Dutse Journal of Pure and Applied Sciences (DUJOPAS) Vol. 4 No. 1 June 2018

ANALYTICAL COMPARISON OF ARNOLDI METHODS FOR THE SOLUTION OF QUADRATIC EIGENVALUE PROBLEMS

Usman Sanusi Department of Mathematics and Computer Science, Umaru Musa Yar'adua University, Katsina

Hamisu Musa Department of Mathematics and Computer Science, Umaru Musa Yar'adua University, Katsina

Abstract

Quadratic Eigenvalue Problems arise in many areas of science and engineering. These applications areas include earthquake engineering, quantum chemistry, chemical reactions, control theory and many others. In this paper, we considered Arnoldi, Second-Order Arnoldi Method and Quadratic Arnoldi Method for extracting a few eigenpairs of the QEP. Numerical experiments are performed to examine the behaviour of each method. In so doing, we considered shift and invert QEP. Relative residual norms of the approximate eigenpairs yield by each method are as well computed. Numerical examples has shown that the SOAR method is the best in terms of convergence behavior, while it has been found that Q-Arnoldi is better in convergence than Arnoldi. We have also found that Q-Arnoldi loses stability when $\|\hat{H}_k\|$ is large but the RQ-Arnoldi developed in this paper does not.

Keywords: Quadratic Eigenvalue Problem, Arnoldi Method, SOAR Method, Q-Arnoldi Method, Shift and Invert QEP.

INTRODUCTION

Consider the λ -matrix of degree k

$$Q(\lambda) = \lambda^{k}M_k + \lambda^{k-1}M_{k-1} + \cdots + M_0 \quad (1)$$

with $M_i \in C^{n \times n}$; $i = 0 : k$. In polynomial eigenvalue problem (PEP), we seek scalar values λ and nonzero vectors x and y satisfying

$$Q(\lambda)x = 0,$$

and

$$y^* Q(\lambda) = 0,$$

where x and y are right and left eigenvectors respectively, associated with the eigenvalue λ .

A particular case and the most important consideration in this paper is when we let $k = 2$ in (1), the quadratic eigenvalue problem (QEP).

$$Q(\lambda)x = 0 \text{ with } Q(\lambda) = \lambda^2 M_2 + \lambda M_1 + M_0, \quad (2)$$

When $k = 1$, we have the generalized eigenvalue problem (GEP)

$$Ax = \lambda Bx, \quad (3)$$

and if $k = 1$; $M_0 = I$, where I is an $n \times n$ matix.

Equation (1) corresponds to the standard eigenvalue problem (SEP)

$$Ax = \lambda x.$$

Unlike SEP and GEP, the QEP has 2n eigenvalues which may be infinite or finite with up to $2n$ left and $2n$ right eigenvectors, and if there are more than n eigenvectors they do not form a linearly independent set (Francoise and Karl, 2001).

QEPs belong to the class of nonlinear eigenvalue problems (NEP) and are said to be regular if $\det Q(\lambda) \not\equiv 0$, $\forall\lambda$ and is called nonregular if otherwise. We will assume throughout this work that Q(λ) is regular.

If the coefficient matrix of λ^2 is nonsingular, Q(λ) is regular with $2n$ finite eigenvalues and when the leading coefficient of Q(λ) is singular the characteristic polynomial of detQ(λ) yields $r < 2n$ finite eigenvalues and the other $2n-r$ are infinite. These infinite eigenvalues are the zero eigenvalues of the λ -matrix $\lambda^2 Q(\lambda^{-1})$ called reverse matrix polynomial in (Francoise and Karl, 2001).

Given an eigenvalue λ_0, the number α of its repetitions as a root of the characteristics polynomial detQ(λ) is called *algebraic multiplicity* of λ_0. When α = 1, the eigenvalue is said to be

simple. The *geometric multiplicity* β of an eigenvalue λ_0, is the dimension of Ker(Q(λ_0)), that is, the geometric multiplicity is the number of linearly independent eigenvectors corresponding to λ_0. For an n x n matrix Q(λ), the largest geometric multiplicity an eigenvalue λ_0 can have is n. If α = β, then the eigenvalues are said to be *semisimple*. It can be easily seen that simple eigenvalues are semisimple but the converse is not true.

Usually the matrices involved in QEPs are very large and sparse. So using MATLAB function *polyeig* or *quadeig* demands high time of computation and requires large computer memory space. Hence Krylov based methods such as standard Arnoldi algorithm, Second Order Arnoldi algorithm (SOAR) (Bai and Su, 2005) and Quadratic Arnoldi (Q-Arnoldi) (Karl, 2008) algorithms were adopted to extract a few eigenpairs of the QEP. Arnoldi algorithm has these limitations: it solves QEP of twice the dimension than the original problem. Secondly, properties of the matrices may be lost in the process of linearization. These limitations have been taken care by the SOAR method (Bai and Su, 2005). However, the SOAR algorithm also has a limitation as it can not perform implicit restart. This limitation can be overcome by Q-Arnoldi algorithm (Karl, 2008). Furthermore, Q-Arnoldi suffers from numerical instability (vectors spanning the Krylov subspace may lose orthogonality). In this paper, we studied Arnoldi methods for the solution of QEP. In addition, we inverted the matrices twice in Q-Arnoldi algorithm and found that the algorithm is numerically stable.

LINEARIZATION

Our procedure involves linearization of (2). Consider a transformation of (2), similar to the linearization of 2nd-order differential equation. Recall that 2nd-order DE is normally reduced to the first order differential equation. That is, reduction of QEP to its linear form.

A common method to solve equation (2) is to rewrite the QEP as a linear problem $Az = \lambda Bz$ of twice the dimension such as, for example,

$$\begin{bmatrix} -M_1 & -M_0 \\ I & 0 \end{bmatrix} \begin{bmatrix} \lambda x \\ x \end{bmatrix} = \lambda \begin{bmatrix} M_2 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \lambda x \\ x \end{bmatrix},$$

that is, QEP(2) can be re-written as

$$(\lambda B + A)z = 0 \qquad (4)$$

where

$$A = \begin{bmatrix} M_1 & M_0 \\ -I & 0 \end{bmatrix}, B = \begin{bmatrix} M_2 & 0 \\ 0 & I \end{bmatrix}, \ z = \begin{bmatrix} \lambda x \\ x \end{bmatrix},$$

known as first companion form. Another way of linearization is to write equation (2) as

$$\begin{bmatrix} 0 & I \\ -M_2^{-1}M_0 & -M_2^{-1}M_1 \end{bmatrix} \begin{bmatrix} x \\ \lambda x \end{bmatrix} = \lambda \begin{bmatrix} x \\ \lambda x \end{bmatrix} \qquad (5)$$

where M_2 is nonsingular.

$$Au = \lambda u,$$

and

$$A = \begin{bmatrix} 0 & I \\ -M_2^{-1}M_0 & -M_2^{-1}M_1 \end{bmatrix}$$

$$u = \begin{bmatrix} x \\ \lambda x \end{bmatrix}$$

By so doing, we transform (2) into $Au = \lambda u$ that takes the form of a standard eigenvalue problem. We say that equation (4) and (5) are linearization of (2). There is no unique way to transform QEPs to GEPs, however, it is always advisable to use the one that preserves the properties of Q(λ) such as symmetry.

