SIAM J. SCI. COMPUT.
Vol. 30, No. 1, pp. 232–245

IMPROVING THE ACCURACY OF GMRES WITH DEFLATED RESTARTING*

S. RÖLLIN† AND W. FICHTNER†

Abstract. In general, restarting the generalized minimum residual method (GMRES) slows down the convergence speed. There exist a number of methods that improve the convergence of restarted GMRES. One of them is GMRES with deflated restarting [*SIAM J. Sci. Comput.*, 24 (2002), pp. 20–37] introduced by Morgan. This method retains a number of harmonic Ritz vectors at each restart to mitigate convergence slowdown. We investigate the accuracy of this method and propose a slight modification of it that is mathematically equivalent. We show that the new implementation has better numerical properties, especially if high accuracy of the solution is required. Numerical results are presented that confirm our theoretical results.

Key words. linear systems, iterative techniques, GMRES, deflation, accuracy

AMS subject classifications. 65F10, 65F15, 65F50, 65B99

DOI. 10.1137/060656127

1. Introduction.

The generalized minimum residual method (GMRES) is one of the most widely used iterative methods for solving large sparse linear systems. A major drawback of GMRES is its increasing computational cost: in iteration step k, the memory requirements as well as the number of vector operations (DAXPY, DDOTS) are proportional to k. In order to circumvent this problem, a restarted version of GMRES is usually used. Unfortunately, one has to take into account a slower convergence rate, since at each restart some information is discarded. Especially in the presence of small eigenvalues in the iteration matrix, the convergence rate of restarted GMRES may be much worse than that of full GMRES.

To retain the convergence rate, a number of techniques have been proposed that compute eigen-information at a restart and use this information to improve the convergence of restarted GMRES; see, e.g., [1, 3, 5]. The focus of this paper is on GMRES with deflated restarting [11] (GMRES-DR), which is one of those methods. It is equivalent to GMRES with eigenvectors [9] and to implicitly restarted GMRes [10].

In this paper, we propose modifications of GMRES-DR that lead to slightly better accuracy. Additionally, we provide other insights into the underlying equations of the algorithm. In the next section, we introduce GMRES-DR with the suggested modifications. Furthermore, we provide an error analysis of a restart of the algorithm that justifies our implementation. Section 3 illustrates our findings with numerical experiments. Finally, we draw some conclusions in the last section.

2. GMRes with deflated restarting.

The GMRES method is a Krylov subspace method that iteratively solves a linear system

$$Ax = b, \qquad A \in \mathbb{R}^{n\times n}, \qquad b, x \in \mathbb{R}^n. \tag{2.1}$$

*Received by the editors April 3, 2006; accepted for publication (in revised form) May 21, 2007; published electronically December 13, 2007.
http://www.siam.org/journals/sisc/30-1/65612.html
†Integrated Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland (roellin@iis.ee.ethz.ch, fw@iis.ee.ethz.ch).

In each iteration, it generates an iterate $x_k \in x_0 + \mathcal{K}_k(A, r_0)$ such that the 2-norm of the residual $r_k = b - Ax_k$ is minimal. This is achieved by building an orthonormal basis $\{v_1, \ldots, v_{k+1}\}$ of the Krylov subspace $\mathcal{K}_{k+1}(A, r_0)$. This orthonormalization process is called the Arnoldi method and leads to the Arnoldi relation

$$AV_k = V_{k+1}\underline{H}_k, \tag{2.2}$$

where $V_{k+1} = [v_1 \ldots v_{k+1}]$ and $\underline{H}_k$ is an upper Hessenberg matrix with $k+1$ rows and k columns. By solving a least squares problem with matrix $\underline{H}_k$, one can find the iterate x_k with minimal residual.

As pointed out in the introduction, each iteration step becomes computationally more expensive. Therefore, in the (restarted) GMRES(m) version, one sets $x_0 = x_m$ at iteration m and restarts the algorithm. Unfortunately, the orthogonal basis built so far is thrown away, and a new basis of the Krylov subspace $\mathcal{K}_k(A, r_m)$ is generated from scratch. Since some information is discarded, the convergence is expected to be slower compared to full GMRES.

In the GMRES with deflated restarting method—GMRES-DR(m,k)—the most valuable information from the orthonormal basis $\{v_0, \ldots, v_m\}$ is retained. To be more precise, when an orthonormal basis with $m+1$ vectors is reached, a restart is carried out, and k harmonic Ritz vectors are computed. Together with the current residual, these vectors are used to construct a new orthogonal basis of dimension $k+1$. It was shown by Morgan [11] that this basis spans a Krylov subspace (see also Proposition 4). Instead of starting from scratch, we extend this subspace to dimension $m+1$ with the Arnoldi method. At this point, the method is restarted again. Details of the method are given in Algorithm 1.

Algorithm 1 GMRES-DR(m,k).

choose an initial guess x_0 and compute $r_0 := b - Ax_0$; set $v_1 := r_0/\|r_0\|$, $\beta := \|r_0\|$.
apply standard GMRES to form V_{m+1} and upper Hessenberg matrix $\underline{H}_m \in \mathbb{R}^{m+1\times m}$.
$c_m := [\beta\, 0 \ldots 0]^T \in \mathbb{R}^{m+1}$.
loop
  solve $d_m := \operatorname{argmin}_{d\in\mathbb{R}^m} \|c_m - \underline{H}_m d\|$; set $x_m := x_0 + V_m d_m$, $r_m := b - Ax_m$.
  if $\|r_m\|$ $(= \|c_m - \underline{H}_m d_m\|)$ is small enough, **then**
    break
  end if
  let H_m be the submatrix of $\underline{H}_m$ consisting of the first m rows.
  $\beta := h_{m+1,m}$, $f := H_m^{-T} e_m$, $e_m = [0 \ldots 0\ 1]^T \in \mathbb{R}^m$.
  compute the k smallest eigenpairs (θ_i, g_i) of $H_m + \beta^2 f e_m^T$.
  orthonormalize the vectors g_i to form the columns of $P_k \in \mathbb{R}^{m\times k}$.
  orthonormalize the vector $\left(\begin{smallmatrix} -\beta f \\ 1 \end{smallmatrix}\right)$ against the columns of $\left(\begin{smallmatrix} P_k \\ 0 \end{smallmatrix}\right)$ to form p_{k+1}.
  set $P_{k+1} := \left(\begin{smallmatrix} P_k & p_{k+1} \\ 0 & \end{smallmatrix}\right)$, $V_{k+1}^{new} := V_{m+1}P_{k+1}$, $\underline{H}_k^{new} := P_{k+1}^T \underline{H}_m P_k$.
  apply the Arnoldi method to extend V_{k+1}^{new} and $\underline{H}_k^{new}$ to V_{m+1} and $\underline{H}_m$.
  $c_m := [r_m^T V_{k+1}^{new}\ \ 0 \ldots 0]^T$, $x_0 := x_m$.
end loop

