SIAM J. SCI. COMPUT.
Vol. 14, No. 2, pp. 461–469, March 1993

A FLEXIBLE INNER-OUTER PRECONDITIONED GMRES ALGORITHM*

YOUCEF SAAD†

Abstract. A variant of the GMRES algorithm is presented that allows changes in the preconditioning at every step. There are many possible applications of the new algorithm, some of which are briefly discussed. In particular, a result of the flexibility of the new variant is that any iterative method can be used as a preconditioner. For example, the standard GMRES algorithm itself can be used as a preconditioner, as can CGNR (or CGNE), the conjugate gradient method applied to the normal equations. However, the more appealing utilization of the method is in conjunction with relaxation techniques, possibly multilevel techniques. The possibility of changing preconditioners may be exploited to develop efficient iterative methods and to enhance robustness. A few numerical experiments are reported to illustrate this fact.

Key words. Krylov subspace methods, GMRES, non-Hermitian systems, preconditioned conjugate gradient, variable preconditioners

AMS(MOS) subject classification. 65F10

1. Introduction.

Krylov subspace techniques have increasingly been viewed as general-purpose iterative methods, especially since the popularization of preconditioning techniques [2] in the mid-seventies. Although iterative methods lack the robustness of direct methods, they are effective for the large class of problems arising from partial differential equations of the elliptic type. An important gap in the literature concerns the development of truly general-purpose iterative solvers that could replace direct methods with a minimum risk of failure. A comparison between existing software based on direct methods and software based on iterative methods reveals that the direct solvers have evolved quite differently and have acquired a level of sophistication that far exceeds that of iterative methods.

In order to be able to enhance robustness of iterative solvers, we should be able to determine, e.g., by means of heuristics, whether or not a given preconditioner is suitable for the problem at hand. If not, one can attempt another possible iterative method/preconditioner and switch periodically if necessary. It is desirable to be able to switch within the outer iteration instead of restarting. For the GMRES algorithm [5], this can be accomplished with the help of a rather simple modification of the standard algorithm, referred to as the flexible GMRES (FGMRES), which is presented in this paper. An important property of FGMRES is that it satisfies the residual norm minimization property over the preconditioned Krylov subspace just as in the standard GMRES algorithm [5].

For motivation, we mention that there are cases in which relaxation-type preconditioners are more attractive than the usual ILU preconditioners. These include the case in which a red-black ordering is used. In this situation, the single-step SSOR preconditioner and the ILU preconditioning may perform very poorly. However, our experience is that if a higher level-of-fill ILU or a multiple-step SSOR (or SOR) preconditioner is used, then the preconditioned method can perform rather well [4]. In this situation, SOR and SSOR have a distinct advantage over the ILU-type preconditioners, in that they preserve their high degree of parallelism, which is of order N.

* Received by the editors November 6, 1991; accepted for publication (in revised form) April 3, 1992.

† Computer Science Department and Minnesota Supercomputer Institute, University of Minnesota, 200 Union Street S.E., Minneapolis, Minnesota 55455.

In contrast, a serious loss of parallelism is incurred for the incomplete factorization preconditioners with high level-of-fill. A few other advantages are discussed in [4]. Thus we wish to be able to apply an arbitrary number of SOR or SSOR steps in the preconditioning phase, for example, in order to solve the preconditioning system $My = v$ to a given tolerance. We may also wish to change the relaxation parameter ω, possibly at each GMRES step, in order to attempt to achieve optimality.

The FGMRES algorithm presented in this paper allows us to incorporate these changes in the preconditioner into the GMRES framework at little additional cost. To be precise, there is no additional cost incurred in the arithmetic but the memory requirement doubles. On the other hand, FGMRES may enable one to utilize the memory more efficiently since the vectors that are normally not being used in a given FGMRES step can be fully exploited to compute a preconditioned vector, e.g., via another GMRES run that uses these vectors. A few tests based on this approach will be presented in §3. We will present some applications of the technique to show how the method can be used to improve the robustness of the standard GMRES algorithm. We should point out that another illustration of the benefits of FGMRES in the finite element framework is described by Tezduyar, Behr, Abadi, and Ray [6].

