Restarted GMRES preconditioned by deflation

J. Erhel [1], K. Burrage [2] and B. Pohl [3]

Research Report No. 94-04
July 1994

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

[1]INRIA, Campus de Beaulieu, 35042 Rennes, France.
[2]Department of Mathematics, University of Queensland, Brisbane 4072, Australia.
[3]Seminar für Angewandte Mathematik, ETH Zürich, 8092 Zürich, Switzerland.

Restarted GMRES preconditioned by deflation

J. Erhel [1], K. Burrage [2] and B. Pohl [3]

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Research Report No. 94-04 July 1994

Abstract

This paper presents a new preconditioning technique for solving linear systems. It is based on an invariant subspace approximation for the restarted GMRES algorithm. It uses the flexible GMRES scheme by designing a new preconditioning after each restart. Numerical examples show that this approach may converge almost as fast as full-GMRES at a, possibly, much lower cost.

Keywords: GMRES, preconditioning, invariant subspace

Subject Classification: 65F10, 65F15

[1]INRIA, Campus de Beaulieu, 35042 Rennes, France.
[2]Department of Mathematics, University of Queensland, Brisbane 4072, Australia.
[3]Seminar für Angewandte Mathematik, ETH Zürich, 8092 Zürich, Switzerland.

1 Introduction

The GMRES algorithm is commonly used to solve large sparse nonsymmetric linear systems. The convergence behaviour of GMRES is related to the eigenvalues and also to the pseudo-eigenvalues (eigenvalues of closed matrices) [11]. Recently, the convergence behaviour of the full-version has been analyzed [6] and superlinear convergence has been related to the convergence of Ritz values. However, because of memory requirements, a restarted version must be used in general. It has been observed that the convergence of the restarted algorithm depends heavily on the dimension of the Krylov subspace and may be slower than in the full case [7]. It appears as if the restarting procedure loses information on the smallest Ritz values. An adaptive procedure is proposed in [10] to choose the restart frequency according to the convergence and work requirements.

This paper presents a preconditioning technique which aims at keeping the information when restarting. The idea is to estimate the invariant subspace corresponding to the smallest eigenvalues. Indeed, the rate of convergence is mostly governed by these smallest eigenvalues.

Many authors have proposed preconditioners or hybrid methods based on eigenvalue estimations. For the Conjugate Gradient algorithm, polynomial preconditioning aims at minimizing a certain norm. Two classical choices lead to the least-squares polynomial or to the minimax polynomial [9, 13]. The quality of the minimax polynomial depends strongly on the eigenvalue estimations. An adaptive procedure which is based on a recursive estimation of the eigenvalues is designed in [1, 2] for both definite and indefinite systems.

Polynomial preconditioning is closely related to hybrid methods which combine, for example, a GMRES algorithm with a Richardson iteration. The idea is to use first GMRES to approximate both the solution and eigenvalues and then to use Richardson iteration using a polynomial derived from the estimated eigenvalues. A survey of hybrid methods which rely on eigenvalue estimations can be found in [12]. These estimations are usually done by the power method or by the Arnoldi technique but they can also be computed from modified moments [5]. Other hybrid solutions do not rely on eigenvalue estimations but use directly a polynomial generated by GMRES itself [12]. An alternative approach discussed in [17, 15] is to build a preconditioner based on the application of GMRES.

In this paper the eigenvalue technique is not used, but rather an invariant subspace approach. This idea has been developed in [8, 16] for the solution of nonlinear parameter-dependent systems of equations, in which a Newton method is used in the invariant subspace corresponding to the eigenvalues of the Jacobian near the unit disk and the usual fixed-point scheme is used in the orthogonal subspace. Therefore, the convergence is accelerated since the eigenvalues in the orthogonal subspace can be made small enough by a deflation approach. This idea has been applied in [4] in the linear case to the iterative

methods based on various splittings of the matrix such as Jacobi or Gauss-Seidel splittings. It is shown there in numerical experiments that the relaxation methods can be dramatically accelerated.

However, the GMRES algorithm cannot be easily described as a fixed-point scheme. Here the convergence is related to the smallest eigenvalues. The full-GMRES version behaves as if the smallest eigenvalues were removed after some iterations. But this is no longer true in the restarted case. Therefore, the aim of this paper is to remove them by a preconditioner. After each restart, a new preconditioner is built and the flexible GMRES method can be used [15]. At each restart, new eigenvectors are estimated in order to increase the invariant subspace. The preconditioner is almost equal to the matrix on the approximated invariant subspace and is taken as the identity on the orthogonal subspace.