As pointed out by Morgan, we might have to adjust k (the number of eigenvectors to compute at a restart) during the algorithm to include both the real and imaginary parts of complex eigenvectors. The θ_i's in Algorithm 1 are called the harmonic Ritz values. In the GMRES(m) method, $\underline{H}_m$ is always an upper Hessenberg matrix. However, in GMRES-DR, $\underline{H}_m$ now is an upper block-Hessenberg matrix, since, in

general, $\underline{H}_k^{new}$ after a restart is a full matrix. In the following, whenever we write $\underline{H}_m$, we mean a matrix with $m+1$ rows and m columns in the form

$$\underline{H}_m = \begin{pmatrix} H_m \\ e_m^T \beta \end{pmatrix}, \tag{2.3}$$

with H_m nonsingular, $e_m = [0 \dots 0\ 1]^T$; i.e., the last row of $\underline{H}_m$ is zero except the rightmost element.

There are some differences between the GMRES-DR method presented in Algorithm 1 and the one originally proposed by Morgan [11]. We will see that both versions are mathematically equivalent and that our version has some advantages in finite precision arithmetics, especially if high accuracy of the solution is required. The main differences of the two versions are the following:

1. construction of P_{k+1} and in particular the last column of this matrix,
2. computation of the vector c_m.

A further difference not apparent in Algorithm 1 is the computation of the last vector of V_{k+1}^{new}. For numerical reasons, Morgan suggested reorthogonalizing this vector. However, we propose another strategy to keep the columns in V_{k+1}^{new} orthonormal. Our approach leads to a higher accuracy but is computationally twice as expensive. We come back to this issue in the section with the numerical experiments.

In the following, we show the equivalence of both implementations of GMRES with deflated restarting. Morgan uses the vector $c_m - \underline{H}_m d_m$ (i.e., the residual of the minimization problem $\min_d \|c_m - \underline{H}_m d\|$) instead of the vector $[-\beta f^T\ 1]^T$ to construct the last column of P_{k+1}. The following theorem proves that both vectors are linearly dependent. Therefore both choices are mathematically equivalent and lead to the same P_{k+1}.

THEOREM 1. *Let $\underline{H}_m$ be as given in (2.3). Then define $f := H_m^{-T} e_m$, $d_m := \operatorname{argmin}_d \|c_m - \underline{H}_m d\|$, v the first m rows of c_m, and ω the last element of c_m. Under these assumptions, the following relation holds:*

$$c_m - \underline{H}_m d_m = \begin{pmatrix} -\beta f \\ 1 \end{pmatrix} \left(\frac{\omega - \beta f^T v}{1 + \beta^2 f^T f} \right). \tag{2.4}$$

Proof. The solution of the minimization problem is

$$d_m = \underline{H}_m^{\dagger} c_m,$$

with $\underline{H}_m^{\dagger}$ the Moore–Penrose inverse. Since $\underline{H}_m$ has rank m, a formula for the Moore–Penrose inverse is

$$\begin{aligned} \underline{H}_m^{\dagger} &= \left(\underline{H}_m^T \underline{H}_m\right)^{-1} \underline{H}_m^T \\ &= \left(H_m^T H_m + \beta^2 e_m e_m^T\right)^{-1} \underline{H}_m^T. \end{aligned}$$

In view of the Sherman–Morrison formula [7], we get

$$\begin{aligned} \underline{H}_m^{\dagger} &= \left(\left(H_m^T H_m\right)^{-1} - \frac{\beta^2 \left(H_m^T H_m\right)^{-1} e_m e_m^T \left(H_m^T H_m\right)^{-1}}{1 + \beta^2 e_m^T \left(H_m^T H_m\right)^{-1} e_m} \right) \underline{H}_m^T \\ &= \left(H_m^{-1} H_m^{-T} - \frac{\beta^2 H_m^{-1} f f^T H_m^{-T}}{1 + \beta^2 f^T f} \right) \begin{bmatrix} H_m^T & \beta e_m \end{bmatrix} \\ &= H_m^{-1} \left(I_m - \frac{\beta^2 f f^T}{1 + \beta^2 f^T f} \right) \begin{bmatrix} I_m & \beta f \end{bmatrix}. \end{aligned}$$

Therefore we can write

$$d_m = \underline{H}_m^{\dagger} c_m = H_m^{-1}\left(v - \frac{\beta^2 f f^T v}{1+\beta^2 f^T f} + \beta\omega f - \beta\omega\frac{\beta^2 f f^T f}{1+\beta^2 f^T f}\right)$$
$$= H_m^{-1}\left(v + \beta f\left(\frac{\omega - \beta f^T v}{1+\beta^2 f^T f}\right)\right).$$

As a result, the first m rows of (2.4) are

$$v - H_m d_m = v - v - \beta f\left(\frac{\omega - \beta f^T v}{1+\beta^2 f^T f}\right).$$

