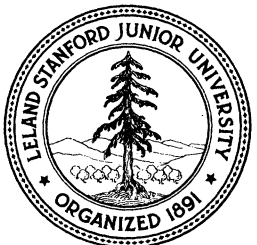
Schwarz Splitting and Template Operators

by

Wei Pai Tang

Department of Computer Science

Stanford University
Stanford, CA 94305



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-018
Exp. Date: Jun 30,

EPORT SECURITY CLASSIFICATION
unclassified

1b. RESTRICTIVE MARKINGS

CURITY CLASSIFICATION AUTHORITY

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release:
Distribution unlimited

ECLASSIFICATION / DOWNGRADING SCHEDULE

RFORMING ORGANIZATION REPORT NUMBER(S)
STAN-CS-87-1167

5. MONITORING ORGANIZATION REPORT NUMBER(S)

AME OF PERFORMING ORGANIZATION
Computer Science Department

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION

DDRESS (City, State, and ZIP Code)
Stanford University
Stanford, CA 94305

7b. ADDRESS (City, State, and ZIP Code)

AME OF FUNDING / SPONSORING RGANIZATION
ONR

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N00014-82-K-0335

DDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UN ACCESSION

TLE (Include Security Classification)
SCHWARZ SPLITTING AND TEMPLATE OPERATIONS

RSONAL AUTHOR(S)
lei Pai Tang

YPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
THESIS	FROM _____ TO _____	JUNE 1987	139

IPPLEMENTARY NOTATION

COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
ELD	GROUP	SUB-GROUP	

ISTRACT (Continue on reverse if necessary and identify by block number)

Schwarz alternating method ($\mathcal{SM}$) is an old mathematical technique dating from 1869. It was commonly believed that $\mathcal{SM}$ was a useful tool for theoretical analysis but not a very practical approach for computations. The earlier experiences showed that $\mathcal{SM}$ converged slowly. In this thesis, $\mathcal{SM}$ is reexamined and generalized. The governing factors of convergence of $\mathcal{SM}$ are explored through the analysis for the model problem. Based on this knowledge, many acceleration schemes can be combined with $\mathcal{SM}$ to yield a new type of iterative method for large sparse matrix problems. In particular, when these techniques are applied to the solution of the model problem, an optimal complexity can be achieved. Some generalizations of $\mathcal{SM}$, namely *Schwarz splittings* ($\mathcal{S}$), are presented here. For many important applications, such as performing parallel computations in a non-shared memory environment, using composite grids and also applying fast solvers in an irregular region, $\mathcal{S}$'s are found to be useful techniques.

In order to identify the types of problems for which $\mathcal{S}$'s are most suitable, a new structure for the linear operators called *template operators* has been developed. Some decay results for the elements of the inverses of sparse operators are given. These results provide a theoretical basis for determining when these $\mathcal{S}$ techniques can be used successfully.

ISTRIBUTION / AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION

NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE (Include Area Code)

22c OFFICE SYMBOL

ORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAG

SCHWARZ SPLITTING
AND
TEMPLATE OPERATORS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Wei Pai Tang
June 1987

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Joseph Oliger
(Principal Adviser)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Garry Rodrigue
University of California, Davis
and
Lawrence Livermore National Laboratory

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Joel Ferziger
Mechanical Engineering Department

Approved for the University Committee on Graduate Studies:

Dean of Graduate Studies

Abstract

Schwarz alternating method ($\mathcal{SAM}$) is an old mathematical technique dating from 1869. It was commonly believed that $\mathcal{SAM}$ was a useful tool for theoretical analysis but not a very practical approach for computations. The earlier experiences showed that $\mathcal{SAM}$ converged slowly. In this thesis, $\mathcal{SAM}$ is reexamined and generalized. The governing factors of convergence of $\mathcal{SAM}$ are explored through the analysis for the model problem. Based on this knowledge, many acceleration schemes can be combined with $\mathcal{SAM}$ to yield a new type of iterative method for large sparse matrix problems. In particular, when these techniques are applied to the solution of the model problem, an optimal complexity can be achieved. Some generalizations of $\mathcal{SAM}$, namely *Schwarz splittings* ($\mathcal{SS}$), are presented here. For many important applications, such as performing parallel computations in a non–shared memory environment, using composite grids and also applying fast solvers in an irregular region, $\mathcal{SS}$'s are found to be useful techniques.

In order to identify the types of problems for which $\mathcal{SS}$'s are most suitable, a new structure for the linear operators called *template operators* has been developed. Some decay results for the elements of the inverses of sparse operators are given. These results provide a theoretical basis for determining when these $\mathcal{SS}$ techniques can be used successfully.

Acknowledgement

The march leading to my Ph.D. has been long and difficult, though on its path I have also found happiness and fulfillment. There is a long list of individuals who have helped me in many ways. I want to thank them for what they have given me.

I am deeply indebted to Professor Joseph Oliger for providing guidance and support through my years at Stanford. The most valuable lesson I learned from him is not to just follow where the path may lead, but to go where there is no path. During the crises of my academic life, his confidence in me and his generous support have made the progress of this thesis possible. Now I am very proud of saying: I am another new member of the large academic Oliger "family".

I offer my most profound thanks to Professors Garry Rodrigue and Joel Ferziger. Their advice has greatly helped me in the course of my thesis. I will also benefit from this advice in my future career.

I am also very thankful to Professors Martin Schultz, Paul Swartztrauber, Micheal Overton, Tony Chan and William Gropp for their academic support when I needed it.

Let me in particular mention my good friend Steve Suhr here. He was the first reader of many of my drafts. He has spent a great deal of his time and energy helping me to improve this presentation. His help is very much appreciated.

I have spent seven years, which are the most valuable period of my academic life, at Stanford. Looking back, I am grateful to Professor Donald Knuth for his introducing me to Stanford. It was truly a turning point in my life. A whole new world of computer science at Stanford inspired me to launch this "long march " to a Ph.D. degree after being away from school for 15 years. Thanks are also due to Professors Gene Golub, Andrew Yao and Ronald Graham. Their encouragement made my study at Stanford a reality. Professor G. Golub also supported me for the first few years.

Without the support and love of many friends, this dissertation simply would not have been written. I especially want to thank Chris Fraley, Anita Mayo, Alan Siegel, Harry Mairson, Jeffrey Victor, Marsha Berger, David Foulser, Billy Wilson, Patrick Worley, William Skamarock, John Hershberger, Jeff Naughton and Doug Baxter. During these years, especially during the several crises towards my Ph.D., they stood up for me, held my hands, gave me strength, criticized my weakness, encouraged my progress, shared my joys ,,,,, . Their friendship has not only made my life at Stanford rich, it will also illuminate my future.

I regret that my mother could not live long enough to see me complete my degree. It was her dream to send me to a prestigious school. Unfortunately, I could not even say good bye to her when she was passing away. I dedicate this work to her with my tears and love.

My dearest daughter Qi has suffered from the lack of her father's love during these years. I had to leave her in China to pursue my study due to the well known reasons. When she grows older, she will understand why I would pay such a high cost to do that. I am wishing that she will follow and continue this march exploring the beauty of science.

Finally, I would like to thank Bao Zhang Tang for her love and support. Mere words cannot fully acknowledge what I have received from her.

This research was supported by NASA Ames Consortium Agreement NASA NCA2-150 and Office of Naval Research Contracts N00014-86-K-0565, N00014-82-K-0335, N00014-75-C-1132.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This dissertation is a reexamination and generalization of a very old mathematical technique –Schwarz Alternating Method ($\mathcal{SAM}$). It was commonly believed that $\mathcal{SAM}$ was a useful tool for theoretical analysis but not a very practical approach for computations. Some preliminary experiments had indicated that $\mathcal{SAM}$ was not promising because it converged too slowly. There was little knowledge of the factors governing the convergence of this method. In this study, an analysis of $\mathcal{SAM}$ for the model problem for elliptic partial differential equations is presented. The convergence factor of $\mathcal{SAM}$ is found to be a function of many components. Based on the analysis, many acceleration schemes can be combined with this idea to yield a competitive new type of iterative method for large sparse matrix problems. Generalizations of $\mathcal{SAM}$ are also introduced in this thesis. We show that $\mathcal{SAM}$ is not only suitable to solve elliptic partial differential equations, it is also a good computational model for many other important applications. Particularly, a new structural view of linear operators is presented which provides a useful tool for analyzing the behavior of sparse operators.

Here the original version of $\mathcal{SAM}$ is first introduced. Following this description the motivations of this study are discussed. Then a brief historical survey is presented. At the end of this introduction, the organization of the rest of this dissertation is outlined.

In the last century, Schwarz [Sch69] found that, for a region consisting of the union of two rectangular regions or disks, he could construct a sequence of solutions of the Laplace equation on subregions which would converge to the solution of the Laplace equation on the union. His method is now called $\mathcal{SAM}$. The description of a simple version of $\mathcal{SAM}$ is as follows:

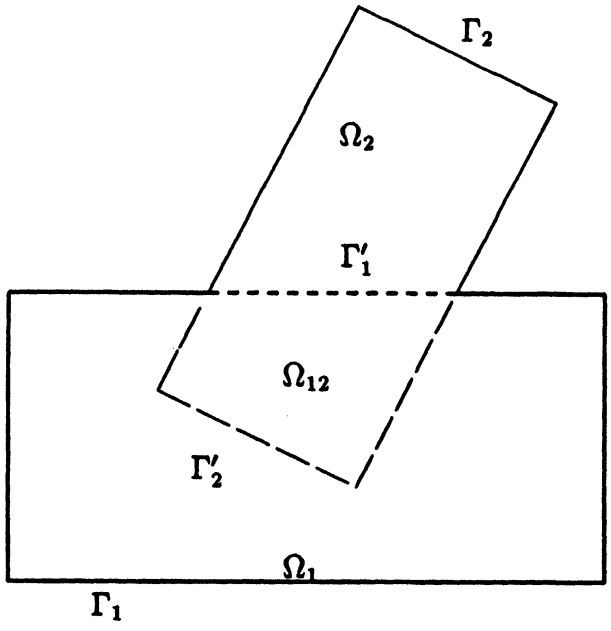


Figure 1.1: Two overlapping subregions

Consider the Dirichlet problem for an elliptic operator L [1]

$$\begin{cases} L(u) = f, & \mathbf{x} \in \Omega, \\ u\,|_{\Gamma_\Omega} = \psi, & \mathbf{x} \in \Gamma_\Omega \end{cases} \tag{1.1}$$

where Ω is a bounded region in k-dimensional space, Γ_Ω is the boundary of Ω, $\mathbf{x} = \{x_1, x_2, \cdots, x_k\}$ is the independent variable.

Schwarz split the solution domain Ω into two overlapping subdomains Ω_1 and Ω_2. Let $\Omega_{12} = \Omega_1 \cap \Omega_2 \neq 0$, Γ_{Ω_1}, Γ_{Ω_2}, $\Gamma_{\Omega_{12}}$ denote the boundaries of Ω_1, Ω_2 and Ω_{12} respectively (see fig.1.1). Let

$$\begin{aligned} \Gamma_{\Omega_1} &= \Gamma_1 + \Gamma_1', \\ \Gamma_{\Omega_2} &= \Gamma_2 + \Gamma_2'. \end{aligned}$$

where

$$\Gamma_1 = \Gamma_\Omega \cap \Gamma_{\Omega_1},$$

[1] We assume that the solution of this problem exists and is unique.

$$
\begin{aligned}
\Gamma_1' &= \Gamma_{\Omega_1} \cap \Gamma_{\Omega_{12}}, \\
\Gamma_2 &= \Gamma_{\Omega} \cap \Gamma_{\Omega_2}, \\
\Gamma_2' &= \Gamma_{\Omega_2} \cap \Gamma_{\Omega_{12}}.
\end{aligned}
$$

From this splitting we can formulate two coupled problems

$$
\begin{cases}
L(u_1) = f, & \mathbf{x} \in \Omega_1, \\
u_1 |_{\Gamma_{\Omega_1}} = \begin{cases} \psi, & \mathbf{x} \in \Gamma_1, \\ u_2, & \mathbf{x} \in \Gamma_1', \end{cases}
\end{cases} \tag{1.2}
$$

$$
\begin{cases}
L(u_2) = f, & \mathbf{x} \in \Omega_2, \\
u_2 |_{\Gamma_{\Omega_2}} = \begin{cases} \psi, & \mathbf{x} \in \Gamma_2, \\ u_1, & \mathbf{x} \in \Gamma_2'. \end{cases}
\end{cases} \tag{1.3}
$$

It is clear that u, the solution of (1.1), is the solution of (1.2) and (1.3). We may also easily show that:

$$
\begin{aligned}
u_1 &= u_2, && \mathbf{x} \in \Omega_{12}, \\
u_1 &= u, && \mathbf{x} \in \Omega_1, \\
u_2 &= u, && \mathbf{x} \in \Omega_2.
\end{aligned}
$$

Thus, the problem (1.1) is equivalent to the pair of problems (1.2) and (1.3). Since there are unknowns which are coupled in the boundary conditions, we cannot solve the two problems independently. But giving an initial guess $u |_{\Gamma_1'} = \psi_0$, we will be able to construct a sequence $\{u_1^{(i)}, u_2^{(i)}\}$ as follows:

$$
\begin{cases}
L(u_1^{(0)}) = f, & \mathbf{x} \in \Omega_1, \\
u_1^{(0)} |_{\Gamma_{\Omega_1}} = \begin{cases} \psi, & \mathbf{x} \in \Gamma_1, \\ \psi_0, & \mathbf{x} \in \Gamma_1', \end{cases}
\end{cases} \tag{1.4}
$$

$$
\begin{cases}
L(u_2^{(i)}) = f, & \mathbf{x} \in \Omega_2, \\
u_2^{(i)} |_{\Gamma_{\Omega_2}} = \begin{cases} \psi, & \mathbf{x} \in \Gamma_2, \\ u_1^{(i-1)}, & \mathbf{x} \in \Gamma_2', \end{cases}
\end{cases} \tag{1.5}
$$

$$\begin{cases} L(u_1^{(i)}) = f, & \mathbf{x} \in \Omega_1, \\ u_1^{(i)}|_{\Gamma_{\Omega_1}} = \begin{cases} \psi, & \mathbf{x} \in \Gamma_1, \\ u_2^{(i)}, & \mathbf{x} \in \Gamma_1', \end{cases} & \\ & i = 1, 2, \cdots. \end{cases} \tag{1.6}$$

It can now be shown that the sequence $\{u_1^{(i)}, u_2^{(i)}\}$ will converge to the solutions $\{u_1, u_2\}$ of (1.2) and (1.3) under certain conditions (see section 2.3). Then, from the solution of (1.2) and (1.3), the solution of (1.1) can be constructed.

Here we have just described the simplest version of SAM. Unlike some other techniques which usually are precise procedures for solving problems, SAM basically gives us only a philosophy for solving a problem. The freedom inherent in SAM provides great opportunity to incorporate many other techniques in order to obtain good performance.

- Freedom in the geometrical shapes of the subproblems. This freedom makes it possible to tailor the subregions to meet the requirements imposed by fast solvers or by grids.

- Freedom in the solution techniques for subproblems. We are able to choose different solution techniques for different subproblems. It is also possible to use different ways to obtain the solution of the same subproblem in the different stages of the computation, allowing us to use an optimal approach at any particular moment and in any particular location. This is a unique feature of SAM.

- Freedom in the numerical model for each subproblem. Special boundary shapes or local behavior of the solution need a special treatment in the modeling. The decoupled subproblems allow us to localize the special treatment to the place where it is needed. Composite grids are a good example of this case.

- Freedom in the number of subproblems. This freedom will permit us to adapt this algorithm to different degrees of parallelism.

- Freedom in the coupling pattern.
  - The type of boundary value for these artificial boundaries in the decomposition.
  - Overlapping area.
  - More than one partitioning.

  Later we will show that proper use of these freedoms can yield an efficient algorithm.

A particularly interesting application of $\mathcal{SAM}$ is for parallel computations. $\mathcal{SAM}$ not only provides parallelism in the algorithm. Another advantage for an efficient implementation of $\mathcal{SAM}$ is the local communication pattern and the hiding of global information exchange. For any current state–of–the–art parallel computer the cost of the communication is always a killer of efficiency. The relatively cheap cost of the hardware provides a possibility of using a large number of processors to solve a big problem. Unfortunately, in the near future the communication cost will prevent us from fully taking advantage of fine grain parallelism in a general purpose computer architecture. The low ratio of communication verse computation, the coarser granularity and the flexibilities we mentioned above make $\mathcal{SAM}$ an attractive candidate as a parallel algorithm. This is one of the major motivations for our study.

Since this alternating method appeared, many application areas have been found. Here we present a brief historical survey of the literature:

In 1869 Schwarz [Sch69] first developed a method he called an alternating method to prove the existence of the solution of the Dirichlet problem for the Laplace equation on a union of two overlapping areas. Soon Neumann [Neu70] observed that a similar idea could be applied to the solution of the Dirichlet problem in a region that is the intersection of two other regions overlapping one another. Later Poincaré [Poi90] developed his *methode de balayage*, which is similar to Schwarz's method, Poincaré was also concerned with existence proofs rather than computation.

During the 30's many Russian mathematicians applied Schwarz's method to problems in elastostatics. They treated the solution process of $\mathcal{SAM}$ as a search

for the minimum of a variational problem. This new way of thinking provided possibilities for enlarging the applicable areas. Gorgidze [Gor34a], [Gor34b] applied S_A^M to a plane problem in the theory of elasticity. Almost at the same time, Mikhlin [Mik34] generalized this idea to a biharmonic problem. He proved the convergence of S_A^M to the solution of the second elastostatic boundary value problem. A more general proof of this method for the second boundary value problems of elasticity in three dimensions was sketched out by Sobolev[Sob36]. He reduced the consideration of convergence of the sequence of approximations to a study of convergence to the minimum of the integral of strain energy.

In the early 50's, Kantorovich and Krylov [KK58] gave a set of sufficient conditions[2] which guarantee the convergence of S_A^M. These conditions encompass most of the areas to which S_A^M can be applied.

After the 60's people started to apply S_A^M to numerical computations rather than to existence proofs or theoretical analysis. Some new algorithms such as ADI methods or Fourier series methods were the state of the art at that time, but they could only be applied to rectangular regions. S_A^M was a very natural way of applying these methods to a union of rectangular regions.

D'Jakonov [DJa62] derived some work estimates for solving Poisson's equation to a given precision on overlapping rectangular regions by S_A^M. The rectangular solutions are by the alternating-direction implicit method, or a similar method of D'Jakonov's, applied to the 5–point difference approximation.

Werner [Wer60], [Wer63] considered application of S_A^M to any linear second–order elliptic P.D.E. with boundary conditions of the third type. He proved the existence of a continuous solution and gave error bounds for a solution which satisfies the differential equation, but only approximates the boundary data. He presented some numerical results for the Laplace equation on an L–shaped region with mixed boundary conditions. The rectangular solutions are expressed as a double finite Fourier series.

We have mentioned the result by Miller[Mil65]. In the same paper he also gives work estimates for several cases. Fairweather and Mitchell [FM66] applied S_A^M to

[2]We will present them in section 2.3 of the next chapter.

a 9-point difference approximation on an JD-shaped region. They used a modified ADI method to solve the subdomain problems.

Dupont[Dup67] generalized their idea to the equation V(aVti) = p, and derived work estimates on overlapping rectangular regions.

Stoutemyer[Sto72] applied *QM* and Neumann's variant to the Laplace equation on the union and intersection of two disks.

As we mentioned earlier, applications of «5^to the composite mesh method have attracted people's attention for some time. Volkov [V6168] first presented a second order composite mesh method for the Dirichlet problem for the Laplace equation; he also used *$lM* to solve the system of linear equations.

Later, Starius [Sta77] generalized this idea to linear second order elliptic equations.

When computer technology advanced to parallel processing, the inherent parallelism in this algorithm obtained new appeal. Kang[KCSQ85] extended the variational form of *QM* to general second order elliptic P.D.E.s, and tried to apply it to parallel computations.

Glowinski, Dinh and Periaux [DGP80] [GDP80] formulated a conjugate gradient variant of *fyM*. Essentially, they reduced the problem to a minimization problem on the intersection of two overlapping regions.

Rodrigue[RS84a], [RS84b] [Rod86], and [RS85] recast *%M* in terms of numerical linear algebra so that classical techniques of acceleration could be applied. A Jacobi splitting of the modified matrix problem was studied in these papers.

Analyses and experiments have shown that the convergence rate of the plain *§lM* can be further improved. Many authors have independently found that SOR acceleration of *$M* works very efficiently. Oliger, Skamarock, Tang [OST86] also noticed that the sensitivity of the relaxation parameter is related to the overlap. Theoretical estimates of the convergence rate and choice of the best relaxation parameter for the model problem are given. In the same paper we mentioned above, Kang also proved the convergence of the SOR acceleration for the finite element method[KCSQ85]. Meier [Mei86] had also proposed a parallel SOR variant of *SAM*.

In the next chapter, an analysis of *fyM* for the continuous model problem in the fc-dimensional case is presented, a few of the important factors which govern the convergence of ^Af axe explored and then a generalization — multi-color *fyM*— and its convergence proof are shown. This generalization is mainly motivated by parallel computation. Because as we mentioned before, *QM* can be viewed as a general methodology for solving a problem, sets of sufficient conditions for convergence of *fylM* when applied to functional equations are then presented.

Following Rodrigue and Simon's idea, we propose a linear algebra analog of this model in Chapter 3. The original matrix is modified into an equivalent, loosely coupled matrix which is called the Schwarz Enhanced Matrix (or %M). Some equivalence theorems and the applicable matrices of the *SEM* are discussed. A Schwarz splitting *{SJS)* of the *SEM* is then defined. If the original matrix is an M-matrix, then *$5* is a convergent splitting.

In Chapters 2 and 3, the analysis concentrates mainly on convergence and generalizations. Another important issue is the characterization of problems for which SAM is most suitable. One particular phenomenon, exponential decay of the inverse of a sparse operator, which contributes to the success of 4j^, is investigated in Chapter 4. We found that matrices were not good structures for the study of this problem. New data structures, *template vector* and *template operator,* for a linear operator in a finite dimensional space are introduced. Some bounds for the norm of the wavefronts are shown. Particularly, a sufficient condition which yields exponential decay of the inverse is given. These results provide some guidance for a successful application of *$lM.*

Detailed analyses of the application of *SJS* to the model problem in one and higher dimensional cases are presented in Chapter 5. The spectral radii of the Jacobi iterative matrices of <9> for these cases are derived. Similar results can also be derived for higher order difference schemes or finite element methods. We show that if the overlapping area is a constant fraction of the subproblems, this algorithm has an optimal order of complexity. This means that the work needed to obtain an approximate solution which is accurate to truncation error is proportional to the number of unknowns.

In Chapter 6, based on the analysis in Chapter 5, we apply several acceleration schemes to the different hierarchical levels of this algorithm. One consequence of the analysis of *$5* in the last chapter is that many classical techniques of acceleration, especially SOR (Successive Over Relaxation) acceleration, can be app[1´]ed to this model. For the model problem, the classical analysis of SOR and many other accelerations, for instance Chebychev acceleration, can be applied to this case without any difficulty. Theoretical analyses and experiments show that the improvement in the performance is significant. The choice of the relaxation parameter in SOR acceleration is the only global information exchange in the algorithm. But if a local relaxation method is used, this global communication can be eliminated. As we show in Chapter 4, the mesh size is only involved in a higher order term of the convergence rate. Also, the low frequency errors dominate the convergence. A multi-level grid strategy is appropriate here. In the last section we present some other parallel implementation strategies to make this algorithm a powerful parallel algorithm.

Chapter 2

Convergence Analyses and Multi–Color SAM

In this chapter an application of $\mathcal{SAM}$ to the continuous model problem for elliptic partial differential equations in the k-dimensional case is given in detail. Through the analysis of this example many important factors which affect the convergence rate of $\mathcal{SAM}$ are disclosed. This analysis provides useful guidance for an efficient implementation of $\mathcal{SAM}$. Motivated by the parallel computations, a new parallel implementation of multi–color $\mathcal{SAM}$ and its convergence proof are presented in Section 2.2. Finally, two sets of very general sufficient conditions for the convergence of $\mathcal{SAM}$ in the literature are listed.

2.1 An Analysis of SAM

In this section we will apply $\mathcal{SAM}$ to the model problem on a uniform cube in k-dimensional space. Through the analysis of the solution process in this problem we can demonstrate many important characteristics of this method . In later chapters we will take advantage of these features to make $\mathcal{SAM}$ a competitive iterative algorithm.

Consider the Dirichlet problem:

$$\begin{cases} \Delta u(x_1, \cdots, x_k) &= f(x_1, \cdots, x_k), \\ u\,|_{\Gamma_\Omega} &= g(x_1, \cdots, x_k). \end{cases}$$

where Ω is a k-dimensional uniform cube for which the lengths of the edges are all equal to a. The restriction to uniform length is only for convenience of discussion; generalization to different lengths in different coordinates is straightforward.

Let us decompose this cube into two[1] overlapping subregions. Suppose the

[1] We can generalize this analysis to the case which has any finite number of subregions.

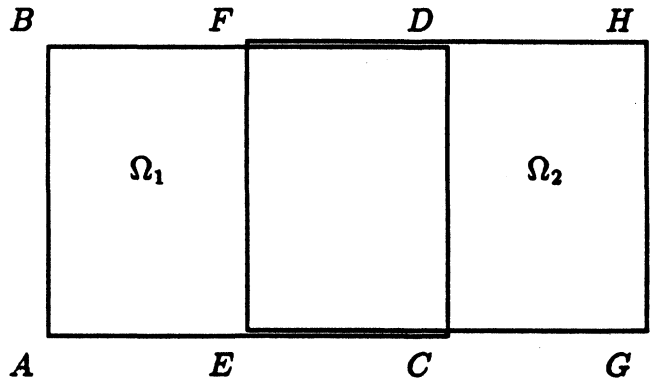


Figure 2.1: Two overlapping rectangular grids

overlapping direction is x_1. Figure 2.1 shows a two dimensional case[2]. Denote c as the width of the overlap, and b as the length of the subcube in the overlapping direction x_1. Here Γ_1' is CD and Γ_2' is EF. Apply the algorithm 1.4–1.6 to these two overlapping regions; the sequence will eventually converge to the solution on the uniform cube. Denote $u_1^{(i)}$, $u_2^{(i)}$ as the i-th approximate solutions in Ω_1 and Ω_2, u_1 and u_2 the true solutions in the two subregions. Let

$$
\begin{aligned}
\varepsilon_1^{(i)} &= u_1^{(i)} - u_1, \\
\varepsilon_2^{(i)} &= u_2^{(i)} - u_2, \\
e_2^{(0)} &= \varepsilon_2^{(0)} \mid_{\Gamma_1'} = \psi_0 - u_2 \mid_{\Gamma_1'}, \\
e_1^{(i)} &= \varepsilon_1^{(i)} \mid_{\Gamma_2'}, \\
e_2^{(i)} &= \varepsilon_2^{(i)} \mid_{\Gamma_1'} .
\end{aligned}
$$

[2]If we may imagine the lines AB, CD, EF and HG are $k-1$-dimensional uniform cubes. Then it can also represent k-dimensional case.

Then the error functions $\varepsilon_1^{(i)}$ and $\varepsilon_2^{(i)}$ will satisfy the following relations:

$$\begin{cases} \Delta\varepsilon_1^{((0))} = 0 & \mathbf{x} \in \Omega_1 \\ \varepsilon_1^{(0)}|_{\Gamma_{\Omega_1}} = \begin{cases} 0 & \mathbf{x} \in \Gamma_1 \\ e_2^{(0)} & \mathbf{x} \in \Gamma_1' \end{cases} \end{cases} \tag{2.1}$$

$$\begin{cases} \Delta\varepsilon_2^{(i)} = 0 & \mathbf{x} \in \Omega_2 \\ \varepsilon_2^{(i)}|_{\Gamma_{\Omega_2}} = \begin{cases} 0 & \mathbf{x} \in \Gamma_2 \\ e_1^{(i-1)} & \mathbf{x} \in \Gamma_2' \end{cases} \end{cases} \tag{2.2}$$

$$\begin{cases} \Delta\varepsilon_1^{(i)} = 0 & \mathbf{x} \in \Omega_1 \\ \varepsilon_1^{(i)}|_{\Gamma_{\Omega_1}} = \begin{cases} 0 & \mathbf{x} \in \Gamma_1 \\ e_2^{(i)} & \mathbf{x} \in \Gamma_1' \end{cases} \end{cases} \tag{2.3}$$

Now we are able to analyze the convergence process by Fourier analysis. Since the error functions of the approximation satisfy the Laplace equation and have boundary values 0 except at one face of the subcube, by expanding the boundary value $e_i^{(j)}$ at Γ_i' in Fourier series, we may express the error function in the whole subcube in terms of the coefficients of the boundary values. Let

$$e_2^{(0)} = \sum \alpha_{i_2 \cdots i_k}^{(0)} \sin\frac{i_2 x_2 \pi}{a} \cdots \sin\frac{i_k x_k \pi}{a}.$$

Let

$$s(i_2, \cdots, i_k) = \alpha_{i_2 \cdots i_k}^{(0)} \sin\frac{i_2 x_2 \pi}{a} \cdots \sin\frac{i_k x_k \pi}{a}$$
$$i_2, \cdots, i_k = 1, 2, \cdots$$

and

$$\tau(i_2, \cdots, i_k) = \sqrt{i_2^2 + \cdots + i_k^2}.$$

Then we have

$$\varepsilon_1^{(0)} = \sum \frac{\sinh \tau(i_2, \cdots, i_k)\frac{x_1}{a}}{\sinh \tau(i_2, \cdots, i_k)\frac{b}{a}} s(i_2, \cdots, i_k).$$

The boundary value at Γ_2' will be

$$e_1^{(0)} = \sum \frac{\sinh \tau(i_2, \cdots, i_k)\frac{(b-c)}{a}}{\sinh \tau(i_2, \cdots, i_k)\frac{b}{a}} s(i_2, \cdots, i_k).$$

By using the same reasoning, we have:

$$\cdot_2 \quad - \sum (\frac{\sinh \tau(i_2, \cdots, i_k) \frac{(b-c)}{a}}{\sinh \tau(i_2, \cdots, i_k) \frac{b}{a}})^2 s(i_2, \cdots, i_k).$$

Thus the amplification factors are:

$$\frac{\sinh \tau(i_2, \cdots, i_k) \frac{(b-c)}{a}}{\text{sinhr}(\text{i}_2, \bullet\bullet\bullet, **)\pounds} \qquad i_2, \cdots, i_k = 1, 2, \cdots \tag{2.4}$$

It is obvious that the amplification factor of the lowest frequency component will dominate the convergence. If we define the convergence factor as the ratio of the two norms of consecutive error functions, the convergence factor of this method for the model problem is:

$$\frac{\sinh \frac{\sqrt{k-1}}{a}(b-c)}{\sinh \frac{\sqrt{k-1}}{a} b}. \tag{2.5}$$

FVom 2.4 and 2.5 we can observe a few important facts of *$$i.* First the overlap ratio *c/b* has a strong influence on the convergence rate. That is: when the overlap increases the convergence factor will improve exponentially. Also, the ratio *b/a* will affect the convergence. It is clear that we should avoid overlapping in a direction for which the width of the subregion is too short in comparison with the other directions. Another important observation is that the amplification factors exponentially decay when the frequencies increase. This is a favorable feature for multilevel grid strategies. We can start the computations with a very coarse grid to obtain coarse frequency information, then reduce the grid size to obtain the higher frequency information. Moreover, 2.4 tells us that the high frequency errors do not make a significant contribution to the error inside of the region. So we might carry on the communication at some coarser grid level in order to reduce the communication cost. Another important feature of 4^, especially propitious for large scale computations, is that the higher the dimension the faster the convergence. In later chapters we will elaborate these characteristics of *$M* in depth and apply several acceleration schemes simultaneously to this model to construct a very efficient algorithm.

