Parallel Numerical Methods for Partial Differential Equations (Ph.D. Thesis)

Sang-Bae Kim

Report Number:
93-090
PARALLEL NUMERICAL METHODS FOR
PARTIAL DIFFERENTIAL EQUATIONS

Sang-Bae Kim

CSD TR-93-090
December 1993
(Revised May 1994)
Parallel Numerical Methods for Partial Differential Equations

Sang-Bae Kim*

December 20, 1993

Abstract

It has been rightly predicted that parallel computing is inevitable. This thesis attempts to study and implement the so-called geometry splitting solution paradigm as a parallel computational framework for solving elliptic partial differential equations (PDEs) on distributed memory machines. First, we formulate, analyze and implement the Schwarz Alternating Method (SAM) for elliptic PDEs defined in one and multidimensional domains. Specifically, we analyze SAM methods whose convergence is controlled by a different parameter in each interface condition or overlapped domain. We derive both analytical and experimental results. Furthermore, we introduce a symmetric version of SAM and make useful observations about its convergence. Second, we implement four non-overlapping geometry splitting approaches based on finite element and difference techniques. One of them is formulated on the extended rectangular domain that encapsulates the given PDE domain and its corresponding grid. This encapsulation method assumes an extension of the PDE problem outside the specified domain of definition. This approach has reduced significantly the grid partitioning overhead without reducing the overall efficiency of the computation. Finally, we have parallelized the well-known ITPACK library and implemented it on the nCUBE II machines. All discretization and solution modules developed in this thesis have been integrated in the parallel ELLPACK environment and their performance has been extensively studied.

*Department of Computer Sciences, Purdue University, W. Lafayette, IN 47907
This work was supported by AFSOR grant F49620-92-J-0069 and NSF grants 9202536-CCR and 9123502-CDA.
TABLE OF CONTENTS

INTRODUCTION ................................................................. 1

1 Multi-Parameterized Schwarz Splitting for One-Dimensional Problems ........................................... 3
   1.1 Preliminary ................................................................. 3
      1.1.1 Schwarz Alternating Method (SAM) ......................... 4
      1.1.2 Generalization of SAM ........................................ 5
      1.1.3 Schwarz Splitting ............................................. 7
      1.1.4 Generalized Schwarz Splitting ............................... 10
   1.2 One-Parameter Schwarz Splitting .................................. 11
      1.2.1 Formulation of the Parameterized Schwarz Splitting .... 13
      1.2.2 Convergence Analysis ........................................ 16
      1.2.3 Determination of the Optimal Parameter .................... 19
   1.3 Multi-Parameter Schwarz Splitting ............................... 26
      1.3.1 Formulation of the Multi-Parameterized Schwarz Splitting 26
      1.3.2 Convergence Analysis ........................................ 27
      1.3.3 Determination of the Optimal Multi-Parameter Set ......... 31
   1.4 Numerical Experiments ............................................. 36
   1.5 Remarks and Discussion ............................................ 38

2 Multi-Parameterized Schwarz Splitting for Multi-Dimensional Problems ........................................... 40
   2.1 Formulation of the Multi-Parameterized Schwarz Splitting for Two-Dimensional Problem .................... 40
   2.2 Convergence Analysis ............................................. 42
   2.3 On the Determination of the Optimal Parameters .......... 43
   2.4 Numerical Experiments ............................................ 49
   2.5 Multi-Parameterized Schwarz Splitting for Higher ($p \geq 3$) Dimensional Problems ....................... 52
      2.5.1 Formulation of the Multi-Parameterized Schwarz Splitting 53
      2.5.2 Convergence Analysis ........................................ 53
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Symmetric Schwarz Splitting</td>
<td>55</td>
</tr>
<tr>
<td>3.1</td>
<td>Symmetric Schwarz Enhanced Matrix (SSEM) and Equation (SSEE)</td>
<td>55</td>
</tr>
<tr>
<td>3.2</td>
<td>Equivalence Theorems</td>
<td>57</td>
</tr>
<tr>
<td>3.3</td>
<td>Splittings of the Symmetric Schwarz Enhanced Matrices</td>
<td>59</td>
</tr>
<tr>
<td>4</td>
<td>Parallel PDE Discretization Techniques for MIMD Architectures</td>
<td>62</td>
</tr>
<tr>
<td>4.1</td>
<td>Domain Decomposition Based Parallel Discretization Procedures</td>
<td>62</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Parallel 5-point Star</td>
<td>67</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Parallel Encapsulated 5-point Star</td>
<td>68</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Parallel Linear Triangular Finite Element Method</td>
<td>73</td>
</tr>
<tr>
<td>4.2</td>
<td>Implementation of Parallel Finite Difference Discretization Modules</td>
<td>77</td>
</tr>
<tr>
<td>4.3</td>
<td>Implementation of Parallel Linear FEM Procedure</td>
<td>78</td>
</tr>
<tr>
<td>5</td>
<td>Parallel Stationary Iterative Methods for Distributed Memory Machines</td>
<td>83</td>
</tr>
<tr>
<td>5.1</td>
<td>Description of Parallel ITPACK Library</td>
<td>83</td>
</tr>
<tr>
<td>5.2</td>
<td>The Interface of the Parallel ITPACK Library</td>
<td>84</td>
</tr>
<tr>
<td>5.3</td>
<td>On the Parallelization of ITPACK Modules</td>
<td>85</td>
</tr>
<tr>
<td>5.3.1</td>
<td>The Parallelization of the Basic ITPACK Subtasks</td>
<td>86</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Red/Black Ordering for the Parallel ITPACK Modules</td>
<td>87</td>
</tr>
<tr>
<td>5.4</td>
<td>The Performance of Parallel ITPACK on the nCUBE II</td>
<td>87</td>
</tr>
<tr>
<td>5.4.1</td>
<td>The Performance of //ITPACK Modules for PDE Problems Defined on a Rectangle</td>
<td>87</td>
</tr>
<tr>
<td>5.4.2</td>
<td>The Performance of //ITPACK Modules for PDE Problems Defined on a General Domain</td>
<td>96</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td></td>
<td>107</td>
</tr>
</tbody>
</table>
INTRODUCTION

This thesis has three research thrusts:

1. to formulate and analyze matrix based decomposition techniques for solving elliptic boundary value problems in multi-dimensional spaces on sequential and parallel machines,

2. to implement and measure the performance of geometry based decomposition techniques for elliptic boundary value problems on distributed parallel machines,

3. to develop and implement a parallel version of ITPACK library and access its performance on the nCUBE II machines.

Numerical versions of the classical mathematical approach of the Schwarz Alternating Method (SAM), introduced by Schwarz a century ago, have been recently explored as parallel computational frameworks for the solution of initial/boundary value problems. These schemes are usually referred to as Schwarz Splittings (SS). One of the objectives of this research is to identify the proper parameters that speed the convergence of these methods and estimate their optimal values for a certain class of partial differential equations (PDEs). In Chapters 1 to 2 of this thesis, we formulate, analyze and implement SAM for a class of elliptic PDEs defined on one and several dimensions domains. This methods is based on a decomposition of the PDE domain into subdomains with overlapping domains. The original PDE problem is reduced to a set of smaller PDE problems on each subdomain whose solutions are coupled through some iterative scheme to produce the global solution. The method SAM can be formulated at a functional or matrix level. In this thesis we have adopted the later. One of the uncertainties in the numerical formulation of SAM that affects its convergence is the selection of the so called interface conditions (artificial boundary conditions). In the context of elliptic boundary value problems the most commonly used auxiliary conditions are of Dirichlet type. Recently, the effect of parameterized \( \alpha \) mixed interface conditions has been considered by a number of researchers. It was shown experimentally that an appropriate choice of the parameter \( \alpha \) relating the weights between the Dirichlet and the Neumann conditions allows one to optimize the convergence rates of the SAMs. In this thesis we complete the analysis of the above
observations and consider the multi-parameter mixed interface conditions where a different parameter ($\alpha_i$) is associated with the $i$-th overlapping area. In the case of one-dimensional boundary value problems, we were able to determine analytically the optimal values of $\alpha_i$'s which minimize the spectral radius of the block Jacobi iteration matrix associated with the SAM. Furthermore, we extend the formulation and some of the results in the two- and higher-than-two- dimensional cases. We also consider a Symmetric SAM as a variation of SAM in Chapter 3.

The second objective of this research is the realization of the SAM paradigm on certain class of computer architectures and its performance evaluation. Specifically, in Chapter 4, we formulate, implement, and analyze the performance of domain decomposition methods based on non-overlapping subdomains and finite element/difference discretization techniques. These parallel discretization framework assumes a decomposition of the corresponding grid/mesh into balanced subdomains with minimum interface nodal points. This optimal decomposition of the grid/mesh is an NP-complete problem. Thus, only heuristic algorithms are applied for its solution. In order to avoid the overhead of decomposition we have developed a new discretization approach which is based on a structure a grid or mesh that encapsulates the actual grid/mesh. This approach has been first formulated and tested on the finite difference equations. The advantage of this approach is that the decomposition of the grid is trivial. We experimentally study the performance of this approach and compare with the domain decomposition ones. The numerical results obtained so far indicate the superiority of the encapsulated approach. The software modules developed have been integrated in the parallel ELLPACK library.

One of the most computational intensive phases of the numerical solution of PDEs is the solution of the discrete equations. Significant number of papers have been written to address this part of the numerical simulation. Another objective of this research is to exploit the parallelism of stationary iterative methods towards the solution of the finite difference and element equations on MIMD machines. Specifically, in chapter 5 we address the parallelization of the well known iterative package ITPACK on distributed memory machines. Furthermore, we present an extensive set of performance data that indicate the almost optimal mappings of these computations on nCUBE II machines. Also, these software modules are available in the ELLPACK library.
Chapter 1
Multi-Parameterized Schwarz Splitting for One-Dimensional Problems

One of the uncertainties in the numerical formulation of *Schwarz Alternating Method (SAM)* that affects its convergence is the selection of the so-called interface conditions. In the context of elliptic boundary value problems, the most commonly used auxiliary conditions are of Dirichlet type. It turns out that the best choice of interface conditions for the SAM approach is an open problem.

In Section 1.1 we provide an introduction to the original SAM and its numerical version, Schwarz Splitting. In Sections 1.2 and 1.3, we consider the convergence properties of Schwarz Splitting with Jacobi-type iteration and parameterized mixed (Dirichlet and Neumann) interface conditions with one and several parameters. The One-Parameter \( \omega \) Schwarz Splitting with Jacobi iterations was considered in [Tan92]. In Section 1.2 we review this One-Parameter case and then derive explicit and implicit analytic expressions for the optimal value of \( \alpha \) involved. In Section 1.3 we formulate and analyze a Multi-Parameterized Numerical SAM framework whose mixed interface conditions in each subdomain are controlled by different parameters. Finally, Section 1.4 and 1.5 include some experimental results which verify the rate of convergence of these methods and a number of remarks.

1.1 Preliminary

SAM was first introduced in 1869 by Schwarz. Although SAM was originally suggested as a method for solving elliptic partial differential equations, it can be considered as a general methodology for problem solving. SAM has not attracted people for many decades because it was not computationally practical. It has been known that SAM converges slowly. However, it has begun to be re-explored since the advent of parallel computers. SAM has a high potential of parallelism, so it has been generalized in many respects. Among them, there is the so-called Schwarz Splitting as a discrete version of SAM (see, e.g. [Mil65], [Rod85], [RS84a], [RS84b], [RS85]).

We introduce the original SAM in Section 1.1.1, its generalization in Section 1.1.2, and its numerical version (Schwarz Splitting) in Section 1.1.3.
1.1.1 Schwarz Alternating Method (SAM)

Schwarz [Sch69] showed that, for a domain consisting of the union of two rectangular domains or disks, one could construct a sequence of solutions of the Laplace equation in the two subdomains which would converge to the solution of the Laplace equation in their union. His method is now called Schwarz Alternating Method. The description of a simple version of SAM is as follows.

Consider the Dirichlet boundary value problem for the second order linear elliptic partial differential equation

\[
\begin{align*}
Lu &= f \text{ in } \Omega, \\
u &= \psi \text{ on } \Gamma_\Omega
\end{align*}
\]  

(1.1)

where \( L \) is the elliptic operator, \( f, \psi \) are given functions, \( \Omega \) is a bounded domain in the two-dimensional space and \( \Gamma_\Omega \) is the boundary of \( \Omega \). Assume that the solution to problem (1.1) exists and is unique.

Following Schwarz's basic idea we decompose the domain \( \Omega \) into two overlapping subdomains \( \Omega_1 \) and \( \Omega_2 \). We assume that \( \Omega_{1,2} = \Omega_{2,1} = \Omega_1 \cap \Omega_2 \neq \emptyset \) and we denote by \( \Gamma_{\Omega_1}, \Gamma_{\Omega_2} \) the boundaries of \( \Omega_1, \Omega_2 \), respectively (see Figure 1.1). We define

\[
\begin{align*}
\Gamma_{1,1} &= \Gamma_{\Omega_1} \cap \Gamma_{\Omega_2}, \\
\Gamma_{2,2} &= \Gamma_{\Omega_2} \cap \Gamma_{\Omega_1}, \\
\Gamma_{1,2} &= \Gamma_{\Omega_1} \cap \Omega_2, \\
\Gamma_{2,1} &= \Gamma_{\Omega_2} \cap \Omega_1,
\end{align*}
\]  

(1.2)

so we have

\[
\begin{align*}
\Gamma_{\Omega_1} &= \Gamma_{1,1} + \Gamma_{1,2}, \\
\Gamma_{\Omega_2} &= \Gamma_{2,2} + \Gamma_{2,1}.
\end{align*}
\]  

(1.3)

We note that \( \Gamma_{1,2} \) and \( \Gamma_{2,1} \) are what are called the artificial boundaries of the subdomains \( \Omega_1 \) and \( \Omega_2 \), respectively. Based on the above decomposition we can formulate...
two coupled problems

\begin{align}
&\begin{cases}
Lu_1 = f & \text{in } \Omega_1, \\
u_1 = \psi & \text{on } \Gamma_1, \\
u_1 = u_2 & \text{on } \Gamma_2
\end{cases} \\
&\begin{cases}
Lu_2 = f & \text{in } \Omega_2, \\
u_2 = \psi & \text{on } \Gamma_2, \\
u_2 = u_1 & \text{on } \Gamma_1
\end{cases}
\end{align}

and

\begin{align}
&\begin{cases}
Lu_1 = f & \text{in } \Omega_1, \\
u_1 = \psi & \text{on } \Gamma_1, \\
u_1 = u_2 & \text{on } \Gamma_2
\end{cases} \\
&\begin{cases}
Lu_2 = f & \text{in } \Omega_2, \\
u_2 = \psi & \text{on } \Gamma_2, \\
u_2 = u_1 & \text{on } \Gamma_1
\end{cases}
\end{align}

It is clear that \( u_1 \), the solution of problem (1.1), is the solution of the coupled problems (1.4) and (1.5), and then

\begin{align}
&u_1 = u \quad \text{in } \Omega_1 \cup \Gamma_{\Omega_1}, \\
&u_2 = u \quad \text{in } \Omega_2 \cup \Gamma_{\Omega_2}, \\
&u_1 = u_2 \quad \text{in } \Omega_{1,2} \cup \Gamma_{\Omega_{1,2}}.
\end{align}

Thus problem (1.1) is equivalent to the pair of problems (1.4) and (1.5). Due to the fact that on the closed region \( \Omega_{1,2} \cup \Gamma_{\Omega_{1,2}} \) the unknown functions \( u_1 \) and \( u_2 \) are coupled, we cannot solve the two problems separately. However, if we give an initial guess on \( \Gamma_{1,2}, u|_{\Gamma_{1,2}} = \psi_0 \), we can construct a sequence of approximating solutions \( \{ u_1^{(i)}, u_2^{(i)} \}, i = 1, 2, 3, \ldots, \) to \( \{ u_1, u_2 \} \) as is shown below

\begin{align}
&\begin{cases}
Lu_1^{(i)} = f & \text{in } \Omega_1, \\
u_1^{(i)} = \psi & \text{on } \Gamma_1, \\
u_1^{(i)} = \begin{cases}
\psi_0 & \text{if } i = 1 \\
u_{1+1}^{(i-1)} & \text{if } i \geq 2
\end{cases} & \text{on } \Gamma_{1,2}
\end{cases} \\
&\begin{cases}
Lu_2^{(i)} = f & \text{in } \Omega_2, \\
u_2^{(i)} = \psi & \text{on } \Gamma_2, \\
u_2^{(i)} = u_2^{(i)} & \text{on } \Gamma_{2,1}
\end{cases}
\end{align}

for \( i = 1, 2, 3, \ldots \). Under certain conditions [CH62], [KK58], it can be proved that the sequence \( \{ u_1^{(i)}, u_2^{(i)} \} \) produced from (1.6) converges to the solutions \( \{ u_1, u_2 \} \) of the pair of problems (1.4) and (1.5), from which the solution of problem (1.1) can be constructed. Note that this method of solving a continuous problem is in complete analogy with the Gauss-Seidel idea when applied for the solution of matrix equation problems.

1.1.2 Generalization of SAM

The classical SAM described in Section 1.1.1 decomposes the solution domain into two subdomains and solves two subproblems alternatingly. As a generalized version of SAM, we decompose the original domain into any finite number of subdomains. In the following, a simple description of a generalized SAM is presented.
First, we consider a $k$-way splitting of the PDE domain such that

$$\Omega = \bigcup_{j=1}^{k} \Omega_j,$$

where

$$\Omega_{j_1,j_2} = \Omega_{j_1} \cap \Omega_{j_2} \neq \emptyset \quad \text{if} \quad |j_1 - j_2| \leq 1,$$

$$\Omega_{j_1,j_2} = \Omega_{j_1} \cap \Omega_{j_2} = \emptyset \quad \text{if} \quad |j_1 - j_2| > 1,$$

for any $j_1, j_2 \in \{1, 2, \cdots, k\}$. Figure 1.2 depicts an instance of the above splitting for a two-dimensional domain with three subdomains. Second, we distinguish the actual boundaries from the artificial ones introduced by the splitting. We denote by $\Gamma_{\Omega_j}$ the boundary of $\Omega_j$ and define

$$\Gamma_{j,j-1} = \begin{cases} \phi & j = 1 \\ \Gamma_{\Omega_j} \cap \Omega_{j-1} & j = 2, \cdots, k, \end{cases}$$

$$\Gamma_{j,j} = \begin{cases} \Gamma_{\Omega_j} \cap \Gamma_{\Omega} & j = 1, \cdots, k, \end{cases}$$

$$\Gamma_{j,j+1} = \begin{cases} \Gamma_{\Omega_j} \cap \Omega_{j+1} & j = 1, \cdots, k-1 \\ \phi & j = k, \end{cases}$$

so we have

$$\Gamma_{\Omega_j} = \Gamma_{j,j-1} + \Gamma_{j,j} + \Gamma_{j,j+1}.$$ 

Giving initial guesses to the common boundaries, say $u|_{\Gamma_{j,j+1}} = \psi_{0,j}, j = 1, \cdots, k-1$, we can construct a sequence of $k$ functions $\{u_1^{(i)}, u_2^{(i)}, \cdots, u_k^{(i)}\}, i = 1, 2, 3, \cdots$, as is shown below

$$\begin{cases} Lu_j^{(i)} = f \quad \text{in} \ \Omega_j \\ u_j^{(i)} = u_{j-1}^{(i)} \quad \text{on} \ \Gamma_{j,j-1} \\ u_j^{(i)} = \psi \quad \text{on} \ \Gamma_{j,j} \quad \text{for} \ j = 1, 2, \cdots, k \\ u_j^{(i)} = \begin{cases} \psi_{0,j} & \text{for} \ i = 1 \\ u_{j+1}^{(i-1)} & \text{for} \ i \geq 2 \end{cases} \quad \text{on} \ \Gamma_{j,j+1} \end{cases} \quad (1.7)$$
for $i = 1, 2, 3, \cdots$. Under certain conditions [CH62], [KK58], the sequence $\{u^{(1)}_i, u^{(2)}_i, \cdots, u^{(i)}_k\}$ converges to the solution of problem (1.1).

Classical SAM is a special case of this generalized SAM. As a more general version of SAM, *multi-color SAM* is presented in [Tan87]. As in Section 1.1.1, note again that this method of solving a continuous problem is in complete analogy with the *Gauss-Seidel* idea when applied for the solution of matrix equation problems. This is so-called *Multiplicative Schwarz Scheme* which has been recently developed further with a variational framework [Lio88]. This Gauss-Seidel type scheme is inherently a sequential process. So the *Additive Schwarz Scheme* was designed by Dryja and Widlund [DW87] to exploit a parallelism. In fact, the Additive Schwarz Scheme can be considered as a continuous analog of the *Jacobi* type scheme for matrix equation problems.

It is known that SAM is very flexible in the sense that, for each subdomain and the associated subproblem, we can choose, e.g., the shape of each subdomain, the coupling pattern in the artificial boundaries, the numerical model, the solution scheme, etc. With these features, SAM can be incorporated into many other techniques such as multilevel, preconditioning, SOR, etc. in order to obtain good performances or in order to improve on the performance of the elliptic partial differential equation solvers. In parallel processing SAM makes it possible to map the subproblems into different topologies of parallel computers.

In this thesis we particularly focus on the coupling pattern in the artificial boundaries. The boundary value problem (BVP) (1.1) is converted to $k$ coupled problems defined over the $k$ subdomains provided that boundary conditions are specified on the interface boundaries $\Gamma_{j,j-1}$, $\Gamma_{j,j+1}$ of the subdomains. These boundary conditions are called the *interface conditions*. In order to solve the $k$ coupled problems, we assume some initial guesses to the solutions of the $k$ coupled problems on the common boundaries, say $u|_{\Gamma_{j,j-1}} = \psi_{0,j}, j = 1, \cdots, k - 1$, and apply some well known iterative scheme to construct a sequence of $k$ functions $\{u^{(1)}_i, u^{(2)}_i, \cdots, u^{(k)}_i\}, i = 1, 2, 3, \cdots$. In (1.7) we actually presented Dirichlet type interface conditions with a Gauss-Seidel like scheme.

### 1.1.3 Schwarz Splitting

Several modern extensions of SAM have been proposed. Miller [Mil65] first introduced a numerical analog of the SAM. Rodrigue and Simon's work [RS84a] is among the very first ones that studied SAM and its properties at the discrete matrix equation level. Further studies revealed that many results of the classical analysis in numerical linear algebra [Var62, You71] could be applied.

In the following we basically use the notation and adopt the terminology in [Tan87]. Suppose that the operator $L$ in equation (1.1) is a second order linear elliptic operator, then the discretized version of the continuous problem can be written as a matrix equation problem of the form
\[
\begin{equation}
Ax = \begin{bmatrix}
A_{11} & A_{12} & A_{13} \\
A_{21} & A_{22} & A_{23} \\
A_{31} & A_{32} & A_{33}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2 \\
f_3
\end{bmatrix}
= f
\end{equation}
\]
(1.8)

where the diagonal blocks \(A_{11}, A_{22}, A_{33}\) are square matrices. The blocks of the unknowns \(x_1, x_2, x_3\) are arranged in such a way that \([x_1^T, x_2^T, x_3^T]^T\) corresponds to the unknowns in \(\Omega_1\), \([x_2^T, x_3^T]^T\) corresponds to the unknowns in \(\Omega_2\) and \(x_2\) corresponds to the unknowns in \(\Omega_{1,2}\), which is the overlapping part of the two subdomains. The numerical SAM for the above problem solves the following subproblems alternatively

\[
\begin{equation}
\begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\begin{bmatrix}
x_1^{(i)} \\
x_2^{(i)}
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2
\end{bmatrix}
- \begin{bmatrix}
A_{13} \\
A_{23}
\end{bmatrix}
\begin{x}
\frac{(i-1)}{}
\end{x}

\begin{equation}
\begin{bmatrix}
A_{22} & A_{23} \\
A_{32} & A_{33}
\end{bmatrix}
\begin{bmatrix}
x_2^{(i)} \\
x_3^{(i)}
\end{bmatrix}
= \begin{bmatrix}
f_2 \\
f_3
\end{bmatrix}
- \begin{bmatrix}
A_{21} \\
A_{31}
\end{bmatrix}
\begin{x}
\frac{(i)}{}
\end{x}
\end{equation}
(1.9)

where \(x_2^{(i)}\) is an auxiliary vector with the same number of components as \(x_2^{(i)}\) and \(x_3^{(0)}\) is an arbitrary initial guess, for \(i = 1, 2, \cdots\). We observe that the procedure in (1.9) is equivalent to a 2 \(\times\) 2 block Gauss-Seidel iteration method for the following matrix equation

\[
\tilde{Ax} = \begin{bmatrix}
A_{11} & A_{12} & 0 & A_{13} \\
A_{21} & A_{22} & 0 & A_{23} \\
A_{31} & 0 & A_{32} & A_{33}
\end{bmatrix}
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2 \\
\tilde{x}_3
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2 \\
f_3
\end{bmatrix}
= \tilde{f}.
\end{equation}
(1.10)

Under certain conditions [RS84a, Mil65, Tan87], the procedure in (1.9) will converge to the solution of equation (1.10) with \(\tilde{x}_2 = x_2^0\), where \([\tilde{x}_1^T, \tilde{x}_2^T, \tilde{x}_3^T]^T\) is a solution of the matrix equation (1.8). Following Tang [Tan87], we call the equation (1.10) the Schwarz Enhanced Equation (SEE) of the original equation (1.8) and the corresponding matrix \(\tilde{A}\) in (1.10) the Schwarz Enhanced Matrix (SEM) of the matrix \(A\) in (1.8).

For the case of the 5 \(\times\) 5 block matrix equation

\[
Ax = \begin{bmatrix}
A_{11} & A_{12} & A_{13} & A_{14} & A_{15} \\
A_{21} & A_{22} & A_{23} & A_{24} & A_{25} \\
A_{31} & A_{32} & A_{33} & A_{34} & A_{35} \\
A_{41} & A_{42} & A_{43} & A_{44} & A_{45} \\
A_{51} & A_{52} & A_{53} & A_{54} & A_{55}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5
\end{bmatrix}
= f,
\end{equation}
(1.11)

the SEE can be defined by

\[
\tilde{Ax} = \begin{bmatrix}
A_{11} & A_{12} & 0 & A_{13} & A_{14} & 0 & A_{15} \\
A_{21} & A_{22} & 0 & A_{23} & A_{24} & 0 & A_{25} \\
A_{31} & 0 & A_{32} & A_{33} & A_{34} & 0 & A_{35} \\
A_{41} & 0 & A_{42} & A_{43} & A_{44} & 0 & A_{45} \\
A_{51} & 0 & A_{52} & A_{53} & A_{54} & 0 & A_{55}
\end{bmatrix}
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2 \\
\tilde{x}_3 \\
\tilde{x}_4 \\
\tilde{x}_5
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5
\end{bmatrix}
= \tilde{f}.
\end{equation}
(1.12)
Note that if \( x_2 = \overline{x}_2 \) and \( x_4 = \overline{x}_4 \) in (1.12), the two matrix equations in (1.11) and (1.12) are the same system of linear equations. From the above two particular cases, we can easily see how we can define the SEE and the SEM for the general case of a \((2k - 1) \times (2k - 1)\) block matrix \((k \geq 4)\)

\[
A = \begin{bmatrix}
A_{11} & A_{12} & \cdots & A_{1,2k-1} \\
A_{21} & A_{22} & \cdots & A_{2,2k-1} \\
\vdots & \vdots & \ddots & \vdots \\
A_{2k-1,1} & A_{2k-1,2} & \cdots & A_{2k-1,2k-1}
\end{bmatrix}.
\] (1.13)

Let \( x \) be partitioned in accordance with the partition of \( A \) in (1.13)

\[
x = [x_1^T, x_2^T, \cdots, x_{2k-1}^T]^T,
\]

then the augmented vector \( \tilde{x} \) obtained from \( x \) is defined by

\[
\tilde{x} = [x_1^T, x_2^T, x_3^T, x_4^T, x_5^T, \cdots, x_{2k-2}^T, x_{2k-1}^T]^T.
\] (1.14)

If the vector \( x = [x_1^T, x_2^T, \cdots, x_{2k-1}^T]^T \) is the solution of \( Ax = f \), then its augmented vector \( \tilde{x} \) satisfies the equation \( \tilde{A}\tilde{x} = \tilde{f} \). Therefore if \( \tilde{A}^{-1} \) exists, the (unique) solution \( \tilde{x} \) of \( \tilde{A}\tilde{x} = \tilde{f} \) is the augmented vector of \( x \). Rodrigue and Simon [RS84a] proved that the necessary and sufficient condition for the existence of \( \tilde{A}^{-1} \) is that all \( A_{2i-2,2i-1} \), \( i = 1, \cdots, k - 1 \), exist. In many problems, these conditions may hold due to some further property of the matrix \( A \); as for example, in the case of \( A \) being a positive definite matrix or a non-singular \( M \)-matrix or even a non-singular \( H \)-matrix (For corresponding definitions see, e.g., [BP79], [Var62, p85], [You71], [FS91, p7]). Thus if the matrix \( A \) in the matrix equation \( Ax = f \) is a positive definite matrix or a non-singular \( M \)-matrix or a non-singular \( H \)-matrix, then the solution \( \tilde{x} \) of the Schwarz Enhanced Equation \( \tilde{A}\tilde{x} = \tilde{f} \) is the augmented vector of \( x \).

As a modern extension of SAM Rodrigue and Simon [RS84a] suggested the following splitting.

**Definition 1.1** \( \tilde{A} = \tilde{M}_S - \tilde{N}_S \) is a Schwarz Splitting (SS) of the SEM of the matrix \( A \) in (1.13) if

\[
\tilde{M}_S = \text{diag}(S_1, S_2, \cdots, S_k)
\]

where

\[
S_1 = \begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix}
\]

\[
S_i = \begin{bmatrix}
A_{2i-2,2i-2} & A_{2i-2,2i-1} & A_{2i-2,2i} \\
A_{2i-1,2i-2} & A_{2i-1,2i-1} & A_{2i-1,2i} \\
A_{2i,2i-2} & A_{2i,2i-1} & A_{2i,2i}
\end{bmatrix}, \quad i = 2, \cdots, k - 1
\]

\[
S_k = \begin{bmatrix}
A_{2k-2,2k-2} & A_{2k-2,2k-1} \\
A_{2k-1,2k-2} & A_{2k-1,2k-1}
\end{bmatrix}
\]
We should always relate a Schwarz Splitting to the corresponding partition. A different block partition of the matrix $A$ will lead to a different Schwarz Splitting. We note that from the above definition, the Schwarz Splitting is essentially a block Jacobi splitting for a particular partition of SEM and SAM is the Gauss-Seidel scheme which corresponds to this partition.

1.1.4 Generalized Schwarz Splitting

As we mentioned in Section 1.1.2, SAM is very flexible in choosing the artificial boundary conditions. Tang [Tan92], among others, proposed a successful coupling on the artificial boundary. Let us introduce his generalized version of SAM and Generalized Schwarz Splitting.

Consider again the Dirichlet boundary value problem (1.1). The classical SAM uses the Dirichlet boundary conditions on the artificial boundary, but G. Rodrigue suggested the use of mixed boundary condition and Tang [Tan92] used the following coupling

$$g(u_1)|_{\Gamma_{1,2}} = g(u_2),$$
$$g(u_2)|_{\Gamma_{2,1}} = g(u_1)$$

(1.15)
on the artificial boundaries $\Gamma_{1,2}, \Gamma_{2,1}$ in (1.2), respectively, where

$$g(u) = \omega u + (1 - \omega) \frac{\partial u}{\partial n},$$

(1.16)
with $\frac{\partial u}{\partial n}$ denoting the outwardly directed normal derivative on the artificial boundaries. We note that for $\omega = 1$, this generalized version reduces to the classical SAM. So equations (1.4) and (1.5) will be

$$\begin{cases}
Lu_1 = f & \text{in } \Omega_1, \\
u_1 = \psi & \text{on } \Gamma_{1,1}, \\
g_1(u_1) = g_1(u_2) & \text{on } \Gamma_{1,2},
\end{cases}$$

and

$$\begin{cases}
Lu_2 = f & \text{in } \Omega_2, \\
u_2 = \psi & \text{on } \Gamma_{2,2}, \\
g_2(u_2) = g_2(u_1) & \text{on } \Gamma_{2,1},
\end{cases}$$

respectively. Thus the numerical Generalized SAM for problem (1.8) solves the following subproblems alternatively

$$\begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & B_2
\end{bmatrix}
\begin{bmatrix}
x_1^{(i)} \\
x_2^{(i)}
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2
\end{bmatrix}
- \begin{bmatrix}
0 & A_{13} \\
C_2 & A_{23}
\end{bmatrix}
\begin{bmatrix}
x_2^{(i-1)} \\
x_3^{(i-1)}
\end{bmatrix},$$

(1.17)

$$\begin{bmatrix}
B'_2 & A_{23} \\
A_{32} & A_{33}
\end{bmatrix}
\begin{bmatrix}
x_2^{(i)} \\
x_3^{(i)}
\end{bmatrix}
= \begin{bmatrix}
f_2 \\
f_3
\end{bmatrix}
- \begin{bmatrix}
A_{21} & C'_2 \\
A_{31} & 0
\end{bmatrix}
\begin{bmatrix}
x_1^{(i)} \\
x_2^{(i)}
\end{bmatrix}.$$
with an initial guess $x_i^{(0)}$, $x_j^{(0)}$, for $i = 1, 2, 3, \ldots$, where $B_2, C_2'$ are arbitrary matrices with $(B_2 - C_2')$ non-singular and

$$A_{22} = B_2 + C_2 = B_2' + C_2'. \quad (1.18)$$

The splittings in (1.18) correspond to the couplings in (1.15). In [Tan92], it is shown that a good choice of the splitting of $A_{22}$ can significantly improve the convergence of SAM. Obviously this procedure is equivalent to a $2 \times 2$ block Gauss-Seidel iteration method for the following matrix equation problem

$$\tilde{A}\tilde{x} = \begin{bmatrix}
A_{11} & A_{12} & 0 & A_{13} \\
A_{21} & B_2 & C_2 & A_{23} \\
A_{31} & 0 & B_2' & A_{33}
\end{bmatrix}
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2 \\
\tilde{x}_3
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2 \\
f_3
\end{bmatrix}
= \tilde{f}. \quad (1.19)$$

We call the matrix equation (1.19) the Generalized Schwarz Enhanced Equation (GSEE) of the original equation (1.8) and the corresponding matrix $\tilde{A}$ in (1.19) the Generalized Schwarz Enhanced Matrix (GSEM) of the matrix $A$ in (1.8). The splitting $\tilde{A} = \tilde{M}_S - \tilde{N}_S$ where

$$\tilde{M} = \begin{bmatrix}
A_{11} & A_{12} & 0 & 0 \\
A_{21} & B_2 & 0 & 0 \\
0 & 0 & B_2' & A_{23} \\
0 & 0 & 0 & A_{32} & A_{33}
\end{bmatrix}$$

is called the Generalized Schwarz Splitting. The procedure described above can be very easily generalized to cover the case of more than two subdomains.

