World Applied Sciences Journal 12 (12): 2337-2342, 2011
ISSN 1818-4952
© IDOSI Publications, 2011

New Iterative Method for Solving Large Sparse Generalized Eigenvalue Problem

S. Pourjafar, S. Kordrostami and A.H. Refahi Sheikhani

Lahijan Branch, Islamic Azad University, Lahijan, Iran

Abstract: The generalized eigenvalue problem $AX = \lambda BX$ has special properties when (A,B) is a symmetric and positive definite pair. In this paper we present a new iterative method for soling large sparse generalized eigenvalue problem based on Krylov subspace process. This method uses three parameters, which are weighted Arnoldi, Inner and outer iterations parameters. In this article those parameters have been optimized to increase the speed of convergence. The method has been tested by some numerical examples.

Key words: Generalized eigenvalue problem . Arnoldi, symmetric . positive definite . weighted FOM

INTRODUCTION

Iterative methods such as the Lanczos algorithm and the inverse vector algorithm are widely used for solving large matrix generalized eigenvalue problems [3-12].

But these methods require solving a linear system of equations of the original size at each iteration of the process and one can use an iterative method such as weighted Arnoldi [2] to solve the systems.

In this paper we present a new method for computing the P-smallest (largest) eigenvalue of pair (A,B). There are three parameters:

- m, which is from the weighted Arnoldi
- The number of inner iterations
- The number of outer iterations.

We show that there is an important relation between these parameters first and then we optimize them.

DEFINITIONS

If $X \in R^n$ and $\|X_2\| = 1$, then the number

$$R(X) = \frac{X^T AX}{X^T BX}$$

is called the generalized Rayleigh quotient. This quotient is well defined if and only the matrix B is positive definite.

Let D be a diagonal matrix, i.e, D = diag $(d_1, d_2, \ldots, d_n)$. If U,V are two vectors of R^n. We define the D-scalar product of $(U \cdot V)_D = V^t DU$. This product is well defined if and only if matrix D is positive definite, i. e, $d_i > 0$, i = 1,2,…,n. The norm associated with this inner product is the D-norm $\|\cdot\|_D$ and define by

$$\|U\|_D = \sqrt{(U.U)_D} = \sqrt{U^T DU} \quad \forall U \in R^n$$

SOLUTION OF LINEAR SYSTEM AX = b BY WEIGHTED FOM METHOD

Azeddin Essai proposed the weighted FOM method [2] to solve the system. In this method a vector $d = (d_1, d_2, \ldots, d_n)^t$ is chosen such that $\|d\|_2 = \sqrt{n}$.

The following algorithm describes the weighted Arnoldi process which uses the D-inner product (.,.) to construct a D-orthonormal basis of the krylov subspace

$$k_m(A, r_0) = \mathrm{Span}\{r_0, A r_0, \ldots, A^{m-1} r_0\}$$

starting with the vector $\tilde{v}_1 = \frac{v}{\|v\|_D}$, where x_0 is an initial guess, $r_0 = b - Ax_0$ and $D = \mathrm{diag}\,(d_1, d_2, \ldots, d_n)$.

Algorithm 1: Arnoldi process

```
For j = 1,…,m
  $w = A\tilde{v}_j$
  For i = 1,…,j
    $\tilde{h}_{ij} = (w, \tilde{v}_i)_D$
    $w = w - \tilde{h}_{ij}\tilde{v}_i$
  End {for}
  $\tilde{h}_{j+1,j} = \|w\|_D$ if $\tilde{h}_{j+1,j} = 0$ stop
  $\tilde{v}_{j+1} = \frac{w}{\tilde{h}_{j+1,j}}$
End {for}
```

Corresponding Author: S. Pourjafar, Lahijan Branch, Islamic Azad University, Lahijan, Iran

Theorem 3.1: The vectors $\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_m$ produced by the weighted Arnoldi algorithm form a D-orthonormal basis of the subspace

$$k_m = \text{span}\{\tilde{v}_1, A\tilde{v}_1, \ldots, A^{m-1}\tilde{v}_1\}.$$

Proof in [2].