Using the same idea and notation we can easily analyze the case of applying Neumann boundary condition on CD and EF. Now equations 1.2 and 1.3 become :

$$\begin{cases} \Delta u_1 = f, & x \in \Omega_1, \\ u_1|_{\Gamma_1} = \psi, \\ \frac{\partial u_1}{\partial n}|_{\Gamma_1'} = \frac{\partial u_2}{\partial n}, \end{cases} \tag{2.6}$$

$$\begin{cases} \Delta u_2 = f, & x \in \Omega_2, \\ u_2|_{\Gamma_2} = \psi, \\ \frac{\partial u_2}{\partial n}|_{\Gamma_2'} = \frac{\partial u_1}{\partial n}. \end{cases} \tag{2.7}$$

The corresponding equations for the error functions are:

$$\begin{cases} \Delta \varepsilon_1^{(0)} = 0, & x \in \Omega_1, \\ \varepsilon_1^{(0)}|_{\Gamma_1} = 0, \\ \frac{\partial \varepsilon_1^{(0)}}{\partial n}|_{\Gamma_1'} = e^0, \end{cases} \tag{2.8}$$

$$\begin{cases} \Delta \varepsilon_2^{(i)} = 0, & x \in \Omega_2, \\ \varepsilon_2^{(i)}|_{\Gamma_2} = 0, \\ \frac{\partial \varepsilon_2^{(i)}}{\partial n}|_{\Gamma_2'} = \frac{\partial \varepsilon_1^{(i-1)}}{\partial n}, \end{cases} \tag{2.9}$$

$$\begin{cases} \Delta \varepsilon_1^{(i)} = 0, & x \in \Omega_1, \\ \varepsilon_1^{(i)}|_{\Gamma_1} = 0, \\ \frac{\partial \varepsilon_1^{(i)}}{\partial n}|_{\Gamma_1'} = \frac{\partial \varepsilon_1^{(i)}}{\partial n}. \end{cases} \tag{2.10}$$

(2.11)

Similarly, we may expand the error in the Neumann boundary condition problem in a Fourier series:

$$e^{(0)} = \sum \alpha_{i_2 \cdots i_k}^{(0)} \sin \frac{i_2 x_2 \pi}{a} \cdots \sin \frac{i_k x_k \pi}{a}.$$

The solution of the error in Ω_1 is as follows:

$$\varepsilon_1^{(0)} = \sum \frac{\sinh \tau(i_2, \cdots, i_k) \frac{x_1}{a}}{\cosh \tau(i_2, \cdots, i_k) \frac{b}{a}} s(i_2, \cdots, i_k).$$

By the same reasoning, we know that the convergence factor of the error on the artificial boundaries is:

$$\frac{\cosh \frac{\sqrt{k-1}}{a}(b-c)}{\cosh \frac{\sqrt{k-1}}{a}b}. \tag{2.12}$$

It is not difficult to see that the Dirichlet boundary condition on these artificial boundaries works better than the Neumann boundary condition. In Chapter 6 we will show that this conclusion is also true for discrete cases. There are also other kinds of combination of different boundary conditions can be imposed on these artificial boundaries. For the Laplace operator, we have shown that Dirichlet boundary conditions are better. This conclusion can also be derived from the decay rate of the Green's function. We will not present the details here. We could not extend this analysis to obtain a general conclusion. I conjecture that for different problems the best choice of the type of boundary conditions may vary. It is a very interesting open problem for future research.

2.2 Multi-Color SAM and Its Convergence

The SAM was originally constructed as a sequential process by Schwarz in 1869. But the inherent parallelism in this idea provides many possibilities of constructing some highly parallelized implementations. In this section a multi-color SAM for the solution of a second order linear elliptic PDE is presented. The convergence proof of this method is also given.

A simplest parallel implementation of SAM is two color or red-black SAM. It is a natural extension of the red-black SOR algorithm. The basic idea of red-black SAM is as follows: construct two sets of partitions of the solution region[3], red and black, given some initial guess on the artificial boundaries in the red set of subregions, solve the red set of subproblems independently. Using the solution of the red set for the value of the artificial boundaries in the black set, solve the subproblems in the black set independently and repeat this process. If the partitioning provides a balanced load for each processor, this implementation is a highly parallel algorithm. But from the analysis in Section 2.1, we know that the error reduction varies from

[3]They have to cover the whole solution region.

different positions in each subproblem. The new boundary values for these artificial boundaries should be taken from *fast convergent zones* of a partitioning in which this artificial boundary resides. If we only have two sets of partitions, there is only one choice for each point on these artificial boundaries. The best partitioning and choice of boundary values is very difficult to accomplish. Instead, we may plan a few sets of partitions so that each point on the artificial boundaries belongs to a fast convergent zone in at least one of the partitions. With more than two sets of partitions, this goal is easier to be fulfilled. The same idea can be applied to the elliptic operator L in any finite dimensional space. This implies that the multi-color SAM can also be used to solve any linear system of equations which has a positive definite coefficient matrix.

To simplify the notation, a description of the algorithm for a two-dimensional problem is given here. The extension to higher-dimensional problems is straightforward.

Let $L_2(\Omega)$ be a Hilbert space with respect to the inner product

$$(u, v) = \int_\Omega uv \, d\Omega,$$

and the norm

$$\| u \| = \sqrt{(u, u)},$$

where Ω is a bounded, connected open set in R^2. $C_1(\bar{\Omega})$ denotes the space of real valued continuously differentiable functions on $\bar{\Omega}$, where $\bar{\Omega} = \Omega \bigcup \Gamma$ and Γ is the boundary of Ω. Let

$$\alpha = (\alpha_1, \alpha_2), \quad | \alpha | = \alpha_1 + \alpha_2, \quad \alpha_1, \alpha_2 > 0$$
$$D^\alpha = \frac{\partial^{|\alpha|}}{\partial x^{\alpha_1} \partial y^{\alpha_2}}, \qquad u^{(\alpha)} = D^\alpha u.$$

Then the Sobolev spaces $H^1(\Omega)$ and $H^{1/2}(\Gamma)$ are defined as

$$H^1(\Omega) \equiv \{u \mid D^\alpha u \in L_2(\Omega), 0 \leq | \alpha | \leq 1\}$$
$$H^{1/2}(\Gamma) \equiv \{u \mid D^\alpha u \in L_2(\Gamma), 0 \leq | \alpha | \leq 1/2\}$$

Particularly,

$$H_0^1(\Omega) \equiv \{u \mid (u \in H^1(\Omega)) \cup (u \mid_\Gamma = 0)\}$$
$$H_g^1(\Omega) \equiv \{u \mid (u \in H^1(\Omega)) \cup (u \mid_\Gamma = g)\}$$

Consider the boundary value problem

$$\begin{cases} L(u) \equiv -[p_1 u_x]_x - [p_2 u_y]_y + qu = f, & x \in \Omega \\ u \mid_\Gamma = g, \end{cases} \tag{2.13}$$

where $p_1, p_2, q \in C_1(\bar{\Omega}), f \in L_2(\Omega), g \in H^{1/2}(\Gamma)$. It is well known that this problem (2.13) is equivalent to the minimization problem

$$I(\Omega, u) = \frac{1}{2} a_\Omega(u, u) - G_\Omega(u), \qquad u \in H_g^1(\Gamma),$$

where

$$a_\Omega(u, v) = \int_\Omega [p_1 u_x v_x + p_2 u_y v_y + quv]\, d\Omega,$$
$$G_\Omega(u) = \int_\Omega fu\, d\Omega.$$

Construct a sequence of partitions of Ω:

$$\Pi_1 : \{\Omega_1^{(1)}, \cdots, \Omega_{i_1}^{(1)}\},$$
$$\Pi_2 : \{\Omega_1^{(2)}, \cdots, \Omega_{i_2}^{(2)}\},$$
$$\vdots$$
$$\Pi_k : \{\Omega_1^{(k)}, \cdots, \Omega_{i_k}^{(k)}\}$$

such that

1. $\Omega_i^{(l)} \cap \Omega_j^{(l)} = \emptyset, \qquad$ if $i \neq j$.

2. $\forall x \in \Omega, \exists \;\; i$ and $j : x \in \Omega_i^{(j)}$.

Let $\Omega^{(l)} = \bigcup_{j=1}^{i_l} \Omega_j^{(l)}$, $i = 1, \cdots, k$. Each Π_i is referred to as a color c_l. So condition (1.) means that any two of the subregions do not overlap if they have the same color. For any $v \in H_g^1(\Omega)$ and $\Omega^{(l)}$, a subspace $\Phi(\Omega^{(l)}, v)$ is defined as follows:

$$\Phi(\Omega^{(l)}, v) = \{u \mid u \in H_g^1(\Omega) \cap (u = v, \text{ if } \; x \in \Omega - \Omega^{(l)})\}.$$

For partition Π_l, the functional $I(\Omega, u)$ can be calculated as follows:

$$I(\Omega, u) \ = \sum_{j=1}^{i_l} I(\Omega_j^{(1)}, u) + I(\Omega - \Omega^{(l)}, u) \ = I(c_l, u).$$

The multi-color SAM is then as follows:

Algorithm. 1 *Choose a initial guess* $u^{(0)} \in H_g^1(\Omega)$. *We construct a sequence* $\{u^{(i)}, c_i\}$ *such that*

$$u^{(i)} \equiv \{v \mid \inf_{w \in \Phi(\Omega^{(c_i)}, u^{(i-1)})} I(c_i, w)\}$$

and $\forall \ I > 0$ *and* $J \in [1, 2, \cdots, k]$, $\exists \ i$ *such that* $i > I$ *and* $c_i = J$.

Since $i \neq j$ implies $\Omega_i^{(l)} \cap \Omega_j^{(l)} = \emptyset$,

$$\inf_{w \in \Phi(\Omega^{(c_i)}, u^{(i-1)})} I(c_i, w) = \sum_{k=1}^{i_{c_i}} \inf_{w \in \Phi(\Omega^{(c_i)}, u^{(i-1)})} I(\Omega_k^{(c_i)}, w).$$

Each $I(\Omega_k^{(c_i)}, w)$ can be computed independently. That is where parallelism comes from.

A rigorous complete proof of convergence is wordy. Here we present an concise version of the proof.

From the construction of the algorithm, $\forall \ i > 0$

$$I(\Omega, u^{(i)}) \leq I(\Omega, u^{(i-1)}).$$

If u denotes the solution of (2.13), then

$$\begin{aligned} \| u^{(i)} - u \|_1^2 \ &\leq \frac{1}{\gamma} a_\Omega(u^{(i)} - u, u^{(i)} - u) \\ &= \frac{2}{\gamma}[I(\Omega, u(i)) - I(\Omega, u)] \\ &\leq \frac{2}{\gamma}[I(\Omega, u(0)) - I(\Omega, u)]. \end{aligned}$$

So there is at least one subsequence $u^{(i_k)}$ such that

$$\lim u^{(i_k)} = \tilde{u}. \tag{2.14}$$

From the construction of the algorithm and (2.14), $\tilde{u} \in H_g^1(\Omega)$ achieves the minimum in every subregion $\Omega_n^{(j)}$. Now, let's prove that $\tilde{u}$ actually is the solution of (2.13).

Let $\Gamma_n^{(j)}$ be the boundary of the subregion $\Omega_n^{(j)}$, which is the nth subregion in color j. $\Gamma_n^{(j)}$ is consists of two parts. First part is $\Gamma_{n1}^{(j)} = \Gamma_n^{(j)} \cap \partial\Omega$ can be empty if there is no common part in the boundaries of $\Gamma_n^{(j)}$ and $\partial\Omega$. The second part is the so-called artificial boundary $\Gamma_{n2}^{(j)}$. It is a union of the pieces which are located in other subregions (of different colors).

From the definition of $\tilde{u}$, we know that it is the solution of the following problem.

$$\begin{cases} L(u_n^{(m)}) = f, & x \in \Omega_n^{(m)}, \\ u_n^{(m)}|_{\Gamma_n^{(m)}} = \begin{cases} g_h, & x \in \partial\Omega, \\ u_n^{(m-1)}, & x \in \Gamma_{n2}^{(m)}, \end{cases} \end{cases} \tag{2.15}$$

$$n = 1, 2, \cdots, m,$$

$$m = 1, 2, \cdots, k$$

where $\Omega_n^{(m)}$ is the n^{th} subregion in color C_m and $u_n^{(m-1)}$ are the solution for partition $n_m - 1$. The coupled problems (2.2) and (2.3) are the simplest case of this problem.

First we know that the solution u of (2.13) is a solution of (2.15). It is also known that if the solution of (2.15) exists, it is unique. Therefore, we may summarize the above discussion as the following theorem:

Theorem 2.1 *For any initial guess $u^0 \in H_g^1(\Omega)$ the sequence constructed in the multicolor ZfcM algorithm converges to the solution of (2.15).*

The extension of this algorithm to a matrix problem is straightforward. We may replace the sequence of subregions $\Omega_n^{(j)}$ by a sequence of diagonal block matrices $A_{jj}^{(j)}$ of the original matrix A, which satisfy the following conditions:

1. Any two of the diagonal blocks do not overlap if they have the same color.

2. Each row of the matrix A is covered by at least by one of the blocks A_{jj}.

Then the rest part of the algorithm is the same as the continuous case. The convergence of the discrete version of multi-color *SIM* is also analogous with the continuous case.

2.3 **Sufficient** Conditions **for Convergence**

There are several proofs of convergence of *%M* [KK58], [CH62], [KCSQ85]. The most general case was given by two Russian mathematicians Kantorovich and Krylov in the 50's [KK58]. They showed that five conditions together are sufficient for convergence of *fyM* to a solution of a boundary-value problem [4]

$$\begin{cases} L(u) = f, & \mathbf{x} \in \Omega, \\ u|_{\Gamma_n} = \psi, & \text{x 6 FQ.} \end{cases} \tag{2.16}$$

These five conditions are as follows:

Uniqueness. Two solutions u and u^f which satisfy equation (2.16) in ft are bounded, have identical values on the boundary FQ (except, perhaps, at a finite set of points) and are identically equal in ft.

Monotonicity [5]. Two bounded functions u and u' which satisfy equation (2.16) in ft and have $u \geq u^f$ on FQ (except, perhaps, at a finite number of points) will satisfy $u \geq u'$ everywhere in ft.

Limit solution. The limit of any monotone and uniformly bounded sequence of solutions to equation (2.16) is also a solution of (2.16).

Maximum principle. **A** solution to (2.16) cannot have either a positive interior maximum or a negative interior minimum. For linear problems this implies the monotonicity condition.

Continuity onto the boundary. If u = / on a boundary segment except perhaps at a point P inside the segment, where / is continuous on this segment, then the solution u(Q) for Q in ft approaches /(P) as Q -+ P.

The numerical analog of *$)M* is straightforward. We can discretize the problems 1.4-1.6, and then solve them numerically. Miller[Mil65] showed that the following

[4] **As mentioned in their book, this same proof can be applied to a more general functional equation.**
[5] **For linear problems, this condition can be derived from the maximum principle.**

conditions are sufficient for convergence of the solution of the numerical $\mathcal{SAM}$ to the original continuous problem:

Existence of a continuous solution. The solution of the continuous problem 2.16 exists. This implies that the solutions of the problems 1.2 and 1.3 exist.

Existence of the discrete solutions. Solutions of the discretized problems 1.2 and 1.3 exist.

Convergent discretization. Discrete approximations of 1.2 and 1.3 are convergent to the continuous solution of 1.2 and 1.3.

Contraction mapping. There exist numbers $Q_1 \leq 1$, $Q_2 \leq 1$, such that $Q_1 Q_2 < 1$ and

$$\|u_1 - \tilde{u}_1\| \leq Q_1 \epsilon_1,$$
$$\|u_2 - \tilde{u}_2\| \leq Q_2 \epsilon_2,$$

where ϵ_1, ϵ_2 are perturbations of the boundary data on Γ_1', Γ_2'; $\tilde{u}_1, \tilde{u}_2$ are the perturbated solutions which correspond to u_1, and u_2.

For elliptic partial differential equations we can also express problem 2.16 in an equivalent variational form; then it is possible to prove that the solution sequence of the corresponding finite element method is a convergent minimization sequence. The independence between convergence and the the ordering of the solutions of these subregions can be easily shown in variational form [KCSQ85].

We can also recast the numerical analog of $\mathcal{SAM}$ as a modified matrix problem, then prove its convergence. From analysis of the linear algebra analog of $\mathcal{SAM}$ for the model problem we can obtain many new results by applying classical acceleration approaches in numerical linear algebra to this method. In Chapters 5 and 6 we will discuss these problems in detail.

Chapter 3

Schwarz Splitting

In this chapter a more general model of $\mathcal{SAM}$ for application to problems in linear algebra, Schwarz Splitting (or $\mathcal{SS}$), is presented. For a matrix equation $Ax = f$, we first introduce a Schwarz Enhanced Equation (or $\mathcal{SEE}$) $\tilde{A}\tilde{x} = \tilde{f}$. The corresponding matrix $\tilde{A}$ is called a Schwarz Enhanced Matrix (or $\mathcal{SEM}$). A necessary and sufficient condition for the equivalence of the original equation and $\mathcal{SEE}$ is shown. In Section 3.3 a few splitting matrices of $\mathcal{SEM}$ are presented. In particular, the Schwarz Splitting (or $\mathcal{SS}$) is defined. Then some relations between the eigenvalues of these splitting matrices and the corresponding splitting matrix for the original matrix are shown. If the original matrix is an M–matrix, then $\mathcal{SS}$ is a convergent splitting. The original $\mathcal{SAM}$ is equivalent to applying a block Gauss-Seidel scheme to the $\mathcal{SS}$. It is clear that other classical acceleration schemes can also be applied to this model.

3.1 Definitions

As we mentioned in our last chapter, the approach of $\mathcal{SAM}$ to a problem is to modify it to produce an equivalent enhanced problem, then to solve the new one iteratively. It is not necessary to view $\mathcal{SAM}$ only as a way of solving elliptic partial differential equations. As we mentioned in the introduction, $\mathcal{SAM}$ can be viewed as a general methodology for problem solving. A similar idea has been applied to a nonlinear problem arising in circuit simulations [Deu85]. It was suggested that the application of $\mathcal{SAM}$ to the system of ODE is promising. Here $\mathcal{SAM}$ is discussed in terms of matrix theory. In Rodrigue and Simon's paper "A generalization of the numerical Schwarz algorithm", $\mathcal{SAM}$ is first recast into numerical linear algebra. Then many results of the classical analyses in linear algebra [Var62] could be applied. This approach

provided possibilities of generalizing and improving *§lM.* We extend this thought further to a general linear system of equations in this chapter.

Consider a matrix problem:

$$Ax = /, \tag{3.1}$$

where A is an N x N nonsingular matrix, / and x are N vectors. A partitioned form of the equation (3.1) will be used in the rest of this thesis. A partitioning is defined by the integers $n_i, n_2, \cdots, n_{2k+i}$ where $n_{2i} > 0$, $n_{2i+i} \geq 0$[1] for all i, and where

$$rti + n_2 + \cdots + n_{2k+1} = N. \tag{3.2}$$

Given a set $\{n_i\}_{i=1}^{2k+1}$ which satisfies (3.2), the $(2k + 1)$ x $(2k + 1)$ partitioned form of the matrix A is then given by

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,2k+1} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,2k+1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{2k+1,1} & A_{2k+1,2} & \cdots & A_{2k+1,2k+1} \end{bmatrix}, \tag{3.3}$$

where A_{ij} is an n_i x r_j submatrix. We always assume that the unknown vector x and the known vector / in the matrix equation Ax = / are partitioned in a form consistent with A. Thus, if A is given by (3.3), then x is assumed to be partitioned as

$$x = [x_u x_2, \cdots, x_{2k+1}]^T, \tag{3.4}$$

where x_i is an n_i x 1 matrix (column vector). A *dual vector* of x

$$\tilde{x} = [x_1, x_2, x_2, x_3, x_4, x_4, x_5, \cdots, x_{2k}, x_{2k}, x_{2k+1}]^* \tag{3.5}$$

is defined such that: all even subvectors x_{2i}, i = l,***,fc are duplicated once in their places, and all odd subvectors remain the same.

A partitioned matrix can also be represented by a directed graph. Consider any 2k + 1 distinct points $P_i, P_2, \cdots, P_{2k+i}$ in the plane, which we shall call *nodes.* For every nonzero entry A_{ij} of the matrix A, we connect the node P_i to the node P_j by

[1] We will explain the reason for this partitioning pattern later.

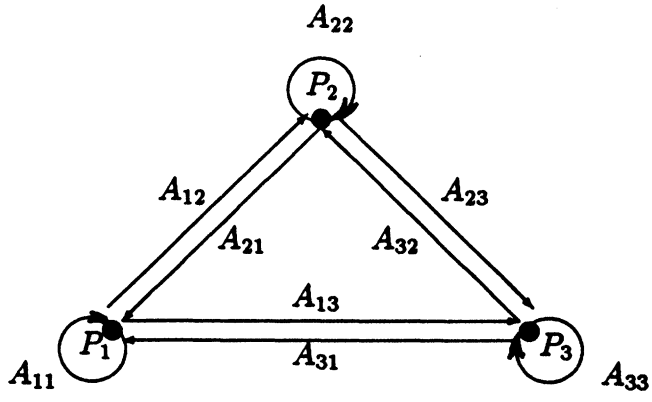


Figure 3.1: Directed graph $G(A)$.

means of a path [2] $\overrightarrow{P_iP_j}$, directed from P_i to P_j, as shown in Figure 3.1. By relating each path $\overrightarrow{P_iP_j}$ to the corresponding $A_{i,j}$, the matrix A is associated with a finite directed graph $G(A)$. As an example, a dense 3×3 partitioned matrix

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,1} & A_{3,2} & A_{3,3} \end{bmatrix}$$

has the directed graph $G(A)$ in Fig. (3.1).

If the operator $L(u)$ in equation (1.1) is a linear second order elliptic operator, then the discretized problem can be written as a matrix equation :

$$Ax = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = f. \tag{3.6}$$

The order of the unknowns is arranged so that $[x_1, x_2]$ corresponds to the unknowns in Ω_1, $[x_2, x_3]$ corresponds to the unknowns in Ω_2 and $[x_2]$ corresponds to the unknowns in Ω_{12}, which is the overlapped part of the two. The numerical SAM for the

[2] For a diagonal entry $A_{i,i} \neq 0$, the path joining the node P_i to itself is called a *loop*. For an illustration, see Figure (3.1).

above problem solves the following subproblems alternatively:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} 4^{\circ} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + \begin{bmatrix} A_{13} \\ A_{23} \end{bmatrix} x_3^{(i-1)},$$

$$\begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} 4^{\circ} \\ 4^{\circ} \end{bmatrix} = \begin{bmatrix} h \\ h \end{bmatrix} + \begin{bmatrix} A_{13} \\ A_{23} \end{bmatrix} {}^{*1} \cdot \tag{3.7}$$

It is not difficult to observe that this procedure is equivalent to a 2 x 2 block Gauss-Seidel iteration for the following matrix equation:

$$\tilde{A}\tilde{x} = \begin{bmatrix} A & \mathrm{Al}_2 & 0 & A_{13} \\ A_{21} & \mathrm{A}_{22} & 0 & A_{23} \\ A_{21} & 0 & A_{22} & A_{23} \\ A_{31} & 0 & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_2' \\ \tilde{x}_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ h \\ h \\ f_3 \end{bmatrix} = \tilde{f}. \tag{3.8}$$

From the convergence proof discussed in the last chapter we know that the procedure (3.7) will converge, the solution of equation (3.8) satisfies $\tilde{x}_2 = \tilde{x}_2'$, and $[x_1, x_2, x_3]^T$ is a solution of equation (3.6). This is to say that the dual vector of the solution of (3.6) is the solution of (3.8) and vice versa[3]. We shall call the equation (3.8) the Schwarz Enhanced Equation (or «3sP) of (3.6) and the corresponding matrix $\tilde{A}$ in (3.8) the Schwarz Enhanced Matrix (or *$EM)* of the matrix *A.* The formation of *SEM* can also be illustrated in terms of a directed graph. As we mentioned before, the original matrix is represented by the directed graph in Fig.(3.1). Let us split node 2 into a pair of dual nodes ($\tilde{P}$ ^ $\tilde{P}$2), and let the incoming path from Pi point to $\tilde{P}_2'$, and the incoming path from P3 point to $\tilde{P}_2$.

The loop path of the original node is duplicated for both dual nodes (see Fig. 3.2). This new directed graph is called the *dual graph* for the *Sfjflf* $\tilde{A}$.

This idea of forming a new equivalent problem can be generalized in two ways: we may enhance the new enhanced equation recursively; or we may partition the matrix *A* into a matrix like (3.3) and then enhance this partitioned matrix. Here

[3]Later we will prove that this conclusion can be true only when A_2^{-1} exists. For most approximations of an elliptic partial differential equation this restriction is satisfied.

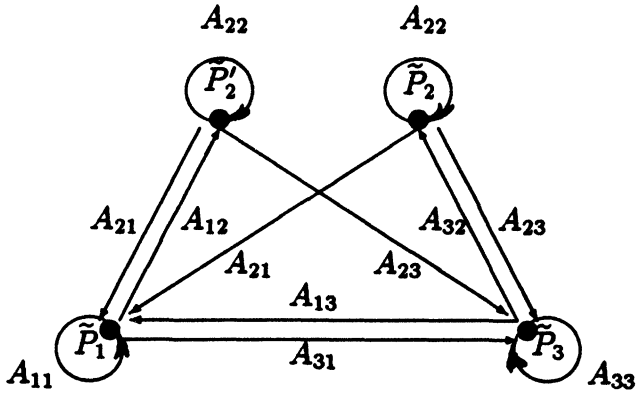


Figure 3.2: Dual graph $G(\tilde{A})$.

we only discuss the latter approach. The results of the following discussion can be easily applied to the former.

Let the matrix A in (3.1) be partitioned in the form (3.3), where $n_{2i} > 0$ and $n_{2i+1} \geq 0$ for all i. The reason for assuming $n_{2i} > 0$ is that we are going to split the node P_{2i} and splitting a null node [4] is meaningless. On the other hand, a null nonsplitting node can be used to cover the case of adjacent nodes. The case of splitting a node into three or more nodes can be covered by recursive splitting, but there is little practical reason to do that. To form the Schwarz Enhanced Matrix $\tilde{A}$, we first split every even node P_{2i} into a pair of dual nodes $(\tilde{P}_{2i}, \tilde{P}_{2i}')$, $i = 1, \cdots, k$, and copy every odd node P_{2i+1} to a new node $\tilde{P}_{2i+1}$. The even nodes P_{2i} are also called *overlap nodes*. The new nodes $\tilde{P}_i$ and $\tilde{P}_i'$ are the nodes of the directed graph $G(\tilde{A})$ for the SEM $\tilde{A}$. Then for each path $\overrightarrow{P_l P_m}$ in the original $G(A)$, we will put a corresponding path (or paths) into the dual graph $G(\tilde{A})$. The rules are listed in the following table. The far right column lists six logical expressions, of which only one can be true for each path in $G(A)$. After identifying the case for which the logical expression in the fourth column is true, we will add the path (or paths) in the third column to $G(\tilde{A})$. The corresponding entry for each path (or paths) is given in the second column. Let S_o and S_e denote the sets of odd numbers and even numbers,

[4] Here we define a node as null node if there is no path to or from this node. Equivalently, we may say that a node P_i is null if $n_i = 0$.

respectively. We have the following table:

Case	Entry	Path	Condition
Case 1	A_{ll}	$\overrightarrow{\tilde{P}_l\tilde{P}_l}$	$(l = m) \cap (l \in S_o)$
Case 2	A_{ll}	$\overrightarrow{\tilde{P}_l\tilde{P}_l}$ $\overrightarrow{\tilde{P}'_l\tilde{P}'_l}$	$(l = m) \cap (l \in S_e)$
Case 3	A_{lm}	$\overrightarrow{\tilde{P}_l\tilde{P}_m}$	$(l \neq m) \cap (l \in S_o) \cap ((m \in S_o) \cup (l > m))$
Case 4	A_{lm}	$\overrightarrow{\tilde{P}_l\tilde{P}'_m}$	$(l \neq m) \cap (l \in S_o) \cap ((m \in S_e) \cap (l < m))$
Case 5	A_{lm}	$\overrightarrow{\tilde{P}_l\tilde{P}_m}$ $\overrightarrow{\tilde{P}'_l\tilde{P}_m}$	$(l \neq m) \cap (l \in S_e) \cap ((m \in S_o) \cup (l > m))$
Case 6	A_{lm}	$\overrightarrow{\tilde{P}_l\tilde{P}_m}$ $\overrightarrow{\tilde{P}'_l\tilde{P}_m}$	$(l \neq m) \cap (l \in S_e) \cap ((m \in S_e) \cap (l < m))$

We may also interpret this table graphically. As in Fig. (3.1) and (3.2), we may lay the $2k+1$ nodes in a straight line, with the nodes in numerical order. Then we split each even node into $\tilde{P}'_{2i}$ and $\tilde{P}_{2i}$, one by one, starting from P_2. All incoming paths from the left side of P_{2i} will point to $\tilde{P}'_{2i}$; and all incoming paths from the right will point to $\tilde{P}_{2i}$. The outgoing path from P_{2i} is then split into two outgoing paths for both new nodes and will point to the same destination as before. All loops of the overlapped nodes will be duplicated for both dual nodes. After we split all even nodes, the new graph is the dual graph $G(\tilde{A})$. Notice that the dual nodes of each pair have exactly the same outgoing paths, with the exception of the two loop paths. The matrix $\tilde{A}$ constructed according to the above rules is called the *Schwarz Enhanced Matrix* (SEM) with respect to the partition (3.3), and the corresponding matrix equation

$$\tilde{A}\tilde{x} = \tilde{f} \tag{3.9}$$

is called the *Schwarz Enhanced Equation* (SEE), where $\tilde{f}$ is the dual vector of f. Here is an example of a 5×5 block matrix equation and its SEE:

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

$$\tilde{A}\tilde{x} = \left[\begin{array}{ccccccc} A_{11} & A_{12} & 0 & A_{13} & A_{14} & 0 & A_{15} \\ A_{21} & A_{22} & 0 & A_{23} & A_{24} & 0 & A_{25} \\ A_{21} & 0 & A_{22} & A_{23} & A_{24} & 0 & A_{25} \\ A_{31} & 0 & A_{32} & A_{33} & A_{34} & 0 & A_{35} \\ A_{41} & 0 & A_{42} & A_{43} & A_{44} & 0 & A_{45} \\ A_{41} & 0 & A_{42} & A_{43} & 0 & A_{44} & A_{45} \\ A_{51} & 0 & A_{52} & A_{53} & 0 & A_{54} & A_{55} \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_2' \\ x_3 \\ x_4 \\ x_4' \\ x_5 \end{array}\right] = \left[\begin{array}{c} f_1 \\ f_2 \\ f_2 \\ f_3 \\ f_4 \\ f_4 \\ f_5 \end{array}\right] = \tilde{f}.$$

If we merge each pair of dual nodes into a single node, and fold each pair of paths from the same dual pair into a single path, the resulting graph is identical to the original one. From the construction of S_{EE}, it is easy to see the following result:

Lemma 3.1 *If vector $x = (x_1, x_2, \cdots, x_{2k+1})^T$ is the solution of equation (3.1), then its dual vector $\tilde{x}$ is the solution of S_{EE} $\tilde{A}\tilde{x} = \tilde{f}$, where $\tilde{f}$ is the dual vector of f.*

The matrices $A_{2i,2i}, i = 1, \cdots, k$ are also called overlapped blocks. If two Schwarz enhanced matrices $\tilde{B}$ and $\tilde{C}$ of the same matrix A, for which the overlapped blocks are $B_{2i,2i}$ and $C_{2i,2i}, i = 1, \cdots, k$, respectively, have such a relation that each $B_{2i,2i}$ is a submatrix of the corresponding $C_{2i,2i}$, then we say $\tilde{C}$ has *more overlap* than $\tilde{B}$. This overlap is closely related to the overlap area of the solution regions for the subregions mentioned in the introduction. As we have shown, for the continuous model problem, if the amount of overlap increases, then the convergence rate will increase too. For the matrix model we have a similar result.

3.2 Equivalence Theorem

A necessary and sufficient condition for the equivalency of equation (3.1) and its S_{EE} (3.9) is given in this section.

Theorem 3.1 *Let $\lambda(A)$, $\lambda(\tilde{A})$ and $\lambda(A_{ii}), i = 1, \cdots, 2k+1$ be the sets of eigenvalues of A, $\tilde{A}$ and $A_{ii}, i = 1, \cdots, 2k+1$, respectively. Then $\lambda(\tilde{A}) \subset \lambda(A) \cup (\bigcup_{i=1}^{k} \lambda(A_{2i,2i}))$.*

Proof. Let λ be an eigenvalue of $\tilde{A}$ and

$$\tilde{x} = (\tilde{x}_1, \tilde{x}'_2, \tilde{x}_2, \cdots, \tilde{x}_{2k+1})$$

be the corresponding eigenvector. Substituting $\tilde{x}$ into the equations which correspond to the dual nodes P'_{2i} and P_{2i} , we have

$$\sum A_{2i,j}\tilde{x}_j = \lambda \tilde{x}'_{2i}, \tag{3.10}$$

$$\sum A_{2i,j}\tilde{x}_j = \lambda \tilde{x}_{2i}. \tag{3.11}$$

As we mentioned in the last section, only one term is different in the left hand sides of the two equations. Subtracting (3.10) from (3.11), we have:

$$A_{2i,2i}(\tilde{x}'_{2i} - \tilde{x}_{2i}) = \lambda(\tilde{x}'_{2i} - \tilde{x}_{2i}), \qquad i = 1, \cdots, k.$$

If $\tilde{x}'_{2i} - \tilde{x}_{2i} \neq 0$ for some i, then we have $\lambda \in \bigcup_{i=1}^{k} \lambda(A_{2i,2i})$. If $\lambda \notin \bigcup_{i=1}^{k} \lambda(A_{2i,2i})$, then $\tilde{x}_{2i}$ has to be equal to $\tilde{x}'_{2i}$ for $i = 1, \cdots, k$. Therefore, $\tilde{x}$ is a dual vector of $x = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \cdots, \tilde{x}_{2k+1})^T$, which will satisfy equation

$$Ax = \lambda x.$$

Thus $\lambda \in \lambda(A)$, which concludes the proof.

