AB-Algorithm and Its Application for Solving Matrix Square Roots

Matthew M. Lin
Chun-Yueh Chiang

NCTS/Math Technical Report
2016-016

National Center for
Theoretical Sciences
Mathematics Division, Taiwan

NCTS

AB-algorithm and its application for solving matrix square roots

Matthew M. Lin * Chun-Yueh Chiang †

September 2, 2016

Abstract

This work is to propose an iterative method to compute a stable subspace of a regular matrix pencil. This approach is to define a sequence of matrix pencils via particular left null spaces. We show that this iteration preserves a discrete-type flow depending only on the initial matrix pencil. Via this recursion relationship, we propose an accelerated iterative method to compute the stable subspace and use it to solve the principal square root of a given matrix, both nonsingular and singular. We show that this method can not only find out the matrix square root, but also construct an iterative approach which converges to the square root with any desired order.

Keywords: Stable subspace, Sherman Morrison Woodbury formula, Square root, Accelerated iterative method, Q-superlinear convergence

1 Introduction

Throughout this paper we shall use the following notation to facilitate our discussions. $\lambda(A)$ and $\lambda(A, B)$ denote the sets of eigenvalues of the matrix A and the matrix pencil $A - \lambda B$, respectively, and let $\rho(A)$ be the spectral radius of the square matrix A. $\mathbb{C}^+$ and $\mathbb{C}^-$ represent the open right and left half complex planes.

Given a regular $n \times n$ matrix pencil $A - \lambda B$ (i.e., $\det(A - \lambda B)$ is not identically zero for all λ) and an integer $m \leq n$, we want to find in this work a full rank matrix $U \in \mathbb{C}^{n\times m}$ such that

$$AU = BU\Lambda, \tag{1}$$

*Department of Mathematics, National Cheng Kung University, Tainan 701, Taiwan. The first author was supported by the Ministry of Science and Technology of Taiwan under grant 104-2115-M-006-017-MY3. (mhlin@mail.ncku.edu.tw)

†Corresponding Author, Center for General Education, National Formosa University, Huwei 632, Taiwan. The second author was supported by the Ministry of Science and Technology of Taiwan under grant 105-2115-M-150-001. (chiang@nfu.edu.tw)

where $\Lambda \in \mathbb{C}^{m\times m}$ and $\rho(\Lambda) < 1$. This problem is related to the so-called *generalized spectral divide and conquer* (SDC) problem [1, 4], which is to find a pair of left and right deflating subspaces $\mathcal{L}$ and $\mathcal{R}$ such that

$$A\mathcal{R} \subset \mathcal{L}, \quad B\mathcal{R} \subset \mathcal{L},$$

corresponding to eigenvalues of the pair $A - \lambda B$ in a specified region $\mathcal{D} \subset \mathbb{C}$. That is, find two nonsingular matrices $U_L = (U_{L_1}, U_{L_2})$ and $U_R = (U_{R_1}, U_{R_2})$ with $\mathcal{L} = \mathrm{span}(U_{L_1})$ and $\mathcal{R} = \mathrm{span}(U_{R_1})$ so that

$$AU_R = U_L \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}, \quad BU_R = U_L \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix},$$

and the eigenvalues of $A_{11} - \lambda B_{11}$ are the eigenvalues of $A - \lambda B$ in the region $\mathcal{D}$. Note that if $A_{11} - \lambda B_{11}$ has no infinite eigenvalues, then B_{11} is invertible and

$$AU_{R_1} = BU_{R_1}(B_{11}^{-1}A_{11});$$

if $A_{11} - \lambda B_{11}$ has no zero eigenvalues, then A_{11} is invertible and

$$AU_{R_1}(A_{11}^{-1}B_{11}) = BU_{R_1}.$$

The region $\mathcal{D}$ in the SDC problem is generally assumed in the interior (or exterior) of the unit disk. Otherwise, the Möbius transformations $(\alpha A+\beta B)(\gamma A+\delta B)^{-1}$ can be applied to transform original region as a rather general region [1].

Our proposed AB-algorithm is to solve (1) by defining a sequence of matrix pencils $\{A_k - \lambda B_k\}$ with $(A_1, B_1) = (A, B)$ and $(A_k, B_k) = (\mathcal{M}_{k-1}A_{k-1}, \mathcal{N}_{k-1}B_1)$ for any integer $k > 1$. Here, $(\mathcal{N}_{k-1}, \mathcal{M}_{k-1})$ is a solution belonging to the left null space of $\begin{bmatrix} A_1 \\ -B_k \end{bmatrix}$, that is,

$$\mathcal{N}_k A_1 - \mathcal{M}_k B_k = 0. \tag{2}$$

Observe that $A_1 U = B_1 U \Lambda^1$. Suppose this process can be continually iterated to obtain the new pencil $A_k - \lambda B_k$ such that $A_k U = B_k U \Lambda^k$. It must be that

$$\begin{aligned} A_{k+1}U &= \mathcal{M}_k A_k U = \mathcal{M}_k B_k U \Lambda^k \\ &= \mathcal{N}_k A_1 U \Lambda^k = \mathcal{N}_k B_1 U \Lambda^{k+1} = B_{k+1} U \Lambda^{k+1}. \end{aligned} \tag{3}$$

This implies that if $\rho(\Lambda) < 1$, and if the sequence $\{B_k\}$ is uniformly bounded, then $\lim_{k\to\infty} A_k U = 0$. Once the sequence $\{A_k\}$ also converges, say $A_\infty := \lim_{k\to\infty} A_k$, we are able to solve the solution U by computing the the right null space of A_∞. Major results on the SDC problem can be found in [13, 16, 15, 17, 2, 5, 6, 1, 8]. See also [4] for a comprehensive review of its applications. Unlike convectional methods, our proposed AB-algorithm theoretically preserves a discrete-type flow property and can be applied to accelerated the iteration with convergence of any desired order. Numerically, it can be applied to compute the matrix square root, which has been widely discussed in [14, 7, 9, 10, 11, 18, 19] and the references therein.

Note that matrix square roots are not unique (even up to sign) or even exist, for example,

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix}^2$$

for any $\theta \in \mathbb{R}$ and $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ does not have a square root. Indeed, let $S \in \mathbb{C}^{n\times n}$ be a matrix having no nonpositive real eigenvalues. Then the quadratic matrix equation

$$X^2 - S = 0 \tag{4}$$

has a unique solution X such that $\lambda(X) \subset \mathbb{C}^+$. We call this the principal square root of S and denote it by $\sqrt{S}$ [7, 10]. By assumption of uniqueness, we apply our AB-algorithm to calculate the square root of a given matrix. We show that under mild adjustments, the speed of convergence can be of any order.

