ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 4, No. 3, 2009, *pp*. 183-190

Weighted Harmonic Projection Method for Computing the Eigenvalues of Nonsymmetric Matrix

Hossein Aminikhah [1, +] and Hossein Moosaei [2]

[1] Department of Mathematics, School of Mathematical Sciences, Shahrood University of Technology, P. O. Box 316, P.C. 3619995161, Shahrood, Iran

[2] Department of Mathematics, Faculty of Sciences, University of Guilan, P.O. Box 1914, P.C. 41938, Rasht, Iran

(*Received March 15, 2009, accepted May 20, 2009*)

Abstract. The harmonic projection method can be used to find interior eigenpairs of large matrices. Given a target point or shift τ to which the needed interior eigenvalues are close, the desired interior eigenpairs are the eigenvalues nearest τ and the associated eigenvectors. In this paper, we present a new algorithm, which is called weighted harmonic projection algorithm for computing the eigenvalues of a nonsymmetric matrix. The implementation of the algorithm has been tested by numerical examples, the results show that the algorithm converges fast and works with high accuracy

Keywords: Harmonic projection method, Nonsymmetric matrix, Weighted Arnoldi, Weighted harmonic.

1. Introduction

Finding eigenvalues is an important task in scientific computation. There are many applications in physics, chemistry, and engineering. These include computing energy levels of atoms, finding vibrational states of molecules, and determining how buildings will vibrate during earthquakes.

Also one of the most important and practical topics in computational mathematics is computing some of the interior eigenvalues close to target point or shift τ and the associated eigenvectors. The harmonic projection method has been accepted to be one of commonly used method for computing interior eigenpairs.

However, it has been shown that the harmonic projection method may converge erratically and even may fail to do so.

In this paper, we present a new algorithm for the harmonic projection algorithm, which is called weighted harmonic projection algorithm for computing some of the interior eigenvalues close to τ of the eigenvalue problem, $AX_i = \lambda_i X_i$, where A is an $n \times n$ matrix ,and (λ_i, X_i) is referred to as an eigenpair of A with $\|X_i\| = 1$, here the norm used is the Euclidean norm. The implementation of the algorithm has been tested by numerical examples, the results show that the algorithm converges fast and works with high accuracy.

The paper is organized as follows. Section 2 describes Arnoldi method and some properties of it. Section 3 describes weighted Arnoldi method and some properties of it. Section 4 describes harmonic projection method for computing some of the interior eigenvalues close to target point or shift τ and the associated eigenvectors. Section 5 describes a new algorithm which is call weighted harmonic projection method. Section 6 reports numerical results on three real world problems. Concluding remarks are given in section 7.

2. Arnoldi Method

This method was developed by Arnoldi [1, 8, 9], in 1951. It work on the Krylov subspace:

$$k_m(A, v) = span\{v, Av, \ldots, A^{m-1}v\}$$

[+] Corresponding author. Tel, Fax: +98-273-3332012.
E-mail address: hossein.aminikhah@gmail.com (H. Aminikhah), hmoosaei@gmail.com(H. Moosaei).

where A is a $n \times n$ matrix and $v \in R^n$ is an arbitrary vector. The method starts with an initial vector and after m-step generates a $n \times m$ matrix V_m such that ${V_m}^H A V_m \cong H_m$.

Therefore each eigenvalue of H_m is an approximation of the corresponding one in A.

The Arnoldi algorithm is based on the Gram-Schmidt orthogonalization procedure, since modified Gram-Schmidt algorithm (MGS) has a good numerical stability. We use MGS in our method. This algorithm is known as the Arnoldi modified Gram-Schmidt algorithm.

The algorithm is:

2.1. Algorithm: ***(Arnoldi MGS process)***

Choose a vector v_1 of norm 1

For $j = 1,\ldots,m$ do

$w := Av_j$

For $i = 1,\ldots,j$ do

$h_{ij} := (w, v_i)$

$w := w - h_{i,j}v_i$

End do

$h_{j+1,j} := \|w\|_2$

$v_{j+1} := \dfrac{w}{h_{j+1,j}}$

End do.

Theorem 2.2 The vectors $v_1, v_2,\ldots, v_m$ produced by the Arnoldi algorithm form an orthonormal basis of the subspace $k_m = span\{v_1, Av_1,\ldots, A^{m-1}v_1\}$.

Proof in [8].

