DELFT UNIVERSITY OF TECHNOLOGY

REPORT 85-08

Aspects of multigrid methods
for problems in three dimensions

by

R. Kettler and P. Wesseling

Reports of the Department of Mathematics and Informatics no. 85-08

Delft 1985

R. Kettler

Delft University of Technology
Dept. of Mathematics and Informatics
P.O. Box 356
2600 AJ DELFT
THE NETHERLANDS

and

Kon. Shell Expl. Prod. Laboratory
RIJSWIJK
THE NETHERLANDS

P. Wesseling

Delft University of Technology
Dept. Of Mathematics and Info
P.O. Box 356
2600 AJ DELFT
THE NETHERLANDS

Paper presented at the Second Copper Mountain Conference on Multigrid
Methods, April 1 - 3, 1985.

Abstract

Smoothing methods are analyzed for use in multigrid methods applied to flow problems in three dimensions. A three-dimensional version of incomplete block factorisation is described. The anisotropic diffusion and convection-diffusion equations are used as model problems. Unlike the two-dimensional case, none of the smoothing methods considered is suitable for all relevant values of the physical parameters.

1. Introduction

With fluid flow problems in mind, we study in this paper the use of multigrid methods for the solution of discretized versions of:

$$-\nabla \cdot (D\nabla\phi) + u\nabla\phi + c\phi = f, \quad x \in \Omega \subset \mathbb{R}^d \quad (1.1)$$

with D a symmetric positive definite $d \times d$ matrix, and u a d -dimensional vector.

Our objective is to develop a multigrid program that requires the user to provide the matrix and the right side of the discretization of (1.1) only, and that does not require the user to adapt the multigrid method to the problem at hand. Such a multigrid program would be an autonomous ("black box") subroutine. In [4,6,7,12,13] autonomous multigrid subroutines are described that perform satisfactorily in two dimensions. Here we study the three-dimensional case $d = 3$.

An autonomous multigrid subroutine for (1.1) should be able to cope with all relevant functions D , u and c . Especially, in the context of flow computations, for D having elements that differ in order of magnitude (anisotropic diffusion, stretched grids), for $L\{|u|/|D|\} \gg 1$. (high Reynolds number flow, L is a characteristic length), and $c \geq 0$. A necessary requirement is, that the smoothing method employed by the multigrid subroutine works in all these cases, i.e. is robust. Because in two dimensional problems incomplete block (incomplete block LU) factorisation (IBLU) (also called incomplete line LU decomposition, ILLU) has been found to be a robust smoothing method [6, 7], we have investigated a generalization of this method to three dimensions. Here we present the results of this investigation, and compare with a few alternative methods.

We note that the application of multigrid methods to (1.1) with $u = 0$ and $d = 3$ has also been investigated in [1].

2. Incomplete block factorisation in three dimensions

We have implemented and analyzed a three-dimensional version of IBLU. This three-dimensional version has been proposed in [10], and is an extension of the two-dimensional incomplete block factorisation described for example in [3, 7].

The discretization of (1.1) and the coarse grid approximations result in a block-tridiagonal matrix $A = L + D + U$ with the following structure:

$$L = \begin{pmatrix} 0 & & & \\ & L_2 & & \\ & & \ddots & \\ & & & L_n \end{pmatrix}, \quad D = \begin{pmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_n \end{pmatrix}, \quad U = \begin{pmatrix} 0 & U_1 & & \\ & 0 & \ddots & \\ & & & U_{n-1} \\ & & & & 0 \end{pmatrix} \quad (2.1)$$

Here n is the number of grid-points in the z -direction; L_i , D_i and U_i are $l_m \times l_m$ matrices, with l , m the number of grid-points in the x - and y -direction, respectively.

