Numerical Algorithms 21 (1999) 261–285

Implicitly restarted and deflated GMRES

C. Le Calvez [a] and B. Molina [b]

[a] *Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie, 4, place Jussieu, F-75252 Paris Cedex 05, France*
E-mail: caroline.lecalvez@lip6.fr

[b] *Departamento de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela*
E-mail: bmolina@reacciun.ve

We introduce a deflation method that takes advantage of the IRA method, by extracting a GMRES solution from the Krylov basis computed within the Arnoldi process of the IRA method itself. The deflation is well-suited because it is done with eigenvectors associated to the eigenvalues that are closest to zero, which are approximated by IRA very quickly. By a slight modification, we adapt it to the FOM algorithm, and then to GMRES enhanced by imposing constraints within the minimization condition. The use of IRA enables us to reduce the number of matrix–vector products, while keeping a low storage.

Keywords: restarted GMRES, restarted FOM, IRA, deflation, minimization with constraints

AMS subject classification: 65F10, 65F15, 64N30

1. Introduction

In recent years, many papers have focused on Krylov type methods combined with deflation techniques [1–3,5,10,15,16] when solving linear systems. In this paper, we will deal with unsymmetric real large sparse matrices and Krylov type methods that explicitly build a Krylov subspace such as the GMRES algorithm and FOM. Within the Arnoldi process that builds an orthonormal basis of the Krylov subspace, the number of vectors that are stored grows at each iteration as well as the computational cost. One remedy consists of restarting the algorithm every m steps and taking the initial guess to be the latest approximate solution obtained. The convergence of the Full GMRES algorithm (to be opposed to the restarted one) is related to the convergence of the Ritz values (see [18]). Concerning FOM, its residual norm does exhibit an erratic behavior because it does not have any minimization condition on its residual, but its convergence is still related to the convergence of the Ritz values. For GMRES and FOM, their full version behaves as if at each step the smallest eigenvalues were removed from the system. When restarting, the information concerning the smallest eigenvalues is lost and must be recovered at each run. Deflation techniques take advantage of the former information to incorporate it into the next run.

These techniques are of two types: a preconditioning approach and an augmented subspace approach. For one or the other, numerical results showed significant improvements in convergence rates compared to the same method with a Krylov subspace of the same size. The first approach was introduced by Erhel et al. in [5] with the restarted GMRES method to minimize the effects of restarting. It adaptively builds a preconditioner that takes advantage of the spectral information gathered by the Arnoldi process during the former runs of GMRES and acts as if the smallest eigenvalues in magnitude of the matrix were removed. This preconditioner remains the same during one whole run of GMRES and has nothing to do with the Flexible GMRES Method [13], that enables to change the preconditioner within the run. It can be combined with any preconditioner, approximating the spectrum of the preconditioned matrix instead of the matrix itself. In [2], the method is improved by updating the preconditioner: the eigenvalues are approximated with an harmonic eigenvalue problem instead of an oblique one and only the better Ritz vectors of all the runs are kept to form the preconditioner. In the original algorithm, the k best Ritz vectors of each run were gathered to form the new preconditioner, which induced more storage. Furthermore, some of the kept Ritz vectors were not always well approximated. In [1] the authors use the recurrence formulas of the Implicitly Restarted Arnoldi (IRA) method [17] to form a new preconditioner. This preconditioner is applied to the previous ones forming the preconditioner of the next run. Because of the recurrence formulas of IRA, the number of matrix–vector products per run is reduced. Part of the matrix–vector products are replaced by implicit shifted QR factorizations on a matrix whose size is equal to the size of the subspace.

The second approach is the one we are interested in. It consists of adding to the Krylov subspace a few approximate eigenvectors associated with the approximate eigenvalues closest to zero. In [10], Morgan adds to the Krylov subspace within the restarted GMRES algorithm some of the Ritz vectors of the previous run. He does a projection onto the following subspace:

$$S_{m,k}(A, r_0, y_1, \ldots, y_k) = \text{Span}\big\{r_0, Ar_0, \ldots, A^{m-k-1}r_0, y_1, \ldots, y_k\big\} \tag{1.1}$$

with r_0 the initial residual of the run, $y_1, \ldots, y_k$, some Ritz vectors computed within the previous run, instead of doing the projection onto the Krylov subspace

$$\mathrm{K}_m(A, r_0) = \text{Span}\big\{r_0, Ar_0, \ldots, A^{m-1}r_0\big\}.$$

The convergence can be much faster than for the restarted GMRES itself.

In the present paper, we introduce three methods based on the IRA method with deflation (see [9]). The first one uses the minimization condition of GMRES, the second one uses the Galerkin condition of FOM, whereas the third one uses GMRES slightly modified by a minimization condition with constraints. In each run, these three methods build a subspace of the same kind as Morgan's excepted that the Ritz vectors are chosen and computed in a more natural way through the recurrence formulas of the IRA method. It is more natural because $S_{m,k}(A, r_0, y_1, \ldots, y_k)$ is now a Krylov subspace itself, from which we can extract a solution as well as approximate eigenvectors

associated with the eigenvalues closest to zero, which we are interested in, and will be better approximated in the next run. The same Krylov subspace that is computed within the Arnoldi process, is used to solve a linear system as well as an eigenvalue problem. Because of this link, the method based on GMRES called the **I**mplicitly **D**eflated **G**eneral **M**inimal **R**esidual algorithm (IDGMRES) converges as fast as or faster than the method of Morgan itself, with far less storage requirements. The other two methods based on FOM and GMRES with constraints are respectively named IDFOM for the **I**mplicitly **D**eflated **F**ull **O**rthogonalization **M**ethod and IDMGMRES for the **I**mplicitly **D**eflated **M**odified **G**eneral **M**inimal **R**esidual algorithm.

We would like to mention that this work was developed independently from the recent work of Morgan [12] where he introduces the IRA method to enrich the Krylov subspace within FOM and GMRES. Nevertheless, we point out the main conceptual differences by comparing his methods to ours.

We first recall the GMRES and FOM algorithms as well as the Augmented GMRES of Morgan in section 2. In section 3 we explain our implicitly restarted and deflated approach, which we apply to GMRES and then extend to FOM and GMRES with constraints. Storage and computational requirements of the new methods as well as numerical results are presented in section 4. We finally conclude in section 5.

2. Background

We are interested in solving the large sparse unsymmetric real system $Ax = b$ with n the size of the matrix. We will recall in this section the basic tools concerning GMRES, FOM [14] and the Augmented GMRES due to Morgan [10].

2.1. GMRES and FOM

GMRES is a Krylov method that computes an approximate solution x_m at step m by doing an oblique projection onto the Krylov subspace $\mathrm{K}_m(A, r_0)$ of size m. Given an initial starting guess x_0 and its corresponding residual r_0, it builds, through the Arnoldi process, a matrix $V_m = [v_1, \ldots, v_m]$ whose columns are an orthonormal basis of $\mathrm{K}_m(A, r_0)$ and an upper Hessenberg matrix H_m, the orthogonal projected matrix of A onto V_m.

The matrices satisfy the well-known relationships

$$AV_m = V_{m+1}\overline{H}_m \tag{2.1}$$

$$= V_m H_m + h_{m+1,m} v_{m+1} e_m^{(m)\mathrm{T}}, \tag{2.2}$$

where $e_i^{(j)}$ is the ith column of the $j \times j$ identity matrix, $\overline{H}_m$ is the upper Hessenberg $(m+1)\times m$ matrix whose coefficients are the $h_{i,j}$, and H_m denotes $\overline{H}_m$ after deleting its last row. When $h_{j+1,j}$ equals zero, an invariant Krylov subspace of A has been found and GMRES computes the exact solution. In the following $h_{j+1,j}$ will be supposed to be different from zero: H_m is supposed to be an unreduced upper Hessenberg matrix.

The GMRES approximate solution is computed such that $x_m - x_0$ belongs to $\mathrm{K}_m(A, r_0)$ with $r_m = b - Ax_m$ being A-orthogonal to $\mathrm{K}_m(A, r_0)$. This leads to computation of $x_m = x_0 + V_m y_m$, where

$$y_m = \arg \min_{y \in \mathbb{R}^m} \left\| \beta e_1^{(m+1)} - \overline{H}_m y \right\|_2 \tag{2.3}$$

and $\beta = \|r_0\|_2$. This $(m+1) \times m$ least squares problem is solved through Givens rotations [14].

FOM is an orthogonal projection method where equation (2.3) has been replaced by $H_m y_m = \beta e_1^{(m)}$. The rest of the algorithm remains unchanged.

Because of memory and computational requirements that increase rapidly with the size m of the Krylov subspace, one usually computes the approximate solution of GMRES and FOM after m steps and restarts the process with x_m as the new initial starting vector. Such methods are said to be restarted and we call a run their m consecutive steps. In the sequel, we will use GMRES and FOM for, respectively, restarted GMRES and restarted FOM and specify the term full when speaking of their full versions.

2.2. Enriching the Krylov subspace

The convergence behavior of the full GMRES was studied by Van der Vorst and Vuik [18]. They showed that its superlinear convergence behavior takes place when a Ritz value is sufficiently close to the corresponding simple eigenvalue, or, for a multiple eigenvalue μ_i of algebraic multiplicity n_i, when n_i Ritz values are close enough to μ_i.