### 1.2 One-Parameter Schwarz Splitting

We consider the two-point boundary value problem

$$Lu \equiv -u''(t) + q u(t) = f(t), \quad t \in (0, 1), \quad (1.20)$$

$$Bu \equiv u(0) = a_0, \quad Bu \equiv u(1) = a_1$$

with $q$ is a constant with $q \geq 0$ and formulate a numerical instance of Generalized SAM based on a $k$-way splitting of the unit interval and finite difference discretizations of the local BVP over each subdomain with mixed interface conditions (1.16) on the artificial boundaries. We introduce parameter $\alpha$ in the Generalized Schwarz Splitting where $\alpha$ is the parameter in the Generalized Schwarz Enhanced Matrix corresponding to the parameter $\omega$ (1.16) of the continuous problem (1.20). The relation between $\omega$ and $\alpha$ will be explained in Section 1.2.1. The Parameterized Schwarz Splitting is a special case of the Generalized Schwarz Splitting.

Tang [Tan92] applied a single parameter $\omega$ on the mixed interface conditions in (1.16). Tang noted that $\omega$ is a function of the parameter $\alpha$ and $h$ where $h$ is the discretization size. Furthermore, he was able to determine all non-zero eigenvalues of
the corresponding block Jacobi iteration matrix in the case of a 3-way decomposition of the domain \((k = 3)\) and to show experimentally the relation between the spectral radius of this matrix and the parameter \(\alpha\). It was observed experimentally that for some value of \(\alpha\) the convergence rate of the Parameterized Schwarz Splitting was optimized. For the general case \(k \geq 4\), a \(2(k-1) \times 2(k-1)\) matrix was derived whose eigenvalue spectrum definitely includes all the non-zero eigenvalues of the Jacobi matrix.

In Sections 1.2.1 and 1.2.2, we summarize the results in [Tan92]. In Section 1.2.3 we determine analytically the exact value, if possible, of the parameter \(\alpha\) that minimizes the spectral radius of the block Jacobi iteration matrix of the Generalized Schwarz Splitting. Specifically, we derive explicitly the optimal value of \(\alpha\) for the cases \(k = 2\) and \(k = 3\) for which the spectral radius of the Jacobi matrix turns out to be zero. In general, for \(k \geq 3\) overlapping subdomains, we present two coupled equations whose roots definitely include all the non-zero eigenvalues of the block Jacobi iteration matrix of the Generalized Schwarz Splitting. These equations can be used to estimate the optimal value of \(\alpha\) numerically.
1.2.1 Formulation of the Parameterized Schwarz Splitting

Let \( T_j(a, b, c) \) denote the tridiagonal \( j \times j \) matrix whose diagonal entries are \( b \) except that its first and last diagonal elements are \( a \) and \( c \), respectively, i.e.,

\[
T_j(a, b, c) = \begin{bmatrix}
a & -1 & 0 & 0 & \cdots & 0 \\
-1 & b & -1 & 0 & \cdots & 0 \\
0 & -1 & b & -1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & -1 & b & -1 \\
0 & 0 & \cdots & 0 & -1 & c
\end{bmatrix}_{j \times j}
\] (1.21)

and let \( T_j(x) \) denote the tridiagonal \( j \times j \) matrix such that all of the diagonal entries are \( x \), i.e.,

\[
T_j(x) = T_j(x, x, x). \quad (1.22)
\]

Following Tang's formulation, we discretize the BVP (1.20) by a second order central divided difference discretization scheme with a uniform grid of mesh size \( h \) which yields the linear system

\[
T_n(\beta) x = f,
\] (1.23)

where

\[
\beta = 2 + qh^2, \quad q \geq 0.
\] (1.24)

For the formulation of the Generalized SAM we split the domain \((0, 1)\) into \( k \) \((\geq 2)\) overlapping subdomains as shown in Figure 1.3. Furthermore, we denote by \( \ell \) the length of the overlap and \( \eta \) the length of each subdomain. Provided \( n + 1 = \frac{1}{h} \), we let \( l + 1 = \frac{\ell}{h} \) and \( m + 1 = \frac{\eta}{h} \) so that \( n = mk - l(k - 1) \). Throughout, we assume that \( l < \frac{m-1}{2} \) which implies that no three subdomains can have a common overlap. The open circled points in Figure 1.3 represent the artificial boundaries (interfaces) of the subdomains on which we force the solutions of the local BVP to satisfy the parameterized mixed interface conditions (1.16) with

\[
\omega = \frac{1 - \alpha}{1 - \alpha + \alpha h}, \quad 0 \leq \alpha < 1.
\] (1.25)

The derivation of this formula for \( \omega \) is not included in [Tan92]. We derive it in the following proposition which gives a standard equal-spaced discretization.

**Proposition 1.1** Consider the one-dimensional two-point boundary value problem

\[
-u''(t) + qu(t) = f(t), \quad t \in (\tau_1, \tau_2)
\] (1.26)

under the mixed boundary conditions

\[
\begin{align*}
\omega_1 u|_{t=\tau_1} + (1 - \omega_1) \frac{\partial u}{\partial n}|_{t=\tau_1} &= U_1, \\
\omega_2 u|_{t=\tau_2} + (1 - \omega_2) \frac{\partial u}{\partial n}|_{t=\tau_2} &= U_2,
\end{align*}
\] (1.27)
where $0 < \omega_i \leq 1$, $i = 1, 2$, and $\frac{\partial u}{\partial n}|_{t=x}$ is the outwardly directed normal derivative to the boundary at a point $t = x$. If one discretizes the continuous problem (1.26)-(1.27) by using a uniform grid of mesh size $h = \frac{2\pi}{m+1}$ and uses finite differences as follows

$$u''(t) \approx \frac{u(t-h) - 2u(t) + u(t+h)}{h^2},$$

$$\frac{\partial u}{\partial n}|_{t=t_i} \approx \frac{u(t_i) - u(t_i + h)}{h},$$

then the resulting linear system is given by the following matrix equation

$$\begin{bmatrix}
\beta - \alpha_1 & -1 & 0 & 0 & \cdots & 0 \\
-1 & \beta & -1 & 0 & \cdots & 0 \\
0 & -1 & \beta & -1 & \cdots & 0 \\
& & & & & \vdots \\
0 & 0 & \cdots & -1 & \beta & -1 \\
0 & 0 & \cdots & 0 & -1 & \beta - \alpha_2
\end{bmatrix} \begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_{m-1} \\
x_m
\end{bmatrix} = \begin{bmatrix}
h^2 f_1 + K_1 U_1 \\
h^2 f_2 \\
h^2 f_3 \\
\vdots \\
h^2 f_{m-1} \\
h^2 f_m + K_2 U_2
\end{bmatrix}$$

where

$$\beta = 2 + q h^2, \quad t_i = \tau_i + i h, \quad f_i = f(t_i), \quad i = 0, 1, \cdots, m+1,$$

$$\alpha_i = \frac{1}{1 - \omega_i + \omega_i h}, \quad K_i = \frac{h}{1 - \omega_i + \omega_i h}, \quad i = 1, 2.$$  

( Remark: Note that the pair of relationships

$$0 < \omega \leq 1 \quad \text{and} \quad \alpha = \frac{1 - \omega}{1 - \omega + \omega h}$$

is equivalent to the pair of relationships

$$0 \leq \alpha < 1 \quad \text{and} \quad \omega = \frac{1 - \alpha}{1 - \alpha + \alpha h}.$$  

Proof: Let $u_i, \quad i = 0, 1, \cdots, m+1,$ denote the approximate values of $u(t_i), \quad i = 0, 1, \cdots, m+1$ (see Figure 1.4). Substituting the expressions of finite difference in (1.28) into two-point BVP (1.26) and its boundary condition (1.27), we get a system of finite difference equations

$$-u_{i-1} + \beta u_i - u_{i+1} = h^2 f_i, \quad i = 1, 2, \cdots, m,$$  

with boundary conditions

$$\omega_1 u_0 + (1 - \omega_1) \left( \frac{u_0 - u_1}{h} \right) = U_1,$$

$$\omega_2 u_{m+1} + (1 - \omega_2) \left( \frac{u_{m+1} - u_m}{h} \right) = U_2.$$
From the first boundary condition (1.30), we get the expression of $u_0$ as follows

$$u_0 = \left( \frac{h}{1 - \omega_1 + \omega_1 h} \right) U_1 + \left( \frac{1 - \omega_1}{1 - \omega_1 + \omega_1 h} \right) u_1$$

$$= K_1 U_1 + \alpha_1 u_1.$$  \hfill (1.32)

Substituting the expression of $u_0$ in (1.32) into the finite difference equation (1.29) with $i = 1$, we obtain the discretization equation at the nodal point $t = t_1$ as follows

$$(\beta - \alpha_1) u_1 - u_2 = h^2 f_1 + K_1 U_1.$$  

Similarly, from (1.31), the discretization equation at the nodal point $t = t_m$ is given by

$$-u_{m-1} + (\beta - \alpha_2) u_m = h^2 f_m + K_2 U_2.$$  

Therefore the result follows.  

For easy exposition of the convergence analysis of the corresponding Generalized Schwarz Splitting, we consider the case of a 3-way ($k = 3$) splitting of the BVP domain. The treatment of the general case is straightforward. For this particular case, the corresponding discrete equation to BVP (1.23) in a block matrix form is given by

$$T_{n,x} = \begin{bmatrix} T_{m-1} & -F \\ -E & T_1 & -F \\ 0 & -E & T_{m-2} & -F \\ 0 & 0 & -E & T_1 & -F \\ 0 & 0 & 0 & -E & T_{m-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix} = f$$  \hfill (1.33)

where the $T_j$ denotes a $j \times j$ tridiagonal matrix defined in (1.22), i.e.,

$$T_j \equiv T_j(\beta).$$  \hfill (1.34)

the matrices $E$ have zero elements everywhere except for a 1 at the rightmost top position, and the matrices $F$ have zero elements everywhere except for a 1 at the leftmost bottom position. The matrices $E$ and $F$ have compatible sizes.
The corresponding Generalized Schwarz Enhanced Equation (GSEE) has the following structure

\[
\tilde{T}_n x = \begin{bmatrix}
T_{m-1} - F & 0 & 0 & 0 & 0 \\
- E & B_1 & 0 & 0 & 0 \\
- E & C_1 & - F & 0 & 0 \\
0 & - E & T_{m-2} - F & 0 & 0 \\
0 & 0 & - E & B_2 & C_2 - F \\
0 & 0 & - E & C_1' & B_1' - F \\
0 & 0 & 0 & 0 & - E & T_{m-1}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5 \\
f_6 \\
f_7
\end{bmatrix} = \bar{f}
\quad (1.35)
\]

where \( B_i, C_i' \) are arbitrary matrices with \( (B_i - C_i') \) non-singular for \( i = 1, 2 \), and

\[ T_i = B_i + C_i = B_i' + C_i', \quad i = 1, 2. \]

GSEE is a sort of generalization of SEE in (1.12). Several splittings can be employed for the matrix \( T_i \). Among them, we consider the Parameterized Schwarz Splitting corresponding to a particular splitting of the matrix \( T_i \). Specifically, we choose the \( l \times l \) matrices \( C_i' \) and \( C_i \) such that all their entries are zero except for an \( \alpha \) in the positions \((1,1)\) and \((l,l)\), respectively. It is not difficult to show that for \( \beta \geq 2 \), \( (B_i - C_i') = T_i(\beta - \alpha, \beta, \beta - \alpha) \) is non-singular (see Corollary 1 in [Var62, p.85]). It turns out that these conditions imply the equivalence of the linear systems (1.33) and (1.35) ([Tan87, Tan92]).

One can easily show that the matrix \( \bar{T}_n \) in (1.35) can be written in the form

\[
\bar{T}_n = \begin{bmatrix}
T_m(\beta, \beta, \beta - \alpha) & - F' \\
- E' & T_m(\beta - \alpha, \beta, \beta - \alpha) & - F' \\
0 & - E' & T_m(\beta - \alpha, \beta, \beta)
\end{bmatrix}
\quad (1.36)
\]

where \( E' \) is the \( m \times m \) matrix with zero elements everywhere except for a 1 in the position \((1, m-l)\) and \(-\alpha\) in the position \((1, m-l+1)\) and \( F' \) is the \( m \times m \) matrix with zero elements everywhere except for a 1 in the position \((m, l+1)\) and \(-\alpha\) in the position \((m, l)\).

The Parameterized Schwarz Splitting (PSS) for the matrix \( \bar{T}_n \) in (1.36) is defined as follows

\[
\bar{T}_n = M - N
\]

\[
= \begin{bmatrix}
T_m(\beta, \beta, \beta - \alpha) & 0 & 0 \\
0 & T_m(\beta - \alpha, \beta, \beta - \alpha) & 0 \\
0 & 0 & T_m(\beta - \alpha, \beta, \beta)
\end{bmatrix}
- \begin{bmatrix}
0 & F' & 0 \\
E' & 0 & F' \\
0 & E' & 0
\end{bmatrix} = \bar{T}_n
\quad (1.37)
\]

1.2.2 Convergence Analysis

The convergence analysis of the Parameterized Schwarz Splitting based on Jacobi iteration is reduced to calculating the spectral radius of the block Jacobi iteration matrix \( J = M^{-1} N \) of the matrix \( \bar{T}_n \) in (1.37). This Jacobi matrix has the form
Moreover, the block tridiagonal structure of $\tilde{T}_n$ of (1.36) implies that $\tilde{T}_n$ possesses Young's block property $A$ (see [Var62], [You71], [BP79], [HY81]). Thus, the convergence of the block Jacobi method implies that its Gauss-Seidel counterpart will converge asymptotically twice as fast, while its optimal SOR counterpart will converge asymptotically much faster. To simplify the presentation we adopt the notation $\rho(A)$ and $\sigma(A)$ for the spectral radius and the spectrum of a matrix $A$, respectively. The analysis of the SOR method requires knowledge of the spectrum of the block Jacobi iteration matrix $\tilde{J}$ in (1.38) or at least of its convex hull. If $\sigma(\tilde{J})$ is a set of real number, it is well known that the Young’s optimal value of the SOR parameter is given by $2/(1 + \sqrt{1 - \rho(\tilde{J})^2})$, (see [You54], [You71], [Var62], [BP79], [HY81]). More generally, if $\sigma(\tilde{J})$ is a set of complex number satisfying some conditions, the optimal SOR parameter can be found by the Young-Eidson’s algorithm (see [YE70], [You71]).

In the following we summarize the observations of [Tan92] in two Lemmas 1.1 and 1.2 and derive the optimal values of the parameter $\alpha$ explicitly for the special cases $k = 2, 3$ and show the conditions that $\alpha$ satisfies in the general case.

**Lemma 1.1** Consider the block Jacobi iteration matrix $\tilde{J}$ in (1.38) and the $4 \times 4$ matrix

$$G_3 = \begin{bmatrix} 0 & g_1 & 0 & 0 \\ g_2 & 0 & 0 & g_3 \\ g_3 & 0 & 0 & g_2 \\ 0 & 0 & g_1 & 0 \end{bmatrix}$$

(1.39)

where

$$g_1 = t_{m-t}^{(1)} - \alpha t_{m-t+1}^{(1)}, \quad g_2 = t_{m-t}^{(2)} - \alpha t_{m-t+1}^{(2)}, \quad g_3 = t_{m-t}^{(2)} - \alpha t_{m-t+1}^{(2)}$$

and

$$[t_{1}^{(1)}, t_{2}^{(1)}, \ldots, t_{m}^{(1)}]^T \text{ and } [t_{1}^{(2)}, t_{2}^{(2)}, \ldots, t_{m}^{(2)}]^T$$

are the last columns of $T_m^{-1}(\beta, \beta, \beta - \alpha)$ and $T_m^{-1}(\beta - \alpha, \beta, \beta - \alpha)$, respectively. Then $\tilde{J}$ and $G_3$ have the same spectra except possibly for some zeros, that is, there holds that

$$\sigma(\tilde{J}) = \sigma(G_3) \cup \{0\}. \quad (1.40)$$

**Proof**: We begin our proof with the observation that all row vectors of $F'$ in (1.38) are zero except the last row of $F'$. Thus only the last columns

$$[t_{1}^{(1)}, t_{2}^{(1)}, \ldots, t_{m}^{(1)}]^T \text{ and } [t_{1}^{(2)}, t_{2}^{(2)}, \ldots, t_{m}^{(2)}]^T$$

in $T_m^{-1}(\beta, \beta, \beta - \alpha)$ and $T_m^{-1}(\beta - \alpha, \beta, \beta - \alpha)$ are used when $T_m^{-1}(\beta, \beta, \beta - \alpha) F'$ and $T_m^{-1}(\beta - \alpha, \beta, \beta - \alpha) F'$ are computed, respectively. Similarly, when $T_m^{-1}(\beta - \alpha, \beta, \beta) E'$
and $T_m^{-1}(\beta - \alpha, \beta, \beta - \alpha) E'$ are computed, only the first columns in $T_m^{-1}(\beta - \alpha, \beta, \beta)$ and $T_m^{-1}(\beta - \alpha, \beta, \beta - \alpha)$ are used and these columns are given by

$$[t_m^{(1)}, \cdots, t_2^{(1)}, t_1^{(1)}]^T \text{ and } [t_m^{(2)}, \cdots, t_2^{(2)}, t_1^{(2)}]^T,$$

respectively. Since $l < \frac{m-1}{2}$, the matrix $\bar{J}$ in (1.38) has only eight non-zero columns. Let $P$ be the $3m \times 3m$ permutation matrix that moves these columns, i.e.,

$$m - l, m - l + 1, m + l, m + l + 1, 2m - l, 2m - l + 1, 2m + l, 2m + l + 1$$

to the last eight columns in the order

$$3m - 8 + i, \quad i = 1, 2, \cdots, 8,$$

respectively. Using the permutation matrix $P$ just defined, $\bar{J}$ can be transformed to $\bar{J}'$ as follows

$$\bar{J}' = P^T \bar{J} P = \begin{bmatrix} 0 & * \ 0 & W \end{bmatrix} \quad (1.41)$$

where the symbol $*$ denotes a possibly non-zero block and

$$W = \begin{bmatrix}
0 & 0 & -\alpha t_{m-1}^{(1)} & t_{m-1}^{(1)} & 0 & 0 & 0 & 0 \\
0 & 0 & -\alpha t_{m-1+1}^{(1)} & t_{m-1+1}^{(1)} & 0 & 0 & 0 & 0 \\
t_{m-1}^{(2)} & -\alpha t_{m-1}^{(2)} & 0 & 0 & 0 & 0 & -\alpha t_{m-1}^{(2)} & t_{m-1}^{(2)} \\
t_{m-1+1}^{(2)} & -\alpha t_{m-1+1}^{(2)} & 0 & 0 & 0 & 0 & -\alpha t_{m-1+1}^{(2)} & t_{m-1+1}^{(2)} \\
0 & 0 & 0 & 0 & t_{m-1+1}^{(1)} & -\alpha t_{m-1+1}^{(1)} & 0 & 0 \\
0 & 0 & 0 & 0 & t_{m-1}^{(1)} & -\alpha t_{m-1}^{(1)} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Since the matrix $W$ has only four independent columns, a similarity transformation on it yields the matrix $G_3$ in (1.39) whose eigenvalues include the four, possibly, non-zero eigenvalues of $W$ (i.e., the only four eigenvalues might be non-zero). Here we present the derivation of $G_3$ from $W$ since it was not included in [Tan92].

For this derivation we let $\bar{P}$ be the permutation matrix that moves the columns 1, 4, 5, 8 to the columns 5, 6, 7, 8, respectively, and define the matrix

$$Q = \begin{bmatrix}
1 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \alpha & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & \alpha & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & \alpha & 1 & 0
\end{bmatrix}$$

From the definition of $\bar{P}$ and $Q$, we can easily show that
Then the relation (1.40) is a direct consequence of (1.41) and (1.42).

It is worth noticing that the roots of the characteristic polynomial of the matrix in (1.39) are the non-identically zero eigenvalues

\[ \pm \sqrt{g_1(g_2 \pm g_3)}. \]  

(1.43)

A similar analysis as in the proof of Lemma 1.1 can be used to cover the \( k \) \((\geq 3)\) case. The corresponding result is given in [Tan92] and is stated here in the following lemma.

**Lemma 1.2** For \( k \) \((\geq 3)\) overlapping subdomains, the non-zero eigenvalues of the Jacobi matrix \( \bar{J} \) are included in those of the following \((k - 1) \times (k - 1)\) block matrix

\[
G_k = \begin{bmatrix}
E & U & 0 & 0 & \cdots & 0 \\
L & D & U & 0 & \cdots & 0 \\
0 & L & D & U & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & L & D & U \\
0 & 0 & \cdots & 0 & L & E^T
\end{bmatrix}
\]  

(1.44)

where

\[
E = \begin{bmatrix} 0 & g_1 \\ g_2 & 0 \end{bmatrix}, \quad L = \begin{bmatrix} g_3 & 0 \\ 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & g_3 \\ g_2 & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & 0 \\ 0 & g_3 \end{bmatrix}.
\]

Specifically, the following relation holds

\[
\sigma(\bar{J}) = \sigma(G_k) \cup \{0\}.
\]  

(1.45)

**Remark 1.1** For \( k = 2 \) overlapping subdomains, it can be seen that the matrix \( G_2 \) is given by

\[
G_2 = \begin{bmatrix} 0 & g_1 \\ g_1 & 0 \end{bmatrix}.
\]  

(1.46)

### 1.2.3 Determination of the Optimal Parameter

In Section 1.2.2 we summarized the results derived by Tang in [Tan92]. In this subsection we address the open problem of determining analytically the exact optimal values of \( \alpha \). This problem can be formulated as follows.
Problem: Determine the optimal value of \( \alpha \) for which the spectral radius of the block Jacobi iteration matrix of the Parameterized Schwarz Splitting is as small as possible.

For the determination of the optimal \( \alpha \), we obtain analytic expressions for \( g_1, g_2, g_3 \) which turn out to be expressions in \( [t_1^{(1)}, t_2^{(1)}, \ldots, t_m^{(1)}]^T \) and \( [t_1^{(2)}, t_2^{(2)}, \ldots, t_m^{(2)}]^T \). A technique for computing these vectors is suggested in [Tan92] (see also [FU69]). Moreover, they can be derived from the analysis of a more general case which is formulated and treated in Section 1.3. The following lemma states the analytic expressions of \( g_1, g_2, g_3 \) while its proof contains an outline of their derivation.

Lemma 1.3 Let \( \theta = \arccosh(\frac{\beta}{2}) \geq 0 \), where \( \beta \) is defined in (1.24). If \( \beta > 2 \), then

\[
\begin{align*}
g_1 &= \frac{\alpha \sinh((m-\ell+1)\theta) - \sinh((m-\ell)\theta)}{\alpha \sinh(m\theta) - \sinh((m+1)\theta)}, \\
g_2 &= \frac{\frac{1}{2} \sinh((m-\ell)\theta) (\alpha - \epsilon^\theta) (\alpha - \epsilon^{-\theta})}{(\alpha \sinh(m\theta) - \sinh((m+1)\theta)) (\alpha \cosh(m\theta) - \cosh((m+1)\theta))}, \\
g_3 &= \frac{(\alpha \sinh(m\theta) - \sinh((m+1)\theta)) (\alpha \cosh((m+1)\theta) - \cosh((m+1)\theta))}{(\alpha \sinh((m+1)\theta) - \sinh(m\theta)) (\alpha \cosh((m+1)\theta) - \cosh(m\theta))}.
\end{align*}
\]

On the other hand, if \( \beta = 2 \), then

\[
\begin{align*}
g_1 &= \frac{(m-\ell+1)\alpha - (m-\ell)}{(m) \alpha - (m+1)}, \\
g_2 &= \frac{(m-\ell) (\alpha - 1)}{(m-1) \alpha - (m+1)}, \\
g_3 &= \frac{(l-\ell+1) \alpha - (l+1)}{(m-1) \alpha - (m+1)}.
\end{align*}
\]

Proof: As we mentioned before, the expressions of \( t_1^{(1)}, t_2^{(2)}, \ldots, m, \) can be obtained from [Tan92] or [FU69] and as a special case from Proposition 1.3 in Section 1.3. Since \( [t_1^{(1)}, t_2^{(1)}, \ldots, t_m^{(1)}]^T \) is the last column of \( T_m^{-1}(\beta, \beta, \beta - \alpha) \), its components satisfy the following system of equations

\[
\begin{align*}
\beta \cdot t_1^{(1)} - t_2^{(1)} &= 0, \\
-t_{p-1}^{(1)} + \beta \cdot t_p^{(1)} - t_{p+1}^{(1)} &= 0, \quad p = 2, \ldots, m - 1, \\
-t_{m-1}^{(1)} + (\beta - \alpha) \cdot t_m^{(1)} &= 1,
\end{align*}
\]

which can be transformed into

\[
\begin{align*}
(\beta - \alpha) \cdot \delta_1 - \delta_2 &= 1, \\
-\delta_{p-1} + \beta \cdot \delta_p - \delta_{p+1} &= 0, \quad p = 2, \ldots, m - 1, \\
-\delta_{m-1} + (\beta - 0) \cdot \delta_m &= 0,
\end{align*}
\]

by substituting \( \delta_{m-p+1} \) for \( t_p^{(1)} \). From the result in Proposition 1.3, we obtain

\[
\delta_p = \begin{cases}
\frac{\sinh((m-p+1)\theta) - 0 \sinh((m-p)\theta)}{\sinh((m+1)\theta) - (\alpha+0) \sinh(m\theta) + \alpha \cdot 0 \sinh((m-1)\theta)} & \text{for } \beta > 2, \\
\frac{m-p+1 - 0 \cdot (m-p)}{(m+1) - (\alpha+0) \cdot m + \alpha \cdot 0} & \text{for } \beta = 2.
\end{cases}
\]
for \( p = 1, 2, \ldots, m \), where \( \theta = \arccosh(\frac{\theta}{2}) \). Considering the case of \( \beta > 2 \), we have
\[
    t_p^{(1)} = \delta_{m-p+1} = \frac{\sinh(p\theta)}{\sinh((m+1)\theta) - \alpha \sinh(m\theta)} \tag{1.49}
\]
for \( p = 1, 2, \ldots, m \). Similarly we find that
\[
    t_p^{(2)} = \frac{\sinh(p\theta) - \alpha \sinh((p-1)\theta)}{\sinh((m+1)\theta) - 2 \alpha \sinh(m\theta) + \alpha^2 \sinh((m-1)\theta)} \tag{1.50}
\]
for \( p = 1, 2, \ldots, m \). From the two expressions in (1.49), (1.50), we obtain for \( g_1, g_2, g_3 \) the following
\[
    g_1 = t_{m-1}^{(1)} - \alpha t_{m-1+1}^{(1)} = \frac{\sinh((m-1)\theta) - \alpha \sinh((m-l+1)\theta)}{\sinh((m+1)\theta) - \alpha \sinh(m\theta)},
\]
\[
    g_2 = t_{m-l}^{(2)} - \alpha t_{m-1}^{(2)} = \frac{\sinh((m-l)\theta) - \alpha \sinh((m-l+1)\theta)}{\sinh((m+1)\theta) - 2 \alpha \sinh(m\theta) + \alpha^2 \sinh((m-1)\theta)},
\]
\[
    g_3 = t_{l+1}^{(2)} - \alpha t_l^{(2)} = \frac{\sinh((l+1)\theta) - 2 \alpha \sinh(l\theta) + \alpha^2 \sinh((l-1)\theta)}{\sinh((m+1)\theta) - 2 \alpha \sinh(m\theta) + \alpha^2 \sinh((m-1)\theta)}.
\]
Using the identities
\[
    \sinh(A) = 2 \sinh(\frac{A}{2}) \cosh(\frac{A}{2}),
\]
\[
    \sinh(A) + \sinh(B) = 2 \sinh(\frac{A+B}{2}) \cosh(\frac{A-B}{2}),
\]
we can factor the numerator and the denominator in \( g_2 \) and \( g_3 \) to obtain (1.47). For the case of \( \beta = 2 \) we can take similar steps as above.

Having obtained the explicit expressions for \( g_1, g_2, g_3 \), we now determine the optimal value of \( \alpha \) for which the spectral radius of the block Jacobi iteration matrix of the Generalized Schwarz Splitting becomes as small as possible. This is given in the Theorem 1.1. In the proof of the theorem, we refer to Proposition 1.2 which uses the matrix polynomial theory to solve a system of difference equations with vectors as unknowns and matrices as coefficients. Similar techniques are also found in [Tan87], [KHHR92], [LHHR92].

