SIAM J. SCI. COMPUT.
Vol. 21, No. 5, pp. 1670–1690

KRYLOV SUBSPACE ACCELERATION OF NONLINEAR MULTIGRID WITH APPLICATION TO RECIRCULATING FLOWS*

C. W. OOSTERLEE† AND T. WASHIO‡

Abstract. This paper deals with the combination of two solution methods: multigrid and GMRES [*SIAM J. Sci. Comput.*, 14 (1993), pp. 856–869]. The generality and parallelizability of this combination are established by applying it to systems of *nonlinear* PDEs. As the "preconditioner" for a nonlinear Krylov subspace method, we use the full approximation storage (FAS) scheme [*Math. Comp.*, 31 (1977), pp. 333–390], a nonlinear multigrid method. The nonlinear Krylov acceleration is applied also on coarse grids, so that recirculating incompressible flow problems discretized with a higher order upwind scheme can be solved efficiently.

Key words. Krylov subspace methods, nonlinear multigrid, parallel computing, incompressible Navier–Stokes equations, second-order upwind discretizations

AMS subject classifications. 65N55, 65F99, 65Y05, 76D05

PII. S1064827598338093

1. Introduction.

In the search for efficient solvers that are generally applicable, we consider multigrid a preconditioner for Krylov subspace methods. Although multigrid solution methods can be $O(N)$ solvers, it is not always easy to choose the optimal components for difficult problems. Complicating factors include convection dominance, anisotropies, nonlinearities, and non-M-matrix properties. All of these phenomena can occur with convection-dominated rotating computational fluid dynamics (CFD) problems discretized by higher order upwind discretization.

It is also not trivial to design *robust* multigrid solvers for large classes of nonlinear CFD problems. By Krylov subspace acceleration, we aim to increase the class of problems for which proven efficient multigrid solvers exist. In the case of an $O(N)$ multigrid solver, we try to improve its convergence rate or to reduce the number of necessary smoothing steps by using the acceleration technique. For large systems of equations, the acceleration technique is, in general, cheaper in CPU time than one implicit smoothing iteration on the finest grid.

It is possible to generalize this Krylov accelerated multigrid solution method, called KMG, to systems of equations and to nonlinear problems. In a natural way, a *nonlinear multigrid method* can be constructed (see [3] [10]), so that linear and nonlinear problems can be handled by the one and the same method. In section 3, it is shown that the same principle holds for the Krylov subspace acceleration. If we switch from GMRES to FGMRES [20], it is not so difficult to generalize the Krylov subspace acceleration to nonlinear situations. With this Krylov accelerated multigrid method many different nonlinear problems can be solved, as we have already shown in [23]. The acceleration method is, in fact, related to the reduced rank extrapolation method (RRE), as it is presented in [21], and to the acceleration cycle presented in [6].

An interesting topic for making the KMG method feasible for solving realistic

*Received by the editors April 29, 1998; accepted for publication (in revised form) September 17, 1998; published electronically April 28, 2000.
http://www.siam.org/journals/sisc/21-5/33809.html
†GMD, Institute for Algorithms and Scientific Computing, D-53754 Sankt Augustin, Germany (oosterlee@gmd.de).
‡C&C Research Laboratories, NEC Europe Ltd., D-53757 Sankt Augustin, Germany (washio@ccrl-nece.technopark.gmd.de).

3D CFD problems is the reduction of storage. Much storage is necessary when a large Krylov subspace is needed for searching an accelerated solution. We discuss a possibility and show results of the use of the Krylov acceleration also on coarse grids. On coarse grids much less storage is needed for a Krylov subspace. We denote the method by *KMG-f* if the Krylov subspace acceleration is only performed on the finest grid, and by *KMG-fc* if the acceleration is performed on fine and coarse grids.

Another advantage of the KMG solver lies in its parallelizability. Both parts can be parallelized in a standard way on the basis of grid partitioning [15]. In this paper, we concentrate on a class of problems that is particularly difficult to solve with standard multigrid methods: rotating flows with dominant convection. In section 4, we explain in some detail the difficulties of rotating flow problems. In [7] and [25], the same types of problems are treated and remedies are given for improving the multigrid convergence rate for the first-order upwind discretization of the convection terms. We investigate the effect of the fine and coarse grid Krylov subspace acceleration for first-order and, especially, for second-order upwind discretizations. Parallel numerical convergence results with the nonlinear KMG solution method are presented in section 6, for example, for the system of incompressible Navier–Stokes equations, where the discretization is based on the primitive variables. The flux-difference splitting discretization of the Navier–Stokes equations [8] with higher order upwind κ-discretizations [13] for the convective terms is second-order accurate.

2. The Krylov acceleration for nonlinear multigrid methods. The linear system discussed in section 2 is

$$Au = b, \tag{2.1}$$

and a multigrid cycle for solving (2.1) is described by the following matrix splitting:

$$Mu_j + (A - M)u_{j-1} = b, \tag{2.2}$$

where u_j represents a current and u_{j-1} a previous solution vector. This formulation is equivalent to

$$u_j - u_{j-1} = M^{-1}(b - Au_{j-1}) = M^{-1}r_{j-1} \quad \text{and} \tag{2.3}$$

$$u_{j+1} - u_j = (I - M^{-1}A)(u_j - u_{j-1}), \tag{2.4}$$

with r being the residual vector and $I - M^{-1}A$ representing the multigrid iteration matrix. Multigrid scheme (2.2) for solving (2.1) is used as a *preconditioner* for GMRES(m). The parameter m represents the number of vectors stored after which the GMRES(m) algorithm will restart. The right preconditioned method then considers the equation

$$AM^{-1}(Mu) = b. \tag{2.5}$$

GMRES(m) searches for a solution u_j in the following subspace:

$$M(u_j - u_{j-m}) \in \text{span}[r_{j-m}, (AM^{-1})r_{j-m}, \ldots, (AM^{-1})^{m-1}r_{j-m}] =: K^m(AM^{-1}, r_{j-m}),$$

where $K^m(AM^{-1}, r_{j-m})$ is the Krylov subspace. GMRES minimizes the residual r_j in the L_2-norm,

$$\min_{M(u_j - u_{j-m}) \in K^m(AM^{-1}, r_{j-m})} \|b - Au_j\|_2, \tag{2.6}$$

whereas the conjugate gradient method (CG) minimizes the residual in the A^{-1}-norm. It is clear that CG requires symmetric positive definite matrices A and M^{-1} and that GMRES is not restricted in this sense. With a powerful preconditioner, like multigrid, one can work with a small subspace (a small m) and still achieve considerable convergence acceleration for many problems. The work for one GMRES iteration consists of a matrix-vector product and a preconditioning step. In our case, the computation is dominated by the multigrid preconditioning.

It is possible to generalize the KMG solution method to nonlinear problems. Moreover, the linear solution method is, in fact, automatically included in the nonlinear method. The nonlinear system treated here is described by

$$R(u) = b - A(u) = 0. \tag{2.7}$$

2.1. The nonlinear multigrid method. Solution method M gives an updated intermediate solution u_M from u_{j-1},

$$u_M = M(R, u_{j-1}). \tag{2.8}$$

(The new iterant u_j is selected from iterant u_M, and an accelerated vector u_A as is discussed in the next section.) In our case, M represents one nonlinear multigrid FAS cycle [3]. Another more generally known research direction is to construct nonlinear solution methods on the basis of a global Newton linearization. The resulting linear system is then solved with a linear multigrid method [10] and/or with Krylov methods.

The basis of our method is *nonlinear* multigrid, for which Jacobians $\partial A/\partial u$ are only evaluated and stored locally (point- or line-wise) in the smoother on every grid level. This is an advantage of FAS compared with methods with a global Newton linearization, which need to store the whole Jacobian matrix. Another related advantage is that, in the case of an exceptionally ill-conditioned Fréchet derivative, global Newton linearization will lead to convergence problems, whereas the FAS convergence, due to the local linearization, might not suffer from the condition of the derivative. This problem occurs when the given continuous problem has no Fréchet derivative. In that case, the condition of the Jacobian deteriorates severely as the grid is refined, and Newton's method convergence deteriorates accordingly.

A difference between linear multigrid and nonlinear FAS is that, in linear multigrid methods on the coarse grid, *corrections* to a fine grid solution are explicitly calculated, whereas, for FAS, the corrections are obtained with coarse grid *approximations* u_H to a solution. The coarse grid equation for the FAS scheme on a coarse grid with grid size H looks like

$$A_H(u_H) = A_H(w_H + \widehat{I}_h^H u_h) = I_h^H(R(u_h)) + A_H(\widehat{I}_h^H u_h) = g_H. \tag{2.9}$$

Operators I_h^H and $\widehat{I}_h^H$ are (usually different) restriction operators transferring grid functions from fine grids, h, to coarse grids, H; u_h represents a current fine grid approximation. The coarse grid *correction* w_H is prolongated and added to the fine grid solution. The FAS scheme coincides with the linear multigrid scheme for linear equations, as can be seen from (2.9) by subtracting $A_H(\widehat{I}_h^H u_h)$.

