NASA/CR-1998-208700
ICASE Report No. 98-37

Robust Multigrid Smoothers for Three Dimensional Elliptic Equations with Strong Anisotropies

Ignacio M. Llorente
Universidad Complutense, Madrid, Spain

N. Duane Melson
NASA Langley Research Center, Hampton, Virginia

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, VA

Operated by Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-97046

August 1998

ROBUST MULTIGRID SMOOTHERS FOR THREE DIMENSIONAL ELLIPTIC EQUATIONS WITH STRONG ANISOTROPIES *

IGNACIO M. LLORENTE† AND N. DUANE MELSON‡

Abstract. We discuss the behavior of several plane relaxation methods as multigrid smoothers for the solution of a discrete anisotropic elliptic model problem on cell-centered grids. The methods compared are plane Jacobi with damping, plane Jacobi with partial damping, plane Gauss-Seidel, plane zebra Gauss-Seidel, and line Gauss-Seidel. Based on numerical experiments and local mode analysis, we compare the smoothing factor of the different methods in the presence of strong anisotropies. A four-color Gauss-Seidel method is found to have the best numerical and architectural properties of the methods considered in the present work. Although alternating direction plane relaxation schemes are simpler and more robust than other approaches, they are not currently used in industrial and production codes because they require the solution of a two-dimensional problem for each plane in each direction. We verify the theoretical predictions of Thole and Trottenberg that an exact solution of each plane is not necessary and that a single two-dimensional multigrid cycle gives the same result as an exact solution, in much less execution time. Parallelization of the two-dimensional multigrid cycles, the kernel of the three-dimensional implicit solver, is also discussed. Alternating-plane smoothers are found to be highly efficient multigrid smoothers for anisotropic elliptic problems.

Key words. multigrid methods, anisotropic discrete operators, plane implicit methods, robust multigrid

Subject classification. Computer Science

1. Introduction and motivation. The efficient solution of systems of partial differential equations is a challenging practical problem from a computational point of view. Computer scientists, applied mathematicians, physicists and engineers are contributing to achieving the shared goal of faster, more accurate numerical solutions while always considering the possibilities and limitations of current computers. In fact, progress in numerical solution of these systems requires the effective combination of advances in algorithm development, understanding underlying physics, and computer hardware.

Today, many in the computational community are engaged in finding and developing the *fastest* methods to solve the systems of equations resulting from the numerical discretizations of mathematical models in physics, chemistry, engineering, medicine, etc. This search is one of the main goals of scientific parallel computing because the numerical solution of these systems of equations is compute-intensive and is the limiting factor in the size of problems and complexity of the physics that can be solved numerically.

Consequently, it is necessary to fully understand a numerical method before it is used as a *tool* to solve practical numerical problems. The definitive measures of the efficiency of a method are execution time and memory usage. Execution time is difficult to predict because it depends on the *numerical* and *architectural properties* of the method as well as the performance of the underlying computer. The numerical properties

* This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the first author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-2199. The first author was supported by the *Dirección General de Enseñanza Superior* of the Spanish Government.

†Departamento de Arquitectura de Computadores y Automática, Universidad Complutense, 28040 Madrid, Spain (email: llorente@eucmax.sim.ucm.es).

‡NASA Langley Research Center, Hampton, VA 23681-2199 (email: n.d.melson@larc.nasa.gov).

include its computational complexity: convergence rate and work per cycle in our particular case. The architectural properties of a method include its memory usage, parallel efficiency and data locality. Good data locality has become a valuable property of a numerical method because it assures an efficient use of the hierarchical memory structure (cache) of current microprocessors.

Standard multigrid techniques are efficient methods for solving many types of partial differential equations (pde's) due to their optimal complexity (work linearly proportional to the number of unknowns), optimal memory requirement, and good parallel efficiency and scalability in parallel implementations. Although highly efficient multigrid methods have been developed for a wide class of problems governed by pde's, they are underutilized in production and commercial codes. One reason for this is that the high level of efficiency is not maintained in anisotropic problems, i.e., the convergence rate of standard multigrid methods degenerates on problems that have anisotropic discrete operators. There is intensive ongoing research aimed at combining the high efficiency of multigrid with good robustness so that multigrid becomes more widely used in production and/or commercial codes.

Several methods have been proposed in the multigrid literature to deal with anisotropic operators. One popular approach is to use semi-coarsening where the multigrid coarsening is not applied uniformly to all of the coordinate directions. By selectively NOT coarsening the grid in a certain direction, the anisotropy can be reduced on the coarser grid. This makes it easier for the smoother to eliminate other components of the high frequency error on the coarser grid. (This approach is basically a 'work-around' for a weak smoother.)

Another approach for dealing with anisotropic problems is to develop and use multigrid smoothers which can eliminate all high frequency errors in the presence of strong anisotropies. The present work explores the capabilities of a class of plane-implicit smoothers in the presence of various anisotropies. These plane-implicit schemes are a natural for structured grids. (It is more difficult to apply these plane-implicit schemes to unstructured grids because there is no *natural* definition of a plane or even a line.) Other intermediate alternatives that combine implicit point or line relaxation with partial and full coarsening have been presented in the multigrid literature [5]. In Section 4 we give a brief introduction to several approaches to plane-implicit smoothers. A comparison between the alternatives is difficult because there are many performance parameters involved that result in a great variety of numerical and architectural properties. We do not find one method always better than the others; rather, we find that each one can be optimum in different situations.

Multi-block structured grids are often used in fluid dynamic applications to capture complex geometries and facilitate parallel implementation without dealing with unstructured grids. Inside each block stretched grids are used to obtain improved discretization accuracy near the boundaries where high gradients in the solution are often present. As a result, the local discrete operator may contain strong anisotropies from both the coefficients of the equation and the highly stretched grid. The objective of this report is to study the behavior of traditional plane relaxation methods as robust smoothers for the multigrid solution using full coarsening of these discrete anisotropic operators. The model problem under study is the solution of the anisotropic elliptic model equation on a cell-centered grid, described in Section 3.

This report presents analytical formulae for the smoothing factors of some plane relaxation smoothers with periodic and Dirichlet boundary conditions. The analytical expressions are verified with several numerical experiments with Dirichlet boundary conditions and cell-centered grids. The formulae provide an accurate prediction of the numerical results. The dependence of the convergence rate on the strength of the anisotropy for the model problem on vertex-centered grids has been previously studied for the two-dimensional (2-D) case, for example by Wesseling in [31], and observed for the three-dimensional (3-D) case

with zebra Gauss-Seidel by Thole and Trottenberg [28]. This report contains the first published numerical results and analysis of the behavior of plane relaxation methods for cell-centered grids. The results, presented in Section 6, are very satisfactory. The performance of some plane smoothers improves considerably for strong anisotropies since they effectively become exact solvers.

The numerical results show that zebra Gauss-Seidel does not perform as well as expected, in fact, the standard (lexicographic order) Gauss-Seidel obtains better convergence rates. This seems to contradict the results presented by Thole and Trottenberg in [28] and by Yavneh in [32] for vertex-centered grids. However, the deterioration of the smoothing factor of Gauss-Seidel with odd-even ordering on cell-centered grids has been previously reported by Gjesdal [8] for the 2-D case.

We show that the use of alternating-plane relaxation smoothers yield very efficient robust multigrid solvers. They are considered in the multigrid literature to have poor numerical and parallel properties because of the expensive and parallel inefficient solution of a large number of 2-D problems. However, we demonstrate in Section 5 that an exact solution of the planes is not needed and that just one 2-D multigrid cycle is sufficient which considerably reduces the execution time of a 3-D smoothing sweep. This result is also shown analytically in [28] by Thole and Trottenberg. On the other hand, the solution of each plane by a 2-D multigrid cycle involves the solution of a very large number of tridiagonal systems of equations (in a general case, band structured with constant bandwidth). This is not a problem in a sequential setting because very efficient band solvers exist. However, in a fine-grain parallel setting the tridiagonal systems may be distributed across many processors which leads to a high volume of interprocessor communication. Section 7 studies different alternatives to bridge this drawback.

2. Factors to consider when comparing numerical methods. Computational scientists are interested in increasing the accuracy of their numerical simulations while reducing the execution time required to obtain the solutions. Therefore, the quality of the solver must be considered in terms of program execution time and memory usage, relative to the accuracy obtained in the numerical simulation. Of course, there are other criteria for a *good* code that candidate methods must satisfy: reliability, robustness, portability, maintainability, etc. However, execution time and memory usage are the definitive metrics for the performance of a numerical method. Execution time depends on the *numerical properties* of the method, the performance of the computer and how the code exploits the underlying computer architecture (*architectural properties*).

The numerical properties are usually studied by way of theoretical complexity studies based on problem size. However, one can reach misleading conclusions regarding the numerical efficiency of a method by just comparing complexities. The complexity studies usually represent an asymptotic limit, but for finite practical problem sizes, the results can improve or deteriorate. One example of this situation is a result shown in this paper where, for practical mesh sizes, the smoothing properties of plane smoothers improve considerably relative to the predicted performance for a grid with an infinite number of points. An improvement of the numerical properties of a method can be considerably more beneficial than the improvement of the architecture properties. Improvements in convergence rate compound with each iteration or multigrid cycle whereas improvements in operations-per-second remain linear. However, as discussed in [26], the programmer must be aware of the underlying architecture to use the full potential of current computers; from workstations to high performance platforms.

With the rapid technological and architectural development occuring in the wide variety of computers currently available, it would seem that it is impossible to consider the underlying computer when developing and comparing numerical methods. However, there are some architecture properties that are *essential,* (and likely to remain *essential* over the next few years) to use the full potential of most current computing systems.

Parallelism is one of these architectural properties. Different levels of parallelism are needed for efficiently exploiting the high number of execution pipelined units in the current superscalar processors and for keeping busy the processors in a parallel computer. The trends in high performance computing show that future computers will consist of thousands of commodity microprocessors. In fact, it is presently very common in the numerical literature to consider the parallelism grade as one of the main comparison factors.

The performance of the current superscalar microprocessors indicates an increasing dependence on efficient usage of hierarchical memory structures. Indeed, the maximum performance that can be obtained in current microprocessors is limited by memory access speed. The peak performance of the microprocessors has increased by a factor four to five every three years by exploiting increasing integration density, reducing the clock cycle, and by implementing architectural techniques to take advantage of multiple levels of parallelism in a program. However, memory access time has been reduced by a factor of only 1.5 to 2.0 over the same period. The disparity between improvements in microprocessor speed and memory access speed seems likely to continue over the next few years. Microprocessors may reach a speed of 4 Gflops at the beginning of the next century [19] without a commensurate memory access speedup. The common technique to bridge this gap and *hide* the memory latency problem is by using a hierarchical memory structure with large fast cache memories close to the processor. As a result, the memory structure has a strong impact on the design and development of a code, and the program must exhibit spatial and temporal data locality to make efficient use of the cache memory and so keep the processor busy.

Therefore data locality is becoming as important as parallel efficiency when studying and comparing different algorithms. The effectiveness of data locality has been well demonstrated in the LAPACK project [1] and major research has just begun to develop cache-friendly iterative methods [6, 25]. On most RISC machines, there is an order-of-magnitude increase in performance going from an out-of-cache implementation to a cache-friendly implementation.

3. The numerical problem. We consider the following anisotropic elliptic partial differential equation

$$a\frac{\partial^2 u(x,y,z)}{\partial x^2} + b\frac{\partial^2 u(x,y,z)}{\partial y^2} + c\frac{\partial^2 u(x,y,z)}{\partial z^2} = f(x,y,z) \tag{3.1}$$

This anisotropic Poisson equation is solved on a 3-D rectangular domain $\Omega \subset \Re^3$ with some suitable boundary conditions.

3.1. Discretization. There are two ways to replace a space continuum by a space structured grid. In finite difference discretizations the domain Ω is divided in cells and the conserved quantities are stored at the vertices of these cells (vertex-centered formulation). In finite volume discretizations, the domain Ω is also divided into cells but the conserved quantities are stored at the centers of these cells (cell-centered formulation). Cell-centered grids have been widely used in CFD for the finite volume solution of the incompressible and compressible Navier-Stokes equations. As our goal is to study the behavior of the smoothers in a CFD setting we focus our attention to cell-centered grids.

The nodes of the discretization are given by

$$\begin{aligned} G = \{&x \in \Omega : x = x_j = jh, j = (j_1, j_2, j_3), \\ &h = (h_1, h_2, h_3), j_\alpha = 0, 1, ..., n_\alpha, h_\alpha = 1/n_\alpha, \alpha = 1, ..., 3\}, \end{aligned}$$

the computational grid is given by

$$\begin{aligned} G = \{&x \in \Omega : x = x_j = (j - s)h, j = (j_1, j_2, j_3), s = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}), \\ &h = (h_1, h_2, h_3), j_\alpha = 1, ..., n_\alpha, h_\alpha = 1/n_\alpha, \alpha = 1, ..., 3\}, \end{aligned} \tag{3.2}$$

and the boundary conditions are evaluated at

$$\begin{aligned}\partial G = \{x \in \partial\Omega : \quad & x = x_j = (j+s)h, j = (j_1, j_2, 0), s = (\frac{1}{2}, \frac{1}{2}, 0), \\ & x = x_j = (j+s)h, j = (j_1, j_2, n_3), s = (\frac{1}{2}, \frac{1}{2}, 0), \\ & x = x_j = (j+s)h, j = (j_1, 0, j_3), s = (\frac{1}{2}, 0, \frac{1}{2}), \\ & x = x_j = (j+s)h, j = (j_1, n_2, j_3), s = (\frac{1}{2}, 0, \frac{1}{2}), \\ & x = x_j = (j+s)h, j = (0, j_2, j_3), s = (0, \frac{1}{2}, \frac{1}{2}), \\ & x = x_j = (j+s)h, j = (n_1, j_2, j_3), s = (0, \frac{1}{2}, \frac{1}{2}), \\ & h = (h_1, h_2, h_3), j_\alpha = 1, 2, ..., n_\alpha, h_\alpha = 1/n_\alpha, \alpha = 1, ..., 3\}\end{aligned}$$

Fig. 1 shows a uniform 32x32x32 grid.

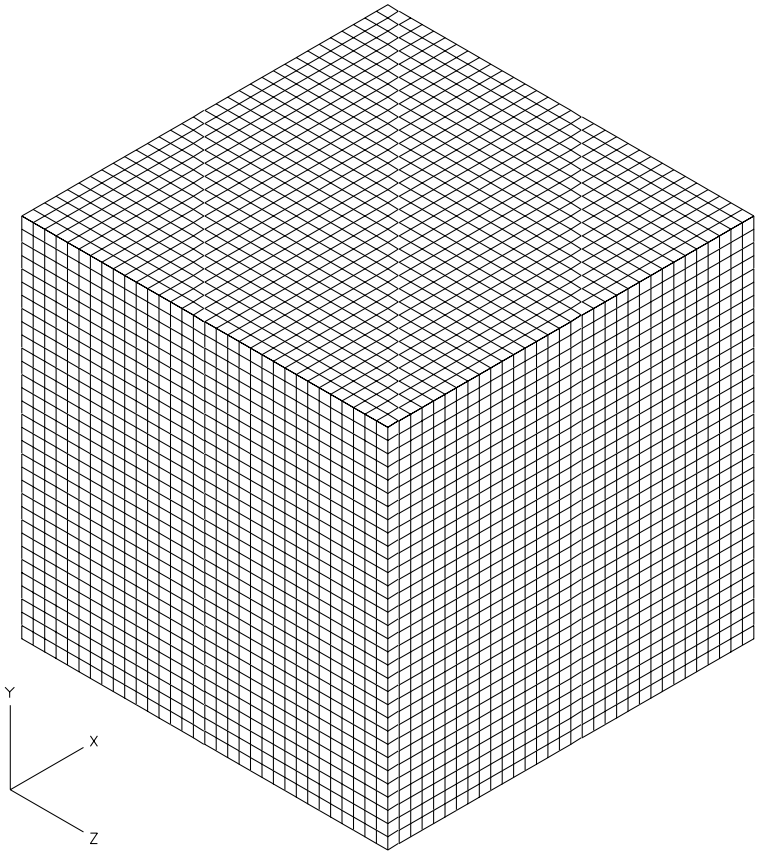

FIG. 1. *32x32x32 uniform grid.*

The following difference equations, involving algebraic relationships between grid points, are obtained

via a discretization of Eq. (3.1) on the cell-centered computational grid (3.2) by a finite volume technique

$$a\frac{u_{i-1,j,k}-2u_{i,j,k}+u_{i+1,j,k}}{h_x^2}+b\frac{u_{i,j-1,k}-2u_{i,j,k}+u_{i,j+1,k}}{h_y^2}+c\frac{u_{i,j,k-1}-2u_{i,j,k}+u_{i,j,k+1}}{h_z^2}=f_{i,j,k}$$
$$i=1,...,n_1, j=1,...,n_2, k=1,...,n_3$$

where $u_{i,j,k}$ is the discrete version of $u(x,y,z)$ function in $(i-\frac{1}{2}, j-\frac{1}{2}, k-\frac{1}{2})$.

