Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

Some new restart vectors for explicitly restarted Arnoldi method

Z. Abadi*[1], S. A. Shahzadeh Fazeli†[2] and S. M. Karbassi ‡[3]

[1,3]Department of Mathematical Science, Yazd University, Yazd, Iran.
[2]Department of Computer Science, Yazd University, Yazd, Iran.

ABSTRACT

The explicitly restarted 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 algorithm. In these methods, we apply two strategies for the updated initial vector in each restart cycles. 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 original ERAM.

Keyword: Large eigenvalue problems, Krylov subspace, Arnoldi method, Explicitly restarted, Restarting vector

ARTICLE INFO

Article history:
Received 3, January 2019
Received in revised form 12, April 2019
Accepted 11 May 2019
Available online 01, June 2019

AMS subject Classification: 15A18, 65F15, 65F50

1 Introduction

Eigenvalues and eigenvectors are highly important in applications. They arise in many areas of mathematics, physics, chemistry and engineering. They arise in analytic geometry

*zeinab.abadi@stu.yazd.ac.ir
†Corresponding author: S. A. Shahzadeh Fazeli. Email: fazeli@yazd.ac.ir
‡smkarbassi@yazd.ac.ir

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[2]. Many algorithms are based on Arnoldi process. In fact, Arnoldi method is an orthogonal projection method on to Krylov subspace[10]. 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 effective. In these cases, we can apply different techniques to improve approximation solutions, such as restarted Arnoldi techniques. In 1980, Saad proposed a restarting of this method, which be called explicitly restarted Arnoldi method (ERAM)[10]. 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 not be 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 poor 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 [14] has suggested this efficient technique.
In addition, the several hybrid methods were proposed to accelerate the convergence and/or to improve the accuracy of the ERAM or IRAM. These methods combine several different methods by parameterized copies of the same method to solve these problems efficiently[1, 4, 5, 6, 8, 9, 11, 12, 13].
The present paper considers two new version of ERAM, for computation of a few extreme eigenvalues and associated eigenvectors of a large sparse matrix. In fact, in these methods we change the subspace by changing the initial vector in each restart cycles. Therefore, this paper organized as follows:
In section2, we review the problem and some related notations. The section 3 presents the Arnoldi method and explicitly restarted Arnoldi method (ERAM). In section 4, we describe two new invariants of ERAM and their algorithms. These algorithms are evaluated in section 5 by a set of matrices, and the results are compared with the original ERAM.

2 General Purpose and Notations

Let A be a large complex non-Hermitian matrix of dimension $n \times n$ and $\mathcal{K}$ be a subspace of $\mathbb{C}^n$. An orthogonal projection method allows to approximate an eigenpair (λ_i, u_i) of A

by a Ritz-elements pair $(\lambda_i^{(m)} \in \mathbb{C}, u_i^{(m)} \in \mathcal{K})$. If the subspace $\mathcal{K} = \mathcal{K}_{m,v}$ where

$$\mathcal{K} = \mathcal{K}_{m,v} = \text{span}\{v, Av, \ldots, A^{m-1}v\}$$

and v is a vector, then the orthogonal projection method is called Krylov subspace method. The subspace $\mathcal{K}_{m,v}$ is called Krylov subspace.
This method approximates k eigenpairs of A by a matrix of order m obtained by an orthogonal projection onto an m-dimensional subspace $\mathcal{K}_{m,v}$ with $k \leq m << n$.
Let V_m be the matrix whose columns $v_1, \ldots, v_m$ are an orthogonal basis of $\mathcal{K}_{m,v}$. The problem is to find $\lambda_i^{(m)} \in \mathbb{C}$ and $y_i^{(m)} \in \mathbb{C}^m$ such that

$$(H_m - \lambda_i^{(m)} I) y_i^{(m)} = 0 \tag{1}$$

where the matrix H_m of dimension $m \times m$, is defined by $H_m = V_m^H A V_m$. Note that V_m^H is the transpose conjugate of V_m and $u_i^{(m)} = V_m y_i^{(m)}$. Therefore, some eigenvalues of A can be approximated by the eigenvalues of the matrix H_m. These eigenvalues can be found by building an orthogonal basis of $\mathcal{K}_{m,v}$ and by solving equation (1). There are different ways of building such basis and the most used process is the Arnoldi's orthogonalization.

