International Journal of Pure and Applied Mathematics
Volume 116 No. 4 2017, 999-1013
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: 10.12732/ijpam.v116i4.16

NEW TECHNIQUES FOR ACCELERATING PROJECTION METHODS FOR COMPUTING EIGENVALUES OF LARGE SPARSE MATRICES

Z. Abadi[1], S.A. Shahzadeh Fazeli[2] §, S.M. Karbassi[3]
[1,3]Department of Mathematics
Yazd University
University Blvd., Safayieh, 89195-741, Yazd, IRAN
[2]Parallel Processing Laboratory
Yazd University
University Blvd., Safayieh, 89195-741, Yazd, IRAN

Abstract: The explicitly Arnoldi method (ERAM) can be used to find some eigenvalues of large and sparse matrices. However, it has been shown that even this method may fail to converge. In this paper, we present two new methods to accelerate the convergence of ERAM method. In the first method, we will use a new variant of Krylov subspace, which we call it p-Krylov subspace; and in the second method, by using Noda iteration in the restart step, we accelerate the convergence of the explicitly Arnoldi method (ERAM). The implementation of the methods have been tested by numerical examples. The results show that we can obtain a good acceleration of the convergence compared to ERAM.

AMS Subject Classification: 15A18, 65F15, 65F50
Key Words: large eigenvalue problems, Krylov subspace, Arnoldi method, explicitly restarting, Noda iteration

1. Introduction

Eigenvalues and eigenvectors are highly important in applications. They arise in

Received: 2017-07-09
Revised: 2017-09-16
Published: November 19, 2017

§Correspondence author

many areas of mathematics, physics, chemistry and engineering. They arise in analytic geometry in connection with finding the particular coordinate system in which a conic in the plane or a quadric surface in three-dimensional space finds its simplest canonical expression. In physics and engineering they arise in connection with finding, for example, the critical frequencies of a vibrating string, suspension bridge or rotating shaft, the critical load of a supporting column or the energy levels of a system in quantum mechanics.

Several methods have been proposed to solve the eigenvalue problems for large-scale matrices. One of the important of these methods is Arnoldi process. This method, was proposed by Arnoldi in 1951[1]. Many algorithms are based on Arnoldi process. In fact, Arnoldi method is an orthogonal projection method on to Krylov subspace[13]. In this method, the problem of finding some eigenpairs of a matrix with large-scale dimension $n \times n$ converts to finding some eigenpairs of a matrix with smaller dimension $m \times m$ ($m << n$). However, this approximation may not be very good. In these cases, we can apply different techniques to improve approximation solutions. For example, restarted Arnoldi techniques. In 1980, Saad proposed a restarting of this method, which called explicitly restarted Arnoldi method (ERAM)[14]. This technique restarts the Arnoldi projection with a better 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. The restarting may be not easy, because one new starting vector must be defined as an explicit linear combination of desired Ritz vectors. If this vector is not carefully chosen, it can lead to a very bad selection for the new starting vector, thus the cost will be increased.

Another important technique for restarting and improving the Arnoldi method is implicitly restarted Arnoldi method (IRAM). Sorensen in [16] has suggested this efficient technique. The idea behind the implicitly restarted Arnoldi method (IRAM) is to reduce the cost by limiting the dimension of the search space. This means that the iteration is stopped after a number of steps (which is bigger than the number of desired eigenvalues), reducing the dimension of the search space without destroying the Krylov space structure, and finally resume the Arnoldi iteration. This efficient technique uses the QR algorithm to restart the Arnoldi projection. In fact, if the accuracy of Ritz elements of a large matrix A is not satisfactory, IRAM applies a QR shifted algorithm on the $m \times m$ matrix which represents A in the projection subspace. Therefore, IRAM is a method that combines the implicitly shifted QR technique with an Arnoldi factorization, and can be viewed as truncated form of the implicitly shifted QR iteration.

