

https://dspace.org/

Schwarz type domain decomposition methods for spectral element discretizations.

Pahl, Shannon S.

1994
https://hdl.handle.net/10539/27920

Downloaded from a DSpace repository. https://dspace.org/

Schwarz Type Domain Decomposition Methods for Spectral Element Discretizations

Shannon S. Pahl
University of the Witwatersrand

December 1993

Degree awarded with distinction on 30 June 1994.

A research report submitted to the Faculty of Science in partial fulfillment of the requirements for the degree of Master of Science at the University of the Witwatersrand, Johannesburg.

Abstract

Most of the theory for domain decomposition methods of Schwarz type has been set in the framework of the h and p-version finite element method. In this study, Schwarz methods are formulated for both the conforming and nonconforming spectral element discretizations applied to linear scalar self adjoint second order elliptic problems in two dimensions.

An overlapping additive Schwarz method for the conforming spectral element discretization is formulated. However, unlike p-finite element discretizations, a minimum overlap strategy common to h-finite element discretizations can be accommodated. Computational results indicate that the convergence rate of the overlapping Schwarz method is similar for the corresponding method defined for the h-finite element discretization. It is also shown that the minimum overlap additive Schwarz method results in significantly fewer floating point operations than finite element preconditioning for the spectral element method.

Iterative substructuring and Neumann-Neumann methods are also formulated for the conforming spectral element discretization. These methods demonstrate greater robustness when increasing the degree p of the elements than the minimum overlap additive Schwarz method. Computationally, convergence rates are also similar for the corresponding methods defined for the h-finite element discretization. In addition, an efficient interface preconditioner for iterative substructuring methods is employed which does not degrade the performance of the method for the problems considered.

The extension of Schwarz methods to the nonconforming spectral element discretization is not as straightforward as for the conforming discretization. A number of strategies to overcome these difficulties are investigated.

Declaration

I declare that this research report is my own, unaided work. It is being submitted for the degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.

Shannon S. Pahl

Seventeenth day of February, 1994.

To those who have consistently supported and

encouraged me during my studies

Acknowledgements

I would like to gratefully acknowledge

my supervisor, Prof. C.J. Wright, for his contribution towards making this study possible;

the Foundation for Research and Development and the University of the Witwatersrand for the partial funding of my studies.

Contents

1 General Introduction 1

1.1 A brief taxonomy of domain decomposition methods 1

1.2 Presentation of the research report 5

2 Spectral Element Methods 6

2.1 Elliptic equations: the conforming case 6

2.1.1 Formulation 6

2.1.2 Numerical properties 9

2.1.3 Solution techniques 12

2.2 Stokes equations 18

2.2.1 Formulation 18

2.2.2 Computational aspects 19

2.3 Elliptic equations: the nonconforming case 20

2.3.1 Formulation 20

2.3.2 Computational aspects 22

3 Schwarz Methods 25

3.1 Basic concepts 25

3.2 Abstract Schwarz methods 27

3.2.1 Additive Schwarz methods 28

3.2.2 Multiplicative Schwarz methods 29

3.2.3 Hybrid Schwarz methods 30

3.3 Instances of global Schwarz methods 31

3.3.1 Schwarz methods for the h-finite element discretization 31

3.3.2 Schwarz methods for the p-finite element discretization 33

3.3.3 Schwarz methods for the spectral element discretization 34

3.4 Instances of boundary Schwarz methods 35

3.4.1 Iterative substructuring methods 38

3.4.2 Neumann-Neumann methods 42

3.5 Further instances of Schwarz methods 46

4 Numerical Experiments **48**
4.1 Numerical experiments with global Schwarz methods 48
4.2 Numerical experiments with boundary Schwarz methods 54

5 Nonconforming Spectral Element Solution Techniques **59**
5.1 Introduction 59
5.2 Solution techniques 59

6 Closure **64**
6.1 Summary of the study 64
6.2 Conclusions 66
6.3 Recommendations for further research 68

A Program Listing **69**

References **74**

Chapter 1

General Introduction

1.1 A brief taxonomy of domain decomposition methods

The recent advent of commercial parallel computers has given impetus to the design and testing of algorithms for parallel computing. Although an initial approach is to revise well established algorithms to suit these architectures, the transition from serial to parallel computing also induces new methodologies which produce algorithms that are inherently parallel.

Domain decomposition (DD) methods for the solution of partial differential equations (PDE's) is the evolution of a new methodology of inherently parallel algorithms. A DD method is founded upon the concept of reformulating a given problem assuming that the computational domain is decomposed into a collection of smaller subdomains. There is a large variety of methods which employ this general philosophy as reflected in the literature and a brief account of the major developments are outlined in this section. In addition to the development of parallel algorithms, DD concepts are also useful in the following sense.

- The original complex domain may be decomposable into subdomains of more regular shape, thus extending the class of problems solvable by fast solvers and other regular domain discretizations.
- Domain decomposition provides a means to reduce the sequential computational complexity of a problem. In general, the complexity of a problem grows more than linearly with its size. The partitioned problem can yield a computationally faster solution provided the solutions to the subproblems can be efficiently combined to form the solution of the original problem.
- Often in large scale problems, the resolution of local phenomena determines the need for local grid refinement. A finer mesh in sub-regions of the domain may be needed where the solution contains large gradients. A partitioned domain provides a natural and efficient means of implementing local grid refinement strategies.
- Domain decomposition can easily accommodate different solution techniques in sub-regions of the computational domain. This facility to capture the behaviour of the physical solution

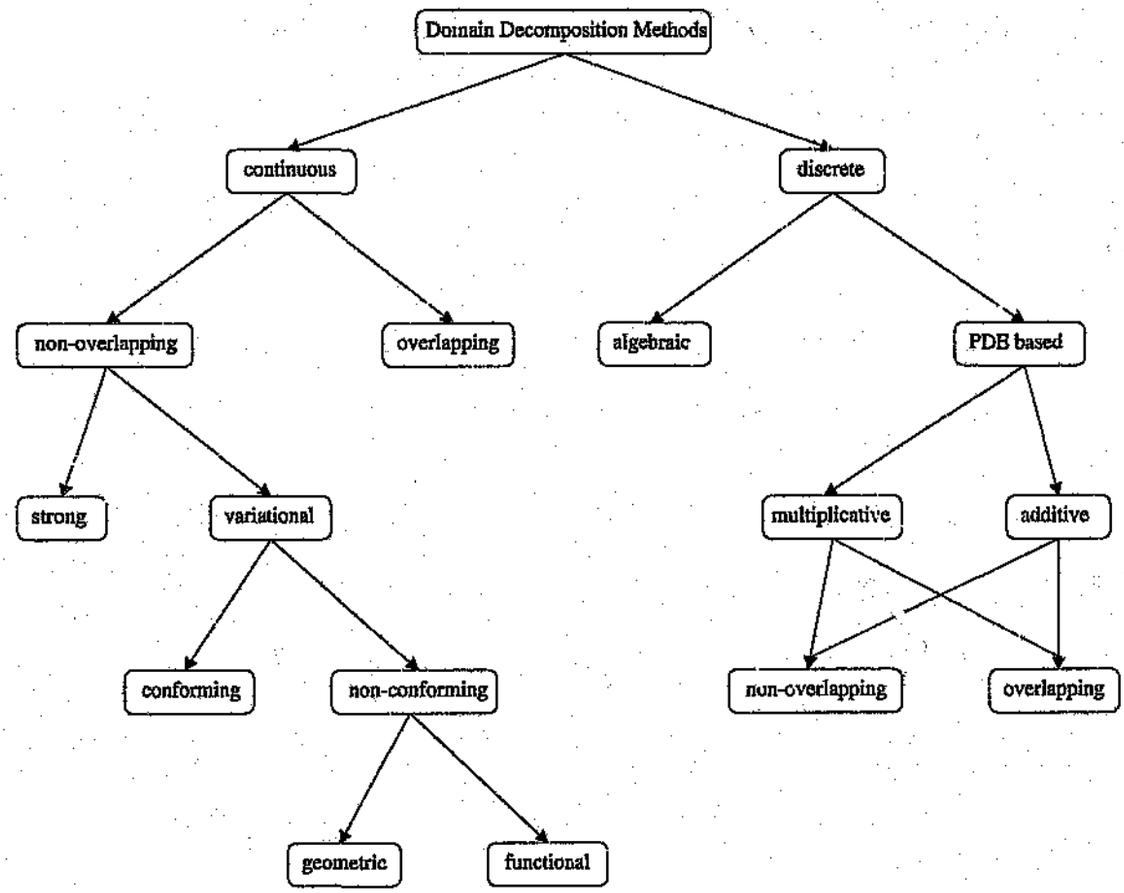


Figure 1.1: A partial taxonomy of domain decomposition methods.

in different sub-regions can be extended to combined different types of PDE's in each sub-region.

Essentially, the basic concept of DD can be applied before and/or after the discretization of the PDE. A partial classification of DD methods is given in figure 1.1. The field of DD is rapidly evolving with seven international conferences having being held in the past seven years [67, 32, 33, 68, 34]. As such, a strict classification may not always be appropriate owing to the generalization and unification of the theory. A brief discussion of the classification depicted in figure 1.1 is given below.

Continuous forms of DD have been primarily motivated by the need to extend regular domain discretizations, such as spectral methods, to more general geometries, and the need to couple different types of PDE's into one model. Continuous forms of DD employing an *overlapping* decomposition of the domain are more commonly known as Schwarz methods. A simple geometric decomposition is depicted in figure 1.2 (a). The solution of the PDE on Ω_1 is computed using zero Dirichlet boundary data along edge Γ_1. The value of this solution along Γ_2 is then used as

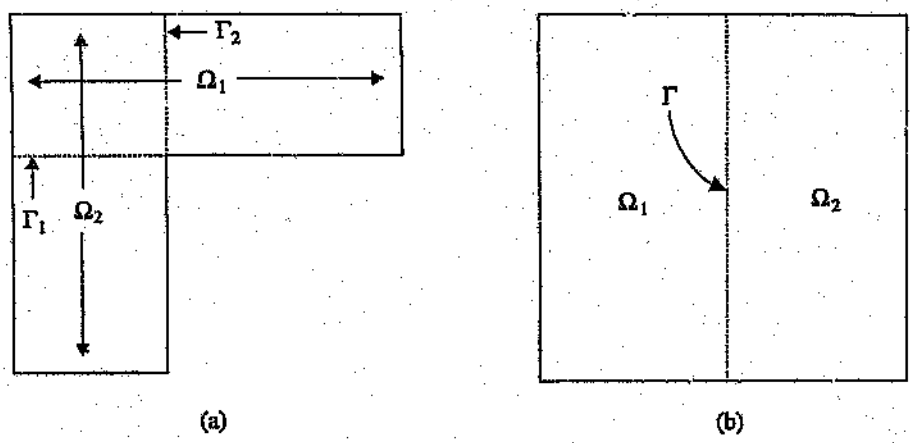


Figure 1.1: An example of an overlapping and non-overlapping decomposition.

boundary data for the problem restricted to Ω_2. The process is repeated until a user specified convergence criteria is reached. Developments in the field of Schwarz methods for pseudo-spectral approximations have been investigated in [29, 129, 99], however, Schwarz methods have gained more popularity in discrete DD methods.

A *strong* or patching form of DD relies on the availability of explicit transmission conditions between subdomains. For instance, the differential problem

$$Lu = f \quad \text{in } \Omega\,,$$

with suitable boundary conditions, is to be solved on the nonoverlapping decomposition in figure 1.2(b). In general, the above problem is split according to

$$Lu_1 = f \quad \text{in } \Omega_1 \qquad Lu_2 = f \quad \text{in } \Omega_2\,,$$

where u_1 and u_2 are the restriction of u to Ω_1 and Ω_2 respectively. Suitable transmission conditions across Γ are needed to ensure that the solution of the decomposed problem is equivalent to the solution of the original problem. These transmission conditions can be generically expressed as

$$\begin{aligned} \Psi(u_1) &= \Psi(u_2) \quad \text{on} \quad \Gamma' \\ \Phi(u_1) &= \Phi(u_2) \quad \text{on} \quad \Gamma'' \end{aligned}$$

where $\Gamma' \subseteq \Gamma$ and $\Gamma'' \subseteq \Gamma$. For second order elliptic problems, $\Gamma' = \Gamma'' = \Gamma$ and the functions Ψ and Φ express continuity of the solution and flux respectively.

The above decomposed problem generally requires further investigation of numerical solution strategies and this is where discrete forms of DD can be of assistance. The advantages of continuous forms of DD are the wide variety of PDE's and discretizations to which the method can be applied. Further, this approach is particularly useful for heterogenous DD – the coupling of different PDE types in different subregions of the computational domain. Further details can be found in [107, 108, 31, 66, 109, 70]. A multilevel scheme has also been investigated in [110].

Variational DD methods have been investigated in [102, 100, 13, 79, 80, 59]. These methods differ from the strong form of DD mainly in the degree of continuity required at the subdomain interfaces and the manner in which the continuity is imposed. An explicit expression imposing the transmission conditions between subdomains is replaced by a more implicit variational scheme. *Nonconforming* methods are generalizations of earlier *conforming* methods. Nonconforming methods permit approximation properties within subdomains to be chosen more independently than in conforming methods. One approach to variational DD [79] consists of formulating a constrained variational principle where the constraints represent some form of coupling (conforming or nonconforming) between the subdomain solutions. Numerically, these constraints are realised through Lagrange multipliers. An alternate approach is to construct an explicit approximation space which satisfies certain projections for the coupling of solutions [82, 1, 13]. Two types of nonconformity have been considered, namely a *functional* and a *geometric* nonconforming notion [12]. The latter approach results from a nonconforming geometric decomposition and encompasses the functional nonconforming case which results from approximations which are not necessarily continuous across the subdomain boundaries. Further details of this type of DD can be found in [13, 9, 97, 74, 10, 11, 14].

Discrete forms of DD have received more attention than continuous forms and primarily for finite element discretizations. The main focus of this interest has been the need to solve large scale finite element problems on parallel computers.

Algebraic DD methods derive solution techniques for a system of linear algebraic equations based only on the algebraic properties of the matrix and the assumption of an association of unknowns with various regions in the domain. These methods are more generally applicable than those DD methods derived from a knowledge about the PDE (*PDE based* methods), at the expense of robustness. Further examples of algebraic DD methods can be found in [98].

The availability of parallel computers with a large number of processors provides the incentive to use DD methods where at least one subdomain is mapped onto each processor. A large number of subdomains necessarily implies that many subdomains do not have access to the physical boundary data. This can result in a reduction of the convergence rate of an iterative DD method when increasing the number of subdomains. Multilevel DD methods are essential in this respect and are capable of maintaining a convergence rate that is independent of the number of subdomains. The core of PDE based DD methods share this multilevel criteria. Two main classes of PDE based methods are those methods based on either an *additive* or on a *multiplicative* update of the subdomain problems. Additive methods permit each subdomain problem to be updated simultaneously while multiplicative methods involve a number of sequential steps. Either class of method can be defined for subdomains related to an *overlapping* or a *non-overlapping* decomposition of the computational domain. A large number of PDE based DD methods are realised as preconditioners for the iterative solution of the system of equations that arise from the discretization of the PDE. The preconditioners are derived from an exact or approximate solution of the PDE restricted to certain regions of the domain. These regions are defined by a geometrical decomposition of the computational domain.

The distinguishing feature of most methods in either class is the manner in which the multilevel criteria is incorporated. In addition, the multilevel feature within an algorithm for two dimensional problems may not necessarily scale efficiently to three dimensions and further considerations have been developed to overcome this problem.

Considerable unity has been achieved for PDE based DD methods. A general theory described by Dryja and Widlund [55, 57], Cai [24] and Xu [128] has begun to emerge and encompasses a wider variety of numerical methods than solely DD methods, such as multigrid, hierarchical basis methods and multilevel preconditioners.

1.2 Presentation of the research report

The *spectral element method* is an instance of a variational DD method and has been successful in extending the class of problems solvable by pseudo-spectral methods. The *conforming* spectral element method of Patera [102] was initially developed with geometric restrictions similar to the geometric conforming finite element method. Further generality was achieved by the *nonconforming* spectral element method where both geometric and functional nonconformity could be accommodated. However, the success of a discretization method depends in part on the availability of efficient solution techniques for the resulting system of equations. Direct methods were initially the prominent solution technique for spectral element methods. Advances in the understanding of the numerical properties of the conforming spectral element method has led to the introduction of efficient preconditioned iterative solution techniques. However, a solution technique based on discrete DD within the Schwarz framework has not been investigated. This study will investigate the use of Schwarz methods for the spectral element discretizations.

In chapter 2, an overview of the theoretical and numerical properties of the conforming and nonconforming spectral element method is given. The performance of two preconditioned iterative solution techniques for the spectral element method is investigated which forms a basis for the Schwarz methods presented in chapter 3.

Most of the theory for Schwarz methods has been developed in the framework of the h and p-finite element method. In chapter 3, Schwarz methods for the conforming spectral element discretization are formulated.

Documented in chapter 4 are the results of experiments conducted with the Schwarz methods of chapter 3.

The possible application of Schwarz methods to the nonconforming spectral element discretization is briefly discussed in chapter 5.

The distinguishing feature of most methods in either class is the manner in which the multilevel criteria is incorporated. In addition, the multilevel feature within an algorithm for two dimensional problems may not necessarily scale efficiently to three dimensions and further considerations have been developed to overcome this problem.

Considerable unity has been achieved for PDE based DD methods. A general theory described by Dryja and Widlund [55, 57], Cai [24] and Xu [128] has begun to emerge and encompasses a wider variety of numerical methods than solely DD methods, such as multigrid, hierarchical basis methods and multilevel preconditioners.

1.2 Presentation of the research report

The *spectral element method* is an instance of a variational DD method and has been successful in extending the class of problems solvable by pseudo-spectral methods. The *conforming* spectral element method of Patera [102] was initially developed with geometric restrictions similar to the geometric conforming finite element method. Further generality was achieved by the *nonconforming* spectral element method where both geometric and functional nonconformity could be accommodated. However, the success of a discretization method depends in part on the availability of efficient solution techniques for the resulting system of equations. Direct methods were initially the prominent solution technique for spectral element methods. Advances in the understanding of the numerical properties of the conforming spectral element method has led to the introduction of efficient preconditioned iterative solution techniques. However, a solution technique based on discrete DD within the Schwarz framework has not been investigated. This study will investigate the use of Schwarz methods for the spectral element discretizations.

In chapter 2, an overview of the theoretical and numerical properties of the conforming and nonconforming spectral element method is given. The performance of two preconditioned iterative solution techniques for the spectral element method is investigated which forms a basis for the Schwarz methods presented in chapter 3.

Most of the theory for Schwarz methods has been developed in the framework of the h and p-finite element method. In chapter 3, Schwarz methods for the conforming spectral element discretization are formulated.

Documented in chapter 4 are the results of experiments conducted with the Schwarz methods of chapter 3.

The possible application of Schwarz methods to the nonconforming spectral element discretization is briefly discussed in chapter 5.

Chapter 2

Spectral Element Methods

2.1 Elliptic equations: the conforming case

2.1.1 Formulation

Let $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, be an open bounded domain with boundary $\partial\Omega$. A general elliptic boundary value problem with homogeneous Dirichlet boundary values is defined by

$$\left.\begin{array}{rll} Lu = f & \text{in} & \Omega \\ u = 0 & \text{on} & \partial\Omega \end{array}\right\} \tag{2.1}$$

where the operator L has the form

$$Lu = -\sum_{i,j=1}^{d} \frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial u}{\partial x_j}\right) + \sum_{i=1}^{d} b_i \frac{\partial u}{\partial x_i} + \sigma u \,.$$

All the coefficients are assumed to be sufficiently smooth functions on Ω. The matrix of coefficients is assumed to be symmetric and uniformly positive definite for all $x \in \Omega$, and $f \in L^2(\Omega)$. As usual, $L^2(\Omega) = \{u \text{ measurable over } \Omega : \int u^2 \, d\Omega < \infty\}$. This study will concentrate on the above problem for $d = 2$, $b_i = \sigma = 0$ with a sufficiently smooth boundary and/or convex polygonal domain. Most of the work presented can be applied to the more general problem in (2.1) with mixed Neumann and Dirichlet boundary values.

The weak formulation of the problem to be considered is: Find $u \in H_0^1(\Omega)$ such that

$$a(u,v) = f(v) \quad \text{for all } v \in H_0^1(\Omega)\,, \tag{2.2}$$

where

$$\begin{aligned} a(u,v) &= \sum_{i,j=1}^{2} \int a_{ij}\frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_j}\, d\Omega\,, \\ f(v) &= \int fv \, d\Omega\,. \end{aligned}$$

Here $H_0^1(\Omega)$ is the Sobelov space $W_0^{1,2}(\Omega)$ defined by those functions which together with their weak derivatives are contained in $L^2(\Omega)$ and satisfy the boundary conditions in (2.1). It is

well known that since $f(\cdot)$ is continuous and $a(\cdot,\cdot)$ is both continuous and H_0^1 elliptic, the Lax-Milgram theorem ensures that a unique solution of (2.2) is guaranteed to exist, see Ciarlet [42].

The discrete equivalent of (2.2) is obtained by formulating the problem on a finite dimensional subspace of $H_0^1(\Omega)$. We construct the finite dimensional subspace by first decomposing the domain as follows.

Let $\overline{\Omega} = \cup_{i=1}^{N} \overline{\Omega}_i$ be a partitioning of Ω into N curvilinear polygonal open *subdomains* of diameter $\mathcal{O}(H)$ such that $\Omega_i \cap \Omega_j = \emptyset$ whenever $i \neq j$. Further, we assume that a subdomain boundary $(\partial\Omega_k)$ comprises four *vertices* and *edges/interfaces* such that $\overline{\Omega}_i$ and $\overline{\Omega}_j$ have either an entire interface or single vertex common to both subdomains whenever $\overline{\Omega}_i \cap \overline{\Omega}_j \neq \emptyset$. This restriction ensures a geometrically conforming decomposition of the domain.

Let $\widehat{\Omega} = (-1,1) \times (-1,1)$ be the *reference* subdomain. The space of polynomial functions of degree p in each variable (ε,η) on $\widehat{\Omega}$ will be denoted by $\mathbb{P}_p$. Suppose $T_k(\varepsilon,\eta)$ is defined to be the affine and/or isoparametric mapping from $\widehat{\Omega}$ onto Ω_k, then the pth order approximation space on Ω_k is

$$\mathbb{P}_{p,k} = \{v(T_k^{-1}(x,y)), \quad v \in \mathbb{P}_p\} .$$

The space of piecewise polynomial functions on Ω is then defined by

$$\widetilde{V}_{p,h} = \{v \in L^2(\Omega) : \; v|_{\Omega_k} \in \mathbb{P}_{p,k} , \quad k = 1,\ldots,N\} .$$

Clearly, $\widetilde{V}_{p,h}$ is not a subspace of $H_0^1(\Omega)$ and therefore additional constraints need to be imposed along the subdomain boundaries. Specific details for constructing such constraints are in general dependent on the basis chosen for $\mathbb{P}_p$ and will be discussed shortly. Generically, we ensure functional conformity by setting

$$V_{p,h} = \widetilde{V}_{p,h} \cap H_0^1(\Omega) . \tag{2.3}$$

The discrete problem corresponding to (2.2) can be written as: Find $u \in V_{p,h}$ such that

$$a(u,v) = f(v) \quad \text{for all } v \in V_{p,h} . \tag{2.4}$$

Since $V_{p,h} \subset H_0^1(\Omega)$, the existence and uniqueness of the discrete solution of (2.4) can be inferred from the Lax-Milgram theorem.

Suppose the exact solution u^* is such that $u^* \in H^m(\Omega)$ for $m > 1$ and f is an analytic function on Ω. An error estimate for the solution of (2.4) is shown in [6] to be given by

$$||u^* - u||_{H_0^1(\Omega)} \leq C h^{\ell} p^{1-m} ||u^*||_{H^m(\Omega)} , \tag{2.5}$$

where $\ell = \min(m-1, p)$ and C is a constant independent of h, p and u^*. Similar results hold when u^* is only regular within Ω_i [6, 85]. A wide class of problems in fluid and structural mechanics have smooth or piecewise analytic solutions. Therefore, an approximation of the type in (2.4) can converge exponentially to the exact solution u^*.

We now discuss the construction of a basis for $\mathbb{P}_p$. The approximation properties of $V_{p,h}$ are independent of the algebraic form of the basis. The form of the basis can however affect computational considerations such as the condition of the discrete equations, the stability and convergence of the solution process and the ability to efficiently accommodate adaptive meshing strategies. Variants of the p-version of the finite element method (PFEM) and the spectral element method (SEM) emerge from the choice of a basis for $\mathbb{P}_p$, the manner in which the constraint in (2.3) is imposed and the choice of quadrature schemes for (2.4). These aspects for the PFEM have been previously addressed in [5, 7, 46, 50].

A basis for $\mathbb{P}_p$ can be derived from the basis functions typically used in pseudo-spectral methods. The set of basis functions are described in Canuto et al. [30] as a tensor product of the one dimensional basis set. Let $I = [-1,1]$ and ε_j, $j = 0,\ldots,p$ be the Gauss-Lobatto-Legendre quadrature points of I. An order p polynomial basis set on I, $\psi = \{h_i\}_{i=0}^{p}$ such that $h_i(\varepsilon_j) = \delta_{ij}$, is defined by

$$h_i(\varepsilon) = \frac{-(1-\varepsilon^2)L_p'(\varepsilon)}{p(p+1)L_p(\varepsilon_j)(\varepsilon-\varepsilon_j)} ,$$

where $L_p(\varepsilon)$ represents the pth order Legendre polynomial on I. A basis for $\mathbb{P}_p$ is simply expressed as a tensor product of the functions in the set ψ. Bases of different order in each variable can also be accommodated, see Canuto et al. [30] for more details. The degrees of freedom defining $v \in \mathbb{P}_p$, are of Lagrange/nodal type defined at the collocation points. That is, $v \in \mathbb{P}_p$ can be expressed as

$$v = \sum_{i,j=0}^{p} \underline{w}_{ij} h_i(\varepsilon) h_j(\eta) \qquad \text{for } \varepsilon, \eta \in I \tag{2.6.a}$$

or

$$v = \sum_{\ell=1}^{(p+1)^2} \underline{v}_\ell \Phi_\ell(\varepsilon,\eta) \qquad \text{where } \Phi_\ell(\varepsilon,\eta) = h_i(\varepsilon)h_j(\eta) \text{ for some } i \text{ and } j\,. \tag{2.6.b}$$

In order to enforce the constraint in (2.3), if $\gamma = \partial\Omega_i \cap \partial\Omega_j$ such that $\gamma \neq \emptyset$ and $v_i \in \mathbb{P}_{p,i} \cap V_{p,h}$ and $v_j \in \mathbb{P}_{p,j} \cap V_{p,h}$, we require $v_i|_\gamma = v_j|_\gamma$. This is easily accomplished as in the finite element method through direct stiffness summation of the degrees of freedom defining v_i and v_j on γ.

Finally, we make use of the tensor-product Gauss-Lobatto-Legendre quadrature formulae to numerically evaluate the integrals in (2.4). The above choice of basis functions leads to significantly reduced complexity counts for this operation. For $u, v \in \mathbb{P}_p$, the numerical approximation of the integrals in $f(\cdot)$ and $a(\cdot,\cdot)$ on $[-1,1]\times[-1,1]$ is

$$f(v) \cong \sum_{m,n=0}^{p} \rho_m \rho_n v(\varepsilon_m,\eta_n) \mathrm{f}(\varepsilon_m,\eta_n) ,$$

$$a(u,v) \cong \sum_{m,n=0}^{p} \rho_m \rho_n \nabla u(\varepsilon_m,\eta_n)\cdot\nabla v(\varepsilon_m,\eta_n) ,$$

where ρ_n, ρ_m and ε_i, η_j are the one dimensional Gauss-Lobatto-Legendre weights and points respectively.

