

**USC-SIPI REPORT #148**

*Multilevel Filtering Elliptic Preconditioners*

*By*

*C.-C. Jay Kuo, Tony F. Chan and Charles Tong*

*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.**

**August 1989**

# Multilevel Filtering Elliptic Preconditioners \*

C.-C. Jay Kuo†

Tony F. Chan ††

Charles Tong†††

August, 1989

## Abstract

We present a class of preconditioners for elliptic problems built on ideas borrowed from the digital filtering theory and implemented on a multilevel grid structure. They are designed to be both rapidly convergent and highly parallelizable. The digital filtering viewpoint allows us to use 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.

**Abbreviated title:** Multilevel Filtering Preconditioners

**AMS(MOS) subject classification:** 65N20, 65F10

---

\* This paper is based on a presentation at the SIAM Symposium on Sparse Matrices, May 22-24, 1989, Salishan Resort, Gleneden Beach, Oregon. The first author was supported by USC Faculty Research and Innovation Fund. The second and third authors were supported in part by the National Science Foundation under contracts NSF-DMS87-14612 and BBS 87 14206, and by the Army Research Office under contract DAAL03-88-K-0085 and part of this work was performed while visiting the Research Institute for Advanced Computer Science, NASA Ames.

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

†† Department of Mathematics, University of California, Los Angeles, CA 90024.

††† Department of Computer Science, University of California, Los Angeles, CA 90024.

## 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 parallelizable and rapidly convergent.

The idea of preconditioning is a simple one but 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 of producing a spectrum for the preconditioned system  $M^{-1}A$  clustering around 1 and having a small condition number) and for which the matrix vector product  $M^{-1}v$  for a given vector  $v$  can be computed efficiently. With such a preconditioner, one then solves in principle the preconditioned system  $M^{-1}Ax=M^{-1}b$  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 SSOR preconditioner [3], the incomplete factorization preconditioners (ILU [25] and MILU [15]) and polynomial preconditioners [2],[19]. 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. Unfortunately, previous works [12],[16] have shown that for many of the classical preconditioners, there is a fundamental tradeoff in the ease of parallelization and the rate of convergence. A principal obstacle to parallelization is the sequential manner in which many preconditioners use in traversing the computational grid --- the data dependence implicitly prescribed by the method fundamentally limits the amount of parallelism available. Re-ordering the grid traversal (e.g. from natural to red-black ordering) or inventing new methods (e.g. polynomial preconditioners) to improve the parallelization alone invariably 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 effective global coupling that 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 which 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.) Since we are using the multilevel iteration within an outer conjugate gradient iteration, we have more flexibility in terms of the choice of inter- and intra-grid level operators, such as interpolation, projection and smoothing. 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. A major difference in the approach taken by Bramble, et al. and this paper from multigrid methods is 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-Vassilevski [6],[7], Kuznetsov [24] and Axelsson [4].

The outline of the paper is as follows. In Section 2, we describe our framework for deriving multilevel filtering preconditioners for a model problem. The basic framework is then extended to more general problems in Section 3. In Section 4, we briefly survey several other preconditioners of the multilevel type. Numerical results for (model, variable coefficient and discontinuous coefficient) problems in 2D and 3D are presented in Section 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 Section 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  $\Omega_h$ ,

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

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

$$A u = f,$$

where  $A$ ,  $u$  and  $f$  correspond to the discrete Laplacian, solution and forcing functions respectively. 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

$$A = W^T \Lambda_A W, \quad (2.2)$$

where  $\Lambda_A$  is a diagonal matrix

$$\text{diag}(\lambda_1, \dots, \lambda_k, \dots, \lambda_{N-1}), \quad \lambda_k = 1 - \cos(k\pi h),$$

and  $W$  is an  $(N-1)^2$  square matrix whose  $k$ th row is

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

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

$$\hat{u}_k = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{n=1}^{N-1} u_n \sin(k\pi nh), \quad \hat{f}_k = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{n=1}^{N-1} f_n \sin(k\pi nh), \quad n = 0, 1, 2, \dots, N.$$

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

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

where

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

is known as the spectrum of the discrete Laplacian.

In order to invert  $A$ , we can make use of (2.2) and obtain

$$A^{-1} = W^T \Lambda_A^{-1} W. \quad (2.5)$$

The above procedure also serves as the general framework for fast Poisson solvers in higher dimensional cases. 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 which are extensible to more general problems and then use them as preconditioners. The fundamental idea is to avoid the use of FFT's but instead use a sequence of filtering operations to approximately achieve the desired spectral decomposition. This explains the motivation and the name of the multilevel filtering (MF) preconditioner proposed in this paper.

## 2.2 Piecewise constant approximation of the spectrum

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

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

where

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

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

$$\Lambda_P = \text{diag}(\hat{P}(1), \hat{P}(2), \dots, \hat{P}(N-1)), \quad (2.6)$$

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

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

where

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

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

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

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  $\lambda$  within band  $B_l$  to be the constant  $c_l$ . For the following discussion, let us consider the choice,

$$c_l = 4^{-(L-l)}. \quad (2.7)$$

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

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

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

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

and

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

Note that the last inequalities in above equations hold independent of  $L$ , or equivalently, the grid size  $h$ . Thus, the condition number  $\kappa$  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 Figure 2.1 for  $N = h^{-1} = 256$  with  $c_l$  defined in (2.7).

### 2.3 Decomposition and synthesis based on filtering

The preconditioning procedure



$$P^{-1}r = W^T \Lambda_P^{-1} W r , \quad (2.8)$$

consists of three building blocks: decomposition ( $W$ ), scaling ( $\Lambda_P^{-1}$ ) and synthesis ( $W^T$ ).

Let us rewrite (2.8) as

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

where  $W_l$ ,  $1 \leq l \leq L$ , 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. Consequently, we have

$$W_l^T W_l w_k = \begin{cases} w_k , & k \in B_l \\ 0 , & \text{otherwise} \end{cases} , \quad (2.9)$$

where  $w_k$  is defined in (2.3). From (2.9), we see clearly that  $W_l$  functions as an ideal bandpass filter for band  $B_l$ . Although it is possible to implement the ideal bandpass characteristics (2.9) with FFT or bandpass filters of size  $O(N)$ , the corresponding implementation is either not easily generalizable or simply too expensive ( $O(N^2)$  complexity). Instead, we want to approximate the ideal bandpass filter  $W_l$  with nonideal bandpass filters  $F_l$

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

in such a way that  $F_l$  can be implemented cost-effectively for general problems. Note that  $F_l$  is in general a dense matrix of size  $(N-1)^2$ . The resulting preconditioner is in form

$$M^{-1}r = \left( \sum_{l=1}^L \frac{1}{c_l} F_l^T F_l \right) r . \quad (2.10)$$

Before the detailed discussion of implementing  $F_l$ ,  $1 \leq l \leq L$ , with digital filters, it is worthwhile to summarize the similarities and differences between the fast Poisson solver (2.5) and the MF preconditioning (2.10). They are both based on the spectral decomposition idea. The fast Poisson solver decomposes a function into its Fourier components through FFT while the MF preconditioner decomposes it approximately 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, FFT is primarily applicable to constant coefficient problems with regular grids and domains. Besides, for the fast Poisson solver we usually require the detailed knowledge of the spectrum. But for the MF preconditioner we only have to estimate how the spectrum varies from one band to another.

In the context of multirate signal processing[13], the separation of a function into several components each of which is confined in 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 ( $F_l$ ,  $1 \leq l \leq L$ ) and synthesizer ( $F^T$ ,  $1 \leq l \leq L$ ), a simple design illustrated by the block diagram of Figure 2.2 will be sufficient for our purpose. This design is based on the cascade of a sequence of elementary filters  $H_L, H_{L-1}, \dots, H_2$ , where the function of  $H_l$  is to preserve Fourier components contained in bands  $B_1, \dots, B_{l-1}$  and to eliminate Fourier components contained in band  $B_l$ . From Figure 2.2, we see that  $F_l$  are related to elementary filters  $H_l$  via

$$F_L = I - H_L, \quad (2.11a)$$

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

$$F_1 = \prod_{p=2}^L H_p. \quad (2.11c)$$

It is not hard to check that we can obtain components in bands  $B_L$  and  $B_1$  from the outputs of  $F_L$  and  $F_1$ . The product of a sequence of elementary filters appearing in (2.11b) leads to the band  $B_1 \cup \dots \cup B_l$ , from which the band  $B_l$  can be separated by using the filter  $I - H_l$ . Thus, the problem of designing the filter bank  $F_l$ ,  $1 \leq l \leq L$ , is transformed into an equivalent one of designing elementary filters  $H_l$ ,  $1 \leq l \leq L$ .

## 2.4 Design of elementary filters

Consider the design of the elementary filter  $H_L$  appearing at the first stage. It is desired that the filter  $H_L = W^T \Lambda_{H_L} W$  has the following ideal lowpass characteristic,

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

where  $\hat{H}_L(k)$  is the  $k$ th element of the diagonal matrix  $\Lambda_{H_L}$  so that we are able to separate the function  $r$  into two bands: the high wavenumber band  $(I - H_L)r$  and the low wavenumber band  $H_L r$ .

We will approximate the above ideal filter with a nonideal lowpass filter of size  $(2J+1)$ ,

$$H_{L,J} = a_0 + \sum_{j=1}^J a_j (E^j + E^{-j}). \quad (2.12)$$

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,

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

Consequently, the filter  $H_{L,J}$  corresponds to a circulant matrix. The above odd-periodic assumption is only used for analyzing and designing filters. The actual implementation of the MGMF algorithm (see Section 3.5) does not rely on this assumption.

There are numerous ways to determine the coefficients  $a_0$  and  $a_j$ 's depending what approximation criteria 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:

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

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

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

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

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 Figure 2.3 with  $N = 2^8 = 256$ . The larger  $J$  becomes, the better the approximation is.

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

Table 2.1: Coefficients of a class of nonideal lowpass filters

As illustrated in Figure 2.2, 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.

$$H_{L-1,J} = a_0 + \sum_{j=1}^J a_j (E^{2j} + E^{-2j}). \quad (2.13)$$

Comparing (2.12) and (2.13), 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 (2.13), we see that the filter  $H_{L-1,J}$  has the spectrum

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

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  $1 \leq k < 2^{L-1}$ ,  $\hat{H}_{L-1}$  behaves like a lowpass filter which preserves components in the region  $1 \leq k < 2^{L-2}$  and filters out components in the region  $2^{L-2} \leq k < 2^{L-1}$ . 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 define the general elementary filter  $H_l$  on a uniform infinite grid as

$$H_{l,J} = a_0 + \sum_{j=1}^J a_j (E^{2^{L-l}j} + E^{-2^{L-l}j}), \quad 2 \leq l \leq L, \quad (2.14)$$

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

$$\hat{H}_{l,J}(k) = a_0 + \sum_{j=1}^J a_j \cos(k\pi j 2^{L-l}h), \quad 2 \leq l \leq L. \quad (2.15)$$

It is clear from (2.14) that the elementary filter  $H_{l,J}$  is symmetric. So is the bandpass filter  $F_l$ . The construction of the bandpass filter  $F_l$  with elementary filters  $H_l$  is illustrated in Figure 2.4, where  $l=L-1$  and  $J=1$  are chosen as example. We know from (2.11) that  $F_{L-1} = (I - H_{L-1}) H_L$ .