When the process is restarted, the convergence is slower. The size m of the Krylov subspace is usually not large enough for a Ritz value to approximate an eigenvalue well. Furthermore, part of the already computed spectral information is lost and must be recovered within the next run. For such reasons, Morgan proposed in the Augmented GMRES algorithm [10] to enrich the Krylov subspace with some eigenvectors. The underlying idea is that adding an eigenvector to the Krylov subspace excludes its corresponding eigenvalue from the spectrum of the matrix. Since only approximate eigenvectors are available through the Ritz vectors, Morgan suggested to keep some of them that were computed during the current run, and to add them within the next one. He investigated the Ritz vectors associated with the smallest Ritz values, because in the normal case or when the matrix is not too highly non-normal, they monitor the convergence. But of course it is not the only ingredient. Approximating a cluster of eigenvalues accelerates the convergence as well. As was suggested above and shown in [18], approximating any eigenvalue sufficiently well, and not only the smallest one, accelerates the convergence.

Another credit to enrich the Krylov subspace with the Ritz vectors associated with the smallest Ritz values is to look at the residual polynomial. At iteration m, x_m belongs to $\mathrm{K}_m(A, r_0)$. We have $x_m = x_0 + s_m(A) r_0$, where s_m is a polynomial of degree strictly less than m with real coefficients, and $r_m = (I - A s_m(A)) r_0$ belongs to

$\mathrm{K}_{m+1}(A, r_0)$. The polynomial $p_m(x) = 1 - xs_m(x)$ is called the residual polynomial. The best one would be the one with its roots equal to the eigenvalues of A associated with the eigenvectors onto which the vector r_0 does have a composant that is not null. But by construction, $p_m(0) = 1$, meaning that it might be hard for a polynomial of small degree, especially with indefinite matrices, to be small in values close to zero (typically the small eigenvalues) and to be equal to 1 in 0.

Nevertheless, choosing the smallest Ritz pairs works well in practice with the Augmented GMRES. For problems where GMRES have trouble in approximating other eigenvalues but not the smallest ones, it is useless to enrich the Krylov subspace with the smallest Ritz pairs. It would be more judicious to enrich it with the ones needed.

We briefly recall the Augmented GMRES method due to Morgan [10].

2.3. The Augmented GMRES

The algorithm computes an approximate solution x_m such that $x_m - x_0$ belongs to $S_{m,k}(A, r_0, y_1, \ldots, y_k)$ defined in (1.1), where $y_1, \ldots, y_k$ are some Ritz vectors and by imposing its corresponding residual r_m to be orthogonal to $AS_{m,k}(A, r_0, y_1, \ldots, y_k)$.

To do this, Morgan uses the Flexible GMRES scheme [13]. He first builds an orthonormal basis for $\mathrm{K}_{m-k}(A, r_0)$ with the Arnoldi process, where V_{m-k} is the corresponding matrix. He next takes the normalized eigenvector y_1, multiplies it by A and orthogonalizes it against the vectors v_i for $i = 1, \ldots, m-k+1$ and normalizes it. It forms the vector v_{m-k+2}. The subsequent vectors v are computed in the same way. Ay_j for $j \leqslant k$ is orthogonalized against the v_i's for $i = 1, \ldots, m-k+j$, and normalized to form the vector $v_{m-k+j+1}$.

Let us call $W_m = [V_{m-k}, y_1, \ldots, y_k]$. The relationships (2.1), (2.2) are now replaced by

$$AW_m = V_{m+1}\overline{H}_m \tag{2.4}$$

$$= V_m H_m + h_{m+1,m} v_{m+1} e_m^{(m)\,\mathrm{T}}. \tag{2.5}$$

The Ritz vectors used in the next run to enrich the Krylov subspace are the Ritz vectors of the form $y_j = W_m u_j$ associated with the largest values in magnitude $1/\mu_j$, where the eigenpairs (u_j, μ_j) solve the harmonic eigenvalue problem

$$W_m^{\mathrm{T}} A^{\mathrm{T}} W_m u_j = \frac{1}{\mu_j} W_m^{\mathrm{T}} A^{\mathrm{T}} A W_m u_j. \tag{2.6}$$

When complex eigenvectors appear, one has to take separately their real and imaginary parts.

Finally, we do the projection onto W_m whose image is exactly $S_{m,k}(A, r_0, y_1, \ldots, y_k)$. We have $x_m = x_0 + W_m y_m$, where

$$y_m = \arg\min_{y \in \mathbb{R}^m} \|r_0 - AW_m y\|_2 = \arg\min_{y \in \mathbb{R}^m} \left\|\beta e_1^{(m+1)} - \overline{H}_m y\right\|_2.$$

As is done with GMRES, y_m is computed through the application of Givens rotation matrices.

3. The implicitly restarted and deflated Krylov methods

We propose to improve the Augmented GMRES by making the deflation more efficient. Besides computing the approximate solution x_m our method improves the approximations of the wanted eigenpairs in order that a "real" deflation can take place. Such a method is possible thanks to the IRA method due to Sorensen [17]. In each run when no deflation takes place, we build the subspace $S_{m,k}(A, r_0, y_1, \ldots, y_k)$, where $y_1, y_2, \ldots, y_k$ are some Ritz vectors we are interested in and r_0 is the final residual of the previous run. This subspace is now a Krylov subspace $\mathrm{K}_m(A, v_1)$, where v_1 is a particular linear combination of the wanted Ritz vectors and it enables us to save storage. This subspace also contains k Krylov subspaces of size $m - k + 1$, each of them associated to one of the Ritz vectors of interest. We have

$$\begin{aligned}\mathrm{K}_m(A, v_1) &= \mathrm{Span}\{y_1, y_2, \ldots, y_k, Ay_1, A^2y_1, \ldots, A^{m-k}y_1\} \\ &= \mathrm{Span}\{y_1, y_2, \ldots, y_k, Ay_i, A^2y_i, \ldots, A^{m-k}y_i\} \quad \text{for each } i = 2, \ldots, k.\end{aligned}$$

Doing a projection onto such a subspace enables us to improve the approximate solution as well as the Ritz vectors. When the convergence of some eigenpairs is reached, the deflation can take place.

We first cover the case with GMRES and make the relation with the recent work of Morgan [12]. We then adapt the technique to FOM and what we call Modified GMRES.

3.1. Implicitly restarted and deflated GMRES (IDGMRES)

We first compute one run of GMRES, which leads to a new starting vector x_0 and the m-length Arnoldi factorization (2.1). From it, we derive a k-length Arnoldi factorization $A\widetilde{V}_k = \widetilde{V}_{k+1}\widetilde{H}_{k+1}$ in such a way that r_0 belongs to $\mathrm{Span}\{\widetilde{V}_{k+1}\}$, where $\mathrm{Span}\{\widetilde{V}_k\} = \mathrm{Span}\{y_1, \ldots, y_k\}$ with $y_1, \ldots, y_k$ being some Ritz vectors of interest that did not yet converge. Through $m - k$ steps of the Arnoldi process, we get a new Arnoldi factorization of length m, from which we can extract a new approximate solution x_m. The process of derivation is then repeated.

Deflation takes place when some Ritz vectors of interest $\tilde{y}_1, \ldots, \tilde{y}_s$ have converged, that is, when $\|A\tilde{y}_i - \lambda_i\tilde{y}_i\|_2$ is negligible. This leads to $\mathrm{Span}\{\widetilde{V}_k\}$ being equal to $\mathrm{Span}\{\tilde{y}_1, \ldots, \tilde{y}_s, y_{s+1}, \ldots y_k\}$ where $\mathrm{Span}\{\widetilde{V}_s\} = \mathrm{Span}\{\tilde{y}_1, \ldots, \tilde{y}_s\}$ is an invariant subspace. We describe in detail how such an algorithm is achieved through the IRA method. We first consider the case where no Ritz pairs converged and then extend it to the case when the deflation takes place.

3.1.1. When no deflation takes place

The very first iteration consists in one run of GMRES. We then express r_m, the next starting residual r_0, with the matrix V_{m+1}. From [14], we have $r_0 = V_{m+1}z_{m+1}$,

where $z_{m+1} = (\zeta_1, \ldots, \zeta_{m+1})^{\mathrm{T}}$. When $\zeta_{m+1} \neq 0$, we can express v_{m+1} in the $(v_i)_{1 \leqslant i \leqslant m}$ and r_0 basis:

$$v_{m+1} = \frac{1}{\zeta_{m+1}}\, r_0 - \sum_{i=1}^{m} \frac{\zeta_i}{\zeta_{m+1}} v_i, \tag{3.1}$$

and the relationship (2.2) becomes

$$\begin{aligned} AV_m &= V_m H_m + h_{m+1,m} \left(\frac{1}{\zeta_{m+1}} r_0 - \sum_{i=1}^{m} \frac{\zeta_i}{\zeta_{m+1}} v_i \right) e_m^{(m)\,\mathrm{T}} \\ &= V_m \underbrace{\begin{pmatrix} & h_{1,m} - \dfrac{\zeta_1}{\zeta_{m+1}} h_{m+1,m} \\ \overline{H}_{m-1} & \vdots \\ & h_{m,m} - \dfrac{\zeta_m}{\zeta_{m+1}} h_{m+1,m} \end{pmatrix}}_{H\mathrm{obl}_m} + \frac{h_{m+1,m}}{\zeta_{m+1}} r_0 e_m^{(m)\,\mathrm{T}}. \end{aligned} \tag{3.2}$$

In the sequel, the scalar d_{m+1} will denote $h_{m+1,m}/\zeta_{m+1}$.