This work is organized as follows. In section 2 we provide properties of the AB-Algorithm. In section 3 we modified this AB-Algorithm so that its convergence can be of any order. In section 4 we report a numerical application to solve the matrix square root, and the concluding remarks are given in section 5.

2 The AB-Algorithm and Its Corresponding Properties

Recall that the idea of the AB-Algorithm depends heavily on the determination of the left null space of $\begin{bmatrix} A_1 \\ -B_k \end{bmatrix}$. Observe that $B_1(A_1+B_1)^{-1}A_1 - A_1(A_1+B_1)^{-1}B_1 = 0$, if $-1 \notin \lambda(A_1, B_1)$ and (A_1, B_1) are two $n\times n$ matrices. That is, $(N_1, M_1) := (B_1(A_1+B_1)^{-1}, A_1(A_1+B_1)^{-1})$ is the left null space of $\begin{bmatrix} A_1 \\ -B_1 \end{bmatrix}$. Using the same procedure, we would like to generate the matrix sequences $\{A_k\}$ and $\{B_k\}$ by defining

$$A_k = A_1(A_1 + B_{k-1})^{-1}A_{k-1}, \tag{5a}$$

$$B_k = B_{k-1}(A_1 + B_{k-1})^{-1}B_1, \tag{5b}$$

once the process can be iterated.

It should be noted that if $\mathcal{M}_{k-1} = B_{k-1}(A_1+B_{k-1})^{-1}$ and $\mathcal{N}_{k-1} = A_1(A_1+B_{k-1})^{-1}$ for any integer $k > 1$, it can be seen that $B_{k-1}(A_1+B_{k-1})^{-1}A_1 = A_1(A_1+B_{k-1})^{-1}B_{k-1}$, which satisfies the assumption (2). For simplicity, we let $\Delta_{i,j} := (A_i + B_j)^{-1}$ so that the sequences $\{A_k\}$ and $\{B_k\}$ in (5) can be rewritten as

$$A_k = A_1\Delta_{1,k-1}A_{k-1} = A_{k-1} - B_{k-1}\Delta_{1,k-1}A_{k-1}, \tag{6a}$$

$$B_k = B_{k-1}\Delta_{1,k-1}B_1 = B_1 - A_1\Delta_{1,k-1}B_1. \tag{6b}$$

Based on (6), we propose the following AB-algorithm for computing the stable subspace of the matrix pencil $A_1 - \lambda B_1$:

Algorithm 2.1. AB-Algorithm

1. *Given a pencil* $A_1 - \lambda B_1$*, initialize a tolerance* $\tau > 0$ *and a positive integer kmax.*
2. *For* $k = 2...$*, iterate*
   - *(a)* $A_k = A_1 \Delta_{1,k-1} A_{k-1}$,
   - *(b)* $B_k = B_{k-1} \Delta_{1,k-1} B_1$,

   until $\|A_k - A_{k-1}\| < \tau$ *or* $k > kmax$.

Note that on the one hand, Algorithm 2.1 provides an alternative approach for finding the invariant stable subspace U (i.e., $A_1 U = U\Lambda$ and $\rho(\Lambda) < 1$) of the matrix A_1 by constructing A_∞ (once it exists) directly as follows:

Remark 2.1. *If no breakdown occurs in Algorithm 2.1 and* $B_1 = I_n$*, for any integer* $k > 1$ *we have*

$$A_k = A_1^k (\sum_{j=0}^{k-1} A_1^j)^{-1}, \tag{7a}$$

$$B_k = (\sum_{j=0}^{k-1} A_1^j)^{-1}. \tag{7b}$$

In other words, to obtain the stable subspace of the matrix A_1*, we only need to focus on the iterations generated by* (7a)*.*

On the other hand, once the iteration is available, we are interested in characterizing the transformation of eigenvalues of the matrix pencil $A_1 - \lambda B_1$ after each iteration. First, we give an observation about the relationship between the eigenvalues of $A_k - \lambda B_k$ and the eigenvalues of $A_1 - \lambda B_1$. Since the proof can be read off from (3), we omit our proof here.

Lemma 2.1. *Let* $A_1 - \lambda B_1$ *be a regular matrix pencil, and let* $\{A_k - \lambda B_k\}$ *be the sequence of matrix pencils generated by Algorithm 2.1, if no breakdown occurs. If* $\lambda \in \lambda(A_1, B_1)$ *with* $\lambda \in \mathbb{R} \cup \{\infty\}$*, then* $\lambda^k \in \lambda(A_k, B_k)$. $(Here, \infty^k := \infty)$

Subsequently, we have the following theorem which gives rise to the appearance of new eigenvalues induced by the AB-algorithm.

Theorem 2.1. *Let* $A_1 - \lambda B_1$ *be a regular matrix pencil, and let* $\{A_k - \lambda B_k\}$ *be the sequence of matrix pencils generated by Algorithm 2.1, if no breakdown occurs. Let* $\{\lambda_1^{(i,k)}, \ldots, \lambda_n^{(i,k)}\}$ *be the set of eigenvalues of the matrix pencils* $A_i - \lambda B_k$ *for any two positive integers* i *and* k*. Then, for* $1 \leq j \leq n$*, the set of eigenvalues has the following properties:*

1. $\lambda_j^{(1,k)} = \begin{cases} \sum_{s=1}^{k} (\lambda_j^{(1,1)})^s, & \lambda_j^{(1,1)} \in \mathbb{R}, \\ \infty, & \lambda_j^{(1,1)} = \infty. \end{cases}$

2. $\lambda_j^{(i,1)} = \begin{cases} \dfrac{(\lambda_j^{(1,1)})^i}{\sum_{s=0}^{i-1} (\lambda_j^{(1,1)})^s}, & \lambda_j^{(1,1)} \in \mathbb{R}, \\ \infty, & \lambda_j^{(1,1)} = \infty. \end{cases}$

3. $\lambda_j^{(i,k)} = \begin{cases} (\lambda_j^{(1,1)})^i \dfrac{\sum_{s=0}^{k-1} (\lambda_j^{(1,1)})^s}{\sum_{s=0}^{i-1} (\lambda_j^{(1,1)})^s}, & \lambda_j^{(1,1)} \in \mathbb{R}, \\ \infty, & \lambda_j^{(1,1)} = \infty. \end{cases}$

