USC-SIPI REPORT #150

Discretization and Solution of Elliptic PDEs - A Digital Signal Processing Approach

by

C.-C. Jay Kuo and Bernard C. Levy

December 1989

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
Powell Hall of Engineering
University Park/MC-0272
Los Angeles, CA 90089 U.S.A.

December 1, 1989

Discretization and Solution of Elliptic PDEs - A Digital Signal Processing Approach[†]

C.-C. JAY KUO[*]

BERNARD C. LEVY[**]

Abstract

A digital signal processing (DSP) approach is used to study numerical methods for discretizing and solving linear elliptic partial differential equations (PDEs). Whereas conventional PDE analysis techniques rely on matrix analysis and on a space-domain point of view to study the performance of solution methods, the DSP approach described here relies on frequency domain analysis and on multidimensional DSP techniques. This tutorial paper discusses both discretization schemes and solution methods. In the area of discretization, mode-dependent finite-difference schemes for general second-order elliptic PDEs are examined, and are illustrated by considering the Poisson, Helmholtz and convection-diffusion equations as examples. In the area of solution methods, we focus on methods applicable to self-adjoint positive definite elliptic PDEs. Both direct and iterative methods are discussed, which include fast Poisson solvers, elementary and accelerated relaxation methods, multigrid methods, preconditioned conjugate gradient methods and domain decomposition techniques. In addition to describing these methods in a DSP setting, an up-to-date survey of recent developments is also provided.

[†] The first author was supported by the USC Faculty Research and Innovation Fund. The second author was supported by the National Science Foundation under Grant ECS-8700903.

[*] Signal and Image Processing Institute and Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089-0272.

[**] Department of Electrical Engineering and Computer Science, University of California, Davis, CA 95616.

I. Introduction

Many physical and engineering systems are described by partial differential equations (PDEs). It is generally impossible to obtain closed-form analytical solutions for these equations due to the irregularity of problem domains, and because coefficients are usually spatially varying. Consequently, the numerical solution of PDEs plays an important role in understanding and simulating a wide variety of physical phenomena. Since the late 1940s, the gradual emergence of high-speed computers, culminating with the introduction of supercomputers, has made it possible for researchers to test and develop new PDE solution techniques. The amount of research activity concerned with the numerical analysis of PDEs has therefore growing very rapidly. Many discretization schemes, computational algorithms, and novel computer architectures have been proposed to solve PDEs efficiently. In spite of these developments, the numerical solution of PDEs is still one of the most challenging areas of numerical analysis due to the versatile and often complicated structure of PDEs, and because of the large amount of variables that need to be computed for two or higher dimensional problems.

In this survey, we focus our attention on the discretization and solution of 2-D second-order linear elliptic PDEs of the form

$$a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2} + c\frac{\partial u}{\partial x} + d\frac{\partial u}{\partial y} + eu = f \ , \qquad (1.1)$$

with $ab > 0$, where the coefficients are in general functions of x and y. Elliptic PDEs are often used to characterize the steady-state behavior of physical systems defined over a bounded domain. In this context, boundary conditions representing experimental conditions are usually imposed on the domain boundary, thus yielding a boundary-value problem. The familiar Laplace, Poisson, Helmholtz and convection-diffusion equations are all special cases of (1.1). The solution of (1.1) has therefore a wide range of applications [13],[82].

Elliptic PDEs can be divided into self-adjoint positive definite, indefinite and nonself-adjoint equations, depending on the eigenvalues of the associated differential operator. If an operator is self-adjoint, it has a real spectrum (eigenvalues). Furthermore, if it is positive definite, all its eigenvalues are positive. The discretization of self-adjoint positive definite differential operators leads to *symmetric positive definite* (SPD) matrices. In contrast, the discretization of nonself-adjoint elliptic operators gives rise to nonsymmetric matrices whose eigenvalues are in general complex. It is customary to use the Poisson, Helmholtz and convection-diffusion equations on the unit square $\Omega = [0,1]^2$ with appropriate boundary conditions as model problems for self-adjoint positive definite, indefinite and nonself-adjoint elliptic PDEs, respectively. They can be expressed as follows.

Poisson equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f \ , \tag{1.2}$$

Helmholtz equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \omega^2 u = f \ , \tag{1.3}$$

Convection-diffusion equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + c\frac{\partial u}{\partial x} + d\frac{\partial u}{\partial y} = f \ . \tag{1.4}$$

Generally speaking, the numerical solution of PDEs involves two tasks: (a) choosing a *discretization scheme* to transform the PDE of interest into a discrete problem that approximates it, and (b) selecting a *solution method* for the discretized problem. These two tasks are usually performed separately for single grid solution techniques, but they are combined for multigrid methods. For expository purposes, since the goals of the discretization and solution steps are different, they will be examined independently. In this paper, we study the discretization of all three model problems (1.2)-(1.4). As to solution methods, the design and analysis of iterative algorithms for solving self-adjoint positive definite elliptic PDEs has reached an advanced state of development, whereas a complete theory is not yet available for indefinite and non-selfadjoint PDEs. Thus, we focus on the solution of self-adjoint positive definite PDEs modeled by the Poisson equation (1.2).

Our exposition relies on a DSP (Digital Signal Processing) approach [28],[31],[74],[79]. From the DSP viewpoint, 2-D differential and finite-difference operators correspond to 2-D analog and digital filters, respectively. The discretization of PDEs specifies an approximation problem, i.e., how to match the spectra of analog and digital filters. The solution of PDEs requires the implementation of a *deconvolution* filter which recovers the input u from the output f given by

(1.1). Thus, the discretization and solution of PDEs can be formulated as multidimensional filter specification and filter design problems, respectively.

A key step in deriving discretization schemes is the selection of a set of test functions for which the discretized operator must behave in the same way as the original differential operator. It turns out that a good set of test functions can be chosen by using concepts of linear systems theory. Roughly speaking, they are obtained by examining the zeros of the system function corresponding to the differential operator. This approach leads to the *mode-dependent* discretization scheme described in Section II.

The discretization procedure leads to a system of finite-difference equations, which are often solved iteratively. The convergence rate of iterative methods is traditionally studied within the framework of *matrix iterative* analysis [12],[87],[93]. This form of analysis uses tools from numerical linear algebra, where special concepts such as those of L-, M-, and consistently ordered matrices and related inequalities are introduced to facilitate the characterization of the convergence property. The advantage of matrix analysis is its general applicability. It can be applied to PDEs with irregular geometries and spatially varying coefficients, or which are discretized with nonuniform grids, as long as the corresponding iteration matrices satisfy the desired properties.

An approach complementing the matrix formulation relies on *model problem* analysis, whereby the convergence rate of a given iterative method is analyzed for a simple model problem. This form of analysis has several advantages. First, it is much simpler and therefore provides some insight into the behavior of the algorithms that we study. Secondly, the estimates that are provided by this approach for parameters such as the optimum relaxation parameter for the SOR (Successive Over-Relaxation) method, or the smoothing rate of multigrid methods, are usually much sharper than comparable estimates provided by matrix analysis. Finally, the actual convergence behavior of an iterative method for a general class of problems can be well predicted by the model problem approach, as long as the model problem is chosen appropriately.

The model problem approach relies heavily on Fourier analysis. In this survey, we show that it is in fact closely related to the digital filtering concept appearing in multidimensional DSP. Several examples are given below. Accelerated relaxation methods such as the SOR and Chebyshev iterative methods can be viewed as parametrized lowpass filters for the error between the initial guess and the true solution, where the parameters are chosen to optimize the filtering characteristics. The incomplete LU preconditioning technique for the conjugate gradient method can be interpreted as corresponding to the approximation of a 2-D noncausal FIR filter by the product of two causal and anticausal 2-D FIR filters. The difficulty in that respect lies in the fact that since 2-D polynomials are generally not factorable, the 2-D causal and anticausal filters obtained by spectral factorization have infinite support, and need therefore to be approximated. Finally, if we consider multigrid solution methods, the interpolation and restriction operators appearing in the description of these algorithms are special cases of sampling-rate conversion operations occurring in multirate signal processing. The details of all the above examples will be discussed below. The main purpose of these examples is to illustrate the fact that many tools and concepts arising in the solution of elliptic PDEs are amenable to interpretation and analysis from the point of view of multidimensional DSP.

This survey contains two parts: the first part (Section II) considers discretization schemes, whereas the second part (Sections III-IX) examines solution methods. Readers seeking to locate quickly topics of interest may want to consult the following table of contents.

Finally, we discuss future extensions and present some concluding remarks in Section X.

II. Mode-Dependent Discretization

Three types of discretization techniques, the finite-difference, finite-element, and spectral methods, are commonly used to discretize spatial partial differential operators. In this section, we focus our attention on mode-dependent finite-difference discretization schemes (which constitute an extension of standard finite-difference methods), since they are particularly interesting from a digital filtering point of view. The reader is referred to [68] and the references therein for a discussion of the relation existing between spectral and mode-dependent discretization methods, and for a brief overview of mode-dependent finite-element methods.

The analysis and design of mode-dependent finite-difference discretization schemes can be formulated in a simple way in the frequency domain. The Laplace and Z-transforms are used to represent the constant-coefficient differential operator and its discrete approximation by polynomial expressions of the transform variables s and z. Then, the selection of a mode-dependent discretization scheme becomes equivalent to requiring that the spectra of the continuous and discretized operators, and their derivatives, should match each other at a number of frequencies in the transform domain. In DSP terms, since we require that the spectra of the continous and discretized operators should be as close as possible, the PDE discretization problem can therefore be viewed as a filter specification and design problem.

A. The Mode-Dependent Finite-Difference Discretization Approach

Consider a function of the form

$$u(x) = \sum_{k=1}^{K} [c_{k0} + c_{k1}x + c_{k2}\frac{x^2}{2!} + \cdots + c_{kn_k}\frac{x^{n_k}}{(n_k)!}] e^{s_k x} ,$$

where each term $x^p e^{s_k x}$, $0 \leq p \leq n_k$, is called a mode of order p at the frequency s_k. We are interested in approximating a linear Rth-order constant-coefficient differential operator operating on $u(x)$,

$$L(D) = \sum_{r=0}^{R} a_r D^r , \tag{2.1}$$

where $D = \frac{d}{dx}$, by a $(r_2 - r_1 + 1)$-point finite-difference operator

$$L_d(E) = \sum_{r=r_1}^{r_2} b_r E^r , \tag{2.2}$$

where E is the shift operator defined on an infinite uniform grid Ω_h with spacing h, i.e. for nh, $(n+r)h \in \Omega_h$, $E^r u(nh) = u((n+r)h)$. L_d corresponds to a forward, backward or central difference operator depending on whether $r_1 = 0$, $r_2 = 0$ or $-r_1 = r_2$, respectively. We denote by

$$P_n(s) = \{u(x) : u(x) = e^{sx} \sum_{k=0}^{n} c_k x^k\} \tag{2.3}$$

the space spanned by polynomials of degree at most n multiplied by the factor e^{sx}. A mode-dependent finite-difference discretization scheme is obtained by selecting the coefficients b_r of L_d such that

$$[L_d(E) - L(D)]u(x) = 0 \quad \text{for } u(x) \in C \text{ and } x \in \Omega_h , \tag{2.4}$$

where C, which is called the coincident space of L_d, is the direct sum of subspaces of the form (2.3), i.e.

$$C = \bigoplus_{k=1}^{K} P_{n_k}(s_k) \tag{2.5}$$

A mode in the coincident space C is called a coincident mode, and its frequency is called a coincident frequency.

The above mode-dependent finite-difference scheme specification can be converted easily to the transform domain. Let $L(s)$ be the spectrum obtained by replacing D with s in (2.1) through the use of the Laplace transform, i.e.,

$$L(s) = \sum_{r=0}^{R} a_r s^r .$$

Let also $L_d(z)$ be the discrete spectrum obtained by using the Z-transform to replace E by z in (2.2), so that

$$L_d(z) = \sum_{r=r_1}^{r_2} b_r z^r = \sum_{r=r_1}^{r_2} b_r e^{rsh} ,$$

where the last equality is due to the fact that since E is related to D via $E = e^{hD}$ [29], we have $z = e^{sh}$. Then, the difference Δ between L and L_d can be expressed in terms of the variable s as

$$\Delta(s) = L_d(e^{sh}) - L(s) , \tag{2.6}$$

and the mode-dependent finite-difference scheme specification (2.4)-(2.5) takes the form (see [68] for a proof)

$$\Delta^{(p)}(s_k) = 0 \ , \quad 0 \le p \le n_k \ , \ 1 \le k \le K \ , \tag{2.7a}$$

where

$$\Delta^{(p)}(s_k) = \frac{d^p \Delta(s)}{ds^p}\bigg|_{s=s_k} \ . \tag{2.7b}$$

It is usually easier to determine the coefficients b_r of a mode-dependent finite-difference discretization scheme by using (2.7) rather than (2.4)-(2.5).

The key element in the specification of a mode-dependent different scheme is the choice of coincident space C. In the following two subsections, we discuss the selection of C for several types of problems.

B. Discretization of Homogeneous Boundary-Value ODEs

Consider an Rth-order ($R=2m$) homogeneous differential equation

$$Lu = 0 \ , \quad \text{with } L = \sum_{r=0}^{2m} a_r D^r \text{ and } a_{2m} = 1 \ , \tag{2.8}$$

on the interval [0,1], with given boundary conditions. We seek to discretize it with a $(2m+1)$-point central difference scheme on a uniform grid with spacing h. The characteristic equation of (2.8) is

$$\begin{aligned} L(s) &= s^{2m} + a_{2m-1}s^{2m-1} + \cdots + a_1 s + a_0 \\ &= \prod_{k=1}^{K} (s - s_k)^{n_k} = 0 \ , \end{aligned} \tag{2.9}$$

with $\sum_{k=1}^{K} n_k = 2m$, where s_k is a *natural frequency* of L of order n_k. Then, the operator L has the $2m$-dimensional nullspace

$$N_L = \bigoplus_{k=1}^{K} P_{n_k-1}(s_k) \ .$$

To determine uniquely a $(2m+1)$-point finite difference scheme, we need to specify a $(2m+1)$-dimensional coincident space C. However, since a homogeneous finite-difference equation can be scaled by an arbitrary constant, a $2m$-dimensional coincident space C is sufficient. An exact discretization for (2.8) is obtained by selecting

$$C = N_L \ . \tag{2.10}$$

For this choice, the relations (2.7) yield

$$L_d(z) = Az^{-m} \prod_{k=1}^{K} (z - z_k)^{n_k} , \quad \text{with } z_k = e^{s_k h} , \tag{2.11}$$

where A is a scaling factor and the multiplication factor z^{-m} is due to the fact that we want $L_d(z)$ to be a central difference scheme. The choice of scaling factor A does not affect the solution of the discretized equation

$$L_d(E) u_d = 0 .$$

However, in order to analyze the discretization error $\Delta(s)$, it is convenient to choose A such that $L_d(e^{sh})$ and $L(s)$ are consistent over fine grids. This constraint implies that A must be proportional to h^{-2m}, as h goes to zero.

1D Laplace equation: For $L(D) = D^2$, we know that $N_L = \{1 , x\}$. The coincident modes have the same frequency $s_k = 0$. According to (2.11), we have

$$L_d(E) = AE^{-1}(E - 1)^2 = A(E - 2 + E^{-1}) . \tag{2.12}$$

If we choose $C = N_L + \{x^2\}$, the constant A is uniquely determined. We obtain $A = h^{-2}$, and in this case (2.12) reduces to the standard 3-point central difference scheme for D^2.

1D convection-diffusion equation: Let $L(D) = D^2 - aD$, with $a \neq 0$. Then, $N_L = \{1 , e^{ax}\}$ and $s_k = 0 , a$, so that in (2.11) we have

$$L_d(E) = AE^{-1}(E - 1)(E - e^{ah}) = A[E - (1+e^{ah}) + e^{ah} E^{-1}] . \tag{2.13}$$

If we select $C = N_L + \{x\}$, we find that $A = a[h(e^{ah} - 1)]^{-1}$, and (2.13) becomes identical to a scheme considered by Allen and Southwell [4].

C. Discretization of Homogeneous Boundary-Value PDEs

Consider a general 2D homogeneous boundary-value PDE on the square $[0,1]^2$

$$L(D_x, D_y) u = 0 , \quad \text{with } L(D_x, D_y) = \sum_{r,s} a_{r,s} D_x^r D_y^s , \tag{2.14}$$

where

$$D_x^r = \frac{\partial^r}{\partial x^r} , \quad D_y^s = \frac{\partial^s}{\partial y^s} ,$$

with appropriate boundary conditions. We discretize (2.14) with the finite-difference scheme

$$L_d(E_x,E_y)u_d = 0 \ , \quad \text{where} \quad L(E_x,E_y) = \sum_{r,s} b_{r,s} E_x^r E_y^s \ , \tag{2.15}$$

and where E_x and E_y denote respectively the shift operators in the x- and y-directions on the uniform grid Ω_{h_x,h_y} obtained by discretizing the unit square with horizontal and vertical meshes h_x and h_y. Relying upon a natural generalization of the 1D case, we have the following correspondences between 2D space domain operators and transform domain variables

$$D_x \longleftrightarrow s_x \ , \ D_y \longleftrightarrow s_y \ , \ E_x \longleftrightarrow z_x \ , \ E_y \longleftrightarrow z_y \ . \tag{2.16}$$

where $s_x = \sigma_x + i\omega_x$ and $s_y = \sigma_y + i\omega_y$, and where the identities $E_x = e^{h_x D_x}$, $E_y = e^{h_y D_y}$, $z_x = e^{h_x s_x}$ and $z_y = e^{h_y s_y}$ are satisfied. For simplicity, we now restrict our attention to the case where $h_x = h_y = h$.

Substituting $u = e^{s_x x + s_y y}$ inside (2.14), we obtain the characteristic equation

$$\sum_{r,s} a_{r,s} s_x^r s_y^s = 0 \ . \tag{2.17}$$

Since the complex equation (2.17) imposes only two real constraints on the real and imaginary parts of the complex variables s_x and s_y, there are infinitely many solutions to this equation and therefore infinitely many modes in N_L. It is not possible to approximate all modes in N_L exactly. Thus, we have to select a finite-dimensional subspace $D_L \subset N_L$, called the dominant-mode space, as the coincident space C for L_d. The determination of D_L depends on a rough estimate of the local behavior of the solution. This information is usually provided by the structure of the PDE operator and of the boundary conditions. In this section, we restrict our attention to the case where the dominant modes are either oscillating or exponentially growing (decaying). In other words, coincident frequencies are selected among the sets

$$\{(s_x,s_y) : (s_x,s_y) = (\sigma_x,\sigma_y)\} \quad \text{or} \quad \{(s_x,s_y) : (s_x,s_y) = (i\omega_x,i\omega_y)\} \ . \tag{2.18}$$

Laplace equation: Let $L(D_x,D_y) = D_x^2 + D_y^2$. Since only one frequency $(s_x,s_y) = (0,0)$ satisfies the characteristic equation and belongs to the sets (2.18), (0,0) is selected as the unique coincident frequency. In this case, the mode-dependent and conventional discretization schemes are identical.

The following 5-point, rotated 5-point and 9-point stencil discretization schemes have been derived by several approaches [29],[60],[75],

$$L_{d,+}(E_x,E_y) = \frac{1}{h^2}(E_x + E_x^{-1} + E_y + E_y^{-1} - 4) , \qquad (2.19)$$

$$L_{d,\times}(E_x,E_y) = \frac{1}{2h^2}(E_xE_y + E_x^{-1}E_y + E_xE_y^{-1} + E_x^{-1}E_y^{-1} - 4) , \qquad (2.20)$$

$$\begin{aligned} L_{d,9}(E_x,E_y) = \frac{1}{6h^2}[&4(E_x + E_x^{-1} + E_y + E_y^{-1}) \\ &+ (E_xE_y + E_x^{-1}E_y + E_xE_y^{-1} + E_x^{-1}E_y^{-1}) - 20] . \end{aligned} \qquad (2.21)$$

It is well known that the accuracy of the above schemes for discretizing the Laplace equation is $O(h^2)$, $O(h^2)$ and $O(h^6)$, respectively.

We present now another derivation of these schemes by matching $L(s_x,s_y)$ and $L_d(z_x,z_y)$ at the coincident frequency (0,0) in the transform domain. As before, we consider the expansion of $\Delta = L_d - L$ around (0,0),

$$\begin{aligned} \Delta(s_x,s_y) = \Delta^{(0,0)}(0,0) + \Delta^{(1,0)}(0,0)s_x + \Delta^{(0,1)}(0,0)s_y + \frac{1}{2}[\Delta^{(2,0)}(0,0){s_x}^2 \\ + \Delta^{(1,1)}(0,0)2s_x s_y + \Delta^{(0,2)}(0,0){s_y}^2] + \sum_{\substack{p+q\geq 3 \\ p,q\geq 0}} \Delta^{(p,q)}(0,0)\frac{1}{p!q!}s_x^p s_y^q , \end{aligned} \qquad (2.22)$$

where

$$\Delta^{(p,q)}(0,0) = \left.\frac{\partial^{p+q}\Delta(s_x,s_y)}{\partial^p s_x \partial^q s_y}\right|_{(s_x,s_y)=(0,0)} ,$$

is a function of the grid size h. Hence, (2.22) is in fact a power series of h. Our derivation attempts to make the order of the residual terms in (2.22) as high as possible.

The discretization schemes (2.19) and (2.20) can be derived by requiring respectively that

$$\Delta^{(0,0)}(0,0) = \Delta^{(1,0)}(0,0) = \Delta^{(0,1)}(0,0) = \Delta^{(2,0)}(0,0) = \Delta^{(0,2)}(0,0) = 0 ,$$

and

$$\Delta^{(0,0)}(0,0) = \Delta^{(1,0)}(0,0) = \Delta^{(0,1)}(0,0) = \Delta^{(1,1)}(0,0) = \Delta^{(2,0)}(0,0) = \Delta^{(0,2)}(0,0) = 0 .$$

Note the similarity between these requirements and (2.7). The above choice of constraints $\Delta^{(p,q)}(0,0) = 0$ has taken into account the specific structure of operators $L_{d,+}$, $L_{d,\times}$ and L. For example, in the case of $L_{d,\times}$, the symmetry properties of $L_{d,\times}$ imply that $\Delta^{(2,0)}(0,0) = \Delta^{(0,2)}(0,0)$, so that among the six constraints which are used to specify $L_{d,\times}(E_x,E_y)$, only five are independent.

By setting the coefficients of low order terms in (2.22) equal to zero, it is possible to obtain various high-order finite-difference discretization schemes. For example, to obtain the 9-point scheme (2.21), we need only to impose the requirement that this scheme should have an accuracy of $O(h^6)$ for modes satisfying the characteristic equation $s_x^2 + s_y^2 = 0$. Then, substituting this equation inside (2.22) and setting coefficients up to order h^5 equal to zero, we obtain nine independent constraints which specify (2.21) uniquely.

Helmholtz equation: Let

$$L(D_x,D_y) = D_x^2 + D_y^2 + \lambda^2 .$$

If s_x and s_y are purely imaginary, the characteristic equation becomes

$$\omega_x^2 + \omega_y^2 = \lambda^2 , \qquad (2.23)$$

which is a circle in the ω_x-ω_y plane, centered at the origin and with radius $|\lambda|$. There are infinitely many natural frequencies and, hence, there are many different ways to select coincident frequencies. Our choice is based on the following two considerations. First, if there is no further information about the dominant modes, a reasonable strategy consists in distributing the coincident frequencies uniformly along the contour (2.23). Second, we want to preserve the symmetry properties of L, so that the resulting discretization scheme will have a simple form and will be easy to implement.

Let us select

$$(\omega_x , \omega_y) = (|\lambda|\cos(\frac{n}{2}\pi+\frac{1}{4}\pi) , |\lambda|\sin(\frac{n}{2}\pi+\frac{1}{4}\pi)) , \qquad 0 \le n \le 3 ,$$

as coincident frequencies as shown in Fig. 2.1(a). With this choice, the discretization can be performed independently in the x- and y-directions. The resulting scheme is

$$L_d(E_x,E_y) = A\,[E_x^{-1} - 2\cos(\frac{|\lambda|}{\sqrt{2}}h) + E_x + \kappa(E_y^{-1} - 2\cos(\frac{|\lambda|}{\sqrt{2}}h) + E_y)] .$$

Two parameters A and κ remain undetermined. The parameter κ is selected such that the discretization error $\Delta(s_x, s_y)$ corresponding to natural frequencies is proportional to $O(h^2)$, and A is used to normalize the above scheme so that L_d is consistent with L. This yields $\kappa = 1$ and $A = h^{-2}$. We obtain the symmetric 5-point stencil discretization operator

$$L_{d,+}(E_x, E_y) = \frac{1}{h^2}[E_x^{-1} + E_x + E_y^{-1} + E_y - 4\cos(\frac{|\lambda|}{\sqrt{2}}h)] . \qquad (2.24)$$

Rotating the above four coincident frequencies in the transform domain and the associated 5-point stencil in the space domain by an angle $\pi/4$, we obtain another mode-dependent 5-point stencil discretization. In this scheme, the coincident frequencies become

$$(\omega_x \ , \ \omega_y) = (\ |\lambda|\cos(\frac{n}{2}\pi) \ , \ |\lambda|\sin(\frac{n}{2}\pi)) \ , \qquad 0 \le n \le 3 \ .$$

as shown in Fig. 2.1(b), and the resulting discretization operator is

$$L_{d,\times}(E_x, E_y) = \frac{1}{2h^2}[E_x^{-1}E_y^{-1} + E_x^{-1}E_y + E_x E_y^{-1} + E_x E_y - 4\cos(|\lambda|h)] \ . \ (2.25)$$

Note that this rotated 5-point stencil can be viewed as corresponding to a discretization on a grid with spacing $\sqrt{2}h$. By appropriately combining (2.24), (2.25) and adding a constant term, we obtain the 9-point stencil discretization operator,

$$L_{d,9}(E_x, E_y) = \frac{\gamma_\times}{\gamma_\times + \gamma_+} L_{d,+}(E_x, E_y) + \frac{\gamma_+}{\gamma_\times + \gamma_+} L_{d,\times}(E_x, E_y) - \frac{\gamma_\times \gamma_+}{\gamma_\times + \gamma_+} \quad (2.26)$$

Then, if

$$\gamma_\times = L_{d,\times}(e^{i\frac{|\lambda|}{\sqrt{2}}h}, e^{i\frac{|\lambda|}{\sqrt{2}}h}) = \frac{1}{h^2}[\cos(\sqrt{2}\,|\lambda|h) + 1 - 2\cos(|\lambda|h)] \ , \quad (2.27a)$$

$$\gamma_+ = L_{d,+}(e^{i|\lambda|h}, 1) = \frac{1}{h^2}[2\cos(|\lambda|h) + 2 - 4\cos(\frac{|\lambda|}{\sqrt{2}}h)] \ , \qquad (2.27b)$$

we are able to match $L_d(z_x, z_y)$ and $L(s_x, s_y)$ at 8 frequencies

$$(\omega_x \ , \ \omega_y) = (\ |\lambda|\cos(\frac{n}{4}\pi) \ , \ |\lambda|\sin(\frac{n}{4}\pi)) \ , \qquad 0 \le n \le 7 \ .$$

as shown in Fig. 2.1(c). Thus, (2.26) is a mode-dependent 9-point stencil discretization operator for the Helmoltz equation. It can be shown that both $L_{h,+}$ and $L_{h,\times}$ have an accuracy of $O(h^2)$ and that $L_{h,9}$ has an accuracy of $O(h^6)$.

Convection-diffusion equation: In this case,

$$L(D_x,D_y) = D_x^2 + D_y^2 - 2\alpha D_x - 2\beta D_y \ .$$

Then, if we consider only real frequencies $(s_x,s_y) = (\sigma_x,\sigma_y)$, the characteristic equation reduces to

$$\sigma_x^2 + \sigma_y^2 - 2\alpha\sigma_x - 2\beta\sigma_y = 0 \ , \qquad (2.28)$$

which is a circle in the σ_x-σ_y plane centered at $(\alpha \ , \ \beta)$ with radius $d = (\alpha^2+\beta^2)^{1/2}$.

The conventional approach for discretizing the above equation uses central differences to approximate the first and second order derivatives separately. This gives

$$L_{d,c}(E_x,E_y) = \frac{1}{h^2}[(1+\alpha h)E_x^{-1} + (1-\alpha h)E_x - 4 + (1+\beta h)E_y^{-1} + (1-\beta h)E_y] \ , \qquad (2.29)$$

which corresponds to selecting a single coincident frequency at the origin. Allen and Southwell [4] combined two 1D mode-dependent schemes, i.e. (2.13), along the x- and y-directions. This yields

$$L_{d,AS}(E_x,E_y) = \frac{1}{h}[\frac{2\alpha}{e^{2\alpha h}-1}(e^{2\alpha h}E_x^{-1} - (1+e^{2\alpha h}) + E_x) + \frac{2\beta}{e^{2\beta h}-1}(e^{2\beta h}E_y^{-1} - (1+e^{2\beta h}) + E_y)] \ , \qquad (2.30)$$

which corresponds to selecting (0,0), $(2\alpha,0)$, $(0,2\beta)$, $(2\alpha,2\beta)$ as coincident frequencies. Motivated by the discussion of the previous section, we can also select the coincident frequencies

$$(\sigma_x \ , \ \sigma_y) = (\alpha + d\cos(\frac{n}{2}\pi + \frac{1}{4}\pi) \ , \ \beta + d\sin(\frac{n}{2}\pi + \frac{1}{4}\pi)) \ , \qquad 0 \le n \le 3 \ ,$$

uniformly along the contour (2.28), which gives the discretization operator

$$L_{d,+}(E_x,E_y) = \frac{1}{h^2}[e^{\alpha h}E_x^{-1} + e^{-\alpha h}E_x + e^{\beta h}E_y^{-1} + e^{-\beta h}E_y - 4\cosh(\frac{d}{\sqrt{2}}h)] \ . \qquad (2.31)$$

The multiplication of E_x and E_y by the factors $e^{-\alpha h}$ and $e^{-\beta h}$ in the space domain corresponds to a shift of the s_x and s_y variables in the transform domain, where s_x and s_y become $s_x - \alpha$ and $s_y - \beta$, respectively. The above scheme shifts therefore the center (α,β) of the circle (2.28) to the origin and interprets the resulting circle as corresponding to a Helmoltz equation with radius d. The coincident frequencies for the three schemes (2.29)-(2.31) are shown in Fig. 2.2. Following a procedure similar to the one used for the Helmoltz equation, we can also design mode-dependent rotated 5-point and 9-point stencil discretization schemes for the convection-diffusion equation. These schemes have an accuracy of $O(h^2)$ and $O(h^6)$, respectively.