3 Arnoldi method

Let the initial guess v_1 be normalized to $v/||v||_2$. The well-known Arnoldi process generates an orthogonal basis $v_1, \ldots, v_m$ of the Krylov subspace $\mathcal{K}_{m,v}$ by using the Gram-Schmidt orthogonalization process. The basic Arnoldi algorithm is as follows[10]:

Algorithm 1. The 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

The above algorithm may break down if $r_j = 0$ for some j. This may happen if the minimal polynomial of v is of degree j. In this case, the subspace $\mathcal{K}_{m,v}$ is invariant under A and the Ritz elements are exact [10]. This method was introduced by Arnoldi[2] in 1951 to reduce a matrix to a Hessenberg form. The reduced matrix H_m is a Hessenberg representation of A in the orthonormal basis V_m of $\mathcal{K}_{m,v}$ when $m = n$. Arnoldi hinted that the process could give good approximations to some eigenvalues of A if stopped before completion [1], i.e., when $m < n$. Today, it is the most common used method. The matrices H_m and V_m issued from the algorithm 1 and the matrix A satisfy the equation:

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

where $f_m = h_{m+1,m} v_{m+1}$ and e_m is the mth vector of the canonical basis of $\mathbb{C}^m$. Once the choice of the orthogonalization process is fixed, the k desired Ritz values (with largest/smallest real part or largest/smallest magnitude) $\Lambda_m = (\lambda_1^{(m)}, \ldots, \lambda_k^{(m)})$ and the corresponding Ritz vectors $U_m = (u_1^{(m)}, \ldots, u_k^{(m)})$ can be calculated as follows[6]:

Algorithm 2. The basic Arnoldi algorithm

Input: $A \in \mathbb{R}^{n \times n}$, $k \in \mathbb{N}$ (number of desired eigenpairs), $tol > 0$
Output: $\Lambda_m = (\lambda_1^{(m)}, \ldots, \lambda_k^{(m)})$, $U_m = (u_1^{(m)}, \ldots, u_k^{(m)})$, r_k

1. Apply Algorithm 1 to generate (V_m, H_m, f_m)
2. Compute the eigenpairs of H_m and select the k desired ones.
3. Compute the k associated Ritz vectors $u_i^{(m)} = V_m y_i^{(m)}$.
4. Compute $r_k = (\rho_1, \ldots, \rho_k)$ with $\rho_i = ||(A - \lambda_i^{(m)} I) u_i^{(m)}||_2$.

If the accuracy of the computed Ritz elements is not good enough, the projection can be restarted again onto a new $\mathcal{K}_{m,v}$.

3.1 Restarted Arnoldi method with a larger subspace

The new $\mathcal{K}_{m,v}$ can be defined with the same initial vector v and a larger m value. It is clear that, according to the hypothesis that v does not belong to any desired invariant subspace, m has to be as large as possible. An important, well-known, shortcoming of this version of the Arnoldi method is its alarmingly large storage space requirement and computation cost for large values of m.

3.2 Explicitly Restarted Arnoldi Method

In this version of the method, the new subspace can be defined with the same subspace size and a new initial vector. Therefore, if the convergence does not occur, then the starting vector is updated (using appropriate methods on the computed Ritz vectors), and the process of Algorithm 1 is restarted until the accuracy of the approximated solution is satisfactory. This update is designed to force the vector to be in the desired invariant subspace. This goal can be reached by some polynomial restarting strategies proposed in [10]. This method is called the explicitly restarted Arnoldi method(ERAM) and its algorithm is given below[10]:

Algorithm 3. Explicitly Restarted Arnoldi Method (ERAM)

Input: $A \in \mathbb{R}^{n\times n}$, $k \in \mathbb{N}$ (number of desired eigenpairs), $tol > 0$
Output: $\Lambda_m = (\lambda_1^{(m)}, \ldots, \lambda_k^{(m)})$, $U_m = (u_1^{(m)}, \ldots, u_k^{(m)})$, r_k

1. Choose a parameter $m \in \mathbb{N}$, $(m << n)$ and an initial vector $v \in \mathbb{R}^{n\times 1}$
2. Apply Algorithm 2 to generate (Λ_m, U_m, r_k).
3. If $\max(\rho_1, \ldots, \rho_k) > tol$, then use Λ_m and U_m to update the starting vector v and go to step 2.

4 Some new variants of ERAM

The restarting strategy is a critical part of explicitly restarted Arnoldi algorithm. In this section, we propose two new methods for the restarting technique to improve the convergence of ERAM.