Maday and Patera [85] show that, given the above basis and quadrature scheme, the convergence of the approximate solution $u \in V_{p,h}$ of (2.4) to the exact solution $u^* \in H_0^m(\Omega)$ is given by

$$\|u^* - u\|_{H^1} \leq Cp^{1-m}\|u^*\|_{H^m} ,$$

when $T_k(\varepsilon, \eta)$ are affine, $a_{ij} = \delta_{ij}$ and f is an analytic function on Ω. Here C is a constant independent of p. This bound includes the effect of numerical integration and compares to the bound in (2.5). Experimental results in [85] indicate that similar convergence behaviour is attainable for more general problems. Computational aspects of the SEM will be discussed in the following two sections.

2.1.2 Numerical properties

Storage and Computational costs

Computational and storage costs associated with subdomain stiffness matrices increase at a faster rate for higher order methods than for lower order finite element methods with the same number of unknowns. Consequently, these operations necessitate efficient implementation in order to maintain the possible advantages of using high order basis functions.

The system of equations $A_i \underline{u}_i = \underline{f}_i$ results from (2.4) restricted to $\mathbb{P}_{p,i}$ after having applied the quadrature rules and inserting the basis functions for $\mathbb{P}_{p,i}$. The system of linear equations corresponding to (2.4) is then obtained through direct stiffness summation of the local contributions. Algebraically, this can be written as follows. Let $\widehat{A} = diag(A_1, \ldots, A_N)$ and $\underline{\widehat{f}} = (\underline{f}_1, \ldots, \underline{f}_N)^T$. Using MATLAB notation, let $\underline{u}(i_r)$ be the subvector of $\underline{u}$ corresponding to $\underline{u}_i$ and let R be the permutation matrix defined by $R(i_r, :)\,\underline{u} = \underline{u}_i$. The system of equations corresponding to (2.4) can be written as

$$R^T \widehat{A} R \underline{u} = R^T \underline{\widehat{f}} . \tag{2.7}$$

Direct solution methods for (2.7) have been investigated previously in [102, 103, 49]. However, for large two and three dimensional problems, iterative methods can provide a computationally cheaper alternative. Iterative methods typically only require matrix vector products with the discretization matrix and therefore an explicit form of $R^T \widehat{A} R$ or even A_i is not required. The process of forming matrix vector products with $R^T \widehat{A} R$ reduces to evaluating matrix vector products with A_i. This operation is described below.

The nature of the basis for $\mathbb{P}_{p,i}$ results in each local stiffness matrix A_i, requiring $\mathcal{O}(p+1)^4$ storage locations. A matrix vector product with A_i therefore requires the same order of multiplications. These operation and storage counts far exceed those of low order finite element methods for the same number of unknowns. Fortunately, the tensor product nature of the basis can be exploited to reduce the operation and storage requirements.

Let K be the local stiffness matrix on the reference subdomain and $\underline{v}$ the vector corresponding

to the degrees of freedom of $v \in \mathbb{P}_p$. In order to compute $K\underline{v}$, $a(v,\cdot)$ in (2.4) must be evaluated for each of the $(p+1)^2$ basis functions. That is, using (2.6.b)

$$(K\underline{v})_\ell = a(v, \Phi_\ell)\,, \qquad \ell = 1, \ldots, (p+1)^2\,. \tag{2.8}$$

Maday and Pater [85] give an $\mathcal{O}(p^3)$ algorithm to compute the entries of $K\underline{v}$ in a componentwise manner (sum factorization). However, we develop an algorithm which involves dense matrix kernels. For exposition purposes only, we assume that the coefficients a_{ij} in $a(\cdot,\cdot)$ are given by a function of the form $g(x,y)\delta_{ij}$.

Let W and K_w be square matrices resulting from a re-shaping of the vectors $\underline{v}$ and $K\underline{v}$ corresponding to the equivalence of (2.6.a) and (2.6.b). Since $\Phi_\ell = h_i(\varepsilon)h_j(\eta)$ for some $i,j \in \{0,\ldots,p\}^2$, (2.8) can be equivalently written as

$$(K_w)_{ij} = \int g(\varepsilon,\eta)\frac{\partial v}{\partial \varepsilon}\frac{\partial (h_i(\varepsilon)h_j(\eta))}{\partial \varepsilon} + g(\varepsilon,\eta)\frac{\partial v}{\partial \eta}\frac{\partial (h_i(\varepsilon)h_j(\eta))}{\partial \eta}\,\mathrm{d}\widehat{\Omega}\,. \tag{2.9}$$

The effects of the map T_k^{-1} are simply reflected in the coefficient $g(\varepsilon,\eta)$ in (2.9).

For the case when $g(\varepsilon,\eta)$ is a constant. Let $\Lambda = diag(\rho_0,\ldots,\rho_p)$ and $D_{ij} = \frac{\partial h_j}{\partial \varepsilon}(\varepsilon_i)$, and that the collocation and quadrature points coincide, a little algebra transforms

$$\begin{aligned} K_w &= g\Lambda W D^T \Lambda D + g D^T \Lambda D W \Lambda \\ &= g\Lambda W H + g H W \Lambda\,. \end{aligned} \tag{2.10}$$

Evaluating (2.10) involves $4(p+1)^3 + 5(p+1)^2 + \mathcal{O}(p)$ multiplications. However, it is possible to compute and store H once, which reduces the operation count to $2(p+1)^3 + 3(p+1)^2 + \mathcal{O}(p)$. Further, the storage requirement for K is easily seen to be $\mathcal{O}(p^2)$.

When $g(\varepsilon,\eta)$ is non-constant, K_w is computed in a slightly different manner. Let $g_{ij} = g(\varepsilon_i,\eta_j)$ and $\Lambda^i = diag(\rho_0 g_{i0},\ldots,\rho_p g_{ip})$. Using similar arguments, (2.9) is equivalent to

$$\begin{aligned} K_w(i,:) &= \rho_{i-1} W(i,:) D^T \Lambda^{i-1} D \;\;, \; i = 1,\ldots,p+1 \\ K_w(:,j) &= \rho_{j-1} D^T \Lambda^{j-1} D W(:,j) + K_w(:,j) \;\;, \; j = 1,\ldots,p+1\,. \end{aligned} \tag{2.11}$$

The operation count for (2.11) is $4(p+1)^3 + 6(p+1)^2$ and the storage requirement for K remains $\mathcal{O}(p^2)$.

The above implicit scheme requires in general $\mathcal{O}(p^2)$ storage locations for A_i and $\mathcal{O}(p^3)$ operations for a matrix vector product with A_i. The local stiffness matrices can be formed explicitly in $\mathcal{O}(p^3)$ operations although the cost of storage and matrix vector products increase to $\mathcal{O}(p^4)$. The computational advantage of the tensor product basis becomes even more substantial in three dimensions. In [85], a componentwise scheme is used to reduce the complexity of the operation and storage counts from $\mathcal{O}(p^6)$ to $\mathcal{O}(p^4)$ and $\mathcal{O}(p^3)$ respectively.

p	4	5	6	7	8	9	10	11	12	13	14	15
$\kappa(A)$	58.2	87.9	124.3	167.7	218.9	278.8	349.1	432.1	530.3	645.4	778.9	931.8

Table 2.1: Condition number of the discretization matrix for $D_x = 1$ and varying p

D_x	1	2	3	4	5	6	7	8
$\kappa(A)$	58.2	139.9	276.0	461.9	696.3	979.1	1310.1	1689.5

Table 2.2: Condition number of the discretization matrix for $p = 4$ and varying D_x

Condition number estimates

An understanding of the growth in the condition number of the discretization matrix is an important aspect for both direct and iterative solution techniques. Numerical experiments to determine the c ition number of the discretization matrix were conducted for the following model problem,

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

where $\Omega = [0,1] \times [0,1]$ and f is chosen such that $u = xy(1-x)(1-y)$. Here Ω is partitioned into $D_x \times D_x$ uniform subdomains.

The condition number of the discretization matrix, $R^T \widehat{A} R$ in (2.7), before imposing the boundary condition $u = 0$ is reported in tables 2.1 and 2.2. Similar experiments have been documented in [5, 105] for the PFEM. Numerical data fitting for the model $\kappa(R^T \widehat{A} R) \propto D_x^{\alpha} p^{\beta}$ reveals that $\alpha = 2$ and $\beta = 2$ for $p \leq 10$, while $\beta \cong 2.5$ for $10 \leq p \leq 15$.

Sparsity structure

The sparsity structure of the local stiffness matrix K is displayed in figure 2.1. The sparsity pattern is identical to that of the pseudo-spectral discretization matrix of the model problem. The bandwidth of this matrix is shown in [48] to be $(p^2 - p)$. The percentage of non-zero entries constituting K is given in table 2.3. Unfortunately, even for various types of matrix re-ordering [58], the associated LU factors are dense. By comparison, a finite element method (FEM) using linear triangular elements was used to generate a discretization matrix of the model problem on $\widehat{\Omega}$ using the same number of unknowns as the SEM on $\widehat{\Omega}$. The $\mathcal{O}(p+1)$ bandwidth of this matrix is clearly illustrated in figure 2.2

p	Non zeros (%)
4	36.0
6	26.5
8	20.9
10	17.4
12	14.8

Table 2.3: Percentage of nonzero entries for a local stiffness matrix

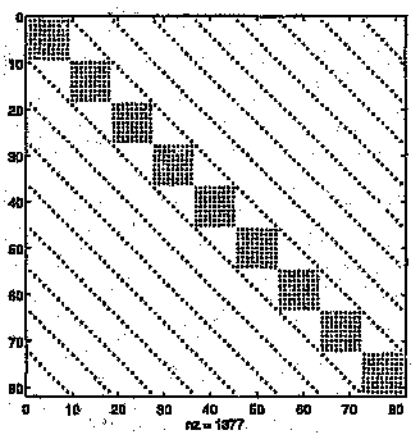


Figure 2.1: Sparsity structure of a local SEM matrix for $p = 8$ and using lexicographic ordering of the unknowns.

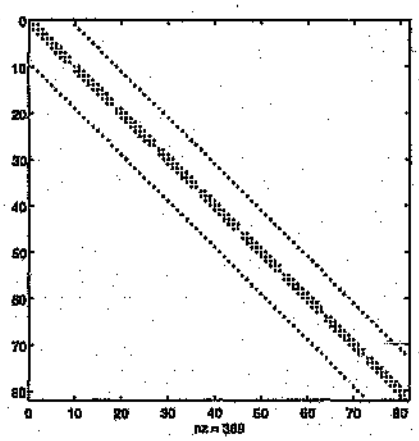


Figure 2.2: Sparsity structure of the finite element discretization matrix with the same number of unknowns and ordering as in figure 2.1.

Convergence tests

In order to appreciate the possible gains when using a high order method for a suitable problem, the model problem is solved using three discretizations: the SEM, PFEM and a FEM using bilinear rectangular elements. The L^∞ error between the exact and computed solution at the collocation points of the SEM is displayed in table 2.4. The advantage of using a high order method over a lower order method for problems which have smooth or piecewise analytic solutions is clearly apparent.

2.1.3 Solution techniques

The system of equations in (2.7) can be equivalently written as

$$A\underline{u} = \underline{f}\,. \tag{2.11}$$

In this section, we investigate the use of preconditioned iterative solution techniques for (2.11). More advanced iterative techniques are presented in chapter 3 and will be applied to the spec-

Number of unknowns	L^∞ error SEM ($D_x = 1$)	PFEM ($D_x = 1$)	FEM(bilinear)
9	1.7 (-3)	6.3 (-4)	5.5 (-2)
16	2.0 (-4)	7.1 (-5)	
25	8.9 (-5)	2.2 (-5)	
36	6.7 (-6)	9.9 (-7)	
49	1.1 (-6)	3.1 (-7)	
64	9.6 (-8)	1.5 (-8)	
81	6.7 (-9)	2.1 (-9)	9.0 (-3)
100	6.7 (-10)	1.1 (-10)	
121	2.2 (-11)	8.8 (-12)	
144	2.7 (-12)	5.0 (-13)	
169	5.6 (-14)	2.6 (-14)	
196	1.4 (-14)	2.9 (-15)	4.1 (-3)

Table 2.4: A test of the convergence of various discretizations for the model problem

tral element discretization based on the concepts developed in this section. Spectral element multigrid solvers for (2.11) have also been investigated in [113, 84].

As was previously indicated, the high order nature of the basis functions used in spectral element discretizations result in less sparse matrices than in h-finite element discretizations. Direct solution methods for equations of the form in (2.11) are robust and particularly suitable when $\underline{f}$ is itself a matrix [69]. However, for large scale problems, the complexity of the solution process is not only governed by floating point operation counts. The potential gain in storage by employing an iterative solution technique for (2.11) over a direct method has already been indicated in section 2.1.2. Further, an iterative solution process is well suited to problems which are steady, contain time dependent coefficients or require adaptive meshes. Iterative methods are also generically well suited to various computer architectures [8, 60, 63].

Preconditioned Conjugate Gradient Methods

The system of equations in (2.11) is symmetric and positive definite which permits the use of the conjugate gradient method as a solution technique. The general problem in (2.1) can also be solved iteratively with an appropriate extension of the conjugate gradient method, notably the GMRES method [115].

The conjugate gradient (CG) method [69] is particularly effective for well conditioned matrices. After n iterations of the CG method, the relative reduction of the error measured in the energy

norm between the exact and computed solution is bounded by

$$2\left(\frac{\sqrt{\kappa(A)}-1}{\sqrt{\kappa(A)}+1}\right)^n, \tag{2.12}$$

where $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$. This implies that the number of CG iterations required for convergence to a specified tolerance is $\mathcal{O}(\sqrt{\kappa(A)})$. Therefore, the number of CG iterations required to solve (2.11) is $\mathcal{O}(\sqrt{D_x^2 p^\beta})$, where $\beta \geq 2$ as indicated in section 2.1.2. This suggests that the convergence rate could be slow even for the model problem. Therefore, a preconditioner for (2.11) is essential. In addition, since the condition number of the discretization matrix resulting from the FEM with the same number of unknowns as the SEM is $\mathcal{O}(D_x^2 p^2)$, the need for a good preconditioner for (2.11) is even more vital than in the h-finite element case. Many types of preconditioning strategies can be followed. Here we use left preconditioning, that is, preconditioning (2.11) results in the *preconditioned system*

$$M^{-1}A\underline{u} = M^{-1}\underline{f},$$

and the CG method is applied to $M^{-1}A$. The preconditioner M^{-1} is chosen such that $M^{-1}A$ is well conditioned and $M\underline{z} = \underline{g}$ is computationally easier to solve than $A\underline{z} = \underline{g}$. Most DD methods can be viewed as a process for constructing the above preconditioned system. The subject of the remaining chapters in this study is the construction of suitable preconditioners based on DD for the high order discretizations presented in this chapter.

Since the matrix A is the realization of the bilinear form $a(\cdot,\cdot)$ in (2.4), the matrix M can also be associated with a bilinear form, $m(\cdot,\cdot)$, defined on $V_{p,h} \times V_{p,h}$. In many instances, it is simpler to define and analyze the bilinear form of the preconditioner. This concept will be used extensively in the next chapter.

The preconditioned conjugate gradient (PCG) method described in [8] will be used in this study. The algorithm is stated in figure 2.3. Note that only the application of A and M^{-1} to a vector is required. Consequently, matrices A and M need not be known explicitly and the techniques of section 2.1.2 can be used for the required operation on A.

Various preconditioners for (2.11) have already been proposed in [85, 60, 49, 61] which have resulted in a substantial improvement in the convergence rate of the PCG solution strategy. We will briefly consider two preconditioners which will serve as comparative references for the DD based preconditioners formulated in chapter 3.

The diagonal preconditioner M_J^{-1}

The classical diagonal preconditioner is typically used as a reference for proposed preconditioners mainly due to its simple and efficient implementation on a wide range of computer architectures. The bilinear form $m_j(\cdot,\cdot)$ for the preconditioner can be thought of as an approximation to the form $a(\cdot,\cdot)$ defined on $V_{p,h} \times V_{p,h}$, and is given by

$$m_j(u_m, u_n) = a(u_m, u_n)\delta_{mn} \quad \text{for all } u_m, u_n \in V_{p,h} .$$

$u^{(0)} = 0$
$r^{(0)} = b$
$z^{(0)} = M^{-1}b$
$p^{(0)} = z^{(0)}$
$\rho^{(0)} = r^{(0)T} z^{(0)}$
repeat until convergence
$q^{(k)} = Ap^{(k)}$
$\alpha^{(k)} = \rho^{(k)} / (p^{(k)T} q^{(k)})$
$u^{(k+1)} = u^{(k)} + u^{(k)} q^{(k)}$
$r^{(k+1)} = r^{(k)} - \alpha^{(k)} q^{(k)}$
$z^{(k+1)} = M^{-1} r^{(k+1)}$
$\rho^{(k+1)} = r^{(k+1)T} z^{(k+1)}$
$\beta^{(k)} = \rho^{(k+1)} / \rho^{(k)}$
$p^{(k+1)} = z^{(k+1)} + \beta^{(k)} p^{(k)}$
end

Figure 2.3: The preconditioned conjugate gradient method

Note that M_J is easily obtained even when A is not known explicitly.

The finite element preconditioner M_F^{-1}

An interesting concept of defining a preconditioner for $a(\cdot,\cdot)$ on a lower order subspace of $V_{p,h}$ has been investigated in [47, 48, 49, 45] for pseudo-spectral methods. Information between the two spaces is transferred using projection and interpolation operators. The extension of these ideas to the SEM is outlined below.

Let G_p designate the set of points $x \in \Omega$ through which the degrees of freedom for the SEM are defined. Denote the finite element space of piecewise linear functions using triangular elements on Ω by V_h and G_h the set of points representing the locations of the Lagrange/nodal degrees of freedom. In order to define the preconditioner, the two sets of grid points must be related by $G_p \subseteq G_h$. An approximation $m_f(\cdot,\cdot)$ to $a(\cdot,\cdot)$ is obtained by defining $a(\cdot,\cdot)$ on $V_h \times V_h$ using the grid G_h. Information between the two spaces is transferred through the restriction and interpolation operators

$$R_p^h : V_{p,h} \to V_h \quad \text{by} \quad (R_p^h u - u)|_x = 0 \quad \text{for all} \quad x \in G_h \,,$$

$$I_h^p : V_h \to V_{p,h} \quad \text{by} \quad (I_h^p v - v)|_x = 0 \quad \text{for all} \quad x \in G_p \,.$$

Let A_h represent the matrix equivalent of $m_f(\cdot,\cdot)$ on $V_h \times V_h$ and $\widetilde{R}_p^h$, $\widetilde{I}_h^p$ represent the analogues

of R_p^h and I_h^p operating on coefficient vectors, then the preconditioner M_F^{-1} can be written as

$$M_F^{-1} = \widetilde{I}_h^p A_h^{-1} \widetilde{R}_p^h .$$

When $G_p \equiv G_h$, the operators $\widetilde{I}_h^p$ and $\widetilde{R}_p^h$ are simply the identity operators and $M_F^{-1} = A_h^{-1}$.

Numerical results

The performance of the above two preconditioners for the model problem of section 2.1.2 was investigated. The PCG method was terminated when the original residual was reduced by five orders of magnitude. The results of these experiments are displayed in tables 2.4 and 2.5.

p/D_x	1	2	4	6	8	10
4	3	5	6	6	6	6
5	3	5	6	6	6	6
6	3	6	6	6	6	6
7	3	6	6	6	6	6
8	3	6	6	6	6	6
9	3	6	6	6	6	6
10	3	6	6	6	6	6

Table 2.4: PCG iteration counts using M_F^{-1}

p/D_x	1	2	4	6	8	10
4	3	9	25	37	49	61
5	3	14	33	49	65	81
6	6	17	41	61	81	101
7	6	23	50	74	98	121
8	10	28	60	87	116	142
9	10	31	70	103	135	166
10	13	35	82	118	154	190

Table 2.5: PCG iteration counts using M_J^{-1}

Given the above computational results, the following conjecture can be made.

Conjecture 1 *The matrix M_F is spectrally equivalent to A, that is, $\kappa(M_F^{-1}A)$ is bounded independently of the number of subdomains and degree polynomials used.*

Conjecture 1 motivates the use of a DD based preconditioner which is spectrally equivalent to M_F, as a preconditioner for (2.11). This concept will be developed further in chapter 3.

A comparison between the two preconditioners cannot be made only by considering the number of PCG iterations. Routine analytical flop counts for the two preconditioning methods can be made. The flop counts include the complexity of the matrix vector product as outlined in section 2.1.2 and the cost of factorizing the finite element matrix using banded methods. Together with the iteration counts listed in tables 2.4 and 2.5, the complexity of each method can be established. Figure 2.4 illustrates a comparison between the complexities of the two methods. For instance, when $D_x = 10$ and $p = 4$, the PCG method with preconditioner M_F^{-1} results in 1.5 times fewer flops than the PCG method with preconditioner M_J^{-1}. Note that this comparison does not

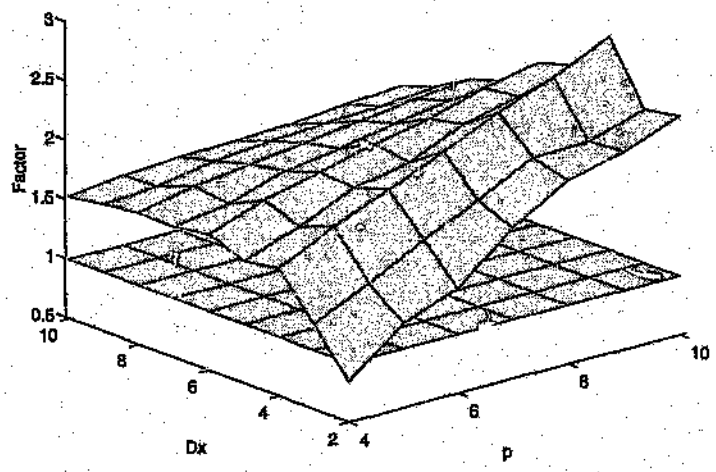


Figure 2.4: A point on the non-constant surface indicates the ratio of flops of the PCG solution of the model problem with preconditioner M_J^{-1} and M_F^{-1} respectively.

include the effects of computer storage requirements, which are considerably increased when using M_F^{-1} as apposed to using M_J^{-1} as a preconditioner.

Finally, we investigate the performance of the finite element preconditioner for multiple right hand sides against a direct solution approach, when $D_x = 1$. The importance of this case is brought about by a number of DD methods which require the solution of independent local problems on each subdomain. Each of the local problems is similar to the original problem for $D_x = 1$ and can be solved either directly or iteratively using the finite element preconditioner. In most cases, the solution of local problems is required to be performed repeatedly.

Once again, flop counts for the direct and iterative solution approach can be made. Deville and Mund [48] demonstrate the advantage of using finite element preconditioning over a direct solution approach when only one solve is required . The results of this comparison for multiple right hand sides is depicted in figure 2.5.

Figure 2.5 indicates that for two dimensional problems, provided sufficient storage is available, a direct solution approach for the local problems may be preferable. This may not be the case for large problems, particularly in three dimensions.

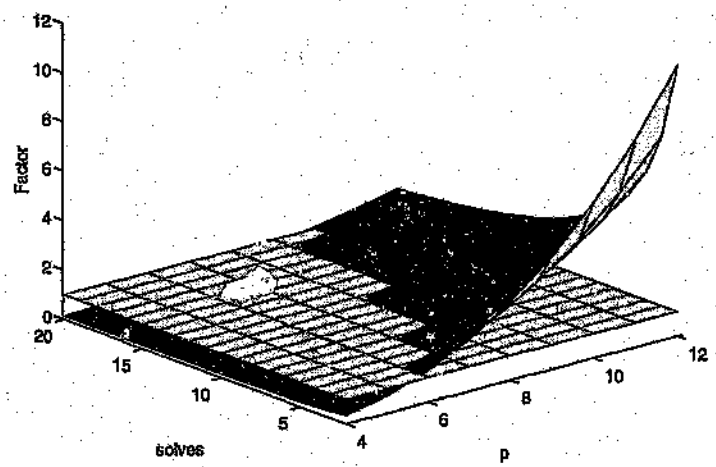


Figure 2.5: A point on the non-constant surface indicates the ratio of flops of a direct method for the solution of the model problem, to the PCG method with preconditioner M_F^{-1}, for multiple right hand sides when $D_x = 1$.

2.2 Stokes equations

The efficient solution of the Stokes equations is often a prerequisite for the numerical solution of the Navier-Stokes equations [86, 75, 61]. In this section, we briefly outline the application of the SEM to the Stokes equations and indicate possible solution techniques for the resulting system of equations.

2.2.1 Formulation

The steady flow of an incompressible fluid for low Reynolds numbers is described by the steady Stokes equations,

$$\begin{aligned} -\Delta u + \nabla p &= f \quad \text{in } \Omega \\ \nabla \cdot u &= 0 \quad \text{in } \Omega\,. \end{aligned} \tag{2.13}$$

For exposition purposes only, nonslip boundary conditions are imposed,

$$u = 0 \qquad \text{on} \qquad \partial\Omega,$$

where $u = (u_1, u_2)$ is the fluid velocity, p is the pressure and $f = (f_1, f_2)$ denotes the body forces. Equation (2.13) is non-dimensionalized to ensure that the fluid viscosity and density are unity.

A well posed weak formulation for (2.13) described in [22, 83] is: Find $(u,p) \in (H_0^1(\Omega))^2 \times L_0^2(\Omega)$ such that

$$\begin{aligned} a(u,v) - b(p,v) &= f(v) \quad \text{for all } v \in (H_0^1(\Omega))^2 \\ -b(q,u) &= 0 \quad \text{for all } q \in L^2(\Omega)\,, \end{aligned} \tag{2.14}$$

where

$$b(p,v) = \int p \nabla \cdot v \, \mathrm{d}\Omega$$

and $a(\cdot,\cdot)$, $f(\cdot)$ are defined in equation (2.2). Here $L_0^2(\Omega) = \{\phi \in L^2(\Omega) : \int \phi \, d\Omega = 0\}$.

The discrete problem is formed by choosing finite dimensional spaces X_h and M_h as subspaces of $(H_0^1(\Omega))^2$ and $L_0^2(\Omega)$ respectively. The spaces X_h and M_h must, however be *compatible* in the sense of the discrete inf-sup condition [22] : there exists a real number $\beta_h > 0$ such that for all $p_h \in M_h$, there exists $v_h \in V_h$ such that

$$\beta_h ||p_h||_{L^2} \leq \frac{b(p_h, v_h)}{a(u_h, v_h)}\,. \tag{2.15}$$

The crucial factor here is the definition of the spaces X_h and M_h satisfying (2.15) with β_h independent of the resolution of the spaces. The compatibility condition ensures that *spurious* pressure modes do not occur, that is, non-zero functions $p_h \in M_h$ do not exist such that $b(p_h, v_h) = 0$ for all $v_h \in X_h$. The existence of such a function implies that if the pair (v_h^*, p_h^*) is a solution to (2.14), then $(v_h^*, p_h + p_h^*)$ is also a solution of (2.14) in which case the convergence of the numerical scheme cannot be guaranteed.

Good surveys outlining the approaches taken to define X_h and M_h in the context of spectral methods can be found in [117, 83]. Further treatment of this problem is given in [72]. Here we adopt the choice of spaces advocated by Maday et al. [83],

$$\begin{aligned} X_h &= (V_{p,h})^2\,, \\ M_h &= \{\phi \in L_0^2(\Omega) : \phi|_{\Omega_k} \in \mathbb{P}_{p-2,k}\}\,. \end{aligned}$$

The SEM for the Stokes equations is defined by constructing a basis for the approximation space and applying appropriate quadrature schemes coinciding with the discussion in section 2.1. The pressure space, however, is expressed in terms of Lagrangian interpolants through Gauss-Legendre quadrature points based on $p-1$ points on $[-1,1]$.

