Numerical Linear Algebra with Applications, Vol. 4(1), 43–66 (1997)

Deflated and Augmented Krylov Subspace Techniques

Andrew Chapman[1,*] and Yousef Saad[2]

[1] *Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN 55455, USA*

[2] *Department of Computer Science, University of Minnesota, Minneapolis, MN 55455, USA*

We present a general framework for a number of techniques based on projection methods on 'augmented Krylov subspaces'. These methods include the deflated GMRES algorithm, an inner–outer FGMRES iteration algorithm, and the class of block Krylov methods. Augmented Krylov subspace methods often show a significant improvement in convergence rate when compared with their standard counterparts using the subspaces of the same dimension. The methods can all be implemented with a variant of the FGMRES algorithm. © 1997 by John Wiley & Sons, Ltd. Numer. Linear Algebra Appl., Vol. 4, 43–66 (1997)

(No. of Figures: 5 No. of Tables: 6 No. of Refs: 11)

KEY WORDS deflated GMRES; inner-iteration GMRES; block GMRES; augmented Krylov subspace; flexible GMRES

1. Introduction

There are three techniques which are sometimes used to enhance the robustness of Krylov subspace methods. The first is to exploit block versions of Krylov subspace methods. These block methods are known to be generally more reliable than their scalar equivalents mainly because they tend to better accommodate clustering of eigenvalues around zero. The second technique which has been suggested more recently [4,2,1], is to explicitly deflate the matrix from eigenspaces which hamper convergence, typically using approximate eigenvectors associated with eigenvalues nearest zero. A third method is the inner–outer iteration using FGMRES, for example, as the outer loop, and any subsidiary iteration for the inner loop,

[*] Correspondence to A. Chapman, NEC systems Laboratorly, 4200 Research Forest Drive, The Woodlands, TX 77381, USA

Contract grant sponsor: NASA; contract grant sponsor: NSF; contract grant number NAG2-904; contract grant number CCR-9214116

CCC 1070–5325/97/010043–24 $17.50 Received 20 October 1995
Revised 19 July 1996

see [7], and the related work [11]. These three techniques have the same goal, but the second approach is more explicitly aimed at enhancing convergence by modifying the spectrum of the original matrix. In this paper we present a framework for implementing these three techniques using a variant of the FGMRES algorithm given by Morgan [4]. We also present some theoretical considerations, and the results of a few numerical experiments. We begin with a description of the deflation method for the right preconditioned system $AM^{-1}Mx = b$.

2. Deflated GMRES

In deflation methods there are two distinct ways of exploiting some knowledge about approximate eigenvectors of AM^{-1} to accelerate convergence. The first is to explicitly deflate the eigenvectors from the matrix, by solving for example,

$$(I - \sigma uv^{\mathrm{H}})AM^{-1}x = (I - \sigma uv^{\mathrm{H}})b$$

Here v is a left eigenvector of AM^{-1} and u is arbitrary but is often chosen to be equal to v. This viewpoint is akin to that of preconditioning and is adopted in [2] and [1].

The second approach is simply to add the desired eigenvectors directly to the Krylov subspace. For eigenvalue problems this is quite natural [6]. For linear systems, an approach of this sort has been proposed by Morgan [4]. Morgan injects eigenvectors $u_1, u_2, \ldots, u_p$ into the Krylov subspace, so that the solution x then belongs to

$$x_0 + \mathrm{span}\{M^{-1}(r_0, AM^{-1}r_0, \ldots, (AM^{-1})^{m-p-1}r_0, u_1, u_2, \ldots, u_p)\}$$

Morgan's algorithm with minor extensions to allow for right preconditioning is given below.

Algorithm 2.1. *Right preconditioned GMRES with deflation*

```
1.  Compute $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$ and $v_1 = r_0/\beta$.
2.  For $j = 1, \ldots, m$ do
3.      Set $w_j = \begin{cases} v_j & \text{if } j \le m - p \\ u_{j-m+p}(eigenvector) & otherwise \end{cases}$
4.      Compute $z := AM^{-1}w_j$;
5.      For $i = 1, \ldots, j$, do $\begin{cases} h_{i,j} := (z, v_i) \\ z := z - h_{i,j}v_i \end{cases}$
6.      Compute $h_{j+1,j} = \|z\|_2$ and $v_{j+1} = z/h_{j+1,j}$
7.  EndDo
8.  Define $V_{m+1} := [v_1, \ldots, v_{m+1}]$, $W_m := [w_1, \ldots, w_m]$,
9.  Compute $x_m = x_0 + M^{-1}W_m y_m$, where $y_m = \mathrm{argmin}_y \|\beta e_1 - \bar{H}_m y\|_2$, and
    $e_1 = [1, 0, \ldots, 0]^{\mathrm{T}}$.
10. If satisfied stop
11. Set $x_0 \leftarrow x_m$, and $p$ to the number of eigenvalues to be used,
12. Calculate $p$ eigenvectors $u_1, \ldots, u_p$ of $AM^{-1}$, and go to 1.
```

The algorithm can be modified for left preconditioning by making the following three changes: set $r_0 = M^{-1}(b - Ax_0)$ in step 1, set $x_m = x_0 + W_m y_m$ in step 9, and replace AM^{-1} with $M^{-1}A$ in steps 4 and 12.

It can be seen that the vectors w_j in Algorithm 2.1, and the Arnoldi vectors v_j, $j = 1, \ldots, m$ are related by the equation,

$$AM^{-1}w_j = \sum_{i=1}^{j+1} h_{ij} v_i$$

In matrix form this becomes,

$$AM^{-1}W_m = V_{m+1}\bar{H}_m$$

where $W_m = [w_1, \ldots, w_m]$ is the $n \times m$ matrix whose column vectors are the w_i's, $i = 1, \ldots, m$. $V_{m+1} = [v_1, \ldots, v_{m+1}]$ is the $n \times (m+1)$ matrix whose column vectors are the v_i's, $i = 1, \ldots, m+1$, and $\bar{H}_m$ is an $(m+1) \times m$ matrix whose non-zero entries are the h_{ij}'s defined by the algorithm.

Now consider the residual vector for an arbitrary vector $z = x_0 + M^{-1}W_m y$ in the affine space $x_0 + \text{span}\{M^{-1}W_m\}$. We have

$$\begin{aligned} b - Az &= b - A(x_0 + M^{-1}W_m y) \\ &= r_0 - AM^{-1}W_m y \qquad (2.1) \\ &= \beta v_1 - V_{m+1}\bar{H}_m y \\ &= V_{m+1}[\beta e_1 - \bar{H}_m y] \qquad (2.2) \end{aligned}$$

Here $e_1 = [1, 0, \ldots, 0]^{\mathrm{T}}$. If we denote by $J_m(y)$ the function

$$J_m(y) = \|b - A[x_0 + M^{-1}W_m y]\|_2$$

we observe that by (2.2) and the fact that V_{m+1} is unitary, we have

$$J_m(y) = \|\beta e_1 - \bar{H}_m y\|_2 \qquad (2.3)$$

Since the algorithm minimizes this norm over all vectors y in $\mathbf{R}^m$ to yield y_m, it is clear that the approximate solution $x_m = x_0 + M^{-1}W_m y_m$ has the smallest residual norm in $x_0 + \text{span}\{M^{-1}W_m\}$.

Note that eigenvectors are calculated at step 12 of Algorithm 2.1 In the first pass through GMRES, no eigenvectors are available, and $p = 0$. For problems with multiple right-hand sides, or in non-linear problems, there may be estimates of eigenvalues available from a previous calculation, or a nearby problem. Convergence can be accelerated if these are used in the first pass through GMRES.

Algorithm 2.1 is similar to the Flexible GMRES (FGMRES) algorithm which is repeated below for reference purposes.

Algorithm 2.2. *Flexible GMRES*

```
Compute $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$ and $v_1 = r_0/\beta$.
For $j = 1, \ldots, m$ do
    Compute $z_j := M^{-1}v_j$;
    Compute $w := Az_j$;
    For $i = 1, \ldots, j$, do $\begin{cases} h_{i,j} := (w, v_i) \\ w := w - h_{i,j}v_i \end{cases}$
```

```
6.         Compute $h_{j+1,j} = \|w\|_2$ and $v_{j+1} = w/h_{j+1,j}$
7.    EndDo
8.    Define $V_{m+1} := [v_1, \ldots, v_{m+1}]$, $Z_m := [z_1, \ldots, z_m]$,
9.    Compute $x_m = x_0 + Z_m y_m$ where $y_m = argmin_y \|\beta e_1 - \bar{H}_m y\|_2$.
10.   If satisfied stop, else set $x_0 \leftarrow x_m$ and go to 1
```

If FGMRES or Algorithm 2.1 are compared with standard GMRES, it can be seen that they each keep a second basis, W_m for Algorithm 2.1, and Z_m for FGMRES. For Algorithm 2.1 V_{m+1} and W_m are related by $AM^{-1}W_m = V_{m+1}\bar{H}_m$. This allows for the calculation of the eigenvectors of the preconditioned matrix AM^{-1} using projection methods as shown in Section 2.1 below. These are the eigenvectors required in deflation. For FGMRES the bases are related by $AZ_m = V_{m+1}\bar{H}_m$, and the same projection methods yield the eigenvectors of A, which are not needed. In Algorithm 2.1 it is necessary to store only the vectors $w_j = u_{j-m+p}$ assigned in step 3, and not the complete base W_m. The vectors $w_j = v_j$ are stored in V_m. In FGMRES all vectors in Z_m must be stored. FGMRES has the advantage that the preconditioner can be varied at each step, and this cannot be done in Algorithm 2.1