Proof. Assume without loss of generality that A_1 and B_1 are upper triangular matrices. Otherwise, let U and V be two unitary matrices such that $U^H A_1 V$ and $U^H B_1 V$ both are upper triangular matrices. Upon using (5), it can be seen that A_k and B_k are also upper triangular, and

$$\lambda_j^{(1,k)} = \begin{cases} (1 + \lambda_j^{(1,k-1)})\lambda_j^{(1,1)}, & \lambda_j^{(1,1)} \in \mathbb{R}, \\ \infty, & \lambda_j^{(1,1)} = \infty, \end{cases}$$

$$\lambda_j^{(i,1)} = \begin{cases} \dfrac{\lambda_j^{(i,i)}}{1 + \lambda_j^{(1,i-1)}}, & \lambda_j^{(1,1)} \in \mathbb{R}, \\ \infty, & \lambda_j^{(1,1)} = \infty. \end{cases}$$

Moreover,

$$\lambda_j^{(i,k)} = \begin{cases} (1 + \lambda_j^{(1,k-1)})\lambda_j^{(i,1)}, & \lambda_j^{(1,1)} \in \mathbb{R}, \\ \infty, & \lambda_j^{(1,1)} = \infty, \end{cases}$$

for $i, k \geq 2$. We remark that $\lambda_j^{(1,i-1)} \neq -1$ since $A_i - \lambda B_i$ is well-defined, and from Lemma 2.1, we have $\lambda_j^{(i,i)} = (\lambda_j^{(1,1)})^i$, which completes the proof of the theorem.

□

Note that Algorithm 2.1 is workable if and only if the sum of matrices A_1 and B_{k-1}, for any integer $k > 1$, is invertible, that is, $-1 \notin \lambda(A_1, B_{k-1})$, for any integer $k > 1$. This capacity can be can be completely characterized by the pth roots of unity, except itself.

Theorem 2.2. *Let $A_1 - \lambda B_1$ be a regular matrix pencil, and let*

$$S_k = \bigcup_{2 \leq p \leq k+1} \{e^{\frac{2q\pi i}{p}} : 1 \leq q \leq p-1\}.$$

If

$$S_k \cap \lambda(A_1, B_1) = \phi,$$

then the sequence of matrix pencils $A_k - \lambda B_k$, for any integer $k \geq 1$, can be generated using Algorithm 2.1, or, generally, all sequences of matrices $\{A_k - \lambda B_k\}$ generated by iterations (5) *with the initial matrix pencil $A_1 - \lambda B_1$ are no breakdown, if*

$$S_\infty \cap \lambda(A_1, B_1) = \phi. \tag{8}$$

Corollary 2.1. *For any positive integers i, j and k, we have $A_k - B_k = A_1 - B_1$, that is, $A_i - A_j = B_i - B_j$, provided that $S_{\max\{i,j,k\}} \cap \lambda(A_1, B_1) = \phi$.*

Proof. The proof is by induction on k. When $k = 1$, the result is evident. Suppose we have proved this corollary for $k = \ell$. Then, by the induction hypothesis

$$\begin{aligned} A_{\ell+1} - B_{\ell+1} &= A_\ell - B_\ell \Delta_{1,\ell} A_\ell - B_\ell \Delta_{1,\ell} B_1 \\ &= A_\ell - B_\ell \Delta_{1,\ell}(A_\ell + B_1) = A_\ell - B_\ell = A_1 - B_1. \end{aligned}$$

□

We remark that Corollary 2.1 implies that $\lim_{k\to\infty} A_k$ exists if and only if $\lim_{k\to\infty} B_k$ exists. Note that in (5), the iterations of the matrix pencils $A_k - \lambda B_k$, for $k \geq 1$, are relative to the initial pencil $A_1 - \lambda B_1$. We would like to derive a more general iterative method, which are easily accessible through any initial pencil $A_i - \lambda B_i$. To this purpose, we shall first introduce the well-known Sherman Morrison Woodbury formula (SMWF).

Lemma 2.2. *[3] Let A be an arbitrary matrix of size n, and let X and Y be two $n \times n$ nonsingular matrices. If $Y^{-1} \pm AX^{-1}A^H$ is nonsingular, then*

$$(X \pm AYB)^{-1} = X^{-1} \mp X^{-1}A(Y^{-1} \pm BX^{-1}A)^{-1}BX^{-1}.$$

This lemma gives a useful method to prove the following result.

Theorem 2.3. *Let the assumption* (8) *holds and $\{A_k - \lambda B_k\}$ be the sequence of matrix pencils obtained by* (5) *with initial $A_1 - \lambda B_1$. Then,*

$$A_{i+j} = A_i(A_i + B_j)^{-1}A_j, \tag{9a}$$

$$B_{i+j} = B_j(A_i + B_j)^{-1}B_i, \tag{9b}$$

where i and j are any two positive integers.

Proof. This proof is divided into two parts. We first fix $j = 1$ and show that the statement (9) is true for any positive integer i. We prove by induction on i.

When $i = 1$, the statement (9) is definitely true from the definition of A_2 and B_2. Suppose (9) is true for $i = s$. It follows from Lemma 2.2 that

$$\begin{aligned}
\Delta_{1,s+1} &= (A_1 + B_s - A_s\Delta_{s,1}B_s)^{-1} \\
&= \Delta_{1,s} + \Delta_{1,s}A_s(A_s + B_1 - B_s\Delta_{1,s}A_s)^{-1}B_s\Delta_{1,s} \\
&= \Delta_{1,s} + \Delta_{1,s}A_s\Delta_{1+s,1}B_s\Delta_{1,s}, \\
\Delta_{1+s,1} &= (A_s - B_s\Delta_{1,s}A_s + B_1)^{-1} \\
&= \Delta_{s,1} + \Delta_{s,1}B_s(A_1 + B_s - A_s\Delta_{s,1}B_s)^{-1}A_s\Delta_{s,1} \\
&= \Delta_{s,1} + \Delta_{s,1}B_s\Delta_{1,s+1}A_s\Delta_{s,1}.
\end{aligned}$$

Thus, we have

$$\begin{aligned}
A_{(s+1)+1} &= A_{1+(s+1)} = A_{s+1} - B_{s+1}\Delta_{1,s+1}A_{s+1} \\
&= A_1 - B_1\left[\Delta_{s,1} + \Delta_{s,1}B_s\Delta_{1,s+1}A_s\Delta_{s,1}\right]A_1 \\
&= A_1 - B_1\Delta_{s+1,1}A_1 = A_{s+1}\Delta_{s+1,1}A_1, \\
B_{(s+1)+1} &= B_{1+(s+1)} = B_1 - A_1\Delta_{1,s+1}B_1 \\
&= B_1 - A_1\left[\Delta_{1,s} + \Delta_{1,s}A_s\Delta_{s+1,1}B_s\Delta_{1,s}\right]B_1 \\
&= B_{s+1} - A_{s+1}\Delta_{s+1,1}B_{s+1} = B_1\Delta_{s+1,1}B_{s+1},
\end{aligned}$$

which completes the proof of the first part.