Theorem 3.2: Denote by $\tilde{V}_m$ a n×m matrix with the column vectors $\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_m$ and by $\tilde{H}_m$ a m×m Hessenberg matrix whose nonzero entries are defined by the algorithm 1. Then the following relations hold:

$$A\tilde{V}_m = \tilde{V}_m \tilde{H}_m + \tilde{h}_{m+1,m}\tilde{v}_{m+1}e_m^t$$
$$\tilde{V}_m^t D\, \tilde{V}_m \cong \tilde{H}_m$$

Proof in [2].

Like all krylov methods, the m^{th}($m \geq 1$) iterate x_m of the weighted FOM belong to the affine krylov subspace $x_0+k_m(A,r_0)$.

The iterate x_m^{WF} of the weighted-FOM method is chosen such as its residual is D-orthogonal to the subspace $k_m(A,r_0)$. That is

$$r_m^{WF} \perp_D k_m(A, r_0)$$

In this method we use the D-inner product and the D-norm, so, in order to compute the solution in the affine subspace $x_0+k_m(A,r_0)$, we construct a D-orthonormal basis of the krylov subspace $k_m(A,r_0)$ by the weighted Arnoldi process.

An iterate x_m of this methods can be written as $x_m = x_0 + \tilde{V}_m y_m$ where $y_m \in R^m$.

Therefore, the corresponding residual r_m = b-Ax_m satisfies

$$\begin{aligned} r_m &= b - A(x_0 + \tilde{V}_m y_m) \\ &= r_0 - A\tilde{V}_m y_m \\ &= \tilde{V}_{m+1}(\tilde{\beta}e_1 - \bar{\bar{H}}y_m) \end{aligned}$$

Where $\tilde{\beta} = \|r_0\|_D$ and e_1 is the first vector of the canonical basis and $\bar{\bar{H}} = \begin{pmatrix} \tilde{H}_m \\ \tilde{h}_{m+1,m}e_m^T \end{pmatrix}$.

Then, the weighted-FOM method consists in finding the vector y_m^{WF} solution of the problem $\tilde{H}_m y_m^{WF} = \tilde{\beta}e_1$.

Algorithm 2: Weighted FOM process for solving AX = b

1. Start: Choose x_0,m and a tolerance ε, compute r_0 = b-Ax_0.
2. Choose the vector d such that $\|d\|_2 = \sqrt{n}$.
3. Compute $\tilde{\beta} = \|r_0\|_D$ and $\tilde{v}_1 = \frac{r_0}{\tilde{\beta}}$.
4. Construct the D-orthonormal basis $\tilde{V}_m$ by the weighted Arnoldi process, starting with the vector $\tilde{v}_1$.
5. Solve the system $\tilde{H}_m Y_m = \tilde{\beta}e_1$ by the QR factorization of $\tilde{H}_m$ and set $x_m = x_0 + \tilde{V}_m Y_m$, $r_m = b - Ax_m$.
6. If $\|r_m\|_2 \leq \varepsilon$ stop
   Else set $x_0 = x_m$, $r_0 = r_m$ and go to 2.

INVERSE VECTOR ITERATION METHOD

Let A and B are two n×n symmetric positive definite matrices, in this method the basic idea is to minimize Rayleigh quotient and to find the smallest (largest) generalized eigenvalue of pair (A,B).
The algorithm is:

Algorithm 3: Inverse vector iteration process.

```
Input: Initial vector $X_1$ and tolerance $\varepsilon$
$\lambda_1 = \frac{X_1^T A X_1}{X_1^T B X_1}$
for j = 1,2,… do
solve $A\tilde{X}_{j+1} = BX_j$
$\lambda_{j+1} = \frac{\tilde{X}_{j+1}^T A \tilde{X}_{j+1}}{\tilde{X}_{j+1}^T B \tilde{X}_{j+1}} = \frac{\tilde{X}_{j+1}^T B X_j}{\tilde{X}_{j+1}^T B \tilde{X}_{j+1}}$
if $\frac{|\lambda_{j+1} - \lambda_j|}{|\lambda_{j+1}|} < \varepsilon$ stop
else $X_{j+1} = \frac{\tilde{X}_{j+1}}{(\tilde{X}_{j+1}^T B \tilde{X}_{j+1})^{1/2}}$     (4.1)  end{for}
```

