Springer INdAM Series 30

Dario Andrea Bini
Fabio Di Benedetto
Eugene Tyrtyshnikov
Marc Van Barel *Editors*

Structured Matrices in Numerical Linear Algebra

Analysis, Algorithms and Applications

Springer

Springer INdAM Series

Volume 30

More information about this series at http://www.springer.com/series/10283

Dario Andrea Bini • Fabio Di Benedetto •
Eugene Tyrtyshnikov • Marc Van Barel
Editors

Structured Matrices in Numerical Linear Algebra

Analysis, Algorithms and Applications

Springer

Editors
Dario Andrea Bini
Department of Mathematics
University of Pisa
Pisa, Italy

Fabio Di Benedetto
Department of Mathematics
University of Genoa
Genoa, Italy

Eugene Tyrtyshnikov
Institute of Numerical Mathematics
Russian Academy of Sciences
Moscow, Russia

Marc Van Barel
Department of Computer Science
KU Leuven
Heverlee, Belgium

ISSN 2281-518X ISSN 2281-5198 (electronic)
Springer INdAM Series
ISBN 978-3-030-04087-1 ISBN 978-3-030-04088-8 (eBook)
https://doi.org/10.1007/978-3-030-04088-8

Library of Congress Control Number: 2019936552

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Solving a mathematical model, by means of computational techniques in general or more specifically by means of numerical analysis, is ultimately reduced to solving problems in linear algebra. In fact, very often models are linear by their nature, while sometimes they are nonlinear but their solution is achieved by means of linearization. This to some extent explains why numerical linear algebra and matrix analysis have undergone such extensive development in recent decades.

In this process, one typically encounters problems like solving linear systems, or solving standard or generalized eigenvalue problems, as well as matrix polynomial equations or polynomial eigenvalue problems where the size of the matrices involved in the model is extremely large or in some cases even infinite. For such problems standard (general-purpose) algorithms cannot work due to their extreme complexity; therefore, one has to exploit the specific properties which originate from the peculiar features of the model. In the language of linear algebra, these properties are translated in terms of *structures* that the matrices involved in the model share; often, structured matrices reveal themselves in a clear form and appear to show all their properties immediately. Sometimes, however, structures are hidden and difficult to discover, and their properties seem hardly exploitable. Their analysis and exploitation is not just a challenge but also a mandatory step which is necessary to design highly effective ad hoc algorithms for the solution of large-scale problems from applications. In fact, general-purpose algorithms, say Gaussian elimination for solving linear systems, cannot be used to solve problems of large size while a smart exploitation of the available structures enables one to design effective solution algorithms even for problems of a huge size.

The importance of matrix structures has grown over the years. Analyzing structures from the theoretical point of view, turning them into effective solution algorithms, constructing software which implements the algorithms, and verifying its effectiveness by direct computation is one of the most exciting challenges that covers abstract theory, design and analysis of algorithms, software implementation, and applications.

This volume presents a selected number of peer-reviewed papers concerning structured matrix analysis and its applications. The topics discussed concern theory,

algorithms, and applications in which structured matrices are involved. The subjects range from abstract topics such as the theory of generalized locally (block) Toeplitz matrices and the analysis of matrix subspaces and quadratic kernels to more numerical issues such as error analysis of algorithms for tensor manipulation and analysis of the derivative of matrix geometric means. Moreover, other structured oriented topics are developed, e.g., analysis of companion pencil and block Fiedler companion matrices, together with analysis of the tridiagonal symmetric eigenvalue problem, computation of bivariate matrix functions, and solution of the saddle point problem. Among the applications are analysis of the stability of gyroscopic systems, numerical solution of 2D hard scattering problems of damped waves, fractional reaction-diffusion equations, and the problem of multi-frame super-resolution reconstruction from video clips.

All the papers correspond to talks presented at the INdAM meeting *Structured Matrices in Numerical Linear Algebra: Analysis, Algorithms and Applications* held in Cortona, Italy, on September 4–8, 2017.

This workshop aimed to continue in both form and spirit the series of conferences on *Structured Matrices* and their applications held in Cortona, Italy, every 4 years between 1996 and 2008 and continued in Leuven, Belgium, in September 2012 and in Kalamata, Greece, in September 2014.

The book will be of interest to graduate students in mathematics and researchers in numerical linear algebra and scientific computing, as well as engineers and applied mathematicians.

Pisa, Italy — Dario Andrea Bini
Genoa, Italy — Fabio Di Benedetto
Moscow, Russia — Eugene Tyrtyshnikov
Heverlee, Belgium — Marc Van Barel
October 2018

Contents

About the Editors

Dario Andrea Bini, a Full Professor of Numerical Analysis since 1986, has held a permanent position at the University of Pisa since 1989. His research mainly focuses on numerical linear algebra problems, on structured matrix analysis and on the design and analysis of algorithms for polynomial and matrix computations. The author of three research books and more than 120 papers, he also serves on the editorial boards of three international journals.

Fabio Di Benedetto is an Associate Professor of Numerical Analysis at the Department of Mathematics of the University of Genoa, where he teaches courses on Numerical Analysis for undergraduate and graduate students, since 2000. His main research interests concern the solution of large-scale numerical linear algebra problems, with special attention to structured matrices analysis with applications to image processing. He is the author of more than 30 papers.

Eugene Tyrtyshnikov, Professor and Chairman at the Lomonosov Moscow State University, is a Full Member of the Russian Academy of Sciences and Director of the Institute of Numerical Mathematics of the Russian Academy of Sciences, Moscow. He completed his Ph.D. in Numerical Mathematics at Moscow State University, and his postdoctoral studies at the Siberian Branch of the Russian Academy of Sciences, Novosibirsk. His research interests concern numerical analysis, linear and multilinear algebra, approximation theory and related applications. He is the associate editor of many international journals and the author of more than 100 papers and 8 books.

Marc Van Barel received his Ph.D. in Computer Engineering (Numerical Analysis and Applied Mathematics) from the KU Leuven, where he is currently a Full Professor at the Department of Computer Science. His work mainly focuses on numerical (multi-)linear algebra, approximation theory, orthogonal functions and their applications in systems theory, signal processing, machine learning, etc. He is the author or co-author of more than 140 papers and 4 books. Currently, he serves on the editorial boards of three international journals.

Spectral Measures

Giovanni Barbarino

Abstract The theory of spectral symbols links sequences of matrices with measurable functions expressing their asymptotic eigenvalue distributions. Usually, a sequence admits several spectral symbols, and it is not clear if a canonical one exists. Here we present a way to connect the sequences with the space of probability measure, so that each sequence admits a uniquely determined measure. The methods used are similar to those employed in the theory of generalized locally Toeplitz (GLT) sequences: a goal of this present contribution is in fact that of explaining how the two concepts are connected.

Keywords Probability measures · Generalized locally Toeplitz sequences · Complete pseudo-metrics · Ergodic formula

1 Introduction

A *matrix sequence* is an ordered collection of complex valued matrices with increasing size, and is usually denoted as $\{A_n\}_n$, where $A_n \in \mathbb{C}^{n\times n}$. We will refer to the space of matrix sequences with the notation

$$\mathscr{E} := \{\{A_n\}_n : A_n \in \mathbb{C}^{n\times n}\}.$$

It is often observed in practice that matrix sequences, $\{A_n\}_n$, generated by discretization methods applied to linear differential equations possess a *spectral symbol*, that is a measurable function describing the asymptotic distribution of the

G. Barbarino (✉)
Scuola Normale Superiore, Pisa, Italy
e-mail: giovanni.barbarino@sns.it

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_1

eigenvalues of A_n. We recall that a spectral symbol associated with a sequence $\{A_n\}_n$ is a measurable function $k : D \subseteq \mathbb{R}^n \to \mathbb{C}$ satisfying

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} F(\lambda_i(A_n)) = \frac{1}{l(D)} \int_D F(k(x))dx$$

for every continuous function $F : \mathbb{C} \to \mathbb{C}$ with compact support, where D is a measurable set with finite Lebesgue measure $l(D) > 0$ and $\lambda_i(A_n)$ are the eigenvalues of A_n. In this case we write

$$\{A_n\}_n \sim_\lambda k(x).$$

We can also consider the singular values of the matrices instead of the eigenvalues. In the same setting, if

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} F(\sigma_i(A_n)) = \frac{1}{l(D)} \int_D F(|k(x)|)dx$$

for every continuous function $F : \mathbb{R} \to \mathbb{C}$ with compact support, where $\sigma_i(A_n)$ are the singular values of A_n, then $\{A_n\}_n$ possesses a *singular value symbol*, and we write

$$\{A_n\}_n \sim_\sigma k(x).$$

The space of matrix sequences is a complete pseudometric space when endowed with a pseudometric inducing the *approximating classes of sequences* (acs) convergence, that we will redefine in the next section. One fundamental property of this metric is that it identifies sequences that differ by a sequence admitting zero as singular value symbol (called zero-distributed sequences). In particular, it has been shown that such sequences share the same singular value symbol, but the distance between two sequences with the same singular value symbol is not usually zero.

The main observation of this note is that for any measurable function $k(x)$, the operator

$$\phi(F) := \int_D F(k(x))dx \qquad \phi : C_c(\mathbb{C}) \to \mathbb{C}$$

is linear and continuous and can be represented by a unique probability measure μ. We call μ a *spectral measure*, and we associate it with any sequence $\{A_n\}_n$ that has $k(x)$ as spectral symbol. It turns out that if a sequence admits a spectral measure, then it is uniquely determined, differently from the spectral symbols. The space of probability spectral measures is moreover a complete metric space with the Lévy–Prokhorov distance π, and it corresponds to a pseudometric d' on matrices called

modified optimal matching distance. The main result is that d' identifies sequences admitting the same spectral symbol, differently from the acs distance.

Theorem 1 *If* $\{A_n\}_n \sim_\lambda f(x)$, *then*

$$\{B_n\}_n \sim_\lambda f(x) \iff d'(\{A_n\}_n, \{B_n\}_n) = 0.$$

A different approach to the uniqueness problem for the spectral symbol is embodied in the theory of GLT sequences. For specific sequences, called *generalized locally Toeplitz* (GLT) sequences, we can choose one of their symbols, and denote it as *GLT symbol* of the sequence

$$\{A_n\}_n \sim_{GLT} k(x, \theta).$$

In the case of diagonal matrix sequences, the choice of one symbol can be seen as a particular sorting of their eigenvalues, as expressed in the following theorem, proved in the last section, and which represents a generalization of the results in [3].

Theorem 2 *Given a diagonal sequence* $\{D_n\}_n$ *and one of its spectral symbols* $k : [0, 1] \to \mathbb{C}$, *then*

$$\{P_n D_n P_n^T\} \sim_{GLT} k(x) \otimes 1$$

for some P_n *permutation matrices.*

The paper is organized in the following way: In Sect. 2 we recall basic definitions such as the acs convergence, the optimal matching distance d, and the theory of GLT sequences. Moreover, we define the modified optimal matching distance d' since it is a slight variation of d, and we discuss how it is connected to d_{acs}. In Sect. 3 we introduce the spectral measures and we study their relationships with the spectral symbols. In particular, we notice how the vague convergence and the Lévy–Prokhorov distance π on the probability measures lead to a reformulation of the definition of spectral symbol/measure. In Sect. 4, we prove that the pseudometrics π and d' are actually equivalent, and we explain how this fact leads to the proofs of the above reported theorems.

2 Prerequisites

2.1 Complete Pseudometrics

The space of matrix sequences that admit a spectral symbol on a fixed domain D has been shown to be closed with respect to a notion of convergence called the approximating classes of sequences (acs) convergence. This notion and this result are due to Serra-Capizzano [11], but were actually inspired by Tilli's pioneering

paper on LT sequences [12]. Given a sequence of matrix sequences $\{B_{n,m}\}_{n,m}$, it is said to be acs convergent to $\{A_n\}_n$ if there exists a sequence $\{N_{n,m}\}_{n,m}$ of "small norm" matrices and a sequence $\{R_{n,m}\}_{n,m}$ of "small rank" matrices such that for every m there exists n_m with

$$A_n = B_{n,m} + N_{n,m} + R_{n,m}, \qquad \|N_{n,m}\| \leq \omega(m), \qquad \mathrm{rk}(R_{n,m}) \leq nc(m)$$

for every $n > n_m$, and

$$\omega(m) \xrightarrow{m\to\infty} 0, \qquad c(m) \xrightarrow{m\to\infty} 0.$$

In this case, we will use the notation $\{B_{n,m}\}_{n,m} \xrightarrow{acs} \{A_n\}_n$.

This notion of convergence has been shown to be metrizable on the whole space $\mathscr{E}$. Given a matrix $A \in \mathbb{C}^{n\times n}$, we can define the function

$$p(A) := \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + \sigma_i(A) \right\},$$

where $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_n(A)$ are the singular values of A, and by convention $\sigma_{n+1}(A) = 0$. The function $p(A)$ is subadditive, so we can introduce the pseudometric d_{acs} on the space of matrix sequences

$$d_{acs}(\{A_n\}_n, \{B_n\}_n) = \limsup_{n\to\infty} p(A_n - B_n).$$

It has been proved [6, 8] that this distance induces the acs convergence already introduced. In other words,

$$d_{acs}\left(\{A_n\}_n, \{B_{n,m}\}_{n,m}\right) \xrightarrow{m\to\infty} 0 \iff \{B_{n,m}\}_{n,m} \xrightarrow{acs} \{A_n\}_n.$$

One fundamental property of this metric is that it identifies sequences whose difference admits zero as singular value symbol (called zero-distributed sequence). In particular, it has been shown that such sequences share the same singular value symbol, in case one of them admits singular value symbol.

Lemma 1 *Let $\{A_n\}_n, \{B_n\}_n \in \mathscr{E}$. We have*

$$\{A_n - B_n\}_n \sim_\sigma 0 \iff d_{acs}(\{A_n\}_n, \{B_n\}_n) = 0.$$

In this case, if $k : D \subseteq \mathbb{R}^n \to \mathbb{C}$ where D is a measurable set with finite Lebesgue measure $l(D) > 0$, then

$$\{A_n\}_n \sim_\sigma k(x) \iff \{B_n\}_n \sim_\sigma k(x).$$

In [2], it has been first proved that the pseudometric d_{acs} on the space of matrix sequences is complete. In Theorem 2.2 of [4], we find sufficient conditions for a pseudometric on $\mathscr{E}$ to be complete. Here we need a different result, but the proof is almost identical.

Lemma 2 *Let d_n be pseudometrics on the space of matrices $\mathbb{C}^{n\times n}$ bounded by the same constant $L > 0$ for every n. Then the function*

$$d(\{A_n\}_n, \{B_n\}_n) := \limsup_{n\to\infty} d_n(A_n, B_n)$$

is a complete pseudometric on the space of matrix sequences.

2.2 Optimal Matching Distance

Let $v, w \in \mathbb{C}^n$ be vectors with components

$$v = [v_1, v_2, \ldots, v_n], \quad w = [w_1, w_2, \ldots, w_n].$$

We recall the pseudometric on $\mathbb{C}^n$ called *optimal matching distance* defined in Bhatia's book [5].

Definition 1 Given $v, w \in \mathbb{C}^n$, the pseudometric of the optimal matching distance is defined as

$$d(v, w) := \min_{\sigma\in S_n} \max_{i=1,\ldots,n} |v_i - w_{\sigma(i)}|,$$

where S_n is the symmetric group of permutation of n objects.

Given $A \in \mathbb{C}^{n\times n}$, let $\Lambda(A) \in \mathbb{C}^n$ be the vector of the eigenvalues. We can extend the distance d to matrices in the following way.

Definition 2 Given $A, B \in \mathbb{C}^{n\times n}$, we define

$$d(A, B) := d(\Lambda(A), \Lambda(B)).$$

Notice that the order of the eigenvalues in $\Lambda(A)$ and $\Lambda(B)$ does not affect the quantity $d(A, B)$. It is easy to see that d is still a pseudometric on $\mathbb{C}^{n\times n}$. This is still not enough for our purposes, since we want a distance that sees two matrices differing for few eigenvalues as very similar. For this reason, we modify the previous metric, and we introduce a new function d' called *modified optimal matching distance*.

Definition 3 Given $v, w \in \mathbb{C}^n$, the modified optimal matching distance is defined as

$$d'(v, w) := \min_{\sigma \in S_n} \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |v - w_\sigma|_i^{\downarrow} \right\},$$

where

$$|v - w_\sigma| = [|v_1 - w_{\sigma(1)}|, |v_2 - w_{\sigma(2)}|, \dots, |v_n - w_{\sigma(n)}|]$$

and $|v - w_\sigma|_i^{\downarrow}$ is the i-th greatest element in $|v - w_\sigma|$, with the convention $|v - w_\sigma|_{n+1}^{\downarrow} := 0$.

Given $A, B \in \mathbb{C}^{n \times n}$, we define

$$d'(A, B) := d'(\Lambda(A), \Lambda(B))$$

and if $\{A_n\}_n, \{B_n\}_n \in \mathscr{E}$, we can also define

$$d'(\{A_n\}_n, \{B_n\}_n) := \limsup_{n \to \infty} d'(A_n, B_n).$$

Notice that $d'(v, w) \leq 1$ for every $v, w \in \mathbb{C}^n$, so $d'(A, B) \leq 1$ for every pair of matrices of the same size, and $d'(\{A_n\}_n, \{B_n\}_n) \leq 1$ for every pair of sequences $\{A_n\}_n, \{B_n\}_n \in \mathscr{E}$. We referred to d' as a distance, but we need to prove it.

Lemma 3 *The function d' is a complete pseudometric on $\mathscr{E}$.*

Proof Let us prove that d' is a pseudometric on $\mathbb{C}^n$. First, it is easy to see that $d'(v, w)$ is always a finite nonnegative real number, and it is symmetric since

$$\begin{aligned} d'(v, w) &= \min_{\sigma \in S_n} \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |v - w_\sigma|_i^{\downarrow} \right\} \\ &= \min_{\sigma \in S_n} \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |w - v_{\sigma^{-1}}|_i^{\downarrow} \right\} = d'(w, v). \end{aligned}$$

Moreover, given any $\tau \in S_n$, we have

$$\begin{aligned} d'(v, w) &= \min_{\sigma \in S_n} \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |v - w_\sigma|_i^{\downarrow} \right\} \\ &= \min_{\sigma \in S_n} \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |v_\tau - w_{\sigma\tau}|_i^{\downarrow} \right\} = d'(v_\tau, w), \end{aligned}$$

so we can permute the elements of the vectors as we like. Let $v, w, z \in \mathbb{C}^n$ and let us sort their elements in such a way that

$$d'(v,w) = \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |v_i - w_i| \right\},$$
$$d'(w,z) = \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |w - z|_i^{\downarrow} \right\},$$

meaning that the permutation realizing the minimum in both cases is the identity, and that $|v_i - w_i| \geq |v_j - w_j|$ whenever $i \leq j$. Moreover, let s, r, q be the greatest indices that satisfy

$$d'(v,w) = \frac{s-1}{n} + |v_s - w_s|, \qquad d'(w,z) = \frac{r-1}{n} + |w_q - z_q|.$$

Let I, J be two sets of indices defined as

$$I = \{1, 2, \dots, s-1\}, \qquad J = \{j : |w_j - z_j| > |w_q - z_q|\}.$$

Notice that $\#I = s-1$ and $\#J = r-1$. Let us consider two cases.

- Suppose $I \cup J = \{1, \dots, n\}$. We obtain that

  $$\#I + \#J = r + s - 2 \geq n$$

  and hence

  $$d'(v,z) \leq 1 \leq \frac{s-1}{n} + \frac{r-1}{n} \leq d'(v,w) + d'(w,z).$$

- Suppose $I \cup J \neq \{1, \dots, n\}$. Let k be the index not belonging to $I \cup J$ that maximizes $|v_i - z_i|$. If we consider the identity permutation, we deduce that

  $$d'(v,z) \leq \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |v - z|_i^{\downarrow} \right\},$$

  but the number of indices such that $|v_i - z_i|$ is greater than $|v_k - z_k|$ is at most $\#I \cup J \leq r + s - 2$, and consequently

  $$d'(v,z) \leq \frac{r+s-2}{n} + |v_k - z_k|.$$

  The index k does not belong to I or to J, so

  $$|v_k - w_k| \leq |v_s - w_s|, \qquad |w_k - z_k| \leq |w_q - z_q|.$$

From the latter we infer that

$$\begin{aligned} d'(v,z) &\leq \frac{r+s-2}{n} + |v_k - z_k| \\ &\leq \frac{s-1}{n} + |v_k - w_k| + \frac{r-1}{n} + |w_k - z_k| \\ &\leq \frac{s-1}{n} + |v_s - w_s| + \frac{r-1}{n} + |w_q - z_q| \\ &= d'(v,w) + d'(w,z). \end{aligned}$$

This shows that d' is a pseudometric on $\mathbb{C}^n$ and consequently it is a pseudometric even on $\mathbb{C}^{n\times n}$. Thanks to Lemma 2, we can conclude that d' is a complete pseudometric on $\mathscr{E}$. □

In the general case, the two pseudometrics have no common features, but, when dealing with diagonal matrices, we can prove the following lemma.

Lemma 4 *Given $\{D_n\}_n, \{D'_n\}_n \in \mathscr{E}$ sequences of diagonal matrices, there exists a sequence $\{P_n\}_n$ of permutation matrices such that*

$$d'(\{D'_n\}_n, \{D_n\}_n) = d_{acs}(\{D'_n\}_n, \{P_n D_n P_n^T\}_n).$$

Proof Let v^n and v'^n be the vectors of the ordered diagonal entries of D_n and D'_n, so that

$$v^n_i := [D_n]_{i,i}, \qquad v'^n_i := [D'_n]_{i,i}.$$

Let $\tau_n \in S_n$ be the permutations satisfying

$$\begin{aligned} d'(D'_n, D_n) &= \min_{\sigma\in S_n} \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |v'^n - v^n_\sigma|^\downarrow_i \right\} \\ &= \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |v'^n - v^n_{\tau_n}|^\downarrow_i \right\}. \end{aligned}$$

Let also P_n be the permutation matrices associated with τ_n. We know that

$$\begin{aligned} p(D'_n - P_n D_n P_n^T) &= \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + \sigma_i(D'_n - P_n D_n P_n^T) \right\} \\ &= \min_{i=1,\dots,n+1} \left\{ \frac{i-1}{n} + |v'^n - v^n_{\tau_n}|^\downarrow_i \right\} \\ &= d'(D'_n, D_n). \end{aligned}$$

As a consequence

$$d_{acs}(\{D'_n\}_n, \{P_n D_n P_n^T\}_n) = \limsup_{n\to\infty} p(D'_n - P_n D_n P_n^T)$$
$$= \limsup_{n\to\infty} d'(D'_n, D_n) = d'(\{D'_n\}_n, \{D_n\}_n).$$

□

2.3 GLT Matrix Sequences

A matrix sequence $\{A_n\}_n$ may have several different singular value symbols, even on the same domain. For specific sequences, called *generalized locally Toeplitz* (GLT) sequences, we can choose one of their symbols, and denote it as *GLT symbol* of the sequence

$$\{A_n\}_n \sim_{GLT} k(x, \theta).$$

where the chosen symbols have all the same domain $D = [0, 1] \times [-\pi, \pi]$. If we denote with $\mathscr{M}_D$ the set of measurable functions on D, and with $\mathscr{G}$ the set of GLT sequences, then the choice of the symbol can be seen as a map

$$S : \mathscr{G} \to \mathscr{M}_D.$$

Both $\mathscr{G}$ and $\mathscr{M}_D$ are $\mathbb{C}$ algebras and pseudometric spaces with the distances d_{acs} and d_m, inducing respectively the acs convergence and the convergence in measure. In [9] and in [2] several properties of the map S are proved.

Theorem 3

1. *S is a homomorphism of $\mathbb{C}$ algebras. Given $\{A_n\}_n, \{B_n\}_n \in \mathscr{G}$, and $c \in \mathbb{C}$, we have that*

$$S(\{A_n + B_n\}_n) = S(\{A_n\}_n) + S(\{B_n\}_n),$$

$$S(\{A_n B_n\}_n) = S(\{A_n\}_n) \cdot S(\{B_n\}_n),$$

$$S(\{cA_n\}_n) = cS(\{A_n\}_n).$$

2. *The kernel of S are exactly the zero-distributed sequences.*
3. *S preserves the distances. Given $\{A_n\}_n, \{B_n\}_n \in \mathscr{G}$ we have*

$$d_{acs}\left(\{A_n\}_n, \{B_n\}_n\right) = d_m(S(\{A_n\}_n), S(\{B_n\}_n)).$$

4. *S is onto. All measurable functions are GLT symbols.*

5. *GLT symbols are singular value symbols:*

$$\{A_n\}_n \in \mathscr{G} \implies \{A_n\}_n \sim_\sigma S(\{A_n\}_n)$$

6. *The graph of S is closed in $\mathscr{G} \times \mathscr{M}_D$. If $\{B_{n,m}\}_{n,m}$ are sequences in $\mathscr{G}$ that converge acs to $\{A_n\}_n$, and their symbols converge in measure to $k(x,\theta)$, then $S(\{A_n\}_n) = k(x,\theta)$.*

The diagonal sampling sequences are denoted as $\{D_n(a)\}_n$, where $a : [0,1] \to \mathbb{C}$ is a measurable function, and

$$D_n(a) = \underset{i=1,\dots,n}{\operatorname{diag}} a\left(\frac{i}{n}\right) = \begin{pmatrix} a\left(\frac{1}{n}\right) & & & \\ & a\left(\frac{2}{n}\right) & & \\ & & \ddots & \\ & & & a(1) \end{pmatrix}$$

It is easy to verify that when $a : [0,1] \to \mathbb{C}$ is an almost everywhere (a.e.) continuous function, we have $\{D_n(a)\}_n \sim_{\sigma,\lambda} a(x)$. Furthermore, if $a(x)$ is continuous, we know that these sequences have as GLT symbol

$$\{D_n(a)\}_n \sim_{GLT} a(x) \otimes 1,$$

where $a \otimes 1 : [0,1] \times [-\pi,\pi] \to \mathbb{C}$ is a function constant in the second variable. This is not true for every $a(x)$ measurable, so we resort to the following result.