D. Historical Notes

Historically, the idea of selecting exponential functions as coincident modes was first suggested by Allen and Southwell [4] for discretizing the convection-diffusion equation. An important feature of this problem is that there are large first-order terms in the governing second-order PDE. Due to these large first-order terms, there exists a boundary layer which cannot be well approximated by polynomials. The use of trigonometric functions as coincident modes was first discussed by Gautschi [41] for the numerical integration of ODEs which have periodic or oscillatory solutions whose periods can be estimated in advance. The advantage of selecting nonpolynomial functions as coincident modes has been recognized for years and applied to PDE problems repeatedly in the literature (see for example the references appearing in [68]). However, until recently, all mode-dependent discretization results were derived by considering one specific equation at a time, and it is only in [68] that a general framework was provided for the study of mode-dependent discretization methods.

III. Solution of Self-Adjoint Positive Definite Elliptic PDEs: Problem Formulation

Once (1.1) has been discretized with a finite-difference or finite-element scheme, the remaining task is to solve a system of linear difference equations of the form

$$A u_d = f_d \ , \tag{3.1}$$

where A is a sparse matrix, and u_d and f_d are discrete approximations of u and f, respectively. Suppose that u_d and f_d are vectors of length N. The solution of (3.1) by Gaussian-elimination requires $O(N^3)$ operations, which is prohibitive for most practical applications. However, if the matrix A is symmetric positive definite (SPD), several direct and iterative methods [12],[51],[87], which require between $O(N)$ and $O(N^2)$ operations, can be used to solve (3.1) efficiently.

In the following, we shall restrict our attention to the case where the coefficient matrix A in (3.1) is SPD. In terms of the differential operator (1.1), this amounts to second-order self-adjoint positive definite elliptic PDEs which can be expressed in the form

$$\frac{\partial}{\partial x}\left(B\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(C\frac{\partial u}{\partial y}\right) + Du = F \ , \tag{3.2}$$

where B and C are positive functions and $D \leq 0$. This subclass of equations includes the Poisson equation, which will be used below as the prototype for equations of the form (3.2).

To study the convergence rate of iterative solution techniques for (3.2), the traditional approach consists in using matrix iterative analysis [12],[51],[87], which relies on a detailed characterization of the structure of iteration matrices. Another approach, which has become popular recently, uses Fourier analysis to study the convergence behavior for a simple model problem. If the model problem is representative of the general class of problems that we want to solve, the convergence behavior for general problems can be inferred from the results obtained for the model problem. Since this second approach analyzes the effect of iterations on each Fourier mode through the use of digital signal processing methods, it is called here the DSP approach.

The advantage of the matrix approach is its general applicability. It can be applied to PDEs with irregular domain geometries, spatially varying coefficients, and when the discretization is performed on nonuniform grids. The only requirement is that the iteration matrices should possess certain properties, such as property A or consistent ordering [51],[92],[93]. In contrast, the DSP approach can only be rigorously applied to a small class of problems. It presents, however, several advantages. First, the matrix approach is in general much more complicated than the DSP approach. Second, for simple problems, the DSP approach yields more accurate estimates of important quantities such as the optimal relaxation parameter for the SOR method, the smoothing rate of multigrid methods, or the eigenvalue distribution of the preconditioned operator obtained by applying a preconditioner to the discretized form of (3.2). Finally, the convergence behavior of iterative algorithms predicted by the DSP analysis of simple model problems is usually consistent with results obtained by performing numerical experiments on complicated problems. Thus, in spite of its simplicity, the DSP approach provides results which are applicable to very general problems.

A. The Model Poisson Problem

The standard model problem for (3.2) is the Poisson equation on the unit square $\Omega = [0,1]^2$

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} = f(x,y) \ . \tag{3.3}$$

with appropriate boundary conditions. It can be discretized on a uniform grid

$$\Omega_h = \{(n_x h \, , \, n_y h) : 0 \leq n_x , \; n_y \leq M\} \, , \tag{3.4}$$

with grid spacing $h = M^{-1}$. Approximating the Laplacian with the 5-point finite-difference scheme (2.19), and denoting by u_{n_x,n_y} the discrete approximation of the solution $u(n_x h, n_y h)$, we obtain the discretized system

$$\frac{1}{h^2}(u_{n_x+1,n_y} + u_{n_x-1,n_y} + u_{n_x,n_y+1} + u_{n_x,n_y-1} - 4u_{n_x,n_y}) = f_{n_x,n_y} \ , \tag{3.5}$$

at points $(n_x h, n_y h)$ which are located in the interior of Ω_h, i.e., for $1 \leq n_x , \; n_y \leq M{-}1$. This system can be rewritten in terms of shift operators as

$$A(E_x,E_y)u_{n_x,n_y} = -\frac{h^2}{4} f_{n_x,n_y} \ , \tag{3.6}$$

with

$$A(E_x,E_y) = 1 - \frac{1}{4}(E_x + E_x^{-1} + E_y + E_y^{-1}) \,. \tag{3.7}$$

Boundary Conditions: For self-adjoint positive definite elliptic PDEs, it has been observed empirically [22] that the convergence behavior of a given iterative algorithm is not significantly affected by the choice of boundary conditions. This implies that we can, without loss of rigor, restrict our attention to Dirichlet or periodic boundary conditions, since these boundary conditions have the advantage that they lend themselves easily to Fourier analysis. For Dirichlet boundary conditions, the solution $u(x,y)$ is specified along the boundary of the domain Ω. In terms of the discretized system (3.5), this means that $u_{n_x,0}$, $u_{n_x,M}$, u_{0,n_y} and u_{M,n_y} are given. Thus, the system (3.5) consists of $(M-1)^2$ equations in $(M-1)^2$ unknowns. Since nonzero boundary values can be moved to the right hand side and treated as part of the driving function, the system (3.5) with Dirichlet boundary conditions can be replaced by an equivalent system with a modified driving function and zero boundary conditions. Without loss of generality, the system (3.5) with zero boundary conditions

$$u_{n_x,0} = u_{n_x,M} = u_{0,n_y} = u_{M,n_y} = 0 \,, \tag{3.8}$$

where $1 \leq n_x,\ n_y \leq M-1$, is therefore called the *model Dirichlet problem.* Similarly, the system (3.5) with periodic boundary conditions

$$u_{n_x,0} = u_{n_x,M} \quad \text{and} \quad u_{0,n_y} = u_{M,n_y} \,, \tag{3.9}$$

where $0 \leq n_x,\ n_y \leq M-1$, is called the *model periodic problem.* It is easy to check that the model periodic problem involves M^2 equations in M^2 variables.

B. Orderings

To specify an algorithm for processing a multidimensional sequence, it is important to indicate the order in which the sequence should be computed. For example, a certain ordering of grid points is needed to implement 2D IIR filters. Similarly, for PDE algorithms, it is necessary to indicate clearly the ordering scheme which is employed, since the numerical performance of a given algorithm depends in general on the ordering [2],[66],[81]. We will focus our attention here on the natural and red-black (or checkered) orderings, since they are the most commonly employed, and are both amenable to Fourier analysis. The *natural ordering* corresponds to a standard rowwise (or columnwise) lexicographic ordering of the grid points. In the *red-black ordering*, the grid points are partitioned into two groups, where a grid point (n_x, n_y) is red if $n_x + n_y$ is even, and black if $n_x + n_y$ is odd. Then, as a group, the red points precede the black points, but within each group, points are ordered according to the natural ordering.

Many PDE algorithms have the feature that numerical operations at a given point require only local information. In this case, it is usually possible to divide the grid points into subsets such that operations performed at points within a subset are independent of each other. In this case, the ordering of points within a subset is not important, since operations at such points can be implemented in parallel on a multiprocessor machine. When solving equation (3.5), this leads us to consider the following parallel versions of the natural and red-black orderings.

Parallel natural ordering:

$$(n_x, n_y) < (m_x, m_y) \quad \text{if} \quad n_x + n_y < m_x + m_y \ . \tag{3.10}$$

Parallel red-black ordering:

$$(n_x, n_y) < (m_x, m_y) \quad \text{if} \quad (n_x, n_y) \text{ red and } (m_x, m_y) \text{ black} \ . \tag{3.11}$$

In (3.10) and (3.11), the order between grid points is denoted by an inequality sign. Note that the above parallel natural ordering does not specify an order for points (n_x, n_y) such that $n_x + n_y$ is constant. Similarly, for the parallel red-black ordering, no order is imposed for points of the same color. This is due to the fact that when the Gauss-Seidel or SOR methods described in Section V below are used to solve (3.5), for the natural ordering, points along constant $n_x + n_y$ lines can be updated in parallel. On the other hand, for the red-black version of the same relaxation methods, all points of identical color can be updated in parallel. From the point of view of parallelism, the red-black ordering is therefore preferable, since only two steps are required to scan all the grid points, instead of $O(N^{1/2})$ steps for the natural ordering. However, the convergence rate of a given iterative algorithm can also be affected by the choice of ordering. For example, it has been shown recently [66] that the rate of convergence of the symmetric successive overrelaxation (SSOR) and of several preconditioned conjugate conjugate gradient methods can be slowed significantly if we use a red-black ordering instead of the natural ordering. Thus, when selecting a given ordering, one has to be careful to examine both the numerical complexity of the resulting algorithm as well as its parallelism.

C. Fourier Analysis

Several different Fourier basis functions will be introduced to expand 2D sequences. A sequence u_{n_x,n_y} defined on Ω_h with zero boundary values can be expanded in a sinusoidal Fourier series of the form

$$u_{n_x,n_y} = \sum_{k_x=1}^{M-1} \sum_{k_y=1}^{M-1} \hat{u}_{k_x,k_y} \sin(k_x \pi n_x h) \sin(k_y \pi n_y h) \ . \tag{3.12}$$

It is easy to see that when $A(E_x,E_y)$ is given by (3.7), we have

$$A(E_x,E_y)\sin(k_x \pi n_x h)\sin(k_y \pi n_y h) = \hat{A}(k_x,k_y)\sin(k_x \pi n_x h)\sin(k_y \pi n_y h) \tag{3.13}$$

with

$$\hat{A}(k_x,k_y) = 1 - \frac{1}{2}[\cos(k_x \pi h) + \cos(k_y \pi h)] \ . \tag{3.14}$$

Therefore, $\sin(k_x \pi n_x h)\sin(k_y \pi n_y h)$ is an eigenfunction of operator $A(E_x,E_y)$ corresponding to the eigenvalue $\hat{A}(k_x,k_y)$. It is worth noting at this point that by imposing the condition that the solution u_{n_x,n_y} is synthesized by a finite number of Fourier sine functions as in (3.12), we are able to ignore the zero boundary conditions (3.8) for the model Dirichlet problem and treat $A(E_x,E_y)$ as a shift-invariant operator defined on an infinite grid.

Next, consider a sequence u_{n_x,n_y} defined on Ω_h which satisfies the periodic boundary conditions (3.9). The sequence u_{n_x,n_y} can be expanded in complex exponential Fourier series as

$$u_{n_x,n_y} = \sum_{k_x=0}^{M-1} \sum_{k_y=0}^{M-1} \hat{u}_{k_x,k_y} e^{i2\pi(k_x n_x + k_y n_y)h} \ . \tag{3.15}$$

Since

$$A(n_x,n_y)\, e^{i2\pi(k_x n_x + k_y n_y)h} = \hat{A}(k_x,k_y)\, e^{i2\pi(k_x n_x + k_y n_y)h} \ , \tag{3.16}$$

where

$$\hat{A}(k_x,k_y) = 1 - \frac{1}{2}[\cos(k_x 2\pi h) + \cos(k_y 2\pi h)] \ , \tag{3.17}$$

we see that $e^{i2\pi(k_x n_x + k_y n_y)h}$ is an eigenfunction of $A(E_x, E_y)$ with eigenvalue (3.17). Consequently, by expressing an arbitrary solution as a finite sum of such eigenfunctions, where k_x and k_y are integers between 0 and $M-1$, we can ignore the periodic boundary conditions (3.9) for the model periodic problem and view $A(E_x, E_y)$ as a shift-invariant operator defined on an infinite grid.

To analyze algorithms with a red-black ordering, we can employ a variant of the above Fourier decompositions, which is known as the two-color Fourier analysis [65],[66]. Consider the model Dirichlet problem, and let u_{n_x,n_y} be a sequence defined on Ω_h with zero boundary values. The restriction of this sequence to the red and black points defines two subsequences: the red sequence u_{r,n_x,n_y}, and the black sequence u_{b,n_x,n_y}. They can be expanded respectively in Fourier series as

$$u_{r,n_x,n_y} = \sum_{(k_x,k_y)\in K_r} \hat{u}_{r,k_x,k_y} \sin(k_x \pi n_x h) \sin(k_y \pi k h)\,, \quad n_x + n_y \text{ even}\,, \tag{3.18a}$$

$$u_{b,n_x,n_y} = \sum_{(k_x,k_y)\in K_b} \hat{u}_{b,k_x,k_y} \sin(k_x \pi n_x h) \sin(k_y \pi k h)\,, \quad n_x + n_y \text{ odd}\,, \tag{3.18b}$$

where for M even,

$$K_b = \{\, (k_x, k_y) \in \mathrm{N}^2 : k_x + k_y \le M-1,\ k_x,\ k_y \ge 1 \text{ or }$$
$$1 \le k_x \le \frac{M}{2} - 1,\ k_y = M - k_x \,\}\,, \tag{3.19a}$$

and

$$K_r = K_b \cup \{\, (M/2,\ M/2) \,\}\,. \tag{3.19b}$$

It is straightforward to check that the Fourier coefficients $\hat{u}_{k_x,k_y}$, $\hat{u}_{M-k_x,M-k_y}$ in the sinusoidal expansion (3.12) and $\hat{u}_{r,k_x,k_y}$, $\hat{u}_{b,k_x,k_y}$ in the red-black expansion (3.18) are related via

$$\begin{pmatrix} \hat{u}_{r,k_x,k_y} \\ \hat{u}_{b,k_x,k_y} \end{pmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{pmatrix} \hat{u}_{k_x,k_y} \\ \hat{u}_{M-k_x,M-k_y} \end{pmatrix}, \qquad (k_x, k_y) \in K_b\,, \tag{3.20a}$$

$$\hat{u}_{r,k_x,k_y} = \hat{u}_{k_x,k_y}\,, \qquad (k_x, k_y) = (M/2, M/2)\,. \tag{3.20b}$$

The expression (3.20) can be interpreted as follows. When the sequence u_{n_x,n_y} is sampled only at the red points, instead of all points of Ω_h, the high frequency component $(M-k_x, M-k_y)$ is aliased into the low frequency component (k_x, k_y), so that two Fourier components coexist in the low frequency region. A similar aliasing phenomenon occurs when u_{n_x,n_y} is sampled at the black points only (see Fig. 3.1). Note also that K_r and K_b differ by the single element $(M/2, M/2)$, so that at the frequency $(M/2, M/2)$ a single Fourier coefficient $\hat{u}_{r,M/2,M/2}$ is used to represent the 2D sequence u_{n_x,n_y}. This frequency can therefore be viewed as being degenerate.

With respect to the two-color decomposition (3.18), the discretized system (3.5) can be rewritten as

$$A(E_x, E_y) \begin{pmatrix} u_{r,n_x,n_y} \\ u_{b,n_x,n_y} \end{pmatrix} = -\frac{h^2}{4} \begin{pmatrix} f_{r,n_x,n_y} \\ f_{b,n_x,n_y} \end{pmatrix}, \tag{3.21}$$

with

$$A(E_x, E_y) = \begin{bmatrix} 1 & -a(E_x, E_y) \\ -a(E_x, E_y) & 1 \end{bmatrix}, \tag{3.22}$$

and

$$a(E_x, E_y) = \frac{1}{4}(E_x + E_x^{-1} + E_y + E_y^{-1}) . \tag{3.23}$$

To obtain a frequency domain representation of the above system, we can substitute the Fourier decomposition (3.18) inside (3.21) and match Fourier components. For a nondegenerate frequency (k_x, k_y), this gives

$$\hat{A}(k_x, k_y) \begin{pmatrix} \hat{u}_{r,k_x,k_y} \\ \hat{u}_{b,k_x,k_y} \end{pmatrix} = -\frac{h^2}{4} \begin{pmatrix} \hat{f}_{r,k_x,k_y} \\ \hat{f}_{b,k_x,k_y} \end{pmatrix}, \tag{3.24}$$

with

$$\hat{A}(k_x, k_y) = \begin{bmatrix} 1 & -\hat{a}(k_x, k_y) \\ -\hat{a}(k_x, k_y) & 1 \end{bmatrix}, \tag{3.25}$$

and where

$$\hat{a}(k_x, k_y) = \frac{1}{2}[\cos(k_x \pi h) + \cos(k_y \pi h)] \tag{3.26}$$

is the Fourier transform of the space domain operator $a(E_x, E_y)$. For the degenerate frequency $(k_x, k_y) = (M/2, M/2)$, we obtain

$$\hat{u}_{r,M/2,M/2} = -\frac{h^2}{4}\hat{f}_{r,M/2,M/2} .$$

Note that the above results rely in part on the fact that for the Dirichlet case, the eigenfunctions of the 2×2 matrix operator $A(E_x, E_y)$ are of the form $\mathbf{v}(k_x, k_y)\sin(k_x \pi n_x h)\sin(k_y \pi n_y h)$ where the 2-vector $\mathbf{v}(k_x, k_y)$ is an eigenvector of the matrix $\hat{A}(k_x, k_y)$.

In the previous two-color Fourier analysis of the red-black ordering, we have assumed that the boundary conditions are of Dirichlet type. For the case of periodic boundary conditions, a similar two-color Fourier analysis can be developed. One needs only to replace the sinusoidal expansions (3.18) by complex exponential Fourier series. Since the analysis is identical to the Dirichlet case, the details are omitted. We find that identities (3.21)-(3.25) remain valid, provided that the function $\hat{a}(k_x, k_y)$ is replaced by

$$\hat{a}(k_x, k_y) = \frac{1}{2}[\cos(k_x 2\pi h) + \cos(k_y 2\pi h)] . \tag{3.27}$$

D. Summary

In this section, we have examined the model Poisson problem with Dirichlet or periodic boundary conditions, and with a natural or red-black ordering. In each case, a Fourier basis has been introduced to expand 2D sequences satisfying the boundary conditions. For such sequences, it has been shown that the system (3.5) can be viewed as a linear shift-invariant (LSI) system in the space domain, and can therefore be analyzed in the frequency domain. The results of our analysis are summarized in Table 3.1.

ordering	B.C.	$A(E_x,E_y)$	Fourier basis functions	$\hat{A}(k_x,k_y)$
natural	Dirichlet	(3.7)	(3.12)	(3.14)
natural	periodic	(3.7)	(3.15)	(3.17)
red-black	Dirichlet	(3.22)	(3.18)	(3.25),(3.26)
red-black	periodic	(3.22)	r-b complex sinusoids	(3.25),(3.27)

Table 3.1: Fourier decomposition for several orderings and boundary conditions.

The Fourier analysis that we have developed in this section has focused on the operator $A(E_x,E_y)$ defined in (3.7) or (3.22). Since this operator is a FIR filter, the ordering of grid points does not play a role in its implementation, so that as far as A is concerned, the distinction between the natural and red/black orderings is really unnecessary. However, when solving (3.5), our actual goal is to implement the inverse filter $A^{-1}(E_x,E_y)$, which is a 2D IIR filter, and for which the choice of ordering does matter. To synthesize this filter, we will rely on the iterated application of deconvolution filters, which will be in general of 2D IIR type, thus explaining our interest in the choice of ordering.

IV. Direct Methods

Several efficient direct methods have been developed for solving elliptic PDEs. These methods usually exploit special features of certain classes of PDEs, and are often restricted to regular domain geometries. They are therefore not as widely applicable as the iterative methods to be discussed in the following sections. Furthermore, except for fast Fourier solvers, direct methods rely mainly on matrix or graph-theoretic techniques. Thus, they do not fit well the DSP viewpoint adopted in this paper. Consequently, in this section we focus primarily our attention on FFT solvers. However, for completeness, several other direct methods, such as cyclic block-reduction and sparse Gaussian elimination methods, are briefly discussed.

A. FFT Solvers

Fast Fourier solvers are applicable to 2-D separable elliptic PDEs of the form

$$(P(x)+Q(y))u(x,y)=f(x,y)\ , \tag{4.1}$$

defined on the unit square $[0,1]^2$, with

$$P(x)=\frac{\partial}{\partial x}\left(p_1(x)\frac{\partial}{\partial x}\right)+p_2(x)\ , \tag{4.1a}$$

$$Q(y)=\frac{\partial}{\partial y}\left(q_1(y)\frac{\partial}{\partial y}\right)+q_2(y)\ , \tag{4.1b}$$

and where $p_1(x)q_1(y)>0$. For simplicity, we assume that the boundary conditions are of Dirichlet type, i.e., $u(x,y)=0$ on the domain boundary. A wider class of boundary conditions is considered in [86].

By discretizing the differential operators $P(x)$ and $Q(y)$ on a uniform grid Ω_h with spacing $h=M^{-1}$, with 3-point central differences in the x- and y- directions, respectively, we obtain a 5-point stencil discretization of (4.1). The discretized system can be denoted as

$$(P_d(n_x)+Q_d(n_y))u_{n_x,n_y}=f_{n_x,n_y}\ . \tag{4.2}$$

FFT solvers require that either $P(x)$ or $Q(y)$ should have constant coefficients. If the coefficients $p_1(x)=p_1$ and $p_2(x)=p_2$ of $P(x)$ are constant, the discretized operator

$$P_d(n_x) = P_d = p_1(E_x - 2 + E_x^{-1})/h^2 + p_2 \tag{4.3}$$

has also constant coefficients. Then, the Fourier transform can be used to transform the discretized equation (4.2), which depends on the two variables n_x and n_y, into a set of *decoupled* equations depending on the single variable n_y. Specifically, due to the separability of equation (4.1), we can express the solution u_{n_x,n_y} and driving function f_{n_x,n_y} in the form

$$u_{n_x,n_y} = \sum_{k_x=1}^{M-1} \hat{u}_{k_x,n_y} \sin(k_x \pi n_x h) , \qquad f_{n_x,n_y} = \sum_{k_x=1}^{M-1} \hat{f}_{k_x,n_y} \sin(k_x \pi n_x h) . \tag{4.4}$$

Substituting (4.4) into (4.2), we obtain $M-1$ independent equations

$$(\hat{P}_d + Q_d(n_y))\hat{u}_{k_x,n_y} = \hat{f}_{k_x,n_y} , \qquad 1 \le k_x \le M-1 , \tag{4.5a}$$

with

$$\hat{P}_d = -2h^{-2} p_1 [1 - \cos(k_x \pi h)] + p_2 . \tag{4.5b}$$

The boundary conditions of the transformed system are also of Dirichlet type, i.e.,

$$\hat{u}_{k_x,0} = \hat{u}_{k_x,M} = 0 . \tag{4.6}$$

Then, for each value of k_x, the system (4.5)-(4.6) can be written in matrix form as a tridiagonal system

$$\begin{bmatrix} a_1 & c_1 & & & & \\ b_2 & a_2 & c_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & b_{M-2} & a_{M-2} & c_{M-2} \\ & & & & b_{M-1} & a_{M-1} \end{bmatrix} \begin{pmatrix} \hat{u}_1 \\ \hat{u}_2 \\ \cdot \\ \cdot \\ \hat{u}_{M-2} \\ \hat{u}_{M-1} \end{pmatrix} = \begin{pmatrix} \hat{f}_1 \\ \hat{f}_2 \\ \cdot \\ \cdot \\ \hat{f}_{M-2} \\ \hat{f}_{M-1} \end{pmatrix} , \tag{4.7}$$

where the k_x dependence of the solution, driving term, and matrix entries has been suppressed. Each such system can be solved directly with the following algorithm of complexity $O(M)$.

```
[LU factorization]
    $\alpha_1 := a_1$
    for $n = 2, 3, \cdots, M-1$
        $\beta_n := b_n / \alpha_{n-1}$
        $\alpha_n := a_n - \beta_n c_{n-1}$
[Forward substitution]
    $v_1 := \hat{f}_1$
    for $n = 2, 3, \ldots, M-1$
        $v_n := \hat{f}_n - \beta_n v_{n-1}$
[Backward substitution]
    $\hat{u}_{M-1} := v_{M-1} / \alpha_{M-1}$
    for $n = M-2, M-3, \cdots, 1$
        $\hat{u}_n = (v_n - c_n \hat{u}_{n+1}) / \alpha_n$
```

Table 4.1: Tridiagonal system solver.

Given the solutions $\hat{u}_{k_x,n_y}$ of systems (4.5)-(4.6) for all k_x, the solution u_{n_x,n_y} of the PDE can then be obtained from the discrete sine transform (4.4).

Fast Fourier solvers rely therefore on the following three steps.

Step 1: Perform a 1-D fast sine transform of f_{n_x,n_y} with respect to n_x to determine the hybrid Fourier coefficients $\hat{f}_{k_x,n_y}$.

Step 2: For each k_x, with $1 \leq k_x \leq M-1$, calculate the hybrid Fourier coefficients $\hat{u}_{k_x,n_y}$ by solving the tridiagonal system corresponding to (4.5)-(4.6).

Step 3: Perform a 1-D fast inverse sine transform to compute the solution u_{n_x,n_y} from the hybrid Fourier coefficients $\hat{u}_{k_x,n_y}$.

In the above discussion, we have assumed that the boundary conditions are of Dirichlet type. However, other choices of boundary conditions, such as Neumann or periodic conditions, are also possible. The effect of a change of boundary conditions is to replace the fast sine transform in steps 1) and 3) above by fast cosine transforms, or FFTs [85],[86]. The complexity of the resulting family of FFT solvers is $O(M^2 \log(M))$. However, it is important to keep in mind that these solvers are restricted to problems with a rectangular domain, and where either $P(x)$ or $Q(y)$ has constant coefficients.

B. Other Direct Methods

The above FFT solver was introduced by Hockney in 1965 [53] for the Poisson problem over a rectangle. In both [53] and [54] Hockney discussed another direct method, called *cyclic reduction.* This method is a Gaussian elimination procedure with a particular ordering. Specifically, assume that in (4.2), the variables u_{n_x,n_y} are scanned column by column, and let $\mathbf{u}_{n_x}$ be the $M-1$ dimensional vector formed by the variables with column index n_x. It is easy to check that the set of vectors $\mathbf{u}_{n_x}$ with $1 \leq n_x \leq M-1$ satisfies a block tridiagonal system. Then, assume that we eliminate one out of every two columns from this system, say the columns with n_x even. The resulting system remains block tridiagonal, although the blocks may start to fill in. By proceeding recursively, after $L = \log(M)$ steps, a single column of variables remains. The resulting system of size $M-1$ can be solved, and its solution can be backsubstituted into the system obtained at the previous level, thus enabling us to compute progressively all columns of the original system. For more details on the cyclic block reduction method, the reader is referred to [85],[86]. This method which was unstable in its original form, was later stabilized by Buneman [20]. The complexity of the resulting procedure is $O(M^2 log(M))$, as for FFT solvers.

The FFT and cyclic block reduction methods can be combined to produce a third technique, called the FACR (Fourier analysis-cyclic reduction) algorithm, whose complexity is $O(M^2\text{loglog}(M))$. The FFT, cyclic reduction and FACR solvers are reviewed by Swarztrauber in [85]. Another survey [86] provides a more elementary introduction to this topic.

The cyclic block reduction procedure can be viewed as a special case of a more general family of direct solvers, called *sparse Gaussian elimination* methods. These methods start from a system of the form

$$Ax = b \ , \tag{4.8}$$

where A is symmetric positive definite. The matrix A is usually sparse. However, when (4.8) is solved by performing a Cholesky factorization $A = LL^T$, the lower triangular matrix L contains in general more nonzero elements than existed in the lower triangular part of A, thus resulting in an increase in the storage and computation time required to solve (4.8) by Gaussian elimination. However, the amount of fill, i.e., the number of additional nonzero entries of L, depends highly on the ordering of the variables. If P denotes an arbitrary permutation matrix, it may be of interest to replace the solution of (4.8) by that of

$$By = c \, , \tag{4.9a}$$

with

$$B = PAP^T \, , \quad y = Px \, , \quad c = Pb \, . \tag{4.9b}$$

An ordering is said to be optimal with respect to fill, if it results in the least possible fill-in, and optimal with respect to operation count if it minimizes the number of operations required to solve (4.9a) by Gaussian elimination. If A is an $N \times N$ matrix, there are $N!$ different orderings of its rows and columns, and the problem of finding the ordering with least fill-in is NP complete. Efforts have therefore focused on obtaining efficient algorithms for finding suboptimal orderings with low fill-in and operation count. Numerous reordering algorithms have been developed based on results from graph theory. This topic is discussed in detail in books by George and Liu [42], and Duff, Erisman and Reid [32].

V. Relaxation Methods and Their Acceleration

A general mechanism for constructing iterative algorithms for the solution of discretized elliptic PDEs consists in using relaxation. In this approach, instead of requiring that the entire system (3.1) of discretized equations should be satisfied, we force only one or a few equations to hold at any given time. For the case of a single equation, the value of the variable u_{n_x,n_y} is updated by forcing the discretization equation to hold at point $(n_x h, n_y h)$, while relaxing it at all other points of the discretization grid Ω_h. By using this procedure sequentially, or if possible in parallel, for all points of Ω_h, an updated value of the solution is obtained at all grid points, and one can then proceed to the next iteration. If the resulting iterative algorithm converges, the complete system (3.1) of discretized equations will eventually be satisfied.

In this section, we describe elementary relaxation methods, such as the Jacobi and Gauss-Seidel iterations, and use a digital filtering viewpoint to analyze their convergence behavior. The major shortcoming of these methods is their slow convergence rate. Several acceleration schemes have been proposed to improve their convergence. Acceleration schemes can be divided into two categories, depending on whether they are stationary or not. In a stationary scheme, the same acceleration procedure is used at each iteration. Thus, we can focus on a single iteration and try to optimize its performance. The best example of such a procedure is the successive over-relaxation (SOR) method. In a nonstationary scheme, the overall performance of the algorithm is optimized by considering more than one iteration at a time. Examples of such schemes include the Chebyshev semi-iterative (CSI) and conjugate gradient (CG) methods. Both stationary and nonstationary acceleration methods are discussed below.