Theorem 4.1: Let $\lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$, where λ_i, i = 1,2,…,n are generalized eigenvalues of symmetric positive definite pair (A,B) and $X_1 < X_2, \ldots, X_n$ are corresponding eigenvectors, then for any initial vector X_1 Algorithm 3 converges to the smallest generalized eigenvalue of pair (A,B) and corresponding eigenvector.
Proof in [1]

Computational remarks:

1) The relation (4.1) in Algorithm 3 guarantees that new eigenvectors are also B-orthonormal i.e, $X_{j+1}^T B X_{j+1} = 1$ and this helps the implementation of

the numerical process but doesn't help the convergence.

2) The main step in this method is to solve $A\tilde{X}_{j+1} = BX_j$, so in this work we try to present a method based on krylov subspace for solving the system.

THEOREMS

Theorem 5.1: The symmetric definite generalized eigenvalue problem has real eigenvalues and linearly independent eigenvectors.
Proof in [1]

Theorem 5.2: The generalized eigenvectors of pair (A,B) are B-orthonormal.
Proof in [1]

Theorem 5.3: Let $\lambda_1,\lambda_2,\ldots,\lambda_n$ are n generalized eigenvalues of symmetric positive definite generalized eigenvalue problem, $AX = \lambda BX$ and $X_1<X_2,\ldots,X_n$ are corresponding generalized eigenvectors and let $\lambda_1<\lambda_2\leq\ldots\leq\lambda_n$, then

i) For any vector X we have; $\lambda_1\leq R(X)\leq\lambda_n$
ii) If the vector X is closed to eigenvector X_r then R(X) has constant value.

Proof

i) Let X be an arbitrary vector, then by theorem 5.1 we have: $X = C_1X_1+\ldots+C_nX_n$. Now from theorem 5.2 and noting that λ_i is an eigenvalue of (A,B) and X_i is corresponding eigenvector we can write:

$$R(X) = \frac{X^TAX}{X^TBX} = \frac{C_1^2\lambda_1 + C_2^2\lambda_2 + \cdots + C_n^2\lambda_n}{C_1^2 + C_2^2 + \cdots + C_n^2} = \frac{\lambda_1(C_1^2 + C_2^2\frac{\lambda_2}{\lambda_1} + \cdots + C_n^2\frac{\lambda_n}{\lambda_1})}{C_1^2 + C_2^2 + \cdots + C_n^2} \quad (5.1)$$

Since

$$C_1^2 + C_2^2\frac{\lambda_2}{\lambda_1} + \cdots + C_n^2\frac{\lambda_n}{\lambda_1} \geq C_1^2 + C_2^2 + \cdots + C_n^2$$

then $R(X)\geq\lambda_1$ and similarly we can show $R(X)\leq\lambda_n$

ii) By (5.1) we have

$$R(X) = \frac{C_r^2\lambda_r + C_r^2\sum_{\substack{i=1\\ i\neq r}}^{n}(\frac{C_i}{C_r})^2\lambda_i}{C_r^2 + C_r^2\sum_{\substack{i=1\\ i\neq r}}^{n}(\frac{C_i}{C_r})^2} = \frac{\lambda_r + \sum_{\substack{i=1\\ i\neq r}}^{n}(\frac{C_i}{C_r})^2\lambda_r}{1 + \sum_{\substack{i=1\\ i\neq r}}^{n}(\frac{C_i}{C_r})^2}$$