Lemma 5 *Given any $a : [0,1] \to \mathbb{C}$ measurable function, and $a_m \in C([0,1])$ continuous functions that converge in measure to $a(x)$, there exists an increasing and unbounded map $m(n)$ such that*

$$\{D_n(a_{m(n)})\}_n \sim_{GLT} a(x) \otimes 1 \qquad \{D_n(a_{m(n)})\}_n \sim_\lambda a(x)$$

Proof Easy corollary of Lemma 3.4 and Theorem 3.1 in [3]. □

3 Spectral Measures

3.1 Radon Measures

Let $\{A_n\}_n \in \mathscr{E}$ be a sequence with a spectral symbol $k(x)$ with domain D. By definition, we have

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} G(\lambda_i(A_n)) = \frac{1}{l(D)} \int_D G(k(x))dx.$$

Let $\phi : C_c(\mathbb{C}) \to \mathbb{C}$ be the functional defined as

$$\phi(G) := \frac{1}{l(D)} \int_D G(k(x))dx.$$

The latter is a continuous and linear map, and if we restrict it to real valued compacted supported functions, it is also a positive operator, since

$$G(x) \geq 0 \quad \forall x \in \mathbb{C} \implies \phi(G) = \frac{1}{l(D)} \int_D G(k(x))dx \geq 0.$$

Let us now recall Riesz theorem [1].

Theorem 4 (Riesz) *Let $\phi : C_c(X) \to \mathbb{R}$ be a positive linear and continuous function, where X is a Hausdorff and locally compact space. There exists a uniquely determined Radon positive measure μ such that*

$$\phi(F) = \int_X F d\mu \qquad \forall F \in C_c(X).$$

If $G \in C_c(\mathbb{C})$ is a complex valued map, we can always decompose it into $G = G_1 + iG_2$ where G_1 and G_2 are real valued and supported on a compact. Since ϕ is linear, we get

$$\phi(G) = \phi(G_1) + i\phi(G_2) = \int_{\mathbb{C}} G_1 d\mu + i \int_{\mathbb{C}} G_2 d\mu = \int_{\mathbb{C}} G d\mu$$

so ϕ induces a unique measure μ. We can thus define a *spectral measure*.

Definition 4 Given $\{A_n\}_n \in \mathscr{E}$, we say that it has a spectral measure μ if

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} G(\lambda_i(A_n)) = \int_{\mathbb{C}} G d\mu$$

for every $G \in C_c(\mathbb{C})$.

Let $G_m \in C_c(\mathbb{C})$ be a sequence of nonnegative real valued maps such that $\|G_m\|_\infty \leq 1$ and

$$G_m(x) = 1 \qquad \forall\ |x| \leq m.$$

We find that

$$\int_{\mathbb{C}} G_m d\mu = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} G_m(\lambda_i(A_n)) \leq 1$$

and hence

$$\mu(\mathbb{C}) = \lim_{m\to\infty} \mu\left(\{x : |x| \le m\}\right) \le \limsup_{m\to\infty} \int_{\mathbb{C}} G_m d\mu \le 1.$$

This proves that all the measures we consider are finite. Since all the finite measures over the Borelian set are Radon, we will now simply say "measure" instead of "Radon measure." We showed that any measurable function induces a finite measure, but we can actually prove that it induces a probability measure, and also that any probability measure is induced by a function.

Lemma 6 *Let $D \subseteq \mathbb{R}^n$ be a measurable set with finite nonzero measure. Then, for any $k \in \mathscr{M}_D$ there exists a probability measure μ such that*

$$\frac{1}{l(D)} \int_D G(k(x))dx = \int_{\mathbb{C}} G d\mu \qquad \forall\, G \in C_c(\mathbb{C}).$$

Let J be the real interval $[0, 1]$. *Then for every probability measure μ there exists a measurable function $k \in \mathscr{M}_J$ such that*

$$\int_0^1 G(k(x))dx = \int_{\mathbb{C}} G d\mu \qquad \forall\, G \in C_c(\mathbb{C}).$$

Proof Given $k \in \mathscr{M}_D$, we already showed that Riesz theorem identifies a unique finite measure μ such that

$$\frac{1}{l(D)} \int_D G(k(x))dx = \int_{\mathbb{C}} G d\mu \qquad \forall\, G \in C_c(\mathbb{C}).$$

Let us consider $M > 0$ and denote

$$\chi_M(x) = \begin{cases} 1 & |x| \le M, \\ 0 & |x| > M. \end{cases}$$

Moreover, let us fix $\varepsilon > 0$, so that for every $M > 0$ we can find $G_M \in C_c(\mathbb{C})$ such that

$$\chi_M(x) \le G_M(x) \le \chi_{M+\varepsilon}(x) \qquad \forall x \in \mathbb{C}.$$

We infer

$$\int_{\mathbb{C}} \chi_{M-\varepsilon} d\mu \le \int_{\mathbb{C}} G_{M-\varepsilon} d\mu = \frac{1}{l(D)} \int_D G_{M-\varepsilon}(k(x))dx \le \frac{1}{l(D)} \int_D \chi_M(k(x))dx,$$

$$\frac{1}{l(D)} \int_D \chi_M(k(x))dx \le \frac{1}{l(D)} \int_D G_M(k(x))dx = \int_{\mathbb{C}} G_M d\mu \le \int_{\mathbb{C}} \chi_{M+\varepsilon} d\mu$$

so that

$$\int_{\mathbb{C}} \chi_{M-\varepsilon} d\mu \leq \frac{1}{l(D)} \int_D \chi_M(k(x)) dx \leq \int_{\mathbb{C}} \chi_{M+\varepsilon} d\mu.$$

When we let ε go to zero, we obtain that the integrals coincide on the indicator functions of closed intervals

$$\int_{\mathbb{C}} \chi_M d\mu = \frac{1}{l(D)} \int_D \chi_M(k(x)) dx.$$

The symbol $k(x)$ is a measurable function, so it is *sparsely unbounded*, meaning that

$$\lim_{M\to\infty} l(\{x : |k(x)| > M\}) = \lim_{M\to\infty} \int_D \chi_{|x|>M}(k(x)) dx = 0.$$

With the latter, we can conclude that μ is a probability measure

$$\mu(\mathbb{C}) = \lim_{M\to+\infty} \int_{\mathbb{C}} \chi_{|x|\leq M} d\mu = \lim_{M\to\infty} \frac{1}{l(D)} \int_D \chi_{|x|\leq M}(k(x)) dx = 1.$$

Given any probability measure μ, we know that the space $(\mathbb{C}, \mu)$ is a *standard probability space*, meaning that it is isomorphic to a space $X = I \sqcup E$, where I is a real finite interval with the Lebesgue measure, and $E = \{x_1, x_2, \dots\}$ is a discrete numerable set with an atomic measure ν. In particular, the isomorphism $\varphi : \mathbb{C} \to X$ satisfies

$$\mu(U) = l \oplus \nu(\varphi(U)) \qquad \forall\, U \in \mathscr{B}(\mathbb{C}).$$

and if the atomic measure is $\nu = \sum_{i=1}^{+\infty} c_i \delta_{x_i}$, then

$$1 = \mu(\mathbb{C}) = l \oplus \nu(X) = l(I) + \sum_{i=1}^{+\infty} c_i.$$

If we call $S = \nu(X) = \sum_{i=1}^{+\infty} c_i$, then we can take $I = [S, 1]$. Let $g : [0, 1] \to X$ be a map defined as

$$g(x) := \begin{cases} x_k & \sum_{i=1}^{k-1} c_i \leq x < \sum_{i=1}^{k} c_i, \\ x & x \geq S. \end{cases}$$

This has the same distribution as $l \oplus \nu$, since for every measurable map $G : X \to \mathbb{C}$ we obtain

$$\int_X G d(l \oplus \nu) = \sum_{i=1}^{+\infty} c_i G(x_i) + \int_S^1 G(x)dx = \int_0^1 G(g(x))dx.$$

Let now $k := \varphi^{-1} \circ g : [0, 1] \to \mathbb{C}$ be a measurable function, and $G \in C_c(\mathbb{C})$. We conclude that

$$\int_{\mathbb{C}} G d\mu = \int_X G \circ \varphi^{-1} d(l \oplus \nu) = \int_0^1 G(\varphi^{-1}(g(x)))dx = \int_0^1 G(k(x))dx.$$

□

A corollary of the latter result is that any sequence with a spectral symbol admits a probability spectral measure, and also the opposite holds. Moreover, if we call $\mathbb{P}$ the set of probability measures on $\mathbb{C}$, then we can also prove that any measure $\mu \in \mathbb{P}$ is a spectral measure.

Corollary 1 *All measures in $\mathbb{P}$ are spectral measures.*

Proof Let J be the real interval $[0, 1]$. Given any $k \in \mathscr{M}_J$, then there exists a sequence of continuous functions $k_m \in \mathscr{M}_J$ converging to k in measure. Using Lemma 5, we find that k is a spectral symbol, so every function in $\mathscr{M}_J$ is a spectral symbol.

Given now a measure $\mu \in \mathbb{P}$, Lemma 6 shows that it is induced by a measurable function in $\mathscr{M}_J$, so μ is also a spectral symbol. This implies that every measure in $\mathbb{P}$ is a spectral measure. □

3.2 Vague Convergence

We notice that every matrix A_n can be associated with an atomic probability measure μ_{A_n} with support on its eigenvalues

$$\mu_{A_n} := \frac{1}{n} \sum_{i=1}^{n} \delta_{\lambda_i(A_n)}.$$

Let us return again to the definition of spectral measure and notice that it can be rewritten as

$$\lim_{n\to\infty} \int_{\mathbb{C}} G d\mu_{A_n} = \int_{\mathbb{C}} G d\mu \qquad \forall G \in C_c(\mathbb{C}).$$

This is actually the definition of *vague convergence* for measures.

The space $\mathbb{P}$ endowed with the vague convergence is a complete metric space, using the *Lévy–Prokhorov metric* [10]

$$\pi(\mu, \nu) = \inf\{\, \varepsilon > 0 \mid \mu(A) \leq \nu(A^\varepsilon) + \varepsilon,\ \ \nu(A) \leq \mu(A^\varepsilon) + \varepsilon\ \forall A \in \mathscr{B}(\mathbb{C})\,\}$$

where

$$A^\varepsilon := \{\, x \in \mathbb{C} \mid dist(x, A) < \varepsilon\,\} = \{\, x + y \mid x \in A, |y| < \varepsilon\,\}.$$

Since every matrix is associated with an atomic probability measure, we can extend the definition of π to matrices and sequences.

Definition 5 Let $A, B \in \mathbb{C}^{n\times n}$ and let μ_A, μ_B be the probability atomic measures associated with their spectra, defined as

$$\mu_A := \frac{1}{n}\sum_{i=1}^{n} \delta_{\lambda_i(A)}, \qquad \mu_B := \frac{1}{n}\sum_{i=1}^{n} \delta_{\lambda_i(B)}.$$

The Lévy–Prokhorov metric on $\mathbb{C}^{n\times n}$ is defined as

$$\pi(A, B) := \pi(\mu_A, \mu_B).$$

The Lévy–Prokhorov metric on $\mathscr{E}$ is defined as

$$\pi(\{A_n\}_n, \{B_n\}_n) := \limsup_{n\to\infty} \pi(\mu_{A_n}, \mu_{B_n}).$$

Again, we need to prove that the latter is actually a pseudometric.

Lemma 7 *The Lévy–Prokhorov metric is a pseudometric on $\mathbb{C}^{n\times n}$ and a complete pseudometric on $\mathscr{E}$.*

Proof The Lévy–Prokhorov metric is an actual metric on the space of probability measures, so all the properties can be transferred to the space of matrices $\mathbb{C}^{n\times n}$, except for the identity of matrices with zero distance, since two different matrices may have the same eigenvalues. Thus it is a pseudometric on $\mathbb{C}^{n\times n}$, and by Lemma 2, it is a complete pseudometric on $\mathscr{E}$. □

Since every matrix is associated with an atomic probability measure, we can also use the same notation for mixed elements, like

$$\pi(A, \nu) := \pi(\mu_A, \nu).$$

The considered notation is useful since the definition of spectral measure is given by

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}G(\lambda_i(A_n))=\int_{\mathbb{C}}G d\mu \qquad \forall G\in C_c(\mathbb{C})$$

and, when $\mu\in\mathbb{P}$, it can be rewritten as

$$\{A_n\}_n\sim_\lambda \mu \iff \pi(A_n,\mu)\xrightarrow{n\to+\infty}0.$$

The distance π on $\mathscr{E}$ is consistent with the distance between their spectral probability measures, as shown in the following result.

Lemma 8 *If $\{A_n\}_n\sim_\lambda\mu$ and $\{B_n\}_n\sim_\lambda\nu$, with $\{A_n\}_n,\{B_n\}_n\in\mathscr{E}$ and $\mu,\nu\in\mathbb{P}$, then*

$$\pi(\{A_n\}_n,\{B_n\}_n)=\pi(\mu,\nu)=\lim_{n\to\infty}\pi(A_n,B_n).$$

Proof Using the triangular property, we infer

$$\pi(\mu,\nu)\le\pi(\mu,A_n)+\pi(A_n,B_n)+\pi(B_n,\nu),$$

$$\pi(\mu,\nu)\ge-\pi(\mu,A_n)+\pi(A_n,B_n)-\pi(B_n,\nu).$$

Thus we obtain

$$\pi(\mu,\nu)\le\liminf_{n\to\infty}\pi(\mu,A_n)+\pi(A_n,B_n)+\pi(B_n,\nu)=\liminf_{n\to\infty}\pi(A_n,B_n),$$

$$\pi(\mu,\nu)\ge\limsup_{n\to\infty}-\pi(\mu,A_n)+\pi(A_n,B_n)-\pi(B_n,\nu)=\limsup_{n\to\infty}\pi(A_n,B_n).$$

By exploiting the latter relationships we conclude that

$$\begin{aligned}\pi(\{A_n\}_n,\{B_n\}_n)&=\limsup_{n\to\infty}\pi(A_n,B_n)\\&\le\pi(\mu,\nu)\le\\&\liminf_{n\to\infty}\pi(A_n,B_n)\le\pi(\{A_n\}_n,\{B_n\}_n).\end{aligned}$$

□

It is noteworthy to stress the importance of the probability condition on the measures. In fact, it is possible to find a sequence that admits a spectral measure but does not admit a spectral symbol, when the spectral measure is not a probability measure. Moreover, the Lévy–Prokhorov metric is defined only on probability

measures and if $\mu_n \in \mathbb{P}$ vaguely converge to a measure not in $\mathbb{P}$, then the sequence μ_n is not even a Cauchy sequence for π.

4 Main Results

4.1 Connection Between Measures

First of all, we prove that π and d' are equivalent pseudometrics on $\mathscr{E}$.

Lemma 9 *If $\{A_n\}_n, \{B_n\}_n \in \mathscr{E}$, then*

$$\pi(\{A_n\}_n, \{B_n\}_n) \leq d'(\{A_n\}_n, \{B_n\}_n) \leq 2\pi(\{A_n\}_n, \{B_n\}_n).$$

Proof Let us first prove that for any $A, B \in \mathbb{C}^{n\times n}$, we have

$$\pi(A, B) \leq d'(A, B) \leq 2\pi(A, B).$$

Let $\Lambda(A)$ and $\Lambda(B)$ be ordered so that

$$i < j \implies |\lambda_i(A) - \lambda_i(B)| \geq |\lambda_j(A) - \lambda_j(B)|$$

and

$$s := d'(A, B) = \frac{k-1}{n} + |\lambda_k(A) - \lambda_k(B)|.$$

In particular, we deduce that

$$|\lambda_i(A) - \lambda_i(B)| \leq s \qquad \forall i \geq k$$

and consequently, for any subset $U \subseteq \mathbb{C}$, we obtain the inequality

$$\#\{\lambda_i(A) \in U,\ i \geq k\} \leq \#\{\lambda_i(B) \in U^s,\ i \geq k\}.$$

Denote with μ_A and μ_B the atomic probability measures associated with A, B. Let $U \in \mathscr{B}(\mathbb{C})$ be any Borelian set and denote the cardinality of the intersection with a n-uple v as

$$Q_U(v) := \#\{i : v_i \in U\}.$$

Formally, $Q_U(v)$ is the number of elements of v inside v, counted with multiplicity. We know that

$$\begin{aligned}\mu_A(U) &= \frac{Q_U(\Lambda(A))}{n}\\ &= \frac{Q_U(\{\lambda_i(A) : i \geq k\})}{n} + \frac{Q_U(\{\lambda_i(A) : i < k\})}{n}\\ &\leq \frac{Q_{U^s}(\{\lambda_i(B) : i \geq k\})}{n} + \frac{k-1}{n}\\ &\leq \frac{Q_{U^s}(\Lambda(B))}{n} + s = \mu_B(U^s) + s.\end{aligned}$$

We symmetrically obtain also the following relation:

$$\mu_B(U) \leq \mu_A(U^s) + s.$$

As a consequence

$$\pi(A, B) = \inf\{\, \varepsilon > 0 \mid \mu_A(U) \leq \mu_B(U^\varepsilon) + \varepsilon,\ \mu_B(U) \leq \mu_A(U^\varepsilon) + \varepsilon\ \forall U \in \mathscr{B}(\mathbb{C})\,\}$$

$$\implies \pi(A, B) \leq s = d'(A, B).$$

Denote now $r = \pi(A, B)$ and let T be any sub-uple of $\Lambda(A)$. If we see T as a set, then it is a finite subset of $\mathbb{C}$, so it is a Borelian set. Given any $\varepsilon > 0$ we know that

$$\mu_A(T) = \frac{Q_T(\Lambda(A))}{n} \leq \mu_B(T^{r+\varepsilon}) + r + \varepsilon = \frac{Q_{T^{r+\varepsilon}}(\Lambda(B))}{n} + r + \varepsilon$$

so we deduce that

$$\frac{Q_T(\Lambda(A))}{n} \leq \frac{Q_{T^r}(\Lambda(B))}{n} + r \implies Q_T(\Lambda(A)) \leq Q_{T^r}(\Lambda(B)) + rn.$$

By using the fact that the map Q is integer valued, we conclude that

$$Q_T(\Lambda(A)) \leq Q_{T^r}(\Lambda(B)) + \lfloor rn \rfloor.$$

The quantity $Q_T(\Lambda(A))$ is actually the cardinality of T seen as a sub-uple of $\Lambda(A)$, so for every subset T of k eigenvalues in A, even repeated, there are at least $k - \lfloor rn \rfloor$ eigenvalues of B that have distance less than r from one of the elements of T.

Let us now build a bipartite graph, where the left set of nodes L contains the elements of $\Lambda(A)$, the right set of nodes R contains the elements of $\Lambda(B)$, and $\lfloor rn \rfloor$ additional nodes. Every additional node is connected to all the elements of L, and an element of $\Lambda(A)$ is connected to an element of $\Lambda(B)$ if and only if their distance

is less than r. If we denote E the set of edges of the graph and N the set of its nodes, then we can define the neighborhood of a subset of nodes $P \subseteq N$ as

$$N(P) := \#\{u \in N : \exists v \in P, (v, u) \in E\}.$$

By using the previous derivations, we know that for any $T \subseteq L = \Lambda(A)$ it holds

$$N(T) \geq \#T - \lfloor rn \rfloor + \lfloor rn \rfloor = \#T.$$

Thanks to Hall's marriage theorem that can be found, for example, in [7], there exists a matching for L, meaning that there exists an injective map $\alpha : L \to R$ such that

$$(u, \alpha(u)) \in E \qquad \forall\, u \in L.$$

Now let us consider the set

$$P := \{u \in L : \alpha(u) \in \Lambda(B)\}.$$

we know that $\#P \geq n - \lfloor rn \rfloor$, and we can enumerate the eigenvalues in $\Lambda(A) = L$ and $\Lambda(B)$ so that

$$\lambda_i(A) \in P, \quad \lambda_i(B) = \alpha(\lambda_i(A)) \qquad \forall\, i \leq n - \lfloor rn \rfloor.$$

Since u and $\alpha(u)$ are connected for all $u \in L$, we deduce that $\lambda_i(A)$ and $\lambda_i(B)$ are connected for at least $n - \lfloor rn \rfloor$ indices. By construction,

$$|\lambda_i(B) - \lambda_i(A)| < r \qquad \forall\, i \leq n - \lfloor rn \rfloor$$

so

$$\begin{aligned} d'(A, B) &= \min_{\sigma \in S_n} \min_{i=1,\ldots,n+1} \left\{ \frac{i-1}{n} + |\Lambda(A) - \Lambda(B)_\sigma|_i^\downarrow \right\} \\ &\leq \min_{i=1,\ldots,n+1} \left\{ \frac{i-1}{n} + |\Lambda(A) - \Lambda(B)|_i^\downarrow \right\} \\ &< \frac{\lfloor rn \rfloor}{n} + r \leq 2r = 2\pi(A, B). \end{aligned}$$

This proves that for any $A, B \in fC^{n \times n}$ we have

$$\pi(A, B) \leq d'(A, B) \leq 2\pi(A, B).$$

Given now $\{A_n\}_n, \{B_n\}_n \in \mathscr{E}$, we conclude

$$\pi(\{A_n\}_n, \{B_n\}_n) = \limsup_{n\to\infty} \pi(A_n, B_n) \leq \limsup_{n\to\infty} d'(A_n, B_n) = d'(\{A_n\}_n, \{B_n\}_n),$$

$$d'(\{A_n\}_n, \{B_n\}_n) = \limsup_{n\to\infty} d'(A_n, B_n) \leq \limsup_{n\to\infty} 2\pi(A_n, B_n) = 2\pi(\{A_n\}_n, \{B_n\}_n).$$

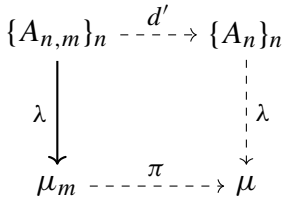
□

The two distances d' and π are equivalent, so they induce the same topology on the space $\mathscr{E}$ and they respect a property of closeness given by the following lemma.

Lemma 10 *Let $\{A_{n,m}\}_n \sim_\lambda \mu_m$, where $\{A_{n,m}\}_n \in \mathscr{E}$ and $\mu_m \in \mathbb{P}$ for every m. If we consider the statements below*

1. $\pi(\mu_m, \mu) \xrightarrow{m\to\infty} 0$,
2. $\{A_n\}_n \sim_\lambda \mu$,
3. $d'(\{A_{n,m}\}_n, \{A_n\}_n) \xrightarrow{m\to\infty} 0$,

where $\{A_n\}_n \in \mathscr{E}$ and $\mu \in \mathbb{P}$, then any two of them are true if and only if all of them are true.



Proof 1.3. $\implies$ 2.) We know that

$$\pi(A_n, \mu) \leq \pi(A_n, A_{n,m}) + \pi(A_{n,m}, \mu_m) + \pi(\mu_m, \mu) \qquad \forall n, m.$$

Given $\varepsilon > 0$, we can find M such that

$$\pi(\mu_m, \mu) \xrightarrow{m\to\infty} 0 \implies \pi(\mu_m, \mu) < \varepsilon \quad \forall m > M,$$

$$d'(\{A_{n,m}\}_n, \{A_n\}_n) \xrightarrow{m\to\infty} 0 \implies d'(\{A_{n,m}\}_n, \{A_n\}_n) < \varepsilon \quad \forall m > M.$$

Using Lemma 9, we obtain

$$\limsup_{n\to\infty} \pi(A_{n,m}, A_n) = \pi(\{A_{n,m}\}_n, \{A_n\}_n) \leq d'(\{A_{n,m}\}_n, \{A_n\}_n).$$

We can then fix $m > M$ and find $N > 0$ such that

$$\pi(A_{n,m}, A_n) \leq 2\varepsilon, \quad \pi(A_{n,m}, \mu_m) \leq \varepsilon \qquad \forall n > N.$$

We obtain that

$$\pi(A_n, \mu) \leq 2\varepsilon + \varepsilon + \varepsilon = 4\varepsilon \qquad \forall n > N,$$

and hence we conclude that

$$\pi(A_n, \mu) \xrightarrow{n\to\infty} 0 \implies \{A_n\}_n \sim_\lambda \mu.$$

2.3. $\implies$ 1.) Thanks to Lemma 8, we know that

$$\{A_{n,m}\}_n \sim_\lambda \mu_m, \quad \{A_n\}_n \sim_\lambda \mu \implies \pi(\mu_m, \mu) = \pi(\{A_{n,m}\}_n, \{A_n\}_n)$$

and, using Lemma 9, we conclude that

$$\pi(\mu_m, \mu) = \pi(\{A_{n,m}\}_n, \{A_n\}_n) \le d'(\{A_{n,m}\}_n, \{A_n\}_n) \xrightarrow{m\to\infty} 0.$$

1.2. $\implies$ 3.) Thanks to Lemma 8, we know that

$$\{A_{n,m}\}_n \sim_\lambda \mu_m, \quad \{A_n\}_n \sim_\lambda \mu \implies \pi(\{A_{n,m}\}_n, \{A_n\}_n) = \pi(\mu_m, \mu)$$

and, using Lemma 9, we conclude that

$$d'(\{A_{n,m}\}_n, \{A_n\}_n) \le 2\pi(\{A_{n,m}\}_n, \{A_n\}_n) = 2\pi(\mu_m, \mu) \xrightarrow{m\to\infty} 0.$$

□

4.2 *Proofs of Theorems*

We can finally prove that d' identifies two sequences if and only if they have the same spectral symbol.

