Linear Algebra and its Applications 523 (2017) 59–78

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

A fixed-point method for approximate projection onto the positive semidefinite cone

Juliano B. Francisco, Douglas S. Gonçalves *

Department of Mathematics - CCFM - Federal University of Santa Catarina, Florianópolis, Brazil

ARTICLE INFO

Article history:
Received 10 May 2016
Accepted 10 February 2017
Submitted by T. Damm

MSC:
15B48
65Fxx
90C22

Keywords:
Positive semidefinite cone
Projection
Iterative methods
Fixed-point

ABSTRACT

The projection of a symmetric matrix onto the positive semidefinite cone is an important problem with application in many different areas such as economy, physics and, directly, semidefinite programming. This problem has analytical solution, but it relies on the eigendecomposition of a given symmetric matrix which clearly becomes prohibitive for larger dimension and dense matrices. We present a fixed-point iterative method for computing an approximation of such projection. Each iteration requires matrix–matrix products whose costs may be much less than $O(n^3)$ for certain structured matrices. Numerical experiments showcase the attractiveness of the proposed approach for sparse symmetric banded matrices.

Given a $n \times n$ symmetric matrix A, we address the problem of computing the projection of A onto $\mathcal{S}_+^n$, the set of symmetric positive semidefinite matrices, according to the Frobenius norm. It is a relevant problem with application in many different areas such as economy, physics and, directly, semidefinite programming [23,22,11]. This problem can be cast as a quadratic semidefinite programming problem:

* Corresponding author.
E-mail addresses: juliano.francisco@ufsc.br (J.B. Francisco), douglas.goncalves@ufsc.br (D.S. Gonçalves).

http://dx.doi.org/10.1016/j.laa.2017.02.014
0024-3795/

$$\begin{aligned} \min_{X \in \mathcal{S}^n} \ & \frac{1}{2}\|A - X\|_F^2 \\ \text{s.t } & X \succeq 0, \end{aligned} \tag{1}$$

where $\mathcal{S}^n$ stands for the set of symmetric matrices of order n, $\|\cdot\|_F$ denotes the Frobenius norm and $X \succeq 0$ means that

$$X \in \mathcal{S}^n_+ = \{Y \in \mathcal{S}^n \,|\, \forall i : \ \lambda_i(Y) \geq 0\},$$

$\lambda_i(Y)$ denoting the i-th eigenvalue of Y.

For the reader interested in projecting a nonsymmetric matrix $A \in \mathbb{R}^{n\times n}$, we underline that

$$\|A - X\|_F^2 = \text{Trace}((A-X)^T(A-X)) = \|\frac{A + A^T}{2} - X\|_F^2 + \|\frac{A - A^T}{2}\|_F^2,$$

for any symmetric matrix X. Hence, in such a case, the objective function of problem (1) should be replaced by $1/2\|(A^T + A)/2 - X\|_F^2$.

It is well-known [21,14] that problem (1) has an analytical solution. Let $A = Q\Lambda Q^\top$ be the spectral decomposition of A. Then, the projection of A onto $\mathcal{S}^n_+$, with respect to the Frobenius norm, is given by

$$A_+ = Q\Lambda_+ Q^\top = Q\,\text{diag}(\max\{0, \lambda_1\}, \ldots, \max\{0, \lambda_n\})\,Q^\top, \tag{2}$$

where $\text{diag}(\cdot)$ represents a diagonal matrix.

However, the projection obtained from Eq. (2) requires the eigendecomposition of the symmetric matrix A whose cost is $O(n^3)$ – approximately $9n^3$ if a QR algorithm is applied after reducing A to tridiagonal form [9]; or approximately $4n^3$ if a divide-and-conquer approach is employed [10].

Certainly, the $O(n^3)$ cost turns prohibitive for larger values of n. In that case, in order to conceive affordable algorithms for computing the projection onto the positive semidefinite cone it is necessary to exploit the particular structure of the matrix A and/or sought properties of the projection.

The aim of the paper is to discuss an alternative method for approximately solving (1), as well as its particular usefulness for some sort of matrices, specifically, symmetric banded ones. We proposed a class of fixed-point methods based on polynomial filtering, which present quadratic convergence in the simplest case (a polynomial of degree three). In such a case, this polynomial has already appeared in computational quantum chemistry for purifying the density matrix, by giving rise to the so-called McWeeny purification [17,19]. In addition, it is not hard to show that there is a straight relation between this polynomial of degree three and the one given by the Newton–Schulz iteration for computing the sign function of a matrix [7,15].

This paper is organized as follows. Section 1 introduces the notation and some basic properties of the projection. The polynomial fixed-point iteration is proposed in Section 2

along with the possible choices for the polynomials and their estimated convergence rates. Section 3 brings a formal convergence analysis and theoretical properties useful in defining stopping criteria. Section 4 discuss some implementation issues for the case of symmetric banded matrices and some numerical results are presented in Section 5. Conclusions are drawn in Section 6.

1. Notation and preliminaries

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix and $A = Q\Lambda Q^\top$ its eigendecomposition, where q_i denotes the i-th column of Q. Notice that $A = A_+ + A_-$, where

$$A_+ = \sum_{\lambda_i>0} \lambda_i q_i q_i^\top \qquad A_- = \sum_{\lambda_i<0} \lambda_i q_i q_i^\top, \tag{3}$$

and thus A_+ is the solution of (1).

Based on the simple identity

$$\max\{x, 0\} = \frac{x + |x|}{2},$$

we can relate expression (2) for A_+ with the polar decomposition of A. Consider the polar decomposition [13] of A:

$$A = UH,$$

where U is orthogonal and H is symmetric positive semidefinite.

It is not hard to show that

$$H = \sqrt{A^2} = Q\,\mathrm{diag}(|\lambda_1|, \ldots, |\lambda_n|)\,Q^\top,$$

and thus, the projection of a symmetric A onto $\mathcal{S}_+^n$ is given by

$$A_+ = \frac{A + H}{2}.$$

Although the singular value decomposition is a stable way of obtaining the polar decomposition, its cost is still $O(n^3)$ which does not scale well. Nevertheless, iterative methods for computing the polar decomposition have been proposed in the literature. The one presented in [13], consists in applying Newton's method to $X^2 = I$, which leads to the iteration

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}). \tag{4}$$

In [13], it was shown that if A is nonsingular and $X_0 = A$, then the X_k converges quadratically to the orthogonal polar factor U. With U at hand, the other polar factor is promptly obtained: $H = U^\top A$.

Other numerical schemes have been proposed as an alternative of the Newton iteration (4) in order to avoid matrix inversion, as for example, the ones devised by McWeeny [19,2], Kovarik [18] and Björck and Bowie [6], which in general are based on the Newton–Schulz iteration and its variations:

$$X_{k+1} = \frac{1}{2} X_k \left(3I - X_k^2\right). \tag{5}$$

2. Fixed-point iteration

Inspired by iterative methods for polar decomposition [13], and based on the ideas of [5], we propose a fixed-point iteration for computing the orthogonal projector P_+ such that

$$A_+ = P_+ A.$$

Let $\lambda_1 \geq \cdots \geq \lambda_p > 0 \geq \lambda_{p+1} \geq \cdots \geq \lambda_n$ be the eigenvalues of A in decreasing order. Analytically, P_+ is given by the expression:

$$P_+ = Q \operatorname{diag}(\operatorname{sgn}(\lambda_1), \ldots, \operatorname{sgn}(\lambda_p), 0, \ldots, 0)\, Q^\top,$$

where $\operatorname{sgn}(\cdot)$ is the sign function.