If X close to X_r then $C_r>>C_i$, $i = 1,2,\ldots,n$, $i\neq r$ so $\left|\frac{C_i}{C_r}\right|$ is a very small number, $(i\neq r)$

therefore

$$1 - \left[\sum_{\substack{i=1\\ i\neq r}}^{n}(\frac{C_i}{C_r})^2\right]^2 \cong 1$$

and we can write

$$R(X) \cong (\lambda_r + \sum_{\substack{i=1\\ i\neq r}}^{n}\varepsilon_i^2\lambda_i)(1 - \sum_{\substack{i=1\\ i\neq r}}^{n}\varepsilon_i^2) \text{ where } \varepsilon_i = \frac{C_i}{C_r} << 1$$

$$R(X) \cong \lambda_r(1 - \sum_{\substack{i=1\\ i\neq r}}^{n}\varepsilon_i^2 + \sum_{\substack{i=1\\ i\neq r}}^{n}\varepsilon_i^2\frac{\lambda_i}{\lambda_r})$$

$$\cong \lambda_r(1 + \sum_{i=1}^{n}(\frac{\lambda_i}{\lambda_r} - 1)\varepsilon_i^2)$$

$$R(X) \cong \lambda_r(1 + O(\varepsilon^2))$$

WEIGHTED FOM-INVERSE VECTOR ITERATION METHOD

Now we are prepared to write the main algorithm of this paper, which is a combination of algorithm 2 and 3, the resulting algorithm is:

Algorithm 4: Weighted FOM-inverse vector iteration method

Input initial vector x_1 whit $\|x_1\| = 1$ and tolerance ε and parameter m.

```
$\lambda_1 = \frac{x_1^TAx_1}{x_1^TBx_1}$
$x_m = x_1$
For k = 1,2,… do
  Compute $r_0 = Bx_k - Ax_m$
  For z = 1,2,… do
    Choose the vector d such as $\|d\|_2 = \sqrt{n}$ and set $D = diag(d)$
    Compute $\tilde{\beta} = \|r_0\|_D$ and $\tilde{v}_1 = \frac{r_0}{\tilde{\beta}}$
    Construct the D-orthonormal basis $\tilde{V}_m$ by the weighted Arnoldi process, starting with $\tilde{v}_1$
    Solve $\tilde{H}_mY_m = \tilde{\beta}e_1$ and set $x_m = x_0 + \tilde{V}_mY_m$, $r_m = Bx_k - Ax_m$
    If $\|r_m\|_2 > \varepsilon$ then $x_0 = x_m$, $r_0 = r_m$
  end {for}
  $\tilde{x}_{k+1} = x_m$
  $\lambda_{k+1} = \frac{\tilde{x}_{k+1}^TA\tilde{x}_{k+1}}{\tilde{x}_{k+1}^TB\tilde{x}_{k+1}}$
```

```
If $\frac{|\lambda_{k+1} - \lambda_k|}{\lambda_{k+1}} < \varepsilon$ then stop
Else $x_{k+1} = \frac{\tilde{x}_{k+1}}{(\tilde{x}_{k+1} B \tilde{x}_{k+1})^{\frac{1}{2}}}$
end {for}
```

As algorithm 4 shows, there are 2 loops in this algorithm, one computes the eigenvector and is called outer iteration, the other solves the system of linear equation at each iteration, which is called inner iteration. Numerical tests show that there is a significant relation between parameter m and inner-outer iterations.

NUMERICAL TEST 1

Let A and B are 1000×1000 matrices as:

$$A = \begin{bmatrix} 3 & 1.2 & 0.42 & 0.8 & 0.3 & 0.8 & & & & & & & & \\ 1.2 & 4 & 1.2 & 0.42 & 0.8 & 0.3 & & & & & & & & \\ 0.42 & 1.2 & 5 & 1.2 & 0.42 & 0.8 & & & & & & & & \\ 0.8 & 0.42 & 1.2 & 6 & 1.2 & 0.42 & & & & & & & & \\ 0.3 & 0.8 & 0.42 & 1.2 & 7 & 1.2 & & & & & & & & \\ 0.8 & 0.3 & 0.8 & 0.42 & 1.2 & 8 & & & & & & & & \\ & 0.8 & 0.3 & 0.8 & 0.42 & 1.2 & & & & & & & & \\ & & 0.8 & 0.3 & 0.8 & 0.8 & & & & & & & & \\ & & & 0.8 & 0.3 & 0.8 & & & & & & 0.8 & & \\ & & & & 0.8 & 0.3 & & & & & & 0.3 & 0.8 & \\ & & & & & 0.8 & & & & & & 0.8 & 0.3 & 0.8 \\ & & & & & & & & & & & 0.42 & 0.8 & 0.3 \\ & & & & & & & & & & & 1.2 & 0.42 & 0.8 \\ & & & & & & 0.8 & 0.3 & 0.8 & 0.42 & 1.2 & 1000 & 1.2 & 0.42 \\ & & & & & & & 0.8 & 0.3 & 0.8 & 0.42 & 1.2 & 1001 & 1.2 \\ & & & & & & & & 0.8 & 0.3 & 0.8 & 0.42 & 1.2 & 1002 \end{bmatrix}_{1000\times 1000}$$

$$B = \mathrm{diag}(2,\ldots,1001)_{1000\times 1000}$$

We apply algorithm 4 to find the smallest eigenvalue with initial vector $\tilde{v}_1$ and the stopping criterion is set as $\varepsilon \leq 10^{-7}$ where $\varepsilon = \frac{|\lambda_{k+1} - \lambda_k|}{\lambda_{k+1}}$.

Number of required inner iterations to achieve a desired accuracy, i.e, 10^{-7}, for m = 2,4,6 are shown in Fig. 1:

We can see that by increasing the value of m the number of outer and inner iterations decrease, therefore the consuming time also decreases. But note that if m is very large then because of loosing orthogonality property. Number of iterations increase.

For example in this example for m>6, the number of outer-iterations and inner iteration are constant, therefore the best m in algorithm 4 for this example is m = 6 (Table 1).

In Fig. 2 the total number of required inner iterations to reach a high accuracy $\varepsilon = 10^{-7}$ for m = 2,4,6 is shown. We can see that when m increases the amount of iterations decreases.

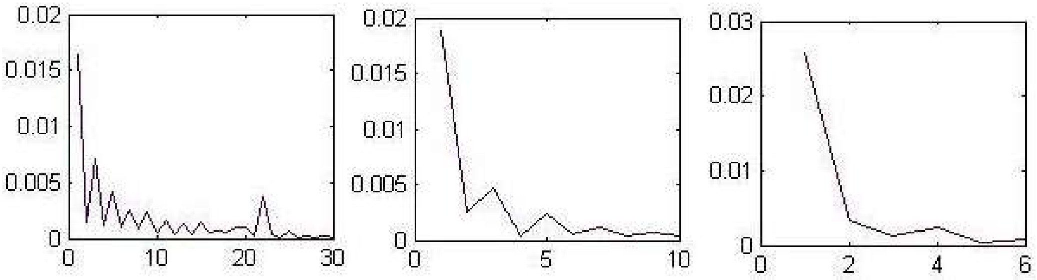

Fig. 1: Showing number of required inner-iterations to achieve a desired accuracy, for m = 2,4,6 (left to right)

Table 1:Shows implementation of Algorithm 4 for pair (A,B) with different value of m and $\varepsilon = 10^{-7}$

m	Inner iteration	Outer iteration	Total inner iteration	Tim
2	30	29	60	53.04
4	10	28	40	18.87
6	6	27	36	12.49
8	6	27	48	24.11

