Electronic Transactions on Numerical Analysis.
Volume 6, pp. 97-105, December 1997.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

SEMICOARSENING MULTGRID FOR SYSTEMS*

J. E. DENDY, JR.†

Abstract. Previously we examined the black box multigrid approach to systems of equations. The approach was a direct extension of the methodology used for scalar equations; that is, interpolation and residual weighting were operator induced, and coarsening employed a Galerkin strategy. The application was to standard coarsening of the unknowns. In this paper we consider a semicoarsening approach and find that there are a few differences in what is generally effective.

Key words. multigrid, parallel computation.

AMS subject classifications. 65N55, 65Y05.

1. Introduction. In [3] we extended to systems of equations the ideas contained in [1] and [2]. More specifically, let us consider multigrid with standard coarsening on a rectangular grid of points; that is, the coarse grid offspring of the grid $G^M = \{x_{i,j} : i = 1, \ldots, m;\ j = 1, \ldots, n\}$ is the grid $G^{M-1} = \{x_{2i-1,2j-1} : i = 1, \ldots, \lceil m/2 \rceil;\ j = 1, \ldots, \lceil n/2 \rceil\}$. And let us consider the system

$$LU = F,$$

i.e.,

$$\sum_{j=1}^{p} L_{ij} U^j = F^i, i = 1, \ldots, p, \tag{1.1}$$

and its discrete approximation on grid G^M:

$$L^M U^M = F^M,$$

i.e.,

$$\sum_{j=1}^{p} L_{ij}^M (U^j)^M = (F^i)^M, i = 1, \ldots, p.$$

We assume that each $(U^j)^M$ is defined on G^M. We also assume that $detL = det(L_{ij}) \neq 0$ and that $detL^M = det(L_{ij}^M) \neq 0$. Let interpolation be denoted by $I_{M-1}^M : (G^{M-1})^p \to (G^M)^p$, where the notation $(G^k)^p$ means

$$\underset{p\ times}{G^k \times \cdots \times G^k}.$$

* Received May 2, 1997. Accepted September 9, 1997. Communicated by J. Jones. This work was performed under the auspices of the U.S. Department of Energy under contract W-7405-ENG-36 and was supported by the Office of Scientific Computing of the Department of Energy under Contract No. KC-07-01-01.

† Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545 U.S.A, (jed@lanl.gov)

ETNA
Kent State University
etna@mcs.kent.edu

And let residual weighting be denoted by $I_M^{M-1} : (G^M)^p \rightarrow (G^{M-1})^p$. Then a two level method is given by:
1. Perform ν_1 smoothing iterations on $L^M u^M = F^M$.
2. Solve $L^{M-1}V^{M-1} = f^{M-1} \equiv I_M^{M-1}(F^M - L^M u^M)$, directly.
3. Perform $u^M \leftarrow u^M + I_{M-1}^M V^{M-1}$.
4. Perform ν_2 smoothing iterations on $L^M u^M = F^M$.
Recursion yields a multigrid method, specifically one V-cycle of a multigrid method. That is, step 2 can be replaced by the two level method on G^{M-1}, etc.; eventually M levels are employed, where M is chosen by the constraint that direct solution on G^1 is inexpensive.

There are several details that need to be prescribed. The smoothing in steps 1 and 4 above was taken in [3] to be collective point Gauss-Seidel with lexicographic ordering. That is, G^k is swept in lexicographic order with U^k being updated at $q \in G^k$ so that the residual is zero at q. This process requires the solution of a $p \times p$ system at q. As in the case of $p = 1$, collective point Gauss-Seidel gives an acceptable smoothing factor unless more than mild anisotropies are present. In the case of general anisotropies, alternating collective line Gauss-Seidel would be needed; this possibility was not investigated in [3].

In [3] and here, we restrict attention to operators with templates of the form

$$\begin{pmatrix} NW & N & NE \\ W & C & E \\ SW & S & SE \end{pmatrix}, \tag{1.2}$$

where NW, N, NE, W, C, E, SW, S, and SE are all $p \times p$ matrices. If (1.2) gives the template of the operator at (k, l), then C is the matrix relating $U_{k,l}^n$, $n = 1, \ldots, p$, and W is the matrix relating $U_{k,l}^n$, $n = 1, \ldots, p$ to $U_{k-1,l}^n$, $n = 1, \ldots, p$, etc. There should be no confusion with the notation in (1.1), where each $L_{i,j}$ operates on U^j.

