SIAM J. MATRIX ANAL. APPL.
Vol. 11, No. 3, pp. 403–429, July 1990

006

MULTILEVEL FILTERING ELLIPTIC PRECONDITIONERS*

C.-C. JAY KUO[†], TONY F. CHAN[‡][¶], AND CHARLES TONG[§][¶]

Abstract. A class of preconditioners for elliptic problems built on ideas borrowed from the digital filtering theory and implemented on a multilevel grid structure is presented. These preconditioners are designed to be both rapidly convergent and highly parallelizable. The digital filtering viewpoint allows for the use of filter design techniques for constructing elliptic preconditioners and also provides an alternative framework for understanding several other recently proposed multilevel preconditioners. Numerical results are presented to assess the convergence behavior of the new methods and to compare them with other preconditioners of multilevel type, including the usual multigrid method as preconditioner, the hierarchical basis method, and a recent method proposed by Bramble–Pasciak–Xu.

Key words. filtering, multigrid, multilevel, parallel computation, preconditioned conjugate gradient, preconditioners

AMS(MOS) subject classifications. 65N20, 65F10

1. Introduction.

Preconditioned conjugate gradient (PCG) methods have been a very popular and successful class of methods for solving large systems of equations arising from discretizations of elliptic partial differential equations. With the advent of parallel computers in recent years, there has been increased research into effectively implementing these methods on various parallel architectures. In this paper, we present a class of preconditioners for elliptic problems built on ideas from the digital filtering theory and implemented on a multilevel grid structure. Our goal is to work towards preconditioners that are both highly parellelizable and rapidly convergent.

The idea of preconditioning is a simple one, but it is now recognized as critical to the effectiveness of PCG methods. Suppose we would like to solve the symmetric positive definite linear system $Ax = b$, where A arises from discretizing a second-order self-adjoint elliptic partial differential operator. A good preconditioner for A is a matrix M that approximates A well (in the sense that the spectrum for the preconditioned matrix $M^{-1}A$ is clustered around 1 and has a small condition number), and for which the matrix vector product $M^{-1}v$ can be computed efficiently for a given vector v. With such a preconditioner, one then solves in principle the preconditioned system $\tilde{A}\tilde{x} = \tilde{b}$, where $\tilde{A} = M^{-1/2}AM^{-1/2}$, $\tilde{x} = M^{1/2}x$ and $\tilde{b} = M^{-1/2}b$, by the conjugate gradient method.

Since an effective preconditioner plays a critical role in PCG methods, many classical preconditioners have been proposed and studied, especially for second-order elliptic problems. Among these are the Jacobi preconditioner (diagonal scaling), the symmetric successive overrelaxation (SSOR) preconditioner [3], and the incomplete factorization

* Received by the editors September 7, 1989; accepted for publication (in revised form) December 27, 1989.

[†] Signal and Image Processing Institute and Department of Electrical Engineering Systems, University of Southern California, Los Angeles, California 90089-0272 (cckuo@brand.usc.edu). The work of this author was supported by the University of Southern California Faculty Research and Innovation Fund.

[‡] Department of Mathematics, University of California at Los Angeles, Los Angeles, California 90024. (chan@math.ucla.edu)

[§] Department of Computer Science, University of California at Los Angeles, Los Angeles, California 90024 (tong@cs.ucla.edu).

[¶] The work of the second and third authors was supported in part by the National Science Foundation under contract NSF-DMS87-14612, and by the Army Research Office under contract DAAL03-88-K-0085. Part of this work was performed while they were visiting the Research Institute for Advanced Computer Science, National Aeronautics and Space Administration Research Center.

preconditioners (ILU [25] and MILU [15]). These preconditioners have been very successful, especially when implemented on sequential computers.

In the parallel implementation of PCG methods, the major bottleneck is often the parallelization of the preconditioner, since the rest of the PCG methods can be parallelized in a straightforward way (the only potential bottleneck is the need for innerproducts, but many parallel computers do support fast inner-product evaluations). Unfortunately, previous works [12], [16] have shown that for many of the classical preconditioners, there is a fundamental trade-off in the ease of parallelization and the rate of convergence. A principal obstacle to parallelization is the sequential manner in which many preconditioners traverse the computational grid—the data dependence implicitly prescribed by the method fundamentally limits the amount of parallelism available. Reordering the grid traversal (e.g., from natural to red-black ordering) or inventing new methods (e.g., polynomial preconditioners [2], [19]) to improve parallelization usually has an adverse effect on the rate of convergence [12], [23].

The fundamental difficulty can be traced to the global dependence of elliptic problems. An effective preconditioner must account for the global coupling inherent in the original elliptic problem. Preconditioners that use purely local information (such as red-black orderings and polynomial preconditioners) are fundamentally limited in their ability to improve the convergence rate. On the other hand, global coupling through a natural ordering grid traversal is not highly parallelizable. The fundamental challenge is therefore to construct preconditioners that maintain global coupling and are highly parallelizable. Ideas along this line have of course been explored in the development of multigrid methods as solution [10], [17] as well as preconditioning techniques [20], [21], and the more recently proposed hierarchical basis preconditioner [8], [29].

We are thus led to the consideration of preconditioners that share global information through a multilevel grid structure (ensuring a good convergence rate) but perform only local operations on each grid level (and hence highly parallelizable). Compared with a purely multigrid iteration, we have more flexibility in terms of the choice of inter- and intragrid level operators (such as interpolation, projection, and smoothing), since we are using the multilevel iteration within an outer conjugate gradient iteration. One preconditioner of this type has been proposed recently by Bramble, Pasciak, and Xu [9] and Xu [28]. The methods that we propose in this paper are quite similar to their preconditioner, and our digital filtering framework can be looked at as providing an alternative view of their method. It also allows the flexibility in deriving several variants. The approach taken in this paper and that of Bramble, Pasciak, and Xu differs from that of multigrid methods in that the smoothing operation in multigrid methods is replaced by a simple scaling operation. Other types of multilevel preconditioners have been studied by Vassilevski [27], Axelsson and Vassilevski [6], [7], Kuznetsov [24] and Axelsson [4].

The outline of the paper is as follows. In § 2, we describe our framework for deriving multilevel filtering preconditioners for a model problem on a single discretization grid. The basic framework is then extended to the multigrid discretization case in § 3. In § 4, we briefly survey several other preconditioners of the multilevel type. Numerical results for (model, variable coefficient, and discontinuous coefficient) problems in two and three dimensions are presented in § 5, comparing the performance of several multilevel preconditioners, including the usual multigrid method as a preconditioner, the hierarchical basis preconditioner, and the method of Bramble–Pasciak–Xu. Some brief concluding remarks are given in § 6.

We note that the main emphasis of the present paper is on the convergence behavior of these multilevel preconditioners—no attempt is made to assess their parallel efficiency. That will be the subject of a forthcoming paper.

2. Multilevel filtering preconditioners: Fundamentals.

2.1. Motivation. Consider the one-dimensional discrete Poisson equation on $[0, 1]$ with zero boundary conditions on a uniform grid Ω_h,

$$\left(-\frac{1}{2}E+1-\frac{1}{2}E^{-1}\right)u_n=f_n, \qquad n=1, \cdots, N-1, \tag{2.1}$$

where $N = h^{-1} = 2^L$, with integer $L > 1$, and E is the shift operator on Ω_h. We denote the above system by

$$Au=f,$$

where A, u, and f correspond, respectively, to the discrete Laplacian, solution, and forcing functions. Clearly, A is a tridiagonal matrix with diagonal elements $-\frac{1}{2}$, 1 and $-\frac{1}{2}$. It is well known that the matrix A can be diagonalized as

$$A = W^T\Lambda_A W, \tag{2.2}$$

where Λ_A is a diagonal matrix

$$\operatorname{diag}(\lambda_1, \cdots, \lambda_k, \cdots, \lambda_{N-1}), \qquad \lambda_k = 1-\cos(k\pi h),$$

and W is an order $(N-1)^2$ orthogonal matrix whose kth row is

$$w_k^T=\left(\frac{2}{N}\right)^{1/2}(\sin(k\pi h), \cdots, \sin(k\pi nh), \cdots, \sin(k\pi(N-1)h)). \tag{2.3}$$

The diagonalization of the matrix A can be interpreted as the decomposition of the driving and solution functions into their Fourier components, i.e.,

$$\hat{u}_k=\left(\frac{2}{N}\right)^{1/2}\sum_{n=1}^{N-1} u_n \sin(k\pi nh), \qquad \hat{f}_k=\left(\frac{2}{N}\right)^{1/2}\sum_{n=1}^{N-1} f_n \sin(k\pi nh),$$

$$k=1,2, \cdots, N-1.$$

One can easily verify that $\hat{u}_k$ and $\hat{f}_k$ are related via

$$\hat{A}(k)\hat{u}_k=\hat{f}_k, \qquad k=1,2, \cdots, N-1,$$

where

$$\hat{A}(k)=\lambda_k=1-\cos(k\pi h), \tag{2.4}$$

is known as the spectrum of the discrete Laplacian.

In order to invert A, we can make use of (2.2) and obtain a fast Poisson solver:

$$A^{-1}=W^T\Lambda_A^{-1}W. \tag{2.5}$$

The above procedure also serves as the general framework for fast Poisson solvers in cases of higher dimension. However, fast Poisson solvers are not generally applicable for nonseparable elliptic operators and irregular domains. Instead, we want to find good approximations to this solution procedure that are extensible to more general problems and then use them as preconditioners. The fundamental idea is to avoid the use of fast Fourier transform (FFT) and to use instead a sequence of filtering operations to approximate the desired spectral decomposition. This explains the motivation and the name of the multilevel filtering (MF) preconditioner proposed in this paper.

Our main idea for deriving the MF preconditioner for A is to divide all admissible wavenumbers into bands and to approximate the spectrum $\hat{A}(k)$ at each band with some constant. To be more precise, consider the following piecewise constant function in the wavenumber domain

$$\hat{P}(k) = c_l, \qquad k \in B_l, \qquad 1 \leqq l \leqq L,$$

where

$$B_l = \{k : 2^{l-1} \leqq k < 2^l \quad \text{and} \quad k \in I\},$$

is the lth wavenumber band. Let Λ_P be the diagonal matrix with $\hat{P}(k)$ as the kth diagonal element, i.e.,

$$\Lambda_P = \text{diag}\,(\hat{P}(1), \hat{P}(2), \cdots, \hat{P}(N-1)),$$

and $P = W^T \Lambda_P W$. Then, the P-preconditioned Laplacian becomes

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

where

$$\Lambda_{P^{-1}A} = (\Lambda_P)^{-1}\Lambda_A = \text{diag}\left(\frac{\lambda_1}{c_1}, \frac{\lambda_2}{c_2}, \frac{\lambda_3}{c_2}, \cdots, \frac{\lambda_{2^{l-1}}}{c_l}, \cdots, \frac{\lambda_{2^l-1}}{c_l}, \cdots, \frac{\lambda_{N-1}}{c_L}\right).$$

The question is how to choose appropriate c_l's to reduce the condition number $\kappa(P^{-1}A)$.

Suppose that we can find c_l's so that

$$C_1 \leqq \frac{\lambda_k}{c_l} \leqq C_2, \qquad k \in B_l, \qquad 1 \leqq l \leqq L,$$

where C_1 and C_2 are positive constants independent of h. Then, P and A are spectrally equivalent. There are many ways to achieve this goal. For example, we can choose any eigenvalue λ within band B_l to be the constant c_l. For the following discussion, let us consider the choice,

$$c_l = 4^{-(L-1)}. \tag{2.6}$$

The ratio of $\hat{A}(k)$ and $\hat{P}(k)$ is then bounded by

$$4^{L-1}[1 - \cos(2^{-L+l-1}\pi)] \leqq \hat{P}^{-1}(k)\hat{A}(k) < 4^{L-l}[1 - \cos(2^{-L+l}\pi)],$$

for $k \in B_l$. The largest and smallest values of $\hat{P}^{-1}(k)\hat{A}(k)$ for $k \in B$ are bounded. They are, respectively,

$$\lambda_{\max}(P^{-1}A) = \max_k \hat{P}^{-1}(k)\hat{A}(k) < \max_{1 \leqq l \leqq L} 4^{L-l}[1 - \cos(2^{-L+l}\pi)] < \frac{\pi^2}{2},$$

and

$$\lambda_{\min}(P^{-1}A) = \min_k \hat{P}^{-1}(k)\hat{A}(k) \geqq \min_{1 \leqq l \leqq L} 4^{L-l}[1 - \cos(2^{-L+l-1}\pi)] \geqq 1.$$

Note that the last inequalities in the equations above hold independent of L, or equivalently, the grid size h. Thus, the condition number κ of the preconditioned operator $P^{-1}A$ is bounded by a constant

$$\kappa(P^{-1}A) < \frac{\pi^2}{2} \approx 4.93.$$

We plot the spectra $\hat{A}(k)$, $\hat{P}^{-1}(k)$, and $\hat{P}^{-1}(k)A(k)$ in Fig. 2.1 for $N = h^{-1} = 256$ with c_l defined in (2.6).