2.1. Estimating eigenvectors

Step 12 of Algorithm 2.1 requires that the p eigenvectors corresponding to the eigenvalues closest to zero be estimated, and returned as the vectors $u_1, u_2, \ldots, u_p$. These are the eigenvectors which hamper convergence. For complex eigenvectors the real and imaginary parts of the eigenvector count as two vectors u_i, and u_{i+1}. To estimate the eigenvectors a projection technique is used onto the subspace span$\{W_m\}$ available at step 12. To this end we use the following relation, where B denotes the matrix $B = AM^{-1}$,

$$BW_m = V_{m+1}\bar{H}_m \tag{2.4}$$

The approximate eigenvectors u to be extracted from the subspace span$\{W_m\}$ can be expressed in the form

$$u = W_m y$$

where y is an m-dimensional vector. Let θ be an associated eigenvalue. In a standard orthogonal projection technique we would require that the residual $(B - \theta I)u$ be orthogonal to the span of W_m which gives us the problem,

$$W_m^{\mathrm{H}}(B - \theta I)W_m y = 0 \tag{2.5}$$

This requires solving the generalized eigenvalue problem,

$$W_m^{\mathrm{H}} B W_m y = \theta W_m^{\mathrm{H}} W_m y$$

A disadvantage to this approach is that the two matrices invoked in the above problem must be formed and this entails non-negligible additional computational work. The first alternative is to use the subspace spanned by V_m instead of W_m to impose the orthogonality condition. This leads to the *oblique* problem,

$$V_m^{\mathrm{H}}(B - \theta I)W_m y = 0 \tag{2.6}$$

or, exploiting the relation (2.4),

$$H_m y = \theta V_m^{\mathrm{H}} W_m y$$

where H_m is the Hessenberg matrix obtained from $\bar{H}_m$ by removing its last row. We now need only to compute one additional matrix namely $V_m^{\mathrm{H}} W_m$.

A third and better alternative is to use the base BW_m for the orthogonality condition as was suggested by Morgan [4,3]. This version of the Rayleigh–Ritz procedure does better at finding eigenvalues nearest zero than the standard projection procedures. The Galerkin condition for this approach gives the problem

$$(BW_m)^{\mathrm{H}}\,(BW_m y - \theta W_m y) = 0 \tag{2.7}$$

It is common to refer to this as a Galerkin procedure for approximating *harmonic* values since if we set $X = BW_m$, then the procedure is equivalent to the (orthogonal) Galerkin condition,

$$X^{\mathrm{T}}(\frac{1}{\theta} Xy - B^{-1}Xy) = 0$$

This means that the process can be viewed as a Galerkin method for approximating the eigenvalues of the inverse of B.

To simplify equation (2.7), let

$$F = (BW_m)^{\mathrm{H}} W_m\ , \qquad G = (BW_m)^{\mathrm{H}}(BW_m)$$

Then the Galerkin condition gives the generalized eigenvalue problem,

$$Gy = \theta F y \tag{2.8}$$

This problem will give us m eigenvalues θ_i, $i = 1, \ldots, m$ and associated eigenvectors y_i. Some of these eigenvalues will be good approximations to eigenvalues of B. The corresponding Ritz vectors given by $W_m y$ will approximate eigenvectors of B.

To calculate the matrices G and F, Morgan uses the identity (2.4). For the matrix G we get,

$$\begin{aligned} G &= W_m^{\mathrm{H}} B^{\mathrm{H}} B W_m \\ &= (V_{m+1}\bar{H}_m)^{\mathrm{H}} V_{m+1}\bar{H}_m \\ &= \bar{H}_m^{\mathrm{H}} V_{m+1}^{\mathrm{H}} V_{m+1} \bar{H}_m \\ &= \bar{H}_m^{\mathrm{H}} \bar{H}_m \end{aligned} \tag{2.9}$$

Similarly for the F matrix,

$$\begin{aligned} F &= (BW_m)^{\mathrm{H}} W_m \\ &= \bar{H}_m^{\mathrm{H}} V_{m+1}^{\mathrm{H}} W_m \end{aligned} \tag{2.10}$$

The generalized eigenvalue problem (2.8) can be solved by any technique for dense generalized eigenvalue problems. In our experiments, the QZ algorithm implemented in the `RGG` subroutine from EISPACK is used.

Another option is to exploit the results of the FGMRES calculation. At the end of the FGMRES process the matrix $\bar{H}_m$ is factored as

$$Q_m \bar{H}_m = \bar{R}_m$$

where Q_m is a unitary matrix of dimension $(m+1) \times (m+1)$ and $\bar{R}_m$ is an upper triangular matrix of size $(m+1) \times m$ whose last row is a zero row. Define R_m to be the matrix obtained from $\bar{R}_m$ by deleting its last row. We clearly have,

$$\bar{H}_m^{\mathrm{H}} \bar{H}_m = \bar{R}_m^{\mathrm{H}} \bar{R}_m = R_m^{\mathrm{H}} R_m$$

Consider now the matrix F and use the relation $\bar{H}_m = Q_m^{\mathrm{H}} \bar{R}_m$

$$\begin{aligned} F &= \bar{H}_m^{\mathrm{H}} V_{m+1}^{\mathrm{H}} W_m \\ &= \bar{R}_m^{\mathrm{H}} Q_m V_{m+1}^{\mathrm{H}} W_m \\ &= R_m^{\mathrm{H}} \lfloor Q_m V_{m+1}^{\mathrm{H}} W_m \rfloor \end{aligned} \tag{2.11}$$

where $\lfloor X \rfloor$ is used to denote the matrix obtained by removing the last row of the matrix X. Therefore the generalized eigenvalue obtained is of the form

$$R_m^{\mathrm{H}} R_m y = \theta R_m^{\mathrm{H}} \lfloor Q_m V_{m+1}^{\mathrm{H}} W_m \rfloor \; y$$

which has the same solutions as

$$R_m y = \theta \lfloor Q_m V_{m+1}^{\mathrm{H}} W_m \rfloor \; y \tag{2.12}$$

In the classical GMRES algorithm, the above problem is in the standard form required by the QZ algorithm since the matrix in the right-hand side is upper-Hessenberg. In this case, the transformation into the standard form required by the RGG routine is avoided. To obtain the matrix on the right-hand side of (2.12) in the general case, notice that the matrix $V_{m+1}^{\mathrm{H}} W_m$ must undergo the same rotations as those that transform $\bar{H}_m$ into R_m. These rotations are saved in the FGMRES algorithm. In terms of storage then, this approach will avoid keeping the original copy of $\bar{H}_m$ and a temporary array for the matrix G_m. It is also likely to be better behaved numerically.

3. Inner–outer FGMRES iteration

In FGMRES the preconditioning operation $z_j := M_j^{-1} v_j$ of step 3 can be thought of as a means of approximately solving $A z_j = v_j$ where M_j^{-1} is the preconditioner $M_j^{-1} \approx A^{-1}$. This is referred to as an inner iteration. An inner iteration for Algorithm 2.1 is given by replacing line 3 by

- Set $w_j = \begin{cases} v_j & \text{if } j \le m - p \\ \text{approximate solution of } AM^{-1} w_j = v_j & \text{otherwise} \end{cases}$

The solution x then belongs to the subspace

$$x_0 + \operatorname{span}\{M^{-1}(r_0, AM^{-1} r_0, \ldots, (AM^{-1})^{m-p-1} r_0, \mu_1, \mu_2, \ldots, \mu_p)\}$$

Here μ_1 is the approximate solution of $AM^{-1}\mu_1 = v_{m-p}$. The vectors μ_i, where $i > 1$, are the approximate solutions of $AM^{-1}\mu_i = \mu_{i-1}$. Note that the approximate solve with AM^{-1} replaces the approximate solve with A of FGMRES. If the inner system is solved exactly, say at step m, then

$$AM^{-1}w_m = v_m$$

so the last column of $AM^{-1}W_m$ in (2.4) is a linear combination of the v_i's for $i = 1, \ldots, m$. Therefore, the matrix $\bar{H}_m$ has its last row equal to zero. As a result there exists a vector y such that $\|\beta e_1 - \bar{H}_m y\| = 0$, provided the matrix H_m obtained by removing the last row of $\bar{H}_m$ is non-singular. In this situation, the minimum of the residual as expressed by (2.2) is zero and the solution is exact at the next outer iteration.

For the inner iteration any method can be used, though for practical purposes it should be a fast approximate solve. In this work standard GMRES is used, terminating when a reduction of less than 0.1 in the residual norm is achieved or 20 steps have been taken.

There are two ways to justify the inner–outer iteration approach. One is simply the argument given above that introducing a vector of the form $B^{-1}v_i$ to the subspace would solve the linear system exactly at the next outer iteration, so introducing a vector approximately equal to $B^{-1}v_i$ should speed up the outer iteration. Second, the steps $m - p$ to m can be viewed as a form of (projected) inverse iteration, in which the systems in the inverse iteration procedure are solved only approximately. The behavior of inner iteration can be illustrated by graphs showing the residual norm versus the number of GMRES steps (shown for example in Figures 1 and 2 in the results section). For indefinite problems there is often an initial rapid decrease in residual norm, followed by stalling. The aim is to take a small number of inner iteration steps which correspond to the initial rapid decrease in the residual norm. The inner iteration is then terminated as any extra run-time is better spent on the outer iteration.