2.2.2 Computational aspects

Following the previous sub-section, discretizing (2.14) results in the symmetric indefinite system of equations,

$$\begin{pmatrix} A_1 & 0 & -D_1^T \\ 0 & A_2 & -D_2^T \\ -D_1 & -D_2 & 0 \end{pmatrix} \begin{pmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \underline{p} \end{pmatrix} = \begin{pmatrix} \underline{f}_1 \\ \underline{f}_2 \\ 0 \end{pmatrix}\,, \tag{2.16}$$

where A_i and D_i correspond to the discrete analogues of $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ on X_h and M_h respectively.

The solution approach taken in [94, 83] is to apply block Gauss elimination to (2.16):

$$\begin{pmatrix} A_1 & 0 & -D_1^T \\ 0 & A_2 & -D_2^T \\ 0 & 0 & S \end{pmatrix} \begin{pmatrix} \underline{u}_1 \\ \underline{u}_2 \\ \underline{p} \end{pmatrix} = \begin{pmatrix} \underline{f}_1 \\ \underline{f}_2 \\ \underline{g} \end{pmatrix},$$

$$\begin{aligned} \text{where} \quad S &= -D_1 A_1^{-1} D_1^T - D_2 A_2^{-1} D_2^T, \\ \underline{g} &= D_1 A_1^{-1} \underline{f}_1 + D_2 A_2^{-1} \underline{f}_2 . \end{aligned}$$

The pressure degrees of freedom are obtained by solving $S\underline{p} = \underline{g}$ followed by back substitution for the values of $\underline{u}_2$ and $\underline{u}_1$. The prohibitive expense of computing S directly motivates the use of an iterative method for the solution of $\underline{p}$. In this case, only matrix vector products with S need be performed. Various preconditioners for S have been proposed and we refer to [85, 83] for further details. The matrix vector product with S can be performed at the expense of solving two elliptic problems. The elliptic solvers described in section 2.1 and chapter 3 can therefore be used as a computational kernel for the solution of the Stokes equations.

An alternative solution strategy for (2.16) is to apply a non-nested iterative method. The development of preconditioners for the iterative solution of the discrete Stokes equations have recently received attention in [45, 124]. The extension of elliptic DD solvers to the Stokes equations has not gained much attention to date and will provide an interesting avenue for future research.

2.3 Elliptic equations: the nonconforming case

In many applications, local mesh refinement is necessary to obtain a solution of guaranteed accuracy with minimum computation. The spectral element discretization places a severe restriction on geometric and functional decomposition. Both h and p refinements can propagate far away from a local region of refinement. Owing to the functional conforming restriction in (2.3), the conforming SEM can only accommodate localized p refinement provided the degrees of freedom along an interface are constrained to yield a continuous solution. This results in a pointwise matching of functions along the interfaces. Similarly, h refinement resulting from a nonconforming geometric decomposition can also be employed. The pointwise matching conditions have been analyzed in [13, 9] and are shown to be inferior, with regard to optimal error estimates, to the integral matching nonconforming methods. The latter approach is outlined in the remaining sections of this chapter.

2.3.1 Formulation

The formulation of the nonconforming SEM has been described in [82, 1]. Given a nonoverlapping decomposition of Ω, the *frame* $F \equiv \cup \partial\Omega_i$ consists of a set of subdomain vertices and edges.

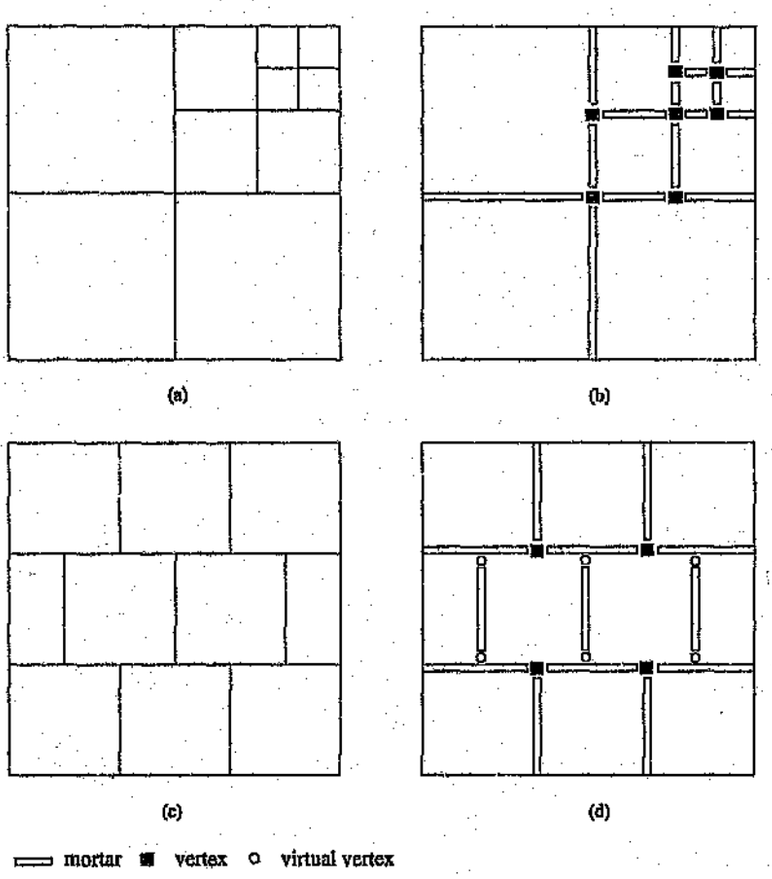


Figure 2.6: Two geometric nonconforming decompositions.

However, we do not require the frame of the partitioning to contain all the subdomain vertices and edges. For instance, two geometric nonconforming decompositions are illustrated in figure 2.6. The set of edges ($\Gamma^{k,j}$ for $j = 1,\ldots,4$; $k = 1,\ldots,N$) which are contained in the frame are denoted by γ^p, $p = 1,\ldots,M$ and are referred to as *mortars* to distinguish between those subdomain edges which are not contained in the frame. The sets of *vertices* and *virtual vertices* are given by $\{\nu^q = \overline{\gamma}^m \cap \overline{\gamma}^n,\ \nu^q \notin \gamma^p\ \ \forall p = 1,\ldots,M\}$ and $\{\overline{\nu}^q = \overline{\gamma}^m \cap \gamma^n\}$ respectively.

Unlike the generic conforming constraint in (2.3), the nonconforming approximation space is not a subset of $H_0^1(\Omega)$. In order to obtain a well posed solution of (2.2), an L^2 projection of the traces of functions on subdomain boundaries are constrained to be equal.

The *mortar space* W is used in the definition of the nonconforming approximation space. In order to define the mortar space, let $W^{k,j} = \{v|_{\Gamma^{k,j}} :\ v \in \mathbb{P}_{p_k,h_k}\}$ and $\overline{W}^{k,j} = \{v|_{\Gamma^{k,j}} :\ v \in \mathbb{P}_{p_k-2,h_k}\}$, then

$$W = \left\{\phi \in C^0(F) :\ \phi|_{\gamma^p} \in W^{k,j},\ p = 1,\ldots,M \text{ for some } k \in \{1,\ldots,N\} \text{ and } j \in \{1,\ldots,4\}\right\}.$$

The nonconforming approximation space is parametrized by the vectors, $\underline{p} = (p_1, p_2, \ldots, p_N)$

and $\underline{h} = (h_1, h_2, \ldots, h_N)$, and is given by

$$V_{\underline{p},\underline{h}} = \{v \in L^2(\Omega) \ : \ v|_{\Omega_k} \in \mathbb{P}_{p_k,h_k} \text{ and for some } \phi \in W\,,$$

$$v(\nu^q) = \phi(\nu^q) \quad \forall\ \nu^q \in \partial\Omega_k\,, \tag{2.20.a}$$

$$v(\widetilde{\nu}^q) = \phi(\widetilde{\nu}^q) \quad \forall\ \widetilde{\nu}^q \in \partial\Omega_k\,, \tag{2.20.b}$$

$$< v|_{\Gamma^{k,j}}, \psi > = < \phi, \psi > \quad \forall\ \psi \in \widetilde{W}^{k,j},\ j = 1, \ldots, 4\} \tag{2.20.c}$$

where $< \cdot, \cdot >$ denotes the usual L^2 inner product. Similar error estimates to those given in section 2.1 hold for the nonconforming method [13, 9] for solutions which are piecewise $H^m(\Omega_k)$.

2.3.2 Computational aspects

Equations (2.20.a), (2.20.b) and (2.20.c) constrain the values of the degrees of freedom along subdomain boundaries. The projection in (2.20.c) can be applied by altering the entries of the discretization matrix. This approach has been used in [1] and [74]. The approach advocated in [79, 80] enforces the projection using Lagrange multipliers which results in a symmetric indefinite saddle point problem. However, there have been no comparative studies to determine which approach is computationally preferable for a given problem. A number of computational considerations are outlined below for the former approach.

The local stiffness matrices are generated using the basis functions described in section 2.1.1. The one dimensional counterparts of these basis functions are used to span the functions in $W^{k,j}$,

$$w \in W^{k,j} \Longrightarrow w = \sum_{i=0}^{p} w_i h_i(T_k^{-1}(s)), \quad s \in \Gamma^{k,j}, \quad T_k^{-1}(s) \in [-1, 1]\,.$$

A basis for $\widetilde{W}^{k,j}$ is given by

$$\psi \in \widetilde{W}^{k,j} \Longrightarrow \psi = \sum_{i=1}^{p-1} \psi_i \eta_i(T_k^{-1}(s)), \quad s \in \Gamma^{k,j}, \quad T_k^{-1}(s) \in [-1, 1]\,,$$

$$\text{where} \quad \eta_i(z) = \frac{(-1)^{p-i} L_p'(z)}{\varepsilon_i - z}, \quad z \in [-1, 1]\,.$$

Note that the functions $\eta_i(\varepsilon)$ and $h_i(\varepsilon)$ are zero on the interior collocation points $\varepsilon_j, j \neq i$, while $\eta_i(\varepsilon)$ are non-zero at the end points. Using suitable mappings and Gauss quadrature, the matrix equivalent of the constraints in (2.20.a)-(2.20.c) for edge $\Gamma^{k,j}$ is

$$B^{k,j} \underline{y}^{k,j} = P^{k,j} \underline{w}^{k,j}\,,$$

where $\underline{y}^{k,j}$ is the vector corresponding to the degrees of freedom along edge $\Gamma^{k,j}$ (excluding subdomain vertices) and $\underline{w}^{k,j}$ are the degrees of freedom on the mortars intersecting edge $\Gamma^{k,j}$, see for example figure 2.7. The nature of the basis for $\widetilde{W}^{k,j}$ and the vertex conditions in (2.20.a)

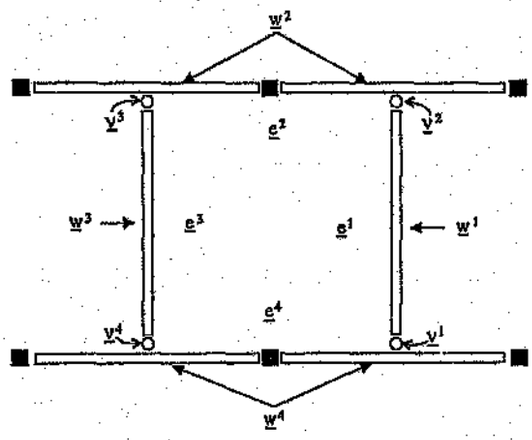


Figure 2.7: A subdomain with associated mortars

and (2.20.b) ensure that the matrix $B^{k,j}$ is diagonal with positive non-zero entries. The degrees of freedom for the function $v|_{\Omega^k}$ along a particular subdomain edge can then be expressed as

$$\underline{v}^{k,j} = Q^{k,j}\underline{w}^{k,j} ,$$

where $Q^{k,j} = (B^{k,j})^{-1}P^{k,j}$. Therefore, the degrees of freedom $\underline{v}_B^k$ corresponding to $v|_{\Omega_k}$ can be written as

$$\underline{v}_B^k = Q^k\underline{w}^k$$

where $\underline{w}^k$ are the degrees of freedom on the mortars intersecting $\partial\Omega_k$ and Q^k is the mortar to subdomain boundary projection.

We now demonstrate the algebraic form of Q^k for a particular subdomain illustrated in figure 2.7. Let $\underline{w}^i, i = 1,\ldots,4$ denote the vectors of degrees of freedom on the mortars and vertices depicted in figure 2.7. Similarly, $\underline{v}^i$ and $\underline{e}^i$, $i = 1,\ldots,4$ are the subdomain vertex and edge degrees of freedom. In particular, $\underline{v}^i$ are the virtual vertices in figure 2.7. The projection Q^k is given by

$$\begin{pmatrix} \underline{v}^1 \\ \underline{v}^2 \\ \underline{v}^3 \\ \underline{v}^4 \\ \underline{e}^1 \\ \underline{e}^2 \\ \underline{e}^3 \\ \underline{e}^4 \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot & C_1^T \\ \cdot & C_1^T & \cdot & \cdot \\ \cdot & C_2^T & \cdot & \cdot \\ \cdot & \cdot & \cdot & C_2^T \\ I & \cdot & \cdot & \cdot \\ \cdot & Q^{k,2} & \cdot & \cdot \\ \cdot & \cdot & I & \cdot \\ \cdot & \cdot & \cdot & Q^{k,4} \end{pmatrix} \begin{pmatrix} \underline{w}^1 \\ \underline{w}^2 \\ \underline{w}^3 \\ \underline{w}^4 \end{pmatrix}, \tag{2.21}$$

The vectors C_1 and C_2 are the vertex pinning conditions in (2.20.a) and (2.20.b).

In order to construct the system of equations for the discrete problem, we reorder A^k according

to the unknowns on the interior and boundary of the subdomain, that is,

$$A^k = \begin{pmatrix} A^k_{II} & A^k_{IB} \\ A^k_{BI} & A^k_{BB} \end{pmatrix}.$$

Define $\widehat{Q}^k$ by

$$\widehat{Q}^k = \begin{pmatrix} I & O \\ O & Q^k \end{pmatrix},$$

then the *modified* local stiffness matrix $\widehat{A}^k$ is obtained from A^k by

$$\widehat{A}^k = \widehat{Q}^{kT} A^k \widehat{Q}^k. \tag{2.22}$$

Let $\widehat{A} = diag(\widehat{A}^1, \ldots, \widehat{A}^N)$, $\underline{\widehat{f}} = (\widehat{Q}^{1^T} f^1, \ldots, \widehat{Q}^{N^T} f^N)^T$ and R be the nonconforming direct stiffness summation operator defined similarly to the conforming operator in section 2.1.2. Then the system of equations for the discrete problem can be written as

$$R^T \widehat{A} R \underline{u} = R^T \underline{\widehat{f}}. \tag{2.23}$$

Various direct and iterative solution techniques for (2.23) have been proposed in [74, 1]. A number of these techniques will be discussed in chapter 5.

Finally, we note that the nonconforming methods considered in this section have been used in conjunction with adaptive schemes (Mavriplis [97]), and extended to the Stokes equations (Maday, Mavriplis and Patera [82]), advection diffusion equations (Le Tallec and Sousa Rodrigues [80]) and linear elastic equations (Le Tallec and Sassi [79]). Further, these concepts also apply to finite element discretizations [80, 13].

Chapter 3

Schwarz Methods

A number of domain decomposition methods can be conveniently described and analyzed as Schwarz methods. These methods have primarily been applied to the h and p-finite element discretizations. In this chapter, the application of Schwarz methods to the spectral element discretization will be outlined. Results of numerical experiments conducted involving the proposed methods are presented in the following chapter.

3.1 Basic concepts

Most DD methods are iterative in nature and make use of iterative methods to accelerate their convergence. A number of basic concepts of iterative methods are stated below to provide a setting for Schwarz methods.

The solution of

$$A\underline{u} = \underline{f} \tag{3.1}$$

is sought, where A is a symmetric positive definite (SPD) matrix defined by the bilinear form $a(\cdot,\cdot)$ in equation (2.2). One class of iterative methods in [71] can be expressed as

$$\underline{u}^{n+1} = \underline{u}^n + M^{-1}(\underline{f} - A\underline{u}^n)\,, \tag{3.2}$$

where M^{-1} is a given SPD approximation to A^{-1}. Various DD methods can be written in this form where the choice of M^{-1} is derived from an exact or approximate solution of the partial differential equation restricted to certain regions of the domain. These regions are defined by a geometrical decomposition of the computational domain. The solution of (3.1) is equivalent to the solution of

$$P\underline{u} = \underline{g}\,, \tag{3.3}$$

where $P = M^{-1}A$ and $\underline{g} = M^{-1}\underline{f}$, and can be solved by a Richardson-type scheme

$$\underline{u}^{n+1} = \underline{u}^n + (\underline{g} - P\underline{u}^n)\,. \tag{3.4}$$

Let $\underline{e}^n = \underline{u}^n - \underline{u}$, then (3.4) and (3.2) can also be written as the *error propagation operator equation*,

$$\underline{e}^{n+1} = (I - P)\underline{e}^n\,. \tag{3.5}$$

The convergence of the linear iterative scheme (3.4) can be accelerated by a CG method [128]. Let $(\cdot,\cdot)_a$ denote the inner product defined by $a(\cdot,\cdot)$ with associated norm $||\cdot||_a$. The convergence factor of (3.4) is given by $\rho_\ell = ||I - P||_a$ where $\rho_\ell < 1$ is necessary for convergence. Since P is SPD with respect to $(\cdot,\cdot)_a$, the CG method can be applied to (3.3) in the inner product of $(\cdot,\cdot)_a$. Let $\kappa(P)$ be the condition number of P, then $\kappa(P) \leq \frac{1+\rho_\ell}{1-\rho_\ell}$, and the convergence factor ρ_c of the CG method applied to (3.3) is

$$\rho_c = \frac{\sqrt{\kappa(P)}-1}{\sqrt{\kappa(P)}+1},$$

which implies that $\rho_c < \rho_\ell$. Therefore, a CG method applied to (3.3) converges at a faster rate than the Richardson scheme in (3.4), or equivalently, a PCG method with preconditioner M^{-1} applied to (3.1) can be used to enhance the convergence rate of (3.2).

We now turn to Schwarz's alternating method [81]. Let Ω be a bounded Lipschitz region in $\mathbb{R}^2$ divided into two subregions, Ω_1 and Ω_2, such that $\Omega_1 \cap \Omega_2 \neq \emptyset$. The solution of the problem,

$$-\Delta u = f \quad \text{in } \Omega$$
$$u = 0 \quad \text{on } \partial\Omega$$

using Schwarz's alternating method, is obtained by computing a sequence of functions u^{n+1} from u^n using the iterative fractional step scheme:

$$\begin{aligned} -\Delta u^{n+\frac{i}{2}} &= f && \text{in } \Omega_i \\ u^{n+\frac{i}{2}} &= 0 && \text{on } \partial\Omega_i \cap \partial\Omega \\ u^{n+\frac{i}{2}} &= u^{n+\frac{i-1}{2}} && \text{on } \partial\Omega_i/(\partial\Omega_i \cap \partial\Omega), \qquad i = 1,2. \end{aligned} \tag{3.6}$$

The function u^0 is taken to be any initialization in $H_0^1(\Omega)$. Recent interest in Schwarz methods has been influenced by the variational formulation of (3.6) as portrayed by Lions [81]. The variational formulation is derived by viewing (3.6) as a process for computing successive corrections to the current solution. For notational convenience in what follows, let $V \quad H_0^1(\Omega)$ and $V_i = H_0^1(\Omega_i)$. The fractional step solution on Ω_i is extended by the previous solution to $\Omega\backslash\Omega_i$ and therefore $u^{n+\frac{i}{2}} - u^{n+\frac{i-1}{2}} \in V_i$. Together with the Dirichlet boundary conditions on the artificial boundaries, (3.6) is equivalent to

$$a(u^{n+\frac{i}{2}} - u, v) = 0 \quad \text{for all } \ v \in V_i,\ u^{n+\frac{i}{2}} - u^{n+\frac{i-1}{2}} \in V_i\,,$$

which can be written as

$$\begin{aligned} a(u^{n+\frac{i}{2}} - u^{n+\frac{i-1}{2}}, v) &= f(v) - a(u^{n+\frac{i-1}{2}}, v) \\ &= a(u - u^{n+\frac{i-1}{2}}, v)\,, \qquad \text{for all } v \in V_i\,, u^{n+\frac{i}{2}} - u^{n+\frac{i-1}{2}} \in V_i\,. \end{aligned} \tag{3.7}$$

Let $\delta_i u^n = u^{n+\frac{i}{2}} - u^{n+\frac{i-1}{2}}$ be the fractional step correction to u^n. Then (3.7) can be written as a sequence of projections of the error $\delta_i u^n$ onto the subspaces [126], that is, (3.7) can be stated

as: Find $\delta_i u^n \in V_i$ such that

$$\begin{aligned} a(\delta_i u^n, v) &= f(v) - a(u^{n+\frac{i-1}{2}}, v) \\ &= a(u - u^{n+\frac{i-1}{2}}, v) \quad \text{for all } v \in V_i\,, \\ u^{n+\frac{i}{2}} &= u^{n+\frac{i-1}{2}} + \delta_i u^n. \end{aligned} \tag{3.8}$$

From (3.8), it is obvious that $\delta_i u^n$ is the projection of $(u - u^{n+\frac{i-1}{2}})$ onto V_i, where the projection operator $P_i : V \to V_i$ is defined by

$$a(P_i w, v) = a(w, v) \quad \text{for all } v \in V_i\,. \tag{3.9}$$

Let $e^n = u^n - u$, then (3.8) and (3.9) implies that

$$\begin{aligned} e^{n+1} &= (I - P_2)(I - P_1)e^n \\ &= E_2 e^n\,. \end{aligned} \tag{3.10}$$

The above ideas are easily generalised to N subdomains by replacing the digit 2 in the above discussion by N.

The *multiplicative Schwarz method* or Schwarz's alternating method, is derived by comparing (3.5) with (3.10) to obtain the *Schwarz operator* $P = I - E_2$. The preconditioned system is solved iteratively using a scheme of the form in (3.4) or by using a PCG method as described earlier. The error propagation operator in (3.10) represents a polynomial operator of degree N in the operators, P_i. Therefore, the number of sequential steps for one application of P to a function is proportional to the number of subdomains. A polynomial error propagation operator of degree one reduces the number of sequential steps and would therefore be ideal from a parallel processing viewpoint. This is easily achieved by retaining only the first order terms in (3.10), namely

$$e^{n+1} = (I - P)e^n\,, \quad P = P_1 + P_2\,. \tag{3.11}$$

The *additive Schwarz method* is obtained by comparing (3.5) with (3.11) and applying an iterative method to the resulting preconditioned system. The above concepts have been generalized by Dryja and Widlund [55, 127, 54] to form a theory known as abstract Schwarz methods.

3.2 Abstract Schwarz methods

In this section, three Schwarz *templates* are specified, namely abstract additive, multiplicative and hybrid Schwarz methods. The specification of each template is in its most general form. An instance of a template is obtained by specifying the following five general quantities.

- An approximation space, V.
- The forms $a(\cdot,\cdot)$ and $f(\cdot)$ defined on the space V.

- A certain decomposition of the space V.
- A set of bilinear forms $b_i(\cdot,\cdot)$ signifying local and coarse problems.
- A collection of projection operators T_i defining the Schwarz operator.

A template specifies the general structure of a Schwarz method while an instance of a template stipulates algorithmic detail for a particular method. The components of each template are discussed first.

3.2.1 Additive Schwarz methods

Let V be a finite dimensional Hilbert space. Consider the abstract variational problem given by: Find $u^* \in V$ such that

$$a(u^*, v) = f(v) \quad \text{for all } v \in V\,. \tag{3.12}$$

The bilinear form $a(\cdot,\cdot)$ must be symmetric, bounded and coercive implying that $a(\cdot,\cdot)$ is an inner product on V. The linear functional $f(v)$ is assumed to be continuous. The space V is decomposed into $N+1$ subspaces such that

$$V = V_0 + V_1 + \cdots + V_N\,.$$

Let $b_i(\cdot,\cdot)$ be a symmetric positive definite bilinear form approximating $a(\cdot,\cdot)$ on $V_i \times V_i$ and define $T_i : V \to V_i$ to be the operators given by

$$b_i(T_i u, v) = a(u, v) \quad \text{for all } v \in V_i,\ u \in V,\ \ i = 0, \ldots, N\,. \tag{3.13}$$

When $b_i(\cdot,\cdot) = a(\cdot,\cdot)$, the operators T_i define orthogonal projections from V onto V_i with respect to the inner product, $(\cdot,\cdot)_a$. The *Schwarz operator* T_{as} is defined by

$$T_{as} = T_0 + T_1 + \cdots + T_N\,,$$

from which $g = \sum_{i=0}^{N} T_i u^*$ is computed by solving the problems

$$b_i(T_i u^*, v) = f(v) \quad \text{for all } v \in V_i\,.$$

Provided T_{as} is invertible, equation (3.12) is equivalent to

$$T_{as} u^* = g \tag{3.14}$$

Theorem 1 *Assume that there exists*

i) a positive constant C_1 such that a decomposition, $v = \sum_{i=0}^{N} v_i$ exists for all $v \in V$ and $v_i \in V_i$ and

$$\sum_{i=0}^{N} b_i(v_i, v_i) \le C_1 a(v, v)\,;$$

ii) a constant C_2 such that

$$a(v,v) \leq C_2 b_i(v,v) \quad \textit{for all } v \in V_i,\ i = 0,\ldots,N;$$

iii) a constant matrix $\mathcal{E}$ with entries ϵ_{ij}, $i,j = 1,\ldots,N$ such that

$$a(v_i,v_j) \leq \epsilon_{ij}\sqrt{a(v_i,v_i)a(v_j,v_j)},\ \ v_i \in V_i,\ v_j \in V_j \qquad i,j = 1,\ldots,N.$$

Then T_{as} is invertible and

$$\kappa(T_{as}) \leq C_1 C_2(\rho(\mathcal{E})+1)\,,$$

where $\rho(\mathcal{E})$ is the spectral radius of $\mathcal{E}$.

Consult Dryja and Widlund [57] for the proof of theorem 1. The constant C_1 is related to the amount of overlap between the subspaces, that is, the dimensions of $V_i \cap V_j$. Increasing the quantity of overlap can decrease C_1 at the expense of having to solve larger local problems in (3.13). The degree to which $b_i(\cdot,\cdot)$ approximates $a(\cdot,\cdot)$ on $V_i \times V_i$ is indicated by the constant C_2. Orthogonality between the subspaces is reflected in $\rho(\mathcal{E})$ and can often be uniformly bounded. The values of these constants will be specified for the algorithms described in section 3.3 and 3.4.

In general, the explicit form of T_{as} is not known, however, the product $T_{as}u$ for a known function u is easily computed using (3.13). Therefore, iterative methods are natural solution techniques for the SPD system in (3.14). Instances of the additive Schwarz operator are given in sections 3.3 and 3.4.

3.2.2 Multiplicative Schwarz methods

The abstract multiplicative Schwarz method [52, 28] is obtained by redefining the Schwarz operator in the abstract additive Schwarz method. The operator T_{ms} is given by

$$T_{ms} = I - E_{N+1}\,,$$

$$E_{N+1} = (I - T_N)(I - T_{N-1})\cdots(I - T_0)\,.$$

Since T_{ms} does not contain any constant term, $g \equiv T_{ms}u^*$ can be computed without the knowledge of u^* and the transformed system corresponding to (3.14) is

$$T_{ms}u^* = g\,. \tag{3.15}$$

Note that the operator T_{ms} is not symmetric. The GMRES method [115] can be used as an iterative solution technique for (3.15). Alternately, a symmetrized version of T_{ms} can be defined by

$$T_{sms} = I - E_{N+1}^T E_{N+1}\,.$$

The following theorem appears in [52].