Define SLE (3.9) as *equivalent* to (3.1) if $\tilde{A}^{-1}$ exists and the solution vector $\tilde{x}$ is a dual vector of the solution x of (3.1). Similarly, we say that SLM $\tilde{A}$ is *equivalent* to matrix A if $\tilde{A}^{-1}$ exists. With this definition and the result from Theorem 3.1 we have

Theorem 3.2 *If a matrix $\tilde{A}$ is a Schwarz enhanced matrix of the nonsingular matrix A, then the following are equivalent:*

1. *Matrix $\tilde{A}$ is equivalent to matrix A.*
2. $0 \notin \bigcup_{i=1}^{k} \lambda(A_{2i,2i})$.

Proof. If $0 \notin \bigcup_{i=1}^{k} \lambda(A_{2i,2i})$, then from Theorem 3.1 we know $\tilde{A}^{-1}$ exists. Applying the strategy used in the proof of Theorem 3.1, we can show that the solution $\tilde{x}$ of $\tilde{A}\tilde{x} = \tilde{f}$ is a dual vector of the solution x of $Ax = f$.

Now we show that $0 \notin \bigcup_{i=1}^{k} \lambda(A_{2i,2i})$ is also a necessary condition. Suppose there is a j such that $0 \in \lambda(A_{2j,2j})$. We know that $A_{2j,2j}$ is singular. Let the rows of matrix $A_{2j,2j}$ be $r_i, i = 1, \cdots, n_j$. There is a constant vector $\alpha = (a_1, \cdots, a_{n_j})^T \neq 0$ such that:

$$\sum_{i=1}^{n_j} a_i r_i = 0.$$

Let the rows of the SEM $\tilde{A}$ for the dual nodes $\tilde{P}_{2j}, \tilde{P}'_{2j}$ be $\mathbf{b}_i$ and $\mathbf{c}_i, i = 1, \cdots, n_j$ respectively, where

$$\begin{aligned} \mathbf{b}_i &= (e_1, \cdots, e_{2j-1}, r_i, \quad 0, e_{2j+1}, \cdots, e_{2k+1}), \\ \mathbf{c}_i &= (e_1, \cdots, e_{2j-1}, 0, \quad r_i, e_{2j+1}, \cdots, e_{2k+1}). \end{aligned}$$

From the definition of a SEM, the only differences between rows $\mathbf{b}_i$ and $\mathbf{c}_i$ are in the positions where the r_i are located. It is easy to verify that:

$$\sum_{l=1}^{n_j} a_l \mathbf{b}_l - \sum_{l=1}^{n_j} a_l \mathbf{c}_l = 0.$$

It means that $\tilde{A}$ is singular. The proof is complete.

If a matrix is a positive definite matrix or an M-matrix[5], any principal minor of this matrix is also a positive definite matrix or M-matrix. Thus, we immediately have

Corollary 1 *Any SEM of a positive definite matrix A is equivalent to A.*

Corollary 2 *Any SEM of an M-matrix A is equivalent to A.*

3.3 Splittings of Schwarz Enhanced Matrices

From the results of the last section we know that the solution of $Ax = f$ is equivalent to the solution of $\tilde{A}\tilde{x} = \tilde{f}$. Here we will analyze the application of some classical

[5]Any $n \times n$ matrix $A = (a_{ij})$ with $a_{ij} \leq 0$ for all $i \neq j$ is an M-matrix if A is nonsingular, and $A^{-1} \geq 0$.

splitting techniques to *SEM** If we want to solve the matrix equation $Ax = /$ where A is an n x n nonsingulax matrix, we consider expressing the matrix A in the form

$$A \wedge M - N, \tag{3.12}$$

where M and N are also n x n matrices. If M is not singular, we say that this expression represents a *splitting* of the matrix A, and associated with this splitting is an iterative method

$$Mx^{\wedge l)} = NxW + f. \tag{3.13}$$

Most important iterative methods can be described from this point of view. It is also called a *linear stationary method of first degree.* The matrix $M\sim^{l}N$ is called the *iterative matrix* of this splitting. The convergence behavior of this splitting is decided by A(M~[1]iV), more specifically, by the spectral radius[6] of the iterative matrix $M\sim^{l}N$ and the distribution of the eigenvalues of this matrix. Particularly, we call a splittings a *convergent splitting* if $p(M\sim^{l}N) < 1$ holds. If the diagonal entries of the matrix A = (ay) are all nonzero, and we express the matrix A as the matrix sum

$$A = D - L - 17,$$

where $D = diag(a\backslash_{9}i, 03,2,' \bullet \bullet \wedge n,n)$ and L and U are , respectively, strictly lower and upper triangular n x n matrices, then the following choices

$$\begin{aligned} M_{PJ} &= D\text{-}_{J} & N_{PJ} &= L + U_{y} \\ M_{PG} &= D\text{-}L; & N_{PO} &= U \end{aligned}$$

give the point Jacobi and point Gauss-Seidel splitting, respectively.

Let $\tilde{A} = \tilde{M}pj - \tilde{N}pj$ be the point Jacobi splitting of *SEM,* and $A\wedge i$ = A — Li — Ui, $i = 1, \bullet \bullet \bullet, 2k+1$ where At_{b} is the diagonal block in (3.3). The eigenvalues of the iterative matrix $\tilde{M}\bar{p}\}\tilde{N}pj$ and the eigenvalues of the point Jacobi iterative matrix of A have the following relation:

[6]The *spectral radius p(A)* of a matrix A is defined as

$$p(A) = \max \text{ I } A_{r} \text{ |}$$

Theorem 3.3

$$\lambda(\widetilde{M}_{PJ}^{-1}\widetilde{N}_{PJ}) \subseteq \lambda(M_{PJ}^{-1}N_{PJ}) \cup [\bigcup_{i=1}^{k} \lambda(D_{2i}^{-1}(L_{2i}+U_{2i}))].$$

Proof. Notice that the iterative matrix $\widetilde{M}_{PJ}^{-1}\widetilde{N}_{PJ}$ is a S_E^M of the matrix $M_{PJ}^{-1}N_{PJ}$ with respect to the same partition of A in (3.3). Apply theorem 3.1 to this point Jacobi iterative matrix, the proof is complete.

This theorem shows that the point Jacobi splitting of S_E^M does not change the performance of the point Jacobi splitting of the original matrix.

For the block Jacobi iterative method in which the matrix A is partitioned into the form of (3.3), a conclusion similar to theorem 3.3 can be derived. Let $\tilde{A}$ be the S_E^M of the matrix A, the block Jacobi splitting for this partitioning of A be

$$A = M_{BJ} - N_{BJ},$$

and the corresponding block Jacobi splitting of the S_E^M be

$$\tilde{A} = \widetilde{M}_{BJ} - \widetilde{N}_{BJ},$$

where $M_{BJ} = diag(A_{1,1}, \cdots, A_{2k+1,2k+1})$ and $\widetilde{M}_{BJ}$ is S_E^M of the matrix M_{BJ}. Then for this special block Jacobi splitting we have

Theorem 3.4

$$\lambda(\widetilde{M}_{BJ}^{-1}\widetilde{N}_{BJ}) = \lambda(M_{BJ}^{-1}N_{BJ}).$$

Proof. As in the case of the point Jacobi iterative matrix, this block Jacobi iterative matrix of $\tilde{A}$ is the S_E^M of the block iterative matrix of A. Since the diagonal blocks of both iterative matrices are zero, the second term on the right hand side in Theorem 3.1 vanishes; and thus the equality holds. If we split the diagonal blocks as $A_{i,i} = M_i - N_i$ and let $M'_{BJ} = diag(M_1, \cdots, M_{2k+1})$, then we have a more general result than Theorem 3.4:

Corollary 3

$$\lambda(\widetilde{M'_{BJ}}^{x\wedge} N'_{BJ}) \subseteq KM'BJ\wedge N'BJ) \cup (\bigcup_{i=1}^{k} A(M\wedge''^{1}AT\wedge.)).$$

Theorem 3.4 tells us that this particular block Jacobi splitting of a *SEM* does not improve the convergence rate when we compare it with the corresponding splitting of the original matrix *A*.

For the point Gauss-Seidel splitting of the dual matrices we also have a similar theorem. Let $\tilde{A} = \widetilde{MPG} - \widetilde{NPG}$ be the point Gauss-Seidel splitting of a *S]d_m* it is not difficult to see that $\widetilde{MPG}$ and $\widetilde{NPG}$ are the *SEM* of the matrices *MPG* and *NPG*, respectively. The following result can easily be obtained from the proof of theorem 3.1.

Theorem 3.5

$$\lambda(\widetilde{M}_{PG}^{-1}\widetilde{N}_{PG}) \subseteq \lambda(Mp\Sigma,Np_G) \cup (\bigcup_{i=1} A((D_{2t} - \cdots - Is*)^{-1}!/*)).$$

This result shows that if we relax each subproblem only once, the convergence factor is independent of the overlap[7]. It is interesting that this non-positive result has a very useful application. When we use the multigrid method in a composite grid environment, one important question is how the overlap will effect the convergence. There are some experiments (see [ST82]) which show that the amount of overlap does not affect the convergence, and thus we can reduce the overlap to a minimum[8] in order to cut down the cost of each sweep. This theorem gives us an explanation. Another extreme is *tyM*^ in which case the relaxation is carried out to convergence of the subproblem. Then the convergence factor is exponentially related to the overlap (see introduction). We might expect that the effect of the overlap on the convergence will increase when we increase the relaxation sweeps in each subproblem.

So far the splitting techniques we have discussed are not very promising. But this picture can be changed. Consider a new partitioning of *$<M*, let

$$\tilde{A} = \widetilde{M}_S - \widetilde{N}_S, \tag{3.14}$$

[7] If we relax the subproblems more than once then the conclusion is not valid

[8] There are other factors which must also be considered. For example, the amount of overlap must be sufficient to ensure the accuracy of the interpolation.

where $\widetilde{M}s = diag(S\backslash, S_2, \cdots, S^*)$, and

$$S_1 = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,1} \end{bmatrix},$$

$$Si = \begin{bmatrix} A_{2i,2i} & A_{2i,2i+1} & A_{2i,2i+2} \\ A_{2i+1,2i} & A_{2i+1,2i+1} & A_{2i+1,2i+2} \\ ^{\wedge}21+2,21 & A_{2i+2,2i+1} & A_{2i+2,2i+2} \end{bmatrix} \quad i — 9 \ldots h —. 1 \bullet$$

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

We define the splitting (3.14) to be a *Schwarz Splitting* (\$>). We should always relate a Schwarz splitting to the corresponding partition. A different partitioning will lead to a different *\$. From this definition we know that a *\$5* is essentially a block Jacobi splitting for a particular partition of *SEM,* and ^f i_s the Gauss-Seidel splitting which corresponds to this partition.

Very often problems in the biological, physical and social sciences can be reduced to problems involving matrices which have some special structure. One common situation is where the matrix is an *M*-matrix. As we mentioned in last section, any *\$EMO{* an M-matrix is equivalent to the original matrix. Now we have the following result:

Theorem 3.6 *A Schwarz splitting of any Schwarz enhanced matrix $\tilde{A}$ is a convergent splitting if A is an M-matrix.*

Proof. We define a splitting $A = M — N$ as a *regular splitting* of A if M is nonsingular with $M \geq 0$, and $N \geq 0$. A well known result for the regular splitting is that if $A^{\sim^{l}} \geq 0$, any regular splitting of the matrix A is a convergent splitting[Var62]. It is clear that if A is an M-matrix then the \$ is a convergent splitting. By the comparison theorem for Af-matrices[9], we can also derive a comparison relation between the splittings we discussed above.

»Let $A = Mi - N_x$ as A/a - N^* be two regular splittings of A, when $A^{"1} \geq 0$. If $N_x > N_2$ then the spectral radii of the matrices $M^{\sim X}N\backslash$ and $M^{\wedge}Ni$ have the following relation:

$$\rho(M_1^{-1}N_1) > \rho(M_2^{-1}N_2)$$

[Var62]

Theorem 3.7

$$\rho(\widetilde{M}_S^{-1}\widetilde{N}_S) < \rho(\widetilde{M'_{BJ}}^{-1}\widetilde{N}'_{BJ}) < \tilde{\rho}(\widetilde{M}_{BJ}^{-1}\widetilde{N}_{BJ}) < \tilde{\rho}(\widetilde{M}_{PJ}^{-1}\widetilde{N}_{PJ}).$$

Proof. **From the construction we have**

$$\widetilde{N}_{PJ} > \widetilde{N}_{BJ} > \widetilde{N'}_{Bj} > \widetilde{N}_S.$$

Application of the comparison theorem concludes the proof.

From this result, we know that $S is the best splitting among these splittings. In Chapter 5 we will derive some quantitative results for $S of the model problem of elliptic PDE's.

Chapter 4

Template Operators and Exponential Decay

In the last two chapters, we have shown that $\mathcal{SAM}$ or $\mathcal{S}$ can be applied to large classes of problems, but we have not addressed the issue of how to recognize the problems for which $\mathcal{SAM}$ is most suitable. Now we will reexamine the analysis of the model problem in the Chapter 2 from a different point of view. More specifically, we will study a particular behavior of the inverse of the same operator, the *exponential decay* phenomenon. First, the Green's functions for the model problem in 1–, 2– and 3–dimensional solution space are discussed. The relation between the decay of the Green's function and the convergence speed of $\mathcal{SAM}$ is studied. Then in Section 2 the decay of the "discrete Green's function" of a matrix is studied. Specifically, the exponential decay of a banded matrix is examined in detail. We have found that the matrix is not a good structure to study this problem. In Section 3, a new structure, *template operator,* for a linear operator in a finite dimensional space is developed. In the last Section, the concepts of influencing and influenced wavefronts are introduced. Then some estimates of the norm of the wavefront are presented. These results provide a theoretical basis for determining when these Schwarz techniques can be used successfully.

4.1 A Key to the Success of SAM

In Chapter 2, the analysis of $\mathcal{SAM}$ for the model problem shows that if the overlap increases the convergence factor of $\mathcal{SAM}$ improves exponentially. Moreover, the higher the spatial dimension, the bigger the improvement. If we combine these analyses with some other techniques, $\mathcal{SAM}$ or $\mathcal{S}$ can be developed to be as competitive as other powerful methods. One of the key facts which makes $\mathcal{SAM}$ become

an optimal iterative method is the exponential relation between the overlap and the convergence factor. More specifically, when the overlapping area increases the convergence factor decreases exponentially. For a Poisson equation, we may also use the decay of the Green's functions to explain this result more intuitively. As we know, the solution $u(P)$ for the model problem with homogeneous boundary condition can be expressed by the corresponding Green's function as

$$u(P) = \int_{\Omega} G(P,Q)f(Q)dQ.$$

The influence of the forcing function $f(Q)$ on the solution $u(P)$ is decided by the value of the Green's function at (P,Q). The Green's functions for model problems in one–, two– and three–dimensional solution space are as follows [1]:

1. One dimensional problem $(0 \leq x \leq 1)$:

$$G(x,\xi) = \begin{cases} (1-\xi)x & for \quad x \leq \xi, \\ (1-x)\xi & for \quad x \geq \xi. \end{cases}$$

2. Two dimensional problem:

$$G(x,y,\xi,\eta) = \frac{1}{2\pi} \ln \frac{1}{\sqrt{(x-\xi)^2 + (y-\eta)^2}}$$

3. Three dimensional problem:

$$G(x,y,z,\xi,\eta,\zeta) = \frac{1}{4\pi\sqrt{(x-\xi)^2 + (y-\eta)^2 + (z-\zeta)^2}}.$$

Let P represent $x, (x,y)$, or (x,y,z) and Q represent $\xi, (\xi,\eta)$, or (ξ,η,ζ). We may observe that when the distance between P and Q increases, the influence of Q on the solution at P decreases. If the overlap is increased, the artificial boundaries are moved away from the boundaries of the subregions. Consequently, the contributions from the error on the boundaries of these subregions diminish *exponentially* with the increasing distance. This observation is not only true for the

[1] Here we list the Green's functions for 2– and 3–dimensional Poisson equations in infinite domains which are easier to explain. The Green's functions for finite domains have a similar decay but are more complicated.

Poisson equation. Actually, it appears in many physical processes. The decay of the Green's function is just a mathematical description of a common physical phenomenon: that the influence between two points will weaken if the distance between them increases. This suggests that $\mathcal{AM}$ and $\mathcal{S}$ can be applied to many important applications successfully. Late on, we will explore this issue mathematically.

Another observation which can be obtained from these Green's functions is that the higher the spatial dimension the faster the decrease of the influence!

The inverses of the matrices A associated with Poisson's equation with Dirichlet boundary value conditions give "discrete Green's functions". A^{-1} should be a good approximation to the Green's function (See Birkhoff's book "*Numerical Solution of Elliptic Problems*"). Thus a similar decay behavior should be true for these inverses[2]. This observation motivates us to seek more kinds of operators for which the inverse has such a decay property.

4.2 Exponential Decay and Banded Matrices

The exponential decay of the off–diagonal elements of the inverse of a diagonally dominant tridiagonal matrix was observed decades ago[Ker70]. There were several papers which discussed the topic of the exponential decay of the inverse of a banded matrix[Dem77], [dB80], etc.. In summary, an estimate of the form

$$| \alpha_{i,j} | \leq C\gamma^{|i-j|} \tag{4.1}$$

was given in these papers, where $\alpha_{i,j}$ is the element of the inverse of a banded matrix. The claims in these paper are somewhat misleading. The first issue is: can we guarantee a decay from (4.1)? The answer is **no**! Without any further conditions on the linear operator the above estimate provides us with no useful information. There are two pitfalls in this statement. First is the "constant" C. In the following example we will show that C can be so big that an exponential increase may happen! The second pitfall is the decay term $\gamma^{|i-j|}$. Even though we do have $\gamma < 1$ here, γ is a function of the order of the matrix n in question. For example, $\gamma \cong 1/(1+n^{-2})$

[2]In next chapter, we will prove that these conclusions can be derived for discrete model problems

for a matrix derived from the model problem. So we will have $\gamma^n \cong e^{-2}$, a nonzero limit, as $n \to \infty$. These two factors lead to a bound which is so weak that virtually anything can happen. Here are some counter examples. The first is:

$$A = \begin{bmatrix} 1 & & & & & & & & \\ -2 & 1 & & & & & & & \\ & -2 & 1 & & & & 0 & & \\ & & -2 & 1 & & & & & \\ & & & \ddots & \ddots & & & & \\ & 0 & & & & \ddots & \ddots & & \\ & & & & & & -2 & 1 & \\ & & & & & & & -2 & 1 \end{bmatrix}_{n \times n}.$$

It has an inverse:

$$A^{-1} = \begin{bmatrix} 1 & & & & & & & & \\ 2 & 1 & & & & & & & \\ 4 & 2 & 1 & & & & 0 & & \\ 8 & 4 & 2 & 1 & & & & & \\ & & & \ddots & \ddots & & & & \\ \vdots & \vdots & & & & & & & \\ & & & & & \ddots & \ddots & & \\ 2^{n-1} & \cdots & & & & & 2 & 1 & \\ 2^n & \cdots & & & & & & 2 & 1 \end{bmatrix}_{n \times n}.$$

The off–diagonal elements of the inverse actually increase exponentially. People may argue that this matrix is not stable [3]. Imposing a stability condition only solves the problem of the big constant C. The second problem still exists. The following example is derived from a boundary value problem for the one–dimensional model

[3] Here we need to consider A to be a family of matrices with respect to the size n, or the mesh size h, if we want to discuss the stability. A commonly used definition of stability in this context is that the norms of the inverses for this family of matrices are bounded by a constant which is independent of n or h.

problem. This matrix is symmetric positive definite, but its inverse still has a growth away from the main diagonal along some rows and columns.

$$A = \begin{bmatrix} 0.92 & -1 & & & & & & & & \\ -1 & 2 & -1 & & & & & & & \\ & -1 & 2 & -1 & & & & & & \\ & & \ddots & \ddots & & & & & & \\ & & & \ddots & \ddots & & & & & \\ & & & & & & & & & \\ & & 0 & & & & -1 & 2 & -1 & \\ & & & & & & & -1 & 2 & \end{bmatrix}_{10\times 10},$$

$$A^{-1} = \begin{bmatrix} 50 & 45 & 40 & 35 & 30 & 25 & 20 & 15 & 10 & 5 \\ 45 & 41.4 & 36.8 & 32.2 & 27.6 & 23 & 18.4 & 13.8 & 9.2 & 4.6 \\ 40 & 36.8 & 33.6 & 29.4 & 25.2 & 21 & 16.8 & 12.6 & 8.4 & 4.2 \\ 35 & 32.2 & 29.4 & 26.6 & 22.8 & 19 & 15.2 & 11.4 & 7.6 & 3.8 \\ 30 & 27.6 & 25.2 & 22.8 & 20.4 & 17 & 13.6 & 10.2 & 6.8 & 3.4 \\ 25 & 23 & 21 & 19 & 17 & 15 & 12 & 9 & 6 & 3 \\ 20 & 18.4 & 16.8 & 15.2 & 13.6 & 12 & 10.4 & 7.8 & 5.2 & 2.4 \\ 15 & 13.8 & 12.6 & 11.4 & 10.2 & 9 & 7.8 & 6.6 & 4.4 & 2.2 \\ 10 & 9.2 & 8.4 & 7.6 & 6.8 & 6 & 5.2 & 4.4 & 3.6 & 1.8 \\ 5 & 4.6 & 4.2 & 3.8 & 3.4 & 3 & 2.4 & 2.2 & 1.8 & 1.4 \end{bmatrix}.$$

Both examples have shown that band structure does not guarantee decay in the inverse. Here is another example. The following matrix is **not banded(!),** but its inverse shows an interesting decay. If we arrange the elements of any row or column of this matrix on a circle with equal spaces and think of the original diagonal element as a central element, the elements on this circle decay away from this central element. This matrix is derived from a periodic boundary value problem. All these examples show that band structure is not a good predictor for the decay of its inverse. Furthermore, band structure is also related to the ordering of the matrix in question. If we reorder a banded matrix as a random sparse matrix, the decay still exists, but it can not be described in terms of distance from the main diagonal.

This suggests that the decay is essentially caused by a locality or compactness of an operator which is independent of the bandness of the matrix or the ordering of the variables.

$$A = \begin{bmatrix} 2.1 & -1 & & & & & & & & -1 \\ -1 & 2.1 & -1 & & & & & & & \\ & -1 & 2.1 & -1 & & & 0 & & & \\ & & & \ddots & \ddots & & & & & \\ & & & & \ddots & \ddots & & & & \\ & & & 0 & & & -1 & 2.1 & -1 & \\ -1 & & & & & & & -1 & 2.1 & \end{bmatrix}_{10\times 10},$$

$$A^{-1} = \begin{bmatrix} 1.702 & 1.286 & 1.001 & .8143 & .7096 & .6758 & .7096 & .8143 & 1.001 & 1.287 \\ 1.287 & 1.702 & 1.287 & 1.001 & .8143 & .7096 & .6758 & .7096 & .8143 & 1.001 \\ 1.001 & 1.287 & 1.702 & 1.287 & 1.001 & .8143 & .7096 & .6758 & .7096 & .8143 \\ .8143 & 1.001 & 1.287 & 1.702 & 1.287 & 1.001 & .8143 & .7096 & .6758 & .7096 \\ .7096 & .8143 & 1.001 & 1.287 & 1.702 & 1.287 & 1.001 & .8143 & .7096 & .6758 \\ .6758 & .7096 & .8143 & 1.001 & 1.287 & 1.702 & 1.287 & 1.001 & .8143 & .7096 \\ .7096 & .6758 & .7096 & .8143 & 1.001 & 1.287 & 1.702 & 1.287 & 1.001 & .8143 \\ .8143 & .7096 & .6758 & .7096 & .8143 & 1.001 & 1.287 & 1.702 & 1.287 & 1.001 \\ 1.001 & .8143 & .7096 & .6758 & .7096 & .8143 & 1.001 & 1.287 & 1.702 & 1.287 \\ 1.287 & 1.001 & .8143 & .7096 & .6758 & .7096 & .8143 & 1.001 & 1.287 & 1.702 \end{bmatrix}.$$

Particularly, for operators in a high dimensional solution space or for operators which are derived from the finite element method, the concept of band can no longer characterize the locality of the operator.

The second issue is how to define a concept of distance between nodes which is meaningfully related to their influence upon each other. Essentially, the purpose of introducing the concept of exponential decay is to characterize the decreasing influence as the distance between two nodes[4] increases. Using a *matrix* data structure, the exponential decay is characterized by the decrease of the off-diagonal elements. For one–dimensional problems, this decay gives a good characterization of decreasing influence. But for higher-dimensional problems, this characterization is not

[4]Here we adopt the terminology *node* and *distance* from the graphical representation of a matrix.

adequate. For example, consider the Dirichlet problem of a Helmholtz equation

$$\begin{aligned} \triangle u - \alpha u &= f, \qquad && x \in [0,1] \times [0,1], \\ u\,|_\Gamma &= g, \end{aligned}$$

where $\alpha = 0.00278$. Let the grid size be 1/6. Using an nine-point stencil, we can construct a 36×36 diagonally dominant, banded, positive definite matrix. We list the first column of the inverse matrix in the following table. Each item in the table has been multiplied by a factor of 10^4. The superscript of each number in the table is the row number of the elements in the inverse.

568 ^{1}	162 ^{2}	59 ^{3}	26 ^{4}	12 ^{5}	5 ^{6}
162 ^{7}	129 ^{8}	69 ^{9}	36 ^{10}	18 ^{11}	8 ^{12}
59 ^{13}	69 ^{14}	52 ^{15}	33 ^{16}	18 ^{17}	8 ^{18}
26 ^{19}	36 ^{20}	33 ^{21}	24 ^{22}	15 ^{23}	7 ^{24}
12 ^{25}	18 ^{26}	18 ^{27}	15 ^{28}	10 ^{29}	5 ^{30}
5 ^{31}	8 ^{32}	8 ^{33}	7 ^{34}	5 ^{35}	2 ^{36}

It is easy to see that the off diagonal elements decrease in an oscillatory manner, since the enforced ordering has destroyed the topological relationship among these variables. Measuring the distance between two nodes by the difference of the row and column numbers here is not suitable to characterize the influence between them. But if we imagine a center in the upper left corner of this table, the elements decay monotonically and exponentially in a wavefront form.

After a careful study of these issues and counter–examples, we have found that the abstract data structures *vector* and *matrix* prevent us from seeing important features in many physical problems. Actually, in a recent paper, Demko [DMS84] had noticed this limitation of the matrix structure. A linear operator in finite dimensional space is often a discrete approximation of a continuous operator for some particular application. Instead of solving the original problem in the entire solution region Ω (say in R^k, where k usually is 1, 2 or 3), we choose only a finite number of nodes (or points) $o_1, o_2, \cdots, o_n$ in Ω and try to find the solutions on these

nodes. A discrete approximation of the original problem

$$\boldsymbol{Ax = b}$$

is then formed, where $x = \{x_i\}$ is the set of unknowns defined on the set of nodes $O = \{o\{\}$ and A is the discrete approximation of the original continuous operator. On the one hand, $X\{$ is a component of an n-dimensional vector. On the other, each Xi is also associated with a node o_i of the solution region $£l$ in R^* or some other solution space. In an abstract vector space, the elements x_i are given a forced linear ordering which in general cannot adequately represent their positions in the solution space. The corresponding data structure for the linear operator in this space is represented by a rectangular matrix. Again, the positions of the coefficients in any one of the rows or columns have little relation with the positions in the solution space. Generally speaking, these abstract data structures have successfully represented the topology of the problems in one spatial dimension and are a good theoretical tool for many analyses. But for operators which are derived from higher-dimensional problems, the enforced linear ordering of the unknowns in the matrix structure has destroyed the proximity relations of the variables and the compactness of the operator. This is an example of how our thinking and theory has been influenced by *sequential filters* which have disfigured many physical features in a particular application. Since sequential arithmetic and two dimensional scratch paper were the means to study mathematics a few hundreds years ago, it is not surprising that people proposed the matrix data structure for the linear operator at that time. Now, the *parallel age* has come. It is the time to free ourselves from this filter.

4.3 Template Operators

In the course of this study, J. Oliger suggested finding a new structure which would preserve the topological structure of the original problem. The discussions led to a new vector space — *Template vector space T£* and a new structure of the linear operator — *Template operator.* Most of the linear operators in finite-dimensional space are derived from discrete approximations of continuous operators. The main

idea of these new structures is to maintain the primary topological structure of the original problem in the discrete approximation. In this way, many characteristics of the original continuous problem can also be easily seen in the finite dimensional approximation.

Let oi,O2,---,o_n be *n* nodes in a solution region *Q,* which usually resides in R*, *k* = 1,2 or 3 [5]. These nodes usually are the positions on which the discrete approximations of a continuous problem are sought. Let *O* denote the set of all nodes o_i.

Definition. 1 ***A*** **template**

is a topological structure of the set O in which all nodes oq* maintain the same proximity relation with each other as they have in the solution region Q.

Intuitively, *T* is the pattern of the distribution of the set *O.* For example, there are four templates in Fig. 4.1. They all have the same number of nodes, but they are associated with four different topological structures. The first three come from R^1, R^2 and R^3 respectively. The second and fourth templates are both from $i2^2$, but they have different topological relationships among the nodes. We consider them to be different templates.

Given a template T, construct n Cartesian products of R* and o,-, *S%* = R^J x o_i, í = 1,2, • • •, n, where R* is an ^-dimensional vector space. Now define set

$$T_n^{\#} = Si \times 5_2 \times \bullet\bullet\bullet \times 5_n.$$

Each element in *TJ* consists of n ordered pairs[6]:

$$(< x_1, o_1 >, < x_2, o_2 >, \cdots, < x_n, o_n >).$$

If there is no confusion, we may also abbreviate the notation as

$$\{x_{o_1}, x_{o_2}, \cdots, x_{o_n}\},$$

[5] **Actually, *Q* can exist in any space.**

[6] **To simplify the typesetting, we will not express all the following concepts by their real topological picture, but by a linear array of n ordered pairs.**

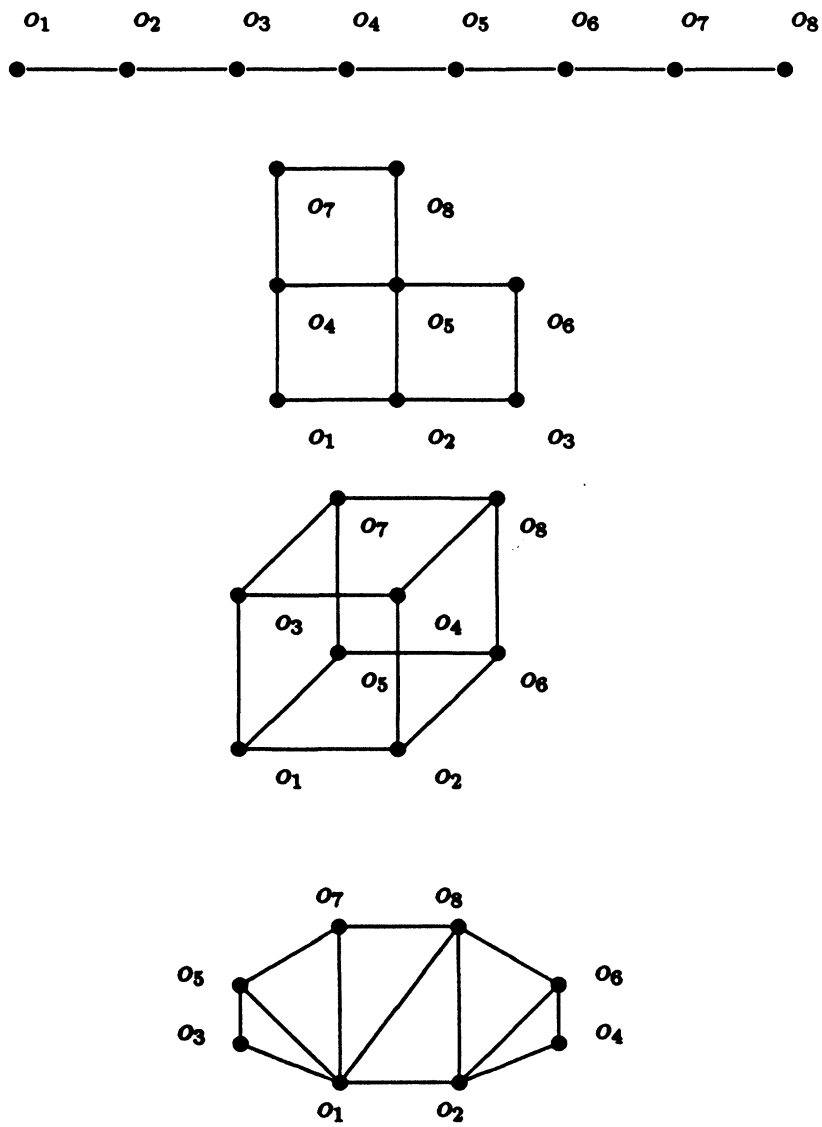


Figure 4.1: Four different templates

where x_i or x_{o_i} are s-dimensional vectors. In physics, x_i or x_{o_i} can be interpreted as the state variables for node o_i. Each x_i or x_{o_i} is defined on the node o_i on the template T.