Numerical results strongly suggest that this technique cannot converge faster than the full-GMRES version since it does not recover all the information kept in the full scheme. But numerical experiments show that for some matrices, it can converge much faster than the restarted version and almost as fast as the full scheme. Moreover, the preconditioned scheme can be faster in CPU time thanks to a lower complexity. It also requires less memory than a full version.

Of course, this technique can be combined with any preconditioner, estimating the invariant subspace of the preconditioned matrix. This approach can be used also to solve consecutive linear systems with the same matrix, as frequently happens in scientific computation. Thus at the convergence of the first linear system, a quite accurate invariant subspace may be computed and used to build a robust preconditioner for the subsequent resolution.

Another advantage of this method is the readily parallelizable use of the preconditioner by means of level 2 dense BLAS operations. Moreover, the algorithm only requires a matrix-vector product and can be applied to so-called matrix-free versions of GMRES where the matrix is not stored.

Thus this paper is organized as follows. In section 2, the GMRES variants are given. Section 3 is devoted to the design of the preconditioner and to implementation issues whereas section 4 describes numerical experiments.

2 Definitions and notations

In this section we recall the basic GMRES algorithms [14] and the new flexible version [15].

Thus consider the linear system $Ax = b$ with $x, b \in \mathbb{R}^n$ and with a non-singular nonsymmetric matrix $A \in \mathbb{R}^{n\times n}$. The Krylov subspace $K(k; A; r_0)$ is defined by $K(k; A; r_0) = span(r_0, Ar_0, \ldots, A^{k-1}r_0)$. The GMRES algorithm uses the Arnoldi process to construct an orthonormal basis $V_k = [v_1, \ldots, v_k]$ for $K(k; A; r_0)$. The full GMRES method allows the Krylov subspace dimension to increase up to n and always terminates in at most n iterations. The restarted GMRES method on the other hand restricts the Krylov subspace dimension to

a fixed value m and restarts the Arnoldi process using the last iterate x_m as an initial guess. It may stall if it gets a residual r satisfying $r^T Ar = 0$. Below is the algorithm for the restarted GMRES.

ALGORITHM : GMRES(m)

ϵ is the tolerance for the residual norm ;
convergence:= false ;
choose x_0 ;
until convergence **do**
  $r_0 = b - Ax_0$;
  $\beta = ||r_0||$;
  $v_1 := r_0/\beta$;
  for $j = 1, \cdots, m$ **do**
    $p := Av_j$;
    for $i = 1, \cdots j$ **do**
      $h_{ij} := v_i^T p$;
      $p := p - h_{ij} v_i$;
    endfor
    $h_{j+1,j} := ||p||_2$;
    $v_{j+1} := p/h_{j+1,j}$;
    if $||b - Ax_j|| < \epsilon$ **then**
      solve $\min_{y_j \in \mathbf{R}^j} ||\beta e_1 - \overline{H_j} y_j||$;
      $x_j := x_0 + V_j y_j$;
      convergence := true ;
    endif;
  endfor ;
  solve $\min_{y_m \in \mathbf{R}^m} ||\beta e_1 - \overline{H_m} y_m||$;
  $x_m := x_0 + V_m y_m$;
  if $||b - Ax_m|| < \epsilon$ convergence := true ;
  $x_0 = x_m$;
enddo

For $1 \leq k \leq m$, the matrix $\overline{H_k} = (h_{ij})$ is a upper Hessenberg matrix of order $(k+1) \times k$ and we get the fundamental relation

$$AV_k \quad = \quad V_{k+1} \overline{H_k}.$$

The GMRES algorithm computes $x = x_0 + V_k y_k$ where y_k solves the least-squares problem $\min_{y_k \in \mathbf{R}^k} ||\beta e_1 - \overline{H_k} y_k||$. Usually a QR factorisation of $\overline{H_k}$ using Givens rotations is used to solve this least-squares problem.

The linear system can be preconditioned either at left or at right solving, respectively, $M^{-1}Ax = M^{-1}b$ or $AM^{-1}(Mx) = b$ where M is the preconditioning matrix.

The flexible GMRES version allows the use of a variable preconditioning matrix. There is currently no general convergence result here because there is

no relationship with a polynomial. In this algorithm the preconditioning matrix M is updated at each restart.