2. Krylov subspaces with variable preconditioning. The basic principle of preconditionings is to use a Krylov subspace method for solving a modified system such as

$$AM^{-1}(Mx) = b.$$

Clearly, the matrix AM^{-1} need not be formed explicitly: we only need to solve $Mz = v$ whenever such an operation is required. Thus a fundamental requirement is that it should be easy to compute $M^{-1}v$ for an arbitrary vector v. In some cases, solving a linear system with the matrix M consists of forming an approximate solution by performing one or a few steps of a relaxation-type method, or a Chebyshev iteration. It is natural to consider preconditioners that do not use only a single step of an iterative method, but as many as are needed to solve a linear system within a given tolerance. In fact, this would be the equivalent of a higher level-of-fill in the usual ILU preconditioners, except that the preconditioner is no longer constant but is allowed to vary from one step to another in the outer iteration. A similar situation in which the preconditioner is "not constant" is when another Krylov subspace method, e.g., one that is based on the normal equations approach, is used as a preconditioner. These applications and others lead us to raise the question of whether or not it is possible to accommodate such *variations in the preconditioners* and still obtain an algorithm that satisfies an optimality property similar to the one satisfied by the original iterative method. This question has been avoided in the past because in the Hermitian case there does not seem to exist a version of the usual preconditioned conjugate gradient algorithm that satisfies a short vector recurrence and that allows the preconditioner to vary at each step. In the non-Hermitian case and for methods that do not rely on short vector recurrences, such as GMRES, variations in the preconditioner can be handled without difficulty, as we now show.

2.1. The algorithm. We start by describing the standard GMRES algorithm with right preconditioning and then show the flexible modification which allows such variations.

ALGORITHM 2.1. GMRES with right preconditioning.

```
1. Start: Choose $x_0$ and a dimension $m$ of the Krylov subspaces. Define an
   $(m+1) \times m$ matrix $\bar{H}_m$ and initialize all its entries $h_{i,j}$ to zero.
2. Arnoldi process:
   (a) Compute $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$, and $v_1 = r_0/\beta$.
   (b) For $j = 1, \ldots, m$ do
       • Compute $z_j := M^{-1}v_j$;
       • Compute $w := Az_j$;
       • For $i = 1, \ldots, j$, do  $\begin{cases} h_{i,j} := (w, v_i), \\ w := w - h_{i,j}v_i; \end{cases}$
       • Compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$.
   (c) Define $V_m := [v_1, \ldots, v_m]$.
3. Form the approximate solution: Compute $x_m = x_0 + M^{-1}V_m y_m$ where
   $y_m = \text{argmin}_y \|\beta e_1 - \bar{H}_m y\|_2$ and $e_1 = [1, 0, \ldots, 0]^T$.
4. Restart: If satisfied stop, else set $x_0 \leftarrow x_m$ and goto 2.
```

The Arnoldi loop simply constructs an orthogonal basis of the preconditioned Krylov subspace

$$\text{Span}\{r_0, AM^{-1}r_0, \ldots, (AM^{-1})^{m-1}r_0\}$$

by a modified Gram–Schmidt process, in which the new vector to be orthogonalized is obtained from the previous vector process. The last step in the above algorithm forms the solution as a linear combination of the preconditioned vectors $z_i = M^{-1}v_i, i = 1, \ldots, m$. Because these vectors are all obtained by applying the same preconditioning matrix M^{-1} to the v's, we need not save them. We only need to apply M^{-1} to the linear combination of the $v's$, i.e., to $V_m y_m$. The question we can now ask is: what if we allowed the preconditioner to change at every step, i.e., z_j would be defined by

$$z_j = M_j^{-1}v_j,$$

but we saved these vectors to use them in updating x_m in step 3? In other words, we would like to consider the following "flexible" modification to the previous algorithm.