A **template vector space** which we may abbreviate as **template space** over R^s is the set $\mathcal{T}_n^s$ with operations, addition and scalar multiplication, which are defined as follows: let

$$\begin{aligned} x &= (< x_1, o_1 >, < x_2, o_2 >, \cdots, < x_n, o_n >), \\ &= \{x_{o_1}, x_{o_2}, \cdots, x_{o_n}\}, \end{aligned}$$

$$\begin{aligned} y &= (< y_1, o_1 >, < y_2, o_2 >, \cdots, < y_n, o_n >), \\ &= \{y_{o_1}, y_{o_2}, \cdots, y_{o_n}\}, \end{aligned}$$

and

$$\begin{aligned} x + y &= (< x_1 + y_1, o_1 >, \cdots, < x_n + y_n, o_n >), \\ &= \{x_{o_1} + y_{o_1}, \cdots, x_{o_n} + y_{o_n}\}, \end{aligned}$$

$$\begin{aligned} \alpha x &= (< \alpha x_1, o_1 >, < \alpha x_2, o_2 >, \cdots, < \alpha x_n, o_n >), \\ &= \{\alpha x_{o_1}, \alpha x_{o_2}, \cdots, \alpha x_{o_n}\}. \end{aligned}$$

Under these definitions, $\mathcal{T}_n^s$ is a linear space.

Each element $x \in \mathcal{T}_n^s$ is called a **template vector.**

For example, Fig. 4.2. presents four template vectors which are associated with the corresponding template in Fig. 4.1.

Let $\sum_{o_i \in \mathcal{O}}$ denote the summation over all nodes $o_i \in \mathcal{O}$. Define the operation of scalar product of two template vectors x and y as follows:

$$(x, y)_T = \sum_{o_i \in \mathcal{O}} (x_{o_i}, y_{o_i}),$$

where (x_{o_i}, y_{o_i}) is the scalar product of two s-dimensional vectors. We may intuitively think of this operation as matching the two template vectors together,

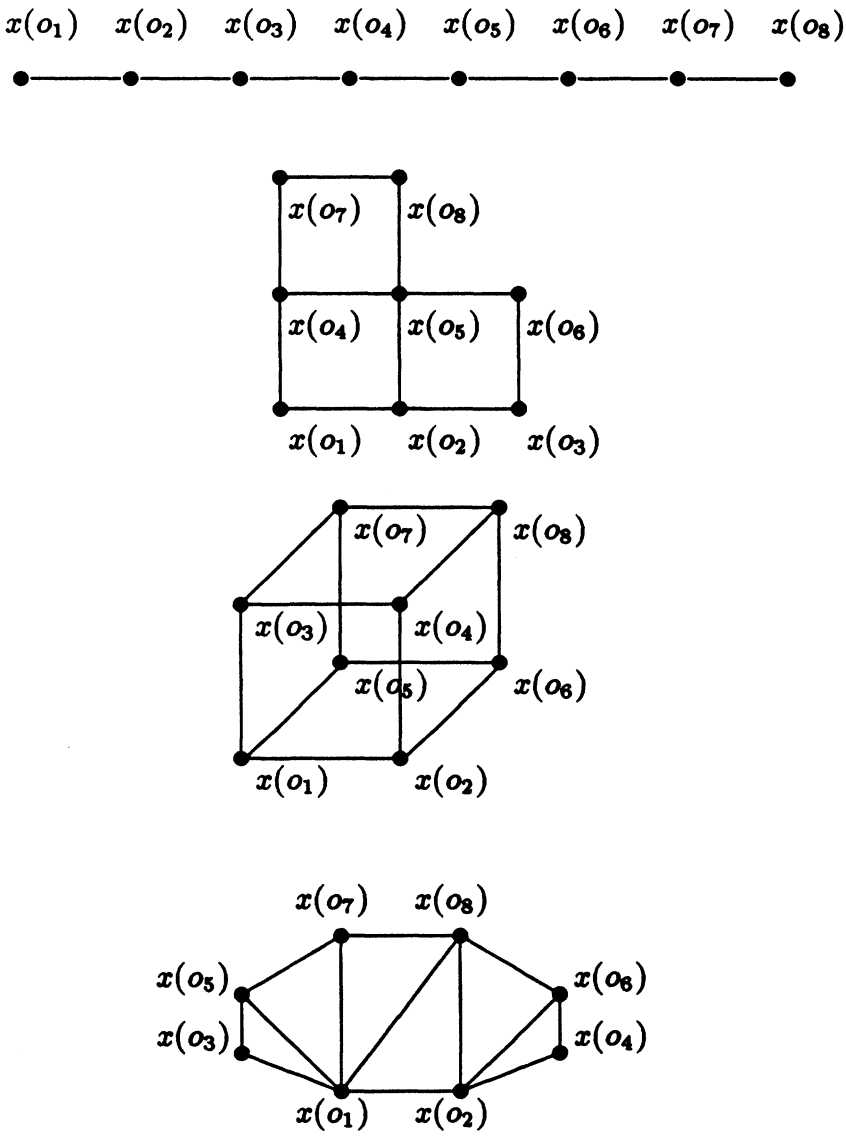


Figure 4.2: Four different template vectors

forming the scalar products for the matching pairs, and then summing the products. By the length (or norm) of a template vector in this space we mean the quantity

$$\text{II } x \text{ }|| = \overline{y/(x,x)_T}.$$

It is easy to see that T_n* is a Euclidian space under this operation.

In order to simplify the notation, the following discussion will assume s = 1. Let T_n denote T_n^l. Thus there is only one state variable on each node. In the appendix, we will remove this restriction and generalize to other spaces.

Given a template vector space T_n, a *template operator space* over T_n can be introduced as follows: let

be the template of T_n. Construct n Cartesian products

$$Q_i = T_n \times o_i, i = 1, \cdots, n.$$

Let

$$C = \text{Qi x } Q_2 \text{ x } .. \bullet \text{ x } Q_n.$$

Each element L € C consists of n ordered pairs

$$L = [< R_u o_x >, < R_2, o_2 >, \cdots, < R_n, o_n >]_l$$

or simply

$$L = [Ro \hat{} Rot, * \bullet \bullet, R_{On}]i$$

where JR, or R_{Oi} is a template vector in T_n associated with the node o_t- in the template T [7]

A **template operator space** over T_n is the set C with two operations addition and scalar multiplication which are defined as follows: let

$$< R_1, o_1 >, < R_2, o_2 >, \cdots, < R_n, o_n >]_l,$$
$$= [R_{o_1}, R_{o_2}, \cdots, R_{o_n}]_l,$$

[7] **Here the subscript *I* of *L* means this expression is a *left form* of a template operator. Late, we will introduce its *right form***

$$L_2 = [<\tilde{R}_U O_X >, < \tilde{R}_2, O_2 >, \cdots, < \tilde{R}_n, o_n >]_l$$
$$= [\tilde{R}_{Ol}, \tilde{R}_{o_2}, \cdots, \tilde{R}_{o_n}]_l,$$

and

$$L_X + L_2 = [< i?l + fll, \tilde{O}! >, \cdots, < Rn + \tilde{Rn}, O_n >]_l$$
$$= [Roi + \tilde{\wedge}01, * * ', -R_{On} + \tilde{Ro}_n]h$$

$$ah = [< \alpha R_1, o_1 >, < \alpha R_2, o_2 >, \cdots, < \alpha R_n, o_n >]_l,$$
$$= [aR_{Ol}, a \wedge, \bullet \bullet \bullet, aR_{o_n}]_l.$$

Under these definition, £ is a linear space.

Let

$$X = \{X_{Ol}, X_{oj}, \cdots, X_{On}\}$$

be a template in T_n and

$$L = [R_{Ol} > R021 \bullet \bullet^{\#} > -Ron]/$$

be a template operator in £. Define the operation of L on x as follows:

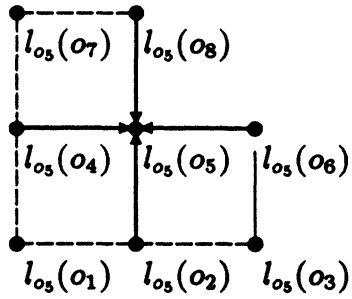
$$y = Lx$$
$$= [R_{o_1}, R_{o_2}, \cdots, R_{o_n}]_l x$$
$$= \{(R_{o_1}, x)_T, (R_{O_2}, x)_T, \cdots, (R_{o_n}, x)_T\}$$

where (R_{Oi},x) denotes the scalar product of the template vectors R_{Oi} and a:. We see that y is again a template vector in T_n. Under this definition, L is a linear operator, mapping T_n into T_n. In another words, L maps x to y, and y is the *image* of x under this mapping. L is called a **template operator** of the template space

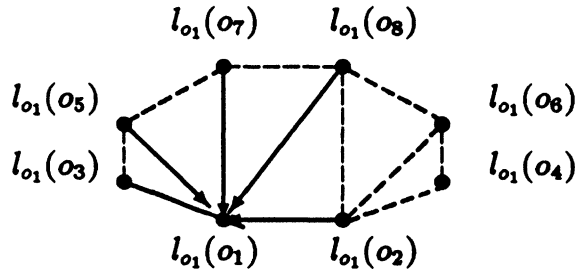
R_{Oi} is called an **operating template or input template** of the template operator L for node o_i. Let $l_{Oi}(oj)$ denote the component of R_{Oi} associated with the node Oj so that

$$R_{oi} = (< l_{o_i}\ o_1), o_1 >, < l_{o_i}(o_2), o_2 >, \cdots, < l_{o_i}(o_n\ ), O_n >).$$

$I_{0i}(o\%)$ is called the **center element** of the operator template for node o_i.



An in–web for node 5 in second template space



An In–Web for Node 1 in Fourth Template Space

Figure 4.3: Two in–webs

Given an operating template R_i, we may construct a directed graph in which for each

$$l_{o_i}(o_j) \neq 0$$

we put a path from node o_j to node o_i. This graph is a picture of how the operating template collects the information from the nodes which have non zero coefficients $l_{o_i}(o_j)$ and forms the value of the image y at node o_i. We call this graph representation of an operating template an **in–web** of the template operator for node o_i. Figure 4.1 shows two examples of in–webs. The in-web in the first picture, with four paths to node 5 from its neighbors, corresponds to a 5-point stencil for node five.

The structure of the left form of a template operator for the second template

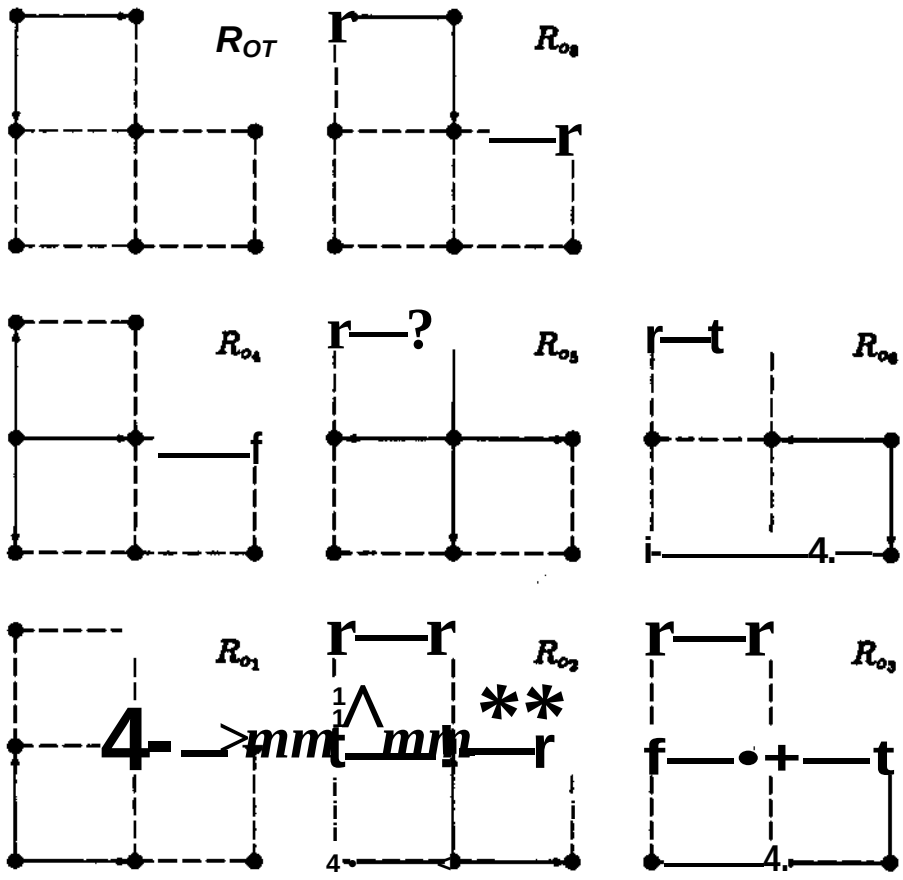


Figure 4.4: A template operator in the second template space

space in Fig. 4.1 has the following arrangement:

$$\begin{array}{lll} Ror & Ro. & \\ R_{o_4} & R_{o_5} & R_{o_6} \\ R_{O1} & R02 & R_{o_3} \end{array}$$

By expanding R_{Oi} to tinveil its internal structure, we may obtain the the picture in Fig. 4.4:

If we map the template vector to a conventional vector (using the same ordering as the nodes have), then the template operator corresponds to an n x n matrix as

follows:

$$A = \begin{bmatrix} l_{o_1}(o_1) & l_{o_1}(o_2) & \cdots & l_{o_1}(o_n) \\ l_{o_2}(o_1) & l_{o_2}(o_2) & \cdots & l_{o_2}(o_n) \\ \vdots & \vdots & \vdots & \\ l_{o_n}(o_1) & l_{o_n}(o_2) & \cdots & l_{o_n}(o_n) \end{bmatrix}_{n \times n}.$$

It is easy to see that the operating template R_{o_i} for node o_i corresponds to the row i of matrix A. Now let

$$e_1 = (< 1, o_1 >, < 0, o_2 >, \cdots, < 0, o_n >),$$

$$e_2 = (< 0, o_1 >, < 1, o_2 >, \cdots, < 0, o_n >),$$

$$\vdots$$

$$e_n = (< 0, o_1 >, < 0, o_2 >, \cdots, < 1, o_n >).$$

It is easy to see that $\{e_i, i = 1, \cdots, n\}$ is the basis of space $\mathcal{T}_n$.

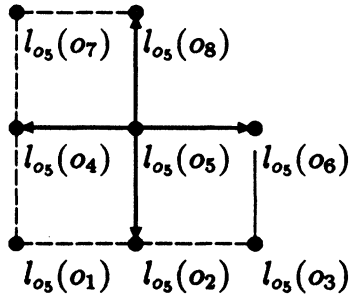
Applying L to the basis, we have

$$Le_i = \{\{l_{o_1}(o_i), l_{o_2}(o_i), \cdots, l_{o_n}(o_i)\}\} \qquad i = 1, \cdots, n.$$

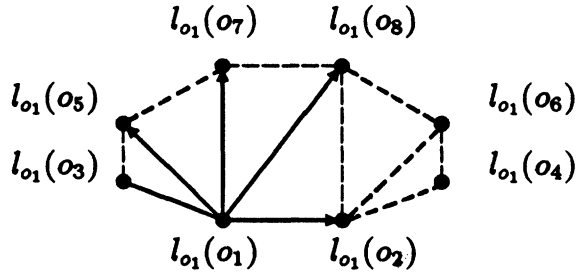
Let C_{o_i} denote Le_i, $i = 1, \cdots, n$. C_{o_i} are another set of templates which can be used to represent the template operator L. We call C_{o_i} the **image template** or **output template** for node o_i, because it is the image of e_i under the mapping L. We may also interpret it as the distribution of the output for a unit source at node o_i under this mapping. Again, o_i is called the center of this image template and $l_{o_i}(o_i)$ the center element of the image template.

We can see from the definition that the image template of the node o_i corresponds to the i-th column in the corresponding matrix we mentioned above.

Analogous to the in-web, another graph called the **out-web** for node o_i may be constructed, in which we include a path from o_i to o_j whenever $l_{o_i}(o_j) \neq 0$ in an image template. In this graph all paths start at node o_i and indicate which nodes are directly influenced by the node o_i. The pictures in Fig. 4.5 are two out-webs for



An out–web for node 5 in second template space



An Out–Web for Node 1 in Fourth Template Space

Figure 4.5: Two out–webs

the node o_5 of the second template space and for the node o_1 in the fourth template space in Figure 4.1, respectively.

Now the **right form** of a template operator can be defined as:

$$L = [< C_1, o_1 >, < C_2, o_2 >, \cdots, < C_n, o_n >]_r$$

or simply

$$L = [C_{o_1}, C_{o_2}, \cdots, C_{o_n}]_r.$$

Then the right product of a template vector and a template operator can be introduced as follows:

$$\begin{aligned} x = yL &= y[O_{o_1}, O_{o_2}, \cdots, O_{o_n}]_r \\ &= \{(y, O_{o_1})_T, (y, O_{o_2})_T, \cdots, (y, O_{o_n})_T\}. \end{aligned}$$

It is interesting to compare this operation with the corresponding operation $x = yA$ in a matrix structure. There, the matrix A keeps the same form while the

vector *y* needs to be transposed as a row vector. Here, the template vector y keeps the same form, but the operator needs to be expressed as a right form in order to obtain an operation consistent with the left form. The rule for deciding when we should use the right or left form of a template operator is simple: if L appears on the right of the operand , the right form is used and vice versa. We will see that with these two kinds of products the multiplication of two template matrices can be expressed very simply.

Let

$$\begin{aligned} A &= [R_{01}, R_{02} \cdots, R_{0n}]_l \\ &= [C_{o_1}, C_{o_2}, \cdots, C_{o_n}]_r, \end{aligned}$$

$$\begin{aligned} B &= [R'_{o_1}, R'_{o_2}, \cdots, R'_{o_n}]_l \\ &= [C'_{o_1}, C'_{o_2}, \cdots, C'_{o_n}]_r \end{aligned}$$

and

$$\begin{aligned} D = AB &= [R''_{o_1}, R''_{o_2}, \cdots, R''_{o_n}]_l \\ &\quad [C''_{o_1}, C''_{o_2}, \cdots, C''_{o_n}]_r \end{aligned}$$

It is easy to verify that

$$\begin{aligned} D = AB &= A[C'_{o_1}, C'_{o_2}, \cdots, C'_{o_n}]_r \\ &= [AC'_{o_1}, AC'_{o_2}, \cdots, AC'_{o_n}]_r \\ &= [R_{o_1}, R_{o_2}, \cdots, R_{o_n}]_l B \\ &= [R_{o_1}B, R_{o_2}B, \cdots, R_{o_n}B]_l \end{aligned}$$

$$\begin{aligned} R''_{o_i} &= R_{o_i}B \\ &= R_{o_i}[C'_{o_1}, \cdots, C'_{o_n}]_r \\ &= \{(R_{o_i}, C'_{o_1})_T, \cdots, (R_{o_i}, C'_{o_n})_T\}, \\ C''_{o_i} &= \wedge G \cdot \text{''}_{0i} \\ &= [R_{o_1}, \cdots, R_{o_n}]_l C'_{o_i} \\ &= \{(R_{o_1}, C'_{o_i})_T, \cdots, (R_{o_n}, C'_{o_i})_T\}, \end{aligned}$$

Each element in the product of two template operators is a scalar product of a operating template of A and an image template of B. An operating template of D is a left product of the operating template of A for the same node and the operator B. An image template of D is a right product of the operator A and the image template for the same node. It is easy to see the relationship with the operation of multiplying a row and a column in the matrix structure.

The transpose of any linear operator L can be simply obtained by swapping the input template and output template for each node.

$$\begin{aligned} L^T &= [C_{o_1}, C_{o_2}, \cdots, C_{o_n}]_l \\ &= [R_{o_1}, R_{o_2}, \cdots, R_{o_n}]_r. \end{aligned}$$

For a self-adjoint operator, we have $R_{o_i} = C_{o_i}, o_i \in \mathcal{O}$.

Corresponding to the concept of diagonal dominance in row or in column for the matrix structure, a template operator is *center dominant* in output or in input if

$$| l_{o_i}(o_i) | > \sum_{\substack{o_j \in \mathcal{O} \\ o_j \neq o_i}} l_{o_i}(o_j), \quad \text{for all} \quad o_i \in \mathcal{O},$$

or

$$| l_{o_i}(o_i) | > \sum_{\substack{o_j \in \mathcal{O} \\ o_j \neq o_i}} l_{o_j}(o_i), \quad \text{for all} \quad o_i \in \mathcal{O},$$

is true.

Although we have also developed other new concepts for template operators, we will not present them here, since they are not directly related to the discussion in the next section. The interested reader can refer to the Appendix B.

4.4 Estimates of the Decay of Inverse Operators

In this section we will concentrate on the discussion of a decay phenomenon for the inverse of a sparse template operator. As with the matrix structure, if there are only few non-zero elements in a template operator we call it a *sparse template operator*.

• stands for first influencing wavefront of the ♡ node

⊕ stands for second influencing wavefront of the ♡ node

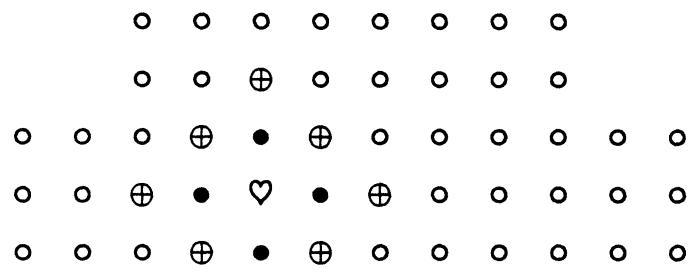


Figure 4.6: First and second influenced wavefronts

To describe the decay phenomenon more precisely, we need to introduce two important concepts: k^{th} *influencing wavefront* and k^{th} *influenced wavefront* of the node o_i in a template operator. Let

$$W_{out}^{(0)}(o_i) = o_i$$

be the center of the influenced wavefront for node o_i,

$$W_{out}^{(1)}(o_i) = \{o_j \mid l_{o_j}(o_i) \neq 0, o_j \neq o_i\}$$

be the set of all nodes except node o_i for which the corresponding elements in the output template C_{o_i} are non-zero. $W_{out}^{(1)}(o_i)$ is called the immediate or **first influenced** wavefront of the node o_i. The k^{th} influenced wavefront of the node o_i can be defined recursively:

Definition. 2 *The* k^{th} **influenced wavefront** *of the node* o_i, $W_{out}^{(k)}(o_i)$, *is a set of nodes* o_l *defined as follows:*

$$W_{out}^{(k)}(o_i) = \bigcup_{o_j \in W_{out}^{(k-1)}(o_i)} W_{out}^{(1)}(o_j) - \bigcup_{\mu=0}^{k-1} W_{out}^{(\mu)}(o_i)$$

Fig. 4.6 shows the first and the second influenced wavefronts of a node o_i.

Likewise, let

$$\mathcal{W}_{in}^{(0)}(o_i) = o_i$$

be the center of the influencing wavefront for node o_i,

$$\mathcal{W}_{in}^{(1)}(o_i) = \{o_j \mid l_{o_i}(o_j) \neq 0, o_j \neq o_i\}$$

be the set of all nodes except node o_i for which the elements in the input template R_{o_i} are non–zero. $\mathcal{W}_{in}^{(1)}(o_i)$ is called the **first influencing** wavefront of node o_i.

Definition. 3 *The k^{th}* **influencing wavefront** *of the node o_i, $\mathcal{W}_{in}^{(k)}(o_i)$, is a set of nodes o_l defined as follows:*

$$\mathcal{W}_{in}^{(k)}(o_i) = \bigcup_{o_j \in \mathcal{W}_{in}^{(k-1)}(o_{o_i})} \mathcal{W}_{in}^{(1)}(o_{o_j}) - \bigcup_{\mu=0}^{k-1} \mathcal{W}_{in}^{(\mu)}(o_{o_i}).$$

It is clear from the definition that

$$\mathcal{W}_{out}^{(\nu)}(o_i) \bigcap \mathcal{W}_{out}^{(\mu)}(o_i) = \emptyset,$$

$$\mathcal{W}_{in}^{(\nu)}(o_i) \bigcap \mathcal{W}_{in}^{(\mu)}(o_i) = \emptyset,$$

and[8]

$$\mathcal{O} = \bigcup_{j=0}^{p} \mathcal{W}_{out}^{(j)}(o_i) = \bigcup_{j=0}^{q} \mathcal{W}_{in}^{(j)}(o_i).$$

where $\mu \neq \nu$ and the p and q are the largest integers for which the $\mathcal{W}_{out}^{(p)}(o_i)$ and $\mathcal{W}_{in}^{(q)}(o_i)$ are not empty. Here p and q both depend on node o_i. To simplify the notation, we will not explicitly express this dependence. For many important applications, $\mathcal{W}_{out}^{(1)}(o_i)$ and $\mathcal{W}_{in}^{(1)}(o_i)$ are compact in the sense that the first influenced and influencing wavefronts are located in a small area in the solution space. For P.D.E. applications, the number of elements in $\mathcal{W}_{in}^{(1)}(o_i)$ or $\mathcal{W}_{out}^{(1)}(o_i)$ is typically bounded by a constant which is independent of the mesh size. We often call this property the **locality** of the operator. In terms of the graph representation, we have the following results which will be useful later.

[8]Here we assume the operator is irreducible.

Lemma 4.1 *The k-th influenced wavefront is the set of nodes to which the shortest path from node* o_i *is of length k.*

Similarly, the k-th influencing wavefront is the set of nodes from which the shortest path to node o_i *is of length k.*

Let $V(oi)$ and $U(<>i)$ be the first influencing and influenced wavefront for node o_i of the fc-th power of template operator L. We have[9]:

Lemma 4.2

$$\overset{k}{\mathrm{U}}\ \wedge in^{(j)}(<>0 = {}^{\mathrm{V}}(*)>$$

and

$$\bigcup_{j=1}^{k} W_{out}^{(j)}(o_i) = U(o_i).$$

The two kinds of wavefronts characterize how influences are propagated to or from other nodes graphically.

The identical template operator / is as follows:

$$I = [I_{o_1}, I_{o_2}, \cdots, I_{o_n}]_l = [I_{o_1}, I_{o_2}, \cdots, I_{o_n}]_r$$

where

$$\mathrm{J}^* \text{ - } \{kM = Sij \mid oj \text{ e } O).$$

The template vector I_{Oi} is a structure corresponding to the base vector in a ordinary vector space; both have only one element which is 1 and the rest are zeros.

Let

$$\begin{aligned} L^{-1} &= [R_{o_1}^{-1}, R_{o_2}^{-1}, \cdots, R_{o_n}^{-1}]_l \\ &= [C_{o_1}^{-1}, C_{o_2}^{-1}, \cdots, C_{o_n}^{-1}]_r \end{aligned}$$

be the inverse of the operator L where

$$\begin{aligned} R_{o_i}^{-1} &= \{l_{o_i}^{-1}(o_1), \cdots, l_{o_i}^{-1}(o_n)\}, \\ C_{o_i}^{-1} &= \{l_{o_1}^{-1}(o_i), \cdots, l_{o_n}^{-1}(o_i)\} \end{aligned}$$

[9] Here, we ignore the possibility of cancellation producing new zeros.

are the input and output templates of the node o_i for L^{-1}, respectively. It is easy to see that the input template $R_{o_i}^{-1}$ of the inverse is a discrete Green's function for node o_i. $(R_{o_i}^{-1}, f)$ is the solution of the equation

$$Lx = f$$

on node o_i. Intuitively, if L is a finite approximation of a linear differential operator and the mesh is fine enough the following is true:

$$(R_{o_i}^{-1}, f) \sim \int_\Omega G(P,Q) f(Q) d\tau_Q$$

and

$$l_{o_i}^{-1}(o_j) \sim G(o_i, o_j)\Delta_Q.$$

Let

$$\begin{aligned} \| R_{o_i}^{-1} \|_2^2 &= \sum_{o_j \in \mathcal{O}} (l_{o_i}^{-1}(o_j))^2, \\ \| C_{o_i}^{-1} \|_2^2 &= \sum_{o_j \in \mathcal{O}} (l_{o_j}^{-1}(o_i))^2. \end{aligned}$$

$\| R_{o_i}^{-1} \|_2^2$ approximates the norm of the Green's function

$$\int_\Omega G(P,Q)^2 d\tau_Q$$

if the linear operator is a discretized linear differential equation and the mesh size is small enough. The norm $\| R_{o_i}^{-1} \|$ is bounded by a constant which is independent of the mesh size if this finite approximation is stable. The structures of the continuous and discrete operators are more consistent for the template space than for the traditional vector space for the matrix structure.

Let χ_A be the characteristic function on the subset A. Construct two new sequences of template vectors

$$\begin{aligned} Q_i^{(k)} &= \chi_{W_{in}^{(k)}(o_i)} \circ R_{o_i}^{-1} \\ &= \{n^{(k)}(o_1), n^{(k)}(o_2), \cdots, n^{(k)}(o_N)\}, \\ & \qquad\qquad k = 1, \cdots, q, \end{aligned}$$

$$
\begin{aligned}
& & & i = 1, \cdots, N, \\
P_i^{(k)} &= \chi_{W_{out}^{(k)}(o_i)} \circ C_{o_i}^{-1} & & \\
&= \{m^{(k)}(o_1), m^{(k)}(o_2), \cdots, m^{(k)}(o_N)\}, & & \\
& & & k = 1, \cdots, p, \\
& & & i = 1, \cdots, N,
\end{aligned}
$$

where p and q are the largest integers for which $W_{in}^{(p)}(o_i)$ or $W_{out}^{(q)}(o_i)$ are not empty, and N is the number of the nodes for this template operator. By the above definition, we have:

$$
\begin{aligned}
n^{(k)}(o_j) &= \begin{cases} l_{o_i}^{-1}(o_j), & o_j \in W_{in}^{(k)}(o_i), \\ 0, & elsewhere, \end{cases} \\
m^{(k)}(o_j) &= \begin{cases} l_{o_j}^{-1}(o_i), & o_j \in W_{out}^{(k)}(o_i), \\ 0, & elsewhere. \end{cases}
\end{aligned}
$$

In another words, $P_i^{(k)}$ only gathers the elements of $C_{o_i}^{-1}$ on the k–th influenced wavefront and $Q_i^{(k)}$ gathers the elements of $R_{o_i}^{-1}$ on the k–th influencing wavefront. Let

$$
\begin{aligned}
\| Q_i^{(k)} \|_2^2 &= \sum_{o_j \in W_{in}^{(k)}(o_i)} (l_{o_i}^{-1}(o_j))^2, \\
\| P_i^{(k)} \|_2^2 &= \sum_{o_j \in W_{out}^{(k)}(o_i)} (l_{o_j}^{-1}(o_i))^2.
\end{aligned}
$$

We have the following theorem:

Theorem 4.1 *If there is an integer $k > 2$ such that $W_{in}^{(k)}(o_i)$ is not empty, then for any such integer k the following inequality holds:*

$$
\| Q_i^{(k)} \|_2^2 \leq \gamma^k \| R_{o_i}^{-1} \|^2 \tag{4.2}
$$

where $\gamma < 1$ and depends only on the condition number of the operator L. A corresponding result for the k^{th} influenced wavefront is also true:

$$
\| P_i^{(k)} \|_2^2 \leq \gamma^k \| C_{o_i}^{-1} \|^2. \tag{4.3}
$$

Proof. The proofs of (4.2) and (4.3) are parallel. Here we present the proof for (4.2). From the definition of an inverse, we have:

$$LC_{o_i}^{-1} = I_{o_i}.$$

Construct two sequences of template vectors:

$$\begin{aligned} R_i^{(k)} &= \pounds_{j=k+1} P_t U) \\ &= \{l^{(k)}(o_1), l^{(k)}(o_2), \cdots, l^{(k)}(o_N)\}, \\ W_i^{(k)} &= LR_i^{(k)} \\ &= \{R_{Ol}, \text{J?oj}, \cdots, Ro \wedge Rl^{(k)} \\ &= \{z^{(k)}(o_1), z^{(k)}(o_2), \cdots, z^{(k)}(o_N)\}, \\ & \qquad\qquad k = 1, \cdots. \end{aligned}$$

By the definition, we have:

$$l^{(k)}(o_j) = \begin{cases} 0, & Oi \in \bigcup_{\mu=1}^{k} W_{out}^{(\mu)}(o_i), \\ l_{o_i}^{-1}(o_j), & o_j \in \bigcup_{\mu=k+1}^{D} W_{out}^{(\mu)}(o_i), \end{cases}$$

$$z^{(k)}(o_j) = \begin{cases} 0, & o_j \in \bigcup_{\mu=1}^{k-1} W_{out}^{(\mu)}(o_i), \\ (\text{i}\,\mathfrak{z}_{o_j}, R_i^{(k)}), & o_j \in W_{out}^{(k)}(o_i) \cup W_{out}^{(k+1)}(o_i), \\ 0, & o_j \in \bigcup_{\mu=k+2}^{0} W_{out}^{(\mu)}(o_i). \end{cases}$$

It is clear that the non-zero patterns of $W_i^{\wedge}$ and $W\$^{k\sim 2\wedge}$ do not overlap. Now,

$$\begin{aligned} \| R_t \| &\sim \left[\sum_{j=k+1}^{2Li} \| P_i^{(k)} \|^2 \right]^{1/2} \\ &= \| L^{-1} L R_i^{(k)} \| \\ &\leq \| L^{-1} \| \| W_i^{(k)} \| \\ &\leq \| L^{-1} \| \| W_i^{(k)} - W_i^{(k-2)} \| \\ &= \| L^{-1} \| \| L(R_i^{(k)} - R_i^{(k-2)}) \| \\ &\leq \kappa_L \| (R_i^{(k)} - R_i^{(k-2)}) \| \\ &\leq \kappa_L \| \text{J}\rangle\!\rangle^{k>} + \text{i}\wedge\text{f}^{k-1)} \| . \end{aligned}$$

The rest of the proof can be completed by the following lemma [Dem77]:

Lemma 4.3 *Let* {a,-},->o *be & sequence of nonnegative numbers. If there is a* $K > 0$ *so that* $Zi \geq k \leq \#<**-i$ *for all* $k \geq 1$, *then* $a_k < [K/(K + l)]^k s_0$ *for all* $k > 1$, *where* $s_0 = \sum_{t>0} a.\%$ -

If a matrix has only real eigenvalues or is positive definite, a stronger result can be obtained. Here we present the result for a positive definite template operator.