$H\mathrm{obl}_m$ can be viewed as the oblique projected matrix of A onto Span$\{V_m\}$ orthogonal to Span$\{W\mathrm{obl}_m\}$, where the vectors $w\,\mathrm{obl}_i$ of $W\mathrm{obl}_m$ and the v_i's are bi-orthonormal, with the further condition that each $w\,\mathrm{obl}_i$ be orthogonal to r_0. Such a basis could be $w\,\mathrm{obl}_i = v_i - \zeta_i/\zeta_{m+1} v_{m+1}$. We get $H\mathrm{obl}_m = W\mathrm{obl}_m^{\mathrm{T}} A V_m$. Computing $H\mathrm{obl}_m$ through (3.2) is possible when H_m is not singular. We have:

Theorem 3.1. If H_m, the orthogonal projected matrix of A onto the Krylov subspace K_m, is not singular, then ζ_{m+1} never vanishes.

Proof. From (6.35) of [14], we have $z_{m+1} = Q_m^{\mathrm{T}}(\gamma_{m+1} e_{m+1}^{(m+1)})$, where Q_m is the product of rotation matrices that transform $\overline{H}_m$ into an upper triangular matrix. From the relationships (6.28) and (6.32) of [14], we get $\zeta_{m+1} = c_m \gamma_{m+1}$, where c_m is the cosine of the mth rotation matrix. If γ_{m+1} vanishes, then $r_0 = 0$, the solution has been found and the process already stopped at the end of the previous run. Having $c_m = 0$ means that H_m is singular, which proves the result. □

This theorem is true for the very first step when $r_0 = \beta v_1$ as well as for the next ones when $r_0 = V_{k+1} s_{k+1}$ with s_{k+1} belonging to $\mathbb{R}^{k+1}$ such that its last component is not equal to zero.

We next compute the eigenpairs (u_i, μ_i) of $H\mathrm{obl}_m$ and the corresponding Ritz pairs $(y_i = V_m u_i, \mu_i)$. Let us number $\mu_{k+1}, \ldots, \mu_m$ the Ritz values which are not wanted (typically in our case the largest ones in magnitude) and $\mu_1, \ldots, \mu_k$ the ones we are interested in. We process $m-k$ implicit shifted QR-steps on the matrix $H\mathrm{obl}_m$

with the "exact" shifts $\mu_{k+1}, \ldots, \mu_m$. For the first shift μ_{k+1}, we compute the QR factorization of the matrix $H\text{obl}_m - \mu_{k+1} I_m$:

$$H\text{obl}_m - \mu_{k+1} I_m = Q_1 R_1$$

and

$$H\text{obl}_m^1 = Q_1^{\mathrm{T}} H\text{obl}_m Q_1 = R_1 Q_1 + \mu_{k+1} I_m.$$

Although stated in its explicit form, $H\text{obl}_m^1$ is computed in an implicit manner called the implicit shifted QR-iteration (see [6,7]). Because of $H\text{obl}_m$ being upper Hessenberg, Q_1 is still upper Hessenberg and so is $H\text{obl}_m^1$. Shifting equation (3.2) leads to

$$\begin{aligned} AV_m - \mu_{k+1} V_m - V_m(H\text{obl}_m - \mu_{k+1} I_m) &= d_{m+1} r_0 e_m^{(m)\,\mathrm{T}}, \\ (A - \mu_{k+1} I_n) V_m - V_m(Q_1 R_1) &= d_{m+1} r_0 e_m^{(m)\,\mathrm{T}}, \\ (A - \mu_{k+1} I_n)(V_m Q_1) - (V_m Q_1)(R_1 Q_1) &= d_{m+1} r_0 e_m^{(m)\,\mathrm{T}} Q_1, \\ A(V_m Q_1) - (V_m Q_1)(R_1 Q_1 + \mu_{k+1} I_n) &= d_{m+1} r_0 e_m^{(m)\,\mathrm{T}} Q_1, \end{aligned} \tag{3.3}$$

and, finally, we get

$$AV_m^1 - V_m^1 H\text{obl}_m^1 = d_{m+1} r_0 e_m^{(m)\,\mathrm{T}} Q_1 \tag{3.4}$$

with $V_m^1 = V_m Q_1$. When we apply the vector $e_1^{(m)}$ to the right-hand side of equation (3.3), we see that $(A - \mu_{k+1} I_n) v_1 = r_{1,1}^1 v_1^1$, where $r_{i,j}^l$ are the coefficients of the matrix R_l.

Let us repeat this process until $m-k$ shifts have been applied. As explained above this scheme is done in an implicit manner. Denoting $Q = Q_1 \ldots Q_{m-k}$, $V_m^{m-k} = V_m Q$ and $H\text{obl}_m^{m-k} = Q^{\mathrm{T}} H\text{obl}_m Q$, the repeated process leads to

$$AV_m^{m-k} - V_m^{m-k} H\text{obl}_m^{m-k} = d_{m+1} r_0 e_m^{(m)\,\mathrm{T}} Q \tag{3.5}$$

with $v_1^{m-k} = V_m^{m-k} e_1^{(m)}$ being a multiple of $\prod_{i=1}^{m-k} (A - \mu_{k+i} I_n) v_1$.

Because of the Hessenberg form of Q_i for $i = 1, \ldots, m-k$, $H\text{obl}_m^{m-k}$ is still upper Hessenberg. Partitioning the matrices $V_m^{m-k} = V_m Q$ and $H\text{obl}_m^{m-k}$ as is done in [17, p. 363],

$$V_m^{m-k} = \big[\widehat{V}_k, \widetilde{V}_{m-k}\big], \qquad H\text{obl}_m^{m-k} = \begin{pmatrix} \widehat{H\text{obl}}_k & \widetilde{M} \\ \widehat{\beta} e_1^{(m-k)} e_k^{(k)\,\mathrm{T}} & \widetilde{H\text{obl}}_{m-k} \end{pmatrix}$$

and writing

$$e_m^{(m)\,\mathrm{T}} Q = (0, \ldots, 0, \alpha_k, \ldots, \alpha_m),$$

as well as equating the first k columns of (3.5), leads to

$$A\widehat{V}_k - \widehat{V}_k \widehat{H\text{obl}}_k = \big(d_{m+1} \alpha_k r_0 + \widehat{\beta} \widetilde{V}_{m-k} e_1^{(m-k)}\big) e_k^{(k)\,\mathrm{T}}. \tag{3.6}$$

Theorem 3.2. If we take the shifts to be some eigenvalues $(\mu_{k+i})_{1 \leqslant i \leqslant m-k}$ of $H\text{obl}_m$ then $\widehat{\beta}$ from (3.6) equals zero and $\text{Span}\{\widehat{V}_k\} = \text{Span}\{y_1, \ldots, y_k\}$, where $(y_i)_{1 \leqslant i \leqslant k}$ are

the Ritz vectors associated to the Ritz values $(\mu_i)_{1\leqslant i\leqslant k}$ of $H\text{obl}_m$ that are not taken as a shift in the implicit shifted QR-iteration.

Proof. We first prove the first assertion: $H\text{obl}_m$ is not singular because it can be expressed as the product of

$$\underbrace{\begin{pmatrix} 1 & & & -\zeta_1/\zeta_{m+1} \\ & \ddots & & \vdots \\ & & \ddots & \vdots \\ & & 1 & -\zeta_m/\zeta_{m+1} \end{pmatrix}}_{C} \times \overline{H}_m,$$

where C is an $m \times (m+1)$ matrix such that $\text{Ker}\{C\} = \text{Span}\{z_{m+1}\}$. We have $\text{Ker}\{\overline{H}_m\} = \text{Ker}\{H_m\} = \{0\}$ because H_m is not singular. This implies that

$$\text{Ker}\{H\text{obl}_m\} = \big\{z \in \mathbb{R}^m \mid \overline{H}_m z = z_{m+1} \text{ or } \overline{H}_m z = 0\big\}.$$

Solving $\overline{H}_m z = z_{m+1}$ is equivalent to solve

$$Q_m \overline{H}_m z = Q_m z_{m+1} = \gamma_{m+1} e_{m+1}^{(m+1)}.$$

But γ_{m+1} does not vanish, otherwise we would have found the solution. Since the last line of the matrix $Q_m\overline{H}_m$ is null, it implies that there does not exist any z satisfying $\overline{H}_m z = z_{m+1}$ and $H\text{obl}_m$ is not singular.

The matrix $H\text{obl}_m - \mu_{k+1} I_m$ is singular since μ_{k+1} is one of the eigenvalues of $H\text{obl}_m$. But $H\text{obl}_m - \mu_{k+1} I_m = Q_1 R_1$ is an unreduced matrix, where its first $m-1$ columns are independent and $r^1_{i,i} \neq 0$, for $i = 1, \ldots, m-1$. $H\text{obl}_m - \mu_{k+1} I_m$ being singular implies that $r^1_{m,m} = 0$ and

$$H\text{obl}^1_m = R_1 Q_1 + \mu_{k+1} I_m = \begin{pmatrix} \widehat{H\text{obl}}_{m-1} & \widetilde{M} \\ 0 & \mu_{k+1} \end{pmatrix}.$$

$\widehat{H\text{obl}}_{m-1}$ is still an unreduced upper Hessenberg matrix. It was computed such that the elements of its subdiagonal are strictly positive. Following the same argumentation with $m-k-1$ more shifts leads to $\widehat{\beta} = 0$ and $\widetilde{H\text{obl}}_{m-k}$ being an upper triangular matrix. The first assertion is proved.