ALGORITHM 2.2. FGMRES: GMRES with variable preconditioning.

```
1. Start: Choose $x_0$ and a dimension $m$ of the Krylov subspaces. Define an
   $(m+1) \times m$ matrix $\bar{H}_m$ and initialize all its entries $h_{i,j}$ to zero.
2. Arnoldi process:
   (a) Compute $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$, and $v_1 = r_0/\beta$.
   (b) For $j = 1, \ldots, m$, do
       • Compute $z_j := M_j^{-1}v_j$;
       • Compute $w := Az_j$;
       • For $i = 1, \ldots, j$, do  $\begin{cases} h_{i,j} := (w, v_i), \\ w := w - h_{i,j}v_i; \end{cases}$
       • Compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$.
   (c) Define $Z_m := [z_1, \ldots, z_m]$.
3. Form the approximate solution: Compute $x_m = x_0 + Z_m y_m$ where $y_m =$
   $\text{argmin}_y \|\beta e_1 - \bar{H}_m y\|_2$ and $e_1 = [1, 0, \ldots, 0]^T$.
4. Restart: If satisfied stop, else set $x_0 \leftarrow x_m$ and goto 2.
```

As can be observed, the only difference from the standard version is that we now save the preconditioned vectors z_i and update the solution using these vectors. It is clear that when $M_j = M$ for $j = 1, \ldots, m$, then the new method is mathematically equivalent to Algorithm 2.2. Note that we can define z_j in step 2(b) without reference to any preconditioner, i.e., we can simply pick a given new vector z_j. We would like to mention that the technique presented above can be viewed as an extension of a strategy presented in [1] in the context of using Krylov subspace methods for solving nonlinear equations. More recently, van der Vorst and Vuik developed a family of algorithms that have the same feature as FGMRES in that they also allow variations in the preconditioner [7].

2.2. Some basic properties. One notable difference between FGMRES and the usual GMRES algorithm is that the action of AM_j^{-1} on a vector v of the Krylov subspace is no longer in the span of V_{m+1}. Instead, it is easy to show that the following equality takes place:

$$AZ_m = V_{m+1}\bar{H}_m \ . \tag{1}$$

This replaces the simpler relation

$$(AM^{-1})V_m = V_{m+1}\ \bar{H}_m,$$

which holds for the standard preconditioned GMRES [5]. Following [5] we will denote by H_m the $m \times m$ matrix obtained from $\bar{H}_m$ by deleting its last row and by $\hat{v}_{j+1}$ the vector w obtained at the end of step 2(b) of the algorithm, i.e., the vector obtained before normalizing w to get v_{j+1}. Then an alternative to (1) that is valid even when $h_{m+1,m} = 0$ is the following:

$$AZ_m = V_m H_m + \hat{v}_{m+1} e_m^T \ . \tag{2}$$

We will now prove an optimality property similar to the one that defines GMRES. Consider the residual vector for an arbitrary vector $z = x_0 + Z_m y$ in the affine space $x_0 + \text{span}\{Z_m\}$. We have

$$\begin{aligned} b - Az &= b - A(x_0 + Z_m y) \\ &= r_0 - AZ_m y \qquad (3) \\ &= \beta v_1 - V_{m+1}\bar{H}_m y \\ &= V_{m+1}[\beta e_1 - \bar{H}_m y]. \qquad (4) \end{aligned}$$

If we denote by $J_m(y)$ the function

$$J_m(y) = \|b - A[x_0 + Z_m y]\|_2,$$

we observe that by (4) and the fact that V_{m+1} is unitary,

$$J_m(y) = \|\beta e_1 - \bar{H}_m y\|_2. \tag{5}$$

Since step 3 of Algorithm 2.2 minimizes this norm over all vectors y in R^m to yield y_m, it is clear that the approximate solution $x_m = x_0 + Z_m y_m$ has the smallest residual norm in $x_0 + \text{Span}\{Z_m\}$. Thus we have proved the following result.