ALGORITHM : PRECGMRES(m)
ϵ is the tolerance for the residual norm ;
convergence:= false ;
choose x_0 ;
choose M ;
until convergence **do**
  $r_0 = M^{-1}(b - Ax_0)$;
  Arnoldi process applied to $M^{-1}A$ to compute V_m ;
  solve $\min_{y_m \in \mathbf{R}^m} ||\beta e_1 - \overline{H_m} y_m||$;
  $x_m := x_0 + V_m y_m$;
  if $||M^{-1}(b - Ax_m)|| < \epsilon$ convergence := true ;
    $x_0 = x_m$;
  else
    update M
  endif;
enddo

3 The construction of the preconditioner

In this section, we describe how to build and to update the preconditioner and we also discuss the convergence properties. In the sequel, it will be assumed that all eigenvalues of A are nondefective. Let $|\lambda_1| \leq |\lambda_2| \leq \ldots \leq |\lambda_n|$ be the eigenvalues of A.

Let P be an invariant subspace of dimension r corresponding to the smallest r eigenvalues of A, and let $Z = (U, W)$ be an orthonormal basis of $\mathbb{R}^n$ where U is an orthonormal basis of P. In this basis, A is similar to a matrix $\tilde{A}$ which is written in the Schur form as

$$\tilde{A} = \begin{pmatrix} T & A_{12} \\ 0 & A_{22} \end{pmatrix}, \tag{1}$$

where $T = U^T A U$ is the restriction of A onto the subspace P.

Let M be a matrix defined by

$$M = Z \begin{pmatrix} T/|\lambda_n| & 0 \\ 0 & I_{n-r} \end{pmatrix} Z^T, \tag{2}$$

where I_{n-r} is the identity matrix. M is nonsingular and its inverse is easily computed by

$$M^{-1} = Z \begin{pmatrix} |\lambda_n| T^{-1} & 0 \\ 0 & I_{n-r} \end{pmatrix} Z^T. \tag{3}$$

In practice, the basis W is unknown so that this matrix form of M^{-1} cannot be used. Therefore M^{-1} can be written as

$$\begin{aligned} M^{-1} &= |\lambda_n| U T^{-1} U^T + (I_n - U U^T), \\ &= I_n + U(|\lambda_n| T^{-1} - I_r) U^T, \end{aligned} \tag{4}$$

which is very easy and cheap to compute.

The preconditioned matrix $M^{-1}A$ is therefore similar in this basis to

$$E = \begin{pmatrix} |\lambda_n| I_r & |\lambda_n| T^{-1} A_{12} \\ 0 & A_{22} \end{pmatrix},$$

so that its eigenvalues are $|\lambda_n|$ and the remaining eigenvalues of A, that is to say the eigenvalues of A_{22}. In particular, the condition number is now $|\lambda_n|/|\lambda_{r+1}|$ instead of $|\lambda_n|/|\lambda_1|$. Hence the GMRES algorithm applied to $M^{-1}A$ converges faster than when applied to A.

However, we do not know exactly the invariant subspace P but only an approximation $\tilde{P}$. Using an orthonormal basis $(\tilde{U}, \tilde{W})$ we get now the same preconditioner M but a nonzero block in the (2,1) position of the matrices $\tilde{A}$ and E:

$$\tilde{A} = \begin{pmatrix} T & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad E = \begin{pmatrix} |\lambda_n| I_r & |\lambda_n| T^{-1} A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \tag{5}$$

where $T = \tilde{U}^T A \tilde{U}$ and $|\lambda_n|$ approximates the largest eigenvalue of A. This gives a perturbed matrix where the perturbation is given by the block A_{21}. If this block is small enough, the eigenvalues of $M^{-1}A$ are close to $|\lambda_n|$ and to the eigenvalues of A_{22} (recall that the eigenvalues are supposed to be nondefective), and we can still expect an improved convergence rate for this preconditioned GMRES.

3.1 Computing the invariant subspace

The GMRES algorithm provides the Hessenberg matrix $H_k = V^T A V$ which is the restriction of A onto the Krylov subspace $K(k; A; r_0)$. The eigenvalues of H_k are called the Ritz values. It is well-known that the extremal Ritz values approximate the extremal eigenvalues of A.