Now suppose that (9) is true for $j = s$ and any i. In particular,

$$\begin{aligned}
\Delta_{i,s+1} &= (A_i + B_s - A_s\Delta_{s,1}B_s)^{-1} \\
&= \Delta_{i,s} + \Delta_{i,s}A_s(A_s + B_1 - B_s\Delta_{i,s}A_s)^{-1}B_s\Delta_{i,s} \\
&= \Delta_{i,s} + \Delta_{i,s}A_s\Delta_{i+s,1}B_s\Delta_{i,s}, \\
\Delta_{i+s,1} &= (A_s - B_s\Delta_{i,s}A_s + B_1)^{-1} \\
&= \Delta_{s,1} + \Delta_{s,1}B_s(A_i + B_s - A_s\Delta_{s,1}B_s)^{-1}A_s\Delta_{s,1} \\
&= \Delta_{s,1} + \Delta_{s,1}B_s\Delta_{i,s+1}A_s\Delta_{s,1}.
\end{aligned}$$

This implies

$$\begin{aligned}
A_{i+(s+1)} &= A_{(i+s)+1} = A_1 - B_1\Delta_{i+s,1}A_1 \\
&= A_1 - B_1\left[\Delta_{s,1} + \Delta_{s,1}B_s\Delta_{i,s+1}A_s\Delta_{s,1}\right]A_1 \\
&= A_{s+1} - B_{s+1}\Delta_{i,s+1}A_{s+1} = A_i\Delta_{i,s+1}A_{s+1}, \\
B_{i+(s+1)} &= B_{(i+s)+1} = B_{i+s} - A_{i+s}\Delta_{i+s,1}B_{i+s} \\
&= B_i - A_i\left[\Delta_{i,s} + \Delta_{i,s}A_s\Delta_{i+s,1}B_s\Delta_{i,s}\right]B_i \\
&= B_i - A_i\Delta_{i,s+1}B_i = B_{s+1}\Delta_{i,s+1}B_i,
\end{aligned}$$

which completes the proof of the theorem. □

Two things are required to be noted. First, Theorem 2.3 implies that the iterative sequence $\{A_k - \lambda B_k\}$ can be formulated explicitly from any two matrix pencils $A_i - \lambda B_i$ and $A_j - \lambda B_j$, where $i + j = k$. The formula also gives rise to a discrete-type flow and can be used to accelerate the iterations given in Algorithm 2.1. Second, it follows from Corollary 2.1 and Theorem 2.3 that $A_k = A_{k-1}\Delta_{1,k-1}A_1 = A_1\Delta_{1,k-1}A_{k-1} = A_{k-1}\Delta_{k-1,1}A_1 = A_1\Delta_{k-1,1}A_{k-1}$. It shows that the iterations A_k and B_k, regardless of the assumptions (5), have the following four equivalent forms by using the same initial matrix pencil:

1.	$A_k^{(1)} = A_1^{(1)}(A_1^{(1)} + B_{k-1}^{(1)})^{-1}A_{k-1}^{(1)}$, $B_k^{(1)} = B_{k-1}^{(1)}(A_1^{(1)} + B_{k-1}^{(1)})^{-1}B_1^{(1)}$;
2.	$A_k^{(2)} = A_1^{(2)}(B_1^{(2)} + A_{k-1}^{(2)})^{-1}A_{k-1}^{(2)}$, $B_k^{(2)} = B_{k-1}^{(2)}(A_1^{(2)} + B_{k-1}^{(2)})^{-1}B_1^{(2)}$;
3.	$A_k^{(3)} = A_1^{(3)}(A_1^{(3)} + B_{k-1}^{(3)})^{-1}A_{k-1}^{(3)}$, $B_k^{(3)} = B_{k-1}^{(3)}(B_1^{(3)} + A_{k-1}^{(3)})^{-1}B_1^{(3)}$;
4.	$A_k^{(4)} = A_1^{(4)}(B_1^{(4)} + A_{k-1}^{(4)})^{-1}A_{k-1}^{(4)}$, $B_k^{(4)} = B_{k-1}^{(4)}(B_1^{(4)} + A_{k-1}^{(4)})^{-1}B_1^{(4)}$.

The next theorem is to know how the eigeninformation is transferred during the iterative process.

Theorem 2.4. *Let $A_1 - \lambda B_1$ be a regular matrix pencil, and let $\{A_k - \lambda B_k\}$ be the sequence of matrices generated by Algorithm 2.1. Suppose that the condition (8) holds and $A_1U = B_1U\Lambda$. Then,*

(a) $A_1U = B_kU\sum_{j=1}^{k}\Lambda^j$.

(b) $A_kU = B_kU\Lambda^k$. *In particular, if $1 \notin \lambda(\Lambda)$, then*

$$A_kU = (B_1 - A_1)U\Lambda^k(I - \Lambda^k)^{-1}. \tag{10}$$

(c) $A_iU\sum_{j=1}^{i}\Lambda^j = B_kU\Lambda^i\sum_{j=1}^{k}\Lambda^j$, *for any two positive integers i and k.*

Proof. Clearly, (a) is true for $k = 1$. Suppose that the statement is true for a positive integer $k = s$; that is,

$$A_1U = B_sU\sum_{j=1}^{s}\Lambda^j.$$

Note that

$$\begin{aligned} A_1U - B_s\Delta_{1,s}A_1U &= (A_1 + B_s)\Delta_{1,s}A_1U - B_s\Delta_{1,s}A_1U \\ &= B_s\Delta_{1,s}A_1U\sum_{j=1}^{s}\Lambda^j = B_s\Delta_{1,s}B_1U\sum_{j=2}^{s+1}\Lambda^j, \end{aligned}$$

so that

$$A_1U = B_s\Delta_{1,s}B_1U\Lambda + B_s\Delta_{1,s}B_1U\sum_{j=2}^{s+1}\Lambda^j = B_{s+1}U\sum_{j=1}^{s+1}\Lambda^j.$$

The result of the first part of (b) has been given in our introduction. We thus omit the proof here. Since

$$A_kU = B_kU\Lambda^k = (A_k + B_1 - A_1)U\Lambda^k = A_kU\Lambda^k + (B_1 - A_1)U\Lambda^k,$$

we see that (10) holds, while $1 \notin \lambda(\Lambda)$. Here, the second equality follows from Corollary 2.1.