2.2. The Krylov acceleration. The nonlinear Krylov method is explained as a finest grid level acceleration and can be seen as an outer iteration for the nonlinear multigrid preconditioner. As explained in detail in [23], the (nonlinear) search directions are constructed from available intermediate solution vectors. Jacobians are

approximated with help of the residual vectors for the intermediate solutions; they are not recomputed explicitly in our Krylov acceleration technique. This technique for accelerating the convergence of solution method (2.8) consists of two steps.

We derive a minimization problem of the residual, which is mathematically equivalent to the minimization performed in GMRES. A method equivalent to the flexible GMRES method (FGMRES) [20] is constructed, which allows a preconditioner to change from iteration to iteration. As a consequence, we need to store, as in FGMRES, twice as many vectors as in standard GMRES, namely, m residual vectors and m solution vectors.

In nonlinear cases, the Krylov subspace span

$$[r_{j-m}, AM^{-1}r_{j-m}, \ldots, (AM^{-1})^{m-1}r_{j-m}]$$

is not available, since we have neither a linear operator A nor the linear multigrid preconditioner M. Instead, we have a nonlinear operator R and a nonlinear multigrid FAS cycle, as discussed above.

Let u_j be a sequence of solutions. If, instead of (2.3), we use (2.4) and induction, then we obtain a different representation of the Krylov subspace K^m:

$$\begin{aligned} K^m(AM^{-1}, r) &:= \operatorname{span}[r_{j-m}, AM^{-1}r_{j-m}, \ldots, (AM^{-1})^{m-1}r_{j-m}] \\ &= M \cdot \operatorname{span}[u_{j-1} - u_{j-m}, u_{j-2} - u_{j-m}, \ldots, u_{j-m+1} - u_{j-m}]. \end{aligned} \tag{2.10}$$

We can define the space in (2.10) for linear and nonlinear equations. From this subspace, we try to find a solution to a minimization problem. Assume we have intermediate solution vectors $u_{j-m}, \ldots, u_{j-1}$ and their residual vectors $R(u_{j-m}), \ldots, R(u_{j-1})$.

Step 1. We accelerate the FAS process (2.8) and search for an improved solution u_A, compared with the FAS solution u_M in the space $u_M + \operatorname{span}[u_{j-m} - u_M, \ldots, u_{j-1} - u_M]$ defined by (2.10),

$$u_{h,A} = u_{h,M} + \sum_{i=1}^{m} \alpha_i (u_{h,j-i} - u_{h,M}). \tag{2.11}$$

For simplicity, we assume $j \geq m$. The main question that now arises is how to approximate $R(u_A)$ by an affine mapping $f : \alpha \to f(\alpha)$ in order to reduce $\|R(u_A)\|_2$, as in (2.6). The basis for the approximation is the following linearization:

$$R(u_A) = R\left(u_M + \sum_{1 \leq i \leq m} \alpha_i (u_{j-i} - u_M)\right) \approx R(u_M) + \sum_{1 \leq i \leq m} \alpha_i \left(\frac{\partial R}{\partial u}\right)_{u_M} (u_{j-i} - u_M). \tag{2.12}$$

There are basically two different possibilities for approximating the right-hand side of (2.12).

- One can explicitly calculate and use the Jacobian at u_M:

$$f(\alpha) = R(u_M) - \sum_{i=1}^{m} \alpha_i \frac{\partial A}{\partial u_M}(u_{j-i} - u_M). \tag{2.13}$$

  In this way, one needs extra calculations for obtaining the Jacobian (plus m matrix-vector multiplications with the Jacobian). When the Jacobian is calculated for another intermediate solution u_i instead of at u_M, a less accurate approximation of $R(u_A)$ is expected.

- One can also use a *Jacobian-free approximation*, which, in our method, looks like

$$f(\alpha) = R(u_M) + \sum_{i=1}^{m} \alpha_i (R(u_{j-i}) - R(u_M)). \tag{2.14}$$

Here, the storage of residual vectors $R(u_{j-i})$ is necessary.

The Jacobian-free approximation is chosen in this work. The minimization of the functional is done in the L_2-norm. It has a direct correspondence to FGMRES in case of linear problems. Jacobians are approximated by the residual vectors for the intermediate solutions. These vectors are already available on the finest grid level in multigrid (before performing a restriction, for example). In this sense, the method is well suited to the nonlinear multigrid method.

The work for solving the minimization problem is relatively small (compared to a multigrid cycle). It requires the minimization of

$$\left\| R(u_M) + \sum_{1 \le i \le m} \alpha_i (R(u_{j-i}) - R(u_M)) \right\|_2. \tag{2.15}$$

This simply reduces (2.15) to the solution of the following linear system:

$$H(m) \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{pmatrix}. \tag{2.16}$$

Here, $H(m) = (h_{ik})$ is defined by

$$\begin{aligned} h_{ik} = {} & (R(u_{j-i}), R(u_{j-k})) - (R(u_M), R(u_{j-i})) \\ & - (R(u_M), R(u_{j-k})) + (R(u_M), R(u_M)), \end{aligned} \tag{2.17}$$

where $(.,.)$ is the standard inner product. Right-hand sides $\beta_1, \ldots, \beta_m$ are defined by

$$\beta_i = (R(u_M), R(u_M)) - (R(u_M), R(u_{j-i})). \tag{2.18}$$

The matrix $H(m)$ is solved with a direct solution method. In principle, it is possible that $H(m)$ is an ill-conditioned matrix, since all residuals $R(u_{j-i})$ are based on solutions u_{j-i} that are converging toward the discrete solution u_h. However, with the powerful multigrid preconditioner, it is observed and expected that the (small) matrices $H(m)$ are still satisfactorily conditioned, so that the α_i-solutions of (2.18) are acceptable. If the $H(m)$ matrix were ill-conditioned for a certain problem, Step 2 in the algorithm described below prevents the algorithm from accepting a possibly bad accelerated solution. If $H(m)$ is a singular matrix, more details are given in [23].

Step 2. Since (2.12) may not be a reasonable approximation *in the nonlinear case* (some of the intermediate solutions may be far away from the desired solution), criteria are needed for selecting u_j from u_M and u_A. The criteria for selecting the accelerated solution u_A as u_j (and for restarting) are the following.

1. The norm of $R(u_A)$ is not too large compared with $R(u_M)$ and intermediate residuals $R(u_{j-i})$:

$$\|R(u_A)\|_2 < \gamma_A \min_{1 \le i \le m} (\|R(u_M)\|_2, \|R(u_{j-i})\|_2). \tag{2.19}$$

2. u_A is not too close to any of the intermediate solutions unless a considerable decrease of the residual norm is achieved:

$$\epsilon_B \|u_A - u_M\|_2 < \min_{1\le i\le m} (\|u_A - u_{j-i}\|_2)$$
$$\text{or}$$
$$\|R(u_A)\|_2 < \delta_B \min_{1\le i\le m} (\|R(u_M)\|_2, \|R(u_{j-i})\|_2). \tag{2.20}$$

In several numerical experiments in [23], we found that taking γ_A somewhat larger than 1 on the finest grid, for example, $\gamma_A = 2$, brings a more reliable convergence for problems that are difficult for the multigrid preconditioner. In general, the method is not very sensitive to the choice of γ_A. Criterion 2 is necessary to prevent stagnation in the convergence. It is possible that $\|R(u_M)\|_2$ becomes significantly larger than the minimum of the residual norms of the intermediate solutions or that $R(u_M) - R(u_{j-i})$ is orthogonal to the span$[R(u_{j-1}), R(u_{j-2}) \dots]$ (illustrated in Figure 2.1) even though the multigrid process leads towards the desired solution. In these cases, a small weight $1 - \sum_{1\le i\le m} \alpha_{j-i}$ will be given to u_M and the weight α_i of a minimal intermediate residual $R(u_{j-i})$ could be close to 1, so that the acceleration process forces the solution back to a previous intermediate solution u_{j-i}.