Finally, the last row of (2.4) is

$$\omega - \beta e_m^T d_m = \omega - \beta f^T\left(v + \beta f\left(\frac{\omega - \beta f^T v}{1+\beta^2 f^T f}\right)\right)$$
$$= (\omega - \beta f^T v) - \beta^2 f^T f\left(\frac{\omega - \beta f^T v}{1+\beta^2 f^T f}\right)$$
$$= (\omega - \beta f^T v)\left(1 - \frac{\beta^2 f^T f}{1+\beta^2 f^T f}\right)$$
$$= \frac{\omega - \beta f^T v}{1+\beta^2 f^T f},$$

which completes the proof. ∎

As already mentioned, a further difference is the computation of the vector c_m of length $m+1$. Morgan uses $c_m = V_{m+1}^T r_m$ to compute it. However, we can save a few inner products, since only the first $k+1$ entries are nonzero. The reason is that the residual r_m within the loop in Algorithm 1 lies in the subspace V_{k+1}^{new}. This is straightforward to show. First, we note that after the minimization at a restart, r_m can be written as

$$r_m = b - Ax_m = V_{m+1}(c_m - \underline{H}_m d_m).$$

The new subspace of dimension $k+1$ is written as

$$V_{k+1}^{new} = V_{m+1}\begin{pmatrix} P_k & \\ 0 & c_m - \underline{H}_m d_m \end{pmatrix} T_{k+1},$$

with T_{k+1} nonsingular. Therefore, we get by multiplying with T_{k+1}^{-1} and applying e_{k+1} from the right,

$$V_{k+1}^{new} T_{k+1}^{-1} e_{k+1} = V_{m+1}(c_m - \underline{H}_m d_m) = r_m.$$

In other words, r_m can be expressed as a linear combination of the columns of V_{k+1}^{new}. This justifies our choice of $c_m := [r_m^T V_{k+1}^{new} \;\; 0 \ldots 0]^T$ for the next cycle. Together with Theorem 1, we conclude that Algorithm 1 is equivalent to the GMRES-DR algorithm originally proposed by Morgan. Next, we will explain why we prefer Algorithm 1.

2.1. Analysis of a restart of GMRes with deflated restarting. As we have shown above, Morgan's choice of P_{k+1} and ours are equivalent in exact arithmetic. However, in finite precision arithmetic, our suggestion is preferable. The Arnoldi relation (2.2) is fulfilled only approximately in a numerical algorithm due to rounding errors. We will show that the error of the Arnoldi relation is smaller in our case. Basically, in Theorem 3 we show the impact of using different vectors in the computation of the harmonic eigenvectors and the construction of the matrix P_{k+1}. In practice, the residual of the minimization problem and the vector f will differ due to rounding errors. As we will see, this reduces the final accuracy. We first introduce some notation.

We denote by (g_i, θ_i) an eigenpair of $H_m + \beta^2 f e_m^T$, i.e.,

$$(H_m + \beta^2 f e_m^T) g_i = g_i \theta_i. \tag{2.5}$$

We emphasize that f may be any vector of length m. In the GMRes-DR algorithm it is $f = H_m^{-T} e_m$, but this special choice is not necessary in Lemma 2 and Theorem 3. Let us choose k linear independent eigenvectors $\{g_i\}$ and set $G_k := [g_1 \dots g_k]$. Furthermore, let $T_k \in \mathbb{R}^{k \times k}$ be nonsingular such that the columns of $P_k := G_k T_k$ are orthonormal, i.e., $P_k^T P_k = I_k$. Since we want to see what happens if the vector $[-\beta f^T \ \ 1]^T$ needed to construct P_{k+1} is perturbed, we replace it with

$$\begin{pmatrix} -\beta(f + \delta f) \\ 1 \end{pmatrix}. \tag{2.6}$$

It follows that P_{k+1} has the form

$$P_{k+1} := \begin{pmatrix} P_k & -\alpha g \\ 0 & \alpha \end{pmatrix}, \tag{2.7}$$

where

$$g := \beta(I_m - P_k P_k^T)(f + \delta f) \quad \text{and} \quad \alpha := \sqrt{1/(g^T g + 1)}. \tag{2.8}$$

It is straightforward to show that this definition of P_{k+1} corresponds to that used in the GMRes-DR algorithm if f is chosen appropriately and $\delta f = 0$. The following lemma is used to prove our main result, Theorem 3.

Lemma 2. *With the above assumptions we have*

$$P_{k+1} P_{k+1}^T \underline{H}_m P_k = \underline{H}_m P_k - \underline{E}_{m,k} P_k, \tag{2.9}$$

with

$$\underline{E}_{m,k} := \beta^2 \begin{pmatrix} I_m - P_k P_k^T - \alpha^2 g g^T \\ \alpha^2 g^T \end{pmatrix} \delta f e_m^T. \tag{2.10}$$

The proof of this lemma is given in the appendix. With the help of Lemma 2, we can prove the following theorem.

Theorem 3. *Let P_{k+1} and $\underline{H}_m$ be defined as before. If we have V_{m+1} orthonormal such that $AV_m = V_{m+1}\underline{H}_m$, then we have*

$$\|AV_k^{new} - V_{k+1}^{new} \underline{H}_k^{new}\| = \|\underline{E}_{m,k}\|,$$

where

$$V_{k+1}^{new} := V_{m+1}P_{k+1} \qquad \textit{and} \qquad \underline{H}_k^{new} := P_{k+1}\underline{H}_m P_k,$$

and $\underline{E}_{m,k}$ *defined as in* (2.10).

Proof. Using Lemma 2, we conclude that

$$\begin{aligned} AV_k^{new} &= AV_m P_k \\ &= V_{m+1}\underline{H}_m P_k \\ &= V_{m+1}(P_{k+1}P_{k+1}^T\underline{H}_m P_k + \underline{E}_{m,k}P_k) \\ &= V_{k+1}^{new}\underline{H}_k^{new} + V_{m+1}\underline{E}_{m,k}P_k. \qquad \square \end{aligned}$$

We learn from Theorem 3 that it is important to use the vector f both to compute the eigenvectors of $H_m + \beta^2 f e_m^T$ and to construct the last column of matrix P_{k+1}. Otherwise, the Arnoldi relation after a restart is affected by errors. Our algorithm uses the same vector for both parts, compared to Morgan's implementation, which needs two (slightly) different vectors. We therefore expect that Algorithm 1 retains the Arnoldi relation to a higher accuracy.