Theorem 1 *If* $\{A_n\}_n \sim_\lambda f(x)$, *then*

$$\{B_n\}_n \sim_\lambda f(x) \iff d'(\{A_n\}_n, \{B_n\}_n) = 0.$$

Proof Let μ be the probability measure associated with $f(x)$. Let also $\{A_{n,m}\}_n$ and μ_m be constant sequences defined as

$$A_{n,m} := A_n \qquad \forall n, m, \qquad\qquad \mu_m := \mu \qquad \forall m.$$

We know by hypothesis that

$$\{A_{n,m}\}_n \sim_\lambda \mu_m, \qquad \pi(\mu_m, \mu) \xrightarrow{m\to\infty} 0,$$

therefore, owing to Lemma 10, we obtain the equivalence

$$\{B_n\}_n \sim_\lambda \mu \iff d'(\{A_{n,m}\}_n, \{B_n\}_n) \xrightarrow{m\to\infty} 0,$$

which can be rewritten as

$$\{B_n\}_n \sim_\lambda f(x) \iff d'(\{A_n\}_n, \{B_n\}_n) = 0.$$

□

The other theorem shows that the GLT symbol represents in fact an ordering of the sequence eigenvalues. Given a sequence $\{A_n\}_n \in \mathscr{E}$ with a spectral symbol $k(x)$, we can consider the diagonal matrices $D_n \in \mathbb{C}^{n\times n}$ containing the eigenvalues of A_n. We get again that $\{D_n\}_n \sim_\lambda k(x)$, so we can focus only on diagonal sequences. A permutation of the eigenvalues is thus formalized as the similarity $P_n D_n P_n^T$ with P_n permutation matrices. In [3], we showed that a function $k(x)\otimes 1$ is a GLT symbol for a diagonal sequence $\{D_n\}_n$ if and only if the piecewise linear functions interpolating the ordered entries of D_n on $[0, 1]$ converge in measure to $k(x)$. Thanks to the existence of the natural order on $\mathbb{R}$, we deduced that for any real diagonal sequence $\{D_n\}_n$, with a real spectral symbol $k(x)$, there exists a sequence of permutations $\{P_n\}_n$ such that

$$\{P_n D_n P_n^T\}_n \sim_{GLT} k(x) \otimes 1.$$

We could not extend the result on the complex plane, due to the lack of a natural ordering. Using the spectral measure theory we developed, we can now bypass the problem, since spectral symbols with the same distribution are now identified into a uniquely determined probability measure.

Theorem 2 *Given a measurable function $k : [0, 1] \to \mathbb{C}$, and a diagonal sequence $\{D_n\}_n$ with spectral symbol $k(x)$, there exists a sequence $\{P_n\}_n$ of permutation matrices such that*

$$\{P_n D_n P_n^T\} \sim_{GLT} k(x) \otimes 1.$$

Proof The space of continuous functions is dense in the space of measurable functions with the convergence in measure. Thus, there exist $k_m(x) \in C[0, 1]$ that converge in measure to $k(x)$. Using Lemma 5, we can find a diagonal sequence $\{D'_n\}_n$ with

$$\{D'_n\}_n \sim_{GLT} k(x) \otimes 1, \qquad \{D'_n\}_n \sim_\lambda k(x).$$

Theorem 1 leads to $d'(\{D_n\}_n, \{D'_n\}_n) = 0$ and owing to Lemma 4, there exist permutation matrices $\{P_n\}_n$ such that

$$d_{acs}\left(\{D'_n\}_n, \{P_n D_n P_n^T\}_n\right) = 0.$$

Using the fact that the GLT space is closed for the pseudometric d_{acs}, and that the distance of the GLT symbols is equal to the distance of the sequences for Theorem 3, we conclude that $\{P_n D_n P_n^T\}_n \sim_{GLT} k(x) \otimes 1$. □

5 Future Works

The theory of spectral measures is still a work in progress, with open questions and many possible extensions.

For example, we have seen that the space of probability measures corresponds to the space of sequences which admit a spectral symbol, but the sequences admitting a general spectral measure (not necessarily a probability measure) are larger. The difference between 1 and the mass of a spectral measure can be interpreted as the rate of eigenvalues not converging to finite values, and consequentially we can admit spectral symbols $f : [0, 1] \to \mathbb{C}^*$, where $\mathbb{C}^* = \mathbb{C} \cup \{\infty\}$ is the Riemann sphere or the Alexandroff compactification of $\mathbb{C}$. The insight on the sequences of matrices is that they may have a fraction of the asymptotic spectrum that diverges to ∞ in modulus, so a spectral symbol with values on $\mathbb{C}^*$ may also catch this new behavior. The introduction of these new functions probably leads to a variation of Corollary 1, where a sequence admits a spectral measure if and only if it admits a spectral symbol with values on $\mathbb{C}^*$. The downside of this extension is that the distance π does not induce the vague convergence on the space of finite measures, so we need to find a new metric that mimics the characteristics of the Lèvy–Prokhorov metric.

All this document is focused on spectral symbols/measures, but the same analysis can be performed using the singular values instead of the eigenvalues, leading to a theory focused into singular value symbols/measures, that will probably have some deep bounds with the GLT symbols. They are similar since both the GLT symbol and the singular value symbol of a sequence are unique, but at the same time they are also very different since the space of measures lacks a group structure, and two sequences with different GLT symbols may have the same singular value symbol.

Eventually, it seems that spectral measures arise naturally even in algebraic geometry (see, for example, [13]) so further connections can be also developed in different areas of mathematics.

References

1. Ambrosio, L., Da Prato, G., Mennucci, A.: Introduction to Measure Theory and Integration. Edizioni della Normale, Pisa (2011)
2. Barbarino, G.: Equivalence between GLT sequences and measurable functions. Linear Algebra Appl. **529**, 397–412 (2017)
3. Barbarino, G.: Diagonal Matrix Sequences and Their Spectral Symbols. http://arxiv.org/abs/1710.00810 (2017)
4. Barbarino, G., Garoni, C.: From convergence in measure to convergence of matrix-sequences through concave functions and singular values. Electron. J. Linear Algebra **32**, 500–513 (2017)
5. Bhatia, R.: Matrix Analysis. Springer, New York (1997)
6. Böttcher, A., Garoni, C., Serra-Capizzano, S.: Exploration of Toeplitz-like matrices with unbounded symbols: not a purely academic journey. Sb. Math. **208**(11), 29–55 (2017)
7. Brualdi, R.A.: Introductory Combinatorics. Pearson/Prentice Hall, Upper Saddle River (2010)
8. Garoni, C.: Topological foundations of an asymptotic approximation theory for sequences of matrices with increasing size. Linear Algebra Appl. **513**, 324–341 (2017)

9. Garoni, C., Serra-Capizzano, S.: Generalized Locally Toeplitz Sequences: Theory and Applications, vol I. Springer, Cham (2017)
10. Prokhorov, Y.V.: Convergence of random processes and limit theorems in probability theory. Theory Probab. Appl. **1**(2), 157–214 (1956)
11. Serra-Capizzano, S.: Distribution results on the algebra generated by Toeplitz sequences: a finite-dimensional approach. Linear Algebra Appl. **328**, 121–130 (2001)
12. Tilli, P.: Locally Toeplitz sequences: spectral properties and applications. Linear Algebra Appl. **97**, 91–120 (1998)
13. Tsfasman M.A., Vlăduţ S.G.: Asymptotic properties of zeta-functions. J. Math. Sci. **84**(5), 1445–1467 (1997)

Block Locally Toeplitz Sequences: Construction and Properties

Carlo Garoni, Stefano Serra-Capizzano, and Debora Sesana

Abstract The theory of block locally Toeplitz (LT) sequences—along with its generalization known as the theory of block generalized locally Toeplitz (GLT) sequences—is a powerful apparatus for computing the spectral distribution of matrices arising from the discretization of differential problems. In this paper we develop the theory of block LT sequences, whereas the theory of block GLT sequences is the subject of the complementary paper (Chap. 3 of this book).

Keywords Singular values and eigenvalues · Block locally Toeplitz sequences · Block Toeplitz matrices · Discretization of differential equations

1 Introduction

A Toeplitz matrix is a matrix whose entries are constant along each northwest–southeast diagonal. A block Toeplitz matrix is a Toeplitz matrix whose entries are blocks, i.e., square matrices of a fixed size s. Any matrix-valued function $f : [-\pi, \pi] \to \mathbb{C}^{s\times s}$ whose components f_{ij} belong to $L^1([-\pi, \pi])$ generates a

C. Garoni
University of Italian Switzerland, Institute of Computational Science, Lugano, Switzerland

University of Insubria, Department of Science and High Technology, Como, Italy
e-mail: carlo.garoni@usi.ch; carlo.garoni@uninsubria.it

S. Serra-Capizzano
University of Insubria, Department of Science and High Technology, Como, Italy

Uppsala University, Department of Information Technology, Division of Scientific Computing, Uppsala, Sweden
e-mail: stefano.serrac@uninsubria.it; stefano.serra@it.uu.se

D. Sesana (✉)
University of Insubria, Department of Science and High Technology, Como, Italy
e-mail: debora.sesana@uninsubria.it

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_2

sequence of block Toeplitz matrices $\{T_n(f)\}_n$ via its Fourier coefficients

$$f_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\theta)\mathrm{e}^{-\mathrm{i}k\theta}\mathrm{d}\theta \in \mathbb{C}^{s\times s}, \quad k \in \mathbb{Z},$$

where the integrals are computed componentwise. Specifically, we have

$$T_n(f) = [f_{i-j}]_{i,j=1}^{n} \in \mathbb{C}^{sn\times sn},$$

i.e., the entries of $T_n(f)$ along the kth northwest–southeast diagonal identified by the equation $i - j = k$ are equal to the kth Fourier coefficient f_k. For example, if

$$f(\theta) = \begin{bmatrix} 2-2\cos\theta & -\mathrm{i}\sin\theta \\ -\mathrm{i}\sin\theta & 1 \end{bmatrix}$$

then we have

$$f_0 = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \qquad f_1 = \begin{bmatrix} -1 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix}, \qquad f_{-1} = \begin{bmatrix} -1 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix},$$

$f_k = 0$ for $|k| > 1$, and

$$T_n(f) = \begin{bmatrix} f_0 & f_{-1} & & & \\ f_1 & f_0 & f_{-1} & & \\ & \ddots & \ddots & \ddots & \\ & & f_1 & f_0 & f_{-1} \\ & & & f_1 & f_0 \end{bmatrix} = \operatorname*{tridiag}_{j=1,\dots,n} \left[\begin{array}{cc|cc|cc} -1 & -\frac{1}{2} & 2 & 0 & -1 & \frac{1}{2} \\ -\frac{1}{2} & 0 & 0 & 1 & \frac{1}{2} & 0 \end{array}\right]. \tag{1}$$

A 'block locally Toeplitz matrix' or 'locally block Toeplitz matrix' is a matrix possessing a local block Toeplitz structure. For instance, denoting by $X \circ Y$ the componentwise (Hadamard) product of the matrices X and Y, a block locally Toeplitz version of (1) is

$$A_n = \begin{bmatrix} a(x_1)\circ f_0 & a(x_1)\circ f_{-1} & & & & \\ a(x_2)\circ f_1 & a(x_2)\circ f_0 & a(x_2)\circ f_{-1} & & & \\ & \ddots & \ddots & \ddots & & \\ & & a(x_{n-1})\circ f_1 & a(x_{n-1})\circ f_0 & a(x_{n-1})\circ f_{-1} & \\ & & & a(x_n)\circ f_1 & a(x_n)\circ f_0 & \end{bmatrix}$$

$$= \operatorname*{tridiag}_{j=1,\dots,n} \left[\begin{array}{cc|cc|cc} -a_{11}(x_j) & -\frac{1}{2}a_{12}(x_j) & 2a_{11}(x_j) & 0 & -a_{11}(x_j) & \frac{1}{2}a_{12}(x_j) \\ -\frac{1}{2}a_{21}(x_j) & 0 & 0 & a_{22}(x_j) & \frac{1}{2}a_{21}(x_j) & 0 \end{array}\right], \tag{2}$$

where $x_j = \frac{j}{n+1}$ for $j = 1, \ldots, n$ and $a : [0,1] \to \mathbb{C}^{2\times 2}$ is continuous on $[0,1]$, in the sense that its components a_{ij} are continuous on $[0,1]$ for all $i, j = 1, 2$. Looking at a relatively small submatrix of A_n (according to a 'local' perspective), one easily recognizes an approximate block Toeplitz structure weighted through the function $a(x)$. For instance, the principal submatrix of A_n corresponding to the first two block rows and columns, i.e.,

$$\begin{bmatrix} a(x_1) \circ f_0 & a(x_1) \circ f_{-1} \\ a(x_2) \circ f_1 & a(x_2) \circ f_0 \end{bmatrix}$$

is approximately equal to

$$\begin{bmatrix} a(x_1) \circ f_0 & a(x_1) \circ f_{-1} \\ a(x_1) \circ f_1 & a(x_1) \circ f_0 \end{bmatrix} = T_2(a(x_1) \circ f),$$

because $x_2 \approx x_1$ and a is continuous. Similarly, if $B_{\lfloor\sqrt{n}\rfloor}$ is a submatrix of A_n of size $\lfloor\sqrt{n}\rfloor$, obtained as the intersection of $\lfloor\sqrt{n}\rfloor$ consecutive block rows and columns of A_n, then $B_{\lfloor\sqrt{n}\rfloor} \approx T_{\lfloor\sqrt{n}\rfloor}(a(x_i) \circ f)$, where $a(x_i)$ is any of the evaluations of $a(x)$ appearing in $B_{\lfloor\sqrt{n}\rfloor}$. The latter assertion remains true if $\lfloor\sqrt{n}\rfloor$ is replaced by any other integer k_n such that $k_n = o(n)$. In conclusion, if we explore 'locally' the matrix A_n, using an ideal microscope and considering a large value of n, then we realize that the 'local' structure of A_n is approximately the block Toeplitz structure generated by $a(x_i) \circ f(\theta)$ for some x_i.

Sequences of block locally Toeplitz (LT) matrices (or block LT sequences for shortness) along with their generalizations (the so-called block generalized locally Toeplitz (GLT) sequences) naturally arise in the discretization of systems of differential equations (DEs) and also in the higher-order finite element approximation of scalar DEs. For example, up to a proper scaling and a possible permutation transformation, the block LT matrix A_n in (2) is the matrix resulting from the classical central second-order finite difference discretization of the following system of DEs:

$$\begin{cases} -a_{11}(x)u_1''(x) + a_{12}u_2'(x) = f_1(x), & x \in (0,1), \\ a_{21}(x)u_1'(x) + a_{22}(x)u_2(x) = f_2(x), & x \in (0,1), \\ u_1(0) = u_1(1) = 0, \\ u_2(0) = u_2(1) = 0. \end{cases}$$

The theory of block LT sequences—and especially its generalization, the theory of block GLT sequences—is a powerful apparatus for computing the asymptotic spectral distribution of such sequences as the matrix size goes to infinity. For instance, these theories allow one to show that the asymptotic spectral distribution

of the sequence $\{A_n\}_n$ is described by the matrix-valued function

$$a(x) \circ f(\theta) = \begin{bmatrix} a_{11}(x)(2-2\cos\theta) & -\mathrm{i}\, a_{12}(x)\sin\theta \\ -\mathrm{i}\, a_{21}(x)\sin\theta & a_{22}(x) \end{bmatrix},$$

in the sense of Definition 2.1 below (this result will be proved in the forthcoming paper [11], which is entirely devoted to applications of the theory of block GLT sequences).

The present paper develops the theory of block LT sequences, which is fundamental to the theory of block GLT sequences. The latter will be developed in the complementary paper [12], whereas for applications we refer the reader to [11]. The paper is organized as follows. In Sect. 2 we collect all the necessary preliminaries. Section 3 focuses on the fundamental notion of approximating classes of sequences. In Sect. 4 we develop the theory of block LT sequences. Section 5 is devoted to final remarks, including a discussion on the analogies and differences between the theory of (scalar) LT sequences [7, Chapter 7] and the theory of block LT sequences.

2 Mathematical Background

2.1 Notation and Terminology

- O_m and I_m denote, respectively, the $m \times m$ zero matrix and the $m \times m$ identity matrix. Sometimes, when the size m can be inferred from the context, O and I are used instead of O_m and I_m.
- $\mathbf{1}_m$ denotes the $m \times m$ matrix whose entries are all equal to 1.
- The eigenvalues and the singular values of $X \in \mathbb{C}^{m\times m}$ are denoted by $\lambda_j(X)$, $j = 1,\dots,m$, and $\sigma_j(X)$, $j = 1,\dots,m$, respectively. The maximum and minimum singular values of X are also denoted by $\sigma_{\max}(X)$ and $\sigma_{\min}(X)$, respectively. The spectrum of X is denoted by $\Lambda(X)$.
- Given $X \in \mathbb{C}^{m\times m}$ and $1 \le p \le \infty$, $\|X\|_p$ denotes the Schatten p-norm of X, which is defined as the p-norm of the vector $(\sigma_1(X),\dots,\sigma_m(X))$; see [3]. The Schatten 1-norm is also called the trace-norm. The Schatten ∞-norm $\|X\|_\infty = \sigma_{\max}(X)$ is the classical 2-norm (or spectral norm) and will also be denoted by $\|X\|$.
- $\Re(X)$ is the real part of the (square) matrix X, i.e., $\Re(X) = \frac{X+X^*}{2}$, where X^* is the conjugate transpose of X.
- $C_c(\mathbb{C})$ (resp., $C_c(\mathbb{R})$) is the space of complex-valued continuous functions defined on $\mathbb{C}$ (resp., $\mathbb{R}$) and with bounded support.
- χ_E is the characteristic (indicator) function of the set E.
- μ_k denotes the Lebesgue measure in $\mathbb{R}^k$. Throughout this paper, unless otherwise stated, all the terminology from measure theory (such as 'measurable set', 'measurable function', 'a.e.', etc.) is always referred to the Lebesgue measure.

- Let $D \subseteq \mathbb{R}^k$, let $r \geq 1$ and $1 \leq p \leq \infty$. A matrix-valued function $f : D \to \mathbb{C}^{r\times r}$ is said to be measurable (resp., continuous, bounded, in $L^p(D)$, in $C^\infty(D)$, etc.) if its components $f_{\alpha\beta} : D \to \mathbb{C}$, $\alpha, \beta = 1, \ldots, r$, are measurable (resp., continuous, bounded, in $L^p(D)$, in $C^\infty(D)$, etc.). The space of functions $f : D \to \mathbb{C}^{r\times r}$ belonging to $L^p(D)$ will be denoted by $L^p(D, r)$ in order to stress the dependence on r.
- Let $f_m, f : D \subseteq \mathbb{R}^k \to \mathbb{C}^{r\times r}$ be measurable. We say that f_m converges to f in measure (resp., a.e., in $L^p(D)$, etc.) if $(f_m)_{\alpha\beta}$ converges to $f_{\alpha\beta}$ in measure (resp., a.e., in $L^p(D)$, etc.) for all $\alpha, \beta = 1, \ldots, r$.
- A function $a : [0, 1] \to \mathbb{C}^{r\times r}$ is said to be Riemann-integrable if its components $a_{\alpha\beta} : [0, 1] \to \mathbb{C}$, $\alpha, \beta = 1, \ldots, r$, are Riemann-integrable. We point out that a complex-valued function g is Riemann-integrable when its real and imaginary parts $\Re(g)$ and $\Im(g)$ are Riemann-integrable in the classical sense. We also recall that any Riemann-integrable function is *bounded* by definition.
- We use a notation borrowed from probability theory to indicate sets. For example, if $f, g : D \subseteq \mathbb{R}^k \to \mathbb{C}^{r\times r}$, then $\{\sigma_{\max}(f) > 0\} = \{\mathbf{x} \in D : \sigma_{\max}(f(\mathbf{x})) > 0\}$, $\mu_k\{\|f - g\| \geq \epsilon\}$ is the measure of the set $\{\mathbf{x} \in D : \|f(\mathbf{x}) - g(\mathbf{x})\| \geq \epsilon\}$, etc.
- A function of the form $f(\theta) = \sum_{j=-d}^{d} f_j \, \mathrm{e}^{\mathrm{i}j\theta}$ with $f_{-d}, \ldots, f_d \in \mathbb{C}^{r\times r}$ is said to be a (matrix-valued) trigonometric polynomial. If $f_{-d} \neq O_r$ or $f_d \neq O_r$, the number d is referred to as the degree of f.
- A matrix-sequence is any sequence of the form $\{A_n\}_n$, where $A_n \in \mathbb{C}^{sn\times sn}$ and s is a *fixed* positive integer. The role of s will become clear later on. A matrix-sequence $\{A_n\}_n$ is said to be Hermitian if each A_n is Hermitian.

2.2 Preliminaries on Matrix Analysis

2.2.1 Matrix Norms

Let $X \in \mathbb{C}^{m\times m}$. Since $\|X\| = \|X\|_\infty = \sigma_{\max}(X)$ and $\mathrm{rank}(X)$ is the number of nonzero singular values of X, we have

$$\sigma_{\max}(X) = \|X\| \leq \|X\|_1 = \sum_{i=1}^{m} \sigma_i(X) \leq \mathrm{rank}(X)\|X\| \leq m\|X\|, \quad X \in \mathbb{C}^{m\times m}. \tag{3}$$

Another important trace-norm inequality is the following [7, p. 33]:

$$\|X\|_1 \leq \sum_{i,j=1}^{m} |x_{ij}|, \quad X \in \mathbb{C}^{m\times m}. \tag{4}$$

If $1 \leq p, q \leq \infty$ are conjugate exponents, i.e., $1/p + 1/q = 1$, then the following Hölder-type inequality holds for the Schatten norms [3]:

$$\|XY\|_1 \leq \|X\|_p \|Y\|_q, \quad X, Y \in \mathbb{C}^{m\times m}. \tag{5}$$

2.2.2 Direct Sums and Hadamard Products

If $X \in \mathbb{C}^{m_1\times m_2}$ and $Y \in \mathbb{C}^{\ell_1\times\ell_2}$, the direct sum of X and Y is the $(m_1+\ell_1)\times(m_2+\ell_2)$ matrix defined by

$$X \oplus Y = \mathrm{diag}(X, Y) = \begin{bmatrix} X & O \\ O & Y \end{bmatrix}.$$

We recall some properties of direct sums.

- The relation $(X_1 \oplus Y_1)(X_2 \oplus Y_2) = (X_1X_2) \oplus (Y_1Y_2)$ holds whenever X_1, X_2 can be multiplied and Y_1, Y_2 can be multiplied.
- If $X \in \mathbb{C}^{m\times m}$ and $Y \in \mathbb{C}^{\ell\times\ell}$, the eigenvalues and singular values of $X \oplus Y$ are given by $\{\lambda_i(X) : i = 1, \ldots, m\} \cup \{\lambda_j(Y) : j = 1, \ldots, \ell\}$ and $\{\sigma_i(X) : i = 1, \ldots, m\} \cup \{\sigma_j(Y) : j = 1, \ldots, \ell\}$, respectively.

In particular, for all $X \in \mathbb{C}^{m_1\times m_2}$ and $Y \in \mathbb{C}^{\ell_1\times\ell_2}$,

$$\begin{aligned} &\|X \oplus Y\| = \max(\|X\|, \|Y\|), \\ &\|X \oplus Y\|_p = \left(\|X\|_p^p + \|Y\|_p^p\right)^{1/p}, \quad 1 \leq p < \infty. \end{aligned} \tag{6}$$

If $X, Y \in \mathbb{C}^{m\times\ell}$, the Hadamard (or entrywise) product of X and Y is the $m \times \ell$ matrix defined by $(X \circ Y)_{ij} = x_{ij}y_{ij}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, \ell$. We recall the following property of Hadamard products [3, p. 23]:

$$\|X \circ Y\| \leq \|X\|\,\|Y\|, \quad X, Y \in \mathbb{C}^{m\times m}. \tag{7}$$

2.3 Preliminaries on Measure and Integration Theory

2.3.1 Measurability

The following lemma can be derived from the results in [3, Section VI.1]. It will be used essentially everywhere in this paper, either explicitly or implicitly.

Lemma 2.1 *Let $f : D \subseteq \mathbb{R}^k \to \mathbb{C}^{r\times r}$ be measurable and let $g : \mathbb{C}^r \to \mathbb{C}$ be continuous and symmetric in its r arguments, i.e., $g(\lambda_1, \ldots, \lambda_r) = g(\lambda_{\rho(1)}, \ldots, \lambda_{\rho(r)})$ for all permutations ρ of $\{1, \ldots, r\}$. Then, the function*

$\mathbf{x} \mapsto g(\lambda_1(f(\mathbf{x})), \ldots, \lambda_r(f(\mathbf{x})))$ *is well-defined* (*independently of the labeling of the eigenvalues of* $f(\mathbf{x})$) *and measurable. As a consequence:*

- *the function* $\mathbf{x} \mapsto g(\sigma_1(f(\mathbf{x})), \ldots, \sigma_r(f(\mathbf{x})))$ *is measurable;*
- *the functions* $\mathbf{x} \mapsto \sum_{i=1}^{r} F(\lambda_i(f(\mathbf{x})))$ *and* $\mathbf{x} \mapsto \sum_{i=1}^{r} F(\sigma_i(f(\mathbf{x})))$ *are measurable for all continuous* $F : \mathbb{C} \to \mathbb{C}$*;*
- *the function* $\mathbf{x} \mapsto \|f(\mathbf{x})\|_p$ *is measurable for all* $p \in [1, \infty]$.