Theorem 4,2 *If L is a positive definite template operator, then*

$$\| R_i^{(k)} \| \leq 2 \left(\frac{\sqrt{\kappa_L} - 1}{\sqrt{\kappa_L} + 1} \right)^k \| R_{o_i}^{-1} \|, \tag{4.4}$$

$$\| C_i^{(k)} \| \leq 2 \left(\frac{\sqrt{\kappa_L} - 1}{\sqrt{\kappa_L} + 1} \right)^k \| C_{o_i}^{-1} \| . \tag{4.5}$$

Proof. The proof of (4.4) is rather easier. Apply the optimal Chebychev iteration to the equation

$$L^T x = I_{Oi}$$

and choose the initial guess $x^{\wedge} = 0$, where I_{Oi} is the output template of the identity operator for node o_i. We know that the solution of this equation is the input template $R_{o_i}^*$ of $L^{\sim J}$ for node o,\ The fc-th iteration $x^{\wedge}$ has the error bound

$$\| x^{(k)} - R_{o_i}^{-1} \| \leq 2 \left(\frac{\sqrt{\kappa_L} - 1}{\sqrt{\kappa_L} + 1} \right)^k \| x^{(0)} - R_{o_i}^{-1} \| .$$

Since $z^{<>} = 0$, $*<*>(<*) = 0, o, \in O - V(o.)$ and

$$\| R_i^{(k)} \| \leq \| x^{(k)} - R_{o_i}^{-1} \| .$$

Using the same argument with

$$Lx = I_{o_i},$$

we can prove (4.5).

These two bounds are rather pessimistic, but we can still derive some very useful results from them. For example, apply the Crank–Nicolson scheme to a parabolic boundary problem

$$\begin{aligned} \frac{\partial u}{\partial t} + \mathcal{L}u &= f, & x \in \Omega \times (0,T], \\ lu &= g, & x \in \Gamma_\Omega \times (0,T], \\ u(x,0) &= u^0(x), \end{aligned}$$

where

$$\begin{aligned} \mathcal{L}u &\equiv -\sum_{i=1}^{k} \frac{\partial}{\partial x_i} a_i \frac{\partial u}{\partial x_i} + bu, \\ lu &\equiv cu + d\frac{\partial u}{\partial n}, \end{aligned}$$

$a_i > 0$, b, c and $d \geq 0$. It is known that the condition number of the resulting linear system has the bound [Kuz87]

$$\kappa_L \leq C\tau h^{-2}$$

where τ is the time step and h is the spatial mesh interval. If we choose $\tau \simeq h$, then asymptotically, we have

$$\| R_i^{(k)} \| \simeq Ce^{-2\sqrt{kd}}$$

where $d \cong kh$ is the distance between the node o_i and the node in the k–th wavefront. There is almost no influence to node o_i from those nodes which are a few wavefronts away. Thus, if we apply $\mathcal{SS}$ to this problem, the overlapping needed is very small.

For a higher–dimensional problem where decay of the norm of the wavefront does occur, the average size of an element in the k–th wavefront even diminishes faster, since the number of elements in k–th wavefronts will increase when k increases. The number of elements in each wavefront is proportional to $k^{(\alpha-1)}$ where α is the dimension of the solution space, but the bound on the rate of decrease in (4.2), (4.3), (4.4) and (4.5) depends only upon the condition number of the operator. Because the condition number of approximations to the model problem is only weakly related to the dimensionality of the solution space, the individual elements of the inverse operator decrease faster in a higher-dimensional grid, explaining the faster convergence of $\mathcal{SM}$ for higher-dimensional problems.

Unfortunately, these bounds still cannot guarantee that the norm of the k-th wavefront of the input or output template of the inverse will decay as k increases. In Section 4.2, we presented illustrative counterexamples and explained how bounds such as (4.2) and (4.3) can permit growth. The conditions of spaxsity or positive-definiteness are not strong enough to ensure that decay will occur. Here we will present a sufficient condition which will yield decay.

For simplicity, in the following discussion we rescale the template operator L so that

$$hXoi) = 1, \quad \text{for all} \quad Oi \in O.$$

If a template operator is output or input strictly center dominant, then we have the following result.

Theorem 4.3 *Let $L = I - B$ be a sparse template operator and $\| B \|_\infty = 7 < 1$, then*

$$\| Pl^{k)} \| L * 7 \| \wedge *''^{1\}} \| L - \qquad (4.6)$$

//$\|B\|_1 = 7 < 1$, then

$$\| Q\backslash^{k)} \| L * 7 \| Qt'' \| L. \qquad (4.7)$$

Proof. First note that $\| B [_\infty < 1$ or $\| B \backslash\backslash_t < 1$ are equivalent to saying that L is input or output center dominant, respectively. Another important fact for the proof is the following: if

$$Oi \in VV\text{≤}2(o<),$$

from Lemma 4.1 we have

$$W_{in}^{(1)}(o_j) \subset \bigcup_{l=k-1}^{q} W_{out}^{(l)}(o_i).$$

This means that the input template R_{Oj} only has non-zero elements in the *(k*—l)-th or **higher influenced wavefronts of** the **node o_i*. Similarly, if**

$$o_j \in W_{in}^{(k)}(o_i),$$

then

$$W_{out}^{(1)}(o_j) \subset \bigcup_{l=k-1}^{p} W_{in}^{(l)}(o_i).$$

The proof of (4.6) and (4.7) axe parallel. Let's prove (4.6).

Vz' we have

$$\begin{aligned} LC_{o_i}^{-1} &= \{(R_{o_1}, C_{o_i}^{-1})_T, \cdots, (R_{o_n}, C_{o_i}^{-1})_T\}, \\ &= I_{o_i} \end{aligned}$$

and

$$\begin{aligned} (R_{o_j}, C_{o_i}^{-1})_T &= \sum_{o_k \in W_{in}^{(1)}(o_j)} l_{o_k}(o_j) l_{o_i}^{-1}(o_k) + l_{o_j}(o_j) l_{o_i}^{-1}(o_j) \\ &- \sum_{(o_j)} l_{o_k}(o_j) l_{o_i}^{-1}(o_k) + l_{o_i}^{-1}(o_j) \end{aligned}$$

•*Bit*

First, we prove: when *k* = p, (4.6) is true. Let *Oj* be the node in W^(°t) such that

$$C^{1}(\circ i) = \| P_{\%}^{(p)} \|_{00}.$$

Then we have

$$\begin{aligned} (R_{o_j}, C_{o_i}^{-1})_T &= \sum_{o_k \in W_{in}^{(1)}(o_j)} l_{o_k}(o_j) l_{o_i}^{-1}(o_k) + l_{o_i}^{-1}(o_j) \\ &= 0, \end{aligned}$$

$$\mathrm{I}\, tfiPi) \mid = \| \mathrm{p}/^{\mathrm{rt}} \mathrm{IL} \leq \sum_{o_k \in W_{in}^{(1)}(o_j)} \mathrm{I}\, \mathrm{U}\circ \mathrm{i})\, \mathrm{II}\ tfW\ \mathrm{I}\bullet$$

$$W_{out}^{(p-1)}(o_i),$$

$$\| P_i^{(p)} \|_\infty \leq \left(\sum_{o_k \in W_{out}^{(p)}(o_i) \cap W_{in}^{(1)}(o_j)} | l_{o_k}(o_j) | \right) \| P_i^{(p)} \|_\infty + \left(\sum_{o_k \in W_{out}^{(p-1)}(o_i) \cap W_{in}^{(1)}(o_j)} | l_{o_k}(o_j) | \right) \| \wedge^{\prime\prime 1)} \mathrm{IL}- \qquad (4\text{-}8)$$

If || *P$^{p)}* H^ ≥ || P/$^{\mathrm{P}\sim x)}$ U^, we could conclude that 1 ≤ 7, a contradiction. Thus we can replace || *P\^{v)}* *W*^ in the right hand side of (4.8) by || *P\^{v}*$^{\sim x)}$ *W*^ and obtain

$$\mathrm{II}\ \mathrm{P}^{(\mathrm{I}>)}\ \mathrm{II} < \text{'v}\mathrm{II}\ \mathrm{P}.\wedge^{\prime\prime 1})\ \mathrm{II}$$
$$\mathrm{II}\ \bullet\text{»«} \quad \|_{00} - \text{'II}\ \bullet\text{*»} \quad \|_{00}\text{'}$$

Using reverse induction, suppose

$$\| P_i^{(k)} \|_\infty \leq \gamma \| P_i^{k-1)} \|_\infty, \qquad k = k+1, \cdots, p$$

is true. Let o_j be the node in $W_{out}^{(k)}(o_i)$ such that

$$l_{o_i}^{-1}(o_j) = \| P_i^{(k)} \|_\infty.$$

Then we have

$$\begin{aligned}(R_{o_j}, C_{o_i}^{-1})_T &= \sum_{o_k \in W_{in}^{(1)}(o_j)} l_{o_k}(o_j) l_{o_i}^{-1}(o_k) + l_{o_i}^{-1}(o_j) \\ &= 0,\end{aligned}$$

$$| l_{o_i}^{-1}(o_j) | = \| P_i^{(k)} \|_\infty \leq \sum_{o_k \in W_{in}^{(1)}(o_j)} | l_{o_k}(o_j) \| l_{o_i}^{-1}(o_k) | .$$

Since

$$W_{in}^{(1)}(o_j) \subset \bigcup_{l=k-1}^{p} W_{out}^{(l)}(o_i),$$

we have

$$\begin{aligned}\| P_i^{(p)} \|_\infty \leq & \left(\sum_{o_k \in \{ \bigcup_{\nu=k}^{p} W_{out}^{(\nu)}(o_i) \} \cap W_{in}^{(1)}(o_j)} | l_{o_k}(o_j) | \right) \| P_i^{(k)} \|_\infty \\ & + \left(\sum_{o_k \in W_{out}^{(k-1)}(o_i) \cap W_{in}^{(1)}(o_j)} | l_{o_k}(o_j) | \right) \| P_i^{(k-1)} \|_\infty. \end{aligned} \tag{4.9}$$

As with in (4.8), $\| P_i^{(k)} \|_\infty \geq \| P_i^{(k-1)} \|_\infty$ will lead to a contradiction. We therefore have

$$\| P_i^{(k)} \|_\infty \leq \gamma \| P_i^{(k-1)} \|_\infty, \qquad k = 1, \cdots, p.$$

The proof is completed.

Notice that $\| B \|_\infty < 1$ does not ensure the conclusion of (4.7). There is a mistake in a theorem dealing with a diagonally dominant tridiagonal matrix in a recent paper of Rong-Qing Jia. He claims that $\| B \|_\infty < 1$ could yield a sharp

estimate for both columns and rows. A simple counter-example is the following matrix, which is row diagonally dominant but not column diagonally dominant:

$$A = \begin{bmatrix} -56.1 & 35 & & & & & & \\ 21 & -55.1 & 34 & & & & & \\ & 21 & -54.1 & 33 & & 0 & & \\ & & 21 & -53.1 & 32 & & & \\ & & & \ddots & \ddots & & & \\ & & & & & \ddots & \ddots & \\ & & & & & 21 & -35.1 & 14 \\ & & & & & & 21 & -34.1 \end{bmatrix}.$$

The inverse of this matrix exponentially decays in its columns but not in its rows.

If we loosen the condition of strictly center dominant to center dominant, then following are true:

$$\| P \|_{L} \leq \| P \|_{\infty} \tag{4-10}$$

or

$$\| Q_i^{(k)} \|_{\infty} \leq \| Q \|_{L} \tag{4.11}$$

We may also derive a simple bound as follows:

Theorem 4.4 *Let $L = I - B$ be a sparse linear operator. If* $\| B \|_{\infty} = \gamma < 1$, *then*

$$\| R_i^{(k)} \| \leq \frac{\gamma^{k+1}}{1-\gamma}, \tag{4.12}$$

$$\| C_i^{(k)} \| \leq \frac{\gamma^{k+1}}{1-\gamma}, \tag{4.13}$$

Proof: **If $\| B \|_{\infty} < 1$, we have $L^{-1} = \sum_{n=0}^{\infty} B^n$ and $C_{o_i} = (\sum_{n=0}^{\infty} B^j) I_{o_i}$. From lemma 4.2, it follows that**

$$R_i^{(k)} = \left(\sum_{n=k+1}^{\infty} B^n \right) I_{o_i}$$

Consequently,

$$\| P_i^{(k)} \|_\infty = \| (\sum_{n=k+1}^{\infty} B^n) I_{o_i} \|_\infty \leq \frac{\gamma^{k+1}}{1-\gamma}.$$

The proof of (4.13) is parallel with the proof of (4.12).

For an M–matrix A, there is a diagonal matrix $D > 0$ such that DA is strictly diagonally dominant. Applying this result to the corresponding template operator, the exponential decay law can then be applied to the new operator.

It is also very interesting that for some problems (e.g. the five point discrete Laplace operator in a rectangle, which we will analyze in detail in the next chapter), the operator is not center dominant in the physical space, but is center dominant in the Fourier space.

The concept of the wavefront also allows us to discuss the exponential decay for some random sparse linear operators without ordering the nodes since there is no ordering relationship involved in the definitions of the influencing and influenced wavefronts.

Chapter 5

Model Problem Analysis

The exponential decay law presented in the last chapter has shown that there is a relationship between the overlap and the convergence of $\mathcal{SAM}$. A general quantitative relation is very hard to derive for arbitrary cases. A common approach is to analyze prototype model problems. In this chapter we present spectral radius analyses of the Schwarz splitting ($\mathcal{S}$) for model problems in one– and higher–dimensional solution spaces. We have found that the convergence speed of $\mathcal{SAM}$ is a function of the overlap, the geometries of the subregions, the frequency of the Fourier component and the dimension of the solution space.

The relationship between convergence and the area of the overlap has been observed previously. Miller [Mil65] proved a result for the case of two overlapping rectangles, while Kantorovich and Krylov mentioned in their convergence proof that the convergence rate is related to the geometries of the subregions. They were mainly interested in solving elliptic equations in irregular regions; an analysis for applications motivated by parallel processing and composite grids has not been carried out. Our analyses extend the earlier work in the following respects:

- The number of the subregions can be an arbitrary finite number.
- A quantitative relation between the convergence and the shapes of the subregions is shown.
- A relation between the convergence and the dimension of the solution space is explored.
- For two– or higher–dimensional solution spaces, the analyses are carried out in Fourier space. The convergence speeds for different frequencies are presented.

- An analysis of higher–order finite difference schemes is carried out.

These analyses have provided guidelines for implementing an efficient parallel algorithm for the solution of elliptic PDE's.

5.1 One–Dimensional Case

There is no practical reason for paralellizing the solution of a one–dimensional model problem, but the analysis of this problem provides some results useful for the higher–dimensional cases. It also makes the whole analysis more complete.

The model problem in one dimension which we will consider is

$$y''(x) = f(x), \qquad x \in (0,1),$$

$$y(0) = \alpha; \quad y(1) = \beta.$$

After discretization using a centered second order method, the resulting linear system is

$$T_n x = b, \tag{5.1}$$

where

$$T_n = Tridiagonal\{1, \quad -2, \quad 1\}_{n\times n}.$$

The $S\!\!\!/\!\!\!4M$ for solving this problem divides the region into k overlapping subregions Ω_i; $i = 1, \cdots, k$ as shown in Figure 5.1. (To simplify the analysis we assume the overlap pattern is uniform. Similar conclusions can be deduced for more general cases.)

Let h be the grid size, ℓ the length of the overlap and η the length of every subregion. Then let $n+1 = \frac{1}{h}$, $l = \frac{\ell}{h}$ and $m+1 = \frac{\eta}{h}$. The circular points in Figure 5.1 are the boundaries of the subregions. A natural way to implement $S\!\!\!/\!\!\!4M$ is to first guess some "reasonable" initial values on the artificial boundaries and then to solve these subproblems separately. Next, use the solutions of these subproblems to update the values on the artificial boundaries and proceed iteratively until the solutions on the overlapping regions converge. If we solve on these subregions in a natural order, each succeeding subregion takes its boundary values from the new

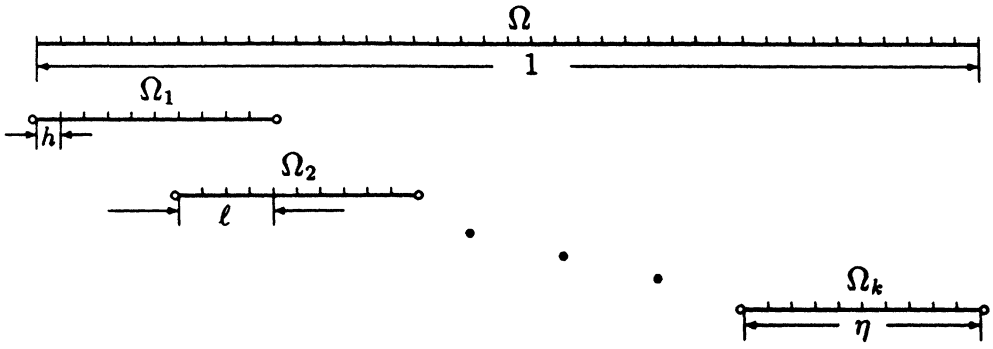


Figure 5.1: One–dimensional overlapping grid.

solution on the previous subregion. As we have shown in Chapter 3, this procedure is equivalent to applying the block Gauss-Seidel method to the Schwarz enhanced equation ($S_E E$):

$$\tilde{T}\tilde{x} = \begin{bmatrix} T_m & F_m & & & & & \\ E_m & T_m & F_m & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & E_m & T_m & F_m \\ & & & & & E_m & T_m \end{bmatrix} \tilde{x} \qquad (5.2)$$

$$= (T_m \otimes I_k + E_m \otimes L_k + F_m \otimes U_k)\tilde{x} = \tilde{b}.$$

The corresponding block Gauss-Seidel iteration for this equation is as follows:

$$(E_m \otimes L_k + T_m \otimes I_k)\tilde{x}^{(k+1)} = -(F_m \otimes U_k)\tilde{x}^{(k)} + \tilde{b}. \qquad (5.3)$$

The quantities above are defined as:

- E_m: an $m \times m$ matrix with zero elements everywhere except for 1 in position $(1, m - l + 1)$.

- F_m: an $m \times m$ matrix with zero elements everywhere except 1 in position (m, l).
- I_k: a $k \times k$ identity matrix.
- L_k: a $k \times k$ matrix with zero elements everywhere except for 1's on the subdiagonal.
- U_k: a $k \times k$ matrix with zero elements everywhere except for 1's on the superdiagonal.

As we showed in Chapter 3, (5.1) and (5.2) are equivalent. Therefore, the convergence analysis of $\mathcal{SAM}$ is reduced to calculating the eigenvalues of the block Jacobi matrix $J = M^{-1}N$ of the $\mathcal{SS}$ where

$$\begin{aligned} M &= T_m \otimes I_k, \\ N &= E_m \otimes L_k + F_m \otimes U_k. \end{aligned}$$

If we multiply out $M^{-1}N$ then

$$\begin{aligned} J &= (T_m \otimes I_k)^{-1}(E_m \otimes L_k + F_m \otimes U_k) \\ &= (T_m^{-1} \otimes I_k)(E_m \otimes L_k + F_m \otimes U_k) \\ &= (T_m^{-1}E_m) \otimes L_k + (T_m^{-1}F_m) \otimes U_k \\ &= \tilde{E}_m \otimes L_k + \tilde{F}_m \otimes U_k, \end{aligned}$$

where $\tilde{E}_m$ and $\tilde{F}_m$ have almost all zero elements except columns $(m - l + 1)$ and l, respectively. The rank of this matrix is clearly at most $2(k - 1)$. After some row and column exchanges J can be transformed to $\tilde{J}$, which is similar to J:

$$\tilde{J} = UJU^T = \begin{bmatrix} 0_{(n-2k)\times(n-2k)} & C_{(n-2k)\times 2k} \\ 0_{2k\times(n-2k)} & G_{2k\times 2k} \end{bmatrix},$$

where

$$G = D' \otimes I_k + E' \otimes L_k + F' \otimes U_k,$$

$$E' = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix}, \quad D' = \begin{bmatrix} 0 & b \\ b & 0 \end{bmatrix}, \quad F' = \begin{bmatrix} 0 & 0 \\ 0 & a \end{bmatrix},$$

$$a = \frac{l}{(m+1)}, \quad b = \frac{m+1-l}{(m+1)}.$$

It is clear that $\lambda_J \in (0 \cup \lambda_G)$, where λ_J and λ_G are the eigenvalues of matrices J and G, respectively. Using matrix polynomial theory we may obtain the following theorem.

Theorem 5.1 *If $a \leq b$ then λ_G satisfies the following equation:*

$$\lambda_G^2 + 2 * a * \cos\theta * \lambda_G + a^2 - b^2 = 0$$

where the parameter θ is the root of the following equation:

$$\left(\cos\theta + \sqrt{\left(\frac{b}{a}\right)^2 - \sin^2\theta}\right) = \frac{\sin((k-1)\theta)}{\sin k\theta}. \tag{5.4}$$

The proof of this theorem is lengthy and has nothing to do with the discussion of SAM. We present it in an appendix.

Let

$$\rho = max\{|\ \lambda_G\ |\} = max\{|\ \lambda_J\ |\},$$

it is easy to show that ρ corresponds to the smallest root θ^* of equation (5.4). In particular, if $k = 2$

$$\rho = b,$$

and if $k = 3$

$$\rho = \sqrt{b}.$$

Now we can immediately observe some important facts about SAM :

1. First, the spectral radius of J only depends on the number of subregions k and the overlapping area a. If both k and a are independent of the mesh size h, then the convergence of SAM is also independent of h. Figure 5.2 shows the distribution of the roots θ when $k = 4, 5, 6, 7$, with $a = 0.4$ and $b = 0.6$ for all four values of k. As k increases, the curve for the left hand side of equation (5.4) remains the same, while the frequency of the jumps in the curve for

the right hand side increases. We can see that the smallest root of equation (5.4) moves leftward when k is increased. This implies that p increases if k is increased. From these pictures we can also see that matrix G has $2k$ distinct real eigenvalues when $a \leq b$.

2. For the cases of $k = 2$ and 3, we notice that when the overlapping area increases, p decreases, and when k increases, p increases. These conclusions also are valid for the general case ($k > 2$). We cannot give a closed form solution for k greater than 5, but the numerical results indicate that these results hold. Figure 5.3 [1] shows the theoretical and computational values of p. The computational results (denoted by ©, A etc..) are very well matched with the theoretical values. This picture also indicates that the conclusions we mentioned are general.

3. Furthermore, the $\bar{SEM}$ of the matrix T_n has Property A^ [You71], thus the Gauss-Seidel iteration can certainly be improved by the SOR acceleration. Since p is known, the optimal relaxation parameter can be estimated exactly (see Chapter 6 for detailed discussion).

[1] Increasing $K = l/m$ corresponds to increasing overlap.

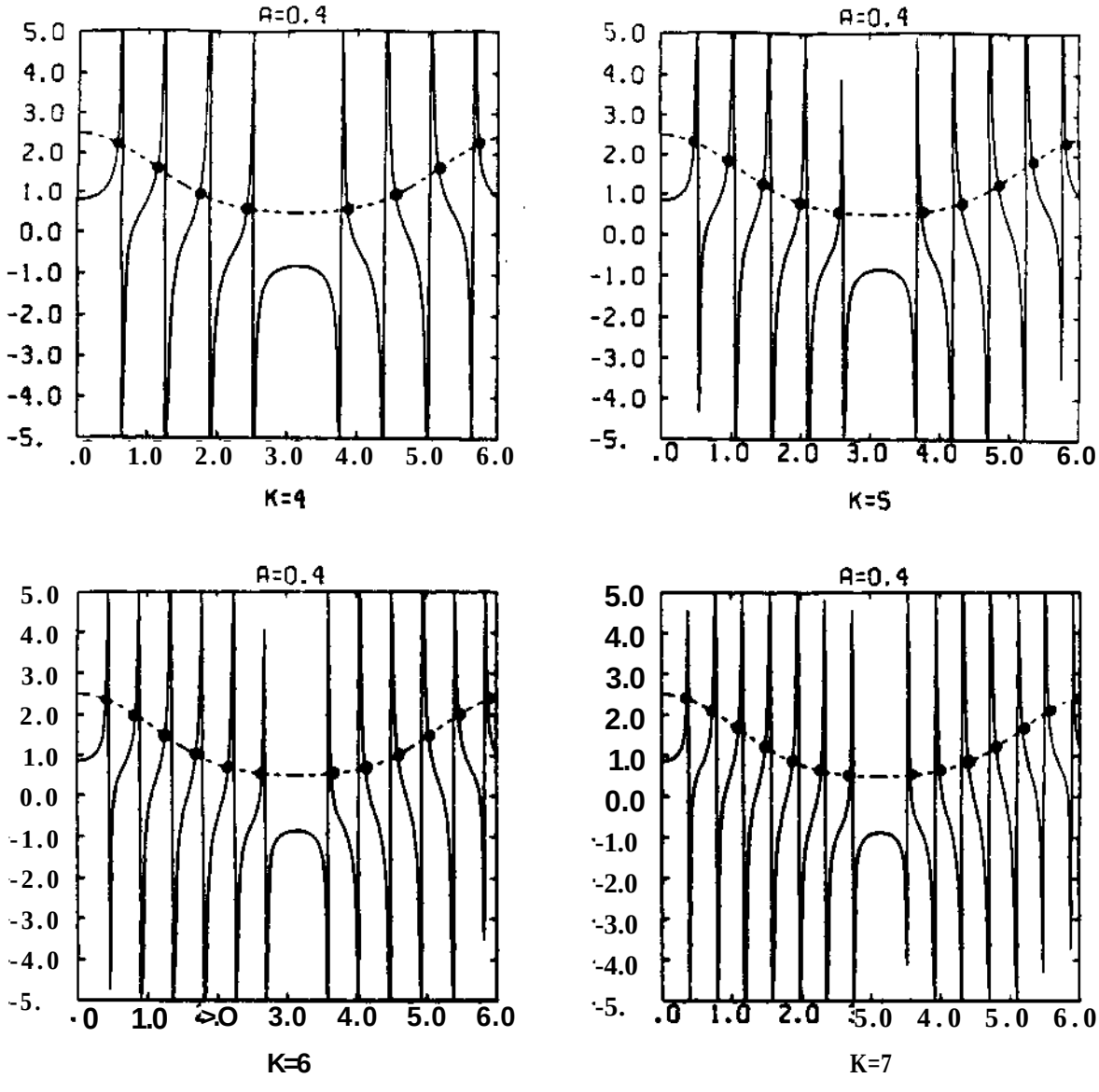


Figure 5.2: The distributions of the roots for different *k*.

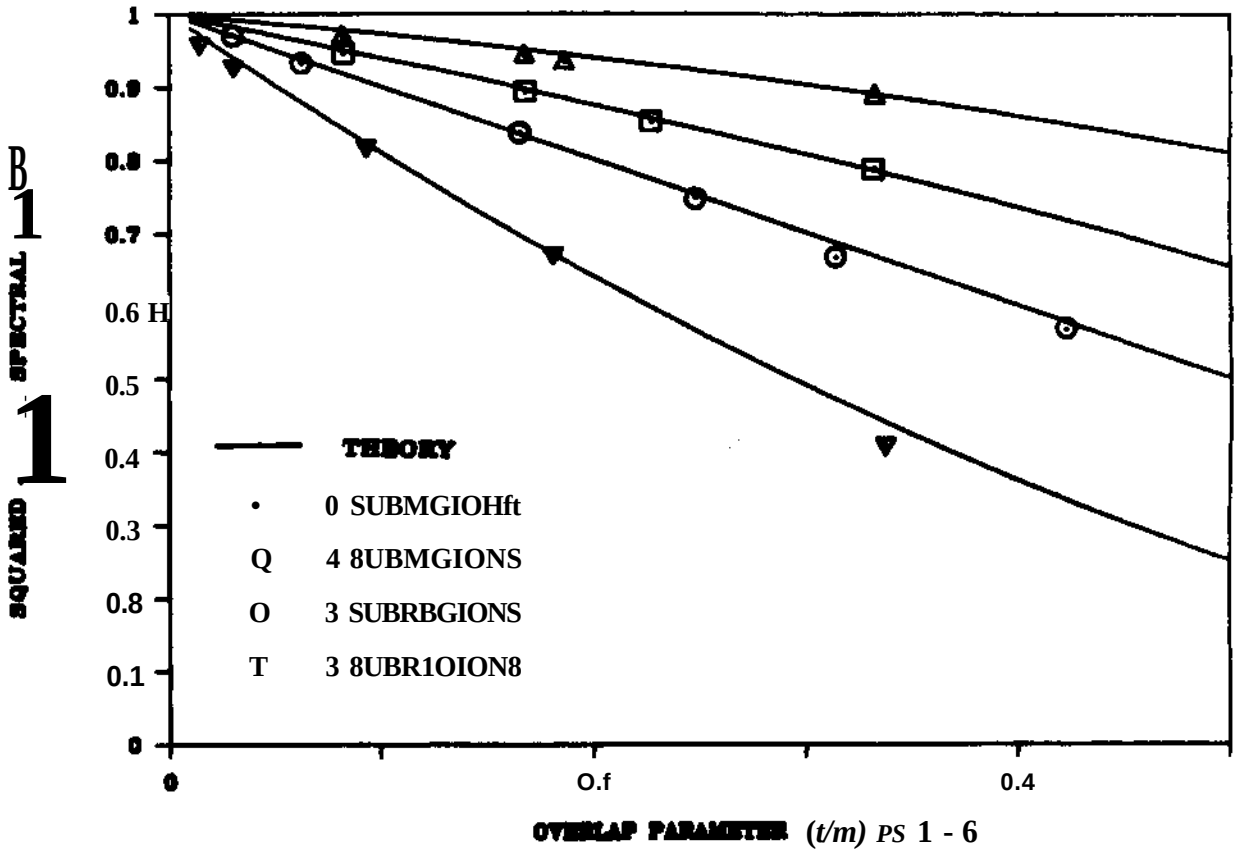


Figure 5.3: **Theoretical and computational values of the squared spectral radius for the block Jacobi iteration matrix in the 1-D case.**

5.2 Two- and Higher-Dimensional Cases

Two-dimensional model problems are commonly used to test numerical methods for the solution of elliptic PDE's. Here we use a method which combines Fourier analysis with the method used in the last section to analyze the application of SAM to the two-dimensional model problem. The same approach can be applied in higher-dimensional cases.

The Poisson equation in two-dimensions is:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f(x,y), \qquad (x,y) \in (0,1) \times (0,1), \tag{5.5}$$

$$U\,|_\Gamma = g(x,y). \tag{5.6}$$

Using central differences we obtain a discretization of this equation:

$$Ax = b, \tag{5.7}$$

where

$$A = T_n \otimes I_n + I_n \otimes T_n.$$

This is of the same form as we obtained in the one-dimensional case, with h being the mesh size and $n+1 = \frac{1}{h}$. If we cover (0,1) on the x axis with k subregions as in the one-dimensional case, then the solution area is covered by k overlapping rectangles as shown below [2].