Let us assume that H_k is decomposed into the Schur form with the eigenvalues ordered by increasing values with the Schur vectors S corresponding to the m smallest eigenvalues. Then the vectors $U = V_k S$ approximate the Schur vectors of A corresponding to the smallest eigenvalues of A. Since the largest Ritz value approximates the largest eigenvalue of A, we can therefore construct a matrix M.

After each restart we estimate new Ritz values which approximate the eigenvalues of $M^{-1}A$ which in turn approximate the eigenvalues of A_{22} and hence

the remaining eigenvalues of A. We increase the size of the invariant subspace to get a more powerful preconditioner by adding new Schur vectors. In order to avoid loss of orthogonality, these vectors are orthogonalized against the previous basis U.

Thus a flexibly-preconditioned-restarted GMRES version is built. In some sense, this algorithm recovers the superlinear convergence of the full-GMRES version which behaves as if the smallest eigenvalues were removed. This approach has some merit when dealing with a restarted version. In this case, the preconditioner keeps the information on the smallest Ritz values which would be lost by the restart. Moreover, this preconditioner is cheap and easily parallelizable, so that it can be faster in CPU time than a full scheme.

Currently a fixed number r of eigenvalues are extracted after each restart. More precisely, if H_m has complex conjugate eigenvalues or eigenvalues of the same modulus, then all the corresponding Schur vectors are extracted. Hence the number of extracted eigenvalues may vary in some extent from one restart to another.

Below is the new flexibly-preconditioned-restarted GMRES:

ALGORITHM : FLEXGMRES(m,r)
ϵ is the tolerance for the residual norm ;
convergence:= false ;
choose x_0 ;
$M := I_n$;
$U := \{\}$;
until convergence **do**
  $r_0 = M^{-1}(b - Ax_0)$;
  Arnoldi process applied to $M^{-1}A$ to compute V_m ;
  solve $\min_{y_m \in \mathbf{R}^m} ||\beta e_1 - \overline{H_m} y_m||$;
  $x_m := x_0 + V_m y_m$;
  if $||M^{-1}(b - Ax_m)|| < \epsilon$ convergence := true ;
    $x_0 = x_m$;
  else
    compute r Schur vectors of H_m noted S_r ;
    orthogonalize $V_m S_r$ against U ;
    increase U by $V_m S_r$;
    $T := U^T A U$;
    $M^{-1} := I_n + U(|\lambda_n| T^{-1} - I) U^T$;
  endif
enddo

Here S_r denotes a set of r Schur vectors in the Hessenberg matrix H_m of order m corresponding to its r smallest eigenvalues. The basis U is increased by adding the new Schur vectors $V_m S_r$ after they have been orthogonalized against the previous basis U.

3.2 Implementation issues and complexity analysis

This new scheme involves the computation of the preconditioner after each restart and the resolution of the preconditioned system at each iteration. Assume that at each restart, until convergence, always r vectors $u = (u_1, \ldots, u_r)$ are added to the basis U of the approximate invariant subspace; then at the j^{th} restart the matrix $T = U^T AU$ is of order $s = r \times j$ and the matrix U has s vectors of size n. The total cost for P restarts in the scheme FLEXGMRES is

$$C_{Total} = \sum_{j=0}^{P-1} (C_{Arnoldi} + C_{Prec} + C_{Basis}), \tag{6}$$

where $C_{Arnoldi}$ represents the cost for one restart, C_{Prec} is the cost to solve the preconditioned system and C_{Basis} is the cost to increase the basis at each restart.

All costs are evaluated in number of floating-point operations. For simplicity, we neglect the terms which are independent of n. The cost for a matrix-vector product is $2\tau n$ where τ is the mean number of nonzeros per row.

The cost for Q restarts of the classical GMRES(m) is then

$$\begin{aligned} Q \times C_{Arnoldi} &= Q * 2(\tau nm + nm(m+1) + 2nm) \\ &= Q * 2nm(\tau + m + 3). \end{aligned}$$