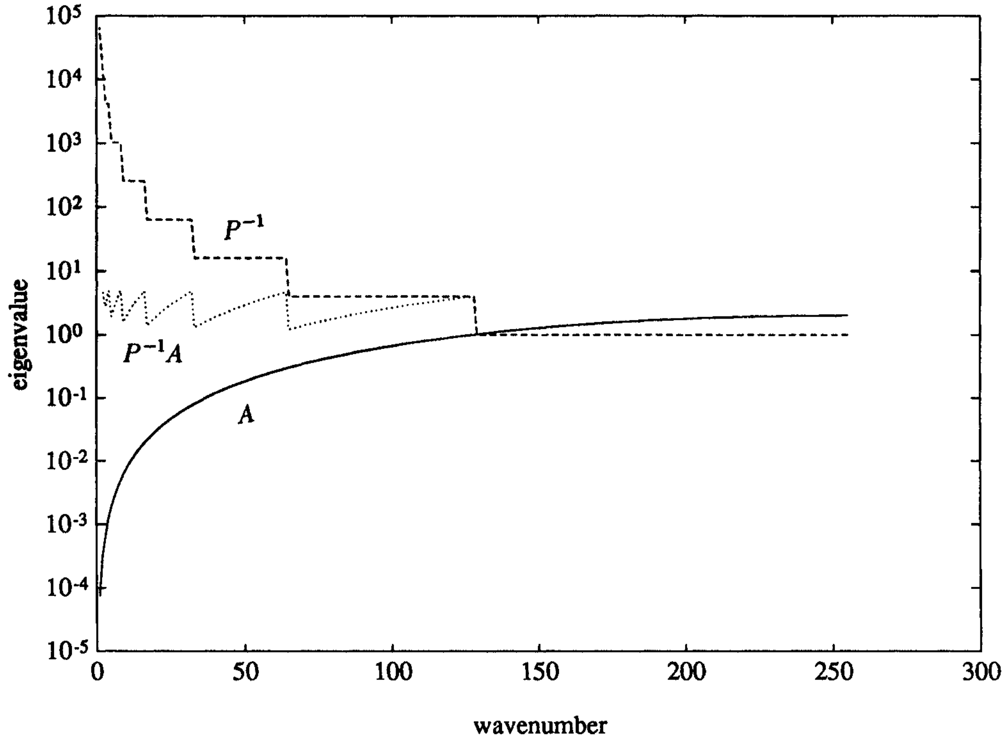


FIG. 2.1. *Spectra of* A, P^{-1} *and* $P^{-1}A$.

2.2. Decomposition and synthesis based on filtering. The preconditioning procedure

$$P^{-1}r = W^T\Lambda_P^{-1}Wr, \tag{2.7}$$

consists of three building blocks: decomposition, scaling, and synthesis. The construction of these building blocks with ideal digital filters will be discussed in this section.

Let us rewrite (2.7) as

$$P^{-1}r = \left(\sum_{l=1}^{L}\frac{1}{c_l}W_l^TW_l\right)r, \tag{2.8}$$

where W_l, $1 \leqq l \leqq L$, 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. If we implement W_l and W_l^T in decomposition and synthesis respectively, FFT and inverse FFT are needed. This is due to the fact that W_l is a mapping from the space domain to the wavenumber domain, whereas W_l^T is a mapping from the wavenumber domain to the space domain. By performing $P^{-1}r$ according to (2.8), we are led to an algorithm similar to the fast Poisson solver (2.5).

Let $F_l = W_l^TW_l$. Then, F_l is a mapping from the space domain to the space domain. In addition, we have

$$F_l = W^T\Lambda_{F_l}W,$$

where Λ_{F_l} is a diagonal matrix whose kth element is

$$\hat{F}_l(k) = \begin{cases} 1, & k \in B_l \\ 0, & \text{otherwise.} \end{cases}$$

The spectral property of F_l is characterized by $\hat{F}_l(k)$. A digital filter is a mapping from the space domain to the space domain satisfying a certain spectral property. Since F_l passes Fourier components in band B_l and blocks components in other bands, it is called a bandpass filter. We might perform the preconditioning (2.8) by implementing F_l's with digital filters in decomposition and a simple addition operation in synthesis. However, the resulting scheme loses a certain symmetrical property in decomposition and synthesis. This turns out to be important in the multigrid context (see § 3).

This motivates us to write (2.7) in another form as

$$P^{-1}r=\left(\sum_{l=1}^{L}\frac{1}{c_l}F_l^TF_l\right)r, \tag{2.9}$$

where bandpass filters $F_l\,(=F_l^T)$ are implemented in both decomposition and synthesis building blocks. In the context of multirate signal processing [13], the separation of a function into several components, each of which is confined to a narrow wavenumber band, is known as the *filter bank analyzer* and the reverse process is the *filter bank synthesizer*. Although there exist many ways to implement the filter bank analyzer and synthesizer, a simple design illustrated by the block diagram of Fig. 2.2 will be sufficient for our purpose. This design, called the single-grid multilevel filtering (SGMF) preconditioner, is based on the cascade of a sequence of elementary filters $H_L,\ H_{L-1},\ \cdots,\ H_2$, where the function of H_l is to preserve Fourier components contained in bands $B_1,\ \cdots,\ B_{l-1}$ and to eliminate Fourier components contained in band B_l. In terms of mathematics, we define

$$H_l=W^T\Lambda_{H_l}W, \tag{2.10a}$$

where Λ_{H_l} is a diagonal matrix with the kth element

$$\hat{H}_l(k)=\begin{cases}1, & k\in B_1\cup\cdots\cup B_{l-1}\\ 0, & k\in B_l.\end{cases} \tag{2.10b}$$

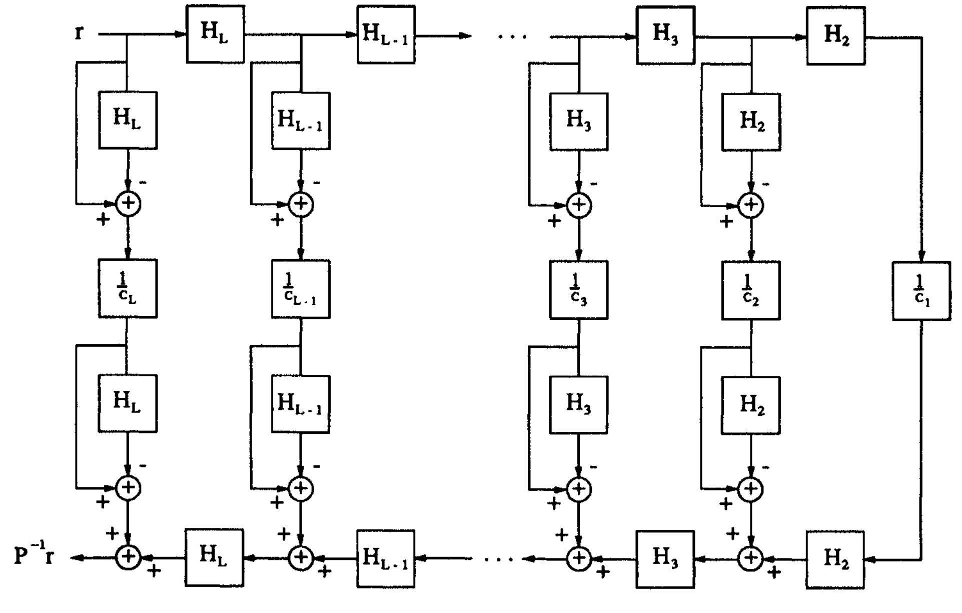


FIG. 2.2. *Block diagram of the* MF *preconditioner with a single discretization grid* (SGMF).

From Fig. 2.2, we see that the filters F_l are related to the filters H_l via

$$F_L = I - H_L, \tag{2.11a}$$

$$F_l = (I - H_l)\left[\prod_{p=l+1}^{L} H_p\right], \qquad 2 \leqq l \leqq L-1, \tag{2.11b}$$

$$F_1 = \prod_{p=2}^{L} H_p. \tag{2.11c}$$

It is easy to verify that F_l's satisfy the desired bandpass characteristics by pre- and post-multiplying (2.11) with W and W^T, respectively. Note also that the values of $\hat{H}_l(k)$ for $k \in B_{l+1} \cup \cdots \cup B_L$ do not influence the bandpass feature of F_l's. This observation simplifies the design of H_l's (see § 2.3).

To save computational work, we can further simplify the SGMF preconditioner in Fig. 2.2 by deleting the paths and the associated work corresponding to $I - H_l$. As given in Fig. 2.3, we have the modified SGMF preconditioner

$$Q^{-1}r = \left(\sum_{l=1}^{L} \frac{1}{d_l} G_l^T G_l\right) r, \tag{2.12}$$

where

$$G_L = I,$$

$$G_l = \prod_{p=l+1}^{L} H_p, \quad \text{for} \quad 1 \leqq l \leqq L-1.$$

Note that bandpass filters F_l in the preconditioner P have been replaced by lowpass filters G_l in the preconditioner Q. By choosing d_l's appropriately, we can make Q behave the same as P. With the preconditioner Q, Fourier components of band B_l exist in the first $L - l + 1$ levels and these components are multiplied by $d_L^{-1}, \cdots, d_l^{-1}$, respectively. Therefore, the scaling constants d_l's are implicitly defined via

$$\sum_{i=l}^{L} \frac{1}{d_i} = \frac{1}{c_l}. \tag{2.13}$$

Solving (2.13) for d_l gives

$$d_L = c_L \quad \text{and} \quad d_l = \frac{1}{c_l^{-1} - c_{l+1}^{-1}}, \qquad l = L-1, \cdots, 1. \tag{2.14}$$

However, we observe from numerical experiments that the parameter sets $\{c_l\}$ and $\{d_l\}$

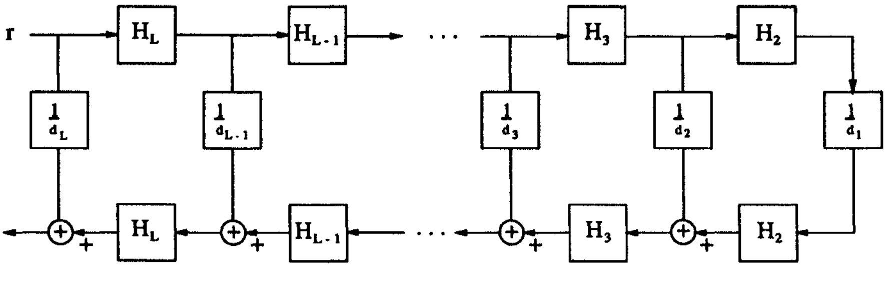


FIG. 2.3. *Block diagram of the modified* SGMF *preconditioner.*

used in Fig. 2.3 give about the same convergence rate. This can be explained by the observation that, for small l, $d_l \approx c_l$, since $c_l^{-1} \gg c_{l+1}^{-1}$.