4. Block-GMRES algorithms

In block-Krylov subspace methods, the subspace of approximants is defined from a set or block of initial vectors. In order to simplify the notation we will assume at first that the block size is only three. Let us take the initial block to be

$$V = [v^1, v^2, v^3]$$

Then the m-th Block-Krylov subspace is defined as

$$\text{span}\,\{V, AV, \ldots, A^{m-1}V\} \tag{4.1}$$

Note that the dimension of the subspace here is $m \times p$, where p is the block dimension. A block-Arnoldi generalization of the Arnoldi process can easily be derived, and a block GMRES algorithm can be defined from it. The standard block GMRES with multiple right-hand sides is described in [10]. This algorithm is outlined briefly below, together with two other implementations.

4.1. Standard block GMRES with multiple right-hand sides

In standard block GMRES with multiple right-hand sides the goal is to solve the linear systems,

$$Ax^{(i)} = b^{(i)}, \quad i = 1, \cdots, p \tag{4.2}$$

or in matrix form,

$$AX = B \tag{4.3}$$

where the columns of the $n \times p$ matrices B and X are the $b^{(i)}$'s and $x^{(i)}$'s respectively. Given a block of initial guesses $x_0^{(i)}$ for $i = 1, \ldots, p$, we define R_0 the block of initial residuals,

$$R_0 \equiv [r_0^{(1)}, r_0^{(2)}, \ldots, r_0^{(p)}]$$

where each column is $r_0^{(i)} = b^{(i)} - Ax_0^{(i)}$. The first step of the block-GMRES algorithm is to compute the QR factorization of the block of initial residuals:

$$R_0 = [v_1, v_2, \ldots, v_p]\, R$$

Here the matrix $[v_1, \ldots, v_p]$ is unitary and R is $p \times p$ upper-triangular. This factorization provides the first p vectors of the block-Arnoldi basis.

Each of the approximate solutions has the form,

$$x^{(i)} = x_0^{(i)} + V_m y^{(i)} \tag{4.4}$$

and, grouping these approximations $x_i^{(i)}$ in a block X, and the $y_i^{(i)}$ in a block Y, we can write,

$$X = X_0 + V_m Y \tag{4.5}$$

The block Arnoldi procedure produces the Krylov base V_{m+p}, and the $(m + p) \times m$ matrix $\bar{H}_m$, related by

$$AV_m = V_{m+p}\bar{H}_m \tag{4.6}$$

Let E_1 be the $(m + p) \times p$ matrix whose upper $p \times p$ principal block is an identity matrix. Then, the relation (4.6) results in

$$\begin{aligned} B - AX &= B - A\,(X_0 + V_m Y) \\ &= R_0 - AV_m Y \\ &= [v_1, \ldots, v_p]R - V_{m+p}\bar{H}_m Y \\ &= V_{m+p}\left(E_1 R - \bar{H}_m Y\right) \end{aligned} \tag{4.7}$$

The block-GMRES approximation $x^{(i)}$ is the unique vector of the form $x_0^{(i)} + V_m y^{(i)}$ which minimizes the 2-norm of the individual columns of the block-residual (4.7). By the orthogonality of the column-vectors of V_{m+p} we get, from (4.7)

$$\|b^{(i)} - Ax^{(i)}\|_2 = \|g^{(i)} - \bar{H}_m y^{(i)}\|_2 \tag{4.8}$$

Here $\bar{g}^{(i)} \equiv E_1 Re_i$ is a vector of length $m + p$ whose components are zero except those from 1 to i which are extracted from the ith column of the upper triangular matrix R. Thus, the function on the right-hand side must be minimized over $y^{(i)}$.

The resulting least-squares problem is similar to the one encountered for GMRES. However, the matrix $\bar{H}_m$ has p entries below the diagonal in place of 1. The usual way of solving the above least-squares problems is to use plane rotations [9]. Now p rotations are needed at each step j, instead of only one, to eliminate the elements below the main diagonal in column j of H_m. For additional details see [8].

4.2. *Separate bases implementation*

The block-Arnoldi algorithm can also be viewed from a different angle by rewriting the block-Krylov subspace as

$$\text{span}\,\{v^1, Av^1, \ldots, A^{m-1}v^1,\ v^2, Av^2, \ldots, A^{m-1}v^2,\ v^3, Av^3, \ldots, A^{m-1}v^3\} \tag{4.9}$$

Instead of trying to use the three vectors simultaneously, as is done in the standard block Arnoldi algorithm, we can generate the Krylov subspace associated with the first vector, then the subspace associated with the second vector and then the third. The result is p *independent* Arnoldi bases. Put together these bases give the system of $m.p$ vectors,

$$V_{m\times p} \;=\; \text{span}\,\{v_1^1, v_2^1, \ldots, v_m^1,\ v_1^2, v_2^2, \ldots, v_m^2,\ v_1^3, v_2^3, \ldots, v_m^3\} \tag{4.10}$$

which is no longer orthonormal but still satisfies a relation of the form

$$AV_{m\times p} = V_{(m+1)\times p}\bar{H}_{m\times p}$$

Here $\bar{H}_{m\times p}$ is a matrix of $(m+1)p$ rows and mp columns. For $p = 3$ it has the form,

$$\bar{H}_{m\times 3} \;=\; \begin{pmatrix} \bar{H}_m^1 & & \\ & \bar{H}_m^2 & \\ & & \bar{H}_m^3 \end{pmatrix}$$

where each of $\bar{H}_m^i$ blocks is an $(m+1)\times m$ matrix obtained from the Arnoldi process for the vector v_1^i. To implement a Block-GMRES solution technique based on this procedure, it is necessary to orthonormalize the columns vectors of $V_{(m+1)\times p}$ using modified Gram–Schmidt or a Householder orthogonalization, to get the QR factorization,

$$V_{(m+1)\times p} = Q\;R$$

where Q is unitary and of size $n\times((m+1)p)$ while R is a square upper-triangular and of dimension $(m+1)\times p$. For modified Gram–Schmidt the operational count is $2((m+1)p)^2 n$, and for Householder orthogonalization it is $4((m+1)*p)^2 n - 4/3((m+1)p)^3$. Once this is done, we would have the relation,

$$\begin{aligned} b - Ax &= b - A[x_0 + V_{m\times p}y] \\ &= \beta v_1^1 - AV_{m\times p}y \\ &= V_{(m+1)\times p}[\beta e_1 - H_{m\times p}y] \\ &= Q[\beta Re_1 - RH_{m\times p}y] \end{aligned}$$

Since Q is unitary the residual norm is minimized by solving the least-squares problem

$$\min_y \|r_{11}\beta e_1 - RH_{m\times p}y\|_2$$

which is similar to the one encountered for the standard GMRES algorithm. This alternative implementation has the drawback that we need to orthonormalize $(m+1)\times p$ vectors, in addition to the local orthonormalization required for each separate Arnoldi basis. The numerical properties of the new basis are also not understood. There are advantages as well. First, the new vectors v_1^2, v_1^3, ... need not be known in advance. There are many applications, where these right-hand sides depend on the previous systems being solved. In these cases, the normalized initial v_1^2, v_1^3, residuals for the other right-hand sides are only available in sequence. A second advantage is the coarse grain parallelism of this approach. The p Arnoldi bases can be generated independently. This is far more than the degree of parallelism allowed at each step of a standard Block Arnoldi procedure.

The standard and separate bases implementations of GMRES are mathematically equivalent. There is another version of block GMRES which is similar to the separate Arnoldi bases strategy and which can be derived from FGMRES. This is *not equivalent* to the standard Block-GMRES algorithm. It is given below.

4.3. FGMRES version

Consider the basis (4.10). At step $m+1$ a new basis vector, namely v_1^2 which is not related in any way to the previous v_j^1's is introduced to the basis. It is possible to use FGMRES or Algorithm 2.1 at this point and to consider this to be a new vector injected to the subspace. Then, v_1^2 is orthogonalized against $v_1^1,\ldots,v_m^1$. The resulting vector is then multiplied by A and orthonormalized against all previous v_i's. At step $2m+1$ a fresh vector, namely v_1^3 is introduced to the subspace and the process continued.

The minimization problem in line 9 is replaced by $y_m^{(i)} = \text{argmin}_y\|r_0^{(i)} - V_{m+k}\bar{H}_m y\|_2$. This is a $n\times m$ least squares problem, where n is the coefficient matrix size, and m is the Krylov subspace size.

