A key to choose subspace size in implicitly restarted Arnoldi method

S. Shahzadeh Fazeli, Nahid Emad, Zifan Liu

A KEY TO CHOOSE SUBSPACE SIZE IN IMPLICITLY RESTARTED ARNOLDI METHOD

S. A. SHAHZADEH FAZELI*, N. EMAD†, AND Z. LIU‡

Abstract. The implicitly restarted Arnoldi method (IRAM) computes some eigenpairs of large sparse non Hermitian matrices. However, the size of the subspace in this method is chosen empirically. A poor choice of this size could lead to the non-convergence of the method. In this paper we propose a technique to improve the choice of the size of subspace. This approach, called multiple implicitly restarted Arnoldi method with nested subspaces (MIRAMns) is based on the projection of the problem on several nested subspaces instead of a single one. Thus, it takes advantage of several different sized subspaces. MIRAMns updates the restarting vector of an IRAM by taking the eigen-information of interest obtained in all subspaces into account. With almost the same complexity as IRAM, according to our experiments, MIRAMns improves the convergence of IRAM.

Key words. Large eigenproblems, Krylov subspace size, Arnoldi method, implicit restarting.

AMS subject classifications. 65F15, 15A18

1. Introduction. In the fluid dynamics, economic modelling and some other domains there are a number of cases where a few eigenvalues with the largest real part or with the largest magnitude of a nonsymmetric matrix are required. Several methods have been proposed to solve this kind of eigenproblems. Arnoldi in 1951 [1] proposed a method which is a variant of the Krylov projection methods. The Arnoldi method is an efficient technique which permits to compute an approximation of desired eigenpairs of an n-size matrix A by those of an $m \times m$ matrix representing A in an m-size Krylov subspace (with $m \leq n$). Meanwhile, when the wanted eigenvalues are clustered the Krylov subspace ought to have overly large size. The Arnoldi method is best at finding a solution to an eigenproblem with well-separated eigenvalues. A drawback of this method is the expense of too much memory space when m is large. This problem is solved by restarting the method as proposed in 1980 by Saad [19].

Saad's approach, called explicitly restarted Arnoldi method (ERAM), restarts the Arnoldi projection with a better subspace. Indeed, this approach offers to choose a small Krylov subspace ($m \ll n$). Then, if the accuracy of the desired Ritz elements computed by Arnoldi method is not satisfactory, ERAM restarts the process using a new Krylov subspace. This new subspace differs by the last one by its initial vector which is formed by an explicit combination of the computed Ritz elements. Although this algorithm is attractive for reason of simplicity, the formation of a restarting vector for the next restart cycle using the approximated eigenvectors of the current cycle might not be good. The restarting is difficult because one new starting vector must be defined as an explicit linear combination of wanted Ritz vectors. If this combination is not carefully chosen, it can lead to a very bad selection for the new starting vector. Moreover, when the starting vectors are complex the cost will increase. Saad proposed some special coefficients for the combination of Ritz vectors such as the weighted linear combination [19]. As himself mentioned it, this method may not well work in practice for many eigenproblems [20]. Moreover, the problem of the choice

*Pure & Applied Mathematics Department, Faculty of Science, Yazd University, Iran (fazeli@yazduni.ac.ir).

†Laboratoire PRiSM, Université de Versailles St-Quentin, 45, avenue. des États-Unis, 78035 Versailles, France (nahid.emad@prism.uvsq.fr).

‡Laboratoire PRiSM, Université de Versailles St-Quentin, 45, avenue. des États-Unis, 78035 Versailles, France (zifan.liu@prism.uvsq.fr).

of the size of the subspace remains. An approach based on the Arnoldi projection onto several Krylov subspaces is proposed by Emad et al [8]. The latter are formed with different initial vectors and have different sizes. This technique is called multiple explicitly restarted Arnoldi method (MERAM). It updates the restarting vector of each ERAM process by taking into account the eigen-information of interest obtained by all ERAM processes. MERAM improves often the convergence of ERAM but the restarting issues intrinsic to ERAM remain [8].

In order to improve the Arnoldi method, Sorensen has suggested an efficient technique which makes use of the QR algorithm to restart the Arnoldi projection [21]. His approach permits to restart the Arnoldi process with an efficient and numerically stable formulation. This approach which is called implicitly restarting Arnoldi method (IRAM) was analyzed, implemented and validated, among others, in [21, 10, 11, 23, 12, 13]. As in ERAM, implicity restarted Arnoldi method makes use of Arnoldi projection to approximate the desired eigenpairs of a large matrix A. If the accuracy of these Ritz elements is not satisfactory, IRAM applies a QR shifted algorithm on the $m \times m$ matrix which represents A in the projection subspace. As these are the non desired eigenvalues which are chosen for shifts, the upper-left block of the matrix issued from QR algorithm concentrates the information corresponding to the desired eigenvalues. IRAM completes an $m-$size Arnoldi projection starting with the submatrix representing this block whose size is the number of wanted eigenvalues. This is equivalent to restart the Arnoldi process with a new initial vector computed implicitly. Morgan showed that IRAM is much better than the other restarting Arnoldi methods such as explicitly restarted ones [14]. However, the problem of choosing the size of subspace remains.

In restarted Arnoldi methods, in order to improve the quality of the subspace during the restart cycles, only the initial vector is taken into account. We propose to make this improvement by taking into account both the initial vector and the size of the subspace. For this, we propose to apply the same idea as the one used in MERAM. This approach called multiple implicitly restarted Arnoldi method with nested subspaces (MIRAMns), is based upon the projection of the problem on several Krylov subspaces instead of a single one. These subspaces differ by their size while the subspaces in multiple restarted methods such as MERAM can differ by their size and their initial vectors [8]. MIRAMns makes use of Arnoldi method to compute the Ritz elements of a large matrix A in a set of ℓ nested Krylov subspaces $\mathbb{K}_{m_i,v}$ (for $i = 1, \ldots, \ell$) with $\mathbb{K}_{m_i,v} \subset \mathbb{K}_{m_{i+1},v}$. If the accuracy of the desired Ritz elements calculated in these subspaces is not satisfactory, MIRAMns selects the "best" of these subspaces. This subspace is one that contains the "best" current Ritz elements. Then a QR shifted algorithm will be applied to the $m_{best} \times m_{best}$ matrix which represents A in this m_{best}-size projection subspace. As these are the non desired eigenvalues which are chosen for shifts, the leading submatrix issued from QR algorithm concentrates the information corresponding to the desired eigenvalues. MIRAMns completes Arnoldi projections on ℓ nested Krylov subspaces starting with this submatrix whose size is the number of wanted eigenvalues.

One of the well known problems of the restarted iterative methods is the sensitivity of the convergence with respect to small perturbations of the subspace size. Indeed, they could not converge with a subspace and converge with the same reduced/extended subspace with nearby sizes. MIRAMns overcomes to this problem by making choice of the "best" size among these subspace sizes. Another advantage of this technique is the better property of convergence with almost the same time

complexity relative to IRAM. Our experiments showed a very good acceleration of convergence with respect to the implicitly restarted Arnoldi method.

This paper has the following organization. An overview of the related work is presented in the next section. The implicitly restating Arnoldi method is discussed in section 3. The multiple implicitly restarted Arnoldi method with nested subspaces and an algorithm of it are presented in section 4. Section 5 is devoted to numerical experiments. Finally we present a conclusion in the last section.

