Electronic Transactions on Numerical Analysis.
Volume 6, pp. 271-290, December 1997.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

KRYLOV SUBSPACE ACCELERATION FOR NONLINEAR MULTIGRID SCHEMES*

T. WASHIO† AND C. W. OOSTERLEE‡

Abstract. In this paper we present a Krylov acceleration technique for nonlinear PDEs. As a 'preconditioner' we use nonlinear multigrid schemes such as the Full Approximation Scheme (FAS) [1]. The benefits of nonlinear multigrid used in combination with the new accelerator are illustrated by difficult nonlinear elliptic scalar problems, such as the Bratu problem, and for systems of nonlinear equations, such as the Navier-Stokes equations.

Key words. nonlinear Krylov acceleration, nonlinear multigrid, robustness, restarting conditions.

AMS subject classifications. 65N55, 65H10, 65Bxx.

1. Introduction. It is well known that multigrid solution methods are optimal $O(N)$ solvers, when all components in a method are chosen correctly. For difficult problems, such as some systems of nonlinear equations, it is far from trivial to choose these optimal components. The influence on the multigrid convergence of combinations of complicated factors, like convection-dominance, anisotropies, nonlinearities or non M-matrix properties (the Bratu problem), is often hard to predict. Problems might then occur with the choice of the best under-relaxation parameter in the smoother, with the choice of the coarse grid correction, or with the transfer operators. It was already found in [9] that, for scalar linear model test problems, many of the eigenvalues of a multigrid iteration matrix are clustered around the origin. In some cases there are some isolated large eigenvalues which limit the multigrid convergence, but are well captured by a Krylov acceleration technique. In this paper we concentrate on nonlinear problems, and aim to construct a nonlinear acceleration scheme analogous to GMRES for linear problems.

Another better known research direction is to construct efficient nonlinear solution methods on the basis of a global Newton linearization. The resulting linear system is then solved with a linear multigrid method [5], or with Krylov-type methods (Newton-Krylov methods). A disadvantage of these methods is that in every linearization step a matrix of Jacobians must be evaluated and stored. Since the basis of our method is *nonlinear* multigrid, Jacobians are only evaluated locally (point- or line-wise) in the smoother on every grid level. The nonlinear Krylov acceleration is performed on the finest grid level, and can be seen as an outer iteration for the multigrid preconditioner. Therefore it is very easy to implement it in already existing codes with a nonlinear multigrid variant as a solver, such as Navier-Stokes or Euler codes. The Krylov acceleration method and its algorithmic descriptions are presented in Section 2.2. Another advantage of the nonlinear acceleration scheme presented is the computational efficiency of our method, with respect to the multigrid preconditioner. The (nonlinear) search directions are constructed from available intermediate solution vectors. Jacobians are approximated by the residual vectors for the intermediate solutions, so that they are not recomputed explicitly in our Krylov acceleration technique. In this sense the method is suited very well to the nonlinear multigrid method.

Numerical results with this approach are presented in Section 3 for nonlinear elliptic scalar PDEs, such as the Bratu problem, where the convergence difficulty in obtaining the

*Receive May 15, 1997. Accepted for publication August 30, 1997. Communicated by S. McCormick.
†C&C Research Laboratories, NEC Europe Ltd., D-53757 Sankt Augustin, Germany (email: washio @ ccrl-nece.technopark.gmd.de)
‡GMD, Institute for Algorithms and Scientific Computing, D-53754 Sankt Augustin, Germany (oosterlee@gmd.de)

second solution for certain parameters with FAS is nicely improved by the Krylov acceleration technique. Also results are presented for systems of the incompressible Navier-Stokes and Euler equations, where the discretization is based on the primitive variables.

2. The Krylov acceleration techniques.

2.1. The Krylov acceleration for linear multigrid methods. In this section, we consider why a Krylov acceleration technique is useful for improving the convergence of linear multigrid cycles in difficult problems. Here 'a Krylov acceleration technique' means applying the multigrid cycle as a preconditioner for a Krylov subspace method. In [9], the convergence behavior of the preconditioned GMRES([11]) with multigrid preconditioners was analyzed. It was shown that GMRES helps remove error components corresponding to isolated eigenvalues far away from 1 from the preconditioned matrix. Here, we derive a minimization problem of the residual, which is mathematically equivalent to the minimization problem treated by GMRES, in order to extend the idea of the acceleration technique to the nonlinear cases.

Suppose that the linear system treated is described by

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

and the multigrid cycle is described by means of the following matrix splitting :

$$Mu_i + (A - M)u_{i-1} = b \ . \tag{2.2}$$

Here, M^{-1} corresponds to the mapping from the right-hand vector to the solution after one cycle of multigrid with a zero initial estimate. Let $r_0 (= b - Au_0)$ be the residual for the initial guess u_0. Then we have three different representations of the Krylov subspace K^i as follows :

LEMMA 2.1.

$$\begin{aligned} K^i(AM^{-1}, r_0) &:= \text{span}[r_0, AM^{-1}r_0, \dots, (AM^{-1})^{i-1}r_0] \\ &= M \cdot \text{span}[u_1 - u_0, u_2 - u_1, \dots, u_i - u_{i-1}] \\ &= M \cdot \text{span}[u_0 - u_i, u_1 - u_i, \dots, u_{i-1} - u_i] \ . \end{aligned}$$

Proof. From (2.2), we have

$$\begin{aligned} u_1 - u_0 &= M^{-1}r_0 \ , \\ u_{k+1} - u_k &= (I - M^{-1}A)(u_k - u_{k-1}) \ . \end{aligned}$$

By induction and using these relations, we obtain

$$\text{span}[M^{-1}r_0, M^{-1}AM^{-1}r_0, \dots, (M^{-1}A)^{i-1}M^{-1}r_0] = \text{span}[u_1 - u_0, \dots, u_i - u_{i-1}] \ .$$

Hence the equivalence of the first and second subspace is obtained. The equivalence of the second and third subspace is trivial. □ Let $\tilde{u}_i$ be an element in the subspace $u_i + \text{span}[u_0 - u_i, u_1 - u_i, \dots, u_{i-1} - u_i]$ which minimizes the L_2-norm of the residual. From Lemma 2.1 any element u in $u_i + \text{span}[u_0 - u_i, u_1 - u_i, \dots, u_{i-1} - u_i]$ can be represented by

$$u = u_0 + M^{-1}(\alpha_1 r_0 + \alpha_2 AM^{-1}r_0 + \cdots + \alpha_i (AM^{-1})^{i-1}r_0) \ . \tag{2.3}$$

By substituting (2.3) to the residual equation we obtain :

$$\begin{aligned} r &= b - Au \\ &= r_0 - \alpha_1 AM^{-1}r_0 - \alpha_2 (AM^{-1})^2 r_0 - \cdots - (AM^{-1})^i r_0 \\ &= P_i(AM^{-1})r_0 \ . \end{aligned}$$

Here, P_i is the i-th order polynomial defined by $P_i(x) = 1 - \sum_{k=1}^{i} \alpha_k x^k$. As a consequence, the L_2-norm of the residual $\tilde{r}_i$ of $\tilde{u}_i$ satisfies the following minimization property :

$$\|\tilde{r}_i\|_2 = \min\{\|P_i(AM^{-1})r_0\|_2 \mid P_i : i\text{-th order polynomial with } P_i(0) = 1\} \quad . \tag{2.4}$$

The residual from u_i, obtained from the multigrid iteration, can be represented as :

$$r_i = b - Au_i = (I - AM^{-1})^i r_0 \quad . \tag{2.5}$$

Hence $\|r_i\|_2$ gives one of the upper bounds of $\|\tilde{r}_i\|_2$. For 'easy' problems for multigrid solution methods, such as nice elliptic problems, all eigenvalues of AM^{-1} are close to 1 so that the residual is efficiently reduced by the multiplication of $I - AM^{-1}$. However, it was found in [9] that there are some eigenvalues isolated from 1 for certain difficult problems, which make the multigrid convergence slow. Of course, in practice one will not compute these eigenvalues, but it is interesting to consider what an acceleration technique does in case isolated eigenvalues exist. Let $\lambda_1, \ldots, \lambda_l$ be these isolated eigenvalues. We then can construct the following polynomial P_i $(i > l)$ in order to estimate $\|\tilde{r}_i\|_2$ from (2.4):

$$P_i(x) = (1-x)^{i-l} \frac{\lambda_1 - x}{\lambda_1} \cdots \frac{\lambda_l - x}{\lambda_l} \quad . \tag{2.6}$$

The operator $(\lambda_j I - AM^{-1})/\lambda_j$ removes the problematic component corresponding to the isolated eigenvalue λ_j, whereas components corresponding to eigenvalues close to 1 are reduced by the operator $(I - AM^{-1})^{i-l}$. The existence of an upper bound like $\|P_i(AM^{-1})r_0\|_2$ for $\|\tilde{r}_i\|_2$ ensures that the search for an optimal solution in the space $u_i + \text{span}[u_0 - u_i, \ldots, u_{i-1} - u_i]$ is meaningful. We can also define this space in nonlinear cases. If u_i is close to the solution and the nonlinear operator can be approximated well by a linearized operator around u_i, then the use of this subspace gives efficiency similar to the linear case.