Theorem 2.3 Denote by V_m a $n \times m$ matrix with column vectors $v_1, v_2,\ldots, v_m$ and by H_m a $m \times m$ Hessenberg matrix whose nonzero entries are defined by the algorithm. Then the following relations hold:

$$A V_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^H,$$

$${V_m}^H A V_m \cong H_m.$$

Proof in [8].

Theorem 2.4 Let ${y_i}^{(m)}$ be an eigenvector of H_m associated with the eigenvalue ${\lambda_i}^{(m)}$ and ${u_i}^{(m)}$ the Ritze approximate eigenvector ${u_i}^{(m)} = V_m {y_i}^{(m)}$. Then

$$(A - \tilde{\lambda}_i^{(m)} I){u_i}^{(m)} = h_{m+1,m} {e_m}^{(H)} {y_i}^{(m)} v_{m+1}$$

$$\left\|(A - \tilde{\lambda}_i^{(m)} I){u_i}^{(m)}\right\|_2 = h_{m+1,m} \left|{e_m}^{(H)} {y_i}^{(m)}\right|$$

Proof in [9].

3. Weighted Arnoldi Method

This algorithm has been developed by [2]. Let $D = diag(d_1, d_2, ..., d_n)$ be a diagonal matrix with $d_i \succ 0, i = 1, 2, ..., n$. If u, v are two vectors of $\mathbb{R}^n$. We define the D-scalar product as

$$(u, v)_D = v^T D u = \sum_{i=1}^{n} d_i u_i v_i$$

The D-norm $\|.\|_D$ associated with this inner product is

$$|| u ||_D = \sqrt{(u, u)_D}, \forall u \in \mathbb{R}^n$$

Now the weighed Arnoldi process, which constructed a D-orthonormal basis of the Krylov subspace

$$k_m(A, v) = span\{v, Av, \ldots, A^{m-1}v\}$$

can be defined as follows [2].

3.1. Algorithm: ***(Weighted Arnoldi process)***
Choose a vector v_1 such that $\|v_1\|_D = 1$
For $j = 1, \ldots, m$ do
$w := Av_j$
For $i = 1, \ldots, j$ do
$h_{ij} := (w, v_i)_D$
$w := w - h_{i,j}v_i$
End do
$h_{j+1,j} := \|w\|_D$
$v_{j+1} := \dfrac{w}{h_{j+1,j}}$
End do.

Let $V_m = \{v_1, v_2, \ldots, v_m\}$,then we have:

$$V_m{}^T D V_m = \mathrm{I}_m$$

Theorem 3.2 Denote by V_m a $n \times m$ matrix with column vectors $v_1, v_2, \ldots, v_m$ and by H_m a $m \times m$ Hessenberg matrix whose nonzero entries are defined by the algorithm. Then the following relations hold:

$$A V_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^H,$$

$$V_m{}^H D A V_m \cong H_m.$$

Proof: similar to above theorem (2.3)

Theorem 3.3 Let $y_i^{(m)}$ be an eigenvector of H_m associated with the eigenvalue $\lambda_i^{(m)}$ and $u_i^{(m)}$ the Ritze approximate eigenvector $u_i^{(m)} = V_m y_i^{(m)}$. Then

$$(A - \tilde{\lambda}_i^{(m)} I) u_i^{(m)} = h_{m+1,m} e_m{}^{(H)} y_i^{(m)} v_{m+1}$$

$$\left\| (A - \tilde{\lambda}_i^{(m)} I) u_i^{(m)} \right\|_D = h_{m+1,m} \left| e_m{}^{(H)} y_i^{(m)} \right|$$

Proof: Similar to above theorem (2.4).

4. Harmonic Projection Method

If τ is not an eigenvalue of A then we have from $AX_i = \lambda_i X_i$ that

$$(A - \tau I)^{-1} X_i = \frac{1}{\lambda_i - \tau} X_i$$

Therefore, the interior eigenvalues near τ are transformed into exterior ones with largest magnitudes of $(A - \tau I)^{-1}$.

For the given τ and a subspace $k_m(A, v)$ the harmonic projection method seeks the pairs $(\tilde{\lambda}_i, \tilde{X}_i)$ satisfying the harmonic projection (for more details see [5, 7]),

$$\tilde{X}_i \in k_m(A, v_1) \tag{1}$$

$$A\tilde{X}_i - \tilde{\lambda}_i \tilde{X}_i \perp (A - \tau I) k_m(A, v_1) \tag{2}$$

and uses them to approximate some eigenvalues of A near τ and the associated eigenvectors.