For a brief description of the derivation of I_{k-1}^k, we temporarily assume symmetry of L^k. For fine grid points coinciding with coarse grid points, I_{M-1}^M is just the identity. For a fine grid point (if, jf) horizontally between two coarse grid points (ic, jc) and $(ic+1, jc)$,

$$(C - S - N)(I_{k-1}^k v^{k-1})_{if+1,jf} = (NW + W + SW)v_{ic,jc}^{k-1} + (NE + E + SE)v_{ic+1,jc}^{k-1},$$

where NW,N, etc. are evaluated at $(if + 1, jf)$ and where it is assumed that $C - S - N$ is invertible. A similar formula holds for fine grid points vertically between two coarse grid points. Then there is enough information to use the operator to express fine grid points in the center of four coarse grid points in terms of these four coarse grid points. Under the assumption of symmetry, we can take $I_k^{k-1} = (I_{k-1}^k)^*$. Finally L^{k-1} is defined by $L^{k-1} = I_k^{k-1} L^k I_{k-1}^k$.

For ease of exposition, let us denote I_{k-1}^k by I with block entries I_{ij}, so that

$$v_i^k = \sum_j I_{ij} v_j^{k-1}. \tag{1.3}$$

In [3] it is shown that in the constant coefficient case, $I_{ij} = 0$ for $i \neq j$. Thus for smooth coefficient problems, one would expect I_{ij}, $i \neq j$, to be small. Thus an alternative which is explored in [3] is to ignore L_{ij}^k, $i \neq j$, reducing the derivation of operator induced interpolation to the scalar case, in terms of the operators L_{ii}, $i = 1, \ldots, p$. Obviously, there are immediate counterexamples to the well-posedness of this procedure. For example if $p = 2$

ETNA
Kent State University
etna@mcs.kent.edu

and $L^M_{ij} = 0$ for $i \neq j$, then by interchange of block rows, the system can be rewritten so that $\tilde{L}^M_{ii} = 0$, $i = 1, 2$. Such obvious counterexamples aside, there is a numerical example in [3] which indicates that it is marginally better to avoid this latter procedure of basing interpolation on scalar blocks, rather than on the whole system.

In practice, isotropic operators seldom appear. Either there are inherent anisotropies in the physical system, or gridding effects introduce them. Because of the necessity of alternating collective line Gauss-Seidel for standard coarsening in the presence of anisotropies, it seems natural to consider the possibility of a semicoarsening procedure. Another reason for considering semicoarsening is that the computation to form $L^{k-1} = I^{k-1}_k L^k I^k_{k-1}$ is considerably simplified. In §2, we introduce some semicoarsening algorithms, and in §3, we give some numerical examples.

2. Semicoarsening. In semicoarsening multigrid procedures, the grid is coarsened in just one direction, which we choose to be y. Thus, the coarse grid offspring of a grid $\{x_{i,j} : i = 1, \ldots, m;\ j = 1, \ldots, n\}$ is the grid $\{x_{i,2j-1} : i = 1, \ldots, m;\ j = 1, \ldots, \lceil n/2 \rceil\}$. The robustness of line relaxation coupled with semicoarsening for constant coefficient, anisotropic, scalar problems was first reported in [9]. For scalar problems with anisotropic and discontinuous coefficients, a semicoarsening method was considered in [5] for three-dimensional scalar problems. The two-dimensional analogue of this method is considered in [[4]] and [[8]]. Both of these papers use a technique due to Schaffer [[7]]; without this technique, the semicoarsening method would not be competitive.

To simplify the exposition, we describe this technique for symmetric scalar equations, p = 1. For odd lines of G^k, I^k_{k-1} is just the identity. For even lines, let

$$A^- V^- + A^0 V^0 + A^+ V^+ = 0$$

be the equation that would give the row $V^0 = (V_{i,j} : i = 1, \cdots, M)$ in terms of the rows $V^- = (V_{i,j-1} : i = 1, \cdots, M)$ and $V^+ = (V_{i,j+1} : i = 1, \cdots, M)$, for j even. Here A^-, A^0, and A^+ are all tridiagonal matrices;

$$
\begin{aligned}
A^- &= tridiag(SW\ S\ SE),\\
A^0 &= tridiag(W\ C\ E), \qquad (2.1)\\
\text{and } A^+ &= tridiag(NW\ N\ NE).
\end{aligned}
$$

Then

$$V^0 = -(A^0)^{-1}(A^- V^- + A^+ V^+). \qquad (2.2)$$

Unfortunately, use of (2.2) yields a nonsparse interpolation, leading to nonsparse coarse grid operators. Schaffer's idea [7] is to assume that $-(A^0)^{-1}A^-$ and $(-A^0)^{-1}A^+$ can each be approximated by diagonal matrices ini the sense that B^- and B^+ are diagonal matrices such that

$$-A^0 B^- e = A^- e \ \text{ and } \ -A^0 B^+ e = A^+ e, \qquad (2.3)$$

where e is the vector $(1, \cdots, 1)^T$. To find B^- and B^+ requires just two tridiagonal solves. The interpolation formula is

ETNA
Kent State University
etna@mcs.kent.edu

$$V^0 = B^- V^- + B^+ V^+.$$

The case for symmetric systems, $p > 1$, is the same, except that now B^- and B^+ are block diagonal matrices — i.e., B^-_{ij} and B^+_{ij} are diagonal — and A^0, A^-, A^+ are block tridiagonal. Thus (2.3) no longer gives enough information to solve for B^- and B^+. One way to get enough information is to require

$$-A^0 B^- e^j = A^- e^j \text{ and } -A^0 B^+ = A^+ e^j, j = 1, \ldots, p,$$

where $e^j = (\delta_{1j}, \ldots, \delta_{pj})^T$, where δ_{ij} is a vector Kronecker delta; i.e., δ_{ij} is the zero vector if $i \neq 0$ and δ_{ij} is the vector of all 1's if $i = j$. The unknowns can be ordered so that A^0 has $2p$ nonzero diagonals.

For symmetric systems I_k^{k-1} can be taken to be $(I_{k-1}^k)^*$ and $A^{k-1} = I_k^{k-1} A^k I_{k-1}^k$. For nonsymmetric systems, following the ideas in [2] leads to forming I_{k-1}^k by redefining A^-, A^0, and A^+ as

$$\begin{aligned} A^- &= blocktridiag(symm(SW)\ symm(S)\ symm(SE)), \\ A^0 &= blocktridiag(W\ C\ E), \\ \text{and} \quad A^+ &= blocktridiag(symm(NW)\ symm(N)\ symm(NE)), \end{aligned} \tag{2.4}$$

where $symm(G) = \frac{1}{2}(G + G^*)$. A more natural choice for A^0, perhaps, is

$$A^0 = blocktridiag(symm(W)\ symm(C)\ symm(E)),$$

but experimentally this choice gives no better results than the above, and the above choice has the advantage of having to compute and store only once the banded LU decomposition of A^0, which is also needed to perform relaxation. Both choices reduce to A^0 in (2.1) when A is symmetric. I_k^{k-1} is formed as $(J_{k-1}^k)^*$, where $(J_{k-1}^k)^*$ is formed from

$$\begin{aligned} A^- &= (blocktridiag(SW\ S\ SE))^*, \\ A^0 &= (blocktridiag(W\ C\ E))^*, \\ \text{and} \quad A^+ &= (blocktridiag(NW\ N\ NE))^*. \end{aligned} \tag{2.5}$$

Again $A^{k-1} = I_k^{k-1} A^k I_{k-1}^k$.

In the above, it may be asked why the symmetric parts of the blocks are used instead of the true symmetric parts. Consider the case $p = 2$, and suppose that $A_{11} = A_{22} = 0$. Then using the true symmetric parts yields

$$\begin{aligned} symm(A^0_{12}) B^{\pm}_{11} &= symm(A^{\pm}_{12}), \\ symm(A^0_{21}) B^{\pm}_{22} &= symm(A^{\pm}_{21}), \\ B^{\pm}_{12} = B^{\pm}_{21} &= 0. \end{aligned}$$

Then the AI part of the coarse grid operator is

$$\begin{pmatrix} 0 & A_{12} \\ A_{21} & 0 \end{pmatrix} \begin{pmatrix} I_{12} & 0 \\ 0 & I_{21} \end{pmatrix} = \begin{pmatrix} 0 & A_{12} I_{21} \\ A_{21} I_{12} & 0 \end{pmatrix},$$

clearly wrong, since this system is a set of decoupled scalar equations, and this methodology leads to the dependence of the coarse grid operator for A_{12} on an interpolation operator induced by A_{21}. But (2.5) yields

$$\begin{aligned} A_{21}^0 B_{11}^{\pm} &= symm(A_{21}^{\pm}) \\ A_{12}^0 B_{22}^{\pm} &= symm(A_{12}^{\pm}) \\ B_{12}^{\pm} = B_{21}^{\pm} &= 0, \end{aligned}$$

and the AI part of the coarse grid operator is

$$\begin{pmatrix} 0 & A_{12}I_{12} \\ A_{21}I_{21} & 0 \end{pmatrix}.$$

A similar argument shows that the residual weighting operator needs to be

$$\begin{pmatrix} I_{12}^* & 0 \\ 0 & I_{21}^* \end{pmatrix}$$

and that to achieve that goal, (2.5) may be used. It is curious, however, that these heuristics suggest using the symmetric part of the blocks instead of the true symmetric part in the case of deriving I_{k-1}^k, whereas in the case of deriving I_k^{k-1} they suggest using the true transpose instead of the transposes of the blocks. We note that the factorization provided by the LINPACK routine used to factor the band matrix A^0 in (2.4) also provides a factorization of A^0 in (2.5), since one matrix is the transpose of the other.

Using the notation of (1.3), we also consider ignoring the nondiagonal components of I_{k-1}^k. That is we consider replacing (2.4) by

$$\begin{aligned} A^- &= blockdiag(symm(SW)\ symm(S)\ symm(SE)), \\ A^0 &= blockdiag(W\ C\ E), \\ \text{and } A^+ &= blockdiag(symm(NW)\ symm(N)\ symm(NE)), \end{aligned} \tag{2.6}$$

and (2.5) by

$$\begin{aligned} A^- &= (blockdiag(SW\ S\ SE))^*, \\ A^0 &= (blockdiag(W\ C\ E))^*, \\ \text{and } A^+ &= (blockdiag(NW\ N\ NE))^*. \end{aligned} \tag{2.7}$$

Again $A^{k-1} = I_k^{k-1} A^k I_{k-1}^k$. This algorithm assumes that the system can, and has, been written in a form in which the block diagonal is nonsingular.

3. Numerical Examples. The first example is the biharmonic equation written as a system:

$$\begin{cases} -\triangle U^1 = F \text{ in } \Omega = (0,1)\times(0,1), \\ U^1 - \triangle U^2 = 0 \text{ in } \Omega, \\ U^2 = 0 \text{ on } \partial\Omega, \\ \frac{\partial U^2}{\partial \nu} = 0 \text{ on } \partial\Omega, \end{cases} \tag{3.1}$$

where F is chosen so that $U^2(x,y) = sin^2\pi x sin^2\pi y$. These boundary conditions are more realistic and harder to solve than specifying U^1 and U^2 on the boundary; both boundary conditions were considered in [3]. (3.1) can be discretized as follows [6]:

ETNA
Kent State University
etna@mcs.kent.edu

$$\begin{cases} -\triangle_0^h + M^h U^2 = F \text{ on } \Omega_h = (h, \ldots, (N-1)h) \times (h, \ldots, (N-1)h), \\ U^1 - \triangle_0^h U^2 = 0 \text{ on } \Omega_h, \end{cases}$$

where $h = \frac{1}{N}$, $\triangle_0^h$ is the five point Laplacian with zero boundary conditions, and

$$M_{ij}^h = \begin{cases} -2h^{-4}, & if \text{(i,j) = (1,j), j=2,}\ldots\text{,N-2,} \\ & \text{(i,j) = (N-1,j), j=2,}\ldots\text{,N-2} \\ & \text{(i,j) = (i,1),i=2,}\ldots\text{,N-2,} \\ & \text{(i,j) = (i,N-1), i=2,}\ldots\text{,N-2} \\ -4h^{-4}, & if \text{(i,j) = (1,1),(1,N-1), (N-1,1),}or\text{(N-1,N-1),} \\ 0, & otherwise. \end{cases}$$

Tables 1 and 2 show the result of applying (2.4-2.5) and (2.6-2.7) respectively to (3.1).

TABLE 1:PERFORMANCE OF (2.4)-(2.5) FOR (3.1)

Size of Problem	Number of Cycles	CF — First Cycle	CF — Last Cycle	average CF
9×9	10^*	2.3×10^1	8.5	5.7
19×19	10^*	9.3×10^3	1.7×10^8	2.1×10^4
39×39	10^*	4.3×10^1	8.9	1.1×10^1

* fails to converge in ten cycles

TABLE 2:PERFORMANCE OF (2.6)-(2.7) FOR (3.1)

Size of Problem	Number of Cycles	CF — First Cycle	CF — Last Cycle	average CF
9×9	10	.13	.14	.13
19×19	13	.45	.19	.22
39×39	19	1.2	.30	.32

This is problem (4.2) in [3]. There the convergence factors for the last cycle for the 9×9, 19×19, and 39×39 problems were, respectively, .12, .21, and .48 for nondiagonal interpolation (analogous to (2.4)-(2.5) and .15, .26, and .57 for diagonal interpolation (analogous to (2.6)-(2.7)).

The second problem is

$$\begin{aligned} -\nabla \cdot (D_{11} \nabla U^1) - \nabla \cdot (D_{12} \nabla U^2) &= F^1, \\ -\nabla \cdot (D_{21} \nabla U^1) - \nabla \cdot (D_{22} \nabla U^2) &= F^2 \text{ in } \Omega = (0, w_2) \times (0, w_2) \\ U^1 &= U^2 = 0 \text{ on } \partial\Omega, \end{aligned} \tag{3.2}$$

where $\bar{\Omega}_1 = [0, w_1] \times [0, w_1] \cup [w_1, w_2] \times [w_1, w_2]$,

$$D_{11}(x,y) = D_{22}(x,y) = \begin{cases} 1000 & \text{if } (x,y) \in \bar{\Omega}_1 \\ 1 & \text{otherwise,} \end{cases}$$

$$D_{12}(x,y) = D_{21}(x,y) = \begin{cases} 999 & \text{if } (x,y) \in \bar{\Omega}_1, \\ 0 & \text{otherwise,} \end{cases}$$

$$F^1(x,y) = F^2(x,y) = \begin{cases} 1 & \text{if } (x,y) \in \bar{\Omega}_1, \\ 0 & \text{otherwise.} \end{cases}$$

Table 3 shows the result of applying (2.4)-(2.5) to (3.2).

TABLE 3:PERFORMANCE OF (2.4)-(2.5) FOR (3.2)

Size of Problem	w_1 and w_2	Number of Cycles	CF — First Cycle	CF — Last Cycle	average CF
15×15	$7., 16.$	7	1.6	.06	.07
31×31	$15., 32.$	8	1.5	.07	.09
63×63	$31., 63.$	9	1.2	.08	.10
63×63	$32., 63.$	10^*	.39	.17	.14

* fails to converge in ten cycles

The results are the same for (2.6)-(2.7) for (3.2) to the number of decimal places reported. The same problem was done in [3] with $w_1 = 23.$ and $w_2 = 40.$ On a 39×39 grid, the convergence factor for the last cycle was .48 and .57 for nondiagonal and diagonal interpolation respectively.

Finally, we consider a problem that mimics the situation that arises in petroleum reservoir engineering when, instead of employing IMPES, equations implicit in pressure and saturation are employed:

$$\begin{aligned} -\nabla \cdot (D_{11}\nabla U^1) - \nabla \cdot (D_{12}\nabla U^2) + \tfrac{\partial U^2}{\partial x} + \tfrac{\partial U^2}{\partial y} &= F^1, \\ -\nabla \cdot (D_{21}\nabla U^1) - \nabla \cdot (D_{22}\nabla U^2) + \tfrac{\partial U^2}{\partial x} + \tfrac{\partial U^2}{\partial y} &= F^2 \text{ in } \Omega = (0,w_2) \times (0,w_2) \\ U^1 &= U^2 = 0 \text{ on } \partial\Omega, \end{aligned} \tag{3.3}$$

where $\bar{\Omega}_1 = [0,w_1] \times [0,w_1] \cup [w_1,w_2] \times [w_1,w_2]$,

$$D_{11}(x,y) = \begin{cases} 1 & \text{if } (x,y) \in \bar{\Omega}_1 \\ 4 & \text{otherwise,} \end{cases}$$

$$D_{12}(x,y) = D_{22}(x,y) = \begin{cases} 1 & \text{if } (x,y) \in \bar{\Omega}_1, \\ 2 & \text{otherwise,} \end{cases}$$

$$D_{21}(x,y) = \begin{cases} .3 & \text{if } (x,y) \in \bar{\Omega}_1, \\ .6 & \text{otherwise,} \end{cases}$$

$$F^1(x,y) = F^2(x,y) = \begin{cases} 1 & \text{if } (x,y) \in \bar{\Omega}_1, \\ 0 & \text{otherwise.} \end{cases}$$

Tables 4 and 5 give the results of (2.4)-(2.5) and (2.6)-(2.7) applied to (3.3).

TABLE 4:PERFORMANCE OF (2.4)-(2.5) FOR (3.3)

Size of Problem	w_1 and w_2	Number of Cycles	CF — First Cycle	CF — Last Cycle	average CF
15×15	$7., 16.$	10	.21	.12	.15
31×31	$15., 32.$	10^*	.38	.17	.24
63×63	$31., 63.$	10^*	.61	.26	.36
63×63	$32., 63.$	10^*	.61	.26	.36

* fails to converge in ten cycles

TABLE 5:PERFORMANCE OF (2.6)-(2.7) FOR (3.3)

Size of Problem	w_1 and w_2	Number of Cycles	CF — First Cycle	CF — Last Cycle	average CF
15×15	$7., 16.$	8	.15	.13	.12
31×31	$15., 32.$	10	.25	.15	.17
63×63	$31., 63.$	10^*	.41	.20	.24
63×63	$32., 63.$	10^*	.41	.20	.24

* fails to converge in ten cycles

These three examples illustrate that (2.6)-(2.7) is at least as effective as (2.4)-(2.5) in these three examples. The comparison for (3.1) is particularly compelling. In [3], with standard coarsening, the method based on nondiagonal interpolation was always superior to the method based on diagonal interpolation. For semicoarsening apparently the reverse is true. One observation is that the influence of interpolation for the methods in [3] is local. And for (2.6)-(2.7), the influence of interpolation becomes weaker as the distance from the interpolated point increases, since the inverse of a diagonally dominant matrix has this property. But for (2.4)-(2.5), no such claim can be made; indeed, for (3.1), the presence of the nonzero terms in M_{ij}^h near the boundary has a global influence on the coarse grid operator.

Acknowledgments. We acknowledge fruitful conversations with John Ruge and Yair Shapira. In particular, Ruge has implemented a similar scheme, and the long range plan was to investigate the differences between the two approaches; however, since both of our lives are currently interrupt-driven, this investigation may never occur.

REFERENCES

[1] J. E. DENDY, JR., *Black box multigrid*, J. Comput. Phys. 48(1982), pp. 366–386.

[2] J. E. DENDY, JR., *Black box multigrid for nonsymmetric problems*, Appl. Math. Comput. 13(1983), pp. 261–283.

[3] J. E. DENDY, JR., *Black box multigrid for systems*, Appl. Math. Comp. 19(1986), pp. 57–74.

[4] J. E. DENDY, JR., M. P. IDA, AND J. M. RUTLEDGE *A semicoarsening multigrid algorithm for SIMD machines*, SIAM J. Sci. Statist. Comp., 13[1992], pp. 1460–1469.

[5] J. E. DENDY, JR., S. F. MCCORMICK, J. W. RUGE, T. F. RUSSELL, S. SCHAFFER, *Multigrid methods for three-dimensional petroleum reservoir simulation*, Proceedings of the Tenth Symposium on Reservoir Simulation, Houston, TX, Feb. 6-8, 1989, pp. 19–25.

[6] L. W. EHRLICH, *Solving the biharmonic equation as coupled finite difference equations,* SIAM J. Numer. Anal. 8[1971], pp. 278–303.

[7] S. SCHAFFER, *A semi-coarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients,* to appear in SIAM J. Sci. Statist. Comput.

[8] R. A. SMITH, A. WEISER, *Semicoarsening multigrid on a hypercube*, SIAM J. Sci. Statist. Comput. 13[1992], pp. 1314–1329.

[9] G. WINTER, *Fourieranalyse zur Konstruktion schneller MGR-Verfahren*, Ph. D. Thesis, Rheinischen Friedrich-Wilhelms-Universität Bonn, 1983.