2.3. Design of elementary filters. Consider the design of the filter H_L appearing at the first stage. The H_L have the following ideal lowpass characteristic,

$$\hat{H}_L(k) = \begin{cases} 1, & 0 \leqq k < 2^{L-1} \\ 0, & 2^{L-1} \leqq k \leqq 2^L. \end{cases} \tag{2.15}$$

From (2.10), we find that H_L is an $(N-1)^2$ full matrix. Thus, the operation $H_L v$ for an arbitrary vector v has a complexity proportional to $O(N^2)$. This is too expensive to perform. Therefore, we seek the approximation of the ideal lowpass filter H_L with a nonideal lowpass filter $H_{L,J}$, which is a symmetric band matrix of bandwidth $O(J)$ with the spectral property $\hat{H}_{L,J}(k) \approx \hat{H}_L(k)$ for $1 \leqq k \leqq N-1$. Consequently, the operation $H_{L,J}v$ only has a complexity proportional to $O(JN)$.

Let us write the nonideal lowpass filter of the form

$$H_{L,J} = a_0 + \sum_{j=1}^{J} a_j(E^j + E^{-j}), \tag{2.16}$$

where the coefficients a_0 and a_j's are to be determined. In order to define the operation

$$H_{L,J}v_n = a_0 + \sum_{j=1}^{J} a_j(v_{n+j} + v_{n-j})$$

for any vector v_n appropriately, the odd-periodic extension of v_n is assumed,

$$v_{-n} = -v_n \quad \text{and} \quad v_{n+2pN} = v_n, \quad \text{for integer } p.$$

This implies that $H_{L,J}$ corresponds to a circulant matrix. The above odd-periodic assumption is used only for analyzing and designing $H_{l,J}$'s in this section. The actual implementation of the MF preconditioner with a multigrid discretization described in § 3 does not rely on this assumption.

There are numerous ways to determine the coefficients a_0 and a_j's depending on what approximation criteria are to be used. The operator $H_{L,J}$ has the eigenfunction $\sin(k\pi nh)$ with the eigenvalue

$$\hat{H}_{L,J}(k) = a_0 + 2\sum_{j=1}^{J} a_j \cos(k\pi jh).$$

Here we consider a class of lowpass filters based on the following two criteria:

(1) $\hat{H}_{L,J}\left(\frac{N}{2}\right) = \frac{1}{2}$ and $\hat{H}_{L,J}(k) - \frac{1}{2} = -\left[\hat{H}_{L,J}(N-k) - \frac{1}{2}\right]$,

(2) $\hat{H}_{L,J}(0) = 1$ and the first jth derivatives $(1 \leqq j \leqq J)$ of $\hat{H}_{L,J}(0)$ are all zero.

The first criterion implies that the function $\hat{H}_{L,J}(k) - \frac{1}{2}$ is odd symmetric with respect to $k = N/2$. A direct consequence of this criterion is that

$$a_0 = \frac{1}{2} \quad \text{and} \quad a_j = 0, j \text{ positive even.}$$

The second criterion, called the *maximally flat* criterion [18], requires the approximation at the origin to be as accurate as possible. It is used to determine a_j with odd j. In Table 2.1, we list coefficients a_j for $J = 1, 3, 5$ obtained according to criteria (1) and (2) and plot their spectra in Fig. 2.4 with $N = 2^8 = 256$. The larger J becomes, the better the approximation is.

TABLE 2.1
Coefficients of a class of nonideal lowpass filters.

J	a_0	a_1	a_3	a_5
1	$\frac{1}{2}$	$\frac{1}{4}$	0	0
3	$\frac{1}{2}$	$\frac{9}{32}$	$\frac{-1}{32}$	0
5	$\frac{1}{2}$	$\frac{150}{512}$	$\frac{-25}{512}$	$\frac{3}{512}$

As illustrated in Figs. 2.2 and 2.3, the low wavenumber band of the function r is used as the input to the filter H_{L-1} at the next stage. The filter H_{L-1} can be constructed with the same set of coefficients used by H_L, i.e.,

$$H_{L-1,J} = a_0 + \sum_{j=1}^{J} a_j(E^{2j} + E^{-2j}). \tag{2.17}$$

Comparing (2.16) and (2.17), we see that the only difference between $H_{L,J}$ and $H_{L-1,J}$ is the position of grid points used for averaging. For the first-stage filter $H_{L,J}$, local averaging is used. For the second-stage filter $H_{L-1,J}$, we consider averaging between points separated by $2h$. This design is due to the following reason. From (2.17), we see that the filter $H_{L-1,J}$ has the spectrum

$$\hat{H}_{L-1,J}(k) = a_0 + 2\sum_{j=1}^{J} a_j \cos(k\pi j2h),$$

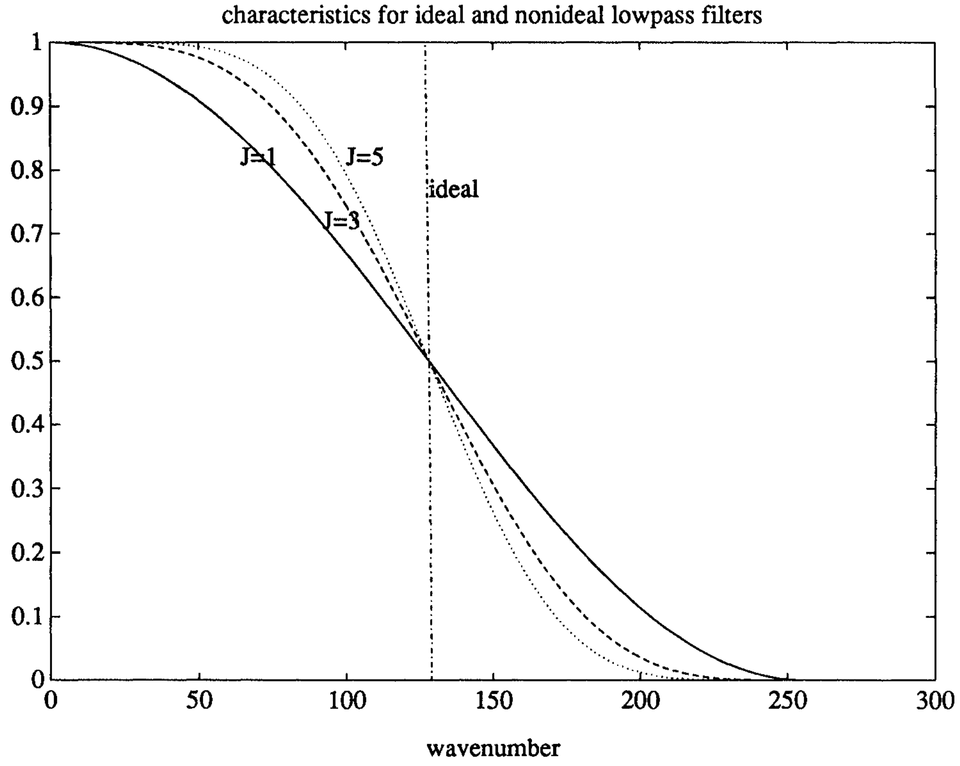


FIG. 2.4. *Spectra of maximally flat lowpass filters $H_{L,J}$ with J = 1, 3, 5.*

and that $\hat{H}_{L-1,J}(k)$ is related to $\hat{H}_{L,J}(k)$ via

$$\hat{H}_{L-1,J}(k)=\hat{H}_{L,J}(2k).$$

Consequently, for functions consisting only of components in low wavenumber region $1 \leqq k < 2^{L-1}$, $\hat{H}_{L-1}$ behaves like a lowpass filter, which preserves components in the region $1 \leqq k < 2^{L-2}$ and filters out components in the region $2^{L-2} \leqq k < 2^{L-1}$. However, note that H_l, $l < L$ is not a lowpass filter with respect to the entire wavenumber band.

By applying the same procedure recursively, we can approximate the general elementary filter H_l on a uniform infinite grid as

$$H_{l,J}=a_0+\sum_{j=1}^{J} a_j(E^{2^{L-l}j}+E^{-2^{L-l}j}), \qquad 2\leqq l\leqq L, \tag{2.18}$$

where the coefficients a_j's are listed in Table 2.1. The spectrum of $H_{l,J}$ is

$$\hat{H}_{l,J}(k)=a_0+2\sum_{j=1}^{J} a_j \cos(k\pi j 2^{L-l}h), \qquad 2\leqq l\leqq L. \tag{2.19}$$

According to (2.11), we can construct nonideal bandpass filters $F_{l,J}$ with nonideal elementary filters $H_{l,J}$,

$$F_{L,J}=I-H_{L,J}, \tag{2.20a}$$

$$F_{l,J}=(I-H_{l,J})\left(\prod_{p=l+1}^{L} H_{p,J}\right), \qquad 2\leqq l\leqq L-1, \tag{2.20b}$$

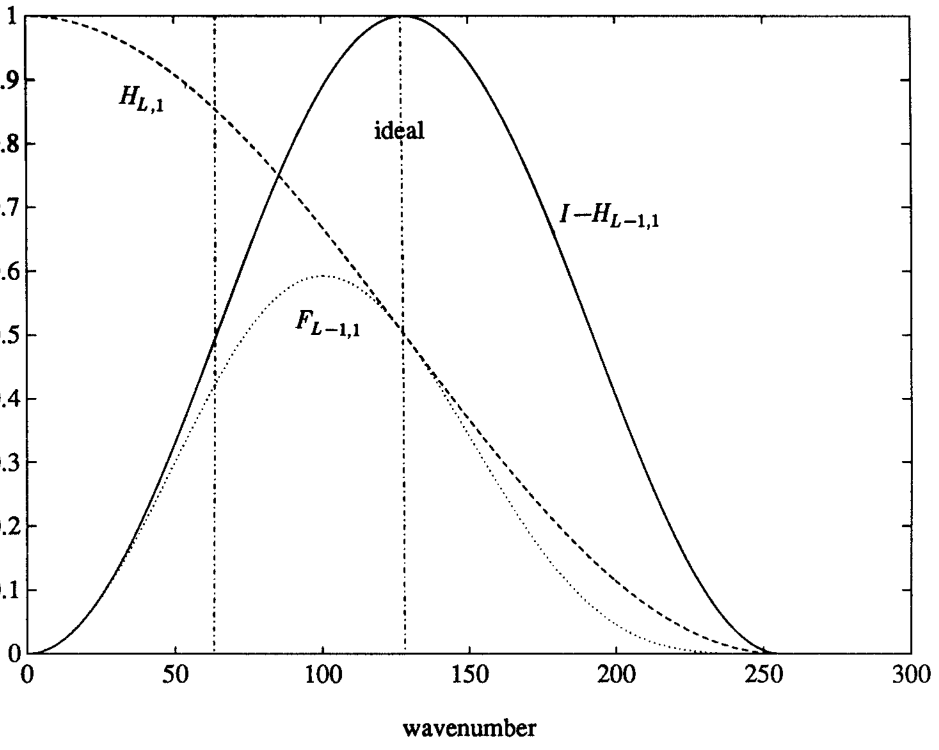


FIG. 2.5. *Spectra of $H_{L,J}$, $I-H_{L-1,J}$ and $F_{L-1,J}$ with $J=1$.*

$$F_{1,J}=\prod_{p=2}^{L} H_{p,J}. \tag{2.20c}$$

To give an example, the construction of $F_{L-1,J}$ with $J=1$ is illustrated in Fig. 2.5. Note that the elementary filter $H_{l,J}$ given by (2.18) is symmetric. So is the bandpass filter $F_{l,J}$. Finally, we obtain the nonideal MF-preconditioner

$$P_J^{-1}r=\left(\sum_{l=1}^{L}\frac{1}{c_l}F_{l,J}^{T}F_{l,J}\right)r, \tag{2.21}$$

which approximates the ideal MF-preconditioner P given by (2.9).

It is worthwhile to summarize the similarities and differences between the fast Poisson solver (2.5) and the SGMF preconditioning (2.21). They are both based on spectral decomposition. The fast Poisson solver decomposes a function into its Fourier components through the FFT, whereas the MF preconditioner approximately decomposes it into a certain number of bands through filtering. The filtering operations, which correspond to local averaging processes, can be easily adapted to irregular grids and domains and variable coefficients. In contrast, the FFT is primarily applicable to constant coefficient problems with regular grids and domains. Besides, for the fast Poisson solver we usually require detailed knowledge of the spectrum. But for the MF preconditioner we have only to estimate how the spectrum varies from one band to another.