2.2. The Krylov acceleration for nonlinear multigrid methods. A nonlinear system treated here is described by

$$F(u) = 0 \quad . \tag{2.7}$$

We have a solution method M :

$$u_{new} = M(F, u_{old}) \quad , \tag{2.8}$$

which gives an updated solution u_{new} from u_{old}. In our case, M represents one nonlinear multigrid cycle, like FAS [1]. Our technique for accelerating the convergence of the solution method (2.8) consists of three steps and is explained as follows :
Assume we have intermediate solution vectors $u_{\max\{0,k-m\}}, ..., u_{k-1}$ and their residual vectors $F(u_{\max\{0,k-m\}}), ..., F(u_{k-1})$. Here, m is the upper bound for the numbers of recent intermediate solution and residual vectors to be stored.
Step 1: Compute a new solution u^M by

$$u^M = M(F, u_{k-1}) \quad . \tag{2.9}$$

Step 2: Find a more optimal solution in the space $u^M + \text{span}[u_{\max\{0,k-m\}} - u^M, \ldots, u_{k-1} - u^M]$ as in the linear case. For simplicity, we assume $k \geq m$. In case $k < m$, we may substitute k instead of m in the following consideration. The search for a new candidate for the solution is based on the following linear approximation of the nonlinear operator F around u^M on the space $u^M + \text{span}[u_{k-m} - u^M, \dots, u_{k-1} - u^M]$:

$$F(u^M + \sum_{1\le i\le m} \alpha_i(u_{k-m-1+i} - u^M)) \simeq F(u^M) + \sum_{1\le i\le m} \alpha_i \left(\frac{\partial F}{\partial u}\right)_{u^M} (u_{k-m-1+i} - u^M)$$

$$\simeq F(u^M) + \sum_{1\le i\le m} \alpha_i (F(u_{k-m-1+i}) - F(u^M)) \ . \tag{2.10}$$

We search for a combination of the parameters $\alpha_1, \dots, \alpha_m$ which minimizes the L_2-norm of the right-hand side in (2.10). With this combination we define a new candidate u^A from

$$u^A = u^M + \sum_{1\le i\le m} \alpha_i (u_{k-m-1+i} - u^M) \ . \tag{2.11}$$

Note that the above minimization problem is equivalent to the minimization for the L_2-norm of

$$\alpha F(u^M) + \sum_{1\le i\le m} \alpha_i F(u_{k-m-1+i}) \ , \tag{2.12}$$

with the restriction :

$$\alpha + \sum_{1\le i\le m} \alpha_i = 1 \ . \tag{2.13}$$

Step 3: Since (2.10) may not be a reasonable approximation or some of the intermediate solutions may be far away from the desired solution, criteria are needed for selecting u_k from u^M and u^A.

2.2.1. Conditions for preventing stagnation. As criteria to select the accelerated solution u^A as u_k, the following conditions can be considered:

Criterion A The residual norm of u^A is not too large compared to those of u^M and the intermediate solutions :

$$\|F(u^A)\|_2 < \gamma_A \min(\|F(u^M)\|_2, \|F(u_{k-1})\|_2, \dots, \|F(u_{k-m})\|_2) \ . \tag{2.14}$$

Criterion B u^A is not too close to any of the intermediate solutions unless a considerable decrease of the residual norm is achieved :

$$\epsilon_B \|u^A - u^M\|_2 < \min(\|u^A - u_{k-1}\|_2, \dots, \|u^A - u_{k-m}\|_2)$$

or

$$\|F(u^A)\|_2 < \delta_B \min(\|F(u^M)\|_2, \|F(u_{k-1})\|_2, \dots, \|F(u_{k-m})\|_2) \ . \tag{2.15}$$

There are some parameters in the above criteria. Regarding γ_A in Criterion **A**, it seems reasonable to take γ_A smaller than 1. This means that we select u^A only when we observe a decrease of the residual norm from the minimal residual norm of the intermediate solutions. However, in the numerical experiments we will find that taking γ_A larger than 1, for example $\gamma_A = 2$, brings much more reliable convergence for problems that are difficult for the multigrid preconditioner used. Criterion **B** is necessary to prevent stagnation in the convergence. In the multigrid process it is a possible that $\|F(u^M)\|_2$ becomes significantly larger than the minimum of the residual norms of the intermediate solutions, even though the multigrid process leads towards the desired solution. In such a case, as can be imagined from (2.12), a small weight α will be chosen and the weight of the minimal intermediate residual may be close to 1, so that the acceleration process forces the solution back to one of the previous intermediate solutions. In order to prevent this phenomenon, one should carefully take the distances between the solutions and the reduction of the residual norm into account, as is done in Criterion **B**. We fix the parameters in Criterion **B** throughout our numerical experiments in Section 3 as follows:

$$\epsilon_B = 0.1 \ , \quad \delta_B = 0.9 \ . \tag{2.16}$$

In some of the numerical experiments, we justified the criterion by removing it and observing the resulting stagnation.

Restarting: Next we consider the truncation parameter m. In linear cases, taking a larger m brings faster convergence, since the expansion in (2.10) is exact. However, in nonlinear cases, this is not true, since the accuracy of the approximation of (2.10) may decrease as m increases. On the other hand, taking a small m does not bring any improvement for problems with several (10 or more, for example) isolated eigenvalues even though it is desired from the restriction of storage capacity. In order to handle this difficulty, first we determine m from the limitation of the storage capacity and we restart the acceleration process as soon as the approximation of (2.10) is judged inaccurate or as soon as the searched subspace for the acceleration is judged inappropriate. We restart the accelerating process as soon as one of the following conditions is found in 'two consecutive iterations':

Condition C

$$\|F(u^A)\|_2 \geq \gamma_C \min(\|F(u^M)\|_2, \|F(u_{k-1})\|_2, \dots, \|F(u_{k-m})\|_2) \ . \tag{2.17}$$

Condition D

$$\epsilon_B \|u^A - u^M\|_2 \geq \min(\|u^A - u_{k-1}\|_2, \dots, \|u^A - u_{k-m}\|_2)$$
$$\text{and}$$
$$\|F(u^A)\|_2 \geq \delta_B \min(\|F(u^M)\|_2, \|F(u_{k-1})\|_2, \dots, \|F(u_{k-m})\|_2) \ . \tag{2.18}$$

These conditions are just the opposite of those used in the criteria to select u^A. However, the parameter γ_C in Condition **C** should be always larger than 1. In the numerical experiments, we use

$$\gamma_C = \max(2, \gamma_A) \ . \tag{2.19}$$

In Criterion **B** and Condition **D**, the same parameters are used. In order to see the benefits of applying this restarting strategy, we compare the convergence with and without it in the first numerical experiment in Section 3.

2.2.2. The Minimization Problem. Here we will describe and estimate the work for solving the minimization problem from (2.10). The minimization of

$$\|F(u^M) + \sum_{1 \leq i \leq m} \alpha_i (F(u_{k-m-1+i}) - F(u^M))\|_2 \tag{2.20}$$

with respect to the parameters $\alpha_1, \dots, \alpha_m$ is simply reduced to the solution of the following linear system :

$$H(m) \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{pmatrix} \quad . \tag{2.21}$$

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

$$\begin{aligned} h_{ij} =& (F(u_{k-m-1+i}), F(u_{k-m-1+j})) - (F(u^M), F(u_{k-m-1+i})) \\ & - (F(u^M), F(u_{k-m-1+j})) + (F(u^M), F(u^M)) \quad , \end{aligned} \tag{2.22}$$

and $\beta_1, \dots, \beta_m$ are defined by

$$\beta_i = (F(u^M), F(u^M)) - (F(u^M), F(u_{k-m-1+i})) \quad . \tag{2.23}$$

If $\{F(u_{k-m}) - F(u^M), \dots, F(u_{k-1}) - F(u^M)\}$ are linearly dependent vectors, then $H(m)$ is singular, so the solution of (2.21) is not unique. In principle, this is not a problem. We could choose just one solution. However, since we are using a direct solver, which is not suited for solving singular systems, we show that adding a small multiple of the identity, which results in a nonsingular system, does not spoil the solution α_i. So, in order to handle the singular case with the direct solver at hand, we compute $\alpha_1, \dots, \alpha_m$ as the solution of

$$(H(m) + \delta I) \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{pmatrix} \quad . \tag{2.24}$$

Here, δ is small positive number defined by

$$\delta = \epsilon \cdot \max\{h_{11}, \dots, h_{mm}\} \tag{2.25}$$

with a small positive ϵ determined according to the arithmetic accuracy. In our case, we choose $\epsilon = 10^{-16}$. The modification in (2.24) makes it easier to compute $\alpha_1, \dots, \alpha_m$, when $H(m)$ is ill-conditioned. The following lemma confirms that this modification produces only negligible errors to one of the solutions of (2.21) when the smallest nonzero eigenvalue of $H(m)$ is much larger than δ.