If we apply SAM to these overlapping subregions, then it is equivalent to applying the Gauss-Seidel method to the following SEE:

$$\tilde{A}\tilde{x} = \begin{bmatrix} W_m & F'_m & & & & & \\ E'_m & W_m & F'_m & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & E'_m & W_m & F'_m \\ & & & & & E'_m & W_m \end{bmatrix}_{k\times k} \tilde{x} \tag{5.8}$$

$$= \{W_m \otimes I_k + (I_n \otimes E_m) \otimes L_k + (I_n \otimes F_m) \otimes U_k\}\tilde{x} = \tilde{b},$$

[2] The subregions are shifted upwards to improve visibility of the overlapping pattern.

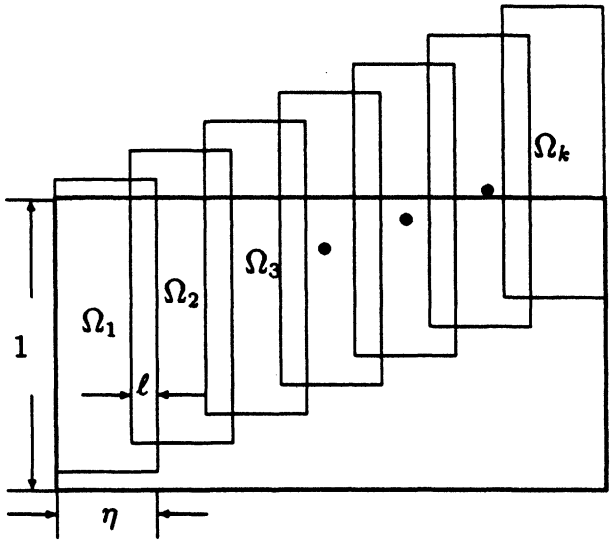


Figure 5.4: Two–dimensional overlapping grid.

where

$$W_m = T_n \otimes I_m + I_n \otimes T_m, \quad E'_m = I_n \otimes E_m, \quad F'_m = I_n \otimes F_m.$$

In order to analyze the convergence of S_M, we need to study the spectral radi of the block Jacobi iterative matrix of the S:

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

where

$$M = W_m \otimes I_k, \qquad N = (I_n \otimes E_m) \otimes L_k + (I_n \otimes F_m) \otimes U_k.$$

We have the following result:

Theorem 5.2 *The matrix J is similar to the matrix*

$$D = \begin{bmatrix} 0_{(n^2-2nk)\times(n^2-2nk)} & C_{(n^2-2nk)\times 2nk} \\ 0_{2nk\times(n^2-2nk)} & G_{2nk\times 2nk} \end{bmatrix}$$

where

$$G = Block - diagonal\{D_i\}, \quad i = 1, \cdots, n\ ,$$

$$D_i = D_i' \otimes I_k + E_i' \otimes L_k + F_i' \otimes U_k,$$

$$E_i' = \begin{bmatrix} \alpha_i & 0 \\ 0 & 0 \end{bmatrix}, \quad D_i' = \begin{bmatrix} 0 & \beta_i \\ \beta_i & 0 \end{bmatrix}, \quad F_i' = \begin{bmatrix} 0 & 0 \\ 0 & \alpha_i \end{bmatrix},$$

$$\alpha_i = \frac{\sinh \kappa m\theta_i}{\sinh m\theta_i}, \quad \beta_i = \frac{\sinh (1-\kappa) m\theta_i}{\sinh m\theta_i},$$

$$\cosh \theta_i = 2 - \cos \frac{i\pi}{n+1} \quad i = 1, \cdots, n \ ,$$

and $\kappa = l/m$ is the overlap ratio. Let ρ_i be the spectral radius of the D_i, then each ρ_i is the convergence factor for the corresponding Fourier component of the error in the approximation.

Proof. Let

$$U = (X_n \otimes I_m) \otimes I_k$$

where X_n is an orthogonal matrix whose columns are the eigenvectors of the matrix T_n, and U is an orthogonal matrix. Note that $UNU^T = N$. Then

$$\begin{aligned} J' = UJU^T &= UM^{-1}NU^T \\ &= (UMU^T)^{-1}N \\ &= \{((X_nT_nX_n^T \otimes I_m) + I_n \otimes T_m)^{-1} \otimes I_k\}N \\ &= \{(D_n \otimes I_m + I_n \otimes T_m)^{-1} \otimes I_k\}N \end{aligned}$$

where D_n is a diagonal matrix whose diagonal elements are the eigenvalues of T_n. We know that there is a $mn \times mn$ permutation matrix P such that

$$P(A \otimes B)P^T = B \otimes A$$

where A and B are any $n \times n$ and $m \times m$ matrices, respectively. So we have

$$P(I_n \otimes E_m)P^T = E_m \otimes I_n, \quad P(I_n \otimes F_m)P^T = F_m \otimes I_n,$$

$$P(D_n \otimes I_m + I_n \otimes T_m)P^T = I_m \otimes D_n + T_m \otimes I_n.$$

Notice that:

$$\begin{aligned} Q &= I_m \otimes D_n + T_m \otimes I_n \\ &= Block\text{-}diagonal\{\tilde{T}_i\}_{n\times n}, \\ \tilde{T}_i &= Tridiagonal\{1, \gamma_i, 1\}_{m\times m} \\ \gamma_i &= -4 + 2\cos(\frac{i\pi}{n+1}), \quad i = 1, \cdots, n. \end{aligned}$$

Let $\tilde{P} = P \otimes I_k$. Then we have

$$\begin{aligned} J'' = \tilde{P}J'\tilde{P}^T &= \tilde{P}\{(D_n \otimes I_m + I_n \otimes T_m)^{-1} \otimes I_k\}\tilde{P}^T\tilde{P}N\tilde{P}^T \\ &= \{(P(D_n \otimes I_m + I_n \otimes T_m)P^T)^{-1} \otimes I_k\}\{(E_m \otimes I_n) \otimes L_k + (F_m \otimes I_n) \otimes U_k\} \\ &= (Q^{-1} \otimes I_k)\{(E_m \otimes I_n) \otimes L_k + (F_m \otimes I_n) \otimes U_k\} \\ &= [Q^{-1}(E_m \otimes I_n)] \otimes L_k + [Q^{-1}(F_m \otimes I_n)] \otimes U_k. \end{aligned}$$

As in the one–dimensional analysis, we can move all of the non-zero columns to the last columns and the theorem follows.

Since the structures of these diagonal blocks are the same as those analyzed in the one–dimensional case, we can find a tight estimate of ρ_J, the spectral radius of J, by using theorem 5.1. But here it is clear that

$$\alpha_i + \beta_i < 1, \quad \alpha_i > 0, \quad \beta_i > 0 \quad i = 1, \cdots, n$$

and thus we cannot derive a closed form of ρ_J for general k, but we may use the Gershgorin theorem to get a very good bound for ρ_J.

Corollary 4

$$\rho_J \leq \alpha_1 + \beta_1.$$

If we denote $\mu = \frac{m}{n}$, it is easy to estimate the asymptotic bound for ρ_J (as $h \to 0$):

Corollary 5 *If* $k = 2$,

$$\rho_J \leq \frac{\sinh((1-\kappa)\mu\pi)}{\sinh(\mu\pi)}.$$

If $k > 2$,

$$\rho_J \leq \frac{\sinh(\kappa\mu\pi) + \sinh((1-\kappa)\mu\pi)}{\sinh(\mu\pi)}.$$

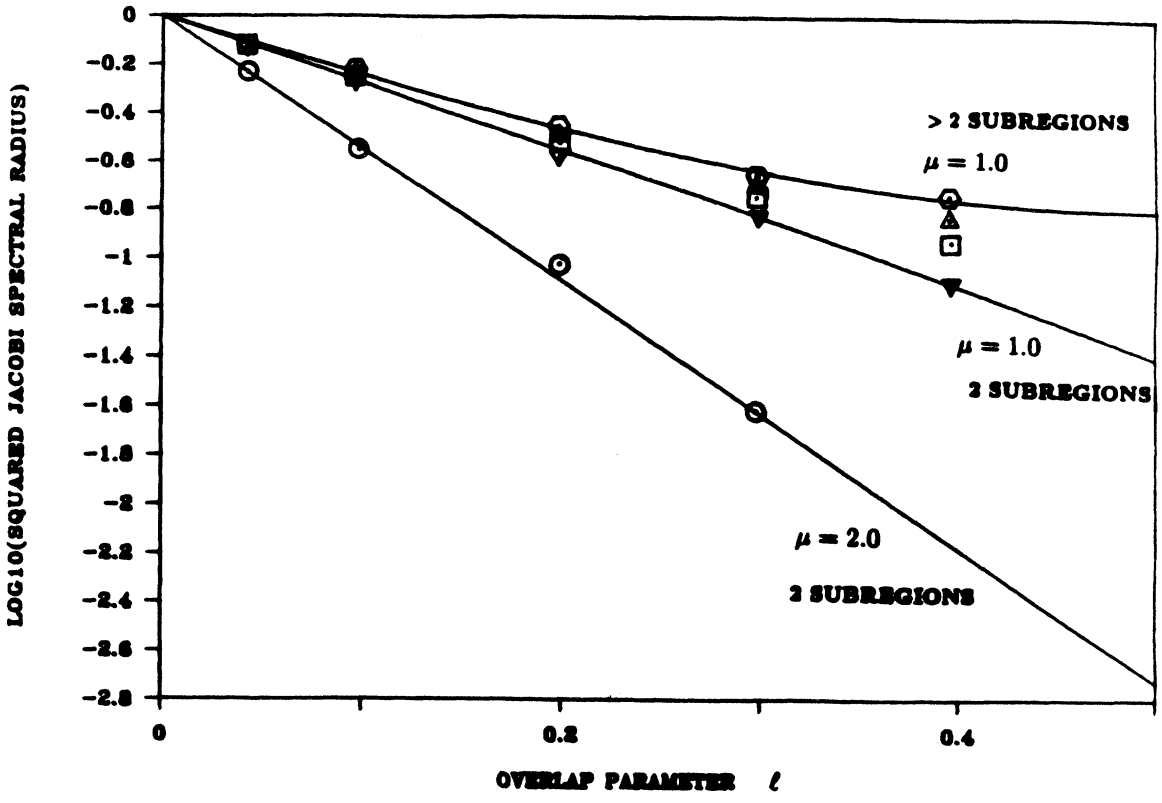


Figure 5.5: Theoretical and computational values of the squared spectral radius for the block Jacobi iteration matrix in the 2–D case.

Figure 5.5 [3] shows that the estimate derived from the Gershgorin theorem is quite accurate. The computational results (denoted by ⊙, △ etc..) are very close to the theoretical curve. Note that the curves are the asymptotic bounds of ρ_J.

From this theorem and its corollaries the following conclusions can be deduced:

- The convergence rate of S_M^M is a function of the overlap ratio κ. If κ is constant[4], then the convergence rate of S_M^M is independent of the mesh size. This is where the conclusion about optimal complexity comes from. If an optimal algorithm is used for solutions of these subregions, the total computational work required for achieving a fixed accuracy is proportional to the number of discrete unknowns.

[3] Increasing ℓ corresponds to increasing overlap. Note that the domain size increases with increasing overlap when the subregion size is held fixed.

[4] This means that κ is independent of the mesh size. We also assume here that the number of subregions is independent of h.

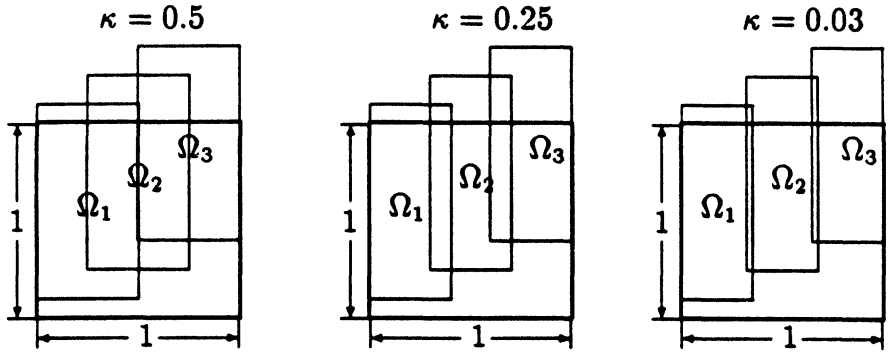


Figure 5.6: Three two–dimensional overlapping grids.

- The convergence rate of S_A^M is also a function of the shape of the subregions, specifically, a function of μ. If the subregions are thin in the overlapping direction (usually caused by increasing the number of subregions), then the convergence rate is slow. This suggests that we should avoid slicing the domain into many thin overlapping subdomains. A multidirection decomposition strategy is proposed in the next chapter.

- As in the one–dimensional case, the S_E^M of the matrix A has property $A^{(\pi)}$. Therefore, some classical acceleration schemes can be applied.

- The convergence factor decreases monotonically when the frequency increases. Furthermore, if the overlap is increased, the errors of high frequencies are damped exponentially faster than for the smaller overlap. The picture above shows three different overlapping grids. The corresponding table presents how the convergence factor ρ_i is changing when the the overlap and frequency are changed. The last column lists the number of iterations needed to reduce the errors of the corresponding Fourier components by a factor of 10^5. An important message which can be obtained from this table is that we should combine the strategies of increasing overlap and using multi–level grids. In the following chapter we will discuss the accelerating strategies in detail.

Frequency *i*	κ	Matrix D_i			Number of iterations
		a_i	A_i	ρ_i	
1	0.5	0.3794	0.3794	0.5366	18
	0.25	0.1987	0.6754	0.7684	44
	0.03	0.0256	0.9602	0.9729	420
6	0.5	0.0095	0.0095	0.0135	3
	0.25	0.0037	0.1555	0.1574	7
	0.03	$0.8058*10^{-4}$	0.8302	0.8306	62
11	0.5	0.0002	0.0002	0.0003	2
	0.25	$0.384U10^{-4}$	0.0338	0.0338	4
	0.03	$0.9612*10^{-6}$	0.7126	0.7126	34
16	0.5	$0.5067*10^{-5}$	$0.5067*10^{-5}$	$0.7166*10^{-5}$	1
	0.25	$0.4423*10^{-7}$	0.0076	0.0076	3
	0.03	$0.1039*10^{-8}$	0.6141	0.6141	24
21	0.5	$0.1409*10^{-7}$	$0.1409+10^{-7}$	$0.1987*10^{-6}$	1
	0.25	$0.5987*10^{-*}$	0.0018	0.0018	2
	0.03	$0.1215*10^{-}$	0.5320	0.5320	19

Table 5.1: Convergence factors for three two-dimensional overlapping grids.

The model problem in a uniform p-dimensional cube is as follows:

$$\sum_{i=1}^{p} \frac{\partial^2 u}{\partial x_i^2} = g.$$

As in the two-dimensional case, the cube is divided into k overlapping subcubes. Figure 5.7 shows a 3-dimensional cube and its decomposition. The subcubes axe shifted upwards to improve the visibility of the pattern of overlap.

The same approach is used for this problem as for the former case. Before discussing the analysis, some notation needs to be defined. Let

$$T_n^{(1)} = T_n = tridiagonal\{1, -2, 1\}_{n \times n}$$

and $I(\backslash n)$ be the $n \times n$ identity matrix. If there is no confusion, we will use $I^{(1)}$ instead of $I\hat{}(n)$. We can recursively define the matrix[5] derived from the model

[5] As in the two-dimensional case, a central difference scheme is used here.

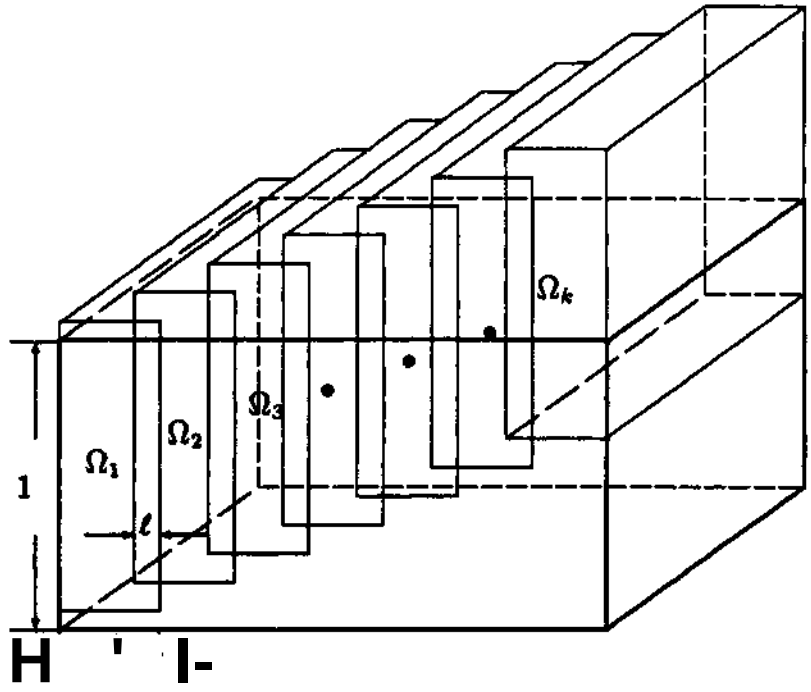


Figure 5.7: Three-dimensional overlapping grid.

problem in a p-dimensional cube as follows:

$$T_p^{(p)} = T^{(p-1)} \otimes I^{(1)} + I^{(p-1)} \otimes T_n^{(1)}, \qquad p = 2, \cdots,$$

where

$$I^{(p)} = I^{(p-1)} \otimes I^{(1)}.$$

Let

$$X^{(1)} = X_n$$

and

$$X^{(p)} = X^{(p-1)} \otimes X^{(1)}, \qquad p = 2, \cdots.$$

Then $X^{(p)}$ is the orthogonal matrix which diagonalizes the matrix $T_n^{(p)}$. That is :

$$X^{(p)} T_n^{(p)} (X^{(p)})^T = D^{(p)}$$

where $JD^{(P)}$ = *Diagonal* $\{-2p + 23c^{08}\copyright\wedge/\wedge + {}^{1})\}_{(n}$p-ixnP-i)$^{\#}$ Thus, the discretization of the p-dimensional model problem can be written as:

$$T_i^{p)} x = b.$$

The corresponding *fyjfi* and its Jacobi iterative matrix of the $\mathfrak{S}$ are as follows:

$$\tilde{T}_n^{(p)}\tilde{x} = \{W_m^{(p)} \otimes I^{(p-1)}(k) + (/(\text{P-D(n)} \ \mathfrak{g}\ \text{J5}_m) \ ® \ L_k + (/ \ \hat{}(n) \ ® \ \text{F}_m) \ ® \ U_k)\tilde{x}$$

$$= \tilde{b},$$

$$J^{(p)} = {}_{M}\text{-}i_N$$

$$= (W^{(p)} \otimes I^{(p-1)}(k))^{-1}((I^{(p-1)}(\text{n}) \ ® \ E_m) \otimes L_k + (I^{(p-1)}(\text{n}) \ ® \ F_m) \otimes U_k),$$

where

$$W_m^{(p)} = T_n^{(p-1)} \otimes I^{(1)}(m) + I^{(p-1)}(n) \otimes T_m^{(1)}.$$

A result which is very similar to that obtained in the two-dimensional case can be obtained as follows:

Theorem 5.3 *The matrix* jW *%s similar to the matrix*

$$\mathcal{X}\mathcal{T} = \begin{bmatrix} 0_{(n^p-(2nk)^{p-1})\times(n^p-(2nk)^{p-1})} & C_{(n^p-(2nk)^{p-1})\times(2nk)^{p-1}} \\ 0_{(2nk)^{p-1}\times(n^p-(2nk)^{p-1})} & G_{(2nk)^{p-1}\times(2nk)^{p-1}} \end{bmatrix}$$

where

$$G = Block\text{——}diagonal\{D\hat{}\},$$

$$D_\nu = D'_\nu \otimes I_k + E'_\nu \otimes L_k + F'_\nu \otimes U_k,$$

$$E'_\nu = \begin{bmatrix} a_v & 0 \\ 0 & 0 \end{bmatrix}, \quad D'_\nu = \begin{bmatrix} 0 & \beta_\nu \\ \beta_\iota, & 0 \end{bmatrix}, \quad F'_\nu = \begin{bmatrix} 0 & 0 \\ 0 & a_v \end{bmatrix}$$

$$\text{-} \ \frac{\sinh \kappa m\theta_\nu}{\sinh m\theta_\nu}, \quad \beta_\nu = \frac{\sinh(\text{l}-K)m\theta_\nu}{\sinh \text{m0,,}},$$

$$\cosh\theta_\nu = 2\text{p} - \sum \cos(\frac{LJT \quad \text{v}}{n+1}),$$

$$\text{!/} = (\text{!!}, \bullet.., \text{i}_\text{p}), \qquad \text{ti}_\text{f} \bullet\bullet\bullet, \text{t}_\text{p} = 1, \bullet\bullet\bullet, \text{n}.$$

The spectral radius of each D_u is the convergence factor for the corresponding Fourier component of the error in the approximation.

Proof. The proof is completely parallel with the two-dimensional case. We only need to change the Fourier transform matrix from $X\hat{}$ to $X(^{p}\sim^{l}K$

There is also a corresponding result for the asymptotic bound:

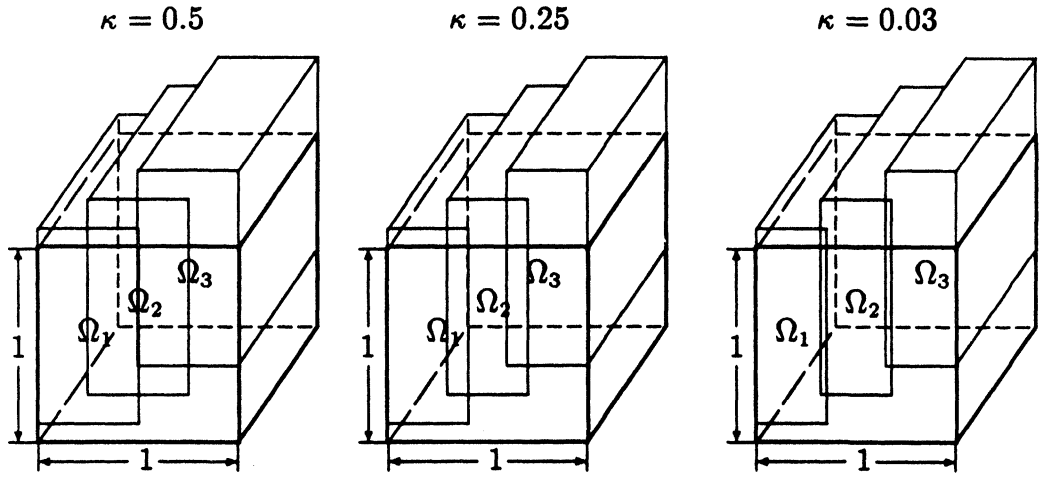


Figure 5.8: Three three-dimensional overlapping grids.

Theorem 5.4 *For the p-dimensional model problem the asymptotic bound for the spectral radius of the block Jacobi iterative matrix of the $\mathcal{SS}$ is:*

$$\rho \leq \frac{\sinh(\sqrt{p-1}\kappa\mu\pi) + \sinh(\sqrt{p-1}(1-\kappa)\mu\pi)}{\sinh(\sqrt{p-1}\mu\pi)}$$

The following picture and table present examples similar to those presented for the two–dimensional case. The same conclusions can also be found in higher–dimensional cases. If we compare this table with Table 5.1, an interesting observation is that the convergence rate of the higher–dimensional case is faster. Actually, we can derive this conclusion directly from Theorem 5.4. A more favorable result is that the errors in the higher frequency components damp even faster than in the two–dimensional case. Thus the strategy of a multi–level grid will be more successful.

5.3 Higher–Order Approximation Cases

In this section we will discuss the convergence behavior of $\mathcal{SAM}$ for higher–order approximations to separable elliptic PDE's.

Frequency $i=j$	κ	Matrix $D_{i,j}$			Number of iterations
		$\alpha_{i,j}$	$\beta_{i,j}$	$\rho_{i,j}$	
1	0.5	0.2998	0.2998	0.4239	14
	0.25	0.1622	0.6164	0.6923	32
	0.03	0.0218	0.9518	0.9627	302
6	0.5	0.0014	0.0014	0.0020	2
	0.25	$0.3748*10^{-3}$	0.0722	0.0724	5
	0.03	$0.9105*10^{-4}$	0.7689	0.7690	44
11	0.5	$0.6625*10^{-5}$	$0.6625*10^{-5}$	$0.9369*10^{-5}$	1
	0.25	$0.6101*10^{-6}$	0.0085	0.0085	3
	0.03	$0.1445*10^{-6}$	0.6207	0.6207	25
16	0.5	$0.3820*10^{-7}$	$0.3820*10^{-7}$	$0.5402*10^{-7}$	1
	0.25	$0.1255*10^{-8}$	0.0011	0.0011	2
	0.03	$0.2383*10^{-9}$	0.5050	0.5050	17
21	0.5	$0.2869*10^{-9}$	$0.2869*10^{-9}$	$0.4058*10^{-9}$	1
	0.25	$0.3543*10^{-11}$	$0.1524*10^{-3}$	$0.1524*10^{-3}$	2
	0.03	$0.5055*10^{-12}$	0.4153	0.4153	14

Table 5.2: Convergence factors for three three–dimensional overlapping grids.

The two–dimensional separable elliptic problem on the rectangle $[0,1] \times [0,1]$ may be stated as follows:

$$\frac{\partial}{\partial x}(p_1(x)\frac{\partial}{\partial x}U) + \frac{\partial}{\partial y}(p_2(y)\frac{\partial}{\partial y}U) + (q_1(x)+q_2(y))U = f(x,y),$$

$$U\,|_\Gamma = g(x,y).$$

where

$$p_1(x), p_2(y) \geq \varepsilon > 0$$

Here the unknown $U(x,y)$ will be approximated by tensor–product B-splines. When the Rayleigh–Ritz–Galerkin discretization using this approximation is applied to the above equation, it gives rise to a matrix equation

$$Ax = b,$$

where the matrix A is of the form

$$A = M_x \otimes S_y + S_x \otimes M_y.$$

If the B-splines are of order fc, then the matrices M_r, M_y, 5_X, S_y are $2k - 1$ banded matrices arising naturally in one-dimensional problems. Moreover, M_x, M_y are symmetric positive definite and 5*, S_y are symmetric semidefinite. We will not discuss the derivation of this system here. Interested readers can refer to the paper [KW84].

Let the decomposition of the solution region be the same as in the two-dimensional case. After a slightly more complicated derivation, we may obtain the $S\hat{}jM$ of the matrix A as follows:

$$\tilde{A} = \begin{bmatrix} W_1 & F_m & & & & & \\ E_m & W_2 & F_m & & & & \\ & \ddots & \ddots & \ddots & & 0 & \\ & & \ddots & \ddots & \ddots & & \\ & 0 & & \ddots & \ddots & \ddots & \\ & & & & E_m & W_{k-1} & F_m \\ & & & & & E_m & W_k \end{bmatrix}$$

where

$$Wi = Mi \circledR Sy + Si \otimes \mathrm{M}_y,$$

$$\mathrm{JZ}_{\Gamma}^{*} = J_{J\backslash\%}{}^{/0\backslash}{}_{xy}\ \delta_y\ \mathrm{T}^{i\cdot}\ \cdot r_{J2i} \otimes \mathrm{Af}_{y,}$$

$$Fi = U\backslash i \circledR Sy + U2\% \otimes \mathrm{My}.$$

Here the matrices $M_{t'}$ and $S\%$ are the B-spline matrices from the subregion H_i, while the matrices Ly and Uij are matrices with zero elements everywhere except for a lower or upper triangular matrix at the position $(l,m - / - d)$ or $(m,/)$ which is related to the boundary conditions on these artificial boundaries. Because the detailed definitions of these matrices depend on the particular approximation, we will not discuss them here. An example will be presented later.

In contrast with the case in the last section, we cannot use a Fourier transformation to diagonalize both matrices S_y and M_y here. Fortunately, a generalized eigensystem will do. Since M_y is symmetric and positive definite and S_y is symmetric, there exists a matrix Z such that

$$Z^T M_y Z = I_n,$$

$$Z^T S_y Z = D_n,$$

where I_n is the identity matrix and D_n is the diagonal matrix. The diagonal elements of D_n are the generalized eigenvalues λ of

$$S_y z = \lambda M_y z.$$

Now a similar approach can be applied to analyze the spectral radius of the Jacobi iterative matrix of the $\mathcal{SS}$. The matrix $Z \otimes I_m$ can be used to diagonalize the matrices M_y and S_y in $\tilde{A}$ such that:

$$\begin{aligned}
(Z^T \otimes I_n) W_i (Z \otimes I_m) &= M_i \otimes D_n + S_i \otimes I_n, \\
(Z^T \otimes I_n) E_i (Z \otimes I_m) &= L_{1i} \otimes D_n + L_{2i} \otimes I_n, \\
(Z^T \otimes I_n) F_i (Z \otimes I_m) &= U_{1i} \otimes D_n + U_{2i} \otimes I_n.
\end{aligned}$$

Let P be the permutation matrix such that $P(A \otimes B)P^T = B \otimes A$, then

$$\begin{aligned}
P(M_i \otimes D_n + S_i \otimes I_n)P^T &= Block - diagonal\{B_i\}, \\
P(L_{1i} \otimes D_n + L_{2i} \otimes I_n)P^T &= Block - diagonal\{\mathcal{L}_i\}, \\
P(U_{1i} \otimes D_n + U_{2i} \otimes I_n)P^T &= Block - diagonal\{\mathcal{U}_i\},
\end{aligned}$$

where $\{B_i\}_{m \times m}$ is a $2d-1$ banded matrix, $\{\mathcal{L}_i\}_{m \times m}$ is a matrix with zero elements everywhere except a $d \times d$ lower triangular submatrix in the position $(1, m-l-d)$, $\{\mathcal{U}_i\}_{m \times m}$ is a matrix with zero elements everywhere except a $d \times d$ upper triangular submatrix in the position (m, l). Now, following the same approach as in the proof of Theorem 5.2, we can prove the following theorem:

Theorem 5.5 *The Jacobi iterative matrix of the $\mathcal{SS}$ is similar to the matrix*

$$D = \begin{bmatrix} 0_{(n^2-2ndk)\times(n^2-2ndk)} & C_{(n^2-2ndk)\times 2ndk} \\ 0_{2ndk\times(n^2-2ndk)} & G_{2ndk\times 2ndk} \end{bmatrix}$$

where

$$G = Block - diagonal\{D_i\}, \quad i = 1, \cdots, n,$$

$$D_i = D_i' \otimes I_k + E_i' \otimes L_k + F_i' \otimes U_k,$$

$$E_i' = \begin{bmatrix} \mathcal{R}_i & 0 \\ 0 & 0 \end{bmatrix}, \quad D_i' = \begin{bmatrix} 0 & \mathcal{Q}_i \\ \mathcal{Q}_i & 0 \end{bmatrix}, \quad F_i' = \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{R}_i \end{bmatrix},$$

$$\mathcal{R}_i = \mathcal{S}_l B_i^{-1} \mathcal{L}_i,$$

$$\mathcal{Q}_i = \mathcal{T}_l B_i^{-1} \mathcal{U}_i,$$

$$\mathcal{S}_l = [\, 0, \cdots, 0, I_k, \overbrace{0, \cdots, 0}^{l} \,]_{k \times m},$$

$$\mathcal{T}_l = [\, 0, \cdots, 0, I_k, \overbrace{0, \cdots, 0}^{l} \,]^T_{m \times k}.$$

Let ρ_i be the spectral radius of the D_i; then each ρ_i is the convergence factor of the error component of the corresponding generalized eigenvalue.

Here we cannot present a general quantitative estimate for ρ_i, but a similar qualitative result such as the results in the last few sections is also true in this case. When we increase the overlap, I_k will move leftward or upwards in $\mathcal{S}_l$ or $\mathcal{T}_l$ ($l = \ell/h$). By the exponential decay law in the last chapter, the norm of $\mathcal{R}_i$ and $\mathcal{Q}_i$ will exponentially decay. If the overlap ratio κ is independent of the mesh size h, then these norms are also independent of h, as can be seen in the following example.

Strictly speaking, the nine–point stencil is not derived from the tensor product B–spline. Since the matrix derived from the nine-point stencil has the simplest tensor product form and also has higher–order accuracy, we present it as an example, discussing the convergence behavior when $\mathcal{SAM}$ is applied to this problem.

