USC-SIPI REPORT #154

Multilevel Filtering Preconditioners: Extensions to More General Elliptic Problems

by

Charles H. Tong, Tony F. Chan
and C.C. Jay Kuo

May 1990

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.

MULTILEVEL FILTERING PRECONDITIONERS : EXTENSIONS TO MORE GENERAL ELLIPTIC PROBLEMS [*]

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

Abstract. We briefly review the concept of multilevel filtering (MF) preconditioning applied to second-order self-adjoint elliptic problems. We then show how to effectively apply this concept to other elliptic problems such as the second-order anisotropic problem, Helmholtz equation, convection-diffusion equation, biharmonic equation, equations on locally refined grids and interface operators arising from domain decomposition methods. Numerical results are given to show the effectiveness of the MF preconditioners on these problems.

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 effective implementation of these methods on various parallel computers. Since effective preconditioning plays a critical role in the competitiveness of the 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, the incomplete factorization preconditioners (ILU and MILU) and polynomial preconditioners. Many such preconditioners have been very successful in giving high performance, especially when implemented on sequential computers.

In the parallel implementation of PCG methods, the major bottleneck is often the parallelization of the preconditioner. The rest of the PCG methods can usually be parallelized in a straightforward way (the inner product computation is also considered a bottleneck but its wide applicability in other methods has prompted many parallel computer manufacturers to develop highly optimized and efficient code for it). Unfortunately, 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 of many preconditioners which are effective in improving the convergence rate (e.g. SSOR, ILU and MILU) is the sequential manner in which these preconditioners use in traversing the computational grid - the data dependence implicitly prescribed by the method 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

[*] This work was supported in part by the Department of Energy under contract DE-FG-03-87-ER-25037 and the Army Research Office under contract DAAL03-88-K-0085. The first author is supported by a DARPA-sponsored Graduate Research Assistantship (award no. 26947F) through UMIACS at University of Maryland. The last author is supported by USC Faculty Research and Innovation Fund.

[†] Dept. of Computer Science, Univ. of Calif. at Los Angeles, CA 90024. E-mail: tong@cs.ucla.edu.

[‡] Dept. of Mathematics, Univ. of Calif. at Los Angeles, CA 90024. E-mail: chan@math.ucla.edu.

[§] Dept. of Electrical Engineering-Systems, Univ. of Southern Calif., CA 90089-0272. E-mail: cckuo@brand.usc.edu

preconditioners) to improve the parallelization alone often has an adverse effect on the rate of convergence [3].

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 limited in their ability to improve the convergence rate. On the other hand, global coupling through a naturally-ordered grid traversal is not highly parallelizable. The challenge is therefore to construct effective global coupling that are highly parallelizable. 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.) Preconditioners of the multilevel type have been proposed recently by other researchers such as Bramble-Pasciak-Xu [2], Axelsson [6], Vassilevski [10], Axelsson-Vassilevski [7] [8] and Kuznetsov [9]. In our previous paper [4], we presented a new class of multilevel filtering (MF) preconditioners for elliptic problems built on ideas from digital filtering theory and implemented on a multilevel grid structure. The MF preconditioner in its multigrid formulation is similar to the conventional multigrid method without smoothing. It is designed to capture the mesh-dependent spectral property of a discretized elliptic operator. The variations of coefficients are handled by the conjugate gradient method with diagonal scaling.

In this paper we will first review briefly the concept of MF preconditioning and numerical results for standard Poisson-like problems with variable and discontinuous coefficient problems. Then we will show the flexibility of the MF preconditioners by adapting them to other elliptic problems that give rise to symmetric or symmetrizable and positive definite systems. These problems include the second-order anisotropic problems, the Helmholtz equation, the convection-diffusion equations for small convection terms, fourth-order elliptic problems such as the biharmonic equation, problems on locally refined grids, and interface operators for domain decomposition methods. We emphasize the ease of tailoring the original MF preconditioner for Poisson-like problems to these problems using the same filtering framework. In particular, the second-order anisotropic problems and problems on locally refined grids can be solved more efficiently by using different types of filters while the other problems require the use of different scaling functions in the course of preconditioning. Extensive numerical experiments are given to demonstrate the effectiveness of the MF preconditioners.

2. MF Preconditioners for Poisson-like Problems.

2.1. Concept and Algorithm. We shall motivate the construction of the MF preconditioner by first considering the following 1D Poisson equation on $\Omega = [0, 1]$

$$-\Delta u = f(x) \tag{1}$$

subject to zero Dirichlet boundary conditions. A standard second-order discretization of the above equation on a uniform grid with grid size $h = \frac{1}{n+1}$ gives rise to a linear system equation denoted by $Au = f$ where A, u and f correspond to the discrete Laplacian, the solution and the forcing functions respectively, and A is a tridiagonal matrix with diagonal elements $-\frac{1}{h^2}, \frac{2}{h^2}, -\frac{1}{h^2}$. It is well known that the matrix A can be diagonalized as

$$A = W\Lambda W^T$$

where W is an orthogonal matrix with elements

$$(W)_{ij} = 2\sqrt{h}\sin ij\pi h,$$

and

$$\Lambda = diag(\lambda_k)\ ,\ \lambda_k = \frac{4}{h^2}\sin^2\frac{k\pi h}{2}.$$

The main idea of the MF preconditioning is to approximate this eigendecomposition of A. First, the eigenfunctions of A are grouped into subsets corresponding to different frequency bands. In matrix form, for $n = 2^L - 1$, we partition W into L bands so that

$$W = [W_1, W_2, \cdots, W_L],$$

where

$$W_l = [w_{2^{l-1}}, \cdots, w_{2^l-1}].$$

with w_j being the jth column of the matrix W. Thus, for example, W_1 and W_L correspond to the lowest and highest frequency bands respectively.

Using the notations introduced above, we can rewrite

$$A = \sum_{l=1}^{L} W_l\Lambda_l W_l^T$$

where

$$\Lambda = diag(\Lambda_l), \Lambda_i = diag(\lambda_{2^{l-1}}, \cdots, \lambda_{2^l-1}).$$

The first approximation comes in when we replace all the eigenvalues (Λ_l) within each band by a constant c_l. Thus, we have a preconditioner $\hat{M}$ such that

$$\hat{M}^{-1}v = \sum_{l=1}^{L}\frac{B_l v}{c_l}$$

where

$$B_l = W_l W_l^T.$$

Note that we have the following property for B_l :

$$B_l v = \begin{cases} v & \text{if } v \in range\,\{W_l\} \\ 0 & \text{if } v \in range\,\{W_l\}^{\perp}. \end{cases}$$

Hence, B_l can be considered as an ideal spatial bandpass filter. Thus applying the preconditioner $\hat{M}$ to a vector (i.e. $M^{-1}v$) consists of three phases : projection of v into the subspace corresponding to each band (operator B_l), scaling by the corresponding approximate eigenvalues c_l, and synthesizing the scaled components (summation).