**Proposition 1.2** Let \( G_k \) \( (k \geq 3) \) be the \( (k-1) \times (k-1) \) block matrix
\[
    G_k = \begin{bmatrix}
        E & U & 0 & 0 & \cdots & 0 \\
        L & D & U & 0 & \cdots & 0 \\
        0 & L & D & U & \cdots & 0 \\
        \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
        0 & 0 & \cdots & L & D & U \\
        0 & 0 & \cdots & 0 & L & E^T
    \end{bmatrix}, \tag{1.51}
\]
where
\[
E = \begin{bmatrix} 0 & g_1 \\ g_2 & 0 \end{bmatrix}, \quad L = \begin{bmatrix} g_3 & 0 \\ g_2 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & g_2 \\ g_2 & 0 \end{bmatrix}, \quad U = \begin{bmatrix} 0 & 0 \\ 0 & g_3 \end{bmatrix}.
\]
Assume \(g_1g_2g_3 \neq 0\), then the eigenvalues \(\lambda\) of the matrix \(G_k\), different from \(0\) and \(\pm(g_2 \pm g_3)\), satisfy the following equation
\[
g_3^2 \lambda(\zeta_1^k - \zeta_2^k) + (g_2^2 g_3 - g_3^2 - g_1^2 g_3)(\zeta_1^{k-1} - \zeta_2^{k-1}) + (g_1 - g_2)^2 \lambda(\zeta_1^{k-2} - \zeta_2^{k-2}) = 0 \tag{1.52}
\]
where \(\zeta_1\) and \(\zeta_2\) are the two roots of
\[
(g_3 \lambda) \zeta^2 - (\lambda^2 + g_3^2 - g_2^2) \zeta + (g_3 \lambda) = 0. \tag{1.53}
\]

**Proof**: The eigenvalue-eigenvector problem for \(G_k\) is equivalent to the boundary value problem of the following system of matrix difference equations
\[
L Z_{i-1} + (D - \lambda I) Z_i + U Z_{i+1} = 0, \quad i = 2, \ldots, k-2, \tag{1.54}
\]
\[
L Z_{k-1} + (D - \lambda I) Z_k = 0
\]
where \(\lambda\) is an eigenvalue of \(G_k\) and \([Z_1^T, Z_2^T, \ldots, Z_{k-1}^T]^T\) is the corresponding eigenvector. We can transform the system of equations (1.54) to the equivalent one
\[
L Z_0 + (D - \lambda I) Z_1 = 0, \quad i = 1, \ldots, k-1, \tag{1.55}
\]
\[
L Z_{k-1} + (D - \lambda I) Z_k = 0
\]
where the first and the last equations constitute the boundary conditions. The matrix difference equations in (1.55) can be solved by the nonmonic matrix polynomial theory [GLR82]. The nonmonic matrix polynomial
\[
\Phi(\zeta) = U \zeta^2 + (D - \lambda I) \zeta + L
\]
\[
= \begin{bmatrix} g_3 - \lambda \zeta & g_2 \zeta \\ g_2 \zeta & g_3 \zeta^2 - \lambda \zeta \end{bmatrix} \tag{1.56}
\]
is the characteristic equation corresponding to the matrix difference equation (1.55). If \((X, J)\) is the finite Jordan pair of (1.56), Theorem 8.3 in [GLR82] gives the general solution of the homogeneous finite difference equations in (1.55) by the expressions
\[
Z_i = X J^i C, \quad i = 0, 1, 2, \ldots, \tag{1.57}
\]
where \(C = [c_0, c_1, c_2]^T\) is a constant vector to be determined by the boundary conditions. The determinant of the matrix polynomial (1.56) is given by
\[
det(\Phi(\zeta)) = -\zeta \left( g_3 \lambda \zeta^2 - (\lambda^2 + g_3^2 - g_2^2) \zeta + g_3 \lambda \right)
\]
\[
= -(\zeta - \zeta_0)(\zeta - \zeta_1)(\zeta - \zeta_2).
\]
where
\[
\begin{align*}
\zeta_0 &= 0, \\
\zeta_1 &= \frac{\left(\lambda^2 + g_3^2 - g_2^2\right) + \sqrt{\left(\lambda^2 + g_3^2 - g_2^2\right)^2 - 4\lambda^2 g_3^2}}{2g_3 \lambda}, \\
\zeta_2 &= \frac{\left(\lambda^2 + g_3^2 - g_2^2\right) - \sqrt{\left(\lambda^2 + g_3^2 - g_2^2\right)^2 - 4\lambda^2 g_3^2}}{2g_3 \lambda}.
\end{align*}
\] (1.58)

Therefore the eigenvalues of the matrix polynomial \( \Phi(\zeta) \) are given by \( \zeta_0, \zeta_1, \zeta_2 \). One notes that
\[
\zeta_1 + \zeta_2 = \frac{\lambda^2 + g_3^2 - g_2^2}{g_3 \lambda} \quad \text{and} \quad \zeta_1 \zeta_2 = 1.
\] (1.59)

By virtue of the assumption \( \lambda \neq \pm (g_2 \pm g_3) \), (1.58) implies \( \zeta_1 \neq \zeta_2 \). Consequently, the matrices \( \Phi(\zeta_i) \) and the eigenvectors of \( \Phi(\zeta) \) corresponding to the eigenvalues \( \zeta_i \) are given as follows
\[
\begin{align*}
\Phi(\zeta_0) &= \begin{bmatrix} g_3 & 0 \\ 0 & 0 \end{bmatrix}, & x_0 &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\
\Phi(\zeta_1) &= \begin{bmatrix} g_3 - \lambda \zeta_1 & g_2 \zeta_1 \\ g_2 \zeta_1 & g_3 \zeta_1^2 - \lambda \zeta_1 \end{bmatrix}, & x_1 &= \begin{bmatrix} 1 \\ \omega_1 \end{bmatrix}, \\
\Phi(\zeta_2) &= \begin{bmatrix} g_3 - \lambda \zeta_2 & g_2 \zeta_2 \\ g_2 \zeta_2 & g_3 \zeta_2^2 - \lambda \zeta_2 \end{bmatrix}, & x_2 &= \begin{bmatrix} 1 \\ \omega_2 \end{bmatrix},
\end{align*}
\]

where
\[
\omega_1 = \frac{\lambda \zeta_1 - g_3}{g_2 \zeta_1} \quad \text{and} \quad \omega_2 = \frac{\lambda \zeta_2 - g_3}{g_2 \zeta_2}.
\] (1.60)

Then, the finite Jordan pair \((X, J)\) is given by
\[
X = \begin{bmatrix} 0 & 1 & 1 \\ 1 & \omega_1 & \omega_2 \end{bmatrix} \quad \text{and} \quad J = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \zeta_1 & 0 \\ 0 & 0 & \zeta_2 \end{bmatrix}.
\] (1.61)

The constant vector \( C \) in (1.57) is determined by the boundary conditions in (1.55). Applying (1.61) and (1.57) to the first boundary condition in (1.55), one gets
\[
\begin{bmatrix} g_3 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & \omega_1 & \omega_2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} 0 & g_2 - g_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & \omega_1 & \omega_2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \zeta_1 & 0 \\ 0 & 0 & \zeta_2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix},
\]

which implies
\[
(g_3 + (g_2 - g_1)\omega_1 \zeta_1) c_1 + (g_3 + (g_2 - g_1)\omega_2 \zeta_2) c_2 = 0.
\] (1.62)

Similarly the second boundary condition gives
\[
(g_3 \omega_1 \zeta_1 + (g_2 - g_1)) \zeta_1^{b-1} c_1 + (g_3 \omega_2 \zeta_2 + (g_2 - g_1)) \zeta_2^{b-1} c_2 = 0.
\] (1.63)
So the equations (1.62) and (1.63) form the simple matrix equation problem

\[
\begin{bmatrix}
g_3 + (g_2 - g_1)w_1 \xi_1 \\
g_3 + (g_2 - g_1)w_2 \xi_2 \\
(g_2w_1 \xi_1 + (g_2 - g_1))\xi_1^{k-1} \\
(g_3w_2 \xi_2 + (g_2 - g_1))\xi_2^{k-1}
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
c_1 \\
c_2
\end{bmatrix} =
\begin{bmatrix}
0 \\
0
\end{bmatrix}. \tag{1.64}
\]

Since equation (1.64) must have a non-trivial solution, the matrix in (1.64) must be singular. Using (1.59) and (1.60), one can get for the determinant \( D \) of the matrix in (1.64) that

\[
-g_2^2 D = g_3^2 g_2 \lambda (\xi_1^k - \xi_2^k) + (g_2^2 g_3^2 - g_2 g_3^2)(\xi_1^{k-1} - \xi_2^{k-1}) + g_2(g_1 - g_2)^2 \lambda (\xi_1^{k-2} - \xi_2^{k-2}).
\]

Since \( g_2 \neq 0 \), the result follows. 

\[\Box\]

**Theorem 1.1** For \( k = 2, 3 \), the optimal value, \( \hat{\alpha} \), of \( \alpha \) that minimizes \( \rho(\tilde{J}) = \rho(\tilde{J}(\alpha)) \) is given by the expressions

\[
\hat{\alpha} = \begin{cases} 
\frac{\sinh((m - l) \theta)}{\sinh((m - l + 1) \theta)}, & \beta > 2, \\
\frac{m - l}{2 - l + 1}, & \beta = 2,
\end{cases} \tag{1.65}
\]

where \( \theta = \text{arccosh}(\frac{g_2}{2}) > 0 \), the \( \beta \) is defined in (1.24) and the \( m \) is an integer such that \( h(m + 1) \) is the length of each subdomain (see page 13).

For \( k \geq 4 \), and except for some trivial cases, the optimal value of \( \alpha \) (\( \hat{\alpha} \)) that minimizes \( \rho(\tilde{J}) = \rho(\tilde{J}(\alpha)) \) is the value of \( \alpha \) that minimizes the largest of the moduli of the (non-identically zero) roots \( \lambda \) of the equation

\[
g_3^2 \lambda \sum_{i=0}^{[x]} S_{k-2i-1} + (g_2^2 g_3^2 - g_2 g_3^2) \sum_{i=0}^{[x]} S_{k-2i-2} + (g_1 - g_2)^2 \lambda \sum_{i=0}^{[x]} S_{k-2i-3} = 0 \tag{1.66}
\]

where \([x]\) is the maximum integer not greater than \( x \) and \( S_i \) is given recursively by

\[
S_0 = 2, \quad S_1 = \frac{\lambda^2 + g_3^2 - g_2^2}{g_3^2}, \\
S_i = S_i S_{i-1} + S_{i-2}, \quad i = 2, 3, \cdots, k - 1. \tag{1.67}
\]

**Proof**: For \( k = 2 \), we have \( \sigma(\tilde{J}) = \sigma(G_2) \cup \{0\} \), where \( G_2 \) is the matrix in (1.46). The eigenvalues of \( G_2 \) are given by

\[
\pm \sqrt{g_1^2}.
\]

So, \( \rho(\tilde{J}) \) can be made zero if and only if \( g_1 = 0 \). The latter condition holds if and only if \( \alpha \) is given by (1.65).
For \( k = 3 \), we have from Tang's result in (1.43) that \( \rho(\bar{J}) \) is given by

\[
\rho(\bar{J}) = \rho(G_3) = \max \left( \sqrt{|g_1(g_2 + g_3)|}, \sqrt{|g_1(g_2 - g_3)|} \right).
\]

(1.68)

We note that \( g_2 + g_3 \) and \( g_2 - g_3 \) cannot be made simultaneously zero since then we would have \( g_3 = 0 \) implying \( \alpha > 1 \). So, \( \rho(\bar{J}) \) in (1.68) can be minimized, in fact can be made zero, if and only if \( g_1 = 0 \). Therefore the optimal value of \( \alpha \) is the same as that of case \( k = 2 \) given in (1.65).

For \( k \geq 4 \), by virtue of Lemma 1.2, we have

\[
\rho(\bar{J}) = \rho(G_k).
\]

For \( \alpha \in [0,1) \), we have \( g_3 \neq 0 \). Therefore assuming that \( g_1 g_2 \neq 0 \) and \( \lambda \neq \pm(g_2 \pm g_3) \), all the assumptions of Proposition 1.2 are satisfied. Consequently, the eigenvalues of \( G_k \) we are interested in are obtained from the solution of the coupled system of equations (1.52) and (1.53). Now, (1.53) will be satisfied with \( \zeta = \zeta_1 \) as well as with \( \zeta = \zeta_2 \). So, we substitute, successively, \( \zeta_1 \) and \( \zeta_2 \) for \( \zeta \) in (1.53), multiplying then the first resulting equation by \( \zeta_1^{-2} \) and the second one by \( \zeta_2^{-2} \) and add the two new resulting equations together. Then we substitute \( S_i = \zeta_1 + \zeta_2, i = 1,2, \cdots, k-1 \), with \( S_1 = \zeta_1 + \zeta_2 = (\lambda^2 + g_3^2 - g_2^2)/(g_2 \lambda) \), and \( S_0 = 2 \) and we obtain equation (1.67). By virtue of the assumption \( \lambda \neq \pm(g_2 \mp g_3) \), (1.58) implies \( \zeta_1 \neq \zeta_2 \). Hence, dividing (1.52) through by \( \zeta_1 - \zeta_2 \) and using equation (1.67), (1.66) is readily obtained.

Remark 1.2 We see that the solutions of (1.66) are the, possibly, non-zero eigenvalues of \( \bar{J} \). So, to solve our problem for \( k \geq 4 \), we have to solve numerically the equation (1.66) in \( \lambda \) which, after getting rid of the denominators that appear, is a polynomial equation of degree \( 2(k - 1) \) that contains only even powers of \( \lambda \). Since its coefficients are functions of \( \alpha \), this simply means that the optimal value of \( \alpha \) in this present general case can only be found computationally by considering a range of values of it in \([0,1)\).

Remark 1.3 The trivial cases \((g_1 g_2 \neq 0 \text{ and } \lambda \neq \pm(g_2 \pm g_3))\), not examined in the theorem, give essentially similar coupled equations (1.66), (1.67).

Remark 1.4 The characteristic polynomial of the matrix \( G_k \) is given by the system of the two coupled equations (1.66), (1.67). In fact, even for \( k = 2,3 \), these polynomials are recovered from the two equations (1.66), (1.67). For instance, the corresponding characteristic polynomials for \( k = 4,5 \) are

\[
\lambda^6 - (2g_1 g_2 + g_2^3)\lambda^4 + (g_1^2 g_2^2 + 2g_1 g_2^2 - 2g_1 g_2 g_3^2)\lambda^2 + (2g_1^2 g_2 g_3^2 - g_1^2 g_3^4 - g_1^2 g_3^2)
\]

and

\[
\lambda^8 - (2g_2^2 + 2g_1 g_2)\lambda^6 + (g_2^4 + 4g_1 g_2^3 + g_1^2 g_2^2 - g_2^2 g_3^2 - 2g_1 g_2 g_3^2)\lambda^4 + (4g_1 g_2^2 g_3^2 + 2g_1^2 g_2 g_3^2 - 2g_1 g_2 g_3^4 - 2g_1 g_2 g_3^2 - 2g_1 g_2 g_3^2)\lambda^2 + (g_1^2 g_3^6 + 3g_1 g_2 g_3^4 - g_1^2 g_3^6 - 3g_1^2 g_3^4 g_3^2),
\]

respectively.
1.3 Multi-Parameter Schwarz Splitting

In this section we consider again the two-point boundary value problem in (1.20) and assume the decomposition for the BV domain defined in the previous section. We will formulate the Multi-Parameterized Schwarz Splitting (MPSS), i.e., the numerical SAM based on finite difference discretization and Jacobi type iteration scheme and assuming the coupling (1.16) with different \( \omega_i \)'s in the artificial boundary between the subdomains \( \Omega_i \) and \( \Omega_{i+1} \). Note that if \( \omega_i = \omega, i = 1, 2, \cdots, k - 1 \), then the present multi-parameter case reduces to the one-parameter case considered in Section 1.2. After formulating the Multi-Parameterized Schwarz Splitting, we solve the following open problem:

Problem: Determine the values of \( \alpha_i \)'s for which the spectral radius of the block Jacobi iteration matrix of the Multi-Parameterized Schwarz Splitting is as small as possible.

1.3.1 Formulation of the Multi-Parameterized Schwarz Splitting

We observed that there are many ways of splitting the matrix \( T_l \) in (1.33). Here we choose the matrices \( B_i, B'_i, C_i, C'_i \) in (1.35) in order to define the Multi-Parameterized Schwarz Splitting. For this formulation, we introduce a set of \( k - 1 \) parameters \( \alpha_i, i = 1, 2, \cdots, k - 1 \), such that each \( \alpha_i \) is associated with \( \omega_i \). As in the case of the One-Parameter (\( \omega \)) SAM, we establish the following relationship (see Proposition 1.1) between \( \omega_i \) in (1.16) and \( \alpha_i \), i.e.,

\[
\omega_i = \frac{1 - \alpha_i}{1 - \alpha_i + \alpha_i h^2}, \quad i = 1, 2, \cdots, k - 1,
\]

where \( h \) is the grid size and \( 0 \leq \alpha_i < 1 \).

Let \( C'_i \) and \( C_i \) be \( l \times l \) matrices with zero elements everywhere except for an \( \alpha_i \) in the position \((1,1)\) and \((l,l)\), respectively. Moreover, we define \( E'_1 \) to be the \( m \times m \) matrix with zero elements everywhere except for a 1 in the position \((1,m-l)\) and \(-\alpha_i\) in the position \((1,m-l+1)\) and \( F'_1 \) to be the \( m \times m \) matrix with zero elements everywhere except for a 1 in the position \((m,l+1)\) and \(-\alpha_i\) in the position \((m,l)\).

Then the matrix \( \widetilde{T}_n(\equiv T_n(\beta)) \) in (1.35) can be written in the form

\[
\begin{bmatrix}
T_m(\beta - \alpha_0, \beta, \beta - \alpha_1) & -F'_1 \\
-E'_1 & T_m(\beta - \alpha_1, \beta, \beta - \alpha_2) & -F'_2 \\
0 & -E'_2 & T_m(\beta - \alpha_2, \beta, \beta - \alpha_3)
\end{bmatrix}
\]
where \( \alpha_0 = \alpha_3 = 0 \). If the number of subdomains \( k \) is more than 3, the matrix \( \tilde{T}_n \) is a block \( k \times k \) matrix of the form

\[
\tilde{T}_n = \begin{bmatrix}
S_1(\beta) & -F'_1 & 0 & 0 & \cdots & 0 & 0 \\
-E'_1 & S_2(\beta) & -F'_2 & 0 & \cdots & 0 & 0 \\
0 & -E'_2 & S_3(\beta) & -F'_3 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & -E'_{k-2} & S_{k-1}(\beta) & -F'_{k-1} \\
0 & 0 & \cdots & 0 & 0 & -E'_{k-1} & S_k(\beta)
\end{bmatrix}
\]  

(1.69)

where

\[
S_i(\beta) \equiv T_m(\beta - \alpha_{i-1}, \beta - \alpha_i), \quad i = 1, 2, \ldots, k,
\]

(1.70)

and \( \alpha_0 = \alpha_k = 0 \). Then the Multi-Parameterized Schwarz Splitting for \( \tilde{T}_n(\beta) \) is defined as

\[
\tilde{T}_n(\beta) = S_{km} - B_{km}
\]

(1.71)

where

\[
S_{km} = \text{diag}(S_1(\beta), S_2(\beta), \ldots, S_k(\beta)),
\]

(1.72)

\[
B_{km} = \begin{bmatrix}
0 & F'_1 & 0 & 0 & \cdots & 0 & 0 \\
E'_1 & 0 & F'_2 & 0 & \cdots & 0 & 0 \\
0 & E'_2 & 0 & F'_3 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & E'_{k-2} & 0 & F'_{k-1} \\
0 & 0 & \cdots & 0 & 0 & E'_{k-1} & 0
\end{bmatrix}
\]

(1.73)

### 1.3.2 Convergence Analysis

The convergence analysis of the Jacobi based Multi-Parameterized Schwarz Splitting is again reduced to calculating the spectral radius of the block Jacobi matrix \( \tilde{J} = M^{-1}N \) of \( \tilde{T}_n \) in (1.71). The \( k \times k \) block-Jacobi matrix \( \tilde{J} \) is given by

\[
\tilde{J} = \begin{bmatrix}
0 & S_1^{-1}F'_1 & 0 & 0 & \cdots & 0 & 0 \\
S_2^{-1}E'_1 & 0 & S_2^{-1}F'_2 & 0 & \cdots & 0 & 0 \\
0 & S_3^{-1}E'_2 & 0 & S_3^{-1}F'_3 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & S_{k-1}^{-1}E'_{k-2} & 0 & S_{k-1}^{-1}F'_{k-1} \\
0 & 0 & \cdots & 0 & 0 & S_{k-1}^{-1}E'_{k-1} & 0
\end{bmatrix}
\]

(1.74)

where \( S_i \equiv S_i(\beta), i = 1, 2, \ldots, k \).

In the following analysis we find matrices of smaller orders whose eigenvalues include the non-zero eigenvalues of the block Jacobi matrix \( \tilde{J} \) in (1.74).
Lemma 1.4 Let 
\[ [\delta_1^{i,j}, \delta_2^{i,j}, \ldots, \delta_m^{i,j}]^T \] 
(1.75) denote the first column of the matrix \( T_m^{-1}(\beta - \alpha_i, \beta, \beta - \alpha_j) \) and \( W \) be the \( 4(k - 1) \times 4(k - 1) \) matrix

\[
W = \begin{bmatrix}
0 & X_{10} & 0 & 0 & 0 & 0 & \cdots & 0 \\
X_{12} & 0 & 0 & Y_{21} & 0 & 0 & \cdots & 0 \\
Y_{12} & 0 & 0 & X_{21} & 0 & 0 & \cdots & 0 \\
0 & 0 & X_{23} & 0 & 0 & Y_{32} & \cdots & 0 \\
0 & 0 & Y_{23} & 0 & 0 & X_{32} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & X_{k-2,k-1} & 0 & 0 & Y_{k-1,k-2} \\
0 & 0 & \cdots & 0 & Y_{k-2,k-1} & 0 & 0 & X_{k-1,k-2} \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & X_{k-1,k} \\
\end{bmatrix}
\]

with

\[
X_{i,i+1} = \begin{bmatrix}
\delta_i^{i,i+1} & -\alpha_i \delta_i^{i,i+1} \\
\delta_i^{i,i+1} & -\alpha_i \delta_i^{i,i+1}
\end{bmatrix}, \quad X_{i,i-1} = \begin{bmatrix}
-\alpha_i \delta_i^{i-1,i} & \delta_i^{i-1,i} \\
-\alpha_i \delta_i^{i-1,i} & \delta_i^{i-1,i}
\end{bmatrix},
\]

for \( i = 1, 2, \ldots, k - 1 \), and

\[
Y_{i,i+1} = \begin{bmatrix}
\delta_{m-1}^{i,i+1} & -\alpha_i \delta_{m-1}^{i,i+1} \\
\delta_{m-1}^{i,i+1} & -\alpha_i \delta_{m-1}^{i,i+1}
\end{bmatrix}, \quad Y_{i+1,i} = \begin{bmatrix}
-\alpha_i \delta_{m-1}^{i+1,i} & \delta_{m-1}^{i+1,i} \\
-\alpha_i \delta_{m-1}^{i+1,i} & \delta_{m-1}^{i+1,i}
\end{bmatrix},
\]

for \( i = 1, 2, \ldots, k - 2 \). Then the eigenvalues of \( W \) include the non-zero eigenvalues of the block Jacobi matrix \( \tilde{J} \) in (1.74), i.e.,

\[
\sigma(\tilde{J}) = \sigma(W) \cup \{0\}.
\] (1.76)

**Proof**: We observe that all the rows of \( E_i' \) are zero except for the first one, hence only the first column in \( S_i^{-1} = T_m^{-1}(\beta - \alpha_i, \beta, \beta - \alpha_j) \) is used in computing \( S_i^{-1} E_i' \), \( i = 2, 3, \ldots, k \) and the vector in (1.75) satisfies the system of equations

\[
(\beta - \alpha_i) \delta_1^{i,j} - \delta_2^{i,j} = 1, \\
-\delta_2^{i,j} + \beta \delta_2^{i,j} - \delta_3^{i,j} = 0, \quad p = 2, \ldots, m - 1, \\
-\delta_2^{m-1,j} + (\beta - \alpha_j) \delta_m^{i,j} = 0.
\] (1.77)

With this notation and the definition of the matrix \( E_i' \), we can see that all column vectors in the matrix \( S_i^{-1} E_i' = T_m^{-1}(\beta - \alpha_i, \beta, \beta - \alpha_i) E_i' \) are zero except for the \((m - l)\)-th and \((m - l + 1)\)-st ones which are given by

\[
[\delta_1^{i-1,j}, \delta_2^{i-1,j}, \ldots, \delta_m^{i-1,j}]^T \quad \text{and} \quad -\alpha_i \delta_1^{i-1,j}, \delta_2^{i-1,j}, \ldots, \delta_m^{i-1,j}]^T,
\]
respectively. Similarly, all columns in the matrix \( S_i^{-1} F_i' = T_{m}^{-1}(\beta - \alpha_{i-1}, \beta, \beta - \alpha_{i}) \) are zero except for the \((l + 1)\)-st and \(l\)-th ones which are given by

\[
[\delta_{m}^{i-1}, \cdots, \delta_{2}^{i-1}, \delta_{1}^{i-1}]^T \text{ and } -\alpha_i [\delta_{m}^{i-1}, \cdots, \delta_{2}^{i-1}, \delta_{1}^{i-1}]^T,
\]

respectively, where we note that

\[
[\delta_{m}^{i}, \cdots, \delta_{2}^{i}, \delta_{1}^{i}]^T
\]

is the last column of \( T_{m}^{-1}(\beta - \alpha_i, \beta, \beta - \alpha_j) \). Hence the matrices \( S_i^{-1} E_{l-1}' \) and \( S_i^{-1} F_i' \) have the following forms

\[
S_i^{-1} E_{l-1}' = \begin{bmatrix}
0 & \cdots & 0 & \delta_{m}^{i-1} & -\alpha_{i-1} \delta_{1}^{i-1} & 0 & \cdots & 0 \\
0 & \cdots & 0 & \delta_{2}^{i-1} & -\alpha_{i-1} \delta_{2}^{i-1} & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & \delta_{m}^{i-1} & -\alpha_{i-1} \delta_{m}^{i-1} & 0 & \cdots & 0 \\
\end{bmatrix}_{m \times m},
\]

\[
S_i^{-1} F_i' = \begin{bmatrix}
0 & 0 & \cdots & -\alpha_{i} \delta_{m}^{i-1} & \delta_{m}^{i-1} & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & -\alpha_{i} \delta_{2}^{i-1} & \delta_{2}^{i-1} & 0 & \cdots & 0 \\
0 & 0 & \cdots & -\alpha_{i} \delta_{1}^{i-1} & \delta_{1}^{i-1} & 0 & \cdots & 0 \\
\end{bmatrix}_{m \times m}.
\]

Therefore, considering \( l < \frac{m-1}{2} \), the matrix \( \tilde{J} \) in (1.74) has exactly \( 4(k-1) \) non-zero columns. Let \( P \) be the \( km \times km \) permutation matrix that moves the columns

\[
i m - l, \ i m - l + 1, \ i m + 1, \ i m + l + 1
\]

to the columns

\[
km - 4(k - 1) + 4(i - 1) + j, \quad j = 1, 2, 3, 4,
\]

respectively, for each \( i = 1, 2, \cdots, k - 1 \). Then \( \tilde{J} \) can be transformed to \( \tilde{J}' \) as follows

\[
\tilde{J}' = P^T \tilde{J} P = \begin{bmatrix}
0 & * \\
0 & W
\end{bmatrix}
\]

and the result in (1.76) is an immediate consequence of (1.78).

The following lemma shows that there is a still smaller matrix whose eigenvalues definitely include the non-zero eigenvalues of the block Jacobi iteration matrix \( \tilde{J} \) in (1.74).

**Lemma 1.5** The eigenvalues of the matrix \( G_k \) include the non-zero eigenvalues of the matrix \( \tilde{J} \), i.e.,

\[
\sigma(\tilde{J}) = \sigma(G_k) \cup \{0\}.
\]

(1.79)
where

\[
G_k = \begin{bmatrix}
0 & x_{10} & 0 & 0 & 0 & 0 & \cdots & 0 \\
x_{12} & 0 & 0 & y_{21} & 0 & 0 & \cdots & 0 \\
y_{12} & 0 & 0 & x_{21} & 0 & 0 & \cdots & 0 \\
0 & 0 & x_{23} & 0 & 0 & y_{32} & \cdots & 0 \\
0 & 0 & y_{23} & 0 & 0 & x_{32} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & x_{k-2,k-1} & 0 & 0 & y_{k-1,k-2} \\
0 & 0 & \cdots & 0 & y_{k-2,k-1} & 0 & 0 & x_{k-1,k-2} \\
0 & 0 & \cdots & 0 & 0 & 0 & x_{k-1,k} & 0 \\
\end{bmatrix}_{2(k-1) \times 2(k-1)}
\]

where the entries of \( G_k \) are

\[
x_{ij} = \delta_{i,j}^i - \alpha_i \delta_{i,j}^i, \quad y_{ij} = \delta_{m-l}^j - \alpha_j \delta_{m-l}^j.
\]

Proof: We define the non-singular matrix

\[
Q = \text{diag}(Q_1, Q_1^T, Q_2, Q_2^T, \ldots, Q_{k-1}, Q_{k-1}^T),
\]

where

\[
Q_i = \begin{bmatrix}
1 & \alpha_i \\
0 & 1
\end{bmatrix}, \quad Q_i^{-T} = \begin{bmatrix}
1 & 0 \\
\alpha_i & 1
\end{bmatrix}, \quad Q_i^{-1} = \begin{bmatrix}
1 & -\alpha_i \\
0 & 1
\end{bmatrix}, \quad Q_i^{-T} = \begin{bmatrix}
1 & 0 \\
-\alpha_i & 1
\end{bmatrix}.
\]

Using the matrices \( Q \) and \( Q^{-1} \), \( W \) can be transformed to \( W' \) via the following similarity transformation

\[
W' = Q^{-1} W Q
\]

\[
= \begin{bmatrix}
0 & X'_{10} & 0 & 0 & 0 & 0 & \cdots & 0 \\
X'_{12} & 0 & 0 & Y'_{21} & 0 & 0 & \cdots & 0 \\
Y'_{12} & 0 & 0 & X'_{21} & 0 & 0 & \cdots & 0 \\
0 & 0 & X'_{23} & 0 & 0 & Y'_{32} & \cdots & 0 \\
0 & 0 & Y'_{23} & 0 & 0 & X'_{32} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & X'_{k-2,k-1} & 0 & 0 & y'_{k-1,k-2} \\
0 & 0 & \cdots & 0 & Y'_{k-2,k-1} & 0 & 0 & x'_{k-1,k-2} \\
0 & 0 & \cdots & 0 & 0 & 0 & x'_{k-1,k} & 0 \\
\end{bmatrix}
\]
where
\[
X'_{i,i+1} = Q_i^{-T}X_{i,i+1}Q_i = \begin{bmatrix}
\delta_{i,i+1}^{i,i+1} & 0 \\
\delta_{i+1}^{i+1} - \alpha_i \delta_{i,i+1}^{i,i+1} & 0
\end{bmatrix},
\]
\[
X'_{i,i-1} = Q_i^{-1}X_{i,i-1}Q_i^T = \begin{bmatrix}
0 & \delta_{i,i-1}^{i,i-1} - \alpha_i \delta_{i,i-1}^{i,i-1} \\
\delta_{i-1}^{i-1} - \alpha_i \delta_{i-1}^{i-1} & 0
\end{bmatrix},
\]
for \( i = 1, 2, \cdots, k - 1 \), and
\[
Y'_{i,i+1} = Q_{i+1}^{-1}Y_{i,i+1}Q_i = \begin{bmatrix}
\delta_{m-l}^{i+1} - \alpha_{i+1} \delta_{m-l+1}^{i+1} & 0 \\
\delta_{m-l+1}^{i+1} & 0
\end{bmatrix},
\]
\[
Y'_{i+1,i} = Q_i^{-T}Y_{i+1,i}Q_{i+1}^T = \begin{bmatrix}
0 & \delta_{m-l}^{i+1,i} \\
\delta_{m-l+1}^{i+1,i} - \alpha_i \delta_{m-l+1}^{i+1,i} & 0
\end{bmatrix},
\]
for \( i = 1, 2, \cdots, k - 2 \). Thus the two matrices \( W \) and \( W' \) have the same eigenvalue spectra, i.e.,
\[
\sigma(W) = \sigma(W'). \tag{1.82}
\]

We now observe that except for \( 2(k - 1) \) columns, all other columns of the matrix \( W' \) are zero vectors. Let then \( \tilde{P} \) be the \( 4(k - 1) \times 4(k - 1) \) permutation matrix that moves the columns
\[
4(i - 1) + 1, 4(i - 1) + 4
\]
to the columns
\[
2(k - 1) + 2(i - 1) + 1, 2(k - 1) + 2(i - 1) + 2,
\]
respectively, for each \( i = 1, 2, \cdots, k - 1 \). Then \( W' \) can be transformed to \( W'' \) as follows
\[
W'' = \tilde{P}^T W' \tilde{P} = \begin{bmatrix}
0 & \ast \\
0 & G_k
\end{bmatrix}.
\]

Thus the eigenvalues of the matrix \( G_k \) definitely include the non-zero eigenvalues of the matrix \( W'' \), i.e.,
\[
\sigma(W'') = \sigma(G_k) \cup \{0\}. \tag{1.83}
\]

From the relations (1.76), (1.82) and (1.83), our conclusion follows.

\[ \blacksquare \]

### 1.3.3 Determination of the Optimal Multi-Parameter Set

Having obtained the matrix \( G_k \) in (1.80) we can show that there is a choice of its elements \( x_{i,i+1} = 0, i = 1, 2, \cdots, k - 1 \) that makes all its eigenvalues equal to zero. This is given in the following lemma.

**Lemma 1.6** If \( x_{i,i+1} = 0, i = 1, 2, \cdots, k - 1 \), then \( \det(G_k - \lambda I) = \lambda^{2(k-1)} \), that is, all the eigenvalues of the matrix \( G_k \) are zero and therefore so is its spectral radius.
Proof: Our assertion will be proved by induction. It is easily checked from (1.80) that the lemma holds true for \( k = 2 \) since then \( G_2 = \begin{bmatrix} 0 & x_{10} \\ 0 & 0 \end{bmatrix} \). Assume that the lemma holds true for any \( k \geq 2 \). Then the choice

\[ x_{i,i+1} = 0, \quad i = 1, \ldots, k - 1, \]

makes \( G_k \) have all its eigenvalues zero, i.e., \( \det(G_k - \lambda I) = \lambda^{2(k-1)} \). Choose \( x_{k,k+1} = 0 \). Then the characteristic polynomial for \( G_{k+1} \) is

\[
\det(G_{k+1} - \lambda I) = \det \left( \begin{bmatrix} \begin{bmatrix} G_k - \lambda I \\ & \vdots \\ & & \vdots \\ 0 & \cdots & 0 & y_{k-1,k} & 0 \\ 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \end{bmatrix} \right) = \det(G_k - \lambda I)(-\lambda)^2 = \lambda^{2(k-1)} \lambda^2 = \lambda^{2k}.
\]

Thus the lemma holds true for \( k + 1 \), which concludes the proof.

There is another choice of the \( x_{i,j} \), namely \( x_{i,i-1} = 0, i = 1, 2, \ldots, k - 1 \), that makes all the eigenvalues zero.

Lemma 1.7 If \( x_{i,i-1} = 0, i = 1, 2, \ldots, k - 1 \), then \( \det(G_k - \lambda I) = \lambda^{2(k-1)} \), that is, all the eigenvalues of the matrix \( G_k \) are zero and therefore so is its spectral radius.

Proof: Let \( \bar{P} \) be the \( 2(k-1) \times 2(k-1) \) permutation matrix that moves the \( i \)th column to the \( (2(k-1) - i + 1) \)st one, \( i = 1, \ldots, 2(k-1) \),

\[
\bar{P} = \begin{bmatrix}
0 & 0 & \cdots & 0 & 0 & 1 \\
0 & 0 & \cdots & 0 & 1 & 0 \\
0 & 0 & \cdots & 1 & 0 & 0 \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
0 & 0 & 1 & \cdots & 0 & 0 \\
0 & 1 & 0 & \cdots & 0 & 0 \\
1 & 0 & 0 & \cdots & 0 & 0
\end{bmatrix}_{2(k-1) \times 2(k-1)}
\]

Then \( G_k \) in (1.80) can be transformed to \( G'_k \) as follows

\[
G'_k = \bar{P}^T G_k \bar{P}
\]
\[
\begin{bmatrix}
0 & x_{k-1,k} & 0 & 0 & 0 & 0 & \cdots & 0 \\
x_{k-1,k-2} & 0 & 0 & y_{k-2,k-1} & 0 & 0 & \cdots & 0 \\
y_{k-1,k-2} & 0 & 0 & x_{k-2,k-1} & 0 & 0 & \cdots & 0 \\
0 & 0 & x_{k-2,k-3} & 0 & 0 & y_{k-3,k-2} & \cdots & 0 \\
0 & 0 & y_{k-2,k-3} & 0 & 0 & x_{k-3,k-2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & x_{21} & 0 & 0 & y_{12} \\
0 & 0 & \cdots & 0 & y_{21} & 0 & 0 & x_{12} \\
0 & 0 & \cdots & 0 & 0 & 0 & 0 & x_{1,0}
\end{bmatrix}_{2(k-1)\times 2(k-1)}
\]

Applying Lemma 1.6 to the matrix \( G_k \), we have

\[
\text{if } x_{i,i-1} = 0, \ i = k - 1, \cdots, 2, 1, \text{ then } \det(G_k - \lambda I) = \lambda^{2(k-1)}.
\]

Since \( \sigma(G_k) = \sigma(G_k') \), statement (1.86) can be rewritten as

\[
\text{if } x_{i,i-1} = 0, \ i = 1, 2, \cdots, k - 1, \text{ then } \det(G_k - \lambda I) = \lambda^{2(k-1)},
\]

which proves our assertion.