As pointed out before, f is chosen arbitrarily in the above theorem. If H_m is nearly singular, the computation of $f = H_m^{-T}e_m$ may be erroneous. Even in this case, we see from Theorem 3 that the Arnoldi relation is retained after a restart. Hence, one might think that f can be chosen arbitrarily in the GMRES-DR method. In other words, the above theorem alone does not implicate the use of the harmonic Ritz vectors. The question comes up of why it is necessary to use these eigenvectors in the GMRES with deflated restarting method. We know that the subspace spanned by the columns of V_{k+1}^{new} has to contain the residual r_m. From Theorem 1 it becomes clear that the special choice $f = H_m^{-T}e_m$ fulfills this constraint. Otherwise, we lose the basic property of GMRES-DR to minimize the residual with the help of a small minimization problem.

What is the consequence if the Arnoldi relation is not fulfilled exactly? Let us have a look at the residual r_{2m-k}, i.e., the residual at the second restart. We assume that the first restart results in the errors shown in Theorem 3,

$$\begin{aligned} \|r_{2m-k}\| &= \|b - A(x_m + V_m d_m)\| \\ &= \|r_m - V_{m+1}\underline{H}_m d_m - \begin{bmatrix} V_{m+1}^{old}\underline{E}_{m,k}P_k & 0\dots 0 \end{bmatrix} d_m\| \\ &\le \|c_m - \underline{H}_m d_m\| + \|V_{m+1}^{old}\underline{E}_{m,k}P_k d_m(1:k)\| \\ &\le \|c_m - \underline{H}_m d_m\| + \|\underline{E}_{m,k}\|\|d_m\|. \end{aligned} \tag{2.11}$$

The vectors V_{m+1}^{old} stem from the first cycle, $\underline{E}_{m,k}$ from the first restart, and d_m is computed at the second restart. We see that the ultimate accuracy of the residual is limited by $\|\underline{E}_{m,k}\|\|d_m\|$. The question is how large this error becomes in an implementation and whether it will degrade the accuracy or not. Certainly, we will see an effect only if we try to make the residuals as small as possible. The numerical results in the next section show that there are indeed differences in the residuals.

Morgan has shown in [11, Theorem 3.3] that the subspace used in GMRES-DR is a Krylov subspace. With the above theorem, we can give another proof of this result.

PROPOSITION 4. *If $\delta f = 0$, then the columns of the matrix V_{m+1} in the* GMRES-DR *algorithm are always a basis of a Krylov subspace.*

Proof. It is clear that the columns of V_{m+1} form a basis of the Krylov subspace $\mathcal{K}_{m+1}(A, r_0)$ before the first restart. From Theorem 3, we know that after a restart the matrix V_{k+1}^{new} satisfies the relation

$$AV_k^{new} = V_{k+1}^{new} \underline{H}_k^{new}.$$

In general, the matrix $\underline{H}_k^{new}$ has no special structure. Stewart [13] has called this relation an orthonormal Krylov decomposition and has shown that it is equivalent to the Arnoldi relation (2.2); see [13, Theorem 2.2]. Therefore, the columns of V_{k+1}^{new} build a basis of a Krylov subspace. This basis is then augmented such that

$$AV_m = V_{m+1} \underline{H}_m.$$

Again, $\underline{H}_m$ has no special property, and we can use the same argumentation as before to show that the columns of V_{m+1} are a basis of a Krylov subspace. ∎

3. Numerical experiments. In this section, we compare two implementations of the GMRES-DR method. The first version corresponds to Algorithm 1. We will call it "MOD" (for "modified") in the following. The only difference in the second implementation is the construction of P_{k+1}, for which we use the vector $c_m - \underline{H}_m d_m$ instead of $[-\beta f^T \; 1]^T$. This corresponds to the algorithm proposed by Morgan [11], except for the computation of c_m. We name this version "ORIG" in the remainder. In both implementations, we carry out two orthogonalization steps in the Arnoldi method. This means that each time we add a new vector to V_k, we orthogonalize this vector against the previous columns of V_k and then immediately reorthogonalize the obtained vector against the same vectors once again. With this technique, we retain the orthogonality up to working precision [6]. We stop the iteration if either 500 iteration steps are carried out or $\|c_n - \underline{H}_n d_n\| < 10^{-15}$. The second criterion is equivalent to $\|b - Ax_n\| = \|r_n\| < 10^{-15}$ in exact arithmetic. But in this latter form, the computation is more expensive, and furthermore the true residuals $\|r_n\|$ stagnate in finite precision arithmetic. Therefore, we stick to the first formulation for the stopping criterion. In the following figures, we depict both $\|c_n - \underline{H}_n d_n\|$ and $\|r_n\|$ to see the difference between those quantities. The chosen tolerance is very small, but as pointed out before, the two implementations differ only if very accurate solutions are required. In GMRES-DR we do a restart for $m = 25$ and retain ten harmonic Ritz vectors.

Example 1. We compare the two implementations for a set of matrices from the Harwell–Boeing collection. As preconditioner we take either SPAI-0 or SPAI-1. If SPAI-0 does not lead to convergence of GMRES-DR within 500 iteration steps, we switch to SPAI-1. Both preconditioners are a variant of the SPAI preconditioner of Grote and Huckle [8]. In SPAI-0 and SPAI-1, the pattern of the preconditioner M is fixed, instead of being augmented during the construction. M is a diagonal matrix for the former, whereas the pattern for the second choice is equal to the pattern of the original matrix [2]. We apply the preconditioner from the left. The right-hand side is set to $b = Ae$, where e is a vector of ones.