For the second assertion, we first prove that for each $1 \leqslant j \leqslant m$, there exists a vector w_j such that $Qe_j^{(m)} = 1/\tau_j \prod_{i=1}^{m-k} (H\text{obl}_m - \mu_{k+i} I_m) w_j$. We will prove it by induction.

1. For $j = 1$: From $H\text{obl}_m - \mu_{k+1} I_m = Q_1 R_1$ and $H\text{obl}^{j-1}_m - \mu_{k+j} I_m = Q_j R_j$, we get

$$r^1_{1,1} Q_1 e_1^{(m)} = (H\text{obl}_m - \mu_{k+1} I_m) e_1^{(m)}$$

and

$$Q_{j-1}^{\mathrm{T}} \cdots Q_1^{\mathrm{T}} (H\text{obl}_m - \mu_{k+j} I_m) Q_1 \cdots Q_{j-1} e_1^{(m)} = r^j_{1,1} Q_j e_1^{(m)}.$$

We finally have

$$(H\text{obl}_m - \mu_{k+j})Q_1 \cdots Q_{j-1} e_1^{(m)} = r_{1,1}^j Q_1 \cdots Q_{j-1} Q_j e_1^{(m)},$$

which leads by induction to the wanted property for $Qe_1^{(m)}$.

2. For $j > 1$: If $Q_1 e_l^{(m)}$ for $l = 1, \ldots, i-1$ can be expressed in terms of $(H\text{obl}_m - \mu_{k+1} I_m) w_l$, it is obvious that $Q_1 e_i^{(m)}$ can also been expressed in the same manner. Now let us suppose that for $i = 1, \ldots, j-1$, $Q_1 \cdots Q_i e_l^{(m)}$ for $1 \leqslant l \leqslant m$ satisfy the wanted property as well as $Q_1 \cdots Q_j e_l^{(m)}$ for $1 \leqslant l \leqslant k-1$, then we have

$$\begin{aligned}
\big(H\text{obl}_m^{j-1} - \mu_{k+j}\big) e_k^{(m)} &= r_{k,k}^j Q_j e_k^{(m)} + \sum_{i=1}^{k-1} r_{i,k}^j Q_j e_i^{(m)}, \\
(H\text{obl}_m - \mu_{k+j}) Q_1 \cdots Q_{j-1} e_k^{(m)} &= r_{k,k}^j Q_1 \cdots Q_{j-1} Q_j e_k^{(m)} \\
&\quad + \sum_{i=1}^{k-1} r_{i,k}^j Q_1 \cdots Q_{j-1} Q_j e_i^{(m)}.
\end{aligned}$$

Then $Q_1 \cdots Q_j e_k^{(m)}$ can be expressed as $\prod_{i=1}^{j} (H\text{obl}_m - \mu_{k+i} I_m)$ times a vector. By induction we get that each vector $Qe_j^{(m)}$ for $1 \leqslant j \leqslant m$ can be expressed as $\prod_{i=1}^{m-k} (H\text{obl}_m - \mu_{k+i} I_m)$ times a vector.

Each vector $Qe_i^{(m)}$ for $1 \leqslant i \leqslant m$ has been purified from $u_{k+1}, \ldots, u_m$ and can be expressed in the $u_1, \ldots, u_k$ basis. Since

$$H\text{obl}_m Q\big(e_1^{(m)}, \ldots, e_k^{(m)}\big) = Q\big(e_1^{(m)}, \ldots, e_k^{(m)}\big) \widehat{H\text{obl}}_k,$$

we have

$$\begin{aligned}
&\dim\big(\text{Span}\big\{Q(e_1^{(m)}, \ldots, e_k^{(m)}\big\}\big) = k \quad \text{and} \\
&\text{Span}\big\{Q\big(e_1^{(m)}, \ldots, e_k^{(m)}\big)\big\} = \text{Span}\{u_1, \ldots, u_k\},
\end{aligned}$$

which means that $\text{Span}\{\widehat{V}_k\} = \text{Span}\{y_1, \ldots, y_k\}$. □

When using the previous theorem, the relationship (3.6) becomes:

$$A\widehat{V}_k = \widehat{V}_k \widehat{H\text{obl}}_k + d_{m+1} \alpha_k r_0 e_k^{(k)\,\mathrm{T}}. \tag{3.7}$$

Taking $w = r_0/\beta$ with $\beta = \|r_0\|_2$, and orthogonalizing it against $\widehat{V}_k$, we get $\widetilde{w} = w - \widehat{V}_k s_k$ with $s_k = \widehat{V}_k^{\mathrm{T}} w = (\sigma_1, \ldots \sigma_k)^{\mathrm{T}}$. Noting $\sigma_{k+1} = \|\widetilde{w}\|$, $\widehat{v}_{k+1} = \text{sign}(\alpha_k)\widetilde{w}/\sigma_{k+1}$ and $\widehat{h}_{k+1,k+1} = d_{m+1}|\alpha_k|\beta\sigma_{k+1}$, we have

$$A\widehat{V}_k = \widehat{V}_k \widehat{H\text{obl}}_k + d_{m+1}\alpha_k\beta \left(\text{sign}(\alpha_k)\sigma_{k+1}\widehat{v}_{k+1} + \sum_{i=1}^{k} \sigma_i \widehat{v}_i \right) e_k^{(k)\,\mathrm{T}}$$

$$= \widehat{V}_k \underbrace{\begin{pmatrix} & \widehat{h\,\mathrm{obl}}_{1,k} + d_{m+1}\alpha_k\beta\sigma_1 \\ \widehat{H\mathrm{obl}}_{k-1} & \vdots \\ & \widehat{h\,\mathrm{obl}}_{k,k} + d_{m+1}\alpha_k\beta\sigma_k \end{pmatrix}}_{\widehat{H}_k} + \widehat{h}_{k+1,k+1}\widehat{v}_{k+1}e_k^{(k)\mathrm{T}}. \quad (3.8)$$

This is an Arnoldi factorization of length k because:

- $\widehat{H}_k$ is an unreduced upper Hessenberg matrix because $\widehat{H\mathrm{obl}}_k$ is unreduced,
- $[\widehat{V}_k, \widehat{v}_{k+1}]$ form an orthonormal basis.

This Arnoldi factorization can be extended to length m by applying $m-k$ Arnoldi steps. Removing the hats, this leads to the relationship (2.1). We have

$$\mathrm{Span}\{V_m\} = \mathrm{Span}\{v_1, \ldots, A^{m-1}v_1\} \quad (3.9)$$

$$= \mathrm{Span}\{y_1, \ldots, y_k, v_{k+1}, \ldots, A^{m-k-1}v_{k+1}\}. \quad (3.10)$$

But v_{k+1} belongs to $\mathrm{Span}\{y_1, \ldots, y_k, r_0\}$ and because of (3.2), $Ay_i = \mu_i y_i + d_{m+1}u_i(m)r_0$ and each Ay_i belongs to $\mathrm{Span}\{y_i, r_0\}$. We then have that Av_{k+1} belongs to $\mathrm{Span}\{y_1, \ldots, y_k, r_0, Ar_0\}$. By induction, we get

$$\mathrm{Span}\{V_m\} = \mathrm{Span}\{y_1, \ldots, y_k, r_0, \ldots, A^{m-k-1}r_0\}. \quad (3.11)$$

On the other hand, r_0 can be expressed in the $\mathrm{Span}\{y_i, Ay_i\}$ basis for each i, and by induction we have

$$\mathrm{Span}\{V_m\} = \mathrm{Span}\{y_1, \ldots, y_k, Ay_i, \ldots, A^{m-k}y_i\} \quad (3.12)$$

for each $i = 1, \ldots, k$.

We can now extract a GMRES solution from $\mathrm{Span}\{V_m\}$. We have $x_m = x_0 + V_m y_m$, where

$$y_m = \arg\min_{y\in\mathbb{R}^m} \left\| \beta\left(s_k^{\mathrm{T}}, \mathrm{sign}(\alpha_k)\sigma_{k+1}, 0\ldots, 0\right)^{\mathrm{T}} - \overline{H}_m y \right\|_2$$

and we repeat the process.

3.1.2. When deflation takes place

So far, we have not taken into account the cases where some Ritz pairs converge. Let us suppose that in a given run after computing equation (3.2), l Ritz pairs converged for the very first time. Let us call $(\tilde{y}_i = V_m\tilde{u}_i, \tilde{\mu}_i)_{1\leqslant i\leqslant l}$ the Ritz pairs that converged. They satisfy

$$A\tilde{y}_i - \tilde{\mu}_i\tilde{y}_i = d_{m+1}r_0 e_m^{(m)\mathrm{T}}\tilde{u}_i,$$

where $|d_{m+1}r_0 e_m^{(m)\mathrm{T}}\tilde{u}_i|$ is considered negligible.

We separate the converged Ritz pairs into two groups: the l_1 first ones we are interested in and the other $l_2 = l - l_1$ ones which are not wanted. We suppose that $l_1 \leqslant k$.