This equation can have different space steps in each dimension which results in a different aspect ratio in each dimension. If the equation is then normalized by the coefficient in the z-direction, the following anisotropic discrete operator is obtained:

$$\epsilon_1(u_{i-1,j,k}-2u_{i,j,k}+u_{i+1,j,k})+\epsilon_2(u_{i,j-1,k}-2u_{i,j,k}+u_{i,j+1,k})+(u_{i,j,k-1}-2u_{i,j,k}+u_{i,j,k+1})=f'_{i,j,k}$$
$$i=1,...,n_1, j=1,...,n_2, k=1,...,n_3 \tag{3.3}$$

where ϵ_1 and ϵ_2 are the strength of the anisotropies.

The structured grids used in the present work allow a relatively easy sequential and parallel implementation using, for example, distribution strategies supported by the current versions of High Performance Fortran (HPF). Furthermore, parallel implementation and cache memory exploitation is possible due to the regular data structures in the structured grids.

Grid stretching is commonly used in CFD grid generation to pack points in regions of high gradients in the solution while avoiding having too many points in more benign regions. In the present work, the stretching of the grid in a given direction is determined by the stretching ratio (quotient between two consecutive space steps) or by a boundary fraction parameter (fraction of uniform spacing used for spacing at the boundary). Figure 2 shows a stretched grid with stretching ratio of 1.5 (boundary fraction 0.01) in the x-direction and uniform in the y- and z-directions. Figure 3 shows a stretched grid with a stretching ratio of 1.5 in all three directions.

In a stretched grid, each cell can have different aspect ratios and so the discretization of Eq. (3.1) is given by the following general discrete operator:

$$\begin{aligned}
&\frac{2a}{\Delta x_i}\left(\frac{1}{\Delta x_i+\Delta x_{i-1}}u_{i-1,j,k}-\left(\frac{1}{\Delta x_i+\Delta x_{i+1}}+\frac{1}{\Delta x_i+\Delta x_{i-1}}\right)u_{i,j,k}+\frac{1}{\Delta x_i+\Delta x_{i+1}}u_{i+1,j,k}\right)+\\
&\frac{2b}{\Delta y_j}\left(\frac{1}{\Delta y_j+\Delta y_{j-1}}u_{i,j-1,k}-\left(\frac{1}{\Delta y_j+\Delta y_{j+1}}+\frac{1}{\Delta y_j+\Delta y_{j-1}}\right)u_{i,j,k}+\frac{1}{\Delta y_j+\Delta y_{j+1}}u_{i,j+1,k}\right)+\\
&\frac{2c}{\Delta z_k}\left(\frac{1}{\Delta z_k+\Delta z_{k-1}}u_{i,j,k-1}-\left(\frac{1}{\Delta z_k+\Delta z_{k+1}}+\frac{1}{\Delta z_k+\Delta z_{k-1}}\right)u_{i,j,k}+\frac{1}{\Delta z_k+\Delta z_{k+1}}u_{i,j,k+1}\right)=\\
&\qquad f_{i,j,k}\\
&\qquad i=1,...,n_1, j=1,...,n_2, k=1,...,n_3
\end{aligned} \tag{3.4}$$

where Dirichlet boundary conditions are imposed at the boundaries of the domain. Note that Eq. (3.4) includes the case with variable coefficients, different values of a, b, and c in different parts of the computational domain. Varying grid aspect ratios and values of the equation coefficients cause the strength of the anisotropies to be different in each cell.

The computational experiments presented in this report study separately both sources of anisotropy: anisotropic equation coefficients on uniform grids and isotropic equation coefficients on stretched grids. These represent different situations; the first represents anisotropies that are uniform thoughout the domain and the

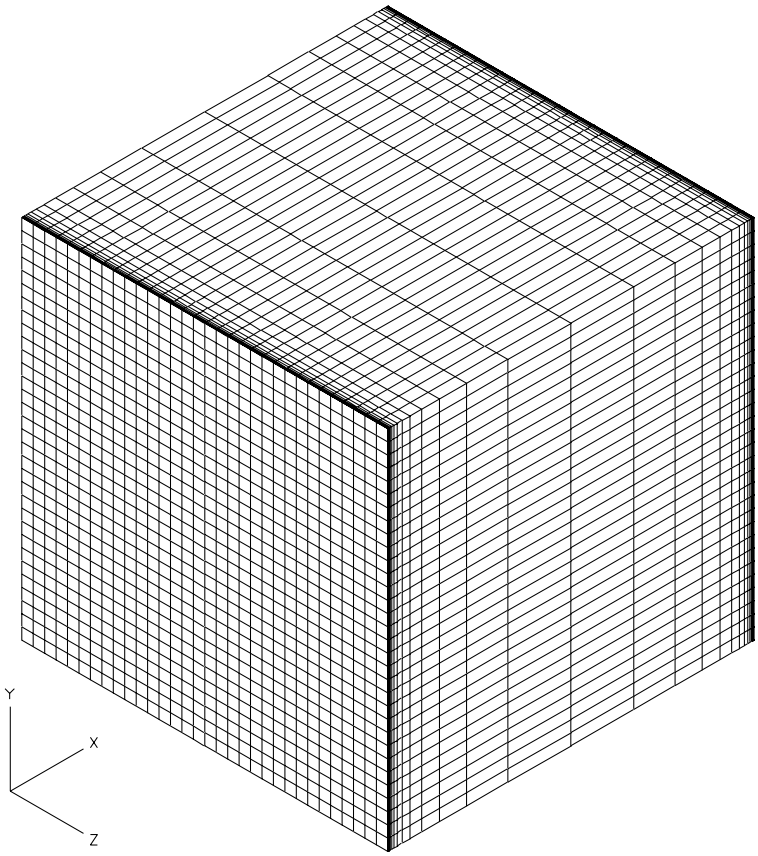

FIG. 2. *32x32x32 stretched grid along x-direction (stretching ratio 1.5 and boundary fraction 0.01).*

latter represents anisotropies that vary from cell to cell in the domain. Note that for exponential stretching in all directions, every cell can have a different level of anisotropy between the coefficients of the discrete operator. On the other hand, when there is exponential stretching in only one direction, two coefficients are similar and the third coefficient, corresponding to the stretching direction, changes in each cell.

If we consider the relative size of the resulting terms in the three coordinate directions, we see that there are several possible scenarios for a given cell:

- all three terms are relatively equal (isotropic with no directions dominant)
- one term is relatively larger than the other two terms (anisotropic with one direction dominant)
- two terms are relatively larger than the third term (anisotropic with two directions dominant)

4. Robust multigrid methods. The multigrid technique has many important advantages from the computational point of view. A well designed multigrid method has, at most, a computational complexity of $O(N \log N)$, where N is the number of equations in the system, to achieve a numerical solution to the level of truncation error [2, 3, 18, 31]. Moreover, these multigrid methods offer very good parallel efficiencies and scalability on parallel computers [15, 16, 17].

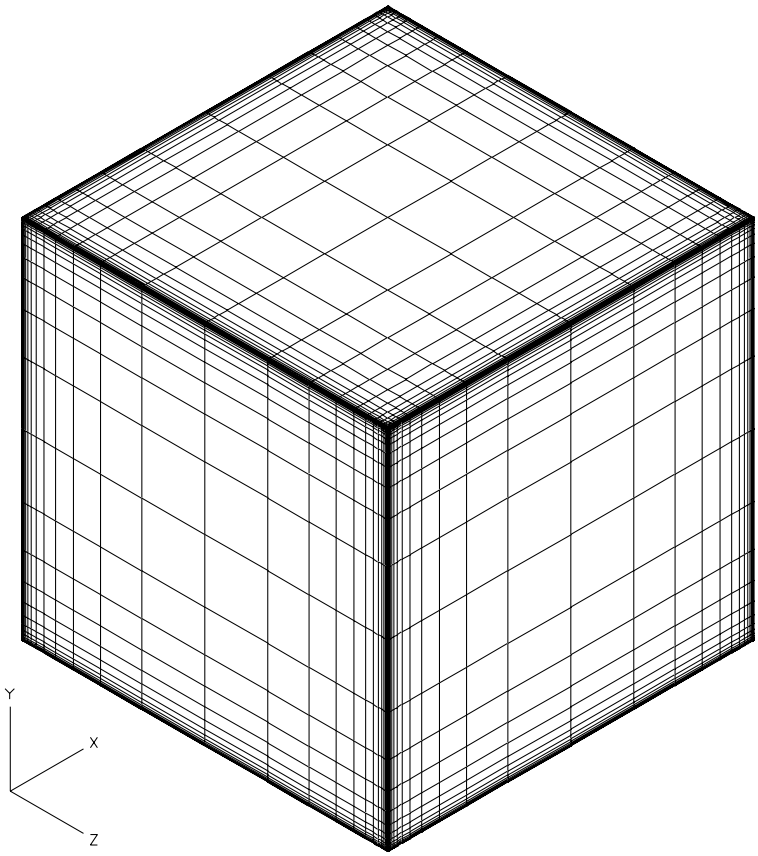

FIG. 3. *32x32x32 stretched grid along all directions (stretching ratio 1.5 and boundary fraction 0.01).*

Multigrid methods rely on a combination of a smoothing process coupled with a coarse grid correction. Multigrid smoothers are often traditional relaxation or incomplete factorization based methods and serve to reduce high frequency components of the error in the solution on the fine grid. The smooth components of the error are more efficiently approximated on coarser grids. Therefore, using a grid hierarchy, we can efficiently reduce the whole frequency domain and achieve the final solution with an optimal operation count.

4.1. The FAS multigrid scheme. The system of equations (3.4) is solved by the *full approximation scheme* (FAS) scheme [2, 18, 31]. This multigrid method is more involved than the simpler correction scheme but can be applied to solve a wider number of problems, such as the solution of nonlinear equations.

In vertex-centered coarsening, a coarse grid is obtained by sampling the next finer grid at every other point. In cell-centered coarsening, a coarse grid is obtained by taking unions of fine grid cells. Consequently, by applying cell-centered coarsening, a sequence $G^l : l = 1, 2, ..., L$ of increasingly coarser grids is obtained

$$G^l = \{x\Omega : x = x_j = (j - s)h^l, j = (j_1, j_2, j_3), s = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}), \\ h^l = (h_1, h_2, h_3), j_\alpha = 1, 2, ..., \frac{n_\alpha}{2^{l-1}}, h^l_\alpha = 2^{l-1}/n_\alpha\} \tag{4.1}$$

The following iterative algorithm represents an FAS (γ_1,γ_2)-cycle to solve the system $Lu = f$ where $\Omega^1 = \Omega$

$u \leftarrow FAS(L, u, f)$

step 1: Apply γ_1 sweeps of the smoothing method on $L^1u^1 = f^1$

Restriction part
FOR l = 2 TO L
step 2: Computation of residual $r^{l-1} = f^{l-1} - L^{l-1}u^{l-1}$
step 3: Restriction of residual $r^l = R^{l-1}r^{l-1}$
step 4: Restriction of current approximation $u^l_{old} = R^{l-1}u^{l-1}$
step 5: Computation of right-hand side $f^l = r^l + L^l u^l_{old}$
step 7: Apply γ_1 sweeps of the smoothing method on $L^l u^l = f^l$

Prolongation part
FOR l = L-1 TO 1
step 8: Correction of current approximation $u^l = u^l - P^l(u^{l+1} - u^{l+1}_{old})$
step 9: Apply γ_2 sweeps of the smoothing method on $L^l u^l = f^l$

The operators that perform the restriction, R (steps 3 and 4), and prolongation, P (step 8), connect the grid levels; the prolongation operator maps data from the coarser level to the current one and the restriction operator takes values from the finer level to the current one. For the restriction operator, we used unweighted averaging for u and a volume weighted sum for the residual. Trilinear interpolation in the computational space was used for the prolongation operator. The smoothing methods are implemented in delta form, which is simpler to program but computationally less efficient.

Plane-implicit smoothers (step 7 and 9) require the solution of a large number of 2-D boundary value problems. For example, an (x,y)-plane smoother requires the solution of n_3 problems ($K = 1, ..., n_3$) given by:

$$\epsilon_1(u_{i-1,j,K} - 2u_{i,j,K} + u_{i+1,j,K}) + \epsilon_2(u_{i,j-1,K} - 2u_{i,j,K} + u_{i,j+1,K}) + (-2u_{i,j,K}) = f^{2-D}_{i,j,K}$$
$$i = 1, ..., n_1, j = 1, ..., n_2$$

where $f^{2-D}_{i,j,K}$ depends on the relaxation method. The resulting 2-D problems are more favorable because the systems may have more diagonal dominance than the original 3-D system. The 2-D problems can be efficiently solved by also using 2-D FAS cycles (Section 5).

Notice that the grids visited in the 2-D coarsening are different from the grids used for the 3-D multigrid. Therefore, the grid metrics for the grid hierarchy to solve the planes does not correspond to the grid metrics for the 3-D grid hierarchy. To precompute all the metrics for the 2-D grid hierarchy would significantly increase the memory requirements of the multigrid code. In fact, the required memory for a 3-D cycle with a point-wise smoother grows as $O(\frac{8}{7}N)$, but to precalculate all the metrics of a plane-wise smoother, it would grow as $O(\frac{4}{3}N)$, which is about 52% larger.

Due to the considerable improvement of convergence rate achieved by multigrid methods, the solution of pde's is moving from *time-critical applications* to *accuracy-critical applications* [17]. In these kinds of

applications, memory usage is the limiting factor for solving larger problems. Savings in computing time are just used to solve bigger problems. Consequently, it is important to maintain the memory requirements of point-wise smoothers when implementing plane-wise smoothers. Therefore, the present scheme is coded using just one 2-D multigrid data structure and recomputing each 2-D system of equations every time a plane is visited. This implementation maintains the memory requirements of the original 3-D multigrid cycle, but increases the execution time. However, because the memory requirements are the same, the performance of the plane smoother can be more exactly compared with the performance of point and line smoothers. This implementation alternative also improves the data locality (temporal and spatial) of the algorithm because the same 2-D multigrid structure is used to solve each plane. The temporal locality is improved because the 2-D data structure fits in cache and the program uses the same data storage addresses when solving each one of the planes. The spatial locality of the data is improved in the current implementation because data are contiguously stored in memory whereas the usage of a global 2-D multigrid hierarchy to store all planes would present different memory access strides depending on the orientation of the plane.

4.2. Robust multigrid methods for anisotropic equations. For a wide class of pde problems, highly efficient multigrid methods have been developed. Unfortunately, this high level of efficiency is not maintained for problems with strong anisotropies.

For example, let us consider the following 2-D model discrete equation

$$\epsilon(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + (u_{i,j-1} - 2u_{i,j} + u_{i,j+1}) = f_{i,j} \qquad i = 1, ..., n_1, j = 1, ..., n_2 \tag{4.2}$$

where $\epsilon > 1$. In this case the standard multigrid algorithm, based on a point-wise smoother, is not a good approach because after few sweeps the error becomes smooth in the x-direction and not in the y-direction.

The two components used in multigrid for error attenuation are the smoothing method, to reduce the high frequency components of the error, and the coarse grid correction, to reduce the low frequency components of the error. Consequently, to maintain good convergence rates on anisotropic operators we have to improve either the smoother or the coarsening process:

- *Keep standard coarsening, but change the smoothing method;* solve simultaneously for those unknowns which are strongly connected (x-line relaxation on Eq. (4.2): lines parallel to x-axis).
- *Keep point-wise smoothing method, but change the coarsening strategy;* define the coarser grid by doubling the mesh size in those directions in which the error is smooth (x-semicoarsening on Eq. (4.2), double the mesh size only in the x-direction).

In the 3-D case, represented by Eq. (3.3), we can extend the previous alternatives by using plane relaxation to solve simultaneously those unknowns which are strongly coupled or by coarsening only those directions in which the error is smooth.

Brandt's fundamental block relaxation rule [2] states that all strongly coupled unknowns (coordinates with relatively larger coefficients) should be relaxed simultaneously. So, if $\epsilon_2 \approx 1$ in Eq. (3.3), we could use x-line relaxation as an efficient smoother. However, if $\epsilon_1 \approx \epsilon_2 \gg 1$ (x,y)-plane relaxation is needed to provide a good smoother. There are other approaches that efficiently combine block and point relaxation with full or partial coarsening, see for example [18].

The block relaxation rule can be extended to blocks that also include relatively small coefficients if the coordinate directions with relatively small coefficients are not coarsened. Although this alternative is more robust, its computational complexity (operation count) increases considerably because it is necessary to solve noncoarsened blocks at each multigrid level. As a result, we can conclude that an efficient smoother with full

coarsening is obtained by block relaxation of the coordinates with relatively larger coefficients if the problem to be solved in the remaining coordinates is isotropic; and the coordinates with relatively smaller coefficients can also be relaxed in the block when the grid is not coarsened in their corresponding directions.