Equivalently, partitioning $Q = [Q_+ \; Q_-]$, where $Q_+ \in \mathbb{R}^{n\times p}$ and $Q_- \in \mathbb{R}^{n\times(n-p)}$, and denoting the identity matrix of order p by I_p, we obtain

$$P_+ = [Q_+ \; Q_-] \begin{bmatrix} I_p & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_+^\top \\ Q_-^\top \end{bmatrix} = Q_+ Q_+^\top.$$

Notice that, in order to obtain P_+, we would like to map the positive eigenvalues of A to one whereas the non-positive eigenvalues are mapped to zero.

Given a symmetric matrix A whose spectrum is denoted by $\sigma(A)$, firstly, we estimate an upper bound

$$\alpha \geq \max_{\lambda\in\sigma(A)} |\lambda| := \rho(A),$$

for $\rho(A)$, the spectral radius of A. This can be accomplished, for instance, by using the Gerschgorin circles' theorem [9,20].

Then, all eigenvalues are shifted and scaled

$$B = \frac{A + \alpha I}{2\alpha},$$

such that the eigenvalues $\bar{\lambda} = (\lambda + \alpha)/2\alpha$ of B are in the interval $[0, 1]$:

$$0 \leq \bar{\lambda}_n \leq \ldots \bar{\lambda}_{p+1} \leq \frac{1}{2} < \bar{\lambda}_p \leq \ldots \bar{\lambda}_1 \leq 1.$$

Further, we remark that,

$$\begin{array}{lll} \bar{\lambda}_i \in [0, 1/2) & \text{if and only if} & \lambda_i \in [-\alpha, 0), \\ \bar{\lambda}_i = 1/2 & \text{if and only if} & \lambda_i = 0, \\ \bar{\lambda}_i \in (1/2, 1] & \text{if and only if} & \lambda_i \in (0, \alpha]. \end{array} \tag{6}$$

Now, the idea is to apply a fixed-point iteration

$$B_{k+1} = \mathcal{P}(B_k), \tag{7}$$

where $B_0 = B$ and $\mathcal{P}$ is an appropriate polynomial. The polynomial $\mathcal{P}$ should be chosen in a way that iteration (7) gradually sends the eigenvalues $\bar{\lambda} < 1/2$ to zero whereas the eigenvalues $\bar{\lambda} > 1/2$ are sent to one. Also, the eigenvalue $\bar{\lambda} = 1/2$ should be a fixed-point of $\mathcal{P}$, thus $\mathcal{P}(1/2) = 1/2$.

Let us denote

$$\widetilde{I} = \begin{bmatrix} I_p & 0 & 0 \\ 0 & (1/2)I_m & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

wherein m is the multiplicity of null eigenvalues. Since $\mathcal{P}(B_k)$ has always the same eigenvectors of A, it follows that $B_* = \lim_{k\to\infty} B_k = Q\widetilde{I}Q^\top$. Therefore, we have that $A_+ = P_+A = B_*A$. In addition, if A is non-singular, which means that $m = 0$, then $B_* = P_+$, the sought projector.

We remark that iteration (7) only requires matrix–matrix products, and thus, can be attractive for computing the projection onto $\mathcal{S}_+^n$ for matrices with suitable structure, as discussed in Section 4.

We construct here a family of polynomials in order to ensure convergence of the fixed-point iteration (7) to $B_* = Q\widetilde{I}Q^\top$.

Let $\mathcal{P}(t) = p_n(t) = a_0 + a_1t + \ldots + a_nt^n$ be a polynomial of degree n. We assume that $t_1^* = 0$, $t_2^* = 1/2$ and $t_3^* = 1$ are fixed points of $p_n(t)$. Furthermore, in order to ensure convergence of order at least s, we also assume that

$$p_n^{(r)}(0) = p_n^{(r)}(1) = 0 \quad \text{for} \quad r = 1, \ldots, s-1 \tag{8}$$

where $s > 1$ and $p_n^{(r)}$ denote the r-th derivative of p_n. Therefore, we have that $a_0 = a_1 = \ldots = a_{s-1} = 0$ and the remaining coefficients are the solutions of the following linear system with s equations and $n - s + 1$ unknowns:

$$\begin{cases} a_s + a_{s+1} + \ldots + a_n = 1 \\ \dfrac{s!}{(s-r)!}a_s + \dfrac{(s+1)!}{(s-r+1)!}a_{s+1} + \ldots + \dfrac{n!}{(n-r)!}a_n = 0, \quad r = 1, \ldots, s-1. \end{cases} \tag{9}$$

Proposition 2.1 states the order of convergence of the proposed fixed-point iterations.

Proposition 2.1. *Let $t_0 \in \mathbb{R}$ and consider the sequence $\{t_k\}$ generated by the iteration $t_{k+1} = \mathcal{P}(t_k)$, where $\mathcal{P}(t)$ is constructed by solving (9). Suppose that $\{t_k\} \subseteq [a_1, a_2]$, with $[0,1] \subseteq [a_1, a_2]$, and it converges to either $t_* = 0$ or $t_* = 1$. Then, for*

$$c_s = \max_{t \in [a_1, a_2]} |\mathcal{P}^{(s)}(t)|,$$

it follows that $|t_{k+1} - t_*| \leq c_s |t_k - t_*|^s$, that is, the convergence is of order s.

Proof. By applying s times the mean value theorem, and using (8), we obtain

$$\begin{aligned} |t_{k+1} - t_*| &= |\mathcal{P}(t_k) - \mathcal{P}(t_*)| = |\mathcal{P}'(\zeta_k^1)||t_k - t_*| = |\mathcal{P}'(\zeta_k^1) - \mathcal{P}'(t_*)||t_k - t_*| \\ &= |\mathcal{P}''(\zeta_k^2)||\zeta_k^1 - t_*||t_k - t_*| = |\mathcal{P}''(\zeta_k^2) - \mathcal{P}''(t_*)||\zeta_k^1 - t_*||t_k - t_*| \\ &= \ldots = |\mathcal{P}^{(s)}(\zeta_k^s)| \left(\textstyle\prod_{j=1}^{s-1} |\zeta_k^j - t_*|\right) |t_k - t_*|, \end{aligned}$$

wherein $|\zeta_k^s - t_*| \leq |\zeta_k^{s-1} - t_*| \leq |\zeta_k^{s-2} - t_*| \leq \ldots \leq |\zeta_k^1 - t_*| \leq |t_k - t_*|$ and $\zeta_k^s \in \mathbb{R}$. Therefore,

$$|t_{k+1} - t_*| \leq c_s |t_k - t_*|^s. \qquad \square$$

Thus, in order to obtain the highest possible convergence rate, we choose s such that $s = n - s + 1$, that is, $s = (n+1)/2$ or $n = 2s - 1$. For example, for quadratic convergence we need a polynomial $\mathcal{P}$ of degree $n = 3$. If cubic convergence is desired, $\mathcal{P}$ must have degree 5. It is worth to remark that the degree of $\mathcal{P}$ is always odd. Further, in this case, system (9) is squared and can be written as $Fa = e_1$, where $e_1^\top = (1, 0, \ldots, 0)$, $F \in \mathbb{R}^{s \times s}$ with $F_{ij} = 1$ when $i = 1$ and $F_{ij} = (s + j - 1)!/(s - i + j)!$ otherwise. In Fig. 1 we display the polynomials for $s = 2$, $s = 3$ and $s = 6$. Hereafter, denote by

$$\mathcal{F} = \{\mathcal{P}_s(t) \mid s = 2, 3, \ldots\},$$

the family of polynomials generated by solving (9).

Example 1 *(Quadratic convergence).* When $s = 2$, we have the linear system

$$\begin{cases} a_2 + a_3 = 1 \\ 2a_2 + 3a_3 = 0, \end{cases}$$

which leads to the polynomial

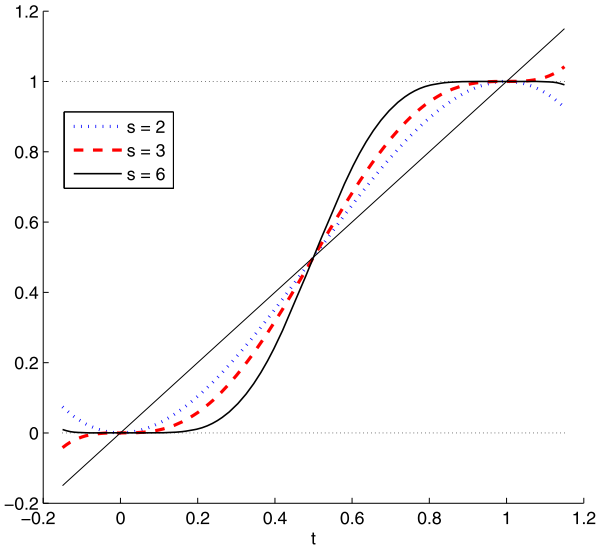

Fig. 1. Polynomials $\mathcal{P}_2(t)$, $\mathcal{P}_3(t)$ and $\mathcal{P}_6(t)$.

$$\mathcal{P}_2(t) = 3t^2 - 2t^3. \tag{10}$$

Example 2 *(Cubic convergence).* When $s = 3$, we have the linear system

$$\begin{cases} a_3 + a_4 + a_5 &= 1 \\ 3a_3 + 4a_4 + 5a_5 &= 0 \\ 6a_3 + 12a_4 + 20a_5 &= 0, \end{cases}$$

which leads to the polynomial

$$\mathcal{P}_3(t) = 10t^3 - 15t^4 + 6t^5. \tag{11}$$

Polynomials (10) and (11) have already appeared in the context of density matrix purification in the electronic structure calculation [17]. Further, with some manipulation, we see that $(2\mathcal{P}_2(t + 1/2) - 1) = \frac{1}{2}t(3 - t^2)$, the well known Newton–Schulz polynomial.

Below we establish the polynomial fixed point iteration for solving (1).

Algorithm 2.1 *(Polynomial fixed point iteration for projection onto $\mathcal{S}_+^n$). Given $A \in \mathbb{R}^{n\times n}$ symmetric, let $\alpha > 0$ such that $\|A\| \leq \alpha$.*
Compute $B = (A + \alpha I)/(2\alpha)$ and set $B_0 = B$.
For $k = 0, 1, 2, \ldots$
Step 1) $A_k = B_k A$.
Step 2) Choose $\mathcal{P}(t) \in \mathcal{F}$.
Step 3) $B_{k+1} = \mathcal{P}(B_k)$.

Some remarks concerning Algorithm 2.1 are in order:

(*i*) A_k is the current approximation of A_+, the projection of A onto $\mathcal{S}_+^n$.
(*ii*) A stop criterion for Algorithm 2.1 is $\|B_{k+1} - B_k\| \leq \hat{\epsilon}$, for some tolerance $\hat{\epsilon} > 0$. The reasoning for this and other stopping rules are discussed in Section 3.
(*iii*) At Step 2, any polynomial in $\mathcal{F}$ could be considered, however, we feel that $\mathcal{P}_2$ and $\mathcal{P}_3$ are more suitable for practical purposes.
(*iv*) Although the sequence $\{A_k\}$ is built at Step 1, in the implementation we skip this step and the approximation $A_* = B_* A$ for A_+ is computed only after convergence of $\{B_k\}$ (or after some stopping criterion is reached).

3. Convergence analysis

Here we present a convergence analysis for Algorithm 2.1 when $\mathcal{P}_2(t)$ or $\mathcal{P}_3(t)$ are used in Step 2. A similar analysis has appeared in [12, pg. 472] for $\mathcal{P}_2(t)$, which has a convergence interval slightly greater than $\mathcal{P}_3(t)$ (namely, $\mathcal{P}_2(t)$ converges in $(1/2 - \sqrt{3}/2, 1/2) \cup (1/2, 1/2 + \sqrt{3}/2)$).

Proposition 3.1. *Let $\{t_k\}_k$ be the sequence generated by $t_{k+1} = \mathcal{P}(t_k)$, with $t_0 \in \mathbb{R}$ and $\mathcal{P}(t) = \mathcal{P}_2(t)$ or $\mathcal{P}(t) = \mathcal{P}_3(t)$. Then,*
(a) if $t_0 \in (1/2 - \sqrt{21}/6, 1/2)$, $\{t_k\}_k$ converges to 0;
(b) if $t_0 \in (1/2, 1/2 + \sqrt{21}/6)$, $\{t_k\}_k$ converges to 1;
(c) if $t_0 = 1/2$, $t_k = 1/2$ for all k.

Proof. Let us define $r_1 = \sqrt{21}/6$. Note that

$$\phi_3(t) = \mathcal{P}_3(t) - t = 12t(t-1)(2t-1)(t - (1/2 + r_1))(t - (1/2 - r_1))$$

and

$$\phi_2(t) = \mathcal{P}_2(t) - t = -t(2t-1)(t-1).$$

Thus, the fixed-points of $\mathcal{P}_3$ are $1/2 - r_1,\ 0,\ 1/2,\ 1$ and $1/2 + r_1$. Further, the fixed-points of $\mathcal{P}_2(t)$ are $0,\ 1/2$ and 1. Therefore, we notice that

(i) $\mathcal{P}_2(t) > t$ for all $t \in (-\infty, 0) \cup (1/2, 1)$;
(ii) $\mathcal{P}_2(t) < t$ for all $t \in (0, 1/2) \cup (1, +\infty)$;
(iii) $\mathcal{P}_2(t) \in (0, 1/2)$ for all $t \in (1/2 - r_1, 1/2)$;
(iv) $\mathcal{P}_2(t) \in (1/2, 1)$ for all $t \in (1/2, 1/2 + r_1)$;
(v) $\mathcal{P}_3(t) > t$ for all $t \in (1/2 - r_1, 0) \cup (1/2, 1) \cup (1/2 + r_1, +\infty)$;
(vi) $\mathcal{P}_3(t) < t$ for all $t \in (-\infty, 1/2 - r_1) \cup (0, 1/2) \cup (1, 1/2 + r_1)$;
(vii) $\mathcal{P}_3(t) \in (1/2 - r_1, 0)$ for all $t \in (1/2 - r_1, 0)$;
(viii) $\mathcal{P}_3(t) \in (0, 1/2)$ for all $t \in (0, 1/2)$;
(ix) $\mathcal{P}_3(t) \in (1/2, 1)$ for all $t \in (1/2, 1)$;
(x) $\mathcal{P}_3(t) \in (1, 1/2 + r_1)$ for all $t \in (1, 1/2 + r_1)$.

Now, whenever $\mathcal{P}(t_k) < t_k$, we have that $t_{k+1} = \mathcal{P}(t_k) < t_k$. By the other hand, if $\mathcal{P}(t_k) > t_k$, it follows that $t_{k+1} = \mathcal{P}(t_k) > t_k$. Therefore, if t^* is a fixed-point of $\mathcal{P}$, $\mathcal{P}(t) < t$ and $\mathcal{P}(t) \in (t^*, t_0]$ for all $t \in (t^*, t_0]$, we have that $\{t_k\}_k$ is decreasing and bounded from below, namely, $\{t_k\}_k$ is convergent. Since

$$\lim_{k\to\infty} t_{k+1} = \mathcal{P}(\lim_{k\to\infty} t_k)$$

we have that

$$\lim_{k\to\infty} t_k = t^*.$$

Analogously, if $\mathcal{P}(t) > t$ and $\mathcal{P}(t) \in [t_0, t^*)$ for all $t \in [t_0, t^*)$, we have that $\lim_{k\to\infty} t_k = t^*$. Therefore, the proof follows from considerations (i) to (x). □

Theorem 3.1. *Let $\{A_k\}_k$ and $\{B_k\}_k$ be the sequences generated by Algorithm 2.1. Then,*

(*i*) *If A is non-singular, $\lim_{k\to\infty} B_k = P_+$. Otherwise, $\lim_{k\to\infty} B_k = B_*$.*
(*ii*) $\lim_{k\to\infty} A_k = A_+$.
(*iii*) *If there exists $k_0 \in \mathbb{N}$ such that $\mathcal{P}(t) = \mathcal{P}_2(t)$ for all $k \geq k_0$, both convergences above are of quadratic order. By the other hand, if $\mathcal{P}(t) = \mathcal{P}_3(t)$ for all $k \geq k_0$, convergence is cubic.*

Proof. To prove (i), we observe that when A is non-singular

$$\lim_{k\to\infty} B_k = B_* = Q\widetilde{I}Q^\top = Q_+Q_+^\top = P_+.$$

Now, notice that $B_{k+1} = \mathcal{P}(B_k) = Q\text{diag}(\mathcal{P}(\bar{\lambda}_1^k), \ldots, \mathcal{P}(\bar{\lambda}_n^k))Q^\top$, that is, the spectral decomposition of B_{k+1} is

$$B_{k+1} = Q\text{diag}(\bar{\lambda}_1^{k+1}, \ldots, \bar{\lambda}_n^{k+1})Q^\top,$$

with $\bar{\lambda}_i^{k+1} = \mathcal{P}(\bar{\lambda}_i^k)$ and $\bar{\lambda}_i^0 = \bar{\lambda}_i$, for $i = 1, \ldots, n$. Now, we have that

$$A_k = Q\text{diag}(\lambda_1\mathcal{P}(\bar{\lambda}_1^k), \ldots, \lambda_n\mathcal{P}(\bar{\lambda}_n^k))Q^\top.$$

Therefore, (ii) follows from Proposition 3.1.

Finally, (iii) follows from Proposition 2.1 and from the fact that

$$\upsilon_1\|B_k - B_*\|_F \leq \|A_k - A_+\|_F \leq \upsilon_2\|B_k - B_*\|_F,$$

where $\upsilon_1 = \min\{|\lambda_i| \mid \lambda_i \neq 0\}$ and $\upsilon_2 = \rho(A)$. □

Next theorem gives us a criterion for stopping Algorithm 2.1 by measuring the distance of two consecutive iterates.

Theorem 3.2. *Let $\{A_k\}_{k\in\mathbb{N}}$ and $\{B_k\}_{k\in\mathbb{N}}$ be the sequences generated by Algorithm 2.1 and $a_1 \leq 0 < 1 \leq a_2$ such that $\sigma(B_k) \subseteq [a_1, a_2]$ for all k. Define*

$$c_s = \max_{t\in[a_1,a_2]} |\mathcal{P}_s^{(s)}(t)|. \tag{12}$$

Given $\epsilon > 0$, if $\mathcal{P} = \mathcal{P}_s$ for all $k \geq k_0$ and

$$\|B_k - B_*\|_F \leq 1/\sqrt[s-1]{2c_s}, \tag{13}$$

then

$$\|B_{k+1} - B_k\|_F \leq \frac{1}{2}\sqrt[s]{\epsilon/c_s}, \tag{14}$$

implies that

$$\|B_{k+1} - B_*\|_F \leq \epsilon \quad \text{and} \quad \frac{\|A_{k+1} - A_+\|_F}{\|A\|_F} \leq \epsilon.$$

Proof. Since for all $k \geq k_0$ the eigenvalues $\{\bar{\lambda}_i^k\}_{i=1}^n$ of matrix B_k are all in an interval $[a_1, a_2]$ (in fact, they are all in $[0, 1]$), from Proposition 2.1,

$$\frac{1}{\rho(A)}\|A_{k+1} - A_+\|_F \leq \|B_{k+1} - B_*\|_F \leq c_s\|B_k - B_*\|_F^s, \tag{15}$$

for all $k \geq k_0$. Thus,

$$\|B_k - B_*\|_F \leq \|B_{k+1} - B_k\|_F + \|B_{k+1} - B_*\|_F \leq \|B_{k+1} - B_k\|_F + c_s\|B_k - B_*\|_F^s.$$

From the previous inequality and (13) and (14), we obtain

$$\frac{1}{2}\|B_k - B_*\|_F \leq (1 - c_s\|B_k - B_*\|_F^{s-1})\|B_k - B_*\|_F \leq \|B_{k+1} - B_k\|_F \leq \frac{1}{2}\sqrt[s]{\epsilon/c_s}.$$

Therefore,

$$\begin{aligned}\|B_{k+1} - B_*\|_F &\leq c_s\|B_k - B_*\|_F^s = c_s\|B_k - B_*\|_F^{s-1}\|B_k - B_*\|_F \\ &\leq \frac{1}{2}\|B_k - B_*\|_F \leq \frac{1}{2}\sqrt[s]{\epsilon/c_s} < \epsilon.\end{aligned}$$

Further, from (15)

$$\frac{\|A_{k+1} - A_+\|_F}{\|A\|_F} \leq \epsilon$$

and the proof is concluded. □

From (10) and (11) we see that $c_s = 6$ when $s = 2$, $c_s = 60$ when $s = 3$ and $[a_1, a_2] = [0, 1]$ in both cases. Consequently, if

$$\|B_k - B_*\|_F \leq \min\{1/12, 1/(2\sqrt{30})\} = 1/12$$

and $\|B_{k+1} - B_k\|_F \leq 1/2\sqrt{\epsilon}\min\{1/\sqrt{6}, 1/\sqrt[3]{60}\} = \sqrt{\epsilon}/(2\sqrt[3]{60})$, it follows that

$$\frac{\|A_{k+1} - A_+\|_F}{\|A\|_F} \leq \epsilon.$$

We emphasize that stop criterion of Theorem 3.2 is interesting since in general we do not have knowledge neither of B_* nor A_+. Nonetheless, we can stop the algorithm an iteration later.

In next theorem we establish an inequality that gives rise to a stop criterion that depends only on the current iterate.

Theorem 3.3. *Let $\bar{B}$ be such that $\bar{B}P_+ = P_+\bar{B}$, $(\bar{B} + P_+ - I)$ is non-singular and $\|\bar{B} - P_+\| \leq \theta < 1$, for some sub-multiplicative and invariant by orthogonal transformations norm $\|\cdot\|$. Then*

$$\frac{1}{\|I\| + \theta}\|\bar{B}^2 - \bar{B}\| \leq \|\bar{B} - P_+\| \leq \frac{1}{1 - \theta}\|\bar{B}^2 - \bar{B}\|. \tag{16}$$

Proof. Since $P_+^2 = P_+$, we have,

$$\bar{B}^2 - \bar{B} = \bar{B}^2 - \bar{B} + P_+ - P_+^2 = (\bar{B} - P_+)(\bar{B} + P_+ - I). \tag{17}$$

Now, since $(\bar{B} + P_+ - I)$ is non-singular,

$$\bar{B} - P_+ = (\bar{B}^2 - \bar{B})(\bar{B} + P_+ - I)^{-1}. \tag{18}$$

From the fact that $(2P_+ - I)^2 = I$ it turns out,

$$\bar{B} + P_+ - I = \bar{B} + 2P_+ - P_+ - I = (2P_+ - I)((2P_+ - I)(\bar{B} - P_+) + I). \tag{19}$$

By hypothesis of norm $\|\cdot\|$ and since $\|\bar{B} - P_+\| \leq \theta < 1$, from the Banach's Lemma it follows that

$$\|(\bar{B} + P_+ - I)^{-1}\| = \|((2P_+ - I)(\bar{B} - P_+) + I)^{-1}\| \leq \frac{1}{1 - \|\bar{B} - P_+\|} \leq \frac{1}{1 - \theta}.$$

Therefore, from (18) we obtain,

$$\|\bar{B} - P_+\| \leq \frac{1}{1 - \theta}\|\bar{B}^2 - \bar{B}\|.$$

Now, from (17) and (19),

$$\|\bar{B}^2 - \bar{B}\| \leq \|\bar{B} - P_+\|(\|\bar{B} - P_+\| + \|I\|) \leq (\|I\| + \theta)\|\bar{B} - P_+\|. \qquad \square$$

Since $B_k P_+ = P_+ B_k$ and $(B_k + P_+ - I)$ is non-singular for all k and thus, from the theorem above, when $\|B_k - P_+\| \leq \theta < 1$ (e.g., $\|B_k - P_+\|_2 \leq 1/2$ for all k) we have that

$$\frac{1}{\|I\| + \theta}\|B_k^2 - B_k\| \leq \|B_k - P_+\| \leq \frac{1}{1-\theta}\|B_k^2 - B_k\|. \tag{20}$$

Moreover, since $\|A_k - A_+\| \leq \|A\|\|B_k - P_+\|$, we also obtain

$$\frac{\|A_k - A_+\|}{\|A\|} \leq \frac{1}{1-\theta}\|B_k^2 - B_k\|. \tag{21}$$

Let us denote by r_{min} the smallest eigenvalue (in magnitude) of $B_0 - 1/2I$, that is, $\gamma_0 = (r_{min} + 1/2)$ is the eigenvalue of B_0 closest to $1/2$. It is worth mention here that the number of iterations required for Algorithm 2.1 declares convergence with a given precision ε in B_k depends directly on how far r_{min} is from zero: smaller values require more iterations for attaining convergence. In addition, since $\mathcal{P}_s$ is monotonically increasing in [0,1], for all k we have that $\gamma_{k+1} = \mathcal{P}_s(\gamma_k)$ will be the eigenvalue of B_{k+1} nearest to $1/2$.

If the norm $\|.\|_2$ is used, notice that $|\gamma_k - 1| \leq 1/2$ or $|\gamma_k| \leq 1/2$ implies the condition $\|B_k - P_+\|_2 \leq 1/2$. Moreover, from (20), we have that

$$2/3|\gamma_k^2 - \gamma_k| \leq \|B_k - P_+\|_2 \leq 2|\gamma_k^2 - \gamma_k|.$$

Namely, one can simply monitoring γ_k in order to stop the iterations.

4. Symmetric banded matrices

Iteration (7) involves matrix–matrix products which costs $O(n^3)$ for general matrices. Moreover, even if two matrices are sparse, their product is not necessarily sparse. Thus, Algorithm 2.1 becomes attractive only when the matrix–matrix multiplication can be performed in much less than $O(n^3)$ flops. This is the case of symmetric banded matrices with narrow band.

Let A be a $n \times n$ symmetric banded matrix with $2b_1 + 1$ diagonals where b_1 is the semi-bandwidth. Consider $B \in \mathbb{R}^{n\times n}$ a banded matrix with semi-bandwidth b_2. It is well known that the cost of the product $C = AB$ is $O(b_1 b_2 n)$ and that the semi-bandwidth of C is at most $b_1 + b_2$.

Thus, if the fill-in and bandwidth can be controlled, Algorithm 2.1 becomes suitable for computing the approximate projection of a symmetric banded matrix onto the positive semidefinite cone.

After each iteration of Algorithm 2.1, the semi-bandwidth of B_k usually increases, and thus, after a small number of iterations the powers of B_k may become expensive to compute. In order to keep the bandwidth under control we employ a band reduction step whenever the semi-bandwidth b of B_k becomes greater than a threshold $\bar{b}$. In the band reduction step we apply an algorithm to reduce the symmetric banded matrix B_k to tridiagonal form:

$$Q_k^\top B_k Q_k = T_k.$$

After that, the fixed-point iterations continue, using the tridiagonal matrix T_k in the place of B_k. Every time a band reduction is done, it is necessary to accumulate the product $Q = Q_0 Q_1 \dots Q_k$ that is used to recover the approximation of the projector P_+ after convergence.

For tridiagonalization of a symmetric banded matrix we used the SBR toolbox [4], specifically, the routine `DSBRDT`, which applies a suitable sequence of blocked Householder transformations in order to reduce a symmetric banded matrix to tridiagonal form.

According to [3], for banded matrices B_k, the update of Q dominates the complexity with $2((d+1)/b)n^3$ versus $6(d+1)n^2$ flops for reduction of B_k alone, where d is the number of upper(lower) subdiagonals to eliminate. For tridiagonalization $d = b - 1$ and the costs are $2n^3$ and $6bn^2$ respectively.

An important remark is that the number of arithmetic operations may be reduced by skipping selected Householder transformations [4]. The amount of savings are considerable whether the eigenvalues of B_k are contained in few narrow clusters. This is the case for the final iterations of Algorithm 2.1 when the eigenvalues are clusterized around 0 and 1.

5. Numerical experiments

The numerical experiments are organized in two subsections. Section 5.1 presents the numerical results for a set of randomly generated banded matrices with varying size and bandwidth. Using such instances we intend to analyze how the ratio bandwidth/size affects the cost of computing powers of the iterated matrix and how often band reductions are required. Section 5.2 brings some non-random examples that can either be generated from `gallery` command from Matlab [16] or downloaded from the University of Florida sparse matrix collection [8]: http://www.cise.ufl.edu/research/sparse/matrices.

In both sections we compare our fixed-point method, with choices $\mathcal{P}_2(t)$ and $\mathcal{P}_3(t)$, with `DSYEV` routine from Lapack and with the benchmark `EIG` built-in routine from Matlab R2016b, in order to validate the proposed approach in terms of performance (CPU time) and quality of the approximate projections.

The algorithms were implemented in Matlab R2016b, with Lapack 3.5.0, and the experiments run on a Macbook Pro, Intel Core i7 2.4Ghz, 8Gb RAM. The source codes are publicly available at

http://mtm.ufsc.br/~douglas/downloads/sdpproj,

apart from the SBR Toolbox [4] that must be downloaded separately.

5.1. Random banded matrices

In order to assess the behavior of Algorithm 2.1 for different matrix sizes and bandwidths, we generated random symmetric banded matrices of increasing size n and semi-bandwidth b. The entries of each lower(upper) subdiagonal were sampled independently from a Gaussian distribution with zero mean and standard deviation 20.

In these experiments we consider the choices $\mathcal{P}(t) = \mathcal{P}_2(t)$ and $\mathcal{P}(t) = \mathcal{P}_3(t)$. The proposed fixed-point methods were compared with the standard approach for computing the projection onto $\mathcal{S}_+^n$ through the eigendecomposition. For computing all eigenvalues and eigenvectors of a real symmetric matrix we considered two routines: `DSYEV` from Lapack [1], which performs a reduction of A to the Hessenberg form and then applies the shifted QR algorithm, and `EIG` from Matlab 2016b [16].

We used $\bar{b} = 0.01n$ as the threshold for band reduction in the fixed-point algorithms. In the `DSBRDT` routine of the SBR toolbox, the tolerance `DRPTOL` was set to 10^{-10}. A Householder transformation is skipped whenever $\|(x_2, \ldots, x_k)^\top\|_2 \leq$ `DRPTOL`, where $x = (x_1, x_2, \ldots, x_k)^\top$ is some vector/sub-column in B_k that should be transformed into $\tilde{x} = \xi e_1$, $e_1 = (1, 0, \ldots, 0)^\top \in \mathbb{R}^k$.

The fixed-point iterations are stopped as soon as one of the two conditions holds:

$$\|B_{k+1} - B_k\| \leq \hat{\epsilon}/\alpha \qquad \text{or} \qquad \|B_k^2 - B_k\| \leq \hat{\epsilon}/\alpha,$$

where $\hat{\epsilon} = 10^{-4}$. Recall the reasoning behind these stopping criteria from Theorems 3.2 and 3.3.

The time reported in the tables includes the whole time for computing an approximation of A_+. This means that after the spectral decomposition, the time spent to obtain the product $Q\Lambda_+Q^\top$ is also considered. Analogously, after convergence of the fixed point iterations the time for computing $B_* = Q\tilde{B}_kQ^\top$, $A_+ \approx B_*A$ (where $\tilde{B}_k$ is the last iterate of Algorithm 2.1 and $Q = Q_0Q_1 \ldots Q_k$ is the cumulative matrix of Householder transformations due to band reductions) is also counted.

In Table 1, we report the time (in seconds) $t(s)$ and the number of iterations k performed by Algorithm 2.1 with $\mathcal{P}_2(t)$ and $\mathcal{P}_3(t)$, namely FP2 and FP3 respectively, for finding the approximate projection onto $\mathcal{S}_+^n$ for random matrices of order n and semi-bandwidth b.

Although the number of iterations of FP3 is smaller than FP2, what was expect due to cubic versus quadratic convergence, each iteration of the former requires one additional matrix multiplication. This increases the iteration cost and also contributes to a faster bandwidth growth, therefore demanding band reductions more often. This explains why the algorithm using $\mathcal{P}_2(t)$ was faster than $\mathcal{P}_3(t)$ in more than 60% of the instances.

Table 1
Experiments on random symmetric banded matrices. FP2 and FP3 stand for the fixed-point method using $\mathcal{P}(t) = \mathcal{P}_2(t)$ and $\mathcal{P}(t) = \mathcal{P}_3(t)$, respectively. The methods based on eigendecomposition are denoted by EIG (Matlab) and DSYEV (Lapack).

		$b = 1$		$b = 2$		$b = 4$		$b = 8$		$b = 16$		$b = 32$		$b = 64$	
	n	k	$t(s)$	k	$t(s)$	k	$t(s)$	k	$t(s)$	k	$t(s)$	k	$t(s)$	k	$t(s)$
FP2	2000	23	2.71	27	2.28	25	2.70	26	3.17	24	3.63	25	3.24	26	3.33
	4000	31	8.06	29	20.95	25	13.91	29	14.48	24	14.06	31	16.92	27	15.51
	6000	31	46.55	32	28.73	29	40.18	27	42.62	30	44.96	26	46.12	28	46.12
	8000	36	114.97	27	71.94	29	110.94	34	105.09	31	107.01	27	108.49	29	111.97
	10000	29	127.69	37	146.77	34	293.08	27	187.26	30	199.13	28	204.86	32	220.08
	12000	34	201.57	30	221.91	34	257.02	31	309.37	30	320.00	28	335.40	30	343.54
	14000	33	290.19	33	324.43	34	384.57	34	469.90	26	444.59	28	490.76	28	476.93
	16000	30	395.90	39	483.18	27	498.57	30	683.23	31	709.13	35	733.45	31	697.91
	18000	31	596.34	30	644.62	33	768.98	30	989.68	40	1004.97	30	1008.92	32	992.86
	20000	32	809.18	31	879.68	29	1014.65	33	1498.64	32	1301.01	36	1415.11	32	1563.12
FP3	2000	15	2.81	18	2.24	16	2.64	17	2.72	16	3.94	17	3.33	17	3.28
	4000	20	11.46	19	21.43	17	16.66	19	18.10	16	16.89	21	19.53	18	18.10
	6000	20	27.86	21	34.31	19	43.77	18	44.50	20	47.09	17	49.17	19	50.41
	8000	24	79.61	18	80.35	19	123.82	22	108.48	21	123.65	18	110.31	19	119.17
	10000	19	120.10	24	146.81	22	168.12	18	172.67	20	203.62	19	189.74	21	208.65
	12000	23	385.15	20	216.13	22	261.52	21	308.01	20	330.38	19	324.00	20	349.47
	14000	22	317.97	22	364.86	22	435.85	22	551.91	17	523.68	18	557.69	18	588.69
	16000	20	393.43	25	549.57	18	539.64	20	766.70	21	784.51	23	872.29	21	831.61
	18000	21	577.37	20	768.35	22	752.21	20	1099.98	27	1103.86	20	1225.02	21	1133.26
	20000	21	711.88	21	1020.29	19	976.23	22	1459.48	21	1515.22	24	1601.19	21	1532.06

(*continued on next page*)

Table 1 (*continued*)

		$b=1$		$b=2$		$b=4$		$b=8$		$b=16$		$b=32$		$b=64$	
	n	k	$t(s)$	k	$t(s)$	k	$t(s)$	k	$t(s)$	k	$t(s)$	k	$t(s)$	k	$t(s)$
EIG	2000	–	0.54	–	1.71	–	1.45	–	1.35	–	1.42	–	1.49	–	1.47
	4000	–	3.15	–	14.78	–	14.16	–	12.19	–	11.58	–	11.51	–	11.45
	6000	–	9.98	–	47.58	–	48.76	–	42.55	–	39.15	–	38.75	–	38.79
	8000	–	23.40	–	108.40	–	106.71	–	98.75	–	97.71	–	99.14	–	99.81
	10000	–	46.40	–	213.99	–	222.13	–	196.49	–	192.64	–	193.26	–	193.14
	12000	–	84.60	–	368.91	–	396.70	–	371.32	–	354.79	–	358.20	–	358.45
	14000	–	125.94	–	564.93	–	629.95	–	567.64	–	538.62	–	537.17	–	542.32
	16000	–	190.41	–	840.00	–	916.40	–	855.64	–	835.96	–	837.71	–	823.73
	18000	–	283.21	–	1172.19	–	1328.93	–	1226.00	–	1195.81	–	1230.99	–	1232.18
	20000	–	397.35	–	1630.04	–	1855.91	–	1765.34	–	1765.87	–	1963.80	–	2033.25
DSYEV	2000	–	5.01	–	3.31	–	2.22	–	2.08	–	2.14	–	2.21	–	2.25
	4000	–	35.05	–	26.93	–	21.99	–	17.48	–	16.64	–	16.63	–	16.79
	6000	–	102.34	–	81.46	–	73.94	–	60.01	–	55.72	–	55.19	–	55.24
	8000	–	208.51	–	136.78	–	155.68	–	134.30	–	126.65	–	125.51	–	125.96
	10000	–	374.20	–	322.71	–	305.58	–	259.35	–	243.38	–	236.52	–	239.13
	12000	–	621.89	–	543.73	–	539.18	–	469.04	–	443.50	–	431.58	–	431.12
	14000	–	927.84	–	843.56	–	783.12	–	748.40	–	695.63	–	679.10	–	666.33
	16000	–	1382.37	–	1197.10	–	1265.34	–	1116.56	–	1049.41	–	1018.05	–	1010.60
	18000	–	1814.32	–	1709.55	–	1786.92	–	1596.50	–	1505.22	–	1458.25	–	1388.47
	20000	–	2558.83	–	2348.73	–	2432.10	–	2234.27	–	2132.79	–	2060.01	–	2037.87

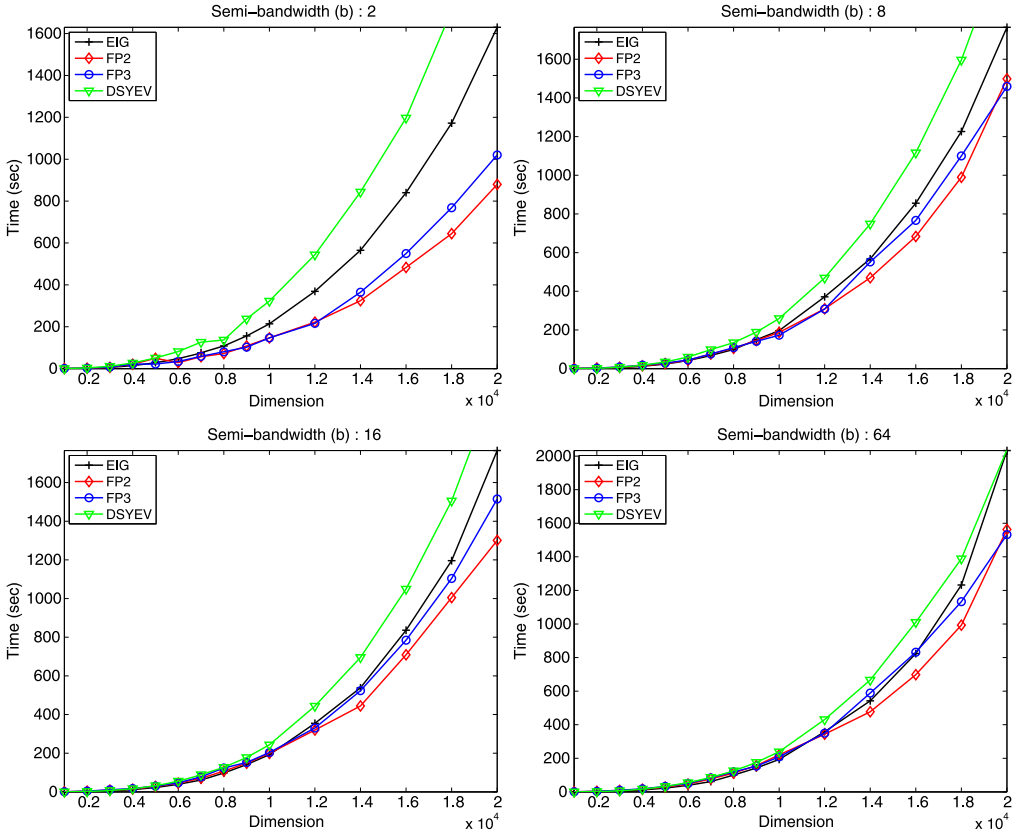

Fig. 2. CPU times for fixed semi-bandwidth b and increasing dimension n.

Table 1 also shows the times of the standard approach for computing the projection A_+ by computing all eigenvalues and eigenvectors through `DSYEV` and `EIG` routines. As we can see from Table 1, the polynomial fixed-point algorithms present a better performance for matrices with relatively narrow band, or in other words, for instances where the ratio b/n is small enough. `EIG` spends less time than the fixed-point iterations in 32 out of 70 instances whereas `DSYEV` outperforms the fixed-point methods only in 15 instances. In these random experiments `DSYEV` does not outperformed `EIG` (Matlab R2016b) in any instance.

Aimed to help reading Table 1, Fig. 2 presents some plots for fixed values of the semi-bandwidth b and increasing values of n.

Let A_* be the approximation of projection produced by the fixed-point method, and assume that the true projection A_+ is the one produced through the spectral decomposition (using `EIG`). We observe that in all instances the distance $\|A_+ - A_*\|_F$ ranges from 10^{-7} to 10^{-6}. This precision can be improved by decreasing the tolerance `DRPTOL` for skipping Householder transformations in the band reduction and also lowering the tolerance $\hat{\epsilon}$ of the stopping criteria.

Table 2
Time spent by the fixed-point with $\mathcal{P}(t) = \mathcal{P}_3(t)$ and the `EIG` routine on sparse narrow band matrices from two collections. The last column present the distance between the approximate projections.

Problem	n	PF3	EIG	
		time (s)	time (s)	$\|A_* - A_+\|_F$
`power` [8]	4941	59.4713	22.0117	6.7099e-09
`cell2` [8]	7055	55.6322	63.9550	2.4009e-08
`bcsstm38` [8]	8032	22.6420	110.5226	5.9065e-09
`ted_B` [8]	10605	74.9348	200.6649	4.6100e-15
`neumann` [16]	3600	9.3645	7.7063	9.1212e-11
`neumann` [16]	6400	43.1780	46.0199	3.0677e-10
`neumann` [16]	10000	145.6844	183.9769	3.6019e-10
`toeppen` [16]	3600	8.5986	7.6830	5.9082e-09
`toeppen` [16]	6400	35.0136	44.0846	5.8503e-09
`toeppen` [16]	10000	118.7514	173.4849	7.2347e-09

The main burden in the fixed-point iterations is the need of band reductions in order to keep the bandwidth under control for allowing an efficient matrix–matrix product (for computing the powers of the iterated matrix).

As discussed in Section 4, since the eigenvectors are needed, each band reduction costs, in the worst case, $2n^3$ flops. Although the number of flops may be reduced by discarding some Householder transformations whenever it is possible, sometimes this strategy is not effective, mainly in the first iterations where the eigenvalues of B are spread in the interval $(0, 1)$. Roughly speaking, in the presented numerical experiments of this subsection a band reduction is required at each four iterations when $\mathcal{P}_2(t)$ is used, whereas $\mathcal{P}_3(t)$ demands a band reduction at each three iterations.

5.2. Sparse narrow banded matrices

We present in this subsection a set of experiments based on sparse narrow banded matrices that either can be generated from `gallery` command from Matlab or downloaded from the University of Florida sparse matrix collection [8]. In fact, some of these matrices are not symmetric, thus, actually we consider A as the closest symmetric matrix (w.r.t. Frobenius norm) for each of them. Such matrices present a narrow band structure which is very sparse, allowing an efficient matrix–matrix product with a small fill-in which, in its turn, contributes to keep the bandwidth moderate and avoids the necessity of several band reductions.

Table 2 brings for each problem the time in seconds spent by the fixed point method using $\mathcal{P}_3(t)$ (FP3) and `EIG`, respectively, to find an approximate projection as well as the distance between these two approximations in the last column. We can observe that, for this class of matrices, FP3 finds a good approximation (at least as good as the one obtained through spectral decomposition) for the true projection with reasonable savings in terms of time, specially for large n.

6. Conclusions and future work

A polynomial fixed-point method for computing an approximation of A_+, the projection of a symmetric matrix A onto the positive semidefinite cone, has been presented, along with its convergence analysis and theoretical properties.

The resulting algorithm demands matrix–matrix products in each iteration and may be adequate for computing the approximate projection of certain structured matrices. If the powers of a matrix are not expensive to compute and the precision in the approximate projection can be relaxed, the savings in terms of computational time can be considerable.

For narrow banded matrices, the numerical experiments show that the method is competitive with respect to the standard approach of computing A_+ through spectral decomposition, and these preliminary computational results encourage us to work on further improvements on the algorithm.

Nevertheless, it is important to mention that, although the focus of the paper is on finding the nearest symmetric positive semidefinite matrix, the ideas behind the fixed-point algorithms may find other interesting applications. For instance, from the discussion of Section 1, the fixed-point algorithm could also be used to approximately compute the polar decomposition.

Finally, suppose that we are not interested in A_+ but on its action over some vector b. We are currently working on how to specialize the presented method to compute A_+b using only matrix-vector products in each iteration.

Acknowledgements

The authors are grateful to anonymous referees for useful comments and suggestions that helped to improve this work.

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, LAPACK Users' Guide, third edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.
[2] L. Auslander, A. Tsao, On parallelizable eigensolvers, Adv. in Appl. Math. 13 (3) (1992) 253–261.
[3] C.H. Bischof, B. Lang, X. Sun, A framework for symmetric band reduction, ACM Trans. Math. Software 26 (4) (2000) 581–601.
[4] C.H. Bischof, B. Lang, X. Sun, Algorithm 807: the SBR Toolbox – Software for Successive Band Reduction, ACM Trans. Math. Software 26 (4) (2000) 602–616.
[5] C. Bischof, X. Sun, A. Tsao, T. Turnbull, A study of the invariant subspace decomposition algorithm for banded symmetric matrices, in: Proceedings of the Fifth SIAM Conference on Applied Linear Algebra, 1994, pp. 321–325.
[6] Å. Björck, C. Bowie, An iterative algorithm for computing the best estimate of an orthogonal matrix, SIAM J. Numer. Anal. 8 (2) (1971) 358–364.
[7] J. Chen, E. Chow, A Newton–Schulz variant for improving the initial convergence in matrix sign computation, Preprint ANL/MCS-P5059-0114, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, 2014.
[8] T.A. Davis, Y. Hu, The University of Florida Sparse Matrix Collection, ACM Trans. Math. Software 38 (1) (2011) 1–25.

[9] G. Golub, C. Van Loan, Matrix Computations, 3rd edition, Johns Hopkins University Press, 1996.
[10] M. Gu, S.C. Eisenstat, A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem, SIAM J. Matrix Anal. Appl. 16 (1) (1995) 172–191.
[11] D. Henrion, J. Malick, Projection methods for conic feasibility problems: applications to polynomial sum-of-squares decompositions, Optim. Methods Softw. 26 (1) (2011) 23–46.
[12] T. Helgaker, P. Jørgensen, J. Olsen, Molecular Electronic-Structure Theory, John Wiley & Sons Ltd, Chichester, 2000.
[13] N.J. Higham, Computing the polar decomposition–with applications, SIAM J. Sci. Statist. Comput. 7 (4) (1986) 1160–1174.
[14] N.J. Higham, Computing a nearest symmetric positive semidefinite matrix, Linear Algebra Appl. 103 (1988) 103–118.
[15] N.J. Higham, Functions of Matrices: Theory and Computation, SIAM, Philadelphia, 2008.
[16] D.J. Higham, N.J. Higham, MATLAB Guide, second edition, SIAM, Philadelphia, 2005.
[17] A. Holas, Transforms for idempotency purification of density matrices in linear-scaling electronic-structure calculations, Chem. Phys. Lett. 340 (2001) 552–558.
[18] Z. Kovarik, Some iterative methods for improving orthonormality, SIAM J. Numer. Anal. 7 (3) (1970) 386–389.
[19] R. McWeeny, Some recent advances in density matrix theory, Rev. Modern Phys. 32 (Apr 1960) 335–369.
[20] C.D. Meyer, Matrix Analysis and Applied Linear Algebra, Society for Industrial and Applied Mathematics, 2001.
[21] N.C. Schwertman, D.M. Allen, Smoothing an indefinite variance-covariance matrix, J. Stat. Comput. Simul. 9 (3) (1979) 183–194.
[22] M.J. Todd, Semidefinite Optimization, Acta Numer. 10 (2001) 515–560.
[23] L. Vandenberghe, S. Boyd, Semidefinite Programming, SIAM Rev. 38 (1) (1996) 49–95.