2.3.2 L^p-Norms of Matrix-Valued Functions

Let D be any measurable subset of some $\mathbb{R}^k$, let $r \geq 1$, and let $1 \leq p \leq \infty$. For any measurable function $f : D \to \mathbb{C}^{r\times r}$ we define

$$\|f\|_{L^p} = \begin{cases} (\int_D \|f(\mathbf{x})\|_p^p d\mathbf{x})^{1/p}, & \text{if } 1 \leq p < \infty, \\ \operatorname{ess\,sup}_{\mathbf{x}\in D} \|f(\mathbf{x})\|_\infty, & \text{if } p = \infty. \end{cases} \tag{8}$$

Note that this definition is well-posed by Lemma 2.1. In the case where $r = 1$, it reduces to the classical definition of L^p-norms for scalar functions. As highlighted in [6, p. 164], there exist constants $A_p, B_p > 0$ such that, for all $f \in L^p(D, r)$,

$$A_p \|f\|_{L^p}^p \leq \sum_{\alpha,\beta=1}^{r} \|f_{\alpha\beta}\|_{L^p}^p \leq B_p \|f\|_{L^p}^p, \qquad \text{if } 1 \leq p < \infty,$$

$$A_\infty \|f\|_{L^\infty} \leq \max_{\alpha,\beta=1,\ldots,r} \|f_{\alpha\beta}\|_{L^\infty} \leq B_\infty \|f\|_{L^\infty}, \qquad \text{if } p = \infty.$$

This means that $L^p(D, r)$, which we defined in Sect. 2.1 as the set of functions $f : D \to \mathbb{C}^{r\times r}$ such that each component $f_{\alpha\beta}$ belongs to $L^p(D)$, can also be defined as the set of measurable functions $f : D \to \mathbb{C}^{r\times r}$ such that $\|f\|_{L^p} < \infty$. Moreover, if we identify two functions $f, g \in L^p(D, r)$ whenever $f(\mathbf{x}) = g(\mathbf{x})$ for almost every $\mathbf{x} \in D$, then the map $f \mapsto \|f\|_{L^p}$ is a norm on $L^p(D, r)$ which induces on $L^p(D, r)$ the componentwise L^p convergence; that is, $f_m \to f$ in $L^p(D, r)$ according to the norm $\|\cdot\|_{L^p}$ if and only if $(f_m)_{\alpha\beta} \to f_{\alpha\beta}$ in $L^p(D)$ for all $\alpha, \beta = 1, \ldots, r$.

2.3.3 Convergence in Measure

The convergence in measure plays a central role in the theory of block LT sequences, as well as in the theory of block GLT sequences [12]. Two useful lemmas about this convergence are reported below.

Lemma 2.2 *Let* $f_m, g_m, f, g : D \subseteq \mathbb{R}^k \to \mathbb{C}^{r\times r}$ *be measurable functions.*

- *If* $f_m \to f$ *in measure and* $g_m \to g$ *in measure, then* $\alpha f_m + \beta g_m \to \alpha f + \beta g$ *in measure for all* $\alpha, \beta \in \mathbb{C}$.

- *If $f_m \to f$ in measure, $g_m \to g$ in measure, and $\mu_k(D) < \infty$, then $f_m \circ g_m \to f \circ g$ in measure and $f_m g_m \to fg$ in measure.*

Proof See, e.g., [7, Lemma 2.3]. □

Lemma 2.3 *Let g_m, $g : D \to \mathbb{C}^{r\times r}$ be measurable functions defined on a set $D \subset \mathbb{R}^k$ with $0 < \mu_k(D) < \infty$. If $g_m \to g$ in measure, then $\sum_{j=1}^r F(\lambda_j(g_m(\mathbf{x})))$ converges to $\sum_{j=1}^r F(\lambda_j(g(\mathbf{x})))$ in $L^1(D)$ for all $F \in C_c(\mathbb{C})$.*

Proof For each fixed $\mathbf{x} \in D$, consider the optimal matching distance between the spectra of $g_m(\mathbf{x})$ and $g(\mathbf{x})$, namely

$$d(\Lambda(g_m(\mathbf{x})), \Lambda(g(\mathbf{x}))) = \min_{\rho} \max_{1\le j\le r} |\lambda_j(g_m(\mathbf{x})) - \lambda_{\rho(j)}(g(\mathbf{x}))|$$

where the minimum is taken over all permutations ρ of $\{1, \dots, r\}$; see also [3, Section VI.1]. For every $\epsilon > 0$ we have the inclusion

$$\{d(\Lambda(g_m), \Lambda(g)) > \epsilon\} \subseteq \{4(\|g_m\| + \|g\|)^{1-1/r}\|g_m - g\|^{1/r} > \epsilon\} = E_{m,\epsilon},$$

because

$$d(\Lambda(g_m), \Lambda(g)) \le 4(\|g_m\| + \|g\|)^{1-1/r}\|g_m - g\|^{1/r};$$

see [3, Theorem VIII.1.5]. The set $E_{m,\epsilon}$ is measurable by Lemma 2.1, and for each $\mathbf{x} \in D \setminus E_{m,\epsilon}$ we have $d(\Lambda(g_m(\mathbf{x})), \Lambda(g(\mathbf{x}))) \le \epsilon$, hence

$$\left|\sum_{j=1}^r F(\lambda_j(g_m(\mathbf{x}))) - \sum_{j=1}^r F(\lambda_j(g(\mathbf{x})))\right| \le r\omega_F(\epsilon), \quad \forall\, F \in C_c(\mathbb{C}),$$

where $\omega_F(\epsilon) = \sup\{|F(u) - F(v)| : u, v \in \mathbb{C},\ |u - v| \le \epsilon\}$ is the modulus of continuity of F. Thus, for every $F \in C_c(\mathbb{C})$, every m and every $\epsilon > 0$,

$$\begin{aligned}
&\left\|\sum_{j=1}^r F(\lambda_j(g_m)) - \sum_{j=1}^r F(\lambda_j(g))\right\|_{L^1} = \int_D \left|\sum_{j=1}^r F(\lambda_j(g_m(\mathbf{x}))) - \sum_{j=1}^r F(\lambda_j(g(\mathbf{x})))\right| \mathrm{d}\mathbf{x} \\
&= \int_{E_{m,\epsilon}} \left|\sum_{j=1}^r F(\lambda_j(g_m(\mathbf{x}))) - \sum_{j=1}^r F(\lambda_j(g(\mathbf{x})))\right| \mathrm{d}\mathbf{x} \\
&\quad + \int_{D\setminus E_{m,\epsilon}} \left|\sum_{j=1}^r F(\lambda_j(g_m(\mathbf{x}))) - \sum_{j=1}^r F(\lambda_j(g(\mathbf{x})))\right| \mathrm{d}\mathbf{x} \\
&\le 2r\|F\|_\infty \mu_k(E_{m,\epsilon}) + \mu_k(D) r\omega_F(\epsilon). \qquad (9)
\end{aligned}$$

We show that

$$\lim_{m\to\infty} \mu_k(E_{m,\epsilon}) = \lim_{\epsilon\to 0} \omega_F(\epsilon) = 0.$$

Once this is done, passing first to the $\limsup_{m\to\infty}$ and then to the $\lim_{\epsilon\to 0}$ in (9), we obtain that $\sum_{j=1}^{r} F(\lambda_j(g_m)) \to \sum_{j=1}^{r} F(\lambda_j(g))$ in $L^1(D)$, and the thesis is proved. The limit relation $\lim_{\epsilon\to 0} \omega_F(\epsilon) = 0$ follows immediately from the Heine–Cantor theorem, so we only have to prove that $\lim_{m\to\infty} \mu_k(E_{m,\epsilon}) = 0$. By (3) and (4),

$$\begin{aligned} E_{m,\epsilon} &= \left\{4(\|g_m\| + \|g\|)^{1-1/r}\|g_m - g\|^{1/r} \geq \epsilon\right\} \\ &\leq \left\{4(\|g_m - g\| + 2\|g\|)^{1-1/r}\|g_m - g\|^{1/r} \geq \epsilon\right\} \\ &\leq \left\{4\Big(\sum_{\alpha,\beta=1}^{r} |(g_m - g)_{\alpha\beta}| + 2\|g\|\Big)^{1-1/r}\Big(\sum_{\alpha,\beta=1}^{r} |(g_m - g)_{\alpha\beta}|\Big)^{1/r} \geq \epsilon\right\}, \end{aligned}$$

and the limit relation $\lim_{m\to\infty} \mu_k(E_{m,\epsilon}) = 0$ follows from the hypothesis that $g_m \to g$ in measure. □

2.4 Singular Value and Eigenvalue Distribution of a Matrix-Sequence

We introduce in this section the fundamental definitions of singular value and spectral distribution for a given matrix-sequence. Recall from Sect. 2.1 that a matrix-sequence is a sequence of the form $\{A_n\}_n$, where $A_n \in \mathbb{C}^{sn\times sn}$ and s is a *fixed* positive integer.

Definition 2.1 (Singular Value and Eigenvalue Distribution of a Matrix-Sequence) Let $\{A_n\}_n$ be a matrix-sequence and let $f : D \subset \mathbb{R}^k \to \mathbb{C}^{r\times r}$ be a measurable matrix-valued function defined on a set D with $0 < \mu_k(D) < \infty$.

- We say that $\{A_n\}_n$ has a (asymptotic) singular value distribution described by f, and we write $\{A_n\}_n \sim_\sigma f$, if

$$\lim_{n\to\infty} \frac{1}{sn} \sum_{j=1}^{sn} F(\sigma_j(A_n)) = \frac{1}{\mu_k(D)} \int_D \frac{\sum_{i=1}^{r} F(\sigma_i(f(\mathbf{x})))}{r} \mathrm{d}\mathbf{x}, \quad \forall\, F \in C_c(\mathbb{R}). \tag{10}$$

- We say that $\{A_n\}_n$ has a (asymptotic) eigenvalue (or spectral) distribution described by f, and we write $\{A_n\}_n \sim_\lambda f$, if

$$\lim_{n\to\infty} \frac{1}{sn} \sum_{j=1}^{sn} F(\lambda_j(A_n)) = \frac{1}{\mu_k(D)} \int_D \frac{\sum_{i=1}^r F(\lambda_i(f(\mathbf{x})))}{r} \mathrm{d}\mathbf{x}, \quad \forall\, F \in C_c(\mathbb{C}). \tag{11}$$

Note that Definition 2.1 is well-posed by Lemma 2.1, which ensures that the functions $\mathbf{x} \mapsto \sum_{i=1}^r F(\sigma_i(f(\mathbf{x})))$ and $\mathbf{x} \mapsto \sum_{i=1}^r F(\lambda_i(f(\mathbf{x})))$ are measurable. Whenever we write a relation such as $\{A_n\}_n \sim_\sigma f$ or $\{A_n\}_n \sim_\lambda f$, it is understood that f is as in Definition 2.1; that is, f is a measurable function taking values in $\mathbb{C}^{r\times r}$ for some $r \geq 1$ and defined on a subset D of some $\mathbb{R}^k$ with $0 < \mu_k(D) < \infty$. We refer the reader to [9, Remark 1] or to the appendix of [13] for the informal meaning behind the distribution relations (10) and (11).

2.5 Zero-Distributed Sequences

A matrix-sequence $\{Z_n\}_n$ is said to be zero-distributed if $\{Z_n\}_n \sim_\sigma 0$. It is clear that, for any $r \geq 1$, $\{Z_n\}_n \sim_\sigma 0$ is equivalent to $\{Z_n\}_n \sim_\sigma O_r$. Theorems 2.1 and 2.2 provide a characterization of zero-distributed sequences together with a sufficient condition for detecting such sequences [7, Theorems 3.2 and 3.3].

Theorem 2.1 *Let $\{Z_n\}_n$ be a matrix-sequence. The following are equivalent.*

1. $\{Z_n\}_n \sim_\sigma 0$.
2. *For all n we have $Z_n = R_n + N_n$, where $\lim_{n\to\infty} (\mathrm{rank}(R_n)/n) = \lim_{n\to\infty} \|N_n\| = 0$.*

Theorem 2.2 *Let $\{Z_n\}_n$ be a matrix-sequence and suppose there exists $p \in [1, \infty)$ such that $\lim_{n\to\infty} (\|Z_n\|_p^p/n) = 0$. Then $\{Z_n\}_n \sim_\sigma 0$.*

2.6 Sparsely Unbounded Matrix-Sequences

The notion of sparsely unbounded matrix-sequences plays an important role within the framework of the theory of block LT and GLT sequences.

Definition 2.2 (Sparsely Unbounded Matrix-Sequence) A matrix-sequence $\{A_n\}_n$ is said to be sparsely unbounded (s.u.) if for every $M > 0$ there exists n_M such that, for $n \geq n_M$,

$$\frac{\#\{i \in \{1, \ldots, sn\} : \sigma_i(A_n) > M\}}{n} \leq r(M),$$

where $\lim_{M\to\infty} r(M) = 0$.

The following proposition provides equivalent characterizations of s.u. matrix-sequences [7, Proposition 5.3]. This will allow us to show in Proposition 2.2 that the product of two s.u. matrix-sequences is s.u., and in Proposition 2.3 that any matrix-sequence enjoying an asymptotic singular value distribution is s.u.

Proposition 2.1 *Let $\{A_n\}_n$ be a matrix-sequence. The following are equivalent.*

1. *$\{A_n\}_n$ is s.u.*
2. $\lim_{M\to\infty} \limsup_{n\to\infty} \dfrac{\#\{i \in \{1,\dots,ns\} : \sigma_i(A_n) > M\}}{n} = 0.$
3. *For every $M > 0$ there exists n_M such that, for $n \geq n_M$,*

$$A_n = \hat{A}_{n,M} + \tilde{A}_{n,M}, \quad \operatorname{rank}(\hat{A}_{n,M}) \leq r(M)n, \quad \|\tilde{A}_{n,M}\| \leq M,$$

where $\lim_{M\to\infty} r(M) = 0$.

Proposition 2.2 *If $\{A_n\}_n, \{A'_n\}_n$ are s.u. then $\{A_nA'_n\}_n$ is s.u.*

Proof By Proposition 2.1, for every $M > 0$ there exists n_M such that, for $n \geq n_M$,

$$A_n = \hat{A}_{n,M} + \tilde{A}_{n,M}, \quad \operatorname{rank}(\hat{A}_{n,M}) \leq r(M)n, \quad \|\tilde{A}_{n,M}\| \leq M,$$
$$A'_n = \hat{A}'_{n,M} + \tilde{A}'_{n,M}, \quad \operatorname{rank}(\hat{A}'_{n,M}) \leq r(M)n, \quad \|\tilde{A}'_{n,M}\| \leq M,$$

where $\lim_{M\to\infty} r(M) = 0$. Thus, for every $M > 0$ and every $n \geq n_M$ we have

$$A_nA'_n = \hat{A}_{n,M}A'_n + \tilde{A}_{n,M}\hat{A}'_{n,M} + \tilde{A}_{n,M}\tilde{A}'_{n,M} = \hat{B}_{n,M} + \tilde{B}_{n,M},$$

where the matrices $\hat{B}_{n,M} = \hat{A}_{n,M}A'_n + \tilde{A}_{n,M}\hat{A}'_{n,M}$ and $\tilde{B}_{n,M} = \tilde{A}_{n,M}\tilde{A}'_{n,M}$ are such that $\operatorname{rank}(B_{n,M}) \leq 2r(M)n$ and $\|\tilde{B}_{n,M}\| \leq M^2$. We conclude that $\{A_nA'_n\}_n$ is s.u. because condition 3 in Proposition 2.1 is satisfied. □

Proposition 2.3 *If $\{A_n\}_n \sim_\sigma f$ then $\{A_n\}_n$ is s.u.*

Proof Let $D \subset \mathbb{R}^k$ be the domain of the matrix-valued function $f : D \to \mathbb{C}^{r\times r}$. Fix $M > 0$ and take $F_M \in C_c(\mathbb{R})$ such that $F_M = 1$ over $[0, M/2]$, $F_M = 0$ over $[M, \infty)$ and $0 \leq F_M \leq 1$ over $\mathbb{R}$. Note that $F_M \leq \chi_{[0,M]}$ over $[0, \infty)$. Then,

$$\begin{aligned}
&\frac{\#\{i \in \{1,\dots,sn\} : \sigma_i(A_n) > M\}}{sn} \\
&= 1 - \frac{\#\{i \in \{1,\dots,sn\} : \sigma_i(A_n) \leq M\}}{sn} = 1 - \frac{1}{sn}\sum_{i=1}^{sn} \chi_{[0,M]}(\sigma_i(A_n)) \\
&\leq 1 - \frac{1}{sn}\sum_{i=1}^{sn} F_M(\sigma_i(A_n)) \overset{n\to\infty}{\longrightarrow} 1 - \frac{1}{\mu_k(D)}\int_D \frac{\sum_{i=1}^{r} F_M(\sigma_i(f(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}
\end{aligned}$$

and

$$\limsup_{n\to\infty} \frac{\#\{i \in \{1,\dots,sn\} : \sigma_i(A_n) > M\}}{sn} \le 1 - \frac{1}{\mu_k(D)} \int_D \frac{\sum_{i=1}^r F_M(\sigma_i(f(\mathbf{x})))}{r} \mathrm{d}\mathbf{x}.$$

Since $\frac{1}{r}\sum_{i=1}^r F_M(\sigma_i(f(\mathbf{x}))) \to 1$ pointwise and $\left|\frac{1}{r}\sum_{i=1}^r F_M(\sigma_i(f(\mathbf{x})))\right| \le 1$, the dominated convergence theorem yields

$$\lim_{M\to\infty} \int_D \frac{\sum_{i=1}^r F_M(\sigma_i(f(\mathbf{x})))}{r} \mathrm{d}\mathbf{x} = \mu_k(D),$$

and so

$$\lim_{M\to\infty} \limsup_{n\to\infty} \frac{\#\{i \in \{1,\dots,sn\} : \sigma_i(A_n) > M\}}{sn} = 0.$$

This means that condition 2 in Proposition 2.1 is satisfied, i.e., $\{A_n\}_n$ is s.u. □

2.7 Block Toeplitz Matrices

A matrix of the form

$$[A_{i-j}]_{i,j=1}^n = \begin{bmatrix} A_0 & A_{-1} & \cdots & \cdots & A_{-(n-1)} \\ A_1 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & A_{-1} \\ A_{n-1} & \cdots & \cdots & A_1 & A_0 \end{bmatrix} \in \mathbb{C}^{sn\times sn}, \tag{12}$$

with blocks $A_k \in \mathbb{C}^{s\times s}$ for $k = -(n-1),\dots,n-1$, is called a block Toeplitz matrix. If $s = 1$, it is simply referred to as a Toeplitz matrix. We have seen in Sect. 1 that a function $f \in L^1([-\pi,\pi],s)$ gives rise via its Fourier coefficients to a sequence of block Toeplitz matrices $\{T_n(f)\}_n$. We call $\{T_n(f)\}_n$ the block Toeplitz sequence associated with f, which in turn is called the generating function of $\{T_n(f)\}_n$.

For each fixed $s, n \in \mathbb{N}$, the map $T_n(\cdot) : L^1([-\pi,\pi],s) \to \mathbb{C}^{sn\times sn}$ is linear, i.e.,

$$T_n(\alpha f + \beta g) = \alpha T_n(f) + \beta T_n(g), \quad \alpha,\beta \in \mathbb{C}, \quad f,g \in L^1([-\pi,\pi],s). \tag{13}$$

Moreover, it is clear from the definition that $T_n(I_s) = I_{sn}$. For every $f \in L^1([-\pi,\pi],s)$, let f^* be its conjugate transpose. It is not difficult to show that

$$T_n(f)^* = T_n(f^*), \quad f \in L^1([-\pi,\pi],s), \quad s,n \in \mathbb{N}. \tag{14}$$

In particular, if f is Hermitian, i.e., $f(\theta)$ is Hermitian for all $\theta \in [-\pi, \pi]$, then the block Toeplitz matrices $T_n(f)$ are Hermitian.

Theorem 2.3 is a fundamental result concerning block Toeplitz matrices. It provides the singular value distribution of block Toeplitz sequences generated by a matrix-valued function $f \in L^1([-\pi, \pi], s)$ and the spectral distribution of block Toeplitz sequences generated by a Hermitian matrix-valued function $f \in L^1([-\pi, \pi], s)$. For the eigenvalues it goes back to Szegő [14], and for the singular values it was established by Avram [1] and Parter [15]. They assumed that $f \in L^\infty([-\pi, \pi], s)$ and $s = 1$; see [4, Section 5] and [5, Section 10.14] for more on the subject in the case of L^∞ generating functions. The extension to $f \in L^1([-\pi, \pi], s)$ with $s = 1$ was performed by Tyrtyshnikov and Zamarashkin [23–25], and the final generalization to $f \in L^1([-\pi, \pi], s)$ with $s \geq 1$ is due to Tilli [21]. We also refer the reader to [10] for a proof of Theorem 2.3 based on the notion of approximating classes of sequences (see Sect. 3); the proof in [10] is made only in the case of eigenvalues for $s = 1$, but the argument is general and can be extended to singular values and matrix-valued generating functions.

Theorem 2.3 *If $f \in L^1([-\pi, \pi], s)$ then $\{T_n(f)\}_n \sim_\sigma f$. If moreover f is Hermitian then $\{T_n(f)\}_n \sim_\lambda f$.*

Important inequalities involving Toeplitz matrices and Schatten p-norms originally appeared in [20, Corollary 4.2]. They have been generalized to block Toeplitz matrices in [17, Corollary 3.5]. We report them in the next theorem for future use.

Theorem 2.4 *Let $f \in L^p([-\pi, \pi], s)$ and $n \in \mathbb{N}$. Then, using the natural convention $1/\infty = 0$, the inequality $\|T_n(f)\|_p \leq n^{1/p} \|f\|_{L^p}$ holds for all $p \in [1, \infty]$.*

The next theorem is the last result we shall need about block Toeplitz matrices.

Theorem 2.5 *Let $f_i \in L^\infty([-\pi, \pi], s)$ for $i = 1, \ldots, q$. Then,*

$$\lim_{n\to\infty} \frac{\left\| \prod_{i=1}^q T_n(f_i) - T_n\left(\prod_{i=1}^q f_i\right) \right\|_1}{n} = 0. \tag{15}$$

Proof For $q = 2$ the result is proved in [6, Proposition 2]. In the general case we proceed by induction. Fix $p \geq 3$ and suppose that the result holds for $q = p - 1$. If $q = p$, using (5) and Theorem 2.4 we obtain

$$\begin{aligned}
&\frac{1}{n} \left\| \prod_{i=1}^p T_n(f_i) - T_n\left(\prod_{i=1}^p f_i\right) \right\|_1 \\
&= \frac{1}{n} \left\| \prod_{i=1}^p T_n(f_i) - \left(\prod_{i=1}^{p-2} T_n(f_i)\right) T_n(f_{p-1} f_p) \right.
\end{aligned}$$

$$
\begin{aligned}
&+\Big(\prod_{i=1}^{p-2} T_n(f_i)\Big) T_n(f_{p-1}f_p) - T_n\Big(\prod_{i=1}^{p} f_i\Big)\Big\|_1 \\
&\le \frac{1}{n}\Big\|\Big(\prod_{i=1}^{p-2} T_n(f_i)\Big)(T_n(f_{p-1})T_n(f_p) - T_n(f_{p-1}f_p))\Big\|_1 \\
&\quad + \frac{1}{n}\Big\|\Big(\prod_{i=1}^{p-2} T_n(f_i)\Big) T_n(f_{p-1}f_p) - T_n\Big(\prod_{i=1}^{p} f_i\Big)\Big\|_1 \\
&\le \frac{1}{n}\Big(\prod_{i=1}^{p-2} \|f_i\|_{L^\infty}\Big)\, \big\|T_n(f_{p-1})T_n(f_p) - T_n(f_{p-1}f_p)\big\|_1 \\
&\quad + \frac{1}{n}\Big\|\Big(\prod_{i=1}^{p-2} T_n(f_i)\Big) T_n(f_{p-1}f_p) - T_n\Big(\Big(\prod_{i=1}^{p-2} f_i\Big)(f_{p-1}f_p)\Big)\Big\|_1 .
\end{aligned}
$$

Now, the first term in the right-hand side tends to zero as $n \to \infty$ by [6, Proposition 2], and the second term tends to zero as $n \to \infty$ by the induction hypothesis. □

3 Approximating Classes of Sequences

The notion of approximating classes of sequences (a.c.s.), which is fundamental to the theory of block LT and GLT sequences, is due to the second author [16]. It should be said, however, that the underlying idea was already present in the pioneering papers by Tilli [22] and Tyrtyshnikov [23].

3.1 The a.c.s. Notion

The formal definition of a.c.s. is given below.

