I.2 Quadratic Eigenvalue Problems

1 Introduction

The quadratic eigenvalue problem (QEP) is to find scalars λ and nonzero vectors u satisfying

$$Q(\lambda)x = 0, \tag{1.1}$$

where

$$Q(\lambda) = \lambda^2 M + \lambda D + K,$$

M, D and K are given $n \times n$ matrices. Sometimes, we are also interested in finding the left eigenvectors y: $y^H Q(\lambda) = 0$. Note that $Q(\lambda)$ has $2n$ eigenvalues λ. They are the roots of $\det[Q(\lambda)] = 0$.

2 Linearization

A common way to solve the QEP is to first linearize it to a linear eigenvalue problem. For example, let

$$z = \begin{pmatrix} \lambda u \\ u \end{pmatrix},$$

Then the QEP (1.1) is equivalent to the generalized eigenvalue problem

$$L_c(\lambda)z = 0 \tag{2.2}$$

where

$$L_c(\lambda) = \lambda \begin{pmatrix} M & 0 \\ 0 & I \end{pmatrix} + \begin{pmatrix} D & K \\ -I & 0 \end{pmatrix} \equiv \lambda G + C.$$

$L_c(\lambda)$ is called a companion form or a linearization of $Q(\lambda)$.

Definition 2.1. *A matrix pencil $L(\lambda) = \lambda G + C$ is called a* **linearization** *of $Q(\lambda)$ if*

$$E(\lambda)L(\lambda)F(\lambda) = \begin{pmatrix} Q(\lambda) & 0 \\ 0 & I \end{pmatrix} \tag{2.3}$$

for some unimodular matrices $E(\lambda)$ and $F(\lambda)$.[1]

For the pencil $L_c(\lambda)$ in (2.2), the identity (2.3) holds with

$$E(\lambda) = \begin{pmatrix} I & \lambda M + D \\ 0 & -I \end{pmatrix}, \quad F(\lambda) = \begin{pmatrix} \lambda I & I \\ I & 0 \end{pmatrix}.$$

There are various ways to linearize a quadratic eigenvalue problem. Some are preferred than others. For example if M, D and K are symmetric and K is nonsingular, then we can preserve the symmetry property and use the following linearization:

$$L_c(\lambda) = \lambda \begin{pmatrix} M & 0 \\ 0 & K \end{pmatrix} + \begin{pmatrix} D & K \\ K & 0 \end{pmatrix}. \tag{2.4}$$

The following is an outline of the linearization solution process for solving the QEP (1.1), which is implemented by MATLAB function `polyeig(K,D,M)`.

[1] A λ-matrix $E(\lambda)$ is called unimodular, if $\det(E(\lambda)) \equiv \pm 1$.

Algorithm 2.1 (Direct QEP method)**.**

1. Linearize $Q(\lambda)$ into $L(\lambda) = \lambda G + C$.
2. Solve the generalized eigenproblem $L(\lambda)z = 0$.
3. Recover eigenvectors of Q from those of L.

REMARK **2.1.** Additional issues on scaling and numerical sensitivity.[2]

3 Arnoldi methods

For large sparse QEP, we can use the Arnoldi algorithm to compute a few eigenpairs of the QEP through the linearized problem.

3.1 Arnoldi method

We can use the Arnoldi procedure to generate an orthonormal basis V_n of the Krylov subspace $\mathcal{K}_n(-G^{-1}C; -G^{-1}b)$, namely,

$$\begin{aligned} \operatorname{span}\{V_n\} &= \mathcal{K}_n(-G^{-1}C; G^{-1}b) \\ &= \operatorname{span}\{G^{-1}b, (-G^{-1}C)G^{-1}b, \ldots, (-G^{-1}C)^{n-1}G^{-1}b\} \end{aligned}$$

Algorithm 3.1 (Arnoldi procedure)**.**

Input: G, C, b, n
Output: $\widehat{H}_n$, V_{n+1}
1. $v_1 = G^{-1}b/\|G^{-1}b\|$
2. for $j = 1, 2, \ldots, n$ do
3. Solve $Gr = -Cv_j$ for r
4. $h_j = V_j^T r$
5. $r = r - V_j h_j$
6. $h_{j+1,j} = \|r\|$
7. stop if $h_{j+1,j} = 0$
8. $v_{j+1} = r/h_{j+1,j}$
9. end for

The governing equation of the Arnoldi procedure is

$$(-G^{-1}C)\, V_n = V_{n+1}\widehat{H}_n, \tag{3.5}$$

where $\widehat{H}_n$ is an $(n+1) \times n$ upper Hessenberg matrix and V_{n+1} is a $2N \times (n+1)$ matrix with orthonormal columns.

A description of the basic Arnoldi method for solving the QEP (1.1) based on linearization (2.2) is as follows.

Algorithm 3.2 (Basic Arnoldi Method for Linearized QEP)**.**

1. Transform the QEP (1.1) to the equivalent generalized eigenvalue problem (2.2).
2. Run the Arnoldi procedure with the matrix $H = -G^{-1}C$ and the vector $v = (\, u^T \ 0\,)^T$ to generate an orthonormal basis $\{v_1, v_2, \ldots, v_n\}$ of the Krylov subspace $\mathcal{K}_n\,(H; v)$. Let $V_n = (v_1, v_2, \ldots, v_n)$.

[2] N. J. Higham, D. S. Mackey, F. Tisseur and S. D. Garvey. Scaling, Sensitivity and Stability in the Numerical Solution of Quadratic Eigenvalue Problems, Internat. J. Numer. Methods Eng., 73(3):344-360, 2008.

3. Solve the reduced eigenvalue problem

$$(V_n^T H V_n)t = \theta t,$$

and obtain the Ritz pairs (θ, y) of the eigenvalue problem of the single matrix H, where $y = V_n t$. Note that by (3.5), $V_n^T H V_n = H_n(1:n, 1:n)$ is an $n \times n$ upper Hessenberg matrix returned directly from the Arnoldi procedure without additional cost.

4. Extract the approximate eigenpairs $(\theta,\ z)$ of the QEP (1.1), and test their accuracy by the residual norms as described in (5.19), where $z = y(N+1:2N)/\|y(N+1:2N)\|_2$.

In practice, one may incorporate the implicit restarting scheme as we discussed for the standard Arnoldi procedure.

4 Q-Arnoldi method

Note that

$$-G^{-1}C = \begin{pmatrix} -M^{-1}D & -M^{-1}K \\ I & 0 \end{pmatrix} = \begin{pmatrix} A & B \\ I & 0 \end{pmatrix}.$$

Let us partition the jth Arnoldi vector v_j into

$$v_j = \begin{pmatrix} u_j \\ w_j \end{pmatrix}$$

where u_j and w_j are vectors of length n. From the second block row of the governing equation (3.5) of the Arnoldi procdure, we have

$$U_n = W_{n+1}\widehat{H}_n. \tag{4.6}$$

We can exploit this relation to avoid the storage of the U-vectors with a slight increase of computational cost, since all products with U_n are to be replaced by products with W_{n+1} and $\widehat{H}_n$. Based upon this observations, we derive the following algorithms.

Algorithm 4.1 (Q-Arnoldi procedure)**.**

Input: A, B, b, n

Output: $\widehat{H}_n$, W_{n+1}

1. $u = b/\|b\|$ and $w_1 = 0$
2. for $j = 1, 2, \ldots, n$ do
3. $\quad r = Au + Bw_j$
4. $\quad t = u$
5. $\quad h_j = \begin{pmatrix} \widehat{H}_{j-1}^T(W_j^T r) + W_{j-1}^T t \\ u^T r + w_j^T t \end{pmatrix}$
6. $\quad r = r - \begin{pmatrix} W_j & u \end{pmatrix} \left(\begin{pmatrix} \widehat{H}_{j-1} & 0 \\ 0 & 1 \end{pmatrix} h_j \right)$
7. $\quad t = t - W_j h_j$
8. $\quad h_{j+1,j} = (\|r\|^2 + \|t\|^2)^{1/2}$
9. $\quad$ stop if $h_{j+1,j} = 0$
10. $\quad u = r/h_{j+1,j}$
11. $\quad w_{j+1} = t/h_{j+1,j}$
12. end for

Consequently, we can derive an Q-Arnoldi method with a simple replacement of the Arnoldi procedure by the Q-Arnoldi procedure at Step 2 of Algorithm 3.2.

5 Second-order Arnoldi procedure

The Arnoldi approach has disadvantages such as the loss of essential structures of the original problem in the process of linearization. For example, when coefficient matrices M, C and K are symmetric positive definite, the transformed generalized eigenvalue problem (2.3) has to be either nonsymmetric where one of G and C has to be nonsymmetric, or a symmetric indefinite where both of G and C are symmetric but neither of them will be positive definite. Researchers have been studying numerical methods which can be applied to the large-scale QEP directly. In these methods, they do not transform the second-order form to an equivalent linear form; instead, they project the second-order form onto a properly chosen low-dimensional subspace to reduce it to a second-order problem directly. The following second-order Arnolid method is an approached proposed recently.

5.1 A second-order Krylov subspace

Let us begin by defining the generalized Krylov subspace induced by a pair of matrices A and B and a vector u.

Definition 5.1. *Let A and B be square matrices of order N, and $u \neq 0$ be an N-vector. Then the sequence*

$$r_0, r_1, r_2, \ldots, r_{n-1}, \tag{5.7}$$

where

$$\begin{aligned} r_0 &= u, \\ r_1 &= Ar_0, \\ r_j &= Ar_{j-1} + Br_{j-2} \quad \text{for } j \geq 2, \end{aligned}$$

is called the ***second-order Krylov sequence*** *of A, B on u. The space*

$$\mathcal{G}_n\left(A, B; u\right) = \operatorname{span}\{r_0, r_1, r_2, \ldots, r_{n-1}\},$$

*is called the n**th second-order Krylov subspace**.*

We now discuss the motivation for the definition of the second-order Krylov subspace $\mathcal{G}_n\left(A, B; u\right)$ in the context of solving the QEP (1.1). Recall that the QEP (1.1) can be transformed to an equivalent generalized eigenvalue problem (2.2). If one applies a Krylov subspace technique to (2.2), then an associated Krylov subspace would naturally be

$$\mathcal{K}_n\left(H; v\right) = \operatorname{span}\left\{v, Hv, H^2v, \ldots, H^{n-1}v\right\}, \tag{5.8}$$

where v is a starting vector of length $2N$, and

$$H = -G^{-1}C = \begin{pmatrix} -M^{-1}D & -M^{-1}K \\ I & 0 \end{pmatrix}. \tag{5.9}$$

Let $A = -M^{-1}D$, $B = -M^{-1}K$ and $v = [u^{\mathrm{T}} \quad 0]^{\mathrm{T}}$, then it immediately derives that the second-order Krylov vectors $\{r_j\}$ of length N defined in (5.7) and the standard Krylov vectors $\{H^j v\}$ of length $2N$ defined in (5.8) is related as the following form

$$\begin{bmatrix} r_j \\ r_{j-1} \end{bmatrix} = H^j v \quad \text{for } j \geq 1. \tag{5.10}$$

In other words, the generalized Krylov sequence $\{r_j\}$ *defines* the entire standard Krylov sequence based on H and v. Equation (5.10) indicates that the subspace $\mathcal{G}_j\left(A, B; u\right)$ of $\mathcal{R}^N$ should be able to provide sufficient information to let us directly work with the QEP, instead of using the subspace $\mathcal{K}_n\left(H; v\right)$ of $\mathcal{R}^{2N}$ for the linearized eigenvalue problem (2.2).

By the observation (5.10), the relationship between the standard Krylov subspace $\mathcal{K}_n(H; v)$ and the second-order Krylov subspace $\mathcal{G}_n(A, B; u)$ can be charaterized by the following theorem.

Theorem 5.1. *Let Q_n be an orthonormal basis of the second-order Krylov subspace $\mathcal{G}_n(A, B; u)$. Let $Q_{[n]}$ denote the following 2 by 2 block diagonal matrix*

$$Q_{[n]} = \begin{pmatrix} Q_n & \\ & Q_n \end{pmatrix} \tag{5.11}$$

Then $H^\ell v \in \operatorname{span}\{Q_{[n]}\}$ for $\ell = 0, 1, 2, \ldots, n-1$. This means that

$$\mathcal{K}_n(H; v) \subseteq \operatorname{span}\{Q_{[n]}\}.$$

We call that the standard Krylov subspace $\mathcal{K}_n(H; \widehat{b}_0)$ is embedded *into the second-order Krylov subspace $\mathcal{G}_n(A, B; r_0)$.*

5.2 Second-order Arnoldi procedure

We now turn to the question of how to construct an orthonormal basis $\{q_i\}$ of $\mathcal{G}_j\,(A, B; u)$. Namely,

$$\operatorname{span}\{q_1, q_2, \ldots, q_j\} = \mathcal{G}_j\,(A, B; u) \quad \text{for } j \geq 1.$$

The following is that so-called ***Second-Order ARnoldi*** (SOAR) procedure to generate the orthonormal basis.

Algorithm 5.1 (SOAR Procedure)**.**

```
1.  q_1 = u/||u||_2
2.  p_1 = 0
3.  for j = 1, 2, ..., n do
4.      r = A q_j + B p_j
5.      s = q_j
6.      for i = 1, 2, ..., j do
7.          t_ij = q_i^T r
8.          r := r - q_i t_ij
9.          s := s - p_i t_ij
10.     end for
11.     t_{j+1 j} = ||r||_2
12.     if t_{j+1 j} = 0, stop
13.     q_{j+1} = r/t_{j+1 j}
14.     p_{j+1} = s/t_{j+1 j}
15. end for
```

Remark **5.1.** The **for**-loop in Lines 6-10 is an orthogonalization procedure with respect to the $\{q_i\}$ vectors. The vector sequence $\{p_j\}$ is an auxiliary sequence. In the next section, we will present a modified version of the algorithm to remove the requirement of explicit references to vectors p_j.

Let $Q_n = (q_1, q_2, \ldots, q_n)$, $P_n = (p_1, p_2, \ldots, p_n)$, $T_n = (t_{ij})_{n\times n}$. Note that T_n is upper Hessenberg. Then the following relations hold:

$$\begin{pmatrix} A & B \\ I & 0 \end{pmatrix} \begin{pmatrix} Q_n \\ P_n \end{pmatrix} = \begin{pmatrix} Q_{n+1} \\ P_{n+1} \end{pmatrix} \widehat{T}_n, \tag{5.12}$$

where $Q_n^T Q_n = I_n$, and

$$\widehat{T}_n = \begin{pmatrix} T_n \\ e_n^T t_{n+1\,n} \end{pmatrix}.$$

This relation assembles the similarity between the SOAR procedure and the Arnoldi procedure. The essential difference between the SOAR procedure and the Arnoldi procedure is that in SOAR, entries t_{ij} of $\widehat{T}_n$ are chosen to enforce the orthonormality among vectors q_j of dimension N, whereas in the Arnoldi method, entries h_{ij} of $\widehat{H}_n$ are determined to ensure the orthonormality among vectors v_j of dimension $2N$.

The following theorem ensures that the vector sequence $\{q_1, q_2, \ldots, q_n\}$ indeed is an orthonormal basis of the second-order Krylov subspace $\mathcal{G}_j\,(A, B; u)$.

Theorem 5.2. *If $t_{i+1,i} \neq 0$ for $i \leq j$, then the vector sequence $\{q_1, q_2, \ldots, q_j\}$ forms an orthonormal basis of the second-order Krylov subspace $\mathcal{G}_j\,(A, B; u)$:*

$$\operatorname{span}\{Q_j\} = \mathcal{G}_j\,(A, B; u) \quad \textit{for } j \geq 1. \tag{5.13}$$

Now let us exploit the relations in Algorithm 5.1 to derive a new version, which reduces memory requirement and floating point operations by almost one half.

By (5.12) and noting that $p_1 = 0$, we have

$$Q_n = P_{n+1}\widehat{T}_n = P_{n+1}(:, 2 : n+1) \cdot \widehat{T}_n(2 : n+1, 1 : n)$$

and

$$AQ_n + BQ_nS_n = Q_nT_n + q_{n+1}e_n^T t_{n+1\,n} = Q_{n+1}\widehat{T}_n, \tag{5.14}$$

where S_n is an $n \times n$ strictly upper triangular matrix of the form

$$S_n = \begin{pmatrix} 0 & \widehat{T}_n(2 : n, 1 : n-1)^{-1} \\ 0 & 0 \end{pmatrix}.$$

Equation (5.14) suggests a method for computing the vector q_{j+1} directly from $q_1, q_2, \ldots, q_j$. This leads to the following algorithm:

Algorithm 5.2 (SOAR procedure with deflation and memory saving)**.**

```
1.   q_1 = u/||u||_2
2.   f = 0
3.   for j = 1, 2, ..., n do
4.       r = Aq_j + Bf
5.       for i = 1, 2, ..., j do
6.           t_ij = q_i^T r
7.           r := r − q_i t_ij
8.       end for
9.       t_{j+1 j} = ||r||_2
10.      if t_{j+1 j} ≠ 0,
11.          q_{j+1} := r/t_{j+1 j}
12.          f = Q_j T̂(2 : j+1, 1 : j)^{-1} e_j
13.      else
14.          reset t_{j+1 j} = 1
15.          q_{j+1} = 0
16.          f = Q_j T̂(2 : j+1, 1 : j)^{-1} e_j
17.          save f and check deflation and breakdown
18.      end if
19.  end for
```

Remark **5.2.** Note that at Line 17, if f belongs to the subspace spanned by previously saved f vectors, then the algorithm encounters breakdown and terminates. Otherwise, there is a deflation at step j; after setting $t_{j+1\,j}$ to 1 or any nonzero constant, the algorithm continues. Those saved f vectors are the p_i vectors correspoding to the vector $q_i = 0$ in Algorithm 5.1.

As a practical consideration, the work in the SOAR procedure and the Arnoldi procedure can be divided between the computation of matrix-vector products Ap and Bq and orthogonalization. The two methods cost the same for computing the former, which varies depending on the sparsity and structures of matrices A and B. The costs of orthogonalization for SOAR and Arnoldi are $3n^2N + 3nN$ and $4n^2N + 10nN$, respectively. It indicates that SOAR costs about 25% less floating point operations than Arnoldi. More importantly, the memory requirement of the SOAR procedure is only half of the memory requirement the Arnold procedure.

5.3 SOAR algorithm

We now follow the orthogonal Rayleigh-Ritz approximation procedure to derive a method which approximates a large QEP by a small QEP. Following the standard derivation, to apply Rayleigh-Ritz approximation technique based on the subspace $\mathcal{G}_n(A, B; u)$ with $A = -M^{-1}D$ and $B = -M^{-1}K$, we seek an approximate eigenpair (θ, z), where $\theta \in \mathcal{C}$ and $z \in \mathcal{G}_n(A, B; u)$, by imposing the following orthogonal condition, also called the Galerkin condition,

$$\left(\theta^2 M + \theta D + K\right) z \quad \perp \quad \mathcal{G}_n(A, B; u),$$

or equivalently,

$$v^T \left(\theta^2 M + \theta D + K\right) z = 0 \quad \text{for all } v \in \mathcal{G}_n(A, B; u). \tag{5.15}$$

Since $z \in \mathcal{G}_n(A, B; u)$, it can be written as

$$z = Q_m g, \tag{5.16}$$

where the $N \times m$ matrix Q_m is an orthonormal basis of $\mathcal{G}_n(A, B; u)$ generated by the SOAR procedure, and g is an m vector and $m \leq n$. When there are deflations, $m < n$. By (5.15) and (5.16), it yields that θ and g must satisfy the reduced QEP:

$$\left(\theta^2 M_m + \theta D_m + K_m\right) g = 0 \tag{5.17}$$

with

$$M_m = Q_m^T M Q_m, \quad C_m = Q_m^T D Q_m, \quad K_m = Q_m^T K Q_m. \tag{5.18}$$

The eigenpairs (θ, g) of (5.17) define the *Ritz pairs* $(\theta,\ z)$. The Ritz pairs are approximate eigenpairs of the QEP (1.1). The accuracy of the approximate eigenpairs (θ, z) can be assessed by the norms of the residual vectors

$$r = (\theta^2 M + \theta D + K) z.$$

By explicitly formulating the matrices M_m, D_m and K_m, essential structures of M, D and K are preserved. As a result, essential spectral properties of the QEP will be preserved.

The following algorithm is a description of the Rayleigh-Ritz projection procedure based on $\mathcal{G}_n(A, B; u)$ for solving the QEP (1.1) directly.

Algorithm 5.3 (SOAR Method for Solving the QEP Directly)**.**

1. Run SOAR procedure with $A = -M^{-1}D$ and $B = -M^{-1}K$ and a starting vector u to generate an $N \times m$ orthogonal matrix Q_m whose columns span an orthonormal basis of $\mathcal{G}_n(A, B; u)$.

2. Compute M_m, C_m and K_m as defined in (5.18).

3. Solve the reduced QEP (5.17) for (θ, g) and obtain the Ritz pairs (θ, z), where $z = Q_m g / \|Q_m g\|_2$.

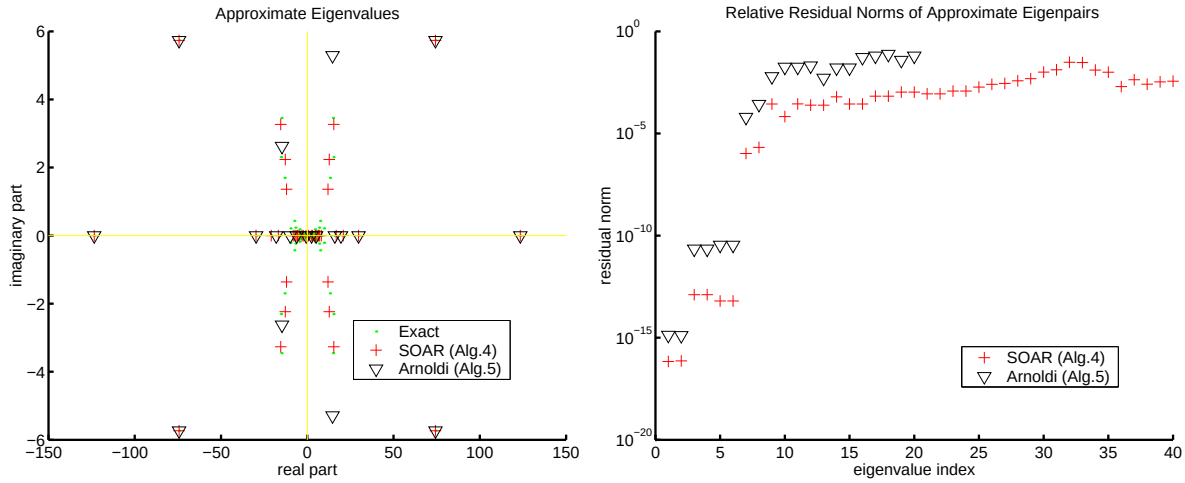

Figure 5.1: Random gyroscopic QEP, exact and approximate eigenvalues (left) and relative residual norms (right) (Example 1).

4. Test the accuracy of Ritz pairs $(\theta,\ z)$ as approximate eigenvalues and eigenvectors of the QEP (1.1) by the relative norms of residual vectors:

$$\frac{\|r\|_2}{|\theta|^2\|M\|_1 + |\theta|\|C\|_1 + \|K\|_1} \tag{5.19}$$

Remark **5.3.** At step 3, to solve the small QEP (5.17), we transform it to a generalized eigenvalue problem in the form of (2.2), and use the direct QEP solver (Algorithm 2.1)

Example 5.1. This example is to show that the convergence rate of SOAR method and Arnoldi method are comparable. However, SOAR method and hybrid method preserve the essential properties of the QEP. Specifically, M, C and K are chosen as 200×200 random matrices with the elements chosen from a normal distribution with mean zero, variance one and standard deviation one. Furthermore, $M^T = M > 0$, $C^T = -C$ and $K^T = K > 0$, as one encounters in a gyroscopic dynamical system. The gyroscopic system is a widely studied system. There are many interesting properties associated with such a system. For example, it is known that the distribution of the eigenvalues of the system in the complex plane is symmetric with respect to both the real and imaginary axes.

The left plot of Figure 5.1 shows the approximate eigenvalues computed by all three algorithms with $n = 20$. The right plot of Figure 5.1 shows the relative residual norms returned by the three algorithms. This example shows that SOAR method preserves the gyroscopic spectral property. Furthermore, the residual norms indicate that the SOAR method has a slight better convergence rate.

6 Further reading

A survey of the quadratic eigenvalue problem can be found in

- F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Review.* vol. 43, pp 234-286, 2001.

The study of the linearization technique in a general setting can be found in

- D. Mackey, N. Mackey, C. Mehl and V. Mehrmann, Vector spaces of linearizations for matrix polynomials, SIAM J. Matrix Anal. Appl. 28:971-1004, 2006

The following recent work to focus on the influence of the linearization process on the accuracy and stability of the computed solution, via a simple model QEP:

- N. J. Higham, D. S. Mackey, F. Tisseur and S. D. Garvey, Scaling, sensitivity and stability in the numerical solution of quadratic eigenvalue problems, Int. J. Numer. Meth. Engng, 73:344-360, 2008

The Q-Arnoldi method is discussed in the following paper:

- K. Meerbergen. The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem. SIAM J. Matrix Anal. Appl. 30:1463-1482, 2008.

The second-order Krylov subspace approach is presented in

- Z. Bai and Y. Su, 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, 2005

A collection of quadratic and other nonlinear eigenvalue problems can be founded in

- T. Betcke, N. J. Higham, V. Mehrmann, C. Schroder and F. Tisseur. NLEVP: A collection of nonlinear eigenvalue problems. Technique report MIMS EPrint 2008.40. School of Mathematics, The University of Manchester, 2008.

  `http://www.manchester.ac.uk/mims/eprints`.