The above discussion is based on the odd-periodic property of the sequence  $v_n$ . However, this may not be easily implementable for general multidimensional problems. 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  $H_{L,J}$  (2.12) with  $J=1$  twice. This is equivalent 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 Section 3.1.

## 2.5 Fourier Analysis and higher dimensional cases

Since the preconditioner  $M^{-1} = \sum_l c_l^{-1} F_l^T F_l$  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 (2.11), we know the following spectral relationship

$$\begin{aligned} \hat{F}_{L,J}(k) &= I - \hat{H}_{L,J}(k), \\ \hat{F}_{l,J}(k) &= (I - \hat{H}_{l,J}(k)) \left[ \prod_{p=l+1}^L \hat{H}_{p,J}(k) \right], \quad 2 \leq l \leq L-1, \\ \hat{F}_{1,J}(k) &= \prod_{p=2}^L \hat{H}_{p,J}(k), \end{aligned} \tag{2.16}$$

where  $\hat{H}_{l,J}(k)$ ,  $1 \leq l \leq L$ , are given by (2.15). Using (2.4), (2.7) and (2.16), we can determine the spectrum of  $M^{-1}A$ ,

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

The spectrum  $\lambda(M^{-1}A)$  is plotted as function of  $k$  with  $J = 1, 3, 5$  and  $h^{-1} = 256$  in Figure 2.5. We should compare these spectra with that in Figure 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 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 the two-dimensional elementary filter by the tensor product of one-dimensional elementary filters along the  $x$ - and  $y$ -directions,

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

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

$$H_{L,1} : \frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}. \quad (2.17)$$

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 function of the grid size  $h$  in Figures 2.6 (a) and (b). These figures show that  $M$  and  $A$  are spectrally equivalent.

### 3. Multilevel filtering preconditioners: generalizations

In Section 2, we discuss 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 PDE problems.

#### 3.1 Multigrid multilevel filtering (MGMF)

The filtering operation described above is performed at every grid points at all levels  $2 \leq l \leq L$ . Since there are  $O(\log N)$  levels and  $O(JN)$  operations per level, where  $N$  and  $J$  denote the order of unknowns and the filter size respectively, 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 Section 2. That is, we construct a sequence of grids  $\Omega_l$  of sizes  $h_l = O(2^{-l})$ ,  $1 \leq l \leq L$ , 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 corresponding block diagram is depicted in Figure 3.1. The preconditioners illustrated in Figures 2.2 and 3.1 are called the SGMF and MGMF preconditioners respectively. Note that the MGMF preconditioner 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  $\Omega_l$  ( $h = 2^{L-l} h$ ) and  $\Omega_{l-1}$  ( $h = 2^{L-l+1} h$ ) can be defined as

$$I_l^{l-1} : \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}^{l-1}, \quad I_{l-1}^l : \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}^l.$$

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 while a upsampling operator followed by



a lowpass filter is equivalent to the interpolation operator [22].

### 3.2 Lowpass versus bandpass filtering

To save computational work, we can further simplify the MGMF preconditioner in Figure 3.1 by deleting the paths and the associated work corresponding to  $I - H_l$ . As given in Figure 3.2, we have the modified MGMF preconditioner

$$Q^{-1}r = \left( \sum_{l=1}^L \frac{1}{d_l} G_l^T G_l \right) r, \quad (3.1)$$

where

$$\begin{aligned} G_L &= I, \\ G_l &= \prod_{p=l+1}^L I_p^{p-1} H_p, \quad \text{for } 2 \leq l \leq L-1, \\ G_1 &= H_2 \prod_{p=3}^L I_p^{p-1} H_p. \end{aligned}$$

Note that the bandpass filters  $F_l$  in the MF preconditioner  $M$  have been replaced by the lowpass filters  $G_l$  in the MF preconditioner  $Q$ . By choosing  $d_l$ 's appropriately, we can make  $Q$  behaves very similarly to  $M$  as described below. With the MF preconditioner implemented by (3.1), Fourier components of band  $B_l$  exist in the first  $L-l+1$  levels and these components are multiplied by  $d_L^{-1}, \dots, d_l^{-1}$  respectively. Therefore, the scaling constants  $d_l$ 's are implicitly defined via

$$\sum_{i=l}^L \frac{1}{d_i} = \frac{1}{c_l}. \quad (3.2)$$

Solving (3.2) for  $d_l$  gives:

$$d_L = c_L \quad \text{and} \quad d_l = \frac{1}{c_l^{-1} - c_{l+1}^{-1}}, \quad l = L-1, \dots, 1. \quad (3.3)$$

However, we find from numerical experiments that the parameter sets  $\{c_l\}$  and  $\{d_l\}$  used in Figure 3.2 give about the same convergence rate. This can be explained by the observation that, for small  $l$ ,  $d_l \approx c_l$  since  $c_l^{-1} \gg c_{l+1}^{-1}$ .

The preconditioning  $Q^{-1}r$  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.

### 3.3 Discretization with nonuniform grids

The above 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  $\lambda_k$  in band  $B_l$  (see Section 2.2) 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

$$H_{L,BPX} : \frac{1}{8} \begin{vmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{vmatrix}, \quad (3.4)$$

which is different from  $H_{L,1}$  given by (2.17). We can derive other filters from (3.4) by applying it more than once. For example, by applying it twice, we get

$$H_{L,TBPX} : \frac{1}{16} \begin{vmatrix} 0 & 0 & 1 & 2 & 1 \\ 0 & 2 & 6 & 6 & 2 \\ 1 & 6 & 10 & 6 & 1 \\ 2 & 6 & 6 & 2 & 0 \\ 1 & 2 & 1 & 0 & 0 \end{vmatrix}. \quad (3.5)$$

In order to eliminate the directional preference, we can apply (3.4) in alternating directions which gives a symmetric filter:

$$H_{L,ADBFX} : \frac{1}{16} \begin{vmatrix} 0 & 1 & 2 & 1 & 0 \\ 1 & 4 & 6 & 4 & 1 \\ 2 & 6 & 8 & 6 & 2 \\ 1 & 4 & 6 & 4 & 1 \\ 0 & 1 & 2 & 1 & 0 \end{vmatrix}. \quad (3.6)$$

### 3.4 Diagonal scaling

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

$$A = D^* \tilde{A} D^*,$$

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  $A u = f$ , we can solve an equivalent problem  $\tilde{A} \tilde{u} = \tilde{f}$ , where  $\tilde{u} = D^* u$  and  $\tilde{f} = D^{-*} f$ , 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 et al.[8].

### 3.5 Summary of practical MGMF algorithm

Given a sequence of grids  $\Omega_l$ ,  $1 \leq l \leq L$ , down-sampling  $(I_{l+1}^l)$  and up-sampling  $(I_l^{l+1})$  operators between grids  $\Omega_l$  and  $\Omega_{l+1}$ , and appropriate elementary filters  $H_l$  defined on  $\Omega_l$ , the algorithm corresponding to the block diagram given by Figure 3.2 can be summarized as follows,

Decomposition:

$$\begin{aligned} v_L &:= r, \\ \text{for } l &= L-1, \dots, 2 \\ v_l &:= I_{l+1}^l H_{l+1} v_{l+1}, \\ v_1 &:= H_2 v_2, \end{aligned}$$

Scaling:

$$\begin{aligned} \text{for } l &= L, \dots, 1 \\ w_l &:= v_l d_l^{-1} \end{aligned}$$

Synthesis:

$$\begin{aligned} s_2 &:= w_2 + H_2 w_1, \\ \text{for } l &= 2, \dots, L \\ s_l &:= w_l + H_l I_{l-1}^l s_{l-1} \end{aligned}$$

$$Q^{-1}r := s_L$$

Table 3.1 Computation of  $Q^{-1}r$

This is the MGMF algorithm implemented in Section 5.

#### 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 fix number of cycles of a conventional multigrid method. This approach has been 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 re-entrant 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  $\psi_j^l$ , 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 vector  $v$  can be written as:

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

and can be computed by a V-cycle with the matrix  $S^T$  corresponds to a fine to coarse grid traversal and  $S$  a coarse to fine traversal. On each level, only local operations are performed. In 2D, the condition number of the preconditioned system can be showned to

grow like  $O(\log^2 h^{-1})$ , which is a very slow growth. Unfortunately, this nice property is lost in 3D where the growth can be proved to be  $O(h^{-1})$ [26],[29]. However, these theoretical results are proven under much weaker regularity assumptions than for the multigrid methods and 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  $\phi_j^l$  are the nodal basis functions and  $h_l$  is measure of the mesh size of grid level  $l$ . Since the form of their preconditioner is very similar to that for the hierarchical basis method, the computations can be arranged in a similar way via a V-cycle. They proved that the condition number of the preconditioned system 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 method.

#### 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 successive Schur complements at a given grid level are 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 also been proposed by Kuznetsov [24]. Later, Axelsson-Vassilevski [6],[7] improved this bound to  $O(1)$  but carrying out recursively more (Chebychev) 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 presentation above, various multilevel preconditioners share some similarities. Most of the multilevel preconditioners are in the form of a multigrid V-cycle (MG, HB, BPX and MF) except AMG methods. 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 operators 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. non-multigrid) 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 as an iterative method (instead of as a preconditioner). Several of the preconditioners have the 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, it seems that to get a spectrally equivalent preconditioner, one has to go beyond the V-cycle and perform more iterations on each grid as in the AMP methods.



## 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 discussed previously. The preconditioners implemented are:

- HB: hierarchical basis preconditioner using linear elements for 2D and trilinear elements for 3D 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 2D problems ( $H_L$  given by (3.4)),
- BPX2: a modified version of BPX preconditioner by filtering twice for 2D problems ( $H_L$  given by (3.5)),
- BPX3: another modified version of BPX preconditioner by filtering twice but using linear elements of different orientations for 2D problems ( $H_L$  given by (3.6)),
- MGMF1: the MGMF preconditioner with the 9-point (2.15) or 27-point filter for 2D and 3D 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 comparison purpose. For the relaxation factor, we use the optimal value  $\omega = 1 - 8\sin^2 \frac{\pi h}{2}$  from [11]. The number of iterations required for RIC can be bounded by  $O(n^{\frac{1}{2}})$ .