Theorem 2 *If the assumptions of theorem 1 hold, then T_{sms} is invertible and*

$$\kappa(T_{sms}) \leq \frac{(1+2\omega^2\rho(\mathcal{E})^2)C_1}{(2-\omega)},$$

where $\omega = \max(1, C_2)$.

No general theory has yet been developed to compare the convergence rates of additive and multiplicative Schwarz methods. However, a theoretical investigation of the two subdomain case in [15], and numerical experiments in [25] suggest that multiplicative methods require significantly fewer iterations than additive methods. The bounds in theorem 1 and 2 merely indicate the possible order of convergence of a particular method.

The application of T_{ms} to $u \in V$ ($u_N = T_{ms}u$), is computed recursively by

$$\begin{aligned} u_0 &= T_0 u \\ u_1 &= u_0 + T_1(u - u_0) \\ &\vdots \\ u_N &= u_{N-1} + T_N(u - u_{N-1}) \end{aligned} \tag{3.16}$$

and using the definition of T_i in (3.13). The symmetrized version of the multiplicative Schwarz method is obtained by applying a reverse ordering of the projections in E_{N+1} to u_N using (3.16).

3.2.3 Hybrid Schwarz methods

Cai [24] and Mandel [92] introduced the concept of *hybrid* Schwarz methods. These methods combine the high degree of parallelism of additive methods, with the faster convergence rate of multiplicative methods. Hybrid Schwarz methods can be obtained by redefining the Schwarz operator in section 3.2.1. The Schwarz operators in (3.14) and (3.15) are polynomial operator equations of degree 1 and $N+1$ respectively in the operators T_i. The most general form of a Schwarz operator can be defined by

$$T = poly(T_0, T_1, \ldots, T_N)$$

such that $poly(0, 0, \ldots, 0) = 0$, which ensures that $g = Tu^*$ can be computed without the knowledge of u^*. The hybrid Schwarz operator T is used to obtain the transformed system,

$$Tu^* = g\,. \tag{3.17}$$

Provided T is invertible, (3.17) is equivalent to (3.12).

The Schwarz method of Cai [24] permits a second level operator, T_0, which represents a coarse approximation of the PDE on the computational domain, to be additive with respect to the local problems in T_{ms}. Mandel [92] incorporates T_0 into T_{as} in a multiplicative fashion. The operators of Cai [24], T_{cai}, and Mandel [92], T_{man}, are given by

$$T_{cai} = T_0 + I - (I - T_N)(I - T_{N-1}) \cdots (I - T_1)\,,$$

$$T_{man} = (I - T_0)(T_1 + T_2 + \cdots + T_N)(I - T_0) + T_0 \,.$$

Experimental and theoretical evidence given in [24] and [91] demonstrate the superiority of the bound on the condition number of T_{cat} and T_{man} to T_{as}. Our results in chapter 4 also substantiate these findings.

The application of T_{man} to a function u is computed in three steps,

1. $\tilde{u}_0 = T_0 u$
2. $\tilde{u} = \sum_{i=1}^{N} T_i (u - \tilde{u}_0)$
3. $\widetilde{u} = \tilde{u} - T_0(\tilde{u} - u)$.

Instances of hybrid Schwarz methods are given in section 3.4.2.

3.3 Instances of global Schwarz methods

The classification of *global* and *boundary* Schwarz methods originates from the application of a Schwarz scheme to either the approximation space V or a subspace associated with subdomain boundaries. Boundary Schwarz methods encompass iterative substructuring methods [120, 118, 20, 52] and Neumann-Neumann methods [52, 91]. Global Schwarz methods for three distinct discretizations are considered.

3.3.1 Schwarz methods for the h-finite element discretization

The Dryja-Widlund additive Schwarz method [127] can be obtained by setting V in the additive Schwarz template to be the piecewise linear finite element space V_h, and $a(\cdot,\cdot), f(\cdot)$ as defined in chapter 2. The subspaces are chosen as follows.

The computational domain is decomposed into two levels of triangulation, a fine triangulation $\mathcal{T}^h$ with element diameter of order h and a coarse decomposition of the domain into N subdomains as described in chapter 2 section 2.1. It is assumed that the triangulations are *shape regular*, see Ciarlet [42], and that no edge of Ω_i cuts through an element on $\mathcal{T}^h$. An overlapping decomposition is constructed by extending each subdomain Ω_i to a larger region Ω_i' such that no edge of Ω_i' cuts through an element on $\mathcal{T}^h$. No subdomain is extended beyond $\partial\Omega$. The finite element subspaces are defined by

$$V_i = \{v \in V : v(x) = 0,\ x \in \Omega \backslash \Omega_i'\},\ i = 1, \ldots, N \,.$$

The convergence rate of DD methods will deteriorate rapidly as more subdomains are introduced [125]; information is passed between subdomains only at the local level. Consequently, the need

for a mechanism to transport information globally is required. In two dimensions, the space V_0 is taken to be the *vertex based coarse space* V^H; the discretization of the PDE by piecewise bilinear functions treating subdomains as elements. Although the vertex based coarse space is particularly effective in two dimensions, more advanced schemes are needed for three dimensional problems. Further details in this regard can be found in [119, 118, 52].

The bilinear forms are chosen as

$$b_i(u,v) = a(u,v), \quad u,v \in V_i,\ i = 0,\ldots,N$$

implying that the *local problems* related to $b_i(\cdot,\cdot)$, $i \neq 0$, are solved exactly. The operators $T_i : V \rightarrow V_i$ are defined by

$$b_i(T_i u, v) = a(u,v) \quad \text{for all } v \in V_i,\ i = 0,\ldots,N\,,$$

which completes the description of the method specified by the additive Schwarz template.

Note that in the notation of section 3.2, $C_2 = 1$. Further, $\rho(\mathcal{E})$ is bounded by the maximum number of subdomains to which a point in the domain can belong. Dryja and Widlund [56] show that C_1 is bounded by $(1 + H/\delta)$, where H represents the maximum subdomain diameter and $\delta = \min(\text{dist}(\Omega_i, \Omega_i'))$. The following theorem is given in the above-mentioned paper.

Theorem 3 *The condition number of the additive Schwarz operator is given by*

$$\kappa(T_{as}) \leq C\left(1 + C\frac{H}{\delta}\right)\,,$$

where C is a constant independent of H, h and δ.

Theorem 3 implies that if the amount of overlap for each subdomain remains a fixed fraction of H, then the condition number of T_{as} remains constant.

A number of solvers for the approximate solution of the local problems can also be accommodated in the definition of $b_i(\cdot,\cdot)$ above, see Smith [122] and Bjørstad and Skogen [17] for further details.

The matrix equivalent of the projections in T_{as} can be described as follows. Let A_i denote the principal submatrix of the discretization matrix A corresponding to the nodal values in Ω_i', and A_H the discretization matrix of (3.12) on the coarse mesh. Since $V_i \subset V$, a function $v_i \in V_i$ is spanned by the basis functions of V. The vector of unknowns $\underline{v_i}$ corresponding to $v_i \in V_i$ is therefore related to $\underline{v}$ corresponding to $v_i \in V$ by $\underline{v_i} = R_i\underline{v}$. The mapping R_i ,$i \neq 0$ therefore represents the restriction of $\underline{v}$ to the nodes in Ω_i' and $R_i^T\underline{v_i}$ is the extension of $\underline{v_i}$ by zero to the nodes in $\Omega\backslash\Omega_i'$. The matrices R_H and R_H^T contain entries corresponding to piecewise linear interpolation and weighted restriction respectively. The operator T_{as} can be written as

$$T_{as} \equiv M_{AS}^{-1}A = R_H^T A_H^{-1} R_H A + R_1^T A_1^{-1} R_1 A + \cdots + R_N^T A_N^{-1} R_N A\,.$$

Theorem 3 implies that the condition number of $M_{AS}^{-1}A$ is bounded independently of H and h provided the amount of overlap remains a fixed fraction of H, that is, M_{AS} is a spectrally

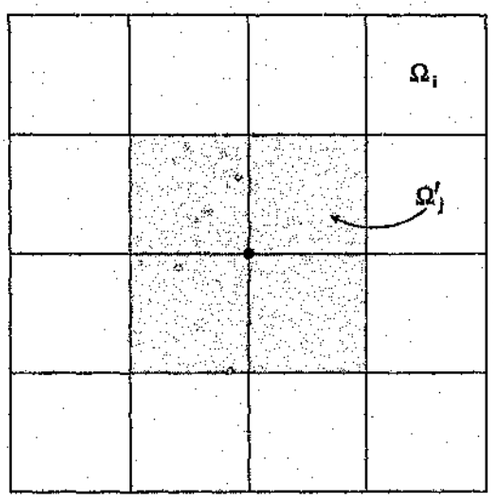


Figure 3.1: The geometric form of an extended subdomain for the PFEM.

equivalent matrix to A. A *minimum overlap strategy* involves using only one mesh point overlap and results in smaller local problems. In this case, theorem 3 indicates that the condition number of the transformed system will depend upon the resolution of the discretization when a fixed number of subdomains are used. However, minimizing the condition number does not necessarily imply minimum computation. Numerical experiments in [25, 23] demonstrate that a minimum overlap strategy can often lead to the least overall amount of computation.

Multiplicative and hybrid Schwarz methods can be specified by simply replacing the above additive operator.

3.3.2 Schwarz methods for the p-finite element discretization

Schwarz methods for high order discretizations have only recently received attention [105]. Constructing an overlapping decomposition for the FEM relied on the assumption that the extended boundaries of the subdomains do not cut through any element; the local spaces must be subspaces of the approximation space. The difficulty with constructing an overlapping decomposition for high order methods is that one element usually forms one subdomain, so that extending a subdomain to include a measure of overlap necessarily implies a relatively large amount of overlap. Pavarino [104] suggested the following additive Schwarz method for the PFEM.

Let V in the additive Schwarz template be the p-finite element space and $a(\cdot,\cdot), f(\cdot)$ as described in chapter 2. An extended subdomain Ω_i' in an overlapping decomposition of Ω consists of the subdomains connected to a single interior vertex, see figure 3.1. The space V is then decomposed into the subspaces,

$$V_i = V_{p,h} \cap H_0^1(\Omega_i') \,, \quad i = 1, \ldots, N$$

where N is the number of interior vertices. The space V_0 is the vertex based coarse space, V^H.

The bilinear forms

$$b_i(u,v) = a(u,v), \quad u,v \in V_i, \quad i = 0,\ldots,N$$

together with the operators, $T_i : V \rightarrow V_i$ defined by

$$a(T_i u, v) = a(u,v), \quad \text{for all } v \in V_i,\ i = 0,\ldots,N$$

complete the description of an additive Schwarz method for the PFEM. The following theorem appears in [104].

Theorem 4 *The condition number of the additive Schwarz operator is bounded by a constant independent of p and H.*

The availability of approximations to $a(\cdot,\cdot)$ in the definition of $b_i(\cdot,\cdot)$ above is more limited for the PFEM than for the FEM. Further, the notion of a minimum overlap strategy cannot be employed to reduce the size of the local problems. The next sub-section demonstrates how both of these problems can be overcome for the SEM.

3.3.3 Schwarz methods for the spectral element discretization

One possible approach to define a Schwarz method for the spectral element discretization has been indicated in the previous sub-section. However, the nodal type basis of the spectral element space affords another Schwarz method unique to the spectral element discretization.

The spectral equivalence of the finite element and spectral element discretization matrices has been demonstrated in chapter 2. We make use of conjecture 1 in section 2.1.3 and the fact that M_{AS} is spectrally equivalent to the finite element discretization matrix (section 3.3.1) to deduce that M_{AS}^{-1} defined on the grid G_p should be a good preconditioner for the spectral element discretization matrix.

Note that with this strategy, a minimum overlap Schwarz method can be defined for the spectral element discretization, a concept which does not carry over directly to the Schwarz method of section 3.3.2. Further, the Schwarz preconditioner is defined on a lower order subspace of the approximation space.

The above strategy can be described as an instance of an additive Schwarz template. We will subsequently refer to the *spectral element space* or simply $V_{p,h}$, as the space $V_{p,h}$ spanned by the basis functions described in chapter 2 section 2.1.1. Let V be the spectral element space and $a(\cdot,\cdot), f(\cdot)$ as described in chapter 2. The subspaces V_i, $i \neq 0$, are the spaces spanned by the basis functions associated with the nodal degrees of freedom within an extended subdomain Ω_i' of section 3.3.1. The space V_0 is taken to be the vertex based coarse space, V^H.

The local bilinear forms are chosen as

$$b_i(u,v) = a(R_p^h u, R_p^h v)\,, \quad u,v \in V_i,\ i \neq 0\,,$$

where $R_p^h : V_{p,h} \to V_h$ by $(R_p^h u - u)|_x = 0$ for $x \in G_p$, and $b_0(\cdot,\cdot) = a(\cdot,\cdot)$ on $V^H \times V^H$. The operators $T_i : V \to V_i$ are defined by

$$b_i(T_i u, v) = a(u,v) \quad \text{for all } v \in V_i,\ i = 0,\ldots,N$$

which completes the description of an additive Schwarz method for the spectral element discretization.

Conjecture 1 *The condition number of the additive Schwarz operator satisfies the estimate*

$$\kappa(T_{as}) \leq C\left(1 + C\frac{H}{\delta}\right)\,,$$

where C is a constant independent of H, p and δ.

Note that it is possible to further approximate the forms $b_i(\cdot,\cdot)$ above using those approximate solvers valid for the local problems in the FEM. The matrix form of the Schwarz operator is similar to that given in section 3.3.1 with the matrix A replaced by the spectral element discretization matrix. Computational results are given in chapter 4.

3.4 Instances of boundary Schwarz methods

Let V be a finite dimensional Hilbert space and consider the variational problem: Find $u \in V$ such that

$$a(u,v) = f(v) \quad \text{for all } v \in V\,. \tag{3.18}$$

Introducing a basis for V, (3.18) can be stated algebraically as: Find a vector $\underline{u}$ such that

$$A\underline{u} = \underline{f}\,. \tag{3.19}$$

Given a conforming partition of Ω , we can divide $\underline{u}$ into subvectors $\underline{u}_I$ and $\underline{u}_B$ containing the degrees of freedom (dof) associated with all the interiors and boundaries of the subdomains. The vector $\underline{u}_I$ is also assumed to be partitioned into blocks with each block corresponding to the unknowns interior to a subdomain. The algebraic problem in (3.19) can be written as

$$\begin{pmatrix} A_{II} & A_{IB} \\ A_{BI} & A_{BB} \end{pmatrix} \begin{pmatrix} \underline{u}_I \\ \underline{u}_B \end{pmatrix} = \begin{pmatrix} \underline{f}_I \\ \underline{f}_B \end{pmatrix}\,,$$

where $\underline{u} = (\underline{u}_I, \underline{u}_B)^T$ and $\underline{f} = (\underline{f}_I, \underline{f}_B)^T$. Since each block of unknowns in $\underline{u}_I$ is associated with only one subdomain, A_{II} is block diagonal and therefore each block of unknowns in $\underline{u}_I$ can be eliminated independently. The solution approach consists of three stages.

1. Compute $\begin{pmatrix} \underline{u}_I^p \\ \underline{u}_B^p \end{pmatrix} = \begin{pmatrix} A_{II}^{-1}\underline{f}_I \\ 0 \end{pmatrix}$

   Since A_{II} is block diagonal, each block of unknowns in $\underline{u}_I^p$ can be computed independently.

2. Solve $\begin{pmatrix} A_{II} & A_{IB} \\ A_{BI} & A_{BB} \end{pmatrix} \begin{pmatrix} \underline{u}_I^H \\ \underline{u}_B^H \end{pmatrix} = \begin{pmatrix} 0 \\ \underline{g} \end{pmatrix}$,

   where $\underline{g} = \underline{f}_B - A_{IB}\underline{u}_I^p$. This operation can be performed in two steps,

   2.1 $S\underline{u}_B^H = \underline{g}$

   2.2 $A_{II}\underline{u}_I^H = -A_{IB}\underline{u}_B^H$

   where $S = A_{BB} - A_{BI}A_{II}^{-1}A_{IB}$.

3. Set $\begin{pmatrix} \underline{u}_I \\ \underline{u}_B \end{pmatrix} = \begin{pmatrix} \underline{u}_I^p \\ \underline{u}_B^p \end{pmatrix} + \begin{pmatrix} \underline{u}_I^H \\ \underline{u}_B^H \end{pmatrix}$.

Steps 1 and 2.2 both require one solve with a block diagonal matrix. Step 2.1 constitutes the only computation not obvious parallelizable. We first outline the construction of S before discussing solution strategies.

Let A^i denote the contribution to A from Ω_i and $\underline{u}^i$ the degrees of freedom defined on $\overline{\Omega}_i$. Vector $\underline{u}^i$ is extracted from $\underline{u}$ using the permutation matrix N^i defined by $\underline{u}_i = N^{iT}\underline{u}$. Then A can be constructed by the method of subassembly,

$$A = \sum_{i=1}^{N} N^i A^i N^{iT} .$$

Ordering the local matrices A^i according to the partition $\underline{u}^i = (\underline{u}_I^i, \underline{u}_B^i)^T$, the contribution to (3.19) from subdomain Ω_i is

$$\begin{pmatrix} A_{II}^i & A_{IB}^i \\ A_{BI}^i & A_{BB}^i \end{pmatrix} \begin{pmatrix} \underline{u}_I^i \\ \underline{u}_B^i \end{pmatrix} = \begin{pmatrix} \underline{f}_I^i \\ \underline{f}_B^i \end{pmatrix} .$$

Matrix A_{II} is block diagonal with blocks corresponding to A_{II}^i. The solution of the problems in steps 1 and 2.2 correspond to the solution of the independent problems related to $(A_{II}^i)^{-1}$.

The contribution to S and $\underline{g}$ from Ω_i is clearly

$$\begin{aligned} S^i &= A_{BB}^i - A_{BI}^i (A_{II}^i)^{-1} A_{IB}^i , \\ \underline{g}^i &= \underline{f}_B^i - A_{IB}^i \underline{u}_I^{p,i} , \end{aligned}$$

where $\underline{u}_I^{p,i} = (A_{II}^i)^{-1}\underline{f}_I^i$, and therefore ordering N^i according to $N^i = (N_I^i\, N_B^i)$, we have

$$\begin{aligned} S &= \sum_{i=1}^{N} N_B^i S^i N_B^{iT} , \\ \underline{g} &= \sum_{i=1}^{N} N_B^i \underline{g}^i N_B^{iT} . \end{aligned}$$

Note that the product of a vector $\underline{y}$ with S can be performed at the expense of one solve with each of the matrices A_{II}^i, that is,

$$S\underline{y} = \sum_{i=1}^{N} N_B^i (A_{BB}^i - A_{BI}^i (A_{II}^i)^{-1} A_{IB}^i) \underline{y}_i, \quad \underline{y}_i = N_B^{iT} \underline{y}. \tag{3.20}$$

In general, the construction of S requires as many solutions of problems involving A_{II}^i as the number of dof on $\partial\Omega_i$. Therefore, a preconditioned iterative solution of the problem in step 2.1 would be competitive with a direct method if the number of iterations remain less than the number of dof on a subdomain boundary. In addition, the independent problems may be solved approximately. A number of authors have proposed various schemes to accommodate approximate solvers, see [122, 52, 20, 56, 26, 41]. Here we discuss one approach advocated by Dryja, Smith and Widlund [52].

The matrix A^{-1} can be factorized as

$$A^{-1} = \begin{pmatrix} I & -A_{II}^{-1} A_{IB} \\ 0 & I \end{pmatrix} \begin{pmatrix} A_{II}^{-1} & 0 \\ 0 & S^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -A_{BI} A_{II}^{-1} & I \end{pmatrix}.$$

If B_{II}^{-1} is a good preconditioner for A_{II} and M^{-1} is the preconditioner for S, then a preconditioner for A is given by

$$B^{-1} = \begin{pmatrix} I & -B_{II}^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} B_{II}^{-1} & 0 \\ 0 & M^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -A_{BI} B_{II}^{-1} & I \end{pmatrix}.$$

Each application of the preconditioner, $B^{-1}\underline{y}$, requires the action of the block diagonal matrix $(B_{II}^i)^{-1}$ twice, and one application of the Schur complement preconditioner. We will restrict the discussion in the remaining sections to the construction of the Schur preconditioner based on Schwarz methods. Therefore, the above ideas are required in the functional notation of [20, 57].

The approximation space is decomposed into a direct sum of subspaces,

$$V = V^{ha} \oplus (V_1 \oplus \cdots \oplus V_N),$$

where $V_i = V \cap H_0^1(\Omega_i)$, such that for $u \in V$,

$$a(u^{p,i}, v) = a(u, v) \quad \text{for all } v \in V_i,\ u^{p,i} \in V_i,$$

and as a consequence

$$a(u^H, v) = 0 \quad \text{for all } v \in V_i,\ u^H \in V^{ha} \text{ with } u^H = u \text{ on } \partial\Omega_i\,. \tag{3.21}$$

Applying this decomposition, (3.18) can be written as: Find $u \in V$, $u = u^H + u^{p,1} + \cdots + u^{p,N}$, such that

$$a(u^{p,i}, v) = f(v) \qquad \text{for all } v \in V_i\,, \tag{3.22.a}$$

$$\left.\begin{aligned} a(u^H, v) &= 0 && \text{for all } v \in V_i\,, \\ a(u^H, v) &= f(v) - \textstyle\sum_{j=1}^{N} a\left(u^{p,j}, v\right) && \text{for all } v \in V^{ha}\,. \end{aligned}\right\} \tag{3.22.b}$$

The computation of the *perpendicular* components of the solution $u^{p,i}$ in (3.22.a) is precisely step 1 above and constitutes the solution of a homogeneous Dirichlet boundary value problem on each subdomain. Similarly, (3.20) and step 2.2 also comprise independent Dirichlet problems on each subdomain. The space V^{ha} is the space of *discrete harmonic functions* which are functions characterized by the property in (3.21). The algebraic analogue of (3.22.b) is given by step 2 with $u_B^H = u$ on $\partial\Omega_i$. Property (3.21) ensures that the discrete harmonic functions are completely described by their values on $\partial\Omega_i$. Therefore, we can solve (3.22.b) restricted to a subspace of V^{ha}, namely $V(\Gamma) = \{v|_{\partial\Omega_i},\, v \in V^{ha},\, i = 1,\ldots,N\}$ where $\Gamma = \cup\partial\Omega_i \backslash \partial\Omega$, and then extend this solution to the interior of the subdomains, thereby forming u^H. Algebraically, this problem is given by step 2.1 and 2.2.

The problem in step 2.1 can be expressed in variational form. Let

$$s_i(u,v) \equiv \underline{u}_B^{iT} S^i \underline{v}_B^i \quad \text{and} \quad s(u,v) \equiv \underline{u}_B^T S \underline{v}_B \,,$$

where $\underline{u}_B, \underline{v}_B$ and $\underline{u}_B^i, \underline{v}_B^i$ are coefficient vectors corresponding to functions $u, v \in V(\Gamma)$ and $u|_{\partial\Omega_i}, v|_{\partial\Omega_i}$ respectively. Step 2.1 can be written as: Find $u \in V(\Gamma)$ such that

$$s(u,v) = \tilde{f}(v) \quad \text{for all } v \in V(\Gamma)\,. \tag{3.23}$$

The solution of (3.23) can be obtained by using a method of Schwarz type. That is, $a(\cdot,\cdot)$ and $f(\cdot)$ in the Schwarz templates are taken to be $s(\cdot,\cdot)$ and $\tilde{f}(\cdot)$. Instances of these methods are discussed in the following two sub-sections. Recall that Schwarz methods transform (3.23) into the equivalent problem

$$Tu = g\,,$$

which is solved using an iterative method. Therefore only the application of T to a function is required and the knowledge of $s(\cdot,\cdot)$ is only required in the form of matrix vector products as described in (3.20).

Boundary Schwarz methods involve groups of *local problems* related to subdomains and a *coarse problem* which can enhance the convergence rate of the DD method considerably. The next two sub-sections outline two general approaches for constructing the local problems, the construction of the coarse problem, and indicates how these methods can be applied to the spectral element discretization.

3.4.1 Iterative substructuring methods

Suppose the decomposition of Ω consists of E edges where an edge Γ_i is the open segment $\partial\Omega_k \cap \partial\Omega_\ell \neq \emptyset$ for some $k \neq \ell$. The Schur complement matrix S can be re-ordered according to

$$\begin{pmatrix} S_{\Gamma\Gamma} & S_{\Gamma\nu} \\ S_{\nu\Gamma} & S_{\nu\nu} \end{pmatrix} = \begin{pmatrix} S_{\Gamma_{11}} & S_{\Gamma_{12}} & \cdots & S_{\Gamma_{1E}} & S_{\Gamma_{1\nu}} \\ S_{\Gamma_{21}} & S_{\Gamma_{22}} & \cdots & S_{\Gamma_{2E}} & \\ \vdots & & \ddots & \vdots & \vdots \\ S_{\Gamma_{E1}} & \cdots & & S_{\Gamma_{EE}} & \\ S_{\Gamma_{E\nu}} & \cdots & & & S_{\nu\nu} \end{pmatrix}.$$

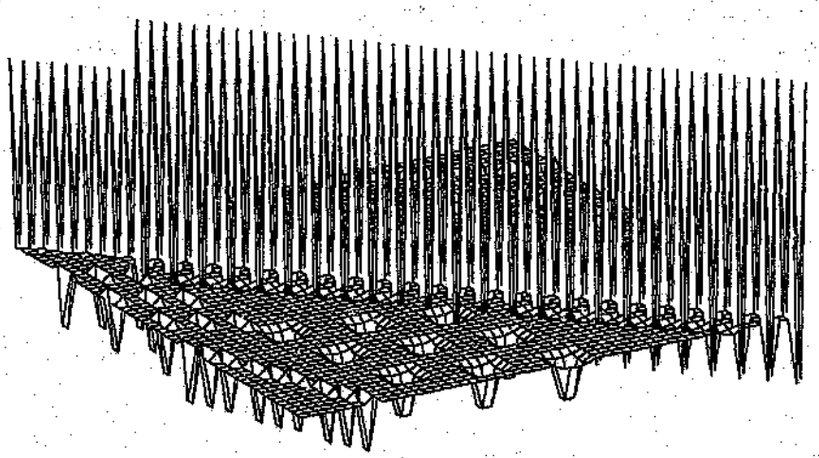


Figure 3.2: A surface plot of the entries of S for $D_x = 4$, $p = 3$ given by the conforming spectral element discretization of the model problem in chapter 2.

Here $S_{\Gamma_{ij}}$ denotes the coupling in S between the dof on edge Γ_i and Γ_j, and $S_{\nu\nu}$ represents the coupling between the vertices. The first approach to construct a preconditioner M^{-1} for S is to eliminate the coupling between the edges and vertices, that is, the preconditioner is derived from the block diagonal partitioning of S,

$$M = \begin{pmatrix} S_{\Gamma_{11}} & & & \\ & S_{\Gamma_{22}} & 0 & \\ & & \ddots & \\ & 0 & & S_{\Gamma_{EE}} \\ & & & & S_{\nu\nu} \end{pmatrix}.$$

This approach is motivated by the fact that for elliptic problems, the absolute magnitude of the entries in S decrease rapidly away from the main diagonal [38]. Figure 3.2 displays a surface plot of the entries of S given by the spectral element discretization of the model problem in chapter 2 section 2.1.2, for $D_x = 4$ and $p = 3$. A similar plot is obtained for h-finite element discretizations. The blocks of M only account for a local coupling of the dof. The submatrix $S_{\nu\nu}$ will be replaced shortly by a component allowing for a global exchange of information.