In this variant the final basis is orthonormal, so it does not suffer from the weakness of the separate bases approach. It does not however offer the same coarse grain parallelism, because the individual blocks are orthogonalized against previous blocks and must be generated sequentially. The two methods are not mathematically equivalent because the FGMRES implementation produces a mixing of the vectors of the basis which are then multiplied by A successively. For instance, once Av_1^2 is orthonormalized against the previous v_j^1's in step $m+1$, the next basis vector v_2^2 combines the vectors Av_1^2 with $v_1^1, v_2^1,\ldots,v_m^1$:

$$h_{m+2,m+1}v_2^2 = Av_1^2 - \sum_{i=1}^{m+1} h_{i,m+1}v_i^1$$

In the next step the above vector is multiplied by A before being orthonormalized against previous v_i's. Thus, a component from Av_1^1 will be introduced. This component will become $A^i v_1^1$ in subsequent steps. At the last step, the basis will contain a component of the vector $A^{(p-1)m}v_1^1$, which obviously is not in the basis (4.9).

Regardless of implementation, there are several ways in which a block-GMRES procedure can be exploited. First, is the standard case of multiple right-hand sides as described above.

A second use of block-Arnoldi is to inject approximate eigenvectors. For example, we can take a block-size of two, the second Krylov subspace being started with the approximate eigenvector associated with the approximate eigenvalue nearest zero. Another option would be to use inner iteration approximate solutions as new vectors.

5. Theoretical considerations

In this section we analyze the effect of adding a nearly invariant subspace to a given Krylov subspace in the FGMRES process. In fact the analysis is not restricted to Krylov subspaces. Instead, we assume that a projection method is applied with a subspace S augmented by a certain 'nearly invariant' subspace $\mathcal{W}$. Two cases are compared. First we discuss a standard 'orthogonal error' method which consists of seeking an approximate solution from the affine subspace $x_0 + K$ where

$$K = S + \mathcal{W} \tag{5.1}$$

whose residual vector is orthogonal to K. Second a minimal residual (min-res) method, such as the FGMRES algorithm, which minimizes the residual vector $b - Ax$ over all x vectors from the subspace $x_0 + K$. With respect to the first approach, the following result can be stated.

Proposition 5.1. *Let $W = [w_1, \ldots, w_p]$ an $n \times p$ matrix whose column vectors form an orthonormal basis of an approximate invariant subspace $\mathcal{W}$. Assume that we have the relation,*

$$AW = WR + E \tag{5.2}$$

where R is non-singular and $W^{\mathrm{H}}E = 0$. Let the approximate solution $\tilde{x}$ obtained from an orthogonal projection process onto K, be written as

$$\tilde{x} = x_S + w$$

where $x_S \in x_0 + S$ and $w \in \mathcal{W}$. Then the residual vector $\tilde{r} = b - A\tilde{x}$, is such that

$$\tilde{r} = (I - WW^{\mathrm{H}})(I - ER^{-1}W^{\mathrm{H}})r_S \tag{5.3}$$

where $r_S \equiv b - Ax_S$.

Proof The residual vector $\tilde{r}$ is such that,

$$\tilde{r} = r_S - AWy$$

The orthogonality condition implies in particular that

$$W^{\mathrm{H}}(r_S - AWy) = 0\ .$$

Exploiting (5.2) and the orthogonality of E and W we clearly have $W^{\mathrm{H}}AW = R$ which results in the equality,

$$y = R^{-1}W^{\mathrm{H}}r_S$$

and

$$\tilde{r} = r_S - AWR^{-1}W^{\mathrm{H}}r_S = r_S - (WR + E)R^{-1}W^{\mathrm{H}}r_S = (I - WW^{\mathrm{H}})r_S - ER^{-1}W^{\mathrm{H}}r_S$$

The relation (5.3) follows by observing that $(I - WW^{\mathrm{H}})E = E$. ■

Observe that when $E = 0$, the process eliminates any components of the residual vector from the (exact) invariant subspace. When E is only approximate, the process also cancels all components of the residual in this approximate invariant subspace but now the components in the orthogonal space may be amplified. This is especially true if R is nearly singular.

The vector x_S in the proposition is not explicitly known. In particular, it is not the result of a projection process onto S. However, it is possible to state a more specific result if additional assumptions are made on the subspace $\mathcal{W}$.

Proposition 5.2. *Let $[w_1, \ldots, w_p, v_1, \ldots, v_m]$ an orthogonal basis of the subspace K where $W = [w_1, \ldots, w_p]$ is an orthonormal basis of the subspace $\mathcal{W}$. Define $V = [v_1, \ldots, v_m]$ and assume that relation (5.2) holds with the conditions that R is non-singular and,*

$$W^{\mathrm{H}}E = V^{\mathrm{H}}E = 0$$

Let the approximte solution $\tilde{x}$ obtained from an orthogonal projection process onto K, be written as

$$\tilde{x} = x_V + w$$

where $x_V \in x_0 + span\{V\}$ and $w \in \mathcal{W}$ and define $r_V \equiv b - Ax_V$. Then, the following properties hold.

1. *The vector x_V is the result of an orthogonal projection process onto* span$\{V\}$ *with the starting vector x_0.*
2. *The residual vector $\tilde{r} = b - A\tilde{x}$, is such that*

$$\tilde{r} = (I - WW^{\mathrm{H}})(I - ER^{-1}W^{\mathrm{H}})r_V \tag{5.4}$$

Proof To prove the first part, set $w = Wy$ and observe that the orthogonal projection method imposes the following conditions on the residual $\tilde{r} = r_V - AWy$:

$$V^{\mathrm{H}}(r_V - AWy) = 0; \quad W^{\mathrm{H}}(r_V - AWy) = 0$$

Since $AW = WR + E$ and by the assumptions, we have $V^{\mathrm{H}}AW = V^{\mathrm{H}}(WR + E) = 0$, so the conditions imply that $V^{\mathrm{H}}r_V = 0$. This proves that the residual $b - Ax_V$ is orthogonal to span$\{V\}$, which establishes the desired result. The proof of the second part of the result is identical with that of the previous proposition, except that x_S and r_S are replaced by x_V and r_V respectively. ■

We now consider the analogous situation obtained when a min-res approach is used. As is well known [8], in a min-res projection method onto an arbitrary subspace K, the initial residual vector is projected out of its components in the suspace AK. More precisely, if P denotes the orthogonal projector onto the subspace spanned by AK, and r_0 is the initial residual $b - Ax_0$ then the residual vector $\tilde{r}$ obtained from a minimal residual method onto K, with the starting vector x_0, is such that,

$$\tilde{r} = (I - P)r_0 \tag{5.5}$$

Equivalently, if the initial error $d_0 = x_* - x_0$ is written as

$$d_0 = v + q, \;\; v \in K, \quad Aq \perp AK \tag{5.6}$$

then $\tilde{r} = (I - P)r_0 = (I - P)Ad_0 = (I - P)(Av + Aq) = Aq$ so the error $\tilde{d} = x_* - \tilde{x}$ after the projection process is

$$\tilde{d} = q$$

Another way of expressing the above result is that $\tilde{x}$ minimizes the 2-norm of the residual of $b - Ax$ for all x in $x_0 + K$. Writing x in the form $x = z + w$ where z belongs to the subspace $x_0 + S$ and $w \in \mathcal{W}$, the optimality result yields,

$$\|\tilde{r}\| = \min_{z,w} \|b - Az - Aw\|_2$$

Consider now the vector $\tilde{z}$ obtained by a min-res projection onto $x_0 + S$, i.e., the vector of $x_0 + S$ which minimizes $\|b - Az\|_2$ and call $r_{\tilde{z}}$ its associated residual vector $r_{\tilde{z}} = b - A\tilde{z}$. From the above equality, we can obviously write

$$\|\tilde{r}\| \le \min_{w} \|b - A\tilde{z} - Aw\|_2 = \min_{w} \|r_{\tilde{z}} - Aw\|_2$$

Recall that

$$\min_{w} \|r_{\tilde{z}} - Aw\|_2 = \|(I - P_{AW})r_{\tilde{z}}\|_2$$

in which P_{AW} is the orthogonal projector onto the subspace $A\mathcal{W}$. Therefore, we can state the following result.

Proposition 5.3. *Let $r_{\tilde{z}}$ be the residual obtained from a min-res projection process onto the subspace S, with starting vector x_0, and let P_{AW} be the orthogonal projector onto the subspace $A\mathcal{W}$. Then, the residual $\tilde{r}$ obtained by a min-res projection process onto $K = S + \mathcal{W}$, with starting vector x_0, satisfies the inequality*

$$\|\tilde{r}\|_2 \le \|(I - P_{AW})r_{\tilde{z}}\|_2$$

From a similar argument to the one above, it is seen that $I - P_{AW}$ projects out the components of W from the error $d_{\tilde{z}} \equiv x_* - \tilde{z}$. The residual obtained from a min-res approach on S is smaller than the residual resulting from this purification. Note that we have not made any assumptions regarding the fact that $\mathcal{W}$ is nearly invariant. If $\mathcal{W}$ is nearly invariant under A then so is $A\mathcal{W}$ since $AW = WR + E$ implies $A(AW) = (AW)R + AE$. In this situation, the orthogonal projector onto AW is close to an orthogonal projector onto $\mathcal{W}$.

6. Numerical experiments

All experiments have been carried out on the matrices listed in Table 1. The matrix VENKAT25 arises from an unstructured Euler solver while BBMAT arises from a Beam and Bailey 2D airfoil problem[1]. BARTHT2A and WIGTO966 were supplied by Tim Barth

[1] These matrices are available through the WWW. See the home pages of the second author at: `http://www.cs.umn.edu`. VENKAT25 and BBMAT and in the 'Simon' collection and have been contributed by H. Simon.