PROPOSITION 2.1. *The approximate solution x_m obtained at step m minimizes the residual norm $\|b - Ax_m\|_2$ over $x_0 + \text{Span}\{Z_m\}$.*

We will now examine a case of breakdown in FGMRES, which occurs when $h_{j+1,j} = 0$ in the last part of step 3 in Algorithm 2.2. In this situation, the vector v_{j+1} cannot be computed and the algorithm breaks down. For the standard GMRES algorithm this is not a problem because when this breakdown occurs, then the approximate solution x_j is exact. In fact, breakdown is equivalent to convergence. In FGMRES this is no longer true. More specifically, we have the following result.

PROPOSITION 2.2. *Assume that $\beta = \|r_0\|_2 \neq 0$ and that $j-1$ steps of* FGMRES *have been successfully performed, i.e., that $h_{i+1,i} \neq 0$ for $i < j$. In addition, assume that the matrix H_j is nonsingular. Then x_j is exact if and only if $h_{j+1,j} = 0$.*

Proof. If $h_{j+1,j} = 0$, then we have the relation $AZ_j = H_j V_j$, and as a result,

$$J_j(y) = \|\beta v_1 - AZ_j y_j\|_2 = \|\beta v_1 - V_j H_j y_j\|_2 = \|\beta e_1 - H_j y_j\|_2.$$

If we assume that H_j is nonsingular, then the above function is minimized for $y_j = H_j^{-1}(\beta e_1)$ and the corresponding minimum norm reached is zero, i.e., x_j is exact.

Conversely, if x_j is exact, then from (2) and (3) we have

$$0 = b - Ax_j = V_j[\beta_1 e_1 - H_j y_j] + \hat{v}_{j+1} e_j^T y_j. \tag{6}$$

If the last component of y_j is zero, then (6) would mean that $H_j y_j = \beta e_1$ and, since $h_{i+1,i} \neq 0$ for $1 \leq i \leq j-1$, a simple back-substitution starting from the last equation will show that all components of y_j are zero. This would imply that $\beta = 0$ and contradict the assumption. Hence $e_j^T y_j \neq 0$. Therefore, since $\hat{v}_{j+1}$ is orthogonal to $v_1, \ldots, v_j$ the only way in which (6) can hold is that $\beta_1 e_1 - H_j y_j = 0$ and $\hat{v}_{j+1} = 0$, which implies $h_{j+1,j} = 0$. □

Note that the only difference between this result and the one in [5] concerning the standard GMRES algorithm is that we must make the additional assumption that H_j is nonsingular since this is no longer implied by the nonsingularity of A.

A consequence of the result is that if at a given step j, we have $Az_j = v_j$, i.e., if the preconditioning is "exact" at step j, then the approximation x_j will be exact if in addition H_j is nonsingular. This is because $w = Az_j$ is linearly dependent on previous v_i's (it is equal to v_j), and as a result we will obtain $\hat{v}_{j+1} = 0$ after the orthogonalization process.

A difficulty with the theory of the new algorithm is that we cannot prove general convergence results such as those in [5]. This is because the subspace of approximants is no longer a standard Krylov subspace and we have no isomorphism with the space of polynomials. However, the optimality property of Proposition 2.1 can be exploited in some specific situations. For example, if within each outer iteration we select *at least one* of the vectors z_j to be a steepest descent direction vector, e.g., for the function $F(x) = \|b - Ax\|_2^2$, then FGMRES is guaranteed to converge, independently of m.

2.3. Practical considerations and applications. The additional cost incurred by the flexible variant over the standard algorithm is only in the extra memory required to save the set of vectors $\{z_j\}_{j=1,\ldots,m}$. On the other hand, the added advantage of *flexibility* may certainly be worth this extra cost. There are a few applications in which this flexibility can be quite helpful, especially in the context of developing robust iterative methods or for developing preconditioners for massively parallel computers. Here is a sample of possible applications.

1. Use of *any* iterative techniques as preconditioners: block-SOR, SSOR, ADI, multigrid, etc., but also GMRES, CGNR, CGS, etc.