2.4. Fourier analysis and higher-dimensional cases. Since the MF preconditioner P_J and the Laplacian A share the same eigenvectors, i.e., Fourier sine functions, the spectrum and condition number of the MF-preconditioned Laplacian can be analyzed conveniently by Fourier analysis. From (2.20), we have the following spectral relationship

$$\hat{F}_{L,J}(k)=1-\hat{H}_{L,J}(k), \tag{2.22a}$$

$$\hat{F}_{l,J}(k)=(1-\hat{H}_{l,J}(k))\left(\prod_{p=l+1}^{L}\hat{H}_{p,J}(k)\right), \qquad 2\leqq l\leqq L-1, \tag{2.22b}$$

$$\hat{F}_{1,J}(k)=\prod_{p=2}^{L}\hat{H}_{p,J}(k), \tag{2.22c}$$

where $\hat{H}_{l,J}(k)$, $1\leqq l\leqq L$, are given by (2.19). Using (2.4), (2.6), and (2.22), we can determine the eigenvalues of $P_J^{-1}A$,

$$\lambda(P_J^{-1}A)=\hat{P}_J^{-1}(k)\hat{A}(k)=\sum_{l=1}^{L}\frac{1}{c_l}\hat{F}_{l,J}^{T}(k)\hat{F}_{l,J}(k)\hat{A}(k).$$

The eigenvalues $\lambda(P_J^{-1}A)$ are plotted as a function of k with $J=1, 3, 5$ and $h^{-1}=256$ in Fig. 2.6. We should compare these spectra with that in Fig. 2.1 based on the ideal filtering assumption. All of them have one common feature. That is, eigenvalues are redistributed in such a way that there exist many local maxima and minima. The condition numbers for $J=1, 3, 5$ are 2.50, 1.88, and 1.93, respectively. Note that these numbers are in fact smaller than the condition number 4.93 obtained with ideal filtering. The precise reason for this phenomenon is still not clear to us. It might be related to the smoothness of the eigenvalue distribution curves. The eigenvalue distribution for $P^{-1}A$ in Fig. 2.1 has many keen edges. However, these edges are smoothed by nonideal digital filters as shown in Fig. 2.6.

The generalization of the MF preconditioner to two- or three-dimensional problems on square or cube domains can be done straightforwardly. For example, we may construct

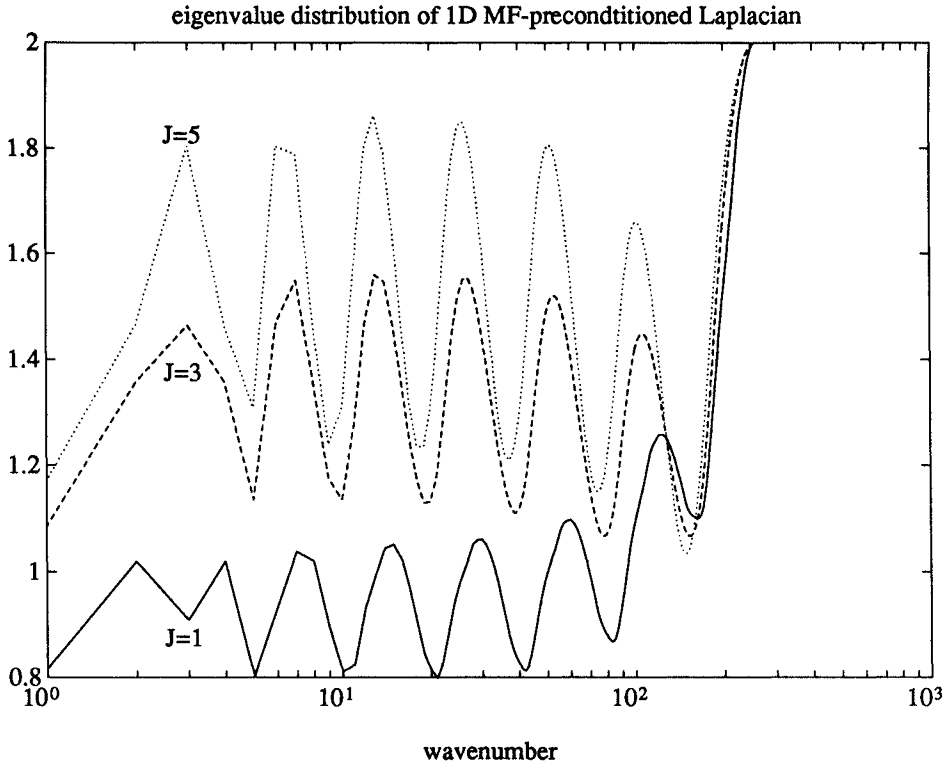


FIG. 2.6. *Eigenvalues of* $P_J^{-1}A$ *with* $J = 1, 3, 5$.

the two-dimensional elementary filter by the tensor product of one-dimensional elementary filters along the x- and y-directions,

$$H_{l,J}=\left(a_0+\sum_{j=1}^{J} a_j(E_x^{2^{L-l_j}}+E_x^{-2^{L-l_j}})\right)\times\left(a_0+\sum_{j=1}^{J} a_j(E_y^{2^{L-l_j}}+E_y^{-2^{L-l_j}})\right),$$

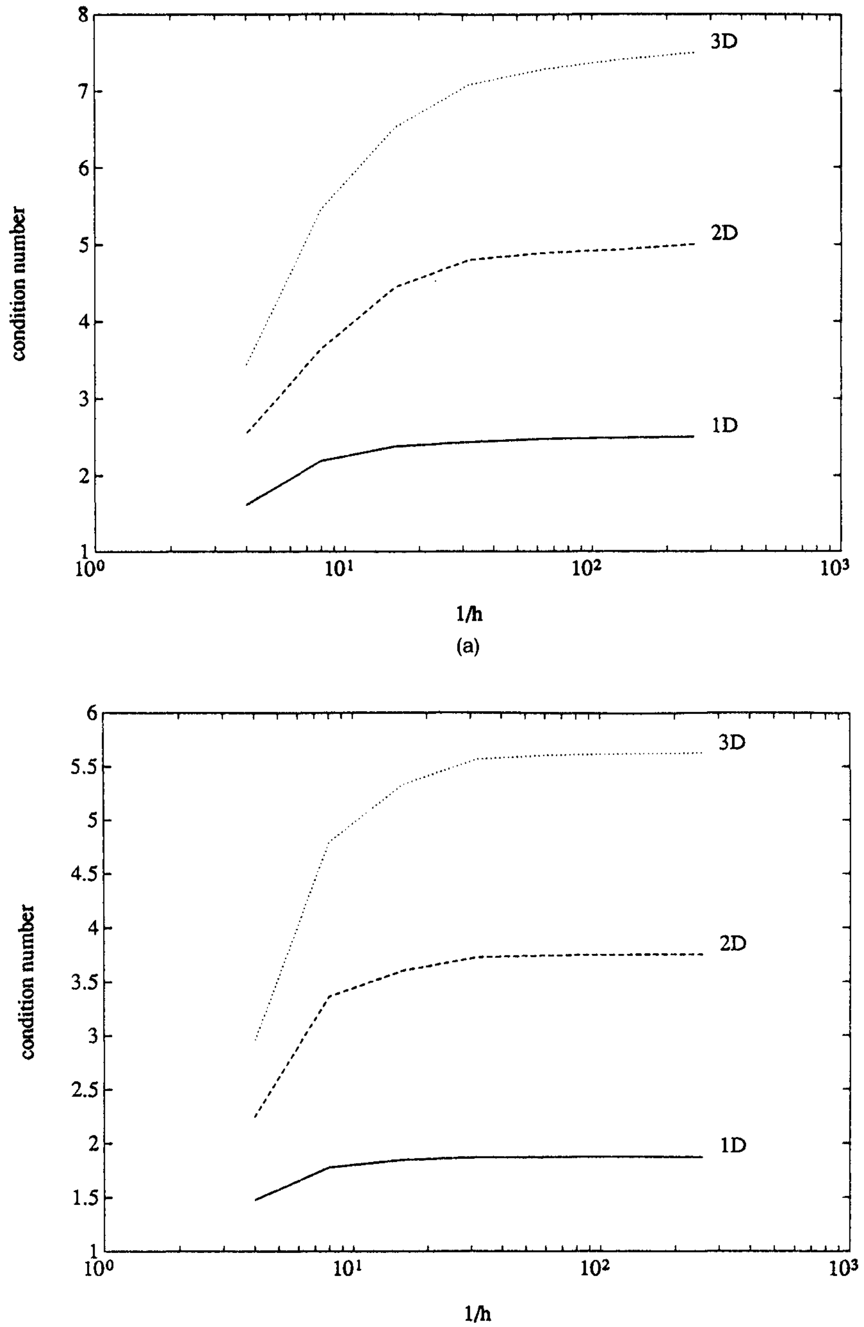
which can be further simplified by using operator algebra [14]. For example, the coefficients for $H_{L,1}$ can be written in stencil form as

$$H_{L,1}:\frac{1}{16}\begin{vmatrix}1&2&1\\2&4&2\\1&2&1\end{vmatrix}. \tag{2.23}$$

Similarly, the three-dimensional elementary filter can be obtained by the tensor product of three one-dimensional filters along the x-, y- and z-directions.

The condition numbers of one-, two-, and three-dimensional MF-preconditioned Laplacians with two types of nonideal filters ($J = 1$ and $J = 3$) are computed and plotted as functions of the grid size h in Figs. 2.7 (a) and (b). These figures show that P_J and A are spectrally equivalent.

The discussion in § 2.3 is based on the odd-periodic property of the sequence v_n. However, this may not be easily implementable for general multidimensional problems with nonrectangular domains. The difficulty arises when the size of $H_{l,J}$ is so large that it operates on points outside the domain. There are two possible solutions. It may be preferable to construct filters of larger size by the repeated application of filters of smaller size. For example, we can apply the filter $H_{L,J}$ (2.16) with $J = 1$ twice. This is equivalent



(a)

(b)

FIG. 2.7. *Condition numbers of the* MF-*preconditioned Laplacian with* (a) $J = 1$ *and* (b) $J = 3$.

to a filter of size 5,

$$H_{L,1}^2 = \left(\frac{1}{4}E^{-1}+\frac{1}{2}+\frac{1}{4}E\right)^2 = \frac{1}{16}E^{-2}+\frac{1}{4}E^{-1}+\frac{3}{8}+\frac{1}{4}E+\frac{1}{16}E^2.$$

Another possibility is to apply smaller filters at points close to boundaries and larger filters at points far away from boundaries. Note also that, for fixed J, the size of the elementary filter $H_{l,J}$ increases as l decreases. However, this problem can be resolved by incorporating the multigrid discretization structure into the above multilevel filtering framework as described in § 3.

3. Multigrid multilevel filtering (MGMF) preconditioners. In § 2, we discussed the construction of the MF preconditioner for the model Poisson problem based on a single discretization grid. This section will discuss the generalization of this preconditioning technique so that it can be implemented more efficiently and applied to more general self-adjoint elliptic partial differential equation (PDE) problems.

The filtering operation described above is performed at every grid point at all levels $2 \leqq l \leqq L$. Since there are $O(\log N)$ levels and $O(JN)$ operations per level, where N and J denote, respectively, the order of unknowns and the filter size, the total number of operations required is proportional to $O(JN \log N)$. However, since waveforms consisting only of low wavenumber components can be well represented on coarser grids, we can use the multigrid philosophy [10], [17] and incorporate the multigrid discretization structure into the filtering framework described in § 2. That is, we construct a sequence of grids Ω_l of sizes $h_l = O(2^{-l})$, $1 \leqq l \leqq L$, to represent the decomposed components. Then, the total number of unknowns is $O(N)$ and consequently the total number of operations per MF preconditioning step is $O(JN)$. Note that J is a constant independent of N.

The block diagram of the multigrid multilevel filtering (MGMF) preconditioner is depicted in Fig. 3.1. It is obtained by inserting down-sampling (I_l^{l-1}) and up-sampling (I_{l-1}^l) operators into the SGMF preconditioner. With the notation commonly used in the multigrid literatures, the down-sampling and up-sampling operators for grids Ω_l ($h_l = 2^{L-l}h$) and Ω_{l-1} ($h_{l-1} = 2^{L-l+1}h$) can be defined as

$$I_l^{l-1}: \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{vmatrix}_l^{l-1}, \qquad I_{l-1}^l: \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{vmatrix}_{l-1}^{l}.$$

It is easy to verify that a lowpass filter followed by a down-sampling operator is the same as the restriction operator in MG methods, whereas an up-sampling operator followed by a lowpass filter is equivalent to the interpolation operator [22].