Table 1. Matrices used in tests

Matrix	sym	n	nnz	Block size		Inner	Krylov
name				Matrix	precon.	iter.	size *m*
VENKAT25	ns	62 424	1 717 792	4	4	20	50
BBMAT	ns	38 744	1 771 722	4	16	20	50
BARTHT2A	ns	14 075	1 311 725	5	5	20	50
WIGTO966	ns	3 864	238 252	4	8	20	50
SAYLR4	s	3 564	22 316	—	11	20	50
SHERMAN5	ns	3 312	20 793	—	3	10	25
EX3	s	1 821	52 685	—	4	10	25
EX7	ns	1 633	54 543	—	4	10	25
EX9	s	3 363	99 471	—	4	10	25

of NASA Ames and Larry Wigton of Boeing, respectively. BARTHT1A is from a 2D, high Reynolds number, airfoil problem, with turbulence modeling. WIGTO966 is for an Euler equation model. The matrices SAYLR4, and SHERMAN5 are from the Harwell-Boeing sparse matrix collection. EX3, EX7, and EX9 are from the fluid dynamics package FIDAP, which is used to model the Navier–Stokes equations by the finite element method. Three of the matrices are symmetric, and the remainder are non-symmetric. This is identified in column 2 of Table 1. Four of the matrices have block structure, and the block size is given in column 5 of Table 1. The columns labeled n and nnz are for matrix dimension and number of non-zeros.

In the experiments the matrices were row then column scaled using the 2-norm, and the resulting systems were solved using GMRES with deflated and augmented Krylov subspace techniques. Results are given in terms of the reduction in residual norm versus the number of GMRES steps. A right-hand side is provided with the matrix BBMAT. It is used in the tests, and the initial guess is set to be a vector of all zeros. For all other matrices the right-hand side is constructed so that the solution is a vector of all ones, and the initial guess is taken to be a vector with entries having random values between 0 and 1.

Two preconditioners were used, block diagonal (BDIAG), and block symmetric successive over relaxation (BSSOR). These are extensions of diagonal and SSOR preconditioning, see for example [8]. The block sizes used for BDIAG and BSSOR preconditioning are listed in column 6 of Table 1. These were chosen based on information on matrix structure, and limited testing. For example, a block size of 16 was used for BBMAT, because this produced significantly better results than using the matrix block size of 4. For the matrices that do not have block structure, preconditioner block size is in some cases not a multiple of matrix size. Here the remaining rows and columns are treated as a single block. A single iteration was used in BSSOR preconditioning, and the relaxation parameter was set to 0.5.

6.1. Deflation and inner iteration

For the deflation and inner iteration tests three subspaces are used. The first is the *standard* Krylov subspace

$$\mathrm{span}\{r_0, Br_0, \ldots, B^{m-1}r_0\} \tag{6.1}$$

Here r_0 is the initial residual $r_0 = b - Ax_0$, and B is the right preconditioned multiplying matrix $B = AM^{-1}$. The *deflated* subspace is given by

$$\mathrm{span}\{r_0, Br_0, \ldots, B^{(m-p)-1}r_0, u_1, u_2, \ldots, u_p\} \tag{6.2}$$

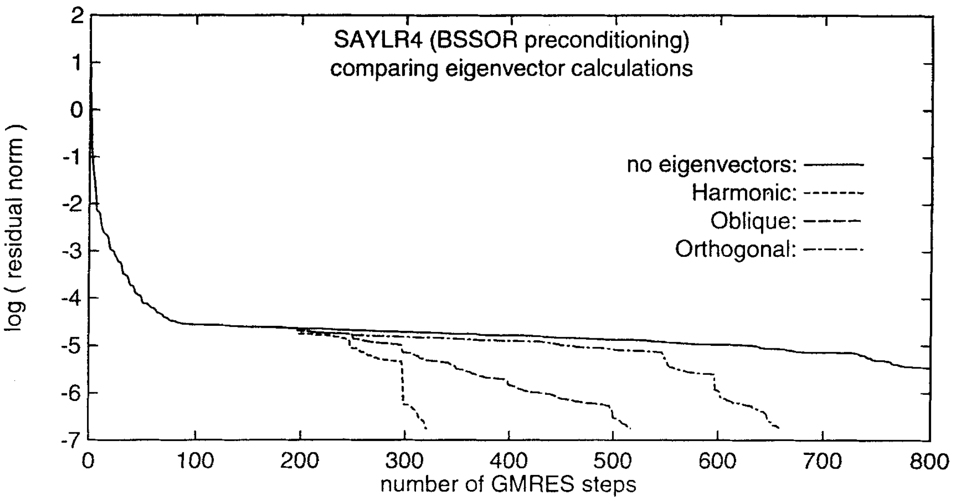


Figure 1. Convergence results for three different methods for calculating eigenvectors. The figure includes results for no added eigenvectors as a base case. In the tests BSSOR preconditioned GMRES is used with a subspace size of 50, including four eigenvectors

Here $u_1, u_2, \ldots, u_p$ are p estimated eigenvectors of B, corresponding to the p eigenvalues closest to zero. There are three methods mentioned in Section 2.1 for estimating eigenvectors, *orthogonal* projection, *oblique* projection and *harmonic* projection, as defined in equations (2.5), (2.6) and (2.7). They are compared for solving SAYLR4 with BSSOR preconditioning in Figure 1. The *harmonic* method was found to give the best results and it is used in all tests.

The inner iteration subspace is given by

$$\text{span}\{r_0, Br_0, \ldots, B^{(m-p)-1}r_0, \mu_1, \mu_2, \ldots, \mu_p\} \tag{6.3}$$

The inner iteration vectors μ were calculated using GMRES, with termination if the residual norm is reduced by more than 0.1. Table 1 gives the maximum number of inner iteration steps allowed for each matrix, 20 steps were allowed for larger matrices, and 10 for the smaller matrices.

The parameter m in the subspaces of equations (6.1), (6.2), and (6.3) is the total number of vectors in the subspace. For larger matrices $m = 50$ was used, and for smaller matrices $m = 25$. This is identified in column 8 of Table 1. The parameter p is the number of eigenvectors or inner iteration vectors. For all matrices $p = 4$ was used.

Convergence results for the three subspaces are given in Table 2. Figures 2 and 3 show the reduction in residual norm as GMRES progresses for matrices WIGTO966 and SAYLR4. To illustrate the different behavior of residual vectors and error vectors we also show in Figure 4 the plot of error norms as GMRES progresses for the matrix SAYLR4. The error norm is given by $\|x - \bar{x}\|_2$, where $\bar{x}$ is the true solution. Timing results are given in Table 3.

In Table 2 it can be seen that convergence for the deflated and inner iteration subspaces is always better than for standard GMRES when measured in terms of the number of outer GMRES steps. Table 3 shows that this is not always true when measured in terms of runtime.

Table 2. Convergence results for GMRES with three different preconditioners and three subspaces. Numbers give the reduction in residual norm in 1 800 iterations, or in parenthesis, the number of GMRES steps to reduce the residual norm by 10^{-8}

Matrix	Subspace	Preconditioner		
		None	BDIAG	BSSOR
VENKAT25	Standard	0.15E-1	0.86E-5	(1136)
	Deflation	0.15E-2	0.85E-6	(994)
	Inner iter.	0.58E-4	(1148)	(399)
BBMAT	Standard	0.78E-2	0.75E-2	0.66E-2
	Deflation	0.96E-2	0.14E-2	0.93E-5
	Inner iter.	0.79E-2	0.70E-2	(1200)
BARTHT2A	Standard	0.43E-2	0.24E-5	(729)
	Deflation	0.37E-3	0.50E-6	(602)
	Inner iter.	0.11E-4	(1047)	(300)
WIGTO966	Standard	0.47E-1	0.28E-1	(431)
	Deflation	0.16E-5	0.12E-4	(248)
	Inner iter.	0.20E-1	0.28E-1	(245)
SAYLR4	Standard	0.43E-6	0.11E-6	(1475)
	Deflation	0.16E-6	(579)	(321)
	Inner iter.	0.30E-6	(1144)	(297)
SHERMAN5	Standard	(786)	(235)	(107)
	Deflation	(288)	(110)	(72)
	Inner iter.	(263)	(72)	(33)
EX3	Standard	0.31E-6	0.34E-6	0.13E-6
	Deflation	0.37E-7	0.10E-6	0.57E-7
	Inner iter.	0.97E-7	0.98E-7	(400)
EX7	Standard	0.62E-7	0.74E-7	0.68E-7
	Deflation	(975)	0.22E-7	0.12E-7
	Inner iter.	0.53E-7	0.34E-7	(249)
EX9	Standard	(645)	0.13E-7	0.15E-7
	Deflation	(231)	(998)	(124)
	Inner iter.	(181)	0.13E-7	(85)

Table 3. Timing results for GMRES with BSSOR preconditioning, and three subspaces. Numbers give the time in seconds for each run on a single processor of a CRAY Y-MP C90. In each case the residual norm is reduced by 10^{-8}

Matrix	Subspace		
	Standard	Deflation	Inner iter.
VENKAT25	490	430	436
BARTHT2A	112	95.7	121
WIGTO966	13.1	7.98	17.3
SAYLR4	26.4	6.24	12.8
SHERMAN5	2.46	1.71	1.68