The situation becomes more involved with real applications, see Eq. (3.4). First, the coefficients of the pde can vary throughout the computational domain. Second, the efficient resolution of many pde's requires stretched grids that have mesh spacing that varies several orders of magnitude in different coordinate directions. As a result, the values of the ϵ parameters and their relative magnitudes vary for different parts of the computational domain. For this type of problem, robustness can be achieved following these directions:

- *Robust multigrid smoothing processes with standard coarsening.* Use alternating-line relaxation (x-line smoothing sweep → y-line smoothing sweep) in 2-D and alternating-plane relaxation ((y,z)-plane smoothing sweep → (x,z)-plane smoothing sweep → (x,y)-plane smoothing sweep) in 3-D. Previous attempts of this approach in the multigrid literature have suggested that it has poor numerical and architectural properties since it requires the solution of many 2-D problems and presents a parallelization challenge because it requires the solution of systems of equations which are distributed across many processors.
- *Point-wise smoothing methods with robust coarse grid correction.* The *algebraic multigrid approach* (AMG) combines a point-wise smoother with a fully adaptive coarsening process. The control of the coarsening is done in an adaptive way by semicoarsening in different directions for different parts of the computational domain [14, 24]. AMG involves very complicated data dependent grid structures that break the regularity of geometric multigrid. Hence AMG is mainly used with geometrically complex applications which have been discretized using unstructured grids where there is no natural definition of a global line or plane on which to apply alternating-block smoothers. The *multiple semicoarsening* method was proposed in order to avoid plane relaxation without dealing with the difficulties of AMG [20, 21]. This method employs more than one semicoarsened grid on each coarser level and a recombination of the corrections from each of the coarse grids to yield an optimal efficiency. It is more expensive (execution time and memory usage) than the alternating-block alternative, but provides two levels of parallelism: parallelism on each grid and parallelism across the grids on each multigrid level [23].
- *Simple smoothing processes combined with appropriate semicoarsening strategies.* A simple way to avoid alternating-plane relaxation, but increasing the execution time, is to use relaxation just in a fixed plane and semicoarsening in the remaining direction [10]. Another alternative, which avoids the problem of recombining corrections of the multiple semicoarsening method, is the *flexible multiple semicoarsening* method. This method uses standard coarsening combined with a semicoarsened smoother. The smoother itself corresponds to a V-cycle employing semicoarsened grids while the multigrid cycle is still based on standard coarsening. The method is easy to parallelize [29].

There is also intensive research to achieve robustness by the combination of Krylov methods, such as *generalized minimum residual* (GMRES) and *conjugate gradient* methods [22, 29] with multigrid.

In the appendix, we present the smoothing properties of some plane relaxation smoothers for strong anisotropies. We first show their predicted smoothing factors from Fourier local mode analysis with periodic and Dirichlet boundary conditions and then we present some numerical results to verify the analytical formula. The numerical results show that Fourier analysis provides an accurate prediction of the smoothing factor.

5. The inner 2-D multigrid. Although plane relaxation is simpler and more robust than a semi-coarsening approach, it has not been widely applied in practical situations because it requires the solution of a 2-D problem for each plane in each smoothing sweep. However, we show that an exact solution for the 2-D problems is not needed. An approximate solution is sufficient and can be obtained by applying just one 2-D multigrid cycle.

Table 5.1 compares the experimental 3-D convergence rates obtained with just one 2-D cycle in each plane, with two cycling strategies: V(1,0) and V(1,1), with the convergence rate obtained with a 2-D exact solver (four 2-D (2,1)-cycles). The results are very interesting. With one 2-D V(1,1)-cycle, we obtain the same convergence as with the exact solver with significantly less computational work.

The amplification factor of a 2-D multigrid cycle can be approximated by the smoothing factor of its smoother. Observe that we require a line relaxation smoother for the 2-D cycles because there may be an anisotropy in each plane, for example to solve the (x,y)-planes we need a y-line smoother when $\epsilon_1 \leq \epsilon_2$ and a x-line smoother otherwise. For robustness in the general case where $\epsilon_1 \leq \epsilon_2$ in part of the plane and $\epsilon_1 \geq \epsilon_2$ in part of the plane, an alternating line scheme can be used.

		cycling strategies of the 2-D cycle		
ϵ_1	ϵ_2	(1,0)	(1,1)	Exact
1	1	0.45	0.34	0.34
1	10^2	0.27	0.25	0.25
1	10^4	1.5×10^{-2}	6.1×10^{-3}	6.1×10^{-3}
1	10^6	1.5×10^{-4}	6.1×10^{-5}	6.1×10^{-5}
1	10^8	1.5×10^{-6}	6.2×10^{-7}	6.2×10^{-7}

TABLE 5.1

Computational convergence factors, ρ_e, of one 3-D V(1,0)-cycle with (x,y)-plane Gauss-Seidel for different ϵ_2 and $\epsilon_1 = 1$ solving the planes with one 2-D V(1,0)-cycle, one 2-D V(1,1)-cycle and an exact solver (four 2-D V(2,1)-cycles)

The behavior of the approximated smoothing method can be analyzed as a perturbation of the exact method. The smoothing factor, $\overline{\rho}_a$, of a plane relaxation smoother when γ_{2-D} multigrid cycles with an amplification factor $\overline{\rho}_{2-D}$ are used to solve each plane can be approximated by

$$\overline{\rho}_a \approx \overline{\rho} + (\overline{\rho}_{2-D})^{\gamma_{2-D}} \tag{5.1}$$

where $\overline{\rho}$ is the smoothing factor of the plane smoother with an exact 2-D solver.

As is shown in Section A.5, the smoothing factor of a 2-D line relaxation method is quite similar to its corresponding 3-D plane version ($\overline{\rho} \approx \overline{\rho}_{2-D}$). Consequently, if we use the same block relaxation method in 2-D and 3-D cycles, the 3-D smoothing factor with one 2-D V(1,0)-cycle is given by

$$\overline{\rho}_a \approx \overline{\rho} + (\overline{\rho}_{2-D}) \approx 2\overline{\rho}$$

and with one 2-D V(1,1)-cycle it could be approximated by

$$\overline{\rho}_a \approx \overline{\rho} + (\overline{\rho}_{2-D})^2 \approx \overline{\rho}$$

Observe that the decrease of the convergence rate for strong anisotropies is also presented in the 2-D smoother because ϵ_1 remains fixed and ϵ_2 increases, so the 2-D problem solved in each plane is also anisotropic. However, if both anisotropy values are increased together, the 3-D problem is anisotropic but

the 2-D problem does not present an anisotropy and so the convergence rate of the 2-D problem remains fixed, becoming the dominant term in (5.1)

$$\overline{\rho}_a \approx \overline{\rho} + (\overline{\rho}_{2-D})^{\gamma_{2-D}} \approx \overline{\rho}_{2-D}^{\gamma_{2-D}}$$

		cycling strategies of the 2-D cycle		
ϵ_1	ϵ_2	(1,0)	(1,1)	Exact
1	1	0.45	0.34	0.34
10^2	10^2	0.37	0.14	0.20
10^4	10^4	0.37	0.14	4.6×10^{-4}
10^6	10^6	0.37	0.14	2.8×10^{-6}
10^8	10^8	0.37	0.14	3.3×10^{-8}

TABLE 5.2

Computational convergence factors, ρ_e, of one 3-D V(1,0)-cycle with (x,y)-plane Gauss-Seidel for different ϵ_2 and ϵ_1 solving the planes with one 2-D V(1,0)-cycle, one 2-D V(1,1)-cycle and an exact solver

This behavior is illustrated by Table 5.2. Considering these results we could conclude that we can not approximate the 2-D solver when both anisotropies are strong. However, we must take into account that one smoothing sweep with the exact solver is considerably more expensive than the smoothing sweep with an approximated solver and so the overall efficiency can be better for the approximated method.

Fig. 4 shows the residual versus work units for 3-D V(2,1)-cycles to solve four anisotropic equations on a 32x32x32 uniform grid with five smoothers:

- point-wise Gauss-Seidel (point),
- y-line Gauss-Seidel (plus),
- (x,y)-plane Gauss-Seidel with exact solver in each plane (square),
- (x,y)-plane Gauss-Seidel with one 2-D V(1,1)-cycle in each plane (star), and
- (x,y)-plane Gauss-Seidel with one 2-D V(1,0)-cycle in each plane (circle),

These results allow a comparison of the performance of the plane approximate solvers with the plane exact solvers and with the behavior of point and line smoothers. The smoother used in the 2-D cycles to solve the planes is y-line Gauss-Seidel. Each symbol is drawn at the completion of a 3-D multigrid cycle to compare the computational complexity of the cycles corresponding to different smoothers. Here a work unit is conservatively defined as the computer time consumed in a residual computation on the finest grid (the time to perform one point-wise iteration on the finest grid is about two work units).

As indicated in Fig. 4, the approximate plane solution version with one 2-D V(1,1)-cycle (star) is more efficient than the approximate version with one V(1,0)-cycle (circle) and the exact version (square). Even when the 2-D problem solved in each plane is isotropic and the remaining direction is effectively decoupled (a=b=10000, c=1), it is not worth using the exact solver because each 3-D cycle consumes too much time. The behavior of the plane smoother for strong anisotropies is absolutely good. The residual reduction per work unit or cycle increases as the anisotropy gets stronger (the two graphics at the bottom of the figure show a similar residual reduction per work unit). In fact, when $\epsilon_2 = 10^4$ the solution is achieved to the level of truncation error in just two 3-D cycles (about 50 work units).

It is illustrative to study the behavior of point and line Gauss-Seidel for these cases. The line smoother is more efficient than the point-wise version for the isotropic problem. The work per cycle is slightly greater in the line version, however the asymptotic convergence rate of the line version is 1.42 times lower (better). On

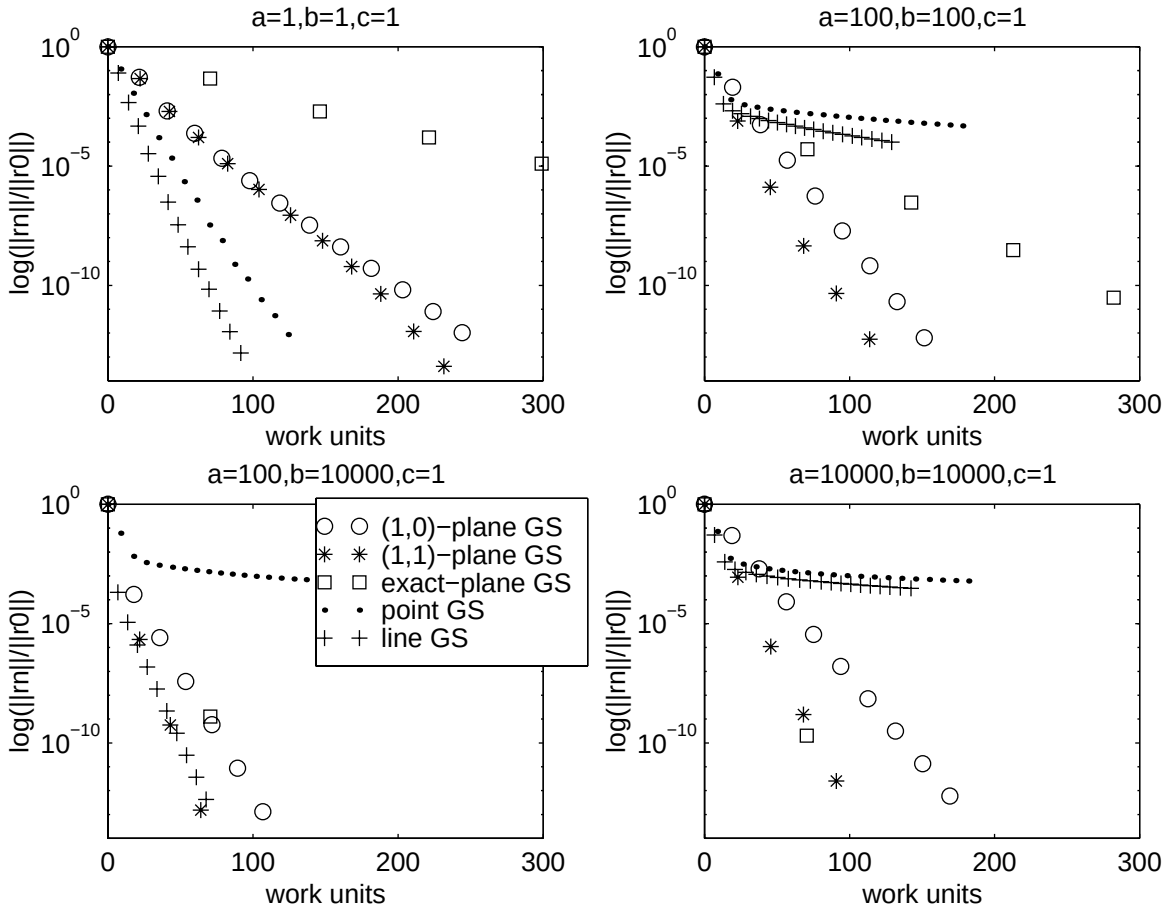

FIG. 4. *Residual versus work units for 3-D V(2,1)-cycles to solve four anisotropic equations on a 32x32x32 uniform grid with the following five smoothers: point-wise Gauss-Seidel (point), y-line Gauss-Seidel (plus), (x,y)-plane Gauss-Seidel with exact solver in each plane (square), (x,y)-plane Gauss-Seidel with one 2-D V(1,1)-cycle in each plane (star), and (x,y)-plane Gauss-Seidel with one 2-D V(1,0)-cycle in each plane (circle). The smoother used in the 2-D cycles is y-line Gauss-Seidel. Each symbol represents a 3-D cycle in order to compare the complexity of the cycles of different smoothers.*

the other hand, the plane smoother exhibits a less efficient behavior for the isotropic case. The performance of the plane smoother is twice as slow as the observed performance of the point smoother. The plane smoother reduces the error more per multigrid cycle than the line smoother but its operation count is so much higher that it ends up being slower form the isotropic case. The computer program used to generate the present results is not fully optimized. It was coded to deal with many different methods and situations and so this result may change some with coding practice.

As was expected, the point smoother gives very poor convergence rates for the anisotropic problems. However, the line smoother performs very fast when $\epsilon_2 \gg \epsilon_1$, as is demonstrated in Section A.5. This is because a single direction dominates the solution in this case.

Fig. 5 shows the residual versus work unit for 3-D V(1,0)-cycles to solve the isotropic equation on four 32x32x32 stretched grids with the following five smoothers:

- point-wise Gauss-Seidel (point),
- alternating-line Gauss-Seidel (plus),

- alternating-plane Gauss-Seidel with exact solver in each plane (square),
- alternating-plane Gauss-Seidel with one 2-D V(1,1)-cycle in each plane (star), and
- alternating-plane Gauss-Seidel with one 2-D V(1,0)-cycle in each plane (circle).

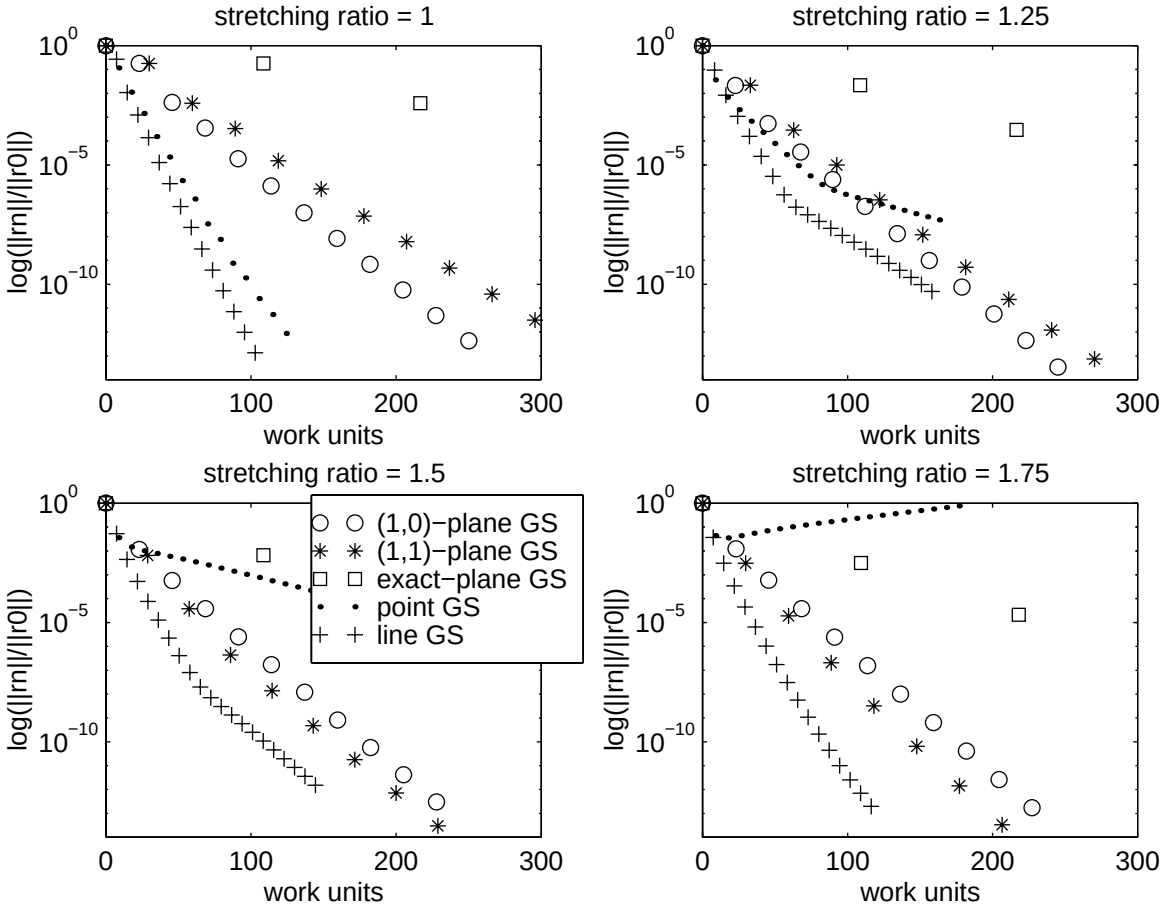

FIG. 5. *Residual versus work unit for 3-D V(1,0)-cycles to solve the isotropic equation on four 32x32x32 stretched grids with the following five smoothers: point-wise Gauss-Seidel (point), alternating-line Gauss-Seidel (plus), alternating-plane Gauss-Seidel with exact solver in each plane (square), alternating-plane Gauss-Seidel with one 2-D V(1,1)-cycle in each plane (star) and alternating-plane Gauss-Seidel with one 2-D V(1,0)-cycle in each plane (circle). The smoother used in the 2-D cycles is alternating-line Gauss-Seidel. Each symbol represents a 3-D cycle in order to compare the complexity with the cycles of different smoothers.*

The smoother used in the 2-D cycles to solve the planes is alternating-line Gauss-Seidel.