Since the implementation of ideal filters is computationally expensive requiring many global operations (e.g. sine transforms), we seek the approximation of ideal filters with nonideal ones which are computationally more efficient. For the construction of efficient nonideal filters, we borrow ideas from standard digital filtering theory [4]. Typically, a bandpass filter is constructed by taking the difference of two lowpass filters, one that filters out all frequencies higher than the highest ones in the band and the other one that lets through all frequencies lower than all frequencies in the band. In turn, the lowpass filters can be approximated by cascading a sequence of elementary filters H_l's, which are simple averaging operators over a small fixed number of grid points separated by spacing proportional to the wavelength of the band W_l.

Mathematically, the effect of using nonideal filters can be summarized by replacing B_l with approximations $\tilde{B}_l$ in the definition of $\hat{M}$ to get our final preconditioner M :

$$M^{-1}v = \sum_{l=1}^{L} \frac{\tilde{B}_l v}{c_l}.$$

In the rest of the paper, we use the following two filters :

- the first order filter defined by :

$$(H_{l,1})_j = \frac{1}{4}(v_{j-2^{L-l}} + 2v_j + v_{j+2^{L-l}})$$

where $(\cdot)_j$ denotes the jth element of the argument, and v is extended periodically by

$$v_{-j} = -v_j, \quad \text{and} \quad v_{n+j} = -v_{n+2-j}.$$

- the filter $H_{l,2}$ obtained by applying $H_{l,1}$ twice :

$$(H_{l,2})_j = \frac{1}{16}(v_{j-2^{L-l+1}} + 4v_{j-2^{L-l}} + 6v_j + 4v_{j+2^{L-l}} + v_{j+2^{L-l+1}}).$$

We call the method introduced above the single grid multilevel filtering (SGMF) preconditioner, which involves computation on the same number of grid points n at all levels (corresponding to the frequency bands). Since there are $L = \log_2(n+1)$ levels and $O(n)$ operations are required per level, the total number of operations required per iteration is thus $O(nL)$.

To further improve the efficiency, we introduce a multigrid version of our preconditioner which we called the multigrid multilevel filtering (MGMF) preconditioner. This is motivated by the fact that waveforms consisting only of low wavenumber components can be well represented on coarser grids. To incorporate the multigrid structure, the operators I_{l+1}^l and I_{l-1}^l, which are the down-sampling and up-sampling operators respectively, are introduced. Note that in the multigrid literatures these operators are commonly known as restriction and interpolation operators. Using the concept of MGMF, we construct a sequence of grids Ω_l of sizes $h_l = O(2^{L-l}h), 1 \leq l \leq L$, to represent the decomposed components. With MGMF, the total number of operations per iteration is $O(n)$, a reduction by a factor of $\log_2 n$ compared to SGMF.

We summarize the MGMF1 preconditioning algorithm as follows :

```
Algorithm MGMF1 : input = $r$, output = $z = M^{-1}r$
Decomposition :
        $v_L := r$
        for $l = L-1, \cdots, 1$
                $v_l := I_{l+1}^{l} H_{l+1,1} v_{l+1}$
        end for
Scaling :
        for $l = 1, \cdots, L$
                $w_l := v_l \div c_l$
        end for
Synthesis :
        $z_1 := w_1$
        for $l = 2, \cdots, L$
                $z_l := w_l + H_{l,1} I_{l-1}^{l} z_{l-1}$
        end for
        $z = z_L$
end MGMF1
```

As it stands, this definition of the preconditioner can be extended to higher dimensions, more general elliptic operators and finite element meshes, as long as we have appropriate definitions for the elementary filters H_l's, the restriction and interpolation operators I_{l+1}^{l} and I_l^{l+1}, and c_l's. For example, filters for the high dimensional cases can be constructed from the tensor product of 1D filters. Moreover, it is well known that the eigenvalues λ_k in the wavenumber band B_l behave like $O(h_l^{-2})$ for general second-order elliptic problems, where h_l denotes the grid spacing for level l [18]. Therefore a general rule for selecting the scaling constant c_l at grid level l is $c_l = O(h_l^{-2})$. For quasiuniform meshes with refinement factor of 2 (so that $h_l \approx 2h_{l+1}$), this leads to the recurrence relation $c_{l+1} = 4c_l$. In [4], we also show how to extend the definition of H_l to general finite element meshes.

The MF preconditioner is designed to capture the mesh-dependent spectral property of a discretized elliptic operator but not the variation of its coefficients. In order to take badly scaled variable coefficients into account, we use diagonal scaling [14]. 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. 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.

The SGMF preconditioner on uniform meshes can be easily analyzed exactly using Fourier methods and we shall compare these results with the experimental results. The MGMF preconditioner on uniform and quasiuniform grids can be analyzed using the same finite element analysis framework used by Bramble, Pasicak and Xu [2], since their multilevel nodal basis preconditioner can be interpreted as a special case of the MF preconditioner with a particular filter. Basically, their results show that the condition number of $\kappa(M^{-1}A)$ (using their multilevel nodal basis preconditioner) is $O(L)$ where $L \approx \log_2 n$. Our experimental results showed that both the SGMF and MGMF preconditioners have comparable performance in terms of iteration counts. Therefore, in our numerical experiments, we will mainly use the more efficient MGMF preconditioners.

On a uniform mesh there is an obvious connection of our multilevel filtering idea with wavelets [17, 19]. Wavelets are orthonormal basis functions for square-integrable functions and are defined on a multilevel structure. These basis functions have compact support in space and almost compact support in the Fourier domain. Thus, wavelets can be considered as efficient bandpass filters. We are exploring the use of wavelets in our multilevel filtering preconditioner framework.

2.2. Numerical Results. In this section, we present numerical results for two- and three-dimensional Poisson, variable coefficient and discontinuous coefficient problems to demonstrate the convergence behavior when MGMF preconditioning is applied. Three variations of the MGMF preconditioning are implemented :

MGMF1 the MGMF preconditioner with 9-point (27-point) filter for 2D (3D) problems. (i.e. $H_{l,1}$)

MGMF2 a modified version of MGMF in which the 9-point (27-point) filter is applied twice. (i.e. $H_{l,2}$)

MGMF3 another modified version of MGMF in which the 9-point (27-point) filter is applied once at the finest grid level (to give smaller amount of work compared to MGMF2) and twice at other grid levels (to achieve a convergence rate between MGMF1 and MGMF2 but close to MGMF2).