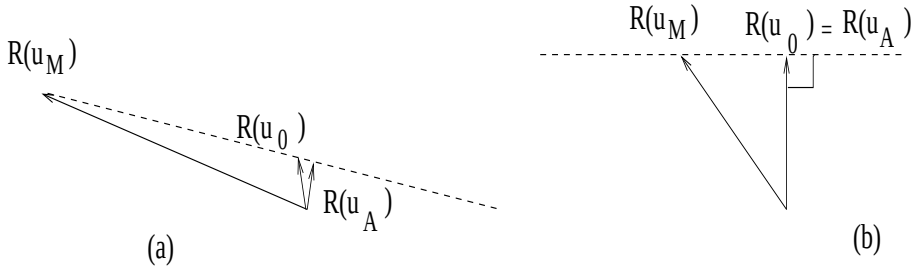


FIG. 2.1. *Two example situations (with $m = 1$) in which the acceleration process stagnates:* (a) $\|R(u_M)\|_2$ *is larger than the previous residual vector and* (b) $R(u_M) - R(u_0)$ *is orthogonal to (the span of) the previous residual vector.*

In order to prevent this phenomenon, one should carefully take into account the distances between the solutions and the reduction of the residual norm as is done in criterion 2. We fix the parameters in criterion 2 in all numerical experiments in section 4 as $\epsilon_B = 0.1$, $\delta_B = 0.9$. The same criteria, (2.19) and (2.20), are used for *restarting* the Krylov subspace, as is presented in [23]. For restarting, γ_A is set to 1 and it takes place if the criteria are not satisfied in two consecutive iterations.

2.3. Parallelism. Our aim is to find efficient parallel solution methods for applications in which the domain is split into a small number of blocks, for example, into 16 blocks. The KMG solution method is parallelized in a straightforward way. The usual way of parallelizing in several application fields with block-structured grids is called grid partitioning [15]. The global grid is split into blocks. Parallelization is done by assigning each block to a process. This parallelization strategy is well suited to both the multigrid preconditioner and the Krylov subspace method.

This means for *the Krylov method* that the inner products necessary for constructing the small system (2.16) are first evaluated locally. The calculated parts of the inner product are then communicated among the nodes and summed, so that the matrix elements in (2.16) are available in all processors. Every processor solves the

system (2.16) and has the α_i values available. In every processor, the updated part of the new accelerated solution u_A is then obtained. Clearly, in this way, the search for an improved solution is not performed per processor, but over the whole domain.

To get an efficient parallel *multigrid method*, the grid is stored with some overlap along the block boundaries on all fine and coarse grids [15]. Communication among the nodes is required for keeping the values in overlap regions up to date. Here, the coarsest grid is chosen such that in every processor there is at least one unknown available. It depends on the parallelization of the smoother whether the parallel multigrid program is similar to or different from the sequential program. For example, using a red-black point smoother gives an identical sequential and parallel multigrid smoother. After a partial smoothing sweep over all red (odd) grid unknowns, communication takes place in which the odd points in the overlap regions are updated. Then, after the second partial sweep over the black points, we have an identical red-black smoother as in the sequential case. Since transfer operators between fine and coarse grids are fully parallel operators, the parallel multiblock multigrid method is almost identical to the sequential single block method. The only difference might be the choice of the coarsest grid.

In case of line smoothers in CFD applications, however, it is common to change the line smoother into a line smoother per block, a line Gauss–Seidel smoother with Jacobi aspects at block boundaries. It is found in [14] that, for several channel flow CFD problems where a domain was split into one and into several blocks, the single block multigrid convergence with alternating Gauss–Seidel line smoothing is regained if, after each x-line sweep per block, an extra communication step is performed and, after each y-line sweep (plus the usual communication), one grid line along the interior block boundaries is smoothed additionally. Without these additions, the convergence rate degrades. In this case, the parallel multigrid algorithm is different from the sequential one and is related more to certain domain-decomposition methods, but the single grid convergence is regained for parallel calculations with a relatively large number of blocks [14]. Here, in section 5, when line smoothers are applied, we do *not* need these additions to regain the single block convergence with KMG.

3. Krylov acceleration on coarse grids. A new idea is to incorporate the nonlinear Krylov subspace acceleration into the multigrid cycle and to apply it also on the coarse grids. A target for realistic applications of KMG is a reduction of the storage needed for building the Krylov subspace (2.10). Of course, the method is parallel, and one can benefit from the distributed storage of MIMD machines or of clusters of workstations. However, in general, it is preferred to use the extra memory for solving finer problems. A good convergence with a small Krylov subspace means, in practice, that the multigrid preconditioner must be "as good as possible." It is, for example, not possible to get a convergence acceleration from a small subspace if a smoother is used that does not reduce a large number of frequencies well.

From the dual view of multigrid, in which the coarse grid equation (2.9) is the main equation and the fine grid is merely available for achieving a better accuracy, it seems logical to also accelerate the coarse grid equation. The size of the Krylov subspace on the finest grid can be reduced by additionally using the Krylov acceleration of the coarse grid problem (2.9). The Krylov subspace vectors on a first coarse grid are four times smaller than the fine grid vectors in 2D and eight times smaller in 3D. This subspace acceleration on the coarse grids is helpful if the solution of the multigrid coarse grid equation (2.9), with the coarse grid smoother and the remaining coarse grid multigrid cycle, leads to multigrid convergence problems. The coarse grid Krylov

acceleration then removes "problematic frequencies" on coarse grids. So, the coarse grid smoothing is not replaced by the subspace acceleration. After the smoothing the intermediate iterants are combined so that a coarse grid solution vector with a smaller residual results. We apply this coarse grid Krylov acceleration in the FAS version of multigrid. It is, however, also possible to perform the acceleration within the linear correction scheme (CS), as it is, in a different terminology, as performed in [7].

The difficulty in accelerating the convergence on a coarse grid is that the right-hand side of the coarse grid equation (2.9) is not constant as on the finest grid. It changes after every restriction from the fine grid during the FAS cycle. A similar approach to the Krylov acceleration on the finest grid is, however, applicable to the coarse grid equation. Assume that we stored a sequence of m_c coarse grid solutions, possibly from different multigrid iterations,

$$\{u_{H,j-1}, u_{H,j-2}, \ldots, u_{H,j-m_c}\},$$

and a sequence of the action of A_H to these solutions,

$$\{A_H(u_{H,j-1}), A_H(u_{H,j-2}), \ldots, A_H(u_{H,j-m_c})\},$$

and that we have an approximate solution $u_{H,M}$ of the coarse grid equation (2.9). Then we construct an accelerated solution $u_{H,A}$ from

$$u_{H,A} = u_{H,M} + \sum_{i=1}^{m_c} \alpha_i (u_{H,j-i} - u_{H,M}) \tag{3.1}$$

and use the following approximation of $g_H - A(u_{H,A})$ (2.9) as in the Krylov acceleration on the finest grid,

$$g_H - A_H(u_{H,A}) \approx g_H - A_H(u_{H,M}) - \sum_{i=1}^{m_c} \alpha_i (A_H(u_{H,j-i}) - A_H(u_{H,M})), \tag{3.2}$$

and minimize the L_2-norm of the right-hand side of (3.2). Here, a different size m_c for the coarse grid Krylov subspace can be chosen. If the right-hand side of the coarse grid equation (2.9) does not change much, the affine space

$$u_{H,M} + \text{span}[u_{H,j-1} - u_{H,M}, u_{H,j-2} - u_{H,M}, \ldots, u_{H,j-m_c} - u_{H,M}] \tag{3.3}$$

is an appropriate subspace to search a better solution.

One can perform the additional coarse grid Krylov acceleration on only one, on certain, or on all coarse grids. In this paper, we mainly use the acceleration on all fine and coarse grids; the method is then denoted by KMG-fc (compared to KMG-f, if the acceleration is performed only on the finest grid). Often, however, it is found that the use of the Krylov subspace acceleration on two or three additional coarse grids already results in a considerable convergence improvement.