To prove (c), we first show that for any positive integer i,

$$A_1U\Lambda^i = A_iU\sum_{j=1}^{i}\Lambda^j.$$

By Theorem 2.3, since $A_i = A_1 - B_1\Delta_{i-1,1}A_1$ and $B_i = B_1\Delta_{i-1,1}B_{i-1}$, we have

$$(A_1 - A_i)U = (B_1\Delta_{i-1,1}A_1)U = B_1\Delta_{i-1,1}B_{i-1}U\sum_{j=1}^{i-1}\Lambda^j = B_iU\sum_{j=1}^{i-1}\Lambda^j.$$

Or, equivalently,

$$A_1U\Lambda^i = A_iU\Lambda^i + B_iU\Lambda^i\sum_{j=1}^{i-1}\Lambda^j = A_iU\sum_{j=1}^{i}\Lambda^j,$$

since $A_iU = B_iU\Lambda^i$.

Second, from (a), we have already proved (c) for $i = 1$ and a given positive integer k. Assume (c) is true for $i = s$; that is,

$$A_sU\sum_{j=1}^{s}\Lambda^j = B_kU\Lambda^s\sum_{j=1}^{k}\Lambda^j.$$

Then

$$A_{s+1}U\sum_{j=1}^{s+1}\Lambda^j = A_1U\Lambda^{s+1} = (A_sU\sum_{j=1}^{s}\Lambda^j)\Lambda = B_kU\Lambda^{s+1}\sum_{j=1}^{k}\Lambda^j.$$

□

3 Modified AB-Algorithm

Let $\{A_k - \lambda B_k\}$ be the sequence of matrices generated by Algorithm 2.1. Before we move on, we should emphasize that the structure of the matrix pencil $A_k -$

λB_k is invariant once the subscripts $i+j=k$; that is, the generation of the sequence $\{A_k - \lambda B_k\}$ is independent of the subscript in A_i, A_j, B_i and B_j. To fully take advantages of this invariance, we would like to design algorithms by applying Theorem 2.3 to generate accelerated iterations with convergence of any desired order as follows.

Algorithm 3.1. Modified AB-Algorithm

1. *Initialize a positive integer* $r > 1$ *and let* $(\widehat{A}_1, \widehat{B}_1) = (A_1, B_1)$*;*
2. *For* $k = 2, \ldots,$ iterate

$$\widehat{A}_k = A_{k-1}^{(r-1)}(A_{k-1}^{(r-1)} + \widehat{B}_{k-1})^{-1}\widehat{A}_{k-1},$$
$$\widehat{B}_k = \widehat{B}_{k-1}(A_{k-1}^{(r-1)} + \widehat{B}_{k-1})^{-1}B_{k-1}^{(r-1)},$$

   until convergence, where $(A_{k-1}^{(r-1)}, B_{k-1}^{(r-1)})$ *is defined in step 4.*

3. *For* $\ell = 1, \ldots, r-2$*, iterate*

$$A_{k-1}^{(\ell+1)} = A_{k-1}^{(\ell)}(A_{k-1}^{(\ell)} + \widehat{B}_{k-1})^{-1}\widehat{A}_{k-1},$$
$$B_{k-1}^{(\ell+1)} = \widehat{B}_{k-1}(A_{k-1}^{(\ell)} + \widehat{B}_{k-1})^{-1}B_{k-1}^{(\ell)},$$

   with $(A_{k-1}^{(1)}, B_{k-1}^{(1)}) = (\widehat{A}_{k-1}, \widehat{B}_{k-1})$.

For clarity, one thing should be emphasized here. Recall that the AB algorithm has been developed to obtain the stable space of the generalized eigenvalue problem $A_1U = B_1U\Lambda$. However, the sequence $\{A_kU\}$ provided in Algorithm 2.1 converges only r-linearly to 0, once the spectral radius of Λ is less than 1, and the sequence $\{B_k\}$ is uniformly bounded. From Algorithm 3.1 it follows that

$$A_{k-1}^{(\ell+1)}U = B_{k-1}^{(\ell+1)}U\Lambda^{(\ell+1)r^{k-2}}, \tag{11a}$$
$$\widehat{A}_kU = \widehat{B}_kU\Lambda^{r^{k-1}}, \tag{11b}$$

for $k = 2, \ldots$, and $\ell = 1, \ldots, r-2$, and $(\widehat{A}_k, \widehat{B}_k) = (A_{r^{k-1}}, B_{r^{k-1}})$. It follows from Theorem 2.4 that

$$\|\widehat{A}_kU\| \leq \frac{\|(B_1 - A_1)U\|}{1 - \|\Lambda\|^{r^{k-1}}}\|\Lambda\|^{r^{k-1}},$$

where $\|.\|$ is a matrix induced norm such that $\|\Lambda\| < 1$. Thus the sequence $\{\widehat{A}_kU\}$ converges to 0 with r-order r. For a full account of the definition of the rate of convergence, the reader is referred to [12].

4 Application of the AB-Algorithms for Solving the Matrix Square Root

In (11a) we see that the sequence $\{\widehat{A}_i U\}$ converges with r-order r to 0. We then in this section use this accelerated techniques to solve the quadratic matrix equation defined in (4), i.e., find the principle square root $\sqrt{S}$ of the matrix S with $\lambda(S) \subset \mathbb{C}^+$. To this end, we relate (4) to the generalized eigenvalue problem

$$A \begin{bmatrix} I_n \\ \sqrt{S} \end{bmatrix} = B \begin{bmatrix} I_n \\ \sqrt{S} \end{bmatrix} \sqrt{S}, \tag{12}$$

where $A = \begin{bmatrix} 0 & I_n \\ S & 0 \end{bmatrix}$ and $B = I_{2n}$. Since $\lambda(S) \subseteq \mathbb{C}^+$, there is no guarantee that the AB-algorithm will converge. To remedy this situation, this matrix $\sqrt{S}$ in (12) must be retreated. One way is to apply the Möbius transformation

$$\mathcal{C}_{\sqrt{S}}(\gamma I_n) = (\gamma I_n - \sqrt{S})(\gamma I_n + \sqrt{S})^{-1},$$

where $\gamma > 0$ and $-1 \notin \lambda(\gamma I_n - \lambda\sqrt{S})$, i.e., -1 is not an eigenvalue of the matrix pencil $\gamma I_n - \lambda\sqrt{S}$; that is, recast (12) in the following equation

$$A_1 \begin{bmatrix} I_n \\ \sqrt{S} \end{bmatrix} = B_1 \begin{bmatrix} I_n \\ \sqrt{S} \end{bmatrix} \mathcal{C}_{\sqrt{S}}(\gamma I_n), \tag{13}$$

where $A_1 = \gamma B - A$ and $B_1 = \gamma B + A$. Observe that $\rho(\mathcal{C}_{\sqrt{S}}(\gamma I_n)) < 1$ since $\lambda(S) \subseteq \mathbb{C}^+$. Upon using the AB-algorithm, it can be easily checked that for any integer $k \geq 1$, A_k and B_k can be expressed as

$$A_k = \begin{bmatrix} Q_k & -I_n \\ -S & Q_k \end{bmatrix}, \; B_k = \begin{bmatrix} Q_k & I_n \\ S & Q_k \end{bmatrix}, \tag{14}$$

respectively, where the sequence $\{Q_k\}$ satisfies $Q_i Q_j = Q_j Q_i$, for any integers $i, j > 0$, and the following iteration

$$Q_{k+1} = (\gamma Q_k + S)(\gamma I_n + Q_k)^{-1} \tag{15}$$

with $Q_1 = \gamma I_n$. Note that once $Q_k = \sqrt{S}$ for some k, it follows that $Q_\ell = \sqrt{S}$ for all $\ell \geq k$.