The operation counts per iteration (just for preconditioning) for each method for 2D and 3D problems are given in Tables 5.1 and 5.2 respectively. The operation counts include addition, multiplication and division (each is counted as one operation) and does not include other overhead operations such as condition checking or data copying. The operations required in each CG step for 2D and 3D problems are  $21N$  and  $25N$

respectively.

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 5.1: Work per iteration for preconditioners (2D)

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

Table 5.2: Work per iteration for preconditioners (3D)

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 2D 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 3D problems, the BPX1 (BPX2) preconditioning using trilinear elements is same as the MGMF1 (MGMF2) preconditioning as shown in Table 5.2. The MG preconditioner has not yet been implemented for 3D 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 chosen to be  $||r^k|| / ||r^0|| \leq 10^{-6}$ . Diagonal scaling is always used except for RIC. The six test problems are:

(1) the 2D model problem with solution  $u = x(x-1)y(y-1) e^{xy}$ ,

$$\Delta u = f, \quad \Omega = (0,1)^2, \quad (5.1)$$

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

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

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

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

where

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

(4) the 3D model problem with solution  $u = x(1-x)y(1-y)z(1-z)e^{xyz}$ ,

$$\Delta u = f, \quad \Omega = (0,1)^3, \quad (5.4)$$

(5) a 3D variable coefficient problem with solution  $u = e^{xyz} \sin \pi x \sin \pi y \sin \pi z$ ,

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

(6) a 3D problem with discontinuous coefficients with  $f = 2x(1-x) + 2y(1-y) + 2z(1-z)$ ,

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

where

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

The number of iterations and operation counts per grid point are plotted in Figures 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 compared to the HB preconditioner, especially for 3D problems. The HB method is competitive with the other multilevel methods only for the discontinuous coefficient problem in 2D.
2. The  $O(\log^2 n)$  convergence rate for all the multilevel methods is evident, except for the 3D HB method. The 3D 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 help to improve the convergence rates for the model Poisson problem in both 2D and 3D (the MGMF2 and BPX3 preconditioners appear to be spectrally equivalent.) However, 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 number of iterations required. However, in terms of amount of work, MGMF3 is better than MGMF1 and MGMF2.

6. For small  $n$  ( approximately  $< 100$  ), the RIC method is actually quite competitive with all the multilevel methods. In fact, for the discontinuous coefficient problems, none of the multilevel preconditioners gives 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 are significantly improved. This may be due to the extra smoothing operations in the multigrid methods which are more adapt 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 non-uniform and refined meshes. Even though the RIC preconditioner shows better convergence rate for strongly discontinuous coefficient problems, it has a low degree of parallelism. The multilevel filtering preconditioners are very similar to the BPX method. What the filtering framework provides is the flexibility of filter design which can lead to more efficient methods.

## References

1. L. M. Adams and E. G. Ong, "A comparison of preconditioners for GMRES on parallel computers," in *Parallel Computations and Their Impact on Mechanics*, ed. A. K. Noor, pp. 171-186, The American Society of Mechanical Engineers, New York, N.Y., 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*, vol. 13, pp. 443-467, 1972.
4. O. Axelsson, "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.*, vol. 48, pp. 479-498, 1986.
6. O. Axelsson and P. Vassilevski, "Algebraic multilevel preconditioning methods, I," Report 8811, Department of Mathematics, Catholic University, The Netherlands, 1988.
7. O. Axelsson and P. Vassilevski, "Algebraic multilevel preconditioning methods, II," Report 1988-15, Institute for Scientific Computation, University of Wyoming, Laramie, Wyoming, 1988.
8. R. E. Bank, T. F. Dupont, and H. Yserentant, "The hierarchical basis multigrid method," Preprint, 1987.
9. J. H. Bramble, J. E. Pasciak, and J. Xu, "Parallel multilevel preconditioners," To appear in *Math. Comp.*.
10. A. Brandt, "Multi-level adaptive solutions to boundary-value problems," *Math. Comp.*, vol. 31, no. 138, pp. 333-390, 1977.
11. T. F. Chan, "Fourier analysis of Relaxed Incomplete Cholesky factorization preconditioners," UCLA, CAM report 88-34, 1988.
12. T. F. Chan, C.-C. J. Kuo, and C. Tong, "Parallel elliptic preconditioners: Fourier analysis and performance on the Connection Machine," *Computer Physics Communications*, vol. 53, pp. 237-252, 1989.
13. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1983.
14. G. Dahlquist and A. Björck, *Numerical Methods*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 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.*, vol. 5, no. 3, pp. 559-573, 1968.
16. A. Greenbaum, C. Li, and H. Z. Chao, "Parallelizing preconditioned conjugate gradient algorithms," *Computer Physics Communications*, vol. 53, pp. 295-309, 1989.
17. W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, Germany, 1985.
18. O. Herrmann, "On the approximation problem in nonrecursive digital filter design," *IEEE Trans. on Circuit Theory*, vol. CT-18, pp. 411-413, 1971.
19. 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.

20. 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.
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," *Applied Mathematics and Computation*, vol. 29, pp. 69-87, 1989.
23. C.-C. J. Kuo and T. F. Chan, "Two-color Fourier analysis of iterative algorithms for elliptic problems with red/black ordering," To appear in *SIAM J. Sci. Stat. Comput.*
24. 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.
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.*, vol. 31, no. 137, pp. 148-162, 1977.
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 98195, 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, N.Y. 14853, 1989.
29. H. Yserentant, "On the multi-level splitting of finite element spaces," *Numer. Math.*, vol. 49, pp. 379-412, 1986.



## Figure Captions

Figure 2.1: Spectra of  $A$ ,  $P^{-1}$  and  $P^{-1}A$ .

Figure 2.2: Blockdiagram of the MF preconditioner with a single discretization grid (SGMF).

Figure 2.3: Spectra of maximally flat lowpass filters  $H_{L,J}$  with  $J = 1, 3, 5$ .

Figure 2.4: Spectra of  $H_L$ ,  $I-H_{L-1}$  and  $F_{L-1}$ .

Figure 2.5: Eigenvalues of  $M^{-1}A$  with  $J = 1, 3, 5$ .

Figure 2.6: Condition numbers of the MF-preconditioned Laplacian with (a)  $J=1$  and (b)  $J=3$ .

Figure 3.1: Blockdiagram of the MGMF preconditioner.

Figure 3.2: Blockdiagram of the modified MGMF preconditioner.

Figure 5.1: (a) Iteration and (b) operation counts for Test Problem 1.

Figure 5.2: (a) Iteration and (b) operation counts for Test Problem 2.

Figure 5.3: (a) Iteration and (b) operation counts for Test Problem 3.

Figure 5.4: (a) Iteration and (b) operation counts for Test Problem 4.

Figure 5.5: (a) Iteration and (b) operation counts for Test Problem 5.

Figure 5.6: (a) Iteration and (b) operation counts for Test Problem 6.

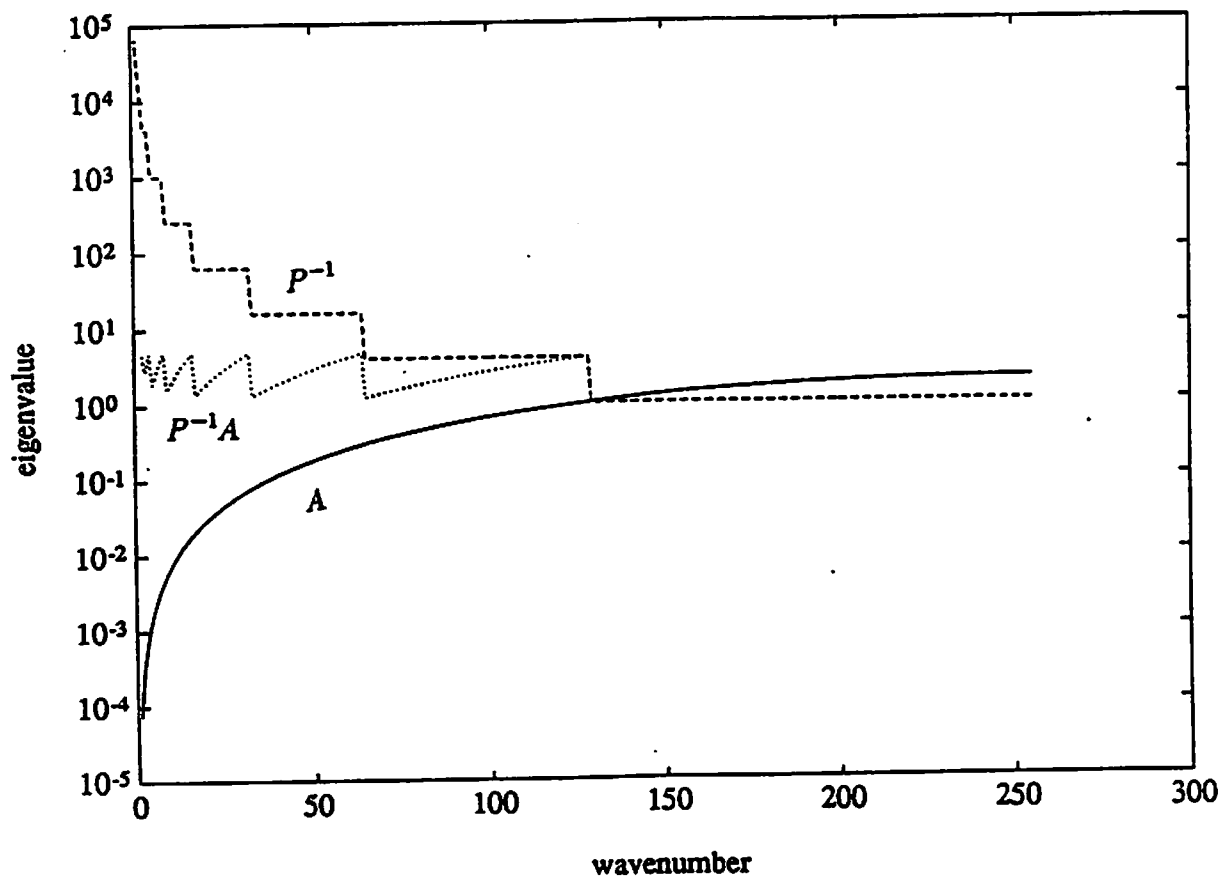


Figure 2.1: Spectra of  $A$ ,  $P^{-1}$  and  $P^{-1}A$ .