We now show that $\lambda B + A$ have the same eigenvalue as Q(λ). From definition 2, we have

$$\lambda B + A = \begin{bmatrix} -I & -(\lambda M_2 M_1) \\ 0 & I \end{bmatrix} \begin{bmatrix} \lambda^2 M_2 + \lambda M_1 + M_0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & -I \\ -I & \lambda I \end{bmatrix}$$

the pencil $\lambda B + A$ is equivalent to

$$\begin{bmatrix} \lambda^2 M_2 + \lambda M_1 + M_0 & 0 \\ 0 & I \end{bmatrix}$$

and

$$\det(\lambda B + A) = \det(\lambda^2 M_2 + \lambda M_1 + M_0).$$

This shows that $\lambda B + A$ has the same eigenvalue as Q(λ).

Another way of rewriting (2) is the second companion form

$$L_2(\lambda) = \begin{bmatrix} -M_0 & 0 \\ 0 & N \end{bmatrix} - \lambda \begin{bmatrix} M_1 & M_2 \\ N & 0 \end{bmatrix},$$

where N is nonsingular. Other form of linearization as shown in (Karl, 2008) that is of important in this paper is as follows

$$L_3(\lambda) = \begin{bmatrix} N & 0 \\ 0 & M_0 \end{bmatrix} - \lambda \begin{bmatrix} 0 & \mathrm{N} \\ -\mathrm{M}_2 & -\mathrm{M}_1 \end{bmatrix},$$

Where

$$\mathrm{A} = \begin{bmatrix} N & 0 \\ 0 & M_0 \end{bmatrix} \quad , \qquad \mathrm{B} = \begin{bmatrix} 0 & N \\ -M_2 & -M_1 \end{bmatrix}$$

And N is an arbitrary nonsingular matrix:

Arnoldi Algorithm

The three algorithms that were introduced in the previous section would be described here, we start with the Arnoldi Algorithm

---

Algorithm 1: Arnoldi Algorithm

Input: $G \in \mathbb{R}^{N \times N}$, $v \in \mathbb{R}^{N} \setminus \{0\}$, $k \leq n$

Output: $\hat{\mathrm{H}}_k$, V_{k+1}

1. $v_1 = \frac{v}{\|v\|}$
2. for $j = 1, 2. . .,\ k$ do
3. s = Gv_j
4. for $i = 1; 2 \ldots, j$ do
5. $h_{i;\,j} = v^{T}{}_i\, s$
6. $s = s - v_i h_{i,j}$
7. end for
8. $h_{j+1;\,j} = \| s \|_2$
9. if $h_{j+1;\,j} = 0$ breakdown
10. $v_{j+1} = s/h_{j+1;\,j}$
11. end for

Now apply Algorithm 1 to

$$\mathrm{G} = \boldsymbol{B}^{-1}\ \mathbf{A} = \begin{bmatrix} -M_2^{-1} M_1 & -M_2^{-1} M_0 \\ I & 0 \end{bmatrix}$$

where **A** and **B** are as defined in equation (4). Take $N = 2n$ and let

$$C = -M_2^{-1}M_1,\ D = -M_2^{-1}M_0 \quad (8)$$

and choose an arbitrary vector v of length $2n$ and an integer k≤ n: Let V_k be the $2n$ x k matrix that contains the vectors

$(v_1, v_2, .., v_k)$ and H_k be the Hessenberg matrix of size k x $(k + 1)$ that contains the entries $h_{i,j}$ as computed by the algorithm in line 5, then the procedure can be re-written more compactly as

$$GV_k - V_kH_k = v_{k+1}e_k^T h_{k+1k} \quad (9)$$

$GV_k - V_{k+1}\hat{H}_k = 0$, or

$$\begin{bmatrix} C & D \\ I & 0 \end{bmatrix} V_k = V_{k+1}\,\hat{H}_k \quad (10)$$

where $V_{k+1} = [V_k \ \ v_{k+1}]$ is an orthonormal matrix of size 2n x (k+1) and $\hat{H}_k$ is upper Hessenbarg defined as

$$\hat{H}_k = \begin{bmatrix} \hat{H}_k \\ e_k^T\ t_{k+1\,k} \end{bmatrix}$$

of size $(k + 1)$ x k . K iterations cost $2N(k + 3)k$ flops, without the cost of matrix-vector multiplication at line 3. Equation (9) is known as Arnoldi recurrence relation.

Second Order Arnoldi Procedure (SOAR)

SOAR is a method developed recently by (Bai and Su, 2005). The method is similar to Arnoldi procedure but with the advantage of extracting small portion of the spectrum of QEP without doubling its dimension. In this way, the properties of the QEP and its essential structures are maintained. We now present the second order Arnoldi procedure. Choose C and D as defined in equation (8) with an arbitrary vector v, an integer k and v having the same length as C and D.

Algorithm 2: Second Order Arnoldi (SOAR) Procedure

Input: $G \in \mathbb{R}^{N \times N}$, $v \in \mathbb{R}^N \setminus \{0\}$, $k \leq n$

Output: $\hat{H}_k$, V_{k+1}

1. $v_1 = \frac{v}{\|v\|}$
2. $p_1 = 0$
3. for $j = 1, 2 \ldots, k$ do
4. $s = Cv_j + Dp_j$
5. $t = v_j$
6. for $i = 1, 2, \ldots, j$ do
7. $h_{i;\,j} = v_i^T s$
8. $s = s - v_i h_{i,\,j}$
9. $t = t - p_i h_{i,\,j}$
10. end for
11. $h_{j+1;\,j} = \|s\|_2$

12 . if $h_{j+1;\,j} = 0$ stop
13. $v_{j+1} = s / h_{j+1\,j}$
14. $p_{j+1} = t / h_{j+1\,j}$
15. end for

Note that in line 4 there is a matrix -vector multiplication of the matrices C and D, therefore, their structure and sparsity can be preserved. This feature can equally be found in the basic Arnoldi procedure, hence SOAR algorithm and basic Arnoldi algorithm coincide in this feature. The loop defined in line 6 through 10 is used to compute a sequence of orthonormal vectors v_i. The vector p_i is just an auxiliary vector.
Also note that in line 12 when norm of s is zero, the algorithm stops at that point. In which we either have breakdown or deflation which can equally be found again in the basic Arnoldi process.