In the two cases of stretching ratio equal to 1.0 and 1.25, the approximate plane solver with one V(1,0)-cycle performs better than either the approximated solver with one V(1,1)-cycle or the exact solver. However, as the stretching ratio increases, the performance for the approximate plane solver with one V(1,1)-cycle improves. All cases show that the exact solver gives much worse performance.

Fig. 5 also shows an unexpected behavior of the alternating-line smoother. This optimal behavior is due to the use of stretching along the three directions that produces high discrepancies for the local values of the anisotropy in each cell, and therefore the local dominance of one of the directions. In fact, the numerical results obtained with grids stretched along just one direction show a poor behavior of the alternating-line

smoother.

6. Comparison of the plane smoothers. Fig. 6 shows residual versus work unit for 3-D V(2,1)-cycles to solve four anisotropic equations on a 32x32x32 uniform grid with the following five (x,y)-plane smoothers:

- Gauss-Seidel (solid),
- zebra Gauss-Seidel (dashed),
- four-color Gauss-Seidel (star),
- Jacobi with damping 0.7 (dotted), and
- Jacobi with partial damping 0.7 (dash dot).

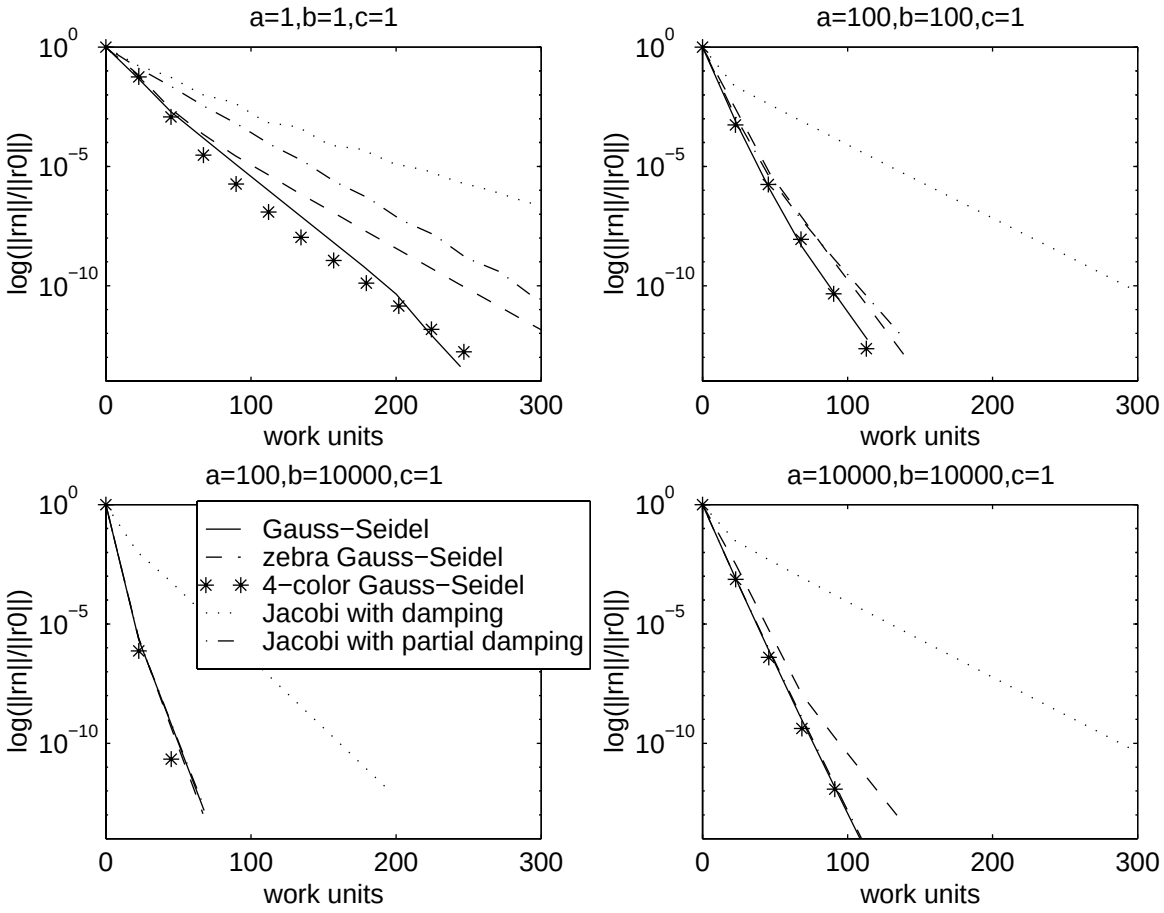

FIG. 6. *Residual versus work unit for 3-D V(2,1)-cycles to solve four anisotropic equations on a 32x32x32 uniform grid with the following five (x,y)-plane smoothers: Gauss-Seidel (solid), zebra Gauss-Seidel (dashed), four-color Gauss-Seidel (star), Jacobi with damping 0.7 (dotted), and Jacobi with partial damping 0.7 (dash dot). One 2-D V(1,1)-cycle is used to solve each plane. The smoother used in the 2-D cycles is the y-line version of the one used in the 3-D cycles.*

The best results are consistently obtained with the four-color plane Gauss-Seidel method.

On the other hand, Fig. 7 shows residual versus work unit for 3-D V(1,1)-cycles to solve the isotropic equation on four 32x32x32 stretched grids with the following five alternating-plane smoothers:

- Gauss-Seidel (solid),

- zebra Gauss-Seidel (dashed),
- four-color Gauss-Seidel (star),
- Jacobi with damping 0.7 (dotted) and
- Jacobi with partial damping 0.7 (dash dot).

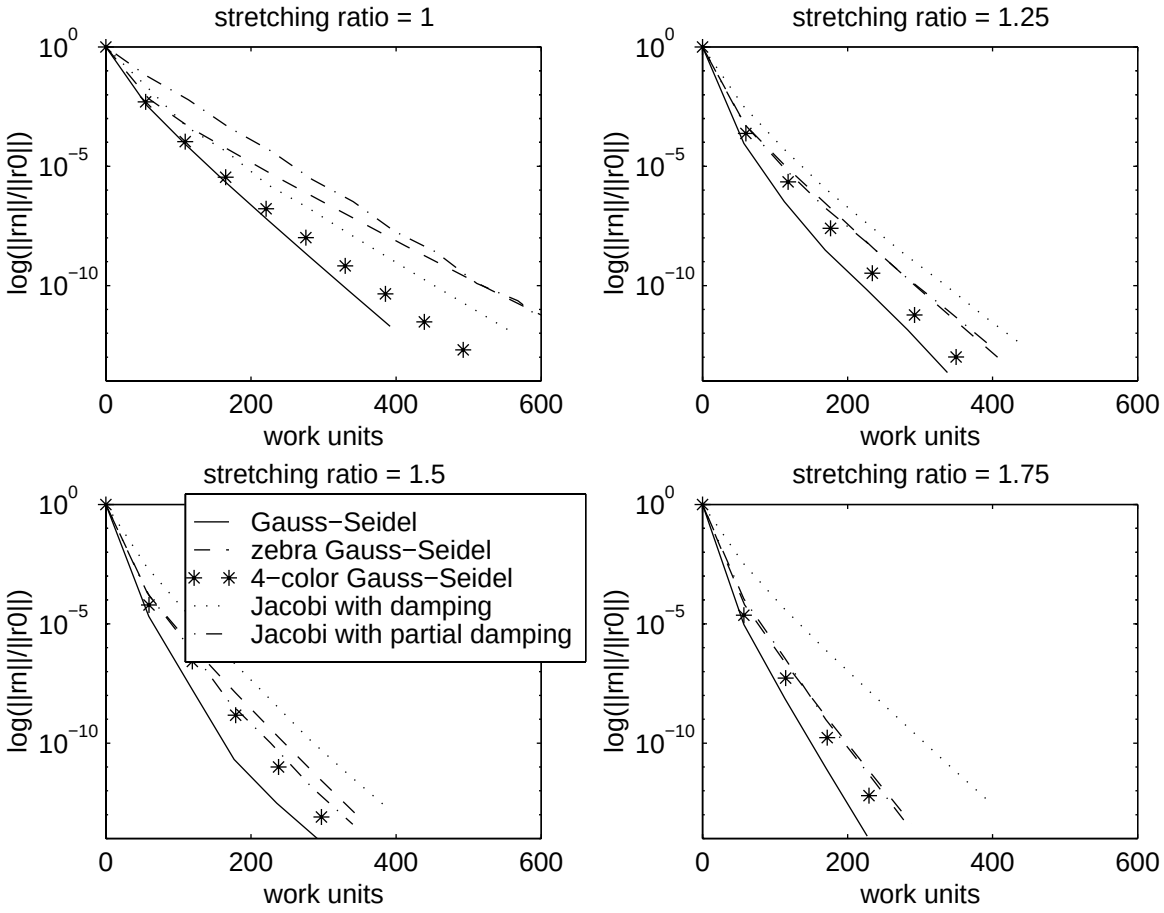

FIG. 7. *Residual versus work unit for 3-D V(1,1)-cycles to solve the isotropic equation on four 32x32x32 stretched grids with the following five alternating-plane smoothers: Gauss-Seidel (solid), zebra Gauss-Seidel (dashed), four-color Gauss-Seidel (star), Jacobi with damping 0.7 (dotted) and Jacobi with partial damping 0.7 (dash dot). One 2-D V(1,1)-cycle is used to solve each plane. The smoother used in the 2-D cycles is the alternating-line version of the one used in the 3-D cycles.*

In this case, the best results are obtained with the lexicographic ordering.

In general, the three Gauss-Seidel plane implicit methods and the Jacobi plane implicit method with partial damping give similar results with anisotropic equations. Jacobi with damping performs worse because its smoothing factor does not improve with the anisotropy. However, four-color Gauss-Seidel performs better in the isotropic case. About the parallel implementation, zebra Gauss-Seidel, four-color Gauss-Seidel and Jacobi methods are fully parallelizable, however the Jacobi method is likely to give better parallel efficiencies because of its coarser granularity.

The improvement of the convergence rate for strong anisotropies is not expected to deteriorate considerably for increasing mesh sizes. The residual versus work units for 3-D V(2,1)-cycles to solve four anisotropic equations on a uniform grid with (x,y)-plane four-color Gauss-Seidel and the isotropic equation

on four stretched grids with alternating-plane four-color Gauss-Seidel for grid sizes 32x32x32, 64x64x64 and 128x128x128 are given in Fig. 8 and 9 respectively. These figures show that the good behavior for strong anisotropies deteriorates slightly for larger grid sizes. The discrepancies in the isotropic case show the dependence of the convergence rate, ρ, on the problem size; it tends to $\overline{\rho}$, the predicted convergence rate for periodic boundary conditions.

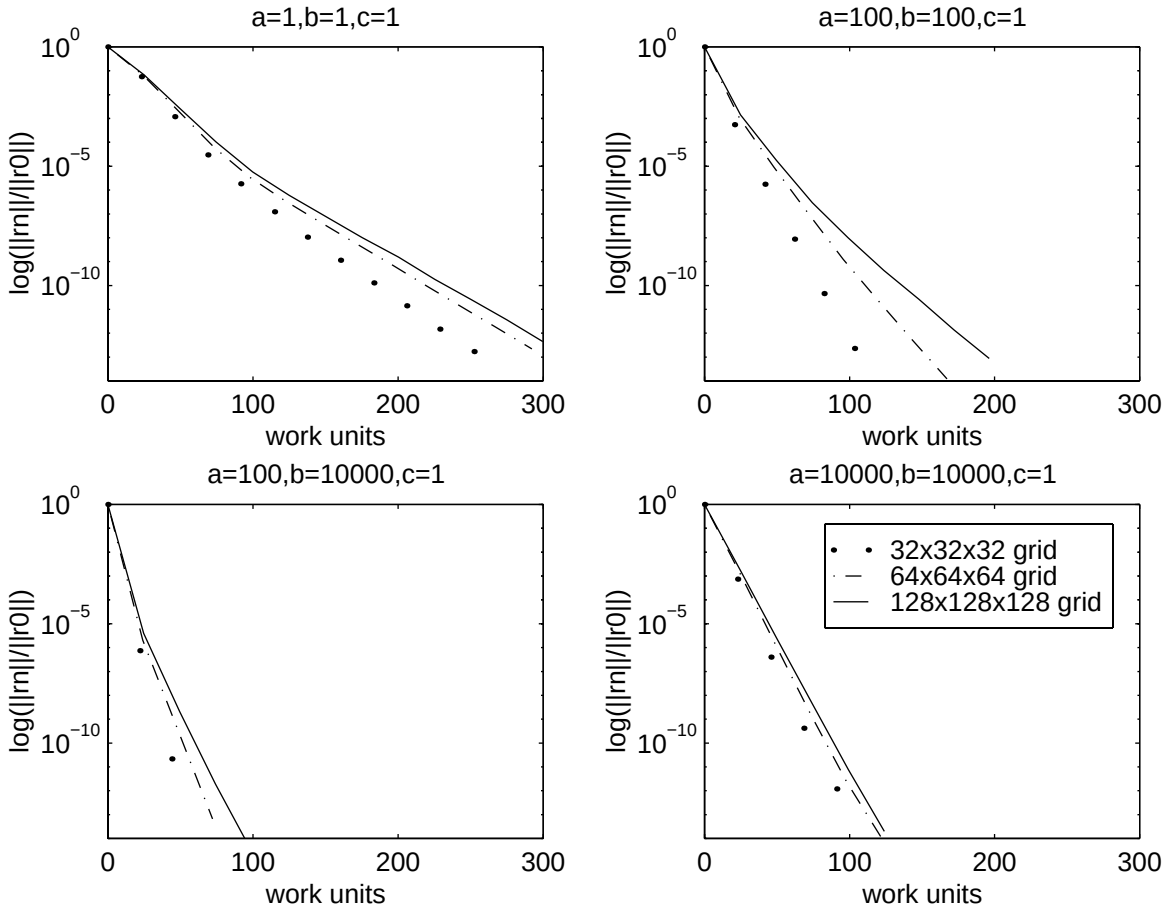

FIG. 8. *Residual versus work unit for 3-D V(2,1)-cycles to solve four anisotropic equations with (x,y)-plane four-color Gauss-Seidel with the following grid sizes: 32x32x32 (dotted), 64x64x64 (dash dot) and 128x128x128 (solid). One 2-D V(1,1)-cycle is used to solve each plane. The smoother used in the 2-D cycles is y-line four-color Gauss-Seidel.*

7. Efficiency considerations. One of the drawbacks mentioned in the literature against plane relaxation smoothers is their inefficiency on parallel computers. A plane relaxation sweep implies the solution of a large number of tridiagonal systems of equations. For example, with (x,y)-plane relaxation and using y-line relaxation inside each plane, it is necessary to solve $n_1 n_3$ systems of equations $(I = 1, ..., n_1; K = 1, ..., n_3)$ in each 3-D smoothing sweep

$$\epsilon_1(-2u_{I,j,K}) + \epsilon_2(u_{I,j-1,K} - 2u_{I,j,K} + u_{I,j+1,K}) + (-2u_{I,j,K}) = f^{1-D}_{I,j,K} \qquad j = 1, ..., n_2 \tag{7.1}$$