Definition 3.1 (Approximating Class of Sequences) Let $\{A_n\}_n$ a matrix-sequence. An approximating class of sequences (a.c.s.) for $\{A_n\}_n$ is a sequence of matrix-sequences $\{\{B_{n,m}\}_n\}_m$ with the following property: for every m there exists n_m such that, for $n \ge n_m$,

$$
A_n = B_{n,m} + R_{n,m} + N_{n,m}, \quad \operatorname{rank}(R_{n,m}) \le c(m)n, \quad \|N_{n,m}\| \le \omega(m), \tag{16}
$$

where n_m, $c(m)$, $\omega(m)$ depend only on m and $\lim_{m\to\infty} c(m) = \lim_{m\to\infty} \omega(m) = 0$.

Roughly speaking, $\{\{B_{n,m}\}_n\}_m$ is an a.c.s. for $\{A_n\}_n$ if, for all sufficiently large m, the sequence $\{B_{n,m}\}_n$ approximates (asymptotically) the sequence $\{A_n\}_n$ in the sense that A_n is eventually equal to $B_{n,m}$ plus a small-rank matrix (with respect to the matrix size sn) plus a small-norm matrix. As proved in [7, Section 5.2], the a.c.s. notion is a notion of convergence in the space of matrix-sequences

$$\mathscr{E} = \{\{A_n\}_n : \ \{A_n\}_n \text{ is a matrix-sequence}\}.$$

More precisely, there exists a pseudometric $d_{\text{a.c.s.}}$ in $\mathscr{E}$ such that $\{\{B_{n,m}\}_n\}_m$ is an a.c.s. for $\{A_n\}_n$ if and only if $d_{\text{a.c.s.}}(\{B_{n,m}\}_n, \{A_n\}_n) \to 0$ as $m \to \infty$. It was shown in [2] that $d_{\text{a.c.s.}}$ is complete, i.e., any Cauchy sequence $\{\{B_{n,m}\}_n\}_m$ in $(\mathscr{E}, d_{\text{a.c.s.}})$ converges to some limit sequence $\{A_n\}_n$. Moreover, for every $\{A_n\}_n, \{B_n\}_n \in \mathscr{E}$ we have

$$d_{\text{a.c.s.}}(\{A_n\}_n, \{B_n\}_n) = 0 \quad \Longleftrightarrow \quad \{A_n - B_n\}_n \sim_\sigma 0. \tag{17}$$

Based on the above topological interpretation, we will use the convergence notation $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ to indicate that $\{\{B_{n,m}\}_n\}_m$ is an a.c.s. for $\{A_n\}_n$.

3.2 The a.c.s. Tools for Computing Singular Value and Eigenvalue Distributions

The importance of the a.c.s. notion resides in Theorems 3.1 and 3.2, which are easily derived from the results in [7, Section 5.3], as shown below.

Theorem 3.1 *Let $\{A_n\}_n, \{B_{n,m}\}_n$ be matrix-sequences and let $f, f_m : D \to \mathbb{C}^{r\times r}$ be measurable functions defined on a set $D \subset \mathbb{R}^k$ with $0 < \mu_k(D) < \infty$. Assume that:*

1. $\{B_{n,m}\}_n \sim_\sigma f_m$ *for every* m;
2. $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$;
3. $f_m \to f$ *in measure.*

Then $\{A_n\}_n \sim_\sigma f$.

Proof Let $F \in C_c(\mathbb{R})$. For every n, m we have

$$\left| \frac{1}{sn} \sum_{j=1}^{sn} F(\sigma_j(A_n)) - \frac{1}{\mu_k(D)} \int_D \frac{\sum_{j=1}^{r} F(\sigma_j(f(\mathbf{x})))}{r} \mathrm{d}\mathbf{x} \right|$$

$$\leq \left| \frac{1}{sn} \sum_{j=1}^{sn} F(\sigma_j(A_n)) - \frac{1}{sn} \sum_{j=1}^{sn} F(\sigma_j(B_{n,m})) \right|$$

$$+\left|\frac{1}{sn}\sum_{j=1}^{sn}F(\sigma_j(B_{n,m}))-\frac{1}{\mu_k(D)}\int_D\frac{\sum_{j=1}^{r}F(\sigma_j(f_m(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}\right|$$

$$+\left|\frac{1}{\mu_k(D)}\int_D\frac{\sum_{j=1}^{r}F(\sigma_j(f_m(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}-\frac{1}{\mu_k(D)}\int_D\frac{\sum_{j=1}^{r}F(\sigma_j(f(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}\right|. \tag{18}$$

The second term in the right-hand side tends to 0 as $n\to\infty$ by hypothesis, while the third one tends to 0 as $m\to\infty$ by Lemmas 2.2 and 2.3 (take into account that $\sum_{j=1}^{r}F(\sigma_j(f_m(\mathbf{x})))=\sum_{j=1}^{r}G(\lambda_j(f_m(\mathbf{x})f_m(\mathbf{x})^*))$ where $G(z)=F(\sqrt{|z|})$ belongs to $C_c(\mathbb{C})$). For the first term we have

$$\lim_{m\to\infty}\limsup_{n\to\infty}\left|\frac{1}{sn}\sum_{j=1}^{sn}F(\sigma_j(A_n))-\frac{1}{sn}\sum_{j=1}^{sn}F(\sigma_j(B_{n,m}))\right|=0;$$

see [7, Lemma 5.3]. Therefore, passing first to the $\limsup_{n\to\infty}$ and then to the $\lim_{m\to\infty}$ in (18), we get the thesis. □

Theorem 3.2 *Let $\{A_n\}_n,\{B_{n,m}\}_n$ be Hermitian matrix-sequences and let $f,f_m:D\to\mathbb{C}^{r\times r}$ be measurable functions defined on a set $D\subset\mathbb{R}^k$ with $0<\mu_k(D)<\infty$. Assume that:*

1. *$\{B_{n,m}\}_n\sim_\lambda f_m$ for every m;*
2. *$\{B_{n,m}\}_n\xrightarrow{\text{a.c.s.}}\{A_n\}_n$;*
3. *$f_m\to f$ in measure.*

Then $\{A_n\}_n\sim_\lambda f$.

Proof Let $F\in C_c(\mathbb{R})$. For all n,m we have

$$\left|\frac{1}{sn}\sum_{j=1}^{sn}F(\lambda_j(A_n))-\frac{1}{\mu_k(D)}\int_D\frac{\sum_{j=1}^{r}F(\lambda_j(f(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}\right|$$

$$\le\left|\frac{1}{sn}\sum_{j=1}^{sn}F(\lambda_j(A_n))-\frac{1}{sn}\sum_{j=1}^{sn}F(\lambda_j(B_{n,m}))\right|$$

$$+\left|\frac{1}{sn}\sum_{j=1}^{sn}F(\lambda_j(B_{n,m}))-\frac{1}{\mu_k(D)}\int_D\frac{\sum_{j=1}^{r}F(\lambda_j(f_m(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}\right|$$

$$+\left|\frac{1}{\mu_k(D)}\int_D\frac{\sum_{j=1}^{r}F(\lambda_j(f_m(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}-\frac{1}{\mu_k(D)}\int_D\frac{\sum_{j=1}^{r}F(\lambda_j(f(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}\right|. \tag{19}$$

The second term in the right-hand side tends to 0 as $n \to \infty$ by hypothesis, while the third one tends to 0 as $m \to \infty$ by Lemma 2.3. For the first term in the right-hand side we have

$$\lim_{m\to\infty} \limsup_{n\to\infty} \left| \frac{1}{sn} \sum_{j=1}^{sn} F(\lambda_j(A_n)) - \frac{1}{sn} \sum_{j=1}^{sn} F(\lambda_j(B_{n,m})) \right| = 0;$$

see [7, Lemma 5.5]. Therefore, passing first to the $\limsup_{n\to\infty}$ and then to the $\lim_{m\to\infty}$ in (19), we get the thesis. □

3.3 The a.c.s. Algebra

Theorem 3.3 collects important algebraic properties possessed by the a.c.s.

Theorem 3.3 *Let* $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ *and* $\{B'_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A'_n\}_n$. *The following properties hold.*

- $\{B^*_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A^*_n\}_n$.
- $\{\alpha B_{n,m} + \beta B'_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{\alpha A_n + \beta A'_n\}_n$ *for all* $\alpha, \beta \in \mathbb{C}$.
- *If* $\{A_n\}_n, \{A'_n\}_n$ *are s.u. then* $\{B_{n,m}B'_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_nA'_n\}_n$.
- *If* $\{C_n\}_n$ *is s.u. then* $\{B_{n,m}C_n\}_n \xrightarrow{\text{a.c.s.}} \{A_nC_n\}_n$.

Proof For the proofs of statements 1 to 3, see [7, Section 5.4]. We prove statement 4. Since $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$, for every m there exists n_m such that, for $n \geq n_m$,

$$A_n = B_{n,m} + R_{n,m} + N_{n,m}, \quad \operatorname{rank}(R_{n,m}) \leq c(m)n, \quad \|N_{n,m}\| \leq \omega(m),$$

where $\lim_{m\to\infty} c(m) = \lim_{m\to\infty} \omega(m) = 0$. Since $\{C_n\}_n$ is s.u., by Proposition 2.1 for every $M > 0$ there exists $n(M)$ such that, for $n \geq n(M)$,

$$C_n = \hat{C}_{n,M} + \tilde{C}_{n,M}, \quad \operatorname{rank}(\hat{C}_{n,M}) \leq r(M)n, \quad \|\tilde{C}_{n,m}\| \leq M,$$

where $\lim_{M\to\infty} r(M) = 0$. Setting $M_m = \omega(m)^{-1/2}$, for every m and every $n \geq \max(n_m, n(M_m))$ we have

$$A_nC_n = B_{n,m}C_n + R'_{n,m} + N'_{n,m},$$

where $R'_{n,m} = R_{n,m}C_n + N_{n,m}\hat{C}_{n,M_m}$ and $N'_{n,m} = N_{n,m}\tilde{C}_{n,M_m}$ satisfy

$$\operatorname{rank}(R'_{n,m}) \leq (c(m) + r(M_m))n, \quad \|N'_{n,m}\| \leq \omega(m)^{1/2}.$$

This shows that $\{B_{n,m}C_n\}_n \xrightarrow{\text{a.c.s.}} \{A_nC_n\}_n$. □

Remark 3.1 As a consequence of Theorem 3.3 and Proposition 2.2, if $\{A_n^{(i,j)}\}_n$ is an s.u. matrix-sequence and $\{B_{n,m}^{(i,j)}\}_n \xrightarrow{\text{a.c.s.}} \{A_n^{(i,j)}\}_n$ for $i = 1, \dots, p$ and $j = 1, \dots, q_i$, then $\left\{\sum_{i=1}^{p}\prod_{j=1}^{q_i} B_{n,m}^{(i,j)}\right\}_n \xrightarrow{\text{a.c.s.}} \left\{\sum_{i=1}^{p}\prod_{j=1}^{q_i} A_n^{(i,j)}\right\}_n$.

3.4 A Criterion to Identify a.c.s.

A useful criterion to show that a sequence of matrix-sequences $\{\{B_{n,m}\}_n\}_m$ is an a.c.s. for another matrix-sequence $\{A_n\}_n$ is reported in the next theorem [7, Corollary 5.3].

Theorem 3.4 *Let $\{A_n\}_n$ be a matrix-sequence, let $\{\{B_{n,m}\}_n\}_m$ be a sequence of matrix-sequences, and let $1 \le p < \infty$. Suppose that for every m there exists n_m such that, for $n \ge n_m$,*

$$\|A_n - B_{n,m}\|_p^p \le \epsilon(m,n)n,$$

where $\lim_{m\to\infty} \limsup_{n\to\infty} \epsilon(m,n) = 0$. Then $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$.

4 Block Locally Toeplitz Sequences

The theory of (scalar) LT sequences dates back to Tilli's pioneering paper [22]. It was then carried forward in [18, 19], and it was finally developed in a systematic way in [7, Chapter 7] and [8, Chapter 4]. In this section we develop the theory of block LT sequences, which has been suggested in [19, Section 3.3] but so far has never been addressed in a systematic way. We recall that the theory of block LT sequences is a necessary intermediate step toward a rigorous mathematical theory of block GLT sequences [12].

4.1 The Block LT Operator

Similarly to the case of (scalar) LT sequences, the theory of block LT sequences begins with the definition of block locally Toeplitz operator.

Definition 4.1 (Block Locally Toeplitz Operator) Let $m, n, s \in \mathbb{N}$, let $a : [0,1] \to \mathbb{C}^{s\times s}$, and let $f \in L^1([-\pi,\pi], s)$. The block locally Toeplitz (LT) operator is defined as the following $ns \times ns$ matrix:

$$LT_n^m(a,f) = \bigoplus_{k=1}^{m} T_{\lfloor n/m \rfloor}\left(a\left(\frac{k}{m}\right) \circ f\right) \oplus O_{s(n \bmod m)}.$$

It is understood that $LT_n^m(a, f) = O_{sn}$ when $n < m$ and that the term $O_{s(n \bmod m)}$ is not present when n is a multiple of m.

In this section we investigate the properties of the block LT operator. We first note that Definition 4.1 reduces to the definition of the (scalar) LT operator [7, Definition 7.1] if $s = 1$. Moreover, for every $m, n, s \in \mathbb{N}$, every $a, b : [0, 1] \to \mathbb{C}^{s\times s}$, every $f, g \in L^1([-\pi, \pi], s)$, and every $\alpha, \beta \in \mathbb{C}$, we have

$$LT_n^m(a, f)^* = LT_n^m(a^*, f^*), \tag{20}$$

$$LT_n^m(\alpha a + \beta b, f) = \alpha LT_n^m(a, f) + \beta LT_n^m(b, f), \tag{21}$$

$$LT_n^m(a, \alpha f + \beta g) = \alpha LT_n^m(a, f) + \beta LT_n^m(a, g). \tag{22}$$

Proposition 4.1 *Let $a : [0, 1] \to \mathbb{C}^{s\times s}$, $f \in L^1([-\pi, \pi], s)$, and $n, m \in \mathbb{N}$. Then,*

$$\|LT_n^m(a, f)\|_1 \le ns \max_{k=1,\dots,m} \left\|a\left(\frac{k}{m}\right)\right\| \|f\|_{L^1}.$$

Proof From Theorem 2.4 and (3), (6), (7), (8), we obtain

$$\begin{aligned}
\|LT_n^m(a, f)\|_1 &= \sum_{k=1}^{m} \left\| T_{\lfloor n/m \rfloor}\left(a\left(\frac{k}{m}\right) \circ f\right)\right\|_1 \le \sum_{k=1}^{m} \left\lfloor \frac{n}{m} \right\rfloor \left\|a\left(\frac{k}{m}\right) \circ f\right\|_{L^1} \\
&= \sum_{k=1}^{m} \left\lfloor \frac{n}{m} \right\rfloor \int_{-\pi}^{\pi} \left\|a\left(\frac{k}{m}\right) \circ f(\theta)\right\|_1 \mathrm{d}\theta \le \sum_{k=1}^{m} \left\lfloor \frac{n}{m} \right\rfloor \int_{-\pi}^{\pi} s \left\|a\left(\frac{k}{m}\right) \circ f(\theta)\right\| \mathrm{d}\theta \\
&\le \sum_{k=1}^{m} s \left\lfloor \frac{n}{m} \right\rfloor \int_{-\pi}^{\pi} \left\|a\left(\frac{k}{m}\right)\right\| \|f(\theta)\| \mathrm{d}\theta \le \sum_{k=1}^{m} s \left\|a\left(\frac{k}{m}\right)\right\| \left\lfloor \frac{n}{m} \right\rfloor \int_{-\pi}^{\pi} \|f(\theta)\|_1 \mathrm{d}\theta \\
&\le ns \max_{k=1,\dots,m} \left\|a\left(\frac{k}{m}\right)\right\| \|f\|_{L^1},
\end{aligned}$$

which completes the proof. □

Remark 4.1 Let $a : [0, 1] \to \mathbb{C}^{s\times s}$ be bounded and take any sequence $\{f_h\}_h \subset L^1([-\pi, \pi], s)$ such that $f_h \to f$ in $L^1([-\pi, \pi], s)$ as $h \to \infty$. Then, by (22) and Proposition 4.1, for every $n, m \in \mathbb{N}$ we have

$$\begin{aligned}
\|LT_n^m(a, f) - LT_n^m(a, f_h)\|_1 &= \|LT_n^m(a, f - f_h)\|_1 \\
&\le ns \sup_{x\in[0,1]} \|a(x)\| \, \|f - f_h\|_{L^1}.
\end{aligned}$$

By Theorem 3.4, this implies that $\{LT_n^m(a, f_h)\}_n \xrightarrow{\text{a.c.s.}} \{LT_n^m(a, f)\}_n$ for each $m \in \mathbb{N}$.

Proposition 4.2 *Let* $a^{(i,j)} : [0,1] \to \mathbb{C}^{s\times s}$ *be bounded and let* $f^{(i,j)} \in L^\infty([-\pi,\pi],s)$ *for* $i = 1,\dots,p$ *and* $j = 1,\dots,q_i$. *Then, for every* $n, m \in \mathbb{N}$,

$$\left\| \sum_{i=1}^{p}\prod_{j=1}^{q_i} LT_n^m(a^{(i,j)}, f^{(i,j)}) - \bigoplus_{k=1}^{m} T_{\lfloor n/m\rfloor}\left(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\left(a^{(i,j)}\left(\frac{k}{m}\right)\circ f^{(i,j)}\right)\right) \oplus O_{s(n \bmod m)} \right\|_1 \le \epsilon(\lfloor n/m\rfloor)n, \tag{23}$$

where

$$\epsilon(\lfloor n/m\rfloor) = \frac{1}{m}\sum_{k=1}^{m}\sum_{i=1}^{p} \frac{\left\| \prod_{j=1}^{q_i} T_{\lfloor n/m\rfloor}\left(a^{(i,j)}(\frac{k}{m})\circ f^{(i,j)}\right) - T_{\lfloor n/m\rfloor}\left(\prod_{j=1}^{q_i}\left(a^{(i,j)}(\frac{k}{m})\circ f^{(i,j)}\right)\right)\right\|_1}{\lfloor n/m\rfloor}$$

and $\lim_{n\to\infty}\epsilon(\lfloor n/m\rfloor) = 0$. *In particular, for every* $m \in \mathbb{N}$ *we have*

$$d_{\text{a.c.s.}}\left(\left\{\sum_{i=1}^{p}\prod_{j=1}^{q_i} LT_n^m(a^{(i,j)}, f^{(i,j)})\right\}_n, \left\{\bigoplus_{k=1}^{m} T_{\lfloor n/m\rfloor}\left(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\left(a^{(i,j)}\left(\frac{k}{m}\right)\circ f^{(i,j)}\right)\right) \oplus O_{s(n \bmod m)}\right\}_n\right) = 0. \tag{24}$$

Proof By the properties of direct sums and the linearity of the map $T_n(\cdot)$, we obtain

$$\begin{aligned}
&\left\| \sum_{i=1}^{p}\prod_{j=1}^{q_i} LT_n^m(a^{(i,j)}, f^{(i,j)}) - \bigoplus_{k=1}^{m} T_{\lfloor n/m\rfloor}\left(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\left(a^{(i,j)}\left(\frac{k}{m}\right)\circ f^{(i,j)}\right)\right) \oplus O_{s(n \bmod m)} \right\|_1 \\
&= \left\| \bigoplus_{k=1}^{m}\sum_{i=1}^{p}\prod_{j=1}^{q_i} T_{\lfloor n/m\rfloor}\left(a^{(i,j)}\left(\frac{k}{m}\right)\circ f^{(i,j)}\right) \oplus O_{s(n \bmod m)} - \bigoplus_{k=1}^{m}\sum_{i=1}^{p} T_{\lfloor n/m\rfloor}\left(\prod_{j=1}^{q_i}\left(a^{(i,j)}\left(\frac{k}{m}\right)\circ f^{(i,j)}\right)\right) \oplus O_{s(n \bmod m)} \right\|_1
\end{aligned}$$

$$
\begin{aligned}
&= \left\| \bigoplus_{k=1}^{m} \sum_{i=1}^{p} \left[\prod_{j=1}^{q_i} T_{\lfloor n/m \rfloor}\left(a^{(i,j)}\left(\frac{k}{m}\right) \circ f^{(i,j)}\right) \right.\right. \\
&\qquad \left.\left. - T_{\lfloor n/m \rfloor}\left(\prod_{j=1}^{q_i}\left(a^{(i,j)}\left(\frac{k}{m}\right) \circ f^{(i,j)}\right)\right)\right] \oplus O_{s(n \bmod m)} \right\|_1 \\
&\le \sum_{k=1}^{m}\sum_{i=1}^{p} \left\| \prod_{j=1}^{q_i} T_{\lfloor n/m \rfloor}\left(a^{(i,j)}\left(\frac{k}{m}\right) \circ f^{(i,j)}\right) \right. \\
&\qquad \left. - T_{\lfloor n/m \rfloor}\left(\prod_{j=1}^{q_i}\left(a^{(i,j)}\left(\frac{k}{m}\right) \circ f^{(i,j)}\right)\right) \right\|_1 \\
&= \lfloor n/m \rfloor \sum_{k=1}^{m}\sum_{i=1}^{p} \frac{1}{\lfloor n/m \rfloor} \left\| \prod_{j=1}^{q_i} T_{\lfloor n/m \rfloor}\left(a^{(i,j)}\left(\frac{k}{m}\right) \circ f^{(i,j)}\right) \right. \\
&\qquad \left. - T_{\lfloor n/m \rfloor}\left(\prod_{j=1}^{q_i}\left(a^{(i,j)}\left(\frac{k}{m}\right) \circ f^{(i,j)}\right)\right) \right\|_1 \\
&\le (n/m) \sum_{k=1}^{m}\sum_{i=1}^{p} \frac{1}{\lfloor n/m \rfloor} \left\| \prod_{j=1}^{q_i} T_{\lfloor n/m \rfloor}\left(a^{(i,j)}\left(\frac{k}{m}\right) \circ f^{(i,j)}\right) \right. \\
&\qquad \left. - T_{\lfloor n/m \rfloor}\left(\prod_{j=1}^{q_i}\left(a^{(i,j)}\left(\frac{k}{m}\right) \circ f^{(i,j)}\right)\right) \right\|_1 .
\end{aligned}
$$

This proves (23). Since $\epsilon(\lfloor n/m \rfloor) \to 0$ as $n \to \infty$ by Theorem 2.5 (recall that $a^{(i,j)}(\frac{k}{m}) \circ f^{(i,j)} \in L^\infty([-\pi,\pi], s)$), Eq. (24) follows immediately from (23), Theorem 2.2, and (17). □

Theorem 4.1 *Suppose that $a^{(i,j)} : [0,1] \to \mathbb{C}^{s\times s}$ is Riemann-integrable and $f^{(i,j)} \in L^\infty([-\pi,\pi], s)$ for $i = 1,\dots,p$ and $j = 1,\dots,q_i$. Then, for every $m \in \mathbb{N}$,*

$$
\begin{aligned}
&\left\{ \bigoplus_{k=1}^{m} T_{\lfloor n/m \rfloor}\left(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\left(a^{(i,j)}\left(\frac{k}{m}\right) \circ f^{(i,j)}\right)\right) \oplus O_{s(n \bmod m)} \right\}_n \\
&\quad \sim_\sigma \sum_{i=1}^{p}\prod_{j=1}^{q_i}\left(a_m^{(i,j)}(x) \circ f^{(i,j)}(\theta)\right),
\end{aligned} \tag{25}
$$

where

$$a_m^{(i,j)}(x) = \sum_{k=1}^{m} a^{(i,j)}\Big(\frac{k}{m}\Big)\chi_{\left[\frac{k-1}{m},\frac{k}{m}\right)}(x). \tag{26}$$

Proof By the properties of direct sums, the singular values of the matrix

$$\bigoplus_{k=1}^{m} T_{\lfloor n/m\rfloor}\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a^{(i,j)}\Big(\frac{k}{m}\Big)\circ f^{(i,j)}\Big)\Big) \oplus O_{s(n \bmod m)}$$

are given by

$$\sigma_\ell\Big(T_{\lfloor n/m\rfloor}\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a^{(i,j)}\Big(\frac{k}{m}\Big)\circ f^{(i,j)}\Big)\Big)\Big), \quad \ell = 1,\dots,s\Big\lfloor\frac{n}{m}\Big\rfloor, \quad k = 1,\dots,m,$$

plus further $s(n \bmod m)$ singular values which are equal to 0. Therefore, by Theorem 2.3, since $\sum_{i=1}^{p}\prod_{j=1}^{q_i}(a^{(i,j)}(\frac{k}{m})\circ f^{(i,j)}) \in L^\infty([-\pi,\pi],s)$, for any $F \in C_c(\mathbb{R})$ we have

$$\begin{aligned}
&\lim_{n\to\infty}\frac{1}{sn}\sum_{r=1}^{sn}F\Big(\sigma_r\Big(\bigoplus_{k=1}^{m}T_{\lfloor n/m\rfloor}\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a^{(i,j)}\Big(\frac{k}{m}\Big)\circ f^{(i,j)}\Big)\Big)\oplus O_{s(n\bmod m)}\Big)\Big)\\
&=\lim_{n\to\infty}\frac{ms\lfloor\frac{n}{m}\rfloor}{sn}\frac{1}{m}\sum_{k=1}^{m}\frac{1}{s\lfloor\frac{n}{m}\rfloor}\sum_{\ell=1}^{s\lfloor\frac{n}{m}\rfloor}F\Big(\sigma_\ell\Big(T_{\lfloor n/m\rfloor}\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a^{(i,j)}\Big(\frac{k}{m}\Big)\circ f^{(i,j)}\Big)\Big)\Big)\Big)\\
&=\frac{1}{m}\sum_{k=1}^{m}\frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{1}{s}\sum_{\ell=1}^{s}F\Big(\sigma_\ell\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a^{(i,j)}\Big(\frac{k}{m}\Big)\circ f^{(i,j)}(\theta)\Big)\Big)\Big)\mathrm{d}\theta\\
&=\frac{1}{2\pi}\int_0^1\int_{-\pi}^{\pi}\frac{1}{s}\sum_{\ell=1}^{s}F\Big(\sigma_\ell\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a_m^{(i,j)}(x)\circ f^{(i,j)}(\theta)\Big)\Big)\Big)\mathrm{d}\theta\mathrm{d}x.
\end{aligned}\tag{27}$$

This concludes the proof. □

Theorem 4.2 *Suppose that $a^{(i,j)} : [0,1] \to \mathbb{C}^{s\times s}$ is Riemann-integrable and $f^{(i,j)} \in L^\infty([-\pi,\pi],s)$ for $i = 1,\dots,p$ and $j = 1,\dots,q_i$. Then, for every $m \in \mathbb{N}$,*

$$\begin{aligned}
&\Big\{\Re\Big(\bigoplus_{k=1}^{m}T_{\lfloor n/m\rfloor}\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a^{(i,j)}\Big(\frac{k}{m}\Big)\circ f^{(i,j)}\Big)\Big)\oplus O_{s(n\bmod m)}\Big)\Big\}_n\\
&\quad\sim_\lambda \Re\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a_m^{(i,j)}(x)\circ f^{(i,j)}(\theta)\Big)\Big).
\end{aligned}\tag{28}$$

where $a_m^{(i,j)}$ is defined in (26).