The preconditioning operation counts for 2D (3D) problems are $9N$, $27N$ and $15N$ ($9N$, $32N$ and $12N$) respectively for MGMF1, MGMF2 and MGMF3 where N is the number of unknowns. These operation counts include also the diagonal scaling.

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

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

$$-\Delta u = f, \Omega = [0,1]^2, \tag{2}$$

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

$$\frac{\partial}{\partial x}\left(e^{-xy}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(e^{xy}\frac{\partial u}{\partial y}\right) = f, \Omega = [0,1]^2, \tag{3}$$

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

$$\frac{\partial}{\partial x}\left(\rho(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\rho(x,y)\frac{\partial u}{\partial y}\right) = f, \Omega = [0,1]^2, \tag{4}$$

where

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

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

$$-\Delta u = f, \Omega = [0,1]^3, \tag{5}$$

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

$$\frac{\partial}{\partial x}\left(e^{-xyz}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(e^{xyz}\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(e^{-xyz}\frac{\partial u}{\partial z}\right) = f, \Omega = [0,1]^3, \tag{6}$$

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

$$\frac{\partial}{\partial x}\left(\rho(x,y,z)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\rho(x,y,z)\frac{\partial u}{\partial y}\right) + \frac{\partial}{\partial z}\left(\rho(x,y,z)\frac{\partial u}{\partial z}\right) = f, \Omega = [0,1]^3, \tag{7}$$

where

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

For comparison purposes, the hierarchical basis (HB) [1] [11] and multigrid (MG(k), where k is the number of pre- and post-smoothings used) preconditioners were also implemented. The operation counts per iteration for the HB and the MG(k) preconditioners are $7N$ and $26 + 32 \times k$ ($8N$ and $26 + 36 \times k$ for 3D) respectively. The number of iterations are shown in the following Tables 1 to 6 ('-' in Tables 4 and 5 means 'data not available'). For different test problems the k in MG(k) that gives the best overall operation count is shown.

The iteration counts shown in the tables do not reflect the overall operation counts for the preconditioners. In Tables 7 and 8 we also show the total operation count required per grid point for each preconditioner. (We show only the data for $n = 255$ and $n = 31$ for the 2D and 3D problems respectively).

TABLE 1

Iteration counts for Test Problem 1

n	*MGMF1*	*MGMF2*	*MGMF3*	*HB*	*MG(2)*
7	10	9	10	16	4
15	11	9	10	24	4
31	12	8	10	34	5
63	13	8	10	44	5
127	15	8	10	54	5
255	16	7	10	64	5

TABLE 2

Iteration counts for Test Problem 2

n	*MGMF1*	*MGMF2*	*MGMF3*	*HB*	*MG(1)*
7	13	12	13	18	7
15	17	14	16	27	8
31	22	17	19	36	10
63	26	18	22	46	12
127	30	20	24	56	13
255	33	21	26	67	15

TABLE 3

Iteration counts for Test Problem 3

n	*MGMF1*	*MGMF2*	*MGMF3*	*HB*	*MG(10)*
7	21	19	20	28	6
15	35	30	33	49	10
31	59	49	51	79	15
63	101	82	86	132	17
127	200	140	143	223	20
255	367	254	269	393	24

TABLE 4

Iteration counts for Test Problem 4

n	*MGMF1*	*MGMF2*	*MGMF3*	*HB*	*MG(2)*
7	11	8	11	20	5
15	13	8	10	30	5
31	13	8	10	45	6
63	14	7	10	70	-

TABLE 5
Iteration counts for Test Problem 5

n	*MGMF1*	*MGMF2*	*MGMF3*	*HB*	*MG(2)*
7	13	11	13	20	5
15	16	12	14	33	6
31	18	13	16	53	7
63	21	14	18	82	-

TABLE 6
Iteration counts for Test Problem 6

n	*MGMF1*	*MGMF2*	*MGMF3*	*HB*	*MG(10)*
7	24	21	24	43	8
15	46	38	41	96	15
31	95	71	74	229	20

TABLE 7
Operation counts per grid points for 2D problems (n=255)

test problem	*MGMF1*	*MGMF2*	*MGMF3*	*HB*	*MG*
1	448	328	340	1664	555
2	990	1008	936	1876	1185
3	11010	12192	9684	11604	8808

TABLE 8
Operation counts per grid points for 3D problems (n=31)

test problem	*MGMF1*	*MGMF2*	*MGMF3*	*HB*	*MG*
4	416	440	350	395	615
5	612	741	592	1749	861
6	3230	4047	2738	7557	8220

We can observe from the tables that filtering twice (MGMF2) always improves the convergence rates over MGMF1 but not the overall operation count. This observation was the main driving force for the design of MGMF3 and we see that MGMF3 requires less work per grid point than MGMF1 and MGMF2. Also, from the tables it appears that both MGMF2 and MGMF3 give condition number of $O(1)$ for Poisson problems in 2D and 3D, of $O(\log n)$ for variable coefficient problems and of $O(n)$ for discontinous coefficient problems. The HB preconditioner does not exhibit competitive performance both in terms of iteration count and operation count, especially for 3D problems, since the condition number behaves like $O(h^{-1})$ instead of $O(\log h^{-1})$ for 2D problems. The HB preconditioner should give better performance for problems with nonuniformly refined grids. The MG preconditioner gives the best convergence rates for all the test problems attempted. However, for smooth problems it performs worse than the MGMF preconditioners in operation counts mainly because of the expensive work spent in the relaxation steps. For discontinuous coefficient problems (e.g. test problems 3), the MG preconditioner sometimes gives better operation counts than the others when the number of relaxation steps is large enough (10 in our experiments). In our previous paper when we used 3 relaxation steps the operation count for problem 3 was found to be the worst of all.

3. MF Preconditioners for Anisotropic Problems. In this section, we extend the concept of multilevel filtering to the second-order anisotropic problems. To achieve high degree of efficiency, the preconditioning step requires some modifications in the design of filters. We first provide justification for such modifications and then we will show the condition number computed by Fourier analysis. Numerical experiments are also included.