The preconditioned system can be solved using dense BLAS2 primitives, which gives a complexity of about

$$C_{Prec} = 4nms. \tag{7}$$

At each restart, the new vectors u must be computed and orthogonalized with the previous set of vectors in order to increase the basis U. Also, the matrix $T = U^T AU$ must be updated and factorized, using the block decomposition

$$T = \begin{pmatrix} U^T AU & U^T Au \\ u^T AU & u^T Au \end{pmatrix}.$$

This gives

$$C_{Basis} = 2nr(\tau + m + r + 4s). \tag{8}$$

The global complexity for P restarts of the new scheme is

$$C_{Total} = P * 2n(m(\tau + m + 3) + r(\tau - r) + r(m + 2r)P). \tag{9}$$

Hence, the scheme will perform better than the classical restarted GMRES scheme if

$$C_{Total} < Q \times C_{Arnoldi}. \tag{10}$$

Figure 1 plots the curve where both costs are the same for $m = 10$ and $r = 1$ and for two values of τ, $\tau = 7$ and $\tau = 50$. Under each curve, FLEXGMRES(m,r) is more efficient than GMRES(m) and above the curve GMRES(m) is more efficient. It can be seen that a modest acceleration in convergence is sufficient to obtain good performances.

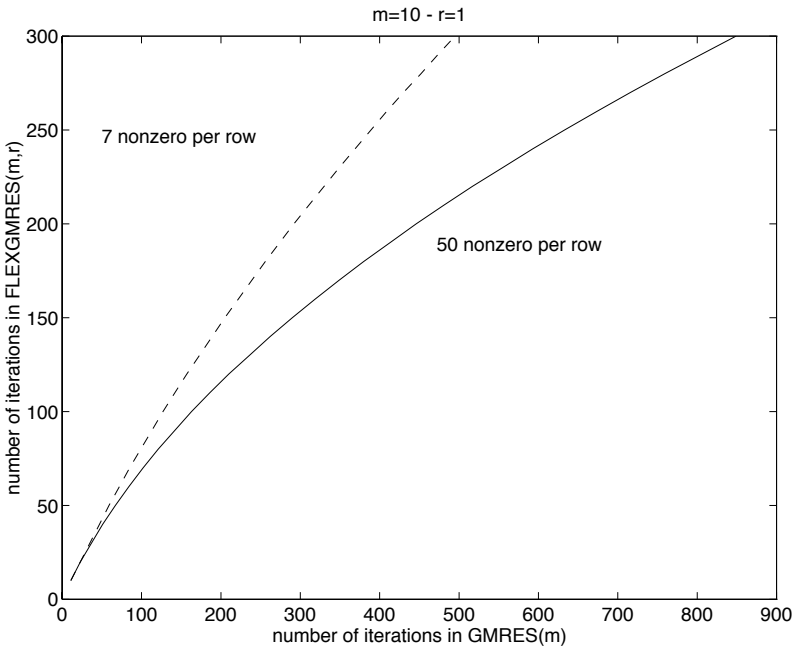

Figure 1: Cost evaluation for GMRES(m) and FLEXGMRES(m,r)

4 Numerical Results

We have tested the algorithm using Matlab and the template of GMRES provided in netlib [3]. The matrices are taken from [7] and have the form $A = SDS^{-1}$ with $A, S, D \in \mathbb{R}^{100\times 100}$ and with $S = (1, \beta)$ a bidiagonal matrix with 1 on the diagonal and β on the upper subdiagonal.

The system $Ax = b$ is solved for right-hand sides $b = (1, \ldots, 1)^T$ and GMRES starts with $x_0 = 0$. In both cases, the restarted and the flexible version, GMRES is restarted every 10 iterations, except otherwise stated.

Results for four different examples of dimension 100 are presented with the

following characteristics:

No.	β	D	condition number $\kappa(S)$
1	0.9	$\mathrm{diag}(1, 2, \ldots 100)$	18.334
2	1.1	$\mathrm{diag}(1, 2, \ldots 100)$	151570
3	0.9	$\mathrm{diag}(1, 100, 200 \ldots, 10000)$	
4	0.9	$\mathrm{diag}(-10, -9, \ldots, -1, 1, 2, \ldots, 90)$	

Figures 2 and 3 show the convergence rate for the full-GMRES version, the restarted version with no preconditioning and the restarted version with the flexible preconditioning estimating one or two eigenvalues at each restart. For Examples 1 and 3 all the methods converge whereas for Examples 2 and 4 the restarted method does not converge. It is interesting to note that the presented flexible scheme does converge. Tables 1 and 2 compare the CPU times for the various schemes.