Given a sequence of grids Ω_l, $1 \leqq l \leqq L$, down-sampling (I_{l+1}^l) and up-sampling (I_l^{l+1}) operators between grids Ω_l and Ω_{l+1}, and appropriate elementary filters H_l defined on Ω_l, the algorithm corresponding to the block diagram given by Fig. 3.1 can be summarized as in Table 3.1.

TABLE 3.1
Computation of $M^{-1}r$.

```
Decomposition:
    $v_L := r$,
    for $l = L-1, \cdots, 1$
    $v_l := I_{l+1}^l H_{l+1} v_{l+1}$,
Scaling:
    for $l = L, \cdots, 1$
    $w_l := v_l d_l^{-1}$
Synthesis:
    $s_1 := w_1$,
    for $l = 2, \cdots, L$
    $s_l := w_l + H_l I_{l-1}^l s_{l-1}$
    $M^{-1}r := s_L$
```

This is the MGMF algorithm implemented in §5.

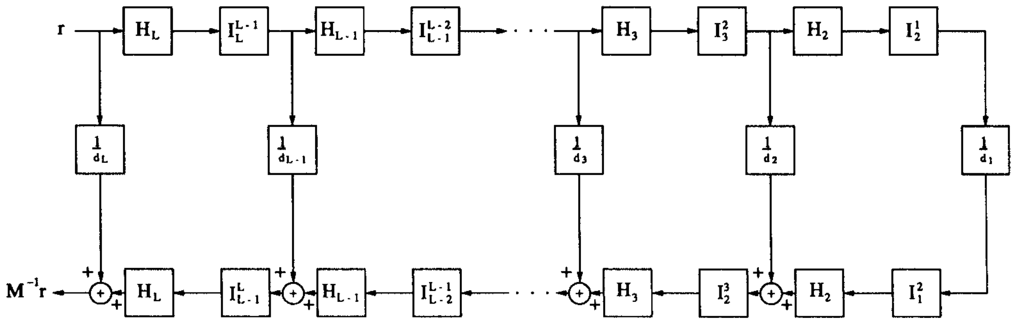


FIG. 3.1. *Block diagram of the modified* MGMF *preconditioner.*

The preconditioning $M^{-1}r$ can be viewed as a degenerate multigrid method, for which we have a sequence of restriction and interpolation operations but where the error smoothing at each grid level is replaced by an appropriate scaling. This observation leads us to generalize the MF preconditioner to the case of nonuniform grids commonly obtained from the finite-element discretization. That is, one can view projection as decomposition and interpolation as synthesis and any multigrid method can be used as an MGMF preconditioner if we replace the potentially more expensive error smoothing by a simple scaling. It is well known that the eigenvalue λ_k in band B_l (see § 2.1) behaves like $O(h_l^{-2})$, where h_l describes approximately the grid spacing for level l [9]. Therefore, a general rule for selecting the scaling constant c_l at grid level l is

$$c_l = O(h_l^{-2}).$$

This generalized version is closely related to the preconditioner by Bramble, Pasciak, and Xu [9]. They derived their preconditioner in the finite-element context discretized with the nested triangular elements. From our filtering framework, the corresponding elementary filter H_L takes the form

$$(3.1)\qquad H_{L,BPX}: \frac{1}{8}\left|\begin{matrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{matrix}\right|,$$

which is different from $H_{L,1}$ given by (2.23). We can derive other filters from (3.1) by applying it more than once. For example, by applying it twice, we get

$$(3.2)\qquad H_{L,TBPX}: \frac{1}{64}\left|\begin{matrix} 0 & 0 & 1 & 2 & 1 \\ 0 & 2 & 6 & 6 & 2 \\ 1 & 6 & 10 & 6 & 1 \\ 2 & 6 & 6 & 2 & 0 \\ 1 & 2 & 1 & 0 & 0 \end{matrix}\right|.$$

In order to eliminate the directional preference, we can apply (3.1) in alternating directions to give a symmetric filter:

$$(3.3)\qquad H_{L,ADBPX}: \frac{1}{64}\left|\begin{matrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 4 & 6 & 4 & 1 \\ 2 & 6 & 8 & 6 & 2 \\ 1 & 4 & 6 & 4 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{matrix}\right|.$$

The MF preconditioner is designed to capture the spectral property (or h-dependency) of a discretized elliptic operator but not the variation of its coefficients. This is also true for the hierarchical basis and BPX preconditioners. In order to take badly scaled variable coefficients into account, we use the MF preconditioner in association with diagonal scaling in our experiments [16]. The diagonal scaling is often used for cases where the diagonal elements of the coefficient matrix A vary for a wide range. Suppose that the coefficient matrix can be written as

$$A = D^{1/2}\tilde{A}D^{1/2},$$

where we choose D to be a diagonal matrix with positive elements in such a way that the diagonal elements of $\tilde{A}$ are of the same order, say, $O(1)$. Then, in order to solve $Au = f$, we can solve an equivalent problem $\tilde{A}\tilde{u} = \tilde{f}$, where $\tilde{u} = D^{1/2}u$ and $\tilde{f} = D^{-1/2}f$, with the MF preconditioner. There exist other ways to incorporate the coefficient information into preconditioners of the multilevel type, say, to use the Gauss-Seidel smoothing suggested by Bank, Dupont, and Yserentant [8].

4. Brief survey of multilevel preconditioners. In this section, we very briefly survey other multilevel preconditioners that have been proposed in the literature and their relationships to one another.

4.1. Multigrid preconditioner (MG). A natural choice for a multilevel preconditioner is to use a fixed number of cycles of a conventional multigrid method. This approach was explored early on in the development of multigrid methods [20], [21]. The basic operations on each grid are interpolation, projection, and smoothing operations, each of which can be easily designed to be highly parallelizable. For example, in the V-cycle strategy, each grid is visited exactly twice in each preconditioning step, once going from fine to coarse grids and once coming back from coarse to fine. However, for highly irregular problems, such as singularities in the solutions due to reentrant corners and highly discontinuous coefficients, it is not clear how to choose the smoothing operations and the performance can deteriorate.

4.2. Hierarchical basis preconditioner (HB). Another preconditioning technique of multilevel type is the hierarchical basis method [8], [29]. The name refers to the space of hierarchical basis functions defined on the grid hierarchy. The usual nodal basis functions are used except that those defined at grid points on a given level which also belong to coarser levels are omitted. Let the hierarchical basis functions be denoted by ψ_j^l, where l denotes the grid level and j the index of the basis function on that level. Then, the action of the inverse of the hierarchical basis preconditioner M on a function v can be written as,

$$M^{-1}v = \sum_l \sum_j (v, \psi_j^l)\psi_j^l,$$

which takes the discretized form SS^Tv_h and can be computed by a V-cycle with the matrix S^T corresponding to a fine-to-coarse grid traversal and S to a coarse-to-fine traversal. On each level, only local operations are performed. In two dimensions, the condition number of the preconditioned system can be shown to grow like $O(\log^2 h^{-1})$, which is very slow. Unfortunately, this nice property is lost in three dimensions, where the growth is $O(h^{-1})$ [26], [29]. However, these theoretical results are proven under much weaker regularity assumptions than for the multigrid methods. Moreover, the computational work per step is $O(h^{-1})$ even for highly nonuniform and refined meshes. For numerical experiments on parallel computers, see [1], [16].

4.3. Method by Bramble-Pasciak-Xu (BPX). Very recently, Bramble-Pasciak-Xu [9], [28] proposed the following preconditioner for second-order elliptic problems in R^d:

$$M^{-1}v = \sum_l h_l^{2-d} \sum_j (v, \phi_j^l)\phi_j^l,$$

where ϕ_j^l are the nodal basis functions and h_l is the measure of the mesh size at grid level l. Since the form of their preconditioner is similar to that for the hierarchical basis preconditioner, the computations can be arranged in a similar way via a V-cycle. They proved that the condition number of the preconditioned operator can be bounded by $O(\log h^{-1})$ for problems with smooth solutions, by $O(\log^2 h^{-1})$ for problems with crack type singularities, and by $O(\log^3 h^{-1})$ for problems with discontinuous coefficients. In 3D, this is a significant improvement over the hierarchical basis preconditioner.

4.4. Algebraic multilevel preconditioners (AMP). Vassilevski [27] proposed a different approach to derive multilevel preconditioners. He used the standard nodal basis functions and a multilevel ordering of the nodes of the discretization, in which nodes at a given level belonging to a coarser grid are ordered after the other nodes. He then considered an approximate block factorization of the stiffness matrix in this ordering, in which the Schur complement at a given grid level is approximated by iteration with the preconditioner of the stiffness matrix recursively defined at the current level. He showed that, with one iteration at each level, the condition number of the preconditioned system can be bounded by $O(\log h^{-1})$. A similar method has been proposed by Kuznetsov [24]. Later, Axelsson-Vassilevski [6], [7] improved this bound to $O(1)$ by carrying out recursively more (Chebyshev) iterations with the preconditioner at each level. Axelsson [4] also showed that the same technique can be applied when hierarchical basis functions are used instead of the nodal basis. Note that when the number of iterations at each level exceeds 1, the grid traversal differs from all the previously mentioned V-cycle based methods. At this time, we have not included non-V-cycle type preconditioners in our numerical comparisons but plan to do so in the future.

4.5. Relationship among multilevel preconditioners. As can be seen from the discussion above, there are similarities among various multilevel preconditioners. Most of the multilevel preconditioners are in the form of a multigrid V-cycle (MG, HB, BPX, and MF, but not AMP). The MF preconditioner is very similar to the BPX method. The MF method allows some flexibility in the choice of filters (basically any multigrid residual averaging operator can be used) and does not depend on the use of a finite-element discretization with nested nodal basis functions. It also allows a single grid (i.e., nonmultigrid) version which may better suit massively parallel architecture computers. On the other hand, the finite-element framework allows an elegant proof of the asymptotic convergence behavior for rather general problems as is done in [9], [28], whereas the filtering framework is rigorously provable for constant coefficient model problems only (although much more detailed information can be obtained for them).

Finally, it is interesting to compare these preconditioners with the conventional multigrid method. Several of the preconditioners have the same form of a conventional multigrid cycle, except that the smoothing operations are omitted. For less regular problems where a good smoothing operator is hard to derive and could be quite expensive, one step of these preconditioners can be substantially less expensive than a corresponding step of the multigrid iteration. In a sense, one can view these preconditioners as efficiently capturing mesh size-dependent part of the ill-conditioning of the elliptic operator and leaves the other sources of ill-conditioning (e.g., discontinuous coefficients) to the conjugate gradient iteration. The combination of multigrid and conjugate gradient holds the promise of being both robust and efficient. However, to get a spectrally equivalent preconditioner, it seems that one must go beyond the V-cycle and perform more iterations on each grid as in the AMP method.

5. Numerical experiments. In this section, we present numerical results for two- and three-dimensional test problems to compare the convergence behavior and the amount of work needed for various preconditioners. The preconditioners implemented are:

HB: hierarchical basis preconditioner using linear elements for two-dimensional and trilinear elements for three-dimensional problems,

MG(i, i): multigrid preconditioner with one V-cycle, where i is the number of pre- and post-smoothings,

BPX1: the BPX preconditioner for two-dimensional problems (H_L given by (3.1)),

BPX2: a modified version of BPX preconditioner by filtering twice for two-dimensional problems (H_L given by (3.2)),

BPX3: another modified version of BPX preconditioner by filtering twice but using linear elements of different orientations for two-dimensional problems (H_L given by (3.3)),

MGMF1: the MGMF preconditioner with the 9-point (2.23) or 27-point filter for two- and three-dimensional problems, respectively,

MGMF2: a modified version of MGMF preconditioner in which the 9-point (or 27-point) filter is applied twice,

MGMF3: another modified version of MGMF preconditioner in which the 9-point (or 27-point) filter is applied once at the finest grid level (to have a smaller amount of work compared to MGMF2) and twice at other grid levels (to achieve a faster convergence rate compared to MGMF1),

RIC: the relaxed incomplete Cholesky preconditioner [5] is included for the purpose of comparison. For the relaxation factor, we use the optimal value $\omega = 1 - 8\sin^2(\pi h/2)$ from [11]. The number of iterations required for RIC can be bounded by $O(n^{1/2})$.