Consider the following 2D second-order anisotropic problem :

$$-\alpha u_{xx} - u_{yy} = f(x,y) \text{ in } \Omega = [0,1]^2 \tag{8}$$

where $\alpha > 1$ and with zero Dirichlet boundary conditions. The discretization of the equation using uniform square mesh with $h = \frac{1}{n+1}$ gives a block-tridiagonal matrix A such that $Au = f$ where u is the solution. In the Fourier domain we can express this as :

$$\hat{A}(j,k)\hat{u}_{j,k} = \hat{f}_{j,k},\ j,k = 1,2,\cdots,n-1 \tag{9}$$

where

$$\hat{u}_{j,k} = \frac{\sqrt{2}}{n}\sum_{l=1}^{n}\sum_{m=1}^{n} u_{l,m}\sin(j\pi lh)\sin(k\pi mh) \tag{10}$$

and

$$\hat{f}_{j,k} = \frac{\sqrt{2}}{n}\sum_{l=1}^{n}\sum_{m=1}^{n} f_{l,m}\sin(j\pi lh)\sin(k\pi mh) \tag{11}$$

such that

$$(12) \qquad \hat{A}(j,k) = (2+2\alpha) - 2(\alpha \cos j\pi h + \cos k\pi h)$$

We can observe from the eigenvalue spectrum of $\hat{A}$ that for $\alpha \gg 1$ the variation in magnitudes of the eigenvalues in the k-direction is relatively small compared to that in the j-direction. To maintain uniform variation of eigenvalues within each band, we divide more wavenumber bands in the j-direction than in the k-direction. We call this technique ***directionally adaptive filtering.*** This can be done in practice by first performing 1D filtering in the j-direction for a number of levels (say number of levels = γ) and after that resuming 2D filtering. This is in contrast to performing 2D filtering for all the levels for the nearly isotropic problems described in the last section. Here γ depends on α as well as the problem to be solved. For second order elliptic problems with quasiuniform grid and $h_l \approx 2h_{l+1}$ it is sufficient to use $\gamma = round(\log_4 \alpha)$. Suppose $\alpha = 4$, then $\gamma = 1$ and the modified $H_{l,1}$ for the finest grid level takes the following stencil form :

$$\frac{1}{4}\left|\begin{matrix} 1 & 2 & 1 \end{matrix}\right|$$

while the filter for the other coarse grid levels have a 2D stencil (tensor product of 1D filter, i.e. $H_{l,1} \times H_{l,1}$).

Note that if the finest level is defined on a $(n+2) \times (n+2)$ grid, then for $\gamma \geq 1$ the next coarse level is defined on $(\frac{n+1}{2}+1) \times (n+2)$ grid instead of $(\frac{n+1}{2}+1) \times (\frac{n+1}{2}+1)$ grid for $\gamma = 0$. It should also be noted that this modified filtering scheme is analogous to the idea of semi-coarsening in the multigrid literatures.

We performed Fourier analysis of the single grid version of this scheme (called SGMF1a) on the 2D anisotropic problem with different α and h. The condition numbers of the preconditioned system are given in Table 9. For comparison purpose, the condition numbers of the preconditioned system using the unmodified SGMF1 preconditioner are also included. Table 9 shows that this modified scheme is quite effective. For example, for $\alpha = 1000$ the condition number grows slowly with n while this is not true for the unmodified SGMF1 preconditioner.

The MGMF1 preconditioning algorithm for the above anisotropic problems can be summarized as follows :

Table 9
condition number for different α and n

	$\alpha = 10$			$\alpha = 100$			$\alpha = 1000$		
n	A	SGMF1a	SGMF1	A	SGMF1a	SGMF1	A	SGMF1a	SGMF1
7	25	3.8	13	25	3.8	38	25	3.8	47
15	103	4.3	21	103	4.7	117	103	4.7	216
31	414	5.4	28	414	5.8	233	414	5.9	849
63	1659	6.6	34	1659	6.8	328	1659	6.9	2142
127	6639	8.2	40	6639	7.9	395	6639	8.0	3480
255	26560	9.7	46	26560	9.0	454	26560	9.0	4396

```
Algorithm MGMF1a : input = $r$, output = $z = M^{-1}r$
        $v_L := r$
Decomposition :
        $count = \gamma$
        for $l = L-1, \cdots, 1$
                if ($count = 0$) then
                        $t_l$ := x-filter1($v_{l+1}$)
                        $v_l$ := y-filter1($t_l$)
                else
                        $count = count - 1$
                        $v_l$ := x-filter1($v_{l+1}$)
                end if
        end for
Scaling :
        for $l = 1, \cdots, L$
                $v_l := v_l \div c_l$
        end for
Synthesis :
        $t_1 := v_1$
        for $l = 2, \cdots, L$
                $t_l := v_l + H_{l,1} I^l_{l-1} t_{l-1}$
        end for
        $z := t_L$
end MGMF1a
```

Next we show the numerical results using the multigrid MF (MGMF1a) preconditioner in conjunction with the conjugate gradient method. Again, we use the standard 5-point discretization on a uniform square mesh with $h = \frac{1}{n+1}$ and the forcing function $f(x,y)$ is such that the solution is $u = x(x-1)y(y-1)e^{xy}$. The stopping criterion used is $\| r^k \| / \| r^0 \| \leq 10^{-5}$ and the initial guess is 0. The iteration counts for different h

and α are shown in Table 10.

TABLE 10
Iteration counts for different α and n

	$\alpha = 10$			$\alpha = 100$			$\alpha = 1000$		
n	A	MGMF1a	MGMF1	A	MGMF1a	MGMF1	A	MGMF1a	MGMF1
7	23	11	18	17	7	19	13	6	20
15	48	13	26	41	10	41	27	9	44
31	97	15	32	90	12	64	63	12	84
63	197	16	36	187	13	83	126	13	140
127	405	19	41	388	15	95	258	15	193
255	839	20	45	812	17	106	608	17	224

The numerical results show that this scheme works very well for a wide range of α. It should be noted that a similar scheme can be applied to the case when $\alpha < 1$ and for the 3D anisotropic problems.

4. MF Preconditioners for Positive Definite Helmholtz Equation. Consider the following 2D Helmholtz equation :

$$(13) \qquad -\Delta u + \beta u = f \text{ in } \Omega = [0,1]^2$$

with zero Dirichlet boundary condition and when β is a positive or a small negative constant so that the discretization matrix A is symmetric and positive definite. (Most Helmholtz problems with negative β, however, give rise to symmetric but indefinite stiffness matrices. We plan to pursue this type of problems in the future). An effective MF preconditioner for this equation requires modifications in the scaling constants c_l's, as explained below. Again we can express $Au = f$ in Fourier domain with $\hat{A}$ as :

$$\hat{A}(j,k) = 4\sin^2(i\pi h) + 4\sin^2(j\pi h) + \beta h^2.$$

The spectrum of $\hat{A}$ differs from that of the Poisson equation by βh^2 and we need to incorporate this offset in the scaling constants $c_l's$. Instead of using $c_{l+1} = 4c_l$ for Poisson equation with $h_l \approx 2h_{l+1}$, The recurrence relation is now given by $c_l = \frac{c_{l+1}+3kh^2}{4}$ with $c_L = 8 + kh^2$ (let us call this scheme SGMF1b). To find the range of β such that all the eigenvalues are real and positive, we can first observe from the equation above that this is indeed the case when $\beta > 0$. And if $\beta < 0$, it is straightforward to find the lower bound of β as

$$\beta > \frac{-8\sin^2(\pi h/2)}{h^2}.$$

For the 2D Helmholtz equation with $n = 256$, this lower bound is about -19.7. In Table 11, we show the condition numbers of the preconditioned systems (SGMF1b) computed by Fourier analysis for a range of β and compare them with those of the unpreconditioned system (A). We also include the condition numbers when the unmodified SGMF1 preconditioner is used. The Fourier results show that for large β, the modifieid preconditioner SGMF1b improves the condition number significantly over SGMF1.