From (2) we have

$$((A-\tau I)V_m)^H(A\tilde{X}_i-\tilde{\lambda}_i\tilde{X}_i)=0$$

Where v_m is an orthonormal basis of $k_m(A,v_1)$ then

$$((A-\tau I)V_m)^H((A-\tau I)\tilde{X}_i-(\tilde{\lambda}_i-\tau)\tilde{X}_i)=0 \tag{3}$$

Since $\tilde{X}_i \in k_m(A,v_1)$ and v_m is an orthonormal basis of $k_m(A,v_1)$ then we can write

$$\tilde{X}_i = V_m g_i \tag{4}$$

Substituting (4) in (3) gives

$$((A-\tau I)V_m)^H((A-\tau I)V_m g_i-(\tilde{\lambda}_i-\tau)V_m g_i)=0 \tag{5}$$

or

$$((A-\tau I)V_m)^H(A-\tau I)V_m g_i=(\tilde{\lambda}_i-\tau){V_m}^H(A-\tau I)^H V_m g_i \tag{6}$$

And by theorem (2.4)

$$(A-\tau I)V_m = V_m(H_m-\tau I)+h_{m+1,m}v_{m+1}e_m^H \tag{7}$$

or

$${V_m}^H(A-\tau I)^H = (H_m-\tau I)^H{V_m}^H + e_m v_{m+1}^H h_{m+1,m}^H \tag{8}$$

Substituting (8) in (5) gives

$$\begin{aligned}&((H_m-\tau I)^H{V_m}^H + e_m v_{m+1}^H h_{m+1,m}^H)(V_m(H_m-\tau I)+h_{m+1,m}v_{m+1}e_m^H)g_i=\\&(\tilde{\lambda}_i-\tau)((H_m-\tau I)^H{V_m}^H + e_m v_{m+1}^H h_{m+1,m}^H)V_m g_i\end{aligned}$$

or

$$\begin{aligned}&((H_m-\tau I)^H(H_m-\tau I)+e_m h_{m+1,m}^H h_{m+1,m}e_m^H)g_i=\\&(\tilde{\lambda}_i-\tau)(H_m-\tau I)^H g_i\end{aligned} \tag{9}$$

If $H_m-\tau I$ is nonsingular then

$$((H_m-\tau I)+(H_m-\tau I)^{-H}e_m h_{m+1,m}^H h_{m+1,m}e_m^H)g_i=(\tilde{\lambda}_i-\tau)g_i \tag{10}$$

Therefore it is sufficient to compute the eigenvalues of (10). Following the above discussion, we present a harmonic projection iteration method.

4.1 Algorithm. ***(Harmonic projection algorithm)***

1. Input, τ, v_1 with $\|v_1\|=1$, l: the numbers of desired eigenvalues.

For $S=0,1,\ldots$ do

2. Run algorithm2.1 (For computing $V_{m+1}=[v_1,v_2,...,v_{m+1}]$ and H_m.)

3. If $H_m-\tau I$ is nonsingular then

Solving $((H_m-\tau I)+(H_m-\tau I)^{-H}e_m h_{m+1,m}^H h_{m+1,m}e_m^H)g_i=(\tilde{\lambda}_i-\tau)g_i$

Else solving $((H_m-\tau I)^H(H_m-\tau I)+e_m h_{m+1,m}^H h_{m+1,m}e_m^H)g_i=(\tilde{\lambda}_i-\tau)(H_m-\tau I)^H g_i$

end if; (For computing $(\tilde{\lambda}_i,g_i)$, i=1,...,m.)

4. Select $\tilde{\lambda}_i$s with respect to the smallest value of $(\tilde{\lambda}_i-\tau)$, s is to approximate the desired eigenvalues i=1,...,l.

5. Take the harmonic Ritz pairs $(\tilde{\lambda}_i,\tilde{X}_i=V_m g_i)$, i=1,...,l as approximations.

6. Set $v^{(s+1)}=\sum_{i=1}^{l}X_i^{(s)}$ and normalized.end for.

Theorem 4.2: Let $\tilde{X}_i^{(m)}=V_m{g_i}^{(m)}$ be the harmonic Ritz vector, $\tilde{\lambda}_i$ be the harmonic Ritz value then:

$$(A-\tilde{\lambda}_i^{(m)}I)X_i^{(m)} = V_{m+1}\begin{bmatrix}(H_m-\tilde{\lambda}_i^{(m)}I)g_i^{(m)}\\ h_{m+1,m}e_m^H g_i^{(m)}\end{bmatrix}$$