The matrix equation derived from the nine–point stencil on a unit square is as follows:

$$(M_x \otimes S_y + S_x \otimes M_y)x = b,$$

where

$$\begin{aligned} M_x &= Tridiagonal\{1, \ -2, \ 1\}, \\ S_x &= Tridiagonal\{1, \ 4, \ 1\}, \\ M_y &= I_n, \\ S_y &= Tridiagonal\{6, \ -12, \ 6\}. \end{aligned}$$

Frequency i	κ	Matrix D_i			Number of iterations
		α_i	β_i	ρ_i	
1	0.5	0.3794	0.3794	0.5366	18
	0.25	0.1987	0.6754	0.7684	44
	0.03	0.0256	0.9602	0.9729	420
6	0.5	0.0094	0.0094	0.0135	3
	0.25	0.0036	0.1547	0.1565	7
	0.03	$0.7939*10^{-4}$	0.8298	0.8298	62
11	0.5	$0.1928*10^{-2}$	$0.1928*10^{-2}$	$0.2726*10^{-2}$	2
	0.25	$0.3481*10^{-4}$	0.0327	0.0327	4
	0.03	$0.8709*10^{-6}$	0.7103	0.7123	34
16	0.5	$0.3950*10^{-5}$	$0.3950*10^{-5}$	$0.5585*10^{-5}$	1
	0.25	$0.3280*10^{-7}$	0.0069	0.0069	3
	0.03	$0.7641*10^{-8}$	0.6086	0.6086	24
21	0.5	$0.8092*10^{-7}$	$0.8092*10^{-7}$	$0.1144*10^{-5}$	1
	0.25	$0.3088*10^{-9}$	0.0015	0.0015	2
	0.03	$0.6097*10^{-9}$	0.5240	0.5240	18

Table 5.3: Convergence factors for three two–dimensional overlapping grids using nine–point stencil.

Since S_y is an identity matrix, the generalized eigenvectors are the same as the Fourier components. Applying the above theorem to this matrix equation we have

$$\mathcal{R}_i = \frac{\sinh \kappa m\theta_i}{\sinh m\theta_i}, \quad \mathcal{Q}_i = \frac{\sinh (1-\kappa) m\theta_i}{\sinh m\theta_i},$$

$$\cosh \theta_i = \frac{5 - 2\cos\frac{i\pi}{n+1}}{2 + \cos\frac{i\pi}{n+1}}, \quad i = 1, \cdots, n\ .$$

It is not very difficult to see that the higher–order approximation has the same asymptotic bound for ρ_J as in the Corollary 2 of the last section. The above table lists the convergence factors for different frequencies and the number of iterations. The decompositions are the same as in Figure 5.6. Compare this table with Table 5.1. The iteration counts are exactly the same except the last one. But for the higher frequency errors the convergence factors are slightly better. We can also prove this conclusion by comparing the $\mathcal{R}_i$, $\mathcal{Q}_i$ with α_i and β_i.

Chapter 6

Acceleration of the Convergence and Numerical Experiments

The discussion of the last chapter has provided insight into the behavior of SAM. Particularly, some possibilities for further improvement in the performance of this method have been mentioned. In this chapter, detailed discussions of the acceleration strategies are presented. Section 1 discusses the SOR (Successive Over Relaxation) acceleration. For the model problem, the classical theory of SOR can be applied here directly. Both theoretical and experimental results show that the improvement is significant. In order for a parallel algorithm to be efficient, global communications should be avoided as much as possible. Here a local relaxation scheme is discussed, and a general convergence proof for this acceleration is shown providing a theoretical basis for the scheme. Section 2 discusses the application of other classical acceleration schemes. Since we have obtained the eigenstructure of the iterative matrix of the plain 4^, many acceleration schemes for *QM* can be analyzed. Particularly, optimal Chebychev acceleration is studied here. There are other powerful accelerations schemes, such as preconditioned conjugate gradient methods which are not mentioned here. This is not a oversight. They are very good iterative methods for a conventional computer, but they require a global information exchange in every iteration and introduce a lot of communication overhead, and the parallel efficiency degenerates. If a new technology, which can reduce the high cost of the global information exchange, appears in the future, these CG types of accelerations will certainly be very interesting for further study. In Section 3, hierarchical computation is discussed. Combining it with the other accelerations makes *$lM* a competitive parallel iterative method for real applications. The remaining sections discuss several other issues which are important in the use of SAM

for real applications, namely decomposition considerations, solution methods for the subregions and convergence checking. Due to lack of much real experience on the new parallel computers and the rapid advances in hardware and software, further studies are needed for these topics.

6.1 SOR Acceleration and Multi-Color Splitting

Among the many possible acceleration methods, the SOR acceleration is an attractive choice. It is easy to implement and its theoretical background is well understood. The local communication pattern of this method is also an appealing feature For parallel computation.

As we discussed in the last chapter, $\mathcal{SM}$ is actually the following block Gauss–Seidel iteration [1]:

$$(E_m \otimes L_k + T_m \otimes I_k)\tilde{x}^{(k+1)} = -(F_m \otimes U_k)\tilde{x}^{(k)} + \tilde{b}. \tag{6.1}$$

Thus, an obvious choice for an acceleration scheme is the SOR acceleration. We can construct a new approximation $\hat{x}^{(k+1)}$,

$$\hat{x}^{(k+1)} = \omega\tilde{x}^{(k+1)} + (1-\omega)\tilde{x}^{(k)},$$

and then attempt to choose an optimal relaxation parameter ω to speed up the convergence. Since the connections between the subregions merely involve artificial boundary values, the relaxations are carried out only for those boundaries. Late in this section we will present a general convergence result: for any choice of the relaxation parameter between 0 and 2, this scheme will converge to the true solution. But as we know, SOR cannot be successfully applied to an arbitrary matrix. The following famous example is due to Kahan [Kah58] :

$$K = \begin{bmatrix} 1 & -a & 0 \\ 0 & 1 & -a \\ -a & 0 & 1 \end{bmatrix},$$

[1] Here we exhibit the case of a one–dimensional model problem.

where $0 < a < 1$. This matrix is an M–matrix and is positive definite. A short calculation will show that the optimal relaxation factor is $\omega = 1$, which is equivalent to saying that SOR acceleration does not help in this case. Some further restrictions on the iterative matrix are needed. For instance, the well known property A and the consistent ordering of the iterative matrix will guarantee a successful relaxation iteration. Fortunately, we are able to construct an algorithm which satisfies these restrictions on the iteration matrix of the multi–color S_4M.

Let us start with some simpler cases. Since we have found the eigenstructure of the iterative matrix for the model problem, the analysis of the application of SOR to the model problem is straightforward. As we see from equation (5.2), this S_EM has property $A^{(\pi)}$ [2]. It is easy to verify that natural ordering and red–black ordering of the subregions will both lead to a consistent ordering in the matrix. Therefore, the classical analysis of the SOR theory can be applied here directly! We have calculated the spectral radius ρ of the Jacobi iterative matrices for the model problems in any dimension, the optimal relaxation factor can be calculated from the following formula:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2}}$$

Table 6.1 lists some comparisons between plain S_4M and its SOR acceleration for the one–dimensional model problem. The third column of this table lists the number of iterations needed to reduce the error by a factor of 10^5 for plain S_4M, while the fourth column lists the same quantities for SOR acceleration with the optimal relaxation factor. The last two columns are the experimental and theoretical optimal relaxation factors for the same cases. We can see that they agree very well. There is a detailed discussion and the results of many experiments are presented in the paper [OST86], which we will not repeat here. As we see in this table, the improvement of the SOR acceleration is significant. For the higher–dimensional problems, we could also make a similar table using the spectral radius we obtained in the last chapter. The next several sections will discuss interesting issues which appear when this method is applied to more general cases.

[2] Property $A^{(\pi)}$ is an extension of Young's famous property A to the block matrix case. Since we consider only block matrices in this chapter, we will later omit the adjective *block*.

SOR Acceleration Results for 1–D Model Problem					
Number of subregions	Overlap ratio	Plain SAM	SAM + SOR	Optimal ω observed	Optimal ω from theory
2	0.333	14	7	1.2	1.15
	0.182	27	10	1.3	1.27
	0.095	52	14	1.42	1.40
	0.039	129	22	1.60	1.57
	0.019	258	30	1.68	1.67
3	0.427	19	10	1.3	1.21
	0.309	26	10	1.3	1.28
	0.250	37	12	1.38	1.33
	0.175	66	16	1.48	1.44
	0.071	141	21	1.62	1.58
	0.030	349	35	1.71	1.71
4	0.333	44	12	1.42	1.36
	0.220	70	16	1.5	1.46
	0.167	97	18	1.52	1.51
	0.083	204	26	1.67	1.63
5	0.333	88	17	1.55	1.50
	0.190	169	26	1.62	1.61
	0.167	193	26	1.65	1.64
	0.083	397	36	1.75	1.74

6.1.1 Multi-Color SAM and Consistent Ordering

As we mentioned above, some restrictions are needed to ensure the success of the relaxation iteration. A well known candidate class of matrices is those which have property $A^{(\pi)}$, or more generally, block p–cyclic matrices. An $n \times n$ matrix A is p–cyclic if there is an permutation matrix P such that PAP^T is of the following

form:

$$PAP^{T} = \begin{bmatrix} A_{1,1} & 0 & \cdots & 0 & M_{1,p} \\ A_{2,1} & A_{2,2} & & 0 & 0 \\ 0 & A_{3,2} & \ddots & 0 & 0 \\ & & \ddots & & \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & A_{p,p-1} & A_{p,p} \end{bmatrix}.$$

But simply having the property $A^{\wedge}$ or being a p-cyclic matrix is not enough. Consistent ordering is also needed. A detailed discussion of these concepts is given in Varga's book [Var62], For the model problem which is decomposed in strip fashion, these requirements are automatically satisfied. The block-tridiagonal structure of the iterative matrix in (4.2) has property $A^{\wedge}$ and is consistently ordered. But, the inherent dependence in the natural ordering of the equation (5.2) prevents an efficient parallel implementation. Instead, red-black ordering is commonly applied. For a general solution region, the decomposition has to be carefully implemented in order to meet these requirements. The multi-color *QM* proposed in Chapter 2 is a way to obtain a block p-cyclic matrix. If we impose an extra restriction on the decomposition such that:

$$\Omega^{(l)} = \bigcup_{j=1}^{i_l} \Omega_j^{(l)} = \Omega, \qquad l = 1, \cdots, k,$$

where k is the number of colors, and i_l is the number of subregions in color l, then the blocks which correspond to a particular color only need to be connected to the previous color in the solution order. It is not difficult to see that the *SEM* for this splitting is a consistently ordered p-cyclic matrix. When p = 2, the p-cyclic matrix is a block 2-cyclic matrix, which is usually called a red-black ordered block matrix. The advantage of the multi-color splitting is the parallelism inherent in this decomposition. Subregions which have the same color can be computed independently. In previous chapters we also mentioned the strategy of locating the artificial boundaries near the middle of other subregions in order to maximize the reduction of the error on these boundaries. If we group the subregions into only

two colors, this requirement is very hard to achieve. This is the major reason we are motivated to propose the multi-color splitting.

6.1.2 The Sensitivity of the Relaxation Factor to the Overlap

The analysis of the relationship between the spectral radius p and the overrelaxation factor uj shows that when p is close to 1 the rate of convergence is much more sensitive to changes in the relaxation factor. A slight perturbation of the relaxation factor can result in a big degradation in performance. This is not good for practical implementations. Since the spectral radius is exponentially related to the overlap, the sensitivity of the rate of convergence drops dramatically if the overlap increases (see Figure 6.1). Although the increased overlap causes more work in each iteration, the total work is still less than it is for a small overlap. The work per iteration increases linearly with the overlap, while the spectral radius decreases exponentially. In Figure 6.1 the relation between the number of iterations and the relaxation factor is shown. These results are all for the two-dimensional model problem in a unit square. We divide the square into 5 overlapping subregions. The six curves in this figure correspond to six overlap patterns, which have different overlapping ratios. As we see in this figure, for the smallest overlap the performance of the method is tremendously sensitive to the choice of the relaxation factor.

This figure strongly suggests a need to increase the overlap. Now a natural question to raise is how to choose the best overlap ratio for a given number of processors. Let us study the two-dimensional model problem again. We divide the unit square successively into 2, 3, 4, 6, 8, 10 overlapping subregions. Then for each case we vary the overlap ratio from 0 to 0.5. Using the spectral analysis in the last chapter, we can calculate the total work needed to reduce the error by a factor of 10^5. Figure 6.2 shows the relation between the overlap and the total work for these six cases. From this figure we can see that although the spectral radius will be minimum for an overlap ratio of 0.5, in terms of the total work the optimal overlap ratio is somewhat less than 0.5. When K decreases, the spectral radius increases. But, if the change of overlap is small the number of iterations needed for reducing the norm of the error by a fixed factor does not change. Thus, the total work will

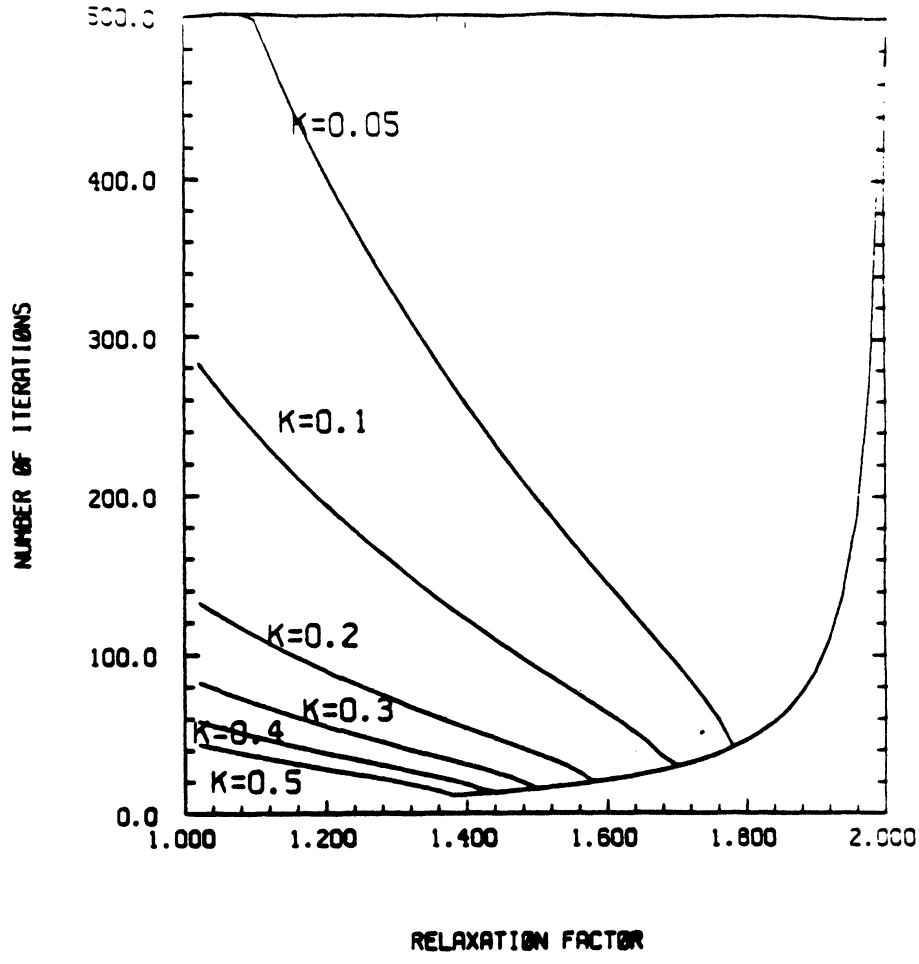


Figure 6.1: The number of iterations as a function of the overrelaxation factor ω

decrease until the number of iterations jumps. That is the reason why these curves show saw-tooth shapes. As the number of the processors increases, however, the optimal overlap ratio will approach 0.5.

6.1.3 A Local Relaxation Strategy

SOR acceleration has a very efficient parallel implementation, but unfortunately, the estimate of the relaxation factor still requires global information exchange in general. This is a well known problem which causes the parallel efficiency to degrade.

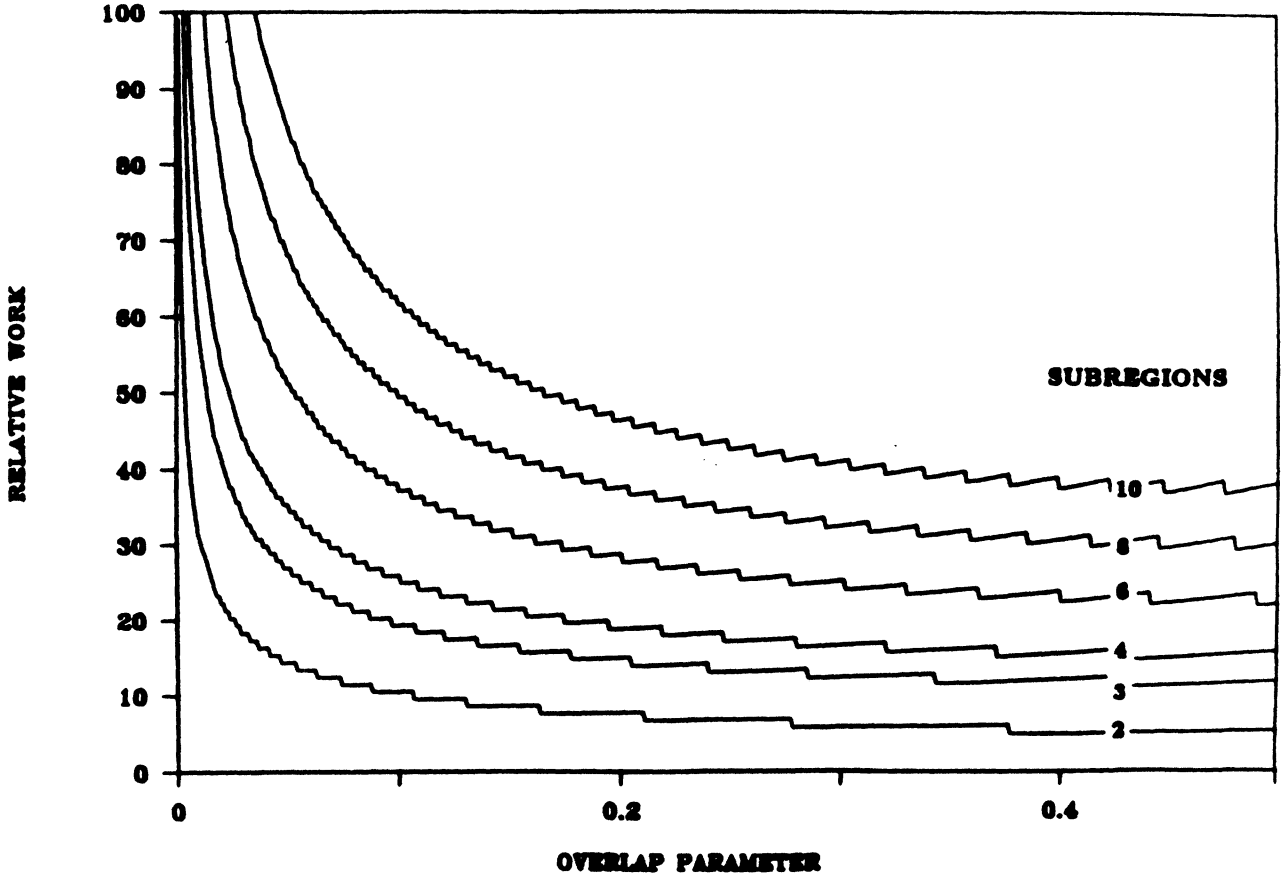


Figure 6.2: The total work as a function of the overlap ratio κ

Recently, a new technique[3] which uses local relaxation factors has received some attention. The motivation for this technique was to try to find a more efficient relaxation technique for irregular meshes or variable coefficient problems which could avoid estimating of the spectral radius of the Jacobi matrix and obtain more error reduction than the uniform relaxation factor. The reasoning behind this technique is very convincing. Since the relaxation is a local operation, the relaxation factor should also be characterized well by local features. Experience has shown that this idea works well for many test problems. Of late, the locality of the communication in this method has obtained the attention of the parallel computation community. C. Kuo, B. Lever and B. Musicus [KLM86] apply this idea to a mesh-connected array.

The basic idea of the local relaxation method is to determine a relaxation factor for each individual grid point. Consider a five-point difference equation

$$CLgXg + a_n x_n + a_e x_e + a_w x_w + a_o x_o = b_0$$

where x_0 is a grid function located at the position (i,j) [4], £,, x_n, x_{ej} and x_w are the grid functions located to the south, north, east and west of x_0, respectively, and a_0, a,, a_n, a_c, a_w are the corresponding coefficients. Suppose that there are N and M grid points in the row i and column j in which x_0 is located. Now we may imagine that there is an N x M rectangular grid and that each grid point has the same difference equation as x_0. Then the spectral radius of the Jacobi matrix for this problem is

$$PJ = \frac{a_o}{} - [\text{vOTT COS} \frac{}{N+1} \cdot + \text{v}\wedge \text{COS} \frac{\pi}{M+1}].$$

Therefore, the optimal relaxation factor for this imaginary rectangular grid is:

$$\omega_{opt} = \frac{2}{1+\sqrt{1-\rho_J^2}}.$$

We will use this u_{opi} as the local relaxation factor for grid point x_0. We can obtain different relaxation factors for each grid point which are only related to the local

[3] It is also called the ad-hoc SOR method. See [Erh81], [Erh84]

[4] The solution region need not be rectangular.

features surrounding the point in this way. When the solution region is rectangular and the coefficients of the PDE are constant, the local relaxation factor is the same as the global optimal relaxation factor. Experimental results show that this relaxation scheme is successful for many test problems.

The same idea can also be applied to the SOR acceleration for $\mathcal{SAM}$. The calculation of the spectral radius for the model problem can be easily generalized to a general second order elliptic PDE with constant coefficients. As we have seen in the last chapter, the estimate of the spectral radius only involves the overlap ratio and information about the shape of the subregions, both of which are local information. If we want to estimate a local relaxation factor for an artificial boundary, which is located in some other subregion, we may imagine two overlapping rectangular subregions and let the shapes of these rectangular regions be as close to the real ones as possible. Then we may use the estimate of the relaxation factor for the rectangular regions as the relaxation factor. Thus, global information exchange can be avoided. In the next subsection we will prove that, for any choice of ω between 0 and 2 for each subregion, the iteration will converge.

This local relaxation method has been successful experimentally. Theoretical analysis of the relationship between these relaxation factors and the convergence rate remains a very interesting open problem.

6.1.4 The Convergence Proof for Local Relaxation

In Chapter 2 a multi-color $\mathcal{SAM}$ for elliptic PDE's was introduced and an extension to a positive definite matrix, called multi-color $\mathcal{SS}$, was mentioned. Here the detailed definition of this splitting is presented. Combining it with the local relaxation method, we may prove the following theorem:

Theorem 6.1 *When the multi-color $\mathcal{SAM}$ is applied to a positive definite matrix, if every $\omega_j^{(i)}$, which is used as the relaxation factor for block j of color i, satisfies*

$$0 < \omega_j^{(i)} < 2,$$

then the relaxation process of the multi-color $\mathcal{SS}$ converges.

Proof. The basic idea of this proof was given three decades ago by Ostrowski [Ost56], who used the same idea to prove convergence of the group relaxations. The new aspect here is that we combine two old techniques, $\mathcal{SYM}$ and group relaxation, to obtain an efficient parallel implementation.

Let

$$Ax = b \tag{6.2}$$

be the linear system of equations, where A is a positive definite matrix. The multi-color $\mathcal{SOR}$ with local relaxation can be described as follows: First we find p permutation matrices P_l, $l = 1, \cdots, p$ such that matrix A can be permuted to p different partitioned matrices, where p is the number of colors:

$$\begin{aligned}
A = \; & P_1 \begin{bmatrix} A^{(1)}_{1,1} & A^{(1)}_{1,2} & & A^{(1)}_{1,k_1} \\ A^{(1)}_{2,1} & A^{(1)}_{2,2} & & A^{(1)}_{2,k_1} \\ \vdots & & \ddots & \vdots \\ A^{(1)}_{k_1,1} & A^{(1)}_{k_1,2} & & A^{(1)}_{k_1,k_1} \end{bmatrix} P_1^T = P_1 A^{(1)} P_1^T \\
= \; & \qquad\qquad \vdots \qquad\qquad\qquad\qquad = \; \vdots \\
= \; & P_p \begin{bmatrix} A^{(p)}_{1,1} & A^{(p)}_{1,2} & & A^{(p)}_{1,k_p} \\ A^{(p)}_{2,1} & A^{(p)}_{2,2} & & A^{(p)}_{2,k_p} \\ \vdots & & \ddots & \vdots \\ A^{(p)}_{k_p,1} & A^{(p)}_{k_p,2} & & A^{(p)}_{k_p,k_p} \end{bmatrix} P_p^T = P_p A^{(p)} P_p^T .
\end{aligned}$$

Local SOR relaxation is then applied to each of the blocks $A^{(l)}_{i,i}$ as follows:

$$A^{(l)}_{i,i} X^{(k+1)}_i = \omega^{(l)}_i \left[-\sum_{j<i} A^{(l)}_{i,j} X^{(k+1)}_i - \sum_{j>i} A^{(l)}_{i,j} X^{(k)}_i + b^{(l)}_i \right] + (1 - \omega^{(l)}_i) A^{(l)}_{i,i} X^{(k)}_i .$$

The motivation for this algorithm came from the fact that the original block relaxation (or group relaxation) still suffered from slow convergence. After we studied the inverse structure of the sparse matrices in Chapter 3, we noticed that the error decay rates for different variables in the same block differed greatly. The plain block relaxation failed to take advantage of the exponential decay of the inverse of a sparse matrix. The multi-color $\mathcal{SOR}$ tries to put every variable within the fast decay area of some color (geometrically, we may say "near the middle of some subregion").

As we show in the model problem cases, it eventually yields a method which has optimal complexity.

The proof of convergence is straightforward. It is well known that for each positive definite operator there is a functional

$$F(X) = (AX,X)-2(b,X)$$

which corresponds to the system of equations, such that the vector which achieves the minimum of this functional is the solution of this system. The iteration process can be viewed as the process of minimizing this functional.

Let us consider each calculation of a block as one step of the algorithm. It is easy to verify that the decrease in the value of the functional for two consecutive iterations is as follows:

$$F(X_i^{(k+1)}) - F(X_i^{(k)}) = -\omega_i^{(l)}(2 - \omega_i^{(l)})(r_i^{(k)}, A_{i,i}^{(l)^{-1}} r_i^{(k)})$$

where rj** is the residual of block *i* in color /. If every $u^{\wedge}$ satisfies $0 < u;f^* < 2$, the sequence of $F(Xi)^{(k)}$ monotonically decreases. Using arguments of Ostrowski, we can prove that this sequence will converge to the solution of (6.2). This concludes our proof of convergence.

6.2 Other Classical Acceleration Schemes

Applying *$5* to equation (3.12) we have

$$\begin{aligned} \widetilde{M}_s \tilde{x}^{(k+1)} &= \widetilde{N}_s \tilde{x}^{(k)} + f, \\ \tilde{x}^{(k+1)} &= \tilde{G}_s \tilde{x}^{(k)} + \widetilde{M}_s^{-1} f, \end{aligned}$$

where $\tilde{G}_s = \widetilde{M}_s^{-1}\widetilde{N}_s$. This is a typical form of the *basic iteration.* Many acceleration schemes for this iteration are available. There is an excellent survey and comparison for them in L. Hageman and D. Young's book, *Applied Iterative Methods.* We will not repeat their comparison here. As we mentioned in the beginning of this chapter, some of these methods are not discussed here due to the high cost of the communication overhead in a parallel computer environment. Among these

different schemes, the popular Chebychev acceleration is rather interesting in this case. The adaptive Chebychev procedure still suffers from the high cost of the heavy global information exchange in each iteration, but if we have some knowledge of the eigenvalues of the iterative matrix $\tilde{G}_s$, then optimal Chebychev acceleration is very attractive. Particularly, for the model problem a detailed analysis of the eigenvalues for the $\mathcal{S}$ is available. The application of the optimal Chebychev acceleration to the model problem is therefore straightforward. Let

$$\begin{aligned} m(A) &= \min \lambda_A, \\ M(A) &= \max \lambda_A. \end{aligned}$$

From Chapter 4, we have

$$m(\tilde{G}_s) = -M(\tilde{G}_s)$$

and

$$M(\tilde{G}_s) \leq \frac{\sinh(\kappa\mu\pi) + \sinh((1-\kappa)\mu\pi)}{\sinh(\mu\pi)}. \tag{6.3}$$

The test results in the last chapter show that the bound (6.3) is very accurate[5]. Moreover, all eigenvalues of the matrix $\tilde{G}_s$ are real provided that the overlap ratio $\kappa < 0.5$[6]. Applying the estimate (6.3) to the classical formula for the convergence rate for optimal Chebychev acceleration, we can expect this acceleration to yield an improvement similar to that offered by the SOR acceleration. Unfortunately, this result cannot be generalized to the other cases as the SOR acceleration can. First of all, there is no local Chebychev acceleration available, and the classical adaptive Chebychev procedure requires an extensive global information exchange. Secondly, the eigenvalues of the iterative matrix $\tilde{G}_s$ can be complex in general. Chebychev acceleration can only be applied to some of the complex eigenvalue cases. How to apply the Chebychev acceleration to a general problem is still an interesting open problem in some sense.

[5]If the result in Theorem 4.1 is used, the exact value of $M(\tilde{G}_s)$ can be estimate by some numerical computations.

[6]There is no advantage in making $\kappa > 0.5$. We need not consider this case.

6.3 Hierarchical Computation

During the last ten years, very important advances in computational science have been made in the area of the hierarchical computation. Among the best known techniques are multigrid techniques[Bra77], adaptive grid methods[Oli84], hierarchical information flow[Oli86]. Although, their approaches, theoretical foundations and applications are very different, one idea behind these techniques is the same. According to the particular application, the computational process is decomposed into several different phases, regions or grids, which we will abstract as a hierarchy. Instead of using one uniform approach for the whole problem, we treat each component of the hierarchy separately, attempting to choose the most efficient way of obtaining the result in that component. The components of the hierarchy will communicate with each other, and after some assembling or iterations, the final result can be obtained in a very efficient way. To use a business expression, we might say that we are only willing to pay what we have to pay. This same philosophy can even be applied to the design of the computer hardware and programming languages. If the designs of the computer and language are "smart" enough, it is certainly worthwhile to run an algorithm in such a way that in the different stages of the computation different precision of arithmetic are used. There is no point in using double precision when the iteration has just started. The same idea can certainly be adapted to the acceleration of S_4M. In Chapters 2 and 4 we have studied the convergence rates of S_4M for different frequencies for both continuous and discrete cases. An important observation from the analyses is that the slow convergence is caused by the low frequency errors. Table 4.1, which lists the number of the iterations required for reducing the error corresponding to particular frequencies by a factor of 10^5, strongly suggests that we should start the computation at the coarse grid. After the low frequency errors converge to the truncation error level at this grid, we should then refine the grid and continue the computation. This procedure can be recursively repeated until the results of the desired accuracy are obtained. We have used the model problem to test this idea. Our results show that the cost is substantially reduced.

The Numbers of Iterations in Each Grid Level							
Number of subregions	Grid Levels (5 is the Finest Grid)						The Equivalent Numbers of Iterations on the Finest Level
	1	2	3	4	5	Total	
5	3	1	1	1	1	7	1.5
10	7	3	2	JL.	2	16	2.7

Table 6.1: Hierarchical computations

The test problem is the two-dimensional model problem on a unit square. The finest grid is [320 x 320]. There are 5 grid levels. The mesh size of each level is double that of the previous one. The iteration starts at the coarsest grid. After the iteration converges to the level of truncated error at this particular grid size, we refine the grid and continue the iteration on the next finer grid, and so on. Table 6.1 lists the number of iterations carried out on each grid level for two different decompositions. The last column lists the total work, measured as the equivalent number of iterations on the finest grid. Although the total number of *fyM* iterations remains the same as it would have been for a single fine grid, The total work needed is reduced to a small fraction of what it would have been.

6.4 Decomposition Considerations

In the early 1950's, Kantorovich and Krylov had noticed that the way the solution region was decomposed would affect the rate of convergence. In our analyses of the model problems we have seen that the rate of convergence is a function of the overlap, the shape of the subregions, the frequency of the errors and the dimension of the solution regions. The first important issue in the consideration of decomposition is the overlap. For the model problem, the overlap can be characterized by a simple quantity *K* (overlap ratio). But, in a general application, *K* can no longer be used for this purpose. In Chapter 5, the exponential decay law was seen to be the reason for the success of ^*M*. From this law, we recognize that the rates of convergence for different variables (or grid points) in *$\M*iteration are very different.

The errors on the artificial boundaries will affect the error in each point by the influence wavefront of the out–web in the inverse operator. Ideally, we would like arrange the subregions in such a way that all the artificial boundaries are located in the "*heart*" of another subregion or subregions[7], in order to maximize the decay of the error in each iteration. For the model problem, this can be easily achieved by setting the overlap ratio κ to 0.5. In general, we need the multi–color splitting to achieve this goal.

Another related issue is the shape of the subregions. For the model problem, the ratio m/n is a very important influence on the convergence rate, where m and n are the height and width of the subregion, respectively. If we would like to partition the solution region into many subregions, we should not dissect the region in only one direction(such as in the strip case). A one–direction dissection would result in having many thin, long subregions, leaving the artificial boundaries very close to the boundaries of those thin subregions. This principle is also applicable to general cases. The subregions should have comparable dimensions in every coordinate. Any small width in one coordinate will result in a short influence wavefront in the inverse of the operator on the thin subregion, causing slow convergence as we saw in Chapter 3. A good way of decomposing the solution region is to dissect the solution region in k directions, where k is the dimension of the solution region. Figure 6.3 shows a dissection in two directions for a two–dimensional problem. A comparison is carried out for two kinds of dissection. The first case decomposes the square into 32 thin strips. The second one is to decompose the square into 36 rectangles. The first one needs 40 iterations while the latter only needs 15 iterations. Even though the latter case has almost twice as many variables in comparison to the former case, the overall work in the two direction dissection is only three fourths of the other one.