Specifically, let $\mathcal{C}_\gamma(\lambda) = \frac{\gamma - \lambda}{\lambda + \gamma}$ be the Möbius transformation with a parameter $\gamma \neq 0$ and $\lambda \neq -\gamma$. Then, the inverse scalar Möbius transformation can be written as

$$\mathcal{C}_\gamma^{-1}(\lambda) = \gamma \frac{1 - \lambda}{1 + \lambda}, \quad \lambda \neq -1.$$

Let $\lambda = e^{\frac{2j\pi i}{n}} \in S_n \backslash \{-1\}$, where $1 \leq j < n$. It follows that the real part of the square of $a := \mathcal{C}_\gamma^{-1}(\lambda)$ is a real negative number, since

$$a^2 = \gamma^2 (\frac{1 - e^{\frac{2j\pi i}{n}}}{1 + e^{\frac{2j\pi i}{n}}})^2 = -\gamma^2 \tan^2(\frac{j\pi}{n}) < 0. \tag{16}$$

From (16) and Theorem (2.2), it follows that the AB-algorithm will terminate prematurely only if $\lambda(S) \subseteq \mathbb{C}^-$; that is, once $\lambda(S) \subseteq \mathbb{C}^+$, or even, $\lambda(S) \subseteq \mathbb{C}^+ \cup \{0\}$, the sequence of matrix pencils $\{A_k - \lambda B_k\}$, initiated by (13), is well-defined.

With an eye on the structure of the matrix pencil $A_k - \lambda B_k$, we look for an accelerated iteration induced by the assumption of $\widehat{A}_k - \lambda \widehat{B}_k$ in Algorithm 3.1.

Algorithm 4.1. (Algorithm for solving matrix square root)

1. *Initialize a positive integer* $r > 1$ *and let* $\widehat{Q}_1 = Q_1 = \gamma I_n$;
2. *For* $i = 2, \ldots,$ *iterate*
$$\widehat{Q}_k := (S + \widehat{Q}_{k-1} Q_{k-1}^{(r-1)})(\widehat{Q}_{k-1} + Q_{k-1}^{(r-1)})^{-1},$$
until convergence, where $\widehat{Q}_{k-1}^{(r-1)}$ *is defined in step 4.*
3. *For* $\ell = 1, \cdots, r-2$, *iterate*
$$Q_{k-1}^{(\ell+1)} := (S + \widehat{Q}_{k-1} Q_{k-1}^{(\ell)})(\widehat{Q}_{k-1} + Q_{k-1}^{(\ell)})^{-1},$$
with $\widehat{Q}_{k-1}^{(1)} = \widehat{Q}_{k-1}$.

Note that $\widehat{Q}_k = Q_{r^{k-1}}$ for $k \geq 1$, and with the assumption of the existence of iterative sequences, we immediately have the following iterative formulae. We omit the proof here because the result can be straightforwardly shown by using induction.

Theorem 4.1. *Assume that the sequences generated by Algorithm 4.1 can be constructed with no break down. Then, we have the following two iterative formulae.*

1. *When* r *is even, let* $q = \frac{r}{2}$. *We have*
$$\widehat{Q}_{k+1} = (\sum_{j=0}^{q} \binom{r}{2j} \widehat{Q}_k^{r-2j} S^j)(\sum_{j=0}^{q-1} \binom{r}{2j+1} \widehat{Q}_k^{r-2j-1} S^j)^{-1}. \qquad (17)$$
2. *While* r *is odd, let* $q = \frac{r-1}{2}$. *We have*
$$\widehat{Q}_{k+1} = (\sum_{j=0}^{q} \binom{r}{2j} \widehat{Q}_k^{r-2j} S^j)(\sum_{j=0}^{q} \binom{r}{2j+1} \widehat{Q}_k^{r-2j-1} S^j)^{-1}, \qquad (18)$$

where the notation $\binom{n}{k}$ *denotes the number of* k*-combinations from the set* $S = \{1, 2, \cdots, n\}$ *of* n *elements.*

Note that if S is a nonsingular matrix, then (17) and (18) can be simply expressed by the following rule:

$$\widehat{Q}_{k+1} = V_m U_m^{-1},$$

where

$$V_m = \sum_{j=0}^{[\frac{m}{2}]} \binom{m}{2j} \widehat{Q}_k^{m-2j} S^j = \frac{1}{2}((\widehat{Q}_k + \sqrt{S})^m + (\widehat{Q}_k - \sqrt{S})^m),$$

$$U_m = \sum_{j=0}^{[\frac{m-1}{2}]} \binom{m}{2j+1} \widehat{Q}_k^{m-2j-1} S^j = \frac{(\sqrt{S})^{-1}}{2}((\widehat{Q}_k + \sqrt{S})^m - (\widehat{Q}_k - \sqrt{S})^m).$$

Importantly, under nonsingularity assumption, a strong result related to the sequences $\{\mathcal{C}_{\sqrt{S}}(Q_i)\}$ and $\{\mathcal{C}_{\sqrt{S}}(\widehat{Q}_i)\}$ hold.

Lemma 4.1. *Suppose that S is nonsingular. Let i, j, and k be any positive integers, and $1 \le i, j \le k$. Then the following properties hold.*

1. *For the sequence $\{Q_k\}$, we have*
   a. $Q_k = \sqrt{S}(I_n + \mathcal{C}_{\sqrt{S}}(Q_1)^k)(I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^k)^{-1}$,
   b. $\mathcal{C}_{\sqrt{S}}(Q_i)^j = \mathcal{C}_{\sqrt{S}}(Q_j)^i$.
2. *For the sequence $\{\widehat{Q}_k\}$, we have*
   a. $\widehat{Q}_k = \sqrt{S}(I_n + \mathcal{C}_{\sqrt{S}}(\widehat{Q}_1)^{r^{k-1}})(I_n - \mathcal{C}_{\sqrt{S}}(\widehat{Q}_1)^{r^{k-1}})^{-1}$,
   b. $\mathcal{C}_{\sqrt{S}}(\widehat{Q}_i)^{r^{k-i}} = \mathcal{C}_{\sqrt{S}}(\widehat{Q}_j)^{r^{k-j}}$.

Proof. It follows from Theorem 2.4 and $Q_1 = \gamma I_n$ that

$$A_k U(I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^k) = (B_1 - A_1)U\mathcal{C}_{\sqrt{S}}(Q_1)^k. \tag{19}$$

Then, (14) and (19) yield

$$(Q_k - \sqrt{S})(I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^k) = 2\sqrt{S}\mathcal{C}_{\sqrt{S}}(Q_1)^k. \tag{20}$$

By adding $2\sqrt{S}(I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^k)$ to both sides of (20), we have

$$(Q_k + \sqrt{S})(I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^k) = 2\sqrt{S}. \tag{21}$$

From (20) and (21) together, it must be that

$$Q_k(I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^k) = \sqrt{S}(I_n + \mathcal{C}_{\sqrt{S}}(Q_1)^k).$$

Since S is nonsingular, it follows that $1 \notin \lambda(\mathcal{C}_{\sqrt{S}}(Q_1))$ so that

$$Q_k = \sqrt{S}(I_n + \mathcal{C}_{\sqrt{S}}(Q_1)^k)(I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^k)^{-1},$$

which is equivalent to

$$\mathcal{C}_{\sqrt{S}}(Q_1)^k = \mathcal{C}_{\sqrt{S}}(Q_k).$$

Since k is an arbitrary positive integer, we have

$$(\mathcal{C}_{\sqrt{S}}(Q_i))^j = \mathcal{C}_{\sqrt{S}}(Q_1)^{ij} = (\mathcal{C}_{\sqrt{S}}(Q_j))^i,$$

for $1 \leq i, j \leq k$.

Also, by Theorem 2.4, Algorithm 3.1, $\widehat{Q}_1 = \gamma I_n$, we have

$$\widehat{A}_k U = \widehat{B}_k U (\mathcal{C}_{\sqrt{S}}(\widehat{Q}_1))^{r^{k-1}},$$

which then completes the proof of part 2a. and part 2b. by applying the same strategies as above. □

Indeed, this iteration in Algorithm 4.1 converges with q-order r for solving the principle square root of a matrix S.

Theorem 4.2. *Suppose that S is a nonsingular matrix. Let $\|.\|$ be a matrix induced norm such that $\|\mathcal{C}_{\sqrt{S}}(\widehat{Q}_1)\| < 1$. Then,*

$$\|\widehat{Q}_{k+1} - \sqrt{S}\| \leq \mu \|\widehat{Q}_k - \sqrt{S}\|^r,$$

for some $\mu > 0$; that is, $\widehat{Q}_k \to \sqrt{S}$ with q-order r.

Proof. Using (20) and $\widehat{Q}_k = Q_{r^{k-1}}$, we see that

$$\widehat{Q}_k - \sqrt{S} = 2\sqrt{S}\mathcal{C}_{\sqrt{S}}(Q_1)^{r^{k-1}}(I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^{r^{k-1}})^{-1}.$$

Without loss of generality we assume that $\widehat{Q}_k \neq \sqrt{S}$ for all k. Otherwise, $\widehat{Q}_\ell = \sqrt{S}$ for all $\ell \geq k$. It follows that

$$\begin{aligned}
\frac{\|\widehat{Q}_{k+1} - \sqrt{S}\|}{\|\widehat{Q}_k - \sqrt{S}\|^r} &\leq \frac{\|2\sqrt{S}\mathcal{C}_{\sqrt{S}}(Q_1)^{r^k}\| \|I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^{r^{k-1}}\|^r}{\|2\sqrt{S}\mathcal{C}_{\sqrt{S}}(Q_1)^{r^{k-1}}\|^r (1 - \|\mathcal{C}_{\sqrt{S}}(Q_1)^{r^k}\|)} \\
&\leq \frac{2\|\sqrt{S}\| \|\mathcal{C}_{\sqrt{S}}(Q_1)^{r^{k-1}}\|^r \|I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^{r^{k-1}}\|^r}{2^r \frac{\|\mathcal{C}_{\sqrt{S}}(Q_1)^{r^{k-1}}\|^r}{\|(\sqrt{S})^{-1}\|^r} (1 - \|\mathcal{C}_{\sqrt{S}}(Q_1)^{r^k}\|)} \\
&\leq 2^{1-r} \|\sqrt{S}\| \|(\sqrt{S})^{-1}\|^r \sup_{k \geq 1} \frac{(1 + \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^{r^{k-1}})^r}{1 - \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^{r^k}} \\
&\leq \mu := \frac{2\|\sqrt{S}\|}{1 - \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^r} \|(\sqrt{S})^{-1}\|^r < \infty.
\end{aligned}$$

□

However, the original sequence $\{Q_k\}$ only converges to $\sqrt{S}$ q-linearly.

Theorem 4.3. *Suppose that S is a nonsingular matrix. Let $\|.\|$ be a matrix induced norm such that $\|\mathcal{C}_{\sqrt{S}}(\widehat{Q}_1)\| < 1$. Then,*

$$\|Q_{k+1} - \sqrt{S}\| \leq \mu \|Q_k - \sqrt{S}\|,$$

for some $\mu \in (0,1)$ and sufficient large k; that is, $Q_k \to \sqrt{S}$ q-linearly with q-factor μ.

Proof. From (20), we have

$$Q_k - \sqrt{S} = 2\sqrt{S}\mathcal{C}_{\sqrt{S}}(Q_1)^k (I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^k)^{-1}.$$

Thus,

$$\begin{aligned} \|Q_{k+1} - \sqrt{S}\| &= \|(Q_k - \sqrt{S})(I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^k)\mathcal{C}_{\sqrt{S}}(Q_1)(I_n - \mathcal{C}_{\sqrt{S}}(Q_1)^{k+1})^{-1}\| \\ &\leq \|\mathcal{C}_{\sqrt{S}}(Q_1)\| \frac{1 + \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^k}{1 - \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^{k+1}} \|Q_k - \sqrt{S}\|. \end{aligned}$$

Since $\dfrac{1 + \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^k}{1 - \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^{k+1}} \to 1$ as $k \to \infty$, there exists a constant k_0 such that

$$\|\mathcal{C}_{\sqrt{S}}(Q_1)\| \frac{1 + \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^k}{1 - \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^{k+1}} < 1$$

for $k \geq k_0$. Let $\mu = \dfrac{1 + \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^{k_0}}{1 - \|\mathcal{C}_{\sqrt{S}}(Q_1)\|^{k_0+1}} \|\mathcal{C}_{\sqrt{S}}(Q_1)\|$, which completes the proof. □

In the next result, we show that the AB-algorithm still converges, while solving the square root of a singular matrix, which is hard to be handled in general. See [18] for further discussion.