LEMMA 2.2. *Assume that H is an $m \times m$ nonzero and nonnegative symmetric matrix, λ is its smallest positive eigenvalue, and β is in the range of H. Assume also that $\bar{\alpha}$ satisfies*

$$H\bar{\alpha} = \beta \quad , \tag{2.26}$$

$$\|\bar{\alpha}\|_2 = \min\{\|\alpha\|_2 \mid H\alpha = \beta\} \quad , \tag{2.27}$$

and α satisfies

$$(H + \delta I)\alpha = \beta \tag{2.28}$$

for a positive number δ. Then

$$\|\alpha - \bar{\alpha}\|_2 \le \frac{\delta}{\lambda + \delta} \|\bar{\alpha}\|_2 \quad . \tag{2.29}$$

Proof. Without loss of generality, we can assume H is a diagonal matrix with diagonal components $\lambda_1, \ldots, \lambda_m$. For i with $\lambda_i > 0$,

$$
\begin{aligned}
|\alpha_i - \bar{\alpha_i}| &= |(\frac{1}{\lambda_i + \delta} - \frac{1}{\lambda_i})\beta_i| \\
&= \frac{\delta}{\lambda_i + \delta}|\bar{\alpha_i}| \\
&\leq \frac{\delta}{\lambda + \delta}|\bar{\alpha_i}| \ .
\end{aligned}
$$

For i with $\lambda_i = 0$, β_i is zero from the assumption. Hence, $\bar{\alpha_i}$ and α_i are also zero. Therefore, inequality (2.29) holds. □

2.2.3. The Algorithm. An algorithmic description of the accelerating process is given below. Here, $\dot{u}_0, \ldots, \dot{u}_{m-1}$ and $\dot{r}_0, \ldots, \dot{r}_{m-1}$ are used to store the intermediate solution and residual vectors of the most recent m iteration steps, namely, $\dot{u}_{mod(k,m)}$ and $\dot{u}_{mod(k,m)}$ are updated to u_k and $F(u_k)$ at the end of the k-th iteration in the acceleration process. The parameter tol is a convergence tolerance with respect to the L_2-norm of the residual.

NLKRY$_m(u, F, M, tol)$ {
    $r := F(u);\ \ \eta := (r, r);\ \ \dot{u}_0 := u;\ \ \dot{r}_0 := r;$
$\sharp$ $q_{11} := \eta;$
    for $k = 1, 2, \ldots$ {
        /⋆ **Computation of** u^M ⋆/
        $u := M(F, u);\ \underline{r := F(u);}\ \underline{\eta := (r, r);}$
        if ($\sqrt{\eta} \leq tol$) return;

        /⋆ **Computation of** u^A ⋆/
        $l := \min(k, m);$
        for $i = 1, \ldots, l$ { $\underline{\xi_i := (r, \dot{r}_{i-1});}$ $\beta_i := \eta - \xi_i;$ }
        for $i = 1, \ldots, l$ { <u>for $j = 1, \ldots, l$</u> { $h_{ij} := q_{ij} - \xi_i - \xi_j + \eta;$ } }
        $\delta := \epsilon \cdot \max(h_{11}, \ldots, h_{ll});$
        Solve $\begin{pmatrix} h_{11}+\delta & \cdots & h_{1l} \\ \vdots & \ddots & \vdots \\ h_{1l} & \cdots & h_{ll}+\delta \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_l \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_l \end{pmatrix};$
        $\underline{u^A := (1 - \sum_{1 \leq i \leq l} \alpha_i) u;}$
        <u>for $i = 1, \ldots, l$</u> { $\underline{u^A := u^A + \alpha_i \dot{u}_{i-1};}$ }
        $\underline{r^A := F(u^A);}$ $\underline{\eta^A := (r^A, r^A);}$

        /⋆ **Selection of the solution** ⋆/
        <u>Select u or u^A as the solution</u> ;
        if (u^A is selected) { $u := u^A;\ r := r^A;\ \eta := \eta^A$ }

        /⋆ **Preparation of the next iteration** ⋆/
        if ($\sqrt{\eta} \leq tol$) return;
        <u>Decide to take the restarting or not</u> ;
        if (the restarting)
            goto $\sharp$;
        else {
            $j := \text{mod}(k, m);$

for $i = 1, \dots, l$ { $\underline{q_{j+1,i} := (r, \dot{r}_i)}$; $q_{i,j+1} := q_{j+1,i}$; }
}
}

Compared to the case where the solution method M is simply applied iteratively, the main overhead of the above algorithm is composed of the underlined operations. Here we assumed that the truncation number m is much smaller than the dimension of the discretized system so that the work to solve the $l \times l$ linear system ($l \leq m$) is negligible. This assumption seems very natural.

At each iteration, the main overhead is $2l + 2$ inner products, l vector updates, evaluation of $F(u^M)$ and $F(u^A)$, examination of the selecting criteria and restarting condition. If the solution method M is a nonlinear multigrid cycle and the truncation number m is 10 or 20, then the total work of these operations is still much less than the solution method itself because the GMRES(m) process is much cheaper than a multigrid preconditioner in linear cases. In Figure 2.1, the combination of a multigrid V-cycle with the Krylov acceleration is shown, for convenience. Note that the solution and the residual vectors only on the finest grid are handled in the Krylov acceleration process.

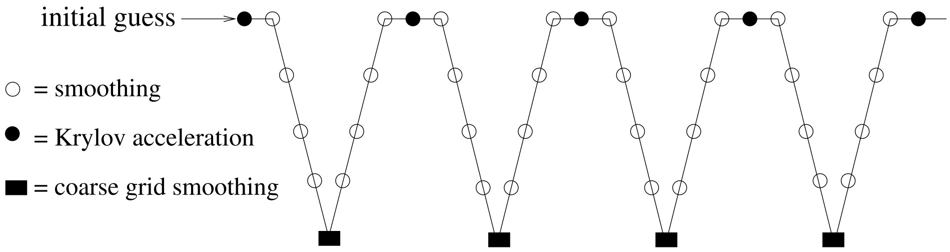

FIG. 2.1. *The structure of a Krylov accelerated multigrid V-cycle.*

3. Numerical experiments. As already mentioned in the introduction, many problems can be solved efficiently with multigrid as a solver. Here, we will concentrate on 'difficult' problems for nonlinear multigrid methods. Four problems are presented, all of which contain certain problem parameters. For certain values of these parameters (c in the Bratu experiment, ϵ in the convection-diffusion example, the Reynolds number in the incompressible Navier-Stokes problem and the Mach number in the Euler test case), the FAS solution method is an excellent solver. However, for other values the convergence of the standard nonlinear multigrid solvers slows down or even diverges. One could then invest more research in an optimal multigrid method that is special for these problem parameters. However we prefer instead to investigate for these cases the effect of our nonlinear Krylov acceleration, with the aim of increased robustness of the standard multigrid method.

3.1. The Bratu problem. The first example treated here is the Bratu problem :

$$\begin{array}{lll} -\Delta u - ce^u = 0 & \text{in} & \Omega = \{(x, y) : 0 < x, y < 1\} \ , \\ u = 0 & \text{on} & \partial\Omega \ . \end{array} \tag{3.1}$$

It is known, cf.[4], that there exists a critical value $c^* \sim 6.808$ for which for $0 < c < c^*$ there are two solutions and for $c > c^*$ there is no solution. The two solutions for $0 < c < c^*$ approach each other as c approaches c^*.

We would like to mention that, for the whole range of possible c values, the first solution is easily obtained. Here, Krylov acceleration is not necessary for $c < 6.5$. To find the second solution solely with multigrid, with FAS [1] or with NLMG [6], is very difficult for small c or c close to the critical value c^*. However, all three Krylov acceleration methods described above give satisfactory convergence for the two solutions with c close to c^*. The difference between the methods is more pronounced when finding the second solution for small c. Hence we test the Bratu problem (3.1) here with $c = 0.2$ and $c = 0.1$. The profiles of the second solutions obtained on the 129^2 grid at $y = 0.5$ are depicted in Figure 3.1. The ratios of ce^u to $4/h^2$ at the maximum of the numerical solution u on the finest 129×129 grid are approximately 0.121 for $c = 0.1$ and 0.0581 for $c = 0.2$ (0.0107 for $c = 1$). These numbers show the loss of diagonal dominance of the Jacobians. In particular, the ratio is already quite large for $c = 0.1$ even on the finest grid. In [5], difficulties in handling linear indefinite problems with multigrid are discussed. For the nonlinear Bratu problem, we cannot find numerical second solutions for $c = 0.1 \sim 0.2$ on the 5^2 grid. On the other hand, on the $9^2 \sim 33^2$ grids, all the "numerical" second solutions we found lost smoothness at their peak. Moreover, we also find solutions which do not have their peak at the center of the domain. There are, however, no such wrong solutions on the 129^2 grid. Therefore, the exact coarse grid correction for this problem is not so helpful as in the usual cases. These facts indicate the difficulty of handling the case $c = 0.1 \sim 0.2$ with the multigrid solution method.