Another interesting issue in the decomposition of the solution region is the automation of the dissection. In principal, this problem is similar to the grid generation problem for the finite element method. There is no intrinsic difficulty in this

[7]The motivation is to put all the artificial boundaries in the quickly converging zone. Then, by the maximum principle, after one iteration the total error will be bounded by the error on these boundaries.

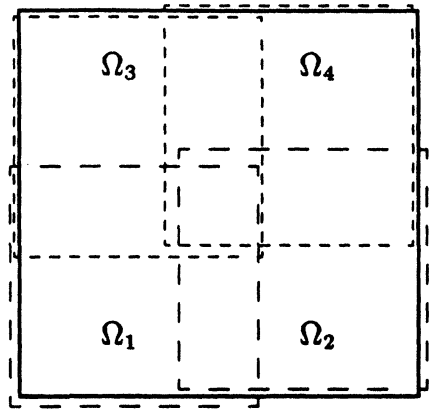


Figure 6.3: A dissection in two directions

problem.

The last issue is the mapping of the decomposition to a parallel computer architecture. Since the efficient communication pattern of the target computer will strongly affect the choice of the decomposition, it is a very hardware dependent issue. We did not include such mappings in this study, but if we would like to make S_4M a really competitive method, they should be carefully treated.

6.5 Solution Methods for the Subregions

The choice of methods for the solution of the problem on the subregions is also an important issue for applications of S_4M. Because of the inherent modularity in the S_4M algorithm, each subregion can be solved using a different method. Depending on the particular application, we may take advantage of this flexibility. For example, we can use a fast solver or even an analytic method, to compute the solution on a regular subregion. Direct and iterative methods each have their own advantages and disadvantages. Iterative methods are generally preferred. This is because, at any step, the result from the last iteration is a very good initial guess for the next iteration. But this does not mean that iterative methods always win. We have

compared the multigrid method with the fast solver for the model problem. In fact, the fast solver wins in the timing comparison. Comparison of the two programs will show that this result should not be a surprise. The multigrid method has more overhead than the fast solver has. The advantage in complexity for the multigrid method only becomes dominant in a very large problem. For sparse problems, if there is enough memory for storing the LU decomposition in each processor, a sparse solver can also be very competitive. The complexity of the work in each iteration is only *O(N)* in this case, where *N* is the number of unknowns.

A strategy of incomplete solution in solving on the subregions has been tried. The basic idea of this strategy is that we really do not need a very accurate solution on the subregions in the early iterations. If an iterative method is used for the solution on the subregions, we can ask if we can stop the iteration at some point before the solution converges? The preliminary results are disappointing. For example, we have applied the multigrid method to solve the subproblems. If the number of V-cycles or W-cycles for solving the subproblems is reduced, the rate of convergence of *§)M* immediately degenerates. The total work needed to converge is also increased. G. Rodrigue has also had a similar experience. Further study is needed on this question. It seems likely that a way can be found to successfully use incomplete solutions.

6.6 Convergence Checking

Until communication cost became an important factor for the performance of a parallel algorithm, convergence checking was never an efficiency issue in implementing an iterative algorithm. Due to the requirement of global information exchange and control, the convergence check in an iterative algorithm has to be carefully implemented. The granularity of *SfaM* is very desirable in this aspect since coarse granularity results in a low frequency of convergence checking.

In addition, the hierarchical computation in *$M* can also be used to reduce the cost of a global convergence checking. It is clear that we dp not need a global check of convergence until the finest grid is reached. During the computation on the coarser

grids, we need only check the local error for each subregion (or processor). Each subregion will keep checking two things: the local error and the maximum update of the variable in the subregion. If the local error is good enough for a particular subregion, the corresponding process can be put to sleep [8]. If the maximum update is large, then any of the neighbor processes which are sleeping should be awakened, since the boundary values of these neighbors have changed significantly. All these information exchanges of convergence information are local and can be combined with the exchange of boundary values. There are no extra communication requests required for the exchange. Thus, the cost of global checking will only be required on the finest grid. As our experience shows that only one or two iterations are needed on this grid, the overall cost of communication is greatly reduced. In general, the global error checking is still a very interesting research problem for any iterative method in a parallel computer environment. Some hardware design considerations can be very helpful in resolving the efficiency problem. For example, if the control processor can check a built-in flag in each processor at a very low cost, then the cost of the checking can be substantially reduced.

[8] In a time sharing system the processor on which the sleeping process was running can be recycled by the system

Chapter 7

Conclusion

This thesis has reexamined and generalized an old mathematical technique — Schwarz alternating method (SAM). Through the convergence analysis for the model problem, the governing factors for the convergence of SAM are explored. Using this knowledge, the performance of SAM can be significantly improved. As a concrete example of the improvement in performance, let's apply SAM to the model problem

$$\begin{aligned} \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} &= f(x,y), && (x,y) \in (0,1) \times (0,1), \\ U\,|_\Gamma &= g(x,y) \end{aligned}$$

where the five–point stencil is used and the mesh size is 1/320. In Table 7.1 we summarize the results from five different ways of applying SAM to this problem. For each implementation, we list the number of iterations and the total relative work needed in reducing the norm of the error by a factor of 10^5, as well as the convergence factor. In the first approach, the unit square is divided into 5 strips and each strip overlaps with its neighbors only by one mesh width. The Jacobi type of SAM with natural ordering is applied. As we might expect, the convergence is very slow. The second approach is to increase the overlap to the optimum, namely, each strip now overlaps with its neighbor by half of its width. The same Jacobi iteration is used. The exponential relationship between the convergence factor and the overlap makes a big improvement in the performance. Next, Gauss–Seidel SAM is applied. The convergence speed is doubled. Then SOR acceleration is incorporated. We list a result for which the optimal ω is used. Again the performance is improved further. Finally, a multilevel grid technique, with five grid levels, is combined with SOR acceleration. The combination of these four modifications yields a significant improvement in performance. As we show in Chapter 6, multi–direction

decomposition can lead to further improvement.

Iteration technique	Convergence factor	Number of iterations	Total relative work
$\mathcal{SAM}$ with minimal overlap	0.99308	1658	1658
$\mathcal{SAM}$ with optimal overlap	0.90097	110	183.33
Gauss–Seidel $\mathcal{SAM}$	0.8117	55	91.67
SOR + $\mathcal{SAM}$ (with optimal ω)	0.395	13	21
Multi–level grid + SOR + $\mathcal{SAM}$	0.395	13	2.1

Table 7.1: A comparison of 5 different implementations of SAM

We have incorporated several acceleration strategies in this example. An important factor for practical application is that these accelerations do not interfere with each other. The various freedoms in $\mathcal{SAM}$ which we mentioned in the introduction allow us to combine many other techniques to improve the performance when we apply $\mathcal{SAM}$ to a particular problem. Particularly, generalizations of $\mathcal{SAM}$, *Schwarz splittings* ($\mathcal{SS}$), are introduced in this thesis. Thus, we can apply this powerful technique to many important applications other than elliptic PDE's.

There is an increasing demand for parallel algorithms, the inherent parallelism, the local communication pattern and the hiding of global information exchange make $\mathcal{SAM}$ an attractive candidate for large scale computations on a parallel computer with non–shared memory. A generalization of $\mathcal{SAM}$— multi–color $\mathcal{SAM}$ — is presented. It preserves the parallelism of the original $\mathcal{SAM}$, while provides a fast convergence. Many parallel implementation issues such as: local relaxation strategy; convergence checking; carrying the exchange of boundary values at coarser grid level even after the computation has proceeded to finer grid level are discussed in this thesis. We also propose some open problems which should be further investigated.

In Chapter 4 we discussed the problems caused by the matrix structure. The abstract form of a matrix creates difficulties for observing many important features of a linear operator. A new structure *template operator*, which is more consistent with the form of the original continuous operator than the matrix is, has been developed. Using this new structure, we have presented the concepts of *influencing* and *influenced wavefronts* which provide tools for quantitatively describing the exponential

decay phenomenon. Several estimates for the exponential decay are shown in this study providing a theoretical basis for determining when $\mathcal{SS}$ can be used successfully.

Although $\mathcal{SAM}$ is an very *old* mathematical technique, the understanding of this approach is still *young*. Particularly, computational experience is very limited. Our study has presented a promising but preliminary investigation. Interesting open problems remain to be solved. We have seen increasing interest in this topic among numerical analysts, and expect $\mathcal{SS}$ to become a competitive and popular iterative technique.

Appendix A

The Eigenvalues of the matrix W_n

Let

$$W_n = \begin{bmatrix} A_1 & A_2 & & & & \\ A_0 & A_1 & A_2 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & A_0 & A_1 & A_2 \\ & & & & A_0 & A_1 \end{bmatrix}, \tag{A.1}$$

where

$$A_0 = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & b \\ b & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 0 \\ 0 & a \end{bmatrix},$$

and $b, a > 0$. We will discuss the calculation of the eigenvalues and eigenvectors for this matrix in this appendix.

Before we calculate the eigenvalues of this matrix, the following result is useful for later discussion.

Lemma A.1 *If $b \geq a$ then*

$$a + b \geq | \lambda_{W_n} | \geq b - a,$$

where λ_{W_n} is any eigenvalue of matrix W_n.

Proof. The left half of the inequality can be derived directly from Gershgorin's theorem. Let

$$z = (z_1, \cdots, z_{2n})^T$$

be the corresponding eigenvector of λ_{W_n} and

$$z_k = \max_i \{| z_i |\} > 0.$$

For $k = 2$ or $k = 2n - 1$, we have

$$bz_2 = \lambda w_n z_1,$$

or

$$bz_{2n-1} = \lambda w_n z_{2n}.$$

So,

$$|\lambda w_n| \geq b > b - a.$$

For $2 < k < (2n - 1)$, we have[1]

$$az_{k-2} + bz_k = \lambda w_n z_{k-1}.$$

Then,

$$\lambda w_n \frac{z_{k-1}}{z_k} - a\frac{z_{k-2}}{z_k} = b.$$

Finally, we have

$$|\lambda w_n| \geq b - a.$$

The eigenvalue and eigenvector problem for W_n is equivalent to the boundary value problem of the matrix difference equation:

$$\begin{cases} A_0 Z_k + (A_1 - \rho I)Z_{k+1} + A_2 Z_{k+2} = 0, & k = 1, \cdots, n, \\ Z_0 = Z_{n+1} = 0 \end{cases} \qquad (A.2)$$

where ρ is an eigenvalue of W_n. It can be solved easily by the nonmonic matrix polynomial theory. Here we will use the same notation in Gohberg's book [GLR82]. It is interesting that the spectral theory of the general matrix polynomials $L(\lambda)$ is, surprisingly, of very recent origin.

---

[1] If $k = 1$ or $k = 2n$ we have

$$bz_2 = \lambda w_n z_1,$$

or

$$bz_{2n-1} = \lambda w_n z_{2n}.$$

The nonmonic matrix polynomial[2]

$$L(\lambda) = A_2\lambda^2 + (A_1 - \rho I)\lambda + A_0$$

corresponds to the matrix difference equation (A.2). The solution of (A.2) can be expressed in terms of a decomposable pair of $L(\lambda)$. Let (X_F, J_F) and (X_∞, J_∞) be a finite and an infinite Jordan pair respectively. The decomposable pair of matrix polynomials $L(\lambda)$ is

$$([X_F, X_\infty], J_F \oplus J_\infty).$$

From Theorem 8.3 in [GLR82] we know that the general solution of the homogeneous finite difference equation (A.2) is given by

$$Z_k = X_F J_F^k g, \qquad k = 0, 1, \cdots. \tag{A.3}$$

A short calculation shows that

$$\det(L(\lambda)) = -\lambda(a\rho\lambda^2 - (a^2 + \rho^2 - b^2)\lambda + a\rho).$$

Then the eigenvalues of the $L(\lambda)$ are

$$\begin{aligned}
\lambda_0 &= 0, \\
\lambda_1 &= \frac{a^2 + \rho^2 - b^2 + \sqrt{((b-a)^2 - \rho^2)((a+b) - \rho^2)}}{2a\rho}, \\
\lambda_2 &= \frac{a^2 + \rho^2 - b^2 - \sqrt{((b-a)^2 - \rho^2)((a+b) - \rho^2)}}{2a\rho}.
\end{aligned}$$

We know that $\lambda_1, \lambda_2 \neq 0$. The eigenvectors of $L(\lambda)$ corresponding to the eigenvalues λ_i and $L(\lambda_i)$ are

$$x_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad L(\lambda_0) = \begin{bmatrix} a & 0 \\ 0 & 0 \end{bmatrix}$$

[2]A matrix polynomial

$$L(\lambda) = \sum_{i=0}^{k} A_i \lambda^i$$

is said to be monic if $A_k = I$, otherwise it is called nonmonic. Here A_i are $m \times m$ matrices.

$$X\backslash = \begin{bmatrix} 1 \\ \frac{Q\backslash\backslash - a}{b\lambda_1} \end{bmatrix} = \begin{bmatrix} 1 \\ \omega_1 \end{bmatrix}, \quad L(\lambda_{\text{ll}}) = \begin{bmatrix} a - p A i & 6A1 \\ 6A1 & aAi^2 - pAi \end{bmatrix}$$

$$x_2 = \begin{bmatrix} 1 \\ \frac{\rho\lambda_2 - a}{6A_2} \end{bmatrix} = \begin{bmatrix} 1 \\ \omega_2 \end{bmatrix}, \quad L\{\backslash^*) = \begin{bmatrix} a - jpA_2 & 6A_2 \\ 6A_2 & aA_2^2 - /0A2 \end{bmatrix}$$

where

$$\omega_1 = \frac{\rho\lambda_1 - a}{b\lambda_1}, \qquad u;_2 = \frac{pA_2 - a}{6A2}$$

Since the Jordan chains which correspond to these eigenvalues all have only one eigenvector each, the finite Jordan pair is as follows:

$$X_F = \begin{bmatrix} 0 & 1 & 1 \\ 1 & <JJ\backslash & U>2 \end{bmatrix}$$

$$J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \backslash i & 0 \\ 0 & 0 & A_2 \end{bmatrix}$$

Now we may use the general solution and the boundary value to determine the eigenvalues and eigenvectors of matrix W_n. Note

$$A_0 X_F = \begin{bmatrix} 0 & a & a \\ 0 & 0 & 0 \end{bmatrix},$$

$$(A_1 - pI)X_F J_F = \begin{bmatrix} 0 & \lambda_1(u>i_L 6 - p) & A_2(w_2 6 - \sim P)' \\ 0 & \lambda_1(b-- \rho\omega_1) & A_3(6 - pa>_2) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -a & -a \\ 0 & -\lambda_1^2 a\omega_1 & -A_2^2 aa;_2 \end{bmatrix},$$

$$A_2 X_F J_F = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \lambda_1^2 aw_1 & AJau>_2 \end{bmatrix},$$

and

$$AQX_F + (A\backslash - pI)X_F J_F + A_2 X_F J_F^2 = 0\text{-}$$

Thus, the general solution (A.3) does satisfy the matrix difference equation (A.2). Now let's determine the constant vector $g = (go,9u92)^T$ to satisfy the boundary

conditions. For the first boundary condition

$$X_F g = 0,$$

we have

$$g_1 = -g_2$$

and

$$g_0 = \frac{a(\lambda_1 - \lambda_2)}{b\lambda_1\lambda_2} g_1$$

From the second boundary condition

$$X_F {J_F}^{n+1} g = 0,$$

λ_1 and λ_2 have to satisfy the following condition:

$$\begin{cases} \lambda_1\lambda_2 = 1 \\ \omega_1{\lambda_1}^{n+1} = \omega_2{\lambda_2}^{n+1} \end{cases} \tag{A.4}$$

The first equality is satisfied from the definition. If $b \geq a$, then λ_1 and λ_2 are complex. the second implies the following condition:

$$a \sin n\theta = \rho \sin (n+1)\theta, \tag{A.5}$$

where

$$\lambda_1 = e^{i\theta}. \tag{A.6}$$

Theorem A.1 *The eigenvalues of the matrix W_n satisfy the following equations:*

$${\lambda_{W_n}}^2 + 2a\cos\theta\lambda_{W_n} + a^2 - b^2 = 0. \tag{A.7}$$

The corresponding eigenvectors are

$$X_F(\theta)[\ J_F(\theta), (J_F(\theta))^2, \cdots, (J_F(\theta))^n\ ]^T g(\theta),$$

where θ is the root of the following equation:

$$\left(\cos\theta + \sqrt{\left(\frac{b}{a}\right)^2 - \sin^2\theta}\right) = \frac{\sin(n\theta)}{\sin((n+1)\theta)}. \tag{A.8}$$

Proof. Equation (A.7) can be directly derived from the definition of λ_1. Then we may solve λ_{W_n} in (A.7) and substitute it into (A.5). After rearranging the terms of $\sin((n+1)\theta)$ and a, (A.8) follows.

Corollary 6 *If $a \leq b$, matrix W_n has $2n$ distinguish real eigenvalues. When n is increased, ρ is also increased*[3].

Let

$$\rho = \max\{|\lambda_{W_n}|\}.$$

A short calculation shows that if $n = 1$

$$\rho = b, \tag{A.9}$$

and if $n = 2$

$$\rho = \sqrt{b(a+b)}. \tag{A.10}$$

Equations (A.9) and (A.10) are true for any a and b.

[3]Here a and b are fixed.

Appendix B

Extensions of the Template Operator

Here we present a few extensions of the template operator introduced in Chapter 4. First, the template operator L over 7^, where $s > 1$, is discussed. Then a more generalized operator -*block template operator*- is considered. In the last section, other kinds of operations on the template vectors using template operators are presented.

B.I Template Operator over T_n*

In Chapter 4, the simplest case of the template operator, L on template space $T^{\wedge 1}$, is examined. A more general case, the template operator on 7^, where $s > 1$, is considered here.

Given a template vector space 7^*, $s > 1$, the template operator space over this space is defined as follows: let

$$T_n = \ll o_1, o_2, \cdots, o_n \gg$$

be the template of $T_n *_y M_{\S}$ be the space of all s x s matrices. Construct n Cartesian products

$$Ni = M_9 \text{ x } o_t; i = 1, \bullet \bullet \bullet, n.$$

Let

$$M^9{}_n = N_x \text{ x } N_2 \text{ x } \bullet \bullet \bullet \text{ x } JV_W.$$

Each element U 6 M^*_n consists of n ordered pairs

where M_{Oi}, i = 1, • • •, n are *sxs* matrices. After defining the operations of addition and scalar product on these elements, it is easy to see that the space M^{9}_{n} is a linear space. We call it a **template matrix space.** Each element m G M'_{n} is called a template matrix. Now construct n Cartesian products

$$Q_i = \mathcal{M}_n^s \times o_i, i = 1, \cdots, n.$$

Let

$$r = Q_1 x Q_2 x \ldots x Q_n .$$

Each element L G C* consists of n ordered pairs

$$L = [< R_U Ox >, < R_2, O_2 >, \bullet\bullet, < Rny\ O_n >]/,$$

or simply

$$L = [R_{0b}\ R^{\wedge}\ ,^{*}\ \bullet\ \bullet,\ <R_{0n}]/.$$

As in the case when 5 = 1, a **template operator space** C* over 7^* with the operations of addition and scalar product can be defined.

Let

and

$$x = \{< xi,oi >, < s_2,O2 >, \bullet\bullet\bullet, < \wedge n > o_n >\}\ G\ TJ.$$

Define the product of R and x as

$$Rx = \sum_{o_i \in O} R_{o_i} x_{o_i}$$

where $R_{Oi}x_{Oi}$ is the product of an *s* x *s* matrix R_{Oi} and a vector x_{Oi}. Here the result Rx is an ^-dimensional vector. Now we can define the operation of a template operator L G C* on **a** template vector a? 6 T_n'. Let

$$L = [R_{o_1}, R_{o_2}, \cdots, R_{o_n}]_l.$$

We define

$$y = Ix = \{R_{o_1}x, R_{o_2}x, \cdots, R_{o_n}x\}.$$

From this definition, we know that $y = Lx \in \mathcal{T}_n^s$ and L maps $\mathcal{T}_n^s$ to $\mathcal{T}_n^s$. We may also introduce the concepts of operating template matrix, image template matrix, right form of the operator and so on. Since they are so similar to the case of $s = 1$, we may leave them to the reader.

B.2 Group Template Operator

In many cases, we need to group the nodes in a template into a few sets, each including several nodes. The number of the nodes can be very different. Schwarz splitting is a good example of an application of this idea. An alternative view is that each node in a template is associated with a state vector which may have different dimensions in different nodes. Here the group template operator is introduced from the second point of view.

Given a template T_n, and n vector spaces V_i, $i = 1, \cdots, n$, where the dimension of V_i is k_i, construct n Cartesian products

$$S_i = V_i \times o_i, i = 1, \cdots, n.$$

The **group template vector space** $\mathcal{G}_n$ is the set

$$\mathcal{G}_n = S_1 \times S_2 \times \cdots \times S_n$$

with two operations —addition and scalar product. Each element

$$x = (< x_1, o_1 >, < x_2, o_2 >, \cdots, < x_n, o_n >)$$

or

$$x = \{x_{o_1}, x_{o_2}, \cdots, x_{o_n}\}$$

in $\mathcal{G}_n$ is called a **group template vector.** Let $\mathcal{M}_{i,j}$ denote the space of all $k_i \times k_j$ matrices. A **group template matrix space** Q_i is the set

$$Q_i = \{\mathcal{M}_{i,1} \times o_1\} \times \{\mathcal{M}_{i,2} \times o_2\} \times \cdots \times \{\mathcal{M}_{i,n} \times o_n\}$$

with two operations —addition and scalar product—. Each element $R \in Q_i$ is n ordered pairs

$$R = \langle\langle < M_1, o_1 >, < M_2, o_2 >, \cdots, < M_n, o_n > \rangle\rangle$$

where M_{o_i} is a $k_i \times k_j$ matrix. Define the *product* of a group template matrix R and a group template vector x as follows:

$$Rx = \sum_{o_j \in \mathcal{O}} M_{o_j} x_{o_j}.$$

The product is a vector of dimension k_i.

Then the **group template operator space** $\mathcal{L}_g$ is the set

$$\mathcal{L}_g = \{Q_1 \times o_1\} \times \{Q_2 \times o_2\} \times \cdots \times \{Q_n \times o_n\}$$

with two operations —addition and scalar product. Each element $L \in \mathcal{L}_g$ is n ordered pairs

$$L = [< R_1, o_1 >, < R_2, o_2 >, \cdots, < R_n, o_n >]_l,$$

or simply

$$L = [R_{o_1}, R_{o_2}, \cdots, R_{o_n}]_l$$

where R_{o_i} is a group template matrix in Q_i. The definition of operating a group template operator L on a group template vector x is the following:

$$y = Lx = \{R_{o_1}x, R_{o_2}x, \cdots, R_{o_n}x\}$$

where $R_{o_i}x$, the product of the group template matrix R_{o_i} and x, is a vector of dimension k_i. The group template operator is a parallel concept of the partitioned matrix in the matrix structure. As in the last section, we leave many of the definitions to the reader.

B.3 A Template Operator Maps $\mathcal{T}_n$ into $\mathcal{T}_m$

So far the template operators we have discussed are mappings for which the domain and range are the same space. Here a more generalized template operator which

maps one template space $\mathcal{T}_n^s$ into another $\mathcal{T}_m^s$ is presented. In order to simplify the notation, we only consider the simplest case, that is the case when $s = 1$.

Let

$$T_n = \ll o_1, o_2, \cdots, o_n \gg$$

and

$$T_m = \ll \tilde{o}_1, \tilde{o}_2, \cdots, \tilde{o}_m \gg$$

be the templates for $\mathcal{T}_n$ and $\mathcal{T}_m$, respectively. The left form of a **template operator** L which maps $\mathcal{T}_n$ into $\mathcal{T}_m$ is

$$L = [< R_1, \tilde{o}_1 >, < R_2, \tilde{o}_2 >, \cdots, < R_m, \tilde{o}_m >]_l$$

where $R_{\tilde{o}_i}$ is the **operating template** for node $\tilde{o}_i$ and is a template vector in $\mathcal{T}_n$. The corresponding right form of this operator is

$$L = [< C_1, o_1 >, < C_2, o_2 >, \cdots, < C_n, o_n >]_r$$

where $C_{o_i} = LI_{o_i}$, the **image template** for node o_i, is a template vector in $\mathcal{T}_m$. We also leave the rest of the definitions for this case to the reader.

Bibliography

[Bra77] A. Brandt. Multi-level adaptive solution to boundary value problems. *Math. Comp.*, 31:333–391, 1977.

[CH62] R. Courant and D. Hilbert. *Methods of Mathematical Physics.* Volume 2, Willey, New York, 1962.

[dB80] Carl de Boor. Dichotomies for band matrices. *SIAM J. Numer. Anal.*, 17:894–907, 1980.

[Dem77] S. Demko. Inverses of band matrices and local convergence of spline projections. *SIAM J. Numer. Anal.*, 14(4):616–619, 1977.

[Deu85] J. T. Deutsch. *Algorithms and Architecture for Multiprocessor-Based Circuit Simulation.* PhD thesis, University of California, Berkeley, Electronics Research Laboratory, U. C. Berkeley, 1985.

[DGP80] R. V. Dinh, R. Glowinski, and J. Periaux. *Applications of Domain Decomposition Techniques to the Numerical Solution of the Navier-Stokes Equations.* Technical Report, INF-LAB, France, 1980.

[DJa62] E. G. D'Jakonov. A method for solving Poisson's equation. *Soviet Math. Dokl.*, 3:320–323, 1962.

[DMS84] S. Demko, W. Moss, and P. Smith. Decay rates for inverses of band matrices. *Mathematics of Computation*, 43(168):491–499, 1984.

[Dup67] T. Dupont. *On the Existence of an Iterative Method for the Solution of Elliptic Difference Equation with an Improved Work Estimate.* Technical Report, Centro Tenuto Internationale Mathmatico Estivo, Corso Tenuto a Ispra, 1967.

[Erh81] L. W. Erhlich. An ad hoc SOR method. *Journal of Computational Physics*, 44:31–45, 1981.

[Erh84] L. W. Erhlich. *The Ad-hoc SOR Method: a Local Relaxation Scheme*, pages 257–269. *Elliptic Problem Solvers II*, Academic Press, edition, 1984.

[FM66] G. Fairweather and A. Mitchell. Some computational results of an improved a.d.i. method for the dirichlet problem. *Computer J.*, 9:298–307, 1966.

[GDP80] R. Glowinski, Q. V. Dinh, and J. Periaux. *Domain Decomposition Methods for Nonlinear Problems in Fluid Dynamics.* Technical Report, INRIA, France, 1980.

[GLR82] I. Gohberg, P. Lancaster, and L. Rodman. *Matrix Polynomials.* Academic Press, 1982.

[Gor34a] A. Gorgidze. The method of successive approximations applied to the plane problem of the theory of elasticity. *Dok. Ak. Nauk SSSR*, 4(5), 1934.

[Gor34b] A. Gorgidze. The method of successive approximations applied to the plane problem of the theory of elasticity. *Dok. Ak. Nauk SSSR*, 4(6), 1934.

[Kah58] W. Kahan. *Gauss-Seidel Methods of Solving Large Systems of Linear Equations.* PhD thesis, University of Toronto, Toronto, Canada, 1958.

[KCSQ85] L. Kang, Y. Chen, L. Sun, and H. Quan. *A Class of New Asynchronous Parallel Algorithms for Solving Partial Differential Equations.* Technical Report 19, Wuhan University, Wuhan, China, 1985.

[Ker70] D. Kershaw. Inequalities on the elements of the inverse of a certain tri–diagonal matrix. *Math. Comput.*, 24:155–158, 1970.

[KK58] L. V. Kantorovich and V. I. Krylov. *Approximate Methods of Higher Analysis.* P. Noordhoff LTD-Groningen-The Netherlands, 4th edition, 1958.

[KLM86] C. C. Kuo, C. B. Levy, and B. R. Musicus. *A Local Relaxation Method for Solving Elliptic PDEs on Mesh-Connected Arrays.* Technical Report, Massachusetts Institute of Technology, Cambridge, MA., 1986.

[Kuz87] Yn. A Kuznetsov. *Domain Decomposition Algorithms for Fast Approximate Implementation of Crank-Nicolson Schemes.* Technical Report, Department of Numerical Mathematics, USSR Academy of Sciences, Moscow, USSR, 1987.

[KW84] L. Kaufman and D. Warner. High-order, fast-direct methods for separable elliptic equations. *SIAM J. Numer. Anal,* 21:672-694, 1984.

[Mei86] U. Meier. *Two Parallel SOR Variants of the Schwarz Alternating Procedure.* Technical Report, Zentralinstitut fűr Angewandte Mathematik, 5170 Jiilich -W. Germany, 1986.

[Mik34] S. Mikhlin. The method of successive approximations applied to the biharmonic problem. *Trudy Seism,* 34:1-14, 1934.

[Mil65] K. Miller. Numerical analogs to the Schwarz alternating procedure. *Numerische Mathematik,* 7:91-103, 1965.

[Neu70] C. Neumann. Zur theorie des logrithmischen und Neuton'schen potentials. *Leipziger Berichte,* 22:264-321, 1870.

[Oli84] Joseph Oliger. Adaptive grid methods for hyperbolic partial differential equations. In F. Santosa, Y. H. Pao, W. W. Symes, and C. Holland, editors, *Inverse Problems for Acoustic and Elastic Waves,* SIAM, 1984.

[OH86] Joseph Oliger. Parallelism and uncertainty in scientific computations. In *International Congress on Computational and Applied Mathematics,* University of Leuven, Belgium, 1986.

[Ost56] A. M. Ostrowski. On the linear iteration procedure for symmetric matrices. ***Rend, mat. e applic,* 14:140-163, 1956.**

[OST86] Joseph Oliger, William Skamarock, and Wei-Pai Tang. *Schwarz Alternating* ***Method and Its SOR accelerations.* Technical Report, Stanford** University, Computer Science Dept., 1986.

[Poi90] H. Poincare. Sur les equations aux derivees partielles de la physique mathematique. *Amer. J. Math.,* 12:211-294, 1890.

[Rod86] G. Rodrigue. Inner/outer iterative methods and numerical Schwaxz algorithms. ***Journal of Parallel Computing,* 2:, 1986.**

[RS84a] G. Rodrigue and J. Simon. A generalization of the numerical schwarz algorithm. In R. Glowinski and J. Lions, editors, *Computing Methods* ***in Applied Sciences and Engineering VI,* pages 273-283, North-Holland,** Amsterdam-New York-Oxford, 1984.

[RS84b] G. Rodrigue and J. Simon. Jacobi splitting and the method of overlapping domains for solving elliptic PDE's,. In R. Vichnevetsky and R. **Stepleman, editors, *Advances in Computer Methods for Partial Differential* Equations *V,*** pages 383-386, IMACS, 1984.

[RS85] G. Rodrigue and P. Saylor. Inner/outer iterative methods and numerical **Schwarz algorithms-ii. In *Proceedings of the IBM Conference on Vector and Parallel Computations for Scientific Computing,* page , IBM, 1985.**

[Sch69] H. A. Schwarz. Ueber einige abbildungsaufgaben. *Jour. f. die reine und angew. Math,* 70:105-120, 1869.

[Sob36] S. L. Soboleff. L'algorithme de Schwarz dans la théorie de relasticité (Schwarz's algorithm in the theory of elasticity). *Doklady, A. N. S.S.S.R.,* 2(3):235-238, 1936.

[ST82] K. Stüben and U. Trottenberg. *Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications*, pages 1–176. Volume 960, Spriger-Verlag, Berlin–Heidelberg–New York, 1982.

[Sta77] G. Starius. Composite mesh difference methods for elliptic boundary value problems. *Numerische Mathematik*, 28:243–258, 1977.

[Sto72] D. Stoutemyer. *Numerical Implementation of the Schwarz Alternating Procedure for Elliptic Partial Differential Equations.* PhD thesis, Stanford University, Stanford, CA 94305, 1972.

[Var62] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, Englewood Cliffs, 1962.

[Vol68] E. Volkov. *The Method of Composite Meshes for Finite and infinite Regions with Piecewise Smooth Boundary.* Technical Report 96, Steklov Institute of Mathematics, 1968.

[Wer60] H. Werner. *Schwarz's alternating method for boundary value problems of the third kind.* Technical Report, University of Southern California, 1960.

[Wer63] H. Werner. Anwendungen und fehlerabschaetzungen fuer das alternierende verfahren von H. A. Schwarz. *ZAMM*, 43:55–61, 1963.

[You71] D. M. Young. *Iterative solution of large linear system.* Academic Press, 1971.