It is possible to perform the coarse grid Krylov acceleration basically at three stages in the multigrid cycle, namely: after coarse grid presmoothing (stage 1), after coarse grid postsmoothing (stage 2), or just before the prolongation (stage 3). The last two possibilities are identical for a V-cycle. They are indicated in Figure 3.1 for a V-cycle on grid level 2. In Figure 3.2, the coarse grid acceleration on the grid levels 2 and 3 in a multigrid W-cycle is also presented. The subspace is truncated, i.e., the first vector is removed from the subspace and the next one is added to the subspace,

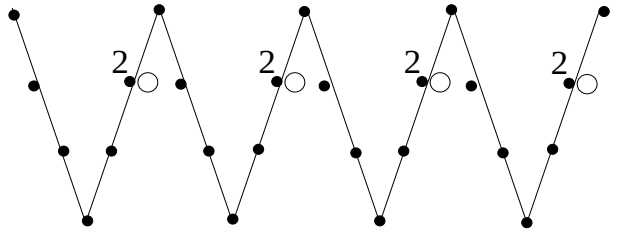


FIG. 3.1. *The Krylov acceleration on level* 2 *in multigrid V*(1,1)*-cycles indicated by* ∘*, acceleration at stage* 2*.*

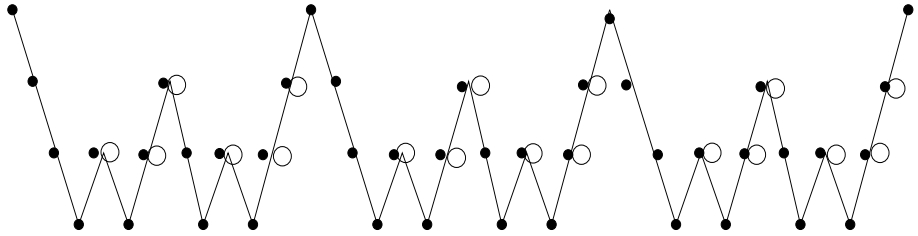


FIG. 3.2. *The Krylov acceleration in multigrid W*(1,1)*-cycles indicated by* ∘*, acceleration after postsmoothing on levels* 2 *and* 3*.*

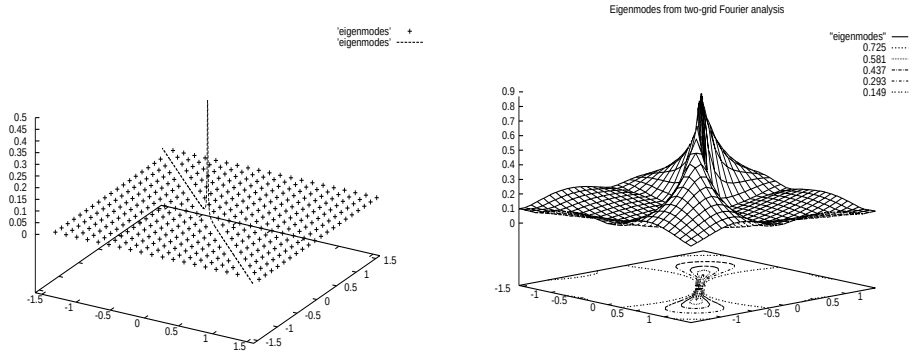
if it already contains m_c vectors. In the actual experiments, we find that performing the subspace acceleration at stage 2 (after the standard postsmoothing) results in the best multigrid convergence. Selection and restarting conditions, based on (2.19), (2.20) with $\gamma_A = 1$, are also necessary in the coarse grid Krylov acceleration.

4. Multigrid for recirculating flow problems. It is well known that rotating convection-dominated convection–diffusion-type problems are difficult to solve efficiently by standard multigrid methods [7], [10]. This class of singularly perturbed problems is an interesting candidate for the coarse grid Krylov acceleration, since it is the coarse grid correction that yields the multigrid convergence difficulties, as we will explain in this section. The standard convection-diffusion equation,

$$-\epsilon\Delta u + a(x,y)\frac{\partial u}{\partial x} + b(x,y)\frac{\partial u}{\partial y} = 0 \ \text{ on } \Omega = (0,1)^2, \tag{4.1}$$

with $0 < \epsilon << 1$, serves as the model problem to explain the difficulties occurring in general rotating flow problems.

4.1. Standard upwind discretization. With a standard upwind discretization for the convective terms, it is difficult to obtain multigrid convergence factors less than 0.5 when a standard $2h$-discretization on the coarse grids is used (first observed in [4]). This can be clearly observed with local mode two-grid Fourier analysis. Figure 4.1(a) presents the eigenmode spectrum obtained from the two-grid Fourier analysis for a convection-dominated convection-diffusion equation (4.1) with $\epsilon = 10^{-5}$ and $a = b = 1$, discretized by first-order upwinding. The smoother used is an alternating line Gauss–Seidel. A two-grid factor of 0.5, as found in Figure 4.1(a), results in multigrid convergence factors that increase recursively toward one when standard multigrid cycles are employed. A general conclusion, which can also be observed in Figure 4.1(a), is that characteristic low frequency error components, which are constant along the characteristics of the advection operator, are not reduced efficiently on coarse grids.



(a) (b)

FIG. 4.1. (a): *The two-grid asymptotic convergence factor* 0.5 *with a standard upwind discretization and* (b): *the factor* 0.95 *with the* ($\kappa = 0$)*-discretization from two-grid Fourier analysis for a convection-diffusion equation.*

The different scaling of convection $(a(x,y)/h,\ b(x,y)/h)$ and diffusion (ϵ/h^2) is not approximated properly with an upwind discretization on the coarse $2h$ grids. A 1D standard upwind discretization can be written as a combination of a central difference discretization plus a Laplace term with artificial viscosity:

$$au_x = a\left(\frac{u_i - u_{i-1}}{h}\right) = a\left(\frac{u_{i+1} - u_{i-1}}{2h}\right) + \left(\frac{ah}{2}\right)\frac{-u_{i-1} + 2u_i - u_{i+1}}{h^2} \tag{4.2}$$

(if $a = \text{constant} > 0$). On a coarse $(2h)$ grid, a factor 0.5 is missing in the scaling of the different terms, due to the appearance of the additional h in the artificial viscosity term [5]. An important observation is that, in the case of scalar convection-dominated flow problems with an inflow and an outflow boundary, convergence difficulties do not occur with line smoothers, for example, in alternating directions, since these smoothers are exact solvers on the fine grid along the characteristic direction and, therefore, they also take care of problematic low frequency error components. This is not the case for rotating convection-dominant flow problems, for which no real inflow and outflow boundary exist. Boundary information is mainly diffusing into the domain and often boundary layers are found in the solution. The coarse grid problem, which is shown with Fourier analysis, is therefore most relevant for rotating convection-dominated problems. The factor 0.5 is observed in the two-grid local mode Fourier analysis, since in this analysis the (positive) influence of boundary conditions on the convergence is not taken into account. In the local mode Fourier analysis [3] one works with the eigenfunctions $\phi(\theta, \mathbf{x}) = e^{i\mathbf{k}\theta}$ with Fourier frequencies θ, which implicitly indicates that the analysis is performed on an infinite grid and that boundary conditions cannot be taken into account. In [7] and [25], improved multigrid solution methods, especially for the upwind discretization of this problem, were constructed by means of the overweighting of residuals, an acceleration via a defect-correction approach on all grids, and/or by the use of a point smoother in the flow direction. Here, we investigate the influence of the additional coarse grid Krylov acceleration on the convergence, which is in some sense a generalization of the techniques proposed in [7].

4.2. Higher order upwind discretizations. Higher order upwind discretizations of convection, like the upwind κ-discretizations [13], are more accurate and therefore more interesting for the discretization of the convective terms. A 1D upwind κ-discretization can be written as a combination of a central difference discretization plus a second-order dissipation term, which is proportional to the third derivative of u:

$$au_x = a\left(\frac{u_{i+1} - u_{i-1}}{2h}\right) + \left(\frac{ah^2(1-\kappa)}{4}\right)\frac{u_{i-2} - 3u_{i-1} + 3u_i - u_{i+1}}{h^3} \tag{4.3}$$

(if $a = \text{const} > 0$). In this paper, we use the discretization for $\kappa = 0$, the Fromm scheme, which is $O(h^2)$ accurate and looks for the standard convection-diffusion equation (4.1) with $a, b = \text{const} > 0$ like

$$[\widehat{L}]_h = \frac{a}{h}[1/4 \quad -5/4 \quad 3/4 \quad 1/4 \quad 0]_h + \frac{b}{h}\begin{bmatrix} 0 \\ 1/4 \\ 3/4 \\ -5/4 \\ 1/4 \end{bmatrix}_h + \frac{\epsilon}{h^2}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}_h . \tag{4.4}$$

For $a < 0$, similar formulae are found. The linear κ-scheme is a first choice for obtaining second-order accurate schemes with a dominating convection term. They work satisfactorily for a large class of CFD problems, including the incompressible Navier–Stokes equations. The κ-schemes are, however, not monotone, which means that they have to be modified (with limiters) for CFD problems containing strong gradients, like shocks. For discretizations with limiters, the method introduced in this paper also applies. For the problems investigated here, the κ-schemes result in satisfactory discretizations.

With respect to multigrid convergence, it is known that using the basic iterative methods as a smoother directly on $\widehat{L}$ leads actually to a *diverging* method. Multistage smoothers or defect-correction approaches are commonly used for this type of discretizations in convection-dominated problems.