The Algorithm generates quantities that are related as follows. Let V_k be n x k matrix with column vector $v_1, \ldots, v_k$, and P_k be a matrix with the same size as V_k with vectors replaced by $p_{1,\ldots,} p_k$, and let H_k be the k x k upper Hessenberg matrix with entries $h_{i\,j}$ as computed in the algorithm, then we have the following fundamental relationships:

$$CV_k + DP_k = V_k H_k + v_{k+1} e_k^T h_{k+1\,k} \qquad (11)$$
$$V_k = P_k H_k + p_{k+1} e_k^T h_{k+k} \qquad (12)$$

Recall that the vector sequence; $v_1, v_2, \ldots, v_k, v_{k+1}$ are orthonormal. Let $\hat{H}$ be an $(k+1)$ x k upper Hessenerg matix with the form

$$\hat{H}_k = \begin{bmatrix} H_k \\ e_k^T t_{k+1\,k} \end{bmatrix}$$

Then we can rewrite equation (9) and (10) as follows:

$$\begin{bmatrix} C & D \\ I & 0 \end{bmatrix} \begin{bmatrix} V_k \\ P_k \end{bmatrix} = \begin{bmatrix} V_{k+1} \\ P_{k+1} \end{bmatrix} \hat{H}_k$$

which established the relationship between SOAR and Arnoldi Algorithms.

Computing the Ritz pairs for SOAR

In this section we will give details on how to approximate a large-scale quadratic eigenvalue problem to a small scale quadratic eigenvalue problem by using the orthonormal Rayleigh-Ritz approximation procedure. In order to apply the technique based on the second order Krylov subspace $K_k(C, D, v)$ with C = $-M_2^{-1}$M$_1$ and D = $-M_2^{-1}$ M$_0$. We find an approximate eigenpair (θ, l) where $\theta \epsilon \mathbb{C}$ and $l \in$ K$_k$(C, B, v), by imposing the Galerkin Condition ,

$$(\theta^2 M_2 + \theta M_1 + M_0) l \perp K_k(C, D, v),$$

which can be written as

$$r^T(\theta^2 M_2 + \theta M_1 + M_0)\, l = 0; \quad \forall r \epsilon K_k(C; D; v): \tag{13}$$

Since $l \in (C, D, v)$, we can write it as

$$l = V_k j, \tag{14}$$

where the n x k matrix V_k is an orthonormal basis of $K_k(C, D, v)$ generated by the algorithm 2 (SOAR procedure), and j is a vector of length $k \leq n$. By equation (13) and (14) it follows that θ and j satify the following reduced QEP.

$$(\theta^2 M_2 + \theta M_{1m} + M_{om})\text{g} = 0 \tag{15}$$

with

$$M_{2m} = V_k^T M_2 V_k, \quad M_{1m} = V_k^T M_1 V_k, \quad M_{0m} = V_k^T M_0 V_k. \tag{16}$$

The eigenpairs (θ, j) of (15) define the Ritz pair (θ, l). Now θ, l are the approximate eigenvalue and eigenvector of the QEP (2) respectively. We can use the norms of the residual vectors

$$\text{re} = (\theta^2 \text{M}_2 + \theta\, \text{M}_1 + \text{M}_0)\, l,$$

to test the accuracy of the eigenpairs (θ, l).

By clearly formulating the above matrices M_{2m}, M_{1m} and M_{0m}, essential structures of M_2, M_1 and M_0 are preserved. Consequently , the essential spectral properties of the quadratic eigenvalue problem will be maintained

QUADRATIC ARNOLDI ALGORITHM

The Q-Arnoldi Algorithm was developed in (Karl, 2008) and is used to compute few eigenpairs of a large QEP possibly sparse through linearization. We will give details on how to use it.

Decomposition of jth Arnoldi vectors

In an attempt to make standard Arnoldi Algorithm (Algorithm 1) more effective. We will split split the jth Arnoldi vectors as follows

$$v_j = \begin{pmatrix} y_j \\ v_j \end{pmatrix} \tag{17}$$

where y_j, $v_j \in C^n$. Then the recurrence relation (9) can be written as

$$\begin{bmatrix} -M_2^{-1}M_1 & -M_2^{-1}M_0 \\ I & 0 \end{bmatrix} \begin{pmatrix} Y_k \\ V_k \end{pmatrix} - \begin{pmatrix} Y_k \\ V_k \end{pmatrix} H_k = h_{k+1\,k} \begin{pmatrix} y_{k+1} \\ v_{k+1} \end{pmatrix} e_k{}^T. \tag{18}$$

From the second block row of the above equation (18), we have

$$Y_k = V_{k+1}\hat{\text{H}}_k \tag{19}$$

With the above expression the cost of storage is reduced by a reasonable amount because one needs to store v_{k+1}, y_{k+1} and V_k only to execute the Arnoldi recurrence relation. All in all we have $(2 + k)n$, which is a reasonable reduction compared to standard Arnoldi Algorithm. Based on this observation, we now present the following Algorithm.

Algorithm 3: Q-Arnoldi Procedure

input: $M_2 \in \mathbb{R}^{n \times n}$; $M_1 \in \mathbb{R}^{n \times n}$; $M_0 \in \mathbb{R}^{n \times n}$; $v \in \mathbb{R}^n$; $y \in \mathbb{R}^n$; $k \leq n$

;

output: $\hat{H}_k$ V_{k+1}

1. Choose v_1 and y_1 so that $\|v_1\|_2^2 + \|y_1\|_2^2 = 1$.
2. for $i = 1, 2, \ldots, k$ do
3. $s = Cy_j + Dv_j$
4. $t = y_j$
5. $h_j = \begin{pmatrix} V_{j-1}t + \hat{H}_{j-1}(V_j^* r) \\ v_j^* t + y_j^* s \end{pmatrix}$
6. $t = t - V_j h_j$
7. $s = s - (V_j \;\; y_j)\left(\begin{pmatrix} \hat{H}_{j-1} & 0 \\ 0 & I \end{pmatrix} h_j\right)$
8. $h_{j+1;\,j} = (\|t\|^2 + \|s\|^2)^{\frac{1}{2}}$
9. if $h_{j+1;\,j} = 0$ *stop*
10. $v_{j+1} = {}^{t}/_{h_{j+1\,j}}$
11. $y_{j+1} = r/h_{j+1\,j}$
12. end for

IMPLEMENTATION OF THE ARNOLDI ALGORITHMS.

This section describes the implementation of the three algorithms in practice. We shall begin with the standard Arnoldi algorithm and then the SOAR Algorithm and finally conclude the discussion with the Q-Arnoldi algorithm.

Algorithm 1

1. Build the matrix **G** and/or call standard Arnoldi algorithm with an arbitrary vector v of length $N = 2n$ and specify the number of iteration k to generate the vectors $\{v_1, v_2, \ldots, v_k\} = V_k$.
2. Solve the problem (reduced one) of the form

$$(V_k{}^T G V_k)y = \theta y$$

to get the Ritz pairs (θ, w) of the matrix G, where $w = V_k y$. Observe that from (8), we have $V_k^T G V_k = H_k(1:k, 1:k)$ which is a square upper Hessenberg matrix.