Lemma 1.6 and Lemma 1.7 are particular cases of a more general result, namely:

**Lemma 1.8** If for any \( j = 0, 1, \ldots, k - 1 \),

\[
\begin{align*}
x_{i,i-1} &= 0, \ i = 1, 2, \cdots, j, \\
x_{i,i+1} &= 0, \ i = j + 1, \cdots, k - 1,
\end{align*}
\]

then \( \det(G_k - \lambda I) = \lambda^{2(k-1)} \), that is, all the eigenvalues of the matrix \( G_k \) are zero and therefore so is its spectral radius.

**Proof:** Using condition (1.87), Lemma 1.7 can be applied to the \( 2j \times 2j \) principal submatrix \( G_{j+1} \) of \( G_k \) to give

\[
\det(G_{j+1} - \lambda I) = \lambda^{2j}.
\]

Then by using the series of relationships in (1.84) with the conditions (1.89) and (1.88), we can easily obtain

\[
\det(G_k - \lambda I) = \lambda^{2(k-1)}
\]

The following proposition provides the expressions of \( \delta_{p,j} \) in (1.77), consequently those of \( x_{i,i-1}, x_{i,i+1} \) in Lemma 1.8.
Proposition 1.3  The solution \([\delta_1, \delta_2, \cdots, \delta_m]^T\) of the system of equations

\[
\begin{align*}
-\delta_{p-1} + (\beta - \alpha_1) \delta_p - \delta_{p+1} &= 0, \quad p = 2, \cdots, m-1, \\
-\delta_{m-1} + (\beta - \alpha_2) \delta_m &= 0,
\end{align*}
\]

where \(0 \leq \alpha_i < 1, \ i = 1, 2, \) and \(\beta \geq 2,\) is given by

\[
\delta_p = \begin{cases} 
\frac{\sinh((m-p+1)\theta) - \alpha_2 \sinh((m-p)\theta)}{\sinh((m+1)\theta) - (\alpha_1 + \alpha_2) \sinh(m\theta) + \alpha_1 \alpha_2 \sinh((m-1)\theta)} & \text{for } \beta > 2 \\
\frac{(m-p+1) - \alpha_2 (m-p)}{(m+1) - (\alpha_1 + \alpha_2) m - \alpha_1 \alpha_2 (m-1)} & \text{for } \beta = 2,
\end{cases}
\]

where \(\theta = \arccosh\left(\frac{\beta}{2}\right).\)

Proof: First consider the case of \(\beta > 2.\) Then for some \(\theta > 0\) we have \(\beta = e^\theta + e^{-\theta}\) since \(\beta = 2 \cosh(\theta).\) In this case, we can change the system of equations into the following one

\[
-\delta_{p-1} + (e^\theta + e^{-\theta}) \delta_p - \delta_{p+1} = 0, \quad p = 1, \cdots, m
\]

(1.90)

with boundary conditions

\[
\delta_0 = \alpha_1 \delta_1 + 1,
\]

\[
\delta_{m+1} = \alpha_2 \delta_m.
\]

(1.91)

The characteristic equation of the difference equation in (1.90) is

\[
r^2 - (e^\theta + e^{-\theta}) r + 1 = 0
\]

which has the two distinct roots \(e^\theta, e^{-\theta}.\) Thus the general solution of the difference equation (1.90) is given by

\[
\delta_p = C_1 (e^\theta)^p + C_2 (e^{-\theta})^p.
\]

(1.92)

To determine the coefficients \(C_1, C_2\) in (1.92), we apply the boundary conditions (1.91) to the general solution (1.92) to get the system of equations

\[
\begin{align*}
C_1 (1 - \alpha_1 e^\theta) + C_2 (1 - \alpha_1 e^{-\theta}) &= 1, \\
C_1 (e^{(m+1)\theta} - \alpha_2 e^{m\theta}) + C_2 (e^{-(m+1)\theta} - \alpha_2 e^{-m\theta}) &= 0.
\end{align*}
\]

Solving this system of linear equations, we obtain

\[
\begin{align*}
DC_1 &= \det \left( \begin{bmatrix} 1 & (1 - \alpha_1 e^{-\theta}) \\ 0 & (e^{-(m+1)\theta} - \alpha_2 e^{-m\theta}) \end{bmatrix} \right) = (e^{-(m+1)\theta} - \alpha_2 e^{-m\theta}), \\
DC_2 &= \det \left( \begin{bmatrix} (1 - \alpha_1 e^\theta) & 1 \\ (e^{(m+1)\theta} - \alpha_2 e^{m\theta}) & 0 \end{bmatrix} \right) = -(e^{(m+1)\theta} - \alpha_2 e^{m\theta}).
\end{align*}
\]

(1.93)
where

\[
D = \det \left( \begin{bmatrix}
1 - \alpha_1 e^\theta & 1 - \alpha_1 e^{-\theta} \\
(1 - \alpha_1 e^\theta)(e^{-(m+1)\theta} - \alpha_2 e^{m\theta}) & (1 - \alpha_1 e^{-\theta})(e^{-(m+1)\theta} - \alpha_2 e^{-m\theta})
\end{bmatrix} \right)
\]

\[
= -(e^{(m+1)\theta} - \alpha_2 e^{m\theta}) + (e^{-(m+1)\theta} - \alpha_2 e^{-m\theta}) - \alpha_1 \left[ -(e^{m\theta} - \alpha_2 e^{(m-1)\theta}) + (e^{-m\theta} - \alpha_2 e^{-(m-1)\theta}) \right]
\]

\[
= -2 \left( \sinh((m+1)\theta) - (\alpha_1 + \alpha_2) \sinh(m\theta) - \alpha_1 \alpha_2 \sinh((m-1)\theta) \right) \tag{1.94}
\]

Note that \( D < 0 \), in (1.94), because \( 0 \leq \alpha_i < 1, \ i = 1, 2 \). Substituting \( DC_1, DC_2 \) in (1.93) into the general solution (1.92), we have

\[
D \delta_p = DC_1 e^{\rho \theta} + DC_2 e^{-\rho \theta}
\]

\[
= -2 \left( \sinh((m-p+1)\theta) - \alpha_2 \sinh((m-p)\theta) \right) \tag{1.95}
\]

Substituting \( D \) of (1.94) into (1.95), we conclude that

\[
\delta_p = \frac{\sinh((m-p+1)\theta) - \alpha_2 \sinh((m-p)\theta)}{\sinh((m+1)\theta) - (\alpha_1 + \alpha_2) \sinh(m\theta) - \alpha_1 \alpha_2 \sinh((m-1)\theta)}.
\]

For the case of \( \beta = 2 \), the characteristic equation for the difference equation in (1.90) has only one solution, \( r = 1 \). So the general solution is of the form

\[
\delta_p = C_1 + C_2 p.
\]

Similar steps to the previous ones lead to

\[
\delta_p = \frac{(m-p+1) - \alpha_2 (m-p)}{(m+1) - (\alpha_1 + \alpha_2) m - \alpha_1 \alpha_2 (m-1)},
\]

which concludes the proof. \( \blacksquare \)

Based on the above lemmas and proposition, we have the following theorem.

**Theorem 1.2** Let \( \theta = \text{arccosh}(\frac{\beta}{2}) \) with \( \beta = 2 + qh^2 \) defined in (1.24) and let the values \( \alpha_i, i = 0, 1, \ldots, k \), be defined as follows:

For \( q > 0 \) (i.e., \( \theta > 0 \)):

\[
\alpha_0 = 0,
\]

\[
\alpha_i = \frac{\sinh((m-l)\theta) - \alpha_{i-1} \sinh((m-l-1)\theta)}{\sinh((m-l+1)\theta) - \alpha_{i-1} \sinh((m-l)\theta)}, \quad i = 1, 2, \ldots, j,
\]

\[
\alpha_i = \frac{\sinh((m-l)\theta) - \alpha_{i-1} \sinh((m-l-1)\theta)}{\sinh((m-l+1)\theta) - \alpha_{i+1} \sinh((m-l)\theta)}, \quad i = j+1, \ldots, k-1,
\]

\[
\alpha_k = 0.
\]

For \( q = 0 \) (i.e., \( \theta = 0 \)):

\[
\alpha_0 = 0,
\]

\[
\alpha_i = \frac{(m-l) - \alpha_{i-1} (m-l-1)}{(m-l+1) - \alpha_{i-1} (m-l)}, \quad i = 1, 2, \ldots, j,
\]

\[
\alpha_i = \frac{(m-l) - \alpha_{i+1} (m-l)}{(m-l+1) - \alpha_{i+1} (m-l)}, \quad i = j+1, \ldots, k-1,
\]

\[
\alpha_k = 0.
\]
for any \( j = 0, 1, \ldots, k - 1 \). Then the spectral radius of the matrix \( G_k \) is zero, implying that the spectral radius of the block Jacobi matrix \( \tilde{J} \) in (1.74) of \( MPSEM(T_n) \) is also zero.

**Proof:** From Proposition 1.3, we have that

\[
\delta_{i,j} = \begin{cases} 
\frac{\sinh((m-p+1)\theta) - \alpha_j \sinh((m-p)\theta)}{\sinh((m+1)\theta) - (\alpha_i + \alpha_j) \sinh(m\theta) + \alpha_i \alpha_j \sinh((m-1)\theta)} & , \theta > 0 \\
\frac{(m-p+1)\alpha_i - \alpha_j}{(m+1) - (\alpha_i + \alpha_j)m + \alpha_i \alpha_j (m-1)} & , \theta = 0.
\end{cases}
\]

Note that the case \( \theta = 0 \) can be obtained from the case \( \theta > 0 \) by a limiting process argument allowing \( \theta \to 0^+ \). From the definition of \( x_{ij} \)'s in (1.81) and for the given \( \alpha_i \)'s, we get, for \( i = 1, 2, \ldots, j \),

\[
x_{i,i-1} = \delta_{i+1,i} - \alpha_i \delta_{i,i-1} = \begin{cases} 
\frac{\sinh((m-I)\theta) - \alpha_{i-1} \sinh((m-I-1)\theta) - \alpha_i \sinh((m-I+1)\theta) - \alpha_{i-1} \sinh((m-I)\theta)}{\sinh((m+1)\theta) - (\alpha_i + \alpha_{i-1}) \sinh(m\theta) + \alpha_i \alpha_{i-1} \sinh((m-1)\theta)} & , \theta > 0 \\
\frac{(m-I) - \alpha_{i-1} (m-I-1) - \alpha_i (m-I+1) - \alpha_{i-1} (m-I)}{(m+1) - (\alpha_i + \alpha_{i-1})m + \alpha_i \alpha_{i-1} (m-1)} & , \theta = 0.
\end{cases}
\]

Similarly, we can obtain that \( x_{i,i+1} = 0 \), for \( i = j + 1, \ldots, k - 1 \). Since the conditions of Lemma 1.8 are satisfied, all the eigenvalues of the matrix \( G_k \) are zero. Hence by virtue of (1.79), the conclusion of the statement follows.

1.4 **Numerical Experiments**

In this section we attempt to measure experimentally the convergence factor of the Classical Schwarz Splitting (SS), the One-Parameter Schwarz Splitting (1PSS), and the Multi-Parameterized Schwarz Splitting (MPSS) methods with different domain splittings. Although we do not present the results here, we have verified Tang's numerical results in [Tan92] using his two-point BVP example in the 1PSS. For the numerical results presented in this thesis we have selected a different two-point BVP (His example does not have a first order term "\(-4u\)"). The two-point boundary value problem is

\[
u''(t) - 4u = 4 \cosh(1), \quad t \in (0, 1),
\]

\[
u(0) = 0, \quad u(1) = 0
\]

whose solution is

\[
u(t) = \cosh(2t - 1.0) - \cosh(1.0).
\]

In all the experiments we take as initial guess of the solution the vector with all its components \(-0.25\). The value \(-0.25\) is midway between the two extreme values of the function \( u(t) \), namely, \( u(0.5) = -0.54 \cdots \) and \( u(0.0) = u(1.0) = 0.0 \). The convergence
factor is computed as the p-th root of relative $\ell_2$ norms of the residual of the corresponding system of equations after p iterations (i.e., $\sqrt[\ell_2]{||Ax^{(p)} - f||_2/||Ax^{(0)} - f||_2}$).

First, in Table 1.1 we show the convergence factor of SS computed after 3, 4 and 8 iterations for different domain splittings, overlaps, and local grid sizes. The results indicate slow convergence.

Second, we present the convergence factor for the 1PSS method. It is worth recalling that Tang [Tan92] found experimentally the optimal value of the parameter of this method for $k = 3$. As we have seen in Section 1.2, we have found the simple equations (1.66), (1.67) that the optimal values of 1PSS satisfy for any value of $k$. In the case of $k = 2$ and 3 the formulas can be solved explicitly while for $k \geq 4$ we solve them numerically. Table 1.2 indicates the computed single parameter value and the convergence factor of the method computed after $k$ iterations where $k$ is the number of subdomains. Notice that in case $k = 3$ our theoretical value of $\alpha$ coincides with the numerical value found by Tang in [Tan92].

Finally, our experiments indicate that MPSS achieves a convergence to relative $\ell_2$ norm of $2 \times 10^{-15}$ after $k$ iterations where $k$ is the number of splittings. This is consistent for all $k$ tried up to $k = 64$. Table 1.3 gives the exact parameters predicted by the theory presented in the previous sections. It is clear that MPSS achieves a rapid convergence within a very small number of iterations. The convergence rate is very sensitive to the computed optimal value of parameter $\alpha_i$'s and the symmetric choice of them reduces the error propagation when we compute the optimal value of parameters $\alpha_i$'s.

The data suggest that 1PSS is faster than SS but slower than MPSS.

---

Table 1.1 The convergence factors of the SS method applied to the problem (1.96) for minimum(min) and half(1/2) overlap for $k = 3, 4$, and 8 domain splittings and for two local grid sizes $m = 10$ and 20. The convergence factors are calculated after $k$ iterations.

<table>
<thead>
<tr>
<th>Num. of subdom. Method (local grid)</th>
<th>3</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS min ($m=10$)</td>
<td>0.55</td>
<td>0.57</td>
<td>0.71</td>
</tr>
<tr>
<td>SS $1/2$ ($m=10$)</td>
<td>0.63</td>
<td>0.63</td>
<td>0.75</td>
</tr>
<tr>
<td>SS min ($m=20$)</td>
<td>0.46</td>
<td>0.51</td>
<td>0.69</td>
</tr>
<tr>
<td>SS $1/2$ ($m=20$)</td>
<td>0.62</td>
<td>0.63</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Table 1.2 The number of iterations, convergence factors, and the parameters of the 1PSS method applied to the boundary value problem (1.96). Two different local grids are used ($m = 10, 20$) and three different domain splittings ($k = 3, 4, 8$). The parameter is computed by numerically solving equations (1.66) and (1.67).

<table>
<thead>
<tr>
<th>Num. of subdom.</th>
<th>Method (local grid)</th>
<th>3</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1PSS min($m=10$)</td>
<td>1.4E-5</td>
<td>1.5E-1</td>
<td>4.8E-1</td>
</tr>
<tr>
<td></td>
<td>1PSS 1/2($m=10$)</td>
<td>1.3E-5 (0.844)</td>
<td>2.1E-1 (0.851)</td>
<td>5.2E-1 (0.907)</td>
</tr>
<tr>
<td></td>
<td>1PSS min($m=20$)</td>
<td>1.3E-5 (0.943)</td>
<td>1.4E-1 (0.947)</td>
<td>4.2E-1 (0.963)</td>
</tr>
<tr>
<td></td>
<td>1PSS 1/2($m=20$)</td>
<td>1.3E-5 (0.909)</td>
<td>2.1E-1 (0.914)</td>
<td>5.0E-1 (0.955)</td>
</tr>
</tbody>
</table>

Table 1.3 The number of iterations and $[k - 1]/2 + 1$ of the parameters (selected to be symmetric) of the MPSS method which give convergence to relative $l_2$ norm of 2E-15. The convergence factors are also included. Two different local grids are used ($m = 10, 20$) and three different domain splittings ($k = 3, 4, 8$). The MPSS results are for boundary value problem (1.96).

<table>
<thead>
<tr>
<th>Num. of subdom.</th>
<th>Method (local grid)</th>
<th>i = 1</th>
<th>i = 2</th>
<th>i = 3</th>
<th>i = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1PSS min($m=10$)</td>
<td>(0.887)</td>
<td>(0.929)</td>
<td>(0.793)</td>
<td>(0.983)</td>
</tr>
<tr>
<td></td>
<td>1PSS 1/2($m=10$)</td>
<td>(0.844)</td>
<td>(0.848)</td>
<td>(0.906)</td>
<td>(0.855)</td>
</tr>
<tr>
<td></td>
<td>1PSS min($m=20$)</td>
<td>(0.943)</td>
<td>(0.946)</td>
<td>(0.967)</td>
<td>(0.949)</td>
</tr>
<tr>
<td></td>
<td>1PSS 1/2($m=20$)</td>
<td>(0.909)</td>
<td>(0.912)</td>
<td>(0.947)</td>
<td>(0.915)</td>
</tr>
</tbody>
</table>

Convergence factor: 1.3E-5, 1.8E-4, 1.3E-2

1.5 Remarks and Discussion

The significance of the Parameterized Schwarz Methods is its potential as a parallel computation framework for solving linear initial/boundary value problems. Another important result is the relation of the rate of convergence of the numerical SAM method to the mesh size of the discretization scheme. It is known (see [Dry89], [OST86], [Tan87]) that for model problems with Dirichlet interface conditions and a fixed aspect ratio of the overlapping area over the subdomains, the rate of convergence of numerical SAM does not depend on the mesh size. This is not true when mixed
interface conditions are used [Tan92]. However, our investigation has shown that there are one-dimensional BV problems where the rate of convergence does not change with the mesh size even for mixed type interface conditions with appropriately chosen convex combinations of Dirichlet and Neumann boundary conditions.

So far we have studied SAM at a discrete algebraic level for a class of multi-parameter boundary value problems and finite difference discretization schemes. Our study generalizes, extends and improves the recent results presented in [Tan87] and [Tan92]. It appears that these results are among a few that study SAM at the discrete matrix equation level. However, there is plethora of studies (see the proceedings of the SIAM conferences on Domain Decomposition Methods) about SAM at various functional levels.

Our work is related to that of [EKCS87], [RKL89], [RS89], [Kan89], and to some of the references cited in them. With the exception of [RKL89], these papers approach the SAM analysis at a functional level. Specifically, paper [EKCS87] deals with 1-D and 2-D boundary value problems assuming a 2-way domain decomposition ($k = 2$), where the values of the approximate solution along the two artificial boundaries are linear combinations of the two previous available ones (iterations). The theoretical and experimental results obtained in [EKCS87] are weaker than ours. This is readily seen by simply comparing the values of the optimal convergence factor ranging from 0.339 to 0.887 (third column of Table 1 in [EKCS87]) against ours. In our case this factor is zero (spectral radius of the block Jacobi iteration matrix).

Paper [RKL89] deals with $k$-way ($k \geq 2$) decompositions of 2-D boundary value problems with Dirichlet boundary conditions along the artificial boundaries. General convergence results (not explicit formulas) are given for Jacobi and/or Gauss-Seidel inner/outer iterative methods. Unfortunately, for mixed boundary value conditions on the artificial boundaries, the regular splitting theory of [RKL89] is not applicable to our case. This is mainly due to the fact that the corresponding $N$ matrices in the $M-N$ regular splittings are not nonnegative matrices as the regular splitting theory requires. It is worth noticing that in our case the non-zero elements of the $N$ matrices are either 1 or $-\alpha_i$ with $\alpha_i \in [0,1)$. Obviously, only for $\alpha_i = 0$ (Dirichlet boundary conditions on the artificial boundaries) the $N$ matrices above are nonnegative. Paper [RS89] applies SAM on 2 and 3-way ($k = 2,3$) decompositions of 2-D boundary value problems. Although mixed boundary value conditions on the artificial boundaries are allowed, they are restricted to cases of Dirichlet/Dirichlet, Dirichlet/Neumann and Neumann/Neumann only. In our work general mixed interface conditions without restrictions are allowed. Finally, paper [Kan89] presents a survey of author's previous works on SAM approach. The results of Theorem 3.6 and 3.7 in [Kan89] seem to be close to our result of Theorem 1.2 in Section 1.3.3. The main difference is that the results in [Kan89] are obtained at a functional equation level, i.e. they use Fourier analysis to determine a set of parameters to make the convergence factor of SAM be zero, while we determine the corresponding set of parameters by examining the spectral radius of the Jacobi iteration matrix of MPSS.
Chapter 2
Multi-Parameterized Schwarz Splitting for Multi-Dimensional Problems

The basic analysis of the Parameterized Schwarz Splitting with the one parameter case for two-dimensional problems was presented in [Tan92]. In this chapter, we develop a similar analysis for two-dimensional problems using a set of parameters \(a_i, i = 1, 2, \ldots, k - 1\), with \(k\) being the number of subdomains. Then we solve the following open problem:

**Problem:** Determine the values of \(a_i\)'s for which the spectral radius of the block Jacobi iteration matrix of the Multi-Parameterized Schwarz Splitting is as small as possible.

This problem was completely solved in the one-dimensional case as was seen in Section 1.3. However, it is still an unsolved problem in the two-dimensional case even when only one parameter, i.e., \(a_i = a, i = 1, 2, \ldots, k - 1\), is used.

Consider the two-dimensional boundary value problem (BVP)

\[
Lu = -\nabla^2 u(x) + q u(x) = f(x), \quad x \in \Omega,
\]

\[
u(x)|_{\Gamma} = g(x)
\]

where \(\Gamma\) is the boundary of \(\Omega \equiv (0,1) \times (0,1)\) and \(q\) is a constant with \(q \geq 0\). We formulate a SAM based on a k-way splitting of the domain \(\Omega\), i.e., we decompose our domain into \(k\) overlapping subdomains \(\Omega_i\) along the \(x_1\)-axis and make a strip-type decomposition of the rectangular domain \(\Omega\) (for instance, see Figure 2.1). Next we apply the mixed interface conditions (1.16) on the two artificial boundaries between subdomains \(\Omega_i\) and \(\Omega_{i+1}\). Let \(\ell\) be the length of the overlap in \(x_1\)-direction and \(\eta\) be the length of each subdomain in the same direction. Figure 2.1 depicts such a 2-way splitting of the unit square \(\Omega\).

### 2.1 Formulation of the Multi-Parameterized Schwarz Splitting for Two-Dimensional Problem

To begin our analysis we use a 5-point finite difference discretization scheme with uniform grid of mesh size \(h = \frac{1}{n+1}\) and then we discretize the BVP in (2.1) to obtain
a linear system of the form

$$A \mathbf{x} = f.$$  \hspace{1cm} (2.2)

The natural ordering of the nodes is adopted starting from the origin and going in the $x_2$-direction first so that the resulting matrix $A$ can be partitioned into block matrices corresponding to the subdomains, respectively.

Using tensor product notation (see [Hal58] and also [LRT64] in which tensor products in connection with BVP's were used for the first time), the matrix $A$ in (2.2) can be written as

$$A = T_n(\beta) \otimes I_n + I_n \otimes T_n(2)$$  \hspace{1cm} (2.3)

where $\beta = 2 + qh^2$ and the $T_j(x)$ is defined in (1.22).

Define $l + 1 = \frac{\ell}{h}$ and $m + 1 = \frac{\eta}{h}$ so that $n = mk - l(k - 1)$ and $l < \frac{m-1}{2}$. As in the one-dimensional case in Section 1.3, the two-dimensional Multi-Parameterized SAM transforms the matrix $A$ in (2.3) into the corresponding Schwarz Enhanced matrix $\tilde{A}$ with parameters $\alpha_i$. More specifically we have

$$\tilde{A} = T_n(\beta) \otimes I_n + I_{km} \otimes T_n(2).$$  \hspace{1cm} (2.4)

Based on (2.4), the Multi-Parameterized Schwarz Splitting for $\tilde{A}$ in (2.4) is defined as

$$\tilde{A} = M - N$$  \hspace{1cm} (2.5)

with

$$M = S_{km} \otimes I_n + I_{km} \otimes T_n(2),$$

$$N = B_{km} \otimes I_n,$$  \hspace{1cm} (2.6)

where $S_{km}$ and $B_{km}$ were defined in (1.72) and (1.73), respectively.
2.2 Convergence Analysis

The convergence analysis of the Multi-Parameterized Schwarz Splitting is reduced to determining the spectral radius of the block Jacobi matrix

$$\tilde{J} = M^{-1}N$$

of $\tilde{A}$ in (2.5). To begin our analysis, we state and prove two lemmas.

**Lemma 2.1** Let $A$ and $B$ be $m \times m$ and $n \times n$ matrices, respectively. Then there exists an $(mn) \times (mn)$ permutation matrix $P$ such that $P(A \otimes B)P^{-1} = B \otimes A$.

**Proof:** The permutation matrix $P$ is the matrix that moves the rows

$$(i - 1)n + j$$

to the rows

$$(j - 1)m + i,$$

for every $i = 1, 2, \cdots, m$ and for every $j = 1, 2, \cdots, n$. ■

**Lemma 2.2** The matrix $\tilde{J} = M^{-1}N$ in (2.7) is similar to the matrix

$$\text{diag}(J_1, J_2, \cdots, J_n)$$

where

$$J_i = (\text{diag}(S_1(\beta + \gamma_i), S_2(\beta + \gamma_i), \cdots, S_k(\beta + \gamma_i)))^{-1} B_{km},$$

$$\gamma_i = 2 + 2 \cos\left(\frac{\pi i}{n+1}\right)$$

for $i = 1, 2, \cdots, n$, where $S_j(x), j = 1, 2, \cdots, k$ are defined in (1.70).

**Proof:** Let $X_n$ be the $n \times n$ orthogonal matrix whose columns are the eigenvectors of the matrix $T_n(2)$. Since the eigenvalues of the matrix $T_n(2)$ are known to be $\gamma_i = 2 + 2 \cos(\frac{\pi i}{n+1}), i = 1, 2, \cdots, n$, we can write

$$X_n^T T_n(2) X_n = D_n \equiv \text{diag}(\gamma_1, \gamma_2, \cdots, \gamma_n).$$

Let $I_{km}$ be the identity matrix of order $km (= k \times m)$ and let $X = I_{km} \otimes X_n$, then its inverse is given by $X^{-1} = I_{km} \otimes X_n^T$. Using $X$, we can construct a new matrix $\tilde{J}'$, which is similar to the matrix $\tilde{J}$, as follows

$$\tilde{J}' = X^{-1} \tilde{J} X = X^{-1} M^{-1} N X = (X^{-1} M X)^{-1} (X^{-1} N X).$$

However, if we replace $X$ and $M$ by their tensor product representations and perform simple operations, we obtain

$$X^{-1} M X = (I_{km} \otimes X_n^T) (S_{km} \otimes I_n) (I_{km} \otimes X_n)$$

$$+ (I_{km} \otimes X_n^T) (I_{km} \otimes T_n(2)) (I_{km} \otimes X_n)$$

$$= S_{km} \otimes (X_n^T I_n X_n) + I_{km} \otimes (X_n^T T_n(2) X_n)$$

$$= S_{km} \otimes I_n + I_{km} \otimes D_n.$$
Similarly,
\[ X^{-1}NX = (I_{km} \otimes X_n^T)(B_{km} \otimes I_n)(I_{km} \otimes X_n) \]
\[ = ((I_{km})B_{km}(I_{km})) \otimes (X_n^T I_n X_n). \]
By Lemma 2.1, there exists a permutation matrix \( P \) such that
\[ \tilde{f}^n = P \tilde{f} P^{-1} \]
\[ = P(X^{-1}MX)^{-1}(X^{-1}NX)P^{-1} \]
\[ = (P(S_{km} \otimes I_n)P^{-1} + P(I_{km} \otimes D_n)P^{-1})^{-1} P(B_{km} \otimes I_n)P^{-1} \]
\[ = (I_n \otimes S_{km} + D_n \otimes I_{km})^{-1} (I_n \otimes B_{km}). \]
On the other hand we have
\[ I_n \otimes S_{km} + D_n \otimes I_{km} \]
\[ = I_n \otimes \text{diag}(S_1(\beta), S_2(\beta), \cdots, S_k(\beta)) + \text{diag}(\gamma_1, \gamma_2, \cdots, \gamma_n) \otimes I_k \otimes I_m \]
\[ = \text{diag}(\text{diag}(S_1(\beta + \gamma_1), S_2(\beta + \gamma_1), \cdots, S_k(\beta + \gamma_1)), \cdots, \text{diag}(S_1(\beta + \gamma_n), S_2(\beta + \gamma_n), \cdots, S_k(\beta + \gamma_n)). \]
So the result (2.8)–(2.9) follows immediately.

From Lemma 2.2, we see that each submatrix \( J_i \) in (2.9) has the same form as the Jacobi matrix (1.74) in the one-dimensional case in (1.3). All submatrices in (2.8) are related to the same set of parameters \( \alpha_i, i = 1, 2, \cdots, k - 1 \). However, the entries of any submatrix in (2.8) are different from those of the other submatrices, hence a set of parameters \( \alpha_i, i = 1, 2, \cdots, k - 1 \), which minimizes the spectral radius of one submatrix will not necessarily minimize those of the other submatrices in (2.8). It follows that in order to minimize the spectral radius of the whole matrix in (2.8) we must find the set of \( \alpha_i \)'s which minimizes the maximum of all spectral radii of the submatrices in (2.8).

### 2.3 On the Determination of the Optimal Parameters

From Lemma 2.2 and Lemma 1.5, we know that the spectral radius of \( \tilde{f} \) in (2.7) is
\[ \rho(\tilde{f}) = \max(\rho(G_k^1), \rho(G_k^2), \cdots, \rho(G_k^n)) \tag{2.11} \]
where \( G_k^i, i = 1, 2, \cdots, n, \) is the matrix \( G_k \) in (1.80) with
\[ x_{ij} = \delta_{i+1,j} - \alpha_i \delta_{i,j}, \quad y_{ij} = \delta_{i,j} - \alpha_i \delta_{i+1,j}, \]
\[ \delta_{i,j}^p = \frac{\sinh((m-p+1)\theta_i) - \alpha_i \sinh((m-p)\theta_i)}{\sinh((m+1)\theta_i) - (\alpha_i + \alpha_j) \sinh(m\theta_i) + \alpha_i \alpha_j \sinh((m-1)\theta_i)}, \]
\[ \theta_i = \arccosh(\frac{2+\alpha_i}{2}). \tag{2.12} \]
Note that $\rho(G_k^2)$ is a function of $\alpha_1, \alpha_2, \ldots, \alpha_{k-1}$. Our goal is to determine the optimal values of $\alpha_1, \alpha_2, \ldots, \alpha_{k-1}$ which minimize the spectral radius $\rho(\tilde{J})$ in (2.11).

In the case of two subdomains ($k = 2$), our Multi-Parameterized SS case is reduced to the One-Parameter SS case with $\alpha_1 = \alpha$. From Remark 1.1 and Lemma 1.3, we know that the spectral radius of $G_k^2$, $i = 1, 2, \ldots, n$, is given by

$$\rho(G_k^2(\alpha)) = |\varphi_i(\alpha)|$$

(2.13)

with $\alpha = \alpha_1$ and that

$$\varphi_i(\alpha) = \frac{\sinh((m - l)\theta_i) - \alpha \sinh((m - l + 1)\theta_i)}{\sinh((m + 1)\theta_i) - \alpha \sinh(m\theta_i)},$$

(2.14)

where $\theta_i = \text{arccosh}(\frac{l + n}{2})$. In the following lemmas, we investigate the properties of the functions $\varphi_i(\alpha), i = 1, 2, \ldots, n$.

**Lemma 2.3** Each function $\varphi_i, i = 1, 2, \ldots, n$, is strictly decreasing in the interval $(0, 1)$ with $\varphi_i(0) > 0$ and $\varphi_i(1) < 0$. Therefore each equation $\varphi_i(\alpha) = 0, i = 1, 2, \ldots, n$, has a unique solution, say $\alpha_i$, in the interval $(0, 1)$.