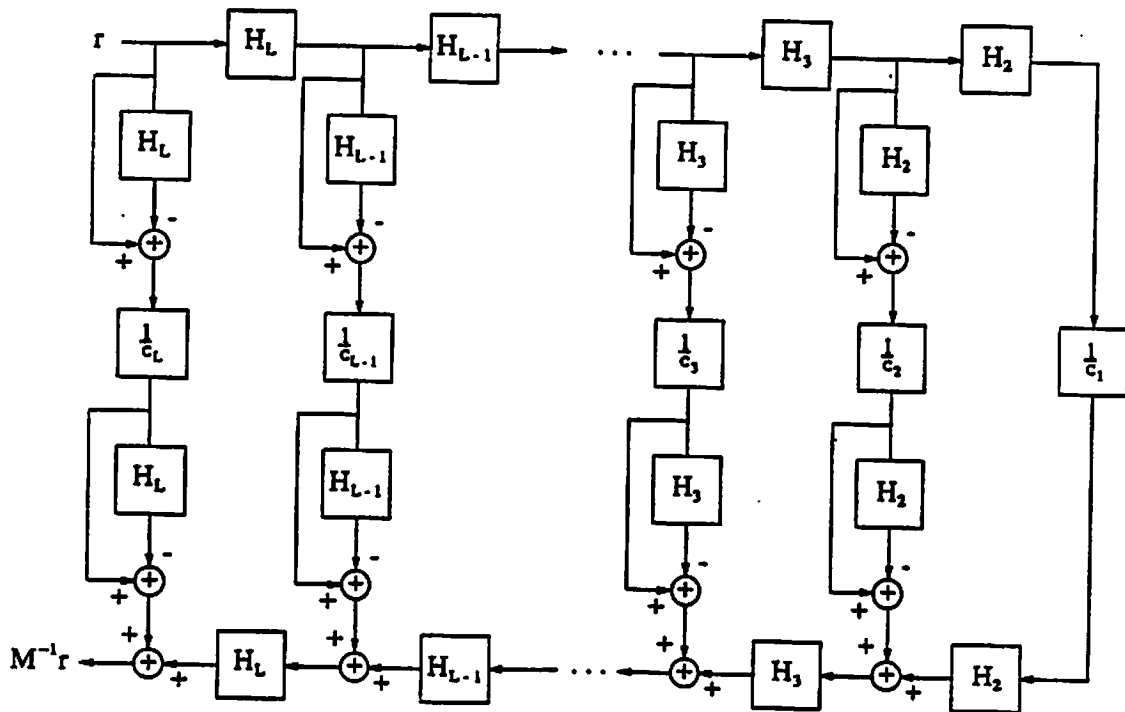


Figure 2.2: Blockdiagram of the MF preconditioner with a single discretization grid (SGMF).

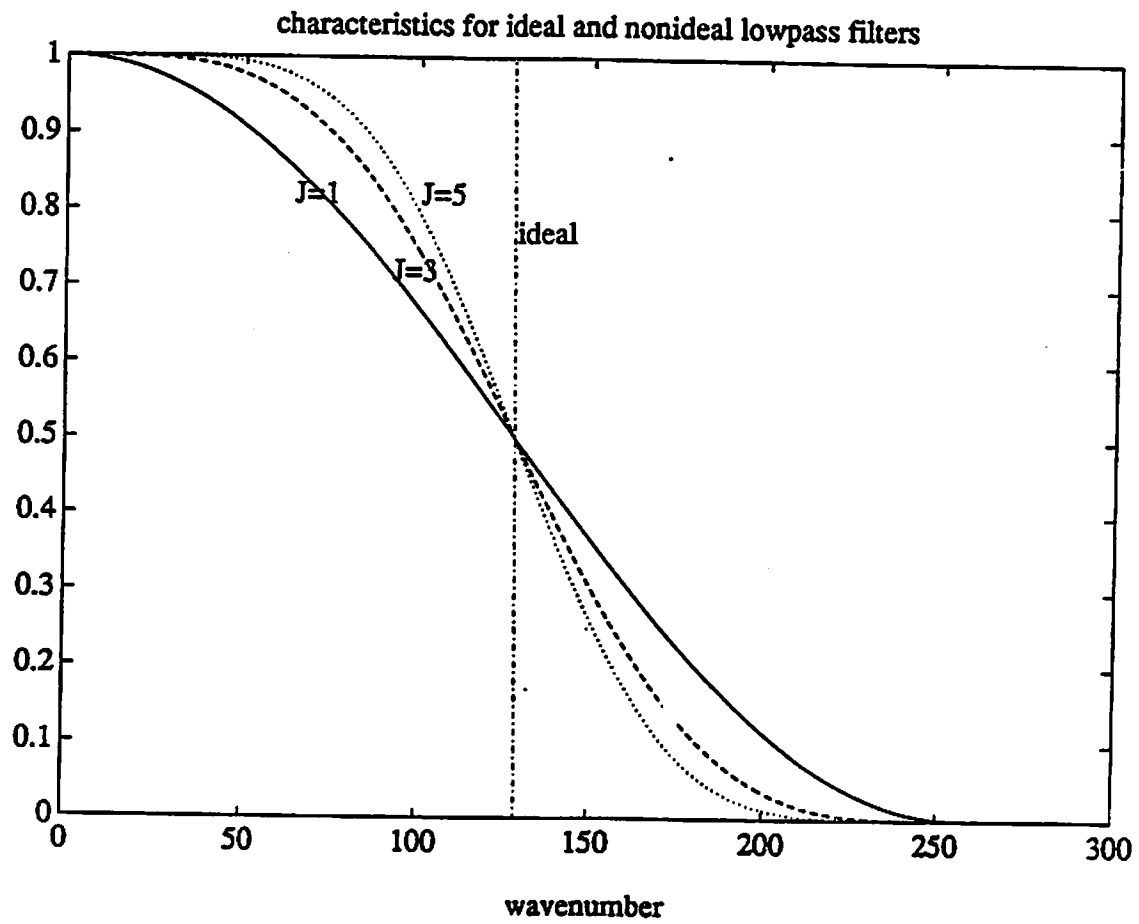


Figure 2.3: Spectra of maximally flat lowpass filters  $H_{L,J}$  with  $J = 1, 3, 5$ .

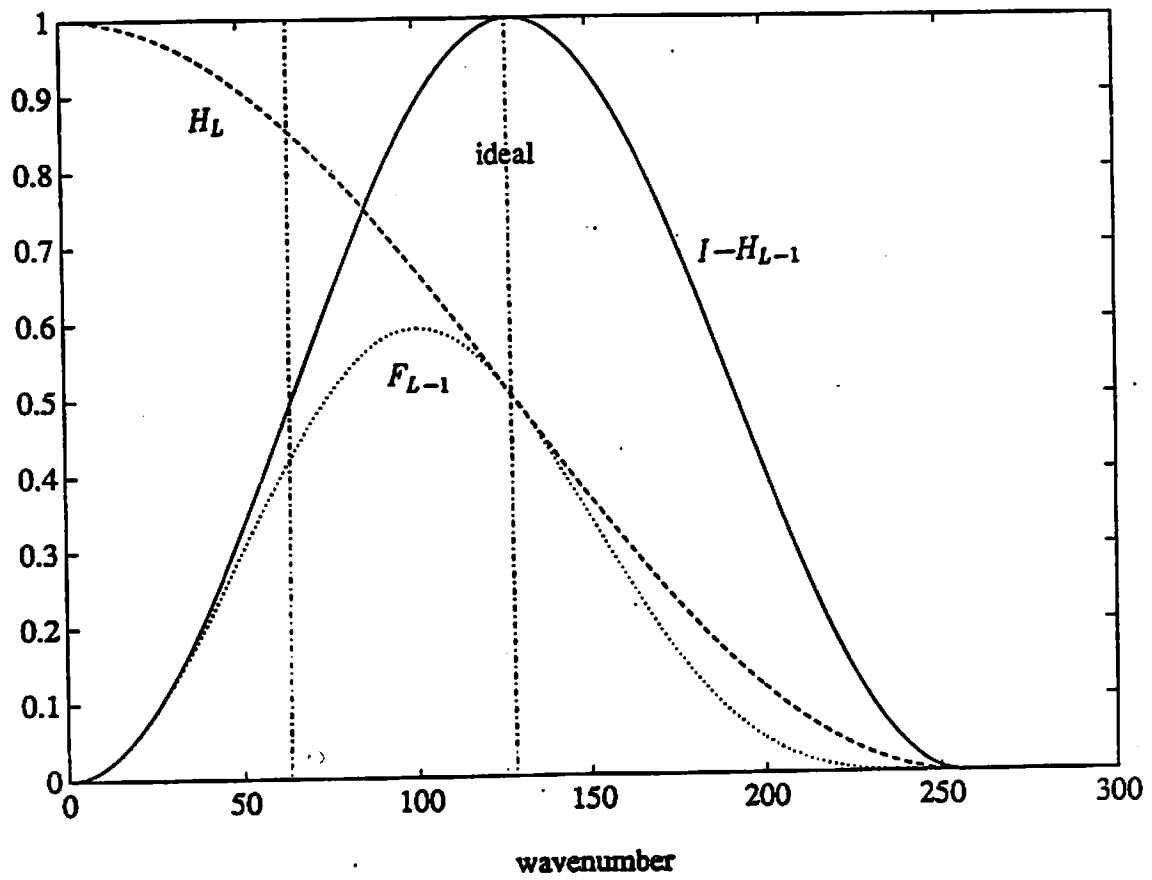


Figure 2.4: Spectra of  $H_L$ ,  $I-H_{L-1}$  and  $F_{L-1}$ .