We depict the convergence behavior of GMRES-DR applied to matrix PORES3 in Figure 3.1. As the preconditioner, SPAI-1 has been used. Both implementations require about the same number of iterations. However, the version with our modification leads to a residual that is about two orders of magnitude better. Or, in other

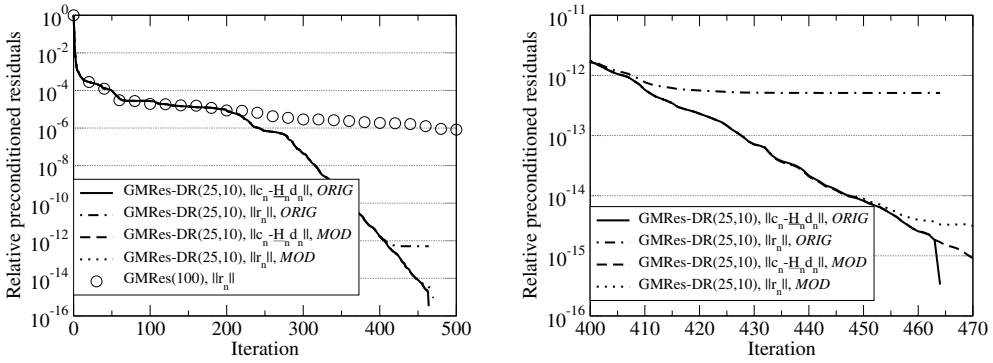

FIG. 3.1. *Relative preconditioned residuals for the matrix* PORES3 *are shown on the left. Two different implementations of* GMRES-DR *are compared. For a comparison of the convergence speed, results for* GMRES(100) *are also given. For better recognizability, a subgraph of the left figure is given on the right.*

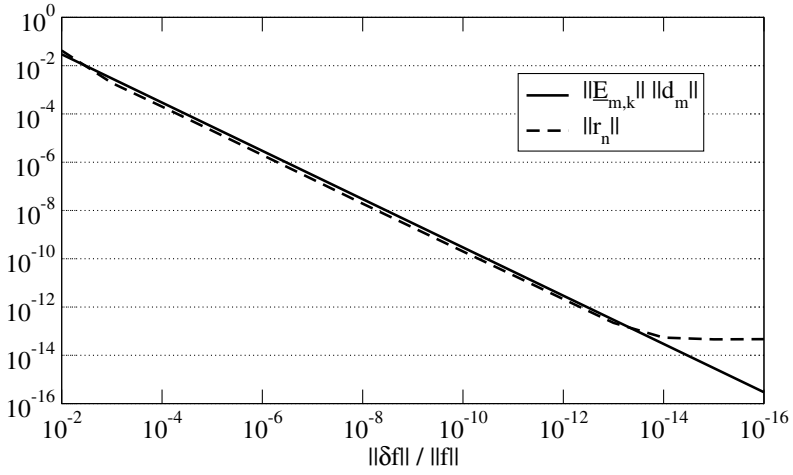

FIG. 3.2. *Final accuracy of the residuals in* GMRES-DR *and the estimate given in* (2.11)*; dependence on* δf.

words, the quantities $\|r_n\|$ and $\|c_n - \underline{H}_n d_n\|$ differ earlier in the original implementation (and thus more) compared to the modified one. This behavior is predicted by Theorem 3: the Frobenius norm of $AV_k^{new} - V_{k+1}^{new}\underline{H}_k^{new}$ after the first restart is about $3 \cdot 10^{-12}$. This is in good agreement with $\|\underline{E}_{m,k}\| = 5 \cdot 10^{-12}$. As stated above, an estimate for the ultimate accuracy of the residual is $\|\underline{E}_{m,k}\|\|d_m\|$. For the example this estimate is $1 \cdot 10^{-11}$, and the norm of the residual at the last iteration is $8 \cdot 10^{-12}$.

As an illustration of how well GMRES-DR performs, we have also included results for standard GMRES(100) in Figure 3.1. Despite the fact that the subspace is four times larger, the residual after 500 iteration steps is about nine orders of magnitude larger.

In order to further support our theory presented before, we artificially perturb the construction of P_{k+1} in Algorithm 1. Namely, we use the vector given in (2.6) with a random vector δf to construct the last column of P_{k+1} at the first restart. For all other restarts, we use $\delta f = 0$. In Figure 3.2 we depict the final accuracy of the residuals

TABLE 3.1
The final relative preconditioned residuals, i.e., $\|r_n\|/\|r_0\|$, and the number of iterations for GMRES-DR(25,10) *and matrices from the Harwell–Boeing collection. Ratio = Res_{ORIG}/Res_{MOD}.*

Matrix	Preconditioner	Res_{MOD}	Res_{ORIG}	Ratio	Iter_{MOD}	Iter_{ORIG}
add20	SPAI-0	2.22e-15	3.12e-14	14.1	429	429
add32	SPAI-0	1.47e-15	1.57e-15	1.1	132	132
pores3	SPAI-1	3.15e-15	5.11e-13	162.4	470	464
saylr3	SPAI-0	3.31e-15	7.09e-14	21.4	454	453
sherman1	SPAI-0	2.60e-15	2.11e-14	8.1	458	458
sherman3	SPAI-1	1.79e-15	2.38e-14	13.3	414	414
sherman4	SPAI-0	1.26e-15	7.40e-15	5.9	138	138
sherman5	SPAI-0	1.70e-15	3.45e-14	20.3	186	185

as well as an estimate according to (2.11). The same matrix and preconditioner have been used as before. We have solved the linear system several times with different vectors δf. As we can see, $\|\underline{E}_{m,k}\|\|d_m\|$ is a good estimate of the final accuracy of the residuals. The difference for a very small δf is obvious: the accuracy of the residuals is limited by finite precision arithmetic. Thus, further improving the accuracy of P_{k+1} does not lead to more accurate residuals.

In Table 3.1 we list more results for matrices from the Harwell–Boeing collection. For all of them, the modified implementation reduces the norm of the residuals further than the original one does. However, the differences are not very large (about one order of magnitude on average). Our findings coincide with the theoretical consideration in the previous section.