The IBLU-factorisation of A is given by

$$(L + \bar{D})\bar{D}^{-1}(\bar{D} + U) = A + R \quad (2.2)$$

with R the error matrix. If one chooses

$$\bar{D} = \begin{pmatrix} \bar{D}_1 & & \\ & \bar{D}_2 & \\ & & \ddots \\ & & & \bar{D}_n \end{pmatrix} \quad (2.3)$$

with $\bar{D}_i$ $lm \times lm$ matrices, and if

$$\bar{D}_1 = D_1, \quad \bar{D}_k = D_k - L_k \bar{D}_{k-1}^{-1} U_{k-1}, \quad k = 2(1)n \quad (2.4)$$

then the factorisation (2.2) is complete ($R = 0$). The factorisation becomes incomplete if one forbids fill-in in $\bar{D}_k$, $k > 1$. For simplicity we describe IBLU only for the sparsity pattern of the standard 7-point finite difference approximation of the Laplacian.

We choose $\bar{D}$ as follows. Let i be a row and j be a column index. Let S_5 be a symbolic name for the sparsity pattern of an $lm \times lm$ matrix with non-zero elements only for $|i-j| = 1, 1, 0$. (hence, S_5 is the familiar 5-point Laplacian sparsity pattern in the (x,y) plane). Similarly, S_3 corresponds to $|i-j| = 1, 0$: the tridiagonal sparsity pattern. By $S_5(A)$ we mean: replace all elements of A outside S_5 by zero, and similarly for S_3 . $\bar{D}$ will have the structure (2.3), with

$$\bar{D}_k = (\bar{L}_k + \Delta_k) \Delta_k^{-1} (\Delta_k + \bar{U}_k) \quad (2.5)$$

as in [3,7,10] for the case of $d = 2$. $\bar{D}_k$ will be the incomplete block factorisation of a matrix $\tilde{D}_k$ with S_5 sparsity pattern:

$$\bar{D}_k = \tilde{D}_k + \bar{R}_k \quad (2.6)$$

The matrices occurring in (2.5) and (2.6) have the following Structure:

$$\bar{L}_k = \begin{pmatrix} 0 & & \\ L_2^k & & \\ & \ddots & \\ & & L_m^k & 0 \end{pmatrix}, \quad \Delta_k = \begin{pmatrix} \Delta_1^k & & \\ & \Delta_2^k & \\ & & \ddots \\ & & & \Delta_m^k \end{pmatrix},$$

$$\bar{U}_k = \begin{pmatrix} 0 & & & & \\ & U_1^k & & & \\ & & \ddots & & \\ & & & U_{m-1}^k & \\ & & & & 0 \end{pmatrix}, \quad \tilde{D}_k = \begin{pmatrix} D_1^k & & & & \\ & L_2^k & & & \\ & & D_2^k & & \\ & & & \ddots & \\ & & & & L_m^k \\ & & & & & D_m^k \end{pmatrix} \quad (2.7)$$

with $D_j^k, U_j^k, L_j^k, \Delta_j^k$ 1×1 matrices; D_j^k and Δ_j^k are tridiagonal, L_j^k and U_j^k are diagonal matrices. The algorithm for constructing L_j^k, Δ_j^k and U_j^k is:

```

 $\tilde{D}_1 := D_1;$ 
for k := 2(1)n do
begin  $\Delta_1^{k-1} := D_1^{k-1};$ 
    for j := 2(1)m do
         $\Delta_j^{k-1} := D_j^{k-1} - S_3\{L_j^{k-1}(\Delta_{j-1}^{k-1})^{-1} U_{j-1}^{k-1}\};$ 
         $\tilde{D}_k := D_k - S_5\{L_k \tilde{D}_{k-1}^{-1} U_{k-1}\}$ 
    end;
 $\Delta_1^n := D_1^n;$ 
for j := 2(1)m do
     $\Delta_j^n := D_j^n - S_3\{L_j^n(\Delta_{j-1}^n)^{-1} U_{j-1}^n\};$ 

```

Given the IBLU factorisation, a smoothing step is applied to the system $Ay = b$ as follows:

$$(L + \bar{D})\bar{D}^{-1}(\bar{D} + U)\delta y = b - Ay, \quad y := y + \delta y. \quad (2.8)$$

A system $(L + \bar{D})\bar{D}^{-1}(\bar{D} + U)z = r$ is solved as follows. We write $z = (z_1, \dots, z_n)^T$ with z_i vectors of length lm , and partition r accordingly. Then we solve $(L + \bar{D})\tilde{z} = r$ with

$$\tilde{z}_1 := \bar{D}_1^{-1} r_1, \quad \tilde{z}_k := \bar{D}_k^{-1} r_k - L_k \tilde{z}_{k-1}, \quad k = 2(1)n. \quad (2.9)$$

Then we solve $(I + \bar{D}^{-1}U)z = \tilde{z}$ with

$$z_n := \tilde{z}_n, \quad z_k := \tilde{z}_k - \bar{D}_k^{-1} U_k z_{k+1}, \quad k = n-1(-1)1 \quad (2.10)$$

Eqs. (2.9) and (2.10) involve solution of systems of the type $\bar{D}_k u = v$. This can be done as follows. We introduce the partition $u = (u_1, \dots, u_m)^T$ with u_i vectors of length l . With the factorisation (2.5) and the partitionings (2.7) we can write

$$\begin{aligned} \tilde{u}_1 &:= (\Delta_1^k)^{-1} v, & \tilde{u}_j &:= (\Delta_j^k)^{-1} (v_j - L_j^k \tilde{u}_{j-1}), \quad j = 2(1)m, \\ & & & (2.11) \\ u_n &:= \tilde{u}_n, & u_j &:= \tilde{u}_j - (\Delta_j^k)^{-1} U_j^k u_{j+1}, \quad j = m-1(-1)1 \end{aligned}$$

Eq. (2.11) involves only the solution of tridiagonal systems.

3. Smoothing analysis

The following smoothing processes will be considered:

PGS: Standard forward Gauss-Seidel (x varying faster than y varying faster than z).

LGS: Line-Gauss-Seidel: simultaneous solution of values on grid-lines in the x-direction, sweeping forward in the y-z plane (y varying faster than z).

PLGS: Plane-Gauss-Seidel: simultaneous solution of values in grid-points in x-y planes, sweeping forward in the z-direction.

IBLU: Incomplete block LU-factorisation.

It is assumed that the coarse grids are constructed by mesh-doubling in all three directions.

PLGS involves accurate solution of two-dimensional subproblems in successive planes. This can be done for example with two-dimensional multigrid or conjugate gradient methods. We expect one PLGS sweep to be more expensive than one IBLU iteration.

Smoothing analysis for IBLU is carried out as follows. It is assumed that the computational grid is infinite, and that all matrices involved are Toeplitz matrices, i.e. all elements on a given diagonal have the same value. All that is required is to obtain these values. This can be done to any required precision with the following iterative algorithm; cf. [7]. In the Toeplitz matrix A we have $D_k = D_1$, $L_k = L_2$, $U_k = U_1$, $L_j^k = L_2^2$, $\Delta_j^k = \Delta_1^1$ etc. for $j, k \in \mathbb{Z}$. The algorithm is:

```

 $\tilde{D}_1 := D_1;$ 
until convergence within desired precision do
  begin
     $\Delta_1^1 := D_1^1$ 
    until convergence within required precision do
       $\Delta_1^1 := D_1^1 - S_3 \{ L_2^2 (\Delta_1^1)^{-1} U_1^1 \};$ 
       $\tilde{D}_1 := D_1 - S_5 \{ L_2 \tilde{D}_1^{-1} U_1 \}$ 
    end;
  end;

```

The error reduction Toeplitz matrix is $(I - \tilde{A}^{-1}A)$ with

$$\tilde{A} = (L + \bar{D})\bar{D}^{-1}(D + U) \quad (3.1)$$

The calculation of reduction factors of Fourier components is then straightforward.

The smoothing analysis of PGS, LGS and PLGS is standard and will not be described.

Table 3.1 gives some smoothing factors for the anisotropic diffusion equation. The discretization is the standard three-dimensional 7-point finite difference approximation. The mesh-size is the same in all three directions.

With respect to table 3.1 we make the following remarks.

a) If there is a dominant plane (as for the first three anisotropic operators) then high frequencies perpendicular to this plane are not well smoothed by any of the methods, except for PLGS in that plane. In problems where the dominant plane has a variable orientation, alternating PLGS (i.e., using not only (x,y) - but also (y,z) - and (z,x) -planes) would work, but this would be an expensive smoothing method compared to IBLU.

b) If there is only one dominant direction, IBLU is always good, but PLGS not.

c) In the case of a dominant plane, there are only a few high frequency modes that are not smoothed by IBLU. Therefore a combination with a conjugate gradient method (as given in [7]) would be effective. This might be an attractive alternative to alternating PLGS in problems where the dominant plane has a variable orientation.

d) Due to the way practitioners choose their discretization, the case of a dominant line is much more common than a dominant plane in three-dimensional oil reservoir flow simulations.

e) The case of a dominant plane can be handled by semi-coarsening, i.e. in multigrid methods the coarse grids retain the same mesh-size as the fine grid in the direction of small diffusion (cf. e.g. [2,11]). This approach falls outside the scope of autonomous subroutines.

f) For anisotropic diffusion problems, pattern relaxation methods (red-black, zebra) behave much like PGS and LGS (see [11]).

operator	ϵ	PGS	LGS	PLGS	IBLU
$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$		.567	.500	.447	.119
$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \epsilon \frac{\partial^2}{\partial z^2}$	10^{-1}	.909	.835	.447	.336
	10^{-2}	.990	.980	.447	.856
	10^{-3}	.999	.998	.447	.984
$\frac{\partial^2}{\partial x^2} + \epsilon \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$	10^{-1}	.909	.835	.833	.299
	10^{-2}	.990	.980	.980	.855
	10^{-3}	.999	.998	.998	.984
$\epsilon \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$	10^{-1}	.909	.909	.833	.379
	10^{-2}	.990	.990	.980	.891
	10^{-3}	.999	.999	.998	.988
$\frac{\partial^2}{\partial x^2} + \epsilon \frac{\partial^2}{\partial y^2} + \epsilon \frac{\partial^2}{\partial z^2}$	10^{-1}	.847	.500	.447	.183
	10^{-2}	.980	.500	.447	.229
	10^{-3}	.998	.500	.447	.248
$\epsilon \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \epsilon \frac{\partial^2}{\partial z^2}$	10^{-1}	.847	.846	.447	.188
	10^{-2}	.980	.981	.447	.221
	10^{-3}	.998	.998	.447	.228
$\epsilon \frac{\partial^2}{\partial x^2} + \epsilon \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$	10^{-1}	.847	.846	.833	.043
	10^{-2}	.980	.981	.980	.076
	10^{-3}	.998	.998	.998	.089

Table 3.1 Smoothing factors for anisotropic diffusion operators in three dimensions.

g) Alternating PLGS will always work. It has been tested extensively in [1] for (1.1) with $u = 0$. The two-dimensional subproblems were solved with a conjugate gradient method. ILU (incomplete point LU factorisation) was also tried in [1], but did not work well, as is to be expected from smoothing analysis. Because of the cost of the PLGS smoother, multigrid was found to be slower than conjugate gradients.

h) In principle, the performance of IBLU may be improved by replacing $\tilde{D}_{k-1}^{-1}$ by a better approximation $\tilde{D}_{k-1}^{-1}$ in the algorithm of section 2.

Next, we turn to the following special case of (1.1), namely the convection-diffusion equation:

$$-\epsilon \Delta \phi + u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} + w \frac{\partial \phi}{\partial z} = f$$

The mesh-size is 1 in all directions; the components u, v, w may be $-1, 0$ or 1 . For easy visualization we place in fig. 3.1 the smoothing factors inside cubes of size 2 with center at the origin. The coordinates of the dot associated with each smoothing factor in fig. 3.1 equal the values of u, v, w . The values not shown for $\epsilon = 0.1, 0.001$ can be inferred from the symmetries present in the case $\epsilon = 1$. We give results only for the two most robust methods of table 3.1: PLGS and IBLU.

As is to be expected, PLGS does not work when the marching direction is against the flow direction. Symmetric PLGS (forwards and backwards) will always work, as does IBLU.

4. Final remarks

We have analyzed the smoothing properties of a number of iterative methods for use in multigrid methods in three dimensions. As test problems, the anisotropic diffusion and the convection diffusion equations have been used. Alternating and symmetric PLGS always works, but is expensive, since it involves solving accurately in planes. Unlike in two dimensions, our version of IBLU does not work in all cases. Namely, it does not work for the anisotropic diffusion equation when there is strong coupling in a two-dimensional surface. If this surface is a plane with fixed orientation, one can combine IBLU with semi-coarsening: coarsening only in the strongly coupled directions. But this precludes the use of what we call autonomous multigrid subroutines.

$\epsilon = 1$, PLGS

$\epsilon = 1$, IBLU

$\epsilon = 0.1$, PLGS

$\epsilon = 0.1$, IBLU

$\epsilon = 0.001$, PLGS

$\epsilon = 0.001$, IBLU

Figure 3.1 Smoothing factors times 1000.

References

1. A. Behie, P.A. Forsyth, Jr.: Multigrid Solution of Three-Dimensional Problems with Discontinuous Coefficients. Appl. Math. Comp. 13, 229-240, 1983.
2. A. Brandt: Guide to Multigrid Development. In [5], pp. 220-312.
3. P. Concus, G.H. Golub, G. Meurant: Block preconditioning for the conjugate gradient method, Lawrence Berkeley Lab. LBL-14856, Berkely, CA (1982), to appear in SIAM J. Sci. and Stat. Comp.
4. J.E. Dendy, Jr.: Black Box Multigrid for Nonsymmetric Problems. Appl. Math. and Comp. 13, 261-284, 1983.
5. W. Hackbusch, U. Trottenberg (eds.): Multigrid methods. Proceedings, Köln-Porz, 1981. Lect. Notes in Math. 960, Springer-Verlag, Berlin, 1982.
6. P.W. Hemker, R. Kettler, P. Wesseling, P.M. de Zeeuw: Multigrid Methods: Development of East Solvers. Appl. Math. Comp. 13, 311-326, 1983.
7. R. Kettler: Analysis and Comparison of Relaxation Schemes in Robust Multigrid and Preconditioned Conjugate Gradient Methods. In [5], pp. 502-534.
8. G. Meurant: The block preconditioned conjugate gradient method on vector computers. BIT 24, 623-633, 1984.
9. J.A. Meijerink, H.A. van der Vorst: Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems. J. Comp. Phys. 44, 134-155, 1981.
10. J.A. Meijerink: Iterative methods for the solution of linear equations based on incomplete factorisation of the matrix. Publication 643, Kon. Shell Expl. Prod. Lab., Rijswijk, The Netherlands, July 1983. Presented at 7th Symp. on Reservoir Simulation, San Francisco, Nov. 16-18, 1983.
11. K. Stüben, U. Trottenberg: Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications. In [5], pp. 1-176.
12. P. Wesseling: Theoretical and practical aspects of a multigrid method. SIAM J. Sci. Stat. Comp. 3, 387-407, 1982.
13. P. Wesseling: A robust and efficient multigrid method. In [5], pp. 614-630.