Let R_{Γ_i} be the permutation matrix such that $R_{\Gamma_i}\underline{u}_B$ returns the entries in $\underline{u}_B$ associated with edge Γ_i. A similar matrix R_ν is introduced for the vertices. In particular, $S_{\Gamma_{ii}} = R_{\Gamma_i} S R_{\Gamma_i}^T$ and $S_\nu = R_\nu S R_\nu^T$. Then $M^{-1}S$ can be written in the additive form

$$M^{-1}S = \sum_{i=1}^{E} R_{\Gamma_i}^T S_{\Gamma_{ii}}^{-1} R_{\Gamma_i} S + R_\nu^T S_\nu^{-1} R_\nu S \,.$$

However, the entries of S and those of $S_{\Gamma_{ii}}$ are not formed explicitly. Instead, the application of

M^{-1} to a vector $\underline{y}$ can be computed by noting that the discrete harmonic functions restricted to Γ_i, $\underline{x}^i = S_{\Gamma_{ii}}^{-1}\underline{y}^i$, are easily computed by solving a *local problem* associated with the two subdomains Ω_k and Ω_ℓ that share edge Γ_i, namely

$$\begin{pmatrix} A_{II}^{k,\ell} & A_{IB}^{k,\ell} \\ A_{BI}^{k,\ell} & A_{BB}^{k,\ell} \end{pmatrix} \begin{pmatrix} \underline{z} \\ \underline{x}^i \end{pmatrix} = \begin{pmatrix} 0 \\ \underline{y}^i \end{pmatrix}, \tag{3.24}$$

where the subscripts B and I represent the index set of boundary and interior dof on Γ_i and $\Omega_k \cup \Omega_\ell$. Note that the value of $\underline{z}$ above is not required in the actual computation. Alternately, the local problems can be solved approximately or $S_{\Gamma_{ii}}$ can be replaced by computationally more efficient matrices (*interface preconditioners*) at the cost of a likely loss in robustness. Various interface preconditioners have been investigated for finite element discretizations. In [35], an interface preconditioner typically used in finite element discretizations was applied to the spectral element method for two subdomains. For more details regarding interface preconditioners, see [37, 36, 39, 20, 51, 18, 76, 122, 123].

The need for a coarse space in DD has been previously discussed in section 3.3.1. In this section, we will only consider two coarse spaces, the vertex based and piecewise constant coarse space. The coarse problem can replace part of the preconditioner or be included as an additional component of the preconditioner. Smith [118] replaces the submatrix $S_{\nu\nu}$ by transforming the basis functions of the approximation space into a partial hierarchical basis. In general, the values of the dof of a function in $V(\Gamma)$ specified by a particular basis are related to the values of the dof under a new basis through the transformation

$$T = \begin{pmatrix} I & R_E^T \\ 0 & I \end{pmatrix},$$

where R_E represents interpolation from the coarse grid values to the edges with the ordering of the coefficients following that of S. When a vertex based coarse problem is used, R_E is simply the linear interpolation operator from the values of the vertices to the edges. A new preconditioner is obtained by first representing S in the new basis, $B = T^T S T$ then applying the same splitting to B as before resulting in

$$\tilde{B} = \begin{pmatrix} S_{\Gamma\Gamma} & 0 \\ 0 & \widetilde{S}_{\nu\nu} \end{pmatrix}.$$

Under this new basis, the edge components of the preconditioner are unchanged and $\widetilde{S}_{\nu\nu}$ is spectrally equivalent to A_H for a vertex based coarse space. We then return to the old basis, $M = T^{-T}\tilde{B}T^{-1}$ from which $M^{-1}S$ can be expressed as

$$M^{-1}S = \sum_{i=1}^{E} R_{\Gamma_i}^T S_{\Gamma_{ii}}^{-1} R_{\Gamma_i} S + R_H^T A_H R_H S,$$

where $R_H = (R_E\, I)$. Preconditioners of this type have been previously considered for the FEM in [52] and the PFEM in [2, 89, 88, 3, 105, 90].

The above scheme can be expressed as an additive Schwarz method [52, 105]. The space V is given by $V_h(\Gamma)$ or the p-finite element space restricted to Γ. We note, however, that the mathematical framework also applies to the spectral element space, $V_{p,h}$. The forms $a(\cdot,\cdot)$ and $f(\cdot)$ in the additive Schwarz template are defined by $s(\cdot,\cdot)$ and $\tilde{f}(\cdot)$. The subspaces are given by

$$V_i = H_0^1(\Gamma_i) \cap V\,, \quad i = 1,\ldots,E\,,$$

and V_0 is the space V^H restricted to Γ.

To define the local bilinear forms, let $\widetilde{\Omega}_i \equiv \Gamma_i \cup \Omega_k \cup \Omega_\ell$ and $\widetilde{V}_i = H_0^1(\widetilde{\Omega}_i) \cap V(\Omega)$. The discrete harmonic extension operators, $\widetilde{\mathcal{H}}_i : V_i \to \widetilde{V}_i$, are defined by

$$\begin{aligned} a(\widetilde{\mathcal{H}}_i u, v) &= 0 \quad \text{for all } v \in \widetilde{V}_i \backslash V_i \\ \widetilde{\mathcal{H}}_i u &= u \quad \text{on } \Gamma_i\,. \end{aligned}$$

The bilinear forms are then given by

$$\begin{aligned} b_0(u,v) &= s(u,v) \\ b_i(u,v) &= a(\widetilde{\mathcal{H}}_i u, \widetilde{\mathcal{H}}_i v)\,, \qquad i = 1,\ldots,E. \end{aligned}$$

The operators $T_{\Gamma_i} : V \to V_i$ and $T_0 : V \to V_0$ are defined by

$$\begin{aligned} b_0(T_0 u, v) &= s(u,v) \quad \text{for all } v \in V_0 \\ b_i(T_{\Gamma_i} u, v) &= s(u,v) \quad \text{for all } v \in V_i,\ i = 1,\ldots,E \end{aligned}$$

which completes the description of the additive Schwarz operator

$$T_{is} = T_0 + T_{\Gamma_1} + \cdots + T_{\Gamma_E}\,.$$

The constant C_1 in section 3.2.1 is bounded by $(1+\log p)^2$ where $p = H/h$ for the h-finite element spaces. A bound for the constants C_2 and $\rho(\mathcal{E})$ are obtained as before. These results are summarized in the following theorem by Dryja, Smith and Widlund [52] and Pavarino [105] for the h and p-finite element spaces respectively.

Theorem 5 *The condition number of the Schwarz operator is given by*

$$\kappa(T_{is}) \leq C(1+\log p)^2\,,$$

where C is a constant independent of p and $p = H/h$ for the h-finite element space.

Conjecture 2 *The condition number of T_{is} satisfies the estimate $\kappa(T_{is}) \leq C(1+\log p)^2$ when V is given by the spectral element space $V_{p,h}(\Gamma)$.*

Results of numerical experiments conducted are given in chapter 4.

Interface preconditioners can be accommodated by defining a suitable approximation to $a(\cdot,\cdot)$ in the definition of $b_i(\cdot,\cdot)$ above. An approximation to the local bilinear forms for the spectral element spaces can be defined by

$$b_i(u,v) = a(\widetilde{\mathcal{H}}_i R_p^h u, \widetilde{\mathcal{H}}_i R_p^h v) \,,$$

where $R_p^h : V_{p,h}(\Gamma) \rightarrow V_h(\Gamma)$ by $(R_p^h v - v)|_x = 0$ for all $x \in G_p \cap \Gamma$ and $\widetilde{\mathcal{H}}_i$ are defined for the h-finite element spaces. That is, the local problems in (3.24) are replaced by computationally more efficient finite element approximations.

The application of T_{is} to a known function u^n requires firstly the evaluation of $s(u^n, v)$ for all $v \in V$. The result of $s(u^n, v)$ for all $v \in V_i$ is obtained as before by recognising that $V_i \subset V$. Let $\mathcal{H}_k$ be the discrete harmonic extension of u^n from $\partial\Omega_k$ to Ω_k given by

$$\begin{aligned} a(\mathcal{H}_k u^n, v) &= 0 \quad \text{for all } v \in V(\Omega) \cap H_0^1(\Omega_k) \\ \mathcal{H}_k u^n &= u^n \text{on } \partial\Omega_k \,. \end{aligned}$$

where $a_k(\cdot,\cdot)$ is the restriction of a $a(\cdot,\cdot)$ to Ω_k. The evaluation of $s(u^n, v)$ is subdivided in a way similar to the computation of stiffness matrices in finite element theory. Since

$$s(u^n, v) = \sum_{k=1}^{N} s_k(u^n|_{\partial\Omega_k}, v|_{\partial\Omega_k}) \,, \tag{3.25}$$

and $s_k(u,w) \equiv a_k(\mathcal{H}_k u, \mathcal{H}_k w)$, applying the definition of the harmonic extension operators, the solution of the homogeneous Dirichlet problems,

$$\begin{aligned} a_k(\mathcal{H}_k u^n, v) &= 0 \quad \text{for all } v \in V(\Omega) \cap H_0^1(\Omega_k) \\ \mathcal{H}_k u^n &= u^n \text{ on } \partial\Omega_k \,, \end{aligned}$$

are used to evaluate the products in (3.25).

Secondly, the local problems

$$b_i(T_{\Gamma_i} u^n, v) = s(u^n, v) \quad \text{for all } v \in V_i$$

require the solution of the Dirichlet problems

$$a(\widetilde{\mathcal{H}}_i T_{\Gamma_i} u^n, v) = \begin{cases} 0 & , \text{ for all } v \in \widetilde{V}_i \backslash V_i \\ s(u^n, v) & , \text{ for all } v \in V_i \,. \end{cases}$$

3.4.2 Neumann-Neumann methods

The second main approach to constructing the local problems consists of maintaining edge and vertex coupling but eliminating the coupling between subdomains.

Summarizing this approach; since $S = \Sigma N_B^i S^i N_B^{iT}$, eliminating the coupling between subdomains results in the preconditioner,

$$M^{-1} = \sum_{i=1}^{N} N_B^i D_i^\dagger (S^i)^{-1} D_i^\dagger N_B^{iT} \,,$$

where $D_i^\dagger$ are diagonal scaling matrices yet to be defined. The action of the inverse of a local Schur complement matrix on a vector $\underline{y}_B^i$ can be implicitly formed by solving the *local Neumann problem*

$$\begin{pmatrix} A_{II}^i & A_{IB}^i \\ A_{BI}^i & A_{BB}^i \end{pmatrix} \begin{pmatrix} \underline{x} \\ \underline{x}_B^i \end{pmatrix} = \begin{pmatrix} 0 \\ \underline{y}_B^i \end{pmatrix} . \tag{3.26}$$

However, the local Schur matrices are singular whenever $\partial\Omega_i \cap \partial\Omega = \emptyset$ and an appropriate pseudo inverse is required.

The Neumann-Neumann method was first introduced in [19, 78, 44] and further developed and extended in [91, 93, 43, 57]. The Schwarz formulation of Neumann-Neumann methods is first described without a coarse space. The coarse operator is subsequently added to the Schwarz operator.

The forms $a(\cdot,\cdot)$ and $f(\cdot)$ for a Schwarz template are taken to be $s(\cdot,\cdot)$ and $\tilde{f}(\cdot)$ as described earlier. Initially, Neumann-Neumann methods were only applied to finite element spaces, that is, $V = V_h(\Gamma)$. However, the following formulation of Neumann-Neumann methods is also applicable to the spectral element space $V_{p,h}(\Gamma)$. The subspaces are chosen as

$$V_i = V(\partial\Omega_i), \quad i = 1, \ldots, N . \tag{3.27}$$

Clearly $V = V_1 + V_2 + \cdots + V_N$.

The following N *counting functions* are required to define the local bilinear forms,

$$\nu_i(x) = \begin{cases} 0 , & \text{if } x \in \Gamma \backslash \partial\Omega_i \\ 1 , & \text{if } x \in \partial\Omega \\ \text{card}\,\{j : \ x \in \partial\Omega_j\}, & \text{if } x \in \partial\Omega_i \end{cases}$$

and the respective pseudo inverses

$$\nu_i^\dagger(x) = \begin{cases} 0 , & \text{if } \nu_i(x) = 0 \\ \nu_i^{-1}(x) & \text{, otherwise} \end{cases}$$

which satisfy the property $\Sigma \nu_i^\dagger(x) = 1$ for all $x \in \Gamma \cup \partial\Omega$.

Let $I_h(\nu_i u)$ for $u \in V_i$ denote linear interpolation onto V, then the local bilinear forms are given by

$$b_i(u,v) = \widehat{s}_i(I_h(\nu_i u), I_h(\nu_i v)) \equiv \widehat{a}_i(\widehat{\mathcal{H}}_i(\nu_i u), \widehat{\mathcal{H}}_i(\nu_i v)) \tag{3.28}$$

where $\widehat{\mathcal{H}}_i$ are defined similarly to $\mathcal{H}_i$ by

$$\begin{aligned} \widehat{a}_i(\widehat{\mathcal{H}}_i u, v) &= 0 \quad \text{for all } v \in V(\Omega) \cap H_0^1(\Omega_i) \\ \widehat{\mathcal{H}}_i u &= u \quad \text{on } \partial\Omega_i \end{aligned}$$

and $\widehat{a}_i(\cdot,\cdot)$ are bilinear forms defined on subdomain Ω_i such that the associated local Schur matrix is nonsingular. One particular choice for $\widehat{a}_i(\cdot,\cdot)$, given by Dryja and Widlund [57], is

$$\widehat{a}_i(u,v) = \int \nabla u \cdot \nabla v \, d\Omega_i + C \int uv \, d\Omega_i$$

where C is a scaling factor. The approach taken in [44], is simply to replace the zero pivot in the factorization of the linear system in (3.26) by an averaged positive pivot. Another approach taken by Dryja and Widlund [57] is to apply Dirichlet boundary conditions at the vertices of individual subdomains. In this case, the N subspaces in (3.27) do not span V. The coarse space can also be used to counteract this problem. This approach is referred to as a *mixed* Neumann-Neumann method and will be completely defined shortly.

The operators $T_i : V \rightarrow V_i$ are defined by

$$b_i(T_i u, v) = s(u, v) \quad \text{for all } v \in V_i$$

which are used to construct the additive Schwarz operator,

$$T_{nn} = T_1 + T_2 + \cdots T_N .$$

Dryja and Widlund [57] have proved that the constants C_1 and C_2 are bounded by H^{-2} and $(1 + \log H/h)^2$ respectively for the h-finite element spaces. The following theorem is proved in the above-mentioned paper.

Theorem 6 *The condition number of T_{nn} satisfies the estimate*

$$\kappa(T_{is}) \leq CH^{-2}(1 + \log H/h)^2 ,$$

where C is a constant independent of H and h.

An approximation to the local bilinear forms in (3.28) for the spectral element spaces can be obtained in a similar fashion as in iterative substructuring methods. The approximation is given by

$$b_i(u, v) = \widehat{s}_i(I_h(\nu_i R_p^h u), I_h(\nu_i R_p^h v)) \equiv \widehat{a}_i(\widehat{\mathcal{H}}_i(\nu_i R_p^h u), \widehat{\mathcal{H}}_i(\nu_i R_p^h v)),$$

where I_h denote linear interpolation onto $V_h(\Gamma)$, and constitutes a finite element approximation to $b_i(\cdot, \cdot)$.

Implementing the above Schwarz method requires evaluating $s(u^n, v)$ for a given function u^n. This process has been outlined earlier. The local problems

$$b_i(T_i u^n, v) = s(u^n, v) \quad \text{for all } v \in V_i$$

are equivalent to computing $w_B^i = T_i u^n$ in

$$\widehat{a}_i(\widehat{\mathcal{H}}_i(\nu_i w_B^i), \widehat{\mathcal{H}}_i(\nu_i v)) = \begin{cases} 0, & \text{for all } v \in V(\Omega) \cap I_0^1(\Omega_i) \\ s(u^n, v), & \text{for all } v \in V_i . \end{cases}$$

The matrix equivalent of these local problems are

$$\begin{pmatrix} A_{II}^i & A_{IB}^i \\ A_{BI}^i & A_{BB}^i \end{pmatrix} \begin{pmatrix} \underline{z} \\ D_i \underline{w}_B^i \end{pmatrix} = \begin{pmatrix} 0 \\ D_i^\dagger \underline{g}_B^i \end{pmatrix}$$

where $\underline{y}_B^i$ is derived from $s(u^n, v)$ and $D_i(D_i^\dagger \equiv D_i^{-1})$ are diagonal matrices with entries corresponding to $\nu_i(x)$ $(\nu_i^\dagger(x))$ evaluated at the nodal dof, $x \in \partial\Omega_i$.

Neumann-Neumann methods with a coarse space

The factor H^{-2} in the estimate of theorem 6 can be removed by adding a coarse space to the decomposition of V. Two coarse spaces will be considered.

The space V is decomposed according to $V = V_0 + V_1 + \cdots + V_N$, where V_0 is the space $V^H(\Omega)$ restricted to Γ. The bilinear form associated with this coarse space is given by

$$b_0^H(u, v) = s(u, v) .$$

The operator $T_H : V \to V^H$ defined by $b_0^H(T_H u, v) = s(u, v)$ for all $v \in V_0$, can be incorporated into T_{nn} in an additive fashion [57],

$$T_{nnh} = T_H + T_1 + \cdots + T_N .$$

The mixed Neumann-Neumann method of Dryja and Widlund [57] can be defined by incorporating T_H into T_{nn} in an additive fashion with the local problems as described earlier. The associated Schwarz operator is denoted by T_{nnm}.

An alternative coarse space developed by Mandel [91] and Dryja and Widlund [57] consists of a subdomain by subdomain piecewise constant space. Therefore, an unstructured decomposition of Ω of the form in chapter 2 section 2.3 can be used, although the underlying approximation space must still be conforming. Let $\mathcal{N}_I$ be the index set of subdomains such that $\partial\Omega_i \cap \partial\Omega = \emptyset$, then

$$V^C = span\{\nu_i^\dagger\}_{i \in \mathcal{N}_I}$$

or alternately, V^C is given by the range of the interpolation operator

$$I^C u(x) = \sum_{i \in \mathcal{N}_I} \overline{u}_{\partial\Omega_i} \nu_i^\dagger(x) , \tag{3.29}$$

where $\overline{u}_{\partial\Omega_i}$ is the average of u over $\partial\Omega_i$.

The bilinear form for the coarse space is defined by

$$b_0^C(u, v) = (1 + \log p)^2 \sum_{i \in \mathcal{N}_I} s_i(u, v) ,$$

with $p = H/h$ for h-finite element spaces and $T^C : V \to V^C$ is defined by $b_0^C(T^C u, v) = s(u, v)$ for all $v \in V_0$.

Dryja and Widlund [57] incorporate the operator T^C into T_{nn} in an additive fashion,

$$T_{nnc} = T^C + T_1 + T_2 + \cdots + T_N .$$

The balancing domain decomposition method of Mandel [91] is described by the hybrid Schwarz method

$$T_{bdd} = (I - T^C)T_{nn}(I - T^C) + T^C .$$

For h-finite element spaces, the constant C_2 in section 3.2.1 for the operators T_{nnh} and T_{nnc} is bounded by $(1+\log H/h)^2$ while C_1 and $\rho(\mathcal{E})$ are independent of H and h. However, the constant C_1 is bounded by $(1+\log H/h)$ for the operator T_{nnm}. The proofs of the respective bounds in the following theorem may be found in [57, 93].

Theorem 7 *For $V = V_h(\Gamma)$, the Schwarz operators T_{nnh}, T_{nnm}, T_{nnc} and T_{bdd} have condition number bounded by*

$$C\left(1+\log\frac{H}{h}\right)^{\alpha}$$

where C is a constant independent of H and h, and $\alpha = 3$ for T_{nnm} otherwise $\alpha = 2$.

Conjecture 3 *For $V = V_{p,h}(\Gamma)$, the Schwarz operators T_{nnh}, T_{nnm}, T_{nnc} and T_{bdd} have condition number bounded by*

$$C\,(1+\log p)^{\alpha}$$

where C is a constant independent of p, and $\alpha = 3$ for T_{nnm} otherwise $\alpha = 2$.

Details of numerical experiments conducted are given in chapter 4.

The matrix A_C corresponding to A_H for the space V^C is computed as follows. Let $\underline{u}$ represent the column vector corresponding to $u \in V$, and $\underline{u}_c$ the coefficients of the coarse interpolant in (3.29), then the mapping between the two sets of coefficients is described by

$$\underline{u}_c = B_C \underline{u}\,,$$

where $B_C(i,:) = (\widehat{D}_i^{\dagger})^T N_B^{iT}$ for $i \in \mathcal{N}_I$, and $\widehat{D}_i^{\dagger}$ is the column vector $diag(D_i^{\dagger})$. Therefore, applying the definition of $b_0^C(\cdot,\cdot)$,

$$A_C = B_C S B_C^T\,.$$

The construction of A_C requires the product of $N_B^i \widehat{D}_i^{\dagger}$ with S for all $i \in \mathcal{N}_I$. This computation can be performed by solving systems on each subdomain Ω_i of the form

$$\begin{pmatrix} A_{II}^i & A_{IB}^i \\ A_{BI}^i & A_{BB}^i \end{pmatrix} \begin{pmatrix} \underline{z} \\ \underline{x}_B \end{pmatrix} = \begin{pmatrix} 0 \\ N_B^j \widehat{D}_j^{\dagger} \end{pmatrix},$$

where j belongs to the index set of subdomains surrounding subdomain Ω_i.

3.5 Further instances of Schwarz methods

We have mainly considered Schwarz methods for the constant coefficient SPD elliptic problem. However, extensions to more general elliptic systems exist. In particular, extensions to the general nonsymmetric indefinite elliptic problem described in chapter 2 and the equations of linear elasticity have been considered in [118, 27, 23, 28]. Schwarz methods for the mixed formulation

of elliptic equations, the Dirichlet biharmonic problem and nonconforming discretizations are given in [95, 96, 130, 116]. In addition, the generalization of the two level Schwarz methods to multilevel Schwarz methods appear in [131, 53, 114]. A multidomain Schwarz alternating method has also been applied to the Stokes equations [65, 64], although there has been no study that incorporates a coarse space into this algorithm. Such an approach may prove competitive with the nested iterative scheme outlined in chapter 2.

Chapter 4

Numerical Experiments

In this chapter, the validity of the conjectures in chapter 3 is investigated experimentally. In addition, we demonstrate the efficiency of the global additive Schwarz method for the spectral element discretization compared with that of the conventional preconditioned iterative solution techniques described in chapter 2.

All numerical experiments conducted solve the model problem

$$-\Delta u = f \quad \text{in } \Omega$$

$$u = 0 \quad \text{on } \partial\Omega \ ,$$

where $\Omega = [0,1] \times [0,1]$ and $f = 2(x + y - x^2 - y^2)$ is chosen such that $u = xy(1-x)(1-y)$. A PCG method was used with the termination criteria that the original residual is decreased by five orders of magnitude. The number of PCG iterations required for convergence to a specified tolerance is proportional to the condition number of the preconditioned system (chapter 2 section 2.1.3). Consequently, the condition number bounds in the conjectures of chapter 3 can be interpreted as a bound on the number of PCG iterations. Therefore, we only report on the PCG iteration counts. The experiments were conducted using MATLAB 4.0 for Windows.

4.1 Numerical experiments with global Schwarz methods

In this section, the Schwarz method of Theorem 1 and conjecture 1 in chapter 3 will be investigated experimentally. Besides reporting on the number of PCG iterations required for convergence, a comparative study is made between the efficiency of the Schwarz method and the finite element preconditioning strategy described in chapter 2. To this end, we consider the computational complexity of each method. However, the complexities do not include the effects of computer storage and memory requirements for a particular problem.

We first investigate the spectral equivalence of M_{AS}, as described in section 3.2.1, to the finite element discretization matrix for a piecewise linear finite element space based on triangular elements. Two finite element grids are considered, a uniformly spaced grid G_h and the grid G_r discussed in chapter 2. The domain is decomposed into $D_x \times D_x$ uniform subdomains with $p+1$

unknowns in each direction on each subdomain. The number of iterations required using one mesh point overlap for the grid G_h is depicted in table 4.1. The results of using the grid G_p and the effect of increasing the amount of overlap is illustrated in table 4.2. The column labelled 'Factor' in table 4.2 is defined as follows.

The solution of the local problems related to A_i^{-1} is performed using banded methods [69]. The approximate complexity of the PCG method with the additive Schwarz preconditioner, as shown by Smith [118], is given by

$$\begin{aligned} D_x^2\,\text{factor}(p+ovrlp) + \text{factor}(D_x-1) + \alpha\,\{\,&\text{CG}(p,D_x) + M_{fem}(p,D_x) + \text{R}(p,D_x) \\ &+ D_x^2\,\text{solve}(p+ovrlp) + \text{solve}(D_x-1)\,\} \end{aligned} \tag{4.1}$$

where

$ovrlp$	=	the amount of overlap used measured in finite element mesh points
$\text{factor}(x)$	=	$\frac{1}{2}(x+1)^2x^2 - \frac{1}{3}(x+1)^3 + \frac{3}{2}(x+1)x^2$
$\text{solve}(x)$	=	$2x^2(x+1) - (x+1)^2$
$\text{CG}(p,D_x)$	=	$4p^2D_x^2$; the complexity of the PCG inner products
$\text{R}(p,D_x)$	=	$2p^2D_x^2$; the complexity of the coarse grid interpolation
α	=	the number of PCG iterations required
$M_{fem}(p,D_x)$	=	the complexity of a matrix vector product with the finite element matrix.

The column labelled 'Factor' in table 4.2 refers to the ratio of the complexities of the Schwarz method using the grid G_p, when the amount of overlap used is one and two mesh points wide respectively. For example, when $D_x = 7$ and $p = 12$ and using the grid G_p, two mesh points overlap results in approximately 18% less floating point operations than using only one mesh point overlap. Figure 4.1 illustrates the relative increase in the computational complexity of the local problems, when the amount of overlap is increased from one to two mesh points. This figure explains the relatively slow increase of 'Factor' in table 4.2 for increasing p, opposed to the significant decrease in the number of iterations when the amount of overlap is increased.

Table 4.2 verifies the bound given in theorem 3 of chapter 3. When p is held constant while varying D_x, the amount of overlap is a fixed fraction of the subdomain size and the number of PCG iterations required for convergence is asymptotically independent on the number of subdomains. The bound can also be verified when D_x is constant while varying p. For example, in conjunction with table 4.3, an identical amount of overlap is used when $D_x = 7$, $p = 12$ using two mesh points overlap, compared with $D_x = 7$, $p = 7$ using one mesh point overlap. However, maintaining $ovrlp = 1$, the amount of overlap is reduced when D_x is fixed and p is increased, which results in an increased number of iterations.

A comparison between the complexity of one application of the additive Schwarz preconditioner to a vector, to one matrix vector product with the finite element discretization matrix is depicted in figure 4.2. The approximate solution of the local problems could be effective in reducing the computational complexity of the preconditioner relative to the matrix vector product. This presumption has been experimentally validated in [122, 17].

p/D_x	2	4	5	6	7	8	9	10	12
4	7	10	10	10	10	10	10	10	10
5	7	10	10	11	11	11	11	11	11
6	7	11	11	11	11	11	11	11	11
7	8	11	11	12	12	12	12	12	12
8	8	11	12	12	12	12	12	12	12
9	8	12	12	13	13	13	13	13	13
10	8	12	13	13	13	13	13	13	13
12	9	12	14	14	14	14	14	14	14

Table 4.1: Number of iterations required using T_{as} for the finite element discretization on the grid G_h with $ovrlp = 1$.