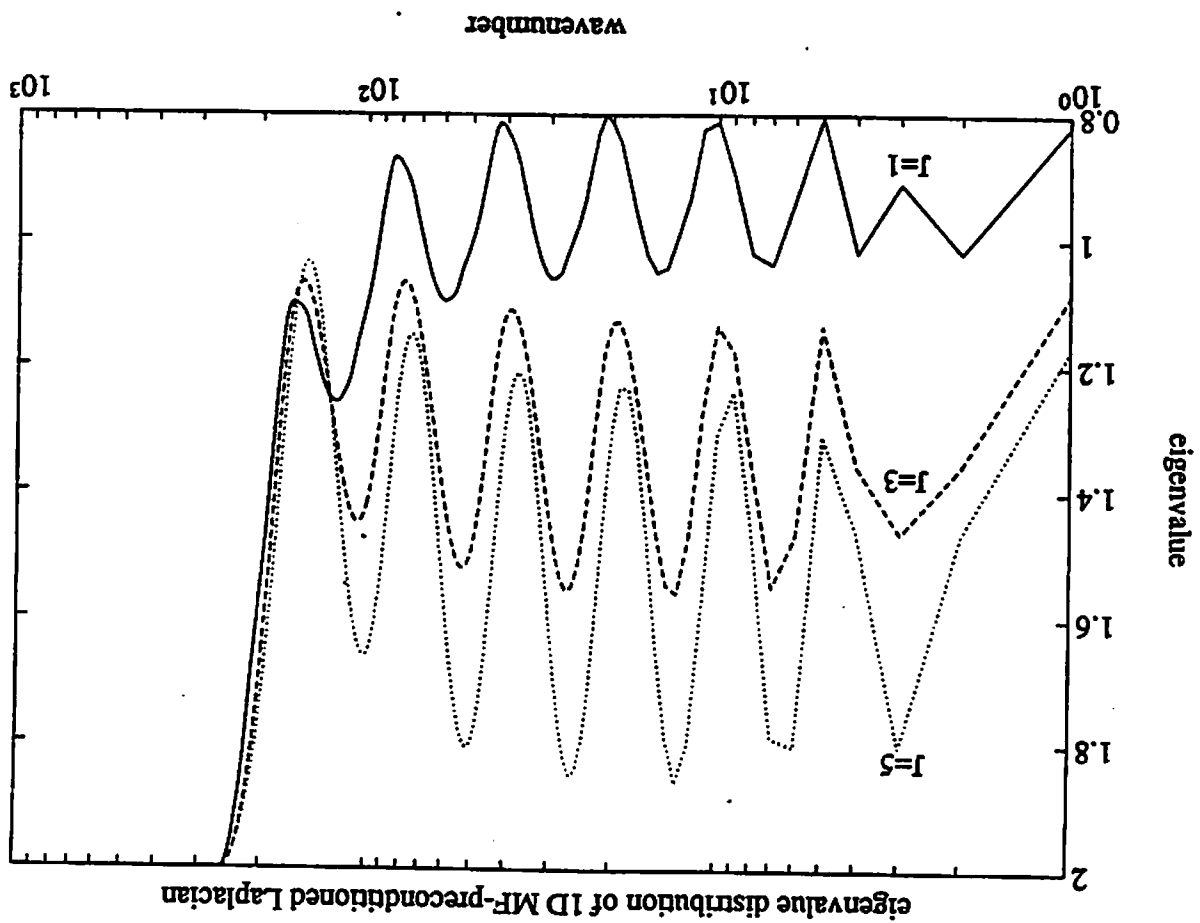


Figure 2.5: Eigenvalues of  $M^{-1}A$  with  $J = 1, 3, 5$ .

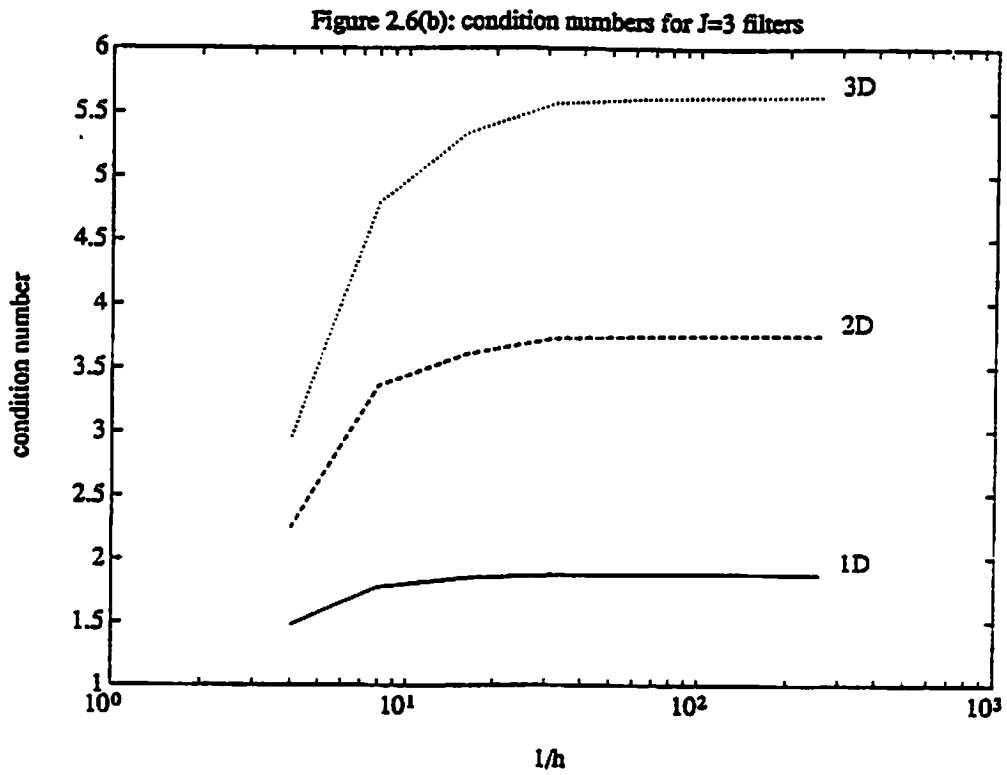
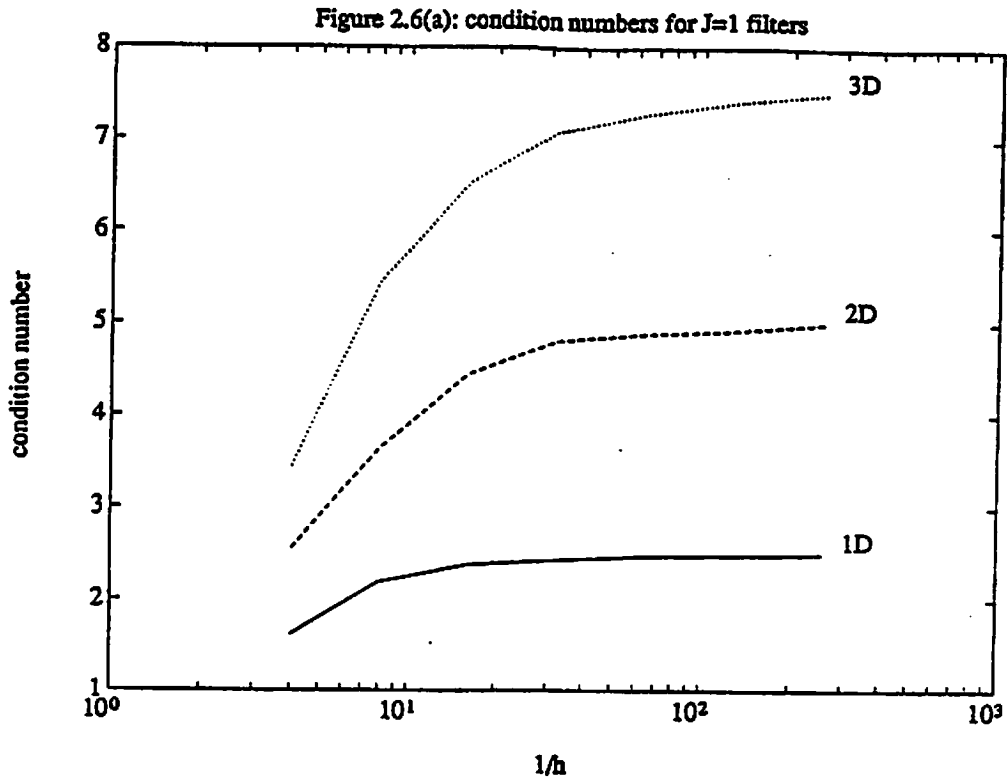


Figure 2.6: Condition numbers of the MF-preconditioned Laplacian with (a)  $J=1$  and (b)  $J=3$ .

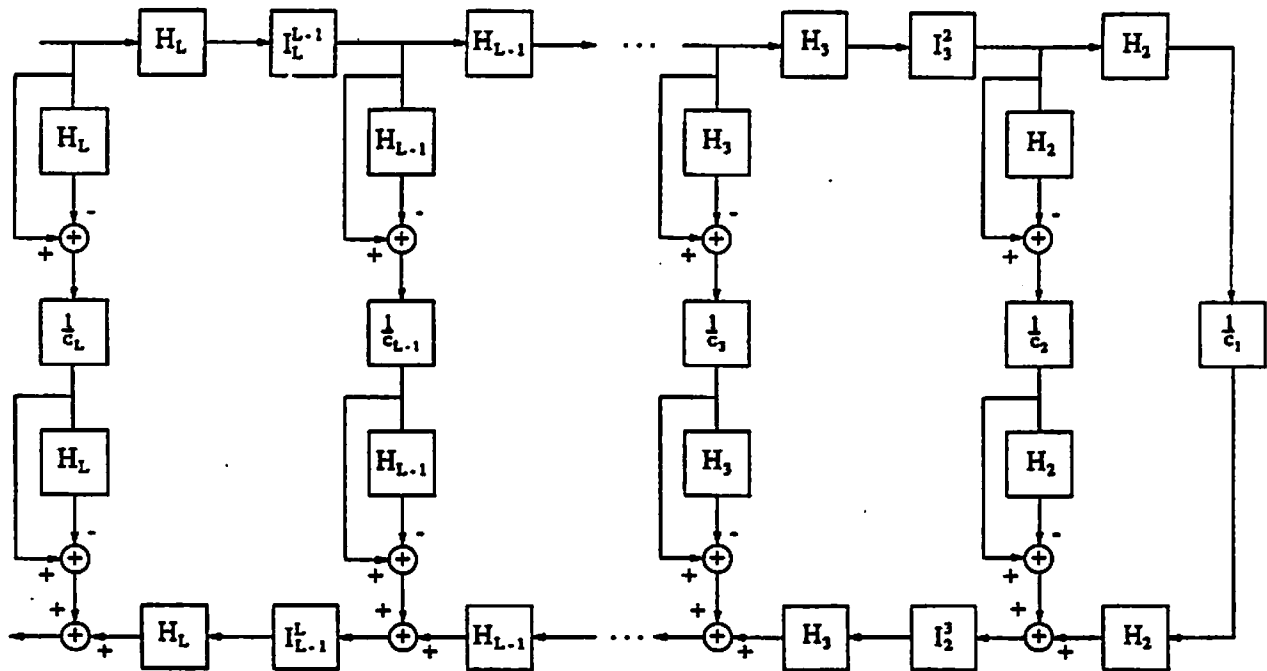


Figure 3.1: Blockdiagram of the MGMF preconditioner.