**Proof:** From the definition of $\varphi_i$ in (2.14), we can compute the derivative of $\varphi_i$ in the unit interval $(0, 1)$ as follows

$$\varphi_i'(\alpha) = \frac{\sinh((m - l)\theta_i)\sinh(m\theta_i) - \sinh((m - l + 1)\theta_i)\sinh((m + 1)\theta_i)}{\sinh((m + 1)\theta_i) - \alpha \sinh(m\theta_i))^2}$$

$$= \frac{\cosh((2m - l)\theta_i) - \cosh((2m - l + 2)\theta_i)}{2((m + 1)\theta_i) - \alpha \sinh(m\theta_i))^2} < 0,$$

for all $\alpha \in (0, 1)$, where the identity $2 \sinh(A) \sinh(B) = \cosh(A + B) - \cosh(A - B)$ is used. Hence the function $\varphi_i$ is strictly decreasing in $(0, 1)$. Since $\theta_i > 0$ and $0 < m - l < m - l + 1 \leq m < m + 1$, we obtain

$$\varphi_i(0) = \frac{\sinh((m - l)\theta_i)}{\sinh((m + 1)\theta_i)} > 0$$

$$\varphi_i(1) = \frac{\sinh((m - l)\theta_i) - \sinh((m - l + 1)\theta_i)}{\sinh((m + 1)\theta_i) - \sinh(m\theta_i)} < 0$$

**Lemma 2.4** If one defines $\varphi_i^+(\alpha) = \varphi_n(\alpha) + \varphi_i(\alpha)$ for $\alpha \in [0, 1], i = 1, 2, \ldots, n - 1$, then each function $\varphi_i^+$ is strictly decreasing in the interval $(0, 1)$ with $\varphi_i^+(0) > 0$ and $\varphi_i^+(1) < 0$. Therefore each equation $\varphi_i^+(\alpha) = 0, i = 1, 2, \ldots, n - 1$, has a unique solution, say $\alpha_i^+$, in the interval $(0, 1)$.

**Proof:** The sum $\varphi_i^+$ of two strictly decreasing functions is also strictly decreasing. From Lemma 2.3, we have that $\varphi_n(0) > 0$, $\varphi_i(0) > 0$ and $\varphi_n(1) < 0$, $\varphi_i(1) < 0$. Hence $\varphi_i^+(0) = \varphi_n(0) + \varphi_i(0) < 0$ and $\varphi_i^+(1) = \varphi_n(1) + \varphi_i(1) < 0$. 


Proposition 2.1  If $\varepsilon \in (0,1)$ is fixed, the functions

$$
\phi_1(\theta) = \frac{\cosh(\varepsilon \theta)}{\cosh(\theta)} \quad \text{and} \quad \phi_2(\theta) = \frac{\sinh(\varepsilon \theta)}{\sinh(\theta)}
$$

are strictly decreasing functions of $\theta \in (0, \infty)$.

Proof: The derivatives of the functions $\phi_1, \phi_2$ are given by

$$
\phi'_1(\theta) = \frac{\varepsilon \sinh(\varepsilon \theta) \cosh(\theta) - \cosh(\varepsilon \theta) \sinh(\theta)}{\cosh^2(\theta)}, \quad (2.15)
$$

$$
\phi'_2(\theta) = \frac{\varepsilon \cosh(\varepsilon \theta) \sinh(\theta) - \sinh(\varepsilon \theta) \cosh(\theta)}{\sinh^2(\theta)}. \quad (2.16)
$$

The denominators $\cosh^2(\theta), \sinh^2(\theta)$ in (2.15), (2.16), respectively, are positive for all non-zero values of $\theta$. Using the identity

$$
2 \sinh(\lambda) \cosh(\mu) = \sinh(\lambda + \mu) + \sinh(\lambda - \mu),
$$

we find for the numerator in (2.15) that

$$
2(\varepsilon \sinh(\varepsilon \theta) \cosh(\theta) - \cosh(\varepsilon \theta) \sinh(\theta))
$$

$$
= \varepsilon(\sinh((1 + \varepsilon)\theta) + \sinh((\varepsilon - 1)\theta)) - (\sinh((1 + \varepsilon)\theta) + \sinh((1 - \varepsilon)\theta))
$$

$$
= -(1 - \varepsilon) \sinh((1 + \varepsilon)\theta) - (1 + \varepsilon) \sinh((1 - \varepsilon)\theta)) < 0.
$$

Similarly we find for the numerator in (2.16) that

$$
2(\varepsilon \cosh(\varepsilon \theta) \sinh(\theta) - \sinh(\varepsilon \theta) \cosh(\theta))
$$

$$
= \varepsilon(\sinh((1 + \varepsilon)\theta) + \sinh((\varepsilon - 1)\theta)) - (\sinh((1 + \varepsilon)\theta) + \sinh((1 - \varepsilon)\theta))
$$

$$
= -(1 - \varepsilon) \sinh((1 + \varepsilon)\theta) + (1 + \varepsilon) \frac{1 - \varepsilon}{1 + \varepsilon} \sinh((1 + \varepsilon)\theta)) \quad < 0.
$$

For the inequality above, we have used, with $\lambda = \frac{1 - \varepsilon}{1 + \varepsilon} \in (0,1)$, the two extreme terms of the relationships

$$
\sinh((1 - \varepsilon)\theta) = \sinh((1 - \lambda)\theta + \lambda(1 + \varepsilon)\theta)
$$

$$
< (1 - \lambda) \sinh(0) + \lambda \sinh((1 + \varepsilon)\theta)
$$

$$
= \frac{1 - \varepsilon}{1 + \varepsilon} \sinh((1 + \varepsilon)\theta))
$$

Note that the strict inequality holds above because $\sinh(\theta)$ is a strictly convex function of $\theta \in (0, \infty)$. 

Lemma 2.5  If one defines $\varphi_i^- (\alpha) = \varphi_n(\alpha) - \varphi_i(\alpha)$ for $\alpha \in [0,1]$, $i = 1, 2, \cdots, n - 1$, then we have $\varphi_i^- (0) > 0$ and $\varphi_i^- (1) < 0$ and each equation $\varphi_i^- (\alpha) = 0, i = 1, 2, \cdots, n - 1$ has a unique solution, say $\alpha_i^-$, in the interval $(0,1)$.
Proof: From the definition of \( \varphi^{-}_i, \varphi_n \) and \( \varphi_l \), we obtain

\[
\varphi^{-}_i(0) = \varphi_n(0) - \varphi_l(0) = \frac{\sinh((m-l)\theta_n)}{\sinh((m+1)\theta_n)} - \frac{\sinh((m-l)\theta_l)}{\sinh((m+1)\theta_l)}.
\]

By Proposition 2.1 we see that \( \frac{\sinh((m-n)\theta)}{\sinh((m+1)\theta)} \) is a strictly decreasing function with respect to \( \theta \). Since \( \theta_n < \theta_l \), it follows that \( \varphi^{-}_i(0) > 0 \). We also have

\[
\varphi^{-}_i(1) = \varphi_n(1) - \varphi_l(1)
\]

\[
= \frac{\sinh((m-l)\theta_n) - \sinh((m-l+1)\theta_n)}{\sinh((m+1)\theta_n) - \sinh(m\theta_n)} - \frac{\sinh((m-l)\theta_l) - \sinh((m-l+1)\theta_l)}{\sinh((m+1)\theta_l) - \sinh(m\theta_l)}
\]

\[= \frac{\cosh((m-l+\frac{1}{2})\theta_n) \sinh(-\frac{1}{2}\theta_n)}{\cosh((m+\frac{1}{2})\theta_n) \sinh(\frac{1}{2}\theta_n)} - \frac{\cosh((m-l+\frac{1}{2})\theta_l) \sinh(-\frac{1}{2}\theta_l)}{\cosh((m+\frac{1}{2})\theta_l) \sinh(\frac{1}{2}\theta_l)} \tag{2.17}
\]

By virtue of the identity

\[\sinh(A) - \sinh(B) = 2 \cosh(\frac{A+B}{2}) \sinh(\frac{A-B}{2}),\]

we see that (2.17) may be expressed as

\[
\varphi^{-}_i(1) = -\frac{\cosh((m-l+\frac{1}{2})\theta_n)}{\cosh((m+\frac{1}{2})\theta_n)} + \frac{\cosh((m-l+\frac{1}{2})\theta_l)}{\cosh((m+\frac{1}{2})\theta_l)}.
\]

By Proposition 2.1 we see that \( \frac{\cosh((m-l+\frac{1}{2})\theta)}{\cosh((m+\frac{1}{2})\theta)} \) is a strictly decreasing function with respect to \( \theta \). Since \( \theta_n < \theta_l \), it follows that \( \varphi^{-}_i(1) < 0 \). From the definition of \( \varphi^{-}_i, \varphi_n \) and \( \varphi_l \), we have

\[
\varphi^{-}_i(\alpha) = \varphi_n(\alpha) - \varphi_l(\alpha) = \frac{q_i(\alpha)}{p_i(\alpha)}
\]

where

\[
p_i(\alpha) = \frac{(\sinh((m+1)\theta_l) - \alpha \sinh(m\theta_l)) (\sinh((m+1)\theta_n) - \alpha \sinh(m\theta_n))}{\sinh(m\theta_l) \sinh((m-l+1)\theta_n) - \sinh(m\theta_n) \sinh((m-l+1)\theta_l)},
\]

\[
q_i(\alpha) = \alpha^2 + A_i \alpha + B_i.
\]

The denominator \( p_i(\alpha) \) of \( \varphi^{-}_i(\alpha) \) is positive for \( 0 \leq \alpha \leq 1 \), so \( \varphi^{-}_i(\alpha) \) and \( q_i(\alpha) \) have the same sign for all \( \alpha \in [0,1] \). Therefore

\[
q_i(0) > 0 \quad \text{and} \quad q_i(1) < 0
\]

since \( \varphi^{-}_i(0) > 0 \) and \( \varphi^{-}_i(1) < 0 \), respectively. However, \( q_i(\alpha) \) is a quadratic, therefore there exists a unique value of \( \alpha \) in \( (0,1) \), say \( \alpha^{-}_i \), for which \( q_i(\alpha) \) vanishes. So then does \( \varphi^{-}_i(\alpha) \) for the same value of \( \alpha = \alpha^{-}_i \). Also we have \( \varphi^{-}_i > 0 \) in \( [0,\alpha^{-}_i \] and \( \varphi^{-}_i < 0 \) in \( (\alpha^{-}_i,1] \).
Lemma 2.6 If $\alpha_i, \alpha_i^+, \alpha_i^-$, $i = 1, 2, \ldots, n - 1$, are defined to be the solutions of the equations $\varphi_i(\alpha) = 0$, $\varphi_i^+(\alpha) = 0$, $\varphi_i^-(\alpha) = 0$, respectively, then we have

$$\alpha_i < \alpha_{i+1} \text{ and } \alpha_i < \alpha_i^+ < \alpha_n < \alpha_i^-,$$

for each $i = 1, 2, \ldots, n - 1$.

Proof: Since the solution $\alpha_i$ of $\varphi_i(\alpha) = 0$, $i = 1, 2, \ldots, n$, is given by

$$\alpha_i = \frac{\sinh((m - l)\theta_i)}{\sinh((m - l + 1)\theta_i)},$$

from Proposition 2.1, we have

$$\alpha_i < \alpha_{i+1}, \quad i = 1, 2, \ldots, n - 1,$$

which implies

$$\alpha_i < \alpha_n, \quad i = 1, 2, \ldots, n - 1.$$

Using Lemma 2.3 we obtain

$$\varphi_n(\alpha_i) > 0 \text{ and } \varphi_i(\alpha_n) < 0. \quad (2.18)$$

From the definitions of $\varphi_i, \varphi_i^+, \varphi_i^-$ and $\alpha_i$, we have

$$\varphi_i^+(\alpha_i) = \varphi_n(\alpha_i) + \varphi_i(\alpha_i) = \varphi_n(\alpha_i),$$
$$\varphi_i^+(\alpha_n) = \varphi_n(\alpha_n) + \varphi_i(\alpha_n) = \varphi_i(\alpha_n),$$
$$\varphi_i^-(\alpha_n) = \varphi_n(\alpha_n) - \varphi_i(\alpha_n) = -\varphi_i(\alpha_n). \quad (2.19)$$

From (2.18) and (2.19), we then obtain

$$\varphi_i^+(\alpha_i) > 0, \quad \varphi_i^+(\alpha_n) < 0 \text{ and } \varphi_i^-(\alpha_n) > 0. \quad (2.20)$$

which imply, by Lemma 2.4 and Lemma 2.5, that

$$\alpha_i < \alpha_i^+ < \alpha_n < \alpha_i^-.$$

Theorem 2.1 Let $\alpha_i^+$ be the solution of the equation $\varphi_i^+(\alpha) = 0$, $i = 1, 2, \ldots, n - 1$, in $(0, 1)$. Then the optimal value $\alpha^o$ of $\alpha$, which minimizes the spectral radius of the matrix $\rho(\tilde{J}) = \rho(\tilde{J}(\alpha))$ in (2.7), is given by

$$\alpha^o = \min\{\alpha_i^+: i = 1, 2, \ldots, n - 1\}$$

Proof: Let $i_0$ be the index so that $\alpha^o = \alpha_{i_0}^+$. We will show that

$$\rho(G_2^o(\alpha^o)) \leq \max\left(\rho(G_2^1(\alpha)), \rho(G_2^2(\alpha)), \ldots, \rho(G_2^n(\alpha))\right), \quad (2.21)$$
for each of three cases $\alpha \in [0, \alpha_i^+)\), $\alpha \in (\alpha_i^+, 1]$ and $\alpha = \alpha_i^+$. Then our assertion will follow from (2.11) and (2.21).

case 1) : $\alpha \in [0, \alpha_i^+)$. We have

$$\alpha < \alpha_i^+, \ i = 1, 2, \cdots, n - 1, \quad \text{(2.22)}$$

because $\alpha_i^+ < \alpha_i^+$, and hence by Lemma 2.6 we have

$$\alpha < \alpha_i^-, \ i = 1, 2, \cdots, n - 1. \quad \text{(2.23)}$$

By Lemmas 2.4 and 2.5, (2.22) and (2.23) we have that

$$\varphi_i^+ (\alpha) > 0 \text{ and } \varphi_i^- (\alpha) > 0, \ i = 1, 2, \cdots, n - 1,$$

i.e.,

$$|\varphi_n (\alpha)| > |\varphi_i (\alpha)|, \ i = 1, 2, \cdots, n - 1,$$

Consequently, we have

$$|\varphi_n (\alpha)| = \max (|\varphi_1 (\alpha)|, |\varphi_2 (\alpha)|, \cdots, |\varphi_n (\alpha)|). \quad \text{(2.24)}$$

On the other hand, by Lemma 2.6 we know $\alpha_i^+ < \alpha_n$ and hence by Lemma 2.3 we obtain

$$|\varphi_n (\alpha_i^+)| < |\varphi_n (\alpha)|. \quad \text{(2.25)}$$

From (2.24) and (2.25), we conclude that

$$|\varphi_n (\alpha_i^+)| \leq \max (|\varphi_1 (\alpha)|, |\varphi_2 (\alpha)|, \cdots, |\varphi_n (\alpha)|),$$

which implies (2.21) by (2.13).

case 2) : $\alpha \in (\alpha_i^+, 1]$. By Lemma 2.3

$$\varphi_i (\alpha) < \varphi_n (\alpha_i^+). \quad \text{(2.26)}$$

Since $\alpha_i^+ < \alpha_n$ by Lemma 2.5, we have, by Lemma 2.3,

$$\varphi_n (\alpha_i^+) < 0. \quad \text{(2.27)}$$

From (2.26) and (2.27), we obtain

$$|\varphi_i (\alpha_i^+)| < |\varphi_i (\alpha)|. \quad \text{(2.28)}$$

Since $\varphi_n (\alpha_i^+) + \varphi_i (\alpha_i^+) = 0$ by the definition of $\alpha_i^+$, we have

$$|\varphi_i (\alpha_i^+)| = |\varphi_n (\alpha_i^+)|. \quad \text{(2.29)}$$

Then (2.28) and (2.29) imply

$$|\varphi_n (\alpha_i^+)| < |\varphi_i (\alpha)|.$$
Since \( 1 \leq i_0 \leq n - 1 \), we can write
\[
|\varphi_n(\alpha_{i_0}^+)| \leq \max(|\varphi_1(\alpha)|, |\varphi_2(\alpha)|, \ldots, |\varphi_n(\alpha)|),
\]
which implies (2.21) by (2.13).

**case \( 5) : \alpha = \alpha_{i_3}^+ \)**. It is obvious that
\[
|\varphi_n(\alpha_{i_3}^+) | \leq \max(|\varphi_1(\alpha)|, |\varphi_2(\alpha)|, \ldots, |\varphi_n(\alpha)|),
\]
which implies (2.21) by (2.13).

For the case of three subdomains \((k = 3)\), we can compute the spectral radii of the matrices \(G_3^t, t = 1, 2, \ldots, n\) in (2.11) from the expression of \(G_3\) in Lemma 1.5 as follows
\[
\rho(G_3^t) = \max \left( \sqrt{\frac{1}{2} \left[ x_{10}x_{12} + x_{21}x_{23} \pm \sqrt{x_{10}^2x_{12}^2 + x_{21}^2x_{23}^2 + 2x_{10}x_{23}(2y_{12}y_{21} - x_{12}x_{21})} \right]} \right) \tag{2.30}
\]
with \(x_{ij}, y_{ij}, \delta p^{ij}, \theta_i\) as given in (2.12). Since \(\delta p^{ij}\) is a function of the parameters \(\alpha_1, \alpha_2\), so is \(G_3^t\). It seems very difficult to determine analytically the optimal values of \(\alpha_1, \alpha_2\) which minimize the spectral radius
\[
\rho(\bar{J}) = \max(\rho(G_3^1), \rho(G_3^2), \ldots, \rho(G_3^n))
\]
of \(\bar{J}\) in (2.7) even if we only consider just the one-parameter case, i.e., \(\alpha_1 = \alpha_2\), for which the expression (2.30) reduces to
\[
\rho(G_3^t) = \max \left( \sqrt{\left| g_1 \pm g_2 \right|} \right)
\]
(see (1.68)), where \(g_1, g_2, g_3\) are defined in (1.47) with \(\theta(= \theta_1) = \arccosh(\frac{\theta + \theta_2}{2})\). This is something we intend to investigate further in the near future.

### 2.4 Numerical Experiments

In this section, we present a numerical experiment for the case of \(k = 2\), in order to confirm the analysis in Theorem 2.1. For this, consider the following model problem
\[
-\nabla^2 u(x, y) = 0, \quad (x, y) \in \Omega = (0, 1) \times (0, 1),
\]
\[
u(x, y) = f(x, y), \quad (x, y) \in \Gamma,
\]
where \(\Gamma\) is the boundary of \(\Omega\) and the solution of (2.31) is
\[
f(x, y) = \sin(2\pi x) \cos(2\pi y).
\]

We report on experiment for the problem (2.31) using \(k = 2, m = 6, l = 1\) and \(n = 11\). Figure 2.2 shows the spectral radii of the eleven submatrices \(J_i(\alpha)\) in (2.8).
Using Theorem 2.1, the optimal value of $\alpha$ can be calculated numerically as $\alpha = 0.654$. Figure 2.3 shows the number of the block Jacobi iterations required to reduce the $\ell_2$-norm of the residual by a factor of $10^{-4}$. Figure 2.4 shows the ratio of the $\ell_2$-norm of the residual relative to its initial norm after five block Jacobi iterations. From Figures 2.3 and 2.4, we see that both the smallest number of iterations and the smallest relative $\ell_2$-norm of the residual are achieved near the value $\alpha = 0.654$ confirming the theoretical analysis done so far.

Figure 2.2 Spectral radii of the blocks $\tilde{J}_i(\alpha)$ in the Jacobi iteration matrix $\tilde{J}(\alpha)$ of two-dimensional problem.
Figure 2.3 Number of iterations versus SS parameter $\alpha$ required to reduce the $\ell_2$-norm of the residual by a factor of $10^{-4}$.

Figure 2.4 The relative $\ell_2$-norm of residual versus SS parameter $\alpha$ after five block Jacobi iterations.
Figure 2.5 A 2-way splitting of the three-dimensional unit cube.

2.5 Multi-Parameterized Schwarz Splitting for Higher \((p \geq 3)\) Dimensional Problems

In this section, we extend the analysis of the two-dimensional Multi-Parameterized Schwarz Splitting of Section 2.1 to that of the \(p \geq 3\) dimensional case. The analysis of the \(p\)-dimensional Schwarz Splitting is very similar to that of the two-dimensional case. The \(p\)-dimensional model problem is the boundary value problem (BVP) defined as follows

\[
Lu \equiv -\nabla^2 u(x) + qu(x) = f(x), \quad x \in \Omega,
\]

\[
u(x)|_{\Gamma} = g(x)
\]

where \(q\) is a constant with \(q \geq 0\) and \(\Gamma\) is the boundary of the \(p\)-dimensional unit cube \(\Omega \equiv \prod_{i=1}^{p}(0, 1)\).

As in the two-dimensional case, we formulate the SAM based on a \(k\)-way splitting of the domain \(\Omega\), i.e., we decompose the domain into \(k\) overlapping subdomains \(\Omega_i\) along the first component (\(x_1\)-direction) axis with overlapping and make a strip-type decomposition on the \(p\)-dimensional unit cube domain \(\Omega\). Next we apply the mixed interface condition (1.16) on the two artificial boundaries between subdomain \(\Omega_i\) and \(\Omega_{i+1}\). Let \(\ell\) be the length of the overlap in the first component direction (\(x_1\)-direction) and \(\eta\) be the length of each subdomain in the same direction. Figure 2.5 depicts an instance of the above splitting for a three-dimensional domain with two subdomains.
2.5.1 Formulation of the Multi-Parameterized Schwarz Splitting

Using a \((2p+1)\)-point finite difference discretization scheme with a uniform grid of mesh size \( h = \frac{1}{n+1} \), we discretize the BVP in (2.32) to obtain a linear system of the form

\[
Ax = f.
\]

Using the tensor product notation, the matrix \( A \) in (2.33) can be written as

\[
A = T_n(\beta) \otimes I_{n^{p-1}} + I_n \otimes \sum_{i=1}^{p-1} (I_{n^{i-1}} \otimes T_n(2) \otimes I_{n^{p-i-1}})
\]

where \( \beta = 2 + qh^2 \) and the \( T_j(x) \) is defined in (1.22).

Define \( l + 1 = \frac{L}{h} \) and \( m + 1 = \frac{M}{h} \) so that \( n = mk - l(k - 1) \) and \( l < \frac{m-1}{2} \). As in the two-dimensional case, the \( p \)-dimensional Multi-Parameterized Schwarz Splitting of \( A \) in (2.33) is defined as

\[
\tilde{A} = M - N
\]

where

\[
M = S_{km} \otimes I_{n^{p-1}} + I_{km} \otimes \sum_{i=1}^{p-1} (I_{n^{i-1}} \otimes T_n(2) \otimes I_{n^{p-i-1}}),
\]

\[
N = B_{km} \otimes I_{n^{p-1}}
\]

and \( S_{km} \) and \( B_{km} \) are defined in (1.72) and (1.73), respectively.

2.5.2 Convergence Analysis

The convergence analysis of the Multi-Parameterized Schwarz Splitting is reduced to determining the spectral radius of the block Jacobi matrix \( \bar{J} = M^{-1}N \) of \( \tilde{A} \) in (2.35).

Lemma 2.7 The matrix \( \bar{J} \) defined above is similar to the matrix

\[
\text{diag}(J_1, J_2, \ldots, J_{n^{p-1}})
\]

where

\[
J_i = \text{diag}(S_i(\beta + \gamma_i), S_2(\beta + \gamma_i), \ldots, S_k(\beta + \gamma_i))^{-1} B_{km},
\]

\[
\gamma_i = \sum_{j=1}^{p-1} \left( 2 + 2 \cos \left( \frac{1}{n+1} \left( \frac{j-1}{n+1} \right) \pi \right) \right).
\]

for \( i = 1, 2, \ldots, n^{p-1} \).
Proof: The proof follows that of the two-dimensional case. Let $X$ be the following tensor product

$$X \equiv I_{km} \otimes \prod_{j=1}^{p-1} (X_n)_j$$

where $(X_n)_i$'s are copies of $X_n$ in (2.10). Given the matrices $M$ and $N$ in (2.36), we obtain

$$X^{-1}MX = S_{km} \otimes I_{n^p-1} + I_{km} \otimes \sum_{j=1}^{p-1} (I_{n^j-1} \otimes D_n \otimes I_{n^p-j-1}),$$

$$X^{-1}NX = B_{km} \otimes I_{n^p-1}.$$

The same steps as in the two-dimensional case can be made to obtain the following matrix $\tilde{J}$ similar to the block Jacobi matrix $\tilde{J}$

$$\tilde{J} = (I_{n^p-1} \otimes S_{km} + (\sum_{j=1}^{p-1} (I_{n^j-1} \otimes D_n \otimes I_{n^p-j-1})) \otimes I_{km})^{-1} (I_{n^p-1} \otimes B_{km}),$$

from which the result can be derived. \qed
Chapter 3
Symmetric Schwarz Splitting

Symmetric matrices often provide advantages over non-symmetric matrices in numerical computation. A basic observation is that the Schwarz Enhanced Matrix (see (1.12) in Section 1.1.3) of a symmetric matrix is not a symmetric matrix. In Section 3.1 we define the Symmetric Schwarz Enhanced Matrix (SSEM) and the Symmetric Schwarz Enhanced Equation (SSEE) and present some of their basic properties. SEM and SEE have much in common with SEM and SEE. In this chapter, we extend some of the basic theorems for SEM and SEE in [Tan87] to the symmetric cases SSEM and SSEE.

3.1 Symmetric Schwarz Enhanced Matrix (SSEM) and Equation (SSEE)

To preserve possible symmetry of the original matrix, we give the following definition of a Symmetric Schwarz Enhanced Matrix, using an idea suggested by Ribbens [Rib90]. Consider the matrix equation in (1.11) where we assume that the matrix $A$ is real symmetric. Then the Symmetric Schwarz Enhanced Equation (SSEE) is defined for three subdomains by

$$
\tilde{A} = \begin{bmatrix}
A_{11} & \frac{1}{2}A_{12} & \frac{1}{2}A_{12} & A_{13} & \frac{1}{2}A_{14} & \frac{1}{2}A_{14} & A_{15} \\
\frac{1}{2}A_{21} & 0 & \frac{1}{2}A_{22} & \frac{1}{2}A_{22} & 0 & \frac{1}{2}A_{24} & \frac{1}{2}A_{24} \\
\frac{1}{2}A_{31} & \frac{1}{2}A_{32} & \frac{1}{2}A_{32} & A_{33} & \frac{1}{2}A_{34} & \frac{1}{2}A_{34} & A_{35} \\
\frac{1}{2}A_{41} & \frac{1}{2}A_{42} & \frac{1}{2}A_{42} & \frac{1}{2}A_{43} & \frac{1}{2}A_{44} & \frac{1}{2}A_{44} & A_{45} \\
\frac{1}{2}A_{51} & \frac{1}{2}A_{52} & \frac{1}{2}A_{52} & A_{53} & \frac{1}{2}A_{54} & \frac{1}{2}A_{54} & A_{55}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5
\end{bmatrix} =
\begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
f_4 \\
f_5
\end{bmatrix} = \tilde{f} \quad (3.1)
$$

where the matrix $\tilde{A}$ is called the Symmetric Schwarz Enhanced Matrix (SSEM). From this particular case, we see how we can define the Symmetric Schwarz Enhanced Equation and the Symmetric Schwarz Enhanced Matrix for the general case of a $(2k - 1) \times (2k - 1)$ block matrix $(k \geq 1)$ in (1.13).

Definition 3.1 Let $I_i$ denotes the identity matrix of the same order as $A_{ii}$, $i = 1, 2, \cdots, 2k - 1$. If $X$ is a $(2k - 1) \times (2k - 1)$ block matrix like $A$ in (1.13), then $\tilde{X}$ is defined to be the matrix $X = \tilde{I}X$ with $\tilde{I} = \operatorname{diag}(I_1, \frac{1}{2}I_2, I_3, \frac{1}{2}I_4, \cdots, \frac{1}{2}I_{2k-2}, I_{2k-1})$.
Definition 3.2 If $X$ is a $(2k - 1) \times (2k - 1)$ block matrix like $A$ in (1.13), then $\overline{X}$ is defined to be the $(k - 1) \times (k - 1)$ block matrix consisting of all even blocks $X_{2i, 2j}$, $i, j = 1, 2, \ldots, k - 1$.

From equation (3.1), we note that if a matrix $A$ is real symmetric, so is its SSEM $\hat{A}$. In the following we prove a basic theorem about preserving the positive definiteness of a matrix $A$.

**Theorem 3.1** If a matrix $A$ is real symmetric positive definite, then so is its SSEM $\hat{A}$.

**Proof:** Assume that $\bar{x} = [x_{1}^{T}, x_{2}^{T}, x_{3}^{T}, x_{4}^{T}, x_{5}^{T}, \ldots, x_{2k-2}^{T}, x_{2k-1}^{T}]^{T}$ is any non-zero vector partitioned in accordance with the block partitioning of $A$. Then we have

$$
\bar{x}^{T} \hat{A} \bar{x} = \sum_{i=1}^{k} \sum_{j=1}^{k} x_{2i-1}^{T} (\sum_{j=1}^{k} A_{2i-1, 2j-1} x_{2j-1} + \sum_{j=1}^{k} \frac{1}{2} A_{2i-1, 2j} x_{2j}) + \sum_{i=1}^{k} x_{2i}^{T} (\sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j-1} x_{2j-1} + \sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j} x_{2j}^{'} ) \\
+ \sum_{i=1}^{k} x_{2i}^{T} (\sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j-1} x_{2j-1} + \sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j} x_{2j} ) \\
+ \sum_{i=1}^{k} x_{2i}^{T} (\sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j-1} x_{2j-1} + \sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j} x_{2j}^{'} ) \\
+ \sum_{i=1}^{k} x_{2i}^{T} (\sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j-1} x_{2j-1} + \sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j} x_{2j}^{'} ) \\
= \frac{1}{2} \sum_{i=1}^{k} x_{2i-1}^{T} (\sum_{j=1}^{k} A_{2i-1, 2j-1} x_{2j-1} + \sum_{j=1}^{k} \frac{1}{2} A_{2i-1, 2j} x_{2j}) \\
+ \frac{1}{2} \sum_{i=1}^{k} x_{2i}^{T} (\sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j-1} x_{2j-1} + \sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j} x_{2j} ) \\
+ \frac{1}{2} \sum_{i=1}^{k} x_{2i}^{T} (\sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j-1} x_{2j-1} + \sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j} x_{2j}^{'} ) \\
+ \frac{1}{2} \sum_{i=1}^{k} x_{2i}^{T} (\sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j-1} x_{2j-1} + \sum_{j=1}^{k} \frac{1}{2} A_{2i, 2j} x_{2j}^{'} ) \\
= \frac{1}{2} x^{T} Ax + \frac{1}{2} x^{T} Ax^{'} \quad (3.2)
$$

where $x = [x_{1}^{T}, x_{2}^{T}, x_{3}^{T}, x_{4}^{T}, x_{5}^{T}, \ldots, x_{2k-2}^{T}, x_{2k-1}^{T}]^{T}$ and $x^{'} = [x_{1}^{T}, x_{2}^{T}, x_{3}^{T}, x_{4}^{T}, x_{5}^{T}, \ldots, x_{2k-2}^{T}, x_{2k-1}^{T}]^{T}$. Since $\bar{x}$ is non-zero, at least one of $x, x^{'}$ is non-zero. Since $A$ is positive definite, at least one term in the final expression of (3.2) is positive and the other is non-negative. Thus $\bar{x}^{T} \hat{A} \bar{x}$ is positive and the result is established.

From the definition of the SSEM, the following lemma can be proved easily.

**Lemma 3.1** If the vector $x = [x_{1}^{T}, x_{2}^{T}, \ldots, x_{2k-1}^{T}]^{T}$ is a solution of $Ax = f$, then its augmented vector (see (1.14))

$$
\bar{x} = [x_{1}^{T}, x_{2}^{T}, x_{3}^{T}, x_{4}^{T}, x_{5}^{T}, \ldots, x_{2k-2}^{T}, x_{2k-1}^{T}]^{T}
$$

is a solution of $\hat{A} \bar{x} = \hat{f}$. 
3.2 Equivalence Theorems

As in the previous chapters, we denote the spectrum and spectral radius of a matrix $A$ by $\sigma(A)$ and $\rho(A)$, respectively. We then have the following relation between the spectrum of a matrix $A$ and that of its SSEM $\hat{A}$.

**Lemma 3.2** If $A$ is symmetric, then we have

$$\sigma(A) \subseteq \sigma(\hat{A}) \subseteq \sigma(A) \cup \sigma((\overline{A}))$$

**Proof:** If $\lambda$ is an eigenvalue of $\hat{A}$ then there exists an eigenvector $x = [x^T_1, x^T_2, \ldots, x^T_{2k-1}]^T$ such that

$$\hat{A}x = \lambda x. \tag{3.3}$$

Using the block-component notation for $x$, (3.3) can be written as follows

$$\sum_{j=1}^{2k-1} A_{2i-1,j} x_j = \lambda x_{2i-1}, \quad i = 1, 2, \ldots, k, \tag{3.4}$$

$$\sum_{j=1}^{2k-1} \frac{1}{2} A_{2i,j} x_j = \lambda x_{2i}, \quad i = 1, 2, \ldots, k - 1.$$

Rewriting (3.4), we obtain

$$\sum_{j=1}^{k} A_{2i-1,j-1} x_{2j-1} + \sum_{j=1}^{k-1} \frac{1}{2} A_{2i-1,2j} x_{2j} + \sum_{j=1}^{k-1} \frac{1}{2} A_{2i-1,2j} x_{2j} = \lambda x_{2i-1}, \quad i = 1, 2, \ldots, k,$$

$$\sum_{j=1}^{k} \frac{1}{2} A_{2i,j-1} x_{2j-1} + \sum_{j=1}^{k-1} \frac{1}{2} A_{2i,2j} x_{2j} = \lambda x_{2i}, \quad i = 1, 2, \ldots, k - 1,$$

which are equivalent to

$$\hat{A} \vec{x} = \lambda \vec{x}$$

where $\vec{x} = [x^T_1, x^T_2, x^T_3, x^T_4, x^T_5, \ldots, x^T_{2k-2}, x^T_{2k-1}]^T$. Therefore $\lambda$ is an eigenvalue of $\hat{A}$ corresponding to the eigenvector $\vec{x}$.