3. Once extracted, we can test the accuracy of the approximate eigenpairs (θ, z) by computing the residual norms as explained in (20), where $z = w(N+1:2N) = \|(N+1:2N)\|_2$

Algorithm 2

1. Run Algorithm 2 (SOAR) with C = $-M_2^{-1}M_1$ and $D = -M_2^{-1}M_0$ and an arbitrary vector v to obtain an $n(k+1)$ orthogonal matrix V_{k+1}.

2. Compute M_{2m}, M_{1m}, and M_{0m} as defined in (16).

3. Solve the reduced problem (15) for (θ, g) and get the pairs (θ, z), where $z = {V_k g}/{\|V_k g\|_2}$.

4. Test the accuracy of the approximate eigenvalues θ and the approximate eigenvectors z of the quadratic eigenvalue problem (2) by the relative norms of residual vectors:

$$\frac{\|re\|_2}{|\theta|^2\|M_2\|_1 + |\theta|\|M_1\|_1 + \|M_0\|_1} \qquad (20)$$

Algorithm 3

1. Run Algorithm 3 (Q-Arnoldi) with $C = -M_2^{-1}M_1$ and $D = -M_2^{-1}M_0$ and an arbitrary vectors v of length n and y of the same length as v such that $\|x\| + \|y\| = 1$ and specify the number of iteration k to generate the basis $\{v_1, v_2, \ldots, v_k\} = V_k$.

2. Solve the reduced problem as described in step 2 of Algorithm 1 above.

3. Compute the residual norms as described in step 3 of Algorithm 1 above.

Numerical Experiments

In this section, we present some numerical experiments to examine the behaviour of the Arnoldi, SOAR, and Q-Arnoldi methods when extracting a few eigenvalues of the QEP (2). We begin with:

Problem 1: This problem has been chosen to demonstrate the convergence behaviours of the three methods. Let M_2; M_1; and M_0 be 2000 x 2000 randomly generated matrices whose elements are normally distributed with variance, standard deviation and mean equal to 1, 1, and 0 respectively. Twenty eigenvalues were extracted by using the Arnoldi, SOAR and Q-Arnoldi algorithms and the results are shown in Figure 1. Note that because SOAR method is applied to

the QEP directly, it produces eigenpairs twice than those produced by both Q-Arnoldi and Arnoldi applied to the linearised QEP. The figure shows that the eigenvalues produced by each method are some how close to each other. In an attempt to show that the three algorithms compute the largest eigenvalues of the QEP (2), we used the MATLAB function eigs to compute twenty largest eigenvalues of the problem. The results are shown in Figure 2, 3 and 4. It is observed that the convergence behaviour of the eigenvalues is similar in each method. Figure 5 shows the computed relative residual norms of problem 1. It can be seen from the figure that after the same number of iterations, it is hard to say which one of the three methods produced small residual.

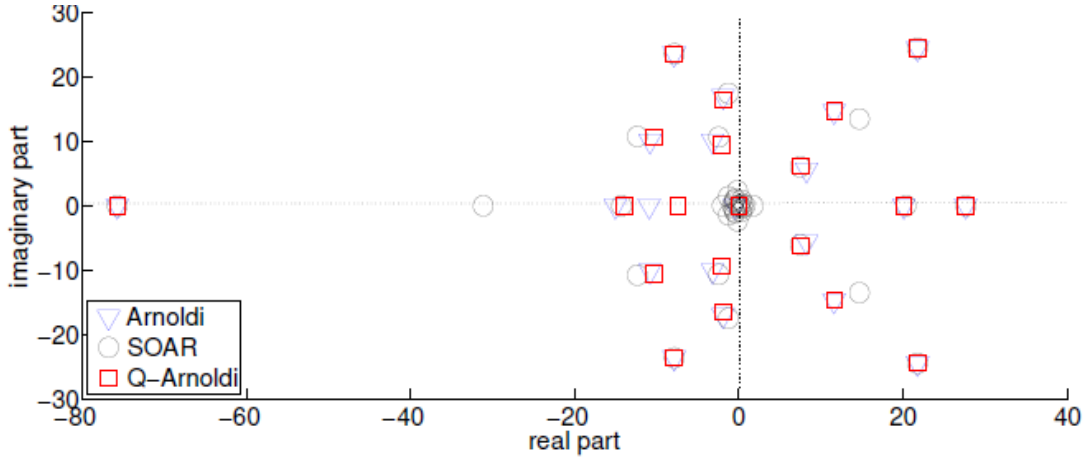

FIGURE 1. *Approximate eigenvalues computed by Arnoldi, SOAR and Q-Arnoldi for a randomly generated QEP (problem 1).*

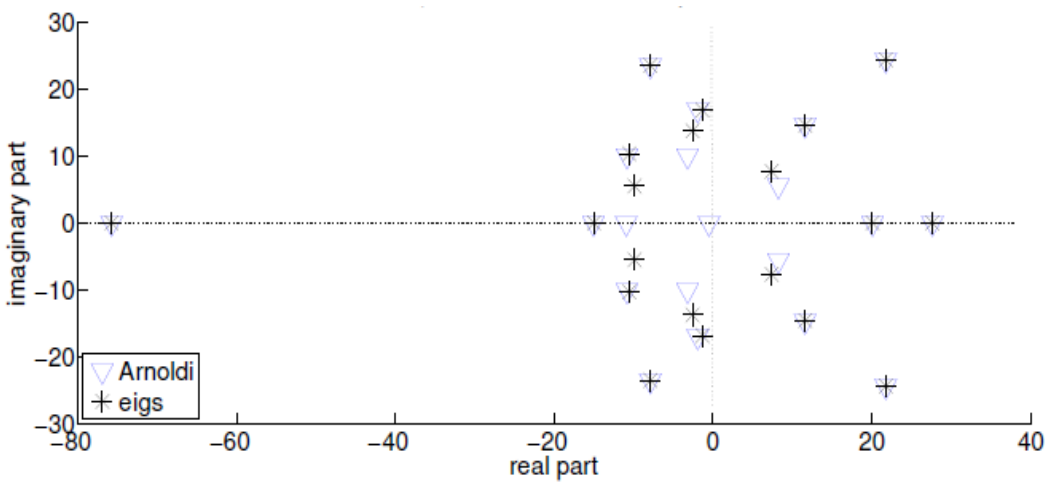

FIGURE 2. *Approximate eigenvalues computed by Arnoldi and the MATLAB function* `eigs` *for a randomly generated QEP (problem 1).*