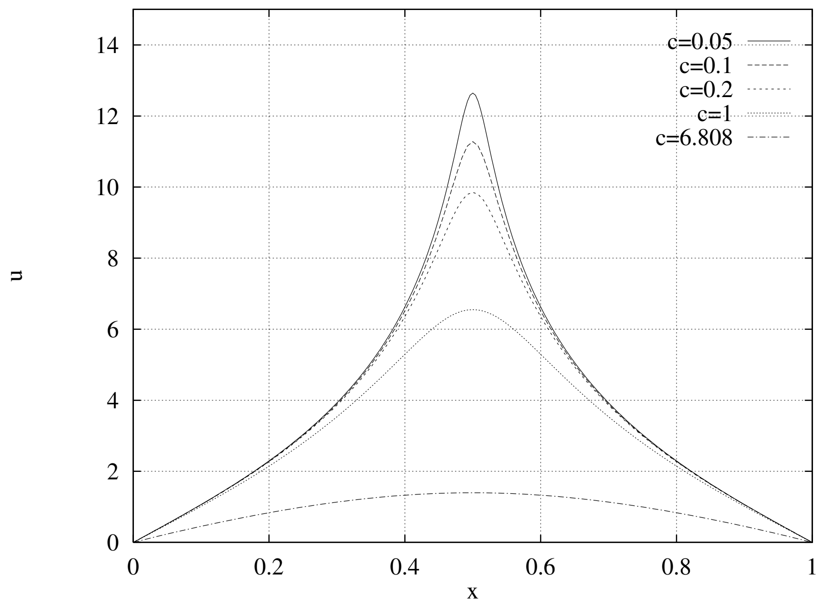

FIG. 3.1. *The profiles of the second solutions at* $y = 0.5$.

We use, as in [6], the damped Jacobi Newton smoother ($\omega = 0.7$) in the nonlinear multigrid cycle. In this smoother, the unknown u is relaxed (by the damped Jacobi iterations) in the discretized linearized equation obtained from

$$-\Delta u - ce^{\tilde{u}}u = g + c(1 - \tilde{u})e^{\tilde{u}} \quad , \tag{3.2}$$

where $\tilde{u}$ is the old solution of the Newton iteration. The Jacobi iteration, however, can easily cause divergence when diagonal dominance is significantly lost in the discretization of the operator $-\Delta - ce^{\tilde{u}}$. This situation frequently occurs when the second solution for a small c is computed on the coarse grids, since $\sup\{u\}$ approaches ∞ as c approaches zero. In order to handle this difficulty, we restart the nonlinear Newton iterations with another linear smoother, if the following condition is detected during the nonlinear iterations:

$$ce^{u_{\max}}/(4/h^2) > 0.1 \quad . \tag{3.3}$$

The smoother for (3.2) in case of (3.3) is a minimization of the L_2-norm of the residual for the search direction determined by the residual vector. The algorithmic description of the smoother is given as follows :

```
SMOOTHER(u_h, c, g_h, ν, ω, μ) {
    w_h := u_h;
    for in = 1, ... , ν {
        b_h := g_h + c(1 − u_h)e^{u_h}; ũ_h := u_h;
        if ( c · e^{u_max}/(4/h^2) < 0.1 )
            for il = 1, ... , μ {
                r_h := b_h − J_h(ũ_h)u_h;
                u_h := u_h + ω diag(J_h(ũ_h))^{−1} r_h; }
        else
            goto ♯;
    }
    return;
♯   u_h := w_h;
    for in = 1, ... , ν {
        b_h := g_h + c(1 − u_h)e^{u_h}; ũ_h := u_h;
            for il = 1, ... , μ {
                r_h := b_h − J_h(ũ_h)u_h; s_h := J_h(ũ_h)r_h;
                α := (r_h, s_h)/(s_h, s_h);
                u_h := u_h + αr_h; }
    }
}
```

Here, ν is the number of the Newton (nonlinear) iterative steps and μ is the number of the linear iterative steps inside the nonlinear step. $J_h(u)$ is the linear mapping obtained from the discretization of $-\Delta - c \cdot e^u$ on grid G_h.

Remark: We have examined also red-black Gauss-Seidel relaxation and nonlinear red-black Gauss-Seidel relaxation (without global Newton linearization on each grid). We observed that red-black Gauss-Seidel relaxation provided more or less similar convergence to the Jacobi relaxation in the case of finding the second solutions for small c. With nonlinear Gauss-Seidel relaxation, it was very hard to obtain the second solutions: in most of cases, the iterates converged to the first solution. These phenomena are caused by the dramatic loss of diagonal dominance of the Jacobians on the coarse grids.

In the Bratu experiment, we choose 129×129 as the finest grid size and 5 as the number of the grid levels, so that the coarsest grid size is 9×9. The FAS W(2,2)-cycle is employed, where the numbers in the brackets correspond to the number of the nonlinear smoothing steps (ν) for pre- and post-smoothing. In each nonlinear smoothing step, only one linear iteration is performed ($\mu = 1$). On the coarsest grid, 10 nonlinear steps are performed. In case of computing the second solution for small c, there is no second solution on the 5^2 grid and many numerical second solutions exist on the coarse grids (finer than the 5^2 grid) due to the great loss of diagonal dominance, as mentioned before. Although the numerical solutions satisfy the discretized equations on the coarse grids, there are no analytic solutions corresponding to them. Therefore, we stop coarsening at 9×9 and perform only 10 iterations of the relaxation there, otherwise the solution on the coarsest grid might converge to one of the undesirable second solutions or to the first solution.

ETNA
Kent State University
etna@mcs.kent.edu

Before starting the acceleration process, one iteration of the FAS cycle is performed. Namely, the starting solution vector u_0 of the acceleration process is given from

$$u_0 := FAS(F, u_{init}) \ .$$

Here we test the following three different acceleration methods in order to see the influence of the selection strategy and the restarting strategy on the speed of convergence and on robustness:

Method M1 Criterion **A** is adopted for the selection of the solution. The restarting strategy is not adopted.

Method M2 Criteria **A** and **B** are adopted for the selection of the solution. The restarting strategy is not adopted.

Method M3 Criteria **A** and **B** are adopted for the selection of the solution. The acceleration process is restarted when Condition **C** or **D** is detected in two consecutive iterations.

Tables 3.1 and 3.2 depict the numbers of iterations needed for the convergence criterion $\|F(u_i)\| < 10^{-6}$ to be fulfilled. Here norm $\|\cdot\|$ is defined, so that the scaling is independent of the grid size (n is the number of the grid points):

$$\|r\| := \sqrt{\frac{\sum_{i,j} r_{i,j}^2}{n}} \ .$$

The initial approximation on the finest grid is given by

$$u_{init}(x,y) = u_c \cdot \min\left(\frac{x}{x_c}, \frac{1-x}{1-x_c}\right) \cdot \min\left(\frac{y}{y_c}, \frac{1-y}{1-y_c}\right) \ . \tag{3.4}$$

As the initial condition, $u_c = 12$ is chosen for obtaining the second solutions for $c = 0.2$ and $c = 0.1$. In Tables 3.1 and 3.2, robustness of the three strategies **M1**, **M2** and **M3** is tested by changing the position of the maximum value u_c of the initial solution (x_c, y_c) away from symmetry. We did not use FMG [1] for obtaining the initial solution on the finest grid, since we are interested here in the computation of the second solutions for small c and it is not trivial to prepare a good initial guess on the coarsest grid for these problems, as mentioned before. In Tables 3.1 and 3.2, the influence of the variation of parameter γ_A between 0.9 and 2 is evaluated as well. $m = 20$ is used as truncation number. In Table 3.1, the results for $c = 0.2$ are presented; the number of the iterations without acceleration are also described. For $c = 0.1$ (Table 3.2), we did not achieve convergence for these initial conditions, so the numbers without acceleration are not presented. In these tables, the CPU times in seconds for the execution on SGI Indigo-2 are also included in brackets.

As can be observed from Tables 3.1 and 3.2, $c = 0.1$ is much more difficult than $c = 0.2$. The difficulty and the difference in convergence speed for the acceleration methods is more pronounced as (x_c, y_c) moves away from the center. From the results in Tables 3.1 and 3.2, **M3** with $\gamma_A = 2$ is most preferable. For some difficult cases, taking $\gamma_A = 0.9$ requires many more iterations than $\gamma_A = 2$.