Definition and notation. Let A be a complex non-Hermitian matrix of dimension $n \times n$, v be an $n-$size initial guess, $\mathbb{K}_{m,v_1}$ be the Krylov subspace spanned by $(v_1, Av_1, \ldots, A^{m-1}v_1)$ with $v_1 = v/\|v\|_2$ and $(v_1, \cdots, v_m)$ be an orthogonal basis of this subspace generated by the Gram-Schmidt orthogonalization process. A relation of the form $AV_m = V_m H_m + f_m e_m^T$ is called an m-step Arnoldi factorization, where $H_m \in \mathbb{C}^{m\times m}$ is an upper Hessenberg matrix with non-negative subdiagonal elements, $V_m \in \mathbb{C}^{n\times m}$ is a matrix with orthonormal columns $v_1, \cdots, v_m$ and $V_m^H f_m = 0$ where f_m is the residual vector of lenght n. This factorization can be used to reduce the eigenproblem with the large order matrix A to a problem with a smaller order matrix H_m. Let $y_i^{(m)}$ be an eigenvector of H_m associated with the eigenvalue $\lambda_i^{(m)}$ and $u_i^{(m)} = V_m y_i^{(m)}$. $(u_i^{(m)}, \lambda_i^{(m)})$ is an approximate eigenpair for A called also Ritz eigenpair and $\rho_{i,m} = \|(A - \lambda_i^{(m)} I)u_i^{(m)}\|_2 = \|(AV_m - V_m H_m)y_i^{(m)}\|_2 = |\beta_m e_m^T y_i^{(m)}|$ is the residual norm associated to this Ritz eigenpair. The number $|\beta_m e_m^T y_i^{(m)}|$ is called also Ritz estimate where $\beta_m = \|f_m\|_2$. In the rest of this paper, we denote by k the number of wanted eigenvalues, by $\{\mu_i^{(m)}\}_{i=1}^p$ the set of unwanted eigenvalues of H_m $(p = m - k)$, by $\|A\|_F$ the Frobenius norm of A, by $H_m(1:k, 1:k)$ the leading $k \times k$ submatrix of H_m and by nrc the number of restart cycles.

2. Related work. The paper [25] explains the relationship between the subspace size in IRAM and its convergence rate so the relevance of the subspace size has been discussed in the open literature. Dynamic selection of restart parameters in Arnoldi methods have been considered previously. Duff and Scott in [7] developed a subspace algorithm combined with Chebychev acceleration. They select dynamically the size of the subspace and the degree of the Chebychev polynomial at each restart cycle. Stathopoulos, Saad and Wu proposed in [24] a technique, called thick restarting, than restarts the Arnoldi algorithm with more eigenvectors that is actually required. A dynamic thick restarting scheme which adjusts the number of retained Ritz vectors at each cycle in IRAM is proposed and the question of which and how many eigenvectors to retain is addressed for symmetrical eigenproblems.

Some authors suggested more similar approaches to the one proposed in this paper. Baker et al. proposed in [3] a simple strategy and provide some heuristic explanation for its effectiveness. The authors define a range of subspace sizes whose minimum and maximum values are respectivily m_{min} and m_{max} and they choose, according to some criterion the subspace size m_i of the ith restart cycle in this discrete interval. Their strategy checks the convergence rate $\| r_{i+1} \|_2 \ / \ \| r_i \|_2$ at the end of each restart cycle, where r_i is the residual vector of ith cycle. The subspace size is initialized by m_{max}. When stagnation is detected, they decrease the restart parameter by a small number d at each cycle until reaching m_{min}. At that point, they increase m_i up to the maximum value m_{max}. For the solution of non-symmetric linear systems by deflated GMRES, Moriya and Nodera proposed in [15] a similar dynamic switching approach for the Krylov subspace dimension. Their strategy consists to

combine the deflated GMRES algorithm and the determination of a restart parameter m dynamically. Indeed, in order to decrease the computation cost, the authors propose to begin with a small restart parameter m_s. If stagnation in encountered then the restart parameter is switched to a larger value m_l. When the restart value is m_l and the stagnation disappears then the restart parameter is switched again to m_s. They use as a criterion for stagnation the angle between the residual vector and search vectors which could be easily computed during a run of GMRES. Dookhitram et al. proposed in [6] a comparable approach to accelerate the convergence of IRAM which is based on a relationship between the residual of the current restart cycle of IRAM and the residual in the previous cycle. Despite the similarity, their technique differs from that proposed by Moriya and Nodera for linear systems since unlike the latter, they do not initialize any angle to avoid a problem dependent strategy and also their switching strategy is based on a different relationship between the residual of the current restart cycle and the residual of the previous cycle.

In the approach proposed in this paper, the dynamic determination of subspace size parameter is monitored and used inside the current restart cycle while in the methods cited above, this determination is done in current restart cycle for being used in the next restart cycle. In other words, the dynamicity of our approach is intra-cycle, while in the approaches mentioned above, the dynamicity is inter-cycle. Some results of the comparison with the methods proposed in [24] and [6] are shown in the section 5.

3. The Sorensen approach. This variant of the Arnoldi method based on restarting technique is called implicitly restarted Arnoldi method (IRAM). That is a technique that combines the implicitly shifted QR mechanism with an Arnoldi factorization and can be viewed as a truncated form of the implicitly shifted QR-iteration [23]. This method involves an implicit application of a polynomial in A to the starting vector. IRAM computes a few eigenvalues ($k \leq m$) such as those of the largest real part or the largest magnitude. An m-step Arnoldi factorization

$$AV_m = V_m H_m + f_m e_m^T, \tag{3.1}$$

is compressed to a factorization of length k with the eigen-information of interest. This is achieved using QR steps to apply $p = m-k$ shifts implicitly. The results after the shift process and equating the first k columns on both sides are

$$AV_m^+ = V_m^+ H_m^+ + f_m e_m^T Q, \tag{3.2}$$

where $V_m^+ = V_m Q, H_m^+ = Q^T H_m Q$, and $Q = Q_1 Q_2 \cdots Q_p$ with Q_j the orthogonal matrix in QR process associated with the shift $\mu_j^{(m)}$ and

$$AV_k^+ = V_k^+ H_k^+ + f_k^+ e_k^T, \tag{3.3}$$

with $f_k^+ = V_m^+ e_{k+1} \widehat{\beta}_k + f_m \sigma_k$ where $\widehat{\beta}_k = H_m^+(k+1,k)$ and $\sigma_k = Q(m,k)$. Using this as a starting point one can apply p additional steps of the Arnoldi process to obtain an m-step Arnoldi factorization. Each shift cycle involves the implicit application of a polynomial in A of degree p to the starting vector v: $\psi(A)v$ with $\psi(\lambda) = \Pi_{j=1}^{p}(\lambda - \mu_j^{(m)})$. The roots of this polynomial are the shifts used in the QR algorithm [23]. The resulting algorithm takes the form of the algorithm 1.

The stopping criterion in above algorithm can be computed by the expression called Ritz estimate: ***(a)-*** $|\beta_m e_m^T y_i^{(m)}|$, or by its mathematically equivalent explicit

Algorithm 1 Implicitly restarted Arnoldi method

Input: (A, V_m, H_m, f_m) with $AV_m = V_m H_m + f_m e_m^T$ an m-step Arnoldi factorization
For $it = 1, \ldots,$ until convergence

1. • Compute $\sigma(H_m)$ the eigenvalue of H_m and their associated eigenvectors,
   • Compute residual norm, if convergence stop,
2. Select set of $p = m - k$ shifts $(\mu_1^{(m)}, \cdots, \mu_p^{(m)})$, based upon $\sigma(H_m)$ or other information and set $q^T \leftarrow e_m^T$,
3. For $j = 1, 2, \ldots, p$
   • Factor $[Q_j, R_j] = qr(H_m - \mu_j^{(m)} I)$;
   • $H_m \leftarrow Q_j^H H_m Q_j; V_m \leftarrow V_m Q_j$;
   • $q^H \leftarrow q^H Q_j$;
4. Set $f_k \leftarrow v_{k+1}\widehat{\beta}_k + f_m \sigma_k$, $V_k \leftarrow V_m(1:n, 1:k)$, $H_k \leftarrow H_m(1:k, 1:k)$
5. Beginning with the k-step Arnoldi factorization $AV_k = V_k H_k + f_k e_k^T$, Apply p additional steps of the Arnoldi process to obtain a new m-step Arnoldi factorization $AV_m = V_m H_m + f_m e_m^T$

formula of the residual norm: ***(b)-*** $\|(Au_i^{(m)} - \lambda_i^{(m)} u_i^{(m)})\|_2$ (for $i = 1, \cdots, k$). Criterion (a) has a computational cost much lower than that of (b). However, criterion (b) may better represent the residual corresponding to the Ritz elements and is more reliable when rounding errors are present. This is because the expression of (b) contains computed Ritz elements and thus takes into account the rounding errors in their calculation. It should be noted that, as explained in [22, 16], when A is Hermitian, the relation ((a)=(b)) may be used to provide computable rigorous bounds on the accuracy of the eigenvalues of H_m as approximations to eigenvalues of A. When A is non-Hermitian the possibility of non-normality precludes such bounds and one can only say that the residual norm $\|(Au_i^{(m)} - \lambda_i^{(m)} u_i^{(m)})\|_2$ is small if $|\beta_m e_m^T y_i^{(m)}|$ is small.