Proof The proof follows the same pattern as the proof of Theorem 4.1. The eigenvalues of the matrix

$$\Re\Big(\bigoplus_{k=1}^{m} T_{\lfloor n/m\rfloor}\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a^{(i,j)}\Big(\frac{k}{m}\Big)\circ f^{(i,j)}\Big)\Big)\ \oplus\ O_{s(n \bmod m)}\Big)$$
$$=\bigoplus_{k=1}^{m} T_{\lfloor n/m\rfloor}\Big(\Re\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a^{(i,j)}\Big(\frac{k}{m}\Big)\circ f^{(i,j)}\Big)\Big)\Big)\ \oplus\ O_{s(n \bmod m)},$$

are given by

$$\lambda_\ell\Big(T_{\lfloor n/m\rfloor}\Big(\Re\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a^{(i,j)}\Big(\frac{k}{m}\Big)\circ f^{(i,j)}(\theta)\Big)\Big)\Big)\Big),\quad \ell=1,\dots,s\Big\lfloor\frac{n}{m}\Big\rfloor,\quad k=1,\dots,m,$$

plus further $s(n \bmod m)$ eigenvalues which are equal to 0. Therefore, by Theorem 2.3, since $\Re(\sum_{i=1}^{p}\prod_{j=1}^{q_i}(a^{(i,j)}(\frac{k}{m})\circ f^{(i,j)}))\in L^\infty([-\pi,\pi],s)$, following the same derivation as in (27) we obtain, for any $F\in C_c(\mathbb{C})$,

$$\lim_{n\to\infty}\frac{1}{sn}\sum_{r=1}^{sn}F\Big(\lambda_r\Big(\Re\Big(\bigoplus_{k=1}^{m}T_{\lfloor n/m\rfloor}\Big(\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a^{(i,j)}\Big(\frac{k}{m}\Big)\circ f^{(i,j)}\Big)\Big)\Big)\Big)\Big)\Big)$$
$$=\frac{1}{2\pi}\int_0^1\int_{-\pi}^{\pi}\frac{1}{s}\sum_{\ell=1}^{s}F\Big(\lambda_\ell\Big(\Re\Big(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\Big(a_m^{(i,j)}(x)\circ f^{(i,j)}(\theta)\Big)\Big)\Big)\Big)\mathrm{d}\theta\mathrm{d}x.$$

This concludes the proof. □

Proposition 4.3 *Let $a:[0,1]\to\mathbb{C}^{s\times s}$ be Riemann-integrable and let $f\in L^1([-\pi,\pi],s)$. Then, for every $m\in\mathbb{N}$,*

$$\{LT_n^m(a,f)\}_n\sim_\sigma a_m(x)\circ f(\theta),$$

where

$$a_m(x)=\sum_{k=1}^{m}a\Big(\frac{k}{m}\Big)\chi_{\left[\frac{k-1}{m},\frac{k}{m}\right)}(x).$$

Proof Fix $m\in\mathbb{N}$ and take any sequence $\{f_h\}_h\subset L^\infty([-\pi,\pi],s)$ such that $f_h\to f$ a.e. and in $L^1([-\pi,\pi],s)$. We have:

- $\{LT_n^m(a,f_h)\}_n\xrightarrow{\text{a.c.s.}}\{LT_n^m(a,f)\}_n$ by Remark 4.1;
- $\{LT_n^m(a,f_h)\}_n\sim_\sigma a_m(x)\circ f_h(\theta)$ by Theorem 4.1;
- $a_m(x)\circ f_h(\theta)\to a_m(x)\circ f(\theta)$ a.e. (and hence also in measure).

We conclude that $\{LT_n^m(a,f)\}_n\sim_\sigma a_m(x)\circ f(\theta)$ by Theorem 3.1. □

4.2 Definition of Block LT Sequences

The notion of block LT sequences is formalized in the next definition.

Definition 4.2 (Block Locally Toeplitz Sequence) Let $\{A_n\}_n$ be a matrix-sequence, let $a : [0, 1] \to \mathbb{C}^{s\times s}$ be Riemann-integrable, and let $f \in L^1([-\pi, \pi], s)$. We say that $\{A_n\}_n$ is a block locally Toeplitz (LT) sequence with symbol $a(x)\circ f(\theta)$, and we write $\{A_n\}_n \sim_{\text{LT}} a(x) \circ f(\theta)$, if $\{LT_n^m(a, f)\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$.

In what follows, whenever we write a relation such as $\{A_n\}_n \sim_{\text{LT}} a(x) \circ f(\theta)$, it is understood that a and f are as in Definition 4.2, i.e., $a : [0, 1] \to \mathbb{C}^{s\times s}$ is Riemann-integrable and $f \in L^1([-\pi, \pi], s)$. Note that Definition 4.2 reduces to the definition of (scalar) LT sequences [7, Definition 7.2] if $s = 1$. LT sequences are then special cases of block LT sequences.

Remark 4.2 The results in [19, Section 3.3] could have led to believe that block LT sequences should be defined in reference to the classical matrix product $a(x)f(\theta)$ instead of the Hadamard product $a(x) \circ f(\theta)$. However, the experience coming from the applications has rejected such a solution as the matrices arising from the discretization of systems of DEs usually invoke the Hadamard product rather than the classical matrix product; see also Sect. 1. In this sense, Definition 4.2 may be seen as a correction to [19, Section 3.3].

Remark 4.3 If $\{A_n\}_n \sim_{\text{LT}} a(x) \circ f(\theta)$ then $\{A_n^*\}_n \sim_{\text{LT}} a(x)^* \circ f(\theta)^* = (a(x) \circ f(\theta))^*$ and $\{\alpha A_n\}_n \sim_{\text{LT}} \alpha\, a(x) \circ f(\theta) = a(x) \circ \alpha f(\theta)$ for all $\alpha \in \mathbb{C}$. This follows immediately from Definition 4.2, the properties of the block LT operator (see (20)–(22)), and Theorem 3.3.

4.3 Zero-Distributed Sequences, Sequences of Block Diagonal Sampling Matrices, and Sequences of Block Toeplitz Matrices

In this section now provide three fundamental examples of block LT sequences: zero-distributed sequences, sequences of block diagonal sampling matrices, and sequences of block Toeplitz matrices.

4.3.1 Zero-Distributed Sequences

We show that any zero-distributed sequence is a block LT sequence with symbol O_s.

Theorem 4.3 *Let $\{Z_n\}_n$ be a matrix-sequence. The following are equivalent.*

1. $\{Z_n\}_n \sim_\sigma 0$.
2. $\{O_{sn}\}_n \xrightarrow{\text{a.c.s.}} \{Z_n\}_n$.
3. $\{Z_n\}_n \sim_{\text{LT}} O_s$.

Proof ($1 \implies 2$) By Theorem 2.1, $Z_n = R_n + N_n$ with $\lim_{n\to\infty}(\text{rank}(R_n)/n) = \lim_{n\to\infty} \|N_n\| = 0$. Hence, the convergence $\{O_{sn}\}_n \xrightarrow{\text{a.c.s.}} \{Z_n\}_n$ follows immediately from Definition 3.1 (take $R_{n,m} = R_n$, $N_{n,m} = N_n$, $c(m)$ and $\omega(m)$ any two positive functions of m that converge to 0 as $m \to \infty$, and n_m any integer such that $\text{rank}(R_n)/n \le c(m)$ and $\|N_n\| \le \omega(m)$ for $n \ge n_m$).

($2 \implies 1$) Since $\{O_{sn}\}_n \xrightarrow{\text{a.c.s.}} \{Z_n\}_n$ and, moreover, $\{O_{sn}\}_n \sim_\sigma 0$, the relation $\{Z_n\}_n \sim_\sigma 0$ follows from Theorem 3.1.

($2 \iff 3$) This equivalence follows from Definition 4.2 and the observation that $LT_n^m(O_s, O_s) = O_{sn}$ and $O_s \circ O_s = O_s$. □

4.3.2 Sequences of Block Diagonal Sampling Matrices

For $n \in \mathbb{N}$ and $a : [0,1] \to \mathbb{C}^{s\times s}$, we define the block diagonal sampling matrix $D_n(a)$ as the following diagonal matrix of size $sn \times sn$:

$$D_n(a) = \underset{i=1,\ldots,n}{\text{diag}}\, a\Big(\frac{i}{n}\Big) = \bigoplus_{i=1}^{n} a\Big(\frac{i}{n}\Big).$$

We are going to see in Theorem 4.4 that $\{D_n(a)\}_n \sim_{\text{LT}} a(x) \circ \mathbf{1}_s$ whenever a is Riemann-integrable. To prove Theorem 4.4 we shall need the following lemmas; cf. [7, Lemmas 5.6 and 7.1].

Lemma 4.1 *Let C be an $\ell \times \ell$ matrix and suppose that*

$$\|C\|_p^p \le \epsilon \ell'$$

where $p \in [1,\infty)$, $\epsilon \ge 0$, and $\ell' \le \ell$. Then we can write C in the form

$$C = R + N, \quad \text{rank}(R) \le \epsilon^{\frac{1}{p+1}} \ell', \quad \|N\| \le \epsilon^{\frac{1}{p+1}}.$$

Lemma 4.2 *For every $m \in \mathbb{N}$ let $\{x(m,k)\}_k$ be a sequence of numbers such that $x(m,k) \to x(m)$ as $k \to \infty$ and $x(m) \to 0$ as $m \to \infty$. Then, there exists a sequence $\{m(k)\}_k \subseteq \mathbb{N}$ such that $m(k) \to \infty$ and $x(m(k),k) \to 0$.*

Theorem 4.4 *Suppose that $a : [0,1] \to \mathbb{C}^{s\times s}$ is Riemann-integrable, then $\{D_n(a)\}_n \sim_{\text{LT}} a(x) \circ \mathbf{1}_s$.*

Proof The proof consists of two steps. We first show that the thesis holds if a is continuous. Then, by using an approximation argument, we show that it holds even in the case where a is only supposed to be Riemann-integrable.

Step 1. We prove that if $a = [a_{\alpha\beta}]_{\alpha,\beta=1}^{s} : [0,1] \to \mathbb{C}^{s\times s}$ is continuous and $\omega_{a_{\alpha\beta}}$ is the modulus of continuity of $a_{\alpha\beta}$ then, for every $m \in \mathbb{N}$,

$$\begin{aligned} &D_n(a) = LT_n^m(a, \mathbf{1}_s) + R_{n,m} + N_{n,m}, \\ &\operatorname{rank}(R_{n,m}) \le sm, \quad \|N_{n,m}\| \le \sum_{\alpha,\beta}^{s} \omega_{a_{\alpha\beta}}\Big(\frac{1}{m} + \frac{m}{n}\Big). \end{aligned} \tag{29}$$

Since $\omega_{a_{\alpha\beta}}(\delta) \to 0$ as $\delta \to 0$ for all α, β, the convergence $\{LT_n^m(a, \mathbf{1}_s)\}_n \xrightarrow{\text{a.c.s.}} \{D_n(a)\}_n$ (and hence the relation $\{D_n(a)\}_n \sim_{\text{LT}} a(x) \circ \mathbf{1}_s$) follows immediately from Definition 3.1 (take $n_m = m^2$, $c(m) = s/m$, $\omega(m) = \sum_{\alpha,\beta=1}^{s} \omega_{a_{\alpha\beta}}(2/m)$).

The matrix $LT_n^m(a, \mathbf{1}_s)$ is the $sn \times sn$ block diagonal matrix given by

$$\begin{aligned} LT_n^m(a, \mathbf{1}_s) &= \bigoplus_{k=1}^{m} T_{\lfloor n/m \rfloor}\Big(a\Big(\frac{k}{m}\Big) \circ \mathbf{1}_s\Big) \ \oplus \ O_{s(n \bmod m)} \\ &= \bigoplus_{k=1}^{m} \Bigg(\bigoplus_{j=1}^{\lfloor n/m \rfloor} a\Big(\frac{k}{m}\Big)\Bigg) \ \oplus \ O_{s(n \bmod m)}. \end{aligned}$$

For $i = 1, \dots, m\lfloor n/m \rfloor$, let $k = k(i)$ be the index in $\{1, \dots, m\}$ such that

$$(k-1)\lfloor n/m \rfloor + 1 \le i \le k\lfloor n/m \rfloor.$$

In other words, k is the index such that the ith diagonal block of $LT_n^m(a, \mathbf{1}_s)$ is given by $(LT_n^m(a, \mathbf{1}_s))_{ii} = a(k/m)$. Using (3)–(4) and taking into account that the ith diagonal block of $D_n(a)$ is given by $(D_n(a))_{ii} = a(i/n)$, for every $i = 1, \dots, m\lfloor n/m \rfloor$ we obtain

$$\begin{aligned} &\big\|(LT_n^m(a, \mathbf{1}_s))_{ii} - (D_n(a))_{ii}\big\| = \Big\|a\Big(\frac{k}{m}\Big) - a\Big(\frac{i}{n}\Big)\Big\| \\ &\le \sum_{\alpha,\beta=1}^{s} \Big|a_{\alpha\beta}\Big(\frac{k}{m}\Big) - a_{\alpha\beta}\Big(\frac{i}{n}\Big)\Big| \le \sum_{\alpha,\beta=1}^{s} \omega_{a_{\alpha\beta}}\Big(\frac{1}{m} + \frac{m}{n}\Big), \end{aligned}$$

where the last inequality follows from the fact that

$$\Big|\frac{k}{m} - \frac{i}{n}\Big| \le \frac{k}{m} - \frac{(k-1)\lfloor n/m \rfloor}{n} \le \frac{k}{m} - \frac{(k-1)(n/m-1)}{n} = \frac{1}{m} + \frac{k-1}{n} \le \frac{1}{m} + \frac{m}{n}.$$

Define the following $sn \times sn$ block diagonal matrices:

$$\tilde{D}_{n,m}(a) = \bigoplus_{i=1}^{m\lfloor n/m \rfloor} a\left(\frac{i}{n}\right) \oplus O_{s(n \bmod m)},$$

$$\hat{D}_{n,m}(a) = O_{sm\lfloor n/m \rfloor} \oplus \bigoplus_{i=m\lfloor n/m \rfloor+1}^{n} a\left(\frac{i}{n}\right).$$

Then,

$$D_n(a) - LT_n^m(a, \mathbf{1}_s) = \hat{D}_{n,m}(a) + \tilde{D}_{n,m}(a) - LT_n^m(a, \mathbf{1}_s) = R_{n,m} + N_{n,m},$$

where $R_{n,m} = \hat{D}_{n,m}(a)$ and $N_{n,m} = \tilde{D}_{n,m}(a) - LT_n^m(a, \mathbf{1}_s)$ satisfy

$$\operatorname{rank}(R_{n,m}) \le s(n \bmod m) < sm,$$

$$\|N_{n,m}\| = \max_{i=1,\dots,m\lfloor n/m \rfloor} \|(LT_n^m(a, \mathbf{1}_s))_{ii} - (D_n(a))_{ii}\| \le \sum_{\alpha,\beta=1}^{s} \omega_{a_{\alpha\beta}}\left(\frac{1}{m} + \frac{m}{n}\right).$$

This completes the proof of (29).

Step 2. Let $a : [0,1] \to \mathbb{C}^{s\times s}$ be any Riemann-integrable function. Take any sequence of continuous functions $a_m : [0,1] \to \mathbb{C}^{s\times s}$ such that $a_m \to a$ in $L^1([0,1], s)$. By Step 1, $\{D_n(a_m)\}_n \sim_{\mathrm{LT}} a_m \circ \mathbf{1}_s$. Hence, $\{LT_n^h(a_m, \mathbf{1}_s)\}_n \xrightarrow{\text{a.c.s.}} \{D_n(a_m)\}_n$, i.e., for every m, h there is $n_{m,h}$ such that, for $n \ge n_{m,h}$,

$$D_n(a_m) = LT_n^h(a_m, \mathbf{1}_s) + R_{n,m,h} + N_{n,m,h},$$
$$\operatorname{rank}(R_{n,m,h}) \le c(m,h)n, \quad \|N_{n,m,h}\| \le \omega(m,h),$$

where $\lim_{h\to\infty} c(m,h) = \lim_{h\to\infty} \omega(m,h) = 0$. Moreover, $\{D_n(a_m)\}_n \xrightarrow{\text{a.c.s.}} \{D_n(a)\}_n$. Indeed, by (4),

$$\|D_n(a) - D_n(a_m)\|_1 = \sum_{j=1}^{n} \left\| a\left(\frac{j}{n}\right) - a_m\left(\frac{j}{n}\right) \right\|_1 = \sum_{j=1}^{n} \left\| (a - a_m)\left(\frac{j}{n}\right) \right\|_1$$
$$\le \sum_{j=1}^{n} \sum_{\alpha,\beta=1}^{s} \left| (a - a_m)_{\alpha\beta}\left(\frac{j}{n}\right) \right| = \epsilon(m,n)n, \tag{30}$$

where

$$\epsilon(m,n) = \frac{1}{n} \sum_{j=1}^{n} \sum_{\alpha,\beta=1}^{s} \left| (a - a_m)_{\alpha\beta}\left(\frac{j}{n}\right) \right|.$$

By the Riemann-integrability of $\sum_{\alpha,\beta=1}^{s} |(a-a_m)_{\alpha\beta}|$, which is a consequence of the Riemann-integrability of $a-a_m$, and by the fact that $a_m \to a$ in $L^1([0,1],s)$, the quantity $\epsilon(m,n)$ satisfies

$$\begin{aligned}\lim_{m\to\infty}\lim_{n\to\infty}\epsilon(m,n) &= \lim_{m\to\infty}\int_0^1 \sum_{\alpha,\beta=1}^{s} \big|(a-a_m)_{\alpha\beta}(x)\big|\,\mathrm{d}x \\ &= \lim_{m\to\infty}\sum_{\alpha,\beta=1}^{s}\|(a-a_m)_{\alpha\beta}\|_{L^1} = 0.\end{aligned}$$

By Theorem 3.4, this implies that $\{D_n(a_m)\}_n \overset{\text{a.c.s.}}{\longrightarrow} \{D_n(a)\}_n$. Thus, for every m there exists n_m such that, for $n \geq n_m$,

$$\begin{aligned}D_n(a) &= D_n(a_m) + R_{n,m} + N_{n,m},\\ \operatorname{rank}(R_{n,m}) &\leq c(m)n, \quad \|N_{n,m}\| \leq \omega(m),\end{aligned}$$

where $\lim_{m\to\infty} c(m) = \lim_{m\to\infty}\omega(m) = 0$. It follows that, for every m,h and every $n \geq \max(n_m, n_{m,h})$,

$$\begin{aligned}D_n(a) &= LT_n^h(a,\mathbf{1}_s) + \big[LT_n^h(a_m,\mathbf{1}_s) - LT_n^h(a,\mathbf{1}_s)\big]\\ &\quad + (R_{n,m}+R_{n,m,h}) + (N_{n,m}+N_{n,m,h}),\\ \operatorname{rank}(R_{n,m}+R_{n,m,h}) &\leq (c(m)+c(m,h))n,\\ \|N_{n,m}+N_{n,m,h}\| &\leq \omega(m)+\omega(m,h),\\ \|LT_n^h(a_m,\mathbf{1}_s) - LT_n^h(a,\mathbf{1}_s)\|_1 &\leq \frac{n}{h}\sum_{j=1}^{h}\left\|a\Big(\frac{j}{h}\Big) - a_m\Big(\frac{j}{h}\Big)\right\|_1 \leq \epsilon(m,h)n,\end{aligned}$$

where the last inequality is proved as in (30). Let $\{m(h)\}_h$ be a sequence such that $m(h)\to\infty$ and

$$\lim_{h\to\infty}\epsilon(m(h),h) = \lim_{h\to\infty} c(m(h),h) = \lim_{h\to\infty}\omega(m(h),h) = 0.$$

Note that such a sequence exists by Lemma 4.2 (apply the lemma with $x(m,h) = \epsilon(m,h)+c(m,h)+\omega(m,h)$). Then, for every h and every $n \geq \max(n_{m(h)}, n_{m(h),h})$,

$$\begin{aligned}D_n(a) &= LT_n^h(a,\mathbf{1}_s) + \big[LT_n^h(a_{m(h)},\mathbf{1}_s) - LT_n^h(a,\mathbf{1}_s)\big]\\ &\quad + (R_{n,m(h)}+R_{n,m(h),h}) + (N_{n,m(h)}+N_{n,m(h),h}),\\ \operatorname{rank}(R_{n,m(h)}+R_{n,m(h),h}) &\leq (c(m(h))+c(m(h),h))n,\end{aligned}$$

$$\|N_{n,m(h)} + N_{n,m(h),h}\| \le \omega(m(h)) + \omega(m(h), h),$$

$$\|LT_n^h(a_{m(h)}, \mathbf{1}_s) - LT_n^h(a, \mathbf{1}_s)\|_1 \le \epsilon(m(h), h)n.$$

The application of Lemma 4.1 allows one to decompose $LT_n^h(a_{m(h)}, \mathbf{1}_s) - LT_n^h(a, \mathbf{1}_s)$ as the sum of a small-rank term $\hat{R}_{n,h}$, with rank bounded by $\sqrt{\epsilon(m(h), h)}\, n$, plus a small-norm term $\hat{N}_{n,h}$, with norm bounded by $\sqrt{\epsilon(m(h), h)}$. This shows that $\{LT_n^h(a, \mathbf{1}_s)\}_n \xrightarrow{\text{a.c.s.}} \{D_n(a)\}_n$, hence $\{D_n(a)\}_n \sim_{\text{LT}} a(x) \circ \mathbf{1}_s$. □

4.3.3 Sequences of Block Toeplitz Matrices

Theorem 4.5 *Suppose that $f \in L^1([-\pi, \pi], s)$, then $\{T_n(f)\}_n \sim_{\text{LT}} \mathbf{1}_s \circ f(\theta)$.*

Proof The proof consists of two steps. We first show that the thesis holds if f is a matrix-valued trigonometric polynomial. Then, by using an approximation argument, we prove the theorem under the sole assumption that $f \in L^1([-\pi, \pi], s)$.

Step 1. We show that if f is a matrix-valued trigonometric polynomial of degree d then

$$T_n(f) = LT_n^m(\mathbf{1}_s, f) + R_{n,m}, \quad \text{rank}(R_{n,m}) \le s(2d + 1)m. \tag{31}$$

Once this is done, the convergence $\{LT_n^m(\mathbf{1}_s, f)\}_n \xrightarrow{\text{a.c.s.}} \{T_n(f)\}_n$ (and hence the relation $\{T_n(f)\}_n \sim_{\text{LT}} \mathbf{1}_s \circ f(\theta)$) follows immediately from Definition 3.1 (take $n_m = m^2$, $c(m) = s(2d + 1)/m$, $\omega(m) = 0$).

Since f has degree d, we can write $f(\theta) = \sum_{j=-d}^{d} c_j \, \mathrm{e}^{\mathrm{i}j\theta}$. Moreover, we have

$$LT_n^m(\mathbf{1}_s, f) = \bigoplus_{k=1}^{m} T_{\lfloor n/m \rfloor}(\mathbf{1}_s \circ f) \oplus O_{s(n \bmod m)} = \bigoplus_{k=1}^{m} T_{\lfloor n/m \rfloor}(f) \oplus O_{s(n \bmod m)}.$$

A direct comparison between the matrix $T_n(f)$ and the matrix $LT_n^m(\mathbf{1}_s, f)$ shows that if $n/m \ge 2d + 1$ then the number of nonzero rows of the difference $T_n(f) - LT_n^m(\mathbf{1}_s, f)$ is at most $s(2dm - d + (n \bmod m))$. Hence, if $n/m \ge 2d + 1$,

$$T_n(f) = LT_n^m(\mathbf{1}_s, f) + R_{n,m}, \quad \text{rank}(R_{n,m}) \le s(2dm - d + (n \bmod m)) \le s(2d + 1)m.$$

This completes the proof of (31) for $n/m \ge 2d + 1$, but it is clear that (31) holds even if $n/m < 2d + 1$, because in this case $s(2d + 1)m$ is greater than the matrix size sn.