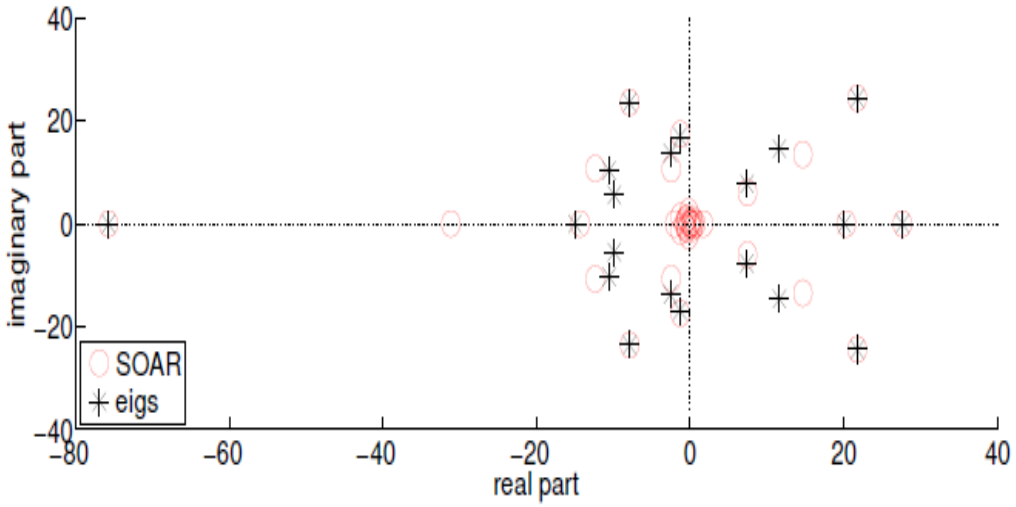

FIGURE 3. *Approximate eigenvalues computed by SOAR and the MATLAB function* `eigs` *for a randomly generated QEP (problem 1).*

Problem 2: In this, we consider a problem called *sign1* in the NLEVP collection (Timo *et al*., 2013). The dimension of the matrices is 10001 with $k = 30$. Figures 6, 7 and 8 show the approximated eigenvalues computed by Arnoldi, SOAR and Q-Arnoldi respectively.

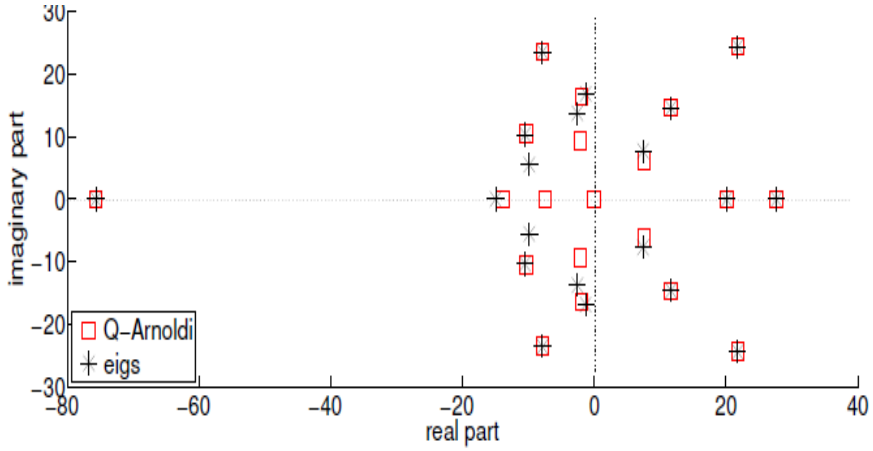

FIGURE 4. *Approximate eigenvalues computed by Q-Arnoldi and the MATLAB function* `eigs` *for a randomly generated QEP (problem 1).*

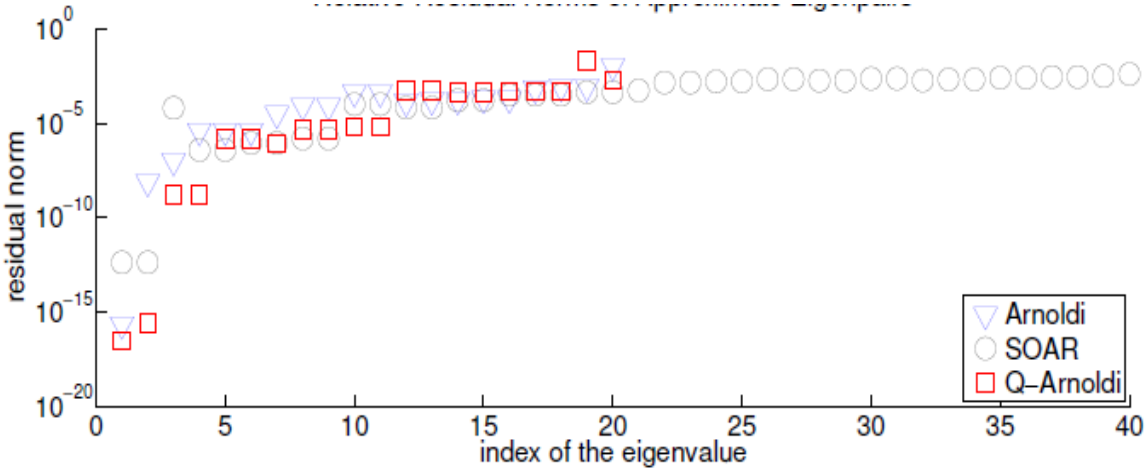

FIGURE 5. *Relative residual norms for problem 1.*

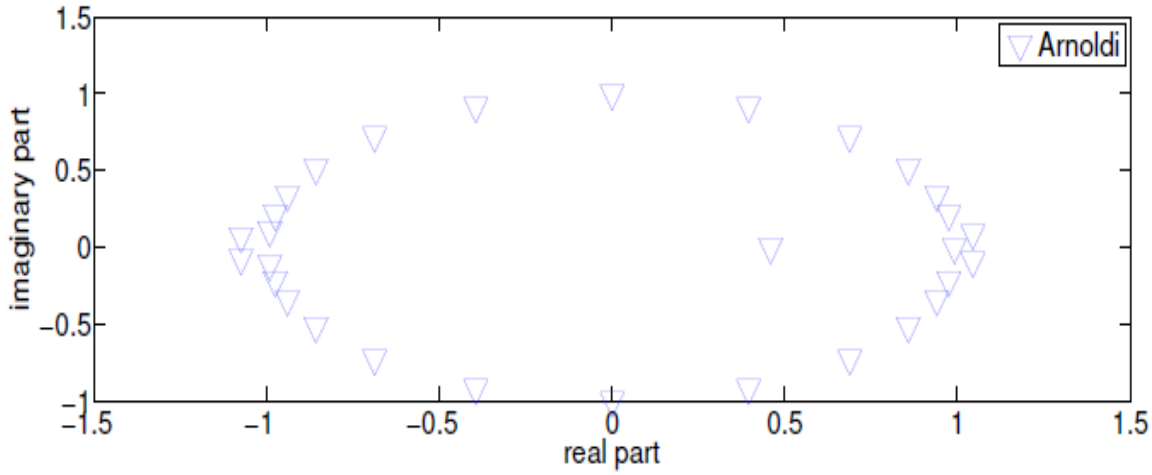

FIGURE 6. *sign1 QEP; approximate eigenvalues computed by Arnoldi algorithm.*

It is reported in (Timo *et al*., 2013) that as the dimension of the matrices increases, the spectrum of the problem is a unit circle. We observed that the reduced QEP by the SOAR method inherits such an interesting property in the process of approximation while Arnoldi and Q-Arnoldi algorithms lose this important property. Note that the eigenvalues are sorted in descending order using the MATLAB function sort. Furthermore, Figure 9 indicates the relative residual norms of the problem. The figure also indicates that SOAR tends to return eigenpairs with a slightly small residual than the other methods.