2. Use of chaotic relaxation-type preconditioners (e.g., in a parallel computing environment).

3. Mixing preconditioners to solve a given problem.

For an example of (3), see the recent work by Tezduyar, Behr, Adadi, and Ray [6], in which two types of preconditioners are alternately applied at each FGMRES step to mix the effects of "local" and "global" interactions. Tezduyar et al. reported good performance with this procedure, much better than using either of the two preconditioners by itself.

Note that any iterative method can now be used as a preconditioner. For example, we will show how to make some nonnegligible improvements to the performance of the basic GMRES algorithm by using GMRES as preconditioner to itself (using for memory space the unused vectors at the ith step of FGMRES(m)).

Preconditioners of particular interest within this framework are relaxation-type techniques. As an example, the SSOR preconditioning matrix defined by

$$M_{\text{SSOR}}(A) = (D - \omega E)D^{-1}(D - \omega F),$$

in which $-E$ is the strict lower part of A, $-F$ is the strict upper part of A, and D is the diagonal of A, has some important advantages, some of which have been briefly discussed in the introduction. In the context of preconditioning, it is customary to just take $\omega = 1$, as the gains from selecting an optimal ω are typically small. However, it is clear that one can use different values of ω at each step of FGMRES and this can open up the possibility of using heuristics to determine the best ω dynamically, by simply monitoring convergence. Alternatively, a mixture of ω's can be initially selected and then used cyclically, instead of arbitrarily taking only $\omega = 1$ as is usually done. We would also like to make an important point concerning SOR as a preconditioner. The usual one-step SOR is not popular as a preconditioner, mainly because it tends to distribute the eigenvalues of the preconditioned matrix around a circle (e.g., for the model problem). This is not very desirable for a CG-type solver. However, we found that when using multiple steps, SOR is often more economical than SSOR as a preconditioner.

We will not discuss these applications here but refer the reader to [4]. Rather, we would like to demonstrate the flexibility of FGMRES by simply combining it with other iterative methods to improve its robustness. We are particularly interested in combinations with CGNR, and with the standard GMRES itself. The reason why we chose CGNR is that we know that the method is globally convergent and as a result, adding one direction vector to the standard Krylov subspace will guarantee global convergence because of the optimality of FGMRES. We should add, however, that guaranteeing global convergence is not the ultimate goal, since the convergence can still be too slow to be of any practical value.

3. Numerical experiments.

For test purposes we consider the problems arising from the centered difference discretization of problems of the form

$$-\Delta u + \gamma(xu_x + yu_y) + \beta u = f \tag{7}$$

on square regions with zero Dirichlet boundary conditions. In our first test we select the parameters $\gamma = 10$ and $\beta = -100$ to make the problem indefinite. The grid consists of a square of 32 internal mesh points in each direction leading to a matrix of size $N = 1024$. The right-hand side is selected once the matrix is constructed so that the solution is known to be $x = (1, 1, \ldots, 1)^T$. In all methods the initial vector is chosen so that its ith component for $i = 1, \ldots, n$ is defined by $x_0(i) = i$. We have compared the following methods.

1. A standard ILU(0) preconditioned GMRES(m) iteration with $m = 20$ direction vectors.

2. The CGNR iteration (conjugate gradient, normal equations) using an ILQ preconditioner with level-of-fill equal to 5; see [3] for details on these preconditioners. The idea of ILQ is to perform an incomplete Gram–Schmidt factorization on the rows of A. The normal equations are preconditioned using this factorization and the conjugate gradient method (CGNR version) is used to solve these equations.

3. The FGMRES iteration using (the unpreconditioned) GMRES itself as a preconditioner. As was briefly mentioned in the introduction, this is run as follows. We observe that at step i of the FGMRES iteration, the space for the vectors $v_{i+2}, \ldots, v_{m+1}$ and $z_k, k = i+1, \ldots, m$ is unused. However, they can be exploited to generate a preconditioned version of v_i by running a standard GMRES iteration using $2m - i - 1$ direction vectors. In fact, the vectors v_j and z_j occupy the same array $wk(1:n, 1:2m)$ and the z_k's are stored backwards starting in column $2m$ to avoid collisions. Note that for fairness in the comparisons, we use here $m = 10$ so that the total number of vectors needed is $2m = 20$, the same as is required for the ILU(0)-GMRES in 1.