We are going to deflate the l_1 wanted Ritz values as well as the l_2 unwanted ones. Deflating them means that after processing the IRA method with deflation, the matrix $\widehat{V}_k$ will span the subspace Span$\{\tilde{y}_1, \ldots, \tilde{y}_{l_1}, y_{l_1+1}, \ldots, y_k\}$ such that Span$\{\widehat{V}_{l_1}\} = \{\tilde{y}_1, \ldots, \tilde{y}_{l_1}\}$ is an invariant subspace and $y_{l_1+1}, \ldots, y_k$ are the Ritz vectors of the current run we are interested in but which did not yet converge. We have then $A\widehat{V}_{l_1} = \widehat{V}_{l_1}\widehat{T}$, where $\widehat{T}$ is an upper quasi-triangular matrix. When extending the Arnoldi factorization to length m, we have that each new vector of the basis is orthogonal to this invariant subspace. This is called *deflation* and it is done implicitly.

To do it, we first process what is called a *Lock* process in [9]. By applying an orthonormal matrix whose columns span the set of the converged Ritz vectors to the right-hand side of equation (3.2), we get

$$A(V_m Q) = (V_m Q)\big(Q^{\mathrm{T}} H\mathrm{obl}_m Q\big) + d_{m+1} r_0 e_m^{(m)\,\mathrm{T}} Q$$

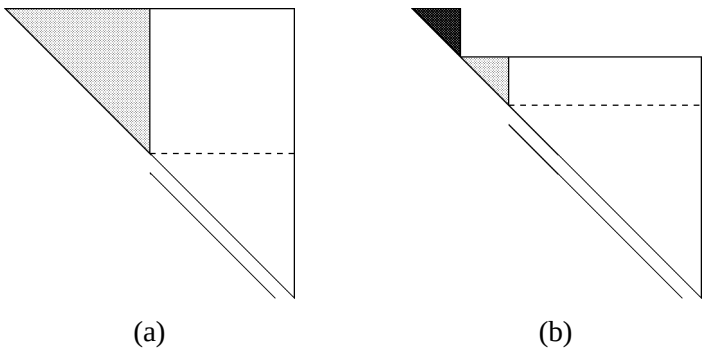
such that the $l \times l$ upper left part of $Q^{\mathrm{T}} H\mathrm{obl}_m Q$ is a quasi-triangular matrix associated with the invariant subspace Span$\{V_m Q[e_1^{(m)}, \ldots, e_l^{(m)}]\} = \{\tilde{y}_1, \ldots, \tilde{y}_l\}$. In order to recover the upper Hessenberg form of $H\mathrm{obl}_m$, we apply the matrix Q', product of Householder matrices, to the right and left sides of $Q^{\mathrm{T}} H\mathrm{obl}_m Q$. The matrix $(QQ')^{\mathrm{T}} H\mathrm{obl}_m(QQ')$ is now upper Hessenberg.

We rename $V_m = V_m QQ'$ and $H\mathrm{obl}_m = (QQ')^{\mathrm{T}} H\mathrm{obl}_m(QQ')$. Because of the convergence of the Ritz pairs $\tilde{y}_1, \ldots, \tilde{y}_l$, we have $e_m^{(m)\,\mathrm{T}} Q = e_m^{(m)\,\mathrm{T}} + w^{\mathrm{T}}$, where $\|w\|_2$ is negligible. We have $e_m^{(m)\,\mathrm{T}} Q' = e_m^{(m)\,\mathrm{T}}$ too. We then get the standard Arnoldi factorization (3.2), where $H\mathrm{obl}_m$ is of the same shape as figure 1(a).

We now apply $m - (k + l_2)$ shifts on the Arnoldi factorization to get

$$A\widehat{V}_{k+l_2} = \widehat{V}_{k+l_2}\widehat{H\mathrm{obl}}_{k+l_2} + d_{m+1}\alpha_{k+l_2} r_0 e_{k+l_2}^{(k+l_2)\,\mathrm{T}},$$

where $\widehat{H\mathrm{obl}}_{k+l_2}$ is a matrix of size $(k+l_2)\times(k+l_2)$, of the same shape as figure 1(a), the triangular part being of size $l \times l$.



(a) (b)

Figure 1. The matrix $H\mathrm{obl}_m$ after (a) the *Lock* process, and (b) the *Purge* process.

We now throw the unwanted Ritz pairs away through the *Purge* process [9]. By a *Lock* process we transform $\widehat{V}_{k+l_2}$ and $\widehat{H\text{obl}}_{k+l_2}$ such that

$$\text{Span}\{\widehat{V}_{k+l_2}[e_1^{(k+l_2)}, \ldots, e_{l-l_1}^{(k+l_2)}]\} = \{\tilde{y}_{l_1+1}, \ldots, \tilde{y}_l\} \tag{3.13}$$

and

$$\text{Span}\{\widehat{V}_{k+l_2}[e_1^{(k+l_2)}, \ldots, e_l^{(k+l_2)}]\} = \{\tilde{y}_1, \ldots, \tilde{y}_l\} \tag{3.14}$$

and the $l \times l$ upper left block of $\widehat{H\text{obl}}_{k+l_2}$ is quasi-triangular.

By solving a Sylvester equation, we transform this Arnoldi factorization into another one where the relationships (3.13) and (3.14) are still valid and $\widehat{H\text{obl}}_{k+l_2}$ is of the same shape as figure 1(b).

We equate the k last columns of the factorization which leads to a k-length factorization similar to the relationship (3.7), with $\widehat{H\text{obl}}_k$ being of the same shape as figure 1(a) where its triangular part is of size $l_1 \times l_1$.

When $l_1 \geqslant k$, the "real" deflation can take place. We have an invariant subspace we are interested in and are going to keep unchanged through the next runs. To do this, we keep the k most interesting Ritz pairs among the l_1 ones that converged and apply a *Lock* process with them to get an Arnoldi factorization of length m such that $H\text{obl}_m$ is of the same shape as figure 1(a), its upper quasi-triangular part being of size $k \times k$. We equate the fist k columns of this Arnoldi factorization, which leads to the equation $AV_k = V_k T_k$, where T_k is quasi-triangular. We next orthonormalize the residual r_0 against V_k to form the vector v_{k+1} and process $m - k$ steps of the Arnoldi process. In the next runs, we no longer perform the *Lock*, implicit shifted QR and *Purge* process, but orthogonalize each new residual against the invariant subspace.

Suppose now that we were able to compute, through the previous runs, an invariant subspace of size t with $t < k$ and that, during the next run, l' new Ritz pairs converged. The whole deflation technique described above remains unchanged and we just have to take into account the l' new Ritz pairs within the l that converged. We thus have $l = t + l'$ and we can split the l Ritz pairs that converged into two groups, the ones we are interested in and the others. This enables us to discard the Ritz pairs we were interested in in the previous runs but no longer want to take into account in the next ones, by purging them.

The resulting IDGMRES is summarized below:

Algorithm $[x_0, r_0, V_m, \overline{H}_m, \textit{iter}] = \text{IDGMRES(A}, x_0, b, m, k, \textit{maxit}, \textit{tol})$

1. Initialization:
   (a) $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$, $\textit{iter} = 0$, $t = 0$
   (b) One run of GMRES(A, $x_0, b, m,$ *tol*): $x_0 = x_m$, $r_0 = r_m$
   (c) $\textit{iter} = \textit{iter} + 1$, if ($\|r_0\|_2/\beta < \textit{tol}$) or ($\textit{iter} = \textit{maxit}$) stop

2. **If** $t < k$ **then**
   (a) Compute $H\text{obl}_m$
   (b) Compute the Ritz pairs ($y_i = V_m u_i, \mu_i$) corresponding to the eigenpairs of $H\text{obl}_m$ (including the Ritz pairs of the invariant subspace)
   (c) Split the Ritz pairs:
      (i) Choose l_1 (not exceeding k) Ritz pairs that converged and we are interested in. Update $t = l_1$
      (ii) **If** $t < k$ **then** split the remaining Ritz pairs into:
         (A) the l_2 unwanted converged Ritz pairs
         (B) the $k - l_1$ Ritz pairs that did not converge but that we are interested in
         (C) the others

      Else $l_2 = 0$
      End If
   (d) If $l_1 + l_2 \neq 0$ apply the *Lock* process
   (e) **If** $t < k$ **then**
      (i) Compute $\widehat{H\text{obl}}_{k+l_2}$ and $\widehat{V}_{k+l_2}$, the resulting matrices of the $m - k - l_2$ implicit shifted QR-steps
      (ii) If $l_2 \neq 0$ apply the *Purge* process
      (iii) Compute $\widehat{H}_k$ and $\widehat{v}_{k+1}$ from (3.8). Set $H_k = \widehat{H}_k$, $V_{k+1} = \widehat{V}_{k+1}$

      Else compute v_{k+1} by orthonormalizing r_0 against V_k
      End If

   Else compute v_{k+1} by orthonormalizing r_0 against V_k
   End if
3. Compute $m - k$ steps of the Arnoldi process
4. Compute the approximate solution:
   (a) Solve $\tilde{y} = \arg\min \|V_{m+1}^{\mathrm{T}} r_0 - \overline{H}_m y\|_2$
   (b) $x_0 = x_0 + V_m \tilde{y}$
   (c) $r_0 = b - A x_0$
5. Restart:
   (a) *iter* = *iter* + 1
   (b) if ($\|r_0\|_2/\beta <$ *tol*) or (*iter* = *maxit*) stop, else go to 2.