Example 2. The matrix and right-hand side for this example are taken from a semiconductor device simulation problem. The linear system has 32040 unknowns and stems from the simulation of a memory cell. We first reorder and scale the matrix with MPS [4]. It turns out that applying MPS is very beneficial in the field of semiconductor device simulation [12]. After that, the matrix is reordered further with the nested dissection algorithm before a (left) preconditioner M is constructed. Here, we use a threshold-based incomplete LU-factorization with a drop tolerance of 10^{-4}. A problem of this example is that even the preconditioned matrix, i.e., MA, has a few very small eigenvalues of order 10^{-9}. These eigenvalues spoil the convergence speed of restarted GMRES and confirm the small stopping tolerance of 10^{-15}.

The convergence behavior of the two implementations of GMRES-DR are compared in Figure 3.3. The differences are more pronounced in this example. Our implementation as shown in Algorithm 1 is much more accurate. When the iteration stops, the true preconditioned residual is about five orders of magnitude better. The results of the original implementation deserve closer attention. Despite the fact that the residual should be nonincreasing, both $\|r_n\|$ as well as $\|c_n - \underline{H}_m d_n\|$ grow in the original implementation, the former at iteration step 65, the latter at step 85 (where the fifth restart is carried out). This is due to the Arnoldi relation that is fulfilled only approximately. We also have depicted the results for standard restarted GMRES. As one can see, the convergence is slower for a Krylov subspace that is twice as large.

3.1. Orthogonalization of the Arnoldi vectors. We now come back to an issue mentioned in the previous section, namely, the orthogonality of the vectors V_{k+1}^{new}. In order to achieve a high accuracy, it is important to keep the basis of the Krylov subspace orthogonal to a high precision. In standard restarted GMRES, the orthogonality is less important, since the complete subspace is thrown away at a restart and, hence, the errors regarding the orthogonality of the basis vectors do not

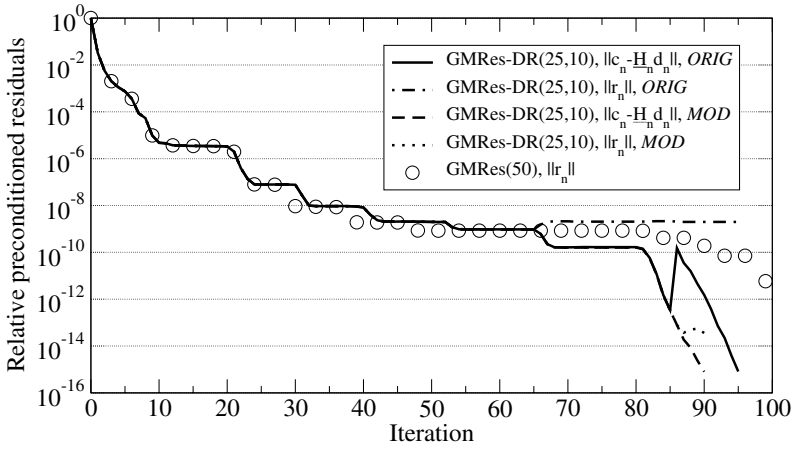

FIG. 3.3. *Preconditioned residuals for the matrix DRAM-18. Two different implementations of* GMRES-DR *are compared. For a comparison of the convergence speed, results for* GMRES(50) *are also given.*

propagate further. This is different in the GMRES-DR method. We just recall from Algorithm 1 that the new vectors at a restart are computed according to $V_{k+1}^{new} := V_{m+1}P_{k+1}$. From this relation we expect that the orthogonality of V_{k+1}^{new} is worse than the one of the vectors V_{m+1} generated during the previous cycle. In other words, the orthogonalization error propagates across different cycles in GMRES-DR.

In the following, we will describe four different possibilities that we have compared numerically to retain the orthogonality. The first method has been used for the numerical experiments so far: we carry out two orthogonalization steps in each Arnoldi step and just compute V_{k+1}^{new} according to the above formula. At a restart, no special emphasis is put on the orthogonality of these vectors. The other three methods use only one orthogonalization step in the Arnoldi method. Thus the orthogonalization cost of the first strategy is twice as large. In order to retain the orthogonality of the vectors V_{k+1}^{new} generated at a restart, we have tested the following possibilities: no reorthogonalization; reorthogonalization of the last vector only, i.e., v_{k+1}^{new} (as suggested by Morgan); or reorthogonalization of all vectors. The first strategy is the cheapest and the last the most expensive of those three possibilities.

In Figure 3.4, we compare all four strategies. It turns out that the first is the optimal strategy. As we can see, the last three strategies do not perform as well as the first one. One problem is that the orthogonality of the Arnoldi vectors is not sufficient, e.g., for the second method. Another deficiency is that due to the reorthogonalization the Arnoldi relation is perturbed at a restart. For example, the last method leads to highly orthogonal vectors, but the Arnoldi relation (2.2) is affected by errors. Further numerical results are given in Table 3.2. They show that carrying out an additional reorthogonalization step is beneficial. We see that reorthogonalizing the last vector only is also a good choice, but slightly less accurate.

We observe another advantage of our proposed strategy in a lot of other numerical examples. Namely, two orthogonalization steps in the Arnoldi method lead often to faster convergence. An example is given in Figure 3.5. As one can see, using only one orthogonalization step leads to a stagnation and therefore to more iteration steps. Our suggestion converges in 18 iteration steps, whereas at least 26 iterations are used for the other methods. Furthermore, at the first restart, the residual increases

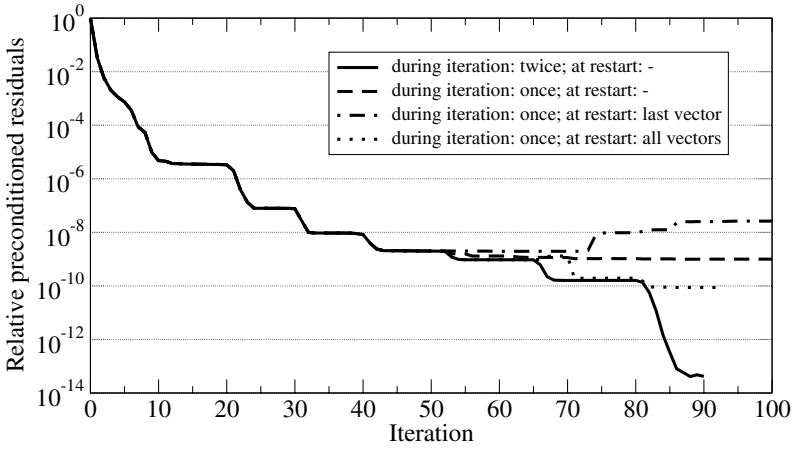

FIG. 3.4. *Effect of reorthogonalization of the Arnoldi vectors. Shown are the preconditioned residuals for the matrix DRAM*-18.