4.1 Version 1

In this version, we use the vector Av instead of v in the start of the ERAM algorithm. We call this algorithm by V1-ERAM. The process of V1-ERAM is as follows:

Algorithm 4. Version 1 of ERAM (V1-ERAM)

Input: $A \in \mathbb{R}^{n\times n}$, $k \in \mathbb{N}$ (number of desired eigenpairs), $tol > 0$
Output: $\Lambda_m = (\lambda_1^{(m)}, \ldots, \lambda_k^{(m)})$, $U_m = (u_1^{(m)}, \ldots, u_k^{(m)})$, r_k

1. Choose a parameter $m \in \mathbb{N}$, $(m << n)$ and an initial vector $v \in \mathbb{R}^{n\times 1}$

2. Set $v := Av$.
3. Apply Algorithm 2 to generate (Λ_m, U_m, r_k).
4. If $\max(\rho_1, \ldots, \rho_k) > tol$, then use Λ_m and U_m to update the starting vector v and go to step 3.

4.2 Version 2

In this version of ERAM, we use vector Av instead of v in the start step and all of restart steps of ERAM algorithm. This strategy is equivalent to using the subspace $\mathcal{K}_{m,v}^{new} = \text{span}\{Av, A^2v, \ldots, A^{m-1}v\}$ instead of $\mathcal{K}_{m,v}$ in the Arnoldi process. Note that, the new subspace $\mathcal{K}_{m,v}^{new}$ is introduced by deleting the vector v from the subspace $\mathcal{K}_{m,v}$. The algorithm of this version of ERAM is as follows:

Algorithm 5. Version 2 of ERAM (V2-ERAM)

Input: $A \in \mathbb{R}^{n\times n}$, $k \in \mathbb{N}$ (number of desired eigenpairs), $tol > 0$
Output: $\Lambda_m = (\lambda_1^{(m)}, \ldots, \lambda_k^{(m)})$, $U_m = (u_1^{(m)}, \ldots, u_k^{(m)})$, r_k

1. Choose a parameter $m \in \mathbb{N}$, $(m << n)$ and an initial vector $v \in \mathbb{R}^{n\times 1}$
2. Set $m := m - 1$.
3. Set $v := Av$.
4. Apply Algorithm 2 to generate (Λ_m, U_m, r_k).
5. If $\max(\rho_1, \ldots, \rho_k) > tol$, then use Λ_m and U_m to update the starting vector v and go to step 3.

---

5 Numerical examples

Algorithms 4, 5 and ERAM algorithm (Algorithm 3) are tested for various matrices by MATLAB software. In this section, we report some of these numerical examples to illustrate the efficiency and reliability of these new versions of ERAM algorithm.

Remark. In step 3 of the ERAM algorithm (algorithm 3), step 4 of algorithm 4 and in step 5 of the algorithm 5, 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[7]. For this reason, we set

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

where the 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 3 in Algorithm 2, Step 4 in Algorithm 4 and Step 5 in Algorithm 5.
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 used 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 as prescribed tolerance, then (λ_i, u_i) is accepted to have converged. Every other stopping criterion can replace the requirement to find k eigenvalues. From now on, 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. The used matrices are taken from the matrix market[3] and presented in Table 1. In this Table, the number of nonzero elements of a matrix is denoted by nnz.

Table 1. The matrix market used matrices

Matrix	Matrix size	nnz
$bcsstk29$	13992	619488
$bcsstm13$	2003	21181
$bfw782b$	782	5982
$bp1600$	822	4841
$pde2961$	2961	14585
$rdb2048l$	2048	12032
$tols4000$	4000	8784
$utm1700a$	1700	21313
$west2021$	2021	7310

The Table 2 presents a comparison between the results obtained by ERAM, V1-ERAM and V2-ERAM algorithms on some large-scale sparse matrices, in terms of the number of restarting and the error. In these examples, we set $tol = 10^{-10}$. We notice from this Table that in terms of the number of the restarts, nv1-ERAM and nv2-ERAM are considerably more efficient than ERAM. We show graphically in Figs. 1-11 the residual norm as a function of iteration number to reach convergence using ERAM, V1-ERAM and V2-ERAM.