A. Elementary Relaxation Methods

Consider the discretization (3.5) of the model Poisson problem. The *Jacobi relaxation* is given by

$$u_{n_x,n_y}^{(m+1)} = \frac{1}{4}\left(u_{n_x+1,n_y}^{(m)} + u_{n_x-1,n_y}^{(m)} + u_{n_x,n_y+1}^{(m)} + u_{n_x,n_y-1}^{(m)} - h^2 f_{n_x,n_y} \right), \quad (5.1)$$

where $u_{n_x,n_y}^{(m)}$ denotes the value of the variable u_{n_x,n_y} at the mth iteration, with $m = 0, 1, 2, \cdots$. From (5.1), we see that given the values $u_{n_x,n_y}^{(m)}$ at all points of

Ω_h, the value $u_{n_x,n_y}^{(m+1)}$ at the next iteration is obtained by forcing equation (3.5) to be locally satisfied at $(n_x h, n_y h)$, independently of whether it is violated at other points of Ω_h.

One way to modify the Jacobi relaxation (5.1) is to partition the grid points into two red and black groups as described in Section III and to perform the iteration

(n_x, n_y) red:

$$u_{n_x,n_y}^{(m+1)} = \frac{1}{4}(u_{n_x+1,n_y}^{(m)} + u_{n_x-1,n_y}^{(m)} + u_{n_x,n_y+1}^{(m)} + u_{n_x,n_y-1}^{(m)} - h^2 f_{n_x,n_y}) , \qquad (5.2a)$$

(n_x, n_y) black:

$$u_{n_x,n_y}^{(m+1)} = \frac{1}{4}(u_{n_x+1,n_y}^{(m+1)} + u_{n_x-1,n_y}^{(m+1)} + u_{n_x,n_y+1}^{(m+1)} + u_{n_x,n_y-1}^{(m+1)} - h^2 f_{n_x,n_y}) . \qquad (5.2b)$$

Thus, one iteration consists of two steps. In the first step, a Jacobi relaxation is performed at all the red points and in the second step, the values obtained at the red points in the first step are used to perform a Jacobi relaxation at the black points. The iteration (5.2) is known as the Gauss-Seidel relaxation for the red-black ordering. The reader is referred to [71] for a detailed comparison of the red-black Gauss-Seidel and Jacobi relaxations.

To analyze the convergence behavior of relaxation methods, it is convenient to view each iteration as corresponding to a digital filtering operation on the solution error. For example, if the Jacobi relaxation converges, the iteration equation (5.1) reduces asymptotically to

$$\overline{u}_{n_x,n_y} = \frac{1}{4}(\overline{u}_{n_x+1,n_y} + \overline{u}_{n_x-1,n_y} + \overline{u}_{n_x,n_y+1} + \overline{u}_{n_x,n_y-1} - h^2 f_{n_x,n_y}) , \qquad (5.3)$$

where $\overline{u}_{n_x,n_y}$ is the exact solution of the discretized problem. Subtracting (5.3) from (5.1), we find that the errors evolve according to

$$e_{n_x,n_y}^{(m+1)} = \frac{1}{4}(E_x + E_x^{-1} + E_y + E_y^{-1})\, e_{n_x,n_y}^{(m)} , \qquad (5.4)$$

where

$$e_{n_x,n_y}^{(m)} = u_{n_x,n_y}^{(m)} - \overline{u}_{n_x,n_y} \qquad (5.5)$$

is the error at the mth iteration. Thus, the Jacobi relaxation can be viewed as a digital filtering process, where at each iteration the FIR filter

$$J = \frac{1}{4}(E_x + E_x^{-1} + E_y + E_y^{-1}) \tag{5.6}$$

is applied to the errors obtained at the previous iteration. Assume that the boundary conditions for the Poisson problem are of Dirichlet type, so that the errors are zero on the domain boundary. To analyze (5.4) in the Fourier domain, we observe that the functions

$$\sin(k_x \pi n_x h)\sin(k_y \pi n_y h)\ , \quad 1 \le k_x,\ k_y \le M{-}1\ ,$$

with $M = h^{-1}$, are eigenfunctions of J which are zero on the domain boundary. They can therefore be used to expand the errors $e_{n_x,n_y}^{(m)}$ in the form (3.12). In the Fourier domain, the iteration (5.4) is diagonalized and takes the form

$$\hat{e}_{k_x,k_y}^{(m+1)} = \hat{J}(k_x,k_y)\hat{e}_{k_x,k_y}^{(m)}\ , \tag{5.7}$$

where the eigenvalues

$$\hat{J}(k_x,k_y) = \frac{1}{2}[\cos(k_x \pi h) + \cos(k_y \pi h)] \tag{5.8}$$

specify the spectrum of J. The spectrum magnitude $|\hat{J}(k_x,k_y)|$ is plotted in Fig. 5.1. We see from this figure that the Jacobi relaxation acts as a bandpass filter. It filters out the middle frequencies, but dampens only slightly the low and high frequencies. Since $|\hat{J}(k_x,k_y)| < 1$ for all feasible wavenumbers, the Jacobi relaxation converges. Its convergence rate is determined by the spectral radius

$$\rho(J) = \max_{1 \le k_x,k_y \le M-1} |\hat{J}(k_x,k_y)| = \cos(\pi h) \approx 1 - \frac{1}{2}\pi^2 h^2\ . \tag{5.9}$$

We see from (5.9) that the number of Jacobi iterations required to reduce the error by a constant factor is proportional to $O(h^{-2})$. In order to determine the total number of iterations needed for convergence, it is useful to observe that since the discretized system is only an approximation of the original continuous problem, the iteration can be stopped when the solution error for the discretized system is of the same order as the discretization error. We saw in Section II.C that the error for a 5-point discretization of the Laplacian is $O(h^2)$. The total number of iterations required by the Jacobi relaxation is therefore $O(h^{-2}\log(h^{-1}))$.

Similarly, denoting by $e_{r,n_x,n_y}^{(m)}$ and $e_{b,n_x,n_y}^{(m)}$ the restriction of the error at the m-th iteration to the red and black points, respectively, we find that the errors for the red-black Gauss-Seidel relaxation evolve according to

$$\begin{pmatrix} e_{r,n_x,n_y}^{(m+1)} \\ e_{b,n_x,n_y}^{(m+1)} \end{pmatrix} = G_{rb} \begin{pmatrix} e_{r,n_x,n_y}^{(m)} \\ e_{b,n_x,n_y}^{(m)} \end{pmatrix} , \tag{5.10a}$$

where

$$G_{rb} = \begin{bmatrix} 1 & 0 \\ J & 0 \end{bmatrix} \begin{bmatrix} 0 & J \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & J \\ 0 & J^2 \end{bmatrix} \tag{5.10b}$$

is the red-black Gauss-Seidel relaxation operator. For Dirichlet boundary conditions, the red and black error functions admit a Fourier decomposition of the form (3.18). With respect to this basis, the error equation (5.10) decouples into a set of 2×2 matrix equations

$$\begin{pmatrix} \hat{e}_{r,k_x,k_y}^{(m+1)} \\ e_{b,k_x,k_y}^{(m+1)} \end{pmatrix} = \hat{G}_{rb}(k_x,k_y) \begin{pmatrix} \hat{e}_{r,k_x,k_y}^{(m)} \\ \hat{e}_{b,k_x,k_y}^{(m)} \end{pmatrix} , \tag{5.11a}$$

$$\hat{G}_{rb}(k_x,k_y) = \begin{bmatrix} 0 & \hat{J}(k_x,k_y) \\ 0 & \hat{J}^2(k_x,k_y) \end{bmatrix} , \tag{5.11a}$$

with $(k_x,k_y) \in K_b$, where K_b is defined in (3.19a). The spectral radius of G_{rb} is therefore given by

$$\rho(G_{rb}) = \max_{k_x,k_y \in K_b} | \hat{J}^2(k_x,k_y) | = \cos^2(\pi h) \approx 1 - \pi^2 h^2 . \tag{5.12}$$

Comparing (5.9) and (5.12), we see immediately that the convergence rate of the red-black Gauss-Seidel algorithm is double that of the Jacobi relaxation. Since both algorithms require the same number of operations per iteration, the red-black Gauss-Seidel algorithm is twice as efficient.

If the natural ordering is adopted, the Gauss-Seidel relaxation takes the form

$$u_{n_x,n_y}^{(m+1)} = \frac{1}{4}(u_{n_x+1,n_y}^{(m)} + u_{n_x-1,n_y}^{(m+1)} + u_{n_x,n_y+1}^{(m)} + u_{n_x,n_y-1}^{(m+1)} - h^2 f_{n_x,n_y}) , \tag{5.13}$$

and is called the lexicographic Gauss-Seidel iteration. The errors dynamics are given by

$$e_{n_x,n_y}^{(m+1)} = G_{lex} e_{n_x,n_y}^{(m)} , \tag{5.14a}$$

where

$$G_{lex} = \frac{E_x + E_y}{4 - E_x^{-1} - E_y^{-1}} \tag{5.14b}$$

is a causal IIR filter. The spectral analysis of the operator G_{lex} with Dirichlet boundary conditions has been performed by Frankel [38], and was studied by Trefethen and LeVeque [73] from a tilted grid viewpoint. For convenience, we consider here the case of periodic boundary conditions. Then, the eigenfunctions of G_{lex} are

$$e^{i2\pi(k_x n_x + k_y n_y)h} \ , \quad 0 \leq k_x , \, k_y \leq M-1 \ ,$$

and, decomposing the errors with respect to this basis, the spectrum of G_{lex} is given by

$$\hat{G}_{lex}(k_x, k_y) = \frac{e^{i2\pi k_x h} + e^{i2\pi k_y h}}{4 - e^{-i2\pi k_x h} - e^{-i2\pi k_y h}} \ . \tag{5.15}$$

Note that $\hat{G}_{lex}(k_x, k_y) = 1$ for $(k_x, k_y) = (0,0)$ and $|\hat{G}_{lex}(k_x, k_y)| < 1$ for all other feasible wavenumbers. This means that the filter G_{lex} does not filter out the d.c. component of the error. However, if $u(x,y)$ is a solution of the model periodic problem, $u(x,y)$ plus a constant is also a solution, and the lexicographic Gauss-Seidel method converges to one of these solutions.

To summarize, the Jacobi, red-black, and lexicographic Gauss-Seidel relaxations admit a digital filtering interpretation, where each iteration consists in applying a filter to the errors obtained at the previous iteration. This filtering process can be studied easily in the frequency domain, by decomposing the errors in terms of properly selected Fourier eigenmodes, and examining each mode independently.

B. SOR Acceleration

The red-black SOR iteration is obtained by introducing a relaxation parameter ω inside the Gauss-Seidel iteration (5.2), i.e.,

(n_x, n_y) red:

$$\begin{aligned} u_{n_x,n_y}^{(m+1)} &= (1-\omega) u_{n_x,n_y}^{(m)} \\ &+ \frac{\omega}{4}(u_{n_x+1,n_y}^{(m)} + u_{n_x-1,n_y}^{(m)} + u_{n_x,n_y+1}^{(m)} + u_{n_x,n_y-1}^{(m)} - h^2 f_{n_x,n_y}) \ , \end{aligned} \tag{5.16a}$$

(n_x, n_y) black:

$$u_{n_x,n_y}^{(m+1)} = (1-\omega) u_{n_x,n_y}^{(m)}$$

$$+ \frac{\omega}{4}(u_{n_x+1,n_y}^{(m+1)} + u_{n_x-1,n_y}^{(m+1)} + u_{n_x,n_y+1}^{(m+1)} + u_{n_x,n_y-1}^{(m+1)} - h^2 f_{n_x,n_y}) . \qquad (5.16b)$$

When $\omega = 1$, the SOR method reduces to the Gauss-Seidel method. The error dynamics for the SOR iteration can be expressed as

$$\begin{pmatrix} e_{r,n_x,n_y}^{(m+1)} \\ e_{b,n_x,n_y}^{(m+1)} \end{pmatrix} = G_{rb}(\omega) \begin{pmatrix} e_{r,n_x,n_y}^{(m)} \\ e_{b,n_x,n_y}^{(m)} \end{pmatrix} , \qquad (5.17a)$$

where

$$G_{rb}(\omega) = \begin{bmatrix} 1 & 0 \\ \omega J & 1-\omega \end{bmatrix} \begin{bmatrix} 1-\omega & \omega J \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1-\omega & \omega J \\ (1-\omega)\omega J & 1-\omega+\omega^2 J^2 \end{bmatrix} \qquad (5.17b)$$

is the red-black SOR iteration operator. With respect to the red-black Fourier decomposition (3.18), the SOR iteration reduces to 2×2 matrix iterations of the form (5.11a), where $\hat{G}_{rb}(k_x,k_y)$ is replaced by

$$\hat{G}_{rb}(\omega,k_x,k_y) = \begin{bmatrix} 1-\omega & \omega \hat{J}(k_x,k_y) \\ (1-\omega)\omega \hat{J}(k_x,k_y) & 1-\omega+\omega^2 \hat{J}^2(k_x,k_y) \end{bmatrix} . \qquad (5.18)$$

Let λ be an eigenvalue of the matrix $\hat{G}_{rb}(\omega,k_x,k_y)$, and let $\mu = \hat{J}(k_x,k_y)$. Then, λ and μ are related via the quadratic equation

$$\lambda^2 - 2(1-\omega+\frac{\omega^2\mu^2}{2})\lambda + (1-\omega)^2 = 0 . \qquad (5.19)$$

Note that as ω varies, the eigenvalues λ_1 and λ_2 move about the complex plane. We are interested in how the quantity $\rho = \max(\,|\lambda_1|, |\lambda_2|)$ depends on ω. When viewed as a function of ω, the discriminant

$$\Delta = 4(1-\omega)\omega^2\mu^2 + \omega^4\mu^4$$

of (5.19) has a single real root which is given by

$$\omega_d = \frac{2}{1+\sqrt{1-\mu^2}} . \qquad (5.20)$$

It is easy to check that $\rho < 1$ if and only if $0 < \omega < 2$. Furthermore, we have

$$\rho = \frac{1}{4}[\omega\,|\mu| + \sqrt{\omega^2\mu^2+4(1-\omega)}]^2 , \quad \text{for } 0 \le \omega \le \omega_d$$

$$\rho = \omega - 1 , \qquad \text{for } \omega_d \le \omega \le 2 .$$

The locus of eigenvalues λ_1 and λ_2 as ω varies is plotted in Fig. 5.2. When $\omega = 0$, the eigenvalues λ_1 and λ_2 coincide at the value 1. As ω increases from 0 to 1, both eigenvalues move toward the origin along the real line but with different speeds. When ω reaches 1, the eigenvalues are 0 and μ^2. When $1 < \omega \leq \omega_d$, one eigenvalue increases its value from 0 and the other continues to decrease. They coincide again at the point $\omega_d - 1$ when $\omega = \omega_d$. The eigenvalues become complex conjugate pair with magnitude $\omega \ - 1$ for $\omega > \omega_d$. Thus, these eigenvalues lie outside of the unit circle for $\omega > 2$. This plot shows that the spectral radius ρ is minimized for $\omega = \omega_d$.

Since $\mu = \hat{J}(k_x, k_y)$, the relaxation parameter ω_d which minimizes the spectral radius of $\hat{G}_{rb}(\omega, k_x, k_y)$ is a function of the wavenumber (k_x, k_y). In order to minimize the spectral radius of the space-domain operator $G_{rb}(\omega)$, we must therefore select for ω the value which minimizes the maximum over all feasible wavenumbers of the spectral radius of $\hat{G}(\omega, k_x, k_y)$. A straightforward analysis [71] shows that the optimal relaxation parameter ω_{opt} is given by the value of ω_d corresponding to the wavenumber $(k_x, k_y) = (1,1)$. Since $\hat{J}(1,1) = \cos(\pi h)$, we obtain

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \cos^2(\pi h)}} \approx 2 - 2\pi h \ , \tag{5.21}$$

and the corresponding spectral radius is

$$\rho(G_{rb}(\omega_{opt})) = \omega_{opt} - 1 \approx 1 - 2\pi h \ . \tag{5.22}$$

We see from (5.22) that the number of iterations required by the red-black SOR iteration to reduce the error by a constant factor is $O(h^{-1})$, so that this algorithm is one order of magnitude faster than the Jacobi or red-black Gauss-Seidel relaxations. However, this rate of convergence is achieved only when the relaxation parameter is equal to its optimal value ω_{opt}, and is sensitive to perturbations of the relaxation parameter away from this value.

An interesting feature of the SOR method is that, since the optimum relaxation parameter ω_{opt} is larger than ω_d for all wavenumber components $(k_x, k_y) \neq (1,1)$, the eigenvalues of $G_{rb}(\omega_{opt})$ have all the same magnitude $\omega_{opt} - 1$. To illustrate this phenomenon, the spectra of the Jacobi and SOR (with ω_{opt}) iteration matrices are plotted in Fig. 5.3. The eigenvalues of the

Jacobi iteration matrix are all real and occur in ± pairs. Their magnitude ranges from 0 to $\cos(\pi h) = 1-O(h^2)$. Thus, different Fourier components converge at different rates, and the slowest converging Fourier component is the one that establishes the convergence rate of the Jacobi method. Through the SOR acceleration, these eigenvalues are redistributed around a circle of radius $\omega_{opt} - 1 = 1-O(h)$ in the complex plane. Since they have the same magnitude, all Fourier components converge at the same rate. Thus the acceleration effect of the SOR method is achieved by balancing the convergence rates of the different Fourier components.

C. Polynomial Acceleration

The SOR procedure is a stationary one-step acceleration technique, in the sense that it optimizes the convergence behavior of one iteration, and uses the same acceleration scheme at every subsequent iteration. There exists an alternative acceleration approach which optimizes the convergence behavior of the overall algorithm, instead of considering only one step. Specifically, if a given iterative procedure requires s steps to converge, we can select a set of acceleration parameters ω_i with $1 \leq i \leq s$ and apply ω_i at the ith iteration to increase the convergence rate. This approach leads to the polynomial acceleration method described below.

Consider the sequence of iterates generated by the iteration

$$w^{(m+1)} = Pw^{(m)} + c \ , \tag{5.23}$$

where P is assumed to have real eigenvalues, and $\rho(P) < 1$, so that (5.23) converges. For example, one possible choice for P is the Jacobi iteration matrix J. The error $e_w^{(m)} = w^{(m)} - \overline{w}$ at the mth iteration is given by

$$e_w^{(m)} = P^m \, e_w^{(0)} \ . \tag{5.24}$$

To improve the convergence of the sequence $\{w^{(m)}\}$, we can generate a new sequence $\{u^{(m)}\}$ by performing a linear combination

$$u^{(m)} = \sum_{i=0}^{m} \alpha_{m,i} \, w^{(i)} \ , \tag{5.25}$$

where the coefficients $\alpha_{m,i}$ are real and satisfy

$$\sum_{i=0}^{m} \alpha_{m,i} = 1 \tag{5.26}$$

for all m. This condition is imposed in order to guarantee that when $w^{(0)} = \overline{u}$, then $u^{(m)} = \overline{u}$ for $m \geq 0$. Let $e^{(m)}$ be the error associated with the new sequence $u^{(m)}$. From (5.24) and (5.25), we can relate $e^{(m)}$ and $e_w^{(m)}$ via

$$e^{(m)} = Q_m(P)e_w^{(0)}, \quad \text{where } Q_m(P) = \sum_{i=0}^{m} \alpha_{m,i} P^m$$

is a matrix polynomial of degree m. Since $e^{(0)} = e_w^{(0)}$, the errors associated with the $\{u^{(m)}\}$ iteration satisfy

$$e^{(m)} = Q_m(P)e^{(0)} . \tag{5.27}$$

The problem is to select the coefficients $\alpha_{m,i}$ so that the error sequence $\{e^{(m)}\}$ converges to zero as fast as possible.

Since $Q_m(P)$ is a polynomial function of P, it has the same eigenvectors as P, and if μ is an eigenvalue of P, the eigenvalue of $Q_m(P)$ corresponding to the same eigenvector is $Q_m(\mu)$. Let S be the discrete spectrum of the matrix P, and let μ_{min} and μ_{max} denote the smallest and largest eigenvalues of P. The polynomial acceleration problem can be formulated as the minimax problem

$$\min_{\alpha_{m,i}} \max_{x \in S} |Q_m(x)| . \tag{5.28}$$

Since the discrete spectrum S is seldom known, the problem (5.28) cannot usually be solved as such. A modified version which is easier to solve consists in replacing S in (5.28) by the continuous spectrum $\overline{S} = \{x : \mu_{\text{min}} \leq x \leq \mu_{\text{max}}\}$. In this case, we can perform the change of variable

$$z(x) = \frac{2x - (\mu_{\text{max}}+\mu_{\text{min}})}{\mu_{\text{max}} - \mu_{\text{min}}} , \tag{5.29}$$

so that (5.28) is transformed into a minimax problem defined on the interval [−1,1]. The solution of this new minimax problem is well known and is given by the Chebyshev polynomial of order m, $T_m(z)$. In terms of the original variable x, the solution is

$$Q_m(x) = T_m(z(x))/T_m(z(1)), \tag{5.30}$$

where the scaling by $T_m(z(1))$ ensures that the coefficient constraint (5.26) is satisfied.

An interesting property of Chebyshev polynomials is that they satisfy the three-term recurrence relation

$$T_{m+1}(z) = 2zT_m(z) - T_{m-1}(z)\ , \quad m \geq 1\ , \tag{5.31}$$

with $T_0(z) = 1$ and $T_1(z) = z$. This property can be exploited to generate the new sequence $\{u^{(m)}\}$ efficiently, instead of using expression (5.25), which has a high computational cost, and requires a large amount of storage. By taking into account the recursions (5.23) and (5.31) inside (5.25), we obtain the following Chebyshev semi-iterative (CSI) acceleration procedure [51],[87] for iteration (5.23):

$$u^{(m+1)} = \beta_{m+1}[\gamma(Pu^{(m)} + c) + (1-\gamma)u^{(m)}] + (1-\beta_{m+1})u^{(m-1)}\ , \tag{5.32}$$

with

$$\gamma = \frac{2}{2-\mu_{\max}-\mu_{\min}}\ , \tag{5.33a}$$

$$\beta_1 = 1\ , \ \beta_2 = (1 - ½\sigma^2)^{-1}\ , \ \beta_{m+1} = (1 - ¼\sigma^2\beta_m)^{-1}\ , \ m \geq 2\ , \tag{5.33b}$$

and

$$\sigma = \frac{\mu_{\max}-\mu_{\min}}{2-\mu_{\max}-\mu_{\min}}\ . \tag{5.33c}$$

To illustrate the redistribution of the eigenvalues of P which is accomplished by the CSI acceleration method, the function $Q_{10}(x)$ describing how the eigenvalues of $Q_m(P)$ depend on those of P for $m = 10$ is plotted in Fig. 5.4. From this figure, we see that unlike the SOR method, where the eigenvalues of $G_{rb}(\omega_{opt})$ were all complex and equal in magnitude, the eigenvalues of the CSI matrix $Q_m(P)$ remain real, and lie in the narrow interval

$$[-2r^{m/2}/(1+r^m)\ , \ 2r^{m/2}/(1+r^m)]\ ,$$

with

$$r = \frac{1 - \sqrt{1-\sigma^2}}{1 + \sqrt{1-\sigma^2}}\ . \tag{5.34}$$

As an example, consider the case where the CSI method is used to accelerate the Jacobi iteration for the model Poisson problem with Dirichlet conditions, so that $P = J$ in (5.23). The resulting algorithm is called the J-CSI method. The asymptotic convergence rate of the J-CSI method can be determined as follows.

From (5.8), we know that

$$\mu_{\max} = \cos(\pi h) \quad \mu_{\min} = -\cos(\pi h) , \tag{5.35}$$

and from (5.33c),

$$\sigma = \cos(\pi h) .$$

Then, observing from Fig. 5.4 that the maximum value of $|Q_m(x)|$ over the interval $[\mu_{\min} , \mu_{\max}]$ is reached for $x = \mu_{\max}$, we find that

$$\rho(Q_m(J)) = |Q_m(\mu_{\max})| = 2\frac{r^{m/2}}{1+r^m} , \tag{5.36a}$$

where, from (5.34),

$$r = \frac{1-\sin(\pi h)}{1+\sin(\pi h)} . \tag{5.36b}$$

According to (5.27), the error of the J-CSI method at the m-th iteration is obtained by multiplying the initial error by $Q_m(J)$. The asymptotic error contraction factor per iteration is therefore

$$\lim_{m \to \infty} \left(\rho(Q_m(J))^{\frac{1}{m}} \right) = r^{1/2} \approx 1 - \pi h . \tag{5.37}$$

This shows that the J-CSI method requires $O(h^{-1})$ iterations to reduce the error by a constant factor. A further improvement in this algorithm was introduced by Golub and Varga [46], who observed that for the the red-black ordering, the recursion (5.32) can be rearranged in such a way that only the odd iterates of the red points and the even iterates of the black points need to be computed, thus cutting the numerical complexity of the algorithm in half. The resulting procedure is called the cyclic CSI method, and its numerical complexity is the same as that of the SOR method.

D. Historical Notes

The development of relaxation methods for the solution of large systems of linear equations was initiated by Gauss, Jacobi, and Seidel in the 19th century, and Ridchardson, Liebmann, and Southwell early in this century. Since a comprehensive account of the history of relaxation methods can be found in a recent paper by Young [94], our comments focus primarily on the application of

Fourier analysis to the study of these methods. The development of the SOR theory in the late 1940s [38],[91],[92] marked the beginning of a period of rapid progress in the area of iterative methods. The Fourier approach adopted in this section has for origin the work of Frankel [38] and Young [91], who used Fourier-like basis functions to analyze the SOR method applied to the naturally ordered Poisson problem with Dirichlet boundary conditions. Recently, LeVeque and Trefethen [73] reinterpreted Frankel's result from a tilted grid viewpoint. The same problem with periodic boundary conditions was analyzed by Chan and Elman [22]. The two-color Fourier analysis of the SOR method for the red-black ordered model Poisson problem with Dirichlet or periodic boundary conditions was developed by Kuo et al. [66],[71]. The use of Chebyshev polynomials was first proposed by Flanders and Shortley [37] for the solution of matrix eigenvalue problems, and subsequently led to the development of the Chebyshev semi-iterative (CSI) method for solving linear systems. A complete discussion of elementary relaxation methods and of the SOR and CSI acceleration procedures can be found in books by Birkhoff and Lynch [12], Hageman and Young [51], Varga [87], and Young [93].

VI. Multigrid Methods

The major limitation of elementary and accelerated relaxation methods is that while the components of the error decrease very rapidly in certain frequency bands, they decay only very slowly in other bands. The region of rapid decay depends on the specific relaxation method that we consider, but it consists typically of middle or high frequencies. On the other hand, the region of slow decay always includes the low frequencies. This phenomenon reflects the fact that the low frequency components of the solution depend on global information, and a large number of iterations are required for propagating information from the edges of the problem domain to its center. Since the error becomes progressively smoother as the iteration proceeds, it is natural to consider switching to a coarser discretization grid, where we can assume temporarily that an exact solver is available. This solver can be used to compute the smooth components of the error on the coarse grid, and the resulting correction can then be interpolated back to the fine grid and combined with the original fine grid solution. Such a solution scheme is called a *two-grid method.* In this approach, the fine grid provides the accuracy required by the approximation while the coarse grid offers a faster convergence rate for the low frequency Fourier components. Naturally, the weakness of the above scheme is that we have assumed that an exact solver is available on the coarse grid. This is generally an unreasonable assumption, but we need only to observe that the problem on the coarse grid can itself be solved by a two-grid method. By proceeding recursively, we obtain a multigrid scheme, where progressively coarser grids are employed, until so few discretization points are involved that a direct solver can be used to compute the error on the coarsest grid. The resulting solution technique is called a *multigrid method.*

Since the two-grid method is the main component of multigrid methods, our first step in this section is to perform a detailed analysis of the two-grid iteration operator. We use two-color Fourier analysis to find the spectrum of this operator for the 1-D and 2-D model Poisson problems. Then, we describe several of the standard recursion patterns, namely the V-cycle, W-cycle, and full-multigrid schemes, that are used to generate multigrid methods from the two-grid iteration.

A. Two-grid Iteration

Consider two discretization grids Ω_h and Ω_{2h}, with mesh sizes h and $2h$, respectively, and let

$$L_h u_h = f_h \tag{6.1}$$

be the equation that we seek to solve on the fine grid, where L_h, f_h and u_h denote the discretized operator, forcing function, and solution, respectively. An $(h, 2h)$ two-grid iteration for solving this equation consists of the following three steps.

Step 1: Presmoothing. Select a relaxation operator S_h for solving (6.1) on the fine grid. Typically, S_h is the Gauss-Seidel relaxation, but other choices are possible, such as the damped Jacobi iteration described below. Then, given an initial estimate $u_h^{(0)}$ of the solution, apply the S_h iteration ν_1 times. If $u_h^{(1)}$ denotes the resulting approximate solution, the corresponding residual is

$$r_h = f_h - L_h u_h^{(1)} . \tag{6.2}$$

Step 2: Coarse-grid correction. The residual r_h can be projected onto the coarse grid Ω_{2h} by using a restriction operator I_h^{2h}, thus yielding $f_{2h} = I_h^{2h} r_h$. Then, since we assume that an exact solver is available on Ω_{2h}, we use this solver, which is denoted here by L_{2h}^{-1}, to find the solution u_{2h} of the coarse grid problem

$$L_{2h} u_{2h} = f_{2h} . \tag{6.3}$$

If I_{2h}^h denotes an interpolation operator for transferring a function defined on Ω_{2h} onto the fine grid Ω_h, we can interpolate the coarse grid correction u_{2h}, and add it to the solution obtained in Step 1, thus yielding

$$u_h^{(2)} = u_h^{(1)} + I_{2h}^h u_{2h} . \tag{6.4}$$

Step 3: Postsmoothing. Using $u_h^{(2)}$ as initial solution, we apply the S_h iteration ν_2 times. The resulting approximate solution is $u_h^{(3)}$.