Step 2. Let $f \in L^1([-\pi, \pi], s)$. Since the set of trigonometric polynomials is dense in $L^1([-\pi, \pi])$ (see, e.g., [7, Lemma 2.2]), there is a sequence of matrix-valued trigonometric polynomials $f_m : [-\pi, \pi] \to \mathbb{C}^{s \times s}$ such that $f_m \to f$ in

$L^1([-\pi,\pi],s)$. By Step 1, $\{T_n(f_m)\}_n \sim_{\text{LT}} \mathbf{1}_s \circ f_m(\theta)$. Hence, $\{LT_n^h(\mathbf{1}_s, f_m)\}_n \xrightarrow{\text{a.c.s.}} \{T_n(f_m)\}_n$, i.e., for every m, h there is $n_{m,h}$ such that, for $n \geq n_{m,h}$,

$$T_n(f_m) = LT_n^h(\mathbf{1}_s, f_m) + R_{n,m,h} + N_{n,m,h},$$
$$\operatorname{rank}(R_{n,m,h}) \leq c(m,h)n, \quad \|N_{n,m,h}\| \leq \omega(m,h),$$

where $\lim_{h\to\infty} c(m,h) = \lim_{h\to\infty} \omega(m,h) = 0$. Moreover, by Theorem 2.4,

$$\|T_n(f) - T_n(f_m)\|_1 = \|T_n(f - f_m)\|_1 \leq n\|f - f_m\|_{L^1}$$

and so $\{T_n(f_m)\}_n \xrightarrow{\text{a.c.s.}} \{T_n(f)\}_n$ by Theorem 3.4. Thus, for every m there exists n_m such that, for $n \geq n_m$,

$$T_n(f) = T_n(f_m) + R_{n,m} + N_{n,m},$$
$$\operatorname{rank}(R_{n,m}) \leq c(m)n, \quad \|N_{n,m}\| \leq \omega(m),$$

where $\lim_{m\to\infty} c(m) = \lim_{m\to\infty} \omega(m) = 0$. It follows that, for every m, h and every $n \geq \max(n_m, n_{m,h})$,

$$\begin{aligned} T_n(f) &= LT_n^h(\mathbf{1}_s, f) + \big[LT_n^h(\mathbf{1}_s, f_m) - LT_n^h(\mathbf{1}_s, f)\big] \\ &\quad + (R_{n,m} + R_{n,m,h}) + (N_{n,m} + N_{n,m,h}), \\ \operatorname{rank}(R_{n,m} + R_{n,m,h}) &\leq (c(m) + c(m,h))n, \\ \|N_{n,m} + N_{n,m,h}\| &\leq \omega(m) + \omega(m,h), \\ \|LT_n^h(\mathbf{1}_s, f_m) - LT_n^h(\mathbf{1}_s, f)\|_1 &= \|LT_n^h(\mathbf{1}_s, f_m - f)\|_1 \leq n\|f_m - f\|_{L^1}, \end{aligned}$$

where the last inequality follows from (6) and Theorem 2.4. Let $\{m(h)\}_h$ be a sequence such that $m(h) \to \infty$ and

$$\lim_{h\to\infty} c(m(h),h) = \lim_{h\to\infty} \omega(m(h),h) = 0.$$

Note that such a sequence exists by Lemma 4.2 (apply the lemma with $x(m,h) = c(m,h) + \omega(m,h)$). Then, for every h and every $n \geq \max(n_{m(h)}, n_{m(h),h})$,

$$\begin{aligned} T_n(f) &= LT_n^h(\mathbf{1}_s, f) + \big[LT_n^h(\mathbf{1}_s, f_{m(h)}) - LT_n^h(\mathbf{1}_s, f)\big] \\ &\quad + (R_{n,m(h)} + R_{n,m(h),h}) + (N_{n,m(h)} + N_{n,m(h),h}), \\ \operatorname{rank}(R_{n,m(h)} + R_{n,m(h),h}) &\leq (c(m(h)) + c(m(h),h))n, \\ \|N_{n,m(h)} + N_{n,m(h),h}\| &\leq \omega(m(h)) + \omega(m(h),h), \\ \|LT_n^h(\mathbf{1}_s, f_{m(h)}) - LT_n^h(\mathbf{1}_s, f)\|_1 &\leq n\|f_{m(h)} - f\|_{L^1}. \end{aligned}$$

The application of Lemma 4.1 allows one to decompose $LT_n^h(\mathbf{1}_s, f_{m(h)}) - LT_n^h(\mathbf{1}_s, f)$ as the sum of a small-rank term $\hat{R}_{n,h}$, with rank bounded by $\sqrt{\|f_{m(h)} - f\|_{L^1}}\, n$, plus a small-norm term $\hat{N}_{n,h}$, with norm bounded by $\sqrt{\|f_{m(h)} - f\|_{L^1}}$. This shows that $\{LT_n^h(\mathbf{1}_s, f)\}_n \xrightarrow{\text{a.c.s.}} \{T_n(f)\}_n$, hence $\{T_n(f)\}_n \sim_{\text{LT}} \mathbf{1}_s \circ f(\theta)$. □

4.4 Singular Value and Spectral Distribution of a Sum of Products of Block LT Sequences

The main results of this section are Theorems 4.6 and 4.7. In order to prove them, we shall need the following lemma, which is a special case of Theorem 4.6.

Lemma 4.3 *If* $\{A_n\}_n \sim_{\text{LT}} a(x) \circ f(\theta)$ *then* $\{A_n\}_n \sim_\sigma a(x) \circ f(\theta)$ *and* $\{A_n\}_n$ *is s.u.*

Proof We have:

- $\{LT_n^m(a, f)\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ by definition of block LT sequences;
- $\{LT_n^m(a, f)\}_n \sim_\sigma a_m(x) \circ f(\theta)$ with $a_m(x) = \sum_{k=1}^m a(\frac{k}{m}) \chi_{\left[\frac{k-1}{m}, \frac{k}{m}\right)}(x)$ by Proposition 4.3;
- $a_m(x) \circ f(\theta) \to a(x) \circ f(\theta)$ a.e. (and hence also in measure) by [7, Lemma 2.9], because $a(x)$ is Riemann-integrable.

We conclude that $\{A_n\}_n \sim_\sigma a(x) \circ f(\theta)$ by Theorem 3.1, and so $\{A_n\}_n$ is s.u. by Proposition 2.3. □

Theorem 4.6 *If* $\{A_n^{(i,j)}\}_n \sim_{\text{LT}} a^{(i,j)}(x) \circ f^{(i,j)}(\theta)$ *for* $i = 1, \ldots, p$ *and* $j = 1, \ldots, q_i$ *then*

$$\left\{\sum_{i=1}^{p} \prod_{j=1}^{q_i} A_n^{(i,j)}\right\}_n \sim_\sigma \sum_{i=1}^{p} \prod_{j=1}^{q_i} \left(a^{(i,j)}(x) \circ f^{(i,j)}(\theta)\right).$$

Proof Let

$$A_n = \sum_{i=1}^{p} \prod_{j=1}^{q_i} A_n^{(i,j)}, \quad A_{n,m} = \bigoplus_{k=1}^{m} T_{\lfloor n/m \rfloor}\left(\sum_{i=1}^{p} \prod_{j=1}^{q_i} \left(a^{(i,j)}\left(\frac{k}{m}\right) \circ f^{(i,j)}\right)\right) \oplus O_{s(n \bmod m)},$$

$$\kappa(x, \theta) = \sum_{i=1}^{p} \prod_{j=1}^{q_i} \left(a^{(i,j)}(x) \circ f^{(i,j)}(\theta)\right), \quad \kappa_m(x, \theta) = \sum_{i=1}^{p} \prod_{j=1}^{q_i} \left(a_m^{(i,j)}(x) \circ f^{(i,j)}(\theta)\right),$$

where

$$a_m^{(i,j)}(x) = \sum_{k=1}^{m} a^{(i,j)}\Big(\frac{k}{m}\Big)\chi_{\left[\frac{k-1}{m},\frac{k}{m}\right)}(x).$$

Since $\{LT_n^m(a^{(i,j)}, f^{(i,j)})\}_n \xrightarrow{\text{a.c.s.}} \{A_n^{(i,j)}\}_n$ by definition of block LT sequences, Lemma 4.3 and Remark 3.1 imply that $\{\sum_{i=1}^{p}\prod_{j=1}^{q_i} LT_n^m(a^{(i,j)}, f^{(i,j)})\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$. Thus, we have:

- $\{A_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ by Proposition 4.2;
- $\{A_{n,m}\}_n \sim_\sigma \kappa_m(x,\theta)$ by Theorem 4.1;
- $\kappa_m(x,\theta) \to \kappa(x,\theta)$ a.e. (and hence also in measure) by [7, Lemma 2.9], because each $a^{(i,j)}(x)$ is Riemann-integrable.

We conclude that $\{A_n\}_n \sim_\sigma \kappa(x,\theta)$ by Theorem 3.1. □

Theorem 4.7 *If* $\{A_n^{(i,j)}\}_n \sim_{\text{LT}} a^{(i,j)}(x) \circ f^{(i,j)}(\theta)$ *for* $i = 1,\dots,p$ *and* $j = 1,\dots,q_i$ *then*

$$\left\{\Re\left(\sum_{i=1}^{p}\prod_{j=1}^{q_i} A_n^{(i,j)}\right)\right\}_n \sim_\lambda \Re\left(\sum_{i=1}^{p}\prod_{j=1}^{q_i}\left(a^{(i,j)}(x)\circ f^{(i,j)}(\theta)\right)\right).$$

Proof The proof is essentially the same as the proof of Theorem 4.6. Define the matrices A_n, $A_{n,m}$ and the functions $\kappa(x,\theta)$, $\kappa_m(x,\theta)$ as in the proof of Theorem 4.6. Since $\{LT_n^m(a^{(i,j)}, f^{(i,j)})\}_n \xrightarrow{\text{a.c.s.}} \{A_n^{(i,j)}\}_n$ by definition of block LT sequences, Lemma 4.3 and Remark 3.1 imply that $\{\sum_{i=1}^{p}\prod_{j=1}^{q_i} LT_n^m(a^{(i,j)}, f^{(i,j)})\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$. Hence, $\{\Re(\sum_{i=1}^{p}\prod_{j=1}^{q_i} LT_n^m(a^{(i,j)}, f^{(i,j)}))\}_n \xrightarrow{\text{a.c.s.}} \{\Re(\sum_{i=1}^{p}\prod_{j=1}^{q_i} A_n^{(i,j)})\}_n$ by Theorem 3.3. Thus, we have:

- $\{\Re(A_{n,m})\}_n \xrightarrow{\text{a.c.s.}} \{\Re(A_n)\}_n$ by Proposition 4.2 and Theorem 3.3;
- $\{\Re(A_{n,m})\}_n \sim_\lambda \Re(\kappa_m(x,\theta))$ by Theorem 4.2;
- $\Re(\kappa_m(x,\theta)) \to \Re(\kappa(x,\theta))$ a.e. (and hence also in measure) by [7, Lemma 2.9], because each $a^{(i,j)}(x)$ is Riemann-integrable.

We conclude that $\{\Re(A_n)\}_n \sim_\lambda \Re(\kappa(x,\theta))$ by Theorem 3.2. □

5 Concluding Remarks

In this paper we have developed the theory of block LT sequences, which, as the reader may have noted, is conceptually very similar to the theory of LT sequences [7, Chapter 7]. The similarity can be inferred not only from the numerous citations to [7] disseminated in the paper but also from the analogies between the

proofs presented herein and the proofs of analogous results presented in [7] (e.g., Proposition 2.3 vs. [7, Proposition 5.4], Theorems 3.1 and 3.2 vs. [7, Corollaries 5.1 and 5.2], Theorems 4.3, 4.4, 4.5 vs. [7, Theorems 7.3, 7.4, 7.5]). It is clear, however, that the block case has involved several technicalities that are not present in the scalar case (e.g., Lemma 2.3 vs. [7, Lemma 2.5], Proposition 4.1 vs. [7, Eq. (7.13)]). Nevertheless, technicalities were not the main difficulty. The main difficulty was to understand the 'right' generalization to the block case of the definitions of LT operator and LT sequences. In particular, the necessity of replacing the classical product $a(x)f(\theta)$ with the Hadamard product $a(x) \circ f(\theta)$ became clear only after considering specific application examples (see Remark 4.2 and Sect. 1). Once the right generalization was found, aside from technicalities, the theory of block LT sequences has been worked out (almost) painlessly by adapting the results/arguments already known within the framework of the theory of LT sequences.

We conclude by recommending that the reader go through the complementary paper [12] so as to gain a complete picture of the theory of block GLT sequences. The related applications can be found in [11].

Acknowledgements Carlo Garoni is a Marie-Curie fellow of the Italian INdAM under grant agreement PCOFUND-GA-2012-600198. The work of the authors has been supported by the INdAM GNCS (Gruppo Nazionale per il Calcolo Scientifico). The authors wish to thank Giovanni Barbarino for useful discussions.

References

1. Avram, F.: On bilinear forms in Gaussian random variables and Toeplitz matrices. Probab. Theory Relat. Fields **79**, 37–45 (1988)
2. Barbarino, G.: Equivalence between GLT sequences and measurable functions. Linear Algebra Appl. **529**, 397–412 (2017)
3. Bhatia, R.: Matrix Analysis. Springer, New York (1997)
4. Böttcher, A., Silbermann, B.: Introduction to Large Truncated Toeplitz Matrices. Springer, New York (1999)
5. Böttcher, A., Silbermann B.: Analysis of Toeplitz Operators, 2nd edn. Springer, Berlin (2006)
6. Donatelli, M., Garoni, C., Mazza, M., Serra-Capizzano, S., Sesana, D.: Spectral behavior of preconditioned non-Hermitian multilevel block Toeplitz matrices with matrix-valued symbol. Appl. Math. Comput. **245**, 158–173 (2014)
7. Garoni, C., Serra-Capizzano, S.: Generalized Locally Toeplitz Sequences: Theory and Applications (Volume I). Springer, Cham (2017)
8. Garoni, C., Serra-Capizzano, S.: Generalized Locally Toeplitz Sequences: Theory and Applications (Volume II). Springer, Cham (2018)
9. Garoni, C., Serra-Capizzano, S., Sesana, D.: Spectral analysis and spectral symbol of d-variate $\mathbb{Q}_p$ Lagrangian FEM stiffness matrices. SIAM J. Matrix Anal. Appl. **36**, 1100–1128 (2015)
10. Garoni, C., Serra-Capizzano, S., Vassalos, P.: A general tool for determining the asymptotic spectral distribution of Hermitian matrix-sequences. Oper. Matrices **9**, 549–561 (2015)

11. Garoni, C., Mazza, M., Serra-Capizzano, S.: Block generalized locally Toeplitz sequences: from the theory to the applications. Axioms **7**, 49 (2018)
12. Garoni, C., Serra-Capizzano, S., Sesana, D.: Block generalized locally Toeplitz sequences: topological construction, spectral distribution results, and star-algebra structure. In: Bini, D.A., et al. (eds.) Structured Matrices in Numerical Linear Algebra. Springer INdAM Series, vol. 30, pp. 59–79. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-04088-8_3
13. Garoni, C., Speleers, H., Ekström, S.-E., Reali, A., Serra-Capizzano, S., Hughes, T.J.R.: Symbol-based analysis of finite element and isogeometric B-spline discretizations of eigenvalue problems: exposition and review. Arch. Comput. Meth. Eng. (in press). https://doi.org/10.1007/s11831-018-9295-y
14. Grenander, U., Szegő, G.: Toeplitz Forms and Their Applications, 2nd edn. AMS Chelsea Publishing, New York (1984)
15. Parter, S.V.: On the distribution of the singular values of Toeplitz matrices. Linear Algebra Appl. **80**, 115–130 (1986)
16. Serra-Capizzano, S.: Distribution results on the algebra generated by Toeplitz sequences: a finite dimensional approach. Linear Algebra Appl. **328**, 121–130 (2001)
17. Serra-Capizzano, S.: More inequalities and asymptotics for matrix valued linear positive operators: the noncommutative case. Oper. Theory Adv. Appl. **135**, 293–315 (2002)
18. Serra-Capizzano, S.: Generalized locally Toeplitz sequences: spectral analysis and applications to discretized partial differential equations. Linear Algebra Appl. **366**, 371–402 (2003)
19. Serra-Capizzano, S.: The GLT class as a generalized Fourier analysis and applications. Linear Algebra Appl. **419**, 180–233 (2006)
20. Serra-Capizzano, S., Tilli, P.: On unitarily invariant norms of matrix-valued linear positive operators. J. Inequal. Appl. **7**, 309–330 (2002)
21. Tilli, P.: A note on the spectral distribution of Toeplitz matrices. Linear Multilinear Algebra **45**, 147–159 (1998)
22. Tilli, P.: Locally Toeplitz sequences: spectral properties and applications. Linear Algebra Appl. **278**, 91–120 (1998)
23. Tyrtyshnikov, E.E.: A unifying approach to some old and new theorems on distribution and clustering. Linear Algebra Appl. **232**, 1–43 (1996)
24. Tyrtyshnikov, E.E., Zamarashkin, N.L.: Spectra of multilevel Toeplitz matrices: advanced theory via simple matrix relationships. Linear Algebra Appl. **270**, 15–27 (1998)
25. Zamarashkin, N.L., Tyrtyshnikov, E.E.: Distribution of eigenvalues and singular values of Toeplitz matrices under weakened conditions on the generating function. Sb. Math. **188**, 1191–1201 (1997)

Block Generalized Locally Toeplitz Sequences: Topological Construction, Spectral Distribution Results, and Star-Algebra Structure

Carlo Garoni, Stefano Serra-Capizzano, and Debora Sesana

Abstract The theory of generalized locally Toeplitz (GLT) sequences is a powerful apparatus for computing the asymptotic singular value and eigenvalue distribution of matrices A_n arising from virtually any kind of numerical discretization of differential equations (DEs). Indeed, when the discretization parameter n tends to infinity, these matrices A_n give rise to a sequence $\{A_n\}_n$, which often turns out to be a GLT sequence or one of its 'relatives', i.e., a block GLT sequence or a reduced GLT sequence. In particular, block GLT sequences are encountered in the discretization of systems of DEs as well as in the higher-order finite element or discontinuous Galerkin approximation of scalar DEs. Despite the applicative interest, a solid theory of block GLT sequences is still missing. The purpose of the present paper is to develop this theory in a systematic way.

Keywords Singular values and eigenvalues · Block generalized locally Toeplitz sequences · Block Toeplitz matrices · Discretization of differential equations

C. Garoni (✉)
University of Italian Switzerland, Institute of Computational Science, Lugano, Switzerland

University of Insubria, Department of Science and High Technology, Como, Italy
e-mail: carlo.garoni@usi.ch; carlo.garoni@uninsubria.it

S. Serra-Capizzano
University of Insubria, Department of Science and High Technology, Como, Italy

Uppsala University, Department of Information Technology, Division of Scientific Computing, Uppsala, Sweden
e-mail: stefano.serrac@uninsubria.it; stefano.serra@it.uu.se

D. Sesana
University of Insubria, Department of Science and High Technology, Como, Italy
e-mail: debora.sesana@uninsubria.it

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_3

1 Introduction

The theory of generalized locally Toeplitz (GLT) sequences stems from Tilli's work on locally Toeplitz (LT) sequences [42] and from the spectral theory of Toeplitz matrices [2, 10–13, 32, 33, 41, 43, 44, 46]. It was then carried forward by the first two authors in [24, 25, 39, 40], and it has been recently extended by Barbarino in [3]. This theory is a powerful apparatus for computing/analyzing the asymptotic spectral distribution of matrices arising from the numerical discretization of continuous problems, such as integral equations (IEs) and, especially, differential equations (DEs). The experience reveals that essentially any kind of numerical methods for the discretization of DEs gives rise to structured matrices A_n whose asymptotic spectral distribution, as the fineness parameter n goes to infinity, can be computed/analyzed through the theory of GLT sequences. We refer the reader to [24, Section 10.5], [25, Section 7.3], and [14, 39, 40] for applications of the theory of GLT sequences in the context of finite difference discretizations of DEs; to [24, Section 10.6], [25, Section 7.4], and [4, 14, 40] for the finite element case; to [6] for the finite volume case; to [24, Section 10.7], [25, Sections 7.5–7.7], and [17, 23, 27, 28, 35] for the case of isogeometric analysis discretizations, both in the collocation and Galerkin frameworks; and to [19] for a further recent application to fractional differential equations. We also refer the reader to [24, Section 10.4] and [1, 37] for a look at the GLT approach to deal with sequences of matrices coming from IE discretizations.

We have to say, however, that, despite the aforementioned progresses, the theory of GLT sequences is still incomplete. In particular, the so-called 'block' GLT sequences have only been introduced in [40, Section 3.3], without any claim to completeness or correctness. Papers pointing toward or referring to block GLT sequences are many (see, e.g., [5, 16, 18, 20, 22]), but none of them enters into the details of the theory. Yet, the topic is worthy of high consideration. Indeed, matrices with a block Toeplitz structure naturally arise in many areas of science such as Markov chains [8], subdivision algorithms [34], Riccati equations [9], in the study of the zeros of orthogonal polynomials with periodic coefficients [45], in the reconstruction of signals with missing data [15], and, above all, in the discretization of systems of constant-coefficient DEs [40, Section 3.3]. Recently, it has been discovered that matrices of this kind also arise in the discretization of scalar constant-coefficient DEs by higher-order finite element methods [26] or discontinuous Galerkin methods [5, 21]. More generally, they are encountered whenever we are in the presence of a finite element discretization in which the Galerkin approximation space consists of piecewise polynomial functions of degree p and smoothness C^k with $p-k > 1$; see [31]. In the case of nonconstant-coefficient DEs, block Toeplitz structures leave the place to locally block Toeplitz structures, that is, the progenitors of block GLT sequences. It is then clear that the computation of the asymptotic spectral distribution of DE discretization matrices with a block structure requires a solid theory of block GLT sequences, which is currently not available.

In this paper, based on the results about block LT sequences obtained in [30], we develop a systematic theory of block GLT sequences, in line with the theory of (scalar) GLT sequences already formalized in [24, 25]. A forthcoming paper [29] will take care of illustrating several applications of the theory developed herein. The paper is organized as follows. In Sect. 2 we collect all the necessary preliminaries. Section 3 focuses on the fundamental notion of approximating classes of sequences. In Sect. 4 we summarize the theory of block LT sequences developed in [30], and in Sect. 5 we address the theory of block GLT sequences. In Sect. 6 we provide a summary of the theory. Section 7 is devoted to final remarks about future lines of research and the connections between the theory of GLT sequences [24, Chapter 8] and the theory of block GLT sequences.

2 Mathematical Background

2.1 Notation and Terminology

- O_m and I_m denote, respectively, the $m \times m$ zero matrix and the $m \times m$ identity matrix.
- $\mathbf{1}_m$ denotes the $m \times m$ matrix whose entries are all equal to 1.
- The eigenvalues and the singular values of $X \in \mathbb{C}^{m\times m}$ are denoted by $\lambda_j(X)$, $j = 1, \ldots, m$, and $\sigma_j(X)$, $j = 1, \ldots, m$, respectively. The maximum and minimum singular values of X are also denoted by $\sigma_{\max}(X)$ and $\sigma_{\min}(X)$, respectively.
- If $X \in \mathbb{C}^{m\times m}$, we denote by $X^\dagger$ the Moore–Penrose pseudoinverse of X.
- Given $X \in \mathbb{C}^{m\times m}$ and $1 \le p \le \infty$, $\|X\|_p$ denotes the Schatten p-norm of X, which is defined as the p-norm of the vector $(\sigma_1(X), \ldots, \sigma_m(X))$; see [7]. The Schatten 1-norm is also called the trace-norm. The Schatten ∞-norm $\|X\|_\infty = \sigma_{\max}(X)$ is the classical 2-norm (or spectral norm) and will also be denoted by $\|X\|$.
- $\Re(X)$ is the real part of the (square) matrix X, i.e., $\Re(X) = \frac{X+X^*}{2}$, where X^* is the conjugate transpose of X.
- If $X, Y \in \mathbb{C}^{m\times \ell}$, the Hadamard (or entrywise) product of X and Y is the $m \times \ell$ matrix defined by $(X \circ Y)_{ij} = x_{ij} y_{ij}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, \ell$.
- $C_c(\mathbb{C})$ (resp., $C_c(\mathbb{R})$) is the space of complex-valued continuous functions defined on $\mathbb{C}$ (resp., $\mathbb{R}$) and with bounded support.
- χ_E is the characteristic (indicator) function of the set E.
- μ_k denotes the Lebesgue measure in $\mathbb{R}^k$. Throughout this paper, unless otherwise stated, all the terminology from measure theory (such as 'measurable set', 'measurable function', 'a.e.', etc.) is always referred to the Lebesgue measure.
- Let $D \subseteq \mathbb{R}^k$, let $r \ge 1$ and $1 \le p \le \infty$. A matrix-valued function $f : D \to \mathbb{C}^{r\times r}$ is said to be measurable (resp., continuous, bounded, in $L^p(D)$, in $C^\infty(D)$, etc.) if its components $f_{\alpha\beta} : D \to \mathbb{C}$, $\alpha, \beta = 1, \ldots, r$, are measurable (resp., continuous, bounded, in $L^p(D)$, in $C^\infty(D)$, etc.). The space of functions f :

$D \to \mathbb{C}^{r\times r}$ belonging to $L^p(D)$ will be denoted by $L^p(D,r)$ in order to stress the dependence on r.