Table 11
condition number for different n and β

	$\beta = 10$			$\beta = 1000$			$\beta = -10$		
n	A	SGMF1b	SGMF1	A	SGMF1b	SGMF1	A	SGMF1b	SGMF1
7	17	1.97	2.18	1.5	2.60	12.5	51	3.36	3.75
15	69	2.61	2.85	3	3.70	31.0	209	4.19	4.84
31	275	3.36	3.60	9	4.63	49.3	839	5.16	6.14
63	1102	4.09	4.33	33	5.04	57.9	3363	6.10	7.38
127	4407	4.84	5.09	129	5.17	60.5	13456	7.06	8.67
255	17629	5.59	9.94	515	5.20	61.2	53831	8.02	9.94

We now show the numerical results using the multigrid formulation of the SGMF1b preconditioner (MGMF1b). Again, we use the standard 5-point discretization on a uniform square mesh with $h = \frac{1}{n+1}$ and the forcing function $f(x,y)$ is such that the solution is $u = x(x-1)y(y-1)e^{xy}$. The stopping criterion used is $\| r^k \| \,/\, \| r^0 \| \leq 10^{-5}$ and the initial guess is 0. The iteration counts for the unpreconditioned (A) and preconditioned CG methods (MGMF1b and MGMF1) with different n and β are shown in Table 12.

Table 12
Iteration counts for Helmholtz equation with different n and β

	$\beta = 10$			$\beta = 1000$			$\beta = -10$		
n	A	MGMF1b	MGMF1	A	MGMF1b	MGMF1	A	MGMF1b	MGMF1
7	15	10	10	4	6	8	18	11	11
15	32	11	11	6	8	11	38	12	12
31	66	12	12	10	10	16	77	13	13
63	133	13	13	22	11	18	156	14	14
127	273	15	15	48	12	19	316	15	15
255	555	16	16	101	13	20	643	17	17

Again, we can see that the numerical results agree with the results from Fourier analysis. For small β, the MGMF1b and MGMF1 requires almost the same number of iterations to achieve convergence. However, for large β, the advantage of using the MGMF1b becomes obvious.

5. MF Preconditioners for Convection-diffusion Equation. Consider the 2D convection-diffusion equation

$$k(x,y) \cdot \nabla u = \epsilon \, \Delta \, u + f(x,y) \text{ in } \Omega = [0,a] \times [0,b] \tag{14}$$

with Dirichlet boundary conditions on $\partial\Omega$. This equation, for example, describes the concentration of a chemical in a solution flowing with a time-independent velocity field $k(x,y)$. We examine the simple case when $k(x,y) = [k,0]^T$, $\epsilon = 1$ and $a = b = 1$ so that the equation becomes

$$- \Delta \, u + k u_x = f(x,y) \text{ in } \Omega = [0,1]^2.$$

The application of MF preconditioning to this convection-diffusion equation requires special handling because the discretization matrix A is nonsymmetric. In this section we only consider the cases where A is symmetrizable (i.e. there exists a diagonal matrix D such that DAD^{-1} is symmetric) and positive definite. In the following we describe two methods to handle the problem. The first is to symmetrize A before applying the PCG algorithm. The symmetrized system resembles the positive-definite Helmholtz equation and thus efficient MF preconditioners are known from the last section. The second is to convert the problem to the self-adjoint form and then apply the original MF preconditioning with diagonal scaling to it. We examine two discretization schemes for the first method and show that the MF preconditioner is effective for both schemes. Also numerical results show that both the first and second methods give good convergence behavior.

Method 1A Central Difference Scheme

This method on a square mesh of side h gives rise to :

$$- \Delta_h \, u_{i,j} + \frac{k}{2h}(u_{i+1,j} - u_{i-1,j}) = f_{i,j},$$

where $\Delta_h u_{i,j}$ is the standard 5-point finite difference discretization for the Poisson equation. This method has an accuracy of $O(h^2)$. However, to obtain stability, the mesh size h has to obey the following criterion [15] :

$$h \leq \frac{2}{k}.$$

Therefore, if k is large (which is typical for many applications), very small mesh size has to be used in order to maintain stability. However, too small a mesh size also means unnecessarily high operation count to arrive at the solution.

Method 1B HODIE Scheme [16]

The HODIE method for the above convection-diffusion equation has the following difference formula :

$$(2\tau+2)u_{i,j}-(\tau+\frac{kh}{2})u_{i-1,j}-(\tau-\frac{kh}{2})u_{i+1,j}-u_{i,j-1}-u_{i,j+1} = kh^2f(ih+\frac{(1-\tau)h}{k},jh)$$

where $\tau = \sqrt{1+\frac{k^2h^2}{3}}$.
An advantage of this method is that it has an accuracy of $O(h^2)$ and is also unconditionally stable.