The above three steps are illustrated in Fig. 6.1. Usually, the numbers ν_1 and ν_2 of pre- and post-smoothing iterations are 0, 1 or 2, and $\nu = \nu_1 + \nu_2$ is 2 or 3. If

$$e^{old} = u_h^{old} - u_h^{(0)} \quad , \quad e^{new} = u_h^{new} - u_h^{(3)} , \tag{6.5a}$$

are the solution errors before and after one full two-grid iteration, the error dynamics for the two-grid iteration can be expressed as

$$e^{new} = M_h^{2h} e^{old} \ , \tag{6.5b}$$

where the two-grid iteration operator M_h^{2h} is given by

$$M_h^{2h} = S_h^{\nu_2} K_h^{2h} S_h^{\nu_1} , \tag{6.6}$$

and K_h^{2h} is the coarse grid correction operator

$$K_h^{2h} = I_h - I_{2h}^h L_{2h}^{-1} I_h^{2h} L_h \ . \tag{6.7}$$

Naturally, the two-grid iteration needs to be repeated until the error becomes sufficiently small. It will be shown below that the two-grid iteration operator M_h^{2h} reduces the error by a constant factor independent of h, so that only $O(\log(h^{-1}))$ iterations are necessary to solve (6.1) within the discretization accuracy $O(h^p)$, where p is a positive integer.

Note that equations (6.2)-(6.6) provide only a general description of the two-grid iteration procedure. In order to obtain an actual two-grid iteration, we need to select the operators S_h, I_h^{2h}, I_{2h}^h, and L_{2h} which have been left unspecified in the above description. In spite of the fact that there exist many different ways to choose these operators and that they need to be adjusted to achieve the best convergence performance for different applications, the efficiency of multigrid methods does not usually depend on this choice. It is the utilization of multiple discretization grids that makes these methods converge very rapidly. In the following subsections, S_h is the red-black Gauss-Seidel iteration operator, L_{2h} is the usual 3-point (resp. 5-point) discretization of the 1-D (resp. 2-D) Poisson operator on the grid Ω_{2h}, and I_h^{2h} and I_{2h}^h are the full weighting restriction and linear interpolation operators, respectively.

B. Solution of the 1-D Poisson Problem

Two-grid method and analysis: Consider an $(h, 2h)$ two-grid method for solving the discretized 1-D Poisson equation

$$\frac{1}{h^2}(u_{n-1} - 2u_n + u_{n+1}) = f_n \quad , \quad 1 \le n \le N-1 \ , \tag{6.8}$$

where the boundary values u_0 and u_N are given, h is the grid spacing, and $N = h^{-1}$ is even. For the 1-D problem (6.8), it will be shown below that it is possible to choose the relaxation, restriction and interpolation operators so that

$M_h^{2h} = 0$. This means that the two-grid method is a direct solver for (6.8). However, this is not true in general for 2-D or 3-D problems.

Quite often, a simple but crude technique, called the *smoothing rate analysis* [16], can be used to study the convergence behavior of two-grid or multigrid methods. This analysis assumes that the coarse-grid correction operator K_h^{2h} annihilates all the low frequency components of the error and preserves its high frequency components, i.e.,

$$\hat{K}_h^{2h}(k) = \begin{cases} 0\,, & 1 \le k < N/2 \\ 1\,, & N/2 \le k \le N-1\,. \end{cases} \tag{6.9}$$

By expressing (6.6) in the frequency domain and using assumption (6.9), we find that the two-grid iteration operator admits the frequency domain representation

$$\hat{M}_h^{2h}(k) = \begin{cases} 0\,, & 1 \le k < N/2 \\ \hat{S}_h^{\nu_1+\nu_2}(k)\,, & N/2 \le k \le N-1\,, \end{cases} \tag{6.10}$$

where $\hat{S}_h(k)$ denotes the spectrum of S_h. The largest magnitude μ of $\hat{S}_h(k)$ for $N/2 \le k \le N-1$ is called the *smoothing factor.* Therefore, the convergence rate of the two-grid method is related to the smoothing factor via

$$\rho(M_h^{2h}) = \mu^{\nu_1+\nu_2} \tag{6.11}$$

To give an example, consider the damped Jacobi iteration,

$$u_n^{(m+1)} = (1-\omega)u_n^{(m)} + \frac{\omega}{2}(u_{n+1}^{(m)} + u_{n-1}^{(m)} - h^2 f_n)\,, \tag{6.12}$$

where ω is a relaxation parameter. The damped Jacobi smoother has the spectrum

$$\hat{J}(\omega,k) = (1-\omega) + \omega\cos(k\pi h)\,, \tag{6.13}$$

whose magnitude parameterized with ω is plotted in Fig. 6.2. We can choose ω to minimize the magnitude of the largest eigenvalue in the high frequency region. The optimal relaxation parameter is $\omega = 2/3$, which is obtained by solving

$$\hat{J}(\omega,\frac{N}{2}) = -\hat{J}(\omega,N)\,, \tag{6.14}$$

and the corresponding smoothing rate is

$$\mu = \max_{N/2 \le k \le N-1} |\hat{J}(\frac{2}{3},k)| = \frac{1}{3}\,. \tag{6.15}$$

The estimated two-grid convergence rate becomes

$$\rho(M_h^{2h}) = (\frac{1}{3})^{\nu_1+\nu_2} . \tag{6.16}$$

We should point out that the assumption (6.9) for the smoothing rate analysis does not actually hold in practice. However, because of its simplicity, this analysis is often useful for estimating the convergence behavior of multigrid methods.

There are situations where the smoothing rate analysis predicts completely wrong results. One such case arises when the red-black Gauss-Seidel relaxation is used as smoother. Following a procedure similar to the one employed for deriving (5.11), we find that with respect to the coefficients $(\hat{e}_{r,k}, \hat{e}_{b,k})$ of the 1-D red-black Fourier series expansion

$$e_n = \sum_{k=1}^{N/2-1} \hat{e}_{r,k} \sin(k\pi nh) , \quad n \text{ even} , \tag{6.17a}$$

$$e_n = \sum_{k=1}^{N/2} \hat{e}_{b,k} \sin(k\pi nh) , \quad n \text{ odd} , \tag{6.17b}$$

the red-black Gauss-Seidel relaxation operator G_{rb} can be represented as

$$\hat{G}_{rb}(k) = \begin{bmatrix} 0 & \cos(k\pi h) \\ 0 & \cos^2(k\pi h) \end{bmatrix}, \quad 1 \le k \le N/2-1 , \tag{6.18}$$

with $\hat{G}_{rb}(N/2) = 0$. The expression (6.18) holds also for high frequency components ($k > N/2$) which are aliased into the low frequency region. Thus, the red-black Gauss-Seidel smoother attenuates rapidly the middle frequency components ($k \approx N/2$) but works poorly for the low and high frequencies. According to the smoothing rate analysis, we have

$$\max_{N/2 \le k \le N-1} | \hat{G}_{rb}(k) | = \cos^2(\pi h) \approx 1 - \pi^2 h^2 . \tag{6.19}$$

This implies a poor convergence of the corresponding multigrid method. However, contrary to this prediction, numerical experiments show that the multigrid method with the red-black Gauss-Seidel smoother is an exact solver for the 1-D Poisson problem and converges very rapidly in the 2-D case. Thus, in order to explain the effectiveness of the red-black Gauss-Seidel smoother, we cannot assume that the condition (6.9) holds. It is necessary to perform a complete two-grid analysis, i.e., to study the spectrum of the coarse-grid corrector K_h^{2h} defined in (6.7), as well as that of the smoother S_h.

We have first to define more precisely the operators appearing in (6.2)-(6.4). The h-grid and $2h$-grid Laplacians are

$$L_h = \frac{1}{h^2}(E_h^{-1} - 2 + E_h) , \quad \text{and} \quad L_{2h} = \frac{1}{(2h)^2}(E_{2h}^{-1} - 2 + E_{2h}) , \tag{6.20}$$

where $E_{2h} = E_h^2$. To restrict a function from Ω_h to Ω_{2h}, we perform an averaging operation with coefficients 1/4, 1/2 and 1/4 and then down-sample the averaged sequence on Ω_{2h}. The restriction operator is denoted by

$$I_h^{2h} : \quad | \frac{1}{4} , \frac{1}{2} , \frac{1}{4} |_h^{2h} . \tag{6.21a}$$

To interpolate a function from Ω_{2h} to Ω_h, we use a linear interpolation scheme for grid points belonging to $\Omega_h - \Omega_{2h}$. The interpolation operator is written as

$$I_{2h}^h : \quad | \frac{1}{2} , 1 , \frac{1}{2} |_{2h}^h . \tag{6.21b}$$

With respect to the red-black Fourier expansion (6.17), the action of the h-grid discretized Laplacian and identity operator I_h on the red/black Fourier vector $(\hat{e}_{r,k}, \hat{e}_{b,k})^T$ can be represented by the 2×2 matrices

$$\hat{L}_h(k) = \frac{2}{h^2} \begin{bmatrix} -1 & \cos(k\pi h) \\ \cos(k\pi h) & -1 \end{bmatrix} , \qquad \hat{I}_h(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} . \tag{6.22a}$$

Observing that in the 1-D case, the points of the coarse grid coincide with the red points of the fine grid, we find that the red-black spectral representations of the restriction and interpolation operators I_h^{2h} and I_{2h}^h correspond respectively to mappings from $(\hat{e}_{r,k}, \hat{e}_{b,k})supT$ onto $\hat{e}_{r,k}$, and from $\hat{e}_{r,k}$ onto $(\hat{e}_{r,k}, \hat{e}_{b,k})^T$, and are given by

$$\hat{I}_h^{2h}(k) = [\frac{1}{2} , \frac{\cos(k\pi h)}{2}] \quad \text{and} \quad \hat{I}_{2h}^h(k) = \begin{bmatrix} 1 \\ \cos(k\pi h) \end{bmatrix} . \tag{6.22b}$$

Furthermore, with respect to the Fourier component $\hat{e}_{r,k}$ the $2h$-grid discretized Laplacian is represented by the spectrum

$$\hat{L}_{2h}(k) = \frac{2(\cos(2k\pi h) - 1)}{(2h)^2} \tag{6.22c}$$

We obtain therefore

$$\hat{K}_h^{2h}(k) = \hat{I}_h(k) - \hat{I}_{2h}^{h}(k)\hat{L}_{2h}^{-1}(k)\hat{I}_h^{2h}(k)\hat{L}_h(k) = \begin{bmatrix} 0 & 0 \\ -\cos(k\pi h) & 1 \end{bmatrix} . \qquad (6.23)$$

Finally, choosing $S_h = G_{rb}$ and $\nu_1 = \nu_2 = 1$ in (6.6), we find that the red-black spectral representation of the two-grid operator is given by

$$\hat{M}_h^{2h}(k) = \hat{G}_{rb}(k)\hat{K}_h^{2h}(k)\hat{G}_{rb}(k) \, . \qquad (6.24)$$

From (6.18) and (6.23), it is easy to check that $\hat{M}_h^{2h}(k)$ are 2×2 zero matrices for $1 \leq k \leq N/2{-}1$ and $\hat{M}_h^{2h}(N/2) = 0$. Thus, the two-grid method with red-back Gauss-Seidel smoothing is a direct solver.

Multigrid methods: The implementation of the two-grid method requires inverting the coarse-grid Laplacian operator L_{2h}. An efficient way to carry out this inversion is to use a $(2h,4h)$ two-grid iteration. By using nested two-grid iterations, we can therefore reduce the original problem to one defined on progressively coarser grids, until a direct solver can be used to invert the discretized operator on the coarsest grid. Thus, if the mesh-size on the finest grid is $h = 2^{-L}$ with $L > 2$, the following nested iteration specifies an L-grid solver:

$$M_h^{2h} = G_{rb}(I_h - I_{2h}^{h} X_{2h} I_h^{2h})G_{rb} \, , \qquad (6.25a)$$

with

$$X_h = \begin{cases} M_h^{2h} \text{ for } h = 2^{-l} \, , \, 2 \leq l \leq L-1 \\ L_h^{-1} \text{ for } h = 1/2 \end{cases} \qquad (6.25b)$$

One can prove by induction that this multigrid algorithm solves the 1-D Poisson problem directly. It is possible to simplify this algorithm to save computations. See [69] for details.

C. Solution of the 2-D Poisson Problem

Let L_h and L_{2h} be the 5-point discretizations of the Laplacian on Ω_h and Ω_{2h}, i.e.,

$$L_h = \frac{1}{h^2}(E_{h,x} + E_{h,x}^{-1} - 4 + E_{h,y} + E_{h,y}^{-1}) \, , \qquad (6.26a)$$

$$L_{2h} = \frac{1}{(2h)^2}(E_{2h,x} + E_{2h,x}^{-1} - 4 + E_{2h,y} + E_{2h,y}^{-1}) \, . \qquad (6.26b)$$

Then, I_h^{2h} and I_{2h}^h denote the full-weighting restriction and linear interpolation operators, given respectively by

$$I_h^{2h} : \left| \begin{matrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{matrix} \right|_h^{2h} , \qquad (6.27a)$$

and

$$I_{2h}^h : \left| \begin{matrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{matrix} \right|_{2h}^{h} . \qquad (6.27b)$$

We consider only the case $1 \le k_x , k_y < N/2$. Each of the 4×4 frequency domain matrices appearing below corresponds to a mapping from the vector space spanned by

$$(\hat{r}_{\mathbf{k}} , -\hat{r}_{\tilde{\mathbf{k}}} , \hat{b}_{\mathbf{k}} , -\hat{b}_{\tilde{\mathbf{k}}})^T ,$$

onto itself, where

$$\mathbf{k} = (k_x , k_y) \quad 1 \le k_x , k_y < \frac{N}{2} , \quad \tilde{\mathbf{k}} = \begin{cases} (N - k_x , k_y) & \text{for } k_x \ge k_y \\ (k_x , N - k_y) & \text{for } k_x < k_y \end{cases} \qquad (6.28)$$

When k_x or k_y is equal to $N/2$, the 4×4 matrices reduce to 2×2 or 1×1 matrices. The analysis of these degenerate cases can be found in [69] and is omitted here. We also use the abbreviations

$$\alpha = \frac{\cos\theta_x + \cos\theta_y}{2} , \; \tilde{\alpha} = \frac{\cos\tilde{\theta}_x + \cos\tilde{\theta}_y}{2} , \; \beta = \cos\theta_x \cos\theta_y , \; \tilde{\beta} = \cos\tilde{\theta}_x \cos\tilde{\theta}_y \qquad (6.29)$$

where $\theta_x = k_x \pi h$, $\theta_y = k_y \pi h$, $\tilde{\theta}_x = \tilde{k}_x \pi h$, and $\tilde{\theta}_y = \tilde{k}_y \pi h$.

The matrices representing operators I_h, L_h, and L_{2h}^{-1} in the frequency domain can be written as

$$\hat{I}_h(k_x,k_y) = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}, \quad \hat{L}_h(k_x,k_y) = \frac{4}{h^2}\begin{bmatrix} -I & J \\ J & -I \end{bmatrix}, \tag{6.30a}$$

$$\hat{L}_{2h}^{-1}(k_x,k_y) = \frac{h^2}{2\delta}, \quad \delta \equiv 2\alpha^2 - \beta - 1 , \tag{6.30b}$$

where 0 is the 2 × 2 zero matrix, I is the 2 × 2 identity matrix, and

$$J = \mathrm{diag}\,(\,\alpha\,,\,\tilde{\alpha}\,)\,. \tag{6.30c}$$

The decomposition shown in Fig. 6.3, which is commonly used in multirate digital signal processing [28], provides a simple physical interpretation of the interpolation and restriction operators, and is also useful for deriving their frequency domain matrices. In this decomposition, the restriction procedure I_h^{2h} is divided into two steps,

Step 1: Lowpass filtering (or averaging) at every point of Ω_h, where the weighting coefficients are specified by the stencil (6.27a).

Step 2: Down-sampling (or injecting) values from Ω_h to Ω_{2h}.

The interpolation operator I_{2h}^h is also decomposed into two steps,

Step 1: Up-sampling values from Ω_{2h} to Ω_h, where we assign 0 to points which belong to $\Omega_h - \Omega_{2h}$.

Step 2: Lowpass filtering at every point of Ω_h, where the weighting coefficients are specified by the stencil (6.27b).

It is relatively easy to find a frequency domain matrix representation for each of the above steps. Combining them together, we obtain

$$\hat{I}_h^{2h}(\theta) = [\,1\;1\;0\;0\,] \times \frac{1}{4}\begin{bmatrix} 1+\beta & 0 & 2\alpha & 0 \\ 0 & 1+\tilde{\beta} & 0 & 2\tilde{\alpha} \\ 2\alpha & 0 & 1+\beta & 0 \\ 0 & 2\tilde{\alpha} & 0 & 1+\tilde{\beta} \end{bmatrix} = \frac{1}{4}\,[\,1+\beta\;\;1+\tilde{\beta}\;\;2\alpha\;\;2\tilde{\alpha}\,] \tag{6.31a}$$

and

$$\hat{I}_{2h}^h(\theta) = \begin{bmatrix} 1+\beta & 0 & 2\alpha & 0 \\ 0 & 1+\tilde{\beta} & 0 & 2\tilde{\alpha} \\ 2\alpha & 0 & 1+\beta & 0 \\ 0 & 2\tilde{\alpha} & 0 & 1+\tilde{\beta} \end{bmatrix} \times \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1+\beta \\ 1+\tilde{\beta} \\ 2\alpha \\ 2\tilde{\alpha} \end{bmatrix}. \tag{6.31b}$$

Thus, in the frequency domain, the down-sampling operation adds the high frequency component $-\hat{r}_{\tilde{\mathbf{k}}}$ to the low frequency component $\hat{r}_{\mathbf{k}}$. This phenomenon is known as *aliasing* [28]. Similarly, the up-sampling operation sets the high-frequency component $-\hat{r}_{\tilde{\mathbf{k}}}$ equal to the low-frequency component $\hat{r}_{\mathbf{k}}$. This duplication effect is called *imaging* [28]. The lowpass filters which are cascaded with the down- and up-sampling operations have for function to reduce the aliasing and imaging effects. For example, when θ_x and θ_y are close to 0, $\alpha \approx 1$, $\beta \approx 1$, $\tilde{\alpha} \approx 0$, and $\tilde{\beta} \approx -1$. Hence, the aliasing and imaging effects occuring between $(\hat{r}_{\mathbf{k}},\hat{b}_{\mathbf{k}})^T$ and $(\hat{r}_{\tilde{\mathbf{k}}},\hat{b}_{\tilde{\mathbf{k}}})^T$ are substantially eliminated by the associated lowpass filters.

From (6.30) and (6.31), we can compute the spectrum $\hat{K}_h^{2h}(k_x,k_y)$ of the coarse-grid correction operator. The frequency domain matrix corresponding to the red-black Gauss-Seidel iteration is

$$\hat{G}_{rb}(k_x,k_y) = \begin{bmatrix} 0 & J \\ 0 & J^2 \end{bmatrix} . \qquad (6.32)$$

Note that $\hat{G}_{rb}(k_x,k_y)$ is a matrix of rank 2 rather than 4. Combining the spectra of the smoothing and coarse-grid correction operators, we obtain the spectrum of the two-grid operator

$$\hat{M}_h^{2h}(k_x,k_y) = \hat{G}_{rb}^{\nu_1}(k_x,k_y)\,\hat{K}_h^{2h}(k_x,k_y)\,\hat{G}_{rb}^{\nu_2}(k_x,k_y) , \qquad (6.33)$$

which is again a matrix of rank 2. In [84] this feature was exploited to find a closed-form expression for the spectral radius of the two-grid operator. If $\nu = = \nu_1 + \nu_2$, we get

$$\rho(M_h^{2h}) = \begin{cases} \dfrac{1}{4} & \nu = 1 \\[2ex] \dfrac{1}{2\nu}\left(\dfrac{\nu}{\nu+1}\right)^{\nu+1} & \nu \geq 2 \,. \end{cases} \qquad (6.34)$$

In (6.34), the maximum of $\rho[\hat{M}_h^{2h}(\theta)]$ occurs at $\theta = (\frac{\pi}{2},0)$ or $(0,\frac{\pi}{2})$ when $\nu = 1$ and at $(\cos^{-1}[(\frac{\nu}{\nu+1})^{1/2}]\,,\, \cos^{-1}[(\frac{\nu}{\nu+1})^{1/2}])$ when $\nu \geq 2$. Note since $M_h^{2h} \neq 0$, the two grid method is not a direct solver in the 2-D case. However, the spectral radius ρ is a constant independent of the grid size h, so that only $O(\log(h^{-1}))$ two-grid iterations are needed to solve (6.1) with an accuracy equal to the 5-point discretization error $O(h^2)$.

Multigrid methods: As in the 1-D case, we can recursively invoke the two-grid method to obtain multigrid algorithms. However, different recursion patterns may be needed for different 2-D or 3-D problems. Three commonly used recursion patterns, the V-cycle and W-cycle and full multigrid algorithms are shown in Fig. 6.4.

From this figure, we see that while the V-cycle multigrid algorithm applies the coarse-grid correction operator once per cycle, the W-cycle algorithm applies it twice. The numerical complexity per cycle of the V-cycle algorithm is therefore smaller than that of the W-cycle algorithm. On the other hand, since the W-cycle algorithm yields a better approximation of L_{2h}^{-1}, it requires fewer cycles to converge. The choice of cycling scheme depends on how the above tradeoff is affected by the problem that we seek to solve. For the model Poisson problem, the V-cycle algorithm works well. It requires just a few cycles (two or three) to converge within a fixed accuracy (independent of h), so that there is no need to use the W-cycle algorithm. However, the W-cycle algorithm is usually superior for difficult problems, such as highly anisotropic or nonlinear problems.

In the full multi-grid (FMG) scheme, instead of solving the discretized problem (6.1) on the fine grid only, we solve it on all grids, starting from the coarsest grid. Once (6.1) has been solved within the discretization accuracy of a given grid, we interpolate the solution to the next finer grid, and use this solution as initial estimate for the V- or W-cycle multigrid algorithm applied to the next problem. The advantage of this approach is that, because we are using a good initial estimate for each each successive problem, only a constant number of V- or W-cycle iterations are needed to solve (6.1) within the discretization error $O(h^p)$ of each grid. The total computational cost of the FMG algorithm is therefore very small, and equals the cost of a constant number of smoothing iterations on the finest grid [16],[48],[84].

D. Historical Notes

The idea of solving elliptic PDEs by using relaxation on multiple grids was first proposed by Fedorenko [36] and Bakhvalov [1] in the 1960s. However, it was not until the work of Brandt [16], Nicolaides [78] and Hackbush [48] in the 1970s that the efficiency of multigrid methods was recognized, and that their convergence properties were fully analyzed. Brandt used Fourier analysis to study the error-smoothing rate in the high frequency region. Subsequently, Stüben and Trottenberg [84] used also a Fourier approach to analyze a complete two-grid method including fine-grid smoothing, restriction, coarse-grid inversion and interpolation. Since all the elements of multigrid methods are already present in a two-grid cycling scheme, the results obtained for this scheme are usually a good indicator of the performance of more general multigrid algorithms. More recently, it was shown in [69] that the analysis of two-grid iterations can be simplified significantly by using two-color Fourier analysis. The book by Briggs [18] and article by Jespersen [58] provide a good introduction to multigrid methods for readers not acquainted with the subject. The proceedings of European multigrid conferences in 1981 [49] and 1985 [50] include several interesting theoretical and practical contributions, particularly concerning the application of multigrid methods to problems of fluid dynamics and aerodynamics. A book edited recently by McCormick [76] contains several articles on various aspects of multigrid theory, as well as an exhaustive multigrid bibliography until 1987. Finally, [48] gives a rigorous mathematical treatment of multigrid methods, and in particular of their convergence properties.

VII. Preconditioned Conjugate Gradient Methods

In the previous two sections, we have examined relaxation methods for solving elliptic PDEs on single and multiple grids. In this section, we consider solution techniques which combine the conjugate gradient algorithm with a preconditioning procedure, whose role is to reduce the condition number of the original system, thereby decreasing accordingly the number of iterations required by the conjugate gradient algorithm.

A. The Preconditioned Conjugate Gradient (PCG) Algorithm

When the conjugate gradient (CG) algorithm was introduced in the 1950s to solve SPD (symmetric positive definite) systems of the form (3.1), it was considered by some researchers as a direct method, since in the absence of roundoff errors, it yields an exact solution in at most N steps, where N is the order of the system. However, because of roundoff errors, this finite termination property does not hold in practice. Furthermore, since the SOR or CCSI methods require only $O(N^{1/2}\log N)$ iterations for the model Poisson problem, the conjugate gradient algorithm would in fact be relatively inefficient if it truly required N steps to solve this problem.

This forced researchers to view the CG method as an iterative method, and in this context it was found that a useful bound for the norm of the error $e^{(m)}$ after m iterations is [8],[9]

$$||e^{(m)}||_A \leq 2\,||e^{(0)}||_A \left[\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right]^m , \tag{7.1}$$

where $\kappa(A)$ denotes the condition number of the matrix A in (3.1), and $||x||_A = (x^T A x)^{1/2}$. For the 2D model Poisson problem, since for Dirichlet or periodic boundary conditions the eigenvalues are given respectively by (3.14) or (3.17), it is easy to check that $\kappa(A) = O(h^{-2}) = O(N)$. Substituting this value inside the bound (7.1), we can conclude that the CG procedure reduces the error by a constant factor in at most $O(N^{1/2})$ iterations, so that its rate of convergence is comparable to that of the SOR and CCSI methods. However, since the CG algorithm requires more operations per iteration than either the SOR or CCSI methods, these two methods are usually preferred.

Although the bound (7.1) is rather conservative, since it does not take into account the clustering of the eigenvalues of A, it provides an important clue for improving the CG method. Specifically, by introducing a SPD preconditioning transformation M, the system (3.1) can be transformed into

$$\tilde{A}\tilde{u}_d = \tilde{f}_d \ , \tag{7.2}$$

where $\tilde{A}$, $\tilde{u}_d$ and $\tilde{f}_d$ are related to A, u_d and f_d via

$$\tilde{A} = M^{-1/2}AM^{-1/2} \ \ , \ \ \tilde{u}_d = M^{1/2}u_d \ \ , \ \ \tilde{f}_d = M^{1/2}f_d \ , \tag{7.3}$$

and $M^{1/2}$ denotes the symmetric square-root of M. From the definition of $\tilde{A}$, we see that it is SPD. If the transformation M is easy to invert, and if the condition number $\kappa(\tilde{A})$ of the transformed system is much less than $\kappa(A)$, it becomes advantageous to apply the CG algorithm to the preconditioned system (7.2) instead of the original system (3.1). Note that since the matrices $\tilde{A}$ and $M^{-1}A$ are related by a similarity transform, we can examine the spectrum of $M^{-1}A$ instead of that of $\tilde{A}$ in order to find the convergence rate of the PCG method. In the following, M and $M^{-1}A$ will be called respectively the preconditioner and the preconditioned operator.

Each iteration of the PCG algorithm consists of the following two steps[45].

Step 1: Preconditioning. Solve

$$Mz_k = r_k \tag{7.4}$$

for z_k.

Step 2: CG iteration. Compute

$$\begin{aligned} \beta_{k+1} &= (z_k, r_k)/(z_{k-1}, r_{k-1}) \\ p_{k+1} &= z_k + \beta_{k+1}p_k \\ \alpha_{k+1} &= (z_k, r_k)/(p_{k+1}, Ap_{k+1}) \\ x_{k+1} &= x_k + \alpha_{k+1}p_{k+1} \\ r_{k+1} &= r_k - \alpha_{k+1}Ap_{k+1} \ . \end{aligned} \tag{7.5}$$

If the spectrum of $\tilde{A}$ has no special clustering feature, and if the condition *number* $\kappa(\tilde{A}) >> 1$, the bound (7.1) for the error norm indicates that the number of PCG iterations required to reduce the error by a constant factor is proportional to $O(\sqrt{\kappa(\tilde{A})})$. Thus, the goal of preconditioning is to find preconditioners M which are easy to invert, since each PCG iteration requires the solution of a system of the form (7.4), and such that the condition number of $\tilde{A}$ is as small as possible.

If both A and M have Fourier functions as eigenfunctions, the spectrum of $M^{-1}A$ can be analyzed directly in the frequency domain. In this context, the design of preconditioners corresponds to an inverse filtering problem. That is, given an FIR filter A, we seek to construct a filter $M^{-1} \approx A^{-1}$ such that M^{-1} can be implemented efficiently. Note that since A^{-1} is a noncausal IIR filter, this last constraint precludes selecting $M = A$.

Many elliptic preconditioners have been proposed in the literature. Depending on whether they rely on operations performed on a single discretization grid, or a sequence of discretization grids, they fall into the category of single-level, or of multi-level preconditioners. Examples of single-level preconditioners include the SSOR (symmetric successive over-relaxation) [6], ILU (incomplete lower and upper factorization) [77], MILU (modified ILU) [33] methods, as well as polynomial preconditioners [5],[59]. Examples of multilevel preconditioners include the multigrid method [61],[62] as well as the HB (hierarchical basis) [95] and MF (multilevel filtering) [67] preconditioners. The design of elliptic preconditioners is an active research area. We do not attempt to survey all existing preconditioning techniques. Instead, our goal is to relate the design and analysis of some preconditioners to familiar concepts in DSP to motivate further research along this line.

B. Preconditioners Based on Incomplete Factorization

Among single-level preconditioners, we focus on those obtained by incomplete factorization. Note that the Cholesky algorithm can be used to factor the coefficient matrix A into a product of lower and upper triangular matrices. However, although A is sparse, its lower and upper triangular factors are usually full, so that the Cholesky algorithm requires $O(N^3)$ operations. We are therefore led to consider preconditioners which require only an approximate factorization of A, i.e., $A \approx LU$, and with a computational complexity of $O(N)$. Efficient approximate factorization procedures of this type can be obtained by requiring that the lower and upper triangular factors L and U should have the same sparsity pattern as A. From the multidimensional signal processing viewpoint, constructing an incomplete factorization is equivalent to factoring of a noncausal IIR filter A^{-1} approximately into the product of two causal IIR filters U^{-1} and L^{-1} of fixed size.