For the proof of the second inclusion, we assume that

$$\hat{A} \vec{x} = \lambda \vec{x} \tag{3.5}$$

where $\vec{x} = [x^T_1, x^T_2, x^T_3, x^T_4, x^T_5, \ldots, x^T_{2k-2}, x^T_{2k-1}]^T$. Using the block-component notation for $\vec{x}$, (3.5) can be written as follows

$$\sum_{j=1}^{k} A_{2i-1,j-1} x_{2j-1} + \sum_{j=1}^{k-1} \frac{1}{2} A_{2i-1,2j} x_{2j} + \sum_{j=1}^{k-1} \frac{1}{2} A_{2i-1,2j} x'_{2j} = \lambda x_{2i-1}, \quad i = 1, 2, \ldots, k, \tag{3.6}$$

$$\sum_{j=1}^{k} \frac{1}{2} A_{2i,j-1} x_{2j-1} + \sum_{j=1}^{k-1} \frac{1}{2} A_{2i,2j} x_{2j} = \lambda x_{2i}, \quad i = 1, 2, \ldots, k - 1, \tag{3.7}$$
\[ \sum_{j=1}^{k} \frac{1}{2} A_{2i,2j-1} x_{2j-1} + \sum_{j=1}^{k-1} \frac{1}{2} A_{2i,2j} x_{2j} = \lambda x_{2i}', \quad i = 1, 2, \cdots, k - 1. \] (3.8)

If \( x_{2i} = x_{2i}' \) for all \( i = 1, 2, \cdots, k - 1 \), then (3.6), (3.7) and (3.8) are reduced to (3.4), which implies \( \lambda \in \sigma(\hat{A}) \). If \( x_{2i_0} = x_{2i_0}' \) for some \( i_0 \in \{1, 2, \cdots, k - 1\} \), then, by subtracting (3.8) from (3.7), we have

\[ \sum_{j=1}^{k-1} \frac{1}{2} A_{2i,2j} (x_{2j} - x_{2j}') = \lambda (x_{2i} - x_{2i}'), \quad i = 1, 2, \cdots, k - 1, \]

i.e., \( \tilde{A}_i \tilde{y} = \lambda \tilde{y} \) with \( \tilde{y} = [(x_2 - x_2')^T, (x_4 - x_4')^T, \cdots, (x_{2k-2} - x_{2k-2}')^T]^T \). Since \( (x_{2i_0} - x_{2i_0}') \neq 0 \), \( \lambda \) is an eigenvalue of \( \tilde{A} \).

We define SSE (3.1) as being equivalent to the original matrix equation (1.11) if \( \hat{A}^{-1} \) exists and the solution vector \( \hat{x} \) is an augmented vector of the solution \( x \) of (1.11). Similarly, for the SSEM, we say that \( \hat{A} \) is equivalent to the matrix \( A \) if and only if \( \hat{A}^{-1} \) exists.

**Theorem 3.2** If \( A \) is a non-singular symmetric matrix, then the following statements are equivalent:

i) The matrix \( \hat{A} \) is equivalent to the matrix \( A \),

ii) \( 0 \notin \sigma(\hat{A}) \).

**Proof:**

i) \( \Rightarrow \) ii): If zero is an eigenvalue of \( \hat{A} \), then it is also an eigenvalue of \( \hat{A} \) and the row vectors of \( \hat{A} \) are linearly dependent. Since the matrix \( \hat{A} \) is a principal submatrix of the matrix \( \hat{A} \), the row vectors of \( \hat{A} \) are linearly dependent too. Hence \( \hat{A} \) is not invertible which contradicts our assumption.

ii) \( \Rightarrow \) i): Since \( A \) is non-singular, by our hypothesis, so is \( \hat{A} = \hat{I}A \). Hence, zero is not an eigenvalue of \( \hat{A} \). On the other hand \( 0 \notin \sigma(\hat{A}) \) implies that zero is not an eigenvalue of \( \hat{A} = \hat{I}\hat{A} \). Therefore by Lemma 3.2, zero cannot be an eigenvalue of \( \hat{A} \), i.e., \( \hat{A} \) is invertible.

Since \( \hat{A} \) is a principal submatrix of \( A \) and any principal submatrix of a positive definite matrix or an \( M \)-matrix \([\text{Var}62, \text{p.85}]\) is also a positive definite matrix or an \( M \)-matrix, respectively, we have the following corollaries.

**Corollary 3.1** The SSEM of any real symmetric positive definite matrix \( A \) is equivalent to \( A \).

**Corollary 3.2** The SSEM of any non-singular \( M \)-matrix \( A \) is equivalent to \( A \).
3.3 Splittings of the Symmetric Schwarz Enhanced Matrices

Let $A$ be a non-singular block matrix in (1.13). To apply an iterative method for the solution of the matrix equation $Ax = b$, we split the matrix $A$ so that

$$A = M - N.$$  \hspace{1cm} (3.9)

If $M$ is non-singular, (3.9) is said to be a splitting of the matrix $A$. Every splitting leads to the first order iterative method

$$Mx^{i+1} = Nx^i + f.$$  

The matrix $M^{-1}N$ is called the iteration matrix of the splitting (3.9). The convergence of the iterative scheme above is guaranteed if and only if $\rho(M^{-1}N) < 1$. In what follows we prove a basic relationship between the spectrum of the iteration matrix and that of the SSEM. Consider the same block partition for the matrices $M$ and $N$ as that of $A$ in (3.9). Particularly, assume that

$$A_{ij} = M_{ij} - N_{ij}$$  \hspace{1cm} (3.10)

for each $i, j = 1, 2, \cdots, 2k - 1$. Then the spectrum of $M^{-1}N$ and that of $(\overline{M})^{-1}\overline{N}$ are related as follows.

**Theorem 3.3** Assume $A$ is symmetric and its SSEM splitting is given by (3.9) and (3.10). Then we have

$$\sigma(M^{-1}N) \subseteq \sigma((\overline{M})^{-1}\overline{N}) \subseteq \sigma(M^{-1}N) \cup \sigma((\overline{M})^{-1}\overline{N}).$$

**Proof**: If $\lambda$ is an eigenvalue of $M^{-1}N$, then there exists a non-zero vector $x = [x_1^T, x_2^T, \cdots, x_{2k-1}^T]^T$ such that

$$Nx = \lambda Mx.$$  \hspace{1cm} (3.11)

However, the augmented vector

$$\bar{x} = [x_1^T, x_2^T, x_2^T, x_3^T, x_4^T, x_4^T, x_5^T, \cdots, x_{2k-2}^T, x_{2k-2}^T, x_{2k-1}^T]^T$$

satisfies

$$\overline{N}\bar{x} = \lambda \overline{M}\bar{x},$$

which implies that $\lambda$ is an eigenvalue of $(\overline{M})^{-1}\overline{N}$.

Conversely, assume that

$$\overline{N}\bar{x} = \lambda \overline{M}\bar{x},$$  \hspace{1cm} (3.11)

for some non-zero vector

$$\bar{x} = [x_1^T, x_2^T, x_2^T, x_3^T, x_4^T, x_4^T, x_5^T, \cdots, x_{2k-2}^T, x_{2k-2}^T, x_{2k-1}^T]^T.$$  \hspace{1cm} (3.12)
If $x_{2i} = x'_{2i}$ for all $i = 1, 2, \ldots, k - 1$, in (3.12), then (3.11) implies

$$Nx = \lambda Mx$$

with $x = [x_1^T, x_2^T, x_3^T, \ldots, x_{2k-2}^T, x_{2k-1}^T]^T$. If $x_{2i_0} \neq x'_{2i_0}$ for some $i_0 \in \{1, 2, \ldots, k - 1\}$, by subtracting the $3i$-th block-row from the $(3i - 1)$-st one, for each $i = 1, 2, \ldots, k - 1$, we obtain that

$$\sum_{j=1}^{k-1} \frac{1}{2} N_{2i,2j} (x_{2j} - x'_{2j}) = \lambda \sum_{j=1}^{k-1} \frac{1}{2} M_{2i,2j} (x_{2j} - x'_{2j}), \quad i = 1, 2, \ldots, k - 1,$$

i.e.,

$$\frac{1}{2} \overline{N} \overline{y} = \frac{1}{2} \lambda \overline{M} \overline{y}$$

with $\overline{y} = [(x_2 - x'_2)^T, (x_4 - x'_4)^T, \ldots, (x_{2k-2} - x'_{2k-2})^T]^T$. Since $(x_{2i_0} - x'_{2i_0})$ is non-zero, $\lambda$ is an eigenvalue of $(\overline{M})^{-1} \overline{N}$.

As in the case of SEM, we can define a Schwarz Splitting for SSEM.

**Definition 3.3** \(\hat{A} = \overline{M}_S - \overline{N}_S\) is a Schwarz Splitting (SS) of the SSEM for \(A\) if

$$\overline{M}_S = \text{diag}(S_1, S_2, \ldots, S_k)$$

where

$$S_1 = \begin{bmatrix} A_{11} & \frac{1}{2} A_{12} \\ \frac{1}{2} A_{21} & \frac{1}{2} A_{22} \end{bmatrix}$$

$$S_i = \begin{bmatrix} \frac{1}{2} A_{2i-2,2i-2} & \frac{1}{2} A_{2i-2,2i-1} & \frac{1}{2} A_{2i-2,i} & \frac{1}{2} A_{2i-1,ii-1} & \frac{1}{2} A_{2i-1,2i} \\ \frac{1}{2} A_{2i-1,2i-2} & \frac{1}{2} A_{2i-1,2i-1} & \frac{1}{2} A_{2i-1,2i} \end{bmatrix}, \quad i = 2, \ldots, k - 1$$

$$S_k = \begin{bmatrix} \frac{1}{2} A_{2k-2,2k-2} & \frac{1}{2} A_{2k-2,2k-1} \\ \frac{1}{2} A_{2k-1,2k-2} & A_{2k-1,2k-1} \end{bmatrix}.$$  

**Lemma 3.3** Let \(S\) be a non-singular M-matrix such that

$$S = \begin{bmatrix} S_{12} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}$$

and let

$$S^o = \begin{bmatrix} 0 & 0 & 0 \\ 0 & S_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Then \(S + S^o\) is also a non-singular M-matrix.
Proof: Since \( S \) is a non-singular M-matrix, by the statement \((M_{35})\) in [BP79, p.137], \( S \) has all its diagonal elements positive and there exists a positive diagonal matrix \( D \) such that \( SD \) is strictly diagonally dominant. It is obvious that \( S + S^o \) has all its diagonal elements positive and \((S + S^o)D = SD + S^oD\) is strictly diagonally dominant. Therefore \( S + S^o \) is a non-singular M-matrix. ■

The above proof was provided to us by Professor Michael Neumann of the University of Connecticut whose kindness is gratefully acknowledged.

Theorem 3.4 The SS of SSEM of any non-singular M-matrix is a convergent splitting, i.e., \( \rho \left( (\tilde{M})^{-1} \tilde{N} \right) < 1 \).

Proof: Let \( D \) be the diagonal matrix consisting of the diagonal entries of the matrix \( A \) and have the same block partition as that of \( A \). If \( A \) is an M-matrix, then \( D \) is a positive diagonal matrix and \( D^{-1}(D-A) \) is non-negative and \( \rho(D^{-1}(D-A)) < 1 \) (see Theorem 3.10 in [Var62, p.84]). Since \( \overline{A} \) is a principal submatrix of \( A \), \( \overline{A} \) is an M-matrix, too. Similarly we have that \( \rho((D^{-1}(\overline{D}-\overline{A})) < 1 \), where \( \overline{D}-\overline{A} = \overline{D} - \overline{A} \). If we let \( M = D, \ N = D - \overline{A} \) in (3.10), then \( \tilde{M} = D, \ \tilde{N} = D - \overline{A} \) and so \( \rho((\tilde{D})^{-1}\tilde{D}-\tilde{A}) < 1 \) by Theorem 3.3. Since \( D \) is diagonally positive, so is \( D \). Since \( D-A \) is non-negative, so is \( D-A \). Note that \( D-A = D-A \). Thus \( \tilde{A} \) is an M-matrix. Let \( \tilde{A} = \tilde{M} - \tilde{N} \) with \( \tilde{M} = \text{diag}(S_1, S_2, \cdots, S_k) \) be the Schwarz Splitting. Since \( A \) is an M-matrix, \( \tilde{N} \) is non-negative. Since \( S_i, i = 1, 2, \cdots, k, \) are M-matrices by Lemma 3.3, we have that the \( S_i^{-1}, i = 1, 2, \cdots, k, \) are non-negative, and hence \( (\tilde{M})^{-1} = \text{diag}(S_1^{-1}, S_2^{-1}, \cdots, S_k^{-1}) \) is non-negative, too. Therefore \( \tilde{A} = \tilde{M} - \tilde{N} \) is a regular splitting. Since any regular splitting of an M-matrix is a convergent splitting (see Theorem 3.13 in [Var62, p84]), the conclusion follows. ■

In this section we have presented the basic theoretical analysis mainly for the real symmetric positive definite case. Before our theory was developed, Ribbens [Rib90] reported that preliminary experiments carried out by him had produced numerical result that seemed to be worth being investigated further. We hope that we will be able to come up with more general conclusions (theoretical and experimental) as soon as we are able to verify experimentally our present theoretical findings.
Chapter 4
Parallel PDE Discretization Techniques for MIMD Architectures

In this chapter we present four parallel discretization procedures for two instances (general and self-adjoint) of the second order elliptic PDE equation

\[ Lu = Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + Fu = G \]  

(4.1)

defined on a two-dimensional domain \( \Omega \subset \mathbb{R}^2 \) and subject to boundary conditions

\[ \alpha u_x + \beta u_y + \gamma u = \delta \quad \text{on} \quad \partial \Omega \equiv \text{boundary of} \quad \Omega. \]  

(4.2)

The parallel discretization techniques considered in this study are based on the so-called non-overlapping domain decomposition approach. They are defined on a predetermined partitioning of a grid or mesh [CHH+91]. Three of these procedures are based on the ELLPACK implementation of 5-point finite difference method [RB85]. The fourth procedure considered is a parallel implementation of the Ritz-Galerkin method in the approximation space of piecewise linear polynomials defined on triangles [AB84], [OBC81], [Hug87], [Joh87] utilizing an partitioning of the underlying triangular mesh.

This chapter is organized as follows. In Section 4.1 we describe the domain decomposition methodology employed and the parallel discretization procedures developed. Sections 4.2 and 4.3 discuss the implementations of these discretization procedures on distributed memory machines and specify the I/O interfaces of the corresponding software modules.

4.1 Domain Decomposition Based Parallel Discretization Procedures

The parallel discretization methodology considered in this thesis for the approximation of the elliptic PDE problem (4.1), (4.2) is defined in terms of the following five steps:

1. **Step 1:** Partitioning the grid or mesh \( \Omega_h \) into \( K \) equally sized subdomains \( \{ \Omega_h^{(k)} \}_{k=1}^K \) whose interface lengths are nearly minimum,
Step 2: Allocate the subdomains \( \{ \Omega_h^{(k)} \}_{k=1}^K \) to the processor set \( \{ P_k \}_{k=0}^{K-1} \) so that the cost of the required communication of data among processors is nearly minimum.

Step 3: Determine the discrete equations in each subdomain and store them in local data structures.

Step 4: Determine the unknowns associated with the coupling of equations across interfaces (shared data) and store them in appropriate local data structures.

Step 5: Sort the shared data with respect to a global numbering of the associated geometric data.

For the implementation of these five steps, we have assumed that there is a one to one correspondence between the geometric data (i.e., grid or mesh points) and the unknowns of the corresponding discrete equations.

The performance of the computation on a parallel machine depends on the quality of the partitioning of the corresponding geometric data structures. It has been shown [CHH+91] that parallel finite element codes based on a mesh partitioning consisting of a set of convex balanced subdomains (i.e., the subdomains are almost of equal size) with minimum interface length achieve close to optimal performance on various distributed machines. An example of such partitioning is depicted later on in Figure 4.6. The determination of a partitioning of a mesh (or a grid) into submeshes (subgrids) with equal size and minimum interface length is in general an NP-complete problem. Thus, several heuristics have been proposed for this problem [CHH+91], [Byu94]. Throughout we assume that this partitioning is predetermined. Furthermore, our implementation of the four discretization procedures utilizes the domain decomposition data structures adopted in Parallel(//) ELLPACK [HR92] for distributed memory machines. We describe the data structures used for 2-D grid and finite element mesh decompositions later on in Tables 4.1 and 4.2. Note that Table 4.1 consists of three parts. The first two parts are needed for the finite difference scheme with rectangular domains. The last two parts are needed for the finite difference scheme with general two-dimensional domains.

For the realization of step 2, we currently adopt the simple \( \text{SHIFT} \) mapping scheme (i.e., subdomain \( k \) is mapped to processor \( (k-1) \)). It has been shown in [CHH+91] that this simple mapping is equally effective as some more sophisticated schemes, at least for two-dimensional problems. However, any mapping scheme can be easily implemented with the four discretization procedures.

In all four discretization procedures the generation of the discretized algebraic equations is done in parallel by following the local numbering of the grid or mesh data in each subdomain. Thus, the associated algebraic data (i.e., equations and unknowns) are ordered according to the local numbering of the discrete geometric data. Some of the data structures used to save these data are similar to the ones used in sequential ELLPACK. The system of linear equations is represented in a sparse matrix format using three arrays, for \( 1 \leq i \leq \text{neqn} \),
\(\text{coef}(i, 1) = \) coefficient of the \(i\)-th unknown in the \(i\)-th equation.
\(\text{coef}(i, j) = \) non-zero coefficient of the \(i\)-th equation \((j \geq 2)\).
\(\text{idco}(i, 1) = i.\) 
\(\text{idco}(i, j) = \) index of the unknown whose coefficient is \(\text{coef}(i, j) (j \geq 2)\).
\(\text{rhs}(i) = \) right side of the \(i\)-th equation.

This data structures exists for each subdomain, so the number \(\text{neqn}\) of equations varies. Additional ones accommodate the \textit{shared data} (unknowns associated with the coupling of equations between subdomains). The shared data are associated with some layers of inner- and outer-interface nodal points among neighboring subdomains, where an \textit{inner-interface point} of a subdomain \(\Omega_h^{(k)}\), is a nodal point which belongs to the subdomain \(\Omega_h^{(k)}\) and has connection with the nodal points in a subdomain neighboring the subdomain \(\Omega_h^{(k)}\) and an \textit{outer-interface point} of a subdomain \(\Omega_h^{(k)}\) is a nodal point which belongs to a subdomain neighboring the subdomain \(\Omega_h^{(k)}\) and has a connection with the nodal points in the subdomain \(\Omega_h^{(k)}\). Tables 4.5 and 4.6 discussed later on define all the output data structures used by the parallel discretization modules.

There are two kinds of indexing used for the nodal points: global numbering of the nodal points of the entire domain and the local numbering of the nodal points that belong to each subdomain. The local numbering of the nodal points is the same with the required numbering of the equations. In our implementation, the communication of the \textit{shared data} among subdomains requires their sorting with respect to the global numbering of the associated geometric data. Note that points in a neighboring subdomain that are part of the discrete equations of a subdomain are locally indexed in both subdomains (with different local indices). This requires the utilization of the \textit{local to global mapping} of these data. Moreover, this mapping is also needed for the postprocessing of the computed solution to handle the absence of parallel reading and writing facilities in the hardware. In the case of a tensor product grid decomposition, this mapping consists of the indices corresponding to lower and upper grid lines of each subdomain in the \(x\) (\(y\)) direction, these are stored in the arrays \(i1bgx(1)\) and \(i1bgx(2)\) \((i1bg(1)\) and \(i1bg(2)\)), respectively. For decompositions of general grids and finite element meshes, this mapping is stored in array \(\text{ignod}\), with \(\text{ignod}(j)\) being the global index of the \(j\)-th local nodal point.
Table 4.1 The list of Parallel(/) ELLPACK domain decomposition data structures for grid in each subdomain $\Omega_h^{(k)}$. Table consists of three parts. The first two parts are needed for the finite difference scheme with rectangular domains. The last two parts are needed for the finite difference scheme with general two-dimensional domains.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ndomx(y)</td>
<td>1</td>
<td>Number of rectangular subdomains in $x(y)$-direction</td>
</tr>
<tr>
<td>idomx(y)</td>
<td>1</td>
<td>Position of the rectangular subdomain $\Omega_h^{(k)}$ in the $x(y)$-direction</td>
</tr>
<tr>
<td>il1grx</td>
<td>1</td>
<td>Number of grid lines in the rectangular subdomain $\Omega_h^{(k)}$ in the $x$-direction</td>
</tr>
<tr>
<td>il1gry</td>
<td>1</td>
<td>Number of grid lines in the rectangular subdomain $\Omega_h^{(k)}$ in the $y$-direction</td>
</tr>
<tr>
<td>ilbgrx</td>
<td>2</td>
<td>Global index of the lower and upper bound grid line in the rectangular subdomain $\Omega_h^{(k)}$ in the $x$-direction</td>
</tr>
<tr>
<td>ilbgry</td>
<td>2</td>
<td>Global index of the lower and upper bound grid line in the rectangular subdomain $\Omega_h^{(k)}$ in the $y$-direction</td>
</tr>
<tr>
<td>nnbd</td>
<td>1</td>
<td>Number of the subdomains neighboring $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>nbd</td>
<td>nnbd</td>
<td>Indices of the subdomains neighboring $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>ilnump</td>
<td>1</td>
<td>Number of grid points plus boundary points in the subdomain $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>ignod</td>
<td>ilnump</td>
<td>Global indices of grid points plus boundary points in the subdomain $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>nsnd</td>
<td>isnd(nnbd+1)−1</td>
<td>((\text{nsnd}(j), j = \text{isnd}(i), \text{isnd}(i + 1)−1): geometric global indices of inner-interface points which are interfaced with the neighboring subdomain $\Omega_h^{(\text{nnbd}(i))}$</td>
</tr>
<tr>
<td>isnd</td>
<td>nnbd+1</td>
<td>See the description for the array nsnd</td>
</tr>
<tr>
<td>nrec</td>
<td>irec(nnbd+1)−1</td>
<td>((\text{nrec}(j), j = \text{irec}(i), \text{irec}(i + 1)−1): geometric global indices of outer-interface points which belong to the neighboring subdomain $\Omega_h^{(\text{nnbd}(i))}$</td>
</tr>
<tr>
<td>irec</td>
<td>nnbd+1</td>
<td>See the description for the array nrec</td>
</tr>
</tbody>
</table>
Table 4.2 The list of ELLPACK domain decomposition data structures for triangular finite element mesh in each subdomain $\Omega_h^{(k)}$. These data structures are not restricted to triangular elements.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnbd</td>
<td>1</td>
<td>Number of the subdomains neighboring $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>nbd</td>
<td>nnbd</td>
<td>Indices of the subdomains neighboring $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>nsnd</td>
<td>isnd(nnbd+1)-1</td>
<td>(nsnd(j), j = isnd(i), isnd(i+1)-1) : geometric local indices of inner-interface mesh points which are interfaced with the neighboring subdomain $\Omega_h^{(\text{numbd}(i))}$</td>
</tr>
<tr>
<td>isnd</td>
<td>nnbd+1</td>
<td>See the description for the array nsnd</td>
</tr>
<tr>
<td>nrec</td>
<td>irec(nnbd+1)-1</td>
<td>(nrec(j), j = irec(i), irec(i+1)-1) : geometric local indices of outer-interface mesh points which belong to the neighboring subdomain $\Omega_h^{(\text{numbd}(i))}$</td>
</tr>
<tr>
<td>irec</td>
<td>nnbd+1</td>
<td>See the description for the array nrec</td>
</tr>
<tr>
<td>numnp</td>
<td>1</td>
<td>Number of mesh points in the subdomain $\Omega_h^{(k)}$ including outer-interface mesh points</td>
</tr>
<tr>
<td>numel</td>
<td>1</td>
<td>Number of elements in the subdomain $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>ignod</td>
<td>numnp</td>
<td>ignod(i) is the geometric global index of the mesh point whose local geometric index is i</td>
</tr>
<tr>
<td>x</td>
<td>numnp</td>
<td>(x(j,i), j=1,2) are the x,y coordinates of the mesh point whose local geometric index is i</td>
</tr>
<tr>
<td>ien</td>
<td>3x numel</td>
<td>(ien(j,i), j=1,3) are the three vertices in i-th (local index) element</td>
</tr>
<tr>
<td>ibnod</td>
<td>numnp</td>
<td>ibnod(i) is i-piece if i-th (local index) node is on boundary and zero otherwise</td>
</tr>
<tr>
<td>nconn</td>
<td>iconn(numnp+1)-1</td>
<td>(nconn(j), j = iconn(i), iconn(i+1)-1) are the (local) indices of the adjacent elements of i-th (local index) mesh point</td>
</tr>
<tr>
<td>iconn</td>
<td>numnp+1</td>
<td>See the description for the array nconn</td>
</tr>
</tbody>
</table>
4.1.1 Parallel 5-point Star

This procedure is a generalization of the sequential 5-point star procedure implemented in the sequential ELLPACK library [RB85]. Both the sequential and parallel codes have almost the same structure and are implemented in two modules. One (q35pmn) is for rectangular domains and the other (q35gmnu) is for general two-dimensional domains. The first parallel module is based on the tensor product partitioning of the global grid on a rectangular domain. In this case each grid point is coupled to maximum of four adjacent grid points and each subdomain has maximum of four (west, east, south, north) neighboring subdomains. Thus, the shared data are associated with one layer of inner- and outer-interface points among neighboring subdomains. The second module implements the 5-point star difference discretization on a partitioning of a general domain. Figure 4.1 depicts a general partitioning of the grid and boundary points. Figure 4.2 displays the column indices idc of the submatrix corresponding to subdomain 4 of the partitioning depicted in Figure 4.1. In this case each interior grid point is coupled to maximum of four adjacent points while the active boundary points (non-Dirichlet boundary conditions) are coupled to maximum of seven grid or boundary points. Hence, the communicated data are associated with two layers of inner- and outer-interface points of neighboring subdomains.

Figure 4.1 A partitioning of a rectangular grid assuming Dirichlet condition on the entire boundary of a general domain. The degrees of freedom corresponding to Dirichlet boundary conditions are not active. The local and global numbering of the grid points in each subdomain is indicated by different fonts specified on the right side of the figure. For subdomain 4, the inner-interface points are 1, 2, 3, 4, 5, 9, 10 and the outer-interface points are 12, 13, 14, 15, 16, 17, 18, 19.
4.1.2 Parallel Encapsulated 5-point Star

For the optimal efficiency, the parallel general 5-point star module needs an optimal partitioning of the grid used. However, it has been observed that the cost of even nearly optimal partitioning can be very substantial for large grids [Chr92]. In order to reduce this cost, we have considered extending the discrete PDE problem to the rectangular domain that contains the original PDE domain. Then the grid of the rectangular domain encapsulates a grid on the original domain. Specifically, we assign identity equations at the exterior grid points. These artificial equations are uncoupled from the active ones. This approach is similar to the one devised by Lynch [Lyn89] to implement the ITPACK routines on vector machines. Its advantage is that the partitioning of the larger grid is rather trivial. This third module for 5-point star difference discretization is called encapsulated 5-point star. For the implementation of this module, all three parts of data structures in Table 4.1 are needed as input data.

Figures 4.3, 4.4, 4.5 show some partitionings of encapsulated two-dimensional grids. Tables 4.3 and 4.4 compare the encapsulated 5-point star approach with the general parallel 5-point star module. The timings(seconds) are measured on the nCUBE II machine and the speedup is the ratio of the computing time using a finitely many processors to the computing time using only one processor assuming that a same grid/mesh is used for both cases. The data in Table 4.3 indicate that the best partitioning for the encapsulated 5-point parallel solution is the $p \times p$ (tensor product decomposition with $p$ subdomains in x/y-direction, respectively [CHH+91]).
with respect to its overhead and the timing of the corresponding PDE solution. From the data in Table 4.3a we conclude that the time for the encapsulated solution is about 25% higher than the parallel general 5-point solution. This is expected since the encapsulation approach utilizes 25% fewer processors. However, if we include the decomposition time then the encapsulation solution is more efficient by a factor 2 to 18 compared to optimal and MRSB (multilevel recursive spectral bisection algorithm [Wu93]) general parallel 5-point star solutions. Note that Sun4-workstation used for the decomposition is even faster than one processor of the nCUBE II. After a similar analysis of the data in Table 4.4, we conclude that the encapsulation technique coupled with \( p \times p \) decomposition produces a more efficient solution. The encapsulated 5-point star procedure has been implemented within the //ELLPACK module q35gma.

Figure 4.3 A 4-way partitioning of an encapsulated rectangular grid assuming **Dirichlet** boundary conditions on part of the boundary and **mixed** boundary conditions on the remaining part. The degrees of freedom corresponding to Dirichlet boundary conditions are not active. The local and global numbering of the grid points in each subdomain is indicated by different fonts specified on the right side of the figure. For subdomain 4, the inner-interface points are 1, 2, 3, 4, 5, 6, 7, 8, 10, 11 and the outer-interface points are 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23.
Figure 4.4 A 4-way partitioning of an encapsulated rectangular grid assuming Dirichlet condition on the entire boundary. The degrees of freedom corresponding to Dirichlet boundary conditions are not active. The local and global numbering of the grid points in each subdomain is indicated by different fonts specified on the right side of the figure. For subdomain 4, the inner-interface points are 1, 2, 3, 4, 5 and the outer-interface points are 7, 8, 9, 10, 11, 12.

Figure 4.5 A 4-way partitioning of an encapsulated rectangular grid with the red-black indexing of the grid points. The degrees of freedom corresponding to Dirichlet boundary conditions are not active. The local and global numbering of the grid points in each subdomain is indicated by the different fonts specified on the right side of the figure.
Table 4.3 The discretization and solution timings ($t$) in second and speedups ($s$), the cost of the partitioning heuristics used, and the ratios of time ($\rho_t$) and speedup ($\rho_s$) of the encapsulated (and MRSB) to the general (optimal) schemes. The data listed are for the PDE problem $u_{xx} + u_{yy} = f$ defined on an L-shaped domain. The function $f$ is chosen so that the true solution is $u = (x^\alpha - x)(y^\alpha - y)/(\alpha^{\alpha/(1-\alpha)} - \alpha^{1/(1-\alpha)})^2$ with $\alpha = 1.5$. A 150 x 150 grid on the encapsulated L-shaped domain is used and single precision arithmetic. The Jacobi-CG solver is terminated when the convergence test is taken to be $5.96 \times 10^{-5}$ (see [RB85]). The L-shaped domain is selected so that the partitioning of the grid is optimal for the general parallel 5-point star module for the machine configuration used.

<table>
<thead>
<tr>
<th>Processors</th>
<th>Encapsulated 5-point star</th>
<th>General 5-point star</th>
<th>MRSB [Wu93]</th>
<th>Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p \times p$</td>
<td>$1 \times p$</td>
<td>$p \times 1$</td>
<td>$t$</td>
</tr>
<tr>
<td>1</td>
<td>351.97</td>
<td>1.00</td>
<td>351.97</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>129.98</td>
<td>2.71</td>
<td>126.46</td>
<td>2.78</td>
</tr>
<tr>
<td>16</td>
<td>35.30</td>
<td>9.70</td>
<td>37.39</td>
<td>9.41</td>
</tr>
<tr>
<td>64</td>
<td>11.26</td>
<td>31.27</td>
<td>13.62</td>
<td>25.83</td>
</tr>
</tbody>
</table>

(a) Timing (sec) and speedup of the two parallel finite difference discretization modules for different partitionings.

<table>
<thead>
<tr>
<th>Processors</th>
<th>Encapsulated 5-point star</th>
<th>General 5-point star</th>
<th>MRSB [Wu93]</th>
<th>Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p \times p$</td>
<td>$1 \times p$</td>
<td>$p \times 1$</td>
<td>$t$</td>
</tr>
<tr>
<td>1</td>
<td>2.67</td>
<td>2.67</td>
<td>2.67</td>
<td>2.67</td>
</tr>
<tr>
<td>4</td>
<td>2.20</td>
<td>3.20</td>
<td>3.11</td>
<td>2.50</td>
</tr>
<tr>
<td>16</td>
<td>1.51</td>
<td>2.93</td>
<td>2.84</td>
<td>2.84</td>
</tr>
<tr>
<td>64</td>
<td>0.76</td>
<td>2.86</td>
<td>2.77</td>
<td>2.77</td>
</tr>
</tbody>
</table>

(b) The cost of the partitions used. Time (sec) is measured on a Sun4-workstation.