Note that if $v = \sum_{j=1}^{n} \gamma_j u_j$, the implicit restarting Arnoldi method with exact shifts provides a specific selection of expansion coefficients γ_j for a new starting vector as a linear combination of the current Ritz vectors for desired eigenvectors. Implicit restarting provides a way to extract eigen-information of interest from large Krylov subspaces while avoiding the storage and numerical difficulties. This is done by continually compressing eigen-information of interest into an k-dimensional subspace of fixed size. This means that IRAM continues an m-step Arnoldi factorization, having kept all Ritz vectors of interest.

Time and space complexities of IRAM. We assume that $m \ll n$. Therefore, in the time complexity expression of IRAM we can disregard terms not containing n. Let nrc be the number of restart cycles excluding the *input* step in the above algorithm. The cost of IRAM in terms of matrix-vector products for nrc restart cycles with `(a)` criterion is $[m + (it - 1) \times (m - k)]$. Indeed, in the first restart cycle the number of matrix-vector products is m and for each of the other restart cycles, the number of matrix-vector products is $p = (m - k)$. The cost of IRAM will be increased by $it \times k$, if `(b)` criterion is used. Noted that the cost of orthogonalization in a restart cycle is $O() = m^2 n + O(n)$?. When A is sparse and p is large, this cost of orthogonalization may dominant the computation.

The space complexity of IRAM is $O(n^2 + m \times n)$.

4. Multiple IRAM with nested subspaces (MIRAMns). The purpose of restarting m-step Arnoldi factorization is to improve the quality of the initial Krylov

subspace $\mathbb{K}_{m,v}$. This objective can be achieved by improvement of the vector v and/or the subspace size m. Indeed, the information obtained through the m-step Arnoldi factorization process is completely determined by the choice of the starting vector v and the subspace size m. The current Arnoldi (explicit/implicit) restarting techniques propose an amelioration of the initial vector v. Regarding the size of the subspace, it is known that the eigen-information of interest may not appear when m is too small [22]. Furthermore, if m is too large, the computation cost of orthogonalization process becomes excessive. The size of the subspace has to be chosen as a compromise between these factors and is chosen empirically according to the number of desired eigen-elements, the size of the original problem, etc. Here, we present a way to increase the quality of the Krylov subspace by improving the size of the subspace. Indeed, to remedy the essential question of the choice of the size of the subspace, this paper suggests the proliferation of these subspace sizes and to select the best one. The size of the subspace is chosen dynamically in every restarting step.

This approach consists to make use of IRAM with a set of Krylov subspaces which differ only by their size which means a set of nested subspaces. Let v be an initial vector and $M = (m_1, \cdots, m_\ell)$ be a set of ℓ subspace-sizes with $m_1 < \cdots < m_\ell$. We built ℓ Arnoldi projections on the subspaces $\mathbb{K}_{m_i,v}$ (for $i = 1, \ldots, \ell$) where $\mathbb{K}_{m_1,v} \subset \mathbb{K}_{m_2,v} \subset \ldots \subset \mathbb{K}_{m_\ell,v}$. We select then the subspace size m_{best} corresponding to the Arnoldi factorization which offers the Ritz estimates for k desired eigenpairs. The steps 2 to 4 of IRAM algorithm (i.e. algorithm 1) are applied then onto this Arnoldi factorization: $AV_{m_{best}} = V_{m_{best}} H_{m_{best}} + f_{m_{best}} e^T_{m_{best}}$. That means only this factorization among the ℓ ones will be compressed to a factorization of length k with the eigen-information of interest. This is achieved using QR steps to apply $p_{best} = m_{best} - k$ shifts implicitly. The results after the shift process and equating the first k columns on both sides are the same as in equation (3.3) with $m = m_{best}$. Beginning with this resulting k-step Arnoldi factorization, we apply then $p_i = m_i - k$ additional steps of Arnoldi factorizations to obtain ℓ new projections onto the updated subspaces (for $i = 1, \cdots, \ell$). This allows again the projection onto ℓ nested subspaces with initial guess determined by the compressed k-step Arnoldi factorization issued from the QR shifted applied to $m_{best}-$step Arnoldi factorization.

We notice that this technique updates the restarting vector v by taking the eigen-information obtained by several subspaces into account. Moreover, for a given restart cycle MIRAMns has almost the same time complexity as IRAM with the largest subspace size. Besides, in MIRAMns as in IRAM, the appearance of spurious eigenvalues may be avoided through complete reorthogonalization of the Arnoldi vectors using the DGKS correction [23, 5]. An algorithm of this method to compute k ($k \leq m_1$) desired Ritz elements of A is presented by the algorithm 2.

In order to select the best results in the step (2) of the algorithm 2 we suppose that $(V_{m_i}, H_{m_i}, f_{m_i})$ is "better" than $(V_{m_j}, H_{m_j}, f_{m_j})$ if $r_k^{m_i} < r_k^{m_j}$ where $r_k^{m_i} = max(\rho_{1,m_i}, \ldots, \rho_{k,m_i})$ is defined by Ritz estimates when `(a)` stopping criterion is used. The $r_k^{m_i}$ value is defined by the residual norm of Rayleigh quotient corresponding to $(V_{m_i}, H_{m_i}, f_{m_i})$ when `(b)` stopping criterion is used.

4.1. Time and space complexities of MIRAMns. As for IRAM, we can disregard the terms not containing n. Recall that m_ℓ is the maximum of $m_1, \ldots, m_\ell$ subspace sizes. The cost of MIRAMns in term of matrix-vector products with `(a)` criterion is $[m_\ell + (nrc - 1) \times (m_\ell - k)]$ where nrc is the number of restart cycles. This cost will be increased by $(\ell \times k \times nrc)$ if `(b)` criterion is used. Still the cost of orthogonalization in Arnoldi process is about $m_l^2 n + O(n)$. As a result, this cost

Algorithm 2 Multiple IRAM with nested subspaces

Input: $(A, V_{m_i}, H_{m_i}, f_{m_i})$ with $AV_{m_i} = V_{m_i}H_{m_i} + f_{m_i}e_{m_i}^T$ the m_i-steps Arnoldi factorization $(i = 1, 2, \ldots, \ell)$

For $it = 1, 2, \ldots$ until convergence

1. • Compute $\sigma(H_{m_i})$ and their associated eigenvectors (for $i = 1, \ldots, \ell$)
   • Compute residual norms. If convergence in one of subspaces then stop.
2. Select the best results in these subspaces and the associated best subspace size m_{best}. Set $m = m_{best}, H_m = H_{m_{best}}$ and $V_m = V_{m_{best}}, f_m = f_{m_{best}}$.
3. Select set of $p = m - k$ shifts $(\mu_1^{(m)}, \ldots, \mu_p^{(m)})$, based upon $\sigma(H_m)$ or perhaps other information and set $q^T \leftarrow e_m^T$.
4. For $j = 1, \ldots, p$
   • Factor $[Q_j, R_j] = qr(H_m - \mu_j^{(m)} I)$;
   • $H_m \leftarrow Q_j^H H_m Q_j; V_m \leftarrow V_m Q_j$,
   • $q \leftarrow q^H Q_j$
5. Set $f_k \leftarrow v_{k+1}\widehat{\beta}_k + f_m q(k)$, $V_k \leftarrow V_m(1:n, 1:k)$, $H_k \leftarrow H_m(1:k, 1:k)$
6. Beginning with the k-step Arnoldi factorization $AV_k = V_k H_k + f_k e_k^T$, apply $p_i = m_i - k$ additional steps of the Arnoldi process to obtain ℓ new m_i-step Arnoldi factorization $AV_{m_i} = V_{m_i}H_{m_i} + f_{m_i}e_{m_i}^T$ (for $i = 1, \ldots, \ell$).

of orthogonalization may still dominant the computation when A is sparse and p is large.

Let CI and CM be the time complexities of ***one restart cycle*** of IRAM(m_ℓ) and MIRAMns($m_1, \cdots, m_\ell$) respectively. If we ignore terms not including n and the cost of stopping criterion, these complexities can be given by $CI = \alpha + O\left(2 \times n \times m_\ell^2\right)$ and $CM = \alpha + O\left(2 \times n \times \left[k \times (m_1 + \ldots + m_\ell) + m_{best}^2 - k \times m_{best}\right]\right)$ where α is the complexity of the common part of both algorithms. To compare these complexities, we can notice that in the worst case for MIRAMns where $m_{best} = m_\ell$, $CM - CI = O\left(2 \times n \times k \times (m_1 + \ldots + m_{\ell-1})\right)$ which is positive and in the best case where $m_{best} = m_1$, $C_M - C_I = O\left(2 \times n \times \left[k \times (m_2 + \ldots + m_\ell) + m_1^2 - m_\ell^2\right]\right)$ which could be positive or negative. That means, according to the k and m_i values, one restart cycle of MIRAMns could be less expensive than the one of IRAM. Consequently, MIRAMns($m_1, \cdots, m_\ell$) could be less expensive than IRAM(m_ℓ).

The space complexity of MIRAMns is $O(n^2 + n \times m_\ell)$.

4.2. Parallelism analysis of MIRAMns . The same parallel programming model as that one used in the implicitly restarted Arnoldi method can be used for the multiple implicitly restarted Arnoldi method with nested subspaces [13]. Then, the distributed parallel programming model defined and implemented in the library P_ARPACK can be applied to MIRAMns also.

Nevertheless, it should be noted that MIRAM (with nested or non nested subspaces) has a great potential for large coarse grain parallelism among its Arnoldi factorizations. Indeed, the computation in different subspaces of MIRAM could be done in parallel. In addition to this coarse grain parallelism among IRAM processes, one can also overlap communication step with computations. Indeed, the communication of the eigen-information of interest of each IRAM process to other IRAM processes can be done asynchronously. The analyse of intra and inter Arnoldi factorizations parallelism in MIRAM and its implementations on large scale distributed systems can be the subject of a future work.

4.3. Suggestion to set subspace sizes. As already stated, a too small value for the size of the subspace would lead to non-appearance of eigen-information of interest in the subspace. Moreover, the choice of a too large value for this parameter could lead to an excessive the computation cost for orthogonalization process of m-step Arnoldi factorization. Therefore, a compromise between a too large value and a too small value for m is required. Setting this value is very much problem dependent. Indeed, it is selected empirically depending on the size of the problem, the number of desired eigenvalues/eigenvectors, etc. As a rule of thumb $m \geqslant 2 \times k$ is suggested by [22, 12] as a reasonable choice. It is good to notice that $m = 2k$ is the default in Matlab. It is also suggested, where possible, to avoid setting k in a way that will split clusters of eigenvalues. As a result, this is true for the choice of m as well.

Let $[m_1, \cdots, m_\ell]$ be the discrete interval of subspace sizes in MIRAMns. The lower bound m_1 should be chosen small enough to provide good orthogonality of basis vectors of Krylov subspace. Applying the rule of thumb described above is a acceptable choice (i.e.: $m_1 \geqslant 2 \times k$). The upper bound m_ℓ should be large enough so that the Krylov subspace can contain all the required eigenvalues. We choose the values within this range distributed fairly uniformly.

The technique presented here is a way to make better the size of the subspace in Arnodi restarted methods. This approach allows improving the quality of the Krylov subspace $\mathbb{K}_{m,v}$ by the refinement of the parameter m at the same time as that of the parameter v by these restarted methods. Nevertheless, the approach belongs empirical and could be improved by analyzing the results of intensive experimentation.

5. Numerical experiments. We have implemented and tested Algorithm 1 (IRAM) and Algorithm 2 (MIRAMns) using MATLAB to compute $k = 2$ eigenvalues of greatest magnitude, except for certain case where convergence is too fast for both IRAM and MIRAMns. The stopping criterion used in IRAM is $r_k^m = \max(\rho_1^m, \ldots, \rho_k^m) < tol$ with $\rho_i^m = |\beta_m e_m^T y_i^{(m)}|/\|A\|_F$ where tol specifies the accuracy requested. The criterion used to select the best subspace size in MIRAMns is $r_k^{m_{best}} = \min(r_k^{m_1}, \ldots, r_k^{m_\ell})$ where $m_1 < \cdots < m_\ell$ are subspace sizes (with $m_1 \geq 2\times k$). Every other stopping criterion can replace the requirement to find k eigenvalues. The tolerance value tol is 10^{-8} for the Figures 5.1 to 5.4(a), 5.7(a), 5.8; 10^{-10} for the Figure 5.6(a); 10^{-12} for the Figures 5.10 and 10^{-14} for the Figures 5.4(b), 5.5, 5.7(b) and 5.9. In all experiments presented here, initial vector is $x = z_n/\|z_n\|_2$ except for the Figure 5.8(a) that initial vector is $x = s_n/\|s_n\|_2$ and the Figures 5.9(a) and 5.10 that initial vector is $x = t_n/\|t_n\|_2$ where $z_n = (1, 1, \cdots, 1)^T$, $s_n = (1, 1, 0.1, \cdots, 0.1)^T$ and $t_n = (1, 1, 0, \cdots, 0)^T$. The initial vectors of IRAM and MIRAMns are the same. The efficiency of these algorithms can thus be measured in terms of the number of restarts cycles (nrc) or the number of matrix-vector products MVP. Our matrices are presented in the table 5.1.

We have used four matrices $af23560.mtx$, $bfw782a.mtx$, $west0989.mtx$ and $sherman3.mtx$ from Matrix Market [2]. $A9\_1000$ is a tridiagonal matrix of order 1000 defined by $a_{i,i} = 3$, $a_{i,i+1} = a_{i,i-1} = 1$. All other entries are zero. The tridiagonal matrix AM_1000 is of dimension $n = 1000$. The diagonal entries are $a_{i,i} = i$, the codiagonal entries are $a_{i,i+1} = -0.1$ and $a_{i,i-1} = 0.1$. All other entries are zero. This example has been taken from [14]. Matrices *roadNet-PA.mtx, com-Youtube.mtx* and $WikiTalk.mtx$ are transition matrices constructed from three social graphs by using Markov chain. These graphs could be found in [26].

Matrix	Size of matrix	nonzero elements
$af23560.mtx$	23560	484256
$bfw782a.mtx$	782	7514
$A9\_1000.mtx$	1000	2998
$west0989.mtx$	989	3537
$AM\_1000.mtx$	1000	2998
$sherman3.mtx$	5005	20033
$roadNet\text{-}PA.mtx$	1088092	3083796
$com\text{-}Youtube.mtx$	1134890	2988374
$WikiTalk.mtx$	2394385	5046614

Table 5.1: General information about the test matrices

5.1. MIRAMns versus IRAM. In all the following figures MIRAMns($m_1, \cdots, m_\ell$) denotes an MIRAMns with subspace sizes $(m_1, \cdots, m_\ell)$, IRAM(m) denotes an IRAM with subspace size m and MVP denotes the number of matrix-vector products. It is important to note that the **main objective** of our experiments is to compare the performance of MIRAMns($m_1, \cdots, m_\ell$) and IRAM(m_ℓ). However, some of experiments have the aim of highlighting the influence of certain parameters on the convergence of these methods. For some typical cases, we present the best subspaces choosen by MIRAMns throughout the restarting cycles, so as to clarify the necessity of using the whole set of subspaces.

The table 5.2 presents the results obtained with IRAM algorithm and the table 5.3 presents the results obtained with MIRAMns and a comparison between IRAM and MIRAMns in term of number of matrix-vector products (MVP), execution time in seconds ($Ex.Time$) and number of restarts (nrc). $Ex.Time$ represents the total execution time: from the beginning of the algorithm (after inputs) upto obtaining the wanted eigenpairs. We can see that in almost half of the experiments presented in table 5.2, IRAM does not converge. The results presented in the table 5.3, show that MIRAMns overcomes these problems of non-convergence.

We show graphically in Figures 5.1 to 5.10 the norm of residual as a function of restart cycle number to reach convergence using MIRAMns and IRAM. We see that there is no convergence for IRAM in figures 5.1 to 5.5, 5.9(a) and 5.10 while MIRAMns reaches convergence. Moreover, the convergence of MIRAMns in figures 5.7, 5.8, 5.9(b) and 5.12(a) is better than IRAM. Specifically, Figure 5.1(a) shows that the curves of convergence of IRAM and MIRAMns undergo oscillations around the residual norm 10^{-6}. However, the peak to peak amplitude of the oscillations corresponding to IRAM is very large while the one corresponding to MIRAMns are quite small. This could become related to a kind of smoothing of the curve of convergence of IRAM by MIRAMns.

Figure 5.1(b) shows the influence of the size of the subspace on the convergence of IRAM. Indeed, we note that an augementation of this size relative to that of Figure 5.1(a) smooths the curve of the convergence of IRAM. We can see also that with the chosen tolerance ($tol = 10^{-8}$), MIRAMns converges but IRAM does not converge. However, with a greater tolerance value such as $tol = 10^{-7}$, IRAM reaches convergence when MIRAMns does not reach it. But it must be remembered that in this case, the parameters of IRAM and MIRAMns no longer meet the criteria for comparison. Indeed, the subspace size of IRAM (22) is larger than $m_l = 10$.

Figure 5.1(a) shows the effect of increasing the size of the subspace on the convergence of IRAM and MIRAMns and highlights the speed of convergence of MIRAMns with respect to that of IRAM. The acceleration of convergence of IRAM by MIRAMns is also shown in Figures 5.4 to 5.10. However, we can see that in Figure 5.4(a), before $tol = 10^{-5}$, IRAM could reach convergence faster than MIRAMns. But this is just an oscillation peak stronger than the others and the residual norms do not decrease continually while those of MIRAMns decreases steadily during the restart cycles. Figures 5.4 and 5.10(b) show further how the convergence curves of IRAM are "smoothed" by MIRAMns. Furthermore, by comparing subfigures in Figure 5.2 and Figure 5.5, we notice that an increase in number of subspaces could improve the speed of convergence for MIRAMns itself as well.

Figures 5.11 and 5.12(b) show the subspaces selected by MIRAMns throughout restart cycles. We see that m_{best} is choosen randomly from interval $[m_1, m_l]$ and for some experiments m_1 is selected as best subspace size more often that m_l (Figure 5.11(b)). This phenomenon of local optimality of small subspaces was also observed by Embree on GMRES [9].

Figures 5.13 and 5.14 show the execution time of MIRAMns versus IRAM throughout restart cycles for AM_1000 and $west0989$ matrices. We notice that the execution times of each restarting cycle is almost the same for both methods. Nevertheless, the total execution time of MIRAM is much smaller than IRAM.

Tables 5.2 and 5.3 and Figures 5.1 to 5.10 indicate that our algorithm improves performances of IRAM. We notice also that this improvement is much more significant when the matrices have clustered eigenvalues such as $af23560$ and $bfw782$ matrices used in our experiments.

5.2. MIRAMns versus other approches. To check the influence of the strategy proposed by Stathopoulos, Saad and Wu in [24], we compared the tick restarted versions of IRAM and MIRAMns. The Figures 5.6 shows this comparison with the number of wanted eigenpairs $k = 15$ and the thick parameters 2 and 5. That means in each restart cycle, a buffer of 2 and 5 extra vectors are kept. We can notice that IRAM could sometimes perform as good as MIRAMns.

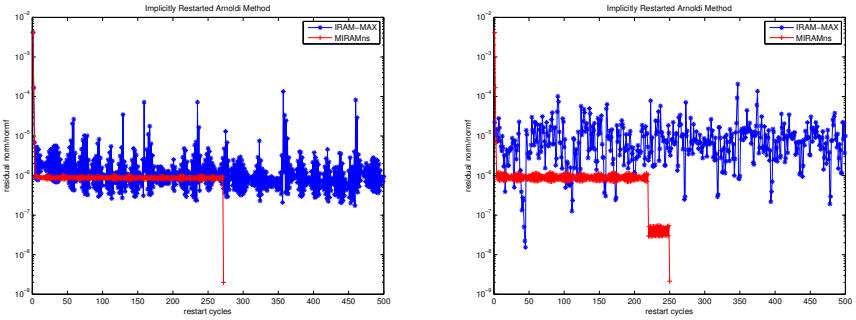
Figure 5.15 shows the results of comparison with DIRA, the method proposed by Dookhitram et al. in [6]. The matrix, initial vector and tolerance value are $bfw782a$, $x = t_n/\|t_n\|_2$ and 10^{-10} respectively. Figure 5.15(a) shows that MIRAMns reaches convergence in 8 restarting cycles while DIRA converges in 12 cycles. The execution times of MIRAMns and DIRA are 0.045 and 0.08 seconds respectively. Figure 5.15(b) shows the convergence of MIRAMns and DIRA in 19 and 67 restarting cycles. Their execution times are 0.19 and 0.25 seconds respectively. According to our experiments MIRAMns gives good results with respect of this approach.

Matrix	m	x	tol	nrc	MVP	Ex.Time	Res.Norm	Fig.
$af23560.mtx$	10	z_n	10^{-8}	* 500	*4002	10.27	no conv.	5.1 (a)
$af23560.mtx$	22	s_n	10^{-8}	* 500	*10002	31.44	no conv.	5.1 (b) 5.2
$af23560.mtx$	25	z_n	10^{-8}	6	140	0.60	2.60e-09	5.12 (a)
$af23560.mtx$	40	z_n	10^{-10}	114	2865	24.02	8.24e-11	5.6 (a)
$bfw782a.mtx$	10	z_n	10^{-8}	* 500	*4002	1.00	no conv.	5.4 (a)
$bfw782a.mtx$	20	z_n	10^{-14}	* 500	*9002	2.18	no conv.	5.4 (b) 5.5
$bfw782a.mtx$	30	z_n	10^{-14}	12	195	0.09	1.98e-15	5.6 (b)
$A9\_1000.mtx$	20	z_n	10^{-8}	309	5564	1.40	9.57e-09	5.7 (a)
$west0989.mtx$	10	z_n	10^{-8}	53	426	0.11	2.75e-09	5.7 (b)
$AM\_1000.mtx$	20	s_n	10^{-8}	22	398	0.12	8.93e-09	5.8 (a)
$sherman3.mtx$	10	z_n	10^{-8}	4	34	0.04	5.40e-13	5.8 (b)
$roadNet\text{-}PA.mtx$	20	t_n	10^{-14}	* 500	*9002	1680.24	no conv.	5.9 (a)
$com\text{-}Youtube.mtx$	20	s_n	10^{-14}	30	482	113.20	3.03e-15	5.9 (b)
$WikiTalk.mtx$	20	z_n	10^{-12}	* 500	*9002	4487.15	no converge	5.10

Table 5.2: *IRAM performances*

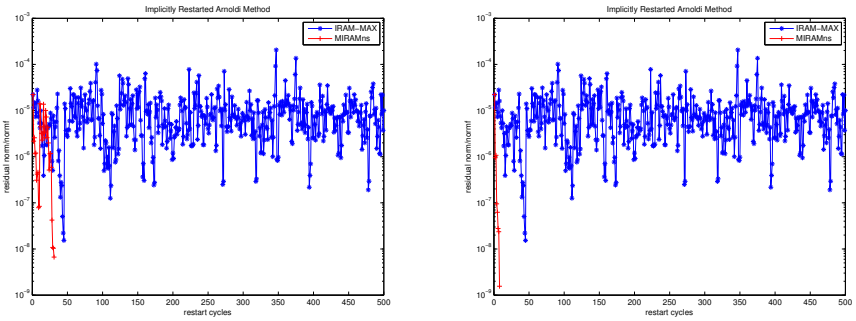
			IRAM			MIRAMns			
Matrix	Fig.	ℓ	m	nrc	MVP	$m_1, \cdots, m_\ell$	it	MVP	Ex.Time
af23560	5.1 (a)	3	10	* 500	* 4002	5, 7, 10	272	2178	7.84
	5.1 (b)	3	22	* 500	* 10002	5, 7, 10	250	2002	7.88
	5.2 (a)	3	22	* 500	* 10002	16, 19, 22	31	622	2.68
	5.2 (b)	5	22	* 500	* 10002	10, 13, 16, 19, 22	8	162	0.82
	5.12 (a)	3	25	6	140	5, 10, 25	6	140	0.63
	5.6 (a)	3	40	114	2865	20, 30, 40	26	665	5.83
bfw782a	5.4 (a)	3	10	* 500	* 4002	5, 8, 10	70	562	0.13
	5.4 (b)	3	20	* 500	* 9002	5, 10, 20	104	1874	0.32
	5.5 (a)	3	20	* 500	* 9002	8, 18, 20	92	1658	0.38
	5.5 (b)	5	20	* 500	* 9002	8, 11, 14, 17, 20	32	578	0.16
	5.6 (b)	3	30	12	195	20, 25, 30	14	225	0.13
A9_1000	5.7 (a)	3	20	309	5564	10, 15, 20	94	1694	0.36
west0989	5.7 (b)	3	30	53	426	5, 8, 10	32	258	0.07
AM_1000	5.8 (a)	3	20	22	398	13, 17, 20	17	308	0.09
sherman3	5.8 (b)	3	10	4	34	5, 8 , 10	2	18	0.02
roadNet-PA	5.9 (a)	6	20	* 500	*9002	5, 8, 11, 14, 17, 20	157	2828	599.7
com-Youtube ($k = 4$)	5.9 (b)	6	20	30	482	5, 8, 11, 14, 17, 20	20	332	108.08
WikiTalk ($k = 2$)	5.10 (a)	6	20	* 500	*9002	4, 7, 10, 13, 16, 20	312	5618	3438.7
WikiTalk ($k = 4$)	5.10 (b)	6	20	* 500	*9002	5, 8, 11, 14, 17, 20	15	272	181.5

Table 5.3: Comparison of IRAM(m) and MIRAMns($m_1, \cdots, m_\ell$)



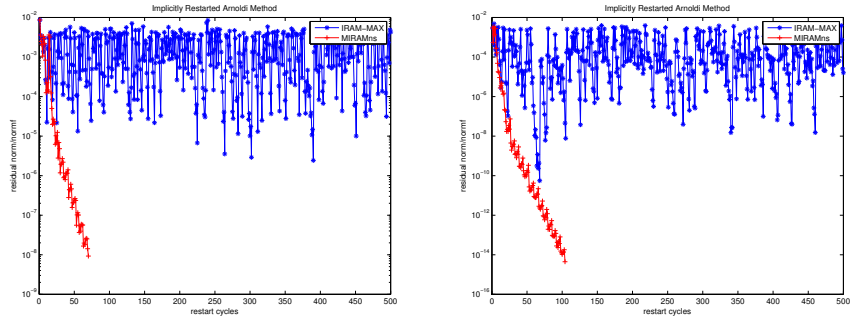
(a) $m = 10$ (b) $m = 22$

Fig. 5.1: MIRAMns$(5, 7, 10)$ versus IRAM(m) with $af23560$ matrix



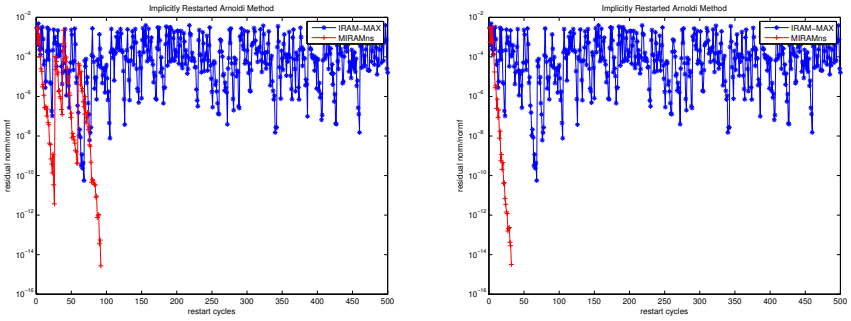
(a) MIRAMns$(16, 19, 22)$ (b) MIRAMns$(10, 13, 16, 19, 22)$

Fig. 5.2: MIRAMns$(m_1, \ldots, m_\ell)$ versus IRAM(22) with $af23560$ matrix



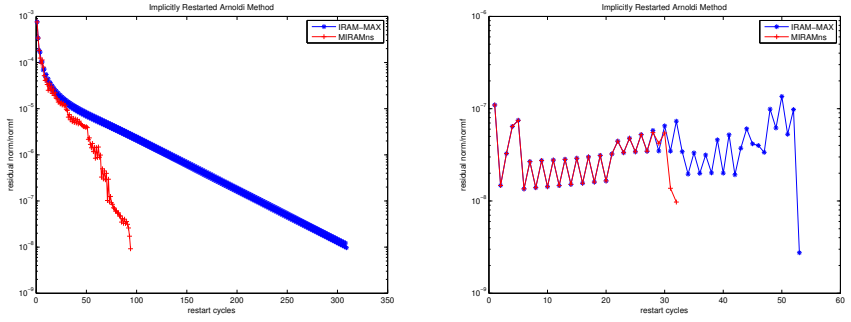
(a) MIRAMns$(5, 8, 10)$, $m = 10$ (b) MIRAMns$(5, 10, 20)$, $m = 20$

Fig. 5.3: MIRAMns$(m_1, \ldots, m_\ell)$ versus IRAM(m) with $bfw782a$ matrix



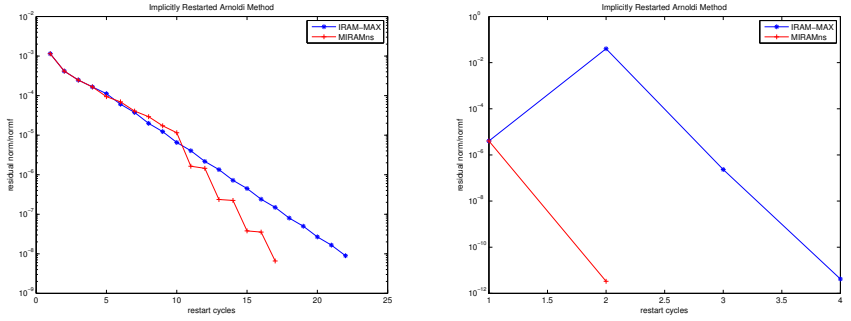
(a) MIRAMns$(8, 18, 20)$ (b) MIRAMns$(8, 11, 14, 17, 20)$

Fig. 5.4: MIRAMns$(m_1, \ldots, m_\ell)$ versus IRAM(20) with $bfw782a$ matrix



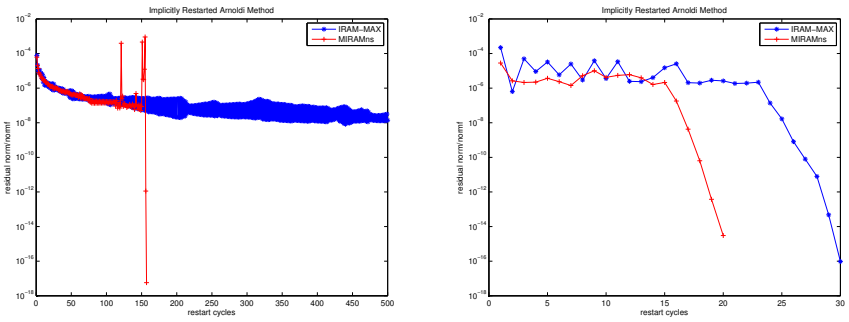
(a) MIRAMns$(10, 15, 20)$, $m = 20$ (b) MIRAMns$(5, 8, 10)$, $m = 10$

Fig. 5.5: MIRAMns$(m_1, \ldots, m_\ell)$ versus IRAM(m) with $A9\_1000$ in (a), $west0989$ in (b)



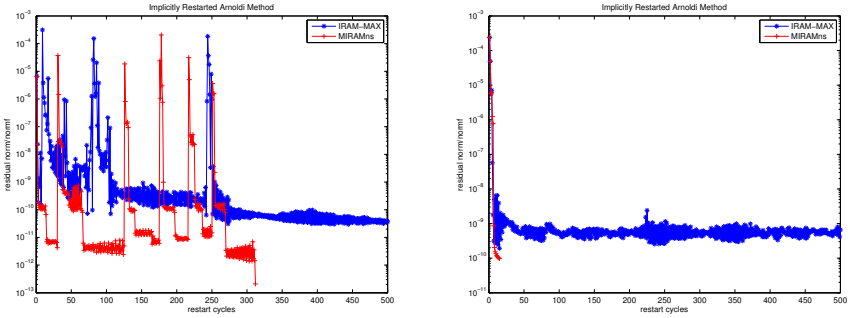
(a) MIRAMns$(13, 17, 20)$, $m = 20$ (b) MIRAMns$(5, 8, 10)$, $m = 10$

Fig. 5.6: MIRAMns$(m_1, \ldots, m_\ell)$ versus IRAM(m) with AM_1000 in (a), $sherman3$ in (b)



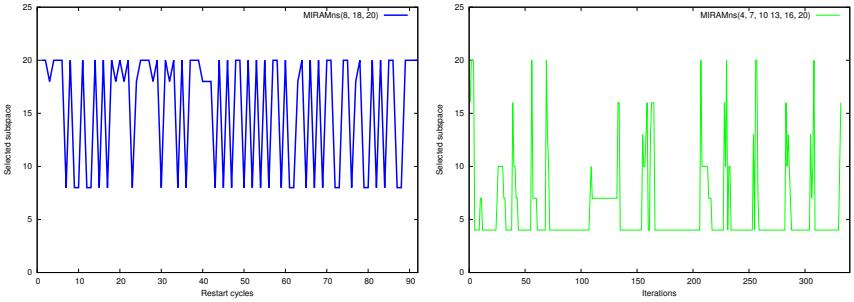
(a) *roadNet-PA* matrix, $k = 2$? (b) *com-Youtube* matrix, $k = 4$

Fig. 5.7: MIRAMns$(5, 8, 11, 14, 17, 20)$ versus IRAM(20)



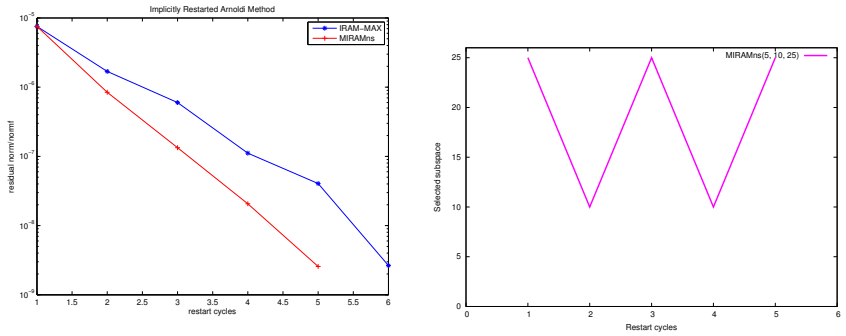
(a) MIRAMns$(4, 7, 10, 13, 16, 20)$, $k = 2$, $tol = 10^{-12}$ (b) MIRAMns$(5, 8, 11, 14, 17, 20)$, $k = 4$, $tol = 10^{-10}$

Fig. 5.8: MIRAMns$(m_1, \ldots, m_\ell)$ versus IRAM(20) with $WikiTalk$ matrix



(a) MIRAMns$(8, 17, 20)$ with $bfw782a$ matrix (b) MIRAMns$(4, 7, 10, 13, 16, 20)$ with $roadNet\text{-}PA$ matrix

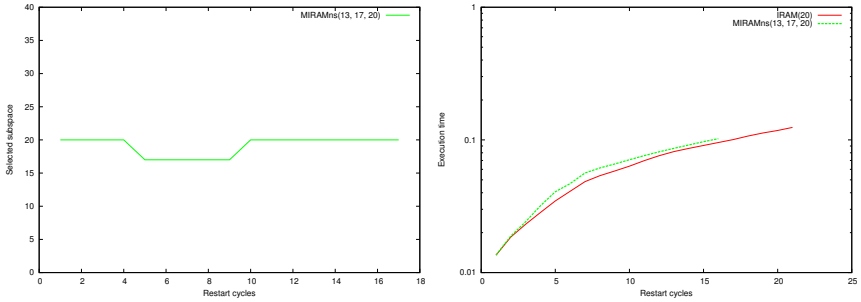
Fig. 5.9: Evolution of m_{best} in MIRAMns$(m_1, \ldots, m_\ell)$ along restart cycles



(a) MIRAMns$(5, 10, 25)$ versus IRAM(25) with $af23560$ matrix

(b) Evolution of m_{best} in MIRAMns$(5, 10, 25)$

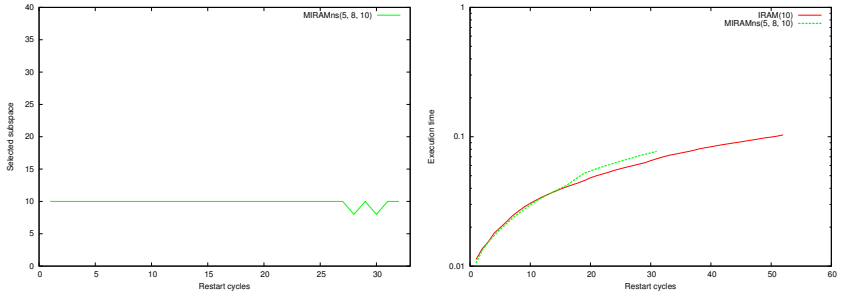
Fig. 5.10: Evolution of m_{best} in MIRAMns$(5, 10, 25)$ along restart cycles with $af23560$ matrix



(a) Subspace sizes along cycles

(b) Execution time

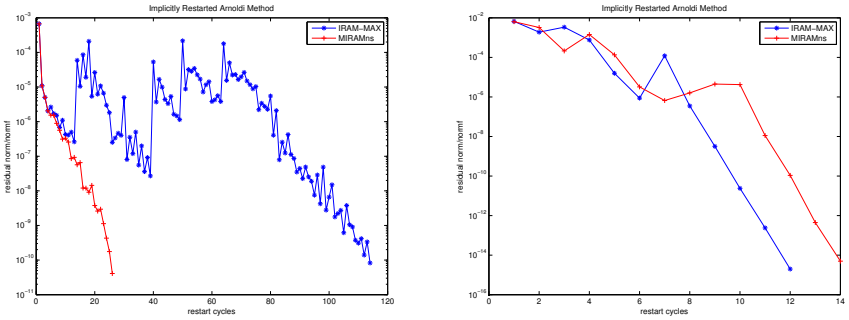
Fig. 5.11: MIRAMns$(13, 17, 20)$ versus IRAM(20) for AM_1000 matrix, tol=10^{-8}



(a) Subspace sizes along cycles

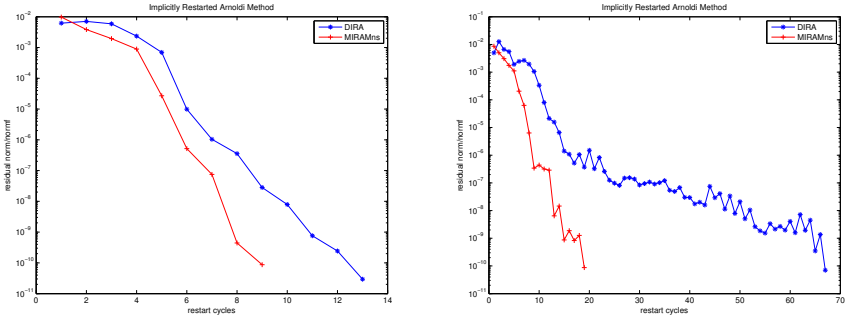
(b) Execution time

Fig. 5.12: MIRAMns(5,8,10) versus IRAM(10) for $west0989$ matrix



(a) MIRAMns$(20, 30, 40)$, $m = 40$, $q = 2$ (b) MIRAMns$(20, 25, 30)$, $m = 30$, $q = 5$

Fig. 5.13: MIRAMns$(m_1, \ldots, m_\ell)$ versus IRAM(m) with $af23560$ matrix, $k = 10$ with a buffer of q extra vectors



(a) MIRAMns$(15, 17, 19)$, $m = 19$, $k = 6$

(b) MIRAMns$(17, 19, 21, 23, 25)$, $m = 25$, $k = 10$

Fig. 5.14: MIRAMns$(m_1, \ldots, m_\ell)$ versus DIRA(m) with k eigenvaluesXXX

6. Conclusion. Due to the empirical choice of subspace size, the implicitly restarted Arnoldi method may not be efficient for computing a few selected eigenpairs of large sparse non-Hermitian matrices. In order to improve this choice, we have proposed to make use of this method with several Krylov subspaces. We have seen that the multiple implicitly restarted Arnoldi method with nested subspaces accelerates the convergence of IRAM with the same number of matrix-vector products in each restart cycle. Our numerical experiments have shown that MIRAMns improves the quality of the Krylov subspaces of IRAM and has consequently better convergence properties. Moreover, the strategy presented in the paper can be applied to many other restarted projection methods. In a general context, this is equivalent to coupling some iterative methods in order to accelerate the convergence of one of them as is the case of hybrid Arnoldi-Chebyshev method described in [4, 18]. We mentioned that for MIRAMns we can use the same parallel programming model as the one used in P_ARPACK [13] which implements the parallel implicitly restarted Arnoldi method. An implementation of MIRAMns within ARPACK is given in the following URL: (https://forge.prism.uvsq.fr). This is because we define a Krylov subspace and make use of the eigen-information of some subspaces nested in it. Another approach consists to make use of IRAM with several Krylov subspaces which differ by both their initial vector and subspace size. This approach, multiple IRAM, has the advantage to update (implicitly) the initial vector of an IRAM by taking into account the eigen-information obtained by several different (non nested) subspaces. The increase cost engendered by these different subspaces could be compensated by the implementation of the method in a large-scale distributed environment as for the multiple explicitly Arnoldi method in [8].

REFERENCES

[1] W. E. Arnoldi, *The Principle of Minimized Iteration in the Solution of Matrix Eigenvalue Problems*, Quart. Journal of Applied Mathematics, 9:pp. 17-29, 1951.

[2] Z. Bai, D. Day, J. Demmel and J.J. Dongarra, *A Test Matrix Collection for Non-Hermitian Eigenvalue Problems*, http://math.nist.gov/MatrixMarket.

[3] A. H. Baker, E. R. Jessup, Tz. V. KoleV, *A Simple Strategy for Varying the Restart Parameter in GMRES(M)*, J. Comput. Appl. Math., 230(2):751-761, August, 2009.

[4] F. Chatelain, *Eigenvalues of matrices*, Wiley,1993.

[5] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, *Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization*, Math. Comp., 1976.

[6] K. Dookhitram, R. Boojhawon, M. Bhuruth, *A new method for accelerating Arnoldi algorithms for large scale Eigenproblems*, Mathematics and Computers in Simulation, 2:387-401, 2010.

[7] I. S. Duff, J. A. Scott, *Computing selected eigenvalues of large sparse unsymmetric matrices using subspace iteration*, ACM Trans. Math. Software 19 (1993) 137159.

[8] N. Emad, S. Petiton, G. Edjlali, *Multiple Explicitly Restarted Arnoldi Method for Solving Large Eigenproblems*, SIAM Journal on Scientific Computing (SJSC), 27(1):253-277, 2005.

[9] M. Embree *TheTortoise and the Hare Restart GMRES* SIAM Review Vol. 45,No. 2,pp. 259266, 2003.

[10] R. B. Lehoucq *Analysis and Implementation of an Implicitly Restarted Iteration*, PhD thesis, Rice University, Houston, Texas, May 1995.

[11] R. B. Lehoucq, D.C. Sorensen *Deflation techniques for an implicitly restarted Arnoldi iteration*, SIAM J. Matrix Analysis and Applications, 17(4):789-821, October 1996.

[12] R. B. Lehoucq, D. C. Sorensen, C. Yang, *ARPACK User's Guide: Solution of Large Scale Eigenvalue Problems With Implicitly Restarted Arnoldi Methods*, SIAM, 1998.

[13] K. J. Maschhoff, D. C. Sorensen, *P_ARPACK: An efficient portable large scale eigenvalue package for distributed memory parallel architectures*, In Jerzy Wasniewski, Jack Dongarra, Kaj Madsen, and Dorte Olesen, editors, Applied Parallel Computing in Industrial Problems and Optimization, volume 1184 of Lecture Notes in Computer Science, Berlin, 1996.

Springer-Verlag.
[14] R. B. Morgan, *On Restarting the Arnoldi Method for Large Nonsymmetric Eigenvalue Problem*, Math. Comp., 65:1213-1230, 1996.
[15] K. Moriya, T. Nodera, *The DEFLATED-GMRES(m, k) method with switching the restart frequency dynamically*, Numer. Linear Algebra Appl., 7:569-584, 2000.
[16] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Inc., 1998.
[17] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, 1993.
[18] Y. Saad, *Chebyshev Acceleration Techniques for Solving Nonsymmetric Eigenvalue Problems*, Math. Com., 42, 567-588, 1984.
[19] Y. Saad, *Variations on Arnoldi's Method for Computing Eigenelements of Large Unsymmetric Matrices*, Linear Algebra Applications, 34, 269-295, 1980.
[20] Y. Saad, *Numerical Solution of Large Nonsymetric Eigen Problem*, Comput. Phys. Commun. 53 (1989), 71-90. MR 90f:65064
[21] D.C. Sorensen, *Implicit Application of Polynomial Filters in a k-step Arnoldi Method.*, SIAM Journal on Matrix Analysis and Applications, 13:357 - 385, 1992.
[22] D.C. Sorensen, *Implicitly Restarted Arnoldi/Lanczos Methods for Large Scale Eigenvalue Calculations*, (invited paper), Kluwer,1995.
[23] D.C. Sorensen, *Implicitly Restarted Arnoldi/Lanczos Methods for Large Scale Eigenvalue Calculations*, In D. E. Keyes, A. Sameh, and V. Venkatakrishnan, eds. Parallel Numerical Algorithms, pages 119-166, Dordrecht, 1997, Kluwer.
[24] A. Stathopoulos, Y. Saad, K. Wu, *Dynamic Thick Restarting of the Davidson, and the Implicitly Restarted Arnoldi Methods*, SIAM J. Sci. Comput., 19:227-245, 1996.
[25] R. B. Lehoucq, *Implicitly Restarted Arnoldi Methods and Subspace Iteration*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 551-562.
[26] *http://snap.stanford.edu/data/*