The ILU and MILU factorizations, which were originally introduced in [77] and [33] respectively, rely on two different rules for constructing L and U. Both factorizations require that L and U should have the same zero entries as the lower and upper triangular parts of A, and that the nonzero off-diagonal entries of A should be equal to the corresponding entries of $M = LU$. The difference between both factorizations lies in the way the diagonal elements of M are specified (see Fig. 7.1). For the ILU factorization, the diagonal elements of A and M are required to be the same, whereas for the MILU factorization we require that, for all rows, the row sum of M must differ from the corresponding row sum of A by a small quantity ch^2, where c is a constant independent of h.

Each row of the matrix factors L and U specifies local finite-difference operators $L(E_x,E_y)$ and $U(E_x,E_y)$. Even if the PDE discretization operator $A(E_x,E_y)$ has constant coefficients the local operators $L(E_x,E_y)$ and $U(E_x,E_y)$ have usually space-dependent coefficients, due to boundary effects. However, for points far away from the domain boundary, these coefficients tend asymptotically to constant values. In the following, we ignore boundary effects and restrict our attention to the asymptotic behavior of incomplete factorization preconditioners.

ILU Preconditioner: For the model Poisson problem with the natural ordering, the local factorization operators $L(E_x,E_y)$ and $U(L_x,L_y)$ take the form [77]

$$L(E_x,E_y) = \frac{1}{4}(a - E_x^{-1} - E_y^{-1}) , \tag{7.6a}$$

$$U(E_x,E_y) = 1 - \frac{1}{a}E_x - \frac{1}{a}E_y , \tag{7.6b}$$

where a is a constant to be determined. Since the only nonzero coefficients of $L(E_x,E_y)$ (resp. $U(E_x,E_y)$) are those of 1, E_x^{-1} and E_y^{-1} (resp. 1, E_x and E_y), L and U have the same sparsity pattern as the lower and upper triangular parts of $A(E_x,E_y)$. The local ILU preconditioner $M_I(E_x,E_y)$ is the product of $L(E_x,E_y)$ and $U(E_x,E_y)$:

$$M_I(E_x,E_y) = \frac{1}{4}[a + \frac{2}{a} - (E_x + E_y + E_x^{-1} + E_y^{-1})$$
$$+ \frac{1}{a}(E_x E_y^{-1} + E_x^{-1} E_y)] \ . \qquad (7.7)$$

Comparing (3.7) and (7.7), we see that the coefficients of the off-diagonal terms E_x, E_x^{-1}, E_y and E_y^{-1} of operator $A(E_x,E_y)$ are matched by those of $M_I(E_x,E_y)$. Note that M_I contains some additional off-diagonal terms of the form $E_x E_y^{-1}$ and $E_x^{-1}E_y$. The ILU factorization imposes the additional requirement that the coefficients of the diagonal terms of M_I and A should be the same. This implies (see Fig. 7.1)

$$a + \frac{2}{a} = 4 \ , \qquad (7.8)$$

so that $a = 2 + \sqrt{2}$. This value of a is in fact observed asymptotically in the ILU factorization of the model Poisson problem with Dirichlet boundary conditions.

Therefore, the ILU-preconditioned Laplacian can be written in operator form as

$$(M_I^{-1}A)(E_x,E_y) = [1 - \frac{1}{4}(E_x + E_y + E_x^{-1} + E_y^{-1}) + \frac{1}{8+4\sqrt{2}}(E_x E_y^{-1} + E_x^{-1}E_y)]^{-1}$$
$$\times [1 - \frac{1}{4}(E_x + E_y + E_x^{-1} + E_y^{-1})] \ . \qquad (7.9)$$

It is straightforward to compute the spectrum of $M_I^{-1}A$ with respect to the Fourier basis functions $e^{i2\pi(k_x n_x + k_y n_y)h}$. We obtain

$$\hat{M}_I^{-1}(k_x,k_y)\hat{A}(k_x,k_y) =$$
$$\frac{1 - \frac{1}{2}[\cos(k_x 2\pi h) + \cos(k_y 2\pi h)]}{1 - \frac{1}{2}[\cos(k_x 2\pi h) + \cos(k_y 2\pi h)] + \frac{1}{4+2\sqrt{2}}\cos((k_x - k_y)2\pi h)} \ , \qquad (7.10)$$

where k_x and k_y are integers between 1 and $N-1$. This spectrum is plotted in Fig. 7.2. From this plot, as well as from a direct analysis, it is easy to check that the spectrum reaches its minimum at the four corners of the domain $1 \le k_x,k_y \le N-1$, and its maximum at the center, i.e., for $k_x = k_y \approx N/2$. Furthermore, the minimum and maximum are proportional to $O(h^2)$ and $O(1)$, respectively. This gives

$$\kappa_I(\tilde{A}) = \frac{\lambda_{\max}(\tilde{A})}{\lambda_{\min}(\tilde{A})} = \frac{\lambda_{\max}(M_I^{-1}A)}{\lambda_{\min}(M_I^{-1}A)} = O(h^{-2}) . \tag{7.11}$$

Since the condition number of $\tilde{A}$ is of the same order as that of A, it is tempting to conclude that the ILU factorization is not a good preconditioner for the CG algorithm. However, from Fig. 7.2, we see that except at the four corners of the (k_x, k_y) domain, the eigenvalues of $\tilde{A}$ are close to 1. A consequence of this eigenvalue clustering property is that the ILU preconditioner has a significant acceleration effect on the CG algorithm which is not reflected by the bound (7.1).

MILU Preconditioner: The MILU preconditioner has the same sparsity pattern as the ILU preconditioner, so that (7.6) and (7.7) also apply. Thus, for the model Poisson problem with the natural ordering, the MILU preconditioner can be represented as

$$M_M(E_x, E_y) = \frac{1}{4}[a + \frac{2}{a} - (E_x + E_y + E_x^{-1} + E_y^{-1}) + \frac{1}{a}(E_x E_y^{-1} + E_x^{-1} E_y)] . \tag{7.12}$$

The difference between the ILU and MILU factorizations lies in how the constant a is determined. For the MILU factorization [33], it is required that the row sum of $M_M(E_x, E_y)$ should differ from the row sum of $A(E_x, E_y)$, which is zero, by a small quantity δ. This gives

$$\frac{1}{4}(a + \frac{4}{a} - 4) = \delta , \tag{7.13}$$

and selecting $\delta = 4^{-1}ch^2$ with $c > 0$, we obtain

$$a = 2 + \frac{ch^2}{2} + \frac{1}{2}\sqrt{8ch^2 + c^2h^4} . \tag{7.14}$$

As was observed above, the spectrum of the ILU preconditioner M_I approximates poorly the spectrum of A at the four corners of the domain $1 \le k_x, k_y \le N-1$. In the modified ILU scheme, the condition (7.13) is imposed in order to guarantee that the preconditioner M_M approximates A well in this region. By performing a Fourier analysis identical to the one employed for the ILU case, the spectrum and condition number of the MILU-preconditioned Laplacian can be evaluated. A surface plot of the spectrum is shown in Fig. 7.3. This plot indicates that the smallest eigenvalues are of order 1, and the largest eigenvalues occur near the end points of the transverse diagonal $k_x + k_y = N$. These eigenvalues are of order h^{-1}, and consequently

$$\kappa_M(\tilde{A}\,) = \frac{\lambda_{\max}(M_M^{-1}A\,)}{\lambda_{\min}(M_M^{-1}A\,)} = O(h^{-1})\,. \tag{7.15}$$

Comparing (7.11) and (7.15), we see that the condition number of the MILU preconditioned system is one order of magnitude smaller than that of the ILU preconditioned system. Numerical experiments have confirmed that the MILU-CG and ILU-CG require respectively $O(h^{-1})$ and $O(h^{-1/2})$ iterations to converge [22].

The ordering of grid points plays in general an important role in determining the form of the coefficient matrix A, and hence of the preconditioners. With the red-black ordering, the ILU and MILU preconditioners take completely different forms and the spectra of preconditioned operators behave very differently. See [66] for more details.

C. Multilevel Preconditioners Based on Filtering

The focus of research on elliptic preconditioners has shifted recently to the design of preconditioners with a multilevel (or hierarchical) grid structure. Since the global features of elliptic operators can be reproduced more easily by multilevel preconditioners, the resulting preconditioned systems have often very small condition numbers, ranging from $O(1)$ to $O(\log^{\alpha} h^{-1})$ where α is a small integer, and hence the corresponding PCG algorithms converge very fast. Another advantage of multilevel preconditioners is that they can be effectively implemented on massively parallel computers [67] and, therefore, are attractive for parallel computation.

Several multilevel preconditioners have been proposed. One such preconditioner is the MG algorithm of Section VI. When combined with the CG method, it yields the MG-CG algorithm. The motivation for using the MG algorithm as a preconditioner is that its speed of convergence is governed by the smoothness of the solution function, whereas the convergence rate of the CG method is not affected by this feature. Consequently, the MG-CG method is more effective than the MG method alone for certain applications, such as the solution of interface problems, where because of presence of several materials, the elliptic PDE has discontinuous coefficients. Two other types of multilevel preconditioners have been proposed by Yserentant [95],[96] and Bramble, Pasciak and Xu [15],[90] in the context of finite-element methods. Yserentant considered a new set of basis functions, known as the hierarchical basis. Bramble et al. introduced a sequence of basis functions which are defined at various discretization levels and called multilevel nodal basis functions. Roughly speaking, the preconditioning step $M^{-1}r$ consists in projecting the residual r onto these basis functions. In the following, we examine yet another preconditioner, the multilevel filtering (MF) preconditioner, which was proposed recently in [67]. This preconditioner relies explicitly on multirate digital signal processing techniques and can be best described in the Fourier domain.

The filtering approach to the design of preconditioners can be described as follows. Suppose that we approximate the spectrum of an elliptic operator by a piecewise constant function. In the space domain, this approximating function corresponds to an operator which (i) splits the input function into several components, where each such component consists of wavenumbers within a narrow band, (ii) scales each component by a constant, and (iii) recombines all the scaled components. The inverse of such an operator is easy to implement, since it has the same form, except that the scaling constants are inverted. In multirate digital signal processing, the decomposition of a signal into components consisting of different wavenumber bands, and vice versa, is accomplished by a filter bank analyzer (resp. synthesizer). Although there exists a number of techniques for designing filter banks (see [28], Chapter 7), the filter bank which is used for the MF preconditioning technique is obtained by cascading a sequence of lowpass filters operating on different discretization grids, in combination with down- and up-sampling operations.

To be more precise, consider the 1D Poisson equation on [0,1] with zero boundary conditions. After discretization on a uniform grid Ω_h with spacing $h = 2^{-L}$, where L is a positive integer, we obtain

$$(-\frac{1}{2}E + 1 - \frac{1}{2}E^{-1})u_n = f_n \ , \qquad 1 \le n \le N{-}1 \ , \tag{7.16}$$

with $N = 2^L$. This system can be rewritten as

$$Au = f \ , \tag{7.17}$$

where A is a tridiagonal matrix with diagonal elements $-1/2$, 1 and $-1/2$. A can be diagonalized as

$$A = W^T \Lambda_A W , \tag{7.18}$$

where

$$\Lambda_A = \mathrm{diag}\,(\lambda_1 , \cdots , \lambda_k , \cdots , \lambda_{N-1}) , \quad \lambda_k = 1 - \cos(k\pi h) , \tag{7.19a}$$

and W is a square matrix of size $N-1$, whose kth row is

$$w_k^T = (\frac{2}{N})^{1/2}(\sin(k\pi h) , \cdots , \sin(k\pi n h) , \cdots , \sin(k\pi(N-1)h)) . \tag{7.19b}$$

The diagonalization of the matrix A can be interpreted as a decomposition of the driving and solution functions into their Fourier components. Futhermore, λ_k is just the spectrum $\hat{A}(k)$ of the 1D Laplacian.

In the wavenumber domain, the spectrum $\hat{A}(k)$ can be approximated by a piecewise constant function

$$\hat{P}(k) = c_l , \quad k \in B_l , \quad 1 \le l \le L , \tag{7.20a}$$

where

$$B_l = \{k \in \mathbf{N} : 2^{l-1} \le k < 2^l\} \tag{7.20b}$$

denotes the lth wavenumber band. Let Λ_P be the diagonal matrix with $\hat{P}(k)$ as kth diagonal element and $P = W^T \Lambda_P W$. Then, the P-preconditioned Laplacian takes the form

$$P^{-1}A = W^T \Lambda_{P^{-1}A} W , \tag{7.21a}$$

with

$$\begin{aligned} \Lambda_{P^{-1}A} &= (\Lambda_P)^{-1}\Lambda_A \\ &= \mathrm{diag}\,(\frac{\lambda_1}{c_1} , \frac{\lambda_2}{c_2} , \frac{\lambda_3}{c_2} , \cdots , \frac{\lambda_{2^{l-1}}}{c_l} , \cdots , \frac{\lambda_{2^l-1}}{c_l} , \cdots , \frac{\lambda_{N-1}}{c_L}) . \end{aligned} \tag{7.21b}$$

The question is how to choose the constants c_l in order to reduce the condition number of $P^{-1}A$. If we select

$$c_l = 4^{-(L-l)} , \tag{7.22}$$

it can be shown [67] that the eigenvalues of $P^{-1}A$ satisfy

$$1 \le \lambda(P^{-1}A) < \frac{\pi^2}{2} \approx 4.93 , \tag{7.23}$$

so that the condition number $\kappa(P^{-1}A)$ is bounded by 4.93, a constant independent of the grid size h. In Figure 7.4, we plot the spectra $\hat{A}(k)$, $\hat{P}^{-1}(k)$ and $\hat{P}^{-1}(k)A(k)$ for $N = h^{-1} = 256$, when c_l is given by (7.22).

For P to be an effective preconditioner, $P^{-1}r$ has to be easily computable for any given vector r. It is clear that $P^{-1} = W^T \Lambda_P^{-1} W$ is a piecewise constant function in the wavenumber domain. The preconditioning procedure

$$P^{-1}r = W^T \Lambda_P^{-1} Wr \ , \tag{7.24}$$

consists therefore of three steps: (i) filter bank analysis, (ii) scaling, and (iii) filter bank synthesis, which are represented here by multiplications by W, Λ_P^{-1} and W^T, respectively. To clarify this comment, we can rewrite (7.24) as

$$P^{-1}r = (\sum_{l=1}^{L} \frac{1}{c_l} W_l^T W_l)r \ , \tag{7.25}$$

where W_l, $1 \le l \le L$, are $(N-1)^2$ square matrices which have the same 2^{l-1} to 2^l-1 rows as W and zero vectors for remaining rows. Then, we have

$$W_l^T W_l w_k = \begin{cases} w_k \ , & k \in B_l \\ 0 \ , & \text{otherwise} \ , \end{cases} \tag{7.26}$$

where w_k is defined in (7.19b). From (7.26), we see that W_l functions as an ideal bandpass filter for the band B_l. Although it is possible to implement the ideal bandpass characteristic (7.26) with FFTs or bandpass filters of size N, the resulting implementations either cannot be extended to more general PDEs, or are too expensive (i.e., of complexity $O(N^2)$. This leads us to approximate the ideal bandpass filter W_l with a nonideal filter F_l with

$$F_l^T F_l w_k \approx \begin{cases} w_k \ , & k \in B_l \\ 0 \ , & \text{otherwise} \ , \end{cases} \tag{7.27}$$

and such that F_l can be implemented cost effectively for general problems. The resulting preconditioner is

$$Q^{-1}r = (\sum_{l=1}^{L} \frac{1}{c_l} F_l^T F_l)r \ . \tag{7.28}$$

The block diagram of Fig. 7.5 describes a procedure for constructing the bandpass filters F_l, with $1 \le l \le L$, in terms of a cascade of elementary low-pass filters H_L, H_{L-1}, $\cdots$, H_2. From Fig.7.5, we see that F_l can be expressed in terms of the filters H_l as

$$F_L = I - H_L \ , \tag{7.29a}$$

$$F_l = (I - H_l)\left[\prod_{p=l+1}^{L} H_p\right], \quad 2 \le l \le L-1 \ , \tag{7.29b}$$

$$F_1 = \prod_{p=2}^{L} H_p \ . \tag{7.29c}$$

Let the elementary filter H_L be an FIR filter of the form

$$H_L = a_0 + \sum_{j=1}^{J} a_j (E^j + E^{-j}) \ , \tag{7.30}$$

where the coefficients a_j are selected so that the spectrum $\hat{H}_L(k)$ approximates an ideal lowpass filter, i.e.

$$\hat{H}_L(k) \approx \begin{cases} 1 \ , & 0 \le k < 2^{L-1} \\ 0 \ , & 2^{L-1} \le k \le 2^L \ . \end{cases} \tag{7.31}$$

The coefficients a_j can be determined by using any standard digital low-pass filter design technique. One specific choice is examined in [67]. The same coefficients are also used for constructing the lth-level elementary filter

$$H_l = a_0 + \sum_{j=1}^{J} a_j (E^{2^{L-l} j} + E^{-2^{L-l} j}) \ , \tag{7.32}$$

with $2 \le l \le L$. Comparing (7.30) and (7.32), we see that the only difference between elementary filters H_L and H_l is that while H_L constructs a weighted average of points separated by a distance of h, the lth-level filter H_l performs the same average over points separated by a distance of $2^{L-l} h$.

Since some of the points needed to perform the above averages may be located outside the domain Ω_h, the system (7.16) is viewed as defined on an infinite grid with an odd-periodic extended driving function, i.e.,

$$f_{-n} = -f_n \quad \text{and} \quad f_{n+2pN} = f_n \tag{7.33}$$

for p integer. The filtering operations that we have just described are performed at every grid point, for all levels $2 \le l \le L$. If the order J of filters H_l is finite, the number of operations required for such an implementation is proportional to $O(N\log N)$, where N is the total number of unknowns. However, since waveforms consisting only of low wavenumber components can be represented accurately on coarser grids, we can incorporate the multigrid structure into the above framework. This is illustrated in Fig. 7.6. The preconditioners shown in Figs. 7.5 and 7.6 are called the SGMF and MGMF preconditioners, respectively. Note that the MGMF preconditioner is obtained by inserting 2:1 down-samplers (I_l^{l-1}) and 1:2 up-samplers (I_{l-1}^l) into the SGMF preconditioner. It is easy to see that the number of operations required by the MGMF preconditioner is proportional to $O(N)$ instead of $O(N\log N)$ for the SGMF case.

The MGMF preconditioner of Fig. 7.6 can be simplified further by deleting paths corresponding to $I - H_l$. The resulting modified MGMF preconditioner is shown in Fig. 7.7. It can be expressed as

$$R^{-1}r = (\sum_{l=1}^{L} \frac{1}{d_l} G_l^T G_l)r \ , \tag{7.34}$$

with

$$G_L = I \ , \tag{7.35a}$$

$$G_l = \prod_{p=l+1}^{L} I_p^{p-1} H_p \ , \quad 2 \le l \le L-1 \ , \tag{7.35b}$$

$$G_1 = H_2 \prod_{p=3}^{L} I_p^{p-1} H_p \ , \tag{7.35c}$$

and where the scaling constants d_l are related to the constants c_l via

$$\sum_{i=l}^{L} \frac{1}{d_i} = \frac{1}{c_l} \ . \tag{7.36}$$

Note that unlike the preconditioner Q, which relied on bandpass filters F_l, the modified preconditioner R is implemented in terms of lowpass filters G_l. A consequence of this feature is that the wavenumber components of the residual r belonging to the band B_l are present at the first $L-l+1$ levels. Since according to Fig. 7.7, these components are multiplied by $d_L^{-1}, \ \cdots \ , d_l^{-1}$ respectively, the preconditioners R and Q will be equivalent only if the constants c_l and d_l satisfy the relation (7.36).

The generalization of the MF preconditioner to multidimensional problems on regular domains is straightforward. For example, the two-dimensional elementary filter H_l can be obtained as the tensor product of one-dimensional elementary filters along the x- and y-directions. It has been shown by Fourier analysis that the condition number of the MF-preconditioned Laplacian implemented with nonideal filters is proportional to $O(1)$ for the 1-D, 2-D and 3-D cases. This implies that the MF-CG method converges in a finite number of iterations independently of h, which has been confirmed by numerical experiments [67].

D. Historical Notes

The conjugate gradient method for solving linear systems of equations was developed in late 1940s and early 1950s by Hestenes, Stiefel and others. For a history of the conjugate gradient algorithm and the closely related Lanczos algorithm, the readers are referred to a recent survey by Golub and O'Leary, which contains an annotated bibliography for the period 1948-1976. A detailed presentation of the SSOR, ILU and MILU preconditioners can be found in the book by Axelsson and Barker [8]. The Fourier analysis of the ILU, MILU and SSOR preconditioners for the naturally ordered Poisson problem with periodic boundary conditions was performed by Chan and Elman [22]. They also observed strong similarities in the eigenvalue distribution of incomplete factorization preconditioners for the Dirichlet and periodic problems. Kuo and Chan [66] used two-color Fourier analysis to study the eigenvalue distribution of the ILU, MILU and SSOR preconditioned Laplacian with the red-black ordering. In the last few years, a growing amount of work has focused on the design of multilevel preconditioners. A brief survey of recent advances in this area can be found in the paper by Kuo, Chan and Tong [67].

VIII. Domain Decomposition Methods

Domain decomposition methods rely on a partition of the domain of definition Ω of a given PDE into subdomains Ω_i with or without overlapping regions. The original problem is then decomposed into smaller problems defined over each subdomain, which can be solved independently, provided that a strategy is developed for evaluating the variables corresponding to overlapping regions, or to interfaces between subdomains. Domain decomposition techniques present several advantages. First, it is often possible to select the subdomains Ω_i in such a way that special solvers, such as fast direct solvers or MG methods, can be applied to the subproblems, even though they are not applicable to the problem defined over the entire domain Ω. This is the case for example when Ω is irregular, but can be represented as the union of regular subdomains Ω_i, or when the PDE has constant parameters over each subdomain, but not over the entire domain, such as for interface problems between different materials. Domain decomposition methods are also attractive from the point of view of parallel computation, since all subproblems can be solved in parallel.

Domain decomposition algorithms can be divided into two categories, depending on whether the subdomains overlap or not. Algorithms with overlapping subdomains fall into the category of Schwartz alternating methods [83], whereas those with nonoverlapping subdomains are called iterative substructuring or capacitance matrix methods. We restrict our attention here to capacitance matrix methods, where the domain is decomposed into regular sudomains, and the capacitance system governing the variables on the interfaces between subdomains is solved by an iterative method, such as the PCG algorithm. Since each iteration requires the solution of problems over each subdomain, it is important to find good preconditioners for the capacitance system. To do so, we use Fourier analysis to study the capacitance system corresponding to a simple model problem consisting of Poisson's equation defined over a rectangle divided horizontally into two subrectangles. This analysis leads to FFT based preconditioners, which are then shown to be effective for more complex domain geometries.

A. Capacitance Matrix Formulation

Consider a discretized elliptic PDE with Dirichlet boundary conditions,

$$Au = f \ , \tag{8.1}$$

whose domain Ω is partitioned into two nonoverlapping subdomains Ω_1 and Ω_2 with an interface region Γ_3, as shown in Fig. 8.1. By partitioning the solution u and driving function f into subvectors u_i and f_i, with $i = 1, 2, 3$, corresponding to the unknowns and driving terms indexed by points of Ω_1, Ω_2 and Γ_3, respectively, (8.1) can be expressed in block form as

$$\begin{bmatrix} A_{11} & & A_{13} \\ & A_{22} & A_{23} \\ A_{13}^T & A_{23}^T & A_{33} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} . \tag{8.2}$$

Using block Gaussian elimination, the system (8.2) can be solved as follows:

Step 1: Determine u_3 by solving the capacitance system

$$Cu_3 = g_3 \tag{8.3}$$

where the capacitance matrix

$$C = A_{33} - A_{13}^T A_{11}^{-1} A_{13} - A_{23}^T A_{22}^{-1} A_{23} \ , \tag{8.4}$$

is the Schur complement of diag $(A_{11} \, , A_{22})$ inside A, and

$$g_3 = f_3 - A_{13}^T A_{11}^{-1} f_1 - A_{23}^T A_{22}^{-1} f_2 \ . \tag{8.5}$$

Step 2: Compute u_1 and u_2 from

$$u_1 = A_{11}^{-1} g_1 \ , \quad u_2 = A_{22}^{-1} g_2 \ , \tag{8.6a}$$

with

$$g_1 = f_1 - A_{13} u_3 \ , \quad \text{and} \quad g_2 = f_2 - A_{23} u_3 \ . \tag{8.6b}$$

In (8.5) and (8.6), we need to invert the matrices A_{11} and A_{22} which describe the coupling among variables of subdomains Ω_1 and Ω_2, respectively. The operation $A_{11}^{-1} w$ (or $A_{22}^{-1} w$), where w is an appropriate vector, is called a subproblem solve. It can often be implemented by using fast direct or MG methods. The solution of the capacitance system (8.3) is more difficult. It is usually not desirable to form the capacitance matrix C explicitly, since the direct computation of the elements of C is very expensive. Instead, when (8.3) is solved by iterative methods such as the PCG algorithm, only the computation of Cw is required, which involves two subproblem solves. Due to the high cost of computing Cw, it is important that iterative methods should converge very fast. Consequently, the design of good preconditioners for the capacitance matrix C is the key to the development of efficient nonoverlapping domain decomposition algorithms.

B. Fourier Analysis of the Capacitance System

As a first step, we consider the case where the matrix A in (8.1) represents the 5-point discretized Laplacian with local operator (3.7), defined over a rectangular domain Ω. We also assume that Ω is decomposed horizontally into two rectangular strips Ω_1 and Ω_2, as shown in Fig. 8.2. In the x-direction, Ω is discretized uniformly with mesh size $h = N^{-1}$, where $N-1$ is the number of internal discretization points. In the y-direction, we assume that the widths L_1 and L_2 of Ω_1 and Ω_2 satisfy

$$L_1 = M_1 h \quad \text{and} \quad L_2 = M_2 h \ , \tag{8.7}$$

where M_1 and M_2 are positive integers.

A consequence of this simple decomposition geometry is that Fourier analysis can be employed to study the capacitance system (8.3). Specifically, we show below that the matrices A_{33}, $A_{13}^T A_{11}^{-1} A_{13}$ and $A_{23}^T A_{22}^{-1} A_{23}$ appearing in the definition (8.4) of C have all for eigenvectors the sine vectors

$$w_k^T = \sqrt{2h}\,(\sin(k\pi h)\,,\ \cdots\ ,\ \sin(k\pi nh)\,,\ \cdots\ ,\ \sin(k\pi(N-1)h))\ , \tag{8.8}$$

with $1 \leq k \leq N-1$.

First, the local operator corresponding to A_{33} can be expressed as

$$1 - \frac{1}{4}(E_x + E_x^{-1})\ .$$

Consequently, by operating with A_{33} on w_k, we obtain

$$A_{33} w_k = [\,1 - \frac{1}{2}\cos(k\pi h)\,]\, w_k = \frac{1}{4}(2 + \sigma_k) w_k\ , \tag{8.9}$$

with

$$\sigma_k = 4\sin^2(\frac{k\pi h}{2})\ . \tag{8.10}$$

Thus, w_k is an eigenvector of A_{33}.

Next, we examine $-A_{13}^TA_{11}^{-1}A_{13}$. Let $A_{11}^{-1}A_{13}w_k \equiv v_k$, so that

$$A_{11}v_k = A_{13}w_k \ . \tag{8.11}$$

The equation (8.11) can be viewed as obtained by discretizing Laplace's equation (the driving function is zero) on Ω_1 with zero boundary values along the east, north and west boundaries and w_k along the south boundary. It turns out that its solution v_k admits the closed-form expression

$$v_k(n_x,n_y) = \sqrt{2h}\,\sin(kn_x\pi h)d_{k,1}(n_y)\ , \tag{8.12}$$

where $d_{k,1}(n_y)$ satisfies the difference equation

$$d_{k,1}(n_y-1) - (2+\sigma_k)d_{k,1}(n_y) + d_{k,1}(n_y+1) = 0\ , \quad 1 \le n_y \le M_1 - 1\ , \tag{8.13}$$

with boundary conditions $d_{k,1}(0) = 1$ and $d_{k,1}(M_1) = 0$. We are interested here in the quantity

$$-A_{13}^TA_{11}^{-1}A_{13}w_k = -A_{13}^Tv_k = \frac{1}{4}v_k(n_x,n_y{=}1) = \frac{1}{4}d_{k,1}(1)w_k\ . \tag{8.14}$$

Thus, w_k is an eigenfunction of $-A_{13}^TA_{11}^{-1}A_{13}$ with eigenvalue $d_{k,1}(1)/4$. The same procedure can be used to analyze the matrix $-A_{23}^TA_{22}^{-1}A_{23}$. This gives

$$-A_{23}^TA_{22}^{-1}A_{23}w_k = \frac{1}{4}d_{k,2}(1)w_k\ , \tag{8.15}$$

where $d_{k,2}(1)$ is obtained by solving

$$d_{k,2}(n_y-1) - (2+\sigma_k)d_{k,2}(n_y) + d_{k,2}(n_y+1) = 0\ , \quad 1 \le n_y \le M_2 - 1\ , \tag{8.16}$$

with boundary conditions $d_{k,2}(0) = 1$ and $d_{k,2}(M_2) = 0$. Combining (8.9), (8.14) and (8.15) yields

$$Cw_k = \frac{1}{4}(\,2+\sigma_k + d_{k,1}(1) + d_{k,2}(1)\,)w_k \equiv \lambda_k w_k\ , \tag{8.17}$$

for $1 \le k \le N{-}1$, so that w_k is an eigenvector of C, as claimed. Further analysis shows [21] that the eigenvalue λ_k associated to w_k can be expressed as

$$\lambda_k = g(k,M_1,M_2)\sqrt{\sigma_k + \tfrac{1}{4}\sigma_k^2}\ , \tag{8.18}$$

where σ_k is given by (8.10), and

$$g(k,M_1,M_2) = \frac{1}{4}\left(\frac{1+\gamma_k^{M_1}}{1-\gamma_k^{M_1}} + \frac{1+\gamma_k^{M_2}}{1-\gamma_k^{M_2}}\right), \tag{8.19a}$$

with

$$\gamma_k = (1+½\sigma_k - \sqrt{\sigma_k + ¼\sigma_k^2})^2 \,. \tag{8.19b}$$

Note that σ_k is the spectrum of the 1-D Laplacian operator $L = 2 - (E_x + E_x^{-1})$ defined on Γ_3. The respective spectra λ_k and σ_k of the capacitance matrix C and Laplacian L, and the function $g(k,M_1,M_2)$, are plotted in Fig. 8.3 for $M_1 = M_2 = 40$ and $h^{-1} = 256$.