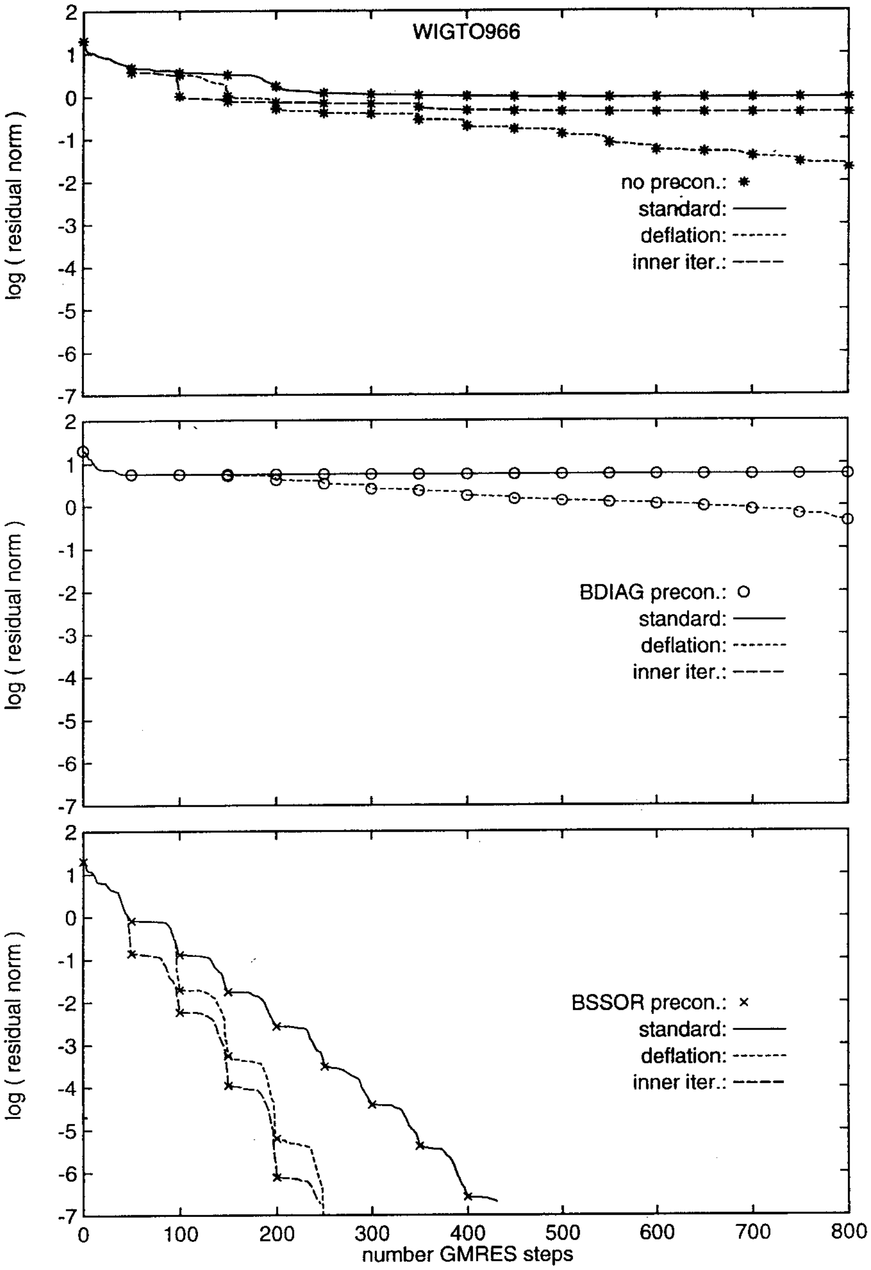


Figure 2. Reduction in residual norm for matrix WIGTO966 solved using GMRES with three different preconditioners and three different subspaces. Markers $*$, o and x are at the first GMRES step of each restart

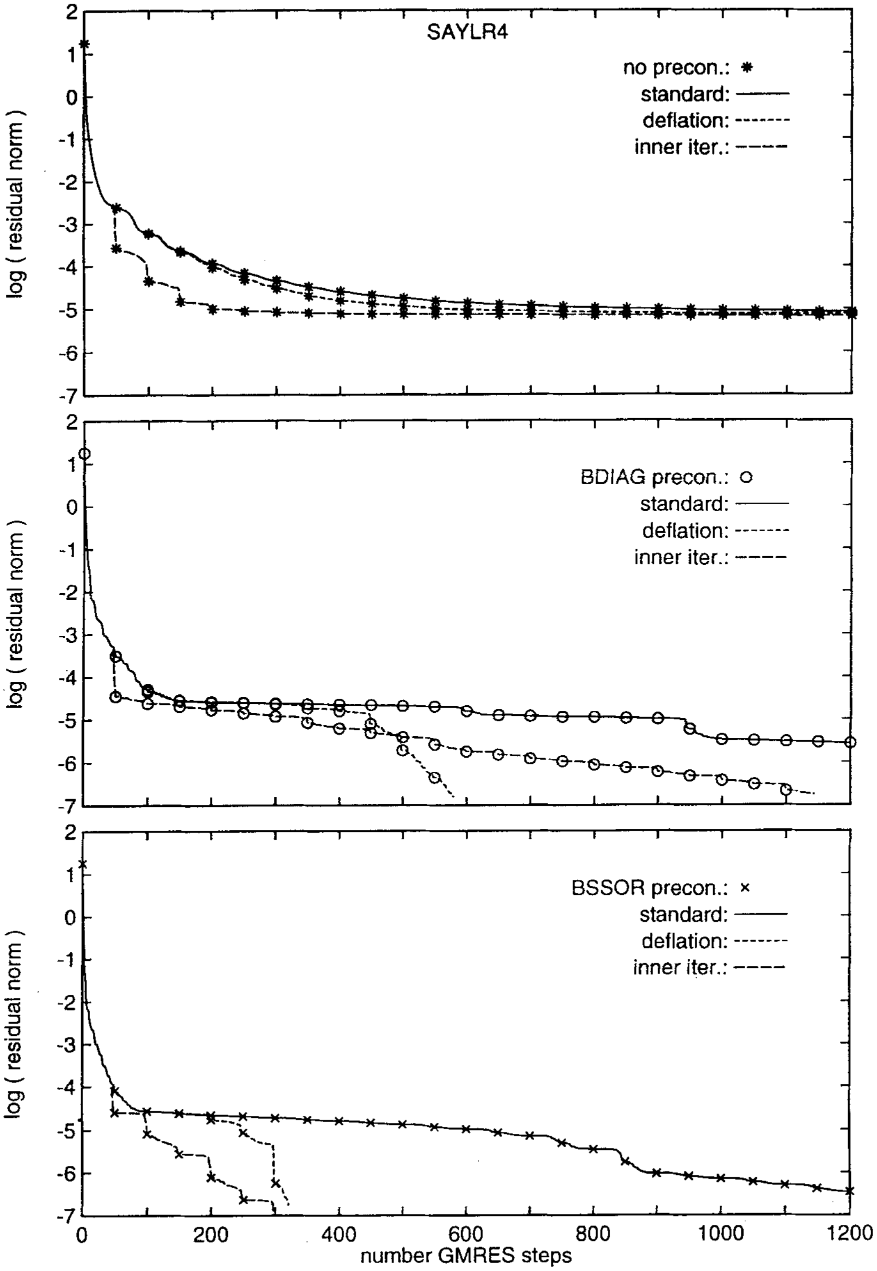


Figure 3. Reduction in residual norm for matrix SAYLR4 solved using GMRES with three different preconditioners and three different subspaces. Markers $*$, o and x are at the first GMRES step of each restart