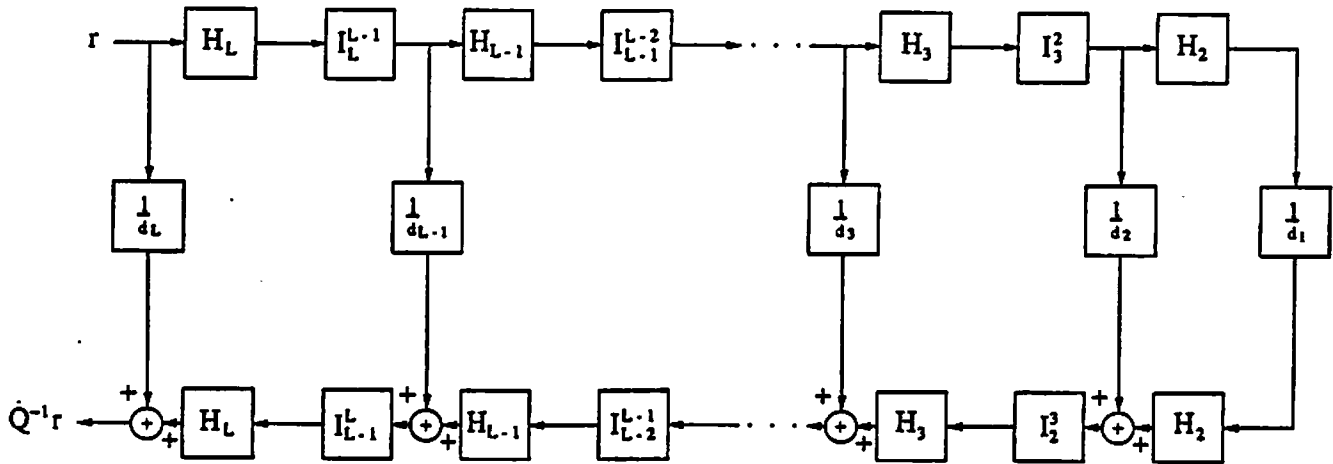


Figure 3.2: Blockdiagram of the modified MGMF preconditioner.

Figure 5.1(a): iteration counts for Test Problem 1

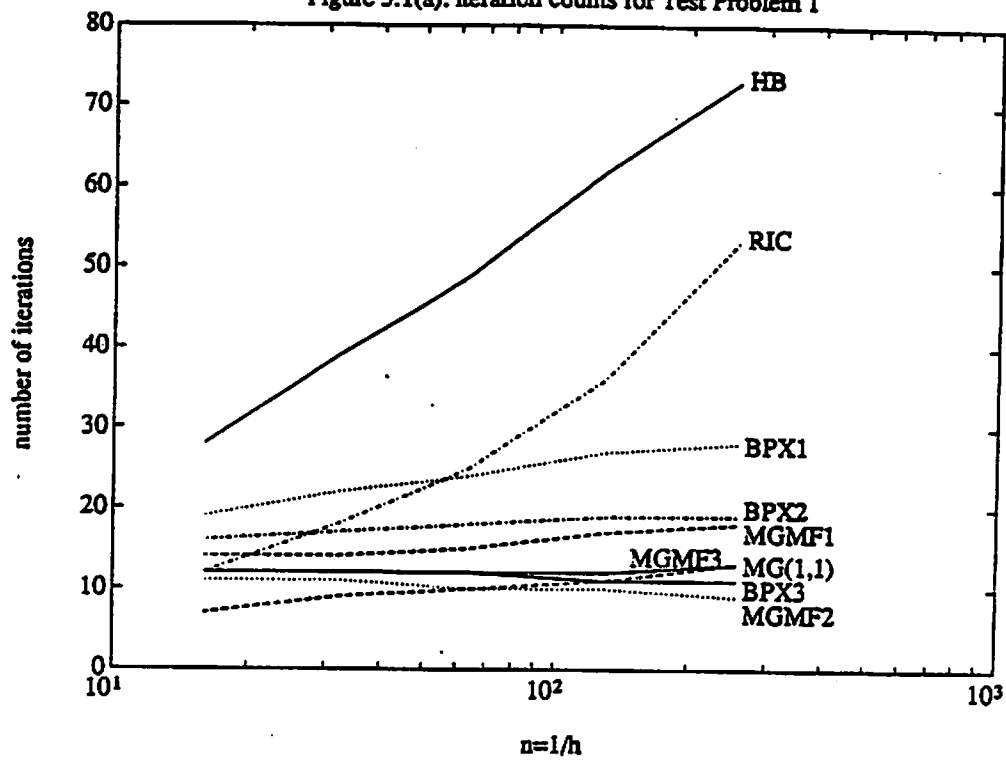


Figure 5.1(b): operation counts per grid point for Test Problem 1

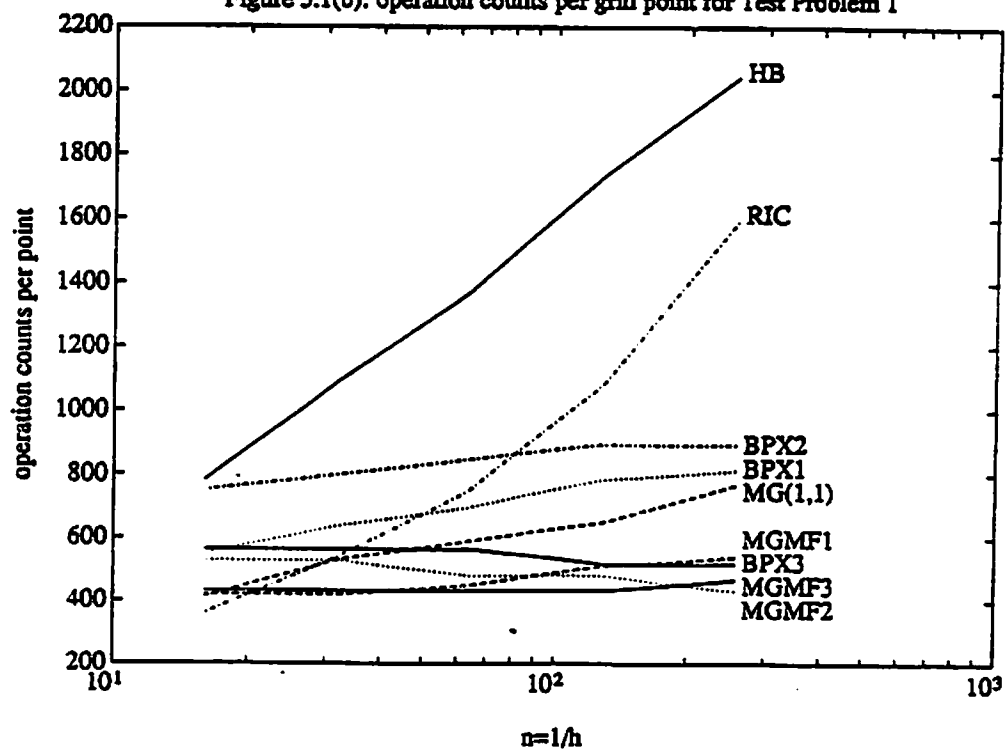


Figure 5.1: (a) Iteration and (b) operation counts for Test Problem 1.

Figure 5.2(a): iteration counts for Test Problem 2

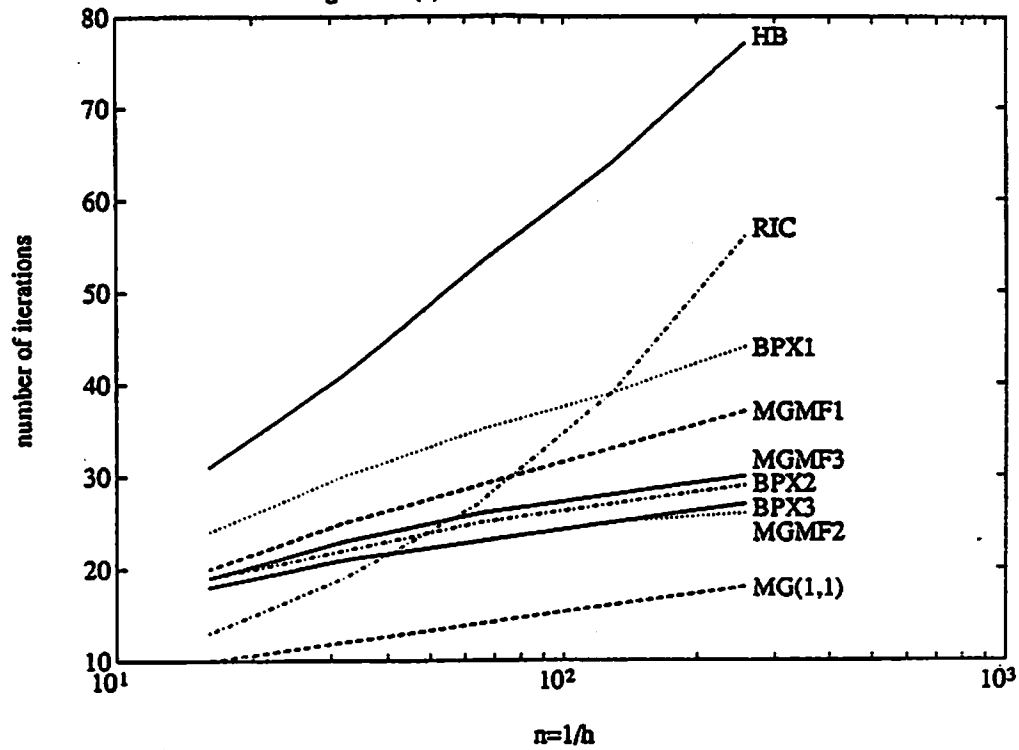


Figure 5.2(b): operation counts per grid point for Test Problem 2

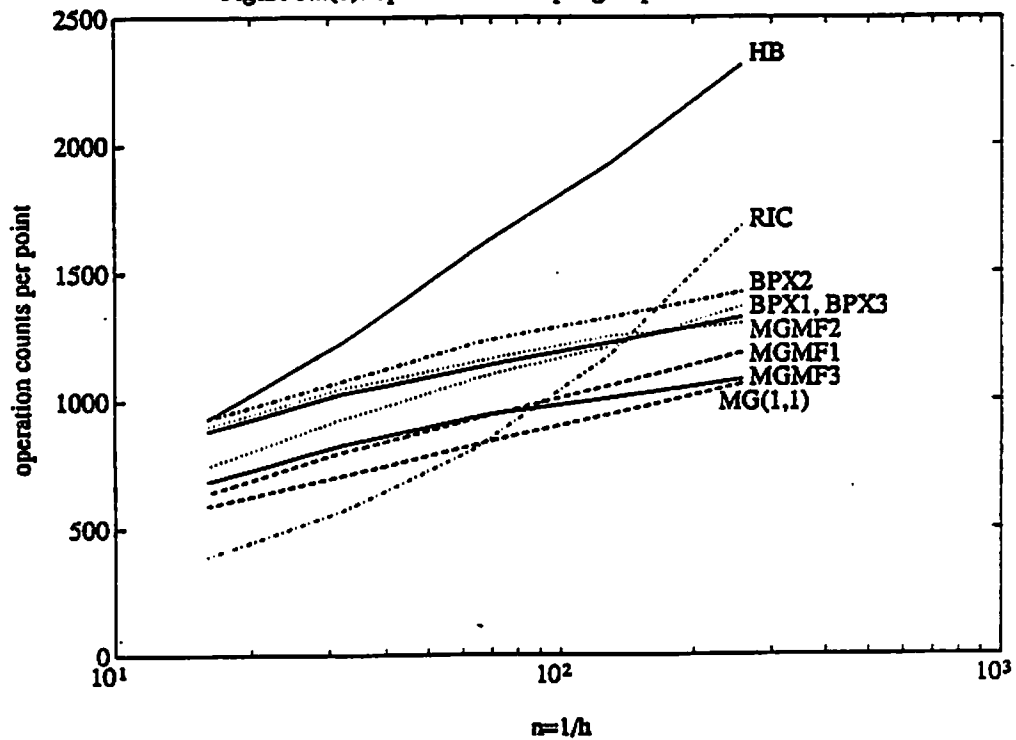


Figure 5.2: (a) Iteration and (b) operation counts for Test Problem 2.