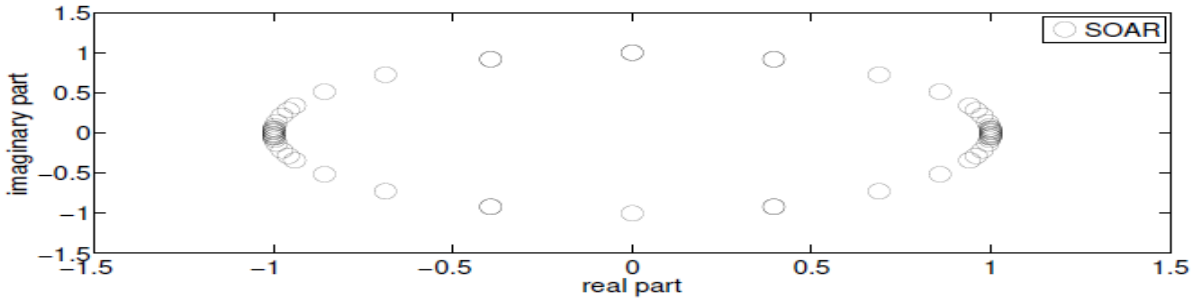

FIGURE 7. *sign1 QEP; approximate eigenvalues computed by SOAR algorithm.*

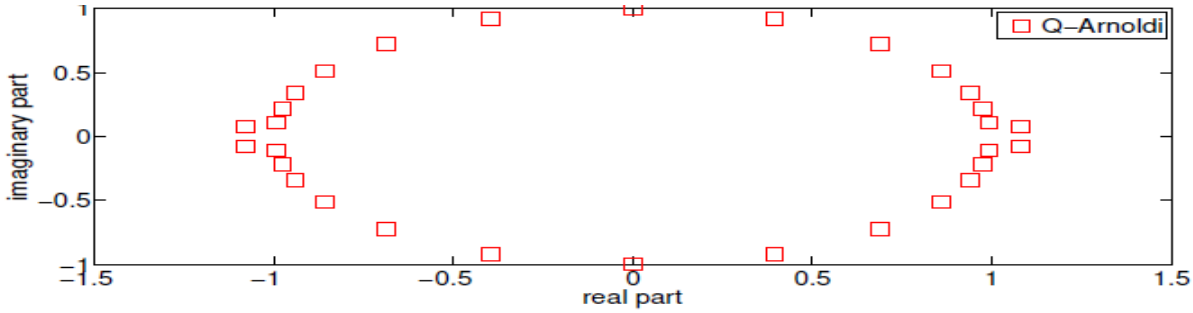

FIGURE 8. *sign1 QEP; approximate eigenvalues computed by Q-Arnoldi algorithm.*

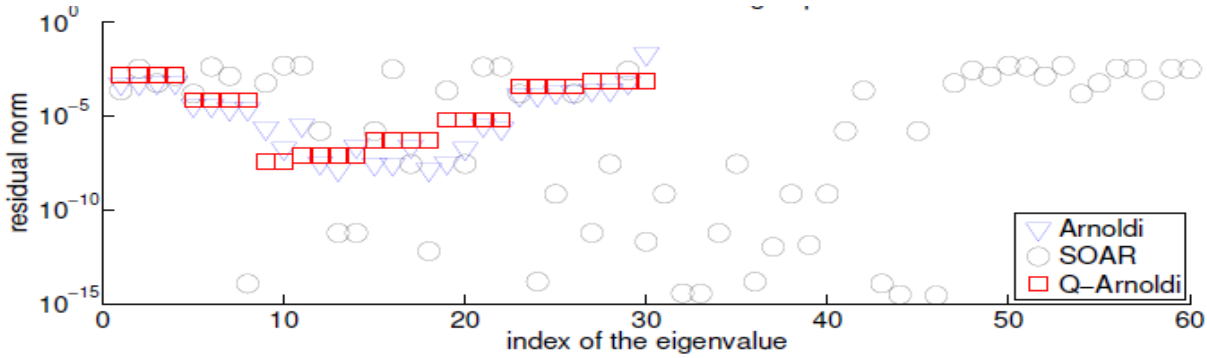

FIGURE 9. *Relative residual norms for problem 3.*

Problem 3: This example demonstrates he lose of orthogonality of the Q-Arnoldi (the one that targets smallest eigenvalue) as expected (Karl, 2008). In this paper we reverse the algorithm to target largest eigenvalue of the QEP. We call the algorithm Reverse Q-Arnoldi (RQ-Arnoldi). In our experiment we found out that the RQ-Arnoldi does not lose orthogonality as it can be seen in table 1.

We have run k= 10 iterations of the Q-Arnoldi and RQ-Arnoldi method for a problem of dimension 10,000 with $M_0 = I, M_1 = \zeta I$ and $M_2 = \zeta\, diag(\mu_1, \ldots, \mu_j)$ where $\mu_j = {}^{1}/_{j}$ for four values of ζ. Table 1 shows that the numerical behavior of the two algorithms for this problem. The lose of the orthogonality occurs when $\|\hat{H}_k\|_2$ is large.

Table 1. Demostaration of th lose of orthogonlity of the Q-Arnoldi and the RQ-Arnoldi does not. $\tau_{\perp} = \|I - V_{k+1}^{*}V_{k+1}\|_F$ is the deiation of Orthogonality and $\tau_R = \left\|Y_k - V_{k+1}\hat{\mathrm{H}}_k\right\|_F$ is the error on the recurrence realtion.