The improvement of the parallel efficiency of tridiagonal solvers has been a focus of intensive research

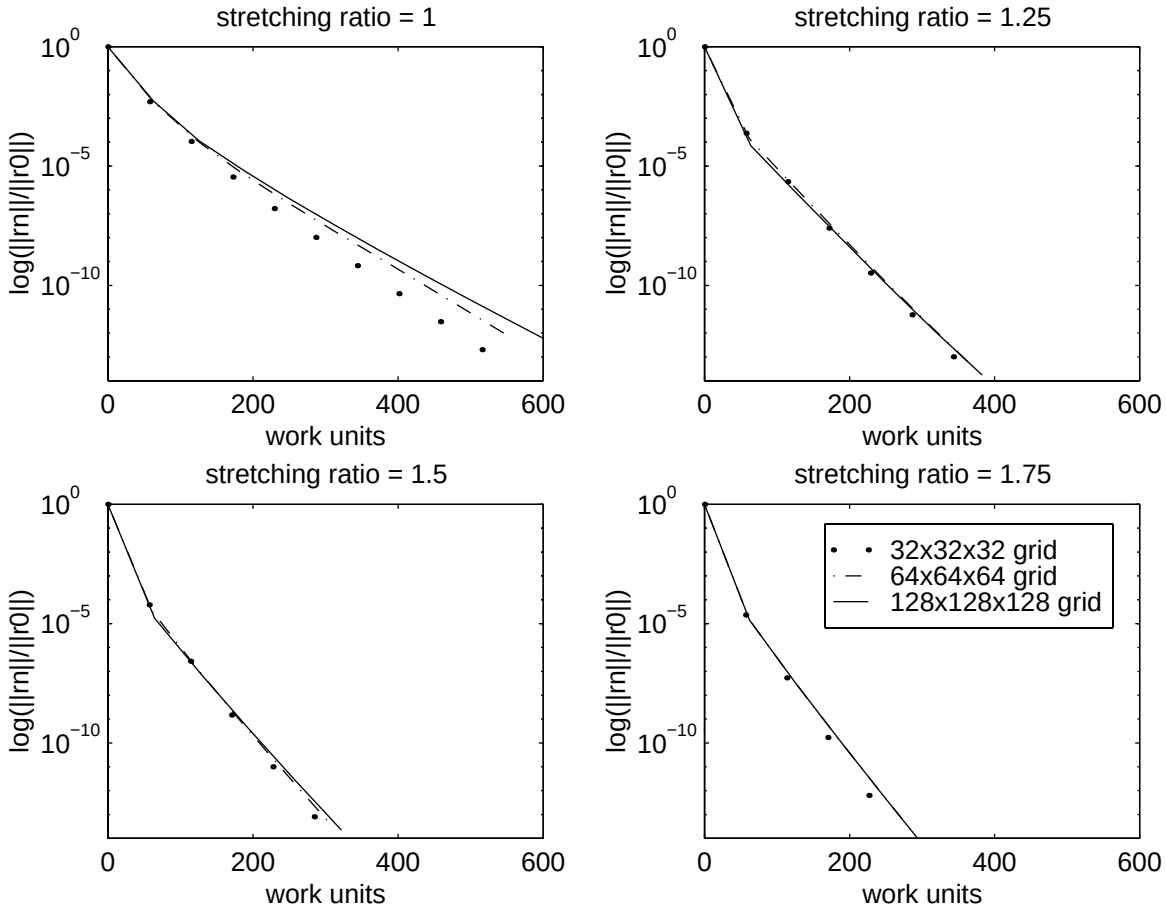

FIG. 9. *Residual versus work unit for 3-D V(1,1)-cycles to solve the isotropic equation on four stretched grids with alternating-plane four-color Gauss-Seidel with the following grid sizes: 32x32x32 (dotted), 64x64x64 (dash dot) and 128x128x128 (solid). One 2-D V(1,1)-cycle is used to solve each plane. The smoother used in the 2-D cycles is the alternating-line four-color Gauss-Seidel.*

in the last few years, see for example [7, 9, 11, 13]. Most of these methods improve their computational count when the system is strongly diagonally dominant, which is a likely situation in Eq. (7.1). Tridiagonal systems could also be solved by a multigrid algorithm. It is expected that the solution could be reached in just one 1-D V(1,1)-cycle (extending the results obtained in Section 5 for the 2-D cycles to solve the planes). The smoothing sweeps can be implemented so as to make more efficient use of cache memory than sequential tridiagonal solvers. The parallel tridiagonal solvers sweep over the data stored in memory at least two times and so the data within the cache is invalidated for large problems. However, the 1-D multigrid method can be implemented to pass over the data just one time for a number of sweeps, improving its temporal locality [6, 25].

Another easy and efficient alternative for dealing with the parallel implementation of the plane solvers is by blocking the smoother. The blocks are distributed across the processors and the plane relaxation is performed only within the blocks of the grid. The blocking of the plane relaxation also allows the application of plane smoothers with multi-block structured grids because only a local definition of the line or plane is needed. The blocking improves the temporal locality of the smoother. If the block fits entirely in cache, the

plane solver can pass over the data just one time for a number of sweeps and so the smoother makes more efficient use of the cache memory. Moreover, it provides an additional advantage. A different smoother could be used in each block depending on the local strength of the anisotropy and the stretching direction. For example, plane smoothers might be confined to the blocks in the region with stronger anisotropies. However, this blocking alternative causes a deterioration of the smoothing properties of the method. Jones and Melson show in [12] that for line smoothers in the 2-D case, true multigrid efficiency is achieved only when the block sizes are proportional to the strength of the anisotropy. Further, the blocks must overlap and the size of the overlap must again be proportional to the strength of the anisotropy. (True multigrid efficiency is the convergence and operation count obtained by multigrid using a point Gauss-Seidel smoother on an isotropic elliptic problem.)

8. Conclusions and future directions. We have shown numerically and analytically the smoothing factors of traditional plane relaxation methods with Dirichlet boundary conditions. The smoothing performance of the following relaxation methods have been investigated for the multigrid solution of a discrete elliptic model equation on a cell-centered grid with strong anisotropies:

- Plane Jacobi with damping
- Plane Jacobi with partial damping
- Plane Gauss-Seidel
- Plane zebra Gauss-Seidel
- Plane four-color Gauss-Seidel
- Line Gauss-Seidel

All of the plane-implicit schemes studied but plane Jacobi with damping show a linear decrease of smoothing factor with increasing anisotropy strength, regardless of the relative strengths of the two anisotropies possible in 3-D problems considered. The good behavior of the plane smoothers deteriorates very slightly with an increase in the number of cells per side in the grid. Consequently, we feel that their excellent performance can be maintained for large practical problems. Although line smoothers give very good results when one of the anisotropies is much larger than the other, line smoothers perform poorly when both anisotropies are similar and, hence, can not be considered for a robust method.

The numerical results indicate that zebra Gauss-Seidel does not perform as well as expected on cell-centered grids. In fact, the lexicographic order Gauss-Seidel obtains better convergence rates and four-color plane Gauss-Seidel becomes an attractive smoother because of its good convergence rates and parallel properties. Plane Jacobi with partial damping is also a very good alternative; it performs worse in the isotropic case but exhibits coarser granularity in a parallel setting.

The practical feasibility of plane relaxation as a multigrid smoother has been demonstrated. The solution of the 2-D boundary-value problem corresponding to each plane can be approximated with just one 2-D multigrid cycle. The same behavior can be expected if 1-D multigrid is applied to the solution of the tridiagonal systems of equations involved in the plane solution. As a result, alternating-plane smoothers are just two times slower than point-wise smoothers in the isotropic case and are robust multigrid smoothers that are orders-of-magnitude faster for anisotropic operators. Moreover, plane smoothers are an alternative that are easy to program, both on sequential and parallel computers.

We are interested in the applicability of plane smoothers with multi-block grids. Therefore, we will continue working on block-structured plane smoothers. In particular, we want to study the behavior of blocked plane smoothers and determine if the results relating block size, overlap, and anisotopy strength obtained by Jones and Melson in [12] hold in the 3-D case and in more complicated pde's and problem

geometries. We feel that the smoothing performance will not suffer excessive deterioration with domain decomposition.

9. Acknowledgements. The authors would like to acknowledge particularly James L. Thomas and Manuel Salas for their valuable comments on the CFD discipline and Fourier local mode analysis. James L. Thomas proposed the use of partial damping in the Jacobi method to improve its behavior for strong anisotropies. We also would like to thank David Sidilkover, Dimitri J. Mavriplis, Thor Gjesdal, Kees Oosterlee and Francisco Gaspar for their valuable comments and for providing a copy of their work.

We would also like to thank Veer Vatsa for his many words of encouragement and his unending enthusiasm in the search for faster multigrid methods.

Appendix A. Smoothing factors of plane relaxation methods.

A multigrid method consists of two different components; a smoother to remove the high-frequency components of the error in the solution and a succession of grid coarsening to eliminate the low-frequency components by using the smoother on such grids. The behavior of a multigrid method strongly depends on the smoother. Indeed, the development of a good smoother is the critical step when developing multigrid algorithms.

A simple and convenient tool to study the smoothing properties of a relaxation method is the Fourier smoothing analysis. The results of this analysis give a measure of the quality of a numerical method. Occasionally, it is too involved to obtain an analytical expression for the smoothing factor and so it must be calculated numerically. However, explicit formulas are preferred because they give more information about the dependence of the smoother on the involved parameters.

The Fourier smoothing analysis does not consider the intergrid transfer process and the discrepancy between the coarse grid and fine grid discrete approximations of the operator, so the actual numerical performance can vary slightly from the predicted performance. To obtain a more accurate prediction, a two-level analysis, which takes into account the operations and discrepancies between levels, must be applied. Results of the Fourier analysis can be found in the literature, for example see [27, 30, 31] for the 2-D case and [32] for the 3-D case (not including results presented in this report).

In spite of the fact that the Fourier analysis gives the same results for vertex-centered and cell-centered grids, we have noticed discrepancies between both cases with zebra plane Gauss-Seidel in our numerical experiments. We show how the lexicographic order performs significantly better than the zebra ordering. These discrepancies have also been reported by Gjesdal [8] for the 2-D case. They may be caused by the coarse grid correction and may be reflected in a two-level analysis.

Our methodology and notations are similar to those used by Wesseling [31]. Assume a matrix representation of the system of equations (3.3) denoted by

$$Lu = f$$

and let the smoothing method given by the following splitting

$$u = Su + M^{-1}f, S = M^{-1}N, M - N = L$$

The error matrix is $M^{-1}N$ and the error after γ iterations of the smoothing method is given by

$$e^{m+1} = S^{\gamma}e^{m}$$

We can expand the error in a Fourier series of the eigenfunctions or local modes $\psi(\theta)$ of S

$$e_j^m = \sum_{\theta \in \Theta} c_\theta^m \psi_j(\theta), \psi_j(\theta) = e^{ij\theta}$$

with $i = \sqrt{-1}$, $j \in I$, and $\theta \in \Theta$

$$I = \{j : j = (j_1, j_2, j_3), j_\alpha = 1, 2, ..., n_\alpha, \alpha = 1, ..., 3\}$$
$$\Theta = \{\theta : \theta = (\theta_1, \theta_2, \theta_3), \theta_\alpha = \frac{2\pi k_\alpha}{n_\alpha}, k_\alpha = -\frac{n_\alpha}{2}, ..., \frac{n_\alpha}{2}, \alpha = 1, ..., 3\}$$

Hence

$$S^\gamma \psi(\theta) = \lambda^\gamma(\theta)\psi(\theta)$$

where $\lambda(\theta)$ is the eigenvalue corresponding to $\psi(\theta)$ and

$$c_\theta^{m+1} = \lambda^\gamma(\theta) c_\theta^m$$

The eigenvalue is the amplification factor, so the largest of the eigenvalues in absolute value, the so called spectral radius, determines the rate of convergence of the relaxation method.

We can distinguish between high-frequency or rough eigenfunctions ($\theta \in \Theta_r$) and low-frequency or smooth eigenfunctions ($\theta \in \Theta_s$). If we want to study the rate of convergence of the relaxation method the *damping factor* is given by the largest of the eigenvalues over all frequencies (Θ). However if we are interested in the *smoothing factor* of the method we must consider just the rough components (Θ_r). Rough and smooth components are given by

$$\Theta_s = \Theta \cap (-\frac{\pi}{2}, \frac{\pi}{2})^3, \Theta_r = \Theta \setminus \Theta_s$$

Let us define ρ as the local mode smoothing factor

$$\rho = sup\{|\lambda(\theta)| : \theta \in \Theta_r\},$$

so that the error over the rough components is multiplied by a factor ρ after each smooth iteration. Note that ρ depends on the problem size n_α because Θ_r depends on the problem size.

However, Θ_r tends to

$$\overline{\Theta}_r = [-\pi, \pi]^3 \setminus (-\frac{\pi}{2}, \frac{\pi}{2})^3$$

as $n_\alpha \uparrow$, and we can define

$$\overline{\rho} = sup\{|\lambda(\theta)| : \theta \in \overline{\Theta}_r\},$$

even so $\overline{\rho}$ may not be independent on n_α because $\lambda(\theta)$ may depend on the problem size. For example, in time dependent problems the dependence of $\lambda(\theta)$ on the temporal step improves considerably the convergence rate of traditional relaxation methods [17].

In the local Fourier analysis it is easier to obtain $\overline{\rho}$, but we have to take into account that for practical values of n_α (especially in the 3-D case where n_α tends to be much smaller than in 1-D and 2-D cases due to computer memory limitations) ρ may be much smaller than $\overline{\rho}$ and so the behavior of the smoothing method might be better than the predicted based on $\overline{\rho}$.

The above definitions consider standard coarsening. With semicoarsening there is at least one direction in which the space step is not doubled. The Fourier modes in those directions are resolved on all grids and so they are not included in Θ_r.

The previous analysis has been performed considering periodic boundary conditions. Though the periodic analysis gives accurate predictions for our problem with Dirichlet boundary conditions for weak anisotropies, it has been observed that better agreement with the numerical results can be obtained by excluding from the analysis the Fourier modes with $\theta^\alpha = 0$ since the error at the boundary is always 0, see [4]. In fact, the numerical results obtained for strong anisotropies are accurately explained using this approximation.

Therefore, when studying the problem with Dirichlet boundary conditions we consider the wave numbers given by

$$\Theta^D = \{\theta : \theta = (\theta_1, \theta_2, \theta_3), \theta_\alpha = \frac{2\pi k_\alpha}{n_\alpha}, k_\alpha \neq 0, k_\alpha = -\frac{n_\alpha}{2}, ..., \frac{n_\alpha}{2}, \alpha = 1, ..., 3\}$$

and the smoothing factor is defined as

$$\rho_D = sup\{|\lambda(\theta)| : \theta \in \Theta_r^D\}$$

The relation between the three smoothing factors is the following

$$\rho_D \leq \rho \leq \overline{\rho}$$

The periodic problem appears to be a pessimistic limit of the Dirichlet problem and the Dirichlet problem tends to the periodic one for very large mesh sizes.

The eigenvalue problem $S\psi(\theta) = \lambda(\theta)\psi(\theta)$ has to be solved in order to determine the three smoothing factors. If the coefficients are constant, the grid size is uniform, and the boundary conditions are periodic, the eigenvalues are given, in stencil notation,by

$$\lambda(\theta) = \frac{\sum_j N(j)\psi_j(\theta)}{\sum_j M(j)\psi_j(\theta)}$$

The finest grid used in the numerical experiments of this section has 32 points in each direction and all levels in the grid hierarchy (4.1) are visited during a cycle. All the numerical experiments reported deal with the numerical solution of Eq. (3.3) on the unit cube $\Omega = (0,1) \times (0,1) \times (0,1)$ with a right-hand side of

$$f(x,y,z) = -(a+b+c)\sin(x+y+z)$$

Dirichlet boundary conditions are enforced on the boundary by evaluating

$$u(x,y,z) = sin(x+y+z)$$

The experimental convergence factor presented in the tables is the asymptotic average residual reduction factor of one 3-D FAS cycle

$$\rho_e = \frac{||r^n||_2}{||r^{n-1}||_2}, n \uparrow \tag{A.1}$$

or the average residual reduction factor when the reduction per cycle is so high that the rounding errors do not allow achieving an asymptotic behavior

$$\rho_e = \frac{||r^n||_2}{||r^0||_2} \tag{A.2}$$

In the latter case the simulation is performed until the initial residual is reduced by a factor 10^{-12}. Lower values for this tolerance factor cannot be reached on 64-bit machines due to roundoff errors.

The 2-D problems in each plane can be solved exactly using enough 2-D FAS cycles. In Section 5 results are presented of a study on the use of multigrid cycles to solve these 2-D problems. In practice, the exact solutions can be replaced by approximate ones, considerably reducing the execution time of the solver.