We also observe that restarting is harmless for all cases and it sometimes brings a great improvement, as in the case of $c = 0.1, x_c = 0.48, y_c = 0.5$, where **M3** is clearly preferable over **M2**. Let us observe this case more closely. In Figure 3.2, the convergence histories for Method **M2** and **M3** with $m = 5, 8, 10, 20$ are depicted ($\gamma_A = 2$). As can be observed from the convergence histories for **M2**, there is no guarantee that larger m brings better convergence. The results show the difficulty in choosing the best m. The convergence is very sensitive with respect to m in Method **M2**. We see that these difficulties are very nicely recovered by adopting the restarting strategy in Method **M3**. Even for small m, convergence

ETNA
Kent State University
etna@mcs.kent.edu

TABLE 3.1
The number of iterations until convergence for $u_c = 12,\ c = 0.2$. Here, "FAS" means no acceleration. The CPU times in second for the execution on SGI Indigo-2 are written in the brackets.

	x_c	0.50		0.48		0.46	
y_c	γ_A	0.9	2	0.9	2	0.9	2
0.50	FAS	91(41.8)		195(90.9)		194(89.0)	
	M1	16(9.7)	16(9.8)	31(20.1)	22(14.1)	29(19.0)	26(16.9)
	M2	16(9.8)	16(9.7)	31(20.2)	22(14.0)	29(18.8)	26(16.9)
	M3	16(9.8)	16(9.7)	31(19.9)	22(13.9)	27(15.1)	26(16.7)
0.48	FAS			197(90.0)		203(92.9)	
	M1			23(14.4)	23(14.7)	38(25.2)	57(38.2)
	M2			23(14.5)	23(14.7)	38(25.1)	39(25.8)
	M3			23(14.4)	23(14.6)	30(17.7)	39(25.8)
0.46	FAS					222(100.9)	
	M1					55(37.3)	69(46.7)
	M2					55(37.0)	47(31.8)
	M3					44(25.3)	41(22.9)

TABLE 3.2
The number of iterations until the convergence for $u_c = 12, c = 0.1$. The CPU times in second for the execution on SGI Indigo-2 are written in the brackets.

	x_c	0.50		0.49		0.48	
y_c	γ_A	0.9	2	0.9	2	0.9	2
0.50	M1	40(29.5)	42(30.6)	35(26.3)	36(27.1)	90(70.5)	78(60.3)
	M2	40(29.5)	23(16.5)	35(26.3)	38(28.3)	90(70.6)	67(51.9)
	M3	34(22.1)	27(19.3)	35(26.3)	39(27.3)	57(39.2)	28(19.2)
0.49	M1			65(50.3)	54(41.6)	53(40.9)	50(38.3)
	M2			65(50.3)	50(38.4)	53(40.9)	53(40.9)
	M3			60(43.2)	41(27.5)	50(34.2)	46(31.9)
0.48	M1					92(71.9)	49(37.5)
	M2					92(71.9)	53(40.8)
	M3					110(73.2)	60(41.4)

is improved, and it is no longer risky to take m as large as possible in order to achieve the best convergence.

3.2. The rotating convection-diffusion equation. Next we consider a linear rotating convection-diffusion problem, tested for example in [9] and the references therein, with Dirichlet boundary conditions:

$$\begin{aligned} -\epsilon\Delta u + (a(x,y)u)_x + (b(x,y)u)_y = 1 \ \text{ on } \Omega = (0,1)^2 \ , \\ u = f(x,y) \ \text{ on } \partial\Omega \ , \end{aligned} \tag{3.5}$$

where

$$\begin{aligned} a(x,y) &= -\sin(\pi x)\cos(\pi y) \ , \\ b(x,y) &= \sin(\pi y)\cos(\pi x) \ , \\ f(x,y) &= \sin(\pi x) + \sin(13\pi x) + \sin(\pi y) + \sin(13\pi y) \ . \end{aligned}$$

The nonlinearity in this problem arises from the discretization of the convection terms. A second order finite volume TVD scheme ([7]) with limiter is used for this purpose. For

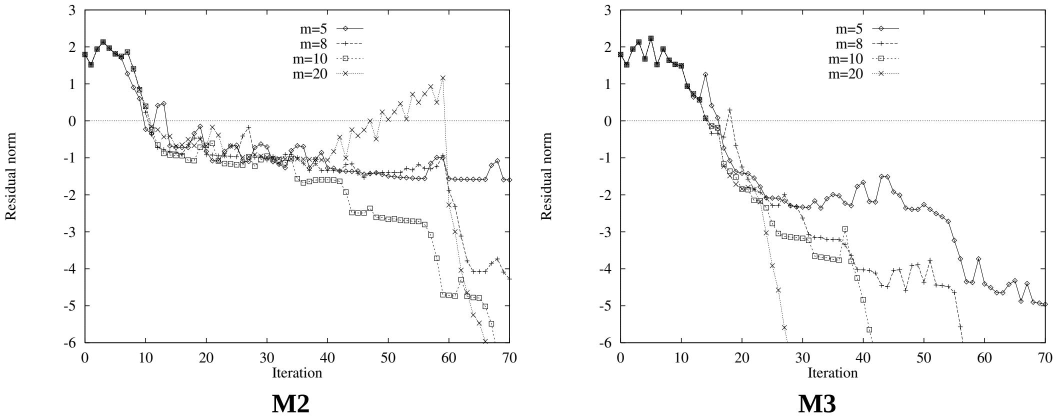

FIG. 3.2. *The convergence histories for the several truncations ($c = 0.1, x_c = 0.48, y_c = 0.5$).*

example, for $(au)_{i+1/2,j}$ when $a(x,y) > 0$, we obtain:

$$(au)|_{i+1/2,j} = \frac{1}{2}(a_{i,j} + a_{i+1,j})[(u_{i,j} + \frac{1}{2}\Psi(R_{i+1/2})(u_{i,j} - u_{i-1,j})] \ . \tag{3.6}$$

Here, with $R_{i+1/2} \equiv (u_{i+1,j} - u_{i,j})/(u_{i,j} - u_{i-1,j})$, nonlinearity enters into the discretization. The function $\Psi(R)$ is the Van Albada limiter:

$$\Psi(R) = \frac{R^2 + R}{R^2 + 1} \ . \tag{3.7}$$

This limiter is often used in CFD calculations, as for example in [8]. Similar formulae are found for $(au)|_{i-1/2,j}$, for $(au)_y$ and for discretizations when $a(x,y) < 0$. A convection dominated test case is investigated: $\epsilon = 10^{-5}$. The rotating convection-diffusion problem with dominant convection is a difficult test for standard multigrid, because of different scaling of convection ($a(x,y)/h,\ b(x,y)/h$) and diffusion (ϵ/h^2), which is not dealt with properly on coarse grids. Characteristic components, which are constant along the characteristics of the advection operator, are not correctly approximated on coarse grids. In channel flow problems, convergence difficulties do not occur, since line smoothers on the fine grids are also taking care of the problematic low frequency error components. This is not true for rotating convection-dominant flow problems. In [2], an optimal multigrid solution method, especially for the first order upwind discretization of this problem, was constructed by means of overweighting of residuals and/or by the use of a point smoother in the flow direction. In [9], standard linear multigrid was used as a preconditioner for GMRES for the first order *linear* discretization of this problem. Here, we want to investigate whether the nonlinear Krylov acceleration results in a similar satisfactory convergence improvement as was found for the standard upwind discretization of (3.5) in [9]. The second order nonlinear discretization is solved directly within multigrid. Multigrid FAS F(2,1)-cycles are used.