The geometric parameters M_1 and M_2 which specify the sizes of subdomains Ω_1 and Ω_2 affect only the function $g(k,M_1,M_2)$. From Fig. 8.3, we see that this function has values of $O(1)$. For large M_1 and M_2 with fixed k, $g(k,M_1,M_2)$ reaches its asymptotic value 0.5 rapidly. Therefore, (8.18) can be simplified as

$$\lambda_k \approx 0.5\sqrt{\sigma_k + ¼\sigma_k^2} \,. \tag{8.20}$$

Since $\sigma_k^2 << \sigma_k$ for small k and $¼\sigma_k^2 \approx \sigma_k$ for large k, an even rougher estimate for λ_k is

$$\lambda_k \approx 0.5\sqrt{\sigma_k} \,. \tag{8.21}$$

In summary, we have shown in this section that if W is the orthonormal matrix of size $N-1$ whose columns are the sine vectors w_k, the capacitance matrix C associated to the partition of a rectangular domain into two horizontal strips admits the eigenvalue/eigenvector decomposition

$$C = W\Lambda W^T \quad \text{with} \quad \Lambda = \text{diag}\{\lambda_1, \cdots, \lambda_k, \cdots, \lambda_{N-1}\} \,. \tag{8.22}$$

C. Preconditioners for the Capacitance Matrix

From (8.22) and (8.18), it is easy to check that, for h sufficiently small, the condition number of the capacitance matrix C is given by

$$\kappa(C) = \frac{\max \lambda_k}{\min \lambda_k} \approx \frac{2\sqrt{2}}{S\pi h} = O(h^{-1}) \,, \tag{8.23}$$

with

$$S = \frac{1}{2}\left(\frac{1+e^{-2\pi L_1}}{1-e^{-2\pi L_1}} + \frac{1+e^{-2\pi L_2}}{1-e^{-2\pi L_2}} \right).$$

It is therefore of interest to design preconditioners M such that $\kappa(M^{-1}C) = O(1)$.

Several such preconditioners have been proposed in the literature. These preconditioners are all of the form

$$M = WDW^T \ , \tag{8.24}$$

and differ only by the choice of diagonal matrix D. Dryja [30], and Golub and Mayers [44] proposed preconditioners with

$$D_D = 0.5 \ \mathrm{diag}\{\sigma_k\} \quad \text{and} \quad D_G = 0.5 \ \mathrm{diag}\{\sqrt{\sigma_k + \tfrac{1}{4}\sigma_k^2}\} \ , \tag{8.25a}$$

respectively. These preconditioners can be motivated by the eigenvalue decomposition (8.22) for C, and approximations (8.21) and (8.20), respectively, for the eigenvalues λ_k of C. More recently, Chan [21] proposed the selection of

$$D_C = \Lambda \ , \tag{8.25b}$$

where Λ is given by (8.22). The preconditioner M given by (8.24), (8.25b) is *exact* for Poisson's equation and the domain decomposition geometry of Fig. 8.2. Finally, observe that all preconditioners of the form (8.24) admit FFT implementations.

An interesting feature of the above preconditioners is that, although they were designed for the case where Ω is a rectangle divided horizontally into two subrectangles, they are applicable to complex domain geometries where Ω is the union of an arbitrary number of rectangles. Consider for example the Poisson equation defined on the L- or C-shaped regions of Figs. 8.4(a) and 8.4(b). For the L-shaped domain of Fig. 8.4(a), Ω can be viewed as obtained by assembling the three elementary rectangles Ω_i with $i = 1, 2, 3$. The corresponding interfaces are Γ_4 and Γ_5. Consider now a decomposition of Ω into two rectangles Ω_1 and $\Omega_{23} = \Omega_2 \cup \Omega_3$. The corresponding capacitance system defined over interface Γ_4 is

$$C_4 u_4 = g_4 \ . \tag{8.26}$$

To precondition this system, we can ignore the presence of Ω_3, and let M_4 be preconditioner given by (8.24), (8.25b) when we partition $\Omega_{12} = \Omega_1 \cup \Omega_2$ into Ω_1 and Ω_2 with interface Γ_4. It was shown by Chan and Resasco [25] that with this choice, the condition number $\kappa(M_4^{-1} C_4)$ is of $O(1)$. A similar result holds for the C-shaped domain geometry of Fig. 8.4(b) [25]. This indicates that preconditioners designed for rectangular domains remain effective for more complex domain geometries. More generally, for an arbitrary problem such as the one depicted in

Fig. 8.1, one may fit the domain with two subrectangles in such a way the geometric parameters M_1 and M_2 can be estimated, and then used to design a preconditioner of the form (8.24)-(8.25).

D. Historical Notes

The first domain decomposition technique for solving elliptic problems was introduced by Schwartz in 1869, who proposed an alternating procedure, where the problem is solved by going in alternance from one subdomain to another. A short history of the early work on domain decomposition methods can be found in [89]. The recent interest in domain decomposition techniques is due to the fact that these methods are intrinsically parallel, and are therefore well adapted to parallel computers. A recent paper by Keyes and Gropp [63] provides a good introduction to domain decomposition methods for readers unfamiliar with this topic. It gives an overview of various domain decomposition techniques, compares their performance, and discusses their parallel implementation. The Fourier analysis of the capacitance matrix for a rectangular domain divided into two subrectangles was first proposed by Chan [21]. The extension of this analysis to the case of a rectangle divided into an arbitrary number of rectangular strips is described in [24]. In [25], Chan and Resasco presented a general framework for the analysis and construction of domain decomposition preconditioners over irregular regions. For a more general perspective on domain decomposition methods, and on their application to a wide variety of PDEs, readers may wish to consult the proceedings of two conferences on domain decomposition methods held in 1987 [43], and 1988 [23].

IX. Parallel Computation

A great deal of progress has been accomplished during the last 20 years in developing vector and parallel computer architectures [55],[56] and algorithms for solving elliptic PDEs. In this section, we focus on algorithms for parallel computers and will give a brief account of the main achievements in this area. For a more thorough review, we refer readers to the work of Ortega and Voigt [80],[81].

As indicated in Section III, one way to parallelize PDE algorithms is to reorder the sequence of grid points to be processed in such a way that a large number of operations can be performed in parallel. For example, the red-black ordering is more attractive than the natural ordering for solving 5-point discretized elliptic PDEs, as far as parallel implementation is concerned. One interesting question that arises in this context is whether the convergence rate of iterative algorithms is affected by the reordering scheme. This problem has been studied in [2],[35],[66],[71],[73]. In particular, the effect of the red-black ordering on SOR and PCG algorithms is discussed in detail in [66]. Briefly speaking, the convergence rate of the SOR algorithm is independent of ordering schemes, but the convergence rate of PCG algorithms depends on the choice of ordering. For the CG method preconditioned by the MILU or SSOR method, the convergence rate of the red-black ordering is one order of magnitude slower than that of the natural ordering [35],[66]. For PCG methods, there exists therefore a tradeoff between the rate of convergence and the degree of parallelism that can be achieved.

No such tradeoff exists for the SOR method, but another difficulty arises when one seeks to implement it in parallel. Specifically, when the coefficients of the PDE are space-dependent, the optimal relaxation parameter depends in general on global information and must be estimated adaptively [51]. The estimation of the relaxation parameter requires global communication between all processors, a feature that slows down the SOR algorithm significantly. To overcome this difficulty, a local relaxation procedure was proposed in [14],[34],[71] where different relaxation parameters are used at every grid point, and are determined on the basis of local information. Since, unlike the conventional SOR algorithm, no global information is needed for determining the optimal local relaxation parameters, the communication time between multiple processors is significantly reduced. Another extension of the red/black SOR algorithm involves the use of more than two colors for ordering the grid points. The motivation for considering multiple coloring schemes is that when elliptic PDEs are discretized on high-order stencils, more than two colors are necessary to decouple all grid points of the same color. For the case of a 9-point stencil discretization, four colors are needed. The extension of the red/black SOR algorithm to multiple coloring schemes can take different forms. For the 9-point discretized Poisson problem, two such extensions have been proposed by Adams, Leveque and Young [3], and by Kuo and Levy [70], which rely respectively on a single- or two-level relaxation scheme. Both of these methods are easily parallelizable on mesh-connected processor arrays.

In parallel implementations of the PCG algorithm, the major bottleneck is usually the parallelization of the preconditioner (7.4), since the remaining steps of the PCG algorithm can be parallelized in a straightforward way. The main difficulty lies in the fact that elliptic PDE problems involve a global coupling of all the grid points. In order to be effective, preconditioners must take into account this global coupling by including a mechanism for transmitting information from one point of the problem domain to another. Consequently, preconditioners that use purely local information, such as the red-black ordered MILU and SSOR and polynomial preconditioners, are fundamentally limited in their ability to improve the convergence rate of the CG algorithm. On the other hand, global coupling through a natural ordering grid traversal is not highly parallelizable. To construct highly parallelizable and effective preconditioners, we are therefore led to consider preconditioners which share global information through a multilevel grid structure, thus ensuring a good convergence rate, but perform only local operations on each grid level, and hence are highly parallelizable. Preconditioners that have this feature include the multigrid method when used as a preconditioner [61],[62], and the hierarchical basis preconditioner [95],[96]. More recently, new multilevel preconditioners have been proposed by Bramble, Pasciak and Xu [15],[90] and Kuo, Chan and Tong [67]. These preconditioners differ from multigrid methods by the fact that the smoothing operation in multigrid methods is replaced by a simple scaling operation, as was shown in Section VII.B. Other types of multilevel preconditioners have been examined in [7],[10],[11],[72],[88]. A detailed comparison of several multilevel elliptic preconditioners can be found in [67].

The parallelization of multigrid methods or multilevel preconditioners on multiprocessor machines is one of the most challenging areas in parallel computing for elliptic PDEs. A significant amount of work has focused on parallelizing standard multigrid algorithms on mesh-connected arrays [17],[40] and hypercubes [26]. Variants of standard multigrid algorithms aiming at achieving more parallelism on massively parallel computers have also been proposed. These parallel multigrid algorithms include the concurrent multigrid method [40] and the superconvergent multigrid method [39]. A thorough survey of the state-of-the-art in this field is presented in [27]. Roughly speaking, two fundamental isssues arise in parallelizing multigrid methods. One is to find an appropriate mapping which assigns adjacent grid points to neighboring processors so that only local communication is required. Since the hierarchy of grids in the multigrid algorithm complicates the flow of data, this is in general not easy. However, for the hypercube machine this mapping problem has been solved by Chan and Saad [26]. The second problem is usually known as that of load balancing. To get maximal parallelism, we need as many processors as there are points at the fine grid level. However, when relaxation is performed on the coarse grid, the majority of the processors become idle. Thus, the problem is to reduce the number of idle processors as much as possible so that the efficiency of the entire multiprocessor system is maximized. One promising way to solve this problem is to perform concurrent iterations at different grid levels. For example, we may use filtering to split the problem into multiple subproblems defined on different grids, where each subproblem corresponds to a different spectral component of the original problem. These subproblems could then be solved simultaneously by performing concurrent relaxations on all grids. However this approach raises many questions: what is the optimal splitting scheme? What is the best filter for dividing a given problem into subproblems? How is the convergence and efficiency of standard multigrid algorithms affected by this decomposition procedure?

Domain decomposition provides a natural way to achieve parallel computation. This approach is particularly suitable for a coarse grain parallel computing environment where there are considerably fewer processors than grid points. One important issue in domain decomposition is the selection of the number of subdomains. On one hand, more subdomains imply more parallelism. On the other hand, the communication cost per iteration and the overall number of iterations tend to increase with the number of subdomains. Thus, the answer is generally architecture- and problem-dependent. The complexity of parallel implementations of domain decomposition techniques on a ring, a two-dimensional mesh, and an n-cube has been studied by Keyes and Gropp [64]. Some performance analysis results and numerical experiments have also been reported in [19],[47],[52],[57].

X. Conclusion and Extension

Digital signal processing (DSP) and the numerical solution of PDEs have been traditionally considered as separate research areas. However, during the last 30 years Fourier analysis has been used increasingly by numerical analysts to analyze and design numerical PDE algorithms. Without surprise, results obtained by Fourier analysis can be reformulated within the DSP framework. Recent research work [65],[67],[68],[70],[71] has focused on bridging the gap between these two separate research areas, and a number of interesting new results have been obtained as a consequence of this effort. In this paper, we have described in detail the link existing between DSP and the numerical solution of PDEs, so that numerical PDE algorithms can be understood by electrical engineers in a more familiar setting. In addition, a number of recent developments on iterative solution techniques for elliptic PDEs have been reviewed so as to provide readers with the most up-to-date knowledge in this area.

The effort to bridge the gap between DSP and numerical differential equations will benefit to researchers in both areas. From the electrical engineering side, researchers will be able to study existing numerical algorithms for differential equations more easily. They will also find numerous interesting and challenging problems in the solution of differential equations, for example, the solution of PDEs consisting of both space and time variables. From the numerical analysis side, researchers will have a new set of tools to analyze and design numerical algorithms. Further advances based on this connection can be expected in the future.

It is worthwhile to emphasize that the DSP approach relies on tools that are usually not used in the matrix context: the theory of multidimensional signals and systems [31] and frequency-domain analysis. To form a matrix equation, a 1-D ordering is required and, therefore, the proximity of grid points in multidimensional meshes is disguised. This phenomenon does not occur for multidimensional DSP techniques, since they are fully adapted to the spatial nature of the signals being studied. The discretized system of equations for the elliptic problem is loosely coupled in the space domain, but totally decoupled in the frequency domain. In other words, transforming the system from the space domain to the frequency domain corresponds to a diagonalization procedure whereby a sparse matrix is transformed into a diagonal matrix, thus leading to a much simpler analysis. Due to its simplicity, the DSP approach provides some valuable insight into the choice of solution method, as well as some guidelines towards the development of more versatile and efficient solution techniques. This point has been demonstrated in the application of digital filtering theory to the design of elliptic preconditioner as discussed in Section VII. Thus, we conclude that the DSP approach can serve as complement to the classical matrix analysis, which is more generally applicable but less transparent.

In this tutorial paper, we have examined discretization schemes and solution methods for solving elliptic PDEs from the DSP viewpoint. We studied mode-dependent finite-difference schemes for three model elliptic PDE problems, i.e., the Poisson, Helmholtz and convection-diffusion equations. The extension of mode-dependent discretization schemes to coupled differential equations and time-dependent problems, such as hyperbolic and parabolic PDEs, is currently being investigated. We also reviewed various methods for solving self-adjoint positive definite elliptic PDEs modeled by the Poisson equation, including direct methods, elementary and accelerated relaxation methods, multigrid methods, preconditioned conjugate gradient methods and the domain decomposition technique. We expect that the DSP viewpoint will also be helpful to develop new efficient algorithms for solving more difficult elliptic PDEs such as indefinite and nonself-adjoint problems modeled by the Helmholtz and convection-diffusion equations.

References

1. N. S. Bakhvalov , "On the convergence of a relaxation method with natural constraints on the elliptic operator," *U.S.S.R. Comp. Math. and Math. Phys.*, vol. 6 , no. 5, pp. 101-135, 1966.

2. L. M. Adams and H. F. Jordan, "Is SOR color-blind," *SIAM J. Sci. Stat.*, vol. 7, no. 2, pp. 490-506, 1986.

3. L. M. Adams, R. J. LeVeque, and D. M. Young, "Analysis of the SOR iteration for the 9-point Laplacian," *SIAM J. Numer. Anal.*, vol. 25, pp. 1156-1180, 1988.

4. D. N. De G. Allen and R. V. Southwell, "Relaxation methods applied to determine the motion, in two dimensions, of a viscous fluid past a fixed cylinder," *Q. J. Mech. and Appl. Math.*, vol. 8, pp. 129-145, 1955.

5. S. F. Ashby, "Polynomial preconditioning for conjugate gradient methods," Ph.D. Thesis, 1987, Department of Computer Science, University of Illinois, Urbana, IL 61801..

6. O. Axelsson, "A generalized SSOR method," *BIT*, vol. 13, pp. 443-467, 1972.

7. O. Axelsson, "An algebraic framework for multilevel methods," Report 8820, Department of Mathematics, Catholic University, The Netherlands, 1988.

8. O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems,* Academic Press, Inc., 1984.

9. O. Axelsson and G. Lindskog, "On the rate of convergence of the preconditioned conjugate gradient methods," *Numer. Math.*, vol. 48, pp. 499-523, 1986.

10. O. Axelsson and P. Vassilevski, "Algebraic multilevel preconditioning methods, I," Report 8811, Department of Mathematics, Catholic University, The Netherlands, 1988.

11. O. Axelsson and P. Vassilevski, "Algebraic multilevel preconditioning methods, II," Report 1988-15, Institute for Scientific Computation, University of Wyoming, Laramie, Wyoming, 1988.

12. G. Birkhoff and R. E. Lynch, *Numerical Solution of Elliptic Problems,* SIAM, Philadelphia, PA, 1984.

13. G. Birkhoff and A. Schoenstadt, ed., *Elliptic Problem Solvers II,* Academic Press, Inc., New York, N.Y., 1984.

14. E. F. Botta and A. E. P. Veldman, "On local relaxation methods and their application to convection-diffusion equations," *J. Comput. Phys.*, vol. 48, pp. 127-149, 1981.

15. J. H. Bramble, J. E. Pasciak, and J. Xu, "Parallel multilevel preconditioners," To appear in *Math. Comp.*.

16. A. Brandt, "Multi-level adaptive solutions to boundary-value problems," *Math. Comp.*, vol. 31, no. 138, pp. 333-390, 1977.

17. A. Brandt, "Multigrid solvers on parallel computers," in *Elliptic Problem Solvers*, ed. M. H. Schultz, pp. 39-83, Academic Press, Inc., New York, N.Y., 1981.

18. W. L. Briggs, *A Multigrid Tutorial,* SIAM, Philadelphia, PA, 1987.

19. L. Brochard, "Efficiency of multicolor domain decomposition on distributed memory systems," in *Domain Decomposition Methods*, ed. T. F. Chan et al., pp. 249-259, SIAM, Philadelphia, PA., 1989.

20. O. Buneman, "A compact non-iterative Poisson solver," Rep. 294, Stanford University Institute for Plasma Research, Stanford, CA, 1969.

21. T. F. Chan, "Analysis of preconditioners for domain decomposition," *SIAM J. Numer. Anal.*, vol. 24, no. 2, pp. 382-390, 1987.

22. T. F. Chan and H. C. Elman, "Fourier analysis of iterative methods for elliptic problems," *SIAM Review*, vol. 31, pp. 20-49, 1989.

23. T. F. Chan, R. Glowinski, J. Periaux, and O. B. Widlund ed., *Domain Decomposition Methods,* SIAM, Philadelphia, PA., 1989.

24. T. F. Chan and D. C. Resasco, "A domain-decomposed fast Poisson solver on a rectangle," *SIAM J. Sci. Stat. Comput.*, vol. 8, no. 1, pp. 14-26, 1987.

25. T. F. Chan and D. C. Resasco, "A framework for the analysis and construction of domain decomposition preconditioners," in *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, ed. R. Glowinski et al., pp. 217-230, SIAM, Philadelphia, PA., 1988.

26. T. F. Chan and Y. Saad, "Multigrid algorithms on hypercube multiprocessor," *IEEE Trans. on Computers*, vol. C-35, no. 11, pp. 969-977, 1986.

27. T. F. Chan and R. S. Tuminaro, "Design and implementation of parallel multigrid algorithms," in *Multigrid Methods: Theory, Applications, and Supercomputing*, Marcel Dekker Inc., 1988.

28. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983.

29. G. Dahlquist and A. Björck, *Numerical Methods,* Prentice-Hall, Inc. , Englewood Cliffs, N.J. , 1974.

30. M. Dryja, "A capacitance matrix method for Dirichlet problem on polygon region," *Numer. Math.*, vol. 39, pp. 51-64, 1982.

31. D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing,* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984.

32. I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices,* Oxford University Press, Oxford OX2 6DP, 1986.

33. T. Dupont, R. P. Kendall, and H. H. Rachford, Jr., "An approximate factorization procedure for solving self-adjoint difference equations," *SIAM J. Numer. Anal.*, vol. 5, no. 3, pp. 559-573, 1968.

34. L. W. Ehrlich, "An ad hoc SOR method," *J. Comput. Phys.*, vol. 44, pp. 31-45, 1981.

35. H. Elman and E. Agron, "Ordering techniques for the preconditioned conjugate gradient method on parallel computers," *Computer Physics Communications*, vol. 53, pp. 253-269, 1989.

36. R. P. Fedorenko, "The speed of convergence of an iterative process," *U.S.S.R. Comp. Math. and Math. Phys.*, vol. 4 , no. 3 , pp. 227-235 , 1964.

37. D. Flanders and G. Shortley, "Numerical determination of fundamental modes," *J. Appl. Phys.*, vol. 21, pp. 1326-1332, 1950.

38. S. P. Frankel, "Convergence rates of iterative treatments of partial differential equations," *Math. Tables Aids Comput.*, vol. 4, pp. 65-75, 1950.

39. P. O. Frederickson and O. A. McBryan, "Parallel superconvergent multigrid," presented in 3rd Multigrid Conference, Copper Mountain, April, 1987.

40. D. Gannon and J. Van Rosendale, "On the structure of parallelism in a highly concurrent PDE solver," *Journal of Parallel and Distributed*

Computing, vol. 3, no. 1, pp. 106-135, 1986.

41. W. Gautschi, "Numerical integration of ordinary differential equations based on trigonometric polynomials," *Numer. Math.*, vol. 3, pp. 381-397, 1961.

42. A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive-Definite Systems,* Prentice-Hall, New Jersey, 1981.

43. R. Glowinski, G. Golub , G. A. Meurant, and J. Periaux ed., *First International Symposium on Domain Decomposition Methods for Partial Differential Equations,* SIAM, Philadelphia, PA., 1988.

44. G. H. Golub and D. Mayers, "The use of preconditioning over irregular regions," Lecture at Sixth Int. Conf. on Computing Methods in Applied Sciences and Engineering, Versailles, Dec. 1983..

45. G. H. Golub and C. F. Van Loan, *Matrix Computations,* The John Hopkins University Press, Baltimore, MD 21211, 1989.

46. G. H. Golub and R. S. Varga, "Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods," *Numer. Math.*, vol. 3, pp. 147-168, 1961.

47. W. D. Gropp and D. E. Keyes, "Domain decomposition on parallel computers," in *Domain Decomposition Methods*, ed. T. F. Chan et al., pp. 260-268, SIAM, Philadelphia, PA., 1989.

48. W. Hackbusch, *Multi-Grid Methods and Applications,* Springer-Verlag, Berlin, Germany, 1985.

49. W. Hackbusch and U. Trottenberg, ed., *Multigrid Methods,* Springer-Verlag, New York, N.Y., 1982.

50. W. Hackbusch and U. Trottenberg, ed., *Multigrid Methods II,* Springer-Verlag, New York, N.Y., 1986.

51. L. A. Hageman and D. M. Young, *Applied Iterative Methods,* Academic Press, Inc. , New York, N.Y., 1981.

52. M. Haghoo and W. Proskurowski, "Parallel efficiency of a domain decomposition method," in *Domain Decomposition Methods*, ed. T. F. Chan et al., pp. 269-281, SIAM, Philadelphia, PA., 1989.

53. R. W. Hockney, "A fast direct solution of Poisson's equation using Fourier analysis," *J. Assoc. Comp. Mach.*, vol. 12, pp. 95-113, 1965.

54. R. W. Hockney, "The potential calculation and some applications," in *Methods of Computational Physics*, ed. B. Adler, S. Fernbach and M. Rotenberg, pp. 136-211, Academic Press, New York and London, 1969.

55. K. Hwang, "Advanced parallel processing with supercomputer architectures," *Proc. IEEE*, vol. 75, no. 10, pp. 1348-1379, 1987.

56. K. Hwang and F. A. Briggs, *Computer Architecture and Parallel Processing,* McGraw-Hill, Inc., New York, N.Y., 1984.

57. K. Hwang and H. C. Wang, "A multigrid Schwarz alternating method for parallel solution of elliptic PDE problems," submitted to Journal of Parallel and Distributed Computing..

58. D. Jesperson, "Multigrid methods for partial differential equations," in *Studies in Numerical Analysis*, ed. G. H. Golub, The Mathematical Association of America, 1984.

59. O. G. Johnson, C. A. Micchelli, and G. Paul, "Polynomial preconditioning for conjugate gradient calculations," *SIAM J. Numer. Anal.*, vol. 20, no. 2, pp. 362-376, 1983.

60. L. V. Kantorovich and V. I. Krylov, *Approximate Methods of Higher Analysis,* Interscinece Publishers, Inc., New York, 1964.

61. R. Kettler, "Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods," in *Multigrid Methods*, ed. W. Hackbusch and U. Trottenberg, pp. 502-534, Springer-Verlag, New York, N.Y., 1982.

62. R. Kettler and J. A. Meijerink, "A multigrid method and a combined multigrid-conjugate gradient method for elliptic problems with strongly discontinuous coefficients in general domain.," Shell publication 604, KSEPL, Rijswijk, The Netherlands.

63. D. E. Keyes and W. D. Gropp, "A comparison of domain decomposition techniques for elliptic partial differential equations and their parallel implementation," *SIAM J. Sci. Stat. Comput.*, vol. 8, no. 2, pp. s166-s202, 1987.

64. D. E. Keyes and W. D. Gropp, "Complexity of parallel implementations of domain decomposition techniques for elliptic partial differential equations," *SIAM J. Sci. Stat. Comput.*, vol. 9, no. 2, pp. 312-326, 1988.

65. C.-C. J. Kuo, "Discretization and solution of elliptic PDEs: a transform domain approach," Ph.D. Thesis, Report LIDS-TH-1687, Laboratory for Information and Decision Systems, MIT, Cambridge, MA., 1987.

66. C.-C. J. Kuo and T. C. Chan, "Two-color Fourier analysis of iterative algorithms for elliptic problems with red/black ordering," to appear in *SIAM J. Sci. Stat. Comput.*.

67. C.-C. J. Kuo, T. F. Chan, and C. Tong, "Multilevel filtering elliptic preconditioners," to appear in *SIAM J. Matrix Analysis and Applications.*.

68. C.-C. J. Kuo and B. C. Levy, "Mode-dependent finite-difference discretization of linear homogeneous differential equations," *SIAM J. Sci. Stat. Comput.*, vol. 9, pp. 992-1015, 1988.

69. C.-C. J. Kuo and B. C. Levy, "Two-color Fourier analysis of the multigrid method with red/black Gauss-Seidel smoothing," *Applied Mathematics and Computation*, vol. 29, pp. 69-87, 1989.

70. C.-C. J. Kuo and B. C. Levy, "A two-level four-color SOR method," *SIAM J. Numer. Analy.*, vol. 26, pp. 129-151, 1989.

71. C.-C. J. Kuo, B. C. Levy, and B. R. Musicus, "A local relaxation method for solving elliptic PDEs on mesh-connected arrays," *SIAM J. Sci. Stat. Comput.*, vol. 8, no. 4, pp. 550-573, Jul. 1987.

72. Y. A. Kuznetsov, "Multigrid domain decomposition methods for elliptic problems," in *Proceedings VIII International Conference on Computational Methods for Applied Science and Eng. Vol. 2*, pp. 605-616, 1987.

73. R. J. LeVeque and L. N. Trefethen, "Fourier analysis of the SOR iteration," *IMA J. Numer. Anal.*, vol. 8, pp. 273-279, 1988.

74. J. S. Lim, *Two-dimensional Signal and Image Processing,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1990.

75. R. Manohar and J. W. Stephenson, "Optimal finite analytic methods," *J. Heat Transfer*, vol. 104, pp. 432-437, 1982.

76. S. F. McCormick, ed., *Multigrid Methods,* SIAM, Philadelphia, PA, 1987.

77. J. A. Meijerink and H. A. van der Vorst, "An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-Matrix," *Math. Comp.*, vol. 31, no. 137, pp. 148-162, 1977.

78. R. A. Nicolaides, "On the l^2 convergence of an algorithm for solving finite element equations," *Math. Comp.*, vol. 31, no. 140, pp. 892-906, 1977.

79. A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1975.

80. J. M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems,* Plenum Press, New York, 1988.

81. J. M. Ortega and R. G. Voigt, "Solution of partial differential equations on vector and parallel computers," *SIAM Review,* vol. 27, no. 2, pp. 149-240, 1985.

82. Martin H. Schultz, *Elliptic Problem Solvers,* Academic Press, Inc., New York, N.Y., 1981.

83. H. A. Schwarz, *Gesammelte Mathematische Abhandlungen,* pp. 133-134, Springer, Berlin, 1980.

84. K. Stüben and U. Trottenberg, "Multigrid methods : fundamental algorithms, model problem analysis, and applications," in *Multigrid Methods,* ed. W. Hackbusch and U. Trottenberg, pp. 1-176, Springer-Verlag, New York, N.Y., 1982.

85. P. N. Swarztrauber, "The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle," *SIAM Review,* vol. 19, no. 3, pp. 490-501, 1977.

86. P. N. Swarztrauber, "Fast Poisson Solvers," in *Studies in Numerical Analysis,* ed. G. H. Golub, pp. 319-370, 1984.

87. R. S. Varga, *Matrix Iterative Analysis,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1962.

88. P. Vassilevski, "Iterative methods for solving finite element equations based on multilevel splitting of the matrix," Preprint, Bulgarian Academy of Science, Sofia, Bulgaria, 1987.

89. O. Widlund, "Domain decomposition algorithms and the bicentennial of the French revolution," *SIAM News,* vol. 22, no. 4, 1989.

90. J. Xu, "Theory of multilevel methods," Ph.D. Thesis, Department of Mathematics, Cornell University, N.Y. 14853, 1989.

91. D. M. Young, ''Iterative methods for solving partial differential equations of elliptic type ,'' Doctoral Thesis , Harvard University , 1950.

92. D. M. Young, ''Iterative methods for solving partial differential equations of elliptic type,'' *Trans. Amer. Math. Soc.*, vol. 76, pp. 92-111, 1954.

93. D. M. Young, *Iterative Solution of Large Linear Systems,* Academic Press, Inc., New York, N.Y., 1971.

94. D. M. Young, ''A historical overview of iterative methods,'' *Computer Physics Communications*, vol. 35, pp. 1-17, 1989.