TABLE 3.2

The final relative preconditioned residuals, i.e., $\|r_n\|/\|r_0\|$*, for* GMRES-DR$(25, 10)$ *and matrices from the Harwell–Boeing collection. Four different strategies to retain the orthogonality of the Arnoldi basis are listed. The last three columns compare the first strategy to the others.*

Matrix	Prec.	$\|r_n\|/\|r_0\|$				Ratio		
		2 - none	1 - none	1 - last	1 - all	1 - none	1 - last	1 - all
add20	SPAI-0	2.15e-15	6.79e-15	2.24e-15	2.36e-15	3.2	1.0	1.1
add32	SPAI-0	1.47e-15	1.45e-15	1.47e-15	1.47e-15	1.0	1.0	1.0
pores3	SPAI-1	3.18e-15	1.81e-12	3.02e-15	5.89e-14	569.6	1.0	18.5
saylr3	SPAI-0	3.42e-15	4.13e-14	5.30e-15	4.31e-14	12.1	1.6	12.6
sherman1	SPAI-0	2.76e-15	1.21e-13	1.02e-14	4.60e-14	44.0	3.7	16.7
sherman3	SPAI-1	1.84e-15	1.48e-12	6.64e-15	7.14e-14	804.1	3.6	38.8
sherman4	SPAI-0	1.24e-15	2.01e-15	2.22e-15	3.47e-15	1.6	1.8	2.8
sherman5	SPAI-0	1.51e-15	3.20e-14	9.68e-15	1.66e-14	21.2	6.4	11.0

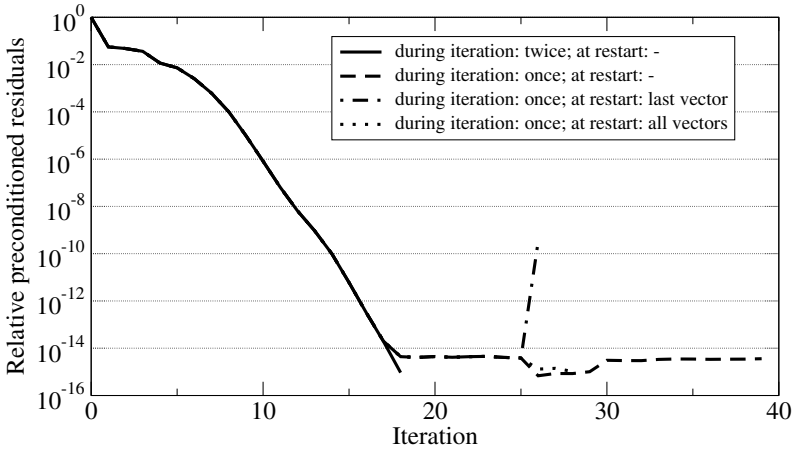

FIG. 3.5. *Effect of reorthogonalization of the Arnoldi vectors. Shown are the preconditioned residuals for matrix Sherman4. The matrix is solved by using the MPS and nested dissection ordering.* ILUT(10^{-2}) *is used as preconditioner.*

more than four orders of magnitude for the strategy that reorthogonalizes the last vector. Again, this is due to the Arnoldi relation that is no longer fulfilled. For this example standard GMRES(25) would also benefit from a reorthogonalization in the Arnoldi method. The difference is quite large: 18 iterations for the case with a reorthogonalization compared to 27 iterations otherwise.

4. Conclusion. In this paper, we have investigated the method GMRES with deflated restarting. In particular, the focus was on the accuracy of the Arnoldi relation during a restart. We have proposed a modification of the algorithm that is mathematically equivalent to the original algorithm introduced by Morgan [11]. Our algorithm minimizes the numerical errors during a restart and therefore is better suited if the linear systems have to be solved with high precision. The presented error analysis justifies our version of GMRES with deflated restarting. The numerical experiments coincide well with the theoretical consideration. Furthermore, we have provided a novel derivation and justifications of the investigated algorithm.

Appendix. Proof of Lemma 2. The columns of P_k are linear combinations of the vectors g_i. Hence, in order to verify (2.9), it is enough to prove the equation

$$P_{k+1}P_{k+1}^T \underline{H}_m g_i = \underline{H}_m g_i - \underline{E}_{m,k} g_i. \tag{A.1}$$

We first show that the following two identities hold.

$$P_k P_k^T H_m g_i = H_m g_i + \beta g e_m^T g_i - \beta^2 (I_m - P_k P_k^T)\delta f e_m^T g_i \tag{A.2}$$

and

$$-\alpha^2 (g^T H_m g_i - \beta e_m^T g_i) = \beta(1 - \alpha^2 \beta g^T \delta f) e_m^T g_i. \tag{A.3}$$

Both are used for the proof of (A.1) later on.