	$D_x = 7$					$p = 8$			
	$ovrlp = 1$	$ovrlp = 1$	$ovrlp = 2$			$ovrlp = 1$	$ovrlp = 1$	$ovrlp = 2$	
p	G_h	G_p	G_p	Factor	D_x	G_h	G_p	G_p	Factor
4	10	10	10	0.68	2	8	10	8	0.93
5	11	12	10	0.84	4	11	14	11	0.96
6	11	13	10	0.94	5	12	15	11	1.02
7	12	14	11	0.95	6	12	15	11	1.02
8	12	15	11	1.02	7	12	15	11	1.02
9	13	16	12	1.02	8	12	15	11	1.02
10	13	18	13	1.07	10	12	15	11	1.03
12	14	21	14	1.18	12	12	15	11	1.03

Table 4.2: Number of iterations required using T_{as} for the finite element discretization on the grids G_h and G_p.

p	Nodal points		
4	1	0.65	0.00
5	1	0.77	0.29
6	1	0.83	0.47
7	1	0.87	0.59
8	1	0.90	0.68
9	1	0.92	0.74
10	1	0.93	0.79
12	1	0.95	0.85

Table 4.3: The last three nodal points on $[-1, 1]$.

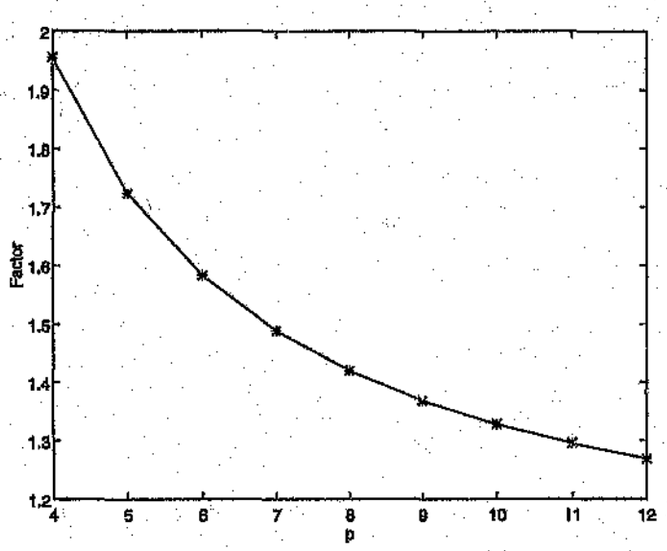


Figure 4.1: The relative increase in the computational complexity of the local problems when the amount of overlap is increased from one to two mesh points.

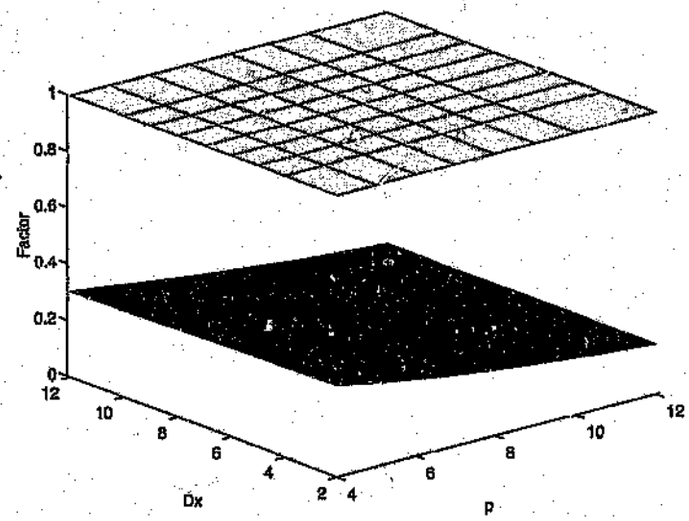


Figure 4.2: A point on the darker surface indicates the ratio of flops of one matrix vector product with M_{fem} to one application of the additive Schwarz preconditioner.

p/D_x	2	4	5	6	7	8	9	10	12
4	7	9	9	9	9	9	9	9	9
5	8	10	11	11	11	11	11	11	11
6	10	11	13	12	12	12	12	12	12
7	11	13	14	14	14	14	14	14	14
8	12	14	16	15	16	16	16	16	16
9	13	15	17	17	17	17	17	17	17
10	14	16	19	18	19	19	19	19	19
12	16	19	22	22	22	22	22	22	22

Table 4.4: Number of iterations required using T_{as} for the spectral element discretization.

We now consider the global additive Schwarz method for the spectral element discretization. The domain is once again decomposed into $D_x \times D_x$ uniform subdomains using degree p polynomials in each direction on each subdomain. The computational results for one mesh point overlap are given in table 4.4. The effect of increasing the amount of overlap is illustrated in table 4.5. The column 'Factor' in table 4.5 refers to the ratio of the complexities of the Schwarz method when the overlap is one and two mesh points wide respectively. Since the local problems remain finite element systems, the complexity model of the Schwarz method for the spectral element discretization is identical to that of (4.1) with M_{fem} replaced by the sum factorization matrix vector product complexity given in section 2.1.2.

The bound in conjecture 1 of chapter 3 is verified in table 4.4 for p constant and varying D_x. As depicted in table 4.5, a similar bound appears to hold for the case of minimum overlap and increasing p. By comparison with the FEM, a similar decrease in the number of iterations for increased overlap is not observed. This may be attributed to the fact that the direct solution of the local problems now corresponds to an approximate solution in the spectral element space. Therefore, in both the spectral element and finite element discretizations, the minimum overlap strategy is to be preferred when p is not large.

The spectral element counterpart of figure 4.2 is given in figure 4.3. In this case, the computational complexity of the additive Schwarz preconditioner is considerably lower that of the matrix vector product.

Given the complexity and iteration counts for the above additive Schwarz method, a comparison can be made between the complexity of the additive Schwarz solution strategy and the finite element preconditioning approach discussed in chapter 2. Figure 4.4 illustrates a comparison between the complexities of the two solution techniques. The advantage of using the additive Schwarz method for medium to large scale problems is clearly apparent. Based on the trends of the results presented and the associated operation counts, it can be deduced that the additive Schwarz preconditioning strategy for the iterative solution of the spectral element equations provides for a good alternative to finite element preconditioning. Further, the application of the

	$D_x = 7$				
	$ovrlp = 1$	$ovrlp = 2$	$ovrlp = 1$	$ovrlp = 2$	
p	FEM	FEM	SEM	SEM	Factor
4	10	10	9	12	0.62
5	12	10	11	13	0.71
6	13	10	12	13	0.79
7	14	11	14	14	0.87
8	15	11	16	15	0.94
9	16	12	17	16	0.95
10	18	13	19	16	1.06
12	21	14	22	19	1.05

Table 4.5: Number of iterations required using T_{as} for the spectral element discretization with $ovrlp = 1$ and 2.

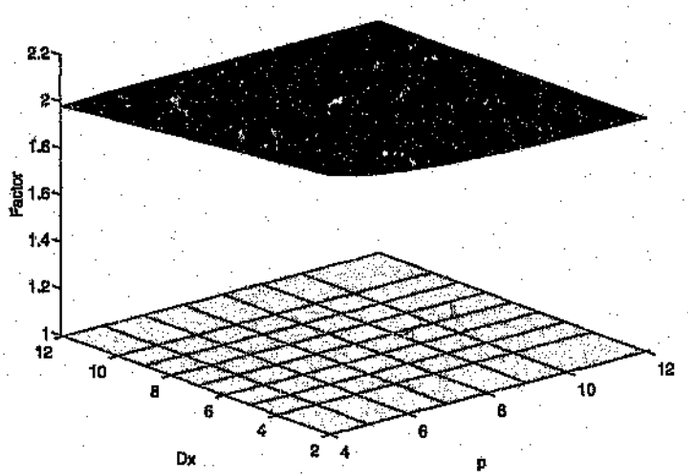


Figure 4.3: A point on the non-constant surface indicates the ratio of flops for one matrix vector product with the spectral element discretization matrix to one application of the additive Schwarz preconditioner, M_{AS}^{-1}.

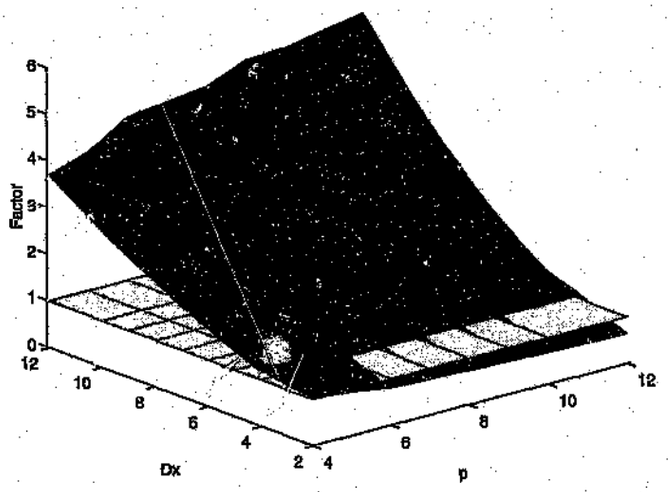


Figure 4.4: A point on the non-constant surface indicates the ratio of flops of the PCG method with M_F^{-1} to the PCG method with M_{AS}^{-1} as a preconditioner.

additive Schwarz preconditioner has favourable opportunities for parallel computing and/or a good blocking of data within a serial environment.

4.2 Numerical experiments with boundary Schwarz methods

Reported in this section are the results of experiments conducted with the Schwarz methods in theorem 5 and 7, and conjecture 2 and 3. Both the grids G_p and G_h are considered for the finite element discretization. The results for the FEM are given in tables 4.6 to 4.9 while the results for the SEM are reported in tables 4.10 and 4.11.

Implementation aspects

We denote T_{ise} by the operator T_{is} in theorem 5 and conjecture 2, when the local bilinear forms are not approximated. The operator T_{isa} for the finite element discretization is the operator T_{is} when the local bilinear forms are approximated by the interface preconditioner J of Dryja [21]. The finite element approximation to the local bilinear forms for the operator in conjecture 2 is used to define the operator T_{isa} for spectral element discretizations.

The operators T_{nnh} and T_{nnc} are implemented by perturbing the local bilinear forms with a scaling of $\int uv \, d\Omega_k$ as described in chapter 3. Note that the form $\int uv \, d\widehat{\Omega}$ discretized using the SEM is simply a diagonal matrix with diagonal entries corresponding to the tensor product of the one dimensional Gaussian quadrature weights. Each local Neumann problem in (3.26) is

$p = 4$

Method/D_x	2	4	5	6	7	8	10	12
T_{nnm}	3	7	7	9	8	10	10	10
T_{nnh}	2	7	7	7	7	7	7	8
T_{nnc}	3	6	8	8	10	11	12	13
T_{bdd}	1	4	4	5	5	6	6	6
T_{isa}	4	7	7	8	8	8	8	8
T_{ise}	4	6	7	8	7	8	8	8
I	4	11	14	17	19	22	28	32

Table 4.6: Iteration counts using the FEM with the grid G_h for increasing D_x.

$D_x = 7$

Method/p	4	5	6	7	8	9	10	12
T_{nnm}	8	9	9	10	11	11	11	12
T_{nnh}	7	8	8	9	9	9	10	10
T_{nnc}	10	11	12	12	12	12	12	12
T_{bdd}	5	6	6	6	6	6	6	7
T_{isa}	8	8	8	10	11	11	12	13
T_{ise}	7	8	8	8	8	8	9	9
I	19	22	24	27	29	31	33	36

Table 4.7: Iteration counts using the FEM with the grid G_h for increasing p.

$p = 4$

Method/D_x	2	4	5	6	7	8	10	12
T_{nnm}	3	8	8	9	9	10	10	10
T_{nnh}	3	7	8	8	8	8	9	9
T_{nnc}	3	6	8	9	11	11	13	14
T_{bdd}	1	4	4	6	6	6	6	7
T_{isa}	4	8	9	10	10	10	11	11
T_{ise}	4	7	7	8	8	8	8	8
I	4	11	12	16	18	20	25	29

Table 4.8: Iteration counts using the FEM with the grid G_p for increasing D_x.

	$D_x = 7$							
Method/p	4	5	6	7	8	9	10	12
T_{nnm}	9	10	11	11	12	13	14	14
T_{nnh}	8	9	10	10	11	11	11	11
T_{nnc}	11	12	12	13	14	14	14	15
T_{bdd}	6	6	6	7	7	7	8	8
T_{isa}	10	11	12	13	13	14	15	16
T_{ise}	8	8	9	9	10	10	10	11
I	18	20	22	24	25	26	27	29

Table 4.9: Iteration counts using the FEM with the grid G_p for increasing p.

	$p = 4$							
Method/D_x	2	4	5	6	7	8	10	12
T_{nnm}	3	9	10	11	11	13	13	14
T_{nnh}	3	6	7	7	7	7	7	7
T_{nnc}	3	7	9	9	11	11	13	13
T_{bdd}	1	4	4	6	6	7	7	7
T_{isa}	4	8	8	9	9	9	9	10
T_{ise}	4	7	8	8	9	9	9	9
I	4	11	13	16	18	20	26	30

Table 4.10: Iteration counts using the SEM for increasing D_x.

	$D_x = 7$							
Method/p	4	5	6	7	8	9	10	12
T_{nnm}	11	13	14	15	15	16	16	17
T_{nnh}	7	8	8	9	9	9	9	10
T_{nnc}	11	12	12	13	14	14	14	15
T_{bdd}	6	6	7	7	7	8	8	8
T_{isa}	9	9	10	10	10	11	11	11
T_{ise}	9	9	10	10	10	11	11	11
I	18	20	22	24	25	27	30	34

Table 4.11: Iteration counts using the SEM for increasing p.

perturbed by a scaling of this matrix. Experimentally, this perturbation provided lower iteration counts for the Neumann-Neumann methods applied to finite element discretizations than the approximation of $\int uv \, d\widehat{\Omega}$ using the finite element basis functions. Since the performance of T_{bdd} is independent of the solution approach taken for the local Neumann problems, see Mandel [91], the zero pivot in (3.26) is replaced by an averaged positive pivot in the factorization of the linear systems.

The scaling of the coarse operator in T_{nnc} and T_{nnh} was taken to be $(1 + \log(p/4))^2$ for the nonuniform grid discretizations, which yielded lower iteration counts when the scaling of $(1 + \log p)^2$ was used.

Discussion of the results

The performance of the above-mentioned Schwarz methods for realistic problems cannot be ascertained solely from experimental results with the model problem. However, a number of theoretical results and computational trends can be verified.

The number of PCG iterations required for convergence for all the Schwarz methods appear to be asymptotically independent of the number of subdomains. Moreover, there is a sub-linear dependence on the number of PCG iterations for increasing p and constant D_x. However, as indicated in theorem 7, the operator T_{nnm} appears to be more sensitive for increasing p than T_{nnc}, T_{nnh} and T_{bdd}. However, the Schwarz methods demonstrate greater robustness for increasing p than the global additive Schwarz methods investigated in section 4.1. We also note the close relationship between the number of PCG iterations required for the spectral element discretization compared with the finite element discretization on the grid G_p.

A remarkable result is that T_{isa} yields no worse iteration counts than T_{ise} for the SEM. That is, a computationally cheaper finite element approximation to the local bilinear forms of T_{is} for the SEM can be made without a performance degradation of the method. However, we have not yet implemented similar approximate local problems for the Neumann-Neumann methods.

The number of iterations for T_{nnh} is consistently lower than for T_{nnm}, a trend which is particularly noticeable in both the nonuniform grid FEM and the SEM. The iteration counts for additive Schwarz methods deteriorate when a piecewise linear coarse space is replaced by a piecewise constant coarse space. Nevertheless, the condition number bounds on the additive Schwarz operators are of the same order when either coarse space is used. In this respect, the robustness of the hybrid Schwarz operator T_{bdd}, which uses a piecewise constant coarse space, is clearly apparent when compared with that of T_{nnc} and even T_{nnh}. Presently, Neumann-Neumann methods with a piecewise constant coarse space is the only boundary Schwarz method that can be employed to solve conforming discretizations on an unstructured domain decomposition similar to chapter 2 section 2.3. In view of applying Schwarz methods to unstructured decompositions, a more robust additive Schwarz method would therefore require a higher order coarse space without the restrictions of present piecewise linear coarse spaces. A recent article by Mandel [92] offers new insight into this problem.

A further distinction between the Neumann-Neumann methods of hybrid and additive Schwarz type, is the relative sensitivity of the additive method to the solution approaches of the local Neumann problems and the scaling of both the Neumann and coarse problems. The hybrid Schwarz method is more robust in this regard. A more detailed understanding is needed concerning the effect of scaling factors and solution approaches for the local Neumann problems on the overall performance of the additive Schwarz method.

Note that T_{nnh} (T_{ise}) requires the solution of one (four) Neumann (Dirichlet) problem(s) on each subdomain for the local problems. Further, since the computational cost to solve one Neumann problem never exceeds twice the cost to solve one Dirichlet problem (see for example figure 4.1) we see that T_{ise} is atleast twice as expensive per iteration as T_{nnh}. However, a more realistic implementation would employ approximate local problems. A proper comparison of these two schemes therefore requires further research.

Chapter 5

Nonconforming Spectral Element Solution Techniques

5.1 Introduction

The nonconforming SEM has been outlined in chapter 2 section 2.3 and demonstrates a number of potential advantages over the conforming SEM. Although the nonconforming SEM is well suited to a PCG solution technique [82], a direct method is the preferred solution approach owing to the lack of good preconditioners [74]. Computational results presented in chapter 4 indicate the efficiencies afforded by a Schwarz method compared with conventional preconditioning techniques for the conforming SEM. Consequently, the application of Schwarz methods to the nonconforming spectral element discretization could provide for an effective tool in the solution of the resulting system of equations. The use of domain decomposed solution techniques for unstructured decompositions has only recently received attention [13, 79]. We outline below a number of the difficulties in applying the Schwarz methods of chapter 3 to the nonconforming spectral element discretization.

5.2 Solution techniques

Diagonal preconditioning

The formation of the modified local stiffness matrices for the nonconforming SEM has been discussed in chapter 2. The contribution from subdomain Ω_k to the linear system of equations in (2.23),

$$R^T \widehat{A} R \underline{u} = R^T \underline{\widehat{f}} \tag{5.1}$$

is given by $\widehat{Q}^{k^T} A^k \widehat{Q}^k \underline{u}^k = \widehat{Q}^{k^T} \underline{\widehat{f}}^k$ or

$$\begin{pmatrix} A^k_{II} & \widehat{A}^k_{IB} \\ \widehat{A}^k_{BI} & \widehat{A}^k_{BB} \end{pmatrix} \begin{pmatrix} \underline{u}^k_I \\ \underline{\widehat{u}}^k_B \end{pmatrix} = \begin{pmatrix} \underline{f}^k_I \\ \underline{\widehat{f}}^k_B \end{pmatrix}, \tag{5.2}$$

where $\widehat{A}^k_{IB} = A^k_{IB}Q^k = (\widehat{A}^k_{BI})^T$, $\widehat{A}^k_{BB} = (Q^k)^T A^k_{BB} Q^k$, $\widehat{f}^k_B = (Q^k)^T f^k_B$ and Q^k is the mortar to subdomain boundary projection. Vector $\widehat{u}^k_B$ now refers to the unknowns located on the mortars bordering subdomain Ω_k, see for example figure 2.7 of chapter 2.

The contribution to the Schur complement matrix S from subdomain Ω_k can be expressed as

$$S^k = \widehat{A}^k_{BB} - \widehat{A}^k_{BI} A^{-1}_{II} \widehat{A}^k_{IB} ,$$

from which S can be assembled through direct stiffness summation $\sum \widehat{N}^k_B S^k \widehat{N}^{k^T}_B$ where $\widehat{N}^k_B$ is defined by $\widehat{u}^k_B = \widehat{N}^{k^T}_B u$.

The first solution strategy is to solve the Schur complement system iteratively using a diagonal preconditioner. This approach may result in lower iteration counts than a diagonal preconditioner for (5.1). The computational kernel of this approach is the formation of a matrix vector product with the Schur matrix,

$$S\underline{y} = \sum \widehat{N}^k_B (\widehat{A}^k_{BB} - \widehat{A}^k_{BI} A^{-1}_{II} \widehat{A}^k_{IB}) \underline{y}^k , \quad \underline{y}^k = \widehat{N}^{k^T}_B \underline{y} .$$

Clearly, the effect of Q^k need only be known implicitly through matrix vector products. This approach requires that the matrices A^k are assembled and A^k_{II} are factorized. However, applying the arguments of section 3.4, a preconditioner for (5.1) can be constructed employing finite element approximations to A^k.

Table 5.1 indicates the number of iterations required to solve the model problem on the geometry in figure 2.6 (c) of chapter 2 using a PCG method for the Schur complement system. The associated preconditioner is denoted I. In this case, the total number of subdomains is given by $D^2_x + 2\left\lfloor \frac{D_x}{2} \right\rfloor$.

Iterative substructuring methods

Iterative substructuring methods were initially developed for conforming discretizations on a structured decomposition of the domain. A second approach consists of an algebraic extension of iterative substructuring methods to the nonconforming SEM. Following the arguments in section 3.4.1, the Schur complement matrix can be re-ordered according to

$$\begin{pmatrix} S_{\gamma_1 1} & S_{\gamma_1 2} & \cdots & S_{\gamma_1 E} & S_{\gamma_1 v} \\ S_{\gamma_2 1} & S_{\gamma_2 2} & & S_{\gamma_2 E} & S_{\gamma_2 v} \\ \vdots & & \ddots & \vdots & \vdots \\ S_{\gamma_E 1} & \cdots & & S_{\gamma_E E} & \\ S_{\gamma_E v} & \cdots & & & S_{vv} \end{pmatrix} ,$$

where in this case, $S_{\gamma_i j}$ denotes the coupling in S between the dof on mortars γ_i and γ_j, and S_{vv} represents the coupling between the *vertices*. Note that a vertex may be the intersection of only two mortars and a mortar may intersect more than two subdomains. A preconditioner M^{-1} for the Schur complement matrix is then obtained (see figures 3.2 and 5.1) by the splitting of S,

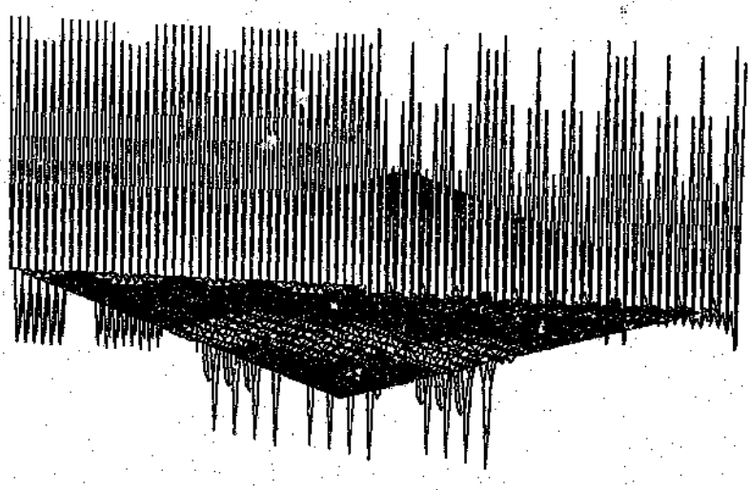


Figure 5.1: The nonconforming spectral element counterpart of figure 3.2 for the geometry in figure 2.6 (c).

$D_x = 4$			$p = 4$		
p	I	M^{-1}	D_x	I	M^{-1}
4	18	12	4	18	12
6	23	14	6	21	15
8	28	16	8	25	20
10	32	17	10	30	23
12	37	18	12	35	28

Table 5.1: The number of PCG iterations for the two preconditioners, I and M^{-1}.

$$M = \begin{pmatrix} S_{\gamma_{22}} & & & 0 \\ & S_{\gamma_{22}} & & \\ & & \ddots & \\ & 0 & & S_{\gamma_{EE}} \\ & & & & S_{\nu\nu} \end{pmatrix}.$$

Once again, there is no need to form the blocks of M directly. Only the computation of $S_{\gamma_{ii}}^{-1}\underline{y}^i$ is required for the prescribed values $\underline{y}^i$ on mortar γ_{ii}. This can be achieved by solving local homogeneous Dirichlet problems or finite element approximations thereof similar to equation (3.24) in chapter 3. However, more than four mortars may surround the boundary of a subdomain and therefore systems of the form in equation (3.24) may be larger than those for conforming discretizations.

As noted earlier, the lack of a means for global transportation of information will degrade the performance of a preconditioner when increasing the number of subdomains. A coarse problem is yet to be included into the above preconditioner and the absences of this component is born out in the computational results depicted in table 5.1. However, it is evident that there is an increased robustness for fixed D_e and increasing p. A recently proposed method by Rønquist [112, 111] may be applicable to the nonconforming discretization and systems of the form in (5.1). This method employs an alternate means for including a coarse problem into the iterative solution of (5.1) and these ideas applied to the Schur complement matrix may provide a mechanism to include a coarse problem into M.

Direct methods

In [74], the Schur complement matrix is formed explicitly and factorized. This approach can be effective for small to medium sized problems when the factored form of the Schur matrix is required. However, for large scale problems, memory requirements are too severe and an iterative solution approach is required.

Neumann-Neumann methods

We briefly comment on the use of Neumann-Neumann methods applied to the nonconforming discretization. The Neumann preconditioner requires the direct formation and factorization of the matrix in (5.2). Since Q^k is in general rectangular (see equation (2.21)), the Neumann problems have a larger null space than the corresponding local problems defined for conforming discretizations. The dimension of these null spaces is easily shown to be dependent on p. Similar problems occur for the nonconforming discretization applied to finite element spaces. Mandel [91] showns that a stable balancing domain decomposition method can result provided the coarse space spans the null spaces of the local Neumann operators. A coarse space with a dimension

dependent on the resolution of the discretization is clearly prohibitive and therefore a Neumann-Neumann method applied to the nonconforming discretization in this manner is not feasible.

An approach suggested by Le Tallec [79] enforces the nonconforming matching conditions using Lagrange multipliers instead of the transformation approach taken in chapter 2 section 2.3. The solution of the elliptic problem is then obtained by solving the larger indefinite saddle point problem. However, Le Tallec [79] demonstrates how a Neumann-Neumann method can be applied in this situation.

Chapter 6

Closure

A brief appraisal of the study is included in this chapter. The work is placed in context by discussing the strong points, shortcomings, suggestions for further research and the conclusions.

6.1 Summary of the study

Domain decomposition methods are one of the most mathematically well understood techniques for the iterative solution of linear equations arising from the discretization of partial differential equations. A wide variety of solution techniques based on domain decomposition have recently been introduced, inspired primarily by the availability of parallel computers, and with particular emphasis on finite element discretizations. Advances in the theory of Schwarz type domain decomposition methods for finite element discretizations has made it possible to transfer a number of these concepts to higher order discretizations. This study attempts to demonstrate the means by which Schwarz methods can be applied to the spectral element discretizations.

Domain decomposition and finite element discretizations

A first approach to formulate Schwarz methods for the spectral element discretization is to exploit the similarities between the PFEM and the SEM, and apply, algorithmically, Schwarz type methods developed for the PFEM. Solution techniques inherited by this approach are the additive Schwarz method of Pavarino [105] and the numerous boundary Schwarz methods [2, 89, 88, 52]. The solution techniques were formulated in a Schwarz framework in order to emphasise similarities and distinguish disparity between the various techniques. The presentation of chapter 3 symbolizes a growing understanding that Schwarz methods are more generally applicable to a myriad of discretizations and solution techniques. Computational results documented in chapter 4 signify experimental evidence of the applicability of Schwarz methods to the spectral element discretization. There is a vast assortment of boundary Schwarz methods and not all of these methods were investigated for the spectral element discretization. The local problems were solved exactly and minor variations based on the choice of an approximation to the local problems were initially omitted. A notable deficiency in both the current literature and in this study, is

a comparative examination of the predominant methods for realistic problems. However, one of the objectives of this study was to establish possible solution strategies rather than to ascertain the most efficient method for a given problem.