The preconditioning operation counts for each method, for two- and three-dimensional problems are given in Tables 5.1 and 5.2, respectively. These operation counts include addition, multiplication, and division (each is counted as one operation), but exclude overhead such as condition checking and data copying. The non-preconditioning operation counts required per PCG step for two-dimensional problems are $21N$, which include $6N$ for three inner products (one more inner product than the basic CG step, since we use the relative residual norm for convergence check), $6N$ for three SAXPY

TABLE 5.1
Work per iteration for preconditioners (2D).

Preconditioner	Operation count per iteration
RIC	9 N
HB	7 N
MG(1.1)	38 N
BPX1	8 N
BPX2	26 N
BPX3	26 N
MGMF1	9 N
MGMF2	27 N
MGMF3	15 N

TABLE 5.2
Work per iteration for preconditioners (3D).

Preconditioner	Operation count per iteration
RIC	13 N
HB	8 N
MGMF1 (BPX1)	9 N
MGMF2 (BPX2)	32 N
MGMF3	12 N

operations, and $9N$ for one matrix vector product. Similarly, the non-preconditioning operation counts per PCG step for three-dimensional problems are $25N$.

From Table 5.1, we observe that the operation counts per iteration for BPX1 and MGMF1 are much less than that of the MG(1, 1) preconditioners, because the former preconditioners do not need smoothing, which is expensive. In general, for two-dimensional problems, MG(i, i) preconditioner takes $(38 + 32 \times (i - 1))N$ operations. For example, MG(3, 3) preconditioning requires $102N$ operations. Also note that the application of filtering twice requires about three times the work of filtering once. This is because by filtering twice the filter stencil is extended from 9-point to 25-point (about three times as many points).

For three-dimensional problems, the operation count for BPX1 (BPX2) preconditioning using trilinear elements is the same as for the MGMF1 (MGMF2) preconditioning as shown in Table 5.2. The MG preconditioner has not yet been implemented for three-dimensional problems.

For all test problems, we use the standard 5- (or 7-) point stencil on a square (or cubic) uniform mesh with $h = n^{-1}$ and $N = (n - 1)^2$ (or $N = (n - 1)^3$), zero boundary conditions and zero initial guesses. Experimental results are given for different values of h and the stopping criterion is $\|r^k\| / \|r^0\| \leqq 10^{-6}$. Diagonal scaling is always used except for RIC. The six test problems are:

(1) the two-dimensional model problem with solution $u = x(x - 1)y(y - 1)e^{xy}$,

$$\Delta u = f, \qquad \Omega = (0, 1)^2, \tag{5.1}$$

(2) a two-dimensional variable coefficient problem with solution $u = xe^{xy} \sin \pi x \times \sin \pi y$,

$$\frac{\partial}{\partial x}\left(e^{-xy}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(e^{xy}\frac{\partial u}{\partial y}\right) = f, \qquad \Omega = (0, 1)^2, \tag{5.2}$$

(3) a two-dimensional problem with discontinuous coefficients with $f = 2x(1 - x) + 2y(1 - y)$,

$$\frac{\partial}{\partial x}\left(\rho(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\rho(x,y)\frac{\partial u}{\partial y}\right) = f, \qquad \Omega = (0, 1)^2, \tag{5.3}$$

where

$$\rho(x,y) = \begin{cases} 10^4 & x > 0.5 \ y \leqq 0.5, \\ 10^{-4} & x \leqq 0.5 \ y > 0.5, \\ 1 & \text{elsewhere.} \end{cases}$$

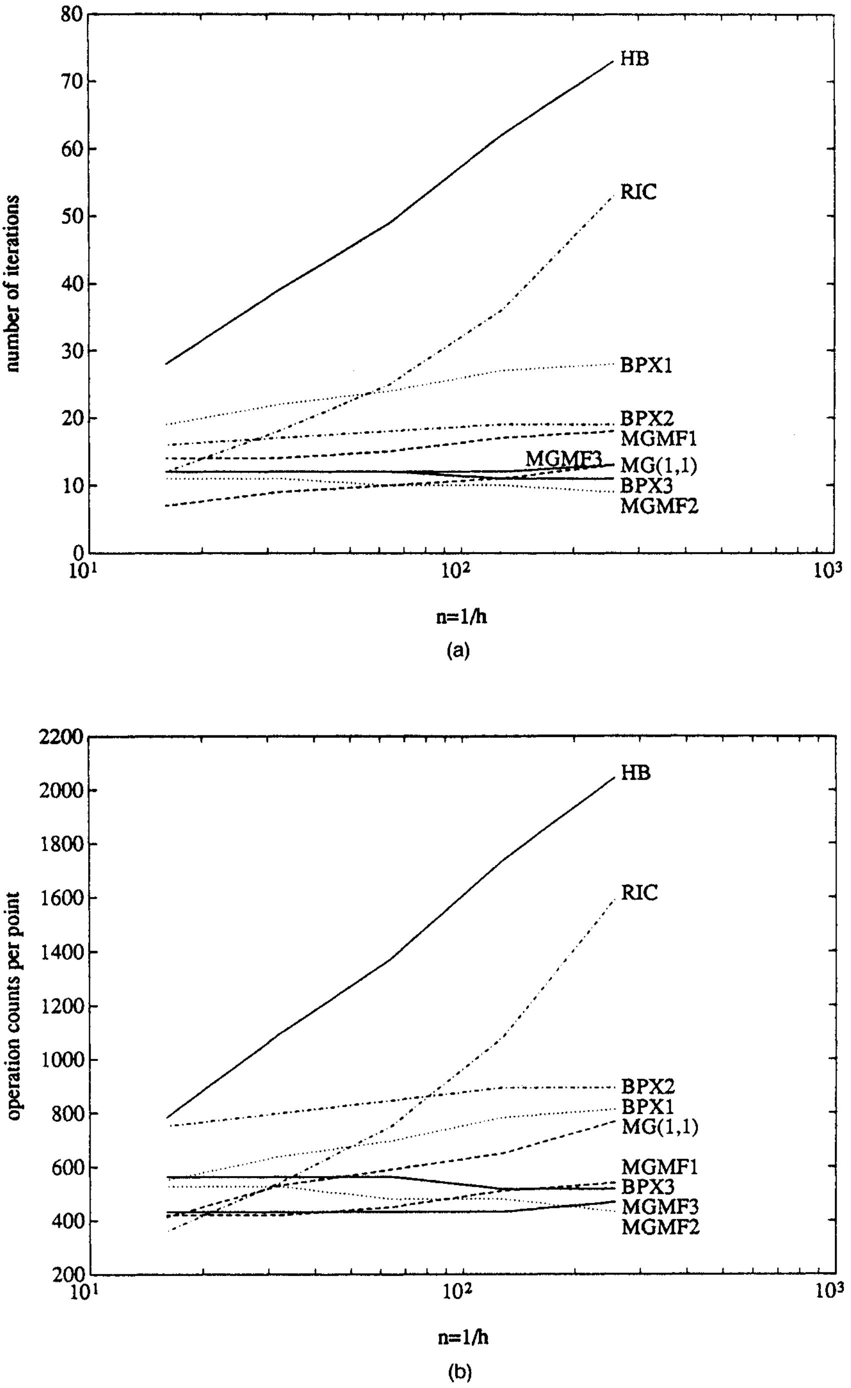
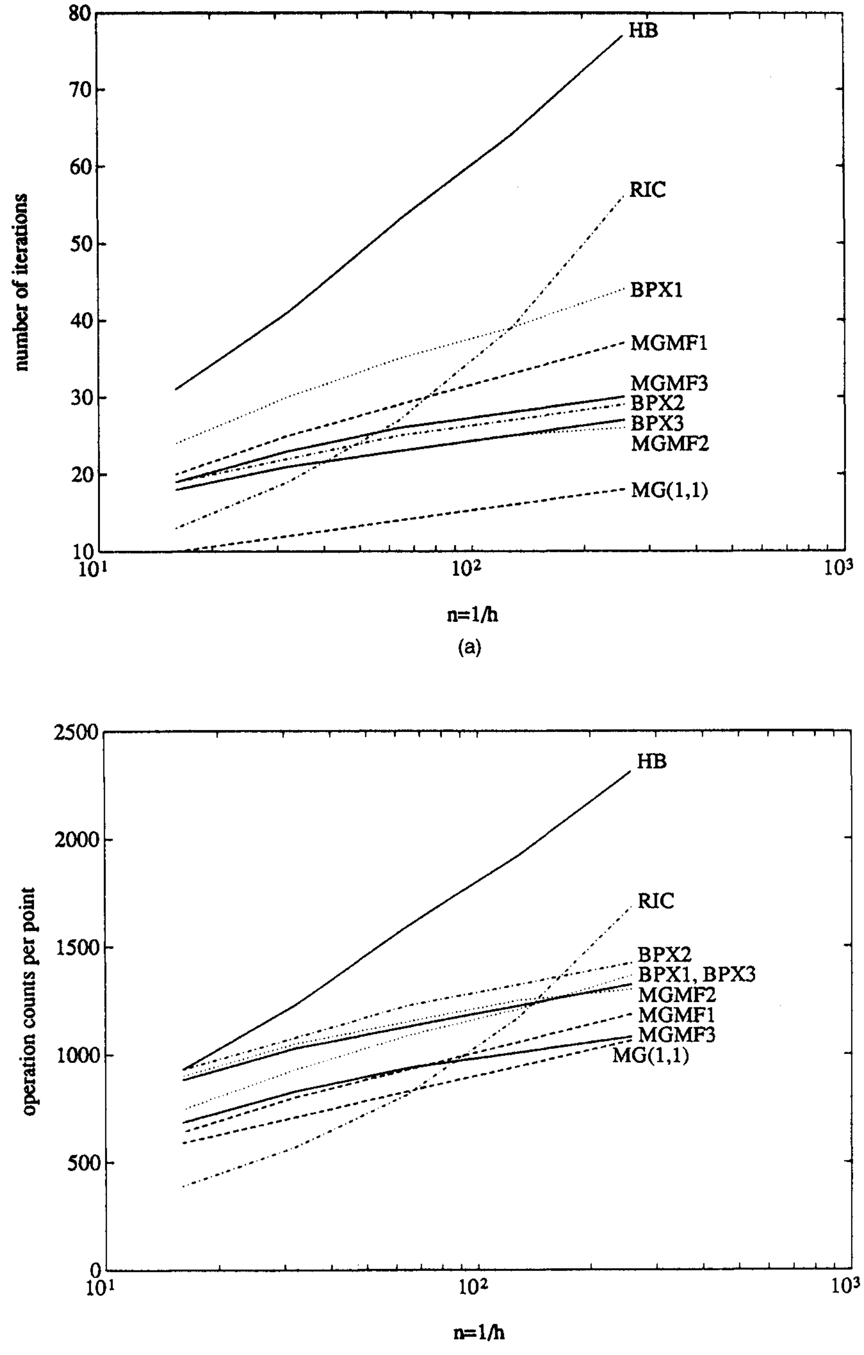


FIG. 5.1. (a) *Iteration and* (b) *operation counts for Test Problem* 1.