Corollary 4.1. *Suppose that S is a singular matrix having $\lambda(S) \subseteq \mathbb{C}^+ \cup \{0\}$ and the null eigenvalues are semisimple. Then,*

1. *$Q_k \to \sqrt{S}$ sublinearly,*
2. *$\widehat{Q}_k \to \sqrt{S}$ q-linearly with q-factor $\frac{1}{r}$.*

Proof. Let P be an invertible matrix so that $\mathrm{diag}(0_p, J_{n-p}) = P\sqrt{S}P^{-1}$ be the Jordan canonical form of $\sqrt{S}$ with $\lambda(J_{n-p}) \subset \mathbb{C}^+$. Upon the use of substitution and Lemma 4.1, we have

$$PQ_kP^{-1} = \mathrm{diag}(Q_k^{(11)}, Q_k^{(22)}),$$

where $Q_k^{(11)}$ is derived directly by (15) and $Q_k^{(22)}$ is followed from Lemma 4.1 such that

$$\begin{aligned} Q_k^{(11)} &= \frac{\gamma I_n}{k}, \\ Q_k^{(22)} &= J_{n-p}(I + \mathcal{C}_{J_{n-p}}(\gamma I_{n-p})^k)(I - \mathcal{C}_{J_{n-p}}(\gamma I_{n-p})^k)^{-1}. \end{aligned}$$

Since $\{Q_k^{(11)}\}$ converges to zero sublinearly and $\{Q_k^{(22)}\}$ converges to zero q-linearly, $\{Q_k\}$ converges to $\sqrt{S}$ sublinearly. In the similar way, we have

$$P\widehat{Q}_k P^{-1} = \mathrm{diag}(\widehat{Q}_k^{(11)}, \widehat{Q}_k^{(22)}),$$

where

$$\begin{aligned}\widehat{Q}_k^{(11)} &= \frac{\gamma I_n}{r^k},\\ \widehat{Q}_k^{(22)} &= J(I_n + \mathcal{C}_J(\gamma I_n)^{r^{k-1}})(I_n - \mathcal{C}_J(\gamma I_n)^{r^{k-1}})^{-1}.\end{aligned}$$

Since $\{\widehat{Q}_k^{(11)}\}$ converges to zero q-linearly with q-factor r, it follows that $\{\widehat{Q}_k\}$ converges to $\sqrt{S}$ q-linearly with q-factor r.

□

5 Conclusion

We have given an algorithm to solve the stable subspace of a matrix pencil $A - \lambda B$, in items of the left null space of a given iteration. This algorithm is shown to preserve a discrete-type flow property. With this property, it allows us to rebuild this algorithm so that its convergence can be up to any desired order. Since the solution of the matrix square root can be interpreted in terms of the stable subspace of a matrix pencil, this interpretation then allows us to compute the result through our AB-algorithm. We show in theory that the speed of convergence has q-order r, and even more, for the singular case, where S is singular having no negative real eigenvalues, and the null eigenvalues are semisimple, the iteration still succeeds with a linear rate of convergence.

Acknowledgment

This research work is partially supported by the Ministry of Science and Technology and the National Center for Theoretical Sciences in Taiwan.

References

[1] Zhaojun Bai, James Demmel, and Ming Gu. An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. *Numer. Math.*, 76(3):279–308, 1997.

[2] Zhaojun Bai and James W. Demmel. On swapping diagonal blocks in real Schur form. *Linear Algebra Appl.*, 186:73–95, 1993.

[3] D. S. Bernstein. *Matrix mathematics:Theory, facts, and formulas with application to linear systems theory.* Princeton University Press, Princeton, NJ, 2005.

[4] Dario A. Bini, Bruno Iannazzo, and Beatrice Meini. *Numerical solution of algebraic Riccati equations*, volume 9 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2012.

[5] J. W. Demmel. Trading off parallelism and numerical stability. In *Linear algebra for large scale and real-time applications (Leuven, 1992)*, volume 232 of *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, pages 49–68. Kluwer Acad. Publ., Dordrecht, 1993.

[6] J. W. Demmel and X. Li. Faster numerical algorithms via exception handling. In *Computer Arithmetic, 1993. Proceedings., 11th Symposium on*, pages 234–241, Jun 1993.

[7] C.R. DePrima and C.R. Johnson. The range of $A^{-1}A^*$ in $GL(n, C)$. *Linear Algebra Appl.*, 9:209 − 222, 1974.

[8] Greg Henry and Robert van de Geijn. Parallelizing the QR algorithm for the unsymmetric algebraic eigenvalue problem: myths and reality. *SIAM J. Sci. Comput.*, 17(4):870–883, 1996.

[9] Nicholas J. Higham. Newton's method for the matrix square root. *Math. Comp.*, 46(174):537–549, 1986.

[10] Nicholas J. Higham. Computing real square roots of a real matrix. *Linear Algebra Appl.*, 88/89:405–430, 1987.

[11] Nicholas J. Higham. Stable iterations for the matrix square root. *Numer. Algorithms*, 15(2):227–242, 1997.

[12] C. T. Kelley. *Iterative methods for linear and nonlinear equations*, volume 16 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995. With separately available software.

[13] V. N. Kublanovskaya. AB-algorithm and its modifications for the spectral problems of linear pencils of matrices. *Numer. Math.*, 43(3):329–342, 1984.

[14] Pentti Laasonen. On the iterative solution of the matrix equation $AX^2 - I = 0$. *Math. Tables Aids Comput.*, 12:109–116, 1958.

[15] A. N. Malyshev. Guaranteed accuracy in spectral problems of linear algebra. I. [translation of Trudy Inst. Mat. (Novosibirsk) **17** (1990), Vychisl. Metody Lineĭn. Algebry, 19–104; MR1067523 (92f:65048)]. *Siberian Adv. Math.*, 2(1):144–197, 1992. Siberian Advances in Mathematics.

[16] A. N. Malyshev. Guaranteed accuracy in spectral problems of linear algebra. II. *Siberian Adv. Math.*, 2(2):153–204, 1992. Siberian Advances in Mathematics.

[17] Alexander N. Malyshev. Parallel algorithm for solving some spectral problems of linear algebra. *Linear Algebra Appl.*, 188/189:489–520, 1993.

[18] Beatrice Meini. The matrix square root from a new functional perspective: theoretical results and computational issues. *SIAM J. Matrix Anal. Appl.*, 26(2):362–376 (electronic), 2004/05.

[19] Sho Mizuno, Yosuke Moriizumi, Tsuyoshi S. Usuda, and Tomohiro Sogabe. An initial guess of Newton's method for the matrix square root based on a sphere constrained optimization problem. *JSIAM Lett.*, 8:17–20, 2016.