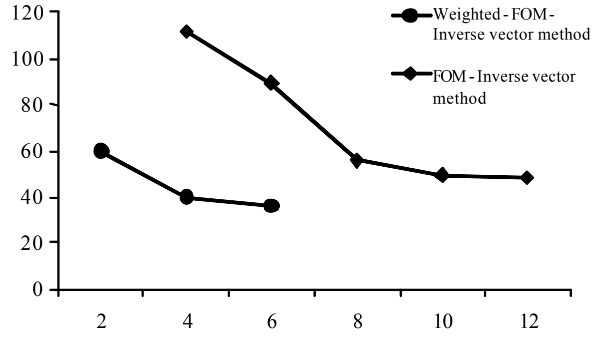

Fig. 2: Comparing the total number of inner iterations needed to reach a desired accuracy for m = 4,6,8,10,12. By FOM inverse vector iteration method [7] and for m = 2,4,6 by W-FOM inverse vector iteration method which can be compared with the W-F-Inverse method

According to Fig. 2, it is noticeable that in the W-F method for m = 6, we reach a to a limited case and the number of total inner iterations which are needed to gain a desirable accuracy (ε=10^{-7}), equals to 36, but in the F method for m = 12 reach to a limited case and the amount of total inner iterations decreases and it equals to 48. Therefore in W-F method the total number of iterations is much less than the F method.

DEFLATION

The algorithm we have described finds the smallest eigenvalue. Once it is done, we can go to find the next smallest eigenvalue by the same procedure through deflation. When P eigenpairs have been found, let V_P be the matrix consisting of the P eigenvectors with $V_P^T B V_P = I$ and E_P be the diagonal matrix consisting of the corresponding eigenvalues. $AV_P = BV_P E_P$. Then we consider

$$(A_P, B) \equiv (A + (BV_P)\Sigma(BV_P)^T, B) \qquad (8.1)$$

where Σ = diag $\{\delta_i - \lambda_i\}$ with δ_i any value chosen to greater than λ_{P+2}. Then it is easy to check that the eigenvalues of (8.1) are the union of $\{\lambda_{P+1}, \lambda_{P+2}, \ldots, \lambda_n\}$ and $\{\delta_1, \ldots, \delta_p\}$. Thus, it's smallest eigenvalue is λ_{P+1} and by applying our method to (8.1), we find λ_{P+1}.

Table 2:Shows implementation of Algorithm4 and deflation for computing 4 smallest eigenvalues of pair (A,B) by $\varepsilon = 10^{-7}$

P	m	Inner -iteration	Outer -iteration	Tim	λ_P
1	6	6	27	12.49	0.58215
2	6	6	65	46.09	0.82667
3	6	6	113	80.57	0.89151
4	6	6	177	125.67	0.92116

Numerical test 2: Let A and B are 1000×1000 matrix used in numerical test 1.

By applying algorithm 4 and deflation process we find 4 smallest generalized eigenvalue of pair (A,B).

The reason why we choose m = 6 is to find the smallest eigenvalue. It was the most appropriate m, therefore we used this m to find the rest of the eigenvalues. As we can see in Table 2 by increasing P, the amount of the outer-iterations needed to reach an appropriate accuracy increases and this is predictable because in the first step we use sparse matrix but in the other steps after deflation the other matrices A,B are not sparse (Fig. 3).

We have also computed four of the smallest eigenvalues of pair (A,B) with the FOM-inverse vector method which can be compared with the weighted-Fom-inverse iberation method (Table 3).

The comparison of the two methods in calculating four of the smallest eigenvalues of pair (A,B) with an accuracy of 10^{-7}. It is noticed that in the weighted-FOM-inverse method the amount of the inner and outer iterations has decreased a lot, therefore the time needed for computations has also decreased.

COMMENTS AND CONCLUSIONS

1) As we know there are some methods for computing the rest of eigenvalues, for example, shift method but the advantage of this kind of deflation compared to the shift method is that for any $\delta_i > \lambda_{P+2}$ (i = 1,2,..,P) the method converges to $(P+1)^{th}$ eigenvalue but in the shift method if we don't use an appropriate shift value it might converge to another eigenvalue.
2) To compute the largest eigenvalue of (A,B), we just need to compute the smallest eigenvalue of (-A,B) and change the sign to obtain the largest eigenvalue of (A,B).
3) The vector d will be chosen as $d_i = \sqrt{n}\frac{|(r_0)_i|}{\|r_0\|_2}$. Such a choice favors the components of the residual which are far away from zero.