A.1. The plane Jacobi method with damping. The (x,y)-plane Jacobi method with damping applied to Eq. (3.3) corresponds to the following splitting, in stencil notation

$$[M]_{k-1} = \omega^{-1} \begin{bmatrix} 0 \end{bmatrix} \quad [M]_k = \omega^{-1} \begin{bmatrix} & -\epsilon_2 & \\ -\epsilon_1 & 2+2\epsilon_1+2\epsilon_2 & -\epsilon_2 \\ & -\epsilon_2 & \end{bmatrix} \quad [M]_{k+1} = \omega^{-1} \begin{bmatrix} 0 \end{bmatrix}$$

$$[N]_{k-1} = \begin{bmatrix} 1 \end{bmatrix} \quad [N]_k = \begin{bmatrix} & 0 & \\ 0 & 0 & 0 \\ & 0 & \end{bmatrix} \quad [N]_{k+1} = \begin{bmatrix} 1 \end{bmatrix}$$

The amplification factor is given by

$$\lambda(\theta) = 1 - \omega + \frac{\omega \cos(\theta_3)}{1 + \epsilon_1(1-\cos(\theta_1)) + \epsilon_2(1-\cos(\theta_2))}$$

Note that $\lambda(\theta)$ is real and symmetric, $\lambda(\theta) = \lambda(-\theta)$, and so we only have to consider $\theta_\alpha \geq 0$ to obtain the smoothing factor.

Periodic boundary conditions

It is easy to find that the smoothing factor is then given by

$$\begin{aligned} \overline{\rho} &= max\{|\lambda(\frac{\pi}{2},0,0)|, |\lambda(0,\frac{\pi}{2},0)|, |\lambda(0,0,\pi)|\} \\ &= max\{|1-\frac{\omega\epsilon_1}{1+\epsilon_1}|, |1-\frac{\omega\epsilon_2}{1+\epsilon_2}|, |1-2\omega|\} \end{aligned} \tag{A.3}$$

Note that ω must be lower than 1 to obtain a smoothing factor lower than 1.

Let us assume $n_1 = n_2 = n_3 = n$ and $\epsilon_1 \leq \epsilon_2$, so the optimum value of the damping parameter to minimize the smoothing factor is given by

$$\omega = \frac{2+2\epsilon_1}{2+3\epsilon_1}, \tag{A.4}$$

and with this optimum damping parameter we have

$$\overline{\rho} = \frac{2+\epsilon_1}{2+3\epsilon_1} \tag{A.5}$$

If $\epsilon_1 < 1$ the optimum smoothing factor tends to 1, for example $\overline{\rho} = 0.99$ for $\epsilon_1 = 10^{-2}$. Therefore, in such cases, (x,y)-plane relaxation is not a good smoother for our problem and we should use (x,z)-plane relaxation.

On the other hand, for $\epsilon_1 \geq 1$ the optimum value of ω depends slightly on ϵ_1, we have $\frac{4}{5} \geq \omega \geq \frac{2}{3}$ and $\frac{3}{5} \geq \overline{\rho} \geq \frac{1}{3}$ for $1 \leq \epsilon_1 \leq \infty$.

For ω lower than the optimum one the smoothing factor is given by

$$\overline{\rho} = 1 - \frac{\omega\epsilon_1}{1+\epsilon_1} \tag{A.6}$$

and for a ω greater than the optimum one

$$\overline{\rho} = |1-2\omega| \tag{A.7}$$

Dirichlet boundary conditions

To perform the study with Dirichlet boundary conditions we exclude from analysis the Fourier modes with $\theta_\alpha = 0$. Doing so, defining $\varphi = \frac{2\pi}{n}$ and considering n large enough to approximate $(1-\cos(\varphi))$ by $\frac{2\pi^2}{n^2}$ we find

$$\begin{aligned}
\rho_D &= max\{|\lambda(\frac{\pi}{2},\varphi,\varphi)|, |\lambda(\varphi,\frac{\pi}{2},\varphi)|, |\lambda(\varphi,\varphi,\pi)|\} \\
&= max\{\, |1-\omega+\frac{\omega\cos(\varphi)}{1+\epsilon_1+\epsilon_2(1-\cos(\varphi))}|, |1-\omega+\frac{\omega\cos(\varphi)}{1+\epsilon_1(1-\cos(\varphi))+\epsilon_2}|, \\
&\qquad |1-\omega-\frac{\omega}{1+\epsilon_1(1-\cos(\varphi))+\epsilon_2(1-\cos(\varphi))}|\} \\
&= max\{\, |1-\omega+\frac{\omega(1-\frac{2\pi^2}{n^2})}{1+\epsilon_1+\epsilon_2\frac{2\pi^2}{n^2}}|, |1-\omega+\frac{\omega(1-\frac{2\pi^2}{n^2})}{1+\epsilon_1\frac{2\pi^2}{n^2}+\epsilon_2}|, \\
&\qquad |1-\omega-\frac{\omega}{1+\epsilon_1\frac{2\pi^2}{n^2}+\epsilon_2\frac{2\pi^2}{n^2}}|\}
\end{aligned}$$

We obtain the following optimum damping parameter (always considering $\epsilon_1 \leq \epsilon_2$)

$$\omega = 2(\frac{\epsilon_1+\epsilon_2\frac{2\pi^2}{n^2}+\frac{2\pi^2}{n^2}}{1+\epsilon_1+\epsilon_2\frac{2\pi^2}{n^2}} + \frac{2+\epsilon_1\frac{2\pi^2}{n^2}+\epsilon_2\frac{2\pi^2}{n^2}}{1+\epsilon_1\frac{2\pi^2}{n^2}+\epsilon_2\frac{2\pi^2}{n^2}})^{-1} \tag{A.8}$$

and the corresponding smoothing factor

$$\rho_D = (\frac{\epsilon_1+\epsilon_2\frac{2\pi^2}{n^2}+\frac{2\pi^2}{n^2}}{1+\epsilon_1+\epsilon_2\frac{2\pi^2}{n^2}} - \frac{2+\epsilon_1\frac{2\pi^2}{n^2}+\epsilon_2\frac{2\pi^2}{n^2}}{1+\epsilon_1\frac{2\pi^2}{n^2}+\epsilon_2\frac{2\pi^2}{n^2}})(\frac{\epsilon_1+\epsilon_2\frac{2\pi^2}{n^2}+\frac{2\pi^2}{n^2}}{1+\epsilon_1+\epsilon_2\frac{2\pi^2}{n^2}} + \frac{2+\epsilon_1\frac{2\pi^2}{n^2}+\epsilon_2\frac{2\pi^2}{n^2}}{1+\epsilon_1\frac{2\pi^2}{n^2}+\epsilon_2\frac{2\pi^2}{n^2}})^{-1} \tag{A.9}$$

If $n \gg \epsilon_1, \epsilon_2$ the previous optimum damping and smoothing factors tend to Eq. (A.4) and Eq. (A.5) respectively. Otherwise, the optimum damping parameter (A.8) tends to 1 and the optimum amplification factor (A.9) tends to 0 as $O(\frac{1}{\epsilon_1})$ if $n \ll \epsilon_1$, or as $O(\frac{1}{\epsilon_2})$ if $n \ll \epsilon_2$. Therefore, the plane Jacobi method with $\omega = 1$ is a very good smoother for strong anisotropies with Dirichlet boundary conditions. Observe that this result totally contradicts the result obtained with periodic boundary conditions.

For damping parameters lower than the optimum one the smoothing factor is given by

$$\rho_D = 1 - w + \frac{1-\frac{2\pi^2}{n^2}}{1+\epsilon_1+\epsilon_2\frac{2\pi^2}{n^2}}\omega$$

and for a damping greater than the optimum one the smoothing factor is given by

$$\rho_D = |1 - w + \frac{1}{1+\epsilon_1\frac{2\pi^2}{n^2}+\epsilon_2\frac{2\pi^2}{n^2}}\omega|$$

Therefore,

- If $n \gg \epsilon_1$ and $n \gg \epsilon_2$ the smoothing factor tends to the periodic case. See for example in Table A.1 how the results for $\epsilon_1, \epsilon_2 \leq 1$ fully agree with Eq. (A.4) for the optimum damping parameter and with Eq. (A.6) and (A.7) for the smoothing factor.
- If $n \ll \epsilon_1$ and $n \ll \epsilon_2$ the smoothing factor tends to

  $$\rho_D = 1 - \omega + O(\frac{1}{\epsilon_1+\epsilon_2})\omega$$

  as ϵ_1 and ϵ_2 become stronger. Table A.1(a), for $\epsilon_1, \epsilon_2 \geq 10^2$, presents some results that verify the previous expression. Note that the smoothing factor falls linearly with the strength of the anisotropies when the damping parameter is equal to one.

ϵ_1	ϵ_2	ω				
		0.2	0.4	0.6	0.8	1.0
10^{-8}	10^{-8}	0.99	0.99	0.99	0.99	0.98
10^{-2}	10^{-2}	0.98	0.98	0.98	0.98	0.98
1	1	0.90	0.80	0.70	0.59	0.98
10^{2}	10^{2}	0.80	0.62	0.42	0.23	0.46
10^{4}	10^{4}	0.80	0.60	0.40	0.20	3.6×10^{-4}
10^{6}	10^{6}	0.80	0.60	0.40	0.20	3.7×10^{-6}
10^{8}	10^{8}	0.80	0.60	0.40	0.20	5.7×10^{-8}

(a) $\epsilon_1 = \epsilon_2$

ϵ_1	ϵ_2	ω				
		0.2	0.4	0.6	0.8	1.0
1	1	0.90	0.80	0.70	0.59	0.98
1	10^{2}	0.81	0.74	0.62	0.50	0.65
1	10^{4}	0.80	0.60	0.40	0.20	9.6×10^{-4}
1	10^{6}	0.80	0.60	0.40	0.20	1.0×10^{-5}
1	10^{8}	0.80	0.60	0.40	0.20	1.0×10^{-7}

(b) $\epsilon_1 = 1$ and various ϵ_2

ϵ_1	ϵ_2	ω				
		0.2	0.4	0.6	0.8	1.0
10^{-4}	10^{-2}	0.99	0.99	0.99	0.99	0.98
10^{-4}	1	0.99	0.99	0.99	0.99	0.98
10^{-4}	10^{2}	0.93	0.87	0.80	0.73	0.66
10^{-4}	10^{4}	0.80	0.60	0.40	0.20	9.8×10^{-4}
10^{-4}	10^{6}	0.80	0.60	0.40	0.20	1.0×10^{-5}
10^{-4}	10^{8}	0.80	0.60	0.40	0.20	1.0×10^{-7}

(c) $\epsilon_1 = 10^{-4}$ and various ϵ_2

TABLE A.1

Computational convergence factors, ρ_e, of one 3-D V(1,0)-cycle with (x,y)-plane Jacobi with damping parameter ω

- If $n \gg \epsilon_1$ and $n \ll \epsilon_2$ the smoothing factor tends to

$$\rho_D = 1 - \omega + O(\frac{1}{\epsilon_2})\omega$$

as ϵ_2 becomes stronger. Table A.1 (b) and (c) give some numerical results that agree with the previous expression. Observe how for low values of ϵ_1 (even though $\epsilon_1 \ll 1$) the method is a very good smoother if ϵ_2 is large enough.

The previous discussion concludes that the periodic case can be considered as an asymptotic limit of the Dirichlet case when n tends to ∞ ($n \gg \epsilon$). However, there is a huge difference between the cases for practical grid sizes. The Dirichlet case presents very good convergence rates with anisotropy values for which

the periodic case does not converge. The multigrid algorithm reaches the solution accurate to the truncation error in just a few cycles for a strong value of just one of the anisotropies.

The behavior of the smoother with Dirichlet boundary conditions is attributed to the fact that as the anisotropy grows, the method becomes an exact solver and so the error is reduced by a factor $1-\omega$. The optimum damping parameter depends strongly on ϵ in this case. One way to avoid this and get good convergence rates for all ω is to apply the damping parameter just to the diagonal component of the method in the explicit direction which is the plane Jacobi method with partial damping. In this way the method becomes an exact solver for strong anisotropies and all damping parameters.

A.2. The plane Jacobi method with partial damping. The (x,y)-plane Jacobi method with partial damping applied to Eq. (3.3) corresponds to the following splitting, in stencil notation

$$[M]_{k-1} = \begin{bmatrix} 0 \end{bmatrix} \quad [M]_k = \begin{bmatrix} & -\epsilon_2 & \\ -\epsilon_1 & 2\omega^{-1} + 2\epsilon_1 + 2\epsilon_2 & -\epsilon_2 \\ & -\epsilon_2 & \end{bmatrix} \quad [M]_{k+1} = \begin{bmatrix} 0 \end{bmatrix}$$

$$[N]_{k-1} = \begin{bmatrix} 1 \end{bmatrix} \quad [N]_k = \begin{bmatrix} & 0 & \\ 0 & 0 & 0 \\ & 0 & \end{bmatrix} \quad [N]_{k+1} = \begin{bmatrix} 1 \end{bmatrix}$$

The amplification factor is given by

$$\lambda(\theta) = \frac{1-\omega(1-\cos(\theta_3))}{1+\omega\epsilon_1(1-\cos(\theta_1))+\omega\epsilon_2(1-\cos(\theta_2))}$$

We only have to consider $\theta_\alpha \geq 0$ because $\lambda(\theta)$ is real and symmetric, $\lambda(\theta) = \lambda(-\theta)$.

Periodic boundary conditions

The amplification factor is given by

$$\begin{aligned} \overline{\rho} &= max\{|\lambda(\frac{\pi}{2},0,0)|, |\lambda(0,\frac{\pi}{2},0)|, |\lambda(0,0,\frac{\pi}{2})|, |\lambda(0,0,\pi)|\} \\ &= max\{|\frac{1}{1+\epsilon_1\omega}|, |\frac{1}{1+\epsilon_2\omega}|, |1-\omega|, |1-2\omega|\} \end{aligned} \tag{A.10}$$

Note that ω must be lower than 1 to obtain an amplification factor lower than 1.

Let us assume $n_1 = n_2 = n_3 = n$ and $\epsilon_1 \leq \epsilon_2$, so the smoothing factor is given by

$$\overline{\rho} = max\{1-\omega, \frac{1}{1+\epsilon_1\omega}, |1-2\omega|\} \tag{A.11}$$

If $\epsilon_1 < 1$, the optimum smoothing factor tends to 1, for example $\overline{\rho} = 0.99$ for $\epsilon_1 = 10^{-2}$. Therefore, in such a case, (x,y)-plane relaxation is not a good smoother and (x,z)-plane relaxation should be used.

If $1 \leq \epsilon_1 \leq 3$, the optimum damping parameter can be obtained by equating the second and the third functions in Eq. (A.11). For example, for $\epsilon_1 = 1$, the optimum damping parameter is equal to 0.78 and the optimum smoothing factor is 0.56. For ω lower than 0.78, the smoothing factor is given by

$$\overline{\rho} = \frac{1}{1+\omega} \tag{A.12}$$

and for a damping parameter greater than 0.78

$$\overline{\rho} = |1-2\omega| \tag{A.13}$$

On the other hand, if $\epsilon_1 \geq 3$ the smoothing factor does not depend on the anisotropy and it is given by

$$\overline{\rho} = max\{1-\omega, |1-2\omega|\}$$

and so the optimum smoothing parameter is $\frac{1}{3}$, corresponding to a damping of $\frac{2}{3}$. For ω lower than the optimum one, the smoothing factor is given by

$$\overline{\rho} = 1-\omega \tag{A.14}$$

and for a damping greater than the optimum

$$\overline{\rho} = |1-2\omega| \tag{A.15}$$

Dirichlet boundary conditions

To perform the study with Dirichlet boundary conditions we exclude from analysis the Fourier modes with $\theta_\alpha = 0$. Doing so, defining $\varphi = \frac{2\pi}{n}$ and considering that n is large enough we obtain

$$\begin{aligned}
\rho_D &= max\{|\lambda(\frac{\pi}{2},\varphi,\varphi)|, |\lambda(\varphi,\frac{\pi}{2},\varphi)|, |\lambda(\varphi,\varphi,\pi)|, |\lambda(\varphi,\varphi,\pi)|\} \\
&= max\{|\frac{1-\omega(1-\cos(\varphi))}{1+\omega\epsilon_1+\omega\epsilon_2(1-\cos(\varphi))}|, |\frac{1-\omega(1-\cos(\varphi))}{1+\omega\epsilon_1(1-\cos(\varphi))+\omega\epsilon_2}|, \\
&\qquad |\frac{1-\omega}{1+\omega\epsilon_1(1-\cos(\varphi))+\omega\epsilon_2(1-\cos(\varphi))}|, |\frac{1-2\omega}{1+\omega\epsilon_1(1-\cos(\varphi))+\omega\epsilon_2(1-\cos(\varphi))}|\} \\
&= max\{|\frac{1-\omega\frac{2\pi^2}{n^2}}{1+\omega\epsilon_1+\omega\epsilon_2\frac{2\pi^2}{n^2}}|, |\frac{1-\omega\frac{2\pi^2}{n^2}}{1+\omega\epsilon_1\frac{2\pi^2}{n^2}+\omega\epsilon_2}|, \\
&\qquad |\frac{1-\omega}{1+\omega\epsilon_1\frac{2\pi^2}{n^2}+\omega\epsilon_2\frac{2\pi^2}{n^2}}|, |\frac{1-2\omega}{1+\omega\epsilon_1\frac{2\pi^2}{n^2}+\omega\epsilon_2\frac{2\pi^2}{n^2}}|\}
\end{aligned}$$

These expressions coincide with (A.10) if $n \gg \epsilon_1, \epsilon_2$.