(4) the three-dimensional model problem with solution

$$u = x(1-x)y(1-y)z(1-z)e^{xyz},$$

$$\Delta u = f, \qquad \Omega = (0,1)^3, \tag{5.4}$$



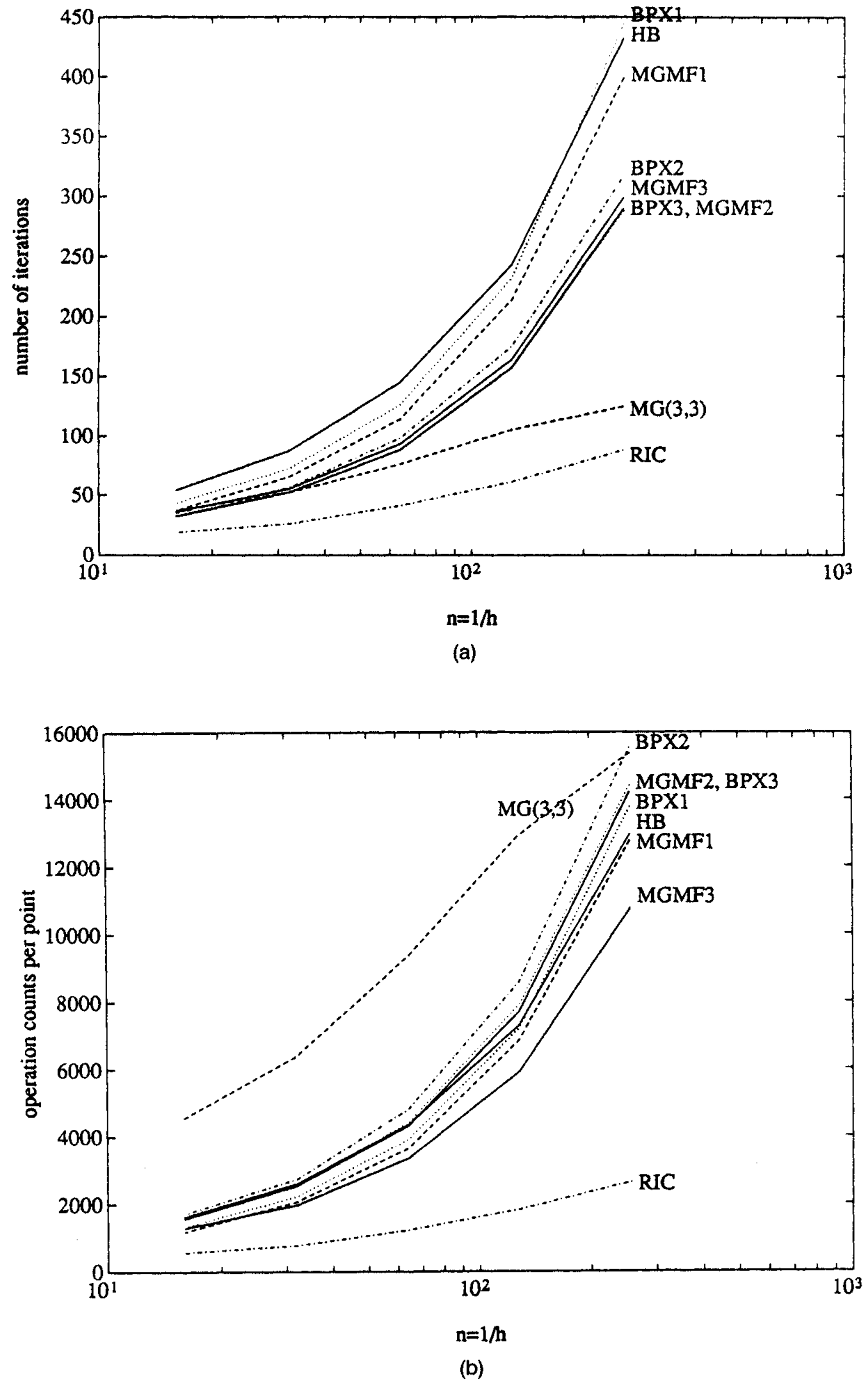
(a)

(b)

FIG. 5.2. (a) *Iteration and* (b) *operation counts for Test Problem* 2.

(5) a three-dimensional variable coefficient problem with solution $u = e^{xyz} \sin \pi x \times \sin \pi y \sin \pi z$,

$$(5.5) \qquad \frac{\partial}{\partial x}\left(e^{-xyz}\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(e^{xyz}\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(e^{-xyz}\frac{\partial u}{\partial z}\right)=f, \qquad \Omega=(0,1)^3,$$



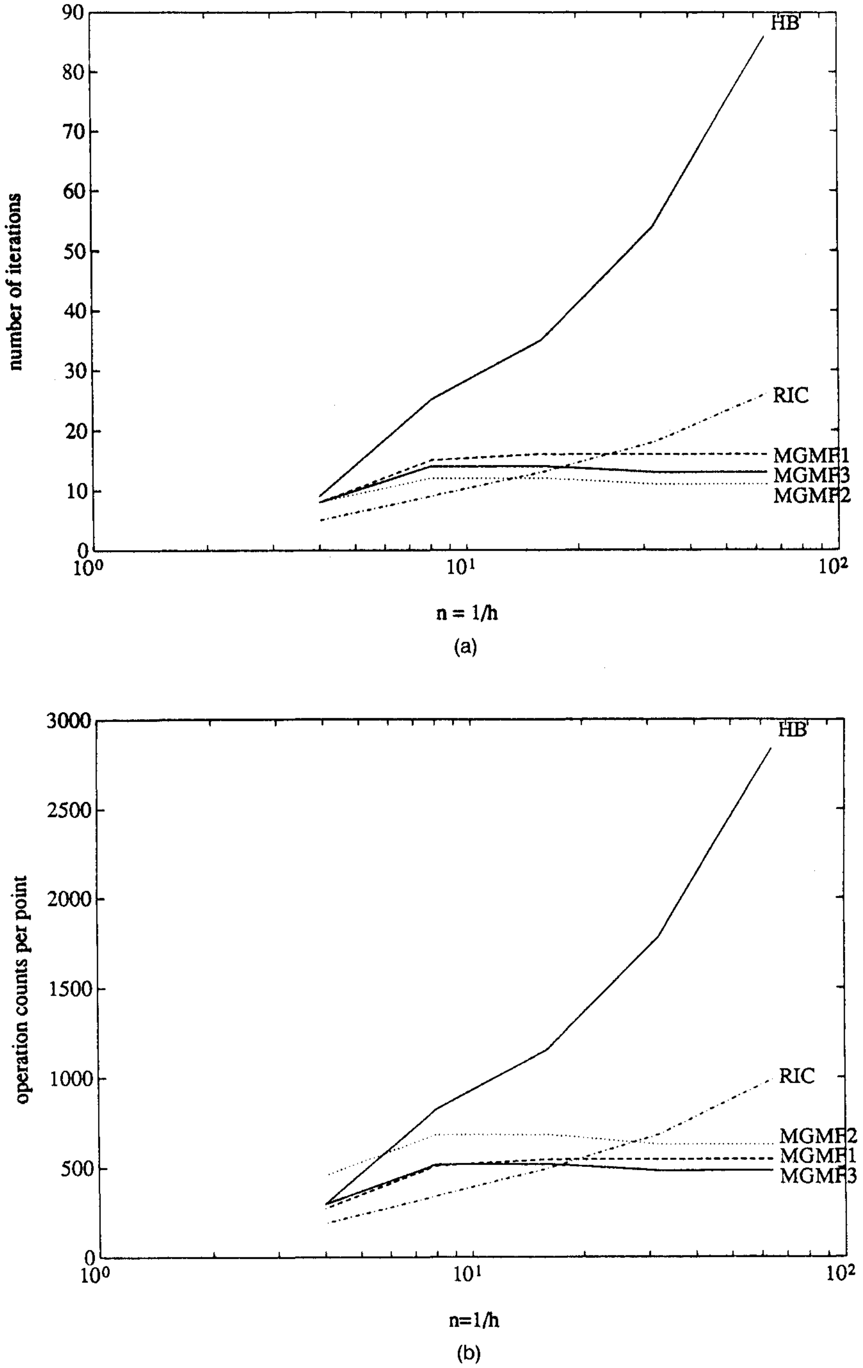
(a)

(b)

FIG. 5.3. (a) *Iteration and* (b) *operation counts for Test Problem* 3.

(6) a three-dimensional problem with discontinuous coefficients with $f = 2x(1-x) + 2y(1-y) + 2z(1-z)$,

$$(5.6) \quad \frac{\partial}{\partial x}\left(\rho(x,y,z)\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\rho(x,y,z)\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(\rho(x,y,z)\frac{\partial u}{\partial z}\right)=f, \qquad \Omega=(0,1)^3,$$



(a)

(b)

FIG. 5.4. (a) *Iteration and* (b) *operation counts for Test Problem* 4.

where

$$\rho(x,y,z)=\begin{cases}10^{-4} & x>0.5 \text{ with } y\leqq 0.5, z\leqq 0.5 \text{ or } y>0.5, z>0.5,\\ 10^{4} & x\leqq 0.5 \text{ with } y>0.5, z\leqq 0.5 \text{ or } y\leqq 0.5, z>0.5,\\ 1 & \text{elsewhere.}\end{cases}$$