Sorensen approach permits to restart the Arnoldi process with an efficient

and numerically stable formulation. This approach was analyzed, implemented and validated, among others in[6, 7, 9, 10, 15, 16, 17]. In [11], Morgan showed that IRAM is much better than the other restarting Arnoldi methods. This successful method is foundation for the very popular eigenvalue software package ARPACK[8]. ARPACK, a set of FORTRAN subroutines that implements the implicitly restarted Arnoldi method, is described in [9]. MATLAB makes this code available through the function eigs.

The present paper considers a new variant of ERAM, called by ERpAM, for computation of a few extreme eigenvalues and associated eigenvectors of a large sparse matrix. This new method is based on constructing a new subspace by selecting some vectors of the Krylov subspace.

This paper is organized as follows:

Section 2, firstly, presents basic Arnoldi algorithm and explicitly restarted Arnoldi method (ERAM). Next, in Subsection 2.3, we introduce a new variant of Krylov subspace and use this subspace to improve ERAM. In Section 3, we apply the Noda iteration to accelerate the convergence of ERAM. A few numerical examples are presented in Section 4 and concluding remarks are contained in Section 5.

2. Restarted Arnoldi Methods

2.1. Basic Arnoldi Algorithm

Arnoldi method is one of the well-known Krylov subspace methods. Let A be a large non-Hermitian matrix of dimension $n \times n$ and v be a $n \times 1$ vector in $\mathbb{C}^n$. The Krylov subspace $K_m(A, v)$ of dimension m, is defined as follows:

$$K_m(A, v) = \text{span}\left\{v, Av, A^2v, \ldots, A^{m-1}v\right\}$$

The Arnoldi method, by using the Gram-Schimdt orthogonalization process, constructs $(v_1, v_2, \ldots, v_m)$ as an orthogonal basis for $K_m(A, v)$. A relation of the form

$$AV_m = V_m H_m + f_m e_m^T$$

is called an m-step Arnoldi factorization, where $V_m = [v_1, v_2, \ldots, v_m]$, H_m is an upper Hessenberg matrix with nonnegative subdiagonal elements, and $V_m^H f_m = 0$ where f is the residual vector of length n. In fact, this factorization converts the eigenvalue problem $Ax = \lambda x$ with the large order matrix A into a problem $H_m y = \lambda y$ with a smaller size matrix H_m. Let $y_i^{(m)}$ be an eigenvector of H_m associated with the eigenvalue $\lambda_i^{(m)}$ and let $u_i^{(m)} = V_m y_i^{(m)}$, then

$(u_i^{(m)}, \lambda_i^{(m)})$ is an approximate eigenpair for A. Called also Ritz eigenpair. The basic Arnoldi algorithm is as following[13]:

Algorithm 1. The basic Arnoldi algorithm

Input: $A \in \mathbb{R}^{n\times n}$, $v \in \mathbb{R}^{n\times 1}$, $m \in \mathbb{N}$ $(m << n)$
Output: $V \in \mathbb{R}^{n\times(m+1)}$, $H \in \mathbb{R}^{m+1\times m}$

1. $v_1 = \frac{v}{||v||}$; $z = Av_1$; $\alpha_1 = v_1^* z$;
2. $r_1 = z - \alpha_1 v_1$; $V_1 = [v_1]$; $H_1 = [\alpha_1]$;
3. for $j = 1, \ldots$ m-1 do
   - $\beta_j = ||r_j||$; $v_{j+1} = r_j/\beta_j$;
   - $V_{j+1} = [V_j, v_{j+1}]$; $\hat{H}_j = \begin{bmatrix} H_j \\ \beta_j e_j^T \end{bmatrix}$;
   - $z = Av_j$;
   - $h = V_{j+1}^* z$; $r_{j+1} = z - V_{j+1}h$;
   - $H_{j+1} = [\hat{H}_j, h]$;

   end for

2.2. ERAM Algorithm

In the Arnoldi algorithm, if the accuracy of the computed Ritz eigenpair is not sufficient, the projection can be restarted onto a new subspace. This new subspace can be defined with the same subspace size and a new initial vector. This technique is called the explicitly restarted Arnoldi method (ERAM). Therefore, if the convergence does not occur, then the starting vector is updated, and the process of Algorithm 1 is restarted until the accuracy of the approximated solution is satisfactory. An algorithm of the explicitly restarted Arnoldi is as following:

Algorithm 2. Explicitly Restarted Arnoldi Method (ERAM)

Input: $A \in \mathbb{R}^{n\times n}$, $k \in \mathbb{N}$ (number of wanted eigenpairs), $tol > 0$
Output: (λ_i, u_i), $i = 1, \ldots k$ (k Ritz eigenpairs of A)

1. Choose a parameter $m \in \mathbb{N}$, $(m << n)$ and an initial vector $v \in \mathbb{R}^{n\times 1}$
2. Apply Algorithm 1 to generate (V_m, H_m, f_m) and compute (λ_i, y_i), eigenpairs of H_m.
3. Set $u_i = V_m y_i$ and select k best Ritz eigenpair of A.
4. If all residual norm of Ritz eigenpairs$< tol$, Stop. Else use (λ_i, u_i) update the starting vector v and go to step 2.

2.3. A New Variant of Krylov Subspace

Consider the large eigenvalue problem $Ax = \lambda x$, where A is a $n \times n$ real, non-Hermitian matrix. (λ, x) is reffered to as an eigenpair of A. We are interested in computing some exterior eigenpairs of A. Let v be a $n \times 1$ vector, $p, m \in \mathbb{N}$ and $p < m$. Assume that $m - 1 = sp + s_0$ such that $0 \leq s_0 < p$. Therefore,

$$\begin{aligned} K_m(A, v) &= \left\{v, Av, \ldots, A^{m-1}v\right\} \\ &= \left\{\underline{v}, Av, \ldots, \underline{A^p v}, \ldots, \underline{A^{2p} v}, \ldots, \underline{A^{sp} v}, A^{sp+1}v, \ldots, A^{sp+s_0}v\right\} \end{aligned}$$

Now we use only vectors which are marked by underline and construct a new subspace $\left\{v, A^p v, A^{2p}v, \ldots, A^{sp}v\right\}$. Simply it can be seen that this new subspace is equal to the subspace $K_{s+1}(A^p, v)$.
Now, we define p-Krylov subspace as bellow:

$$K_m^p(A, v) = \text{span}\left\{v, A^p v, (A^p)^2 v, \ldots, (A^p)^s v\right\},$$

where $m-1 = sp+s_0$ and $0 \leq s_0 < p$. In this method, we use $K_m^p(A, v)$ instead of $K_m(A, v)$ in Arnoldi method, i.e. the Arnoldi process builds up an orthonormal basis $\{v_i\}_{i=1}^{s+1}$ of $K_m^p(A, v)$. We called this version of Arnoldi algorithm by p-Arnoldi method. This process is as follows:

Algorithm 3. p-Arnoldi algorithm (pAM)

Input: $A \in \mathbb{R}^{n\times n}$, $v \in \mathbb{R}^{n\times 1}$, $m \in \mathbb{N}$ $(m << n)$, $p \in \mathbb{N}$
Output: $V \in \mathbb{R}^{n\times(s+1)}$, $H \in \mathbb{R}^{s\times s}$, $r \in \mathbb{R}^{n\times 1}$

1. Set $s = [\frac{m-1}{p}]$
2. Set $v_1 = \frac{v}{||v||}$; $\quad z = v_1$

3. for $i = 1, \ldots, p$ do
   set $z = Az$;
   end;

4. Set $\alpha_1 = v_1^* z$; $\quad r_1 = w - \alpha_1 v_1$; $\quad V_1 = [v_1]$; $\quad H_1 = [\alpha_1]$;

5. for $j = 1, \ldots, s$ do
   set $\beta_j = ||r_j||$; $\quad v_{j+1} = \frac{r_j}{\beta_j}$;
   set $V_{j+1} = [V_j, v_{j+1}]$; $\quad \hat{H}_j = \begin{bmatrix} H_j \\ \beta_j e_j^T \end{bmatrix}$;
   set $z = v_j$;
   for $i = 1, \ldots, p$
   set $z = Av_j$
   end;
   Set $h = V_{j+1}^* z$; $\quad r_{j+1} = z - V_{j+1} h$;
   Set $H_{j+1} = \left[\hat{H}_j, h\right]$;
   end