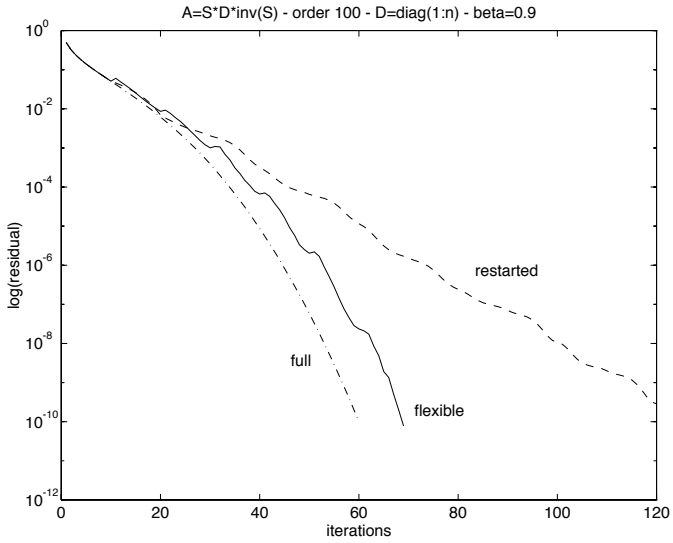

Figure 2: Convergence rates for Example 1

method	restart	nb. of eig.	iterations	CPU time
full	100	0	60	6.4000
restarted	10	0	120	5.4833
flexible	10	1	118	5.5167
flexible	10	2	89	4.4667
flexible	10	3	85	4.3833
flexible	10	4	75	4.0500
flexible	10	5	76	4.3833
flexible	10	6	69	4.0833

Table 1: Results for Example 1

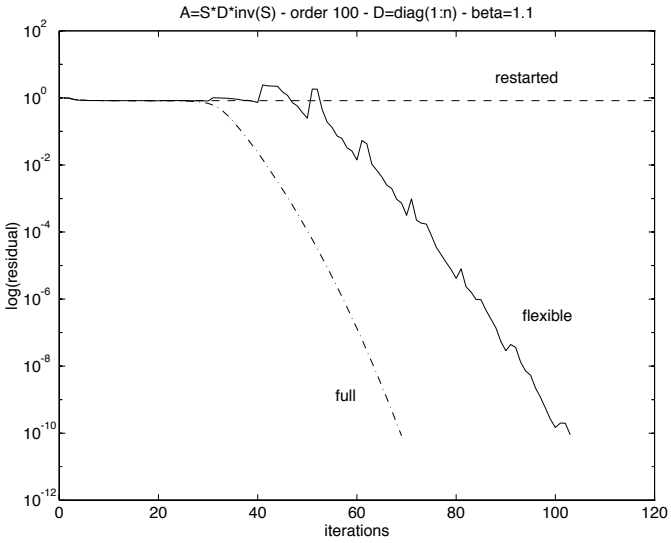

Figure 3: Convergence rates for Example 2

method	restart	nb. of eig.	iterations	CPU time
full	100	0	69	7.7667
restarted	10	0	∞	*
flexible	10	6	∞	*
flexible	10	8	120	6.6500
flexible	10	14	103	7.8833

Table 2: Results for Example 2

As shown in Figure 4, the restarted method does not converge for Example

3, but the flexible method converges slightly slower than the full method. Table 3 gives the CPU time.

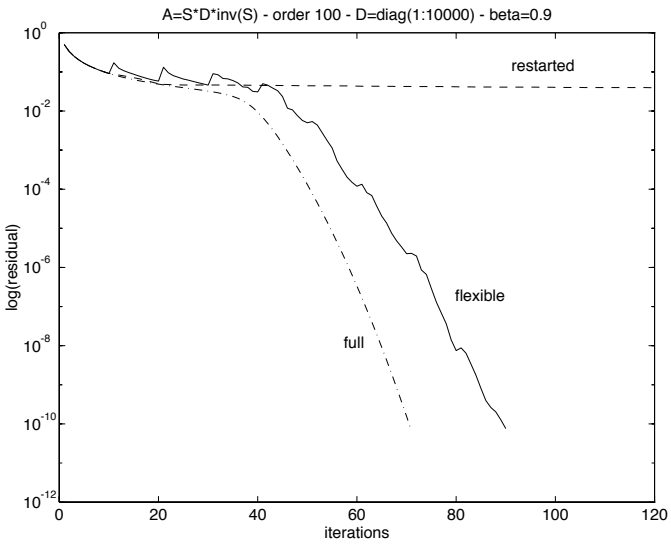

Figure 4: Convergence rates for Example 3

method	restart	nb. of eig.	iterations	CPU time
full	100	0	71	8.2667
restarted	10	0	∞	*
flexible	10	8	90	5.5667