Method 2 Transformation to Self-adjoint Form

The above convection-diffusion equation can easily be transformed into the following self-adjoint form

$$-\nabla\cdot(e^{-kx}\cdot\nabla u) = e^{-kx}f(x,y)$$

and then the standard 5-point finite difference approximation can be applied to obtain $O(h^2)$ accuracy. (However, it should be noted that if k is not a constant, it may not be able to transform such problems into self-adjoint forms)

The discretization matrices from the first method 1A and 1B above are nonsymmetric but symmetrizable when h is in the range of stability. We can symmetrize these matrices before applying preconditioned conjugate gradient methods. The symmetrized matrices in general are equivalent to the discretization of certain Helmholtz equations. Consequently, the MF-preconditioning techniques for the Helmholtz equation can be used here. Recall that for the Helmholtz equation, only the scaling constants need to be modified. Using the same technique, we can derive the scaling recurrence for these methods as :

$$c_l = \frac{c_{l+1}+3s}{4},$$

where s is different for different methods :

- Central difference :

$$s = \frac{2-2\sqrt{(1+\frac{kh}{2})(1-\frac{kh}{2})}}{4+4\sqrt{(1+\frac{kh}{2})(1-\frac{kh}{2})}}, c_L = 6+2\sqrt{(1+\frac{kh}{2})(1-\frac{kh}{2})}$$

- HODIE Method :

$$s = \frac{2\sqrt{1+\frac{k^2h^2}{3}}-2\sqrt{1+\frac{k^2h^2}{12}}}{4+4\sqrt{1+\frac{k^2h^2}{12}}}, c_L = 4+2\tau+2\sqrt{(\tau+\frac{h^3}{2})(\tau-\frac{h^3}{2})}$$

We use the following test problem :

$$- \Delta u + k u_x = f(x,y) \text{ in } \Omega = [0,1]^2$$

where $f(x,y) = -\pi_2(1 - \frac{e^{x/k}}{e^{1/k}}) \sin \pi y$ so that the solution is given by $u(x,y) = (1 - \frac{e^{x/k}}{e^{1/k}}) \sin \pi y$. The stopping criterion is $\| r^k \| / \| r^0 \| \leq 10^{-10}$ and zero initial guess is used. The iteration counts are given in Table 13 for $k = 30$ and different $h = \frac{1}{n+1}$.

TABLE 13
Iteration Counts for Convection Diffusion equation

n	$1A$	$1B$	2
15	26	21	21
31	27	25	24
63	27	26	26
127	27	27	27
255	27	27	27

We can observe from Table 13 that all methods used here require about the same number of iterations and the convergence rates seem to depend only slightly on n. However, it was observed that the self-adjoint form gives the best accuracy. However, when transformation to self-adjoint form is not possible, both the central difference and HODIE methods seem to give reasonable accuracy. The central difference method is both easy to use and reasonably accurate when the convective term is not too large. When the convective term is large and high accuracy is needed, then the HODIE method is quite promising.

6. MF Preconditioners for Biharmonic Equation. Consider the following biharmonic equation in 2D :

$$- \Delta^2 u = f \text{ in } \Omega = [0,1]^2 \tag{15}$$

with first boundary conditions :

$$u(x,y)|_\Gamma = 0 \tag{16}$$

and

$$\frac{\partial u}{\partial n} = 0 \tag{17}$$

We discretize this equation using 13-point second-order centered finite difference approximation with $h = \frac{1}{n+1}$:

$$\begin{aligned} 20u_{i,j} &- 8(u_{i+1,j}+u_{i-1,j}+u_{i,j+1}+u_{i,j-1}) \\ &+ 2(u_{i+1,j+1}+u_{i-1,j+1}+u_{i+1,j-1}+u_{i-1,j-1}) \\ &+ u_{i+2,j}+u_{i-2,j}+u_{i,j+2}+u_{i,j-2} = h^4 f_i,j \end{aligned}$$

for $i,j = 2, n-1$. If we let the boundary condition be $u = g(x,y)$ and its first derivative be $b(x,y)$. Then the difference equation for $i = 1$, and $j = 3, \cdots, n-2$ is :

$$\begin{aligned} 21u_{1,j} &- 8(u_{2,j}+u_{1,j+1}+u_{1,j-1})+2(u_{2,j+1}+u_{2,j-1})+u_{3,j}+u_{1,j+2}+u_{1,j-2} \\ &= h^4(f_i,j+8g_{0,j}-2(g_{0,j+1}+g_{0,j-1})-2hb_{0,j}) \end{aligned}$$

since

$$\frac{\partial u}{\partial n} = -\frac{\partial u}{\partial x} \text{ on } x = 0,$$

and using central differencing, we get

$$-\frac{(u_{1,j}-u_{-1,j})}{2h} = b_{0,j}.$$

Also, at $i = j = 1$, we have

$$\begin{aligned} 22u_{1,1} &- 8(u_{2,1}+u_{1,2})+2(u_{2,2})+u_{3,j}+u_{1,3} \\ &= h^4(f_i,j+8(g_{0,j}+g_{1,0})-2(g_{0,j+1}+g_{0,j-1}+g_{2,0})-2h(b_{0,1}+b_{1,0})). \end{aligned}$$

The difference equations for other near boundary grid points can be derived similarly.

The eigenvalue spectrum of $\hat{A}$ can be approximated by :

$$\hat{A}(j,k) = (4-2(\cos(i\pi h)+\cos(j\pi h))^2 \tag{18}$$

which is the square of that of the Poisson equation.

Since the eigenvalues in B_l for this equation behave like $O(h_l^{-4})$, a natural extension of the MF preconditioner involves changing the scaling recurrence $c_{l+1} = 4c_l$ to $c_{l+1} = 16c_l$ (again, $h_l \approx 2h_{l+1}$ is assumed). In Table 14, we show the result of the Fourier analysis on the MF-preconditioned biharmonic equation. In the table, SGMF1c, SGMF2c and SGMF3c represent the original SGMF1, SGMF2 and SGMF3 preconditioners with the new scaling.

We see that the condition number of A bgrows about 16 times with each halving of h. The use of SGMF1c has effectively helped to reduce the condition number. Nevertheless, SGMF2c helps to reduce the condition number even more dramatically.

To verify the Fourier results, we implement the SGMF1c, SGMF2c and SGMF3c preconditioners for the Biharmonic equation where the $f(x,y)$, $g(x,y)$ and $b(x,y)$ are such that the solution is $u = x(x-1)y(y-1)\sin(\pi x)\sin(\pi y)$. The stopping criterion

TABLE 14
Condition number for SGMF preconditioning for biharmonic equation

n	*No preconditioning*	*SGMF1c*	*SGMF2c*	*SGMF3c*
7	690	25	5.3	17
15	1.1×10^4	108	5.6	66
31	1.7×10^5	438	7.2	256
63	2.8×10^6	1814	8.7	1017
127	4.4×10^7	7367	10.2	4061
255	7.0×10^8	29705	11.7	16238

TABLE 15
Iteration Counts for SGMF-preconditioned PCG for biharmonic equation

n	*No preconditioning*	*SGMF1c*	*SGMF2c*	*SGMF3c*
7	10	9	10	9
15	42	17	12	16
31	160	36	14	30
63	586	82	17	57
127	2218	177	23	113
255	8587	366	33	220

TABLE 16
Iteration Counts for MGMF-preconditioned PCG for biharmonic equation

n	*No preconditioning*	*MGMF1c*	*MGMF2c*	*MGMF3c*
7	10	10	10	10
15	42	27	22	24
31	160	40	29	32
63	586	56	30	37
127	2218	80	35	40
255	8587	120	43	48

is $\| r^k \| / \| r^0 \| \leq 10^{-6}$ and the initial guess is zero. The iteration counts are shown in Table 15.