Now, we use this version of Arnoldi method in ERAM algorithm. Therefore, we have the following procedure to compute approximation of k eigenvalues of matrix A with large magnitude (or small magnitude or etc...):

Algorithm 4. Explicitly restarted p- Arnoldi method (ERpAM)

Input: $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$ and $tol > 0$
Output: k Ritz pair of A: (λ_i, u_i), $\quad i = 1, \ldots, k$

1. Choose parameters m, p and an initial vector v
2. Set $s = [\frac{m-1}{p}]$
3. Compute (V_s, H_s, f) by p-Arnoldi algorithm (Algorithm 4)
4. Compute s eigenpairs of H_s: (λ_i, y_i), $\quad i = 1, \ldots, s$
5. Select k eigenvalues of H_s with large magnitude $\lambda_1, \lambda_2, \ldots, \lambda_k$ and eigenvectors associated with their, $y_1, y_2, \ldots, y_k$

6. Set $u_i = V_s y_i, \qquad i = 1, \ldots, k$
7. If all residual norm of Ritz eigenpairs$< tol$, Stop
8. Else update v by using $(\lambda_i, u_i),\ i = 1, \ldots s$ and goto step 3

Remark. In step 4 of Algorithm 2 and in step 8, of ERpAM algorithm (Algorithm 4) we compute the restarting vector with a linear combination of k Ritz vectors:

$$v(j) = \sum_{i=1}^{k} \alpha_i u_i(j)$$

where $u_i(j)$ denotes ith Ritz vector computed at the iteration j. There are several ways to choose the scalar values α_i. In this paper, we use the method proposed in[3]. For this reason, we set

$$v = \sum_{i=1}^{k} l_i(\lambda) u_i$$

where k coefficient $l_i(\lambda)$ are defined by

$$l_i(\lambda) = \prod_{j=1,\ j \neq i}^{k} \left(\frac{\lambda - \lambda_j}{\lambda_i - \lambda_j} \right)$$

with

$$\lambda = \frac{\lambda_{\min} + \bar{\lambda} - (\lambda_{\min}/n)}{2}, \qquad \bar{\lambda} = \frac{\sum_{j=1}^{k} \lambda_j}{k}$$

and $\lambda_{\min}$ is the eigenvalue with the smallest residual norm.
Now, we choose the vector v as follows:

$$v = \sum_{i=1}^{k} l_i(\lambda) u_i,$$

and use this vector for step 4 in Algorithm 2 and step 8 in Algorithm 4.

3. Noda-Arnoldi Hybrid Method

In this section, firstly, we assume the following definition and theorem:

Definition 1. Let $B = [b_{ij}] \in \mathbb{R}^{n\times n}$. If all of the entries of B are nonnegative (positive), then we write $B \geq 0 (> 0)$. Moreover, a nonnegative matrix B is said to be reducible if there exists a permutation matrix P such that

$$P^T BP = \begin{bmatrix} E & F \\ 0 & G \end{bmatrix},$$

where E and G are square matrices. Otherwise, matrix B is irreducible.

Theorem 2. *[4].Let B an irreducible nonnegative matrix. If $v > 0$ is not an eigenvector of B, then*

$$\min\left(\frac{Bv}{v}\right) < \rho(B) < \max\left(\frac{Bv}{v}\right).$$

(Notation: for a pair of vectors u and v,with $v > 0$, we have: $\max\left(\frac{u}{v}\right) = \max_i\left(\frac{u^{(i)}}{v^{(i)}}\right)$*)*

For an irreducible nonnegative matrix B, recall that the largest eigenvalue $\rho(B)$ is simple. Now, let x be the unit length positive eigenvector corresponding to $\rho(B)$.
In 1971, Noda presented an inverse iteration for computing the maximal eigenvalue of a nonnegative irreducible matrix B[12]. This iteration consists of three steps as follows:

1. $(\bar{\lambda}_k I - B)y_{k+1} = x_k$
2. $x_{k+1} = \frac{y_{k+1}}{||y_{k+1}||}$
3. $\bar{\lambda}_{k+1} = \max\left(\frac{Bx_{k+1}}{x_{k+1}}\right)$

Now, we use this idea for updating the initial vector in restarting step of ERAM algorithm, to accelerate the convergence of eigenvalue problem. For this reason, firstly, we consider the following algorithm:

Algorithm 5. Noda iteration algorithm

Input: $A \in \mathbb{R}^{n\times n}$, $v_0 \in \mathbb{R}^{n\times 1}$ and $k \in \mathbb{N}$
Output: v, λ

1. $v_0 = \frac{v_0}{||v_0||}, \qquad \lambda_0 = \max\left(\frac{Av_0}{v_0}\right)$
2. for $i = 0, \ldots, k-1$ do
   (a) solve the linear system $(\lambda_i I - A)y_{i+1} = v_i$
   (b) $v_{i+1} = \frac{y_{i+1}}{||y_{i+1}||}$
   (c) $\lambda_{i+1} = \max\left(\frac{Av_{i+1}}{v_{i+1}}\right)$
3. set $v = v_k$ and $\lambda = \lambda_k$

The above algorithm is based on the inverse iteration like shifted method by a Rayleigh quotient for approximation of the eigenvalues. In this algorithm, if $v_0 > 0$, and A is a nonnegative irreducible matrix, after k steps (for big enough value of k) λ_k converges to $\rho(A)$, and v_k will be approximation of associated eigenvector of $\rho(A)$[5].
Here, we use this algorithm to accelerate the convergence of ERAM algorithm. Because of this, we assume the following algorithm:

Algorithm 6. Explicitly restarted Noda-Arnoldi method (ERNAM)

Input: $A \in \mathbb{R}^{n\times n}$, $k \in \mathbb{N}$ (number of desired eigenpairs), $tol > 0$
Output: (λ_i, u_i), $i = 1, \ldots k$ (k Ritz eigenpairs of A)

1. Choose a parameter $m \in \mathbb{N}$, $(m << n)$ and an initial vector $v_0 \in \mathbb{R}^{n\times 1}$
2. Perform Alg. 5 for input: A,v and an positive integer number s, for gaining better initial vector v.
3. Apply Alg.1 to generate (V_m, H_m, f_m) and compute (λ_i, y_i), eigenpairs of H_m.
4. Set $u_i = V_m y_i$ and select k best Ritz eigenpair of A.
5. If all residual norm of Ritz eigenpairs$< tol$, Stop. Else use (λ_i, u_i) update the starting vector v_0 and go to step 2.

Matrix	Matrix size	nnz
$af23560$	23560	484256
$bcsstk29$	13992	619488
$ck656$	656	3884
$dwb512$	512	2500
$dwg961b$	961	10591
$plat1919$	1919	32399
$tols4000$	4000	8784

Table 1: The matrix market used matrices

4. Numerical Experiments

Algorithm 4 and Algorithm 6 are tested for various matrices by MATLAB software. In this section, we report some of these numerical examples for comparing ERAM algorithm by using p-Krylov subspace method (ERpAM) versus ERAM algorithm by using original Arnoldi method, and, moreover, Noda-Arnoldi hybrid method (ERNAM) versus ERAM algorithm .

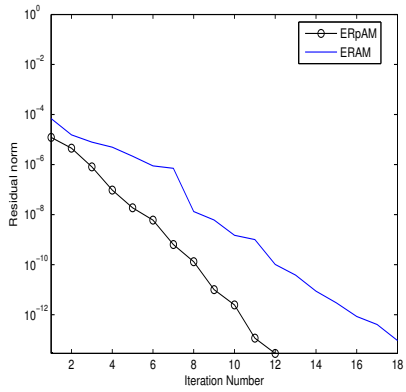
All the numerical experiments presented in this section were carried out on an Intel(R) Core(TM) i7-2630QM CPU @ 2.00GHz using MATLAB software. In all examples we use the initial vector $v = (1, 1, \ldots, 1)^T$. Moreover, if the relative maximum residual norm $r_j = \max(\rho_1, \rho_2, \ldots, \rho_k) < tol$ where $\rho_i = \frac{||(A-\lambda_i)u_i||}{||A||_F}$ with tol a prescribed tolerance, then (λ_i, u_i) is accepted to have converged. Every other stopping criterion can replace the requirement to find k eigenvalues. In all the Tables below, we denote by $iter$ the number of restarts and $error$ the maximum of above relative residual norms. Also $*$ represents that in the method, convergence has not been reached. Moreover, we denote the type of desired eigenvalues by $which$. If $which$ ='lm', then the eigenvalues of largest magnitude are desired. Similarly, 'sm', 'lr' and 'sr' are used for eigenvalues of smallest magnitude, largest real part and smallest real part, respectively. The used matrices are taken from the matrix market[2] and presented in Table 1. In this Table the number of nonzero elements of a matrix is denoted by nnz.

4.1. Arnoldi Versus p-Arnoldi

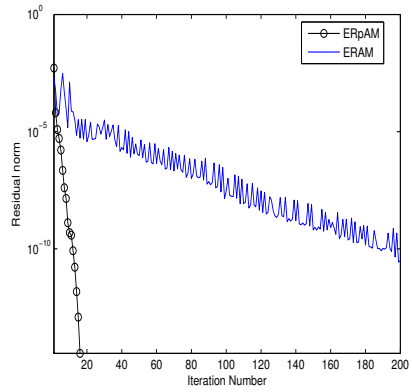
We report some numerical examples to illustrate the efficiency and reliability of ERpAM algorithm (Algorithm 4) versus ERAM (Algorithm 2). The Table 2 presents a comparison between the results obtained by ERAM and ERpAM al-

				ERAM		ERpAM		
Matrix	m	*which*	k	error	iter	p	error	iter
$af23560$	40	'lr'	1	9.2846e-14	18	2	2.8539e-14	12
$af23560$	40	'lr'	3	3.0408e-11	200*	2	3.4703e-15	16
$ck656$	10	'lm'	1	1.0042e-15	24	2	1.1513e-14	6
$tols4000$	20	'lm'	1	7.9190e-14	60	2	9.0425e-14	35

Table 2: Comparison of ERAM (Algorithm 2) and ERpAM (Algorithm 4)



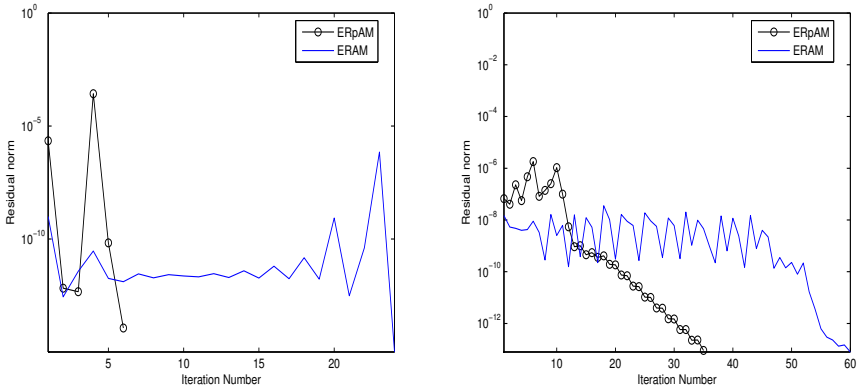
(a) $af23560$, $m = 40$, $k = 1$, $p = 2$



(b) $af23560$, $m = 40$, $k = 3$, $p = 2$

Figure 1: ERpAM versus ERAM with emphasis on number of restart cycles for matrices $af23560$

gorithms on some large-scale sparse matrices, in terms of the number of restarting and the error. In these examples we set $tol = 10^{-13}$. We notice from this Table that in terms of the number of the restarts, ERpAM is considerably more efficient than ERAM. We show graphically in Figs. 1-2 the residual norm as a function of iteration number to reach convergence using ERAM and ERpAM.



(a) $ck656$, $m = 10$, $k = 1$, $p = 2$ (b) $tols4000$, $m = 20$, $k = 1$, $p = 2$

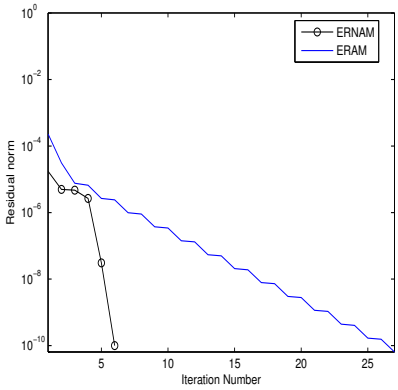
Figure 2: ERpAM versus ERAM with emphasis on number of restart cycles for matrices $ck656$, $tols4000$

			ERAM		ERNAM	
Matrix	m	k	error	iter	error	iter
$bcsstk29$	20	1	6.4631e-11	27	9.9888e-11	6
$dwb512$	20	3	5.7396e-06	200*	5.5882e-11	148
$dwg961b$	20	1	1.2267e-10	200*	6.1642e-11	37
$plat1919$	20	5	5.8850e-11	33	4.6677e-11	23

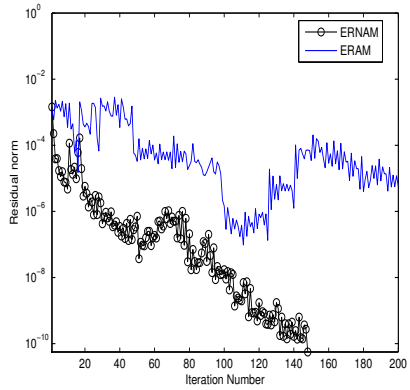
Table 3: Comparison of ERAM (Algorithm 2) and ERNAM (Algorithm 6)

4.2. Arnoldi Versus Noda-Arnoldi

In this section, we report the result of some examples, to compare Noda-Arnoldi hybrid method with ERAM. Table 3. presents numerical comparison of ERAM (Algorithm 2) and ERNAM (Algorithm 6) for $bcsstk29$, $dw512$, $dwg961b$ and $plat1919$ matrices. The type of desired eigenvalues in these examples is 'lr'and $tol = 10^{-10}$. Moreover, Figs. 3-4 , indicate that our ERNAM algorithm is better than ERAM.

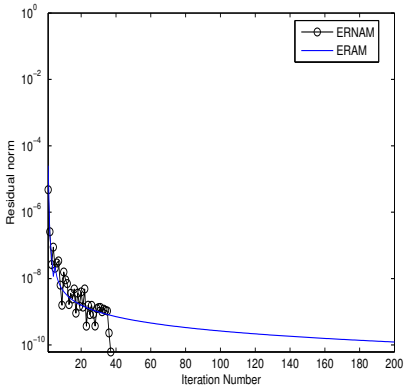


(a) $bcsstk29$, $m = 20$, $k = 1$

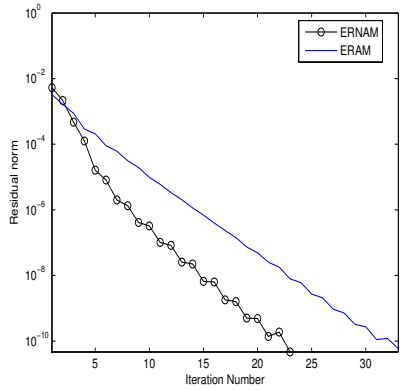


(b) $dwb512$, $m = 20$, $k = 3$

Figure 3: ERNAM versus ERAM with emphasis on number of restart cycles for matrices $bcsstk29$ and $dwb512$



(a) $dwg961b$, $m = 20$, $k = 1$



(b) $plat1919$, $m = 20$, $k = 5$

Figure 4: ERNAM versus ERAM with emphasis on number of restart cycles for matrices $dwg961b$ and $plat1919$

5. Conclusion

In this paper, we proposed two new techniques for computing eigenvalues of large-scale sparse matrices. Numerical examples indicate that new methods presented here often performs much better than the original explicitly restart-

ing Arnoldi method (ERAM). We expect the resulting algorithms to be more powerful.

References

[1] W.E. Arnoldi , The Principle of Minimiaed Iteration in the Solution of Matrix Eigenvalue Problems, *Quart. J. Appl. Math.*, **9** (1951), 17-29.

[2] Z. Bai , D. Day, J. Demmet and J. Dongarra , A Test Matrix Collection for Non-Hermitian Eigenvalue Problems, *http://math.nist.gov/MatrixMarket*, (1996).

[3] N. Emad, S.A. Shahzadeh-Fazeli and J. Dongarra, An asynchronous algorithm on the Netsolve global computing system, *Future Generation Computer Systems*, **22** (2006), 279-290. **doi**: 10.1016/j.future.2005.10.003

[4] R.A. Horn and C.R. Johnson , *Matrix Analysis*, The Cambridge University Press, Cambridge, UK (1985). **doi**: 10.1017/CBO9780511810817

[5] Z. Jia, W.W. Lin and C.S. Liu, An Inexact Noda Iteration for Computing the Smallest Eigenpair of an Irreducible M-Matrix, *Numer. Math.*, **130**,4 (2015), 645-679.

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

[7] R.B. Lehoucq, D.C. Sorensen, Deflation techniques for an implicitly restarted Arnoldi iteration, *SIAM J. Matrix Analysis and Applications*, **17**, 4 (1996), 789-821. **doi**: 10.1137/S0895479895281484

[8] R.B. Lehoucq, D.C. Sorensen, P.A. Vu and C. Wang, *ARPACK: An implementation of an implicitly restarted Arnoldi method for computing some of the eigenvalues and eigenvectors of a large sparse matrix*, Houston, Texas (1996).

[9] R.B. Lehoucq, D.C. Sorensen and C. Yang , *ARPACK User's Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, (1997).

[10] K.J. Maschhoff, D.C. Sorensen, J. Wasniewski, J. Dongarra, K. Madsen, D. Olesen(eds.), *Applied Parallel Computing in Industrial Problems and Optimization*, volume 1184 of Lecture Notes in Computer Science, Springer-Verlag, Berlin (1996).

[11] R.B. Morgan, On Restarting the Arnoldi Method for Large Nonsymmetric Eigenvalue Problem, *Math. Comp.*, **65** (1996), 1213-1230. **doi**: 10.1090/S0025-5718-96-00745-4

[12] T. Noda, Note on the computation of the maximal eigenvalue of a non-negative irreducible matrix, *Number. Math.*, **17** (1971), 382-386. **doi**: 10.1007/BF01436087

[13] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press (1993). **doi**: 10.1137/1.9781611970739

[14] Y. Saad, Variations on Arnoldi's Method for Computing Eigenelements of Large Unsymmetric Matrices, *Linear Algebra Applications*, **34** (1980), 269-295. **doi**: 10.1016/0024-3795(80)90169-X

[15] S.A. Shahzadeh Fazeli, N. Emad and Z. Liu, A key to choose subspace size in implicitly restarted Arnoldi method, *Numerical Algorithms.*, **70**, 2 (2015), 407426. **doi**: 10.1007/s11075-014-9954-5

[16] D.C. Sorensen, Implicit Application of Polynomial Filters in a k-step Arnoldi Method, *SIAM Journal on Matrix Analysis and Applications*, **13** (1992), 357-385. **doi**: 10.1137/0613025

[17] D.C. Sorensen, Implicitly Restarted Arnoldi/Lanczos Methods for Large Scale Eigenvalue Calculations. In: *D. E. Keyes, A. Sameh, V. Venkatakrishnan (eds.)*, Parallel Numerical Algorithms, Kluwer, Dordrecht (1997), 119-166. **doi**: 10.1007/978-94-011-5412-3_5