Proof:

Since we have $AV_m = V_m H_m + h_{m+1,m}v_{m+1}e_m^H$ then

$$AX_i^{(m)} = AV_m g_i^{(m)} = V_m H_m g_i^{(m)} + h_{m+1,m}v_{m+1}e_m^H g_i^{(m)}$$

So

$$\begin{aligned}(A-\tilde{\lambda}_i^{(m)}I)X_i^{(m)} &= AV_m g_i^{(m)} - \tilde{\lambda}_i^{(m)}V_m g_i^{(m)}\\ &= V_m H_m g_i^{(m)} + h_{m+1,m}v_{m+1}e_m^H g_i^{(m)} - \tilde{\lambda}_i^{(m)}V_m g_i^{(m)}\\ &= V_m(H_m-\tilde{\lambda}_i^{(m)}I)g_i^{(m)} + h_{m+1,m}v_{m+1}e_m^H g_i^{(m)} = V_{m+1}\begin{bmatrix}(H_m-\tilde{\lambda}_i^{(m)}I)g_i^{(m)}\\ h_{m+1,m}e_m^H g_i^{(m)}\end{bmatrix}\end{aligned}$$

5. Weighted Harmonic Projection Method

In this section we construct a new algorithm for modified the above algorithm and it is called weighted harmonic projection algorithm.

In this algorithm we will use as D-norm substituted 2-norm.

5.1 Algorithm. ***(weighted Harmonic projection algorithm)***

1. Input, τ, v_1 with $\|v_1\|_D = 1$, l: the numbers of desired eigenvalues.

For $S = 0,1,\ldots$ do

2. Run algorithm3.1 (For computing $V_{m+1} = [v_1, v_2, ..., v_{m+1}]$ and H_m.)

3. If $H_m - \tau I$ is nonsingular then

Solving $((H_m-\tau I)+(H_m-\tau I)^{-H}e_m h_{m+1,m}^H h_{m+1,m}e_m^H)g_i = (\tilde{\lambda}_i-\tau)g_i$

Else solving $((H_m-\tau I)^H(H_m-\tau I)+e_m h_{m+1,m}^H h_{m+1,m}e_m^H)g_i = (\tilde{\lambda}_i-\tau)(H_m-\tau I)^H g_i$

End if; (For computing $(\tilde{\lambda}_i, g_i)$, i=1,...,m .)

4. Select $\tilde{\lambda}_i$ s with respect to the smallest value of $(\tilde{\lambda}_i-\tau)$' s is to approximate the desired eigenvalues i=1,...,l .

5. Take the harmonic Ritz pairs $(\tilde{\lambda}_i, \tilde{X}_i = V_m g_i)$, i=1,...,l as approximations.

6. Set $v^{(s+1)} = \sum_{i=1}^{l} X_i^{(s)}$ and D-normalized.endfor

Theorem 5.2: Let $\tilde{X}_i^{(m)} = V_m g_i^{(m)}$ be the harmonic Ritz vector, $\tilde{\lambda}_i$ be the harmonic Ritz value then:

$$\left\|(A-\tilde{\lambda}_i^{(m)}I)X_i^{(m)}\right\|_D^2 = \left\|V_m(H_m-\tilde{\lambda}_i^{(m)}I)g_i^{(m)}\right\|_D + h_{m+1,m}{}^2\left\|v_{m+1}e_m^H g_i^{(m)}\right\|_D$$

Proof:

Since we have $AV_m = V_m H_m + h_{m+1,m}v_{m+1}e_m^H$ then

$$AX_i^{(m)} = AV_m g_i^{(m)} = V_m H_m g_i^{(m)} + h_{m+1,m}v_{m+1}e_m^H g_i^{(m)}$$

So

$$
\begin{aligned}
(A-\tilde{\lambda}_i^{(m)}I)X_i^{(m)} &= A\,V_m g_i^{(m)} - \tilde{\lambda}_i^{(m)} V_m g_i^{(m)} \\
&= V_m H_m g_i^{(m)} + h_{m+1,m} v_{m+1} e_m^H g_i^{(m)} - \tilde{\lambda}_i^{(m)} V_m g_i^{(m)} \\
&= V_m (H_m - \tilde{\lambda}_i^{(m)} I) g_i^{(m)} + h_{m+1,m} v_{m+1} e_m^H g_i^{(m)}
\end{aligned}
$$