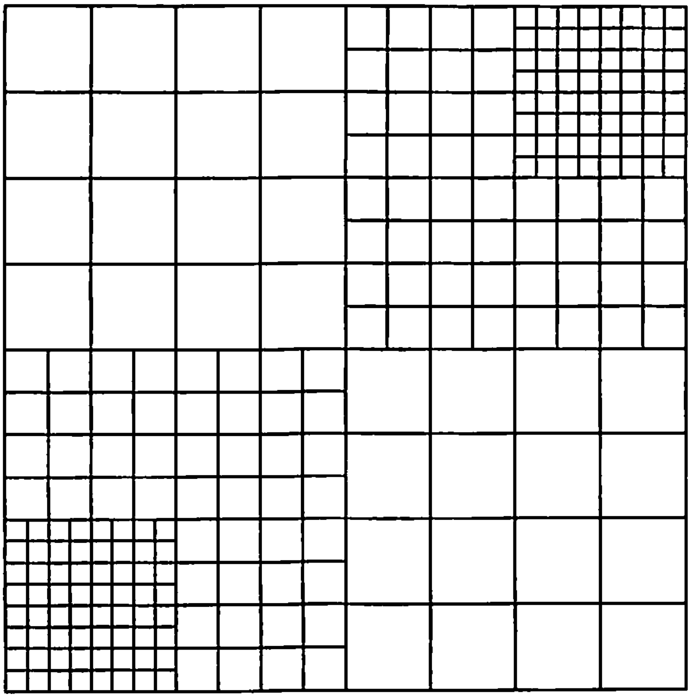
Next we show (in Table 16) the iteration counts when the multigrid formulation of SGMF1c, SGMF2c and SGMF3c (i.e. MGMF1c, MGMF2c and MGMF3c) are applied to the same problem.

We observe a close correlation between the numerical and Fourier results for the SGMF preconditioners. Indeed, SGMF2c improves significantly over SGMF1c with only a little increase in cost per iteration. SGMF3c improves somewhat over SGMF1c but is still not good enough compared to SGMF2c. Therefore, SGMF2c requires the least operation counts out of the three. Looking into the numerical results for the MGMF preconditioners, we first observe that both MGMF1c and MGMF3c give better convergence rates than their SGMF counterparts. We cannot explain why this is the case, nor can we explain why MGMF3c performs much better than predicted by the corresponding Fourier results. Finally, with a little arithmetic, it is not difficult to show that MGMF3c gives the least overall operation counts.

7. MF Preconditioners for Problems with Locally Refined Grids. In this section, we shall consider the application of the MF preconditioners to second-order elliptic problems with local mesh refinement. Such mesh refinements are necessary for accurate modeling of problems with various type of singular behavior. We consider the discretization scheme for locally mesh refined grids by McCormick and Thomas [12]. This discretization scheme was motivated by the desire to preserve the highly regular grid structure (to maintain efficiency on parallel computer architectures) as well as to satisfy the need for local resolution in many physical models. For example, the mesh in Fig. 1 would be effective if the forcing function $f(x,y)$ behaves like a δ function distribution at the points $(1,1)$ and (n,n) (both lower left and upper right corners).

The Fourier analysis cannot be applied here because of the presence of nonuniform grids. However, as was shown in our previous paper [4], the parallel multilevel preconditioner proposed by Bramble, Pasciak and Xu [2] can be considered as a special case of MF preconditioners with appropriately chosen filters. We can borrow the finite element analysis result from them and we would expect the MGMF preconditioners to be effective also for meshes with local refinement. Below we show the MGMF algorithm for this problem. Here $\hat{I}_i^j$ and $\hat{H}_l$ are restriction (or interpolation) and elementary filtering operators restricted to the locally refined grids only. Moreover, we can use the same recurrence relation $c_l = 4c_{l+1}$ and we have the following algorithm:

FIG. 1. *Locally Refined Grids - Example 2*



```
**Algorithm** MGMF1d : input = $r$, output = $z = M^{-1}r$
Decomposition :
    $v_L := r$
    (* filtering at refined levels *)
    **for** $l = L - 1, \cdots, J - k$
        $v_l := \hat{I}^l_{l+1}\hat{H}_{l+1,1}v_{l+1}$
    **end for**
    (* filtering on uniform grid levels *)
    **for** $l = L - k - 1, \cdots, 1$
        $v_l := I^l_{l+1}H_{l+1,1}v_{l+1}$
    **end for**
Scaling :
    **for** $l = 1, \cdots, L$
        $v_l := v_l \div c_l$
    **end for**
Synthesis :
    $z_1 := v_1$
    **for** $l = 2, \cdots, L - k$
        $z_l := v_l + H_{l,1}I^l_{l-1}z_{l-1}$
    **end for**
    **for** $l = L - k + 1, \cdots, L$
        $z_l := v_l + \hat{H}_{l,1}\hat{I}^l_{l-1}z_{l-1}$
    **end for**
    $z = z_L$
**end** MGMF1d
```

We solve a Poisson equation on the grid

- shown in Fig. 1 but with refinement only at the upper right corner and the forcing function is $f(x,y) = 2^{-l}\delta(1-h, 1-h)$, and
- shown in Fig. 1 and the forcing function is $f(x,y) = 2^{-l}(\delta(h,h)+\delta(1-h,1-h))$ where l is the number of level of refinements used and h is the grid size for the nonrefined grid.

We use the discretization scheme for the domain and the interfaces proposed by McCormick [12] for aligned grid. The stopping criterion and initial guess are the same as before. The iteration counts for different number of levels and different h are given in Table 17 and 18. The iteration counts for unpreconditioned CG method and the parallel multilevel preconditioner (BPX) [2] are also included for comparison purpose.

TABLE 17
Iteration Counts for Poisson equation with refinements at upper right corner only

n	no. of levels	*CG*	*MGMF1*	*BPX*
15	0	26	9	12
15	1	37	10	14
15	2	45	11	16
15	3	53	12	17
31	0	48	9	13
31	1	70	10	15
31	2	88	11	17
31	3	109	12	18
63	0	84	10	14
63	1	126	11	15
63	2	166	11	17
63	3	210	12	19
127	0	133	10	14
127	1	219	11	15
127	2	309	12	17
127	3	395	13	19

The tables show the effectiveness of the MF preconditioner compared to the unpreconditioned CG method and the PCG method with parallel multilevel preconditioner. The convergence rates seem to be quite insensitive to the number of refinement levels used.