The vectors g_i are eigenvectors of $P_k P_k^T$ with eigenvalue one, i.e., $P_k P_k^T g_i = g_i$ holds:

$$P_k P_k^T G_k = P_k P_k^T G_k T_k T_k^{-1} = P_k T_k^{-1} = G_k.$$

By using the identity (2.5) twice, it follows that

$$\begin{aligned}
P_k P_k^T H_m g_i &= P_k P_k^T (g_i \theta_i - \beta^2 f e_m^T g_i) \\
&= g_i \theta_i - P_k P_k^T \beta^2 f e_m^T g_i \\
&= H_m g_i + \beta^2 f e_m^T g_i - P_k P_k^T \beta^2 f e_m^T g_i \\
&= H_m g_i + \beta^2 (I_m - P_k P_k^T) f e_m^T g_i \\
&= H_m g_i + \beta g e_m^T g_i - \beta^2 (I_m - P_k P_k^T)\delta f e_m^T g_i,
\end{aligned}$$

which proves (A.2). We conclude that

$$\begin{aligned}
g^T H_m g_i &= \beta (f^T + \delta f^T)(I - P_k P_k^T) H_m g_i \\
&= \beta (f^T + \delta f^T)\big(-\beta g e_m^T g_i + \beta^2 (I_m - P_k P_k^T)\delta f e_m^T g_i\big) \\
&= -\beta^2 (f^T + \delta f^T)\big(g - \beta (I_m - P_k P_k^T)\delta f\big) e_m^T g_i \\
&= -\beta^2 (f^T + \delta f^T) g e_m^T g_i + \beta\beta^2 (f^T + \delta f^T)(I_m - P_k P_k^T)\delta f e_m^T g_i \\
&= -\beta (g^T g - \beta g^T \delta f) e_m^T g_i.
\end{aligned}$$

For the last identity we have used $\beta(f^T + \delta f^T)g = g^T g$. Next, we prove (A.3):

$$\begin{aligned}
-\alpha^2(g^T H_m g_i - \beta e_m^T g_i) &= -\alpha^2(-\beta(g^T g - \beta g^T \delta f)e_m^T g_i - \beta e_m^T g_i) \\
&= \alpha^2\beta(g^T g - \beta g^T \delta f + 1)e_m^T g_i \\
&= \beta\big(\alpha^2(g^T g + 1) - \alpha^2\beta g^T \delta f\big)e_m^T g_i \\
&= \beta(1 - \alpha^2\beta g^T \delta f)e_m^T g_i.
\end{aligned}$$

Finally, we come back to (A.1):

$$\begin{aligned}
P_{k+1}P_{k+1}^T \underline{H}_m g_i &= \begin{pmatrix} P_k P_k^T + \alpha^2 g g^T & -\alpha^2 g \\ -\alpha^2 g^T & \alpha^2 \end{pmatrix} \begin{pmatrix} H_m \\ \beta e_m^T \end{pmatrix} g_i \\
&= \begin{pmatrix} P_k P_k^T H_m g_i + \alpha^2 g g^T H_m g_i - \alpha^2\beta g e_m^T g_i \\ -\alpha^2 g^T H_m g_i + \alpha^2 \beta e_m^T g_i \end{pmatrix} \\
&= \begin{pmatrix} P_k P_k^T H_m g_i + \alpha^2 g(g^T H_m g_i - \beta e_m^T g_i) \\ \beta(1 - \alpha^2\beta g^T \delta f)e_m^T g_i \end{pmatrix} \\
&= \begin{pmatrix} P_k P_k^T H_m g_i - \beta g e_m^T g_i + \alpha^2\beta^2 g g^T \delta f e_m^T g_i \\ \beta e_m^T g_i - \alpha^2\beta^2 g^T \delta f e_m^T g_i \end{pmatrix} \\
&= \begin{pmatrix} H_m g_i - \beta^2(I_m - P_k P_k^T)\delta f e_m^T g_i + \alpha^2\beta^2 g g^T \delta f e_m^T g_i \\ \beta e_m^T g_i - \alpha^2\beta^2 g^T \delta f e_m^T g_i \end{pmatrix} \\
&= \begin{pmatrix} H_m g_i \\ \beta e_m^T g_i \end{pmatrix} - \begin{pmatrix} \beta^2(I_m - P_k P_k^T)\delta f e_m^T g_i - \alpha^2\beta^2 g g^T \delta f e_m^T g_i \\ \alpha^2\beta^2 g^T \delta f e_m^T g_i \end{pmatrix}.
\end{aligned}$$

This completes the proof of Lemma 2. ∎

Acknowledgment. We would like to thank the referees who made valuable suggestions and helped us to improve our manuscript.

REFERENCES

[1] J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel, *Adaptively preconditioned GMRES algorithms*, SIAM J. Sci. Comput., 20 (1998), pp. 243–269.

[2] O. Bröker, M. J. Grote, C. Mayer, and A. Reusken, *Robust parallel smoothing for multigrid via sparse approximate inverses*, SIAM J. Sci. Comput., 23 (2001), pp. 1396–1417.

[3] C. Le Calvez and B. Molina, *Implicitly restarted and deflated GMRES*, Numer. Algorithms, 21 (1999), pp. 261–285.

[4] I. S. Duff and J. Koster, *On algorithms for permuting large entries to the diagonal of a sparse matrix*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 973–996.

[5] J. Erhel, K. Burrage, and B. Pohl, *Restarted GMRES preconditioned by deflation*, J. Comput. Appl. Math., 69 (1996), pp. 303–318.

[6] L. Giraud, J. Langou, M. Rozložník, and J. van den Eshof, *Rounding error analysis of the classical Gram-Schmidt orthogonalization process*, Numer. Math., 101 (2005), pp. 87–100.

[7] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[8] M. J. Grote and T. Huckle, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp. 838–853.

[9] R. B. Morgan, *A restarted GMRES method augmented with eigenvectors*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 1154–1171.
[10] R. B. Morgan, *Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1112–1135.
[11] R. B. Morgan, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.
[12] O. Schenk, S. Röllin, and A. Gupta, *The effects of unsymmetric matrix permutations and scalings in semiconductor device and circuit simulation*, IEEE Trans. Computer-Aided Design Integrated Circuits Systems, 23 (2004), pp. 400–411.
[13] G. W. Stewart, *A Krylov–Schur algorithm for large eigenproblems*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 601–614.