Table 3: Results for Example 3

The convergence curves in Figure 5 show a drop of the residual in the full case when the Krylov subspace is almost of size n, the order of the linear system. The restarted method is very slow and the flexible scheme converges in about 300 iterations. Table 4 shows the CPU time.

method	restart	nb. of eig.	iterations	CPU time
full	100	0	88	32.6667
restarted	10	0	∞	*
flexible	10	7	∞	*
flexible	10	8	788	20.9500
flexible	10	17	179	6.6167

Table 4: Results for Example 4

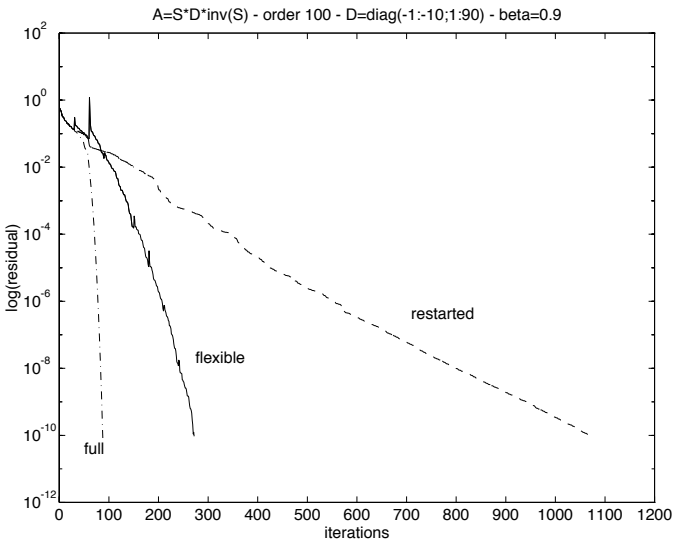

Figure 5: Convergence rates for Example 4

These examples show that the flexible scheme converges almost as fast as the full scheme. Moreover, in the above examples, the flexible scheme converges when the restarted version stalls, provided the size of the invariant subspace is large enough. For a fixed frequency of restarting, the CPU time achieves a minimal value when increasing the size of the invariant subspace then increases. However, this size must be large enough to guarantee convergence in difficult cases. A good tradeoff seems to keep this size increasing when memory space is sufficient.

5 Perspectives

This paper presents a new flexible GMRES scheme defining a variable preconditioner based on the estimation of Schur vectors and on deflation techniques. Examples presented here show that this preconditioned restarted scheme converges whereas the unpreconditioned restarted scheme may sometimes stall. In the above examples the convergence rate of the flexible scheme was slower than the full scheme but faster than the restarted scheme. Preliminary timings indicate however that the flexible scheme requires less CPU time than the two other schemes.

Further work needs to be done in considering different strategies for updating the preconditioning matrix, including the development of an adaptive approach.

Other possible approaches include a version where the dimension of U does not increase very much but approximates an invariant subspace more accurately.

Finally, it is intended to extend this work to a parallel implementation in a MIMD environment.

Acknowledgements:
Part of this work was done while Kevin Burrage was a guest professor of the Seminar für Angewandte Mathematik at ETH Zürich and while Jocelyne Erhel and Bert Pohl were Ethel Raybould Fellows in the Department of Mathematics at the University of Queensland in Australia.
Kevin Burrage gratefully acknowledges the financial support provided by the Intel Corporation.

References

[1] S.F. Ashby. *Polynomial preconditioning for conjugate gradient methods.* PhD thesis, Univ. of Illinois at Urbana-Champaign, December 1987.

[2] Steven F. Ashby, Thomas A. Manteuffel, and Paul E. Saylor. Adaptive polynomial preconditioning for hermitian indefinite linear systems. *BIT*, 29:583–609, 1989.

[3] R. Barret, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods.* SIAM / netlib, 1993.

[4] Kevin Burrage, Jocelyne Erhel, and Bert Pohl. A deflation technique for linear systems of equations. *Research Report No. 94-02, Seminar für Angewandte Mathematik, ETH Zürich*, 1994.

[5] D. Calvetti, G.H. Golub, and L. Reichel. An adaptive chebyshev iterative method for nonsymmetric linear systems based on modified moments. *Numer. Math.*, 67:21–40, 1994.

[6] H.A. Van der Vorst. The superlinear convergence behaviour of gmres. *Journal of Computational and Applied Mathematics*, 48:327–341, 1993.