4. The FGMRES iteration using ILU(0)-GMRES as a preconditioner. This is similar to the previous method except that the method used to precondition is ILU(0)-GMRES instead of the unpreconditioned GMRES.

5. The FGMRES iteration using k steps of the unpreconditioned CGNR iteration as a preconditioner. In our first test we took $k = 5$ and in the second, $k = 2$. Other values of k have also been tested and performed similarly.

Figure 1 is a plot of the residual norm achieved by these five methods against the number of operations (in millions) required to reach that level for the first test problem. We chose to plot the residuals versus the number of operations because the number of iterations is no longer significant for making comparisons since the cost of each iteration can be quite different for each of the methods compared. Observe that the FGMRES iteration using the unpreconditioned GMRES (method 3) converges, although slowly, whereas the standard GMRES(20) (not shown) as well as the ILU(0) preconditioned GMRES(20) both stagnate. More interesting is the convergence of FGMRES with ILU(0)-GMRES as a preconditioner. ILU(0)-GMRES(20) alone fails to converge (stopped after 700 steps). Used as a preconditioner, the technique converges in 15 outer iterations and yields the second best performance in this test. In this test CGNRILQ(5) performed quite well.

In our second test we took the same partial differential equation as before, but with the parameters $\gamma = 1000$ and $\beta = 10.0$, to make the problem highly nonsymmetric. The grid is the same as before and so the size of the matrix is still 1024. The right-hand side and the starting vector are generated similarly to the previous example. We have run the same methods as before, except that the number of CGNR steps used in FGMRES-CGNR is two instead of five. The results are shown in Fig. 2. Note that this time, CGNR-ILQ does not perform as well. In fact, some of the conclusions of the previous test are reversed. Thus, FGMRES-ILU(0)-GMRES is now outperformed by the simpler ILU(0)-GMRES(20). The FGMRES with unpreconditioned GMRES outperforms the FGMRES with ILU(0)-preconditioned GMRES by a slight margin. In addition, FGMRES with CGNR preconditioner using just two steps of CGNR is now the overall best.

4. Conclusion.

An interesting observation from the above experiments is that for indefinite and/or highly nonsymmetric matrices the performance of a given precon-