95. H. Yserentant, ''On the multi-level splitting of finite element spaces,'' *Numer. Math.*, vol. 49, pp. 379-412, 1986.

96. H. Yserentant, ''Hierarchical bases give conjugate gradient type methods a multigrid speed of convergence,'' *Appl. Math. and Comp.*, vol. 19, pp. 347-358, 1986.

Figure Captions

Fig. 2.1 Coincident frequencies of the mode-dependent (a) 5-point, (b) rotated 5-point, and (c) 9-point stencil discretization of the Helmholtz equation.

Fig. 2.2 Coincident frequencies of the (a) central difference, (b) Allen-Southwell, and (c) uniformly distributed mode-dependent 5-point discretizations of the convection-diffusion equation.

Fig. 3.1 (a) Conventional and (b) folded two-color Fourier domains where $\theta_x = k_x \pi h$ and $\theta_y = k_y \pi h$.

Fig. 5.1 The spectrum magnitude of the Jacobi iteration operator.

Fig. 5.2 Root loci of λ_1 and λ_2 with fixed μ.

Fig. 5.3 A typical eigenvalue map in the complex plane for (a) Jacobi iteration and (b) SOR iteration with the optimal relaxation parameter, where the case $h = \frac{1}{16}$ and $\omega = 1.757$ is plotted.

Fig. 5.4 A typical plot of the eigenvalues of the Chebyshev semi-iterative method as function of the eigenvalues of the Jacobi method, where the case $h = \frac{1}{16}$, $\mu_{\max} = -\mu_{\min} = 0.98$ and $Q_{10}(x)$ is shown.

Fig. 6.1 Structure of an $(h, 2h)$ two-grid method.

Fig. 6.2 The spectrum of the 1-D damped Jacobi smoother parameterized with ω.

Fig. 6.3 Decomposition of the (a) restriction and (b) interpolation operators.

Fig. 6.4 Illustrations of (a) V-cycle, (b) W-cycle, and (c) full multigrid methods.

Fig. 7.1 Stencil representation of local operators for the (a) ILU and (b) MILU preconditioners.

Fig. 7.2 Typical surface plot of the spectrum of the ILU preconditioned Laplacian where $\theta_x = 2\pi k_x h$, $\theta_y = 2\pi k_y h$ and $h = 0.02$.

Fig. 7.3 Typical surface plot of the spectrum of the MILU preconditioned Laplacian where $\theta_x = 2\pi k_x h$, $\theta_y = 2\pi k_y h$, $h = 0.02$ and $c = 70$.

Fig. 7.4 Spectra of A, P^{-1} and $P^{-1}A$.

Fig. 7.5 Block diagram of the MF preconditioned with a single discretization grid (SGMF).

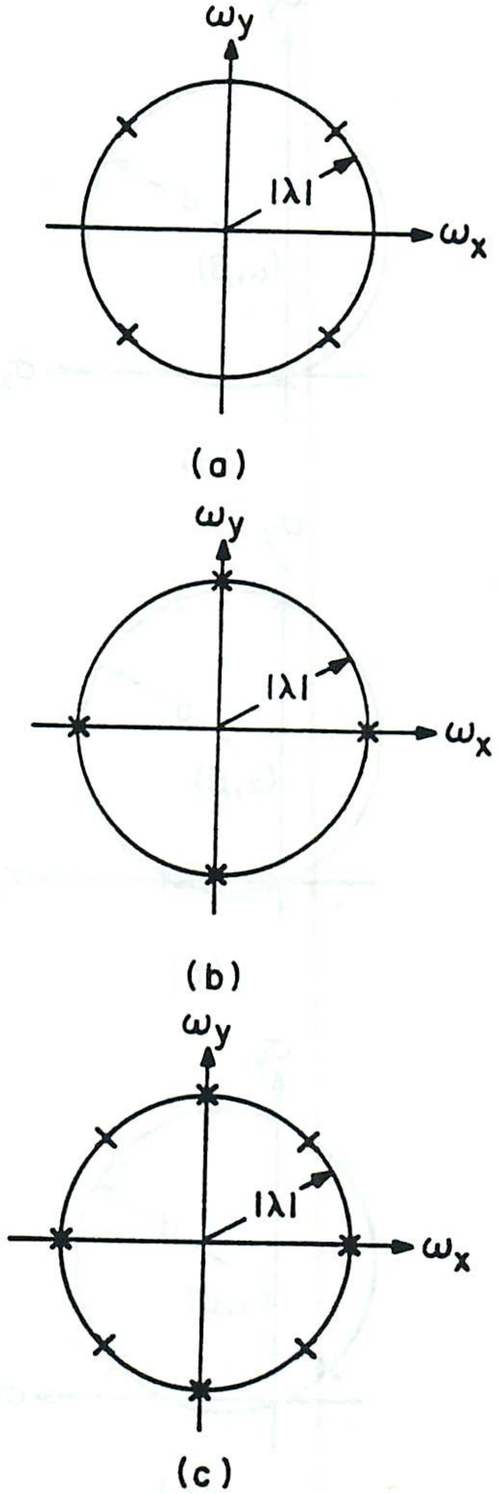

Fig. 2.1 Coincident frequencies of the mode-dependent (a) 5-point, (b) rotated 5-point, and (c) 9-point stencil discretization of the Helmholtz equation.

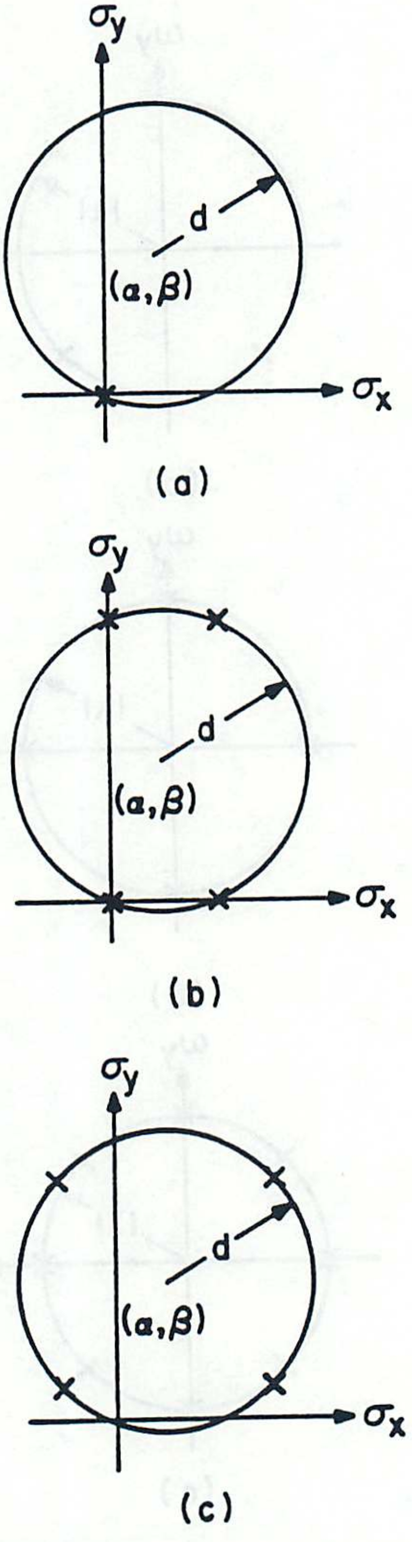

Fig. 2.2 Coincident frequencies of the (a) central difference, (b) Allen-Southwell, and (c) uniformly distributed mode-dependent 5-point discretizations of the convection-diffusion equation.

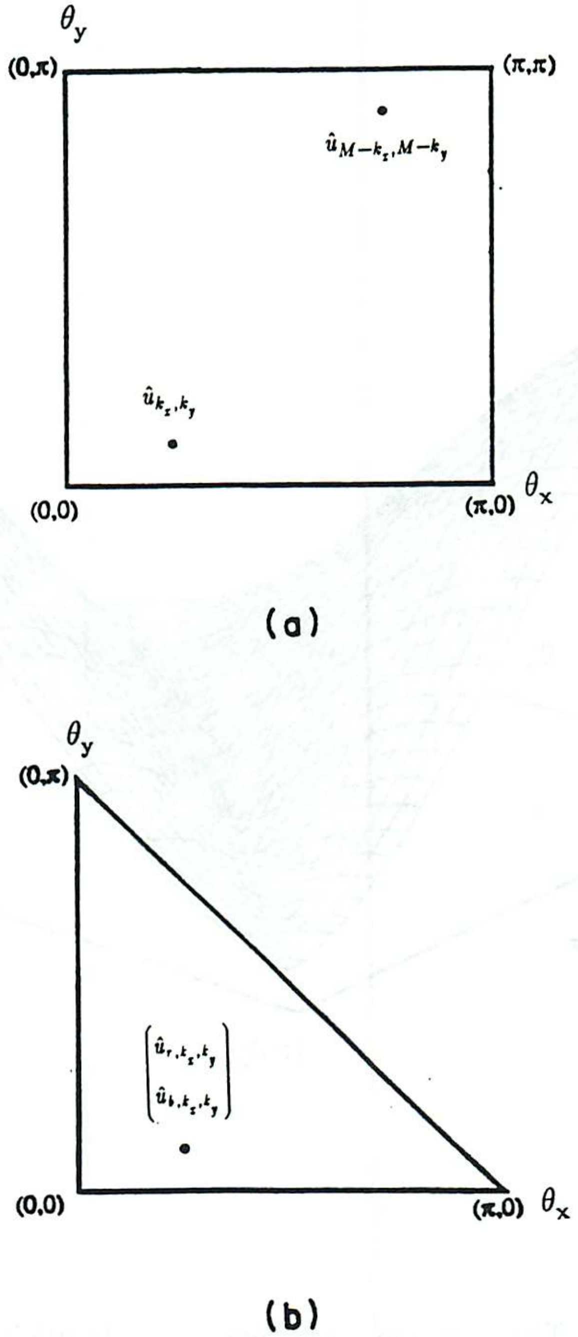

Fig. 3.1 (a) Conventional and (b) folded two-color Fourier domains where $\theta_x = k_x \pi h$ and $\theta_y = k_y \pi h$.

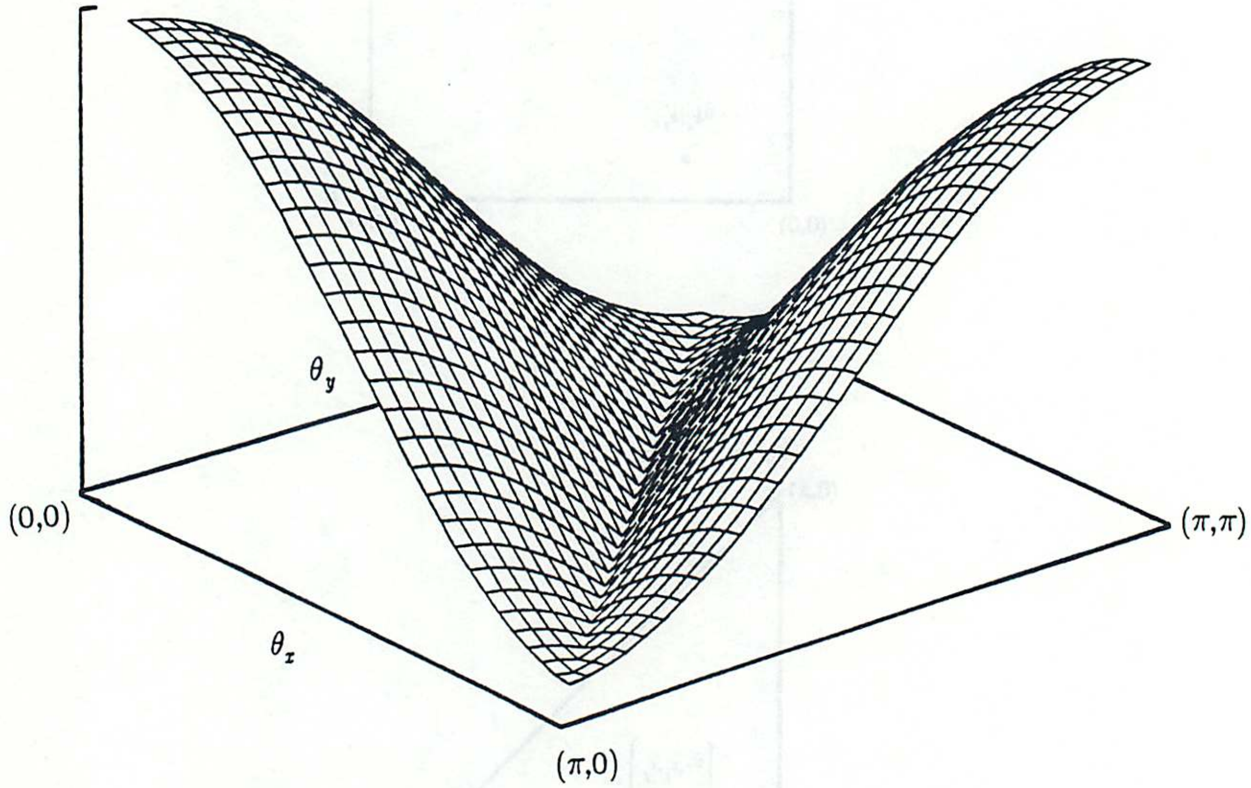

Fig. 5.1 The spectrum magnitude of the Jacobi iteration operator.

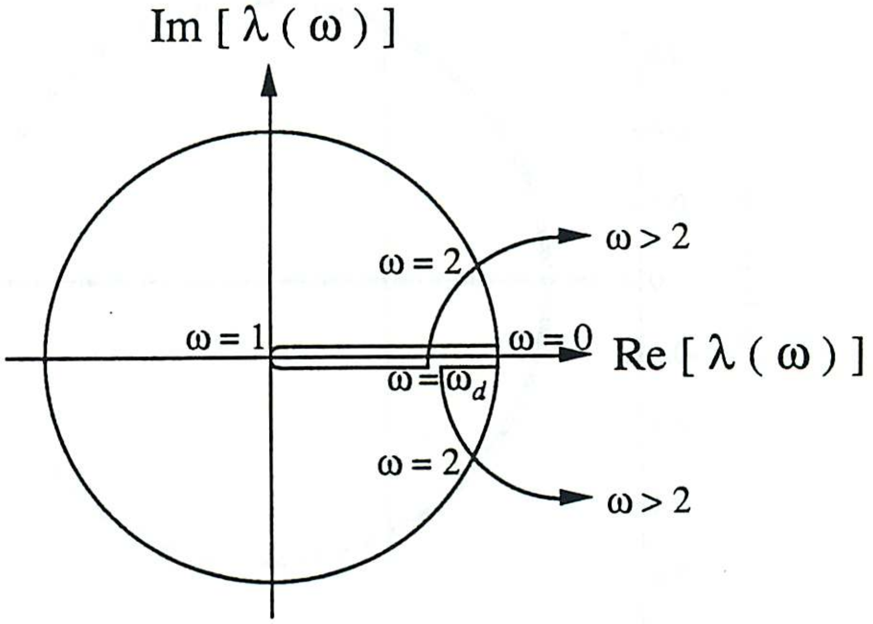

Fig. 5.2 Root loci of λ_1 and λ_2 with fixed μ.

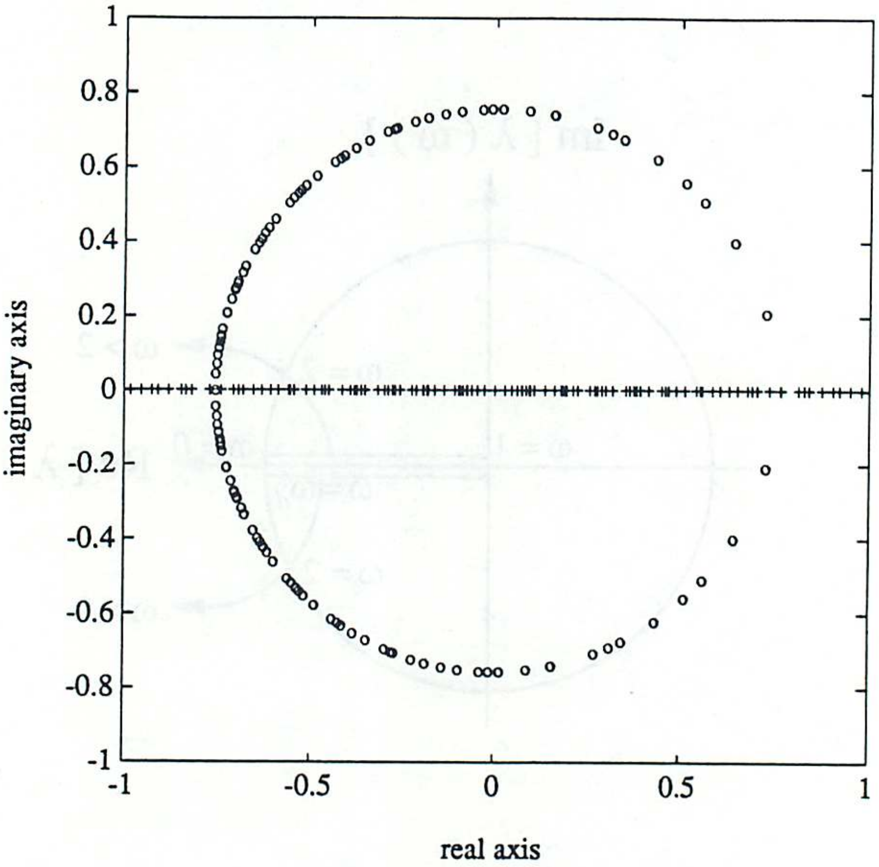

Fig. 5.3 A typical eigenvalue map in the complex plane for (a) Jacobi iteration and (b) SOR iteration with the optimal relaxation parameter, where the case $h = \frac{1}{16}$ and $\omega = 1.757$ is plotted.

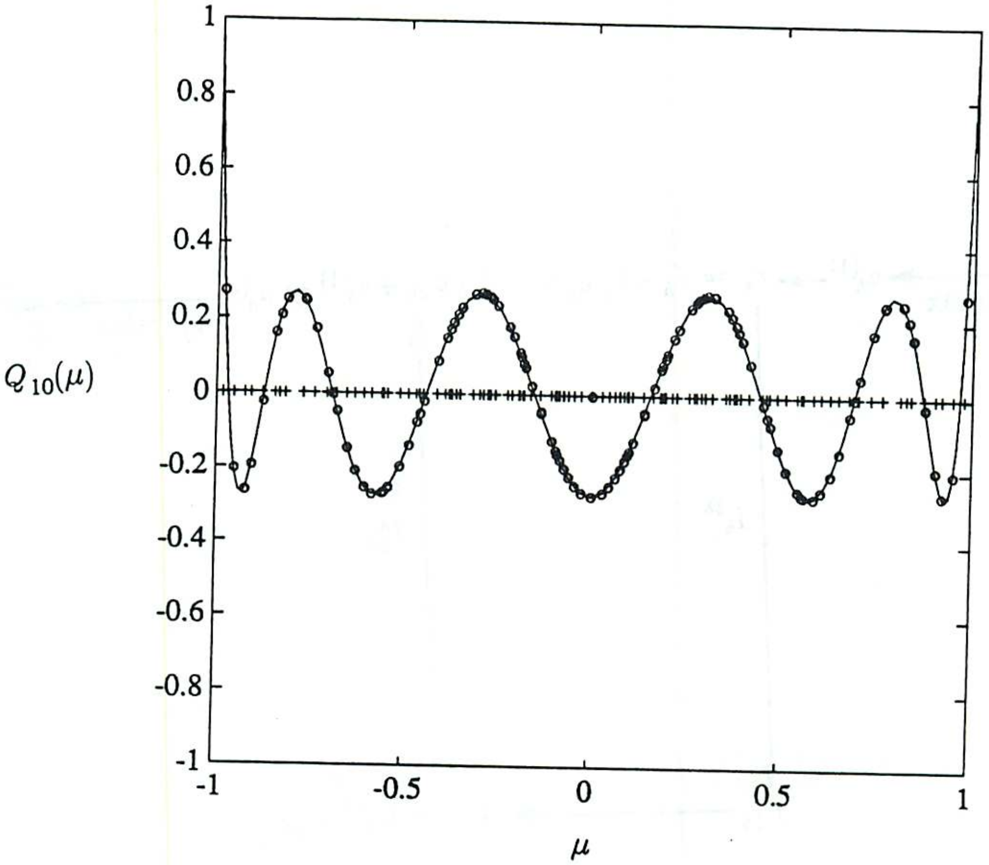

Fig. 5.4 Typical eigenvalue distribution for the Chebyshev semi-iterative method plotted as function of the eigenvalues of the Jacobi iteration ($h = \frac{1}{16}$, $\mu_{\max} = -\mu_{\min} = 0.98$ and $m = 10$).

$$\begin{tikzcd}
u_h^{\mathrm{old}} = u_h^{(0)} \arrow[r, "\nu_1 \text{ relax}"'] & u_h^{(1)} \arrow[r] & r_h = f_h - L_h u_h^{(1)} \arrow[d, "I_h^{2h}"] & I_{2h}^h u_{2h} + u_h^{(1)} = u_h^{(2)} \arrow[r, "\nu_2 \text{ relax}"'] & u_h^{(3)} = u_h^{\mathrm{new}} \\
 & & f_{2h} \arrow[r] & u_{2h} = L_{2h}^{-1} f_{2h} \arrow[u, "I_{2h}^h"'] &
\end{tikzcd}$$

Fig. 6.1 Structure of an $(h,2h)$ two-grid method.

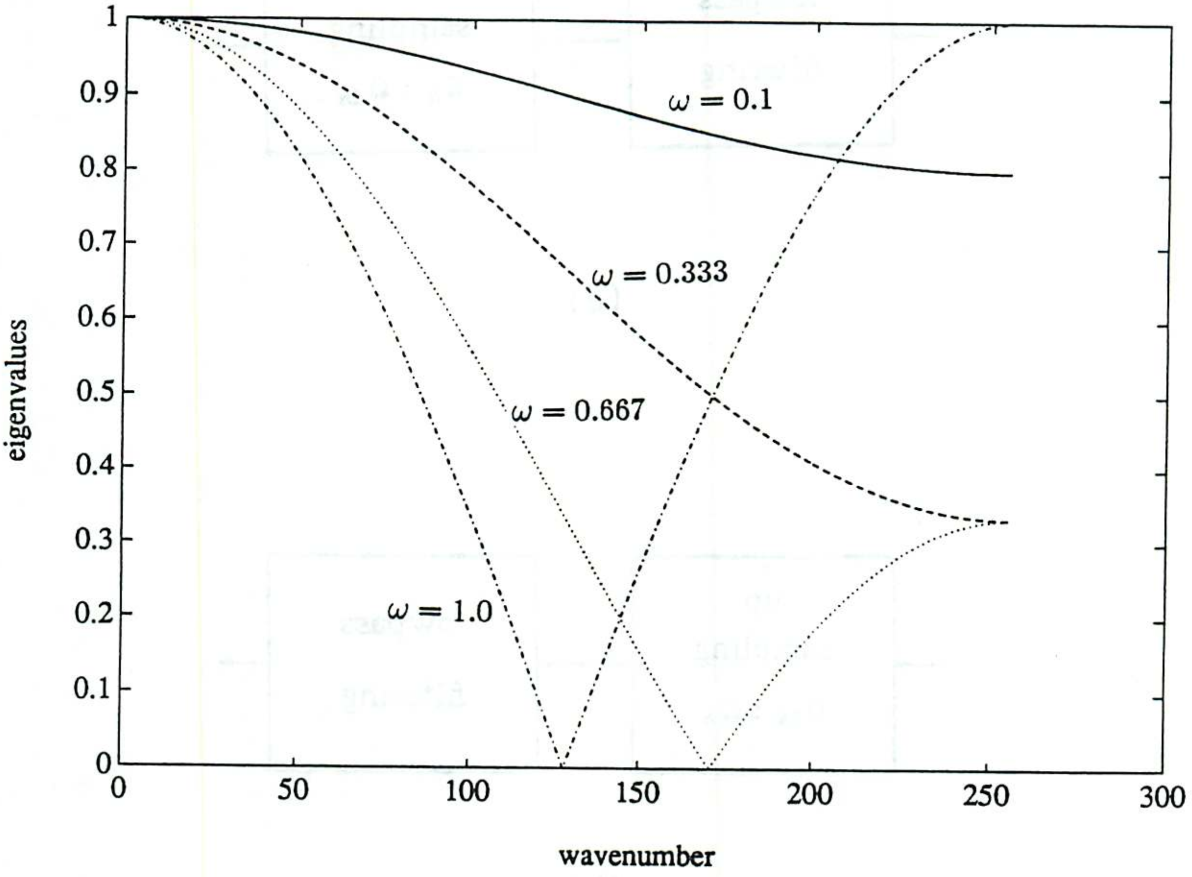

Fig. 6.2 The spectrum of the 1-D damped Jacobi smoother parameterized with ω.

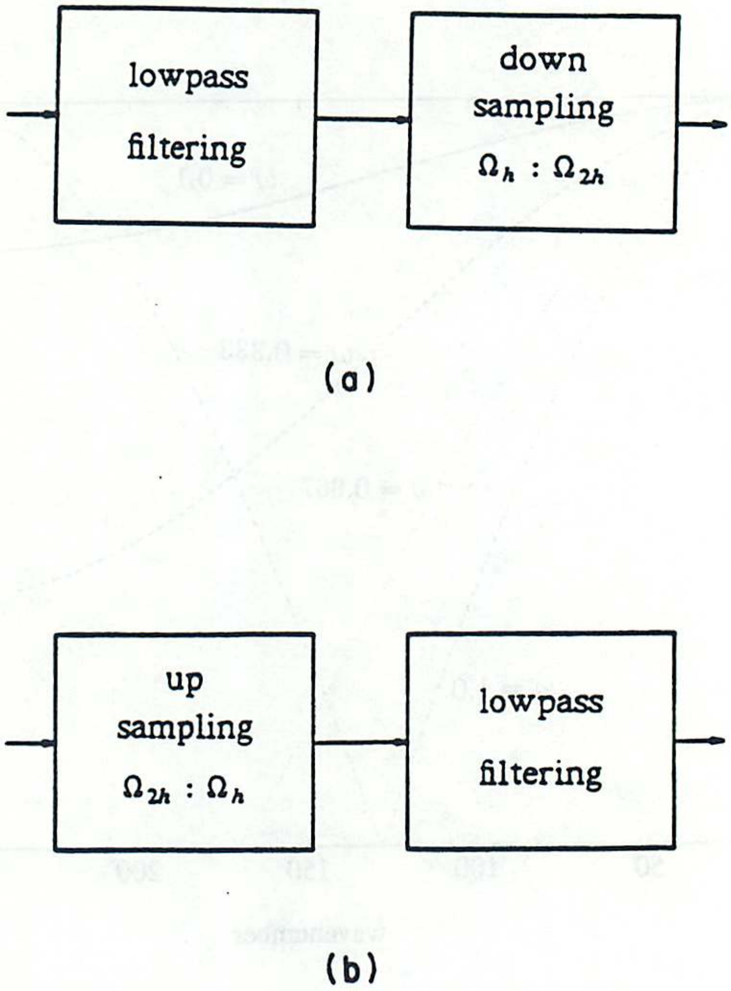

Fig. 6.3 Decomposition of the (a) restriction and (b) interpolation operators.

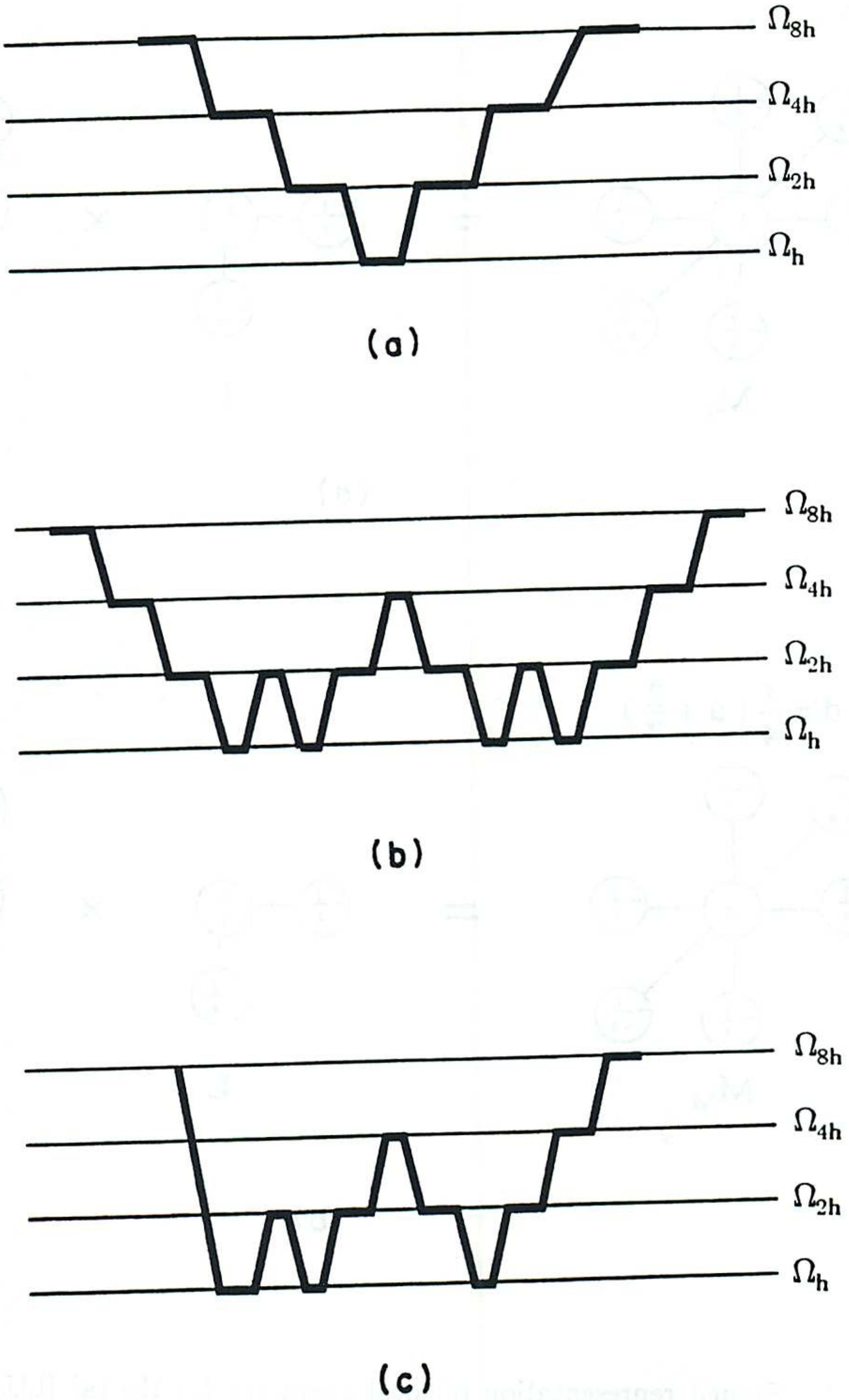

Fig. 6.4 Illustrations of (a) V-cycle, (b) W-cycle, and (c) full multigrid methods.