Approximate local problems

Results of numerical experiments conducted in [25, 17] with Schwarz methods for low order finite element discretizations confirm the presumption that an approximation to the local problems can reduce the total amount of computation. The nature of high order methods obviates a greater necessity for efficient local solvers than for lower order finite element methods.

The finite element preconditioner for pseudo-spectral methods was extended to the SEM. The results of numerical experiments conducted are recapitulated in conjecture 1 of chapter 2 and constitute a crucial link between the SEM and the FEM. This connection suggests that Schwarz methods for the finite element discretizations are applicable to the spectral element discretizations. This concept enabled the formulation of a global additive Schwarz method for the spectral element discretization. In addition, the formulation enabled a minimum overlap strategy to be accommodated for the SEM. In contrast, there appears to be no minimum overlap Schwarz method for the PFEM [105]. The concept of finite element preconditioning for the SEM was also used to introduce approximate local problems for boundary Schwarz methods defined for the spectral element discretizations.

The impact of increased memory requirements and setup costs for the approximate solvers was not explored. Nevertheless, one should take into consideration the added efficiency and storage reduction afforded by the sum factorization techniques which can be used to counteract these overheads.

Theoretical aspects

An experimental method of investigation was employed in order to explore the validity of the multitude of solution techniques developed for the SEM. Although the solution techniques are devised within reasonable theoretical grounds, no proofs are developed and only the mathematical framework is formulated. Perhaps the most rudimentary proof technique would be to prove conjecture 1 of chapter 2 and then infer the theoretical results available for Schwarz methods applied to finite element discretizations. The absence of theoretical justification is a recurring issue throughout this study, in both the presentation of existing and proposed methods. Notwithstanding, the work raises a number of intriguing theoretical aspects by indicating experimentally what may be provable.

Numerical experiments

Schwarz methods for spectral element discretizations were only investigated, numerically, for the model elliptic problem in two dimensions. This does not imply that these methods are not applicable to more general elliptic problems in complex geometries. Focus was placed on the primary concepts of the methods, thereby avoiding further programming details necessary for a more general implementation. The necessary foundation for solution strategies based on domain decomposition for the SEM has therefore been established in order to investigate the more arduous problems.

Nonconforming discretizations

The nonconforming SEM represents an important extension of the conforming SEM allowing for more efficient mesh refinement strategies to be employed. The development of Schwarz type solution strategies for the nonconforming SEM has not been adequately resolved in this study. Global and Neumann-Neumann Schwarz methods do not appear to be directly applicable to the nonconforming discretization although iterative substructuring methods seem to be a more viable approach. The development of Schwarz methods for the nonconforming spectral element discretization would be of independent interest than solely for the spectral element discretizations. Further research is therefore required in this direction.

Navier-Stokes solvers

The solution of the Stokes equations is an important computational kernel in many Navier-Stokes solvers. A brief discussion demonstrating the use of elliptic solvers as a kernel for the solution of the Stokes equations was outlined in chapter 2. Another possibility to include Schwarz methods into a Navier-Stokes solver is to employ Schwarz methods to solve the Stokes equations directly. A number of ideas in this regard are detailed in [65], however, these methods do not incorporate a coarse space which proved invaluable for elliptic equations.

Schwarz methods have not been developed with the purpose of being included within a computational kernel. Therefore, further experimentation and development is required if Schwarz methods are to be employed in this fashion.

6.2 Conclusions

The following conclusions have been made from this study.

- The finite element preconditioner for pseudo-spectral methods extends naturally to the SEM and maintains the spectral equivalence property. Further, finite element preconditioning for the SEM was shown to result in significantly fewer floating point operations than diagonal preconditioning.

- The concept of finite element preconditioning for the SEM was used to formulate a global additive Schwarz method for the spectral element discretization. An experimental investigation of this method revealed the following:

  - The number of PCG iterations required for convergence using the minimum overlap strategy is independent of the number of subdomains and approximately linearly dependent on the degree of polynomials used within each subdomain. The number of PCG iterations are also similar to the number of iterations required by a minimum overlap additive Schwarz method for the finite element discretization on a nonuniform grid.
  - Increasing the amount of overlap can reduce the number of PCG iterations, however, this does not necessarily lead to a reduction in the number of float point operations.
  - One application of the additive Schwarz preconditioner is computationally cheaper than one matrix vector product with the discretization matrix.
  - The global additive Schwarz method for the spectral element discretization requires significantly fewer floating point operations than the two preconditioned iterative solution techniques investigated in chapter 2, particularly for large scale problems. In addition, the memory requirements for the additive Schwarz preconditioner are significantly lower than that required by the finite element preconditioner.

- Boundary Schwarz methods were also formulated for the spectral element discretization. Numerical experiments with these methods revealed the following:

  - Boundary Schwarz methods defined for the spectral element discretization require a similar number of PCG iterations to the corresponding methods defined for a finite element discretization on a nonuniform grid. The number of PCG iterations also demonstrate a sub-linear dependence on the resolution of the discretization within each subdomain.
  - A finite element approximation to the local problems in iterative substructuring methods can be made without degrading the performance of the method.
  - Neumann-Neumann methods augmented with a piecewise linear coarse space are more robust than the corresponding methods augmented with a piecewise constant coarse space.

- The means by which Schwarz methods can be applied to the nonconforming discretization remains an open question. Neumann-Neumann methods of the form used in finite element discretizations are not directly applicable to the nonconforming discretization. Iterative substructuring methods appear to be a more viable approach than a Neumann-Neumann method.

6.3 Recommendations for further research

- Theory
  - The proof of the conjectures in chapter 3 is an instinctive route for further investigation, given the computational results in chapter 4. A recent article by Pavarino and Widlund (October 1993) [106] marks the beginning of the theoretical investigation for iterative substructuring methods.
  - Undoubtedly, the future success of the SEM lies in its ability to accommodate adaptive strategies which promote efficient solution techniques. The extension of Schwarz type schemes to an adaptive variant of the conforming SEM is essential in this respect. Alternately, a basis for the approximation space can be sought which preserves the high order accuracy and computational efficiency of the SEM while enabling adaptive strategies and Schwarz methods to be accommodated.
- Extensions to more general problems
  - This study has only considered the model elliptic problem in two dimensions. Future work should address three dimensional problems (Dryja, Smith and Widlund [52]), nonsymmetric problems (Cai and Widlund [28]) and the Stokes equations (Fortin and Aboulaich [64]).
- Experiments
  - Comparative studies of the various Schwarz methods applied to finite element and spectral element discretizations for realistic problems on advanced computer architectures should also be the topic for further research.

Appendix A

Program Listing

A number of MATLAB programs are listed below for some of the Schwarz methods that were implemented. Those program listings which are similar to the listings given below are not supplied. Further, a number of procedures are not listed since these operations are easily implemented. The procedures not listed are outlined below.

[A, b, Vertex, Interface, Boundary, Interior] = **bifemlib**(NDx, 1);
Given the number of rectangular elements in one direction (NDx), this procedure return the discretization matrix A and corresponding right hand side b for a bilinear FEM. The remaining vectors specifies the nodes associated with the vertices, interfaces, boundary and interior of the domain where it is assumed that the nodes are numbered according to a lexicographical ordering.

[A, b, Vertex, Interface, PBoundary, Interior, SKi, SPKi] = **semlib**(p*NDx, NDx);
The SEM counterpart of BIFEMLIB. The parameter p is the order of polynomials used. In addition, SKi is the stiffness matrix on the reference subdomain and SPKI is the perturbation matrix required for the Neumann-Neumann methods.

[A, b, Vertex, Interface, Boundary, Interior, Struct, FKi, FPKi] = **shblib**(p*NDx, NDx, Spacing, Overlap);
The linear-triangular basis FEM counterpart of BIFEMLIB. Spacing = 1 or 2 for coarse grid G_{h} and G_{p}. Overlap =1,2... specifies the amount of overlap used measured in grid widths. Matrices FKi and FPKI are as specified in Section 3. Struct is specified by the procedure, GetStruct, below.

```
function Struct = GetStruct(NDx, NDy, Nx, Ny, Boundary,
overlap)

  NodesInRow = Nx*NDx - (NDx-1);
  NodesInCol = Ny*NDy - (NDy-1);
  for k=1:NDx
    Base = (2-overlap):(Nx-1+overlap);
    Base = Base + ((1-overlap) *NodesInRow).*
                  ones(size(Base));
    Base = Base + ones(size(Base))*( (k-1)*(Nx-1) );
    Factor=0;
    for j=1:NDy
      for i=1:(Ny-2+2*overlap)
        Struct = [Struct (Base +
                  ones(size(Base))*Factor)];
        Factor = Factor + NodesInRow;
      end
      Factor = Factor - NodesInRow*(2*overlap-1);
    end
  end
  Struct = reshape(Struct,((Nx-2+2*overlap)^2),NDx*
                   NDy);
  BStruct = Struct;

  for j=1: NDx*NDy
    sub=zeros(size(Struct(:,j)));
    for i=1: (Nx-2+2*overlap)^2
      y = find( (Struct(i,j).*ones(size(Boundary)) -
      Boundary) == 0);
      if length(y)==0 sub(i) = 0; else sub(i)=1; end
    end

    y = find( (Struct(:,j) < 1) | (Struct(:,j) >
              NodesInRow^2));
    sub(y) = ones(size(y));
    rownum = floor( (Struct(:,j)- ones(size(Struct(:,j)
                    )) ) ./ NodesInRow);
    Struct(:,j) = Struct(:,j) - (NodesInRow+1).*
                  ones(size( Struct(:,j))) - 2.*(
                  rownum - ones(size(rownum)) );
    y = find(sub);
    if (length(y) > 0)
      Struct(y,j)=zeros(length(y),1);
    end
  end
% ---------- end of function ----------
```

```
% Program BDD

% the main calling program which implements T_{bdd} for
both the FEM and SEM

for spacing =1:2
direct      = 1;
DD          = [4];
NN          = [3];
number      = 1;

iter=[];
t0 = clock;
for i = 1 : length(DD)
  for j = NN(i) : NN(i)
    NDx = DD(i);
    N   = j;
    sh
    results1= [results1; [NDx N iter ]]
  end
end
end
t1 = etime(clock, t0)
disp('DD   N   iters');
results1
```

```
Program sh

  NDy          = NDx;
  X1           = 0;
  X2           = 1;
  Y1           = 0;
  Y2           = 1;
  NumNodes     = ( NDy*(N+1)  -(NDy-1) )*(NDx*(N+1)-
                 (NDx-1) );
  NumElements  = NDy*NDx;
  NumVertex    = (NDx-1)*(NDy-1);
  NInterior    = (N-1)^2;
  SchurLength  = (NDx-1)*( NDy*(N+1) -(NDy-1)-2 ) +
```

```
                (NDx-1)*( NDx*(N-1) );
  overlap      = 1;  % make calling routines compatable;

% -------------- Preprocessor--------------

  cd h:\addsh-h
  [PA, Pb, PVertex, PInterface, PBoundary,PInterior,
  Struct1, FKi, FPKi] = shlib(N*NDx, NDx, spacing ,
  overlap);

  clear PInterior;
  cd h:\sem
  if runsem==1
    [A, Pb, Vertex, Interface, PBoundary,Interior, SKi,
    SPKi]= semlib(N*NDx, NDx);
    clear Interior;
  else
    % FPKi = zeros(size(FKi));
  end
  cd h:\bdd

  Pmap = 1:NumNodes;   Pmap(PBoundary) =[];
  Pb = Pb(Pmap);
  clear Pmap;

  cd h:\libs
  Course = getcourse(NDx);
  [Rt, TStruct, Int, Bound] = getrt (N-1,
  length(PInterface),length(PVertex),Course, spacing,
  NDx);
  clear Rt;
  cd h:\bdd

if runsem==1
  g   =  getg(SKi, SKi, Struct1, TStruct,Bound, Int,Pb,
  SchurLength);
  clear Pb;
else
  g   =  getg(FKi, FKi, Struct1, TStruct,Bound, Int,Pb,
  SchurLength);
  clear Pb;
end

if runsem==1
  [Ah, IntDom] = getah(Int, Bound, Struct1,TStruct,
   SchurLength, NDx, SKi);
  [Uf, iter] = pcg3(FKi, SKi, g, 0.0001,Ah, Struct,
  TStruct,  Bound, Int, Precon,IntDom);
else
  [Ah, IntDom] = getah(Int, Bound, Struct1, TStruct,
  SchurLength, NDx, FKi);
  [Uf, iter] = pcg3(FKi, FKi, g, 0.0001, Ah, Struct1,
  TStruct, Bound, Int, Precon,IntDom);
end

% ---------- End of program ------------

function course = GetCourse(DD)

  Intbase1 = 1:DD:(DD-1)*(DD-1);
  Intbase2 = ones(size(Intbase1)) + Intbase1;
  Intbase3 = ( DD*(DD-1)+1 ):(DD^2);
  Vertbase = 1:(DD-1);
  for k=1:(DD-1)
    course(1,Vertbase) = Intbase1;
    course(2,Vertbase) = Intbase2;
    for i=1: DD-1
      course(3,Vertbase(i)) = Intbase3(i);
      course(4,Vertbase(i)) = Intbase3(i+1);
    end
    Intbase1 = Intbase2;
    Intbase2 = ones(size(Intbase2)) + Intbase2;
    Intbase3 = DD.*ones(size(Intbase3)) + Intbase3;
    Vertbase = (DD-1).*ones(size(Vertbase)) + Vertbase;
  end

% end function.


function [Rt, TStruct, Int, Bound, VS] = GetRt(block,
LS, NumVertex, Data, type, DD)

  Rt=zeros(LS, NumVertex);
  alpha = zeros(block,1);
  if type == 1
    h=1/(block+1);
    for i=1:block alpha(i) = 1-i*h; end
  else
    X = jacobl(block+1);
    alpha = ( 0.5.*( X' + ones(size(X')) ));
    alpha( [1, (block+2)] )=[];
  end

  for j=1:NumVertex
    vert = Data(1,j);
    row  = (block*(vert-1)+1):(block*vert);
    Rt(row,j) = flipud(alpha);
    vert = Data(2,j);
    row  = (block*(vert-1)+1):(block*vert);
    Rt(row,j) = alpha;
    vert = Data(3,j);
    row  = (block*(vert-1)+1):(block*vert);
    Rt(row,j) = flipud(alpha);
    vert = Data(4,j);
    row  = (block*(vert-1)+1):(block*vert);
    Rt(row,j) = alpha;

  end
  leftbot  = 1;
  rightbot = DD+1;
  lefttop  = 2;
  righttop = rightbot+1;
  Dom      = zeros(4, NumVertex);
  Vertex = 1:NumVertex; Vertex = reshape(Vertex,  DD-1,
  DD-1);
  Vertex =(Vertex');
  for i=1:DD-1
    for j=1:DD-1
      Dom(1, Vertex(j,i)) = leftbot;
      Dom(2, Vertex(j,i)) = rightbot;
      Dom(3, Vertex(j,i)) = lefttop;
      Dom(4, Vertex(j,i)) = righttop;
      leftbot  = lefttop;  lefttop  = lefttop+1;
      rightbot = righttop; righttop = righttop+1;
    end
    leftbot  = lefttop;  lefttop  = lefttop+1;
    rightbot = righttop; righttop = righttop+1;
  end

  int1 = 5:(block+4);
  int2 = (int1(block)+1):(int1(block)+block);
  int3 = (int2(block)+1):(int2(block)+block);
  int4 = (int3(block)+1):(int3(block)+block);
  TStruct  = zeros(4*block+4, DD^2);

  for i=1: NumVertex
    vert = Data(1,i);
    row  = (block*(vert-1)+1):(block*vert);
    TStruct(int1, Dom(1,i)) = row';
    TStruct(int3, Dom(2,i)) = row';
    vert = Data(2,i);
    row  = (block*(vert-1)+1):(block*vert);
    TStruct(int1, Dom(3,i)) = row';
    TStruct(int3, Dom(4,i)) = row';
    vert = Data(3,i);
    row  = (block*(vert-1)+1):(block*vert);
    TStruct(int2, Dom(1,i)) = row';
    TStruct(int4, Dom(3,i)) = row';
    vert = Data(4,i);
    row  = (block*(vert-1)+1):(block*vert);
    TStruct(int4, Dom(4,i)) = row';
    TStruct(int2, Dom(2,i)) = row';
    TStruct(2,Dom(1,i))=i;
    TStruct(3,Dom(2,i))=i;
    TStruct(1,Dom(3,i))=i;
    TStruct(4,Dom(4,i))=i;
  end

piece = 5:(4*block+4);
addto = TStruct(piece, :)~=0;
TStruct(piece,:) = TStruct(piece,:) + NumVertex.*addto;

N  = block+2;
vv = [N N^2 (N^2-N+1) 1];
s1 = [(2*N):N:(N^2-1)];
s2 = [(N^2-N+2):(N^2-1)];
s3 = [(1+N):N:(N^2-N)];
s4 = [2:(N-1)];
Bound = [vv s1 s2 s3 s4];
Int = 1:N^2; Int(Bound)=[];

% end function

function g = getg(FKi, Ki, FStruct, TStruct, Bound,
Int, bf,  SchurLength)
```

```
  [q p]  = size(FStruct);
  D      = (1/2).*ones(size(Bound));
  D(1:4) = (1/4).*[1 1 1 1];

  g = zeros(SchurLength,1);
  for i=1:p
    Del      = find( FStruct(Bound,i)==0 );
    Boundary = Bound;  Boundary(Del)=[];
    Di       = D;      Di(Del)      =[];
    bsub     = FStruct( Boundary,i);
    isub     = FStruct( Int,i );
    fi       = bf(isub);
    fb       = Di'.*bf(bsub);
    Aii      = Ki(Int, Int);
    Aib      = Ki(Int, Boundary);
    Abb      = Ki(Boundary,Boundary);
    sub      = TStruct( find(TStruct(:,i) > 0), i);
    g(sub)   = g(sub) + fb - Aib'*( Aii\fi );
  end

% end function.


function [Ah, IntDom] = GetAh(Int, Bound, FStruct,
TStruct, SchurLength, DD, Ki)

  [q p]  = size(FStruct);
  D      = (1/2).*ones(size(Bound));
  D(1:4) = (1/4).*[1 1 1 1];
  Ah     = zeros((DD-2)^2);

  for i=1:p
    Del      = find( FStruct(Bound,i)==0 );
    if length(Del) ==0
      IntDom = [IntDom i];
    end
  end

  for i=1:length(IntDom)
    Rhs = zeros(SchurLength,1);
    sub = TStruct(find(TStruct(:,IntDom(i))> 0),
    IntDom(i) );
    Rhs(sub) = D';
    Vec = smult(Ki, FStruct, TStruct, Bound, Int, Rhs);
    for j=1:length(IntDom)
      sub = TStruct(find(TStruct(:,IntDom(j)) > 0),
                    IntDom(j) );
      Ah(j,i) = D*Vec(sub);
    end
  end

% end function.


function Sy = smult(Ki, FStruct,TStruct, Bound, Int, y)

  [q p] = size(FStruct);
  Sy    = zeros(size(y));
  for i=1:p
    Del      = find( FStruct(Bound,i)==0 );
    Boundary = Bound;  Boundary(Del)=[];
    Kii      = Ki(Int, Int);
    Kbb      = Ki(Boundary, Boundary);
    Kib      = Ki(Int, Boundary);
    sub      = TStruct( find(TStruct(:,i) >0), i);
    X        = Kii\(Kib*y(sub));
    Sy(sub)  = Sy(sub) + Kbb*y(sub) - Kib'*X;
  end

% end function.


function [Ub, iter] = PCG3(FKi, Ki, g, PreconEps, Ah,
FStruct, TStruct, Bound, Int, Precon, IntDom)

  Xk = zeros(size(g));
  Rk = g;
  Sk=zeros(size(Rk));

% solve using precon.
  Rk = balresid(Ah, FKi, Ki, IntDom, FStruct,
                TStruct, Bound, Int, Rk, Precon);
  Sk = applyprecon(Ah, FKi, Ki, IntDom, FStruct,
                TStruct, Bound, Int, Rk, Precon);
  Pk = Sk;
  Gk = Rk'*Sk;
  iter = 0
  while ( sqrt(abs( (Rk'*Rk)/(g'*g)) ) > PreconEps )
    % A mult
    Qk = smult(Ki, FStruct, TStruct, Bound, Int, Pk);
    Tk = Pk'*Qk;
    alphak = Gk/Tk;
    Xk = Xk + alphak.*Pk;
    Rk = Rk - alphak.*Qk;
    %solve using precon.
    Sk = applyprecon(Ah, FKi, Ki, IntDom, FStruct,
                  TStruct, Bound, Int, Rk, Precon);
    Gk1= Gk;
    Gk = Rk'*Sk;
    betak = Gk/Gk1;
    Pk = Sk + betak.*Pk;
    iter = iter+1
    sqrt(abs( (Rk'*Rk)/(g'*g)))
  end

  Ub = Xk;

% End of function.


function s = Balresid(Ah, FKi, Ki, IntDom, FStruct,
TStruct, Bound, Int, bf,  Precon)


  D      = (1/2).*ones(size(Bound));
  D(1:4) = (1/4).*[1 1 1 1];
  Len    = length(IntDom);
  rbar = zeros(Len,1);
  for i=1:Len
    sub      = TStruct(
    find(TStruct(:,IntDom(i)) > 0), IntDom(i));
    rbar(i) = D*bf(sub);
  end
  Lamda = Ah\rbar;
  U = zeros(size(bf));
  for i=1:Len
    sub    = TStruct(
    find(TStruct(:,IntDom(i)) > 0), IntDom(i));
    U(sub) = U(sub) + D'.*Lamda(i);
  end
  s = bf - smult(Ki, FStruct, TStruct, Bound, Int, U);

% end function.


function soln = ApplyPrecon(Ah, FKi, Ki, IntDom,
FStruct, TStruct, Bound, Int, bf, Precon)

  [q p]  = size(FStruct);
  soln   = zeros(size(bf));
  D      = (1/2).*ones(size(Bound));
  D(1:4) = (1/4).*[1 1 1 1];
  Len    = length(IntDom);
  rbar = zeros(Len,1);
  for i=1:Len
    sub      = TStruct(
    find(TStruct(:,IntDom(i)) > 0), IntDom(i));
    rbar(i) = D*bf(sub);
  end

  Lamda = Ah\rbar;
  U = zeros(size(bf));
  for i=1:Len
    sub    = TStruct(
    find(TStruct(:,IntDom(i)) > 0), IntDom(i));
    U(sub) = U(sub) + D'.*Lamda(i);
  end
  s = bf - smult(Ki, FStruct, TStruct, Bound, Int, U);

  for i=1: p
    Del      = find( FStruct(Bound,i)==0 );
    Boundary = Bound;  Boundary(Del)=[];
    Di       = D;      Di(Del)      =[];
    sub      = TStruct( find(TStruct(:,i) > 0), i);
    A        = Ki([Int Boundary],[IntBoundary]);
    len1     = length(Int);
    len2     = length(Boundary);
    len      = len1+len2;
    Rhs      = zeros(len,1);
    Rhs( (len1+1):len) = s(sub).*Di';
    [fq fr ] = qr(full(A));
    if (length(Bound) == length(Boundary))
    fr(length(fr),length(fr))=0.01; end
    TSoln    = fr\(fq'*Rhs);
    soln(sub)= soln(sub) + (TSoln((len1+1):len )).*Di';
  end
```

```
  Su = smult(Ki, FStruct, TStruct, Bound,Int, soln);
  corr = zeros(size(rbar));
  for i=1: Len
    sub      = TStruct(find(TStruct(:,IntDom(i)) > 0),
               IntDom(i));
    corr(i) = D*Su(sub);
  end
  mu = Ah\(rbar - corr);
  for i=1: Len
    sub      = TStruct(find(TStruct(:,IntDom(i)) > 0),
    IntDom(i));
    soln(sub)= soln(sub) + mu(i).*D';
  end

% end function.


% Program BPS

% Main program which implements T_{is} for the SEM and
FEM

for spacing =1:2
direct     = 1;
DD         = [ 7 7 7 7  7  7 ];
NN         = [ 6 7 8 9 10 12 ];
runsem     = 0;

iter=[];
t0 = clock;
for i = 1 : length(DD)
  for j = NN(i) : NN(i)
    NDx = DD(i);
    N   = j;
    scale = 1;
    sh
    results1= [results1; [NDx N iter est]]
  end
end
end

disp('DD   N   iters');
results1

% Program Sh

% Constants:
  NDy          = NDx;
  X1           = 0;
  X2           = 1.
  Y1           = 0;
  Y2           = 1;
  NumNodes     = ( NDy*(N+1)  -(NDy-1) )*( NDx*(N+1) -
                 (NDx-1));
  NumElements  = NDy*NDx;
  NumVertex    = (NDx-1)*(NDy-1);
  NInterior    = (N-1)^2;
  SchurLength  = (NDx-1)*( NDy*(N+1)  -(NDy-1)-2 ) +
                 (NDx-1)*(NDx*(N-1) );
  overlap      = 1;  % make calling routines compatable;

% ---------- Preprocessor ----------

  cd h:\addsh-h
  [Ah, gh, Vertex,  Interface, HBoundary, Interior ] =
  bifemlib(NDx, 1);
  Ah = scale.*Ah;
  [PA, Pb, PVertex, PInterface, PBoundary, PInterior,
  Struct1, FKi, FPKi] = shhlib(N*NDx, NDx, spacing ,
                               overlap);
  cd h:\sem
  if runsem==1
    [A, b, Vertex, Interface, Boundary, Interior, SKi,
    SPKi ] = semlib(N*NDx, NDx);
  end
  cd h:\bps

  Ah(HBoundary,:)=[];  Ah(:,HBoundary)=[];
  gh(HBoundary)=[];
  Hmap = 1: (NDx+1)^2; Hmap(HBoundary)=[];
  Pmap = 1:NumNodes;   Pmap(PBoundary) =[];
  PA = PA(Pmap,:);
  PA = PA(:,Pmap);
  Pb = Pb(Pmap);
  clear Pmap;

  cd h:\libs
  Course = getcourse(NDx);
  [Rt, TStruct, Int, Bound] = getrt(N-1,
  length(PInterface),length(PVertex), Course, spacing,
  NDx);
  cd h:\bps

if runsem==1
  [g, schur1]  =  getg(SKi, SKi, Struct1, TStruct,
  Bound, Int, Pb,  SchurLength, N);
  [g1, schur]  =  getg(FKi, FKi, Struct1, TStruct,
  Bound, Int,Pb,  SchurLength, N);
  clear Pb;
else
  [g, schur]   =  getg(FKi, FKi, Struct1, TStruct,
  Bound, Int,Pb,  SchurLength, N);
  clear Pb;
end

% Ap = precon, A = matrix.
  if runsem==1
    [Uf, iter, est] = pcg3(FKi, SPKi, SKi, g, 0.0001,
    Ah, Rt, Struct1, TStruct, Bound, Int, Precon, NDx,
    N, schur);
  else
    [Uf, iter, est] = pcg3(FKi, FPKi, FKi, g, 0.0001,
    Ah, Rt, Struct1, TStruct, Bound, Int, Precon, NDx,
    N, schur);
  end

% ---------- End of program ----------


function [g, schur] = getg(FKi, Ki, FStruct, TStruct,
Bound, Int, bf,  SchurLength, N)

Aii    = Ki(Int, Int);
Aib    = Ki(Int, Bound);
Abb    = Ki(Bound,Bound);
schur  = Abb - Aib'*(Aii\Aib);
schur  = 2.*schur( 5:(5+N-2), 5:(5+N-2) );

% end function.


function [Ub, iter, est] = PCG3(FKi, PKi, Ki, g,
PreconEps, Ah, Rt, FStruct, TStruct, Bound, Int,
Precon, DD, N, schur)

   Jn = schur;
   for i=1:4
      index = [1+(i-1)*(N-1): i*(N-1)];
      Bn(index,index) = Jn;
   end

  Xk = zeros(size(g));
  Rk = g;
  Sk=zeros(size(Rk));

% solve using precon.
  Sk = applyprecon(Ah, FKi, PKi, Ki, Rt, FStruct,
                   TStruct,Bound, Int, Rk, Bn);
  Pk = Sk;
  Gk = Rk'*Sk;
  iter = 0
  while ( sqrt(abs( (Rk'*Rk)/(g'*g)) ) > PreconEps )
    % A mult - subdomain solves
    Qk = smult(Ki, FStruct, TStruct, Bound, Int, Pk);
    Tk = Pk'*Qk;
    alphak = Gk/Tk;
    Xk = Xk + alphak.*Pk;
    Rk = Rk - alphak.*Qk;
    %solve using precon.
    Sk = applyprecon(Ah, FKi, PKi, Ki, Rt, FStruct,
                     TStruct,Bound, Int, Rk, Bn);
    Gk1= Gk;
    Gk = Rk'*Sk;
    betak = Gk/Gk1;
    Pk = Sk + betak.*Pk;
    iter = iter+1
    sqrt(abs( (Rk'*Rk)/(g'*g)))
  end
Ub = Xk;

% End of function.


function soln = ApplyPrecon(Ah, FKi, PKi, Ki, Rt,
FStruct, TStruct, Bound, Int, bf,  Bn)
```

```
% -----setup-----

   len1 = length(Ah);
   Vert = 1:len1;
   bhe  = bf; bhe(Vert)=[];
   bhv  = bf(Vert);
   bh   = bhv + Rt'*bhe;


% ---------solves----------

   hsoln = zeros(size(bh));
   hsoln = Ah\bh;
   [q p] = size(FStruct);
   fsoln = zeros(size(bf));
   TStruct(1:4,:) = zeros(size(TStruct(1:4,:)));
   for i=1:p
     Del      = find( FStruct(Bound,i)==0 );
     Del      = [Del; [1:4]'];
     Boundary = Bound;
     Boundary(Del)=[];
     extract  = 1:length(Boundary);
     Bu       = Bn(extract,extract);
     sub      = TStruct( find(TStruct(:,i) > 0), i);
     fsoln(sub) =  Bu\bf(sub);
   end

% ---------restore----------
   soln = fsoln +  [ hsoln ;Rt*hsoln];

% end function.


% Program GASM

% Main program which implements the global
ASM for the SF'I and SEM

DD         = [7];
NN         = [4];
spacing    = 2;
overlap    = 2;
semfem     = 1;

iter=[];
iter1=[];
t0 = clock;
for i = 1 : length(DD)
  for j = NN(i) : NN(i)
    NDx = DD(i);
    N   = j;
    sh
    results1= [results1; [NDx N iter iter1]]
  end
end
disp('DD   N   iters');
results1

%function [iter, history] = Sh(NDx, N,spacing)

% Constants:
   NDy          = NDx;
   X1           = 0;
   X2           = 1;
   Y1           = 0;
   Y2           = 1;
   NumNodes     = ( NDy*(N+1)  -(NDy-1) )*(NDx*(N+1) -
                  (NDx-1) );
   NumElements  = NDy*NDx;
   NumVertex    = (NDx-1)*(NDy-1);
   NInterior    = (N-1)^2;
   history      = zeros(1, 20);

% --------- Preprocessor ----------------

[Ah, gh, Vertex, Interface, HBoundary,Interior, HNodes]
= bifemlib(NDx, 1);

 Ah(HBoundary,:)=[];  Ah(:,HBoundary)=[];
 gh(HBoundary)=[];
 Hmap = 1:  (NDx+1)^2; Hmap(HBoundary)=[];

 [A, b, FVertex, FInterface, FBoundary,FInterior,
 Struct1, Struct2] = shhlib(N*NDx, NDx, spacing ,
                            overlap);
 map = 1:NumNodes; map(FBoundary)=[];
 A = A(map,:);
 A = A(:,map);
 b = b(map);

Rht = getrt(N-1,  length(Interface), length(Vertex),
           spacing, Struct1);

if (semfem==1)
    cd h:\sem
    [B, b, Vertex, Interface, Boundary, Interior ] =
    semlib(N*NDx, NDx);
    cd h:\addsh-h
    map = 1:NumNodes; map(Boundary)=[];
    B = B(map,:);
    B = B(:,map);
    b = b(map);
end

%A = precon, B = matrix.
if semfem==1
 [Uf, history, iter iter1] = pcg3(A, B, b, 0.0001, Ah,
 Rht, Struct2);
else
  [Uf, history, iter iter1] = pcg3(A ,A, b,0.0001, Ah,
  Rht, Struct2);
end

% ----- End of program -------------


function [Ub, history, iter, iiter] = PCG3(A, Bu, g,
          PreconEps, Ah, Rht, FStruct)

  Xk = zeros(size(g));
  Rk = g;
  Sk=zeros(size(Rk));
  %solve using precon.
  [Sk, prev, iiter] = applyprecon(Ah, A,Rht, FStruct,
  Rk, prev);
  Pk = Sk;
  Gk = Rk'*Sk;
  iter = 0;
  history(1) = 1;

  while ( sqrt(abs( (Rk'*Rk)/(g'*g)) ) >PreconEps )
    % A mult - subdomain solves
    Qk = Bu*Pk;
    Tk = Pk'*Qk;
    alphak = Gk/Tk;
    Xk = Xk + alphak.*Pk;
    Rk = Rk - alphak.*Qk;
    %solve using precon.
    [Sk,  prev, iiter] = applyprecon(Ah, A,
    Rht, FStruct, Rk, prev);
    Gk1= Gk;
    Gk = Rk'*Sk;
    betak = Gk/Gk1;
    Pk = Sk + betak.*Pk;
    iter = iter+1;
    sqrt(abs( (Rk'*Rk)/(g'*g)))
  end
  Ub = Xk;

% End of function.

functi.  [soln, prev, iter] = ApplyPrecon(Ah, Af, Rht,
                                FStruct, Vp, prev)

% ---------setup-----------
   bf = Vp;
   bh = Rht'*Vp;
% ----------solves----------
  hsoln = zeros(size(bh));
  hsoln = Ah\bh;
  [q p] = size(FStruct);
  fsoln = zeros(size(bf));

  for i=1:p
    sub = FStruct( find(FStruct(:,i) > 0),i);
    fsoln(sub) = fsoln(sub) + (Af(sub,sub))\bf(sub);
  end
%------------restore-------
soln = Rht*hsoln + fsoln;

% end function.
```