In [18], an alternating symmetric line smoother, the KAPPA smoother, was proposed, analyzed, and evaluated for higher order upwind discretizations with the κ-scheme. The higher order upwind discretization is treated directly in the multigrid solution method, not by an outer defect-correction iteration. The KAPPA smoother is based on a splitting into a first-order upwind part and a remaining part (Splitting II in [18]). With smoothers based on this splitting, fast convergence is obtained for many convection-dominated channel flow problems with inflow and outflow boundaries. For rotating flow problems with a dominating convection term, however, convergence problems can be expected again with standard multigrid methods, as we see from the two-grid Fourier analysis results (shown in Figure 4.1(b) for (4.1) with $a = b = 1$ and $\epsilon = 10^{-5}$). A two-grid convergence factor of at best 0.95 is expected, which is far from satisfactory. The multigrid convergence will tend to 1. With the Krylov acceleration on fine and coarse grids, we expect to improve the convergence rate. The robust symmetric alternating line variant of the KAPPA smoother is explained briefly, since it is the basis for the numerical examples in the next section. A symmetric alternating line smoother smooths all lines in the forward x-direction and in the forward y-direction. Then, the relaxation takes place in backward directions. The other multigrid components are standard components.

Higher order upwind discretizations (4.3), (4.4) of (4.1) have the general form

$$\widehat{L}u = \sum_{\mu\in J}\sum_{\nu\in J} a^{(2)}_{\mu\nu} u_{i+\mu,j+\nu} \tag{4.5}$$

with coefficients $a^{(2)}_{\mu\nu}$ and a set of indices $J = \{-2,-1,0,1,2\}$. A part of the symmetric alternating line KAPPA smoother is the y-line smoother for a forward ordering of lines, which we explain here in more detail. It is constructed with a special splitting of $\widehat{L}$ into L^0, L^+, and L^-. The x-line smoother can be explained similarly. First, we explain the superscripts: 0 indicates operator parts corresponding to grid points currently treated, + refers to grid points with already updated unknowns, and − means it is still to be updated, as in [22]. In the case of a forward y-line smoother, 0 represents the line $i = i_c = \text{const}$ where the unknowns are currently updated, + indicates the lines $i < i_c$, and − refers to the unknowns on the lines $i > i_c$. The L^0 parts in the KAPPA smoother are not the operator elements $a^{(2)}$ from the higher order upwind discretization of the grid points under consideration but the first-order upwind operator elements. L^0 and L^+ look (with the stencil notation) like

$$L^0 := \begin{pmatrix} & & 0 & & \\ & & a^{(1)}_{01} & & \\ 0 & 0 & a^{(1)}_{00} & 0 & 0 \\ & & a^{(1)}_{0-1} & & \\ & & 0 & & \end{pmatrix}, \quad L^+ := \begin{pmatrix} & & 0 & & & \\ & & 0 & & & \\ a^{(2)}_{-20} & a^{(2)}_{-10} & 0 & 0 & 0 & 0 \\ & & 0 & & & \\ & & 0 & & & \end{pmatrix}. \tag{4.6}$$

$L^- = \widehat{L} - L^0 - L^+$. So $a^{(1)}_{00}$, etc., denote operator elements from the first-order accurate upwind discretization. With the definitions of L^0, L^+, and L^- given, the splitting for obtaining u_{j+1} for the grid points $i = i_c$ looks, with underrelaxation, like

$$L^0 u_* = f - (\, L^- u_j + L^+ u_{j+1}), \tag{4.7}$$

$$u_{j+1} = \omega u_* + (1-\omega) u_j \tag{4.8}$$

for $i = i_c$, after which the next line of points $i = i_c+1$ is relaxed. With the notation as explained above, it is possible that iteration indices j and $j+1$ appear on a right-hand side.

5. Numerical results. In this section, we show convergence results for KMG-f and KMG-fc applied to linear and nonlinear rotating 2D problems on MIMD machines.

5.1. Rotating convection-diffusion problems. The first problem investigated is the 2D scalar convection-diffusion equation (4.1) with a dominating rotating convection term:

$$a(x,y) = -\sin(\pi x)\cdot\cos(\pi y), \quad b(x,y) = \sin(\pi y)\cdot\cos(\pi x).$$

Dirichlet boundary conditions are prescribed:

$$u|_\Gamma = \sin(\pi x) + \sin(13\pi x) + \sin(\pi y) + \sin(13\pi y).$$

Parameter ϵ in (4.1) varies between 10^{-3} and 10^{-5}.

This equation discretized with standard upwinding is studied in many papers, for example, in [7], [25], [17], [23]. Here, we also evaluate the convergence with Fromm's

TABLE 5.1

Multigrid and fine and coarse grid accelerated multigrid convergence for the 2D rotating convection-diffusion equation for different values of ε.

Method	$O(h)$-discretization		
	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$
Multigrid	0.48 (8; 87 s)	0.66 (14; 149 s)	0.66 (19; 200 s)
Fine grid acc. ($m = 2$)	0.22 (6; 70 s)	0.40 (8; 92 s)	0.29 (9; 103 s)
Fine grid acc. ($m = 15$)	0.15 (6; 70 s)	0.28 (7; 81 s)	0.21 (8; 93 s)
Fine ($m = 2$) + coarse grid ($m_c = 5$) acc.	0.16 (6; 73 s)	0.27 (6; 73 s)	0.22 (7; 85 s)
	$O(h^2)$-discretization		
	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$
Multigrid	0.10 (5; 85 s)	0.45 (6; 101 s)	0.90 (63; 986 s)
Fine grid acc. ($m = 2$)	0.08 (4; 73 s)	0.31 (7; 123 s)	0.85 (42; 704 s)
Fine grid acc. ($m = 15$)	0.07 (4; 73 s)	0.30 (7; 123 s)	0.72 (24; 421 s)
Fine ($m = 2$) and coarse grid ($m_c = 5$) acc.	0.07 (4; 76 s)	0.25 (6; 112 s)	0.65 (14; 250 s)

discretization (4.4) for the convection, as discussed in the previous section. The finest grid consists of 256^2 cells.

In [17], we investigated the convergence of a multigrid as a preconditioner with GMRES acceleration for the standard upwind discretization, where we used a relatively large number of vectors m in the Krylov subspace, namely, $m = 20$. Here, the aim is to use a small m and to investigate the coarse grid Krylov subspace acceleration. The performance of KMG-f is compared to the KMG-fc solution method. In the latter case, we have a restart parameter m on the finest grid and m_c for the size of the subspace on the coarse grids. Parameter m varies between 2 and 15, when the Krylov acceleration takes place only on the finest grid. The sizes are fixed, $m = 2$ and $m_c = 5$, when Krylov acceleration takes place on the fine and coarse grids.

A robust standard FAS multigrid method based on the W(0,1)-cycle consisting of 9 levels is chosen as the preconditioner. The calculation starts on the coarsest grid (full multigrid (FMG)) to improve an initial approximation on the finest grid. A robust smoother for the $O(h)$ discretization is the symmetric alternating line Gauss–Seidel smoother [24]. For the $O(h^2)$ discretization, it is the symmetric alternating KAPPA line smoother discussed in the previous section. An underrelaxation parameter is not needed for this scalar test problem.

Table 5.1 compares the asymptotic convergence of multigrid with the average convergence (after a large number of iterations) of KMG. Also, the number of iterations plus cpu time, on an RS6000 workstation, for reducing the residual by 6 orders of magnitude is indicated in brackets.

Table 5.1 shows a very satisfactory convergence improvement with KMG-fc, especially for the convection-dominant cases. For the first-order upwind discretization on this 256^2 grid, the improvement is seen for $\epsilon = 10^{-4}$ and $\epsilon = 10^{-5}$. A gain of 12 iterations in reducing the initial residual by six orders of magnitude is observed with $\epsilon = 10^{-5}$ for the KMG-fc solution method. The improvement is also significant for the discretization with the Fromm scheme. Here, the improvement is best for $\epsilon = 10^{-5}$. For this test, the coarse grid Krylov subspace acceleration with $m = 2$ on the finest grid performs better than the fine grid Krylov acceleration with $m = 15$.