$\zeta = 1$	Q-Arnoldi	RQ-Arnoldi
$k(\hat{H}_k)$	$1.4e+03$	$5.6e+02$
$\|\hat{H}_k\|_2$	1.6	$1.04e+04$
$\tau_{\perp}$	$2.5e-13$	$8.0e-13$
τ_R	$3.5e-17$	$8.5e-16$
$\zeta = 10^4$		
$k(\hat{H}_k)$	$8.5e+06$	$3.7e+06$
$\|\hat{H}_k\|_2$	$1.4e+04$	$9.9e+03$
$\tau_{\perp}$	$1.6e-05$	$1.2e-12$
τ_R	$4.3e-17$	$8.5e-16$
$\zeta = 10^6$		
$k(\hat{H}_k)$	$8.7e+08$	$3.8e+08$
$\|\hat{H}_k\|_2$	$1.4e+06$	$9.9e+03$
$\tau_{\perp}$	0.2152	$2.9e-12$
τ_R	$9.0e-17$	$8.5e-16$
$\zeta = 10^9$		
$k(\hat{H}_k)$	$4.0e+11$	$3.8e+11$
$\|\hat{H}_k\|_2$	$2.1e+08$	$9.9e+03$
$\tau_{\perp}$	5.4372	$1.0e-12$
τ_R	$1.3e-08$	$8.7e-16$

Where $V_{k+1} = \begin{pmatrix} V_k & V_{k+1} \\ V_{k+1}\hat{\mathrm{H}}_k & y_{k+1} \end{pmatrix}$

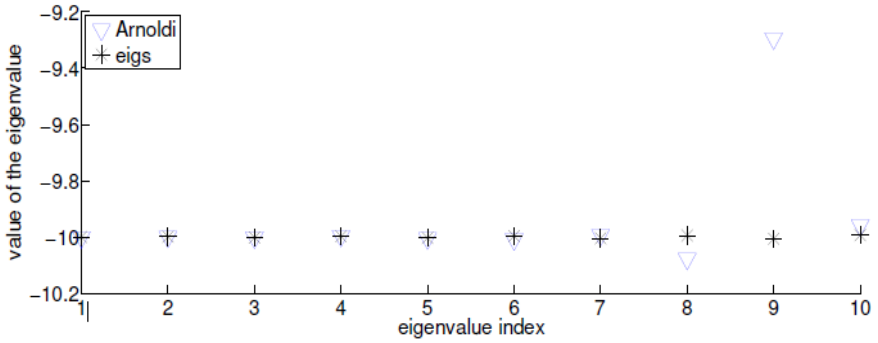

FIGURE 10. *spring QEP; approximated eigenvalues close to -10 computed by Arnoldi and* `eigs`.

Problem 4: Here we examine the partial eigenvalues of the problem called spring in NLEVP collection. This problem arises when an engineer wishes to analyse the damped mass-spring system. To find eigenvalues of interest, that is, close to a shift s, we solved the shift and invert problem of the form

$$(\mu^2\widehat{M}_2 + \mu\widehat{M}_1 + \widehat{M}_0)x = 0 \qquad (21)$$

Whrer $\mu = 1/(\lambda - \sigma), \widehat{M}_2 = \sigma^2 M_2 + \sigma M_1 + M_0, \quad \widehat{M}_1 = M_1 + 2\sigma M_2, and\ \widehat{M}_0 = M_2$. The largest magnitude (in modulus) eigenvalue μ approximates λ of the original problem closest to a shift σ. λ is then given by $\sigma + 1/\mu$. With shift σ = 10, We run the three algorithms with *k=10* and the dimension of the problem was chosen to be 10,000. We used MATLAB function ***eigs*** to compute the eigenvalue close to a shift σ. The results are shown in Figure 10, 11, and 12. It can be observed that with the same number of iterations, SOAR yields highest number of eigenvalues

that are close to those produced by ***eigs***. Figure 13 reports the scaled residual norms of the problem. The figure indicates that the Q-Arnoldi has smallest residual norms.

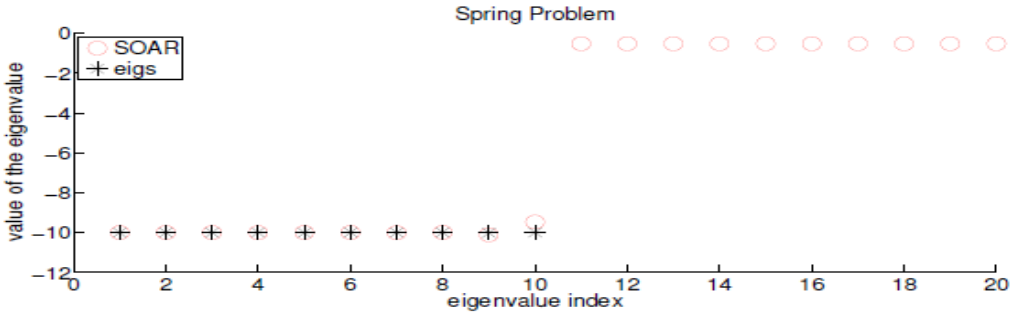

FIGURE 11. *spring QEP; approximated eigenvalues close to -10 computed by SOAR and* `eigs`.

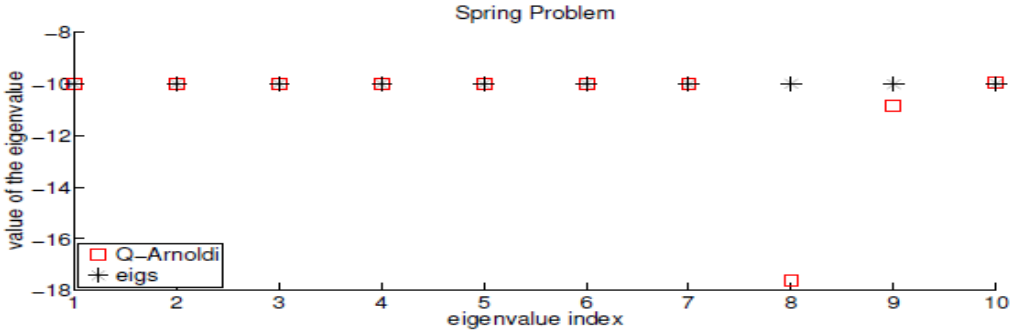

FIGURE 12. *spring QEP; approximated eigenvalues close to -10 computed by Q-Arnoldi and* `eigs`.