8. MF Preconditioners for Schur Complement Systems. Consider solving a 2D second-order elliptic problem on a domain divided into 2 subdomains by an interface. If we use a 5-point discretization and order unknowns in the the subdomains Ω_1 and

TABLE 18

Iteration Counts for Poisson equation with refinements at both corners

n	no. of levels	*CG*	*MGMF1*	*BPX*
15	0	26	9	12
15	1	54	11	15
15	2	63	12	17
15	3	75	16	18
31	0	48	9	13
31	1	86	11	16
31	2	117	13	17
31	3	140	13	19
63	0	84	10	14
63	1	126	12	16
63	2	190	12	18
63	3	235	14	19
127	0	133	10	14
127	1	204	12	16
127	2	297	13	18
127	3	391	14	20

Ω_2 first followed by those on the interface Γ_3, we obtain the following linear system:

$$Au = \begin{bmatrix} A_1 & 0 & A_{13} \\ 0 & A_2 & A_{23} \\ A_{31} & A_{32} & A_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

By applying block Gaussian elimination to eliminate the unknowns u_1 and u_2, we obtain the following system for the interface unknowns u_3 :

$$Su_3 = \tilde{f}_3$$

where

$$S = A_3 - A_{31}A_1^{-1}A_{13} - A_{32}A_2^{-1}A_{23}$$

and

$$\tilde{f}_3 = f_3 - A_{31}A_1^{-1}f_1 - A_{32}A_2^{-1}f_2.$$

A standard approach in domain decomposition methods is to solve the Schur complement system $Su_3 = \tilde{f}_3$ with the preconditioned conjugate gradient method. Many preconditioners have been proposed in the literature [20]. A typical one is Dryja's preconditioner [22], which is defined to be the square root of the negative one-dimensional Laplacian and which can be inverted by the use of FFTs in $O(n \log n)$ time where n is the number of unknowns on the interface. Recently, Smith and Widlund [21] proposed a hierarchical basis preconditioner for S which is cheaper than Dryja's preconditioner, requiring only $O(n)$ work per iteration. Here we propose to use the MF preconditioner for S. To do this, we can retain the multilevel filtering framework and we only need to modify the scaling constants c_l's. We know that the eigenvalues for the Schur complement in the frequency band B_l behaves like $O(h_l^{-1})$ [22]. Therefore, it is sufficient to use the recurrence $c_{l+1} = 2c_l$. In Table 19 we compare the number of iterations to obtain convergence for different n for the Poisson equation on a rectangular $2n \times n$ grid decomposed into two equal subdomains.

TABLE 19
iteration count versus n

n	No precond	Dryja	MGMF1	MGMF2	HB
7	4	4	4	4	4
15	8	6	7	6	7
31	16	6	9	7	8
63	27	6	9	7	10
127	39	6	9	7	12

We observe that MGMF2 performs better than MGMF1 and the hierarchical basis (HB) preconditioner. All but the HB preconditioner show convergence rates independent of n. Moreover, MGMF2 performs almost as well as Dryja's preconditioner. The MGMF appears to offer convergence rate comparable to Dryja's preconditioner and at the same time is relatively easy to use and costs about the same as the HB preconditioner.

9. Conclusion.

In our previous paper [4] and the first part of the present paper we show the competitiveness of the MF preconditioners compared with some other preconditioners such as the hierarchical basis preconditioner, multigrid preconditioner and others. In this paper we have further demonstrated the ease with which we can extend the MF preconditioners to effectively solve other more general elliptic problems. The flexibility of filter and scaling block design offers different ways of achieving high degree of efficiency for these problems.

REFERENCES

[1] R. E. Bank, T. F. Dupont and H. Yserentant, *The hierarchical basis multigrid method,* Numer. Math. 52, pp. 427-458, 1988.

[2] J. H. Bramble, J. E. Pasciak and J. Xu, *Parallel multilevel preconditioners,* To appear in Math. Comp.

[3] T. F. Chan, Jay C.C. Kuo and C. H. Tong, *Parallel elliptic preconditioners: Fourier analysis and performance on the Connection Machine,* Computer Physics Communications 53, pp. 237-252, 1989.

[4] C.-C. J. Kuo, T. F. Chan and Charles Tong, *Multilevel Filtering Elliptic Preconditioners,* UCLA CAM Report 89-23, August 1989. To appear in SIAM J. Matrix Analysis.

[5] Charles H. Tong, *The Preconditioned Conjugate Gradient Method on the Connection Machine,* International Journal of High Speed Computing, 2nd Issue : Scientific Applications of the Connection Machine, 1989.

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

[7] O. Axelsson and P. Vassilevski, *Algebraic multilevel preconditioning methods, I,* Report 8811, Department of Mathematics, Catholic University, The Netherlands, 1988.

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

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

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

[11] H. Yserentant, *On the multi-level splitting of finite element spaces,* Numer. Math. 49, pp.379-412, 1986.

[12] S. McCormick and J. Thomas, *The Fast Adaptive Composite Grid (FAC) Method for of Elliptic Equations,* Mathematics of Computation, Vol. 46, No. 174, April 1986, pp. 439-456

[13] S. McCormick, *Multilevel Adaptive Methods for Partial Differential Equations.*

[14] A. Greenbaum, C. Li and H. Z. Chao, *Parallelizing preconditioned conjugate gradient algorithms,* Computer Physics Communications, vol. 53, pp. 295-309, 1989.

[15] G. Birkhoff and Lynch, *Numerical Solutions for Elliptic Problems,* SIAM, Philadelphia, 1984.

[16] R. E. Lynch and J. R. Rice, *A high-order difference method for differential equations,* Math. Comp., 34 (1980), pp. 333-372.

[17] G. Strang and G. J. Fix, *Wavelets and Dilation Equations : A brief Introduction,* SIAM Review, December 1989, pp. 614-627.

[18] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method,* Prentice-Hall series in automatic computation, 1973,

[19] V. E. Henson, W. L. Briggs, *Wavelets: What are they and what do they have to do with Multigrid?* A presentation given at the Copper Mountain Conference on Iterative Methods, April 2-5, 1990.

[20] *Domain Decomposition Methods for Partial Differential Equations,* edited by T. F. Chan, R. Glowinski, J. Periaux, O. B. Widlund, SIAM, Philadelphia, 1989.

[21] B. Smith and O. Widlund, *A domain Decomposition Algorithm based on a change to a hierarchical basis,* submitted to SIAM J. Sci. Stat. Comput.

[22] M. Dryja, *A capacitance matrix method for Dirichlet problem on polygon region,* Numer. Math., 39, 1982, pp. 51-54.