Table 2. Comparison of ERAM (Algorithm 3), V1-ERAM (Algorithm 4) and V2-ERAM (Algorithm5)

			ERAM		V1-ERAM		V2-ERAM	
Matrix	m	k	error	iter	error	iter	error	iter
$bcsstk29$	20	3	8.38e-11	119	8.44e-11	118	9.16e-11	83
$bcsstm13$	10	3	2.00e-11	21	2.71e-11	19	7.23e-11	17
$bfw782b$	20	3	1.44e-05	200*	8.02e-12	164	1.54e-05	200*
$bp1600$	20	3	1.46e-11	46	3.77e-11	25	3.16e-11	14
	10	1	2.20e-11	200*	8.99e-11	115	3.31e-11	146
$pde2961$	20	3	7.79e-11	46	5.59e-11	41	9.82e-11	28
$rdb2048l$	20	3	7.39e-04	200*	2.81e-08	200*	7.94e-11	51
$tols4000$	20	1	9.89e-11	39	6.90e-11	15	7.39e-11	17
$utm1700a$	20	2	4.04e-10	200*	8.73e-11	96	8.78e-11	134
$west2021$	20	3	5.34e-11	7	6.07e-12	2	9.86e-11	5
	20	5	2.09e-12	8	1.70e-11	4	1.17e-11	4