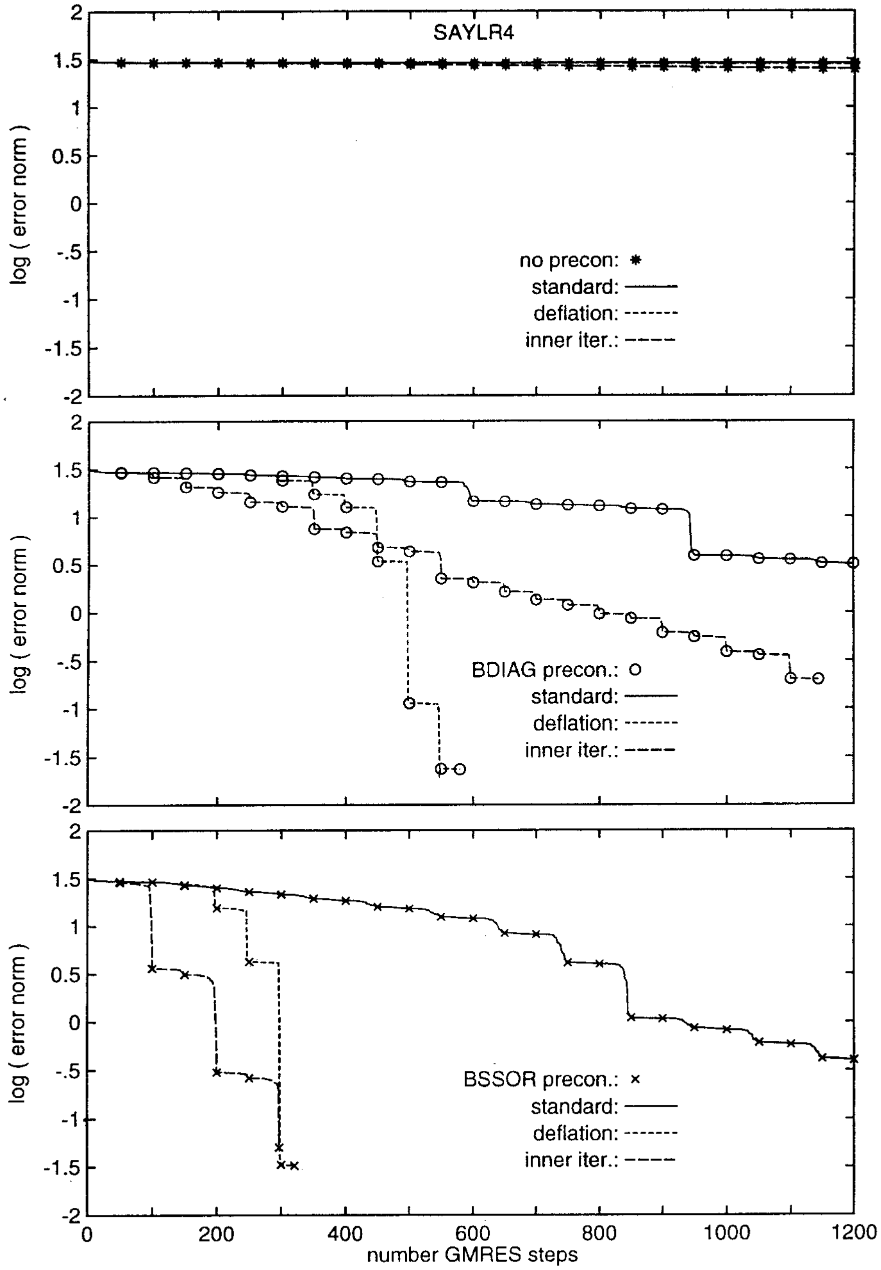


Figure 4. Reduction in the error norm for matrix SAYLR4 solved using GMRES with three different preconditioners and three different subspaces. Markers $*$, o and x are at the first GMRES step of each restart

This is due to the extra work involved in calculating eigenvectors, or in approximately solving the inner iteration. For deflation, an $m \times m$ generalized eigenvalue/vector problem is solved to calculate the y vectors, and a linear combination of m basis vectors are required to obtain each eigenvector. By contrast, the inner iteration requires the approximate solution of a $n \times n$ linear system, where n is the size of the coefficient matrix. For the larger matrices, $4 \times 20 = 80$ inner GMRES steps are used for every 50 outer GMRES steps, and for the smaller matrices, $4 \times 10 = 40$ inner GMRES steps were used for every 25 outer GMRES steps. The inner iteration procedure takes more time than deflation, and this is reflected in Table 3. Figure 3 shows cases where an initial decrease in the residual norm is followed by stagnation. This is referred to as stalling, and it sometimes occurs for indefinite matrices, that is, when the symmetric part of the matrix is not positive definite. Figure 3 shows that for the BSSOR and BDIAG preconditioned problems, deflation and inner iteration break the stalling behavior, and this results in convergence.

It can be seen in Figure 2 for the matrix WIGTO966 with BSSOR preconditioning that deflation is effective in speeding up convergence early in the solution process. Eigenvectors are first calculated at GMRES step 50, and first added at steps 97, 98, 99, and 100. They are effective from the first time that they are added. For the matrix SAYLR4 on the other hand, Figure 3 shows that deflation is effective later in the solution process, at step 450 for BDIAG preconditioning, and at step 250 for BSSOR preconditioning. This brings up the question of predicting when to add eigenvectors, and how many. Morgan discusses this issue in [4] (Section 5), and gives three tests, based on the reduction in residual norm that occurs when eigenvectors are added, and the accuracy of the estimated eigenvectors. The tests are shown to work for some problems, but not for others.

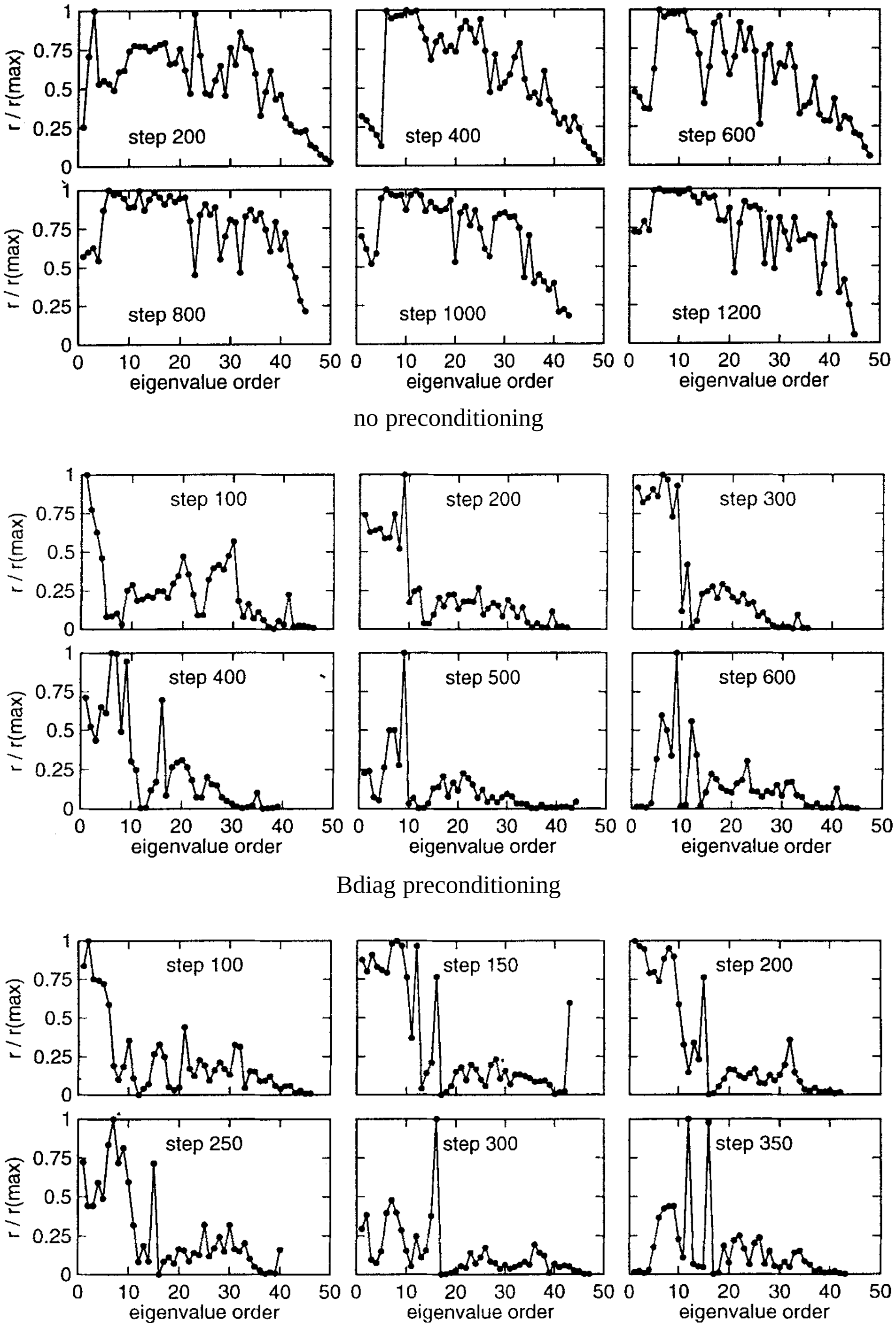
Another heuristic test was found to give an indication when adding eigenvectors will be effective. It is based on the residual norms for the estimated eigenvectors. Recall that in the projection method the number of eigenvectors estimated is equal to the dimension m of the augmented Krylov subspace (6.2), and p eigenvectors corresponding to the eigenvalues closest to zero are used in deflation. The test involves comparing the residual in the eigenvectors used in deflation with the maximum residual for all the estimated eigenvectors. The residual norm is given by

$$r = \frac{\|Bu - \theta u\|_2}{\|Bu\|_2 + \|\theta u\|_2}$$

Figure 5 shows three sets of plots of the normalized residuals $r/r_{\max}$, where $r_{\max}$ is the largest among the m residual norms. This is done for solving SAYLR4 with deflation in the three cases of no preconditioning, BDIAG preconditioning and BSSOR preconditioning. In the plots the x axis is for eigenvalue order from smallest (in modulus) to largest. The eigenvectors used in deflation correspond to the x axis co-ordinates 1, 2, 3 and 4. The step number given in the plots is the GMRES step at which the eigenvectors are added.

To interpret the plots, it is necessary to also look at Figure 3. There it can be seen that deflation is not effective for the case of no preconditioning. The $r/r_{\max}$ plots for no preconditioning show that the residual in the eigenvectors used in deflation is never small. For BDIAG preconditioning, deflation is effective at step 500, and here $r/r_{\max}$ for x co-ordinate 1, 2, 3 and 4 is small. The same occurs for BSSOR preconditioning. This behavior observed for SAYLR4 also occurs for the other matrices. Therefore, it seems that when $r/r_{\max}$ is *small* for the eigenvectors used in deflation, deflation will be effective.