We have presented an algorithm that uses the IRA method to reduce the drawbacks of restarting. From the same Krylov subspace, we can improve in each run the approximation of the invariant subspace we are interested in as well as the approximate solution.

Recently, Morgan developed an algorithm based on the same idea with a different approach, that he called GMRES-IR [12]: From an Arnoldi factorization of length m, he computes the eigenpairs of the harmonic eigenvalue problem (2.6). He then takes the unwanted harmonic Ritz values as shifts in the QR-iteration that he applies to the factorization $AV_m = V_{m+1}\overline{H}_m$. The main differences with our approach are that:

1. We compute the eigenpairs of the matrix $H\text{obl}_m$ instead of the harmonic eigenvalue problem.
2. Using the IRA method, we apply exact shifts since we do the QR-iteration on equation (3.2) and take the eigenvalues of $H\text{obl}_m$ as shifts. The shifts used within the QR-iteration of Morgan are not exact since they are part of the harmonic Ritz values and the QR-iteration is applied to $AV_m = V_{m+1}\overline{H}_m$.
3. We include the *Lock* and *Purge* deflation that enhances the stability of the algorithm.
4. Our algorithm is less expensive from a computational point of view.

If we have a more accurate look at the first item, we can demonstrate that the eigenvalues of the matrix $H\text{obl}_m$ are exactly the harmonic eigenvalues.

Theorem 3.3. Solving $H\text{obl}_m u = \lambda u$, where $H\text{obl}_m = W\text{obl}_m^{\mathrm{T}} A V_m$, is mathematically equivalent to solving the harmonic eigenvalue problem (2.6).

Proof. In [12], Morgan has demonstrated that solving the harmonic eigenvalue problem (2.6) is equivalent to solving the following one:

$$\big(H_m + h_{m+1,m}^2 f e_m^{(m)\mathrm{T}}\big)u = \lambda u,$$

where $f = H_m^{-\mathrm{T}} e_m^{(m)}$. Furthermore, we have $H\text{obl}_m = H_m - h_{m+1,m} v e_m^{(m)\mathrm{T}}$ where $v = (1/\zeta_{m+1})(\zeta_1, \ldots, \zeta_m)^{\mathrm{T}}$ with $z_{m+1} = (\zeta_1, \ldots, \zeta_{m+1})^{\mathrm{T}} = V_{m+1}^{\mathrm{T}} r_0 - \overline{H}_m y_m$. But

$$\big(H_m + h_{m+1,m}^2 f e_m^{(m)\mathrm{T}}\big)y_m = \big(V_m^{\mathrm{T}} + h_{m+1,m} f v_{m+1}^{\mathrm{T}}\big)r_0,$$

which implies that

$$v = \big(V_m^{\mathrm{T}} r_0 - H_m y_m\big)/\big(v_{m+1}^{\mathrm{T}} r_0 - h_{m+1,m} e_m^{(m)\mathrm{T}} y_m\big) = -h_{m+1,m} f.$$

Solving the harmonic eigenvalue is then equivalent to solving $H\text{obl}_m u = \lambda u$. □

Solving the harmonic eigenvalue problem is particularly efficient to approximate eigenpairs whose eigenvalues are close to zero. We can therefore expect that computing the Ritz pairs from $H\text{obl}_m u = \lambda u$ will give good results and enhance the convergence.

3.2. Implicit deflation for other Krylov type methods

By a slight modification, we extend the idea of implicitly deflating GMRES to FOM and what we called Modified GMRES.

Concerning FOM, the process is simplified since we have the well-known relationship $r_m = \alpha_m v_{m+1}$ [14, (6.16)]. We do not have to form the matrix $H\text{obl}_m$ but can keep H_m and apply the IRA method on it. The resulting algorithm, called IDFOM, is very similar to IDGMRES except for the following:

- 1(b) One run of FOM(A,x_0,b,m,*tol*): $x_0 = x_m$, $r_0 = r_m$.
- 2(a), (b) Compute the Ritz pairs of H_m.
- 2(e) (i) Compute $\overline{H}_{k+l_2}$ and V_{k+l_2}, the resulting matrices of the $m-k-l_2$ implicit shifted QR-steps.
- 2(e) (iii) Does not exist.
- 4(a) Compute $H_m\tilde{y} = V_m^{\mathrm{T}} r_0$.

We next introduce MGMRES. We are looking for a solution that satisfies a minimal condition on the residual norm with the additional constraint for the residual to be orthogonal to the previous one. It reminds us of the suitable properties of the Conjugate Gradient that offers a minimal condition on the error as well as a short-term recurrence leading to orthogonal residuals. However, our solution does not involve a short recurrence and the minimal condition is a residual one.

We are looking for the solution $x_m = x_0 + V_m y_m$ such that

$$y_m = \arg \min_{y \in \mathbb{R}^m \text{ and } (r_0, r_0 - AV_m y) = 0} \|r_0 - AV_m y\|_2.$$

This is equivalent to solving

$$y_m = \left(\beta \Big/ \sum_{i=1}^{m} h_{1,i}\ y_m^G(i) \right) y_m^G, \tag{3.15}$$

where y_m^G is the GMRES solution of equation (2.3). Fore more details, see [8]. We can see that the resulting algorithm MGMRES needs the computation of the GMRES vector y_m^G and induces very few modifications compared to the GMRES algorithm itself.

Like we derived IDGMRES from GMRES, we can derive an Implicitly restarted and Deflated Modified GMRES from MGMRES. The resulting algorithm IDGMRES is almost the same as IDMGRES except for the computation of y_m that has to be multiplied by a scalar. This slight change induces a surprising acceleration in convergence. More details can be found in [8].

4. Comparison costs and numerical results

4.1. Computational and storage requirements

We now compare the methods GMRES, FOM, MGMRES, AGMRES, IDFOM, IDGMRES and IDMGMRES from a storage point of view, as well as from a computational one. We compare the methods for an equal size m of the subspace onto which the projection is done and from which the approximate solution is extracted. Since we deal with large sparse matrices, we will make the assumption that $m \ll n$, and will not take into account arrays of order m.

Concerning the method AGMRES, we will consider two implantations for which we build the same subspace $S_{m,k}(A, r_0, y_1, \ldots, y_k)$. The first one is called AGMRES+: instead of doing the k matrix–vector products Ay_i for $i = 1, \ldots, k$, AGMRES+ computes them in a cheaper way thanks to equation (2.6) and stores them for the next run. On the contrary, AGMRES does not require any more storage and explicitly computes the matrix–vector products with A.

The storage requirements of all the methods are summarized below:

Storage	GMRES	AGMRES	AGMRES+
Vector of size n	$m+4$	$m+4+k$	$m+4+2k$

where GMRES stands for GMRES, FOM, MGMRES, IDGMRES, IDFOM and IDMGMRES.

The computational requirements are summarized below:

Computation	GMRES	IDGMRES	AGMRES/AGMRES+
Matvec product with A	m	$m-k$	m / $m-k$
DDOT of length n	$\frac{(m+1)(m+2)}{2}$	$\frac{(m+1)(m+2)-k(k+1)}{2}$	$\frac{(m+1)(m+2)}{2}+(m-k)k$
DAXPY of length n	$\frac{m(m+1)}{2}$	$\frac{(m+1)(m+6)-(k-1)k}{2}$	$\frac{m(m+1)}{2}$
Matvec product of size $(n \times s) \times s$	2	(1) 2 (2) $2+2m-2k$ (3) $2+4m-3k$	$2+m$ / $2+m+k$

where s is of the same magnitude as k or m and GMRES stands for GMRES, FOM and MGMRES, and IDGMRES stands for IDGMRES, IDFOM and IDMGMRES.

The item numbered (2) that appears within the implicitly deflated methods only takes place when the *Lock* process is applied. The third item must be taken into account when both the *Lock* and *Purge* process take place. We point out that such a case only appears at the very last runs when the approximate solution almost converges. The

Lock and *Purge* process do not need to be taken into account for a great part of the runs.

We can see that IDGMRES, IDFOM and IDMGMRES are among the cheapest methods. They save matrix–vector products with A and dot products within the Arnoldi process and do not require further storage. This is not the case for the Augmented GMRES which can save either storage or matrix–vector products, but not both. The lowest storage method of Augmented GMRES is still more expensive than GMRES, FOM, MGMRES, IDFOM, IDGMRES and IDMGMRES.

4.2. Numerical experiments

The numerical experiments have been done in Matlab and follow the same scheme.

The right-hand side is the unit vector and the initial one is zero. The process is stopped whenever $\|r\|_2/\beta < \mathit{tol}$ with $\mathit{tol} = 1.0\text{E}{-9}$ or the maximum number of iterations $\mathit{maxit} = 200$ is reached. In the tables, m stands for the length of the Arnoldi factorization, whereas k is the number of Ritz vectors taken into account within each run for the methods IDFOM, IDGMRES, IDMGMRES, AGMRES and AGMRES+. Because of complex conjugate eigenvalues, this number of wanted Ritz vectors is either equal to k or $k+1$, depending on the run. The *Iter* and *Reduct*. data respectively show the number of runs reached and the reduction of the residual norm achieved ($\|r\|_2/\beta$) when the process stopped. The *Matvec* data shows the total number of matrix–vector products with A computed during the whole process. The *tolvp* parameter is the tolerance used within the stopping criterion for the convergence of the Ritz pairs (y_i, μ_i). We consider that they converged if

$$\frac{\|Ay_i - \mu_i y_i\|_2}{\|y_i\|_2} < \mathit{tolvp}.$$

We took *tolvp* equal to 1.0E−06.