<table>
<thead>
<tr>
<th>Processors</th>
<th>4</th>
<th>16</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partitioning</td>
<td>$\rho_t$</td>
<td>$\rho_s$</td>
<td>$\rho_t$</td>
</tr>
<tr>
<td>$p \times p$</td>
<td>1.46</td>
<td>0.69</td>
<td>1.46</td>
</tr>
<tr>
<td>$1 \times p$</td>
<td>1.42</td>
<td>0.70</td>
<td>1.50</td>
</tr>
<tr>
<td>$p \times 1$</td>
<td>1.50</td>
<td>0.67</td>
<td>1.50</td>
</tr>
<tr>
<td>MRSB [Wu93]</td>
<td>1.00</td>
<td>1.00</td>
<td>1.01</td>
</tr>
</tbody>
</table>

(c) Ratios of encapsulated to general (optimal) performance in solving the linear system of equations.
Table 4.4 The discretization and solution timings\( (t) \) in second and speedups\( (s) \), the cost of the partitioning heuristics used, and the ratios of time\( (\rho_t) \) and speedup\( (\rho_s) \) of the general to the encapsulated schemes. The data listed are for the PDE problem \( u_{xx} + u_{yy} = f \) defined on an L-shaped domain. The function \( f \) is chosen so that the true solution is \( u = (x^a - x)(y^a - y)/(\alpha^{a/(1-a)} - \alpha^{1/(1-a)})^2 \) with \( \alpha = 1.5 \). A 150 \times 150 grid on the encapsulated L-shaped domain is used and single precision arithmetic. The Jacobi-CG solver is terminated when the convergence test is taken to be \( 5.96 \times 10^{-5} \) (see [RB85]). The L-shaped domain is selected so that the partitioning of the grid is optimal for the encapsulated parallel 5-point star module for the machine configuration used.

<table>
<thead>
<tr>
<th>Processors</th>
<th>MRSB [Wu93]</th>
<th>( p \times p )</th>
<th>( 1 \times p )</th>
<th>Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( t )</td>
<td>( s )</td>
<td>( t )</td>
<td>( s )</td>
</tr>
<tr>
<td>1</td>
<td>606.58</td>
<td>1.00</td>
<td>606.58</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>157.83</td>
<td>3.84</td>
<td>156.17</td>
<td>3.88</td>
</tr>
<tr>
<td>16</td>
<td>43.59</td>
<td>13.92</td>
<td>44.06</td>
<td>13.77</td>
</tr>
<tr>
<td>64</td>
<td>15.12</td>
<td>40.12</td>
<td>14.89</td>
<td>40.74</td>
</tr>
</tbody>
</table>

(a) Timing\( (\text{sec}) \) and speedups of the two parallel finite difference discretization modules for different partitionings.

<table>
<thead>
<tr>
<th>Processors</th>
<th>MRSB [Wu93]</th>
<th>( p \times p )</th>
<th>( 1 \times p )</th>
<th>Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>22.87</td>
<td>10.40</td>
<td>10.52</td>
<td>3.29</td>
</tr>
<tr>
<td>4</td>
<td>187.72</td>
<td>24.43</td>
<td>12.51</td>
<td>2.11</td>
</tr>
<tr>
<td>16</td>
<td>251.03</td>
<td>20.19</td>
<td>18.79</td>
<td>1.44</td>
</tr>
<tr>
<td>64</td>
<td>299.72</td>
<td>27.23</td>
<td>31.99</td>
<td>0.72</td>
</tr>
</tbody>
</table>

(b) The cost of the partitions used. Time\( (\text{sec}) \) is measured on a Sun4-workstation.

<table>
<thead>
<tr>
<th>Processors</th>
<th>( \rho_t )</th>
<th>( \rho_s )</th>
<th>( \rho_t )</th>
<th>( \rho_s )</th>
<th>( \rho_t )</th>
<th>( \rho_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MRSB [Wu93]</td>
<td>0.77</td>
<td>1.29</td>
<td>0.76</td>
<td>1.31</td>
<td>0.86</td>
<td>1.16</td>
</tr>
<tr>
<td>( p \times p )</td>
<td>0.76</td>
<td>1.31</td>
<td>0.77</td>
<td>1.29</td>
<td>0.85</td>
<td>1.18</td>
</tr>
<tr>
<td>( 1 \times p )</td>
<td>0.78</td>
<td>1.28</td>
<td>0.76</td>
<td>1.32</td>
<td>0.85</td>
<td>1.18</td>
</tr>
</tbody>
</table>

(c) Ratios of general to encapsulated performance in solving the linear system of equations.
4.1.3 Parallel Linear Triangular Finite Element Method

In this section we describe the parallel implementation of the Ritz-Galerkin method based on linear triangular piecewise polynomials on an partitioning of the finite element mesh. Table 4.2 defines the //ELLPACK domain decomposition data structures for the finite element method (FEM) meshes used in the implementation of this discretization procedure. This module is applicable to second order self-adjoint elliptic PDEs

\[ Lu = -(p u_x)_x - (p u_y)_y + q u = g \]  \hspace{1cm} (4.4)

defined on a domain \( \Omega \subset \mathbb{R}^2 \) and subject to boundary conditions

\[ u = \gamma \quad \text{on} \quad \partial \Omega_1, \]
\[ \frac{\partial u}{\partial n} + \sigma u = \xi \quad \text{on} \quad \partial \Omega_2 = \partial \Omega - \partial \Omega_1 \]  \hspace{1cm} (4.5)

where \( \partial \Omega \) denotes the boundary of \( \Omega \) and \( \partial \Omega_1 \) a piece of it. The domain \( \Omega \) can have holes while its boundary \( \partial \Omega \) is assumed to be piecewise smooth. The FEM discretization of the PDE problem (4.4), (4.5) is defined on its variational formulation.

In Figure 4.7, there is a pseudo-code to determine the FEM coefficient matrix (element stiffness matrix) corresponding to the element \( E_i, i = 1, 2, \ldots, L \). The global stiffness matrix \( (A_{ij})_{i,j=1}^N \) and the right hand side \( (b_i)_{i=1}^N \) of the matrix equation are obtained by assembling the element matrices and element right hand sides using the global numbering of the mesh data (see [AB84] for details).

For the description of the parallel FEM module we introduce the notation \( \{ \Omega_h^{(k)} \}_{k=1}^K \) for the subdomains of a partitioning of the FEM mesh. An example of such a partitioning is given in Figure 4.6. Throughout, we assume that the interfaces between neighboring subdomains consists of one layer of elements. Furthermore, we use the following notation:

\[ \partial \Omega_h^{(k)} \equiv \text{boundary of} \ \Omega_h^{(k)}, \]
\[ \partial \Omega_h^{(k)}_{\Omega_1} \equiv (\Omega_h^{(k)} \cup \partial \Omega_h^{(k)}) \cap \partial \Omega_1, \]
\[ \partial \Omega_h^{(k)}_{\Omega_2} \equiv (\Omega_h^{(k)} \cup \partial \Omega_h^{(k)}) \cap \partial \Omega_2, \]
\[ \partial \Omega_h^{(k)}_{\Omega_2} \equiv \partial \Omega_h^{(k)} - (\partial \Omega_h^{(k)}_{\Omega_1} \cup \partial \Omega_h^{(k)}_{\Omega_2}). \]

If the \( E_i^{(k)}, i = 1, 2, \ldots, L^{(k)} \), are the elements of \( \Omega_h^{(k)} \cup \partial \Omega_h^{(k)} \), the \( \{ \alpha_i^{(k)} \}_{i=1}^M^{(k)} \) are the nodal points of FEM mesh, the \( \{ \phi_i^{(k)} \}_{i=1}^m^{(k)} \) are the FEM basis functions, \( A^{(k)} \) and \( b^{(k)} \) are the stiffness matrix and right hand side of the matrix equation associated with the subdomain \( \Omega_h^{(k)} \), respectively, then the element stiffness matrix in subdomain \( \Omega_h^{(k)} \) is defined by the pseudo-code listed in Figure 4.8. The parallel FEM module consists of code for generating the element matrices in each subdomain and for assembling them into a block of the global FEM equations. The generation of the global equations requires the use of local data from subdomain's outer-interface with its neighbors.
Figure 4.6 A 4-way partitioning of a triangular finite element mesh. The degrees of freedom corresponding to Dirichlet boundary conditions are not active. The local and global numbering of the grid points in each subdomain is indicated by different fonts specified on the right side of the figure. For subdomain 4, the inner-interface points are 47, 40, 32, 33, 25, 26 (global indices) and the outer-interface points are 46, 39, 31, 22, 23, 24, 16, 17 (global indices).
procedure plcOfem

bilinear functional:

\[
a(u,v)_l = \int_{E_l} \left( p_{u_x}v_x + p_{u_y}v_y + q_{uv} \right) dx dy + \int_{\partial \Omega \cap E_l} (p \xi u v) ds,
\]

\[
G(v)_l = \int_{E_l} (gv) dx dy + \int_{\partial \Omega \cap E_l} (p \xi v) ds.
\]

\( E_l \) is the \( l \)-th triangular element of the FEM mesh \( \Omega_h \).
\( \alpha_i \) is the nodal points of \( \Omega_h \) such that

\[
\{ \alpha_i \}_{i=1}^N \subset \Omega \cup \partial \Omega_2 \quad \text{and} \quad \{ \alpha_i \}_{i=N+1}^M \subset \partial \Omega_1.
\]

\( \{ \phi_i \}_{i=1}^M \) is the set of piecewise linear basis functions defined on \( \partial \Omega_h \).

loop \( l = 1 \) until \( L \) do

loop \( m = 1 \) until 3 do

\( \alpha_i := m \)-th vertex in \( l \)-th element

if \( 1 \leq i \leq N \) then

\( b_i := b_i + G(\phi_i)_l \)

loop \( n = 1 \) until 3 do

\( \alpha_j := n \)-th vertex in \( l \)-th element

if \( 1 \leq j \leq N \) then

\( A_{ij} := A_{ij} + a(\phi_j, \phi_i)_l \)

elseif \( N + 1 \leq j \leq M \) then

\( b_i := b_i - \gamma(\alpha_j) a(\phi_j, \phi_i)_l \)

endif

endloop

endif

endloop

endloop

end plcOfem

Figure 4.7 The pseudo-code for generating the stiffness matrix.
procedure parallel pic0fem

C for the k-th subdomain \( \Omega_k^{(k)} \).

C \{ \alpha_i^{(k)} \}_{i=1}^{N_0^{(k)}} \subset \Omega_k^{(k)} \cup \partial \Omega_{h2}^{(k)}, \quad \{ \alpha_i^{(k)} \}_{i=N_0^{(k)}+1}^{N^{(k)}} \subset \partial \Omega_{h2}^{(k)}, \quad \{ \alpha_i^{(k)} \}_{i=N^{(k)}+1}^{M^{(k)}} \subset \partial \Omega_{h1}^{(k)}.

C \quad a(u, v)^{(k)} = \int_{\Omega_k^{(k)}} (pu_u v_x + pu_v v_y + quv) \, dx dy + \int_{\partial \Omega_{h2}^{(k)} \cap \Omega_k^{(k)}} (p\sigma uv) \, ds,

C \quad G(v)^{(k)} = \int_{\Omega_k^{(k)}} (gv) \, dx dy + \int_{\partial \Omega_{h2}^{(k)} \cap \Omega_k^{(k)}} (p\xi v) \, ds.

C

loop \( l = 1 \) until \( L^{(k)} \) do

C loop \( m = 1 \) until 3 do

C \quad \alpha_i^{(k)} := m-th vertex in \( l \)-th element in subdomain \( \Omega^{(k)} \)

C \quad if (1 ≤ i ≤ N_0^{(k)}) then

C \quad \quad b_i^{(k)} := b_i^{(k)} + G(\phi_i^{(k)})^{(k)}

C \quad loop \( n = 1 \) until 3 do

C \quad \quad \alpha_j^{(k)} := n-th vertex in \( l \)-th element in subdomain \( \Omega^{(k)} \)

C \quad \quad if (1 ≤ j ≤ N^{(k)}) then

C \quad \quad \quad A_{ij}^{(k)} := A_{ij}^{(k)} + a(\phi_j^{(k)}, \phi_i^{(k)})^{(k)}

C \quad \quad elseif (N^{(k)} + 1 ≤ j ≤ M^{(k)}) then

C \quad \quad \quad b_i^{(k)} := b_i^{(k)} - \gamma(\alpha_j^{(k)}) a(\phi_j^{(k)}, \phi_i^{(k)})^{(k)}

C \quad endif

C \quad endloop

C \quad endif

C \quad endloop

C \quad endloop

C \quad end parallel pic0fem

Figure 4.8 The pseudo-code for generating the local stiffness matrix associated with the subdomain \( \Omega_k^{(k)} \).
4.2 Implementation of Parallel Finite Difference Discretization Modules

In this section we describe the additional routines utilized for the parallel implementation of the finite difference procedures considered in Section 4.1. The domain decomposition data needed for these modules are provided by the subroutine q2dcps. In the case of the $p \times q$ tensor product partitioning of the grid for rectangular or encapsulated-general domains, the subroutine q2dcps generates the data listed in Table 4.1. In the case of general domains, the subroutine q2dcps reads the predetermined domain decomposition data from a file.

The parallel code for generating the local difference equations is the same as the sequential code. In addition, the parallel code must handle the communication information related to the data shared among the subdomains. Specifically, these are the local indices of the grid points (unknowns) associated with the shared (coupled) unknowns. The routines q35pco (for q35pmn) and q35gco (for q35gmn) generate this information. Examples of the output of these routines depicted in Figures 4.4 and 4.5 are shown in Figures 4.9 and 4.10 for two different orderings of the equations and unknowns. Since the subroutine q35gmn also realizes the encapsulated 5-point star module, it has additional code to handle the generation of fictitious equations at the external points due to the encapsulation. The I/O interfaces of these routines are defined by Tables 4.1 and 4.5. Table 4.1 consists of three parts. The first two parts are input data for q35pmn. The last two parts are used in q35gmn. If the tensor product partitioning is used with q35gmn, then all three parts are used.

![Figure 4.9](image-url) The data structures nsnd, isnd, nrec, irec for subdomain 4 of the mesh partitioning depicted in Figure 4.4 assuming a natural ordering.
4.3 Implementation of Parallel Linear FEM Procedure

Throughout we refer to this module as \texttt{p1c0fem}. Its implementation receives as input a partitioning of the underlying finite element mesh. Its implementation assumes that the elements between the nodal interfaces of the two (or more) subdomains consist of one layer. Tables 4.2 and 4.6 describe the I/O interfaces for this module.

The local numbering employed for the nodal points (and equations) numbers all the nodal points of the subdomain first and the outer-interface points last. In the case of Dirichlet boundary conditions, the equations at the corresponding nodal points are not explicitly generated and the boundary unknowns are eliminated during the discretization process. In order to match the local information between subdomains and to support the postprocessing of the solution, the local-to-global mapping of the geometric/solution data is used. This mapping of geometric data is supplied by the mesh partitioning routine and is stored in the array \texttt{ignod}. In the case of algebraic data (i.e., equations and unknowns) this mapping is generated by \texttt{p1c0fem} and is stored in the array \texttt{id}. Its inverse is stored in the array \texttt{iid}. The indices of the neighboring subdomains to each subdomain are generated by the domain decomposition tool and stored in the array \texttt{nbd}. The arrays \texttt{nsnd} and \texttt{nrec} contain the local indices of the inner-interface and outer-interface mesh points, respectively, which are provided by mesh partitioning routine. Then \texttt{p1c0fem} module reorganize the information in the arrays \texttt{nsnd} and \texttt{nrec}. It is important to notice that the indices in \texttt{nsnd} and \texttt{nrec} are grouped so that the indices associated with a neighboring subdomain \texttt{nbd(i)} belong to the same subset, and then the indices in each subset are sorted according the local-to-global mapping \texttt{ignod}. As we have already pointed out, in our implementation, this sorting is needed to guarantee the correct association of the communicating local data. So, the outputs are similar with those of the 5-point star modules depicted in Figures 4.4. The structure of \texttt{p1c0fem} module is depicted in Figure 4.11 and the definitions of the subroutines involved are given in Table 4.7.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.10.png}
\caption{The data structures \texttt{nsnd}, \texttt{isnd}, \texttt{nrec}, \texttt{irec} for subdomain 4 of the mesh partitioning depicted in Figure 4.5 assuming red-black ordering.}
\end{figure}
Remark 4.1 The pic0fem module assumes that the end points of a piecewise representation of \( \partial \Omega \) are vertices of some elements in \( \Omega \). Furthermore, these points are considered as points of \( \partial \Omega \) rather than \( \partial \Omega_2 \).
Table 4.5 The list of output data structures of all three 5-point star modules (i.e., for both the subroutines q35pmn and q35gmn) and for each subdomain $\Omega_h^{(k)}$.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nnbd</td>
<td>1</td>
<td>Number of the subdomains neighboring $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>nbd</td>
<td>nnbd</td>
<td>Indices of the subdomains neighboring $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>coef</td>
<td>neqn x ncoe</td>
<td>The $\text{coef}(i,j)$ is $(i,\text{idco}(i,j))$-th entry of the coefficient submatrix for the subdomain $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>idco</td>
<td>neqn x ncoe</td>
<td>The $\text{idco}(i,j)$ is the column index of the entry $\text{coef}(i,j)$ in the coefficient submatrix for the subdomain $\Omega_h^{(k)}$, see (4.3)</td>
</tr>
<tr>
<td>rhs</td>
<td>neqn</td>
<td>Right hand side of coefficient submatrix equation for the subdomain $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>ncoe</td>
<td>1</td>
<td>Maximum number of non-zero entries over all the rows of the coefficient submatrix of $\Omega_h^{(k)}$</td>
</tr>
<tr>
<td>neqn</td>
<td>1</td>
<td>Sum of non-interface active grid points and number of inner-interface active grid points, number of equations generated, row dimension of the submatrix</td>
</tr>
<tr>
<td>neqa</td>
<td>1</td>
<td>Sum of $\text{neqn}$ and number of outer-interface active grid points, column dimension of the submatrix</td>
</tr>
<tr>
<td>nsnd</td>
<td>isnd(nnbd+1)−1</td>
<td>Algebraic local indices of the grid points on which the values $\text{nsnd}(j)$ ($j = \text{isnd}(i),\text{isnd}(i+1)−1$) are sent to $\text{nbd}(i)$-th processor when the matrix equation is solved by a parallel iterative solver</td>
</tr>
<tr>
<td>isnd</td>
<td>nnbd+1</td>
<td>See the description for the array $\text{nsnd}$</td>
</tr>
<tr>
<td>nrec</td>
<td>irec(nnbd+1)−1</td>
<td>Algebraic local indices of the grid points on which the values $\text{nrec}(j)$ ($j = \text{irec}(i),\text{irec}(i+1)−1$) are received from $\text{nbd}(i)$-th processor when the matrix equation is solved by a parallel iterative solver</td>
</tr>
<tr>
<td>irec</td>
<td>nnbd+1</td>
<td>See the description for the array $\text{nrec}$</td>
</tr>
</tbody>
</table>
Table 4.6 The list of output data structures of the tri-linear FEM module (i.e., for the subroutines `plc0fem`) for each subdomain $\Omega_k^{(k)}$.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Dimension</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>tabl</code></td>
<td><code>numnp</code></td>
<td>The <code>tabl(i)</code> has the value of Dirichlet condition if $i$-th (local index) mesh point is on Dirichlet boundary, zero otherwise.</td>
</tr>
<tr>
<td><code>id</code></td>
<td><code>numnp</code></td>
<td>The <code>id(i)</code> is the algebraic index of $i$-th (local index) node. The non-outer-interface active mesh points are numbered first and then numbered outer-interface active mesh points.</td>
</tr>
<tr>
<td><code>iid</code></td>
<td><code>neqn</code></td>
<td>Inverse mapping of <code>id</code>, i.e., <code>iid(i)</code> is the geometric local index of the node whose algebraic local index is $i$.</td>
</tr>
<tr>
<td><code>coef</code></td>
<td><code>neqn x ncoe</code></td>
<td>See Table 4.5.</td>
</tr>
<tr>
<td><code>idco</code></td>
<td><code>neqn x ncoe</code></td>
<td>See Table 4.5.</td>
</tr>
<tr>
<td><code>rhs</code></td>
<td><code>neqn</code></td>
<td>See Table 4.5.</td>
</tr>
<tr>
<td><code>ncoe</code></td>
<td><code>1</code></td>
<td>Number of non-interface active mesh points plus number of inner-interface active mesh points, number of equations generated, row dimension of the submatrix.</td>
</tr>
<tr>
<td><code>neqn</code></td>
<td><code>1</code></td>
<td>Sum of <code>neqn</code> and number of outer-interface active mesh points, column dimension of the submatrix.</td>
</tr>
<tr>
<td><code>neqa</code></td>
<td><code>1</code></td>
<td>Sum of <code>neqn</code> and number of outer-interface active mesh points, column dimension of the submatrix.</td>
</tr>
<tr>
<td><code>nsnd</code></td>
<td><code>isnd(nnbd+1)-1</code></td>
<td>$(nsnd(j), j = isnd(i), isnd(i+1)-1)$ are the algebraic local indices of the mesh points on which the values are sent to <code>nbd(i)</code>-th processor when the matrix equation is solved by a parallel iterative method (nsnd is sorted by the global index <code>ignod</code> of the mesh points).</td>
</tr>
<tr>
<td><code>isnd</code></td>
<td><code>nnbd+1</code></td>
<td>See the description for the array <code>nsnd</code>.</td>
</tr>
<tr>
<td><code>nrec</code></td>
<td><code>irec(nnbd+1)-1</code></td>
<td>$(nrec(j), j = irec(j), irec(i+1)-1)$ are the algebraic local indices of the mesh points on which the values are sent to <code>nbd(i)</code>-th processor when the matrix equation is solved by a parallel iterative solver (nrec is sorted by the global index <code>ignod</code> of the mesh points).</td>
</tr>
<tr>
<td><code>irec</code></td>
<td><code>nnbd+1</code></td>
<td>See the description for the array <code>nrec</code>.</td>
</tr>
</tbody>
</table>
Table 4.7 Description of subroutines in plc0fem module.

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plc0fem</td>
<td>Main driver of plc0fem module</td>
</tr>
<tr>
<td>eqset</td>
<td>Generates local algebraic index id and its inverse mapping iid using ibnod and nrec</td>
</tr>
<tr>
<td>sort</td>
<td>Sorts nsnd and nrec according to the global index ignod, after excluding the non-active mesh points</td>
</tr>
<tr>
<td>compbc</td>
<td>Generates Dirichlet boundary condition values tabl using id and ibnod</td>
</tr>
<tr>
<td>plc0qb</td>
<td>Builds the submatrix of the discrete equations, i.e., coef, idco and rhs</td>
</tr>
<tr>
<td>qdcshl</td>
<td>Calculates integration-rule weights, shape functions and local derivatives for the master element (see [AB84])</td>
</tr>
<tr>
<td>qdcshg</td>
<td>Calculates global derivatives of shape functions and Jacobian determinants for each element</td>
</tr>
<tr>
<td>qdck</td>
<td>Forms stiffness matrix for each element</td>
</tr>
<tr>
<td>qdckb</td>
<td>Forms additional stiffness matrix for mixed boundary condition points</td>
</tr>
</tbody>
</table>
Chapter 5
Parallel Stationary Iterative Methods for Distributed Memory Machines

In this chapter we discuss the implementation of the ITPACK library [KRDG82] in the parallel //ELLPACK environment [HR92] and report on its performance on the nCUBE II parallel machine. In this study, we are concerned with the solution of the algebraic equations obtained from the discretization of second order elliptic PDEs on rectangles and general domains with mixed boundary conditions using finite difference and finite element approximations. The modules in parallel ITPACK (//ITPACK) can be applied to any system of linear equations stored according to the //ELLPACK distributed storage scheme. The parallelization methodology employed is based on a partitioning defined on the discrete geometric data structures (i.e., meshes and grids) associated with the selected PDE solver. The performance results obtained so far indicate almost optimal computational and space efficiency of the //ITPACK modules. This chapter is organized as follows. In Section 5.1, we describe the parallelization methodology applied to implement the //ITPACK library on distributed memory machines and discuss its applicability. Section 5.2 defines the input/output interfaces of this library. In Section 5.3 we discuss the parallelization approach used and indicate the modifications made to the sequential code of the ITPACK modules. Finally, in Section 5.4, data about the computational performance of the //ITPACK modules is presented.

5.1 Description of Parallel ITPACK Library

Throughout we assume that the reader is familiar with the theoretical aspects of the iteration methods implemented in the sequential modules of the ITPACK library. A detailed description of each ITPACK module can be found in [KRDG82] and [RB85]. The parallel ITPACK modules are based on the sequential ones, thus their theoretical behavior has not been altered. In this section, we discuss the parallelization methodology adopted for their implementation within the parallel ELLPACK environment. The //ITPACK library reported on here is a generalization of the one presented in [CHK+92] which is based on $p \times q$ tensor product decomposition of an orthogonal grid and using 5-point finite difference discretization schemes.
Table 5.1 The names of the parallelized ITPACK modules, the orderings scheme assumed, and the iteration method implemented. The abbreviations N and RB stand for natural and red/black ordering, respectively.

<table>
<thead>
<tr>
<th>IITPACK module</th>
<th>Indexing</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOR</td>
<td>RB</td>
<td>Successive Over-Relaxation</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>N/RB</td>
<td>Jacobi conjugate gradient(CG)</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>N/RB</td>
<td>Jacobi with Chebyshev acceleration(SI)</td>
</tr>
<tr>
<td>RSCG</td>
<td>RB</td>
<td>Reduced system CG</td>
</tr>
<tr>
<td>RSSI</td>
<td>RB</td>
<td>Reduced system SI</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>RB</td>
<td>Symmetric SOR CG</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>RB</td>
<td>Symmetric SOR SI</td>
</tr>
</tbody>
</table>

The new version of the IITPACK library assumes that the algebraic equations are partitioned in a row-wise splitting and the corresponding blocks are stored in the local memories of the selected physical processors together with the communication information (i.e., local indices of the shared components of the unknown vector). The interface to the IITPACK library is defined later. The partitioning of the equations is obtained indirectly from a non-overlapping decomposition of the discrete PDE geometric data structures (i.e., grid or mesh) in subdomains. Thus, the current implementation of IITPACK is independent of the PDE domain geometry and the PDE equation discretization. The ELLPACK system provides the appropriate infrastructure to support domain decompositions.

The IITPACK library consists of seven modules listed in Table 5.1 which use the indicated indexing schemes and numerical methods. Next, we discuss in some detail the parallelization of the ITPACK modules starting with the definition of the module interface.

5.2 The Interface of the Parallel ITPACK Library

For the parallelization of the ITPACK modules, the coefficient matrix and right side vector, are assumed to be decomposed into blocks and the coupling information between the blocks (as defined by the set of unknowns shared by the blocks of equations) is provided. The set of unknowns involved in each block and assigned to a processor is assumed to consist of two parts. One consists of the local unknowns (called interior) and the other consists of the shared ones (called exterior). The non-zero coefficients of the block equations are stored in the (coeff, idco) arrays while the array rhs holds the right side. See (4.3) for details. The communication information, i.e., the indices of the vector of shared unknowns, is stored in the data structures (nsm, nrec). These indices are assumed sorted according to a global ordering of the equations and unknowns. This guarantees proper matching of the communication
information. An example of these data structures is provided in Table 4.5 in Chapter 4. The local unknown vector is assumed to contain both the uncoupled unknowns \((\text{unkn}(j), j = 1, \text{neqn})\) and the shared or coupled ones \((\text{unkn}(j), j = \text{neqn} + 1, \text{neqa})\).

### 5.3 On the Parallelization of ITPACK Modules

The parallelization of the ITPACK routines is based primarily on the parallel implementation of level 1 and 2 sparse BLAS involved in these computations. These parallel BLAS are realized on the distributed algebraic data structures like in Table 4.5 in Chapter 4. Specifically, we have parallelized the matrix-vector multiplication and inner vector product operations. The sequential ITPACK modules have been re-implemented using these two BLAS operations together with some memory space allocation adjustments. Figure 5.1 gives the basic tasks involved in a sequential ITPACK module.

Next, we describe the nodal (processor) code in terms of the sequential subtasks. We point out the ones affected and discuss the actual modifications made.

1. Initialize parameters.
2. Check the matrix dimension.
3. Scale the system so that the diagonal of the matrix is 1.
4. Remove rows and columns when the off diagonal elements are very "small".
5. Initialize workspace pointers.
6. Select indexing scheme and determine appropriate information.
7. Permute the system according to the selected indexing scheme information.
8. Check for sufficient workspace and initialize some parameters.
   — start iterative process —
9. Initial setups for the iterative process.
10. Iterate.
11. Check convergence.
   —— post processing of solution ——
12. Put solution in place.
13. Reverse the permutation of the system.
15. Unscale the system.
16. Setup return parameters.

Figure 5.1 The basic subtasks in a sequential ITPACK module.
5.3.1 The Parallelization of the Basic ITPACK Subtasks

The first four subtasks of the nodal program are identical to the sequential ones. Subtask 5 is modified to initialize the local data structures. No modifications are needed for subtask 6. In addition to some changes in the declaration statements of subtask 7, the array nsnd is included in the permutation (see subroutine q5i9mg) defined by the selected indexing scheme. Subtasks 8 and 9 are used unchanged. The most computationally intensive part of an ITPACK routine is subtask 10. If $A$ denotes the local submatrix involved in the iterative procedure, then the matrix-vector multiplication

$$
\sum_{j=1}^{\text{neqa}} A_{ij} u_j \quad \text{for } i = 1, \ldots, \text{neqn},
$$

has to take place. This operation requires that the values $(u(j), j = \text{neqn} + 1, \text{neqa})$ are updated with the data received from the neighboring processors. The subroutine q9mesg(data, msgtype, irbflag) is used to communicate the shared vector components between neighboring processors. This local communication depends on the ordering scheme used. The irbflag variable is to select the appropriate code based on the ordering scheme used. For the natural ordering, the irbflag is zero. In the case of red-black ordering, setting irbflag = 1 implies that local information defined on red points only is exchanged, while for irbflag = -1 the information on black points only is exchanged. For each $i = 1, 2, \ldots, \text{nnbd}$, the subroutine q9mesg sends the values

$$
\{u(\text{nsnd}(j)) : j = \text{isnd}(i), \text{isnd}(i + 1) - 1)\}
$$

to the processor responsible for the nbd(i)-th submatrix and receives the values

$$
\{u(\text{nrec}(j)) : j = \text{irec}(i), \text{irec}(i + 1) - 1)\}
$$

from the processor responsible for the nbd(i)-th submatrix. The actual code implementing the above level 2 BLAS operation is shown in Figure 5.2.

call q9mesg(u,150, 0) 
      do 20 jj = 2,maxnz 
          call q5i9gr (n,u,idco(1,jj),work) 
          do 10 ii = 1,neqn 
              10 rhs(ii) = rhs(ii) - coef(ii,jj)*work(ii) 
      20 continue 

Figure 5.2 This code segment implements the multiplication of the local submatrix $A_{ij}$ with the local vector $u$ after its subvector $(u_j, j = \text{neqn} + 1, \text{neqa})$ is updated with values received from the neighbor processors.
Additional computations this subtask involves are the inner product multiplications with subvectors that correspond to the submatrices of the original system. The local inner products are implemented by the function rlbldo which corresponds to the routine rlbldo of the sequential code. The local results are summed up by the routine globcom.

5.3.2 Red/Black Ordering for the Parallel ITPACK Modules

The red/black ordering can be applied for only 5 point-star finite difference methods with Dirichlet boundary conditions. Note that, in the mixed boundary condition problem, each equation is connected to a maximum of seven adjacent grid points. The currently implemented red/black module uses the geometric(grid) information rather than the connectivity of the algebraic equations. The red/black module first computes the global geometry position corresponding to the local algebraic index of each grid point. If a grid point has a position \( (i,j) \) in the whole domain and \( i + j \) is even/odd then we label it red/black respectively. This way the red/black module is totally parallelized.

After the application of the red/black ordering, the subroutine q4rbmg sorts the communicating information (i.e., local coupled unknown indices) in nsnd and nrec into the red and black subsets of the unknowns. This grouping of the unknowns according to the color of the corresponding grid points makes it easy for the local communication routine q9mesg(data,msgtype,irbflag) (see Figure 5.2) to exchange the data between processors.

5.4 The Performance of Parallel ITPACK on the nCUBE II

In this section we study the performance of the //ITPACK library for 5-point star finite difference and finite element equations based on linear triangular elements. The nCUBE II parallel machine is used.

5.4.1 The Performance of //ITPACK Modules for PDE Problems Defined on a Rectangle

In order to measure the performance of //ITPACK modules on the nCUBE II machine we consider five PDE problems defined on the unit square defined in Table 5.2. First, the data in Tables 5.3 and 5.4 compare the performance of the sequential and parallel ITPACK codes on a single nCUBE II processor. Data for the parallel version using 64 processors is also given. The ITPACK modules are applied on the 5-point star discretization equations (Table 5.3) and a linear FEM discretization (Table 5.4) corresponding to problem A with \( \alpha = 1.5 \). After 100 iterations, the data in Table...
Table 5.2 A set of PDE problems defined on the unit square for the performance evaluation of the //ITPACK modules.