Then

$$
\begin{aligned}
&\left\| (A-\tilde{\lambda}_i^{(m)}I)X_i^{(m)} \right\|_D^2 \\
&= \left\| V_m (H_m - \tilde{\lambda}_i^{(m)} I) g_i^{(m)} + h_{m+1,m} v_{m+1} e_m^H g_i^{(m)} \right\|_D^2 \\
&= (V_m (H_m - \tilde{\lambda}_i^{(m)} I) g_i^{(m)} + h_{m+1,m} v_{m+1} e_m^H g_i^{(m)})^H D (V_m (H_m - \tilde{\lambda}_i^{(m)} I) g_i^{(m)} + h_{m+1,m} v_{m+1} e_m^H g_i^{(m)}) \\
&= (V_m (H_m - \tilde{\lambda}_i^{(m)} I) g_i^{(m)})^H D (V_m (H_m - \tilde{\lambda}_i^{(m)} I) g_i^{(m)}) + (h_{m+1,m} v_{m+1} e_m^H g_i^{(m)})^H D (h_{m+1,m} v_{m+1} e_m^H g_i^{(m)}) \\
&= \left\| V_m (H_m - \tilde{\lambda}_i^{(m)} I) g_i^{(m)} \right\|_D + h_{m+1,m}{}^2 \left\| v_{m+1} e_m^H g_i^{(m)} \right\|_D .
\end{aligned}
$$

6. Numerical Results

In this section we report some numerical experiments on tree problems. The algorithm has been tested using MATLAB 7.0.4on a Pentium IV CPU 3.06GHz with main memory 1 gigabyte. And machine precision $eps \approx 2.22 \times 10^{-16}$. T1,T2 denotes times in seconds for algorithms 4.1 and 5.1 and m denote step of Arnoli process and Iter1 and Iter2 are numbers restarted for algorithms 4.1 and 5.1 and R1 and R2 are accuracy for algorithms 4.1 and 5.1 and τ is arbitrary value.

6.1 Example: Matrix A is a banded matrix select from [3], which use in Quantum Chemistry, i.e.

$$
A = \begin{bmatrix}
1 & 0.21 & 1.2 & 0 & 0.13 & 1.42 & & & & \\
0.11 & 2 & 0.21 & 1.2 & 0 & 0.13 & 1.42 & & & \\
0.12 & 0.11 & 3 & 0.21 & 1.2 & 0 & \ddots & \ddots & & \\
0 & 0.12 & 0.11 & 4 & \ddots & \ddots & \ddots & \ddots & \ddots & \\
0.34 & 0 & 0.12 & \ddots & \ddots & \ddots & \ddots & \ddots & 0.13 & 1.42 \\
0.45 & 0.34 & 0 & \ddots & \ddots & \ddots & 0.21 & 1.2 & 0 & 0.13 \\
 & 0.45 & \ddots & \ddots & \ddots & 0.11 & 197 & 0.21 & 1.2 & 0 \\
 & & \ddots & \ddots & \ddots & 0.12 & 0.11 & 198 & 0.21 & 1.2 \\
 & & & \ddots & 0.34 & 0 & 0.12 & 0.11 & 199 & 0.21 \\
 & & & & 0.45 & 0.34 & 0 & 0.12 & 0.11 & 200
\end{bmatrix}_{200\times 200}
$$

The result has shown that in Table1.

6.2 Example: This example attempts to illustrate the efficiency of the adaptive shifts strategy. The matrix is due to Morgan [4]. It is a tridiagonal matrix with 1, 2, 2.05, 2, 1, 3, 4, 5,…, 998 on the main diagonal, -0.1 in each superdiagonal position and 0.1 in each subdiagonal position.
The result has shown that in Table2.

6.3 Example: Consider the constant coefficient convention-diffusion equation

$$-\Delta u(x,y) + p_1 u_x(x,y) + p_2 u_y(x,y) - p_3 u(x,y) = \lambda u(x,y)$$

On a square region $[0,1]\times[0,1]$ with the boundary condition $u(x,y)=0$ where p_1, p_2 and p_3 are positive constant. Discretization by five differences on uniform $n \times n$ grid points using the row wise natural ordering gives a block tridiagonal matrix of the form

$$A=\begin{bmatrix} T & (\beta+1)I & & & & \\ (-\beta+1)I & T & (\beta+1)I & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & & \ddots & (\beta+1)I \\ & & & \ddots & (-\beta+1)I & T \end{bmatrix}$$

Where

$$T=\begin{bmatrix} 4-\theta & \gamma-1 & & & & \\ -\gamma-1 & 4-\theta & \gamma-1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & & \ddots & \\ & & & & \ddots & \gamma-1 \\ & & & \ddots & -\gamma-1 & 4-\theta \end{bmatrix}$$

Where $\beta=(1/2)p_1h, \gamma=(1/2)p_2h, \theta=p_3h^2$ and $h=1/(n+1)$. The order of A is $N=n^2$. By taking $p_1=1, p_2=p_3=0$ and $n=6$, we can obtain a 36×36 matrix A (36).