TABLE 5.2
Accuracy of the higher order upwind discretization for rotating convection.

h:	1/16	1/32	1/64	1/128	1/256
$\|u_h - u\|_2$	6.8e-3	1.5e-3	2.1e-4	2.3e-5	2.2e-6

TABLE 5.3
Convergence of multigrid preconditioned Krylov methods.

h:	Acceleration	# levels	ρ_h, (# its., time in s)
1/64	Fine grid ($m = 15$)	1	0.61 (26, 48 s)
	Fine and coarse grid ($m = 2, m_c = 5$)	2	0.58 (22, 41 s)
	Fine and coarse grid ($m = 2, m_c = 5$)	3	0.60 (22, 41 s)
1/128	Fine grid ($m = 15$)	1	0.64 (29, 212 s)
	Fine and coarse grid ($m = 2, m_c = 5$)	2	0.59 (23, 168 s)
	Fine and coarse grid ($m = 2, m_c = 5$)	3	0.58 (23, 169 s)
	Fine and coarse grid ($m = 2, m_c = 5$)	4	0.60 (25, 185 s)
1/256	Fine grid ($m = 15$)	1	0.63 (31, 909 s)
	Fine and coarse grid ($m = 2, m_c = 5$)	2	0.58 (26, 760 s)
	Fine and coarse grid ($m = 2, m_c = 5$)	3	0.52 (21, 622 s)
	Fine and coarse grid ($m = 2, m_c = 5$)	4	0.54 (22, 655 s)
	Fine and coarse grid ($m = 2, m_c = 5$)	5	0.54 (23, 685 s)

Standard multigrid solves this problem well for several ϵ-values. For certain ϵ-values, the problem is not solved efficiently by standard multigrid without the acceleration, which is often observed.

The second test investigates the *h-independent convergence* of KMG-fc for convection-diffusion equation (4.1) with a dominating rotating convection term:

$$a(x, y) = -\sin(2\pi x) \cdot \cos(2\pi y), \quad b(x, y) = \sin(2\pi y) \cdot \cos(2\pi x), \quad \epsilon = 10^{-5}.$$

It can be seen that $a = b = 0$ is found on four points in the domain. Dirichlet boundary conditions and right-hand side f are prescribed such that an analytical solution results, namely,

$$u = 1 - (x - 1/4)^3 - (y - 3/4)^3.$$

It was examined in [5] that, for rotating flow problems, standard upwind discretization does not reach $O(h)$ accuracy, due to "anisotropic artificial viscosity." This is also observed for this test problem with the standard upwind discretization. Here, we investigate the Fromm's discretization for which it is shown, in Table 5.2, where the error between the numerical and analytical solutions is presented for several grid sizes, that a very satisfactory accuracy is reached. It can be seen that the numerical solution is at least second-order accurate.

Table 5.3 shows the average convergence (for a large number of iterations), plus the number of iterations and cpu time (in seconds), to reduce the residual by 10^{-6} for three grid sizes. KMG-f with $m = 15$ is compared to KMG-fc, where $m = 2$ and $m_c = 5$. We evaluate the influence of the number of levels on which Krylov acceleration is applied (indicated by # levels in the table). The multigrid preconditioner is the same as in the previous experiment (KAPPA smoother). FAS W(1,1)-cycles are used.

In Table 5.3, it is observed that the KMG-fc solution method results in a satisfactory level-independent convergence rates for this test problem. The use of the Krylov subspace acceleration on three grids is most efficient in Table 5.3. The final residual

is then obtained after about 22 iterations for all grid sizes. Also, the cpu time needed for reducing the initial residual by six orders of magnitude is smallest in case of the acceleration on three levels. The use of the Krylov subspace acceleration on more coarse grids, however, does not influence the convergence dramatically.

5.2. Incompressible flow problems. Next, incompressible flow examples are treated. This nonlinear system of PDEs is discretized with a modern upwind method, namely, with a flux-difference splitting discretization [8], in which the higher order upwind κ-discretization [13] is applied for convective fluxes. Originally, these upwind methods were developed for compressible flow problems. An overview of the flux splitting discretization methods is given in [11]. In [8], it is shown that the discretization concept also applies for incompressible equations.

The 2D system of the incompressible Navier–Stokes equations is written in the conservative form

$$\frac{\partial \mathbf{f}}{\partial x}(\mathbf{u}) + \frac{\partial \mathbf{g}}{\partial y}(\mathbf{u}) = \frac{\partial \mathbf{f_v}}{\partial x}(\mathbf{u}) + \frac{\partial \mathbf{g_v}}{\partial y}(\mathbf{u}), \tag{5.1}$$

where $\mathbf{u} = (u, v, p)^T$. For the incompressible Navier–Stokes equations, we have

$$\mathbf{f} = \begin{bmatrix} u^2 + p \\ uv \\ c^2 u \end{bmatrix}, \qquad \mathbf{g} = \begin{bmatrix} uv \\ v^2 + p \\ c^2 v \end{bmatrix},$$

$$\mathbf{f}_v = \begin{bmatrix} \frac{1}{Re}\partial u/\partial x \\ \frac{1}{Re}\partial v/\partial x \\ 0 \end{bmatrix}, \qquad \mathbf{g_v} = \begin{bmatrix} \frac{1}{Re}\partial u/\partial y \\ \frac{1}{Re}\partial v/\partial y \\ 0 \end{bmatrix}.$$

Here, c is a constant reference velocity (here, $c = 1$). The 2D vertex-centered discretization of (5.1) on a collocated grid is flux-difference splitting with the κ-discretization. The second-order accuracy of the resulting flux-difference discretization is checked for several steady flow problems at low and high Reynolds numbers in [8] for 2D problems and in [16] for 3D problems. For incompressible Navier–Stokes equations, it is not necessary to implement a limiter. For many different (2D and 3D) problems at low and high Reynolds numbers, oscillations (for example, in the pressure distribution, as they occur near discontinuities for compressible flow problems) did not appear.

5.3. Driven cavity problem. We want to show the benefits of the KMG-fc solution method for some multiblock applications at high Reynolds numbers. A first rotating flow test problem is the driven cavity flow at a high Reynolds number, Re = 10000. With a moving top wall $u = 1$, and velocities prescribed zero on all walls, recirculating flow is generated with small recirculation zones near the boundary. Figure 5.1(a) presents streamlines for this flow example. A calculation with the stationary incompressible Navier–Stokes equations is performed on a 193^2 grid, with stretching near all boundaries to capture the recirculation zones and boundary layers. Figure 5.1(b) shows that the centerline velocity profiles u at $x = 0.5$ and v at $y = 0.5$ agree well with the reference data from [9].

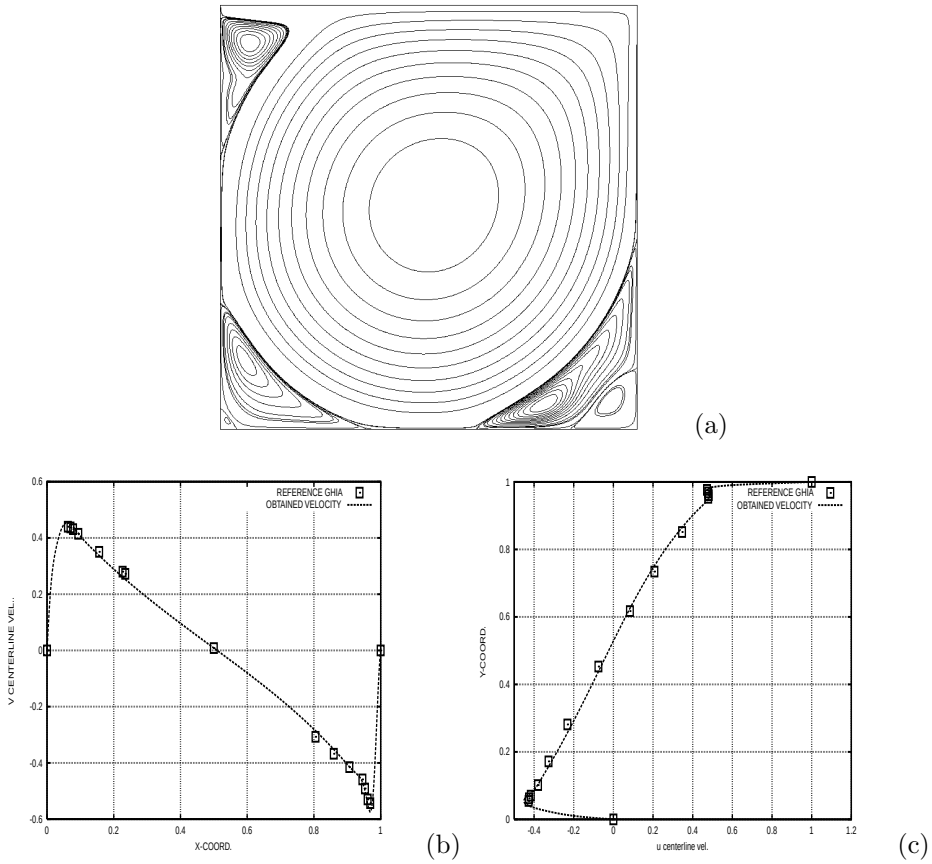


FIG. 5.1. *The driven cavity flow problem at* $\mathrm{Re} = 10000$. (a) *Streamlines;* (b) *and* (c) *Centerline velocity profiles compared to reference data from* [9].