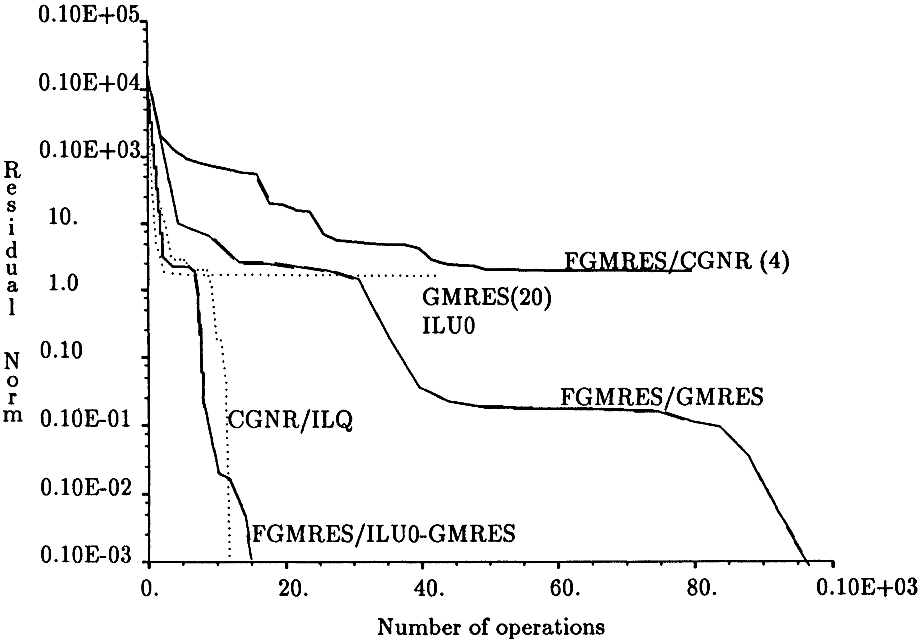

FIG. 1. *Performance of different iterative methods for first test problem.*

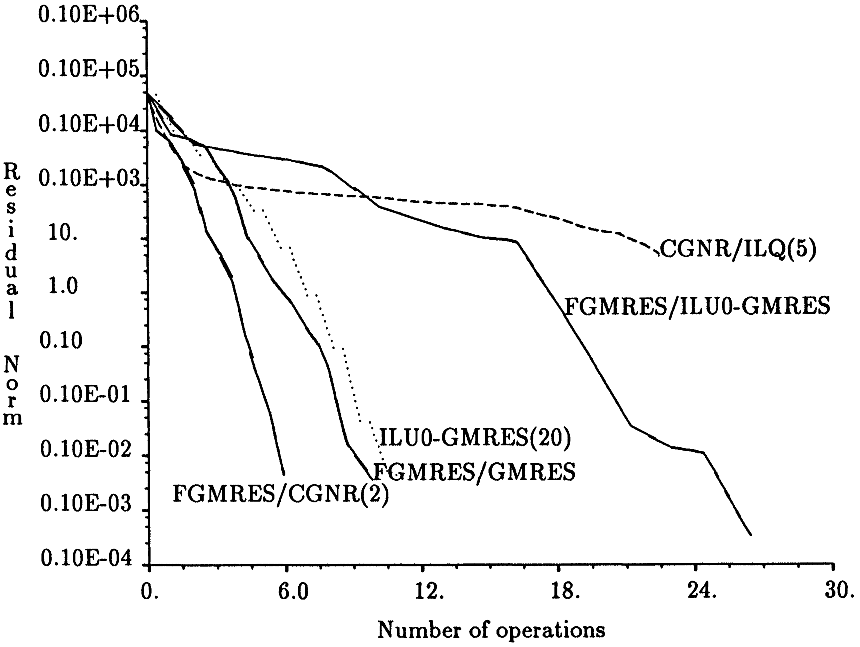

FIG. 2. *Performance of different iterative methods for second test problem.*

ditioner can be unpredictable. In these situations, it is essential to be able to switch preconditioners in order to improve robustness. FGMRES is an algorithm that allows arbitrary changes in the preconditioner and can be used to this end. There are many other uses of the flexible variant of GMRES. In addition, the difference in the coding of the two methods is so small that they can both be implemented, with no loss of efficiency, in a single subroutine that incorporates an option parameter.

Acknowledgments. The author would like to acknowledge the support of the Minnesota Supercomputer Institute, which provided the computer facilities and other resources for this research. The author is indebted to Henk van der Vorst for pointing out an error in an earlier version of this paper.

REFERENCES

[1] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[2] J. A. Meijerink and H. A. van der Vorst, *An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix*, Math. Comp., 31 (1977), pp. 148–162.

[3] Y. Saad, *Iterative solution of indefinite symmetric systems by methods using orthogonal polynomials over two disjoint intervals*, SIAM J. Numer. Anal., 20 (1983), pp. 784–811.

[4] ———, *Highly parallel preconditioners for general sparse matrices*, Tech. Rep. 92-087, Army High Performance Computing Research Center, Minneapolis, MN, 1992.

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

[6] T. E. Tezduyar, M. Behr, S. K. A. Abadi, and S. E. Ray, *A mixed* CEBE/CC *preconditioning for finite element computations*, Tech. Rep. UMSI 91/160, Univ. of Minnesota, Minnesota Supercomputing Institute, Minneapolis, MN, June 1991.

[7] H. A. van der Vorst and C. Vuik, GMRESR: *A family of nested GMRES methods*, Tech. Rep. 91-80, Delft University of Technology, Mathematics and Informatics, Delft, the Netherlands, 1991.