The result has shown that in Table3.

Table1 for example 6.1

	Iter1	Iter2	R1	R2	T1	T2
$m=5,\tau=0$	7099	278	15.623143	.00004444	92.22909	9.03890
$m=10,\tau=2$	2187	19	1.3615790	.00007155	40.24374	6.72804
$m=15,\tau=0$	530	45	0.8251351	.00009459	16.76781	2.83183
$m=20,\tau=0$	230	6	0.8293846	.00003977	11.95818	2.32617

Table2 for example 6.2

	Iter1	Iter2	R1	R2	T1	T2
$m=6,\tau=0$	1194	972	16.32030	.00000823	138.6159	65.8151
$m=15,\tau=2$	1701	13	2.467152	.00000684	236.6630	4.5171
$m=25,\tau=0$	471	27	3.703124	.00000829	88.0428	8.8303
$m=35,\tau=6$	313	216	6.064665	.00001026	87.3226	72.6157

Table3 for example 6.3

	Iter1	Iter2	R1	R2	T1	T2
$m=6,\tau=0$	5162	666	0.559396	.00008479	66.09368	13.6818
$m=15,\tau=1$	540	145	0.006053	.00009004	17.25598	3.60487
$m=25,\tau=3$	1075	117	0.004467	.00007116	28.25612	3.42415
$m=10,\tau=1$	2398	297	0.085080	.00009788	41.16828	10.9874

From these tables we see that:

- For the number of iterations: Weighted harmonic projection method is always better than harmonic projection method, respectively.
- For the computational time: Weighted harmonic projection method is better than harmonic projection method.
- For the computational Residual: Weighted harmonic projection method is better than harmonic

projection method and the convergence is better than that.

- Therefore the results show that weighted harmonic projection algorithm works better than harmonic projection algorithm and it gives the results with high accuracy.

7. Conclusion

1. The first method (algorithm 4.1) does not have a good convergence, as was shown before; this method gives a little improvement to the approximated eigenvalues.
2. The weighted harmonic projection algorithm (algorithm 5.1) has the following properties:
   a) This algorithm is very simple to run.
   b) It can be run on any PC.
   c) The eigenvalue problem can be computed with a high accuracy and less consuming time.
   d) The approximation obtained by this method can be improved more easily compared to the other method.

8. References

[1] W.E. Arnoldi. The principle of minimized iterations in solution of the matrix eigenvalue problem. *Quart.Appl. Math.*. 1951, **9**: 17-29.

[2] A. Essai. Weighted FOM and GMRES for solving nonsymmetric linear systems. *Numer. Algorithm*. 1998, **18**: 277-292

[3] S.H. Lam. The role of CSP in reduce chemistry modeling. *in: IMA Symposium.* Minnesota University, 1999.

[4] R.B. Morgan. On restarting the Arnoldi method for large nonsymmetric eigenvalue problems. *Math. Comput.*. 1996, **65**: 1213-1230

[5] R.B. Morgan and M. Zeng. Harmonic projection method for large non-symmetric eigenvalue problem. *Number Linear Algebra.* 1998, **5**: 33-35.

[6] H. Saberi Najafi, H.Ghazvini. Weighted restarting method in the weighted Arnoldi algorithm for computing the eigenvalues of a nonsymmetric matrix. *Appl. Math. Comput.*. 2006, **175**: 1279-1287.

[7] H. Saberi Najafi, H. Moosaei. A new restarting method in the harmonic projection algorithm for computing the eigenvalues of a nonsymmetric matrix. *Appl. Math. Comput.* . 2008, **198**: 143-149.

[8] Y. Saad. *Iterative method for sparse linear system.* USA: PWS Publishing company, a division of international Thomson publishing Inc., 1996.

[9] Y. Saad. *Numerical method for large eigenvalue problem*. Manchester University press in Algorithm and architectures for advanced scientific computing , 1992.