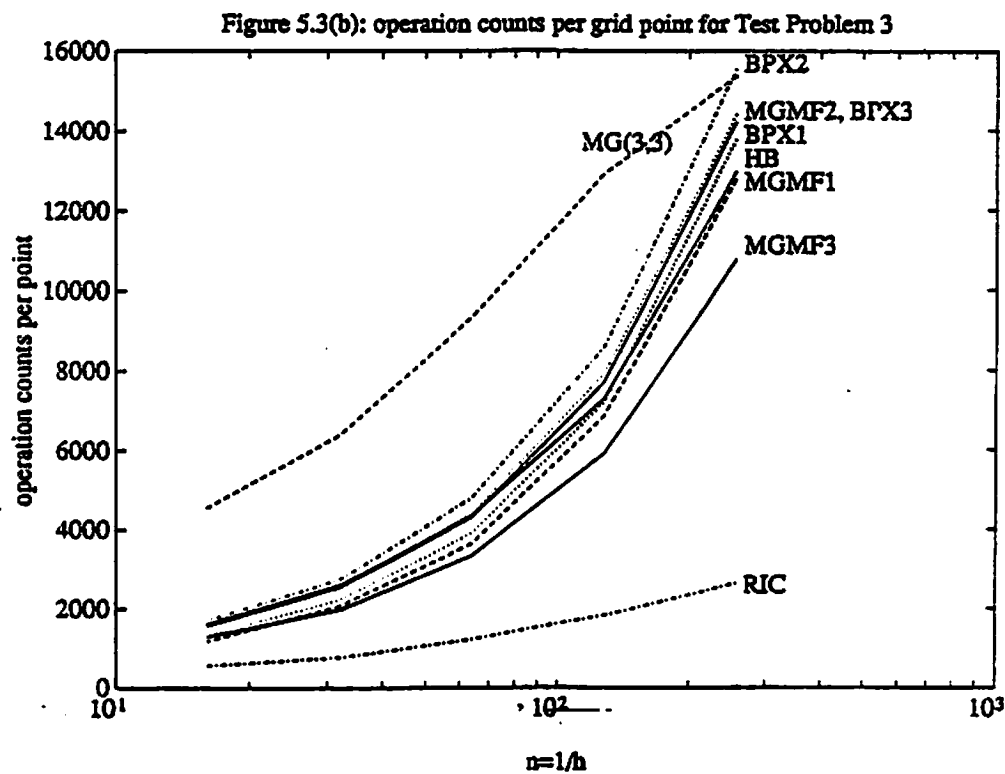
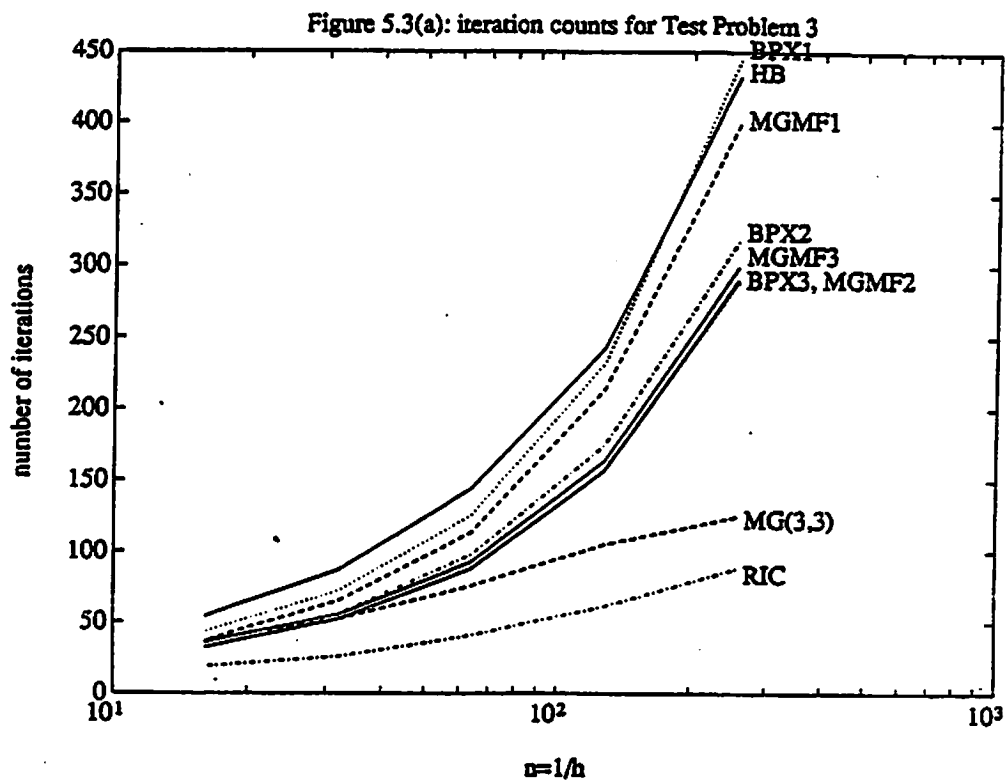


Figure 5.3: (a) Iteration and (b) operation counts for Test Problem 3.

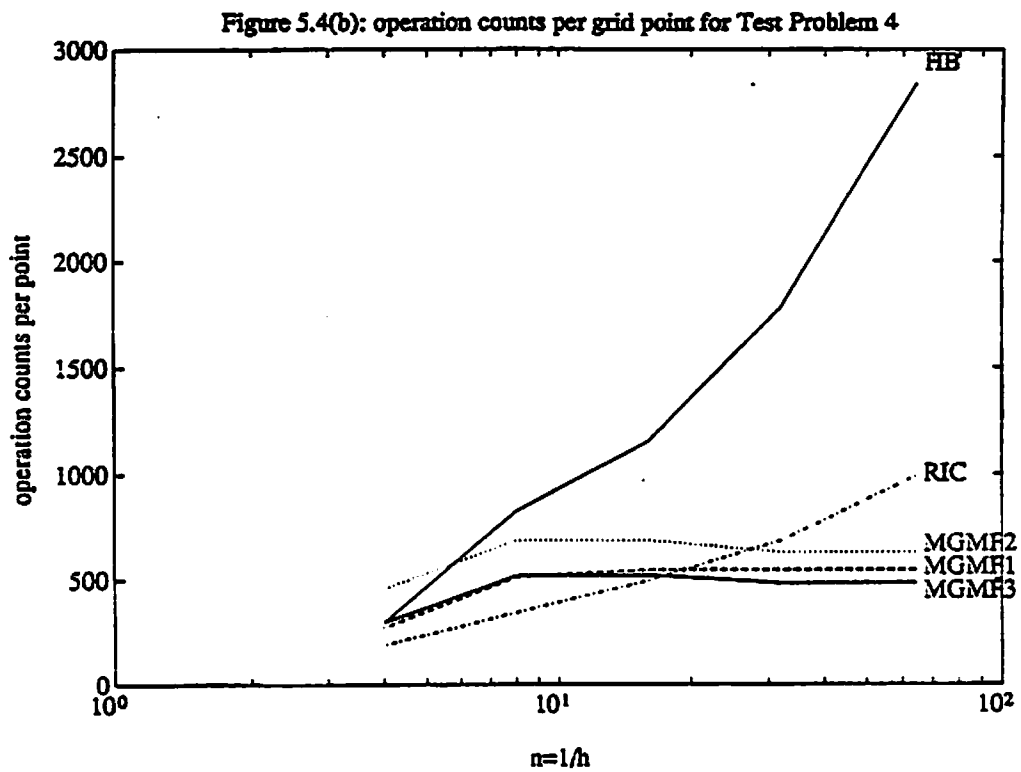
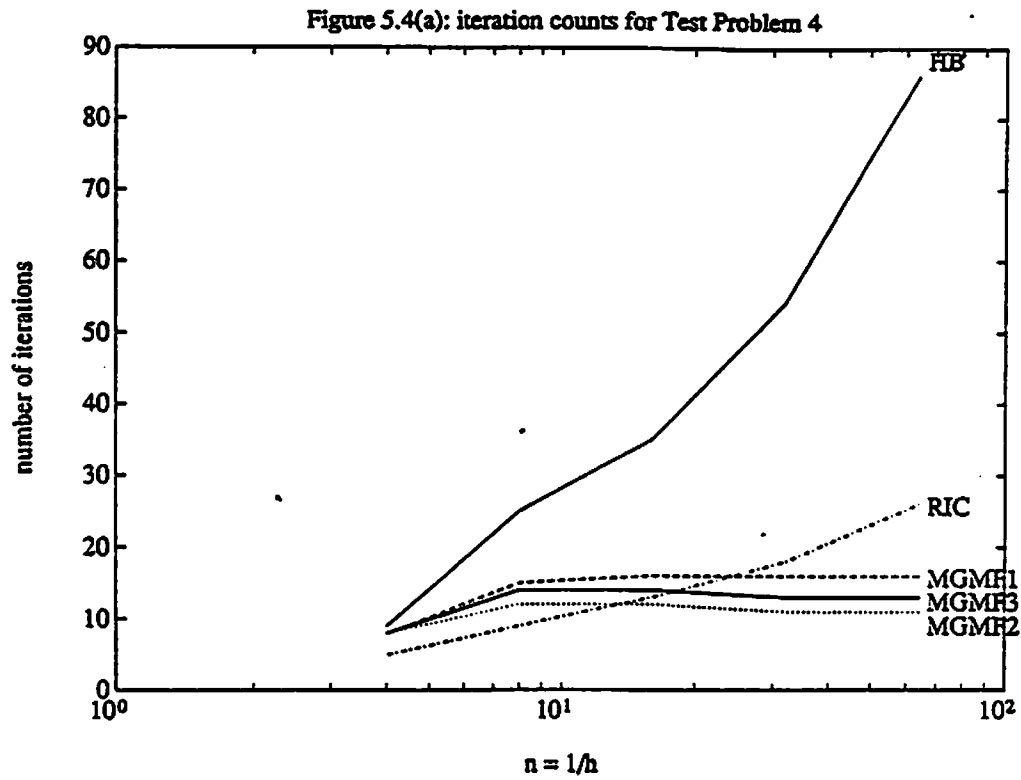


Figure 5.4: (a) Iteration and (b) operation counts for Test Problem 4.