[7] Yunging Huang and Henk van der Vorst. Some observations on the convergence behavior of gmres. Technical Report 89-09, Faculty of technical mathematics and informatics, Delft, 1989.

[8] H. Jarausch and W. Mackens. Numerical treatment of bifurcation problems by adaptive condensation. In Weber Kupper, Mittelmann, editor, *Numerical Methods for bifurcation problems.* Birkhauser, 1987.

[9] Olin G. Johnson, Charles A. Michelli, and George Paul. Polynomial preconditioners for conjugate gradient calculations. *SIAM J. Numer. Anal.*, 20(2):362–376, April 1983.

[10] Wayne Joubert. On the convergence behaviour of the restarted gmres algorithm for solving nonsymmetric linear systems. *to appear in Journal of Numerical Linear Algebra with Applications.*

[11] Noel M. Nachtigal, Satish C. Reddy, and Lloyd N. Trefethen. How fast are nonsymmetric matrix computations ? *SIAM J. Matrix Anal. Appl.*, 13(3):778–795, July 1992.

[12] Noel N. Nachtigal, Lothar Reichel, and Lloyd N. Trefethen. A hybrid gmres algorithm for nonsymmetric linear systems. *SIAM J. Matrix Anal. Appl.*, 13(3):796–825, July 1992.

[13] Y. Saad. Practical use of polynomial preconditioning for the conjugate gradient method. *SIAM J. Sci. Stat. Comput.*, 6(4):865–881, 1985.

[14] Y. Saad and M.H. Schultz. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[15] Youcef Saad. A flexible inner-outer preconditioned gmres algorithm. *SIAM J. Sci. Stat. Comput.*, 14(2):461–469, March 1993.

[16] Gautam M. Shroff and Herbert B. Keller. Stabilization of unstable procedures: the recursive projection method. *SIAM J. Numer. Anal.*, 30(4):1099–1120, August 1993.

[17] H.A. van der Vorst and C. Vuik. Gmresr : a family of nested gmres methods. Technical report, jun 1992.

Research Reports

No.	Authors	Title
94-04	J. Erhel, K. Burrage, B. Pohl	Restarted GMRES preconditioned by deflation
94-03	L. Lau, M. Rezny, J. Belward, K. Burrage, B. Pohl	ADVISE - Agricultural Developmental Visualisation Interactive Software Environment
94-02	K. Burrage, J. Erhel, B. Pohl	A deflation technique for linear systems of equations
94-01	R. Sperb	An alternative to Ewald sums
93-07	R. Sperb	Isoperimetric Inequalities in a Boundary Value Problem in an Unbounded Domain
93-06	R. Sperb	Extension and simple Proof of Lekner's Summation Formula for Coulomb Forces
93-05	A. Frommer, B. Pohl	A Comparison Result for Multisplittings Based on Overlapping Blocks and its Application to Waveform Relaxation Methods
93-04	M. Pirovino	The Inverse Sturm-Liouville Problem and Finite Differences
93-03	R. Jeltsch, X. Wang	Uniqueness of Piecewise Lipschitz Continuous Solutions of the Cauchy-Problem for 2×2 Conservation Laws
93-02	W.-A. Yong	Difference approximations to the global $W^{1,\infty}$ solutions of the isentropic gas equations
93-01	Ch. Lubich, K. Nipp, D. Stoffer	Runge-Kutta solutions of stiff differential equations near stationary points
92-15	N. Botta	Is the transonic flow around a cylinder always periodic?
92-14	K. Nipp, D. Stoffer	Invariant manifolds of numerical integration schemes applied to stiff systems of singular perturbation type - Part I: RK-methods
92-13	K. Nipp	Smooth attractive invariant manifolds of singularly perturbed ODE's
92-12	D. Mao	A Shock Tracking Technique Based on Conservation in One Space Dimension
92-11	K. Nipp, D. Stoffer	Attractive invariant manifolds for maps: Existence, smoothness and continuous dependence on the map
92-10	M. Fey, R. Jeltsch	A Simple Multidimensional Euler Scheme
92-09	M. Fey, R. Jeltsch	A New Multidimensional Euler Scheme
92-08	M. Fey, R. Jeltsch, P. Karmann	Numerical solution of a nozzle flow
92-07	M. Fey, R. Jeltsch, P. Karmann	Special aspects of reacting inviscid blunt body flow