Table 3: Shows the comparison between FOM-Inverse and Weighted-FOM- Inverse methods

FOM-inverse vector iteration method				Weighted-FOM-inverse vector iteration method			
P	Total inner iteration	Outer iteration	Tim	P	Total inner iteration	Outer iteration	Tim
1	48	45	8.06	1	36	27	12.49
2	48	149	72.67	2	36	65	46.09
3	48	289	141.18	3	36	113	80.57
4	48	374	178.114	4	36	177	125.67
	192	857	400.02		144	382	264.82

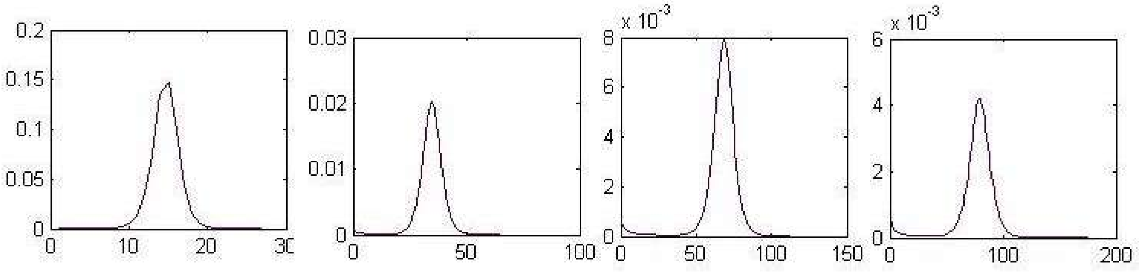

Fig 3: Showing number required outer-iterations for compute 4 smallest generalized eigenvalue of pair (A,B) with accuracy 10^{-7}

4) This algorithm has the advantage of letting us choose an appropriate m, so as to reach the necessary accuracy, meaning the smallest inner-outer iterations needed.

REFERENCES

1. Biswa Nath Datta, 1994. Numerical Linear Algebra and Applications. Brooks, New York.
2. Essai, A., 1998. Weighted Fom and GMRES for solving nonsymmetric linear systems, Numer. Algorithm, 18: 277-299.
3. Saad, Y., 1981. Krylov subspace methods for solving large unsymmetric linear systems. Math. Compute. 37: 105-126.
4. Saberi Najafi, H. and E. Khaleghi, 2004. A new restarting method in the Arnoldi algorithm for computing the eigenvalues of a nonsymmetric matrix. Appl, Math., 156: 59-57.
5. Saberi Najafi, H. and H. Ghazvini, 2006. Weighted restarting method in the weighted Arnoldi algorithm for computing the eigenvalues of a nonsymmetric matrix. Appl, Math., 175: 1279-1287.
6. Saberi Najafi, H. and A. Refahi, 2007. A new restarting method in the Lanczos algorithm for generalized eigenvalue problem. Appl, Math, 184: 421-428.
7. Saberi Najafi, H. and A. Refahi, Fom-inverse vector iteration method for computing a few smallest (largest) eigenvalues of pair (A,B), Appl, Math, In Press.
8. Saad, Y., 1992. Numerical Solution of Large Nonsymmetric Eigenvalue Problems. Halsted Press, New York.
9. Saad, Y., 1996. Iterative method for Sparse Linear system, PWS publishing company, a division of International Thomson Publishing Inc, USA.
10. Sorenson, D.C., 1992. Implicit application of polynomial filters in a K-step Arnoldi method. Siam J. Matrix Appl., 13: 357-385
11. Morgen, B., 1996. Restarting the Amoldi method for large nonsymmetric eigenvalue problems. Math. Comput., pp: 1213-1230.
12. Stathopoulos, A., Y. Saad and K. Wu, 1998. Dynamic thick restarting of the Daridson and the implicitly restarted Arnoldi Algorithm. SIAM J. Sci. Comput., 19: 227-245.