References

[1] G. Anagnostou, Y. Maday, C. Mavriplis, and A. T. Patera, *On the mortar element method: Generalizations and implementation*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., 1990, pp. 157–173.

[2] I. Babuška, A. Craig, J. Mandel, and J. Pitkäranta, *Efficient preconditioning for the p-version finite element method in two dimensions*, SIAM J. Numer. Anal., 28 (1991), pp. 624–661.

[3] I. Babuška and H. C. Elman, *Some aspects of parallel implementation of the finite-element method on message passing architectures*, J. Comp. App. Math., 27 (1989), pp. 157–187.

[4] I. Babuška, H. C. Elman, and K. Markley, *Parallel implementation of the hp-version of the finite element method on a shared-memory architecture*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1433–1459.

[5] I. Babuška, M. Griebel, and J. Pitkäranta, *The problem of selecting the shape functions for a p-type finite element*, Int. J. Numer. Meth. Eng., 28 (1989), pp. 1891–1908.

[6] I. Babuška and M. Suri, *The p and h-p versions of the finite element method, an overview*, Comp. Meth. App. Mech. Eng., 80 (1990), pp. 5–26.

[7] I. Babuška, B. A. Szabo, and I. N. Katz, *The p-version of the finite element method*, SIAM J. Numer. Anal., 18 (1981), pp. 515–545.

[8] R. Barrett et al., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, 1993.

[9] F. Ben Belgacem and Y. Maday, *A spectral element methodology tuned to parallel implementations*, tech. report, Laboratoire d'Analyse Numerique, University Pierre et Marie Curie, Paris, 1992.

[10] C. Beque, C. Bernardi, N. Debit, Y. Maday, G. Kariadakis, C. Mavriplis, and A. Patera, *Nonconforming spectral element - finite element approximations for partial differential equations*, Comp. Meth. App. Mech. Eng, 75 (1989), pp. 109–123.

[11] C. Bernardi, N. Debit, and Y. Maday, *Coupling finite element and spectral methods: First results*, Math. Comp., 54 (1990), pp. 21–39.

[12] C. Bernardi and Y. Maday, *Approximation results for spectral methods with domain decomposition*, App. Numer. Math., 6 (1990), pp. 33–52.

[13] C. Bernardi, Y. Maday, and A. T. Patera, *A new nonconforming approach to domain decomposition: the mortar element method*, in Nonlinear Partial Differential Equations and their Applications, H. Brezis and J. L. Lions, eds., Pitman and Wiley, 1992.

[14] C. BERNARDI, Y. MADAY, AND G. SACCHI-LANDRIANI, *Nonconforming matching conditions for coupling spectral and finite element methods*, App. Numer. Math., 6 (1990), pp. 65–84.

[15] P. E. BJØRSTAD, *Multiplicative and additive Schwarz methods: Convergence in the two-domain case*, in Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., 1989, pp. 147–159.

[16] P. E. BJØRSTAD AND J. MANDEL, *On the spectra of sums of orthogonal projections with applications to parallel computing*, BIT, 31 (1991), pp. 76–88.

[17] P. E. BJØRSTAD AND M. D. SKOGEN, *Domain decomposition algorithms of Schwarz type, designed for massively parallel computers*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[18] P. E. BJØRSTAD AND O. B. WIDLUND, *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, SIAM J. Numer. Anal., 23 (1986), pp. 1097–1120.

[19] J. F. BOURGAT, R. GLOWINSKI, P. LE TALLEC, AND M. VIDRASCU, *Variational formulation and algorithm for trace operator in domain decomposition calculations*, in Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., 1989, pp. 3–16.

[20] [illegible] LE, J. E. PASCIAK, AND A. H. SCHATZ, *The construction of preconditioners for [illegible] by substructuring I*, Math. Comp., 47 (1986), pp. 103–134.

[21] ——, *[illegible] method for elliptic problems on regions partitioned into substructures*, Math. Comp., 46 (1986), pp. 361–369.

[22] F. BREZZI, *On the existence, uniqueness and approximations of saddle-point problems arising from Lagrangian multipliers*, RARIO Anal. Numer., 8 (1974), p. 129.

[23] X. CAI, *An additive Schwarz algorithm for nonselfadjoint elliptic equations*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., 1990, pp. 232–244.

[24] ——, *An optimal two-level overlapping domain decomposition method for elliptic problems in two and three dimensions*, SIAM J. Sci. Comput, 14 (1993), pp. 239–247.

[25] X. CAI, W. D. GROPP, AND D. E. KEYES, *A comparison of some domain decomposition algorithms for nonsymmetric elliptic problems*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[26] ——, *Convergence rate estimate for a domain decomposition method*, Numer. Math., 61 (1992), pp. 153–169.

[27] X. CAI AND O. B. WIDLUND, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Stat, 13 (1992), pp. 243–259.

[28] ——, *Multiplicative Schwarz algorithms for some nonsymmetric and indefinite problems*, SIAM J. Num. Anal., 30 (1993).

[29] C. CANUTO AND D. FUNARO, *The Schwarz algorithm for spectral methods*, SIAM J. Numer. Anal., 25 (1988), pp. 24–40.

[30] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1988.

[31] C. Carlenzoli, A. Quarteroni, and A. Valli, *Spectral domain decomposition methods for compressible Navier-Stokes equations*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[32] T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., *Second International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, 1989.

[33] ——, eds., *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, 1990.

[34] ——, eds., *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, 1992.

[35] T. F. Chan and D. Goovaerts, *Schur complement domain decomposition algorithms for spectral methods*, App. Numer. Math., 6 (1990), pp. 53–64.

[36] T. F. Chan and D. E. Keyes, *Interface preconditionings for domain decomposed convection-diffusion operators*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., 1990, pp. 245–262.

[37] T. F. Chan and T. Mathew, *The interface probing technique in domain decomposition*, Tech. Report 91-62, Department of Mathematics, University of California, Los Angeles, February 1991.

[38] T. F. Chan and T. P. Mathew, *An application of the probing technique to the vertex space method in domain decomposition*, in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, Y. A. Kuznetsov, G. Meurant, J. Périaux, and O. B. Widlund, eds., 1991, pp. 101–111.

[39] T. F. Chan, T. P. Mathew, and J. Shao, *Fourier and probe variants of the vertex space domain decomposition algorithm*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[40] T. F. Chan, T. P. Mathew, and J.-P. Shao, *Efficient variants of the vertex space domain decomposition algorithm*, tech. report, Department of Mathematics, University of California, Los Angeles, January 1992.

[41] H. Cheng, *Iterative Solution of Elliptic Finite Element Problems on Partially Refined Meshes and the Effect of Using Inexact Solvers*, PhD thesis, Courant Institute of Mathematical Sciences, New York University., October 1993.

[42] P. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, 1978.

[43] L. C. Cowsar, J. Mandel, and M. F. Wheeler, *Balancing domain decomposition for mixed finite elements*, Tech. Report 93-08, Department of Mathematical Sciences, Rice University, March 1993.

[44] Y. 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. A. Kuznetsov, G. Meurant, J. Périaux, and O. B. Widlund, eds., 1991, pp. 112–128.

[45] P. DEMARET AND M. O. DEVILLE, *Chebyshev pseudospectral solution of the Stokes equations using finite element preconditioning*, J. Comp. Phy., 83 (1989), pp. 463–484.

[46] L. DEMKOWICZ, J. T. ODEN, W. RACHOWICZ, AND O. HARDY, *Toward a universal h-p adaptive finite element strategy, part 1. constrained approximation and data structure*, Comp. Meth. App. Mech. Eng., 77 (1989), pp. 79–112.

[47] M. DEVILLE AND E. MUND, *Chebyshev pseudospectral solution of second-order elliptic equations with finite element preconditioning*, J. Comp. Phy., 60 (1985), pp. 517–533.

[48] M. O. DEVILLE AND E. H. MUND, *Finite-element preconditioning for pseudospectral solutions of elliptic problems*, SIAM J. Sci. Stat Comput., 11 (1990), pp. 311–342.

[49] M. O. DEVILLE, E. H. MUND, AND A. T. PATERA, *Iterative solution of isoparametric spectral element equations by low order preconditioning*, J. Comp. App. Math., 20 (1987), pp. 189–197.

[50] P. DEVLOO, J. T. ODEN, AND P. PATTANI, *An h-p adaptive finite element method for the numerical simulation of compressible flow*, Comp. Meth. App. Mech. Eng., 70 (1988), pp. 203–235.

[51] M. DRYJA, *A capacitance matrix method for Dirichlet problem on polygon region*, Numer. Math, 39 (1982), pp. 51–64.

[52] M. DRYJA, B. F. SMITH, AND O. B. WIDLUND, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, Tech. Report 638, Courant Institute of Mathematical Sciences. New York University., June 1993.

[53] M. DRYJA AND O. B. WIDLUND, *Multilevel additive methods for elliptic finite element problems*, Tech. Report 507, Courant Institute of Mathematical Sciences. New York University., 1990.

[54] ——, *Towards a unified theory of domain decomposition algorithms for elliptic problems*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., 1990, pp. 3–21.

[55] ——, *Additive Schwarz methods for elliptic finite element problems in three dimensions*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[56] ——, *Domain decomposition algorithms with small overlap*, Tech. Report 606, Courant Institute of Mathematical Sciences. New York University., 1992.

[57] ——, *Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems*, Tech. Report 626, Courant Institute of Mathematical Sciences. New York University., March 1993.

[58] I. S. DUFF, A. M. ERISMAN, AND J. K. REID, *Direct Methods for Sparse Matrices*, Claridon Press, 1986.

[59] C. FARHAT, *A saddle-point principle domain decomposition method for the solution of solid mechanics problems*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[60] P. F. FISCHER AND A. T. PATERA, *Parallel spectral element solution of the Stokes problem*, J. Comput. Phy., 92 (1991), pp. 380–421.

[61] P. F. FISCHER AND E. M. RØNQUIST, *Spectral element methods for large scale parallel Navier-Stokes calculations*, in Second Int. Conf. on Spectral and High Order Methods for PDE's. Proceedings of ICOSAHOM 92. To appear in Comput. Meth. Appl. Mech. Eng., 1993.

[62] P. F. Fischer, E. M. Rønquist, D. Dewey, and A. T. Patera, *Spectral element methods: Algorithms and architectures*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., 1988.

[63] P. F. Fischer, E. M. Rønquist, and A. T. Patera, *Parallel spectral element methods for viscous flow*, in Parallel Supercomputing Methods, Algorithms and Applications, G. Carey, ed., John Wiley and Sons, 1989, pp. 223–237.

[64] M. Fortin and R. Aboulaich, *Schwarz's decomposition method for incompressible flow 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., 1988, pp. 333–349.

[65] M. Fortin, R. Aboulaich, C. Kari, and F. Pitt, *Domain decomposition methods for the Navier-Stokes equations*, in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, Y. A. Kuznetsov, G. Meurant, J. Périaux, and O. B. Widlund, eds., 1991, pp. 310–318.

[66] D. Funaro, *Domain decomposition methods for pseudo-spectral approximations part 1. second order equations in one dimension*, Numer. Math., 52 (1988), pp. 329–344.

[67] R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, 1988.

[68] R. Glowinski, Y. A. Kuznetsov, G. Meurant, J. Périaux, and O. B. Widlund, eds., *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, 1991.

[69] G. Golub and C. van Loan, *Matrix Computations*, The Johns Hopkins University press, 2nd ed., 1989.

[70] D. Gottlieb and R. S. Hirsh, *Parallel pseudo-spectral domain decomposition techniques*, J. Sci. Comput., 4 (1989), pp. 309–325.

[71] L. Hageman and D. Young, *Applied Iterative Methods*, Academic Press, 1981.

[72] W. Heinrichs, *Splitting techniques for the pseudospectral approximation of the unsteady Stokes equations*, SIAM J. Numer. Anal., 30 (1993), pp. 19–39.

[73] R. Henderson and G. E. Karniadakis, *Hybrid spectral element methods for flows over rough walls*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[74] R. D. Henderson and G. E. Karniadakis, *Unstructured spectral element methods for incompressible Navier-Stokes equations*, in 8th Int. Conf. on Finite Element Methods in Fluids, 1992.

[75] G. E. Karniadakis, *Spectral element simulations of laminar and turbulent flows in complex geometries*, Appl. Num. Math., 6 (1990), pp. 85–105.

[76] D. E. Keyes and W. D. Gropp, *A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 166–202.

[77] K. Z. Korczak and A. T. Patera, *An isoparametric spectral element method for solution of the Navier-Stokes equations in complex geometry*, J. Comput. Phy., 62 (1986), pp. 361–382.

[78] P. Le Tallec, Y. H. De Roeck, and M. Vidrascu, *Domain decomposition methods for large linearly elliptic three-dimensional problems*, J. Comput. Appl. Math., 34 (1991), pp. 93–117.

[79] P. Le Tallec and T. Sassi, *Domain decomposition with nonmatching grids*, tech. report, University of Paris-Dauphine, CEREMADE, 1993.

[80] P. Le Tallec and J. Sousa Rodrigues, *Domain decomposition method with nonmatching grids applied to fluid dynamics*, tech. report, University of Paris-Dauphine, 1993.

[81] P. L. 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., 1988, pp. 1–42.

[82] Y. Maday, C. Mavriplis, and A. T. Patera, *Nonconforming mortar element methods: Application to spectral discretizations*, in Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., 1989, pp. 392–418.

[83] Y. Maday, D. Meiron, A. T. Patera, and E. M. Rønquist, *Analysis of iterative methods for the steady and unsteady Stokes problem: Application to spectral element discretizations*, SIAM J Sci. Comput, 14 (1993), pp. 310–337.

[84] Y. Maday and R. Munoz, *Spectral element multigrid II. Theoretical justification*, J. Sci. Comput., 3 (1988), pp. 323–353.

[85] Y. Maday and A. T. Patera, *Spectral element methods for the incompressible Navier-Stokes equations*, in State-of-the-art Surveys on Comput. Mech., A. K. Noor and J. T. Oden, eds., ASME, 1989, pp. 71–143.

[86] Y. Maday, A. T. Patera, and E. M. Rønquist, *An operator-integration-factor splitting method for time-dependent problems: Application to incompressible fluid flow*, J. Sci. Comput., 5 (1990), pp. 263–292.

[87] Y. Maday and E. M. Rønquist, *Optimal error analysis of spectral methods with emphasis on non-constant coefficients and deformed geometries*, Comput. Meth. Appl. Math. Eng., 80 (1990), pp. 91–115.

[88] J. Mandel, *Iterative solvers by substructuring for p-version finite element method*, Comp. Meth. Appl. Mech. Eng., 80 (1990), pp. 117–128.

[89] ——, *Two-level domain decomposition preconditioning for the p-version finite element method in three dimensions*, Int. J. Numer. Meth. Eng., 29 (1990), pp. 1095–1108.

[90] ——, *Adaptive iterative solvers in finite elements*, in Solving Large Scale problems in Mechanics, M. Papadrakakis, ed., John Wiley and Sons, 1993.

[91] ——, *Balancing domain decomposition*, Communications in Numerical Methods in Engineering, 9 (1993), pp. 233–241.

[92] ——, *Hybrid domain decomposition with unstructured subdomains*, in Sixth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Y. A. Kuznetsov, J. Périaux, A. Quarteroni, and O. B. Widlund, eds., AMS, 1993, to appear.

[93] J. Mandel and M. Brezina, *Balancing domain decomposition: Theory and performance in two and three dimensions*, tech. report, Computational Mathematics Group. University of Colorado at Denver, March 1993.

[94] L. MANSFIELD, *Solution of the Stokes problem on distributed-memory multiprocessors*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[95] T. P. MATHEW, *Schwarz alternating and iterative refinement methods for mixed formulations of elliptic problems, part II: Convergence theory*, Tech. Report 92-05, Department of Mathematics, University of California, Los Angeles, January 1992.

[96] ——, *Schwarz alternating and iterative refinement methods for mixed formulations of elliptic problems, part I: Algorithms and numerical results*, Tech. Report 92-04, Department of Mathematics. University of California, Los Angeles, January 1992.

[97] C. MAVRIPLIS, *Adaptive mesh strategies for the spectral element method*, Tech. Report 92-36, ICASE, Langley Research Center, 1992.

[98] G. MEURANT, *Incomplete domain decomposition preconditioners for nonsymmetric problems*, in Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., 1989, pp. 219–225.

[99] Y. MORCHOISNE, *Inhomogeneous flow calculations by spectral methods: mono-domain and multi-domain techniques*, in Spectral methods for Partial Differential Equations, SIAM, 1984, pp. 181–208.

[100] S. A. ORSZAG, *Spectral methods for problems in complex geometries*, J. Comput. Phy., 37 (1980), pp. 70–92.

[101] J. E. PASCIAK, *Two domain decomposition techniques for Stokes problems*, in Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., 1989, pp. 419–430.

[102] A. T. PATERA, *A spectral element method for fluid dynamics: Laminar flow in a channel expansion*, J. Comput. Phy., 54 (1984), pp. 468–488.

[103] ——, *Fast direct Poisson solvers for high-order finite element discretizations in rectangularly decomposable domains*, J. Comput. Phy., 65 (1986), pp. 474–480.

[104] L. F. PAVARINO, *An additive Schwarz method for the p-version finite element method*, Tech. Report 580, Courant Institute of Mathematical Sciences. New York University., September 1991.

[105] ——, *Domain Decomposition Algorithms for the p-version Finite Element Method for Elliptic Problems*, PhD thesis, Courant Institute of Mathematical Sciences. New York University., 1992.

[106] L. F. PAVARINO AND O. B. WIDLUND, *Iterative substructuring methods for spectral elements in three dimensions*, Tech. Report 648, Courant Institute of Mathematical Sciences. New York University., October 1993.

[107] A. QUARTERONI, *Domain decomposition and parallel processing for the numerical solution of partial differential equations*, Surv. Math. Ind., 1 (1991), pp. 75–118.

[108] A. QUARTERONI, F. PASQUARELLI, AND A. VALLI, *Heterogeneous domain decomposition: Principles, algorithms, applications*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[109] A. QUARTERONI AND G. SACCHI-LANDRIANI, *Domain decomposition preconditioners for the spectral collocation method*, J. Sci. Comp., 3 (1988), pp. 45–76.

[110] A. Quarteroni and E. Zampieri, *A multilevel domain decomposition method for elliptic problems*, Tech. Report 730, I.A.N-C.N.R, 1989.

[111] E. M. Rønquist, *A domain decomposition method for elliptic boundary value problems: Application to unsteady incompressible fluid flow*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[112] ——, *An elliptic solver based on conjugate gradients, domain decomposition and deflation*, tech. report, Nektonics Inc. 875 Main Street, Cambridge, MA 02139, USA, 1993.

[113] E. M. Rønquist and A. T. Patera, *Spectral element multigrid I. Formulation and numerical results*, J. Sci. Comput., 2 (1987), pp. 389–406.

[114] U. Rüde, *Mathematical and Computational Techniques for Multilevel Adaptive Methods*, Frontiers in Applied Mathematics 13, SIAM, 1993.

[115] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[116] M. Sarkis, *Two-level Schwarz methods for nonconforming finite elements and discontinuous coefficients*, Tech. Report 629, Courant Institute of Mathematical Sciences. New York University., 1993.

[117] M. R. Schumack, W. W. Schultz, and J. P. Boyd, *Spectral method solution of the Stokes equations on nonstaggered grids*, J. Comput. Phy., 94 (1991), pp. 30–58.

[118] B. F. Smith, *Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity*, PhD thesis, Courant Institute of Mathematical Sciences. New York University., 1990.

[119] ——, *A domain decomposition algorithm for elliptic problems in three dimensions*, Numer. Math, 60 (1991), pp. 219–234.

[120] ——, *An iterative substructuring algorithm for problems in three dimensions*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[121] ——, *An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 364–378.

[122] ——, *A parallel implementation of an iterative substructuring algorithm for problems in three dimensions*, SIAM J. Sci. Comput., 14 (1993), pp. 406–423.

[123] B. F. Smith and O. B. Widlund, *A domain decomposition algorithm using a hierarchical basis*, SIAM J. Sci. Stat. Comput, 11 (1990), pp. 1212–1220.

[124] A. Wathen and D. Silvester, *Fast iterative solution of stabilised Stokes systems part 1: Using simple diagonal preconditioners*, SIAM J. Numer. Anal., 30 (1993), pp. 630–649.

[125] 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., 1988, pp. 113–128.

[126] ——, *Optimal iterative refinement methods*, in Second International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., 1989, pp. 114–126.

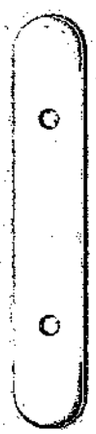
[127] ——, *Some Schwarz methods for symmetric and nonsymmetric elliptic problems*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[128] J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Review, 34 (1992), pp. 581–683.

[129] P. Zanolli, *Domain decomposition algorithms for spectral methods*, Tech. Report 491, Department of Mathematics, University of Cattolica di Brescia, 1985.

[130] X. Zhang, *Domain decomposition algorithms for the Biharmonic Dirichlet problem*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, J. S. Scroggs, and R. G. Voigt, eds., 1992.

[131] ——, *Multilevel Schwarz methods*, Num. Math., 63 (1992), pp. 521–539.





Author: Pahl Shannon S.

Name of thesis: Schwarz type domain decomposition methods for spectral element discretizations.

PUBLISHER:

University of the Witwatersrand, Johannesburg
©2015

LEGALNOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page)for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.