Noname manuscript No.
(will be inserted by the editor)

Detecting a hyperbolic quadratic eigenvalue problem by using a subspace algorithm

Marija Miloloža Pandur

Received: date / Accepted: date

Abstract We consider the quadratic eigenvalue problem (QEP) $\mathbf{Q}(\lambda)x := (\lambda^2 M + \lambda D + K)x = 0$. A Hermitian QEP is hyperbolic if M is positive definite and $(x^H Dx)^2 - 4(x^H Mx)(x^H Kx) > 0$ for all nonzero vectors x. Although there exist many algorithms for detecting hyperbolicity, most of them are not suitable for large QEPs. Motivated by this, we propose a new basic subspace algorithm for detecting large hyperbolic QEPs. Furthermore, we propose a specialized algorithm and its preconditioned variant. Our algorithms can be easily adapted to detect a large overdamped QEP (a hyperbolic QEP with D positive definite and K positive semidefinite). Numerical experiments demonstrate the efficiency of our specialized algorithms.

Keywords quadratic eigenvalue problem · hyperbolic · overdamped · gap · subspace algorithm · LOBPeCG

Mathematics Subject Classification (2000) 15A18 · 65F15 · 65F30

Acknowledgements The author would like to thank Professor Krešimir Veselić for giving the idea for this paper and for valuable discussion. She would also like to thank Professors Françoise Tisseur and Heinrich Voss for providing the MATLAB codes of the algorithms in [7] and [18], respectively.

1 Introduction

We consider the quadratic eigenvalue problem

$$\mathbf{Q}(\lambda)x := (\lambda^2 M + \lambda D + K)x = 0 \tag{1.1}$$

where M, D, K are complex matrices of order n, $x \neq 0$ is the eigenvector, λ the corresponding eigenvalue. The pair (λ, x) is called an eigenpair.

M. Miloloža Pandur
Department of Mathematics, University of Osijek, Trg Ljudevita Gaja 6, 31000 Osijek, Croatia
Tel.: +385-31-224819
E-mail: mmiloloz@mathos.hr

The standard approach to solving a QEP of order n is to use an appropriate linearization of order $2n$ and solve the corresponding generalized eigenvalue problem (GEP). Two common linearizations of $\mathbf{Q}(\lambda)$ from (1.1) are as follows:

$$\begin{bmatrix} 0 & I \\ -K & -D \end{bmatrix} - \lambda \begin{bmatrix} I & 0 \\ 0 & M \end{bmatrix} \tag{1.2}$$

and

$$\begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix} - \lambda \begin{bmatrix} D & M \\ M & 0 \end{bmatrix} =: A - \lambda B. \tag{1.3}$$

Definition 1.1 The QEP (1.1) and the corresponding quadratic matrix polynomial $\mathbf{Q}(\lambda)$ are called: *Hermitian* if M, D and K are all Hermitian; *hyperbolic* if they are Hermitian, M is positive definite and

$$\mathbf{d}(x) := (x^H Dx)^2 - 4(x^H Mx)(x^H Kx) > 0, \quad \text{for all nonzero } x \in \mathbb{C}^n; \tag{1.4}$$

overdamped if they are hyperbolic, D is positive definite and K is positive semidefinite.

The Hermitian $\mathbf{Q}(\lambda)$ from (1.1) with M positive definite is hyperbolic if and only if the Hermitian matrix pencil (1.3) is definite [11, Theorem 3.6], meaning that some real linear combination of the matrices A and B is positive definite. A hyperbolic QEP appears for example in dynamic analysis of structural mechanical systems, such as a damped mass-spring oscillator [19,23]. More about applications of QEPs can be found in [20].

Definition 1.2 Let a matrix $U \in \mathbb{C}^{n \times p}$ have full column rank. The quadratic matrix polynomial

$$U^H \mathbf{Q}(\lambda) U = \lambda^2 U^H MU + \lambda U^H DU + U^H KU$$

is called a *compressed*, or more precisely *U-compressed*, quadratic matrix polynomial for the given $\mathbf{Q}(\lambda)$ from (1.1). Finding scalars λ and nonzero vectors $y \in \mathbb{C}^p$ such that $U^H \mathbf{Q}(\lambda) U y = 0$ holds is called a *compressed*, or more precisely *U-compressed*, QEP for the given QEP (1.1).

For the given Hermitian QEP we are interested in detecting if it is hyperbolic. In this paper, we propose new subspace algorithms for detecting a hyperbolic QEP. Proposed subspace algorithms are based on iterative testing of small compressed QEPs formed by using search subspaces of small dimensions.

Here we briefly list some algorithms for detecting hyperbolicity (overdampedness), where n is the order of a given Hermitian $\mathbf{Q}(\lambda)$. In [21], Veselić proposes a J-Jacobi method for detecting the definiteness of the particular symmetric linearization of the $\mathbf{Q}(\lambda)$, which is equivalent to the overdampedness of the given $\mathbf{Q}(\lambda)$. This test requires the initial $11n^3/3$ flops and then at most $12sn^3$ flops, where s is the number of the performed sweeps. In [11], Higham, Tisseur and Van Dooren propose the level set algorithm. It computes all $4n$ eigenvalues of a QEP with complex matrices of order $2n$ and then checks some simple conditions to detect the (non)definiteness of the matrix pencil (1.3). In [8, Algorithm 1], Guo and Lancaster propose an algorithm for detecting hyperbolicity by computing all $2n$ eigenvalues of the given QEP. It is more efficient than the one proposed in [11]. Furthermore, they propose an algorithm for detecting overdampedness in [8, Sect. 4]; it also detects hyperbolicity after proper shifting of $\mathbf{Q}(\lambda)$. That algorithm finds two solutions (the so-called solvents) of the corresponding matrix equation of order n using a matrix iteration [8, Algorithm 2]; about $19n^3/3$ flops are needed for one iteration. Then at most one extremal eigenvalue of each solvent must be computed. In [6], Guo, Higham and Tisseur accelerate the algorithm given in [8, Sect.4]: by using the same matrix iteration, but without the computation of the solvents. It requires at most $5n^3/3$ flops for the preprocessing step, and $n^3/3$ for $k = 0$ or roughly $20kn^3/3$ flops for $k \geq 1$, where k is the number of iterations in the overdamping test [6, Algorithm 5.1]. In [7], Guo, Higham and Tisseur propose an improved arc algorithm for detecting a definite Hermitian matrix pencil and its modification to detect a hyperbolic quadratic. In [18], Niendorf and Voss propose the hyperbolicity test. More details about algorithms in [7,18] are given in Sect. 4.2. Finally, in [2], Ali proposes a bisection-like method for detecting hyperbolicity of a given banded QEP, which is suitable for large banded QEPs.

Most of the aforementioned algorithms are not suitable for large QEPs. However, our algorithms are particularly suitable for large QEPs.

This paper is organized as follows. In Sect. 2 we recall some results on hyperbolic QEPs. In Sect. 3 we propose a basic subspace algorithm for detecting hyperbolicity (Algorithm 3.1) of a Hermitian QEP. Furthermore, we propose a specialized algorithm for detecting hyperbolicity (Algorithm 3.2) and its modification which uses the preconditioners, both based on the algorithms proposed in [15]. Sect. 4 contains numerical experiments. Some concluding remarks are given in Sect. 5.
We use the following notation: $A \succ 0$ ($\succeq 0$) means that A is a Hermitian positive definite (positive semidefinite) matrix; $A \prec 0$ ($\preceq 0$) means that A is a Hermitian negative definite (negative semidefinite) matrix, and I_n is an identity matrix of order n.

2 Preliminaries

Here we list some properties of a hyperbolic $\mathbf{Q}(\lambda)$ that are needed for the understanding of our algorithms. It is known that Hermitian $\mathbf{Q}(\lambda)$ with a positive definite leading coefficient matrix is hyperbolic if and only if there exists a *shift* $\lambda_0 \in \mathbb{R}$ such that $\mathbf{Q}(\lambda_0)$ is a negative definite matrix [16, Lemmas 31.15 and 31.23].

Theorem 2.1 ([8,22]) *Let $\mathbf{Q}(\lambda)$ of order n be hyperbolic. Then its eigenvalues are all real. Denote its eigenvalues by $\lambda_i^{\pm}$ and arrange them in the order of*

$$\lambda_n^- \leq \cdots \leq \lambda_1^- < \lambda_1^+ \leq \cdots \leq \lambda_n^+. \tag{2.1}$$

Then

a) $\mathbf{Q}(\lambda) \prec 0$ *for all* $\lambda \in (\lambda_1^-, \lambda_1^+)$*;*
b) $\mathbf{Q}(\lambda)$ *is overdamped if and only if* $\lambda_n^+ \leq 0$.

c) *(Cauchy-type interlacing inequalities) Let* $U \in \mathbb{C}^{n\times p}$ *such that* $\operatorname{rank} U = p$. *Denote the eigenvalues of the hyperbolic* $U^H\mathbf{Q}(\lambda)U$ *by* $\mu_i^{\pm}$ *and arrange them in the order of*

$$\mu_p^- \le \cdots \le \mu_1^- < \mu_1^+ \le \cdots \le \mu_p^+. \tag{2.2}$$

Then

$$\lambda_i^+ \le \mu_i^+ \le \lambda_{i+n-p}^+ \quad \text{for} \quad i = 1,\ldots,p, \tag{2.3}$$

$$\lambda_j^- \ge \mu_j^- \ge \lambda_{j+n-p}^- \quad \text{for} \quad j = 1,\ldots,p. \tag{2.4}$$

Definition 2.1 Let $\mathbf{Q}(\lambda)$ be hyperbolic. The open interval $(\lambda_1^-, \lambda_1^+)$ from (2.1) is called the *gap*. Every λ_i^+ (λ_i^-) from (2.1) is called an eigenvalue of a *positive* (*negative*) type.

Now we recall the definition of a weakly hyperbolic QEP, which generalizes a hyperbolic QEP.

Definition 2.2 The QEP (1.1) and the corresponding quadratic matrix polynomial $\mathbf{Q}(\lambda)$ are called: *weakly hyperbolic* if they are Hermitian, M is positive definite and

$$\gamma := \min_{\|x\|_2=1} \left((x^HDx)^2 - 4(x^HMx)(x^HKx)\right) \ge 0, \quad x \in \mathbb{C}^n; \tag{2.5}$$

weakly overdamped if they are weakly hyperbolic, D is positive definite and K is positive semidefinite.

Suppose that $\mathbf{Q}(\lambda)$ is Hermitian with a positive definite leading coefficient matrix and $\mathbf{Q}(\lambda) \ne 0$ for all λ. Then, $\mathbf{Q}(\lambda)$ is weakly hyperbolic if and only if there exists $\lambda_0 \in \mathbb{R}$ such that $\mathbf{Q}(\lambda_0)$ is a negative semidefinite matrix [16, Corollary 31.8 and Lemma 31.23].

Theorem 2.2 ([16,22]) *Let* $\mathbf{Q}(\lambda)$ *of order n be weakly hyperbolic.*

(a) In (2.5)*, if* $\gamma = 0$*, then* $\mathbf{Q}(\lambda)$ *has* $2n$ *real eigenvalues that can be of order* $\lambda_n^- \le \cdots \le \lambda_1^- = \lambda_1^+ \le \cdots \le \lambda_n^+$.
(b) $\mathbf{Q}(\lambda_1^+) \preceq 0$.
(c) The Cauchy-type interlacing inequalities hold.

We give two more statements about a hyperbolic $\mathbf{Q}(\lambda)$. Corollary 2.1 is a direct consequence of the Cauchy-type interlacing inequalities. Proposition 2.1 is a counterpart of [13, Theorem 3.10] for a definite matrix pencil.

Corollary 2.1 *If* $\mathbf{Q}(\lambda)$ *of order n is hyperbolic, then for every* $U \in \mathbb{C}^{n\times p}$ *with* $\operatorname{rank} U = p$*, the gap of* $U^H\mathbf{Q}(\lambda)U$ *contains the gap of* $\mathbf{Q}(\lambda)$.

Proposition 2.1 *Let* $\mathbf{Q}(\lambda)$ *be Hermitian of order n with a positive definite leading coefficient matrix. Let the gaps of hyperbolic quadratics* $U^H\mathbf{Q}(\lambda)U$*, taken for all* $U \in \mathbb{C}^{n\times p}$ *with* $\operatorname{rank} U = p$*, have a non-void intersection* $\mathscr{I}$*. Then* $\mathbf{Q}(\lambda)$ *is hyperbolic, and* $\mathscr{I}$ *is the gap of* $\mathbf{Q}(\lambda)$.

Proof For every $\lambda \in \mathscr{I}$ a matrix $U^H\mathbf{Q}(\lambda)U$ is negative definite. Let $0 \neq u \in \mathbb{C}^n$. Complete u to a matrix U with full column rank. Then $u^H\mathbf{Q}(\lambda)u$ is a diagonal element of the negative definite matrix $U^H\mathbf{Q}(\lambda)U$, and thus the former must be negative. So, $\mathbf{Q}(\lambda)$ is a negative definite matrix for every $\lambda \in \mathscr{I}$. This proves hyperbolicity of the quadratic $\mathbf{Q}(\lambda)$, as well the fact that $\mathscr{I}$ is contained in the gap of $\mathbf{Q}(\lambda)$. Corollary 2.1 implies the equality of these intervals. □

Suppose $\mathbf{Q}(\lambda)$ from (1.1) is hyperbolic. If there exists $x \neq 0$ such that $\mathbf{d}(x) \approx 0$, for $0 < \mathbf{d}(x)$ from (1.4), then $\mathbf{Q}(\lambda)$ is close to a nonhyperbolic one. Indeed, in [11],

$$\phi(M,D,K) := \min_{\|x\|_2=1} \sqrt{\mathbf{d}(x)},$$

is used as a natural measure of the degree of hyperbolicity. There holds

$$\phi(M,D,K) \geq d(M,D,K),$$

where $d(M,D,K)$ is a distance from the hyperbolic $\mathbf{Q}(\lambda)$ to the nearest nonhyperbolic one (for details, see [9,11]). Therefore, small perturbations in the coefficient matrices may cause the loss of hyperbolicity. In [8, Sect. 5], the number $\frac{1}{4}(\lambda_1^- - \lambda_1^+)^2$ is proposed as another measure of the degree of hyperbolicity. If $\lambda_1^- \approx \lambda_1^+$, the hyperbolic $\mathbf{Q}(\lambda)$ is close to the weakly hyperbolic $\widetilde{\mathbf{Q}}(\lambda)$ with $\widetilde{\lambda}_1^- = \widetilde{\lambda}_1^+$.

3 Subspace algorithms

In this section, we propose subspace algorithms for detecting a hyperbolic QEP. Our algorithms make use of three important facts:

i) *a necessary condition*: if a given QEP is hyperbolic, then so is every compressed QEP;
ii) *a sufficient condition*: suppose QEP (1.1) is Hermitian with a positive definite leading coefficient matrix; if $\mathbf{Q}(\lambda_0) \prec 0$ for some $\lambda_0 \in \mathbb{R}$, then the given QEP is hyperbolic;
iii) if a given QEP is hyperbolic, the Cauchy-type interlacing inequalities hold.

First, we propose a basic subspace algorithm: Algorithm 3.1.

Remark 3.1 Some remarks about Algorithm 3.1 are as follows:

- Before forming $\mathbf{d}(u)$, we normalize u such that $\|u\|_2 = 1$. If we find some vector u such that $0 \leq \mathbf{d}(u) \approx 0$, we can terminate our algorithm (lines 5–6). According to the discussion at the end of Sect. 2, this implies that $\mathbf{Q}(\lambda)$ may be weakly hyperbolic. If we find some vector u such that $\mathbf{d}(u) < 0$, then $\mathbf{Q}(\lambda)$ is not hyperbolic by definition (lines 7–8).
- We choose p such that $p \ll n$, so $U_i^H\mathbf{Q}(\lambda)U_i$ is a small quadratic. In order to solve the corresponding small QEP (line 11), one can use the QZ algorithm for the GEP of an appropriate linearization [10,20].

Algorithm 3.1 A basic subspace algorithm for detecting a hyperbolic QEP

Input: Hermitian matrices $M, D, K \in \mathbb{C}^{n\times n}$ such that M is positive definite; an initial basis matrix $U_0 \in \mathbb{C}^{n\times p}$ such that rank $U_0 = p$; tolerance ε;
Output: λ_0 such that $\mathbf{Q}(\lambda_0)$ is negative definite if $\mathbf{Q}(\lambda) = \lambda^2 M + \lambda D + K$ is hyperbolic.

1: set $\mathscr{I}_0 = (-\infty, +\infty)$;
2: **for** $i = 0, 1, 2, \ldots$ **do**
3:     **for** $j = 1, 2, \ldots, p$ **do**
4:         form $\mathbf{d}(u) = (u^H Du)^2 - 4(u^H Mu)(u^H Ku)$, where u is the jth column of U_i
5:         **if** $0 \le \mathbf{d}(u) < \varepsilon$ **then**
6:             STOP: $\mathbf{Q}(\lambda)$ may be weakly hyperbolic.
7:         **else if** $\mathbf{d}(u) < 0$ **then**
8:             STOP: $\mathbf{Q}(\lambda)$ is not hyperbolic.
9:         **end if**
10:     **end for**
11:     solve the QEP $U_i^H \mathbf{Q}(\lambda) U_i$;
12:     **if** $U_i^H \mathbf{Q}(\lambda) U_i$ is not hyperbolic **then**
13:         STOP: $\mathbf{Q}(\lambda)$ is not hyperbolic.
14:     **else**
15:         return its gap $\mathscr{I}_{def}$;
16:         intersect $\mathscr{I}_i \leftarrow \mathscr{I}_i \cap \mathscr{I}_{def}$;
17:         **if** $\mathscr{I}_i = \emptyset$ **then**
18:             STOP: $\mathbf{Q}(\lambda)$ is not hyperbolic.
19:         **else if** length($\mathscr{I}_i$)$< \varepsilon$ **then**
20:             STOP: $\mathbf{Q}(\lambda)$ may be weakly hyperbolic.
21:         **else**
22:             let $\mu_i \in \mathscr{I}_i$;
23:             **if** $\mathbf{Q}(\mu_i) \prec 0$ **then**
24:                 STOP: $\mathbf{Q}(\lambda)$ is hyperbolic.
25:             **else**
26:                 form a new subspace span$\,U_{i+1}$ of dimension p;
27:             **end if**
28:         **end if**
29:     **end if**
30: **end for**
31: if hyperbolicity is detected, return $\lambda_0 = \mu_i$ such that $\mathbf{Q}(\lambda_0)$ is negative definite.

- For detecting if a compressed QEP is hyperbolic (lines 12 and 14) we can use [8, Algorithm 1] based on [8, Theorem 3]. Alternatively, we can delete line 11 and check if $U_i^H \mathbf{Q}(\lambda) U_i$ is hyperbolic using the algorithm from [8, Sect. 4].
- If $U_i^H \mathbf{Q}(\lambda) U_i$ in not hyperbolic, then the necessary condition is violated and $\mathbf{Q}(\lambda)$ is not hyperbolic (lines 12–13).
- If $\mathscr{I}_i = \emptyset$, then $\mathbf{Q}(\lambda)$ is not hyperbolic (lines 17–18) by Corollary 2.1.
- We can simply choose μ_i as the middle of the interval $\mathscr{I}_i$ (line 22).
- If $\mathbf{Q}(\mu_i) \prec 0$ for some some real μ_i, then $\mathbf{Q}(\lambda)$ is hyperbolic by the sufficient condition (lines 23–24). The negative definiteness of $\mathbf{Q}(\mu_i)$ is easily checked by computing the Cholesky factorization, which is a numerically stable algorithm, of the matrix $-\mathbf{Q}(\mu_i)$.
- The dimension of search subspaces can vary for different i, but using the small fixed dimension p assures fixed memory requirements and the fixed numerical cost per iteration for solving a QEP of order p.

- If the length of the current approximation interval $\mathscr{I}_i$ is very small (line 19), this signals that in the case of the given hyperbolic $\mathbf{Q}(\lambda)$, the gap is even smaller. Therefore, such $\mathbf{Q}(\lambda)$ is close to the weakly hyperbolic quadratic with $\gamma = 0$, and iterating further will not provide any more useful information if the tolerance is approximately equal to the unit roundoff.
- We can terminate our algorithm if the iteration index i reaches a predetermined value $i_{\max}$ (e.g. $i_{\max} = 100$). Based on our numerical experiments (only part of them is presented in Sect. 4), this often indicates that the given QEP is close to the weakly hyperbolic quadratic with $\gamma = 0$. Indeed, by decreasing the lengths of the gaps of the hyperbolic QEPs, we notice an increase in the total number of iterations to terminate Algorithm 3.2.

A nontrivial question is how to choose new search subspaces in our subspace algorithm. If we want the gaps of the hyperbolic compressed quadratics to shrink as i increases, we should reuse some information from the previous steps (see Proposition 3.1). Furthermore, since we are solving the whole small compressed QEP, we can apply the Rayleigh-Ritz procedure [15, Algorithm 7.1] and use Ritz values as approximations to some eigenvalues of the original QEP. Since we aim to find a shift from the gap $(\lambda_1^-, \lambda_1^+)$ of a hyperbolic QEP, then it is appropriate to apply the Rayleigh-Ritz procedure by extracting inner eigenvalues. In [15], Liang and Li propose several subspace algorithms. These algorithms compute a few largest or a few smallest eigenpairs of the given type of a large hyperbolic $\mathbf{Q}(\lambda)$. We propose a combination of two aforementioned algorithms that compute eigenpairs closest to the gap of the given hyperbolic $\mathbf{Q}(\lambda)$, namely the Locally Optimal Block Extended Conjugate Gradient method (LOBeCG) from [15, Sect. 11.2] and our Algorithm 3.1, to get a specialized subspace algorithm for detecting a hyperbolic QEP; see Algorithm 3.2.

The subspace $\mathscr{K}_m(\mathbf{Q}(\mu),x) := \operatorname{span}\{x, \mathbf{Q}(\mu)x, \ldots, [\mathbf{Q}(\mu)]^{m-1}x\}$ is called the mth Krylov subspace of a matrix $\mathbf{Q}(\mu)$ on a vector x. Notice that for a Ritz pair (μ, x), the vector $\mathbf{Q}(\mu)x$ is a residual vector for the given QEP, and is contained in the mth Krylov subspace for $m \geq 2$. Instead of using Krylov subspace $\mathscr{K}_m(\mathbf{Q}(\mu_{i;j}^{\pm}), x_{i;j})$ in (3.1), we can use the preconditioned Krylov subspace

$$\mathscr{K}_m(T_i\mathbf{Q}(\mu_{i;j}^{\pm}), x_{i;j}) = \operatorname{span}\{x_{i;j}, T_i\mathbf{Q}(\mu_{i;j}^{\pm})x_{i;j}, \ldots, [T_i\mathbf{Q}(\mu_{i;j}^{\pm})]^{m-1}x_{i;j}\},$$

where $T_i = -\mathbf{Q}(\mu_i)^{-1}$ is an indefinite preconditioner in the ith iteration step, and get *the Locally Optimal Block Preconditioned Extended Conjugate Gradient (LOBPeCG) algorithm for detecting a hyperbolic QEP*. We will refer to this algorithm as *the preconditioned variant of Algorithm 3.2*. For more details about generating a basis matrix for the subspace (3.1) or the version with the preconditioners, we refer the interested readers to [15, Sect. 10–11]; a Modified Gram-Schmidt (MGS) procedure is used, and some form of reorthogonalization is recommended, for example, see [5]. Using definite preconditioners in [15, Sect. 12] significantly accelerates the LOBPeCG algorithms compared to LOBeCG algorithms. Using indefinite preconditioners, not necessarily in every iteration step, in the preconditioned variant of our Algorithm 3.2 can accelerate the detection of (non)hyperbolicity; see Sect. 4.1.

Algorithm 3.2 LOBeCG for detecting a hyperbolic QEP

Input: Hermitian matrices M, D, $K \in \mathbb{C}^{n\times n}$ such that M is positive definite; integers k, $k_\pm$ and an initial approximation $X_0 \in \mathbb{C}^{n\times k}$ such that rank $X_0 = k$, $k \geq k_\pm \geq 1$; an integer $m \geq 2$; tolerance ε;
Output: λ_0 such that $\mathbf{Q}(\lambda_0)$ is negative definite if $\mathbf{Q}(\lambda) = \lambda^2 M + \lambda D + K$ is hyperbolic.

for $j = 1, 2, \ldots, k$ **do**
  form $\mathbf{d}(x_0) = (x_0^H D x_0)^2 - 4(x_0^H M x_0)(x_0^H K x_0)$, where x_0 is the jth column of X_0
  if $0 \leq \mathbf{d}(x_0) < \varepsilon$ **then**
    STOP: $\mathbf{Q}(\lambda)$ may be weakly hyperbolic.
  else if $\mathbf{d}(x_0) < 0$ **then**
    STOP: $\mathbf{Q}(\lambda)$ is not hyperbolic.
  end if
end for
solve the QEP $X_0^H \mathbf{Q}(\lambda) X_0$;
if $X_0^H \mathbf{Q}(\lambda) X_0$ is not hyperbolic **then**
  STOP: $\mathbf{Q}(\lambda)$ is not hyperbolic.
else
  return its eigenpairs $(\mu_{0;j}^\pm, y_j^\pm)$, $j = 1, \ldots, k_\pm$;
  $X_0 \leftarrow X_0[y_{k_-}^-, \ldots, y_1^-, y_1^+, \ldots, y_{k_+}^+]$, $X_{-1} = []$, $\mathbb{J} = \{1 \leq j \leq k_\pm\}$, $\mu_0 = (\mu_{0;1}^- + \mu_{0;1}^+)/2$;
  for $j = 1, 2, \ldots, k_- + k_+$ **do**
    form $\mathbf{d}(x_0) = (x_0^H D x_0)^2 - 4(x_0^H M x_0)(x_0^H K x_0)$, where x_0 is the jth column of X_0
    if $0 \leq \mathbf{d}(x_0) < \varepsilon$ **then**
      STOP: $\mathbf{Q}(\lambda)$ may be weakly hyperbolic.
    else if $\mathbf{d}(x_0) < 0$ **then**
      STOP: $\mathbf{Q}(\lambda)$ is not hyperbolic.
    end if
  end for
end if
for $i = 0, 1, 2, \ldots$ **do**
  if $\mathbf{Q}(\mu_i) \prec 0$ **then**
    STOP: $\mathbf{Q}(\lambda)$ is hyperbolic and μ_i is from the gap;
  end if
  compute a basis matrix U_i of the subspace

$$\mathscr{U}_i = \sum_{j \in \mathbb{J}} \mathscr{K}_m(\mathbf{Q}(\mu_{i;j}^\pm), x_{i;j}) + \text{span}\, X_{i-1}, \tag{3.1}$$

  where $x_{i;j}$ is a column of X_i;
  for $j = 1, 2, \ldots, (m+1)(k_- + k_+)$ **do**
    form $\mathbf{d}(u) = (u^H D u)^2 - 4(u^H M u)(u^H K u)$, where u is the jth column of U_i
    if $0 \leq \mathbf{d}(u) < \varepsilon$ **then**
      STOP: $\mathbf{Q}(\lambda)$ may be weakly hyperbolic.
    else if $\mathbf{d}(u) < 0$ **then**
      STOP: $\mathbf{Q}(\lambda)$ is not hyperbolic.
    end if
  end for
  solve the QEP $U_i^H \mathbf{Q}(\lambda) U_i$;
  if $U_i^H \mathbf{Q}(\lambda) U_i$ is not hyperbolic **then**
    STOP: $\mathbf{Q}(\lambda)$ is not hyperbolic.
  else
    return its eigenpairs $(\mu_{i+1;j}^\pm, y_j^\pm)$, $j = 1, \ldots, k_\pm$ and let $X_{i+1} = U_i[y_{k_-}^-, \ldots, y_1^-, y_1^+, \ldots, y_{k_+}^+]$;
    if $\mu_{i+1;1}^+ - \mu_{i+1;1}^- < \varepsilon$ **then**
      STOP: $\mathbf{Q}(\lambda)$ may be weakly hyperbolic.
    else
      let $\mu_{i+1} = (\mu_{i+1;1}^- + \mu_{i+1;1}^+)/2$;
    end if
  end if
end for
if hyperbolicity is detected, return X_i, interval $(\mu_{i;1}^-, \mu_{i;1}^+)$ and $\lambda_0 = \mu_i$ such that $\mathbf{Q}(\lambda_0)$ is negative definite.

The following proposition assures a monotonicity property of our subspace algorithms.

Proposition 3.1 *If the ith step of Algorithm 3.2 (or its preconditioned variant) is executed then*

$$\mu^{+}_{i+1;j} \le \mu^{+}_{i;j} \quad \textit{for} \quad j = 1, \dots, k_+, \tag{3.2}$$

$$\mu^{-}_{i;j} \le \mu^{-}_{i+1;j} \quad \textit{for} \quad j = 1, \dots, k_-. \tag{3.3}$$

Proof Let $U_{i-1} = [X_{i-1}, W_{i-1}]$. Hence for $j = 1, \dots, k_\pm$, $(\mu^{\pm}_{i;j}, y^{\pm}_j)$ are inner eigenpairs of the hyperbolic $U^H_{i-1}\mathbf{Q}(\lambda)U_{i-1}$ and X_i is the corresponding matrix of Ritz vectors. Therefore, $U_i = [X_i, W_i]$ and $\mu^{\pm}_{i+1;j}$ for $j = 1, \dots, k_\pm$ are inner eigenvalues of the hyperbolic $U^H_i\mathbf{Q}(\lambda)U_i$. For $U = [e_1, \dots, e_r]$, where e_j is the jth column of the identity matrix and $r = k_- + k_+$,

$$U^H(U^H_i\mathbf{Q}(\lambda)U_i)U = X^H_i\mathbf{Q}(\lambda)X_i$$

is a compressed quadratic of the hyperbolic quadratic $U^H_i\mathbf{Q}(\lambda)U_i$. Since $\mu^{\pm}_{i;j}$ are inner eigenvalues of the hyperbolic $X^H_i\mathbf{Q}(\lambda)X_i$, by using the Cauchy-type interlacing inequalities we obtain (3.2) and (3.3). □

After detecting hyperbolicity, we can delete the lines corresponding to the detection of hyperbolicity and continue with Algorithm 3.2 or with its preconditioned variant (using $T = -\mathbf{Q}(\lambda_0)^{-1} \succ 0$). Then the intervals $(\mu^{-}_{i+1;1}, \mu^{+}_{i+1;1})$ shrink as i increases. These intervals converge to the gap, that is, convergence of $\mu^{+}_{i+1;1}$ and $\mu^{-}_{i+1;1}$ to some eigenvalue is proven in [15, Sect. 11] for $k = k_+ = 1$, $k_- = 0$ and $k_+ = 0$, $k = k_- = 1$, respectively; but convergence to the extremal eigenvalues $\lambda^{\pm}_1$ is guaranteed only if $\lambda^{-}_2 < \mu^{-}_{i+1;1}$ for some i and $\mu^{+}_{i'+1;1} < \lambda^{+}_2$ for some i'. An asymptotic estimate for the convergence of the sequences $\{\mu^{\pm}_{i;1}\}$ is given in [15, Theorems 8.2, 9.1] for a single vector version ($k = 1$).

3.1 Comparison of Algorithm 3.2 with some other algorithms

In [12, Sect. 3.6.2], Keller proposes the method of a coordinate relaxation for detecting a definite matrix pencil, and the author expands that method to a subspace algorithm in [17, Sect. 2]. These methods can be applied to the Hermitian linearization (1.3) in order to detect hyperbolicity of $\mathbf{Q}(\lambda)$. For efficiency, in every iteration of the algorithm in [17], we can test negative definiteness of $\mathbf{Q}(\mu_i)$ of order n instead of positive definiteness of $A - \mu_i B$ of order $2n$.

A modification of the mentioned subspace algorithm for detecting a definite matrix pencil to a subspace algorithm for detecting a hyperbolic $\mathbf{Q}(\lambda)$ is given as Algorithm 3.2. Applying the subspace algorithm directly to $\mathbf{Q}(\lambda)$ has two important advantages over applying the subspace algorithm to its Hermitian linearization. First, we work directly with coefficient matrices of order n of the given $\mathbf{Q}(\lambda)$, instead of using the linearization of order $2n$, which reduces the storage requirement. Second,

Table 3.1 Operation count per iteration of Algorithm 3.2 and the algorithm in [17] applied to the Hermitian linearization. Here, $k = k_- + k_+ \geq 2$.

Operations	Algorithm 3.2 Cost (flops)	Algorithm in [17] Cost (flops)
Evaluating $\mathbf{Q}(\mu_i)$	$4n^2$	$4n^2$
Cholesky for $-\mathbf{Q}(\mu_i)$	$n^3/3$	$n^3/3$
Forming a basis matrix for $\mathscr{U}_i$	$6n^2(m-1)k + O(m^3k^2n)$	$16n^2(m+1)k - 8n^2 + O((m+1)^2k^2n)$
Computing one $\mathbf{d}(x)$	$6n^2 + 6n$	$16n^2 + 8n$
Computing compressed matrices	$6n^2(m+1)k + 6n(m+1)^2k^2$	$16n^2(m+1)k + 8n(m+1)^2k^2$
Solving GEP of order $2(m+1)k$ $((m+1)k)$ using the QZ algorithm	$368(m+1)^3k^3$	$46(m+1)^3k^3$
Cholesky for the compressed quadratic (pencil) of order $(m+1)k$	$(m+1)^3k^3/3$	$(m+1)^3k^3/3$
Computing Ritz vectors	$2n(m+1)k^2$	$4n(m+1)k^2$

assuming $k, m \ll n$, the cost per iteration of Algorithm 3.2 is smaller than the cost per iteration of the algorithm in [17] applied to the linearization (1.3): see Table 3.1. In the algorithm in [17], $\mathbf{d}(x)$ refers to computing $x^H Ax$ and $x^H Bx$. If $-\mathbf{Q}(\mu_i)$ is not positive definite, the Cholesky factorization can break down in the early stages of the factorization. The algorithm in [12] has the smallest cost per iteration in comparison to Algorithm 3.2 and the algorithm in [17]; see Table 3.2. Nevertheless, to confirm that a given $\mathbf{Q}(\lambda)$ is hyperbolic, the algorithm in [12] needs to compute λ_1^- and λ_1^+. For large $\mathbf{Q}(\lambda)$ it is very slow. For example, for $\mathbf{Q}(\lambda)$ from Example 4.1 with $n = 2000$ and $\nu = 0.51961525$, the algorithm in [12] applied to (1.3), terminates after 1447 iterations with CPU time 4.19 (cf. Table 4.1).

Notice that operation counts given in Tables 3.1 and 3.2 refer to full matrices. For sparse matrices the cost for evaluating $\mathbf{Q}(\mu_i)$, the Cholesky decomposition for $-\mathbf{Q}(\mu_i)$, forming a basis matrix and new directions, computing $\mathbf{d}(x)$, compressed matrices and Ritz vectors is much smaller.

Table 3.2 Operation count per iteration of the algorithm in [12] applied to the Hermitian linearization.

Operations	Cost (flops)
Computing compressed matrices	$64n^2 + 64n$
Solving two GEPs of order 2	$O(1)$
Computing Ritz vectors	$12n$
Forming new directions	$64n^2 + 28n$

3.2 Subspace algorithms for detecting overdampedness

Overdamped quadratics are an important subset of hyperbolic quadratics, therefore we propose a modification of Algorithm 3.1 into a new algorithm: Algorithm 3.3 for detecting an overdamped QEP.

Algorithm 3.3 A basic subspace algorithm for detecting an overdamped QEP

Input: Hermitian matrices $M, D, K \in \mathbb{C}^{n\times n}$ such that M and D are positive definite, K is positive semidefinite; an initial basis matrix $U_0 \in \mathbb{C}^{n\times p}$ such that rank $U_0 = p$; tolerance ε;
Output: λ_0 such that $\mathbf{Q}(\lambda_0)$ is negative definite if $\mathbf{Q}(\lambda) = \lambda^2 M + \lambda D + K$ is overdamped.

1: set $\mathscr{I}_0 = (-\infty, +\infty)$;
2: **for** $i = 0, 1, 2, \ldots$ **do**
3:     **for** $j = 1, 2, \ldots, p$ **do**
4:         form $\mathbf{d}(u) = (u^H Du)^2 - 4(u^H Mu)(u^H Ku)$, where u is the jth column of U_i
5:         **if** $0 \le \mathbf{d}(u) < \varepsilon$ **then**
6:             STOP: $\mathbf{Q}(\lambda)$ may be weakly overdamped.
7:         **else if** $\mathbf{d}(u) < 0$ **then**
8:             STOP: $\mathbf{Q}(\lambda)$ is not overdamped.
9:         **end if**
10:     **end for**
11:     solve the QEP $U_i^H \mathbf{Q}(\lambda) U_i$;
12:     **if** $U_i^H \mathbf{Q}(\lambda) U_i$ is not overdamped **then**
13:         STOP: $\mathbf{Q}(\lambda)$ is not overdamped.
14:     **else**
15:         return its gap $\mathscr{I}_{def}$;
16:         intersect $\mathscr{I}_i \leftarrow \mathscr{I}_i \cap \mathscr{I}_{def}$;
17:         **if** $\mathscr{I}_i = \emptyset$ **then**
18:             STOP: $\mathbf{Q}(\lambda)$ is not overdamped.
19:         **else if** length($\mathscr{I}_i$)$< \varepsilon$ **then**
20:             STOP: $\mathbf{Q}(\lambda)$ may be weakly overdamped.
21:         **else**
22:             let $\mu_i \in \mathscr{I}_i$;
23:             **if** $\mathbf{Q}(\mu_i) \prec 0$ **then**
24:                 STOP: $\mathbf{Q}(\lambda)$ is overdamped.
25:             **else**
26:                 form a new subspace spanU_{i+1} of dimension p;
27:             **end if**
28:         **end if**
29:     **end if**
30: **end for**
31: if overdampedness is detected, return $\lambda_0 = \mu_i$ such that $\mathbf{Q}(\lambda_0)$ is negative definite.

Remark 3.2 All remarks in Remark 3.1, except the first part in the third remark, apply directly to Algorithm 3.3 by replacing the word “hyperbolic” by the word “overdamped”. An additional remark is as follows.

– For detecting if the compressed QEP is overdamped (lines 12 and 14), we can detect if it is hyperbolic and check if its largest eigenvalue is nonpositive; see Theorem 2.1, item *b)*. If $K \succ 0$, then we can use the two-sided J-Jacobi method [21, Sect. 2] or the implicit J-Jacobi method [21, Sect. 3]. In the case of a nonoverdamped compressed QEP, the J-Jacobi methods terminate when some of the criteria in [21, Sect. 2] are violated.

Similarly to Algorithm 3.3, we can adapt Algorithm 3.2 and its preconditioned variant to detect only overdamped QEPs.

4 Numerical experiments

In Sect. 4.1, we give several numerical experiments to demonstrate the behaviour of Algorithm 3.2 and its preconditioned variant, for $m = 2$ in (3.1) and for different dimensions p of search subspaces. We use $k = k_{\pm} = 1,2,3$, that is, $p = 6,12,18$, respectively. Search subspaces in Algorithm 3.2 are of

> **type A**: $\mathscr{U}_i = \operatorname{span}\{X_i, R_i, X_{i-1}\}$, where R_i is the residual matrix of the current iteration matrix X_i.

In the preconditioned variant of Algorithm 3.2 search subspaces are of

> **type B**: $\mathscr{U}_i = \operatorname{span}\{X_i, T_i R_i, X_{i-1}\}$, where X_i and R_i are as above, while $T_i = \mathbf{Q}(\mu_i)^{-1}$ is the preconditioner used *only* in every fifth iteration (in other iterations, $T_i = I_n$). Indefinite linear systems used for computing the columns of the matrix $T_i R_i$ can be solved only approximately (although, for simplicity of our codes, we use the MATLAB backslash operator).

In some cases, after the MGS in the inner product induced by the positive definite matrix M is applied to the current basis matrix, the final dimension of the search subspace is less than p. We use tolerance $\varepsilon = n \times 10^{-16}$, where n is the order of $\mathbf{Q}(\lambda)$.

In Sect. 4.2, we compare the preconditioned variant of Algorithm 3.2 with **type B** of search subspaces and $p = 6$, with algorithms proposed in [7,18]. The same tolerance ε as above is used in the algorithms from [7,18].

All experiments have been performed in MATLAB R2015a on Intel Core i3-4150 CPU 3.50 GHz, 6 GB RAM. An initial search subspace and an initial vector are randomly chosen (we use vectors with uniformly distributed components) in all our experiments. For the Cholesky factorization we use the MATLAB's built-in function `chol` in our algorithms and in the algorithm proposed in [18], but a hand-written function for the algorithm proposed in [7]. For detecting an overdamped compressed quadratic we use the implicit J-Jacobi method [21], while for detecting a hyperbolic compressed quadratic we use the MATLAB's built-in function `polyeig`.

4.1 Comparison of different dimensions of search subspaces

Example 4.1 (cf. [6–8,18]) Consider the quadratic eigenvalue problem (1.1) with

$$M := I_n, \quad D := \nu \begin{bmatrix} 20 & -10 & & & \\ -10 & 30 & -10 & & \\ & \ddots & \ddots & \ddots & \\ & & -10 & 30 & -10 \\ & & & -10 & 20 \end{bmatrix}, \quad K := \begin{bmatrix} 15 & -5 & & & \\ -5 & 15 & -5 & & \\ & \ddots & \ddots & \ddots & \\ & & -5 & 15 & -5 \\ & & & -5 & 15 \end{bmatrix},$$

where $\nu > 0$ is a real parameter. Notice that all three matrices are positive definite. This QEP comes from a finite element model of a damped mass-spring system and these matrices can be generated by using the NLEVP MATLAB toolbox [3, 4] by the command `nlevp('spring',n,1,10*ones(n,1))`. It is known that for $\nu \leq 0.5196152422$ $\mathbf{Q}(\lambda)$ is not overdamped, and for $\nu \geq 0.5196152423$ it is overdamped. The exact turning value ν^*, for which $\mathbf{Q}(\lambda)$ is weakly overdamped, is somewhere in $(0.5196152422, 0.5196152423)$. For $n = 200, 2000$ and different parameters ν, we compare the algorithm with subspaces of **type A** to the algorithm with subspaces of **type B**, both adapted for overdamped quadratics.

Table 4.1 Comparison of Algorithm 3.2 and its preconditioned variant (given in brackets) for detecting (non)overdamped quadratics from Example 4.1 for $n = 2000$, $p = 6$ and with different values of ν. A number of the attempted Cholesky factorizations (CPU time) is given in the first (second) row for the nonoverdamped and the overdamped cases.

ν	0.2	0.4	0.5	0.5196	0.519615	0.51961524	0.5196152422
	0	1	3	12 (5)	17 (9)	23 (10)	28 (10)
	0.0002	0.007	0.03	0.14 (0.1)	0.41 (0.18)	0.71 (0.21)	1.02 (0.25)
ν	1	0.53	0.5197	0.519650	0.519616	0.51961525	0.5196152423
	1	1	2	8 (6)	10 (6)	14 (7)	17 (11)
	0.0028	0.0032	0.0116	0.08 (0.06)	0.12 (0.07)	0.23 (0.10)	0.42 (0.27)

Our algorithms detect correctly the type of the quadratics in all cases. Increasing dimension p, for the fixed type of search subspaces, fixed n and ν, does not result in an increase of the total number of the attempted Cholesky factorizations, but it results in an increase in CPU time. For the fixed type of search subspaces and fixed ν the total number of the attempted Cholesky factorizations for $n = 200$ and $n = 2000$ is almost the same. Therefore, in Table 4.1, we give the results only for $n = 2000$ and $p = 6$. For example, for $\nu = 0.519616$ the algorithm with subspaces of **type A** (**type B**) detects overdampedness after the 10th (6th) attempted Cholesky factorization is successfully completed with CPU time 0.12 (0.07). For $\nu = 0.2$ nonoverdampedness is detected by definition. As parameter ν is moving away from ν^*, the total number of iterations decreases. When ν is close to ν^*, the total number of iterations is significantly smaller when using **type B** of search subspaces than when using **type A**.

Example 4.2 In this experiment we use the following method from [14] for generating quadratic matrix polynomials with prescribed eigenpairs. Let (λ_k, v_k), $k = 1,\ldots,2n$ such that

$$\Lambda_1 := \mathrm{diag}(\lambda_1,\ldots,\lambda_n), \Lambda_2 := \mathrm{diag}(\lambda_{n+1},\ldots,\lambda_{2n}), \quad \Lambda_1, \Lambda_2 \in \mathbb{R}^{n\times n},$$
$$V_1 := [v_1,\ldots,v_n], V_2 := [v_{n+1},\ldots,v_{2n}], \quad V_1, V_2 \in \mathbb{R}^{n\times n},$$

V_1, V_2 are nonsingular, $V_1V_1^T = V_2V_2^T$ and $\Gamma := V_1\Lambda_1V_1^T - V_2\Lambda_2V_2^T$ is nonsingular. Then the quadratic matrix polynomial $\mathbf{Q}(\lambda) = \lambda^2 M + \lambda D + K$ with

$$M := \Gamma^{-1}, \quad D := -M(V_1\Lambda_1^2V_1^T - V_2\Lambda_2^2V_2^T)M,$$
$$K := -M(V_1\Lambda_1^3V_1^T - V_2\Lambda_2^3V_2^T)M + D\Gamma D,$$

has eigenpairs (λ_k, v_k), $k = 1, \dots, 2n$. If $0 > \lambda_1 \geq \cdots \geq \lambda_n > \lambda_{n+1} \geq \cdots \geq \lambda_{2n}$, $V_1 V_1^T = V_2 V_2^T$, V_1 and V_2 are nonsingular, then Γ is nonsingular and $\mathbf{Q}(\lambda)$ is overdamped with $K \succ 0$, see [6]. If $\lambda_{\min}(\Lambda_1) > \lambda_{\max}(\Lambda_2)$, V_1 is nonsingular, and $V_2 = V_1 U$ for some orthogonal matrix U, then $\mathbf{Q}(\lambda)$ is hyperbolic, see [1].

We use the following two eigenvalue distributions, as in [6]:

type 1: λ_k, $k = 1, \dots, 2n$, is uniformly distributed in $[-100, -1]$;
type 2: λ_k, is uniformly distributed in $[-100, -6]$ for $k = n+1, \dots, 2n$ and $[-5, -1]$ for $k = 1, \dots, n$.

With random orthogonal matrices U_1, U_2 we take $V_1 = U_1$ and $V_2 = V_1 U_2$. We construct a set of 80 overdamped quadratics of order $n = 500$: 40 of each type; and similar for quadratics of order $n = 1000$. The corresponding matrices in the constructed quadratics are full.

Table 4.2 The minimum, average and maximum number of the attempted Cholesky factorizations, and the average CPU time of Algorithm 3.2 (the first row for fixed p) and its preconditioned variant (the second row for fixed p) for detecting 40 overdamped quadratics of **type 1** from Example 4.2.

	$n = 500$ (0.09)				$n = 1000$ (0.04)			
p	min	mean	max	*CPU*	min	mean	max	*CPU*
6	1	10	36	0.16	1	18	68	0.74
	1	6	16	0.11	1	11	26	0.50
12	1	9	32	0.40	1	17	90	1.30
	1	6	15	0.27	1	7	16	0.57
18	1	8	22	0.71	1	13	52	1.7
	1	6	11	0.46	1	7	16	0.86

Table 4.3 The minimum, average and maximum number of the attempted Cholesky factorizations, and the average CPU time of Algorithm 3.2 for detecting 40 overdamped quadratics of **type 2** from Example 4.2.

	$n = 500$ (1.17)				$n = 1000$ (1.08)			
p	min	mean	max	*CPU*	min	mean	max	*CPU*
6	2	2	3	0.03	2	3	3	0.12
12	2	2	3	0.05	2	2	3	0.12
18	2	2	3	0.08	2	2	3	0.17

Tables 4.2 and 4.3 show the minimum, average, and maximum number of the attempted Cholesky factorizations of our algorithms, adapted for overdamped quadratics, applied to the quadratics of **type 1** and **type 2**, respectively. We also give the average CPU time and the average length of the gap (in brackets). For every fixed p, the first and the second row in Table 4.2 correspond to subspaces of **type A** and **type B**, respectively. There was no need to use the preconditioner for **type 2**, therefore the results in Table 4.3 are given only for subspaces of **type A**. An increase in the dimension of search subspaces gives a smaller average number of iterations of

our algorithms for **type 1**, but with a greater average CPU time. The average number of the attempted Cholesky factorizations, as well as the average CPU time, was smaller when using **type B** of search subspaces than when using **type A**. The results for **type 2** for different p are almost the same. The lengths of the gaps for **type 2** are greater than for **type 1**, which explains the smaller number of the attempted Cholesky factorizations for **type 2** as opposed to **type 1**.

Remark 4.1 Here we list a summary of the conclusions for our Algorithm 3.2 and its preconditioned variant, from experiments in Sect. 4.1.

- We can use our algorithms for quadratics with full and with sparse matrices.
- Detection of (non)hyperbolicity can be very fast, after just several attempted Cholesky factorizations (especially when the quadratic is far from being hyperbolic, or when the gap is relatively large).
- The preconditioned variant of Algorithm 3.2 (which uses the preconditioners e.g. only in every fifth iteration) can have a smaller number of total iterations and shorter CPU time, than Algorithm 3.2.
- An increase in the dimension of search subspaces can reduce the total number of iterations, but it often increases CPU time.

4.2 Comparison with other algorithms

We compare our subspace algorithm with two algorithms that are suitable for large QEPs: the hyperbolicity test [18, Algorithm 2] that uses the safeguarded iteration [18, Algorithm 1] and the arc algorithm [7, Algorithm 2.3] adapted for quadratics [7, Sect. 4.1]. Since in every iteration step all three algorithms use the Cholesky factorization for confirming that a given quadratic is hyperbolic, we compare them with the total number of the attempted Cholesky factorizations. We also compare them in terms of CPU time.

We now describe the hyperbolicity test and the arc algorithm. The safeguarded iteration computes one eigenpair of a Hermitian nonlinear eigenvalue problem that admits variational characterizations of eigenvalues. It converges globally and monotonically if the initial vector is from the allowed subset, and under certain assumptions, quadratically to a simple eigenvalue. In the case of a hyperbolic $\mathbf{Q}(\lambda)$, the hyperbolicity test computes a sequence $\{\sigma_k\}$ of approximations of λ_1^+ using the safeguarded iteration, and checks if $\mathbf{Q}(\sigma)$ is negative definite for some σ which is smaller than the computed approximations (the upper sweep). If the relative distance of the approximations becomes very small and hyperbolicity in not detected, then the hyperbolicity test uses the safeguarded iteration to compute a sequence of approximations of λ_1^- and checks the negative definitness of $\mathbf{Q}(\omega)$, where ω is in the middle of two types of approximations (the lower sweep). The nonhyperbolicity of the given quadratic is detected by finding some nonzero vector for which the inequality (1.4) does not hold or when monotonicity of the approximations is violated. The most expensive part in one iteration from the upper sweep is finding the extremal eigenpair of the matrix $\mathbf{Q}(\sigma_k)$. This is done by using the nonlinear Arnoldi method [18, Algorithm 3] which requires preconditioners. In our experiments, we do not update a preconditioner; we just use one LU factorization in the first iteration of the hyperbolicity test.

For the given Hermitian matrix pencil, the arc algorithm detects if this pencil is definite. This algorithm determines the range of some function, which is a part of the unit circle: an arc or two diametrically opposite points. By using the middle of the current arc, it checks the positive definiteness of some linear combination of the given matrices. The nondefiniteness of a given pencil $A-\lambda B$ is detected by finding some nonzero vector x for which holds $x^H(A+iB)x=0$ or for which the length of the current arc is greater than π. For detecting a hyperbolic $\mathbf{Q}(\lambda)$, the arc algorithm uses the linearization (1.3); but in order to work directly with the given quadratic, it tests the negative definiteness of the matrix $\mathbf{Q}(\beta/\alpha)$ for some real β/α. If β/α is very large, this approach used for detecting hyperbolicity can have numerical instability problems. If negative definiteness is not detected, then a matrix $R_1^{-1}R_2$ must be formed, where R_1 is upper triangular of order k with real, positive diagonal elements, R_2 is of order $k\times n-k$, n is the order of $\mathbf{Q}(\lambda)$ and the corresponding Cholesky factorization breaks down in the $k+1$st step.

The hyperbolicity test and the arc algorithm adapted for quadratics can detect if a given quadratic is close to a weakly hyperbolic one with $\gamma=0$: if the approximations of the eigenvalues $\lambda_1^{\pm}$ are very close to each other and if the length of the arc is very close to π, respectively. We use the following abbreviations in the following experiments:

- SA for our subspace algorithm – a preconditioned variant of Algorithm 3.2,
- HT for the hyperbolicity test, and
- ARC for the arc algorithm.

Example 4.3 We use the same quadratics as in Example 4.1 for $n=2000$. For a fixed value of ν we run SA, adapted for overdamped quadratics, HT and ARC for 50 times. The attempted Cholesky factorizations in ARC are done without pivoting.

All algorithms correctly detect the type of the quadratics. Table 4.4 contains the average number of the attempted Cholesky factorizations (the first row) and average CPU time (the second row). The third row for HT contains the average number of the dimensions of the required search spaces in the nonlinear Arnoldi method. Detection of every quadratic is completed within the upper sweep of HT, and we do not restart the nonlinear Arnoldi method (the maximum dimension is 91). For the fixed value of ν, SA and HT have a very similar average number of the attempted Cholesky factorizations, while ARC has the largest average number. When ν is far away from the critical ν^*, all three algorithms detect the correct type of the quadratics almost immediately. For nonoverdamped quadratics SA and ARC have a very similar average CPU time, while for the overdamped quadratics SA is significantly faster than ARC. The slowest algorithm is HT, due to the most time consuming part: finding the extremal eigenpair of a matrix of the order n in every iteration.

Table 4.4 Comparison of 50 runs of SA, HT and ARC for detecting the (non)overdamped quadratics from Example 4.3 for the fixed value of ν. The average number of the attempted Cholesky factorizations (CPU time) is given in the first (second) row for every algorithm. The average number of dimensions of the required search spaces in HT is given in brackets.

	nonoverdamped						
ν	0.2	0.4	0.5	0.5196	0.519615	0.51961524	0.5196152422
SA	0	1	3	5	10	10	10
	0.004	0.006	0.02	0.04	0.18	0.19	0.18
HT	0	0	1	4	6	8	9
	0.002	0.96	1.66	2.44	2.58	2.50	2.59
	(0)	(29)	(44)	(64)	(69)	(69)	(71)
ARC	1	3	5	10	13	16	18
	0.03	0.03	0.04	0.09	0.12	0.16	0.19
	overdamped						
ν	1	0.53	0.5197	0.51965	0.519616	0.51961525	0.5196152423
SA	1	1	2	4	6	7	11
	0.006	0.002	0.008	0.03	0.06	0.08	0.23
HT	1	1	4	5	6	8	10
	0.38	0.61	2.14	2.22	2.29	2.46	2.54
	(12)	(19)	(59)	(62)	(65)	(69)	(70)
ARC	1	3	6	6	9	12	16
	0.38	0.38	0.41	0.40	0.43	0.46	0.50

Example 4.4 We use the same method for constructing quadratics as in Example 4.2. We construct a set of 80 Hermitian quadratics of order 1000, where λ_k for $k = 1,\ldots,1000$ are normally distributed with mean value -3 and standard variation 1, and λ_k for $k = 1001,\ldots,2000$ are uniformly distributed in $[-106,-6]$, cf. [18, Example 3.2]. With random orthogonal matrices U_1, U_2 we take $V_1 = U_1$ and $V_2 = V_1U_2$. For 26 of these examples $\max_{k=1001,\ldots,2000} \lambda_k < \min_{k=1,\ldots,1000} \lambda_k$, meaning that the corresponding quadratics are hyperbolic.

Table 4.5 The minimum, average and maximum number of the attempted Cholesky factorizations (in the first row) and CPU time (in the second row) of SA, HT and ARC for detecting 80 (non)hyperbolic quadratics from Example 4.4.

	min	mean	max
SA	2	22	46
	0.07	1.0	2.2
HT	2	2	3
	1.0	1.3	2.1
ARC	4	9	13
	19.5	46.6	65.4

SA, HT and ARC correctly detect the type of the quadratics in all cases. The results are shown in Table 4.5. Detection of every quadratic is completed within the upper sweep of HT, and we do not restart the nonlinear Arnoldi method. We use complete pivoting in the attempted Cholesky factorizations in ARC since the results are better compared to the Cholesky factorizations without pivoting. Although SA has the largest average number of the attempted Cholesky factorizations, SA is the fastest in 55 problems, and HT in the remaining 25.

Example 4.5 In this experiment, we use sparse uniformly distributed random matrices of order $n = 10000$ and form sparse positive definite M and K with the following code:

```
dens=0.00005;
M=sprand(n,n,dens);
M=(M*M')/2;M=M+20*speye(n);
M=M/max(max(abs(M)));
```

(and similarly for K). Therefore the elements of M,K are in $[0,1]$ and M,K have approximately 12500 nonzero elements. We use $D = M + \alpha K$ for different values of α. We choose $\alpha = 0.75, 0.9999, 1$ for which the quadratic $\mathbf{Q}(\lambda)$ is nonoverdamped, and $\alpha = 1.25, 1.5$ for which the quadratic $\mathbf{Q}(\lambda)$ is overdamped. For the fixed value of α we run SA, adapted for overdamped quadratics, HT and ARC for 20 times. The attempted Cholesky factorizations in ARC are done without pivoting.

Table 4.6 Comparison of 20 runs of SA, HT and ARC for detecting the (non)overdamped quadratics from Example 4.5 for the fixed value of α. The average number of the attempted Cholesky factorizations (CPU time) is given in the first (second) row for every algorithm. The average number of dimensions of the required search spaces in HT is given in brackets.

	nonoverdamped			overdamped	
α	0.75	0.9999	1	1.25	1.5
SA	0	0	1	1	1
	0.004	0.004	0.04	0.03	0.03
HT	0	0	4	1	1
	0.003	0.003	56.4	52.5	52.6
	(0)	(0)	(71)	(66)	(66)
ARC	2	5	13	1	1
	0.5	0.6	0.8	9.3	9.5

The results are shown in Table 4.6. In all runs, SA and ARC correctly detect nonoverdampedness. For $\alpha < 1$, SA and HT detect (non)overdampedness by definition. In every run of HT for $\alpha = 1$, the first approximation of λ_1^+ is equal to -1. Since $\mathbf{Q}(-1) = 0$, HT uses the lower sweep. In 13 runs HT correctly detects that $\mathbf{Q}(\lambda)$ is not hyperbolic, while in 7 other cases it converges without a decision. For $\alpha \neq 1$ HT correctly detects (non)overdampedness. For $\alpha < 1$ the results for HT and ARC are drastically improved compared to the results for $\alpha = 1$. For $\alpha \geq 1$ SA is convincingly the fastest algorithm.

Remark 4.2 Here we list a summary of the conclusions about three algorithms, i.e. SA, ARC and HT, from experiments in Sect. 4.2.

– In some cases, all three algorithms detect (non)hypebolicity in a similar number of the attempted Cholesky factorizations.

– In most cases, HT has the smallest number of the attempted Cholesky factorizations.
– In most cases, SA has the shortest CPU time.

5 Conclusion

Numerical experiments show that by using a very simple type of search subspaces of very small dimensions, proposed subspace algorithms can detect very quickly the (non)hyperbolicity of a large Hermitian quadratic eigenvalue problem. Algorithm 3.2 can be more efficient with the preconditioners than without them. Our experiments show that algorithms from [7,18] and the preconditioned variant of Algorithm 3.2 can detect (non)hyperbolicity in a similar number of the iterations. In comparison with the algorithms from [7,18], the preconditioned variant of Algorithm 3.2 has the shortest CPU time in almost all experiments.

References

1. Al-Ammari, M., Tisseur, F.: Hermitian matrix polynomials with real eigenvalues of definite type. Part I: Classification. Linear Algebra Appl. **436**, 3954–3973 (2012)
2. Ali, A.T.: A bisection method for the banded hyperbolic quadratic eigenvalue problem. Ph.D. thesis, The University of Texas at Arlington (2016)
3. Betcke, T., Higham, N.J., Mehrmann, V., Schröder, C., Tisseur, F.: NLEVP: A collection of nonlinear eigenvalue problems. URL https://github.com/ftisseur/nlevp
4. Betcke, T., Higham, N.J., Mehrmann, V., Schröder, C., Tisseur, F.: NLEVP: A collection of nonlinear eigenvalue problems. ACM Trans. Math. Software **39**(2), 7:1–7:28 (Feb. 2013)
5. Daniel, J.W., Gragg, W.B., Kaufman, L., Stewart, G.W.: Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization. Math. Comput. **30**(136), 772–795 (1976)
6. Guo, C.H., Higham, N.J., Tisseur, F.: Detecting and solving hyperbolic quadratic eigenvalue problems. SIAM J. Matrix Anal. & Appl. **30**(4), 1593–1613 (2009)
7. Guo, C.H., Higham, N.J., Tisseur, F.: An improved arc algorithm for detecting definite Hermitian pairs. SIAM. J. Matrix Anal. & Appl. **31**(3), 1131–1151 (2009)
8. Guo, C.H., Lancaster, P.: Algorithms for hyperbolic quadratic eigenvalue problems. Math. Comput. **74**(252), 1777–1791 (2005)
9. Hachez, Y., Van Dooren, P.: Elliptic and hyperbolic quadratic eigenvalue problems and associated distance problems. Linear Algebra Appl. **371**, 31–44 (2003)
10. Hammarling, S., Munro, C.J., Tisseur, F.: An algorithm for the complete solution of quadratic eigenvalue problems. ACM Trans. Math. Softw. **39**(3), 18:1–18:19 (2013)
11. Higham, N.J., Tisseur, F., Van Dooren, P.M.: Detecting a definite Hermitian pair and a hyperbolic or elliptic quadratic eigenvalue problem, and associated nearness problems. Linear Algebra Appl. **351–352**, 455–474 (2002)
12. Keller, D.: Divide-et-Impera-Verfahren für das verallgemeinerte Eigenwertproblem. Ph.D. thesis, FernUniversität-Gesamthochschule-Hagen (1994)
13. Kovač-Striko, J., Veselić, K.: Trace minimization and definiteness of symmetric pencils. Linear Algebra Appl. **216**, 139–158 (1995)
14. Lancaster, P.: Inverse spectral problems for semisimple damped vibrating systems. SIAM J. Matrix Anal. & Appl. **29**(1), 279–301 (2007)
15. Liang, X., Li, R.C.: The hyperbolic quadratic eigenvalue problem. Forum of Mathematics, Sigma **3** (2015). DOI https://doi.org/10.1017/fms.2015.14
16. Markus, A.S.: Introduction to the Spectral Theory of Polynomial Operator Pencils, vol. 71. AMS, Providence, RI, USA (1998)
17. Miloloža Pandur, M.: Computing interior eigenvalues and corresponding eigenvectors of definite matrix pairs. Ph.D. thesis, Faculty of Science, Department of Mathematics, University of Zagreb (2016). In Croatian

18. Niendorf, V., Voss, H.: Detecting hyperbolic and definite matrix polynomials. Linear Algebra Appl. **432**, 1017–1035 (2010)
19. Tisseur, F.: Backward error and condition of polynomial eigenvalue problems. Linear Algebra Appl. **309**, 339–361 (2000)
20. Tisseur, F., Meerbergen, K.: The quadratic eigenvalue problem. SIAM Rev. **43**(2), 235–286 (2001)
21. Veselić, K.: A Jacobi eigenreduction algorithm for definite matrix pairs. Numer. Math. **64**, 241–269 (1993)
22. Veselić, K.: Note on Interlacing for Hyperbolic Quadratic Pencils. In: Behrndt J., Förster KH., Trunk C. (eds.) Recent Advances in Operator Theory in Hilbert and Krein Spaces. Operator Theory: Advances and Applications, vol. 198. Birkhäuser Basel, Basel (2010)
23. Veselić, K.: Damped Oscillations of Linear Systems. A Mathematical Introduction, *Lecture Notes in Mathematics*, vol. 2023. Springer, Heidelberg (2011)