<table>
<thead>
<tr>
<th>problem</th>
<th>definition</th>
<th>true solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(u_{xx} + u_{yy} = f)</td>
<td>(u(x, y) = \frac{(x^{n}-x)(x^{n}-y)}{(a(1-x^{2})-a(1-y^{2}))^2})</td>
</tr>
<tr>
<td>B</td>
<td>(u_{xx} + u_{yy} = f)</td>
<td>(u(x, y) = \frac{(x^{n}-x)(x^{n}-y)}{(a(1-x^{2})-a(1-y^{2}))^2})</td>
</tr>
<tr>
<td>C</td>
<td>(u_{xx} + u_{yy} - [100 + \cos(2\pi x) + \sin(3\pi y)] u = f)</td>
<td>(u(x, y) = \frac{-0.31[5.4 - \cos(4\pi x)]\sin(\pi x)}{(y^{2} - y)[5.4 - \cos(4\pi y)]} \frac{(1+(4(4x-0.5)^{2}+4(y-0.5)^{2})^{2})^{-1} - 0.5}{100 + \cos(2\pi x) + \sin(3\pi y)})</td>
</tr>
<tr>
<td>D</td>
<td>(-x^n u_{xx} - y^n u_{yy} - \alpha x^{n-1} u_{xx} - \alpha y^{n-1} u_{yy} + (xy)^n u = f)</td>
<td>(u(x, y) = 3e^{x+y}(x - x^2)(y - y^2))</td>
</tr>
<tr>
<td>E</td>
<td>(u_{xx} + u_{yy} = f)</td>
<td>(u(x, y) = 1)</td>
</tr>
</tbody>
</table>

5.3 indicate that the parallel code on a single nCUBE processor is slightly faster than the sequential code. Moreover, it exhibits more than 50% efficiency on a 64 processor configurations for this rather small problem (10,000 equations). The fifth column of Table 5.3 suggests that the convergence (number of iterations) varies among the ITPACK modules. The fastest convergence is obtained by the RSCG module followed by SSOR-CG. Table 5.3 indicates similar behavior for the applicable ITPACK modules to the FEM equations. The performance evaluation of Tables 5.5 and 5.6 differ from the previous two in that the iteration goes until it converges, or stops after 1000 iterations. All //ITPACK modules but the two SSOR modules converge for the termination test specified (1000 iterations). For the convergent modules, we observe that the sequential ones converge faster. This might be due to the differences of the BLAS routines used. We notice speedups of 34.11 to 47.09 for the difference equations and 35.38 to 43.12 for finite element ones. In Tables 5.7 - 5.10, we see similar performances. Tables 5.11 and 5.12 also give data for a model problem but with mixed boundary conditions. The data indicate similar performance.

Next, we estimate the timing of each phase of the PDE solver based on the //ITPACK modules and discuss their scalability. For this we have selected a parallel 5-point star discretization module that computes the optimal decomposition for a given machine configuration and grid size or accepts input values for the \(p \times q\) decomposition. Table 5.13 presents timings and error data for all phases of the numerical solution of the PDE problem E. The Jacobi-CG parallel iterative solver is used for these computations. Observe that the communication cost is a small percentage of
the entire computation cost (smaller than 3% for a 200 \times 200 grid on a 64 processor configuration). This is true for all the //ITPACK modules.

Table 5.14 indicates a very impressive speedup and Table 5.15 shows almost 96% scaled speedup (the ratio of the computing time using a finitely many processors to the computing time using only one processor assuming that the problem size is proportional to the number of processors.) for the SOR module. The data are similar for all the //ITPACK modules. The speedups are computed with respect to the time of the //ITPACK modules on a single processor. Table 5.16 shows the behavior of the applicable //ITPACK modules for the finite element equations. Their performance is similar. We believe that the results presented here show both the scalability of parallel iterative methods and the almost optimal behavior of the parallel implementation of the //ITPACK modules for PDE problems defined on rectangular domains.
Table 5.3 The timings in seconds of the sequential and parallel ITPACK modules to solve the system of 5-point star equations obtained from the discretization of the problem A with $\alpha = 1.5$. A $100 \times 100$ grid is used and single precision arithmetic. The iteration process is terminated after 100 iterations. The speedup is given in parenthesis after the time in 64 processors. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points. The error for the module which does not converge in 1000 iterations is indicated by *.

<table>
<thead>
<tr>
<th>module</th>
<th>Time on one nCUBE II proc (sequential version)</th>
<th>Time on one nCUBE II proc (parallel version)</th>
<th>Time on 64 nCUBE II processors</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOR</td>
<td>62.70</td>
<td>57.19</td>
<td>1.58 (36.20)</td>
<td>2.66E-01</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>71.00</td>
<td>65.80</td>
<td>1.72 (38.26)</td>
<td>1.10E-02</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>65.80</td>
<td>60.17</td>
<td>1.52 (39.59)</td>
<td>4.01E-01</td>
</tr>
<tr>
<td>RSCG</td>
<td>61.15</td>
<td>55.87</td>
<td>1.63 (34.28)</td>
<td>7.91E-04</td>
</tr>
<tr>
<td>RSSI</td>
<td>57.87</td>
<td>52.48</td>
<td>1.56 (33.64)</td>
<td>2.94E-02</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>133.92</td>
<td>129.22</td>
<td>3.58 (36.09)</td>
<td>2.14E-03</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>126.99</td>
<td>122.01</td>
<td>3.34 (36.53)</td>
<td>6.08E-02</td>
</tr>
</tbody>
</table>

Table 5.4 The timings in seconds of the applicable sequential and parallel ITPACK modules to solve the system of FEM (linear triangular elements) equations obtained from the discretization of the problem A with $\alpha = 1.5$. A mesh of 13780 elements is used and single precision arithmetic. The iteration process is terminated after 100 iterations. The speedup is given in parenthesis after the time in 64 processors. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>module</th>
<th>Time on one nCUBE II proc (sequential version)</th>
<th>Time on one nCUBE II proc (parallel version)</th>
<th>Time on 64 nCUBE II processors</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobi-CG</td>
<td>84.30</td>
<td>84.27</td>
<td>2.48 (33.98)</td>
<td>8.25E-04</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>81.66</td>
<td>81.64</td>
<td>1.95 (41.87)</td>
<td>2.85E-01</td>
</tr>
</tbody>
</table>
Table 5.5 The timings in seconds of the sequential and parallel ITPACK modules to solve the system of 5-point star equations obtained from the discretization of the problem A with $\alpha = 1.5$. A $100 \times 100$ grid is used and single precision arithmetic. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ [RB85] or after 1000 iterations. The speedup is given in parenthesis after the time in 64 processors. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points. The error for the module which does not converge in 1000 iterations is indicated by *.

<table>
<thead>
<tr>
<th>module</th>
<th>Time on one nCUBE II proc (sequential version)</th>
<th>Time on one nCUBE II proc (parallel version)</th>
<th>Time on 64 nCUBE II processors</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOR</td>
<td>142.53</td>
<td>141.81</td>
<td>4.00 (35.45)</td>
<td>7.94E-04</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>130.96</td>
<td>133.74</td>
<td>2.84 (47.09)</td>
<td>8.08E-04</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>523.89</td>
<td>532.91</td>
<td>13.17 (40.46)</td>
<td>7.84E-04</td>
</tr>
<tr>
<td>RSCG</td>
<td>61.19</td>
<td>62.45</td>
<td>1.64 (38.08)</td>
<td>7.86E-04</td>
</tr>
<tr>
<td>RSSI</td>
<td>460.39</td>
<td>467.59</td>
<td>13.71 (34.11)</td>
<td>7.86E-04</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>1206.48</td>
<td>1226.70</td>
<td>33.69 (36.41)</td>
<td>2.94E-03*</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td></td>
<td></td>
<td>29.99 ( )</td>
<td>6.08E-02*</td>
</tr>
</tbody>
</table>

Table 5.6 The timings in seconds of the applicable sequential and parallel ITPACK modules to solve the system of FEM (linear triangular elements) equations obtained from the discretization of the problem A with $\alpha = 1.5$. A mesh of 13780 elements is used and single precision arithmetic. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ [RB85] or after 1000 iterations. The speedup is given in parenthesis after the time in 64 processors. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>module</th>
<th>Time on one nCUBE II proc (sequential version)</th>
<th>Time on one nCUBE II proc (parallel version)</th>
<th>Time on 64 nCUBE II processors</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobi-CG</td>
<td>130.36</td>
<td>129.12</td>
<td>3.65 (35.38)</td>
<td>7.08E-04</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>721.02</td>
<td>711.50</td>
<td>16.50 (43.12)</td>
<td>7.09E-04</td>
</tr>
</tbody>
</table>
Table 5.7  The timings in seconds of the sequential and parallel ITPACK modules to solve the system of 5-point star equations obtained from the discretization of the problem B with $\alpha = 4.5$. A $100 \times 100$ grid is used and single precision arithmetic. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ [RE85] or after 1000 iterations. The speedup is given in parenthesis after the time in 64 processors. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points. The error for the module which does not converge in 1000 iterations is indicated by *.

<table>
<thead>
<tr>
<th>module</th>
<th>Time on one nCUBE II proc (sequential version)</th>
<th>Time on one nCUBE II proc (parallel version)</th>
<th>Time on 64 nCUBE II processors</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOR</td>
<td>191.07</td>
<td>190.78</td>
<td>5.32 (35.86)</td>
<td>2.31E-04</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>149.01</td>
<td>149.73</td>
<td>3.87 (38.69)</td>
<td>3.38E-04</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>537.71</td>
<td>538.10</td>
<td>13.19 (40.80)</td>
<td>2.61E-04</td>
</tr>
<tr>
<td>RSCG</td>
<td>70.04</td>
<td>70.68</td>
<td>2.17 (32.57)</td>
<td>3.21E-04</td>
</tr>
<tr>
<td>RSSI</td>
<td>468.49</td>
<td>470.20</td>
<td>13.70 (34.32)</td>
<td>2.35E-04</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>1236.38</td>
<td>1237.02</td>
<td>34.01 (36.37)</td>
<td>1.96E-02*</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>1320.53</td>
<td>1325.91</td>
<td>30.00 (44.20)</td>
<td>2.69E-01*</td>
</tr>
</tbody>
</table>

Table 5.8  The timings in seconds of the applicable sequential and parallel ITPACK modules to solve the system of FEM (linear triangular elements) equations obtained from the discretization of the problem B with $\alpha = 4.5$. A mesh of 13780 elements is used and single precision arithmetic. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ [RB85] or after 1000 iterations. The speedup is given in parenthesis after the time in 64 processors. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>module</th>
<th>Time on one nCUBE II proc (sequential version)</th>
<th>Time on one nCUBE II proc (parallel version)</th>
<th>Time on 64 nCUBE II processors</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobi-CG</td>
<td>180.56</td>
<td>182.82</td>
<td>5.21 (35.09)</td>
<td>6.87E-04</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>710.71</td>
<td>717.52</td>
<td>16.42 (43.70)</td>
<td>6.55E-04</td>
</tr>
</tbody>
</table>
Table 5.9: The timings in seconds of the sequential and parallel ITPACK modules to solve the system of 5-point star equations obtained from the discretization of the problem C. A 100 x 100 grid is used and single precision arithmetic. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ [RB85] or after 1000 iterations. The speedup is given in parenthesis after the time in 64 processors. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points. The error for the module which does not converge in 1000 iterations is indicated by *.

<table>
<thead>
<tr>
<th>module</th>
<th>Time on one nCUBE II proc (sequential version)</th>
<th>Time on one nCUBE II proc (parallel version)</th>
<th>Time on 64 nCUBE II processors</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOR</td>
<td>62.07</td>
<td>60.93</td>
<td>1.77 (34.42)</td>
<td>4.06E-04</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>73.96</td>
<td>77.30</td>
<td>1.92 (40.26)</td>
<td>4.37E-04</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>506.56</td>
<td>530.36</td>
<td>13.08 (40.55)</td>
<td>4.39E-04</td>
</tr>
<tr>
<td>RSCG</td>
<td>31.70</td>
<td>31.75</td>
<td>1.07 (29.67)</td>
<td>4.36E-04</td>
</tr>
<tr>
<td>RSSI</td>
<td>464.74</td>
<td>460.05</td>
<td>13.57 (33.90)</td>
<td>4.38E-04</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>1257.88</td>
<td>1280.20</td>
<td>52.98 (24.16)</td>
<td>1.73E-03*</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>1081.79</td>
<td>1067.81</td>
<td>29.66 (36.00)</td>
<td>4.37E-04</td>
</tr>
</tbody>
</table>

Table 5.10: The timings in seconds of the applicable sequential and parallel ITPACK modules to solve the system of FEM (linear triangular elements) equations obtained from the discretization of the problem C. A mesh of 13780 elements is used and single precision arithmetic. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ [RB85] or after 1000 iterations. The speedup is given in parenthesis after the time in 64 processors. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>module</th>
<th>Time on one nCUBE II proc (sequential version)</th>
<th>Time on one nCUBE II proc (parallel version)</th>
<th>Time on 64 nCUBE II processors</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobi-CG</td>
<td>71.98</td>
<td>71.93</td>
<td>2.36 (30.48)</td>
<td>1.10E-03</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>115.53</td>
<td>114.75</td>
<td>2.69 (42.66)</td>
<td>1.08E-03</td>
</tr>
</tbody>
</table>
Table 5.11 The timings in seconds of the sequential and parallel ITPACK modules to solve the system of 5-point star equations obtained from the discretization of the problem D. A 100 x 100 grid is used and single precision arithmetic. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ [RB85] or after 1000 iterations. The speedup is given in parenthesis after the time in 64 processors. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points. The error for the module which does not converge in 1000 iterations is indicated by *. 

<table>
<thead>
<tr>
<th>module</th>
<th>Time on one nCUBE II proc (sequential version)</th>
<th>Time on one nCUBE II proc (parallel version)</th>
<th>Time on 64 nCUBE II processors</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOR</td>
<td>129.89</td>
<td>131.75</td>
<td>3.73 (35.32)</td>
<td>1.72E-04</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>195.50</td>
<td>195.49</td>
<td>5.60 (34.91)</td>
<td>1.55E-04</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>529.38</td>
<td>528.54</td>
<td>13.12 (40.29)</td>
<td>1.31E-04</td>
</tr>
<tr>
<td>RSCG</td>
<td>89.50</td>
<td>89.46</td>
<td>3.11 (28.77)</td>
<td>1.37E-04</td>
</tr>
<tr>
<td>RSSI</td>
<td>466.35</td>
<td>463.09</td>
<td>13.67 (33.88)</td>
<td>1.31E-04</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>1253.66</td>
<td>1299.63</td>
<td>40.56 (32.04)</td>
<td>8.92E-04*</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>1086.19</td>
<td>1082.54</td>
<td>30.35 (35.67)</td>
<td>2.15E-04</td>
</tr>
</tbody>
</table>

Table 5.12 The timings in seconds of the applicable sequential and parallel ITPACK modules to solve the system of FEM (linear triangular elements) equations obtained from the discretization of the problem D. A mesh of 13780 elements is used and single precision arithmetic. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ [RB85] or after 1000 iterations. The speedup is given in parenthesis after the time in 64 processors. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>module</th>
<th>Time on one nCUBE II proc (sequential version)</th>
<th>Time on one nCUBE II proc (parallel version)</th>
<th>Time on 64 nCUBE II processors</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobi-CG</td>
<td>190.48</td>
<td>191.43</td>
<td>5.29 (36.19)</td>
<td>4.05E-04</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>712.91</td>
<td>715.59</td>
<td>15.81 (45.26)</td>
<td>3.99E-04</td>
</tr>
</tbody>
</table>
Table 5.13 The performance of a parallel implementation of Jacobi-CG to solve the system of 5-point star equations obtained from the discretization of the problem E. A 200 x 200 grid is used and the process is terminated after 100 iterations. The performance is measured in terms of the timings in seconds of the various phases of the solution for different machine configurations. In addition, we list cost of each solution phase as percentage of the overall cost. The maximum abs error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>Time and error measure</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexing (% of total)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Solution (% of total)</td>
<td>95.565</td>
<td>95.500</td>
<td>96.373</td>
<td>96.333</td>
<td>96.136</td>
<td>96.041</td>
<td>95.876</td>
</tr>
<tr>
<td>Comm. (% of total)</td>
<td>0.000</td>
<td>0.114</td>
<td>0.205</td>
<td>0.606</td>
<td>1.039</td>
<td>1.746</td>
<td>2.710</td>
</tr>
<tr>
<td>Discretization (sec)</td>
<td>17.227</td>
<td>8.801</td>
<td>4.496</td>
<td>2.352</td>
<td>1.266</td>
<td>0.656</td>
<td>0.352</td>
</tr>
<tr>
<td>Indexing (sec)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Solution (sec)</td>
<td>484.602</td>
<td>242.781</td>
<td>121.695</td>
<td>61.875</td>
<td>31.590</td>
<td>16.109</td>
<td>8.359</td>
</tr>
<tr>
<td>Total solve time (sec)</td>
<td>501.836</td>
<td>251.586</td>
<td>126.191</td>
<td>64.230</td>
<td>32.859</td>
<td>16.773</td>
<td>8.719</td>
</tr>
<tr>
<td>Communication (sec)</td>
<td>0.008</td>
<td>0.281</td>
<td>0.254</td>
<td>0.379</td>
<td>0.332</td>
<td>0.281</td>
<td>0.227</td>
</tr>
<tr>
<td>Time per iteration (sec)</td>
<td>4.846</td>
<td>2.428</td>
<td>1.217</td>
<td>0.619</td>
<td>0.316</td>
<td>0.161</td>
<td>0.084</td>
</tr>
<tr>
<td># of iterations</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 5.14 The speedup obtained on the nCUBE II by each ITPACK module for the problem E and grid size 200 x 200 using 100 iterations.

<table>
<thead>
<tr>
<th>Processors</th>
<th>Jacobi-CG</th>
<th>Jacobi-SI</th>
<th>RSCG</th>
<th>RSSI</th>
<th>SSOR-CG</th>
<th>SSOR-SI</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.99</td>
<td>1.99</td>
<td>1.99</td>
<td>1.99</td>
<td>1.99</td>
<td>1.99</td>
</tr>
<tr>
<td>4</td>
<td>3.98</td>
<td>4.00</td>
<td>3.93</td>
<td>3.94</td>
<td>3.96</td>
<td>3.97</td>
</tr>
<tr>
<td>8</td>
<td>7.81</td>
<td>7.86</td>
<td>7.65</td>
<td>7.67</td>
<td>7.73</td>
<td>7.72</td>
</tr>
<tr>
<td>16</td>
<td>15.27</td>
<td>15.36</td>
<td>14.72</td>
<td>14.79</td>
<td>15.01</td>
<td>14.96</td>
</tr>
<tr>
<td>32</td>
<td>29.92</td>
<td>30.25</td>
<td>28.32</td>
<td>28.59</td>
<td>29.14</td>
<td>29.08</td>
</tr>
<tr>
<td>64</td>
<td>57.56</td>
<td>58.69</td>
<td>52.89</td>
<td>54.06</td>
<td>55.62</td>
<td>55.52</td>
</tr>
</tbody>
</table>
Table 5.15  The timings of the SQR module for one iteration when the number of equations per processor is held constant as the number of processors increases.

<table>
<thead>
<tr>
<th># Processors</th>
<th>Grid size</th>
<th>Time per iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>102 × 52</td>
<td>.573</td>
</tr>
<tr>
<td>2</td>
<td>102 × 102</td>
<td>.574</td>
</tr>
<tr>
<td>4</td>
<td>202 × 102</td>
<td>.578</td>
</tr>
<tr>
<td>8</td>
<td>202 × 202</td>
<td>.584</td>
</tr>
<tr>
<td>16</td>
<td>402 × 202</td>
<td>.590</td>
</tr>
<tr>
<td>32</td>
<td>402 × 402</td>
<td>.592</td>
</tr>
<tr>
<td>64</td>
<td>802 × 402</td>
<td>.599</td>
</tr>
</tbody>
</table>

Table 5.16  The speedup of some //ITPACK modules on the nCUBE II used to determine a linear triangular FEM approximation to the problem E with a mesh of 13780 elements using 100 iterations.

<table>
<thead>
<tr>
<th>Processors</th>
<th>Jacobi-CG</th>
<th>Jacobi-SI</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2.00</td>
<td>2.01</td>
</tr>
<tr>
<td>4</td>
<td>3.94</td>
<td>3.98</td>
</tr>
<tr>
<td>8</td>
<td>7.21</td>
<td>7.59</td>
</tr>
<tr>
<td>16</td>
<td>13.30</td>
<td>14.83</td>
</tr>
<tr>
<td>32</td>
<td>22.89</td>
<td>26.77</td>
</tr>
<tr>
<td>64</td>
<td>34.29</td>
<td>42.62</td>
</tr>
</tbody>
</table>

5.4.2 The Performance of //ITPACK Modules for PDE Problems Defined on a General Domain

Here we attempt to see the effect of varying the geometry of the PDE domains on the computational behavior of the //ITPACK modules. For this, we consider a Helmholtz-type problem

\[ u_{xx} + u_{yy} - [100 + \cos(2\pi x) + \sin(3\pi y)]u = f(x, y) \quad (5.1) \]

with solution

\[
  u(x, y) = -0.31[5.4 - \cos(4\pi x)]\sin(\pi x)(y^2 - y)[5.4 - \cos(4\pi y)] \\
  [(1 + (4(x - 0.5)^2 + 4(y - 0.5)^2)^4)^{-1} - 0.5]. \quad (5.2)
\]

The PDE (5.1) is defined on a domain (see Figure 5.3) with boundary consisting of lines connecting the points (1, 0), (0, 0), (0, 0.5), (0.5, 1) and (1, 1) plus the half circle.
\[ x = 1 + 0.5 \sin(t) \\
y = 0.5 - 0.5 \cos(t) ; \quad t \in [0, \pi].\]

We refer to the PDE problem (5.1), (5.2) as problem F and the same problem with Dirichlet conditions on the half circle and Neumann conditions at the rest as problem G.

![Figure 5.3 The PDE domain of the boundary value problem (5.1).](image)

Furthermore, we consider another model problem

\[ u_{xx} + u_{yy} = f \]  

with solution

\[ u(x, y) = \frac{(x^\alpha - x)(y^\alpha - y)}{(\alpha^{\alpha/(1-\alpha)} - \alpha^{1/(1-\alpha)})^2} \]

with \( \alpha = 4.5 \). The PDE (5.3) is defined on the L-shaped domain with boundary consisting of lines connecting points \((0, 0)\), \((0, 1)\), \((0.5, 1)\), \((0.5, 0.5)\), \((1.5, 0.5)\), \((1, 0)\), and subject to Dirichlet boundary conditions. We refer to the PDE problem (5.3), (5.4) as problem H.

Tables 5.17 to 5.22 list the execution times of each //ITPACK module for problem F and different nCUBE II configurations. The data in Tables 5.17 to 5.19 allow us to evaluate the efficiency of the parallel processing. Efficiency is the speedup observed divided by the number of processors used. We compute that the efficiencies per iteration varies from 0% to 10% for solving the finite difference equations. In the case of finite element equations these modules exhibit efficiency 99% (2 processors) to 63% (64 processors) for rather small size system of equations. The data in tables 5.20 to 5.22 show that the //ITPACK modules exhibit similar performance for the default termination parameter. Finally, these data suggest that the difference of efficiencies between the general and encapsulated 5-point star solutions varies from 0% to 10% in favor of general. These data do not include the decomposition cost for both cases. It is observed in Section 4.1.2 that this decomposition cost can be significant for the
for the general 5-point star solution. Tables 5.23 to 5.28 indicate the performance of the two applicable //ITPACK modules (Jacobi-SI and CG) for the finite element and finite difference equations obtained from the discretization of the Problem G. Again only a small difference in the per efficiency performance is observed between the general and encapsulated methods. The efficiency of the two modules varies from 98% (2 processors) to 79% (64 processors). Taking into account the small size of the systems considered, we conclude that this performance is close to optimal. The data for the finite element equations indicate similar performance.

Finally, Tables 5.29 5.34 measure the performance of the //ITPACK modules on Problem H. The per iteration efficiency varies from 56% to 99%. SOR and RSCG exhibit the best performance while the SSOR based modules exhibit the worst.
Table 5.17 The timings\( (t) \) and speedups\( (s) \) of the //ITPACK modules to solve the problem \( F \) using the *general* parallel 5-point star module and for a \( 150 \times 150 \) grid with the \( p \times q \) decomposition of the non-rectangular domain. The iteration process is terminated after 100 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SOR</td>
<td>110.79</td>
<td>55.67</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.99</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>129.97</td>
<td>65.35</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.99</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>119.32</td>
<td>59.88</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.99</td>
</tr>
<tr>
<td>RSCG</td>
<td>119.04</td>
<td>55.39</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.99</td>
</tr>
<tr>
<td>RSI</td>
<td>103.01</td>
<td>51.76</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.99</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>271.50</td>
<td>136.46</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.99</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>245.74</td>
<td>123.39</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.99</td>
</tr>
</tbody>
</table>

Table 5.18 The timings\( (t) \) and speedups\( (s) \) of the //ITPACK modules to solve the problem \( F \) using the *encapsulated* parallel 5-point star module and for a \( 150 \times 150 \) grid with the \( p \times q \) decomposition of the non-rectangular domain. The iteration process is terminated after 100 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SOR</td>
<td>111.02</td>
<td>56.72</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>130.50</td>
<td>66.72</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>118.88</td>
<td>60.68</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>RSCG</td>
<td>109.63</td>
<td>56.12</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.95</td>
</tr>
<tr>
<td>RSI</td>
<td>102.83</td>
<td>52.33</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>270.89</td>
<td>138.53</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>241.97</td>
<td>123.63</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.96</td>
</tr>
</tbody>
</table>
Table 5.19 The timings (t) and speedups (s) of the //ITPACK modules to solve the problem F using the parallel FEM for 17331 element with the $p \times q$ decomposition of the non-rectangular domain. The iteration process is terminated after 100 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>t</td>
<td>105.75</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>t</td>
<td>103.40</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.20 The timings (t) and speedups (s) of the //ITPACK modules to solve the problem F using the general parallel 5-point star module and for a 150 x 150 grid with the $p \times q$ decomposition of the non-rectangular domain. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ [RB85] or after 1000 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SOR</td>
<td>t</td>
<td>149.85</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>t</td>
<td>446.83</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>t</td>
<td>1032.59</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>RSCG</td>
<td>t</td>
<td>137.12</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>RSSI</td>
<td>t</td>
<td>904.24</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>t</td>
<td>2483.55</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>t</td>
<td>2099.13</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Table 5.21: The timings (t) and speedups (s) of the /ITPACK modules to solve the problem F using the encapsulated parallel 5-point star module and for a 150 x 150 grid with the p x q decomposition of the non-rectangular domain. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ \cite{RB85} or after 1000 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>/ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SOR</td>
<td>t 150.50</td>
<td>76.89</td>
</tr>
<tr>
<td></td>
<td>s 1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>t 446.87</td>
<td>228.36</td>
</tr>
<tr>
<td></td>
<td>s 1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>t 1033.81</td>
<td>528.35</td>
</tr>
<tr>
<td></td>
<td>s 1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>RSCG</td>
<td>t 121.71</td>
<td>69.98</td>
</tr>
<tr>
<td></td>
<td>s 1.00</td>
<td>1.95</td>
</tr>
<tr>
<td>RSSI</td>
<td>t 901.78</td>
<td>461.18</td>
</tr>
<tr>
<td></td>
<td>s 1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>t 2487.80</td>
<td>1277.48</td>
</tr>
<tr>
<td></td>
<td>s 1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>t 2103.87</td>
<td>1077.71</td>
</tr>
<tr>
<td></td>
<td>s 1.00</td>
<td>1.95</td>
</tr>
</tbody>
</table>

Table 5.22: The timings (t) and speedups (s) of the /ITPACK modules to solve the problem F using the parallel FEM for 17331 element with the p x q decomposition of the non-rectangular domain. The iteration process is terminated when the ITPACK convergence test is taken to be $5.96 \times 10^{-5}$ \cite{RB85} or after 1000 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>/ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>t 119.34</td>
<td>60.10</td>
</tr>
<tr>
<td></td>
<td>s 1.00</td>
<td>1.99</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>t 160.45</td>
<td>80.63</td>
</tr>
<tr>
<td></td>
<td>s 1.00</td>
<td>1.99</td>
</tr>
</tbody>
</table>
Table 5.23 The timings(t) and speedups(s) of the //ITPACK modules to solve the problem G using the *general* parallel 5-point star module and for a 150 × 150 grid with the \( p \times q \) decomposition of the non-rectangular domain. The iteration process is terminated after 100 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>t</td>
<td>172.99</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>t</td>
<td>165.40</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.24 The timings(t) and speedups(s) of the //ITPACK modules to solve the problem G using the *encapsulated* parallel 5-point star module and for a 150 × 150 grid with the \( p \times q \) decomposition of the non-rectangular domain. The iteration process is terminated after 100 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>t</td>
<td>174.11</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>t</td>
<td>165.92</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.25 The timings(t) and speedups(s) of the //ITPACK modules to solve the problem G using the parallel FEM for 17331 element with the \( p \times q \) decomposition of the non-rectangular domain. The iteration process is terminated after 100 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>t</td>
<td>109.21</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>t</td>
<td>106.55</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Table 5.26 The timings\((t)\) and speedups\((s)\) of the //ITPACK modules to solve the problem G using the \textit{general} parallel 5-point star module and for a 150 \(\times\) 150 grid with the \(p \times q\) decomposition of the non-rectangular domain. The iteration process is terminated when the ITPACK convergence test is taken to be 5.96 \(\times\) \(10^{-5}\) [RB85] or after 1000 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>1380.93</td>
<td>701.55</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.97</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>1428.99</td>
<td>726.06</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.97</td>
</tr>
</tbody>
</table>

Table 5.27 The timings\((t)\) and speedups\((s)\) of the //ITPACK modules to solve the problem G using the \textit{encapsulated} parallel 5-point star module and for a 150 \(\times\) 150 grid with the \(p \times q\) decomposition of the non-rectangular domain. The iteration process is terminated when the ITPACK convergence test is taken to be 5.96 \(\times\) \(10^{-5}\) [RB85] or after 1000 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>1364.99</td>
<td>688.59</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.98</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>1415.74</td>
<td>712.21</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.99</td>
</tr>
</tbody>
</table>

Table 5.28 The timings\((t)\) and speedups\((s)\) of the //ITPACK modules to solve the problem G using the parallel FEM for 17331 element with the \(p \times q\) decomposition of the non-rectangular domain. The iteration process is terminated when the ITPACK convergence test is taken to be 5.96 \(\times\) \(10^{-5}\) [RB85] or after 1000 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>131.97</td>
<td>67.31</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.96</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>181.38</td>
<td>92.32</td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>1.98</td>
</tr>
</tbody>
</table>
Table 5.29 The timings\((t)\) and speedups\((s)\) of the \texttt{//ITPACK} modules to solve the problem \(H\) using the \textit{general} parallel 5-point star module and for a \(150 \times 150\) grid with the \(p \times q\) decomposition of the non-rectangular domain. The iteration process is terminated after 100 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>\texttt{//ITPACK module}</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SOR</td>
<td>(t)</td>
<td>97.14</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>(t)</td>
<td>115.99</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>(t)</td>
<td>104.86</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>RSCG</td>
<td>(t)</td>
<td>96.82</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>RSSI</td>
<td>(t)</td>
<td>90.70</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>(t)</td>
<td>241.75</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>(t)</td>
<td>218.97</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.30 The timings\((t)\) and speedups\((s)\) of the \texttt{//ITPACK} modules to solve the problem \(H\) using the \textit{encapsulated} parallel 5-point star module and for a \(150 \times 150\) grid with the \(p \times q\) decomposition of the non-rectangular domain. The iteration process is terminated after 100 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>\texttt{//ITPACK module}</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SOR</td>
<td>(t)</td>
<td>98.12</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>(t)</td>
<td>114.99</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>(t)</td>
<td>104.73</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>RSCG</td>
<td>(t)</td>
<td>97.64</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>RSSI</td>
<td>(t)</td>
<td>91.77</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>(t)</td>
<td>242.99</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>(t)</td>
<td>233.58</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Table 5.31 The timings\((t)\) and speedups\((s)\) of the //ITPACK modules to solve the problem H using the parallel FEM for 17331 element with the \(p \times q\) decomposition of the non-rectangular domain. The iteration process is terminated after 100 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>(t)</td>
<td>59.16</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>(t)</td>
<td>57.82</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.32 The timings\((t)\) and speedups\((s)\) of the //ITPACK modules to solve the problem H using the general parallel 5-point star module and for a 150 \(\times\) 150 grid with the \(p \times q\) decomposition of the non-rectangular domain. The iteration process is terminated when the ITPACK convergence test is taken to be \(5.96 \times 10^{-5}\) [RB85] or after 1000 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SOR</td>
<td>(t)</td>
<td>260.56</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>(t)</td>
<td>438.78</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>(t)</td>
<td>912.73</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>RSCG</td>
<td>(t)</td>
<td>153.98</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>RSSI</td>
<td>(t)</td>
<td>788.73</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>(t)</td>
<td>2229.38</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>(t)</td>
<td>2308.99</td>
</tr>
<tr>
<td></td>
<td>(s)</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Table 5.33 The timings\( (t) \) and speedups\( (s) \) of the //ITPACK modules to solve the problem \( H \) using the \textit{encapsulated} parallel 5-point star module and for a 150 \( \times \) 150 grid with the \( p \times q \) decomposition of the non-rectangular domain. The iteration process is terminated when the //ITPACK convergence test is taken to be \( 5.96 \times 10^{-5} \) [RB85] or after 1000 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>SOR</td>
<td>t</td>
<td>259.35</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>t</td>
<td>442.33</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>t</td>
<td>918.33</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>RSCG</td>
<td>t</td>
<td>138.98</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>RSSI</td>
<td>t</td>
<td>789.04</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>SSOR-CG</td>
<td>t</td>
<td>2233.15</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>SSOR-SI</td>
<td>t</td>
<td>2317.79</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.34 The timings\( (t) \) and speedups\( (s) \) of the //ITPACK modules to solve the problem \( H \) using the parallel FEM for 17331 element with the \( p \times q \) decomposition of the non-rectangular domain. The iteration process is terminated when the //ITPACK convergence test is taken to be \( 5.96 \times 10^{-5} \) [RB85] or after 1000 iterations. The error is the maximum of the absolute values of the estimated solution value minus the true solution value over all grid points.

<table>
<thead>
<tr>
<th>//ITPACK module</th>
<th>number of processors</th>
<th>error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Jacobi-CG</td>
<td>t</td>
<td>95.47</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
<tr>
<td>Jacobi-SI</td>
<td>t</td>
<td>530.38</td>
</tr>
<tr>
<td></td>
<td>s</td>
<td>1.00</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