A symmetric alternating line smoother is adopted, which is based on a splitting into a first order upwind part and a remaining part, and is presented in [10]. With smoothers based on this splitting, fast convergence is obtained for many convection dominated problems. This smoother is explained here, since it is also the basis for the next examples. All other multigrid components are standard components. Second order discretizations of (3.5) with (3.6) have the general form:

$$L_2 u = \sum_{\mu\in J}\sum_{\nu\in J} a^{(2)}_{\mu\nu} u_{i+\mu,j+\nu} \tag{3.8}$$

with coefficients $a^{(2)}_{\mu\nu}$ and a set of indices $J = \{-2,-1,0,1,2\}$. A part of the symmetric alternating line smoother is the x-line smoother for a forward ordering of lines, which we explain in more detail. It is constructed with a special splitting of L_2 into L^0, L^+ and L^-. First we explain the superscripts: 0 indicates operator parts corresponding to grid points currently treated, $+$ means already updated and $-$ means still to be updated , as in [13]. In case of a forward x-line smoother, 0 represents the line $j = j_c =$ const. where the unknowns are currently updated, $+$ indicates the lines $j < j_c$, which were already updated, and $-$ are the lines $j > j_c$ still to be updated. However, the L^0 parts of operator L_2 are not just the operator elements of the grid points under consideration. For L^0 operator elements from the first order operator are chosen. L^0 and L^+ look (with the stencil notation) like:

$$L^0 := \begin{pmatrix} & & 0 & & \\ & & 0 & & \\ 0 & a^{(1)}_{-10} & a^{(1)}_{00} & a^{(1)}_{10} & 0 \\ & & 0 & & \\ & & 0 & & \end{pmatrix}, \quad L^+ := \begin{pmatrix} & & 0 & & \\ & & 0 & & \\ 0 & 0 & 0 & 0 & 0 \\ & & a^{(2)}_{0-1} & & \\ & & a^{(2)}_{0-2} & & \end{pmatrix}. \tag{3.9}$$

Here $a^{(1)}_{00}$ etc. denote operator elements from a first order accurate upwind discretization. With the definitions of L^0, L^+ and L^- given we define the following splitting for obtaining u^* for the grid points $j = j_c$ under consideration:

$$L^0 u_* = f + L^0 u_n - (\,(L_2 - L^+)u_n + L^+ u_{n+1}\,)\ . \tag{3.10}$$

With the notation as explained above, it is possible that iteration indices n (i in the previous chapters, changed to n in order to avoid confusion) and $n+1$ appear in a right-hand side, since all neighboring grid points appear in the right-hand side. Inserting an underrelaxation parameter ω in (3.10) leads to:

$$u_{n+1} = \omega u_* + (1-\omega)u_n \tag{3.11}$$

for $j = j_c$, after which the next line of points $j = j_c + 1$ is relaxed. We use $\omega = 0.9$.

The test with rotating convection is performed on a 129^2 grid. For the Krylov acceleration, we take the best method from the previous experiment, Method **M3**. We choose $\gamma_A = 2$, and study the effect of restart parameter m, which is set to 2, 5 and 10. Of course, the smallest m that results in good convergence is the most interesting one with respect to the storage demands of the acceleration method. Figure 3.3 compares FAS convergence to FAS + Krylov convergence. It can be seen that $m = 2$ already gives very satisfactory results. For this problem, the dependence on m is relatively small. The average convergence factor for FAS is found to be 0.81, whereas the FAS+Krylov convergence with $m = 2$ is 0.66. With Krylov acceleration, the second order residual is reduced with additionally 4 to 5 orders of magnitude after 50 iterations compared to the FAS convergence.

We already stated that the costs of the Krylov acceleration is negligible. Of course, it is necessary to compare the costs in order to make the comparison. For the problem presented above, 25 FAS F(2,1)-cycles cost 295 seconds on a common workstation, while 25 FAS+Krylov cycles with $m = 10$ take 319 seconds, 314 seconds with $m = 5$ and 312 seconds with $m = 2$. The cost of Krylov acceleration is less than 10 percent of the total CPU time.

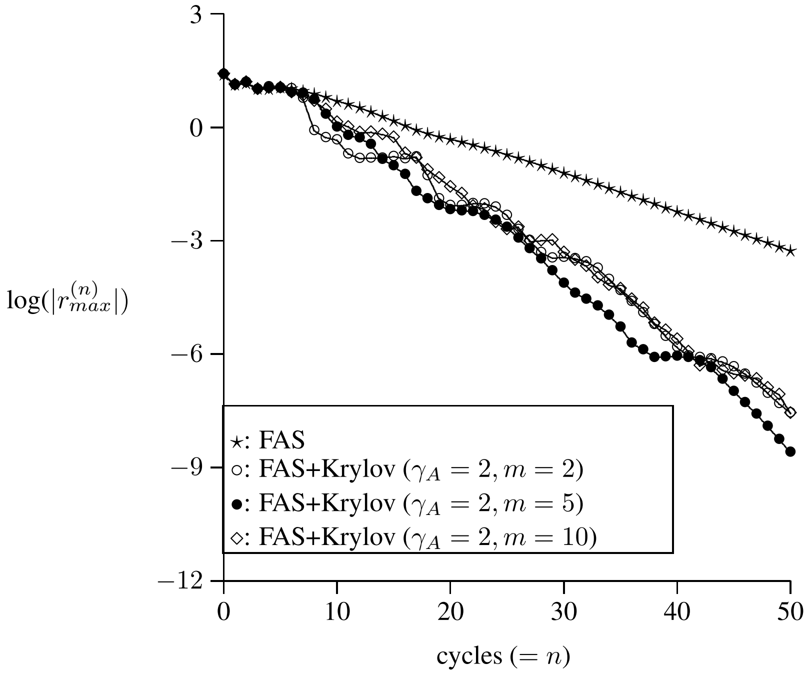

FIG. 3.3. *Convergence of FAS and FAS + Krylov ($\gamma_A = 2$) for the rotating convection-diffusion problem, $\epsilon = 10^{-5}$ with F(2,1) cycles on a 129^2 grid. Symmetric alternating line-smoother $\omega = 0.9$.*

3.3. The driven cavity problem at $Re = 10000$. Next, an incompressible flow example is treated. The 2D steady incompressible Navier-Stokes equations are written as a system of equations as follows:

$$\frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \frac{\partial \mathbf{f_v}}{\partial x} + \frac{\partial \mathbf{g_v}}{\partial y} \ , \tag{3.12}$$

where $\mathbf{f}$ and $\mathbf{g}$ are the components of the convective flux vector, and $\mathbf{f_v}$ and $\mathbf{g_v}$ are the viscous fluxes:

$$\mathbf{f} = \begin{bmatrix} u^2+p \\ uv \\ c^2u \end{bmatrix}. \ \mathbf{g} = \begin{bmatrix} uv \\ v^2+p \\ c^2v \end{bmatrix}, \ \mathbf{f_v} = \begin{bmatrix} \frac{1}{Re}\partial u/\partial x \\ \frac{1}{Re}\partial v/\partial x \\ 0 \end{bmatrix}, \ \mathbf{g_v} = \begin{bmatrix} \frac{1}{Re}\partial u/\partial y \\ \frac{1}{Re}\partial v/\partial y \\ 0 \end{bmatrix}.$$

Here, u and v are Cartesian velocity unknowns, p is pressure, c is a constant reference velocity and Re is the Reynolds number defined as $Re = \overline{U} \cdot L/\nu$, with $\overline{U}$ a characteristic velocity, L a characteristic length and ν the kinematic viscosity. A 2D vertex-centered discretization (on a nonstaggered collocated grid) of (3.12) is used with Dick's flux difference splitting, presented in [3]. Differences of the convective fluxes with respect to $\mathbf{u}$ can be written as

$$\Delta\mathbf{f} = A_1\Delta\mathbf{u} \ , \quad \Delta\mathbf{g} = A_2\Delta\mathbf{u} \ , \tag{3.13}$$

with $\mathbf{u} = (u, v, p)^T$ and A_1, A_2 being discrete Jacobians.

Integration of the convective part of (3.12) over a control volume $\Omega_{i,j}$ gives,

$$\int_{\Omega_{i,j}} (\frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y}) \partial\Omega = F \cdot dS|^{i+1/2,j}_{i-1/2,j} + F \cdot dS|^{i,j+1/2}_{i,j-1/2} \ , \tag{3.14}$$

ETNA
Kent State University
etna@mcs.kent.edu

where $F = \mathbf{n} \cdot \mathbf{F}$ with $\mathbf{F} = (\mathbf{f}, \mathbf{g})^T$ and $\mathbf{n} = (n_x, n_y)^T$ is the outward normal vector on the volume side and dS the length of the volume side. The formula used in the discretization, explained in detail in [3], is obtained with a second order accurate approximation. For example, for $F_{i+1/2,j}$ we have

$$F_{i+1/2,j} = \frac{1}{2}(F(\mathbf{u}_L) + F(\mathbf{u}_R) - |A(\mathbf{u}_L, \mathbf{u}_R)|(\mathbf{u}_R - \mathbf{u}_L)) \ , \tag{3.15}$$

where

$$A = n_x A_1 + n_y A_2 = \begin{bmatrix} n_x\overline{u} + r & n_y\overline{u} & n_x \\ n_x\overline{v} & n_y\overline{v} + r & n_y \\ n_x c^2 & n_y c^2 & 0 \end{bmatrix} \ , \tag{3.16}$$

with $r = n_x\overline{u} + n_y\overline{v}$.