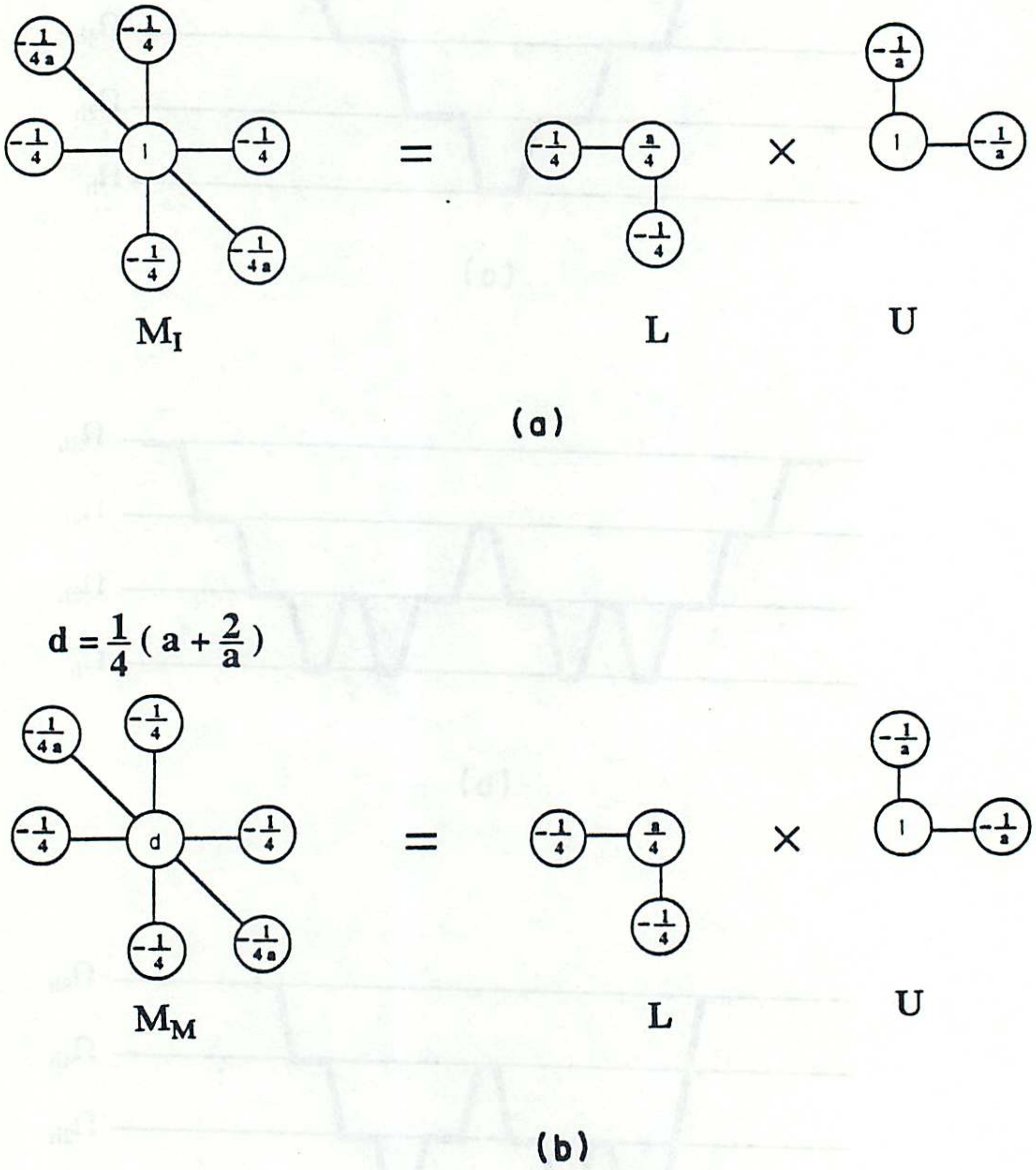

Fig. 7.1 Stencil representation of local operators for the (a) ILU and (b) MILU preconditioners.

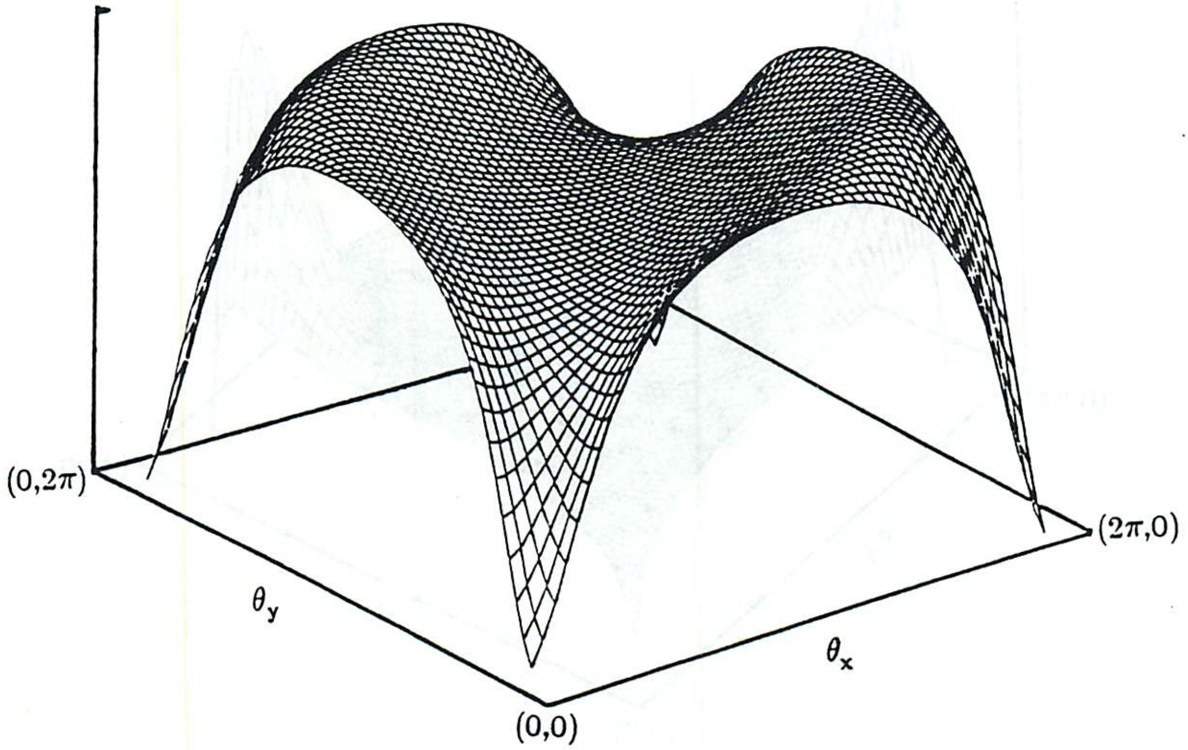

Fig. 7.2 Typical surface plot of the spectrum of the ILU preconditioned Laplacian where $\theta_x = 2\pi k_x h$, $\theta_y = 2\pi k_y h$ and $h = 0.02$.

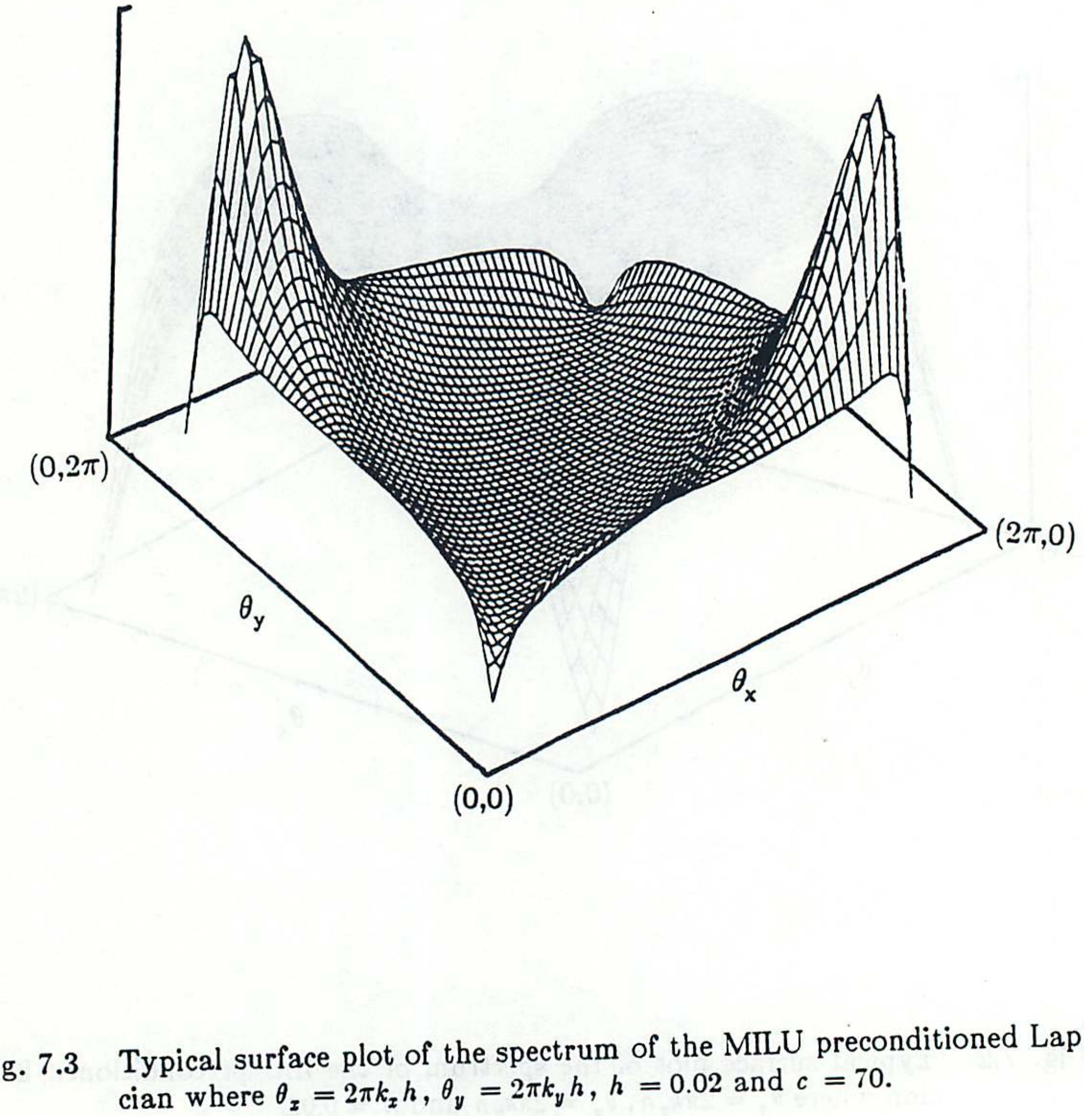

Fig. 7.3 Typical surface plot of the spectrum of the MILU preconditioned Laplacian where $\theta_x = 2\pi k_x h$, $\theta_y = 2\pi k_y h$, $h = 0.02$ and $c = 70$.

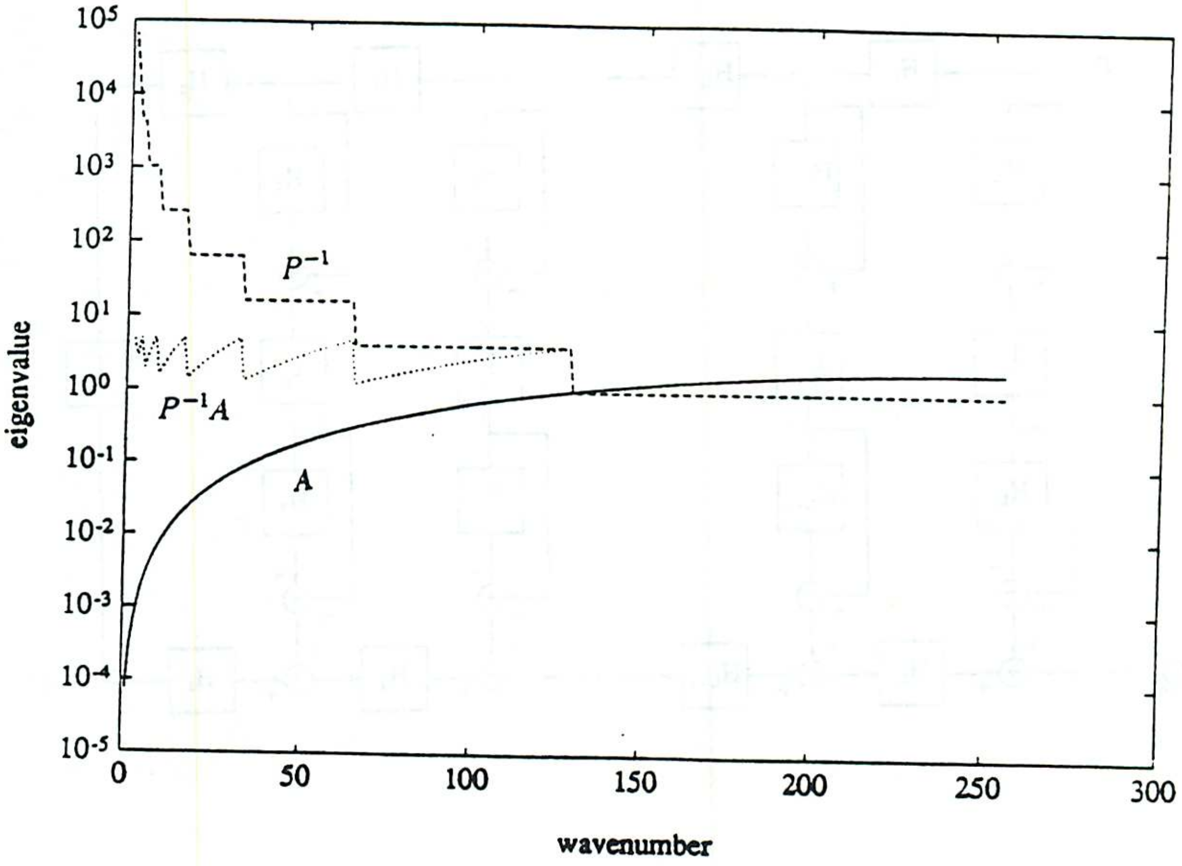

Fig. 7.4 Spectra of A, P^{-1} and $P^{-1}A$.

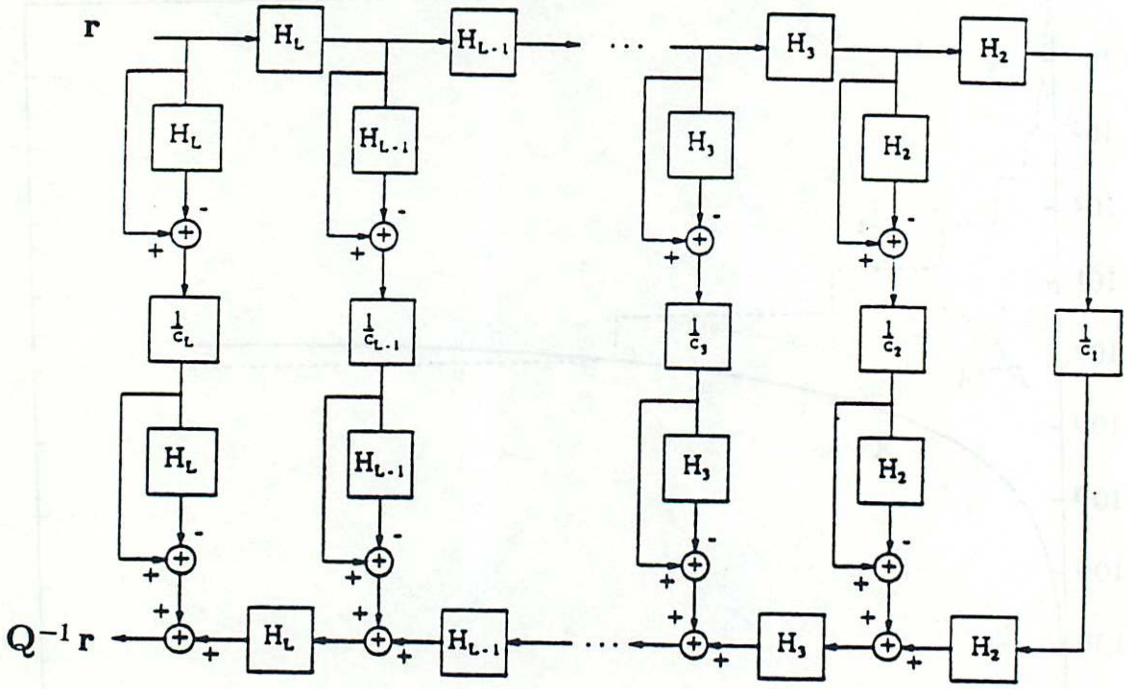

Fig. 7.5 Block diagram of the MF preconditioned with a single discretization grid (SGMF).

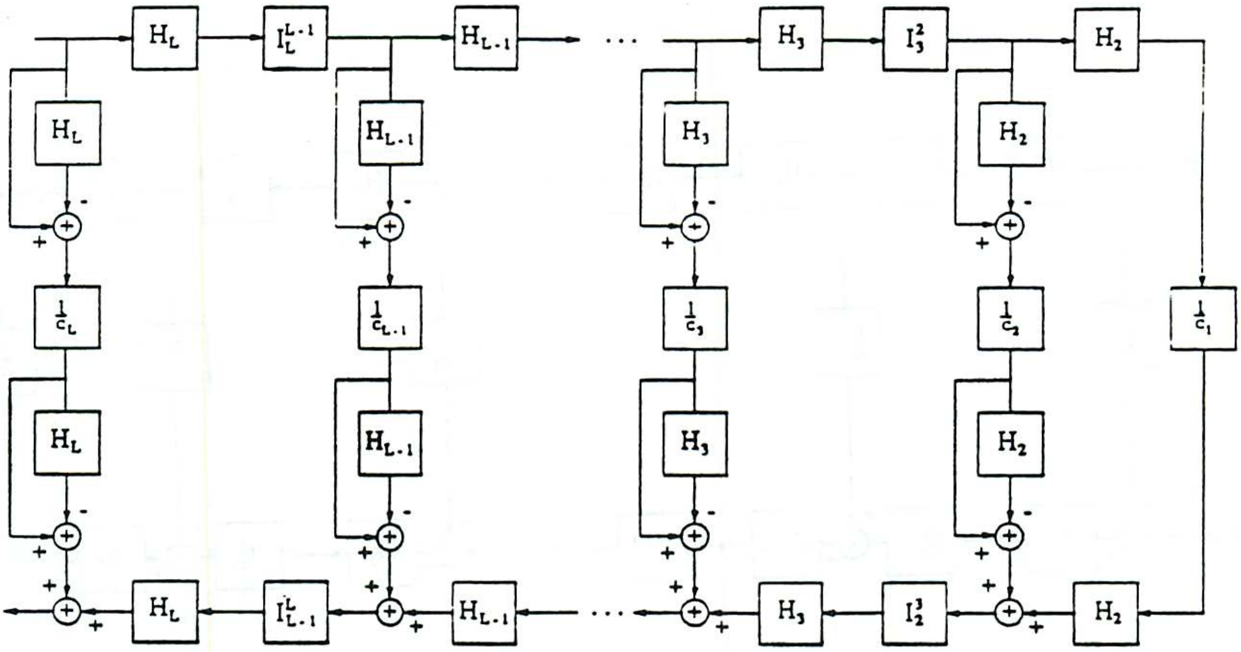

Fig. 7.6 Block diagram of the MGMF preconditioner.

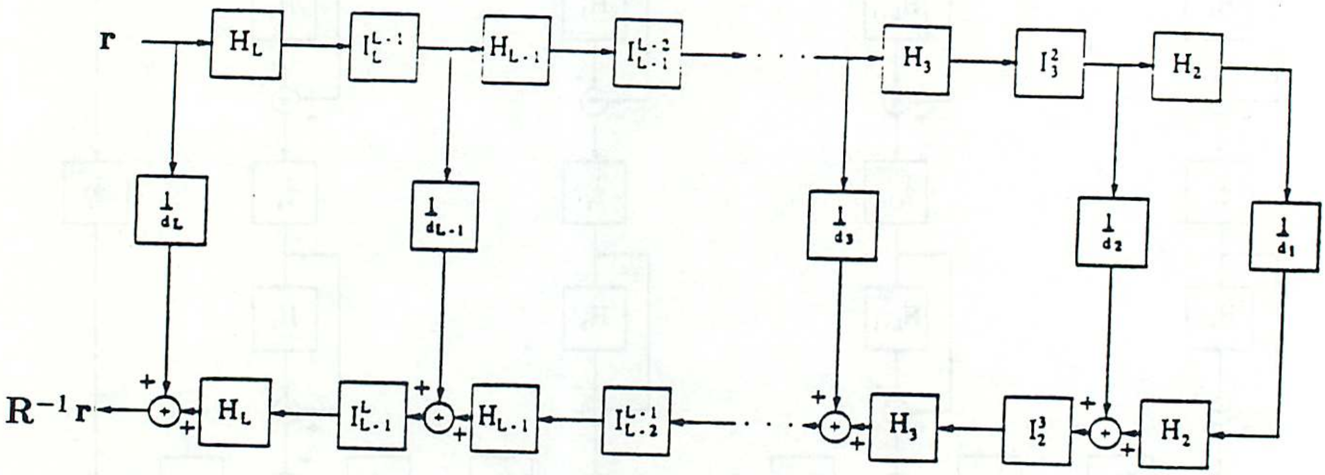

Fig. 7.7 Block diagram of the modified MGMF preconditioner.

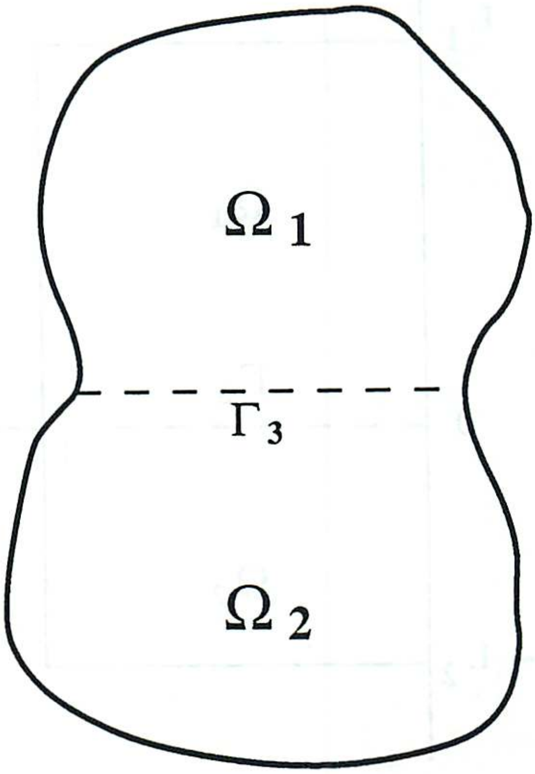

Fig. 8.1 A general domain and its partitioning.

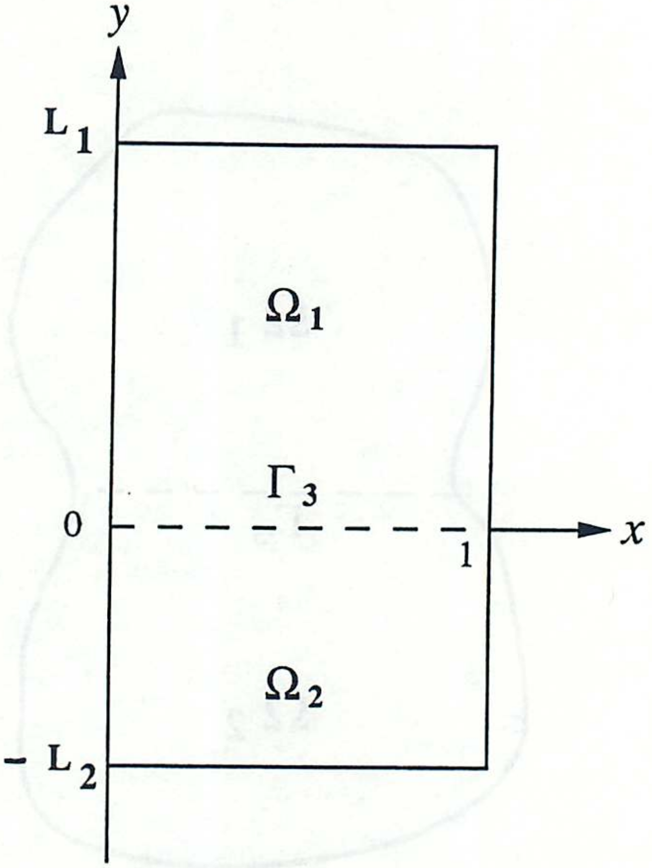

Fig. 8.2 A rectangular domain and its partitioning.

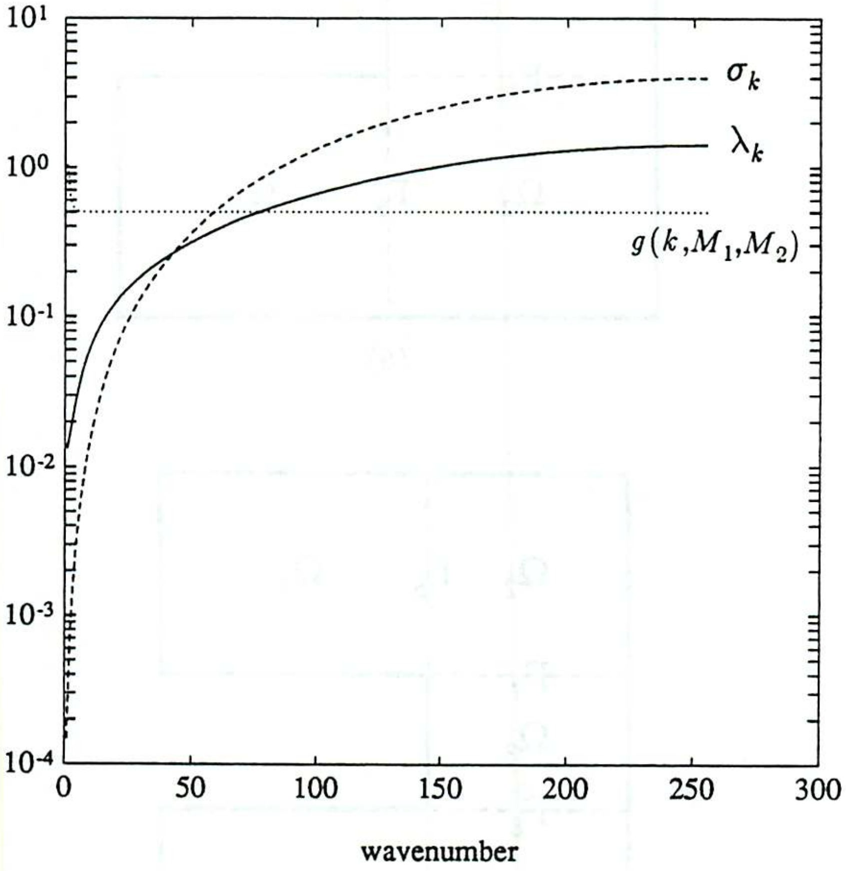

Fig. 8.3 Plots of λ_k, σ_k and $g(k,M_1,M_2)$ as functions of the wavenumber k.

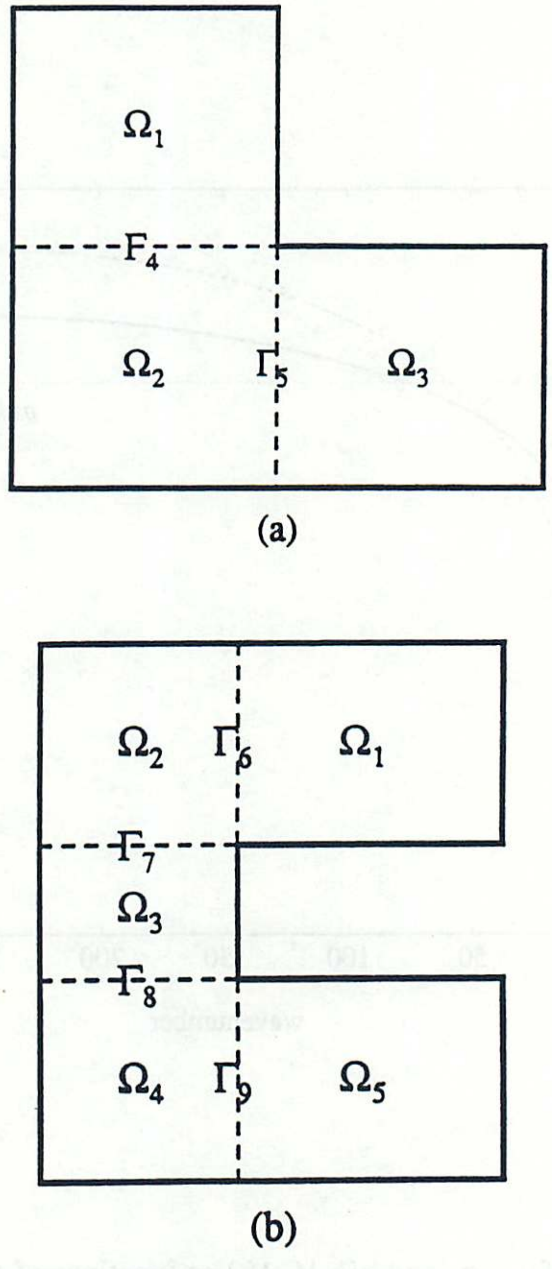

Fig. 8.4 (a) L-shaped and (b) C-shaped domains and their partitionings.