Note that *small* is not quantified and may depend on the problem. Any adaptive procedure for deciding whether to add eigenvectors, and how many to add, will probably only work



no preconditioning

Bdiag preconditioning

BSSOR preconditioning

Figure 5. Residual for estimated eigenvectors of matrix SAYLR4. Eigenvalue order is from smallest to largest. The step number at which the eigenvectors are added is given

Table 4. Recycling eigenvectors for multiple RHS

Matrix	Number of iterations to converge for successive RHS				
BARTHT2A	(602)	(597)	(697)	(641)	(626)
WIGTO966	(248)	(214)	(236)	(229)	(217)
SAYLR4	(321)	(147)	(154)	(160)	(162)

Table 5. Convergence results for block GMRES with two right hand sides. Columns 2 and 3 are for the standard and separate bases implementations. Column 4 is for the FGMRES version. Numbers in parentheses are the number of GMRES iterations, and alongside is the reduction in residual norm

Matrix	Standard implementation	Separate bases implementation	FGMRES version
WIGTO966	(1 200) 0.2764E-05	(1 200) 0.3427E-05	(1 200) 0.5660E-08
	(1 200) 0.9102E-05	(1 200) 0.9851E-05	(1 200) 0.2249E-07
SAYLR4	(1 200) 0.7848E-06	(1 200) 0.8841E-06	(1 200) 0.2017E-06
	(1 200) 0.2323E-05	(1 200) 0.2572E-05	(1 200) 0.6504E-06
SHERMAN5	(200) 0.1472E-09	(200) 0.1472E-09	(200) 0.1144E-13
	(200) 0.1626E-09	(200) 0.1626E-09	(200) 0.2815E-06

for a class of problems, and *small* will have to be quantified for that class. Another point to note is that deflation is often effective later in the solution process, and inner iteration is shown in Figures 2 and 3 to be effective earlier on. In any adaptive procedure, it may be worthwhile to use an inner iteration early in the process, when the estimated eigenvectors are inaccurate, and deflation later, when they become more accurate.

We also tried adding eigenvectors with smallest residual error, instead of those corresponding to the eigenvalues closest to zero. This was not effective. Morgan [4] discusses keeping track of which eigenvectors are added, and releasing them when their component in the solution has been resolved. This is in place of continuing to add them if they continue to correspond to an eigenvalue closest to zero. This was not found to be effective.

The results of Figure 5 show that the eigenvectors tend to become more accurate as the solution process progresses. Thus, if a problem with multiple right-hand sides is being solved, knowledge of eigenvectors gained in solving for one right-hand side should be used in solving for the next right-hand sides. This would also apply to non-linear problems, if estimates of eigenvalues are available from the solution of a nearby problem. Table 4 shows results for solving the same problem repetitively, using eigenvectors calculated in preceding solutions. The table gives the number of iterations to reduce the residual norm by 10^{-8} using GMRES and deflation with BSSOR preconditioning for matrices BARTHT2A, WIGTO966, and SAYLR4. In the tests the number of eigenvectors added is four, and the number of iterations to converge for BARTHT2A, WIGTO966, and SAYLR4 for the same problem, but without deflation is 729, 431, and 1 475, respectively. The results show that recycling eigenvectors helps for WIGTO966 and SAYLR4, but not for BARTHT2A.

6.2. *Block-GMRES*

Block-GMRES is usually applied when solving linear systems with multiple right-hand sides. It is also sometimes used as a strategy to enhance convergence for the single right-hand side case. In this situation, the first block of the block Arnoldi process consists of the

Table 6. Convergence results for block GMRES with added eigenvectors. Numbers in parentheses give the number of GMRES iterations to reduce the residual norm by 10^{-8}. The tests are for the WIGTO966 matrix with BSSOR preconditioning and a total subspace size 48

Number added	Standard implementation		FGMRES version	
eigenvectors	Dimension	Iterations	Dimension	Iterations
0	1×48	(419)	48×1	(419)
1	2×24	$2 \times (312) = (624)$	24×2	(279)
2	3×16	$3 \times (287) = (861)$	16×3	(278)

initial residual, to which are added $p-1$ vectors which can be random vectors for example. One strategy to enhance convergence is to define these additional vectors to be approximate eigenvectors.

We consider first the case of multiple right-hand sides. Table 5 gives results for the standard and separate bases implementations of block GMRES, and compares these two results with the FGMRES version of block GMRES. In the tests two right-hand sides are used, and they are constructed so that the solutions are the vectors $(1, 1, 1, \ldots, 1)^{\mathrm{T}}$ and $(1, 2, 3, \ldots, n)^{\mathrm{T}}$. The initial guesses are vectors with random values between 0 and 1. BSSOR preconditioning was used in the tests, and the total number of vectors in the subspace is 50. The standard and separate bases implementations of block-GMRES give similar results as expected, and the difference between results are due to round-off error. Results for the FGMRES version differ. As a general observation, the results for the first right-hand side, are consistently better that the standard Block-GMRES. For the second right-hand side, the results are not consistent. Recall that FGMRES is not designed for systems with several right-hand sides. The solutions for the right-hand sides b^i with $i > 1$ are actually defined somewhat artificially. The reason why the results are better than the standard block-GMRES for the first right-hand side may be due to the fact that the subspace mixes vectors of the form $p(A)r_0$ where now p is of degree pm as was explained earlier.

Convergence results for the case of block GMRES with added eigenvectors are given in Table 6. For a block-size of p, the first vector is the residual vector and the $p-1$ remaining vectors are eigenvectors. Here p is varied from 1 (standard GMRES) to 3. The total subspace dimension is kept constant and equal to 48. Thus, for $p = 2$ the initial block is of size two (one residual vector and one estimated eigenvector) and the block-dimension m is 24, resulting in a total dimension of $2 \times 24 = 48$. For FGMRES, a similar strategy is applied. A Krylov subspace of dimension $m = 48/p$, is first generated from the initial residual vector and then $p-1$ other Arnoldi-like processes follow, each started with an estimated eigenvector. The tests are for matrix WIGTO966 with BSSOR preconditioning and a total of 48 vectors in the subspace. On this matrix the FGMRES version performs consistently better than the standard block-GMRES when adding eigenvectors in the single right-hand side case. It is interesting that the number of block–steps for block-GMRES, (312) and (287) for $p = 2, 3$, respectively, is similar to the number of matrix vector multiplications for the FGMRES version: (279) and (278).

7. Conclusion

Robustness of Krylov subspace methods can be substantially improved by enriching these subspaces with either approximate eigenvectors or inner iteration vectors. There are several reasons why this strategy may seem unlikely to be helpful in realistic situations. First, for very large matrices, removing a small number of eigenvalues out of a very large number of them close to zero may seem to have a limited effect on the solver. Another argument is that difficulties are often caused by high non-normality rather than small eigenvalues. In the highly non-normal case, it is indeed unlikely that deflation methods will help significantly. In many practical situations however, stalling of Krylov subspace methods seems to be caused by a small number of eigenvalues. Once those eigenvalues are removed from the picture, convergence can be significantly enhanced.

Block methods can be useful for solving linear systems with multiple right-hand sides. However, when used as a strategy for enhancing robustness our experiments seem to indicate that eigenvalue deflation may achieve a similar improvement at a much lower cost.

Acknowledgements

This work was supported by NASA under Grant NAG2-904 and by NSF under grant number CCR-9214116. The Minnesota Supercomputer Institute has provided computer facilities and an excellent research environment to conduct this research. We wish to thank Tim Barth from NASA Ames and Larry Wigton from Boeing for supplying matrices, and Larry Wigton and Edmond Chow for discussion on this work. We thank referees, whose reports lead to corrections and improvements in this paper.

REFERENCES

1. J. Erhel, K. Burrage and B. Pohl. Restarted GMRES preconditioned by deflation. Technical Report, IRISA, Rennes, France, 1994. To appear in *Journal of Computational and Applied Mathematics*, Dec. 1995.
2. S. A. Kharchenko and A. Yeramin. Eigenvalue translation based preconditioners for the GMRES(k) method. *Numerical Linear Algebra with Applications*, 2, 51–77, 1995.
3. R. B. Morgan. Computing interior eigenvalues of large matrices. *Linear Algebra and its Applications*, 154, 289–309, 1991.
4. R. B. Morgan. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Analysis and Applications*, 16(4), 1154–1171, 1995.
5. A. Ruhe. Implementation aspects of band Lanczos algorithms for computation of eigenvalues of large sparse symmetric matrices. *Math. Comp.*, 33, 680–687, 1979.
6. Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Halstead Press, New York, 1992.
7. Y. Saad. A flexible inner–outer preconditioned GMRES algorithm. *SIAM J. Sci. Stat. Comput.*, 14, 461–469, 1993.
8. Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing, New York, 1995. To appear.
9. Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7, 856–869, 1986.
10. V. Simoncini and E. Gallopoulos. An iterative method for nonsymmetric systems with multiple right-hand sides. *SIAM J. Sci. Comput.*, 1995 (to appear).
11. H. A. van der Vorst and C. Vuik. GMRESR: a family of nested GMRES methods. *Numerical Linear Algebra with Applications*, 1, 369–386, 1994.