Although our algorithm was developed to be processed on large sparse matrices, we present here matrices of relatively small size. This is due to the fact that we made our tests in Matlab on a PC Pentium II to see the convergence behavior of the new methods. The assumption of the latter section, $m \ll n$, is not verified, and we focus on the most expensive operations if we were using large sparse matrices, typically the matrix–vector products with A.

We present two matrices. The first one was introduced in Morgan's paper [10] to compare the Augmented GMRES to GMRES itself. The second one comes from the Harwell–Boeing collection.

For each of the matrices tested, we compare the IDFOM, IDGMRES and IDMGMRES methods with GMRES, FOM, MGMRES, AGMRES and AGMRES+, for an equal length m of the Arnoldi factorization. We made the experiments for m equal to 20, 30, 40 and 50, whereas k was taken equal to 1, 3 or 6. We also made some comparisons with the Bi-CGSTAB and QMR methods. The latter is implanted in its

Table 1
Matrix EX1, *tol* = 10E−9, *maxit* = 200, Precon = no.

m	Perf.	GMRES	FOM	MGMRES
20	Iter	200	200	200
	Matvec	4001	4001	4001
	Reduct.	2.1929E−02	3.9344E−00	7.3190E−02
30	Iter	200	200	200
	Matvec	6001	6001	6001
	Reduct.	2.0120E−02	1.4008E−03	2.7226E−02
40	Iter	200	200	200
	Matvec	8001	8001	8001
	Reduct.	2.0594E−02	1.7539E+19	2.3135E−02
50	Iter	200	182	200
	Matvec	10001	9064	10001
	Reduct.	2.0077E−02	8.6687E−10	3.1710E−02

nontranspose-free version and involves matrix–vector products with A^{T} too. A maximum of 2000 iterations were allowed for the last two methods as well as for the Full version of GMRES and FOM.

In the sequel, we will call nb_{method} the number of matrix–vector products with A achieved by the *method* to converge. This involves the matrix–vector products with A^{T} for the QMR residual.

First matrix. The first matrix shows the problems raised by restarting when using GMRES or FOM for a positive definite matrix with its four smallest eigenvalues of slightly different magnitude than the others. The matrix is a bidiagonal one of size 1000, with 0.1 on each position of the super diagonal. The values on its main diagonal are 0.01, 0.02, 0.03, 0.04 for the four first indices and $10, \ldots, 1005$ for the next ones. The Full GMRES converges in 228 matrix–vector products while the Full FOM converges in 230.

As is shown in table 1, GMRES and MGMRES do not converge for all the cases tested, and FOM only does for $m = 50$. In fact, the residual norms of GMRES and FOM stagnate, whereas the MGMRES ones oscillate between two values. This is due to the restart and their difficulty to approximate the eigenvalues that are close to zero.

The Bi-CGSTAB and QMR methods (see table 2) converge very fast in number of runs like the Full version of GMRES or FOM, but they involve twice as many matrix–vector products.

In view of table 3, we can make the following remarks:

- Except for IDFOM, $k = 1$ is not sufficient to approximate the eigenvalues closest to zero well.
- As k increases, the number of matrix–vector products of all the deflation methods decreases. Taking $k = 3$ instead of $k = 1$ leads to a fast convergence instead of a

Table 2
Matrix EX1, *tol* = 10E−9, *maxit* = 2000, Precon = no.

Perf.	QMR	Bi-CGSTAB
Iter	227	399
Matvec A	228	799
Matvec A^{T}	227	0
Reduct.	8.9750E−10	7.4079E−10

stagnant process, and taking $k = 6$ accelerates this convergence too. The eigenpairs closest to zero are faster accurately approximated as k increases but never enough to deflate. Taking k bigger does slightly improve the process and sometimes slows it down, but enables the deflation of the Ritz eigenpairs. For example, taking k equal to $m/2$ does not greatly reduce the number of matrix–vector products, but at the end of the process, when the methods converged, so did few Ritz eigenpairs associated with the eigenvalues closest to zero, although no deflation occured for the cases shown in table 3.

- AGMRES and AGMRES+ almost perform equivalently in number of runs whereas the number of matrix–vector products of AGMRES+ is much less than for AGMRES. However, AGMRES+ requires more storage. We have

$$nb_{\text{AGMRES}} \geqslant nb_{\text{AGMRES+}}.$$

- IDGMRES converges faster than AGMRES+ (except for $m = 50$ and $k = 6$). AGMRES+ becomes competitive with IDGMRES only when $k = 6$. For all the cases tested except $m = 50$, $k = 6$, we have

$$nb_{\text{AGMRES+}} \geqslant nb_{\text{IDGMRES}}.$$

- We have for $k = 6$ and all the m tested

$$nb_{\text{AGMRES}} \geqslant nb_{\text{IDFOM}} \geqslant nb_{\text{IDGMRES}}.$$

- The IDMGMRES method does not perform better than IDGMRES.
- The IDGMRES exhibits a more regular convergence behavior than IDFOM, for which the convergence is unpredictable.
- For $m \geqslant 40$ and $k = 3$, 6, the number of matrix–vector products of AGMRES, AGMRES+, IDGMRES and IDMGMRES is less than for the Bi-CGSTAB and QMR methods.
- The smallest number of matrix–vector products is reached by AGMRES+ for $m = 50$, $k = 6$ in 244 matrix–vector products and IDGMRES for $m = 50$ and $k = 6$ in 246 matrix–vector products. Other tests showed that for $m = 40$ and $k = 10$, IDGMRES converged in 237 matrix–vector products, which is almost the number of matrix–vector products for the Full GMRES method to converge.

Table 3
Matrix EX1, *tol* = 10E−9, *tolvp* = 10E−6, *maxit* = 200, Precon = no.

m/k		AGMRES	AGMRES+	IDFOM	IDGMRES	IDMGMRES
20/1	Iter	200	200	200	200	200
	Matvec	4200	3801	3802	3802	3712
	Reduct.	2.0254E−02	2.0254E−02	7.1219E−01	2.0250E−02	4.0363E−02
20/3	Iter	168	172	200	96	78
	Matvec	3516	2912	3404	1633	1327
	Reduct.	9.9770E−10	9.9918E−10	6.7017E−07	9.8248E−10	5.4268E−10
20/6	Iter	20	20	19	19	26
	Matvec	397	269	272	268	366
	Reduct.	9.0890E−10	9.0890E−10	7.8643E−10	9.4811E−10	2.4981E−10
30/1	Iter	200	200	200	200	200
	Matvec	6200	5801	5802	5802	5703
	Reduct.	2.0077E−02	2.0077E−02	7.1017E−08	2.0077E−02	2.1759E−02
30/3	Iter	29	29	17	23	88
	Matvec	898	784	448	616	2345
	Reduct.	8.5696E−10	8.5629E−10	9.9454E−10	9.7928E−10	3.1020E−11
30/6	Iter	11	11	11	11	11
	Matvec	318	253	256	252	269
	Reduct.	8.8821E−10	8.8821E−10	8.3377E−10	8.7220E−10	9.4485E−10
40/1	Iter	200	200	81	200	200
	Matvec	8200	7801	3119	7802	7802
	Reduct.	2.0077E−02	2.0077E−02	8.4790E−10	2.077E−02	2.2873E−02
40/3	Iter	11	11	12	10	15
	Matvec	412	373	423	371	557
	Reduct.	8.9534E−10	8.9565E−10	9.5688E−10	8.5029E−10	3.1323E−11
40/6	Iter	8	8	8	8	8
	Matvec	293	249	253	248	278
	Reduct.	9.4717E−10	9.6941E−10	7.7630E−10	9.7923E−10	5.9823E−10
50/1	Iter	200	200	78	200	200
	Matvec	10200	9801	3770	9802	9703
	Reduct.	2.0076E−02	2.0076E−02	8.5044E−10	2.0077E−02	5.4974E−02
50/3	Iter	7	7	13	7	9
	Matvec	356	330	577	314	426
	Reduct.	3.6822E−11	3.6447E−11	9.2228E−10	9.7071E−10	2.4308E−10
50/6	Iter	6	6	6	6	7
	Matvec	274	244	248	246	315
	Reduct.	9.2912E−10	9.3046E−10	8.9910E−10	9.4345E−10	1.2899E−13

As was already shown in [10], deflation techniques bring great improvements compared to their corresponding methods. For this matrix, IDGMRES performs the best and almost as well as the Full GMRES method, whereas the restarted version of GMRES stagnates.

Table 4
Matrix SHERMAN1, *tol* = 10E−9, *maxit* = 200, Precon = no.

m	Perf.	GMRES	FOM	MGMRES
20	Iter	200	200	127
	Matvec	4001	4001	2541
	Reduct.	1.6845E−06	1.3962E−05	7.1200E−10
30	Iter	144	164	71
	Matvec	4306	4902	2131
	Reduct.	9.9701E−10	9.3368E−10	3.3343E−10
40	Iter	85	95	40
	Matvec	3364	3791	1601
	Reduct.	9.9711E−10	9.8299E−10	5.2212E−10
50	Iter	58	61	34
	Matvec	2873	3047	1701
	Reduct.	9.9772E−10	9.8951E−10	4.1368E−10

Table 5
Matrix SHERMAN1, *tol* = 10E−9, *maxit* = 2000, Precon = no.

Perf.	QMR	Bi-CGSTAB
Iter	564	433
Matvec A	565	867
Matvec A^{T}	564	0
Reduct.	9.8652E−10	9.8153E−10

Second matrix. The second matrix is an unsymmetric 1000×1000 matrix arising in oil reservoir simulation. It is the matrix SHERMAN1 of the Harwell–Boeing collection. Its eigenvalues are all negative in the range [−5.045, −3.235E−02], where its five smallest eigenvalues are: 3.235E−02, −1.018E−02, −1.113E−01, −1.511E−01, −1.921E−01.

As can be seen in tables 4 and 5, QMR and Bi-CGSTAB perform much better than the restarted GMRES, FOM and MGMRES, although MGMRES performs much better than GMRES and FOM themselves. Nevertheless, the Full GMRES converges in 375 matrix–vector products, while the Full FOM converges in 379 matrix–vector products.

Considering table 6, we make the following remarks:

- For each of the deflation methods and each fixed m, the number of total matrix–vector products decreases as k increases.
- As in the previous case, AGMRES+ converges in the same number of runs as AGMRES but its number of matrix–vector products with A is strictly less than for AGMRES. This makes it competitive with IDGMRES. We have

$$nb_{\text{AGMRES+}} \simeq nb_{\text{IDGMRES}}.$$

Table 6
Matrix SHERMAN1, *tol* = 10E−9, *tolvp* = 10E−6, *maxit* = 200, Precon = no.

m/k	Perf.	AGMRES	AGMRES+	IDFOM	IDGMRES	IDMGMRES
20/1	Iter	142	142	149	141	69
	Matvec	2964	2682	2825	2665	1313
	Reduct.	9.9311E−10	9.9312E−10	9.3415E−10	9.9642E−10	8.5251E−10
20/3	Iter	122	122	128	122	62
	Matvec	2550	2068	2178	2073	1058
	Reduct.	9.9160E−10	9.9158E−10	8.8339E−10	9.9634E−10	7.3851E−10
20/6	Iter	101	101	108	100	58
	Matvec	2101	1406	1510	1405	819
	Reduct.	9.9509E−10	9.9509E−10	8.9183E−10	9.9579E−10	2.3213E−10
30/1	Iter	64	64	66	63	37
	Matvec	1957	1831	1902	1827	1075
	Reduct.	9.9158E−10	9.9158E−10	9.5594E−10	9.9958E−10	9.3685E−10
30/3	Iter	53	53	56	52	30
	Matvec	1615	1409	1494	1402	814
	Reduct.	9.9520E−10	9.9522E−10	9.8049E−10	9.9012E−10	9.7897E−10
30/6	Iter	42	42	44	41	31
	Matvec	1281	999	1051	987	751
	Reduct.	9.8583E−10	9.8583E−10	9.9522E−10	9.9988E−10	5.3735E−10
40/1	Iter	37	37	39	37	26
	Matvec	1485	1413	1504	1409	1016
	Reduct.	9.9607E−10	9.9554E−10	9.7896E−10	9.9460E−10	2.1861E−10
40/3	Iter	30	30	33	29	22
	Matvec	1213	1099	1206	1056	818
	Reduct.	9.9314E−10	9.9045E−10	9.0914E−10	9.9965E−10	4.1344E−10
40/6	Iter	26	26	27	25	23
	Matvec	1022	852	911	848	789
	Reduct.	9.8497E−10	9.8500E−10	9.9776E−10	9.8840E−10	4.4913E−10
50/1	Iter	26	26	27	26	18
	Matvec	1283	1230	1284	1232	884
	Reduct.	9.8014E−10	9.9709E−10	8.9342E−10	9.8834E−10	8.9597E−10
50/3	Iter	21	21	22	21	17
	Matvec	1052	973	1037	973	803
	Reduct.	9.8191E−10	9.9889E−10	9.7640E−10	9.8813E−10	6.5222E−10
50/6	Iter	18	18	19	18	17
	Matvec	890	777	818	771	755
	Reduct.	9.8258E−10	9.7531E−10	9.7884E−10	9.8049E−10	3.4586E−10

- Except for IDFOM $m = 40$, 50 and $k = 1$, we have

$$nb_{\text{FOM}} \geqslant nb_{\text{GMRES}} \geqslant nb_{\text{AGMRES}} \geqslant nb_{\text{IDFOM}} \geqslant nb_{\text{IDGMRES}} \geqslant nb_{\text{IDMGMRES}}.$$

- IDMGMRES performs better than the QMR and Bi-CGSTAB methods when $m = 20$, $k = 6$ and $m = 30$, 40, 50 and $k \geqslant 3$.
- IDGMRES and AGMRES+ perform better than QMR and Bi-CGSTAB for $m = 40$, 50 and $k = 6$.
- The smallest number of matrix–vector products is reached by IDMGMRES for $m = 30$, $k = 6$ in 751 matrix–vector products and $m = 50$, $k = 6$ in 755 matrix–vector products.

For this matrix, IDMGMRES is more efficient than IDGMRES. It is far from the convergence results of the Full version of GMRES or FOM but it performs better than the QMR and Bi-CGSTAB methods as well as the restarted GMRES and FOM. Concerning the other deflation techniques, they always perform better that their corresponding related methods FOM and GMRES. AGMRES+ and IDGMRES perform better too than QMR and Bi-CGSTAB when m and k are large enough.

5. Conclusion

We presented three new methods based on GMRES and FOM, that introduce spectral information of the previous run into the next one via the IRA method. For all the tests done, they perform better in number of total matrix–vector products than the QMR and Bi-CGSTAB methods, and for nontrivial cases they also perform better than GMRES and FOM, attempting to be closer to their Full versions. The convergence behavior of the method based on GMRES is always as good as or even better than the Augmented GMRES of Morgan [10] with a lower storage. Using the IRA method enables us to process a "real" deflation, that makes the resulting methods more robust. It also enables us to save dot products within the Arnoldi process. Since these dot products are done in sequence, they usually represent a great bottleneck in parallel environments. Saving some of them allows us to think that these new methods can exhibit acceleration from a CPU time point of view on parallel computers.

We dit not study the numerical instabilities that might arise from computing $H\text{obl}_m$, the oblique projected matrix of A, and its eigenvalues instead of solving the harmonic eigenvalue problem and will study them in a future work. We will study in detail too the method MGMRES and its deflated version IDMGMRES, especially from a theoretical aspect.

Acknowledgements

The authors would like to thank Richard Lehoucq for giving the Matlab code of the IRA method as well as the referees for their helpful remarks.

References

[1] J. Baglama, D. Calvetti, G.H. Golub and L. Reichel, Adaptively preconditioned GMRES algorithms, SIAM J. Sci. Comput. 20 (1999) 243–269.

[2] K. Burrage and J. Erhel, On the performance of various adaptive preconditioned GMRES strategies, Technical Report 1081, IRISA, France (1997) 303–318.

[3] A. Chapman and Y. Saad, Deflated and Augmented Krylov subspace techniques, Numer. Linear Algebra Appl. 4 (1997) 43–66.

[4] M. Eiermann and O. Ernst, On some recurrent theorems concerning Krylov subspace methods, Technical Report (1998).

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

[6] J.G. Francis, The QR transformation: A unitary analogue to the LR transformation, Parts I and II, Comput. J. 4 (1961) 265–272, 332–345.

[7] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2nd ed. (Johns Hopkins Univ. Press, Baltimore, MD, 1989).

[8] C. Le Calvez, Accélération de méthodes de Krylov pour la résolution de systèmes linéaires creux sur machines parallèles, Ph.D. thesis, Université des Sciences et Techniques de Lille (December 1998).

[9] R.B. Lehoucq and D.C. Sorensen, Deflation techniques for an implicitly restarted Arnoldi iteration, SIAM J. Matrix Anal. Appl. 17 (1996) 789–821.

[10] R.B. Morgan, A restarted GMRES method augmented with eigenvectors, SIAM J. Matrix Anal. Appl. 16(4) (1995) 1154–1171.

[11] R.B. Morgan, On restarting the Arnoldi method for large nonsymmetric eigenvalue problems, Math. Comp. 65 (1996) 1213–1230.

[12] R.B. Morgan, Implicitly restarted GMRES and Arnoldi methods for nonsymmetric systems of equations, SIAM J. Matrix Anal. Appl., to appear.

[13] Y. Saad, A flexible inner–outer preconditioned GMRES algorithm, SIAM J. Sci. Statist. Comput. 14(2) (1993) 461–469.

[14] Y. Saad, *Iterative Methods for Sparse Linear Systems* (PWS Publishing Company, 1996).

[15] Y. Saad, Analysis of augmented Krylov subspace methods, SIAM J. Matrix Anal. Appl. 18(2) (1997) 435–449.

[16] Y. Saad, M. Yeung, J. Erhel and F. Guyomarc'h, A deflated version of the Conjugate Gradient Algorithm, Technical Rapport, Supercomputing Institute, University of Minnesota (May 1998).

[17] D.C. Sorensen, Implicit application of polynomial filters in a k-step Arnoldi method, SIAM J. Matrix Anal. Appl. 13 (1992) 357–385.

[18] H.A. van der Vorst and C. Vuik, The superlinear convergence behaviour of GMRES, J. Comput. Appl. Math. 48 (1993) 327–341.