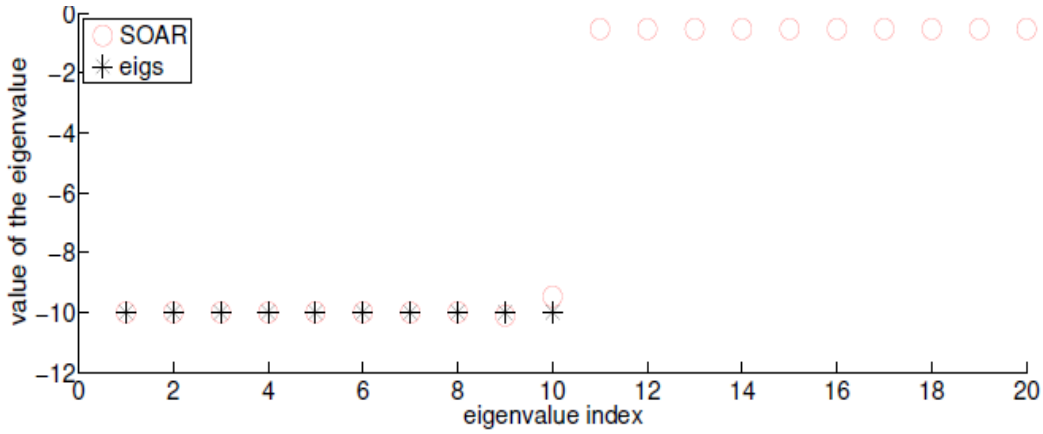

FIGURE 11. *spring QEP; approximated eigenvalues close to -10 computed by SOAR and* `eigs`.

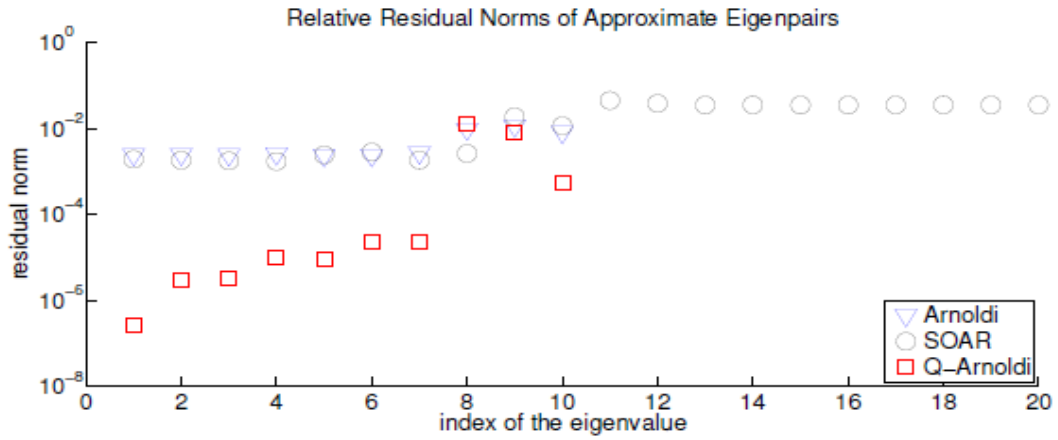

FIGURE 13. *Relative residual norms for problem 5.*

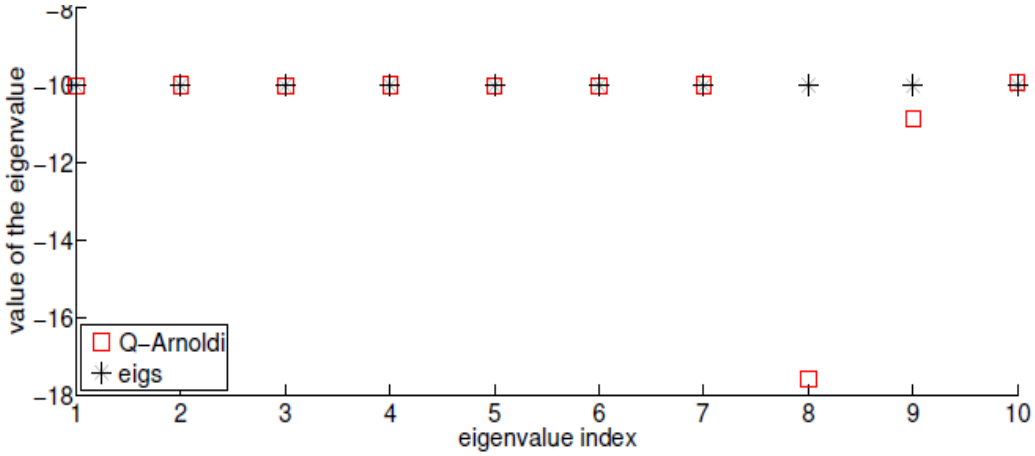

FIGURE 12. *spring QEP; approximated eigenvalues close to -10 computed by Q-Arnoldi and* `eigs`.

CONCLUSIONS

We have performed numerical experiments to understand the behaviour of Arnoldi, SOAR and Q-Arnoldi methods. From the results obtained, it can be concluded that SOAR is fastest in convergence. It preserves the essential properties of the QEP. However Q-Arnoldi is faster in convergence than Arnoldi. We have seen that Q-Arnoldi loses orthogonality when $\|\hat{H}_k\|$ is large but the RQ-Arnoldi introduced in our experiment does not. Furthermore, we performed shift and invert transformation (21) to compute eigenvalues closest to a given 's' of which SOAR method is slightly better in convergence compared to Q-Arnoldi and Arnoldi. We learnt that SOAR can not perform explicit restart while Q-Arnoldi performs. In terms of residual norms both SOAR and Q-Arnoldi produced almost the same result in most of the problems considered.

REFERENCES

Biswa, N, D. (1995). *Numerical Linear Algebra and Applications* . Brooks/Cole Publishing Company

Carmen, C. and Jose, E, R. (2003). Restarted Q-Arnoldi-type methods exploiting symmetry in quadratic eigenvalue problems. *Linear Algebra and its Applications 371, 31-44.*

Françoise, T. and Karl, M. (2001). The Quadratic Eigenvalue Problem. *SIAM Review 43, No. 2, pp. 235-286.*

Gene, H, G. and Charles, F, V. (2013). Matrix Computations , *4th Edition. Johns Hopkins University Press.*

Gohberg I, P. and Rodman, L. (1982). Matrix Polynomials. *Academic Press, New York*.

Karl, M. (2008). The quadratic Arnoldi method for the Solution of the Quadratic Eigenvalue Problem. *SIAM J. Matrix Anal, Appl., Vol. 30, No. 4, pp. 1463-1482*.

Micheal, B. (2005). The Polynomial Eigenvalue Problem . PhD Thesis. *The University of Manchester.*

Seymour L, Marc L. (2001). Linear Algebra. *Third Edition. Schaum's outline series.*

Timo, B. Nicholas, J, H. Volker, M. Christian, S. and Françoise, T. (2013). NLEVP: A Collection of Nonlinear Eigenvalue Problems . *ACM Transactions on Mathematical Software (TOMS) 39, pp.7:1:7:28.*

Tisseur, F. (2000). Backward error and condition of polynomial eigenvalue problems. *Linear Algebra Appl. 309: 339-361.*

Yousef, S. (1992). Numerical Methods for Large eigenvalue Problems . *Manchester University Press.*

Zhaojun, B. and Yangfeng, S. (2005). SOAR: A second order Arnoldi method for the Solution of the Quadratic Eigenvalue Problem. *SIAM J. Matrix Anal. Appl., Vol. 26, No. 3, pp. 640-659*.