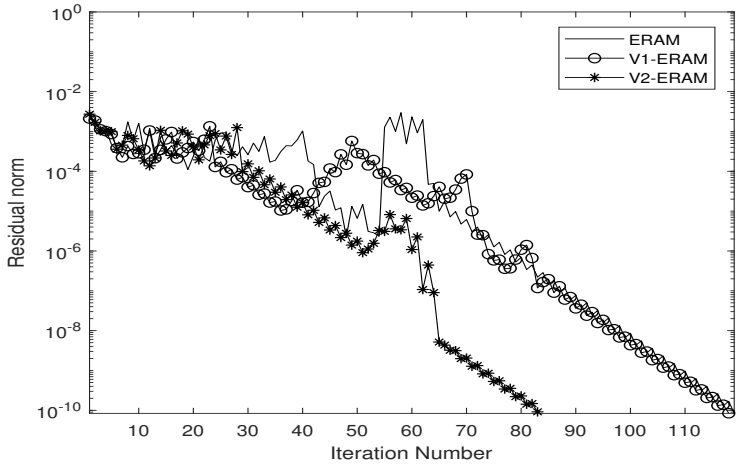

Figure 1: The residual norm for matrix *bcsstk*29 with $m = 20$ andp $k = 3$

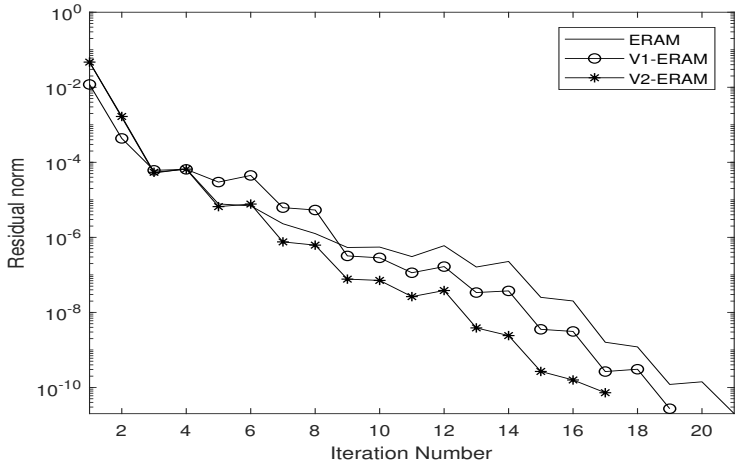

Figure 2: The residual norm for matrix *bcsstm*13 with $m = 10$ and $k = 3$

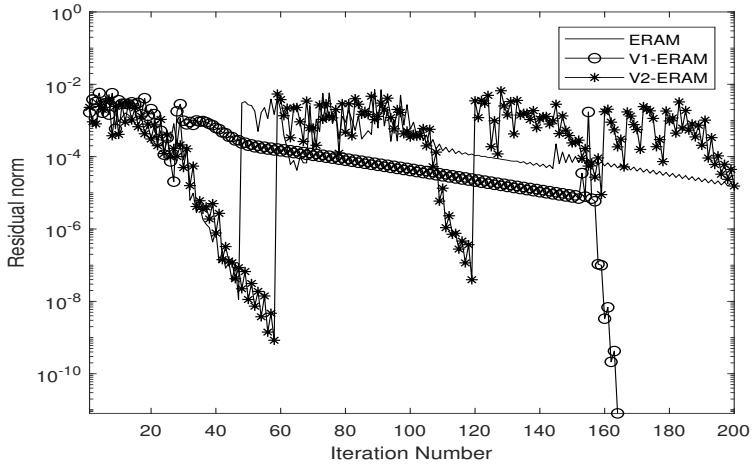

Figure 3: The residual norm for matrix $bfw782b$ with $m = 20$ and $k = 3$

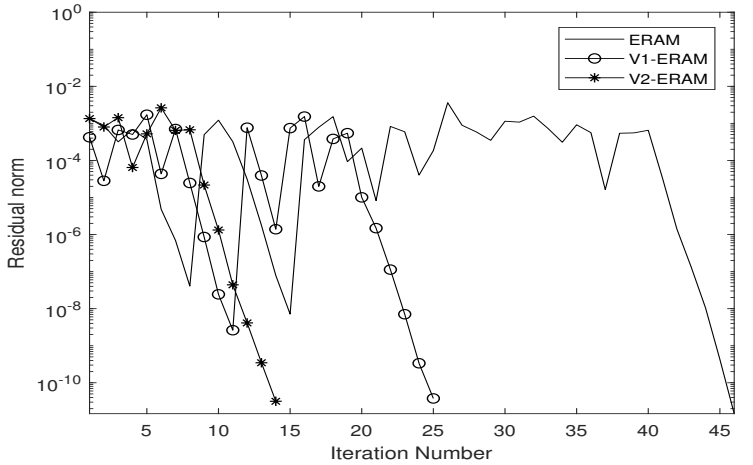

Figure 4: The residual norm for matrix $bp1600$ with $m = 20$ and $k = 3$

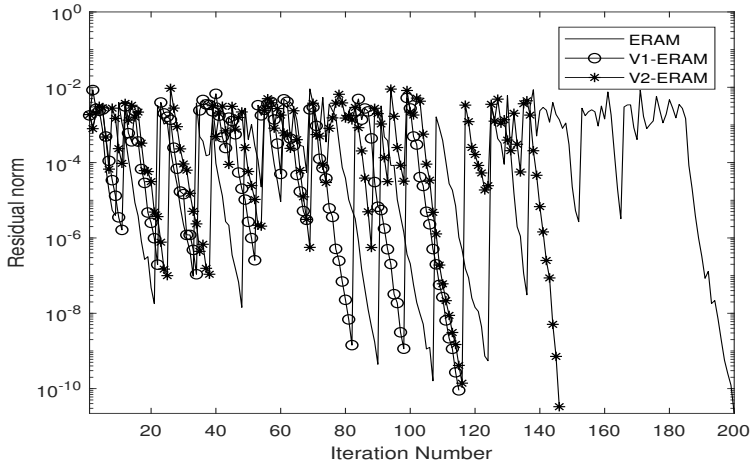

Figure 5: The residual norm for matrix $bp1600$ with $m = 10$ and $k = 1$

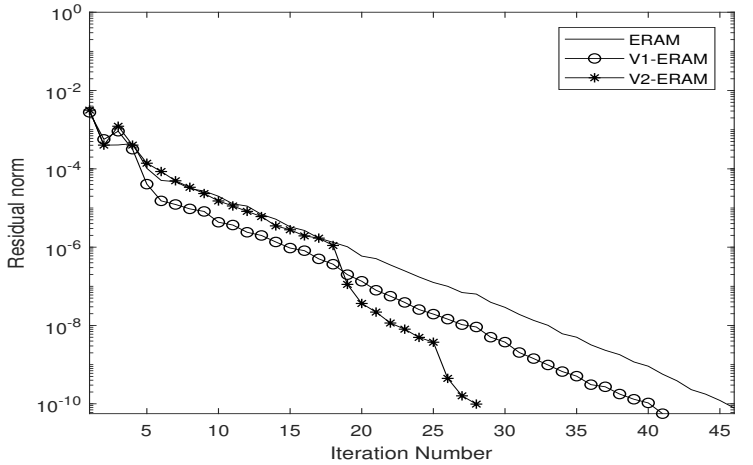

Figure 6: The residual norm for matrix $pde2961$ with $m = 20$ and $k = 3$

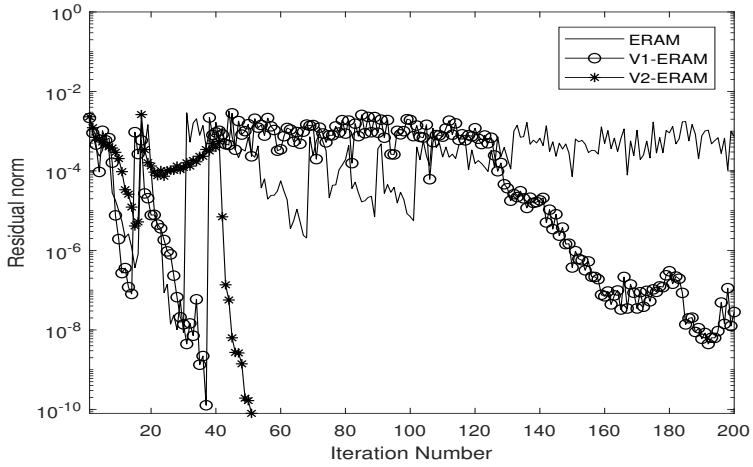

Figure 7: The residual norm for matrix $rdb2048l$ with $m = 20$ and $k = 3$

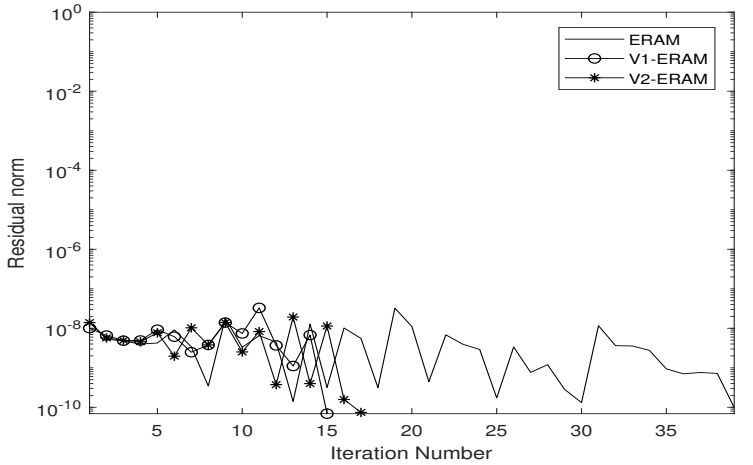

Figure 8: The residual norm for matrix $tols4000$ with $m = 20$ and $k = 1$

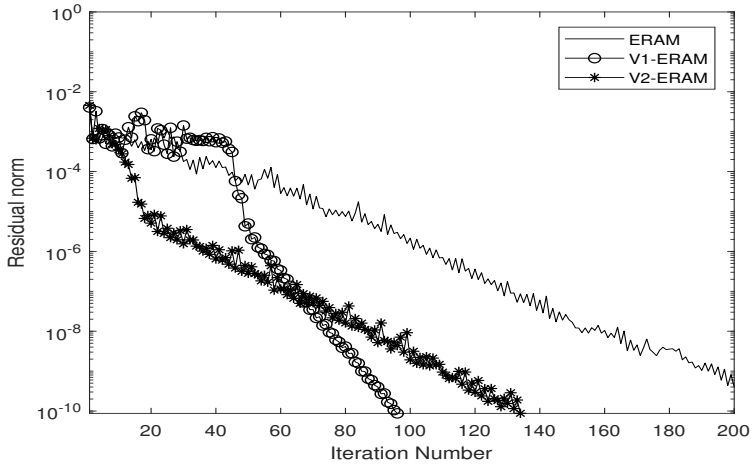

Figure 9: The residual norm for matrix $utm1700a$ with $m = 20$ and $k = 2$

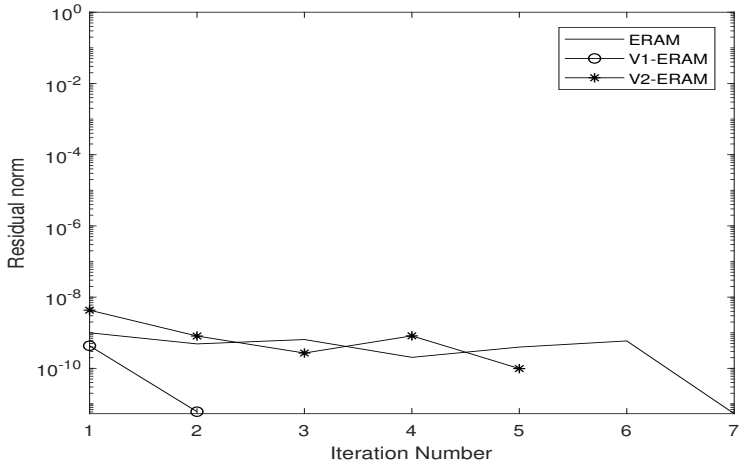

Figure 10: The residual norm for matrix $west2021$ with $m = 20$ and $k = 3$

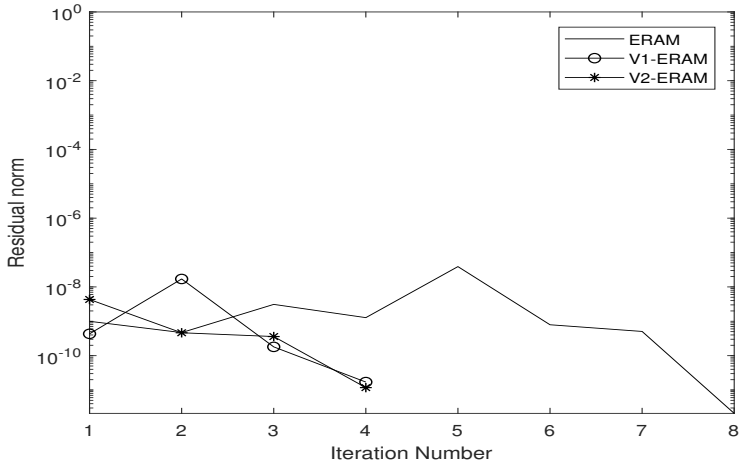