- Let $f_m, f : D \subseteq \mathbb{R}^k \to \mathbb{C}^{r\times r}$ be measurable. We say that f_m converges to f in measure (resp., a.e., in $L^p(D)$, etc.) if $(f_m)_{\alpha\beta}$ converges to $f_{\alpha\beta}$ in measure (resp., a.e., in $L^p(D)$, etc.) for all $\alpha, \beta = 1, \ldots, r$.
- A function $a : [0,1] \to \mathbb{C}^{r\times r}$ is said to be Riemann-integrable if its components $a_{\alpha\beta} : [0,1] \to \mathbb{C}$, $\alpha, \beta = 1, \ldots, r$, are Riemann-integrable. We point out that a complex-valued function g is Riemann-integrable when its real and imaginary parts $\Re(g)$ and $\Im(g)$ are Riemann-integrable in the classical sense. We also recall that any Riemann-integrable function is *bounded* by definition.
- We use a notation borrowed from probability theory to indicate sets. For example, if $f, g : D \subseteq \mathbb{R}^k \to \mathbb{C}^{r\times r}$, then $\{\sigma_{\max}(f) > 0\} = \{\mathbf{x} \in D : \sigma_{\max}(f(\mathbf{x})) > 0\}$, $\mu_k\{\|f - g\| \geq \epsilon\}$ is the measure of the set $\{\mathbf{x} \in D : \|f(\mathbf{x}) - g(\mathbf{x})\| \geq \epsilon\}$, etc.
- A matrix-sequence is any sequence of the form $\{A_n\}_n$, where $A_n \in \mathbb{C}^{sn\times sn}$ and s is a *fixed* positive integer. The role of s will become clear later on. A matrix-sequence $\{A_n\}_n$ is said to be Hermitian if each A_n is Hermitian.

2.2 Preliminaries on Measure and Integration Theory

2.2.1 Measurability

The following lemma can be derived from the results in [7, Section VI.1]. It will be used essentially everywhere in this paper, either explicitly or implicitly.

Lemma 2.1 *Let $f : D \subseteq \mathbb{R}^k \to \mathbb{C}^{r\times r}$ be measurable and let $g : \mathbb{C}^r \to \mathbb{C}$ be continuous and symmetric in its r arguments, i.e., $g(\lambda_1, \ldots, \lambda_r) = g(\lambda_{\rho(1)}, \ldots, \lambda_{\rho(r)})$ for all permutations ρ of $\{1, \ldots, r\}$. Then, the function $\mathbf{x} \mapsto g(\lambda_1(f(\mathbf{x})), \ldots, \lambda_r(f(\mathbf{x})))$ is well-defined (independently of the labeling of the eigenvalues of $f(\mathbf{x})$) and measurable. As a consequence:*

- *the function $\mathbf{x} \mapsto g(\sigma_1(f(\mathbf{x})), \ldots, \sigma_r(f(\mathbf{x})))$ is measurable;*
- *the functions $\mathbf{x} \mapsto \sum_{i=1}^r F(\lambda_i(f(\mathbf{x})))$ and $\mathbf{x} \mapsto \sum_{i=1}^r F(\sigma_i(f(\mathbf{x})))$ are measurable for all continuous $F : \mathbb{C} \to \mathbb{C}$;*
- *the function $\mathbf{x} \mapsto \|f(\mathbf{x})\|_p$ is measurable for all $p \in [1, \infty]$.*

2.2.2 Convergence in Measure

In this section we collect some key properties of the convergence in measure, which plays a central role in the theory of block GLT sequences. We first recall that the convergence in measure is induced by a pseudometric. More precisely, let $r \geq 1$ and $D \subset \mathbb{R}^k$ with $0 < \mu_k(D) < \infty$, and set

$$\mathfrak{M}(D,r) = \{\kappa : D \to \mathbb{C}^{r\times r} : \kappa \text{ is measurable}\}.$$

Then, there exists a pseudometric d_{measure} on $\mathfrak{M}(D,r)$ such that $\kappa_m \to \kappa$ in measure if and only if $d_{\text{measure}}(\kappa_m,\kappa) \to 0$ as $m \to \infty$. For example, we can take

$$d_{\text{measure}}(\kappa,\xi) = \sum_{i,j=1}^{r} \int_D \frac{|\kappa_{ij}(\mathbf{x}) - \xi_{ij}(\mathbf{x})|}{1 + |\kappa_{ij}(\mathbf{x}) - \xi_{ij}(\mathbf{x})|} \mathrm{d}\mathbf{x}, \quad \kappa, \xi \in \mathfrak{M}(D,r);$$

see, e.g., [36, p. 102]. Further properties of the convergence in measure that we shall need later on are collected in the next lemmas [30, Lemmas 2.2 and 2.3].

Lemma 2.2 *Let $f_m, g_m, f, g : D \subseteq \mathbb{R}^k \to \mathbb{C}^{r\times r}$ be measurable functions.*

- *If $f_m \to f$ in measure and $g_m \to g$ in measure, then $\alpha f_m + \beta g_m \to \alpha f + \beta g$ in measure for all $\alpha, \beta \in \mathbb{C}$.*
- *If $f_m \to f$ in measure, $g_m \to g$ in measure, and $\mu_k(D) < \infty$, then $f_m \circ g_m \to f \circ g$ in measure and $f_m g_m \to fg$ in measure.*

Lemma 2.3 *Let $g_m, g : D \to \mathbb{C}^{r\times r}$ be measurable functions defined on a set $D \subset \mathbb{R}^k$ with $0 < \mu_k(D) < \infty$. If $g_m \to g$ in measure, then $\sum_{j=1}^{r} F(\lambda_j(g_m(\mathbf{x})))$ converges to $\sum_{j=1}^{r} F(\lambda_j(g(\mathbf{x})))$ in $L^1(D)$ for all $F \in C_c(\mathbb{C})$.*

2.2.3 Technical Lemma

We conclude this section on measure and integration theory by stating and proving a technical lemma that we shall need in Sect. 5.

Lemma 2.4 *Let $f : D \to \mathbb{C}^{r\times r}$ be a measurable function defined on a set $D \subset \mathbb{R}^k$ with $0 < \mu_k(D) < \infty$, and assume that*

$$\frac{1}{\mu_k(D)} \int_D \frac{\sum_{j=1}^{r} F(\sigma_j(f(\mathbf{x})))}{r} \mathrm{d}\mathbf{x} = F(0), \quad \forall\, F \in C_c(\mathbb{R}).$$

Then $f = O_r$ a.e.

Proof Suppose by contradiction that $\mu_k\{f \neq O_r\} = \mu_k\{\sigma_{\max}(f) > 0\} > 0$. Then, there exists $\epsilon > 0$ such that $\mu_k\{\sigma_{\max}(f) \geq \epsilon\} > 0$. Take a real function $F \in C_c(\mathbb{R})$ such that $F(0) = 1 = \max_{y\in\mathbb{R}} F(y)$ and $F(y) = 0$ for $|y| \geq \epsilon$. Then,

$$\begin{aligned}
&\frac{1}{\mu_k(D)} \int_D \frac{\sum_{j=1}^{r} F(\sigma_j(f(\mathbf{x})))}{r} \mathrm{d}\mathbf{x} \\
&= \frac{1}{r\mu_k(D)} \left[\int_{\{\sigma_{\max}(f)<\epsilon\}} \sum_{j=1}^{r} F(\sigma_j(f(\mathbf{x}))) \mathrm{d}\mathbf{x} + \int_{\{\sigma_{\max}(f)\geq\epsilon\}} \sum_{j=1}^{r} F(\sigma_j(f(\mathbf{x}))) \mathrm{d}\mathbf{x} \right]
\end{aligned}$$

$$\leq \frac{1}{r\mu_k(D)}\left[\int_{\{\sigma_{\max}(f)<\epsilon\}} r\mathrm{d}\mathbf{x} + \int_{\{\sigma_{\max}(f)\geq\epsilon\}} (r-1)\mathrm{d}\mathbf{x}\right]$$
$$= \frac{r\mu_k\{\sigma_{\max}(f)<\epsilon\} + (r-1)\mu_k\{\sigma_{\max}(f)\geq\epsilon\}}{r\mu_k(D)} < 1 = F(0),$$

which is a contradiction to the assumption. □

2.3 Singular Value and Eigenvalue Distribution of a Matrix-Sequence

We introduce in this section the fundamental definitions of singular value and spectral distribution for a given matrix-sequence. Recall from Sect. 2.1 that a matrix-sequence is a sequence of the form $\{A_n\}_n$, where $A_n \in \mathbb{C}^{sn\times sn}$ and s is a *fixed* positive integer.

Definition 2.1 (Singular Value and Eigenvalue Distribution of a Matrix-Sequence) Let $\{A_n\}_n$ be a matrix-sequence and let $f : D \subset \mathbb{R}^k \to \mathbb{C}^{r\times r}$ be a measurable matrix-valued function defined on a set D with $0 < \mu_k(D) < \infty$.

- We say that $\{A_n\}_n$ has a (asymptotic) singular value distribution described by f, and we write $\{A_n\}_n \sim_\sigma f$, if

$$\lim_{n\to\infty} \frac{1}{sn}\sum_{j=1}^{sn} F(\sigma_j(A_n)) = \frac{1}{\mu_k(D)}\int_D \frac{\sum_{i=1}^r F(\sigma_i(f(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}, \quad \forall\, F \in C_c(\mathbb{R}). \tag{1}$$

- We say that $\{A_n\}_n$ has a (asymptotic) eigenvalue (or spectral) distribution described by f, and we write $\{A_n\}_n \sim_\lambda f$, if

$$\lim_{n\to\infty} \frac{1}{sn}\sum_{j=1}^{sn} F(\lambda_j(A_n)) = \frac{1}{\mu_k(D)}\int_D \frac{\sum_{i=1}^r F(\lambda_i(f(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}, \quad \forall\, F \in C_c(\mathbb{C}). \tag{2}$$

Note that Definition 2.1 is well-posed by Lemma 2.1, which ensures that the functions $\mathbf{x} \mapsto \sum_{i=1}^r F(\sigma_i(f(\mathbf{x})))$ and $\mathbf{x} \mapsto \sum_{i=1}^r F(\lambda_i(f(\mathbf{x})))$ are measurable. Whenever we write a relation such as $\{A_n\}_n \sim_\sigma f$ or $\{A_n\}_n \sim_\lambda f$, it is understood that f is as in Definition 2.1; that is, f is a measurable function taking values in $\mathbb{C}^{r\times r}$ for some $r \geq 1$ and defined on a subset D of some $\mathbb{R}^k$ with $0 < \mu_k(D) < \infty$. We refer the reader to [26, Remark 1] or to the appendix of [31] for the informal meaning behind the spectral distribution (2); a completely analogous meaning can be given also for the singular value distribution (1).

2.4 Zero-Distributed Sequences

A matrix-sequence $\{Z_n\}_n$ is said to be zero-distributed if $\{Z_n\}_n \sim_\sigma 0$. It is clear that, for any $r \geq 1$, $\{Z_n\}_n \sim_\sigma 0$ is equivalent to $\{Z_n\}_n \sim_\sigma O_r$. Theorem 2.1 provides a characterization of zero-distributed sequences [24, Theorem 3.2].

Theorem 2.1 *Let $\{Z_n\}_n$ be a matrix-sequence. The following are equivalent.*

1. $\{Z_n\}_n \sim_\sigma 0$.
2. *For all n we have $Z_n = R_n + N_n$, where $\lim_{n\to\infty} (\mathrm{rank}(R_n)/n) = \lim_{n\to\infty} \|N_n\| = 0$.*

2.5 Sparsely Unbounded and Sparsely Vanishing Matrix-Sequences

The notions of sparsely unbounded and sparsely vanishing matrix-sequences play an important role within the framework of the theory of block GLT sequences.

Definition 2.2 (Sparsely Unbounded Matrix-Sequence) A matrix-sequence $\{A_n\}_n$ is said to be sparsely unbounded (s.u.) if for every $M > 0$ there exists n_M such that, for $n \geq n_M$,

$$\frac{\#\{i \in \{1, \ldots, sn\} : \sigma_i(A_n) > M\}}{n} \leq r(M),$$

where $\lim_{M\to\infty} r(M) = 0$.

As highlighted in the next propositions, the product of two s.u. matrix-sequences is s.u. and any matrix-sequence enjoying an asymptotic singular value distribution is s.u. For the related proofs, see [30, Propositions 2.2 and 2.3].

Proposition 2.1 *If $\{A_n\}_n, \{A'_n\}_n$ are s.u. then $\{A_n A'_n\}_n$ is s.u.*

Proposition 2.2 *If $\{A_n\}_n \sim_\sigma f$ then $\{A_n\}_n$ is s.u.*

Strictly related to the notion of sparsely unbounded matrix-sequences is the notion of sparsely vanishing matrix-sequences.

Definition 2.3 (Sparsely Vanishing Matrix-Sequence) A matrix-sequence $\{A_n\}_n$ is said to be sparsely vanishing (s.v.) if for every $M > 0$ there exists n_M such that, for $n \geq n_M$,

$$\frac{\#\{i \in \{1, \ldots, sn\} : \sigma_i(A_n) < 1/M\}}{n} \leq r(M),$$

where $\lim_{M\to\infty} r(M) = 0$.

Remark 2.1 If $\{A_n\}_n$ is s.v. then $\{A_n^\dagger\}_n$ is s.u. This follows from the fact that the singular values of $A^\dagger$ are $1/\sigma_1(A), \ldots, 1/\sigma_r(A), 0, \ldots, 0$, where $\sigma_1(A), \ldots, \sigma_r(A)$ are the nonzero singular values of A ($r = \mathrm{rank}(A)$).

Remark 2.2 We know from [24, Remark 8.6] that $\{A_n\}_n$ is s.v. if and only if

$$\lim_{M\to\infty} \limsup_{n\to\infty} \frac{\#\{i \in \{1,\ldots,sn\} :\ \sigma_i(A_n) < 1/M\}}{n} = 0. \tag{3}$$

Proposition 2.3 is the analog of Proposition 2.2 for s.v. matrix-sequences.

Proposition 2.3 *If $\{A_n\}_n \sim_\sigma f$ then $\{A_n\}_n$ is s.v. if and only if f is invertible a.e.*

Proof Let $D \subset \mathbb{R}^k$ be the domain of the matrix-valued function $f : D \to \mathbb{C}^{r\times r}$. Fix $M > 0$ and take $F_M, G_M \in C_c(\mathbb{R})$ such that

$$\begin{cases} F_M = 1 \text{ over } [0, 1/(2M)], \\ F_M = 0 \text{ over } [1/M, \infty), \\ 0 \le F_M \le 1 \text{ over } \mathbb{R}, \end{cases} \qquad \begin{cases} G_M = 1 \text{ over } [0, 1/M], \\ G_M = 0 \text{ over } [2/M, \infty), \\ 0 \le G_M \le 1 \text{ over } \mathbb{R}. \end{cases}$$

Note that, once we have defined F_M, we may simply take $G_M = F_{M/2}$. By construction we have $F_M \le \chi_{[0,1/M)} \le G_M$ over $[0,\infty)$, hence

$$\begin{aligned} \frac{1}{sn}\sum_{i=1}^{sn} F_M(\sigma_i(A_n)) &\le \frac{1}{sn}\sum_{i=1}^{sn} \chi_{[0,1/M)}(\sigma_i(A_n)) \\ &= \frac{\#\{i \in \{1,\ldots,sn\} :\ \sigma_i(A_n) < 1/M\}}{sn} \le \frac{1}{sn}\sum_{i=1}^{sn} G_M(\sigma_i(A_n)) \end{aligned}$$

Passing to the limit as $n \to \infty$, we obtain

$$\begin{aligned} \frac{1}{\mu_k(D)}\int_D \frac{\sum_{i=1}^r F_M(\sigma_i(f(\mathbf{x})))}{r}\mathrm{d}\mathbf{x} &\le \limsup_{n\to\infty} \frac{\#\{i \in \{1,\ldots,sn\} :\ \sigma_i(A_n) < 1/M\}}{sn} \\ &\le \frac{1}{\mu_k(D)}\int_D \frac{\sum_{i=1}^r G_M(\sigma_i(f(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}. \end{aligned}$$

Since both $\frac{1}{r}\sum_{i=1}^r F_M(\sigma_i(f(\mathbf{x})))$ and $\frac{1}{r}\sum_{i=1}^r G_M(\sigma_i(f(\mathbf{x})))$ converge to $\frac{1}{r}\sum_{i=1}^r \chi_{\{\sigma_i(f)=0\}}(\mathbf{x})$ a.e. and $\left|\frac{1}{r}\sum_{i=1}^r F_M(\sigma_i(f(\mathbf{x})))\right|, \left|\frac{1}{r}\sum_{i=1}^r G_M(\sigma_i(f(\mathbf{x})))\right| \le 1$, by the dominated convergence theorem we get

$$\begin{aligned} \lim_{M\to\infty}\int_D \frac{\sum_{i=1}^r F_M(\sigma_i(f(\mathbf{x})))}{r}\mathrm{d}\mathbf{x} &= \lim_{M\to\infty}\int_D \frac{\sum_{i=1}^r G_M(\sigma_i(f(\mathbf{x})))}{r}\mathrm{d}\mathbf{x} \\ &= \int_D \frac{\sum_{i=1}^r \chi_{\{\sigma_i(f)=0\}}(\mathbf{x})}{r}\mathrm{d}\mathbf{x}. \end{aligned}$$

Thus,

$$\lim_{M\to\infty}\limsup_{n\to\infty}\frac{\#\{i\in\{1,\dots,sn\}:\ \sigma_i(A_n)<1/M\}}{sn}=\frac{1}{\mu_k(D)}\int_D\frac{\sum_{i=1}^r\chi_{\{\sigma_i(f)=0\}}(\mathbf{x})}{r}\mathrm{d}\mathbf{x},$$

which is equal to 0 if and only if f is invertible a.e. By Remark 2.2, this means that $\{A_n\}_n$ is s.v. if and only if f is invertible a.e. □

2.6 Block Toeplitz Matrices

A matrix of the form

$$[A_{i-j}]_{i,j=1}^n=\begin{bmatrix}A_0 & A_{-1} & \cdots & \cdots & A_{-(n-1)}\\ A_1 & \ddots & \ddots & & \vdots\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ \vdots & & \ddots & \ddots & A_{-1}\\ A_{n-1} & \cdots & \cdots & A_1 & A_0\end{bmatrix}\in\mathbb{C}^{sn\times sn},\tag{4}$$

with blocks $A_k\in\mathbb{C}^{s\times s}$ for $k=-(n-1),\dots,n-1$, is called a block Toeplitz matrix. If $s=1$, it is simply referred to as a Toeplitz matrix. Given a function $f:[-\pi,\pi]\to\mathbb{C}^{s\times s}$ belonging to $L^1([-\pi,\pi],s)$, its Fourier coefficients are denoted by

$$f_k=\frac{1}{2\pi}\int_{-\pi}^{\pi}f(\theta)\mathrm{e}^{-\mathrm{i}k\theta}\mathrm{d}\theta\in\mathbb{C}^{s\times s},\quad k\in\mathbb{Z},\tag{5}$$

where the integrals are computed componentwise. The nth block Toeplitz matrix associated with f is defined as

$$T_n(f)=[f_{i-j}]_{i,j=1}^n\in\mathbb{C}^{sn\times sn}.$$

We call $\{T_n(f)\}_n$ the block Toeplitz sequence associated with f, which in turn is called the generating function of $\{T_n(f)\}_n$. Note that $T_n(I_s)=I_{sn}$.

3 Approximating Classes of Sequences

The notion of approximating classes of sequences (a.c.s.), which is fundamental to the theory of block GLT sequences, is attributed to the second author [38], though the underlying idea was already present in the pioneering papers by Tilli [42] and Tyrtyshnikov [43]. Here is the formal definition.

Definition 3.1 (Approximating Class of Sequences) Let $\{A_n\}_n$ a matrix-sequence. An approximating class of sequences (a.c.s.) for $\{A_n\}_n$ is a sequence of matrix-sequences $\{\{B_{n,m}\}_n\}_m$ with the following property: for every m there exists n_m such that, for $n \geq n_m$,

$$A_n = B_{n,m} + R_{n,m} + N_{n,m}, \quad \operatorname{rank}(R_{n,m}) \leq c(m)n, \quad \|N_{n,m}\| \leq \omega(m), \tag{6}$$

where n_m, $c(m)$, $\omega(m)$ depend only on m and $\lim_{m\to\infty} c(m) = \lim_{m\to\infty} \omega(m) = 0$.

As explained in [30, Section 3.1], there exists a complete pseudometric $d_{\text{a.c.s.}}$ in the space of matrix-sequences

$$\mathscr{E} = \{\{A_n\}_n : \{A_n\}_n \text{ is a matrix-sequence}\}$$

such that

$$d_{\text{a.c.s.}}(\{A_n\}_n, \{B_n\}_n) = 0 \iff \{A_n - B_n\}_n \sim_\sigma 0 \tag{7}$$

and $\{\{B_{n,m}\}_n\}_m$ is an a.c.s. for $\{A_n\}_n$ if and only if $d_{\text{a.c.s.}}(\{B_{n,m}\}_n, \{A_n\}_n) \to 0$ as $m \to \infty$. We will therefore use the convergence notation $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ to indicate that $\{\{B_{n,m}\}_n\}_m$ is an a.c.s. for $\{A_n\}_n$.

In the remainder of this section we summarize the properties of a.c.s. that we shall need later on. Properties **ACS 1** – **ACS 3** are Theorems 3.1–3.3 from [30]. The only property which has not been proved in previous works is **ACS 4**, which will be proved below.

ACS 1. If there exist matrix-sequences $\{B_{n,m}\}_n \sim_\sigma f_m$ such that $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ and $f_m \to f$ in measure, then $\{A_n\}_n \sim_\sigma f$.

ACS 2. Suppose each A_n is Hermitian. If there exist Hermitian matrix-sequences $\{B_{n,m}\}_n \sim_\lambda f_m$ such that $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ and $f_m \to f$ in measure, then $\{A_n\}_n \sim_\lambda f$.

ACS 3. If $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ and $\{B'_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A'_n\}_n$ then

- $\{B^*_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A^*_n\}_n$,
- $\{\alpha B_{n,m} + \beta B'_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{\alpha A_n + \beta A'_n\}_n$ for all $\alpha, \beta \in \mathbb{C}$,
- $\{B_{n,m}B'_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_nA'_n\}_n$ provided that $\{A_n\}_n$, $\{A'_n\}_n$ are s.u.,
- $\{B_{n,m}C_n\}_n \xrightarrow{\text{a.c.s.}} \{A_nC_n\}_n$ provided that $\{C_n\}_n$ is s.u.

ACS 4. Suppose $\{A_n - B_{n,m}\}_n \sim_\sigma g_m$ for some $g_m : D \to \mathbb{C}^{r\times r}$. If $g_m \to O_r$ in measure then $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$.

Proof of the last property Let $C_{n,m} = A_n - B_{n,m}$. For any $\ell \in \mathbb{N}$, choose $F_\ell \in C_c(\mathbb{R})$ such that $F_\ell = 1$ over $[0, 1/(2\ell)]$, $F_\ell = 0$ over $[1/\ell, \infty)$, and $0 \leq F_\ell \leq 1$

over $\mathbb{R}$. Note that $F_\ell \le \chi_{[0,1/\ell]}$ over $[0,\infty)$. For every m, ℓ, we have

$$\begin{aligned}\frac{\#\{i \in \{1,\dots,sn\} : \sigma_i(C_{n,m}) > 1/\ell\}}{sn} &= 1 - \frac{\#\{i \in \{1,\dots,sn\} : \sigma_i(C_{n,m}) \le 1/\ell\}}{sn} \\ = 1 - \frac{1}{sn}\sum_{i=1}^{sn} \chi_{[0,1/\ell]}(\sigma_i(C_{n,m})) &\le 1 - \frac{1}{sn}\sum_{i=1}^{sn} F_\ell(\sigma_i(C_{n,m})) \overset{n\to\infty}{\longrightarrow} c(m,\ell),\end{aligned} \tag{8}$$

where

$$c(m,\ell) = 1 - \frac{1}{\mu_k(D)}\int_D \frac{\sum_{j=1}^r F_\ell(\sigma_j(g_m(\mathbf{x})))}{r}\mathrm{d}\mathbf{x}.$$

For every fixed ℓ we have $c(m,\ell) \to 0$ as $m \to \infty$, by Lemmas 2.2 and 2.3 (take into account that $\sum_{j=1}^r F_\ell(\sigma_j(g_m(\mathbf{x}))) = \sum_{j=1}^r G_\ell(\lambda_j(g_m(\mathbf{x})g_m(\mathbf{x})^*))$ where $G_\ell(z) = F_\ell(\sqrt{|z|})$ belongs to $C_c(\mathbb{C})$). Hence, there exists a sequence $\{\ell_m\}_m$ of natural numbers such that $\ell_m \to \infty$ and $c(m,\ell_m) \to 0$. By (8), for each m we have

$$\limsup_{n\to\infty} \frac{\#\{i \in \{1,\dots,sn\} : \sigma_i(C_{n,m}) > 1/\ell_m\}}{sn} \le c(m,\ell_m). \tag{9}$$

Let $C_{n,m} = U_{n,m}\Sigma_{n,m}V_{n,m}^*$ be a singular value decomposition of $C_{n,m}$. