NASA Contractor Report 181834

ICASE REPORT NO. 89-24

ICASE

OPERATOR INDUCED MULTIGRID ALGORITHMS USING SEMIREFINEMENT

Naomi Decker

John Van Rosendale

Contract No. NAS1-18605
April 1989

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

Operator Induced Multigrid Algorithms Using Semirefinement *

Naomi Decker , John Van Rosendale

Abstract

This paper describes a variant of multigrid, based on zebra relaxation, and a new family of restriction/prolongation operators. Using zebra relaxation in combination with an operator-induced prolongation leads to fast convergence, since the coarse grid can correct all error components. The resulting algorithms are not only fast, but are also "robust," in the sense that the convergence rate is insensitive to the mesh aspect ratio. This is true even though line relaxation is performed in only one direction.

Multigrid becomes a direct method if one uses an operator-induced prolongation, together with the "induced" coarse grid operators. Unfortunately, this approach leads to stencils which double in size on each coarser grid. In this paper, we show how the use of an implicit three point restriction can be used to "factor" these large stencils, in order to retain the usual five or nine point stencils, while still achieving fast convergence. This algorithm achieves a V-cycle convergence rate of 0.03 on Poisson's equation, using 1.5 zebra sweeps per level, while the convergence rate improves to 0.003 if optimal nine point stencils are used. This paper presents numerical results for two and three dimensional model problems, together with a two level analysis explaining these results.

*Research supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-18605 while in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23665-5225.

1. Introduction

It is well known that zebra-based multigrid algorithms are fast. They usually have a spectral radius comparable to that of the best multigrid algorithms based on point relaxation, though their cost per V-cycle is somewhat greater. Zebra-relaxation is generally used in combination with semicoarsening, the grid is coarsened only in the direction orthogonal to the line relaxation direction. With semicoarsening, the geometric series $1 + 1/2 + 1/4 + ...$ enters the complexity bounds, rather than the usual series $1+1/4+1/16+...$ Since the first series sums to 2 rather than 4/3, and since the cost of a zebra sweep is slightly more than the cost of an SOR sweep, zebra-based algorithms are more expensive than point-relaxation based algorithms..

Despite this cost difference, zebra-based methods are attractive. It is been known for some time that zebra-based algorithms can be "robust" [6], though this fact is not widely appreciated. A "robust" algorithm is one in which the convergence rate is insensitive to mesh aspect ratios. In real applications highly stretched grids are ubiquitous, so this property is of great importance. Multigrid methods based on alternating line relaxation are robust, while methods based on point relaxation are not. It is a pleasant fact that zebra-based algorithms can be robust, even when one performs the line relaxation in just one direction.

Zebra relaxation also has another important property. After an odd zebra half-sweep, the residual is zero on the odd rows of the fine grid. This means that the coarse grid operator can "see" all components of the residual. Thus if one chose the proper coarse grid operator, one could, in principle, annihilate all fine grid error in one step. This possibility hinges on two requirements:

- one must use exactly the right coarse grid operator
- the prolongation to the fine grid must introduce no new residual on odd lines.

The first requirement here is problematic, but the second is easy to achieve. The family of prolongations known as "operator induced" prolongations has exactly the property required, see for example [3], [5]. Suppose we are looking at the fine grid operator, L_1, resulting from the discretization of the negative of the Laplacian on a grid Ω_1 using the 5-point stencil:

$$L_1 \hat{=} \frac{1}{dy^2} \begin{bmatrix} & -1 & \\ -a & 2a+2 & -a \\ & -1 & \end{bmatrix}$$

where $a = (dy/dx)^2$. Let Ω_1 be a grid with mesh sizes dx and dy, while Ω_2 is the grid formed by semicoarsening, having mesh sizes $2 * dx$ and dy. Then the prolongation

operator, P, given by the stencil

$$P \hat{=} \frac{1}{dy^2} \begin{bmatrix} & -1 & \\ a & 2a+2 & a \\ & -1 & \end{bmatrix}$$

is operator inducing, in the sense that

$$L_1 P : \Omega_2 \to \Omega_1$$

produces zeroes on odd columns of the fine grid. By direct calculation, the result of applying $L_1 P$ to a delta function (of height dy^4) on the coarse grid is:

$$\begin{array}{ccccc} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -4a-4 & 0 & 0 \\ -a^2 & 0 & 2a^2+8a+6 & 0 & -a^2 \\ 0 & 0 & -4a-4 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array}$$

Now suppose we perform a V-cycle algorithm, in which an odd zebra sweep is performed on the fine grid immediately before the coarse grid correction is computed. Thus there is residual only along even columns of the fine grid. This residual must be in the range of the operator $L_1 P$, as can be seen by a simple dimensionality argument; the number of coarse grid mesh points, and the number of points along even columns of the fine grid match.

Given this property, that the residual is in the range of $L_1 P$, one can solve the fine grid linear system in one step if the right coarse grid operator is inverted. That operator is easy to construct; one just drops the zero columns from the results of applying $L_1 P$ to a delta function:

$$\frac{1}{dy^2}\begin{bmatrix} & 1 & \\ & -4a-4 & \\ -a^2 & 2a^2+8a+6 & -a^2 \\ & -4a-4 & \\ & 1 & \end{bmatrix}$$

Notice that this stencil is larger than the fine grid stencil. This is the fundamental difficulty with this approach; the stencils approximately double in size at each coarser grid level. Use of these growing stencils in a k-level algorithm would result in a rather inefficient direct method.

Rather than using these growing stencils, our approach is instead to rely on simple five and nine point stencils, but to adjust the restriction operator Q, and perhaps the coarse grid operator L_2 as well, so that the operator

$$QL_2$$

closely approximates the optimal coarse-to-fine grid operator L_1P. To do this effectively, we must allow implicit restriction operators. Using three point implicit restrictions, which require inversion of tridiagonal systems, leads to a V-cycle algorithm which has very fast convergence. In effect, one is closely approximating a direct method. Table 1 gives the observed convergence rate per V-cycle of this algorithm applied to Laplace equation, as a function of problem size in two and three dimensions. We refer to this as the ZOOM algorithm, which stands for Zebra Optimized Operator Multigrid.

The remainder of this paper is organized as follows. Section 2 describes the design of our implicit restriction operators. Section 3 gives a two level Fourier analysis of our algorithm, and section 4 shows how this Fourier analysis can used to derive correction terms which can be added to the naïve choice of coarse grid operators to accelerate convergence. After that, section 5 compares the computational complexity of this algorithm with that of competing algorithms, and section 6 shows how these results extend to three dimensional problems. The paper finishes with a conclusion section giving a preliminary assessment of the utility of this method.

problem size	two dimensional results	three dimensional results
4	.0025	.0025
8	.0025	.0025
16	.0025	.0025
32	.0025	.0025
64	.0027	.0026
128	.0028	–
256	.0028	–

Table 1: ZOOM convergence rate, as a function of mesh size

2. Design of Coarse Grid Operators

As stated in the last section, if we invert the induced coarse grid operator, the V-cycle zebra algorithm would be a direct method. This operator has the stencil

$$\frac{1}{dy^2}\begin{bmatrix} & 1 & \\ & -4a-4 & \\ -a^2 & 2a^2+8a+6 & -a^2 \\ & -4a-4 & \\ & 1 & \end{bmatrix}$$

so performing zebra relaxation would involve pentadiagonal solves. Unfortunately, the stencils of these induced operators continue to grow, roughly doubling for each coarser level.

Rather than attempting to cope with these unwieldy stencils, we approximate these induced coarse grid stencils by combinations of 5-point or 9-point stencils. The possibility of doing this is based on the observation that the operator induced prolongation

P, given by the stencil

$$\frac{1}{dy^2}\begin{bmatrix} & -1 & \\ a & 2a+2 & a \\ & -1 & \end{bmatrix}$$

contains second differences in the y-direction, but is only an averaging operator in the x-direction. Thus the three point operator Q,

$$\frac{1}{dy^2}\begin{bmatrix} & -1 & \\ 0 & 4a+2 & 0 \\ & -1 & \end{bmatrix},$$

formed by summing the values of P along each row, should be close to P when applied to "smooth" functions.

Given Q, we can form the coarse grid approximation of the fine grid operator,

$$QL_2P^{-1},$$

where L_2 is the (negative of the) five point Laplacian on the coarse grid:

$$\frac{1}{dy^2}\begin{bmatrix} & -1 & \\ -a/4 & 2+a/2 & -a/4 \\ & -1 & \end{bmatrix}$$

The idea here is that the unwanted effect of the second difference in P is canceled by the identical second difference operator built into Q.

In order to examine the effectiveness of this approach, we need to compare the fine grid operator L_1 with the coarse grid correction operator:

$$L_1 \cong QL_2P^{-1}$$

While L_1 has a finite stencil, the right hand side is a global operator, making comparison awkward. To circumvent this we multiply both sides by P, arriving at the relationship

$$L_1P \cong QL_2,$$

which allows easy comparisons.

With the choices above, the operator QL_2 has the form:

$$\frac{1}{dy^4}\begin{bmatrix} 0 & 1 & 0 \\ a/4 & -4.5a-4 & a/4 \\ -a^2-a/2 & 2a^2+9a+6 & -a^2-a/2 \\ a/4 & -4.5a-4 & a/4 \\ 0 & 1 & 0 \end{bmatrix}$$

Thus the difference between QL_2 and L_1P is

$$\frac{1}{dy^4}\begin{bmatrix} a/4 & -a/2 & a/4 \\ -a/2 & a & -a/2 \\ a/4 & -a/2 & a/4 \end{bmatrix},$$

which looks like u_{xxyy}. Therefore, since $dx^2Q \approx 4 - dx^2u_{yy}$, the coarse grid operator is approximating $Q^{-1}L_1P$ to second order. We return to this in more detail when we consider other coarse grid operators which are higher order approximations, see Section 4.

3. Analysis

In this section we present the analysis of our algorithm for the case of the positive definite Helmholtz equation on the unit square. This analysis is remarkably simple because the subspaces involved in both the relaxation and coarse grid correction are identical. Assume the equation is discretized on a uniform mesh using the standard five point stencils for the second order derivatives. Let N_x and N_y be the number of points in the x and y directions, respectively. The complex fourier modes represented on this mesh are given by

$$u^{\theta,\pi}(m,n) = \exp(i\theta m + i\phi n) \qquad \begin{array}{ll} \text{for} & m = 1,\cdots,N_x-1, \\ \text{and} & n = 1,\cdots,N_y-1 \end{array}$$

where $\theta = \pi j/N_x$, $j = -N_x+1,\cdots,N_x-1$ and $\phi = \pi k/N_y$, $k = -N_y+1,\cdots,N_y-1$.

The zebra-relaxation (by columns) and the coarse grid corrections (semi-coarsening by deleting columns) mix the fourier modes, so it is not possible to use a simple scalar amplification factor for the analysis. Applying either operator to a single fourier mode, $u^{\theta,\phi}$, results in a linear combination of two modes, $u^{\theta,\phi}$ and $u^{\theta+\pi,\phi}$. These two modes form an invariant subspace under both the zebra-relaxation and the coarse grid correction, and therefore form an invariant subspace of the iteration matrix of the two level multigrid algorithm. Thus, we can think in terms of a 2×2 "matrix amplification factor", depending on θ and ϕ, and the spectral radius is a bound on the asymptotic convergence rate for any error in that subspace. The maximum over all θ and ϕ is a bound on the asymptotic convergence rate of the two grid algorithm.

Zebra relaxation

Consider the zebra relaxation acting on the subspace corresponding to a fixed θ and ϕ. The symbol of the discrete operator is given by

$$\hat{L}_1 = 4N_x^2 \sin^2(\theta/2) + 4N_y^2 \sin^2(\phi/2) + K,$$

in other words,

$$L_1 u^{\theta,\phi} = \hat{L}_1 u^{\theta,\phi}.$$

We define the symbol for the other mode in the subspace as $\hat{P}$, where

$$L_1 u^{\theta+\pi,\phi} = \hat{P} u^{\theta+\pi,\phi}.$$

Thus,

$$\hat{P} = 4N_x^2 \cos^2(\theta/2) + 4N_y^2 \sin^2(\phi/2) + K.$$

For the odd-column relaxation we have (see [4],[6])

$$G^O u^{\theta,\phi} = \frac{1}{\hat{L}_1 + \hat{P}} (\hat{P} u^{\theta,\phi} + \hat{L}_1 u^{\theta+\pi,\phi}). \tag{1}$$

(Note that it is easy to check that this formula is consistent with the property that, after an odd-column relaxation, the residual vanishes on odd lines.) By Equation 1, the symbol of the odd-column relaxation is:

$$\hat{G}^O = \frac{1}{\hat{L}_1 + \hat{P}} \begin{pmatrix} \hat{P} & \hat{P} \\ \hat{L}_1 & \hat{L}_1 \end{pmatrix}$$

Similarly for the even relaxation we have:

$$\hat{G}^E = \frac{1}{\hat{L}_1 + \hat{P}} \begin{pmatrix} \hat{P} & -\hat{P} \\ -\hat{L}_1 & \hat{L}_1 \end{pmatrix}$$

Coarse grid correction

The coarse grid correction is a combination of the restriction, prolongation and coarse grid operators. The symbol of the prolongation operator for the (θ, ϕ) mode is $\hat{P}$. Recall that this is also the symbol of the fine grid operator for the $(\theta + \pi, \phi)$ mode.

For the restriction operator we have:

$$R\, u^{\theta,\phi} = \hat{Q}^{-1} \cos^2(\theta/2)\, u^{\theta,\phi}$$

$$R\, u^{\theta+\pi,\phi} = \hat{Q}^{-1} \sin^2(\theta/2)\, u^{\theta+\pi,\phi}$$

where $\hat{Q} = 4N_x^2 + 4N_y^2 \sin^2(\theta/2) + K$ is the symbol of the restriction, Q. On coarse grid points (even columns) the two modes in the subspace are identical,

$$u^{\theta,\phi} = u^{\theta+\pi,\phi},$$

and therefore the coarse grid solve can be represented by a scalar symbol, $\hat{L}_2^{-1}$.

Putting it all together, we see that the coarse grid correction has the symbol

$$\hat{T} = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

where

$$\begin{aligned} a &= 1 - \hat{L}_1 \hat{P} (\hat{Q}\hat{L}_2)^{-1} \cos^2(\theta/2), \\ b &= -\hat{P}^2 (\hat{Q}\hat{L}_2)^{-1} \sin^2(\theta/2), \\ c &= -\hat{L}_1^2 (\hat{Q}\hat{L}_2)^{-1} \cos^2(\theta/2), \\ d &= 1 - \hat{L}_1 \hat{P} (\hat{Q}\hat{L}_2)^{-1} \sin^2(\theta/2). \end{aligned}$$

The complete two grid iteration matrix is therefore:

$$\hat{G}^E \hat{T} \hat{G}^O = \frac{\hat{P}(a-c) + \hat{L}_1(b-d)}{(\hat{L}_1 + \hat{P})^2} \begin{pmatrix} \hat{P} & \hat{P} \\ -\hat{L}_1 & -\hat{L}_1 \end{pmatrix} \tag{2}$$

Since each of the 2×2 amplification factors is singular, the largest eigenvalue is just the trace (sum of the diagonal elements). We are thus left with a simple formula for the non-zero eigenvalue and the spectral radius of the two grid algorithm is given by:

$$\rho_{TG}(\hat{G}^E \hat{T} \hat{G}^O) = \left| \frac{(\hat{P} - \hat{L}_1)^2}{(\hat{P} + \hat{L}_1)^2} \left(I - \hat{P}\hat{L}_1 (\hat{Q}\hat{L}_2)^{-1} \right) \right| \tag{3}$$

From this formula it is clear that if $\hat{Q}\hat{L}_2 = \hat{P}\hat{L}_1$ then the two grid method would be direct. In our two grid algorithm the coarse grid operator is the product of the inverse of the coarse grid operator and the approximate inverse to the prolongation (which is performed either in the prolongation or the restriction). A good strategy is to take, as we have done, the coarse grid operator to be an approximation of the fine grid operator. However, it is also apparent that to get even better convergence rates, the factor, $|I - \hat{P}\hat{L}_1(\hat{Q}\hat{L}_2)^{-1}|$, only needs to be small when $|(\hat{P} - \hat{L}_1)^2/(\hat{P} + \hat{L}_1)^2|$ is not small. In the next section we use this analysis to predict 'optimal' coarse grid operators, i.e., the coarse grid operators which are consistent with the fine grid operator, but contain additional correction terms which can further reduce the convergence rates.

4. Accelerated convergence

In certain situations it is possible to accelerate the convergence rate of the ZOOM algorithm. For example, consider the Helmholtz equation:

$$-u_{xx} - u_{yy} + Ku = f.$$

Using the same rectangular mesh as in the last section, the symbol of the fine grid operator, L_1, is

$$\hat{L}_1 = 4N_x^2 \sin^2(\theta/2) + \hat{L}_y,$$

where

$$\hat{L}_y \equiv 4N_y^2 \sin^2(\phi/2) + K.$$

The symbols of the prolongation and restriction operators are:

$$\hat{P} = 4N_x^2 \cos^2(\theta/2) + \hat{L}_y,$$

and

$$\hat{Q} = 4N_x^2 + \hat{L}_y.$$

From Equation 3 we see that we want to choose the coarse grid operator, L_2 (with symbol $\hat{L}_2$), so that $\hat{L}_2\hat{Q} \approx \hat{L}_1\hat{P}$. The symbol of the "natural" coarse grid operator is

$$\tilde{L}_2 = 4(N_x/2)^2 \sin^2\theta + \hat{L}_y$$

and we want to find a correction term to be added to $\tilde{L}_2$ to optimize the convergence rate of the algorithm. Thus we choose our coarse grid operator, $\hat{L}_2$ to approximate the quantity $\hat{Q}^{-1}\hat{L}_1\hat{P}$, where

$$\begin{aligned} \hat{L}_1\hat{P} &= (4N_x^2 \cos^2(\theta/2) + \hat{L}_y)(4N_x^2 \sin^2(\theta/2) + \hat{L}_y) \\ &= (4N_x^4 \sin^2\theta + 4N_x^2\hat{L}_y + \hat{L}_y^2). \end{aligned}$$

The goal is to approximate $\hat{Q}^{-1}$ as closely as possible without allowing the stencil of the coarse grid operator to grow. For small ϕ,

$$\hat{Q} \approx 4N_x^2 + K.$$

For larger values of ϕ, the $(\hat{P} - \hat{L})/(\hat{P} + \hat{L})$ term in Equation 3 is small (the smoothing reduces the high frequencies in the y-direction) so the correction term is less important for the high frequency modes. Therefore we want to choose

$$\hat{L}_2 = \tilde{L}_2 - \frac{N_x^2}{4N_x^2 + K} \sin^2\theta L_y.$$

For $K = 0$ we compare $\hat{Q}\hat{L}_2 - \hat{L}_1\hat{P}$ and $\hat{Q}\tilde{L}_2 - \hat{L}_1\hat{P}$. For the operator with no correction term:

$$\begin{aligned} \hat{Q}\tilde{L}_2 - \hat{L}_1\hat{P} &= 4N_y^2 \sin^2(\phi/2)\sin^2\theta \\ &= O(1). \end{aligned}$$

but with the correction term:

$$\begin{aligned}\hat{Q}\hat{L}_2 - \hat{L}_1\hat{P} &= -\frac{4N_y^2}{N_x^2}\sin^4(\phi/2)\sin^2\theta \\ &= O(dx^2).\end{aligned}$$

Thus L_2 approximates $Q^{-1}L_1P$ to fourth order whereas with no correction the approximation is only second order.

The analytic operator which would give us the coarse grid operator, $\hat{L}_2$ (discretizing directly on the coarse grid with the standard difference formulae) is

$$-u_{xx} - u_{yy} + Ku - \frac{N_x^2}{4(4N_x^2+K)}dx^2(-u_{yy}+Ku)_{xx},$$

where dx is the fine grid spacing in the x direction. More generally, we consider the family of coarse grid operators of the form:

$$-u_{xx} - u_{yy} + Ku + \delta dx^2 u_{yyxx} - \alpha dx^2 K u_{xx}$$

Since we want to use multigrid recursively, these correction terms will accumulate. Thus, on grid level k, the correction terms will be:

$$\begin{aligned}\delta_k &= \delta\,(1+\frac{1}{4}+\frac{1}{16}+\cdots+\frac{1}{4^{k-2}}) \\ \alpha_k &= \alpha\,(1+\frac{1}{4}+\frac{1}{16}+\cdots+\frac{1}{4^{k-2}})\end{aligned}$$

For the case $K = 0$, only the δ correction is needed. Table 2 lists the asymptotic convergence rates of the ZOOM algorithm using various amounts of this correction. The above analysis indicates that for the two grid algorithm we should use $\delta \approx .0625$. The theoretical two grid rates (solving the coarse grid equations exactly) are obtained from Equation 3. Experimental results both for the two grid and using all available coarse grids indicate that the convergence rates are better for a slightly smaller correction, $\delta = .0525$.

Figure 1 examines the effect of the fine grid aspect ratio dx/dy, on the convergence rates for three different values of δ. We see that $\delta = .0525$ is an excellent choice, though the convergence rate remains uniformly good with all three choices. Notice that for the ZOOM algorithm, we are particularly interested in large aspect ratios since we are coarsening in the x direction only.

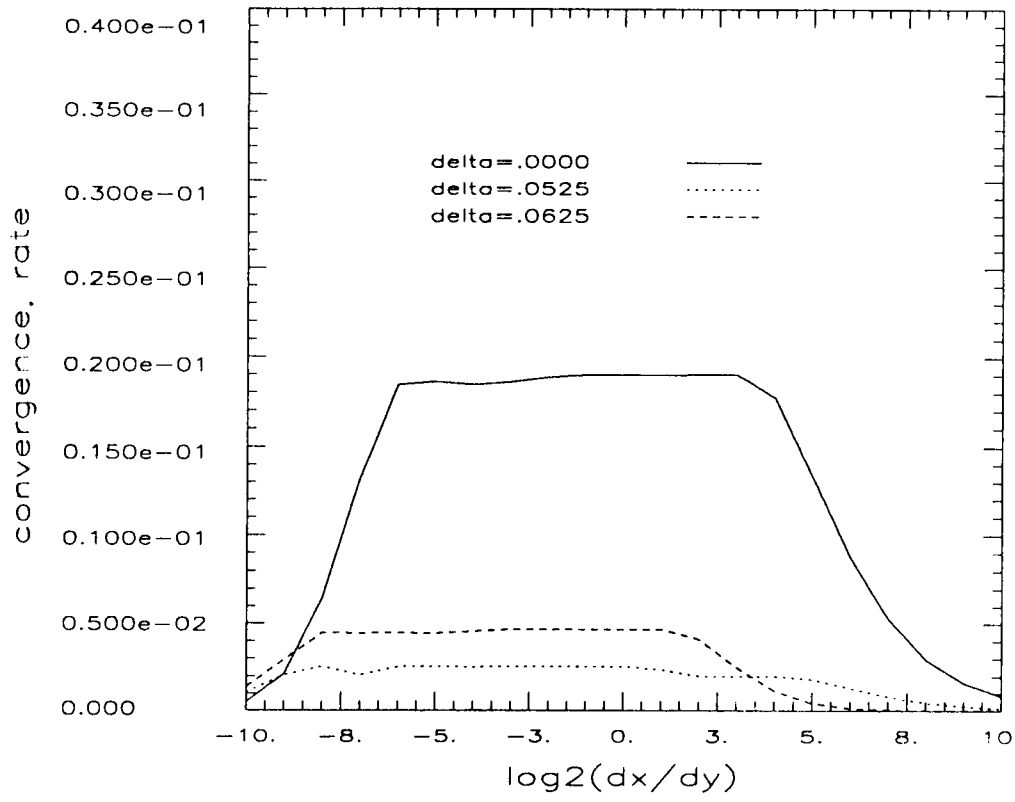

Figure 1: Dependence of convergence rate on mesh aspect ratio

correction δ	theoretical ρ_{TG}	experimental 2 level	experimental k level
0.00	.0190	.0186	.0237
0.01	.0158	.0153	.0199
0.02	.0125	.0121	.0159
0.03	.0092	.0088	.0118
0.04	.0059	.0055	.0077
0.05	.0027	.0024	.0035
0.0525	.0026	.0024	.0025
0.06	.0041	.0040	.0040
0.07	.0067	.0065	.0065
0.08	.0101	.0096	.0097
0.09	.0137	.0131	.0135
0.10	.0175	.0168	.0177

Table 2: Convergence rates for various amounts of coarse grid operator corrections, (32×32 grid)

Returning to the positive definite Helmholtz operator, $K > 0$, we consider the effect of using only the δ correction or using the full Helmholtz correction, with both δ and α. Figure 2 shows a graph of the theoretical two grid convergence rates for various values of K, on a 32 by 32 grid. The solid line is for $\delta = \alpha = 0$, the dotted line for $\delta = .0525$, $\alpha = 0$, and the dashed line is the convergence rate for $\delta = \alpha = .0525$. Thus, with the full correction, we expect uniformly good convergence rates for all $K > 0$. This is important for the three dimensional version of ZOOM, since the operators inverted in the plane solves contain zeroth order terms. In the k-level experiments, we see exactly the same behavior, see Tables 3-5.

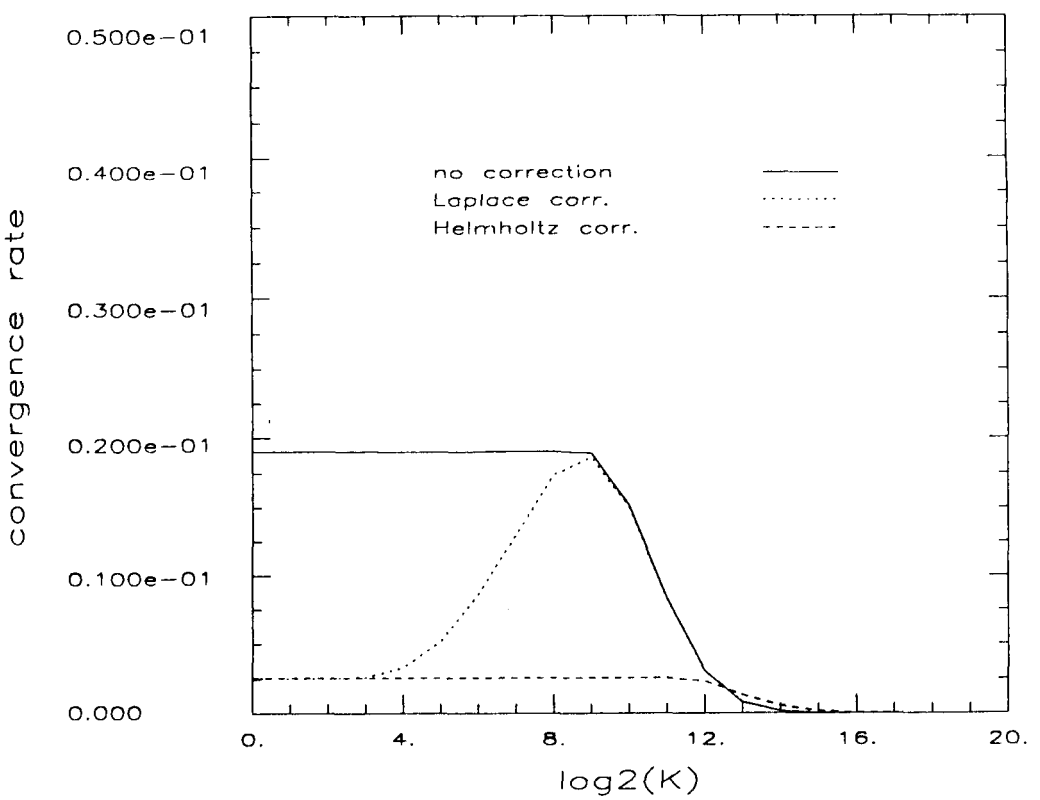

Figure 2: Dependence of convergence rate on Helmholtz term

	Helmholtz term, K									
size	0	1	3	10	30	100	300	1000	3000	10000
4	.0176	.0173	.0165	.0132	.0065	.0011	.0000	.0000	.0000	.0000
8	.0233	.0233	.0230	.0208	.0175	.0098	.0022	.0002	.0000	.0000
16	.0243	.0243	.0241	.0228	.0233	.0202	.0130	.0032	.0003	.0000
32	.0245	.0245	.0243	.0235	.0243	.0235	.0215	.0154	.0050	.0005
64	.0245	.0245	.0244	.0240	.0242	.0241	.0236	.0223	.0177	.0065

Table 3: k grid convergence rates, $\alpha = \gamma = 0.0$ (no corrections)

	Helmholtz term, K									
size	0	1	3	10	30	100	300	1000	3000	10000
4	.0025	.0023	.0042	.0064	.0045	.0009	.0001	.0000	.0000	.0000
8	.0025	.0039	.0068	.0110	.0135	.0088	.0021	.0001	.0000	.0000
16	.0025	.0044	.0073	.0120	.0182	.0186	.0126	.0031	.0003	.0000
32	.0025	.0045	.0075	.0122	.0190	.0216	.0209	.0153	.0050	.0005
64	.0026	.0046	.0076	.0122	.0192	.0222	.0230	.0221	.0176	.0065

Table 4: k grid convergence rates, $\gamma = 0.0525$, $\alpha = 0.0$ (no helmholtz correction)

	Helmholtz term, K									
size	0	1	3	10	30	100	300	1000	3000	10000
4	.0025	.0024	.0025	.0025	.0025	.0016	.0004	.0001	.0000	.0000
8	.0025	.0025	.0025	.0025	.0025	.0025	.0021	.0005	.0001	.0000
16	.0025	.0025	.0025	.0025	.0025	.0025	.0025	.0023	.0008	.0001
32	.0025	.0025	.0025	.0027	.0025	.0025	.0025	.0025	.0025	.0011
64	.0026	.0026	.0026	.0026	.0025	.0026	.0025	.0025	.0025	.0025

Table 5: k grid convergence rates, $\alpha = \gamma = 0.0525$ (full correction)

5. Comparison with other algorithms

In this section we compare the efficiency of the ZOOM algorithm with that of three other multigrid methods. The first of these is a standard V-cycle algorithm based on red-black Gauss-Seidel iteration, as given in [1],[4]. This algorithm is not robust, in the sense that the convergence rate degenerates on stretched grids. All multigrid algorithms based on point iterative relaxation lack robustness. This algorithm was included to show how the ZOOM algorithm compares to a standard algorithm. The second is a zebra-based algorithm described by Winter [6], which uses semicoarsening, and has the same robustness properties as the ZOOM algorithm. The third is an alternating direction zebra algorithm, [4], which has fast convergence and is also robust. There are obviously a wide variety of other multigrid algorithms we could have compared against, but the comparisons given should be representative, and adequately demonstrate the efficacy of our approach.

To make fair comparisons between these algorithms, one must take into account the cost per iteration. Table 6 gives the operations required in a V-cycle for each of the four algorithms being compared. Details, such as the order of even and odd relaxation sweeps, and the like are omitted; the interested reader is referred to the references cited. The approximate cost of these operations is given in Tables 7 and 8. The cost of the operations involved in point Gauss-Seidel algorithms is given in Table 7, while Table 8 gives the cost of the operations involved in zebra-based algorithms. The entries here are floating point operations per mesh point. The reader is cautioned that the particular values occurring are sensitive to assumptions, such as which symmetries can be exploited, and whether the tridiagonal factorizations are stored or recomputed.

	operator	relaxation	restriction	prolongation
Red-black GS algorithm				
fine grid	5 point	1.5 red-black GS	bilinear	bilinear
coarse grids	"	2.0 red-black GS	"	"
Winter's algorithm				
fine grid	5 point	1.5 odd-even zebra	injection	injection
coarse grids	"	"	"	"
Alternating direction				
fine grid	5 point	2 odd-even zebra	bilinear	bilinear
coarse grids	"	"	"	"
ZOOM algorithm				
fine grid	5 point	1 odd-even zebra	implicit	5 point
coarse grids	9 point	1.5 odd-even zebra	"	9 point

Table 6: Operations in methods being compared

	standard Laplacian		variable coefficients	
	5 point	9 point	5 point	9 point
red-black GS	4	8	8	16
bilinear restriction	2.5	6	6	20.5
bilinear prolongation	1	2	1	2

Table 7: Cost of operations in red-black GS multigrid

	standard Laplacian		variable coefficients	
	5 point	9 point	5 point	9 point
zebra relaxation	7	11	12	20
explicit restriction	3	5	5	9
implicit restriction	3.5	5.5	6	10
operator induced prolongation	1	2.5	2	6

Table 8: Cost of operations in zebra-based multigrid

Cost per V-cycle

Given these tables, we can estimate the effectiveness of each algorithm. The number of floating point operations per V-cycle of the red-black Gauss-Seidel algorithm is:

$$(1.5 \cdot 4 + 2.5 + 1) + (2 \cdot 4 + 2.5 + 1) \cdot (\frac{1}{4} + \frac{1}{16} + \ldots) = 13.33$$

$$(1.5 \cdot 8 + 6 + 1) + (2 \cdot 8 + 6 + 1) \cdot (\frac{1}{4} + \frac{1}{16} + \ldots) = 26.67$$

in the constant coefficient, and variable coefficient cases respectively. The first term in each equation is the cost of the fine grid operations, while the second term is the cost of the coarse grid correction. For Winter's algorithm we have

$$(1.5 \cdot 7 + 3 + 0.5) \cdot (1 + \frac{1}{2} + \frac{1}{4} + \ldots) = 24.8$$

$$(1.5 \cdot 12 + 5 + 0.5) \cdot (1 + \frac{1}{2} + \frac{1}{4} + \ldots) = 41.6$$

where the series $1 + 1/2 + 1/4 + \ldots$ enters because of the semicoarsening. For the alternating direction zebra algorithm we have:

$$(2 * 7 + 3 + 0.5) \cdot (1 + \frac{1}{4} + \frac{1}{16} + \ldots) = 23.33$$

$$(2 * 12 + 5 + 0.5) \cdot (1 + \frac{1}{4} + \frac{1}{16} + \ldots) = 39.33$$

algorithm	convergence rate	cost per V-cycle	cost per factor of 10 reduction
red-black GS			
constant coef.	.0741	13.33	11.8
variable coef.		26.67	23.6
Winter's zebra algorithm			
constant coef.	.0741	28	24.8
variable coef.		47	41.6
Alternating direction zebra			
constant coef.	.0230	23.33	14.2
variable coef.		39.33	24.0
ZOOM algorithm			
constant coef.	.0024	36	13.7
variable coef.		66	25.2

Table 9: Comparison of algorithms

Finally for the ZOOM algorithm we have:

$$(7 + 3.5 + 1) + (1.5 \cdot 11 + 5.5 + 2.5) \cdot (1/2 + 1/4 + \ldots) = 36$$

$$(12 + 6 + 2) + (1.5 \cdot 20 + 10 + 6) \cdot (1/2 + 1/4 + \ldots) = 66$$

The performance of each of these algorithms is summarized in Table 9. The first column gives the required number of floating point operations per mesh point, as computed above. The second column gives the two-level convergence rate for each algorithm. It would have been better to use k-level rates, but we did not have k-level rates for all four algorithms.

Now combining these results, we estimate the relative effectiveness of each algorithm. Define the "cost per factor of 10 reduction" as:

$$cost \ / \ log_{10}(\rho)$$

where ρ is the convergence rate. Not surprisingly, the point Gauss-Seidel algorithm is slightly faster than these robust algorithms, though the difference is rather slight. Among the robust algorithms, the ZOOM algorithm and the alternating direction algorithm appear fastest. However, if one looks at stretched grids, the convergence rate of the alternating direction algorithm degenerates, while that of both the ZOOM algorithm and Winter's algorithm do not, Table 10. Thus the ZOOM seems preferable to either of the other robust algorithms.

ϵ	Alternating direction [4]	Winter's algorithm [6]	ZOOM algorithm
100	.119	.015	.0010
10	.082	.063	.0020
2	.019	.074	.0024
1	.023	.074	.0025
0.5	.019	.074	.0025
0.1	.082	.074	.0025
0.01	.119	.074	.0024

Table 10: Two grid convergence rates for $L = -\epsilon u_{xx} - u_{yy}$

Parallel Implications

The ZOOM algorithm also has advantages as a parallel algorithm. Algorithms based on full coarsening are attractive on sequential machines, where the cost of operations on coarse grids is unimportant. However, on parallel architectures, operations on coarse grids are important, since they limit parallelism. Thus semicoarsening, which increases sequential cost, is not a disadvantage on parallel architectures.

In considering the parallel speed of this kind of algorithm, there is one issue in ZOOM which needs to be addressed. In the ZOOM algorithm, each coarser level has half as many tridiagonals as the next finer level, but the size of these tridiagonals does not change. This is clearly awkward on parallel architectures. One would prefer the exact opposite, that the number of tridiagonals remains the same, but the size of each tridiagonal halves on each coarser grid. Though we cannot achieve this while retaining robustness, we can implement ZOOM using x-direction zebra on odd levels and y-direction zebra on even levels, still using semicoarsening. Two level analysis shows that this does not change the algorithms numerical properties, and all coarse grids are now approximately square.

Given this modification, a direct comparison of ZOOM and the alternating direction algorithm shows ZOOM to be much faster on sufficiently parallel architectures. The number of zebra sweeps per V-cycle is the same (including the implicit restricts), so the execution times of these algorithms should be comparable on sufficiently parallel architectures. However, the convergence rate of ZOOM is ten times better on unstretched meshes, and almost a hundred times better on highly stretched meshes. Thus ZOOM seems clearly superior on parallel architectures.

The comparison between ZOOM and Winter's algorithm as parallel algorithms is closer. These algorithms are very similar, so the use of parallel architectures effects both of these algorithms identically. Thus the difference in the performance of these

algorithms on parallel architectures, will be the same as the difference in their sequential complexity.

6. Three Dimensional Problems

The ZOOM algorithm uses zebra relaxation, and is thus natural in two dimensions. However, it can be used in three dimensions as well, if one performs the analogous three dimensional zebra relaxation. To solve the system of equations for each plane in the zebra relaxation, we simply use the two dimensional ZOOM algorithm.

Suppose, for example, one is solving Poisson's equation in three dimensions. The seven point Laplacian in three dimensions is:

$$\begin{bmatrix} & 0 & \\ 0 & -a^2 & 0 \\ & 0 & \end{bmatrix}$$

$$\ddots$$

$$\begin{bmatrix} & -dy^{-2} & \\ -dx^{-2} & 2dx^{-2}+2dy^{-2}+2a^2 & -dx^{-2} \\ & -dy^{-2} & \end{bmatrix}$$

$$\ddots$$

$$\begin{bmatrix} & 0 & \\ 0 & -a^2 & 0 \\ & 0 & \end{bmatrix}$$

where a looks like $1/dz^2$, and varies from grid to grid in semi-coarsening.

To perform three dimensional zebra relaxation, we need to invert the five point Helmholtz operator

$$\begin{bmatrix} & -dy^{-2} & \\ -dx^{-2} & 2dx^{-2}+2dy^{-2}+2a^2 & -dx^{-2} \\ & -dy^{-2} & \end{bmatrix} \tag{4}$$

A similar Helmholtz operator needs to be inverted in the implicit restriction. Since the convergence of the two dimensional ZOOM algorithm was not harmed by positive definite Helmholtz terms, this presents no difficulty.

Acceleration of convergence works exactly as in two dimensions. For the operator

$$-u_{xx}-u_{yy}-u_{zz}+Ku \tag{5}$$

we need to add the term

$$\delta dz^2(-u_{xx} - u_{yy} - Ku)_{zz} \tag{6}$$

where dz is the fine grid mesh spacing, and δ is chosen exactly as in two dimensions.

size	no correction	with correction	
		1 inner iteration	2 inner iterations
4	.0080	.0063	.0025
8	.0165	.0061	.0025
16	.0206	.0063	.0025
32	.0231	.0063	.0025
64	.0239	.0063	.0025

Table 11: three dimensional Laplace equation

The performance of the three dimensional zoom algorithm is given in table 11. The second column shows the spectral radius, when the correction term is neglected. The convergence rate here is similar to that in the two dimensional case. The next column gives the rate of convergence when the correction term is included, if one uses a single ZOOM V-cycle for each of the plane solves required, including those required in the implicit restrictions. While a single ZOOM V-cycle reduces the error by a factor of 400, the error remaining still harms convergence of the three dimensional algorithm slightly. The last column gives the convergence rate when two ZOOM V-cycles are used to solve the equations on each plane. Two V-cycles reduce the error on each plane by a factor of more than 100,000, and so provide, in effect, exact solutions. In this case, the convergence of the two and three dimensional ZOOM algorithms are seen to be identical.

Despite this fast convergence, the three dimensional ZOOM algorithm is not competitive with three dimensional multigrid algorithms based on point relaxation, unless one requires robustness. This is both because of the cost of the plane solves, and because of the slow geometric series caused by semicoarsening $1 + 1/2 + 1/4 + ...$ rather than usual series $1 + 1/8 + 1/64 + ...$ with full coarsening. However, when robustness is an issue, the ZOOM algorithm is competitive. As in two dimensions, the convergence rate of the algorithm is insensitive to mesh aspect ratio.

7. Conclusions

The ZOOM algorithm is new and relatively untried, but is clearly promising. The algorithm has been shown to be effective on simple self-adjoint problems, and we are in the process of studying it on problems dominated by first derivative terms. We also expect that it will make an effective preconditioner for conjugate gradient iteration on the indefinite Helmholtz equation, since the coarse grid operators in ZOOM closely approximate the fine grid operators, and thus should have very similar eigenfunctions and eigenvalues [2].

The fundamental limitation to this method is its restriction to tensor product grids. It is also sensitive to rapidly varying problem coefficients. At locations where coefficients jump, it is impossible to form a prolongation stencil which properly "operator-induces;" one must make numerical compromises. However, given that the coefficient jumps are not extreme, analysis suggests convergence rates of 0.03 or better can still be achieved. With this lower convergence rate, there is no need to include the accelerating correction terms on coarse grids, so the cost of the algorithm can be reduced by use five point stencils.

One of the most interesting applications of this method is in the solution of elliptic equations on Chebyshev grids. Such problems arise when one uses an implicit discretization of the pressure equation in the Navier Stokes equations, and in a variety of other applications. The Chebyshev grid causes a slow variation in the coefficients in the difference stencil, with the rate of variation going to zero as the grid is refined. Very high and very low mesh aspect ratios are also encountered along the domain boundaries. Thus the ZOOM algorithm should be ideal for this application, and we anticipate convergence rates comparable to those obtained on uniform grids.

References

[1] A. Brandt. Guide to multigrid development. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods*. Springer Verlag, 1981. Lecture Notes in Mathematics, 960.

[2] N.H. Decker. *The Fourier Analysis of Multigrid-Type Iterative Methods*. PhD thesis, University of Wisconsin – Madison, 1987.

[3] J.E. Dendy. Black box multigrid. *Journal of Computational Physics*, 48:366–386, 1982.

[4] K. Stüben and U. Trottenberg. On the construction of fast solvers for elliptic equations. Technical Report IMA-Report Mr. 82.0201, GMD MBH Bonn, 1982.

[5] D. Kamowitz. *Theoretical and Experimental Results for a Variety of Multigrid Algorithms*. PhD thesis, University of Wisconsin – Madison, 1986.

[6] G. Winter. *Fourieranalyse zur Konstruktion schneller MGR-Verfahren*. PhD thesis, Rheinischen Friedrich-Wilhelms-Universität zu Bonn, 1982.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-181834 ICASE Report No. 89-24		

4. Title and Subtitle	5. Report Date
OPERATOR INDUCED MULTIGRID ALGORITHMS USING SEMIREFINEMENT	April 1989
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Naomi Decker John Van Rosendale	89-24
	10. Work Unit No. 505-90-21-01

9. Performing Organization Name and Address	11. Contract or Grant No.
Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225	NAS1-18605
	13. Type of Report and Period Covered: Contractor Report

12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code
National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	

15. Supplementary Notes

Langley Technical Monitor:
Richard W. Barnwell

Final Report

16. Abstract

This paper describes a variant of multigrid, based on zebra relaxation, and a new family of restriction/prolongation operators. Using zebra relaxation in combination with an operator-induced prolongation leads to fast convergence, since the coarse grid can correct all error components. The resulting algorithms are not only fast, but are also "robust," in the sense that the convergence rate is insensitive to the mesh aspect ratio. This is true even though line relaxation is performed in only one direction. Multigrid becomes a direct method if one uses an operator-induced prolongation, together with the "induced" coarse grid operators. Unfortunately, this approach leads to stencils which double in size on each coarser grid. In this paper, we show how the use of an implicit three point restriction can be used to "factor" these large stencils, in order to retain the usual five or nine point stencils, while still achieving fast convergence. This algorithm achieves a V-cycle convergence rate of 0.03 on Poisson's equation, using 1.5 zebra sweeps per level, while the convergence rate improves to 0.0003 if optimal nine point stencils are used. This paper presents numerical results for two and three dimensional model problems, together with a two level analysis explaining these results.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Multigrid, zebra relaxation, stretched grids	64 - Numerical Analysis Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	23	A03