Figure 11: The residual norm for matrix *west*2021 with $m = 20$ and $k = 5$

6 Conclusion

In this paper, we proposed two new strategies for choosing the initial vector in the explicitly restated Arnoldi method for computing eigenvalues of large-scale sparse matrices. Numerical examples indicate that our new methods presented here often perform much better than the original emplicitly restarting Arnoldi method (ERAM). Indeed, we expected the resulting algorithms are more powerful.

References

[1] Abadi, Z., Shahzadeh Fazeli, S.A., Karbassi, S.M., New techniques for accelerating projection methods for computing eigenvalues of large sparse matrices, International Journal of Pure and Applied Mathematics 116(4)(2017) 999-1013.

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

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

[4] Dookhitram, K., Boojhawon, R., Bhuruth, M. A., new method for accelerating Arnoldi Algorithms for large scale eigenproblems, Math. Comput. Simul., 2(2010) 387-401.

[5] Duan, C., Jia, Z. A., global Arnoldi method for large non-Hermitian eigenproblems with special applications to multiple eigen problems, Taiwanese Journal of Mathematics, 15(4)(2011) 1497-1525.

[6] Emad, N., Petiton, S., Edjlali, G., Multiple explicitly restarted Arnoldi method for solving large eigenproblems, SIAM J. Sci. Comput. SJSC., 27(1)(2005) 253-277.

[7] Emad, N., Shahzadeh-Fazeli, S. A., Dongarra, J., An asynchronous algorithm on the Netsolve global computing system, Future Generation Computer Systems, 22(2006) 279-290.

[8] Lehoucq, R. B., Implicitly Restarted Arnoldi Methods and Subspace Iteration, SIAM J. Matrix. Anal. Appl., 23(2001) 551-562.

[9] Saad, Y., Chebyshev acceleration techniques for solving non symmetric eigenvalue problems, Math. Comp., 42(1994) 567-588.

[10] Saad, Y., Numerical Methods for Large Eigenvalue Problems, Manchester University Press (1993).

[11] Saberi Najafi, H., Shams Salary, M., Developing an improved shift-and-invert Arnoldi method, Applications and Applied Mathematics: An International Journal, 5(1)(2010) 167-180.

[12] Shahzadeh Fazeli, S.A., Emad, N., Liu, Z., A key to choose subspace size in implicitly restarted Arnoldi method, Numerical Algorithms, 70(2)(2015) 407426.

[13] Sleijpen, G.L.G., Van der Vorst, H.A., A Jacobi-Davidson iteration method for linear eigenvalue problems, SIAM Rev., 42(2000) 267-293.

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