Figure 5.2 presents the single block convergence for this problem. Also, the average reduction factor of the sum of the three equations

$$\rho = \left(\frac{\sum_{i=1}^{3} |r_{h,25}^{(i)}|_\infty}{\sum_{i=1}^{3} |r_{h,5}^{(i)}|_\infty} \right)^{\frac{1}{20}} \tag{5.2}$$

is presented in the figure. In all figures, it can be seen that the first five iterations are different from the asymptotic convergence behavior. Therefore, they are not considered in (5.2). In Figure 5.2, again the pure multigrid convergence is compared to the convergence of KMG-f with $m = 15$ and to KMG-fc with the acceleration on 6 levels with $m = 2$ and $m_c = 5$. The coupled KAPPA smoother is used, so that the second-order accurate discretization of the 3 equations is solved directly in the multigrid method. A W(1,1)-cycle is chosen with 7 multigrid levels. Underrelaxation parameter ω is set to 0.85 when the Krylov acceleration is applied on the coarse grids.

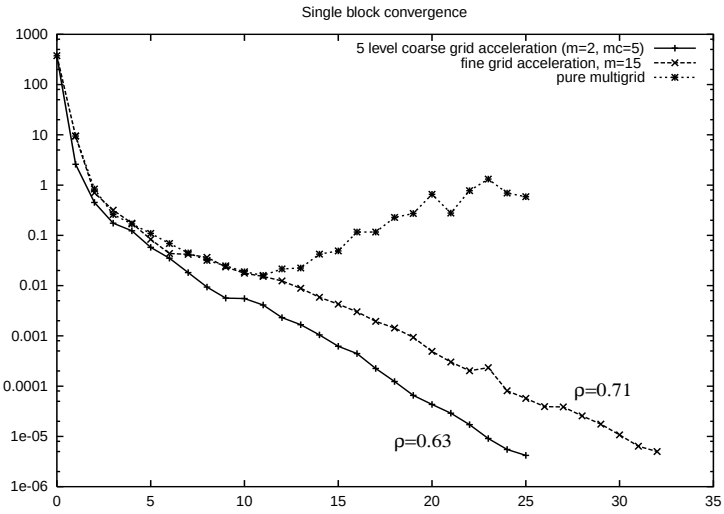


FIG. 5.2. *Single block convergence versus the number of iterations for driven cavity test problem at* Re = 10000.

It must be chosen smaller if the coarse grid acceleration is not applied, namely, 0.6 in order to achieve convergence.

It can be seen in Figure 5.2 that the multigrid solution method is not converging on this stretched grid. This is observed for several values of the underrelaxation parameter ω and for different numbers of smoothing iterations. Also, it can be seen that KMG-fc is converging faster than KMG-f with a relatively large Krylov subspace of 15. To achieve a residual of 10^{-4}, approximately five iterations are saved additionally. The costs in cpu time are similar for both methods.

In Figure 5.3, we compare the single block KMG-fc method with its multiblock version. The 193^2 stretched grid is split into 1×2, 2×2, and 2×4 load-balanced blocks. The problems are solved on 2, 4, and 8 processors of an IBM-SP2 machine. Line smoothing is now performed within a block, as mentioned above. Figure 5.3 compares the reduction of the residual with the wall-clock time. The average reduction factor ρ is also presented in the figure. It can be seen that the average reduction factor is not influenced much by the Jacobi aspect of the KAPPA smoothers at the interior boundaries. An extra update along the interior boundaries is not necessary for this test problem. Efficiencies of about 85% are achieved with this method on 8 processors.

5.4. The twin roller example. The second incompressible Navier–Stokes application is the numerical simulation of steel flow in a twin roller (see Figure 5.4). Similar problems are also treated in [2], [12]. The twin roller process integrates both the casting and the rolling process, which are treated separately in the conventional steel production. Liquid steel is entering the simulation via a nozzle (see Figure 5.4) in a pool between two rotating rolls, which are water cooled. A solid steel strip is the resulting product. The twin roller is a multiphysics problem, with structural mechanics equations considered where the solidification takes place, for example, below the "kissing point" (Figure 5.4) and near the rolls. Here, we study the convergence of the KMG method for a second-order accurate simulation of the liquid steel, which is considered to behave as a Newtonian fluid [12]. In the real computations, the influence of the temperature on the flow is modeled by an energy equation. We do not

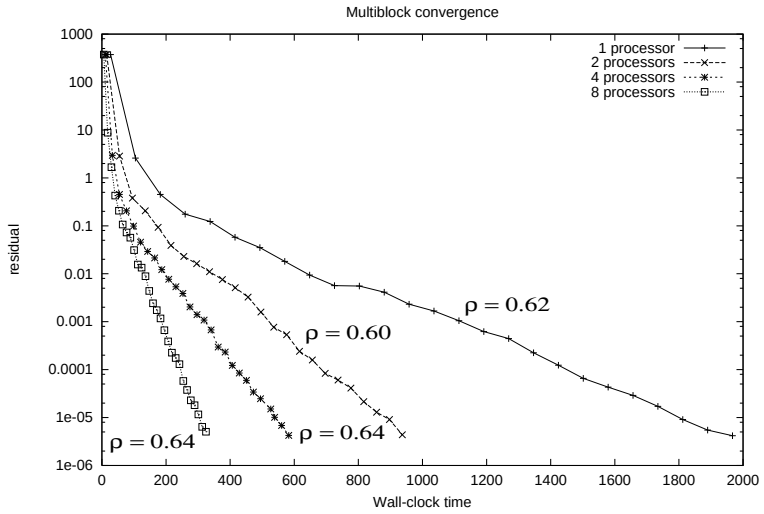


FIG. 5.3. *Convergence versus wall-clock time for the multiblock driven cavity test problem at* $\mathrm{Re} = 10000$.

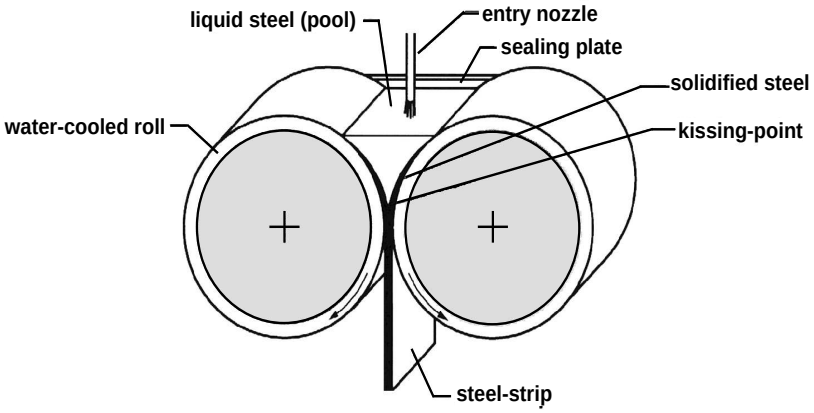


FIG. 5.4. *The twin-roller model example for the incompressible Navier–Stokes problems.*

do this here (although it is not an essential difficulty for the solution method).

A fast and robust solution method for the CFD part is a necessary requirement for the overall multiphysics problem. We compute a 2D incompressible Navier–Stokes flow in a domain with a submerged entry nozzle. Other inflow boundaries (other nozzles) are treated, for example, in [2]. Symmetric boundary conditions are placed at the centerline of the pool, so that half of a domain needs to be calculated. The computational domain is shown in Figure 5.5. In the domain, a boundary-fitted curvilinear grid is generated. Dirichlet boundary conditions for u and v at the roll describe the speed of the roll. A free surface condition is placed at the top wall and on Neumann-type boundary conditions at the outlet. Different types of extreme stretched cells, especially in the lower part of the domain, are found in the numerical grid, which is divided into six blocks for parallel processing. Four blocks contain

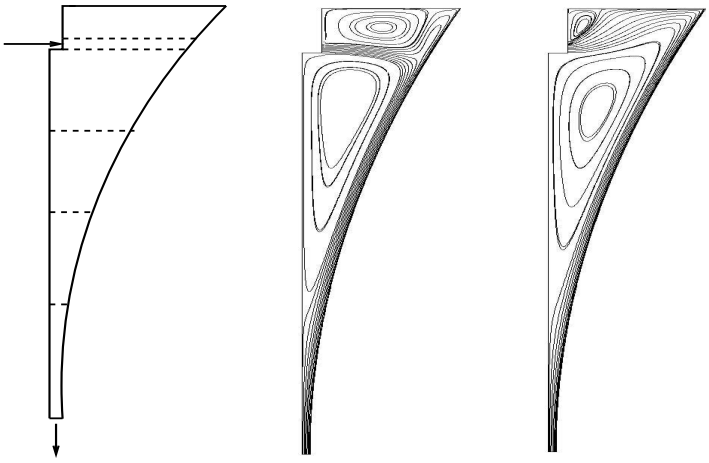


FIG. 5.5. *The domain with six blocks for the incompressible twin roller problem and the streamlines for* Re = 100000 *and* Re = 10000.

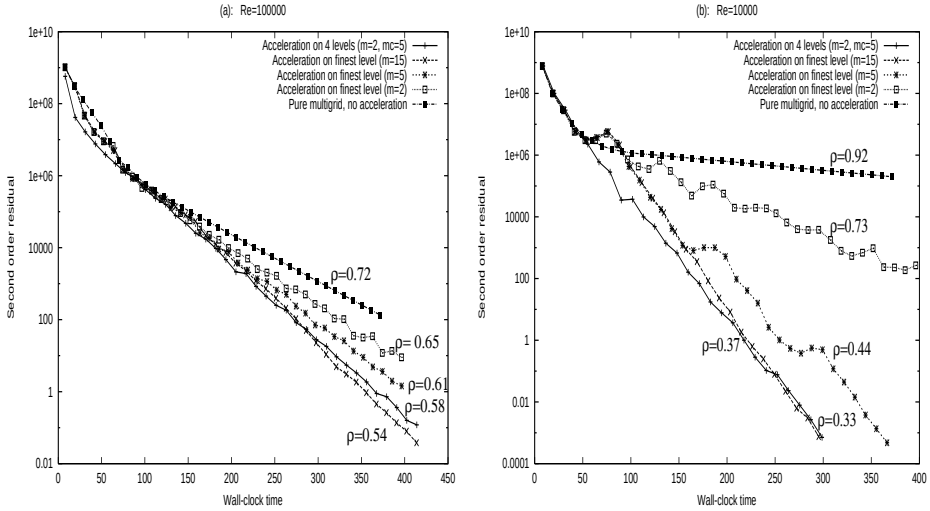


FIG. 5.6. *The six block convergence of the residual from the second-order accurate discretization for two Reynolds numbers.*

128×80 cells, one consists of 64×64 cells and one of 128×64 cells. A load imbalance exists, which is not critical. Here, the Reynolds number varies between $\mathrm{Re} = 10^4$ and $\mathrm{Re} = 10^5$, which are standard parameters considered in [2] and [12]. Large circulation zones appear in these flow problems, as can be observed from the streamlines for the two Reynolds numbers in Figure 5.5.

The multigrid FAS scheme used for solving this problem is the same as for the driven cavity problem. It consist of an F(0,1) cycle with four multigrid levels. The smoother is again the coupled symmetric alternating KAPPA smoother with $\omega = 0.85$. Figure 5.6 shows the reduction of the residual from second-order accurate discretization versus the wall-clock time and ρ as in (5.2) for both Reynolds numbers. Standard FAS multigrid is compared to KMG-f with different sizes of the

subspace; $m = 2, m = 5$, and $m = 15$; and with KMG-fc. In the latter case, we choose $m = 2, m_c = 5$. It can be seen that the convergence of the latter method is comparable to the convergence of KMG-f with $m = 15$ for both tests. Furthermore, we observe unexpected FAS multigrid convergence behavior in Figure 5.6. The convergence is satisfactory for the high Reynolds number 100000. It slows down dramatically for $\mathrm{Re} = 10000$. The convergence of KMG-fc is regular, fast, and reliable.

6. Conclusions. We presented a parallel nonlinear Krylov acceleration strategy for solving nonlinear equations. This strategy is similar to FGMRES for linear equations. It is possible to use the acceleration strategy in combination with a nonlinear preconditioner, like a nonlinear multigrid method. The acceleration method is cheap, relative to a nonlinear multigrid preconditioner, since it uses intermediate solutions and residuals that are already calculated in the multigrid iteration. Based on proper selection criteria, it is then decided whether to adopt the accelerated intermediate solution or the solution obtained from the preconditioner. By means of the Krylov acceleration, the multigrid solution method is made much more robust: larger ranges of problem parameters can be solved efficiently. With the additional Krylov subspace acceleration on the coarse grids of the multigrid preconditioner, rotating flow problems discretized by higher order upwind discretizations can be solved efficiently and the size of the Krylov subspace can be reduced.

Acknowledgment. We thank Ullrich Becker-Lemgau for providing the metal flow example.

REFERENCES

[1] D. Bai and A. Brandt, *Local mesh refinement multilevel techniques*, SIAM J. Sci. Comput., 8 (1987), pp. 109–134.
[2] U. Becker-Lemgau, M. G. Hackenberg, B. Steckel, and R. Tilch, *Parallel multigrid in the simulation of metal flow*, in Parallel Computing: Fundamentals, Applications and New Directions, Advances in Parallel Computing 12, E. H. D'Hollander, G. R. Joubert, F. J. Peters, and U. Trottenberg, Elsevier, New York, 1998, pp. 195–202.
[3] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.
[4] A. Brandt, *Multigrid Solvers for Non-Elliptic and Singular-Perturbation Steady-State Problems,* Report, Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel, 1981.
[5] A. Brandt and I. Yavneh, *Inadequacy of first-order upwind difference schemes for some recirculating flows*, J. Comput. Phys., 93 (1991), pp. 128–143.
[6] A. Brandt and V. Mikulinsky, *On recombining iterants in multigrid algorithms and problems with small islands*, SIAM J. Sci. Comput., 16 (1995), pp. 20–28.
[7] A. Brandt and I. Yavneh, *Accelerated multigrid convergence and high-Reynolds recirculating flows*, SIAM J. Sci. Comput., 14 (1993), pp. 607–626.
[8] E. Dick and J. Linden, *A multigrid method for steady incompressible Navier-Stokes equations based on flux-difference splitting*, Internat. J. Numer. Methods Fluids, 14 (1992), pp. 1311–1323.
[9] U. Ghia, K. N., Ghia, and C. T. Shin, *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method*, J. Comput. Phys., 48 (1982), pp. 387–411.
[10] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.
[11] C. Hirsch, *Numerical Computation of Internal and External Flows*, Vol. 2, Wiley, Chichester, New York, 1990.
[12] J. Jestrabek, *Stahlbandherstellung nach dem Zweirollerverfahren – Modellierung des Strömungs- und Temperaturfeldes*, Ph.D. thesis, RWTH Aachen, Verlag Stahleisen, Düsseldorf, Germany, 1995.

[13] B. van Leer, *Upwind-difference methods for aerodynamic problems governed by the Euler equations*, in Large Scale Computations in Fluid Mechanics, Lectures in Appl. Math. 22, II, B. Enquist, S. Osher, and R. Somerville, eds., AMS, Providence, RI, 1985, pp. 327–336.

[14] G. Lonsdale and A. Schüller, *Multigrid efficiency for complex flow simulations on distributed memory machines*, Parallel Comput., 19 (1993), pp. 23–32.

[15] O. A. McBryan, P. O. Frederickson, J. Linden, A. Schüller, K. Solchenbach, K. Stüben, C. A. Thole, and U. Trottenberg, *Multigrid methods on parallel computers—A survey of recent developments*, Impact Comput. Sci. Engrg., 3 (1991), pp. 1–75.

[16] C. W. Oosterlee and H. Ritzdorf, *Flux difference splitting for three-dimensional steady incompressible Navier–Stokes equations in curvilinear coordinates*, Internat. J. Numer. Methods Fluids, 23 (1996), pp. 347–366.

[17] C. W. Oosterlee and T. Washio, *An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems*, SIAM J. Sci. Comput., 19 (1998), pp. 87–110.

[18] C. W. Oosterlee, F. J. Gaspar, T. Washio, and R. Wienands, *Multigrid line smoothers for higher order upwind discretizations of convection-dominated problems*, J. Comput. Phys., 139 (1998), pp. 274–307.

[19] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Comput., 7 (1986), pp. 856–869.

[20] Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[21] D. A. Smith, W. F. Ford, and A. Sidi, *Extrapolation methods for vector sequences*, SIAM Rev., 29 (1987), pp. 199–234.

[22] K. Stüben and U. Trottenberg, *Multigrid methods: Fundamental algorithms, model problem analysis and applications*, in Multigrid Methods, Lecture Notes in Math. 960, W. Hackbusch, U. Trottenberg, eds., Springer, Berlin, 1982, pp. 1–176.

[23] T. Washio and C. W. Oosterlee, *Krylov subspace acceleration for nonlinear multigrid schemes*, Electron. Trans. Numer. Anal., 6 (1997), pp. 271–290.

[24] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley, Chichester, 1992.

[25] I. Yavneh, C. H. Venner, and A. Brandt, *Fast multigrid solution of the advection problem with closed characteristics*, SIAM J. Sci. Comput., 19 (1998), pp. 111–125.