Second order accuracy results from the determination of the left and right state vectors $\mathbf{u}_L$ and $\mathbf{u}_R$, which are obtained with the Fromm scheme. In (3.15), these state vectors are, for example,

$$\begin{aligned} \mathbf{u}_L &\leftarrow \mathbf{u}_{i,j} + \frac{1}{4}(\mathbf{u}_{i-1,j} - \mathbf{u}_{i,j}) + \frac{1}{4}(\mathbf{u}_{i,j} - \mathbf{u}_{i+1,j}) \ , \\ \mathbf{u}_R &\leftarrow \mathbf{u}_{i+1,j} + \frac{1}{4}(\mathbf{u}_{i,j} - \mathbf{u}_{i+1,j}) + \frac{1}{4}(\mathbf{u}_{i+1,j} - \mathbf{u}_{i+2,j}) \ . \end{aligned} \tag{3.17}$$

The fluxes on the other volume boundaries in (3.14) are treated in the same way. The viscous fluxes $\mathbf{f_v}, \ \mathbf{g_v}$ are discretized with the Peyret control volume technique. For incompressible Navier-Stokes equations, it is not necessary to implement a limiter. For many (2D and 3D) different problems at low and high Reynolds numbers, oscillations did not appear (for example, in the pressure distribution, as they occur near discontinuities for compressible flow problems).

A well-known difficult 2D test case is the lid-driven cavity flow in a unit square at $Re = 10000$. We solve this problem on a 193^2 grid with stretching. The streamlines resulting for this problem (with top wall moving with $u = 1$) are shown in Figure 3.4. With the 193^2 stretched grid, the centerline velocity profiles (not shown here) agree very well with reference results.

The multigrid FAS scheme used for solving this problem is the same as for the rotating convection-diffusion problem. The smoother is now a coupled collective symmetric alternating line smoother, based on splitting (3.10), with $\omega = 0.4$. W(2,1)-cycles are used. For the Krylov acceleration again we adopt method **M3** with $\gamma_A = 2$. We study the effect of the size of parameter m, which varies between 2 and 10. The results are presented in Figure 3.5, from which it can be seen that the FAS convergence is problematic, at least for the first 30 iterations. The multigrid convergence problems for this driven cavity example at $Re = 10000$ are not surprising from the considerations on the rotating flows in Subsection 3.2. The accelerated FAS cycles, however, show a regular convergence almost independent of parameter m. With larger m, slightly better convergence is obtained. We find with $m = 2$ an average convergence of 0.71, with $m = 5$ it is 0.66 and with $m = 10$ it is 0.63. Similar convergence was found with $\gamma_A = 0.9$. Also here, the FAS convergence is accelerated very satisfactorily even with small values of m. We would like to mention that the classical multigrid approach for solving this problem is by means of defect-correction [3], [8]. Without Krylov acceleration, defect-correction converges and reduces the second order residual after 100 cycles to 10^{-3}. Experimentally, it is observed that it is not possible to improve the convergence of defect-correction with this Krylov acceleration.

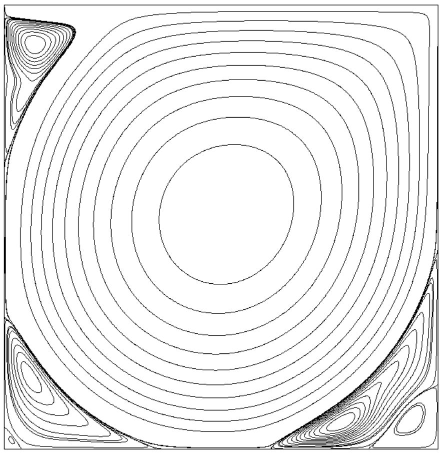

FIG. 3.4. *Streamlines for the lid driven cavity flow problem at $Re = 10000$.*

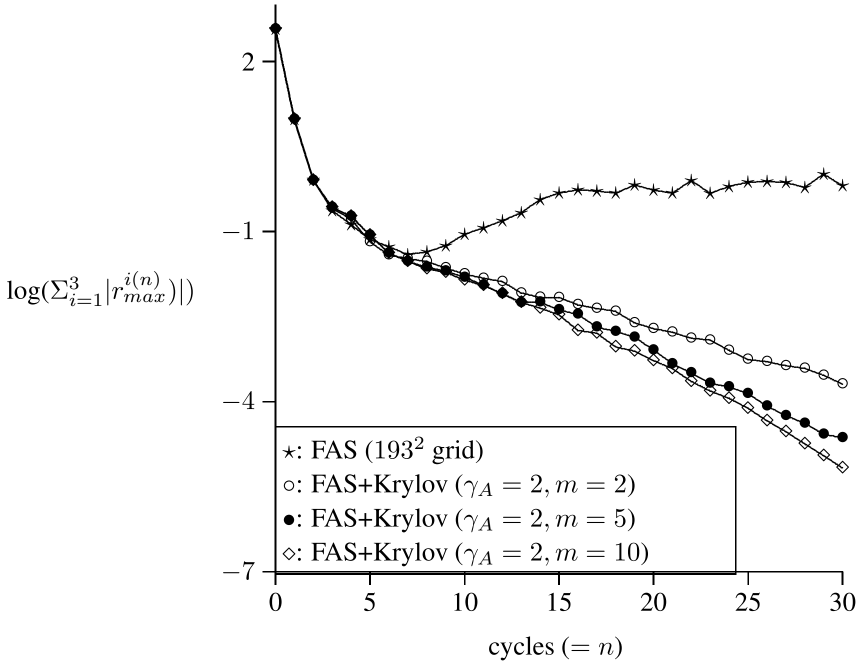

FIG. 3.5. *Convergence of FAS and FAS + Krylov ($\gamma_A = 2$) for the driven cavity problem, $Re = 10000$ with W(2,1) cycles on a 193^2 stretched grid. Symmetric alternating line smoother.*

3.4. An Euler channel flow problem. A last example is compressible Euler flow in a channel with a bump. The 2D steady compressible Euler equations are written in their

ETNA
Kent State University
etna@mcs.kent.edu

differential form as follows:

$$\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (E+p)u\end{bmatrix}+\frac{\partial}{\partial y}\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (E+p)v\end{bmatrix}=0\ ,$$

$$p=(\gamma-1)(E-\frac{1}{2}\rho(u^2+v^2))\ , \tag{3.18}$$

where ρ is density, u and v are Cartesian velocity components, E is total energy, p is pressure, and γ (assumed to be constant) is the ratio of the specific heats at constant pressure and constant volume.

The vertex-centered finite volume discretization adopted for the Euler equations is described briefly. It is based on the cell-centered discretization as in [12], [8]. For the finite volume discretization the domain Ω is divided into control volumes $\Omega_{i,j}$. For each quadrilateral, equation (3.18) must hold in integral form:

$$\oint_{\partial\Omega_{i,j}}(f(\mathbf{u})n_x+g(\mathbf{u})n_y)dS=0\ , \tag{3.19}$$

where $(n_x,n_y)^T=(cos\phi,sin\phi)^T$ is the outward normal vector on $\partial\Omega_{i,j}$, and $\mathbf{u}$ is the state vector.

The rotational invariance of the Euler equations is used, and the discretization results in

$$\sum_{(k,l)\in N_{i,j}}F(\mathbf{u}^L,\mathbf{u}^R)\partial S_{k,l}=0\ , \tag{3.20}$$

where $N_{i,j}$ indexes the set of cells neighboring $\Omega_{i,j}$, $\partial S_{k,l}$ is the length of the boundary between $\Omega_{i,j}$ and $\Omega_{k,l}$ and $F(\mathbf{u}^L,\mathbf{u}^R)$ is an approximate Riemann solver, which depends on $\mathbf{u}^L$ and $\mathbf{u}^R$, the left and right states, along the cell boundary. The approximate solution $F(\mathbf{u}^L,\mathbf{u}^R)$ of the 1D Riemann problem is solved with an approximate Riemann solver proposed by Osher in its P-variant form ([12]):

$$F(\mathbf{u}^L,\mathbf{u}^R)=\frac{1}{2}(\tilde{f}(\mathbf{u}^L)+\tilde{f}(\mathbf{u}^R)-\int_{\mathbf{u}^L}^{\mathbf{u}^R}|A(\mathbf{u})|d\mathbf{u})\ , \tag{3.21}$$

where $|A(\mathbf{u})|(=A^+(\mathbf{u})-A^-(\mathbf{u}))$ is a splitting of the Jacobian matrix into matrices with positive and negative eigenvalues, and $\tilde{f}$ represents the transport of mass, momentum and energy across $\partial S_{k,l}$ along the normal vector. The states $\mathbf{u}^L$ and $\mathbf{u}^R$ on $\partial S_{k,l}$ in (3.21) are approximated by the Fromm scheme (3.17). State vector $\mathbf{u}=(u,v,c,z)^T$ is chosen, where $c\equiv\sqrt{\gamma p/\rho}$ is the speed of sound and $z\equiv ln(p\rho^{-\gamma})$ is an unscaled entropy. In order to avoid wiggles that may appear with this scheme near discontinuities, the van Albada limiter as in (3.6) is implemented here. For further details on the discretization adopted here, see [12].

A transonic channel problem is evaluated at Mach 0.85. The bump in the channel is a 4.2% circular bump, the height of the channel is 2.1, its length is 5 and the bump length is 1. The domain is discretized with 97×65 cells, which results in a multigrid method with 5 levels. With the smoother as explained in (3.10), the second order discretization for the Euler equations is solved directly by multigrid. The underrelaxation parameter is $\omega=0.7$. For this test, where a shock appears in the solution, we use the second order smoother only on the finest grid; on the coarse grid, we discretize and smooth with the first order accurate upwind discretization. A V(1,0)-cycle is used for the transonic test. FAS convergence is improved with more smoothing iterations, but a smoothing iteration is relatively expensive. Here, we test if we can improve multigrid convergence with only one smoothing iteration by the computationally cheaper Krylov acceleration. For Krylov acceleration, again we adopt Method **M3** with $\gamma_A = 2$ and study the effect of parameter m, which varies between 2 and 10. The results are presented in Figure 3.6. Observe that FAS V(1,0)-convergence is already

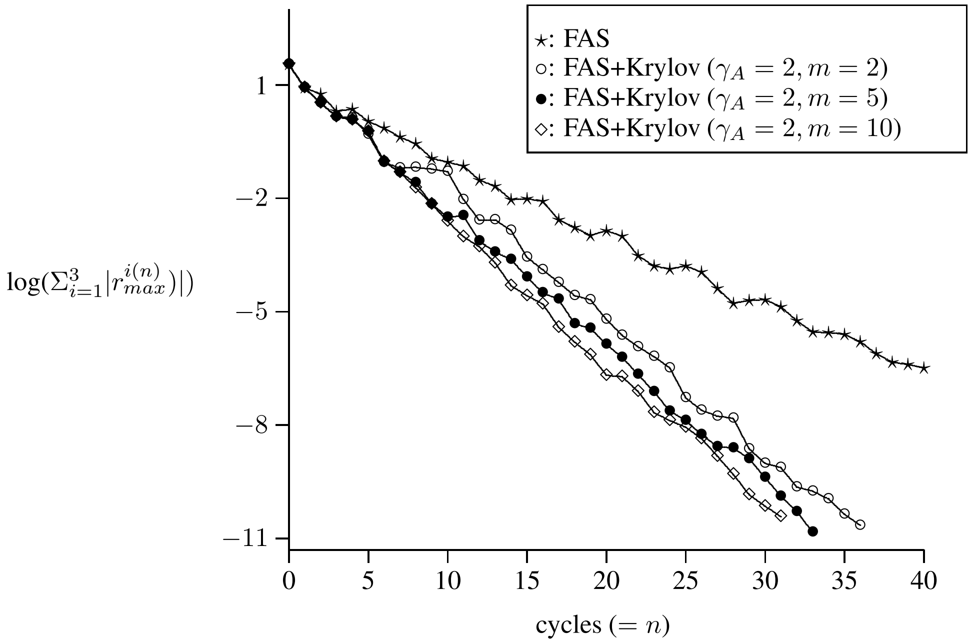

FIG. 3.6. *Convergence of FAS and FAS + Krylov ($\gamma_A = 2$) for the Euler channel flow problem, Mach=* 0.85 *with V(1,0) cycles on a* 97×65 *grid. Symmetric alternating line smoother* $\omega = 0.7$.

satisfactory. However, acceleration improves the residual by 5 orders of magnitude after 40 iterations for the transonic test. Larger values of m further improve the convergence. The solution for the transonic test with the domain parameter given is shown in Figure 3.7.

4. Conclusions. In this paper, we have presented a nonlinear Krylov acceleration strategy for nonlinear equations. The strategy is similar to GMRES for linear equations. It is possible to use the acceleration strategy in combination with a nonlinear preconditioner, such as a nonlinear multigrid method. The strategy is cheap, relative to a nonlinear multigrid preconditioner, since it uses intermediate solutions and residuals that are already calculated in the multigrid iteration. Jacobians need not be re-evaluated explicitly. Based on proper selection criteria, it is then decided whether to adopt the accelerated intermediate solution or the solution obtained from the preconditioner. This is necessary, since the nonlinear Krylov acceleration is based on a linear approximation of Jacobians, which might not be accurate if intermediate solutions are far from the discrete solution. Furthermore, the acceleration will then never disturb satisfactory convergence of a nonlinear preconditioner for 'easy' problems. Criteria for restarting are also prescribed and evaluated. A significant convergence improvement has been presented for several nonlinear problems, which are known to be difficult problems for nonlinear multigrid: scalar equations, like the Bratu problem and a rotating convection-diffusion problem, and systems of incompressible Navier-Stokes and compressible Euler equations. By means of the nonlinear Krylov acceleration, the solution method is made much more robust: larger ranges of problem parameters can be efficiently treated.

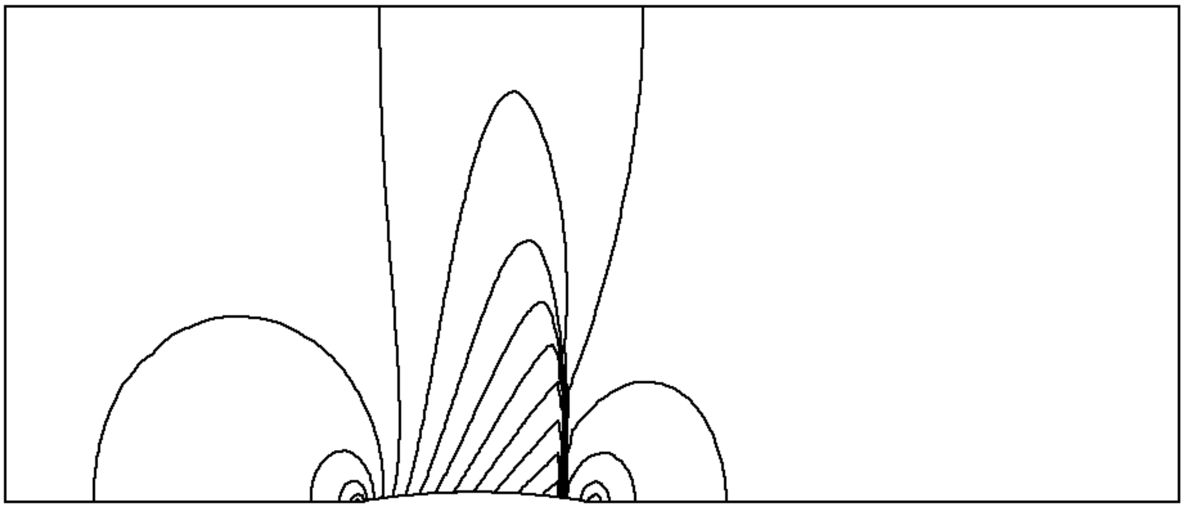

FIG. 3.7. *Isobars for the transonic Euler channel flow (Mach 0.85) on a* 97×65*-grid.*

Acknowledgment. The reviewer's valuable comments are gratefully acknowledged.

REFERENCES

[1] A. BRANDT, *Multi–level adaptive solutions to boundary–value problems*, Math. Comp., 31(1997), pp. 333–390.

[2] A. BRANDT AND I. YAVNEH, Accelerated multigrid convergence and high-Reynolds recirculating flows, SIAM J. Sci. Comp., 14(1993), pp. 607–626.

[3] E. DICK AND J. LINDEN, *A multigrid method for steady incompressible Navier-Stokes equations based on flux difference splitting*, Int. J. Num. Methods in Fluids, 14(1992), pp. 1311–1323.

[4] R. GLOWINSKI, H. B. KELLER AND L. REINHART, *Continuation-conjugate gradient methods for the least squares solution of nonlinear boundary value problems*, SIAM J. Sci. Stat. Comp., 6(1985), pp. 793–832.

[5] W. HACKBUSCH, *Multi-grid methods and applications.* Springer, Berlin, Germany (1985).

[6] ———, *Comparison of different multigrid variants for non-linear equations*, Z. angew. Math. Mech., 72(1992), pp. 148–151.

[7] C. HIRSCH, *Numerical computation of internal and external flows,* Vol. 2. John Wiley, Chichester (1990).

[8] B. KOREN, *Defect correction and multigrid for an efficient and accurate computation of airfoil flows*, J. Comp. Phys., 77(1988), pp. 183–206.

[9] C.W. OOSTERLEE AND T. WASHIO, *An evaluation of parallel multigrid as a solver and a preconditioner for singular perturbed problems*, SIAM J. Sci. Comp., 19(1998), pp. 87–110.

[10] C.W. OOSTERLEE, F.J. GASPAR, T. WASHIO AND R. WIENANDS, *New multigrid smoothers for higher order upwind discretizations of convection-dominated problems*, GMD Arbeitspapier 1049, GMD St. Augustin, Germany (1997). To appear in J. Comp. Phys.

[11] Y. SAAD AND M.H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Comp., 7(1986), pp. 856–869.

[12] S.P. SPEKREIJSE, *Multigrid solution of the steady Euler equations*, CWI Tract 46, CWI Amsterdam, 1988.

[13] 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. and Trottenberg, eds., Springer, Berlin, Germany, 1982, pp. 1–176.