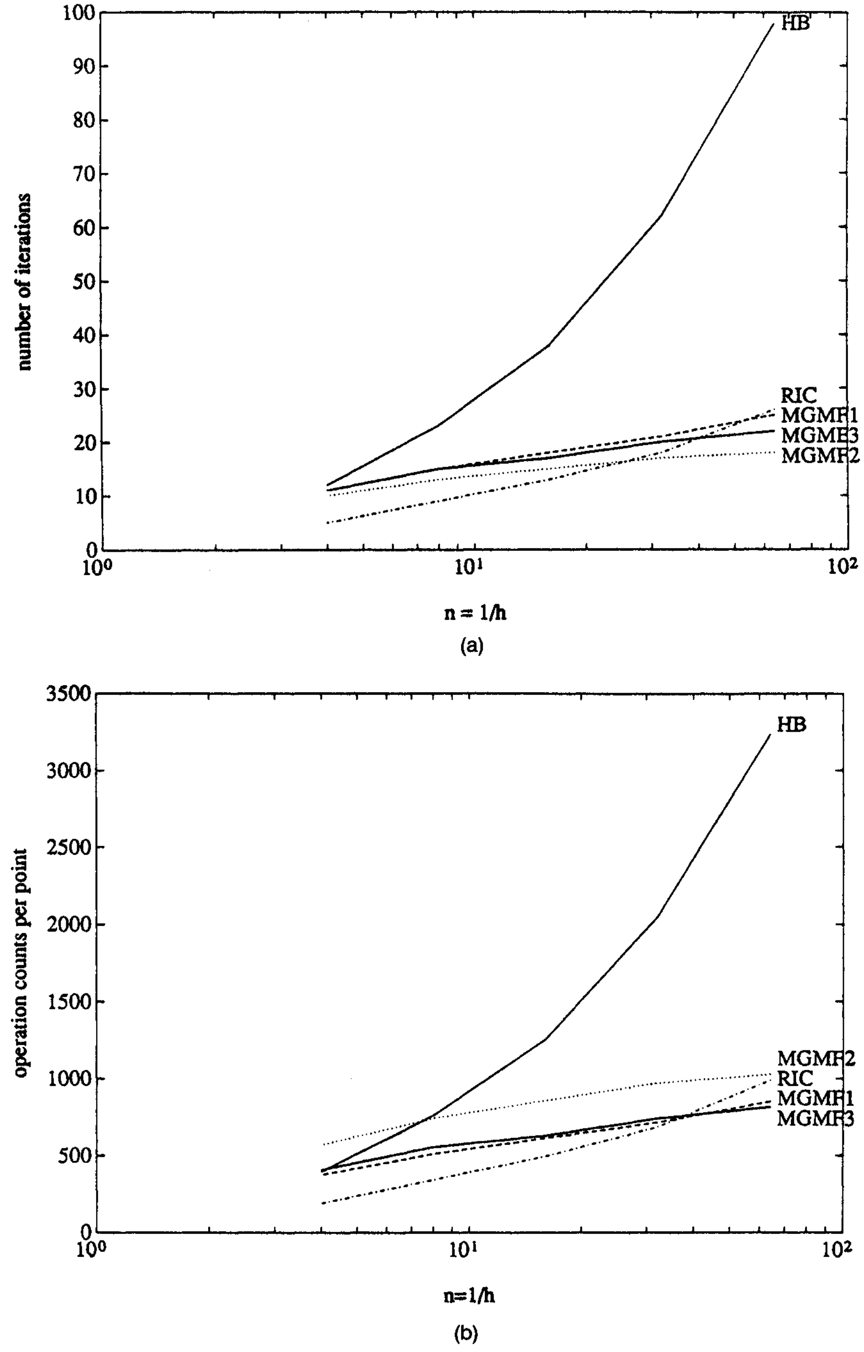
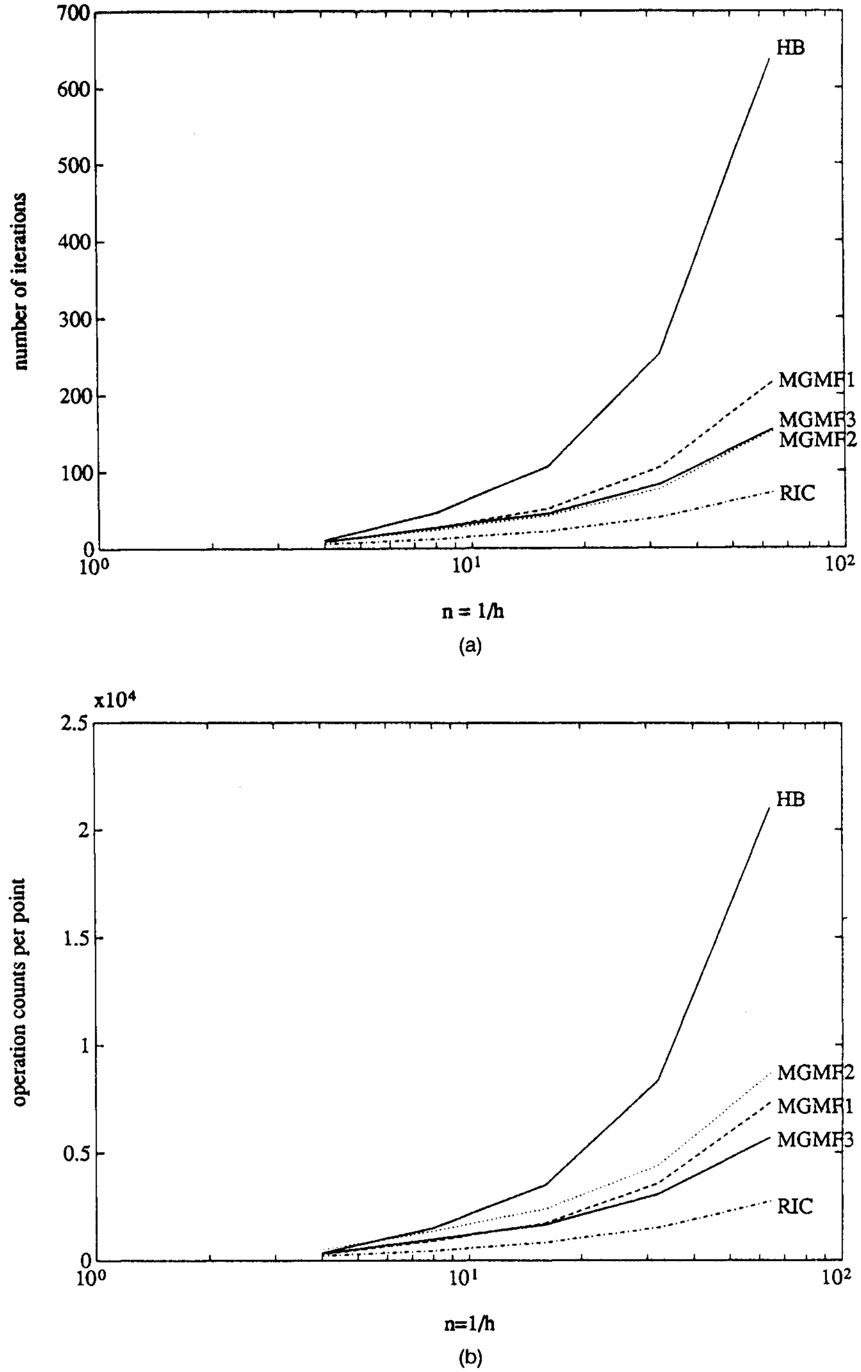


FIG. 5.5. (a) *Iteration and* (b) *operation counts for Test Problem* 5.

The number of iterations and operation counts per grid point are plotted in Figs. 5.1–5.6 (a) and (b), respectively. We can make the following observations from these figures.

(1) The BPX and MGMF preconditioners have better convergence behavior than the HB preconditioner, especially for three-dimensional problems. The HB method is competitive with the other multilevel methods only for the discontinuous coefficient problem in two dimensions.



(a)

(b)

FIG. 5.6. (a) *Iteration and* (b) *operation counts for Test Problem* 6.

(2) The $O(\log^{\alpha} n)$ convergence rate for all the multilevel methods is evident, except for the three-dimensional HB method. The three-dimensional HB method behaves like $O(h^{-0.59})$ and $O(h^{-0.70})$ for problems (5.4) and (5.5), which are close to the predicted theoretical result $O(h^{-0.5})$. However, for the discontinuous coefficient problem (5.6), it converges more slowly, like $O(h^{-1.26})$.

(3) In general, the MGMF methods perform slightly better than the corresponding BPX methods. Recall that the only difference between the two methods is the choice of the elementary filters.

(4) Filtering twice (BPX2, BPX3, and MGMF2) does improve the convergence rates for the model Poisson problem in either two or three dimensions (the MGMF2 and BPX3 preconditioners appear to be spectrally equivalent.) For variable and discontinuous coefficient problems, filtering twice does not seem to improve the convergence rates enough to compensate for the extra work involved.

(5) The MGMF3 method is designed to incorporate the desired features of MGMF1 and MGMF2, i.e., the good convergence property due to filtering twice and the smaller amount of work due to filtering once at the finest grid level. It turns out that it works very well. MGMF3 behaves better than MGMF1 but worse than MGMF2 in the number of iterations required. However, in terms of amount of work, MGMF3 is better than MGMF1 and MGMF2.

(6) For small n (<100), the RIC method is competitive with all the multilevel methods. In fact, for the discontinuous coefficient problems, none of the multilevel preconditioners gives a better convergence rate than the RIC preconditioner. It appears that the RIC preconditioner captures the variation of the coefficients especially well. Its performance deteriorates as n gets large, as predicted by its inferior asymptotic convergence rate.

(7) The MG preconditioner is among the most efficient methods for problems with smooth coefficients. However, it has some difficulties with problems with discontinuous coefficients. In fact, for Problem (5.3), MG(1, 1) requires too many iterations to fit on the plot. Instead we show the results for the MG(3, 3) method, which converges in a reasonable number of iterations but still requires the most work of all the methods. We have noticed that the performance of the multigrid methods are somewhat sensitive to the initial guess. In experiments with random initial guesses, we have observed that the performance of the multigrid methods is significantly improved. This may be due to the extra smoothing operations in the multigrid methods which are more adept at annihilating the high frequency errors inherent in the random initial guess.

6. Conclusions.

The experimental results show that the class of multilevel filtering preconditioners compares favorably with the hierarchical basis and the RIC preconditioners, at least for problems with smooth coefficients and quasi-uniform grids such as used in our experiments. For these types of problems, the multilevel filtering and the BPX methods behave quite similarly to the multigrid preconditioner. What these new methods offer is the saving of smoothing operations which are difficult to make effective for irregular problems, while preserving the nice asymptotic convergence rates of multigrid preconditioners. The relative performance of the hierarchical basis method should improve for irregular problems on highly nonuniform and refined meshes. Even though the RIC preconditioner shows better convergence rates for strongly discontinuous coefficient problems, it has a low degree of parallelism. The multilevel filtering preconditioners are also similar to the BPX method. What the filtering framework provides is the flexibility of filter design, which can lead to more efficient methods.

Acknowledgment. The authors thank the referees for their help in improving the presentation of this paper.

REFERENCES

[1] L. M. ADAMS AND M. E. G. ONG, *A comparison of preconditioners for* GMRES *on parallel computers*, in Parallel Computations and Their Impact on Mechanics, A. K. Noor, ed., pp. 171–186, The American Society of Mechanical Engineers, New York, 1987.

[2] S. F. ASHBY, *Polynomial preconditioning for conjugate gradient methods*, Ph.D. thesis, 1987, Department of Computer Science, University of Illinois, Urbana, IL 61801.

[3] O. AXELSSON, *A generalized* SSOR *method*, BIT, 13 (1972), pp. 443–467.

[4] ———, *An algebraic framework for multilevel methods*, Report 8820, Department of Mathematics, Catholic University, The Netherlands, 1988.

[5] O. AXELSSON AND G. LINDSKOG, *On the eigenvalue distribution of a class of preconditioning methods*, Numer. Math., 48 (1986), pp. 479–498.

[6] O. AXELSSON AND P. VASSILEVSKI, *Algebraic multilevel preconditioning methods*, I, Report 8811, Department of Mathematics, Catholic University, The Netherlands, 1988.

[7] ———, *Algebraic multilevel preconditioning methods*, II, Report 1988-15, Institute for Scientific Computation, University of Wyoming, Laramie, WY, 1988.

[8] R. E. BANK, T. F. DUPONT, AND H. YSERENTANT, *The hierarchical basis multigrid method*, Numer. Math., 52 (1988), pp. 427–458.

[9] J. H. BRAMBLE, J. E. PASCIAK, AND J. XU, *Parallel multilevel preconditioners*, Math. Comp., to appear.

[10] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[11] T. F. CHAN, *Fourier analysis of Relaxed Incomplete Cholesky factorization preconditioners*, CAM report 88-34, University of California, Los Angeles, CA, 1988.

[12] T. F. CHAN, C.-C. J. KUO, AND C. TONG, *Parallel elliptic preconditioners: Fourier analysis and performance on the Connection Machine*, Comput. Phys. Comm., 53 (1989), pp. 237–252.

[13] R. E. CROCHIERE AND L. R. RABINER, *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[14] G. DAHLQUIST AND A. BJÖRCK, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[15] T. DUPONT, R. P. KENDALL, AND H. H. RACHFORD, JR., *An approximate factorization procedure for solving self-adjoint difference equations*, SIAM J. Numer. Anal., 5 (1968), pp. 559–573.

[16] A. GREENBAUM, C. LI, AND H. Z. CHAO, *Parallelizing preconditioned conjugate gradient algorithms*, Comput. Phys. Comm. 53 (1989), pp. 295–309.

[17] W. HACKBUSCH, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, New York, 1985.

[18] O. HERRMANN, *On the approximation problem in nonrecursive digital filter design*, IEEE Trans. Circuit Theory, CT-18 (1971), pp. 411–413.

[19] O. G. JOHNSON, C. A. MICCHELLI, AND G. PAUL, *Polynomial preconditioning for conjugate gradient calculations*, SIAM J. Numer. Anal., 20 (1983), pp. 362–376.

[20] R. KETTLER, *Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods*, in Multigrid Methods, W. Hackbusch and U. Trottenberg, eds., pp. 502–534, Springer-Verlag, Berlin, New York, 1982.

[21] 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.

[22] C.-C. J. KUO AND B. C. LEVY, *Two-color Fourier analysis of the multigrid method with red/black Gauss-Seidel smoothing*, Appl. Math. Comput., 29 (1989), pp. 69–87.

[23] C.-C. J. KUO AND T. F. CHAN, *Two-color Fourier analysis of iterative algorithms for elliptic problems with red/black ordering*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 000–000.

[24] Y. A. KUZNETSOV, *Multigrid domain decomposition methods for elliptic problems*, in Proceedings VIII International Conference on Computational Methods for Applied Science and Engineering, Vol. 2, pp. 605–616, 1987.

[25] 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., 31 (1977), pp. 148–162.

[26] M. E. G. ONG, *The* 3D *linear hierarchical basis preconditioner and its shared memory parallel implementation*, Preprint, Department of Applied Mathematics, University of Washington, Seattle, WA, 1988.

[27] P. VASSILEVSKI, *Iterative methods for solving finite element equations based on multilevel splitting of the matrix*, Preprint, Bulgarian Academy of Science, Sofia, Bulgaria, 1987.

[28] J. XU, *Theory of multilevel methods*, Ph.D. thesis, Department of Mathematics, Cornell University, Ithaca, NY, 14853, 1989.

[29] H. YSERENTANT, *On the multi-level splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.