Otherwise, if $n \ll \epsilon_1, \epsilon_2$, and always considering $\epsilon_1 \leq \epsilon_2$, the optimum damping factor remains $\frac{2}{3}$ with a corresponding optimum smoothing factor

$$\rho_D = \frac{\frac{1}{3}}{1+\frac{2}{3}\frac{2\pi^2}{n^2}(\epsilon_1+\epsilon_2)}$$

Note that the previous optimum amplification factor tends to 0 as $O(\frac{1}{\epsilon_1+\epsilon_2})$. For damping parameters lower than the optimum one the smoothing parameter is given by

$$\rho_D = \frac{1-\omega}{1+\omega\epsilon_1\frac{2\pi^2}{n^2}+\omega\epsilon_2\frac{2\pi^2}{n^2}}$$

and for a damping greater than the optimum one the smoothing factor is given by

$$\rho_D = |\frac{1-2\omega}{1+\omega\epsilon_1\frac{2\pi^2}{n^2}+\omega\epsilon_2\frac{2\pi^2}{n^2}}|$$

On the other hand, if ϵ_1 is small enough the optimum smoothing parameter tends to 1 for increasing values of ϵ_2. For damping parameters lower than the optimum one the smoothing parameter is given by

$$\rho_D = \frac{1-\omega\frac{2\pi^2}{n^2}}{1+\omega\epsilon_1+\omega\epsilon_2\frac{2\pi^2}{n^2}}$$

and for a damping greater than the optimum one the smoothing factor is given by

$$\rho_D = |\frac{1-2\omega}{1+\omega\epsilon_1\frac{2\pi^2}{n^2}+\omega\epsilon_2\frac{2\pi^2}{n^2}}|$$

Therefore,

ϵ_1	ϵ_2	ω					
		0.2	0.4	0.6	0.8	0.9	1.0
10^{-8}	10^{-8}	0.99	0.99	0.99	0.99	0.99	0.98
10^{-2}	10^{-2}	0.99	0.99	0.99	0.99	0.99	0.98
1	1	0.81	0.70	0.65	0.55	0.77	0.98
3	3	0.80	0.59	0.41	0.57	0.76	0.95
10	10	0.76	0.56	0.35	0.53	0.71	0.88
10^2	10^2	0.65	0.42	0.23	0.31	0.39	0.46
10^4	10^4	2.9×10^{-3}	1.2×10^{-3}	6.2×10^{-4}	4.1×10^{-4}	3.7×10^{-4}	3.6×10^{-4}
10^6	10^6	3.1×10^{-5}	1.2×10^{-5}	6.3×10^{-6}	4.2×10^{-6}	3.0×10^{-6}	3.7×10^{-6}
10^8	10^8	3.1×10^{-7}	1.2×10^{-7}	6.4×10^{-8}	4.4×10^{-8}	4.0×10^{-8}	5.7×10^{-8}

(a) $\epsilon_1 = \epsilon_2$

ϵ_1	ϵ_2	ω					
		0.2	0.4	0.6	0.8	0.9	1.0
1	1	0.81	0.70	0.65	0.55	0.77	0.98
1	10	0.77	0.66	0.58	0.55	0.72	0.92
1	10^2	0.73	0.58	0.48	0.40	0.53	0.65
1	10^4	5.8×10^{-3}	2.6×10^{-3}	1.6×10^{-3}	1.2×10^{-3}	1.0×10^{-3}	9.6×10^{-4}
1	10^6	6.7×10^{-5}	2.8×10^{-5}	1.7×10^{-5}	1.2×10^{-5}	1.1×10^{-5}	1.0×10^{-5}
1	10^8	6.7×10^{-7}	2.8×10^{-7}	1.7×10^{-7}	1.2×10^{-7}	1.1×10^{-7}	1.0×10^{-7}

(b) $\epsilon_1 = 1$ and various ϵ_2

ϵ_1	ϵ_2	ω					
		0.2	0.4	0.6	0.8	0.9	1.0
10^{-4}	10^{-2}	0.99	0.99	0.99	0.99	0.99	0.98
10^{-4}	1	0.99	0.99	0.99	0.99	0.99	0.98
10^{-4}	10	0.99	0.98	0.97	0.96	0.95	0.98
10^{-4}	10^2	0.91	0.84	0.77	0.72	0.69	0.66
10^{-4}	10^4	5.9×10^{-3}	2.6×10^{-3}	1.6×10^{-3}	1.2×10^{-3}	1.0×10^{-3}	9.6×10^{-4}
10^{-4}	10^6	6.7×10^{-5}	2.8×10^{-5}	1.7×10^{-5}	1.2×10^{-5}	1.1×10^{-5}	1.0×10^{-5}
10^{-4}	10^8	6.7×10^{-7}	2.8×10^{-7}	1.7×10^{-7}	1.2×10^{-7}	1.1×10^{-7}	1.0×10^{-7}

(c) $\epsilon_1 = 10^{-4}$ and various ϵ_2

TABLE A.2

Computational convergence factors, ρ_e, of one 3-D V(1,0)-cycle (x,y)-plane Jacobi with partial damping parameter ω

- If $n \gg \epsilon_1$ and $n \gg \epsilon_2$ the smoothing factor tends to the periodic case. See for example how the analytical expressions obtained for the periodic case (Eqs. A.12 and A.13 for $\epsilon_1 = 1$ and A.14 and A.15 for $\epsilon_1 = 3$) accurately agree with the numerical results presented in Table A.2(a).

- If $n \ll \epsilon_1$ and $n \ll \epsilon_2$ the smoothing factor for a damping lower than the optimum one, $\frac{2}{3}$, tends to

$$\rho_D = \frac{1-\omega}{\omega O(\epsilon_1+\epsilon_2)}$$

and for a damping parameter greater than the optimum one tends to

$$\rho_D = |\frac{1-2\omega}{\omega O(\epsilon_1+\epsilon_2)}|$$

as ϵ_1 and ϵ_2 become stronger. Table A.2(a), for $\epsilon_1, \epsilon_2 \geq 10^2$, contains some results that verify that the smoothing factor falls linearly with the anisotropy and is a function of ω.

- If $n \gg \epsilon_1$ and $n \ll \epsilon_2$ the smoothing factor for a damping parameter lower than the optimum one tends to

$$\rho_D = \frac{1}{\omega O(\epsilon_2)}$$

and for a damping greater than the optimum one tends to

$$\rho_D = |\frac{1-2\omega}{\omega O(\epsilon_2)}|$$

as ϵ_2 becomes stronger. The optimum damping parameter moves from 0.78 to 1. Tables A.2 (b) and (c) give some numerical results that verify the previous expressions. This method is a very good smoother for very low values of ϵ_1 when ϵ_2 is large enough.

A.3. The plane Gauss-Seidel method. The (x,y)-plane Gauss-Seidel method with partial damping applied to Eq. (3.3) corresponds to the following splitting, in stencil notation

$$[M]_{k-1} = \begin{bmatrix} & -1 & \\ & & \end{bmatrix} [M]_k = \begin{bmatrix} & -\epsilon_2 & \\ -\epsilon_1 & 2+2\epsilon_1+2\epsilon_2 & -\epsilon_2 \\ & -\epsilon_2 & \end{bmatrix} [M]_{k+1} = \begin{bmatrix} & 0 & \\ & & \end{bmatrix}$$

$$[N]_{k-1} = \begin{bmatrix} & 0 & \\ & & \end{bmatrix} [N]_k = \begin{bmatrix} & 0 & \\ 0 & 0 & 0 \\ & 0 & \end{bmatrix} [N]_{k+1} = \begin{bmatrix} & 1 & \\ & & \end{bmatrix}$$

The amplification factor is given by

$$\lambda(\theta) = \frac{e^{i\theta_3}}{2 - e^{-i\theta_3} + 2\epsilon_1(1-\cos(\theta_1)) + 2\epsilon_2(1-\cos(\theta_2))}$$

Note that $\lambda(\theta)$ is complex but symmetric, $|\lambda(\theta)| = |\lambda(-\theta)|$, and so we only have to consider $\theta_\alpha \geq 0$.

Periodic boundary conditions

It is easier to find the smoothing factor if we rewrite the amplification factor as

$$|\lambda(\theta)| = \frac{1}{\sqrt{(2-\cos(\theta_3)+2\epsilon_1(1-\cos(\theta_1))+2\epsilon_2(1-\cos(\theta_2)))^2+\sin^2(\theta_3)}}$$

From the previous expression we obtain

$$\begin{aligned} \overline{\rho} &= max\{|\lambda(\frac{\pi}{2},0,0)|, |\lambda(0,\frac{\pi}{2},0)|, |\lambda(0,0,\frac{\pi}{2})|\} \\ &= max\{|\frac{1}{1+2\epsilon_1}|, |\frac{1}{1+2\epsilon_2}|, |\frac{1}{\sqrt{5}}|\} \end{aligned} \tag{A.16}$$

Let us assume $n_1 = n_2 = n_3 = n$ and $\epsilon_1 \leq \epsilon_2$, the smoothing factor is then given by

$$\overline{\rho} = max\{\frac{1}{1+2\epsilon_1}, \frac{1}{\sqrt{5}}\}$$

Therefore, the smoothing factor is $\frac{1}{\sqrt{5}}$ for $\epsilon_1 \geq \frac{\sqrt{5}-1}{2} \approx 0.6$ and $\frac{1}{1+2\epsilon_1}$ for $\epsilon_1 < 0.6$. For $\epsilon_1 < 0.6$ the smoothing factor tends to 1 and (x,y)-plane relaxation is not a good smoother; we should use (x,z)-plane relaxation.

ϵ_1	ϵ_2	
10^{-8}	10^{-8}	0.99
10^{-2}	10^{-2}	0.96
0.66	0.66	0.43
1	1	0.34
10^{2}	10^{2}	0.20
10^{4}	10^{4}	4.6×10^{-4}
10^{6}	10^{6}	2.8×10^{-6}
10^{8}	10^{8}	3.3×10^{-8}

(a) $\epsilon_1 = \epsilon_2$

ϵ_1	ϵ_2	
1	1	0.34
1	10^{2}	0.25
1	10^{4}	6.1×10^{-4}
1	10^{6}	6.1×10^{-6}
1	10^{8}	6.2×10^{-8}

(b) $\epsilon_1 = 1$ and various ϵ_2

ϵ_1	ϵ_2	
10^{-4}	10^{-2}	0.99
10^{-4}	1	0.97
10^{-4}	10^{2}	0.50
10^{-4}	10^{4}	6.1×10^{-4}
10^{-4}	10^{6}	6.1×10^{-6}
10^{-4}	10^{8}	6.2×10^{-8}

(c) $\epsilon_1 = 10^{-4}$ and various ϵ_2

TABLE A.3

Computational convergence factors, ρ_e, of one 3-D V(1,0)-cycle with (x,y)-plane Gauss-Seidel

Dirichlet boundary conditions

Excluding from analysis the Fourier modes with $\theta_\alpha = 0$, defining $\varphi = \frac{2\pi}{n}$ and considering that n is large enough we have

$$\rho_D = max\{|\lambda(\frac{\pi}{2}, \varphi, \varphi)|, |\lambda(\varphi, \frac{\pi}{2}, \varphi)|, |\lambda(\varphi, \varphi, \frac{\pi}{2})|\}$$

$$= max\{|\frac{1}{\sqrt{(2-\cos(\varphi)+2\epsilon_1+2\epsilon_2(1-\cos(\varphi)))^2+\sin^2(\varphi)}}|, |\frac{1}{\sqrt{(2-\cos(\varphi)+2\epsilon_1(1-\cos(\varphi))+2\epsilon_2)^2+\sin^2(\varphi)}}|,$$
$$|\frac{1}{\sqrt{(2+2\epsilon_1(1-\cos(\varphi))+2\epsilon_2(1-\cos(\varphi)))^2+1}}|\}$$
$$= max\{|\frac{1}{\sqrt{(1+\frac{2\pi^2}{n^2}+2\epsilon_1+2\epsilon_2\frac{2\pi^2}{n^2})^2+\frac{2\pi^2}{n^2}}}|, |\frac{1}{\sqrt{(1+\frac{2\pi^2}{n^2}+2\epsilon_1\frac{2\pi^2}{n^2}+2\epsilon_2)^2+\frac{4\pi^2}{n^2}}}|,$$
$$|\frac{1}{\sqrt{(2+2\epsilon_1\frac{2\pi^2}{n^2}+2\epsilon_2\frac{2\pi^2}{n^2})^2+1}}|\}$$

These expressions coincide with (A.16) if $n \gg \epsilon_1, \epsilon_2$. For example Table A.3(a) shows that the experimental convergence rate when $\epsilon = 0.66$ is 0.43, which verifies the analytical prediction $\frac{1}{\sqrt{5}}$ of the periodic case.

Otherwise, the smoothing factor is

$$\rho_D = |\frac{1}{\sqrt{(2+2\epsilon_1\frac{2\pi^2}{n^2}+2\epsilon_2\frac{2\pi^2}{n^2})^2+1}}|$$

when $n \ll \epsilon_1, \epsilon_2$. Note that in this case the smoothing factor tends to 0 as $O(\frac{1}{\epsilon_1+\epsilon_2})$ for strong anisotropies. This dependence of the smoothing factor on the anisotropy is verified numerically by the results presented in Table A.3.

However, for small values of ϵ_1 the smoothing factor tends to

$$\rho_D = \frac{1}{\sqrt{(1+\frac{2\pi^2}{n^2}+2\epsilon_1+2\epsilon_2\frac{2\pi^2}{n^2})^2+\frac{2\pi^2}{n^2}}}.$$

If ϵ_1 is small enough, ρ_D can be approximated by

$$\rho_D \approx \frac{1}{1+2\epsilon_2\frac{2\pi^2}{n^2}} \tag{A.17}$$

and the smoothing factor decreases as $O(\frac{1}{\epsilon_2})$ for increasing ϵ_2 values. This behavior is also exhibited by the numerical experiments presented in Table A.3 (b) and (c). Again we find that very good convergence rates can be achieved even though one anisotropy is lower than one.

The numerical results show that it does not pay to use SOR ($\omega > 1$) or damped Gauss-Seidel ($\omega < 1$) as a smoother.

A.4. The plane zebra Gauss-Seidel method. The analytical study for this case is more involved because the Fourier modes are not invariant under this method. That is, the zebra ordering does not preserve the modes. However, the study can be performed considering that the operation of an iteration on a $\psi(\theta)$ mode results in a combination of the mode and its harmonics [31]. In this section, we will restrict ourselves to the presentation and discussion of the numerical results.

In reference [32], Yavneh presents results of his study of the zebra Gauss-Seidel method in all combinations of block and point relaxation with full and partial coarsening for periodic boundary conditions. He indicates that the smoothing factor with r relaxation sweeps for the present case is given by

$$\overline{\rho} = max\{(\frac{1}{1+\epsilon_1})^2, (\frac{2r-1}{2r})^2[\frac{1}{2(2r-1)}]^{\frac{1}{r}}\} \tag{A.18}$$

	cycling strategies of the 3-D cycle						
	(1,0)	(0,1)	(2,0)	(0,2)	(1,1)	(2,1)	(1,2)
GS	0.34	0.34	0.14	0.14	0.13	0.08	0.08
ZGS	0.36	0.42	0.20	0.13	0.24	0.17	0.17
4cGS	0.28	0.35	0.14	0.14	0.12	0.10	0.10

TABLE A.4

Computational convergence factors, ρ_e, of 3-D cycles for different V-cycles strategies with (x,y)-plane Gauss-Seidel (GS), (x,y)-plane zebra Gauss-Seidel (ZGS) and (x,y)-plane four-color Gauss-Seidel (4cGS) as smoothers

The numerical results presented in Table A.4 diverge considerably from those that we expected because the lexicographic ordering performs better than the zebra ordering and the result for the isotropic case is larger than the 0.25 predicted by Eq. (A.18). The translation of the grid points by half space step, that occurs in cell-centered grids in relation to vertex-centered grids, does not affect the Fourier analysis but it could affect the behavior of the zebra method in the coarse grid correction, as is shown in [8] for the 2-D case. Based on the good results with a four-color ordering reported in [8] for the 2-D case, we applied this ordering and obtained more robustness. (See Table A.4.)

Tables A.5(a), (b), and (c) show the behavior of the four-color and zebra Gauss-Seidel methods as smoothers. The methods have good behavior for strong values of the anisotropy where the convergence rate decreases (improves) linearly with the strength of the anisotropy as $O(\frac{1}{\epsilon_1+\epsilon_2})$. The four-color ordering presents similar convergence rates to the lexicographic ordering and parallelizes easily, so that it is an attractive smoother.

A.5. The line Gauss-Seidel method. We also include line Gauss-Seidel in this analysis because of its good behavior observed for strong anisotropies in a single direction. Its performance improves considerably with Dirichlet boundary conditions when one of the anisotropies is stronger than the other; i.e, when one direction dominates. The application of Fourier analysis to study the smoothing properties of y-line Gauss-Seidel as applied to Eq. (3.3) is very cumbersome, here we present numerical results and some explicit formula for the smoothing factor obtained by studying the behavior of the numerical results. The formulas help us to explain the behavior of the method in a qualitative way, however they do not represent its accurate behavior.

If $\epsilon_1 > 1$ the smoothing factor can be approximated by

$$\rho_D \approx \frac{\epsilon_1}{1+\epsilon_1+\epsilon_2\frac{2\pi^2}{n^2}}$$

that is equal to 0.5 for $\epsilon_1 = 1$ and tends to 1 for increasing values of ϵ_1, this behavior is observed in the results of Table A.6(a). If $\epsilon_2 \gg \epsilon_1 > 1$ the previous expression can be approximated by

$$\rho_D \approx \frac{1}{1+\frac{\epsilon_2}{\epsilon_1}\frac{2\pi^2}{n^2}} \tag{A.19}$$

and the method presents good convergence rates. (See Table A.6(b).) Eq. (A.19) shows that the smoothing factor depends on the $\frac{\epsilon_2}{\epsilon_1}$ ratio. The results presented in Table A.6(c) also show this dependence on the quotient between anisotropies.

On the other hand, if $\epsilon_1 < 1$ the smoothing factor can be approximated by

$$\rho_D \approx \frac{1}{\epsilon_1+\epsilon_2\frac{2\pi^2}{n^2}+1}$$

ϵ_1	ϵ_2	ZGS	4cGS
10^{-8}	10^{-2}	0.99	0.99
10^{-2}	10^{-2}	0.96	0.96
1	1	0.36	0.28
10^{2}	10^{2}	0.11	0.12
10^{4}	10^{4}	1.2×10^{-3}	7.4×10^{-4}
10^{6}	10^{6}	1.2×10^{-5}	7.4×10^{-6}
10^{8}	10^{8}	1.2×10^{-7}	7.4×10^{-8}

(a) $\epsilon_1 = \epsilon_2$

ϵ_1	ϵ_2	ZGS	4cGS
1	1	0.36	0.28
1	10^{2}	0.12	0.16
1	10^{4}	2.4×10^{-3}	1.5×10^{-3}
1	10^{6}	2.4×10^{-5}	1.5×10^{-5}
1	10^{8}	2.4×10^{-7}	1.5×10^{-7}

(b) $\epsilon_1 = 1$ and various ϵ_2

ϵ_1	ϵ_2	ZGS	4cGS
10^{-4}	10^{-2}	0.99	0.99
10^{-4}	1	0.97	0.97
10^{-4}	10^{2}	0.45	0.46
10^{-4}	10^{4}	2.4×10^{-3}	1.5×10^{-3}
10^{-4}	10^{6}	2.4×10^{-5}	1.5×10^{-5}
10^{-4}	10^{8}	2.4×10^{-7}	1.5×10^{-7}

(c) $\epsilon_1 = 10^{-4}$ and various ϵ_2

TABLE A.5

Computational convergence factors, ρ_e, of one 3-D V(1,0)-cycle with (x,y)-plane Gauss-Seidel (ZGS) and (x,y)-plane four-color Gauss-Seidel (4cGS)

that tends to 1 for decreasing values of ϵ_1. (See Table A.6(a).) However if $\epsilon_2 \gg n$ the smoothing factor can be approximated by

$$\rho_D \approx \frac{1}{\epsilon_2 \frac{2\pi^2}{n^2} + 1}$$

In this case the smoothing factor decreases linearly with ϵ_2 and not with $\frac{\epsilon_2}{\epsilon_1}$. (See Table A.6(d).)

The expressions for the periodic case are obtained by letting n go to ∞ in the previous expressions. As expected, the convergence rate improves linearly with ϵ_2 when $\epsilon_1 = 1$. However, another very important result is that when $\epsilon_1 > 1$ the convergence rate grows with $\frac{\epsilon_2}{\epsilon_1}$ and when $\epsilon_1 < 1$ the convergence rate grows with ϵ_2. Consequently, when one of the anisotropies is stronger than the other, then only one term dominates and the line smoother gives good convergence rates for practical mesh sizes. That is the explanation for the good behavior exhibited by the alternating-line smoother on highly stretched grids along all directions.

A.6. Conclusions about the smoothing factors. We have studied the performance of (x,y)-plane relaxation methods as smoothers when $\epsilon_1 \leq \epsilon_2$. All of them do not reduce the high components of the error when both anisotropies, ϵ_1 and ϵ_2, are lower than 1 or when ϵ_1 is lower than 1 and ϵ_2 is not large enough. In such cases, (y,z)-plane relaxation would perform as a better smoother. On the other hand, if $\epsilon_1 > \epsilon_2$, (x,z)-plane relaxation smoother should be used when ϵ_1 and ϵ_2 are lower than 1 or when ϵ_2 is lower than 1 and ϵ_1 is not large enough. Robustness can be achieved by applying the plane relaxation smoothers alternatingly.

The results for periodic boundary conditions presented by Yavneh in [32] show that red-black block relaxation without coarsening the coordinates relaxed in a block yields the same efficiency as a point-wise smoother with full coarsening for the lower-dimensional problem defined by the coordinates that are relaxed in odd-even ordering. Therefore, plane relaxation without coarsening of the coordinates with relatively smaller or larger coefficients provides a very good smoother.

The question that arises here is whether it is possible to extend this result to include the case of block relaxation with full coarsening for the methods under study. In this case the coordinates relaxed in the block must be those with relatively larger coefficients, as is stated by the fundamental block relaxation rule.

Comparing the analytical results obtained in this section for plane relaxation in the 3-D case with the results presented by Wesseling in [31] for line relaxation in the 2-D case we conclude that,

- first, and in general, the smoothing factor obtained by a line relaxation method in the 2-D case coincides with the smoothing factor obtained for plane relaxation in 3-D. Indeed, the analytical expressions presented in this section are similar to the formula obtained in [31] changing ϵ to ϵ_1 (we supposed that $\epsilon_1 \leq \epsilon_2$ in the analytical development)
- second, the value of the smoothing factor for a plane implicit method (which depends on the strength of the anisotropy) is somewhere between the value of the smoothing factor for a point-wise scheme for the 2-D isotropic problem and the value of the smoothing factor of the point-wise method for the isotropic 1-D problem.

Table A.7 contains analytic results obtained in this section and reported by Wesseling [31] and by Yavneh [32] that verify these conclusions.

On the other hand, the value of the smoothing factor obtained by a line relaxation method in the 3-D case is somewhere between the smoothing factor obtained with the point-wise version for the isotropic 3-D problem and the smoothing factor obtained with the point-wise version for the lower-dimensional problem defined by the coordinates that are not relaxed in the block; i.e., pointwise for the remaining 2-D problem.

This observation is verified by the results presented in [32] for the 3-D y-line zebra Gauss-Seidel, $\overline{\rho} = 0.32$ for $\epsilon_1 = 1$ and $\overline{\rho} = (\frac{\epsilon_2}{1+\epsilon_2})^2$ for $\epsilon_1 \geq 2$, and the experimental convergence factor of $\frac{1}{2}$ obtained for the line 3-D line Gauss-Seidel (Table A.6(a)).

Consequently, as we expected, the smoothing factor of a block smoother with full coarsening approaches

- for periodic boundary conditions, the smoothing factor obtained with the point-wise version with full coarsening for the lower-dimensional problem defined by the coordinates that are not relaxed in the block, and
- for Dirichlet boundary conditions, an exact solver

as the anisotropy strength grows.

ϵ_1	ϵ_2	
10^{-8}	10^{-8}	0.99
10^{-2}	10^{-2}	0.96
1	1	0.48
10^{2}	10^{2}	0.95
10^{4}	10^{4}	0.98
10^{6}	10^{6}	0.99
10^{8}	10^{8}	0.99

(a) $\epsilon_1 = \epsilon_2$

ϵ_1	ϵ_2	
1	1	0.48
1	10^{2}	0.31
1	10^{4}	8.1×10^{-4}
1	10^{6}	8.3×10^{-6}
1	10^{8}	8.4×10^{-8}

(b) $\epsilon_1 = 1$ and various ϵ_2

ϵ_1	ϵ_2		ϵ_1	ϵ_2	
10^{2}	1	0.96	10^{4}	10^{2}	0.99
10^{2}	10^{2}	0.95	10^{4}	10^{4}	0.98
10^{2}	10^{4}	0.49	10^{4}	10^{6}	0.50
10^{2}	10^{6}	6.1×10^{-4}	10^{4}	10^{8}	6.1×10^{-4}
10^{2}	10^{8}	6.2×10^{-8}	10^{4}	10^{10}	6.2×10^{-8}

(c) $\epsilon_1 = 10^2$ and 10^4 and various ϵ_2

ϵ_1	ϵ_2		ϵ_1	ϵ_2	
10^{-4}	10^{-2}	0.99	10^{-2}	10^{-2}	0.96
10^{-4}	1	0.97	10^{-2}	1	0.95
10^{-4}	10^{2}	0.50	10^{-2}	10^{2}	0.50
10^{-4}	10^{4}	6.1×10^{-4}	10^{-2}	10^{4}	6.1×10^{-4}
10^{-4}	10^{6}	6.1×10^{-6}	10^{-2}	10^{6}	6.1×10^{-6}
10^{-4}	10^{8}	6.2×10^{-8}	10^{-2}	10^{8}	6.2×10^{-8}

(d) $\epsilon_1 = 10^{-4}$ and 10^{-2} and various ϵ_2

TABLE A.6

Computational convergence factors, ρ_e, of one 3-D V(1,0)-cycle with y-line Gauss-Seidel

	Jacobi with damping		Gauss-Seidel		Zebra Gauss-Seidel	
	$\epsilon_1 = 1$	$\epsilon_1 = \infty$	$\epsilon_1 = 1$	$\epsilon_1 \geq 1$	$\epsilon_1 = 1$	$\epsilon_1 \geq 0.18$
3-D plane relaxation	$\omega_{opt} = \frac{4}{5}\rho_{opt} = \frac{3}{5}$	$\omega_{opt} = \frac{2}{3}\rho_{opt} = \frac{1}{3}$	$\frac{1}{\sqrt{5}}$	$\frac{1}{\sqrt{5}}$	0.25	0.125
2-D line relaxation	$\omega_{opt} = \frac{4}{5}\rho_{opt} = \frac{3}{5}$	$\omega_{opt} = \frac{2}{3}\rho_{opt} = \frac{1}{3}$	$\frac{1}{\sqrt{5}}$	$\frac{1}{\sqrt{5}}$	0.25	0.125
2-D point relaxation	$\omega_{opt} = \frac{4}{5}\rho_{opt} = \frac{3}{5}$		$\frac{1}{2}$		0.25	
1-D point relaxation	$\omega_{opt} = \frac{2}{3}\rho_{opt} = \frac{1}{3}$		$\frac{1}{\sqrt{5}}$		0.125	

TABLE A.7

Explicit formula for the smoothing factors of different methods with block and point relaxation

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide, Second Edition*, SIAM, 1995.

[2] A. Brandt, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, Tech. Report GMD-Studien 85, May 1984.

[3] W. L. Briggs, *A multigrid tutorial*, SIAM, 1988.

[4] T. F. Chan and H. C. Elman, *Fourier analysis of iterative methods for elliptic boundary value problems*, SIAM Rev., 31 (1989), pp. 20–49.

[5] D. J. Mavriplis, *Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes*, in Proceedings of the AIAA CFD Conference, June 1997.

[6] C. C. Douglas, *Caching in with multigrid algorithms: Problems in two dimensions*, Paral. Alg. and Appl., 9 (1996), pp. 195–204.

[7] C. C. Douglas, S. Malhotra, and M. H. Schultz, *Transpose free alternating direction smoothers for serial and parallel methods*. MGNET at http://www.mgnet.org/, 1997.

[8] T. Gjesdal, *Smoothing analysis of multicolor pattern schemes*, J. Comp. Appl. Math., 85 (1997), pp. 345–350.

[9] J. C. Agüi and J. Jimenez, *A binary tree implementation of a parallel distributed tridiagonal solver*, Parallel Comput., 21 (1995), pp. 233–241.

[10] J. E. Dendy Jr., M. P. Ida, and J. M. Rutledge, *A semi-coarsening multigrid algorithm for simd machines*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1460–1469.

[11] J. López, O. Platas, F. Argüello, and E. L. Zapata, *Unified framework for the parallelization of divide and conquer based tridiagonal systems*, Parallel Comput., 23 (1997), pp. 667–686.

[12] J. E. Jones and N. D. Melson, *A note on multi-block relaxation schemes for multigrid solvers*, Tech. Report 97-15, ICASE, 1997.

[13] Krechel, Plum, and Stüben, *Parallelization and vectorization aspects of the solution of tridiagonal linear systems*, Parallel Comput., 14 (1990), pp. 31–49.

[14] A. Krechel and K. Stüben, *Operator dependent interpolation in algebraic multigrid*, Tech. Report 1048, GMD, 1997.

[15] J. Linden, G. Lonsdale, H. Ritzdorf, and A. Schüller, *Scalability aspects of parallel multigrid*, Future Generation Computer Systems, 10 (1994), pp. 429–440.

[16] I. M. Llorente and F. Tirado, *Relationships between efficiency and execution time of full multigrid methods on parallel computers*, IEEE Trans. on Parallel and Distributed Systems, 8 (1997), pp. 562–573.

[17] I. M. Llorente, F. Tirado, and L. Vázquez, *Some aspects about the scalability of scientific applications on parallel architectures*, Parallel Comput., 22 (1996), pp. 1169–1195.

[18] O. A. McBryan, P. O. Frederickson, J. Linden, A. Schüller, K. Solchenbach, K. Stüben, C. Thole, and U. Trottenberg, *Multigrid methods on parallel computers-a survey of recent developments*, Impact of Computing in Science and Engineering, 3 (1991), pp. 1–75.

[19] P. Messina, *High-performance computers: The next generation*, Computer in Physics, 11 (1997), pp. 444–466.

[20] W. Mulder, *A new multigrid approach to convection problems*, J. Comput. Phys., 83 (1989), pp. 303–

323.

[21] N. H. Naik and J. V. Rosendale, *The improved robustness of multigrid elliptic solvers based on multiple semicoarsened grids*, SIAM J. of Num. Anal., 30 (1993), pp. 215–229.

[22] K. Oosterlee and T. Washio, *An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems, Part I: The standard grid sequence*, Tech. Report 980, GMD, 1996 (to appear in SIAM J. Sci. Comp.).

[23] A. Overman and J. V. Rosendale, *Mapping robust parallel multigrid algorithms to scalable memory architectures*, in Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods, 1993.

[24] J. Ruge and K. Stüben, *Algebraic multigrid (AMG)*, in Multigrid methods, S. McCormick, ed., vol. 5 of Frontier in Applied Mathematics, SIAM, 1986.

[25] L. Stals and U. Rüde, *Techniques for improving the data locality of iterative methods*, Tech. Report MRR97-038, School of Mathematics Sciences of the Australian National University, 1997.

[26] K. Stüben, *Parallel and robust multigrid with applications*, in Proceedings of the Euroconference on Supercomputation in Nonlinear and Disordered Systems: Algorithms, Applications and Architectures, L. Vázquez, F. Tirado, and I. M. Llorente, eds., World Scientific, 1997.

[27] K. Stüben and U. Trottenberg, *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, in Multigrid methods, W. Hackbusch and U. Trottenberg, eds., vol. 960 of Lecture Notes in Mathematics, Springer-Verlag, 1982, pp. 1–176.

[28] C. Thole and U. Trottenberg, *Basic smoothing procedures for the multigrid treatment of elliptic 3d operators*, Appl. Math. and Comput., 19 (1986), pp. 333–345.

[29] T. Washio and K. Oosterlee, *An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems, Part II: Flexible 2D and 3D multiple semicoarsening*, Tech. Report 1012, GMD, 1996 (to appear in SIAM J. Sci. Comp.).

[30] P. Wesseling, *A survey of smoothing analysis results*, in Multigrid Methods III, W. Hackbusch and U. Trottenberg, eds., Birkhäuser–Verlag, 1991, pp. 105–127.

[31] ——, *An introduction to multigrid methods*, John Wiley & Sons, 1992.

[32] I. Yavneh, *Multigrid smoothing factors for red-black Gauss-Seidel relaxation applied to a class of elliptic operators*, SIAM J. of Num. Anal., 32 (1995), pp. 1126–1138.