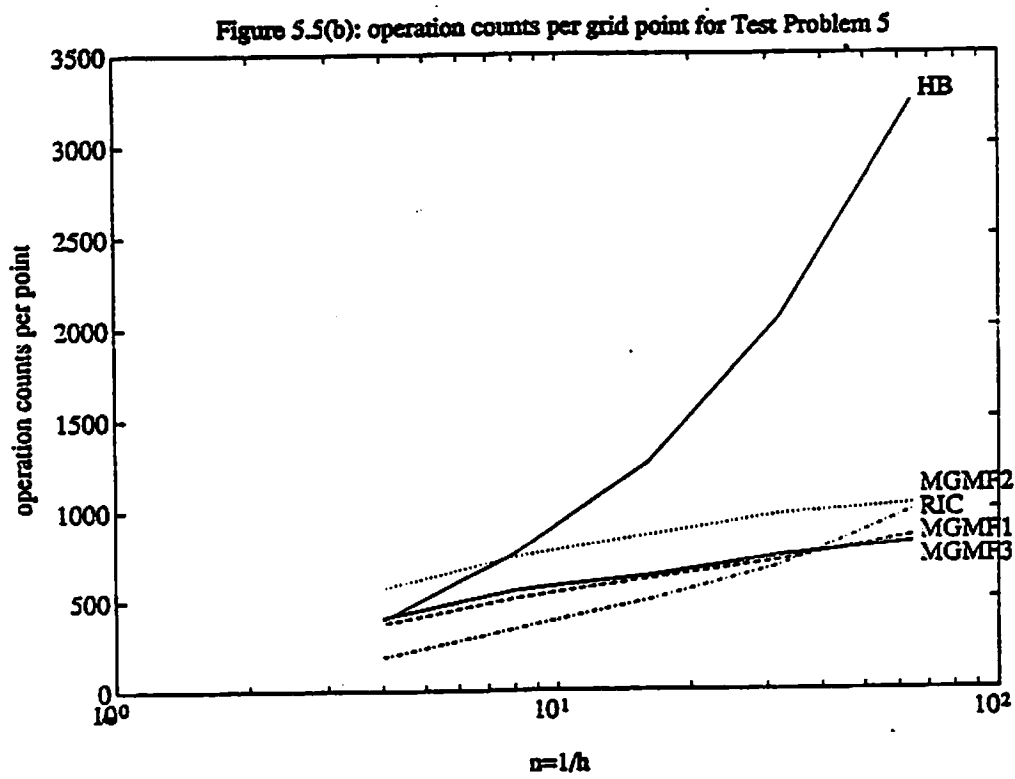
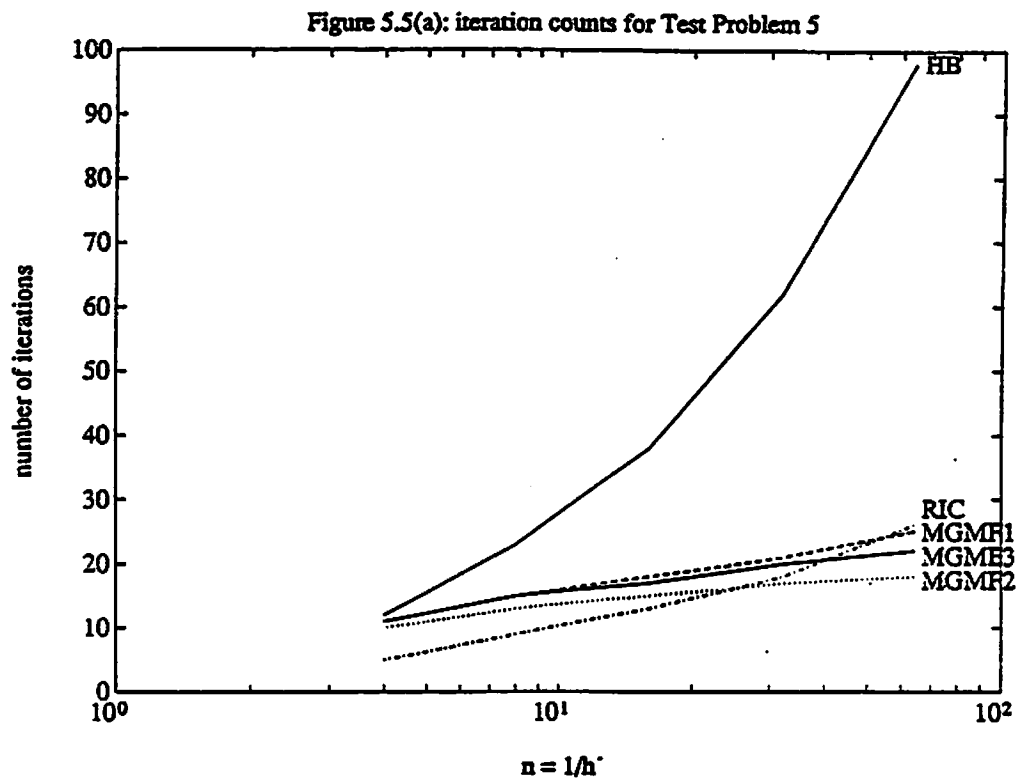


Figure 5.5: (a) Iteration and (b) operation counts for Test Problem 5.

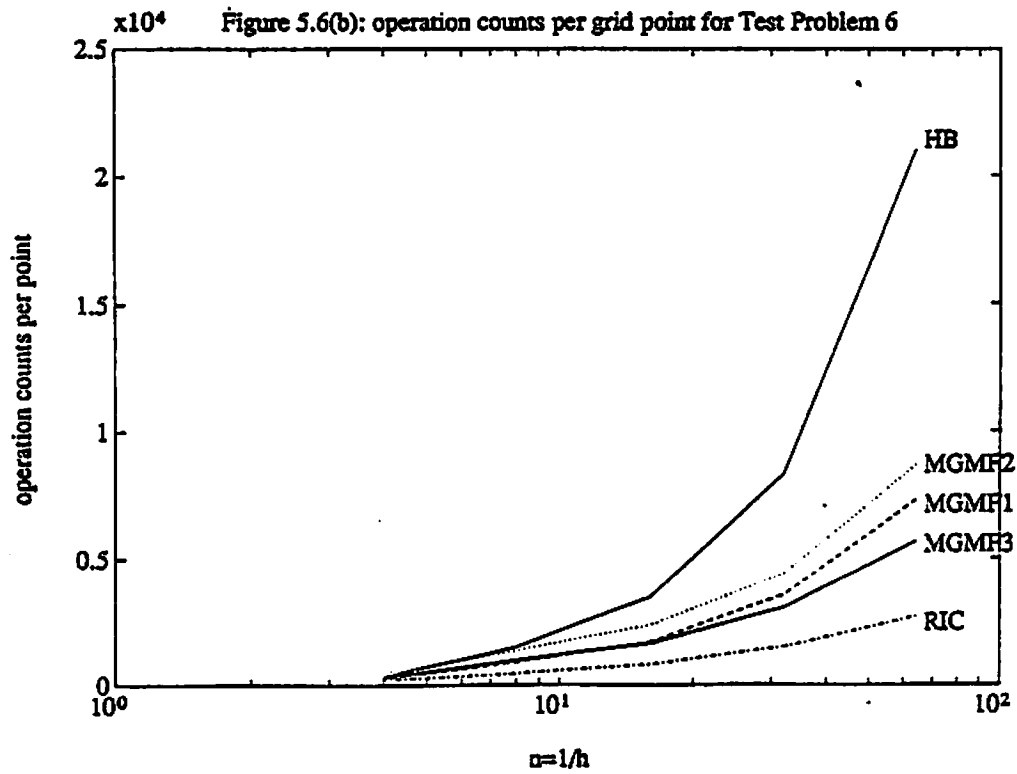
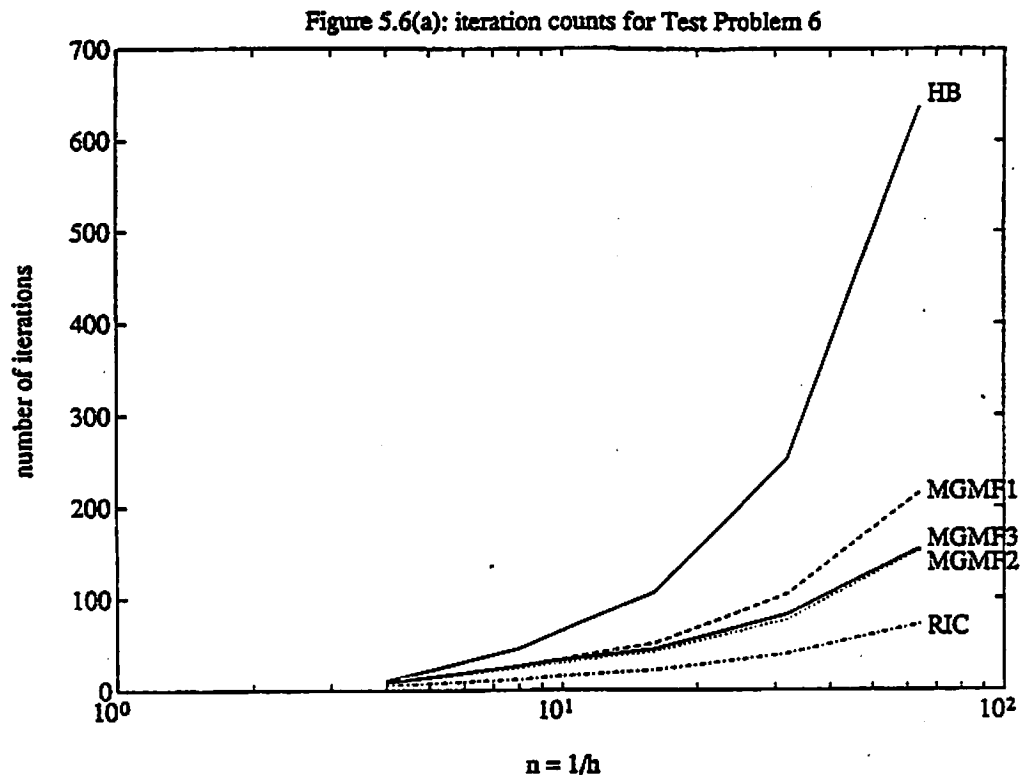


Figure 5.6: (a) Iteration and (b) operation counts for Test Problem 6.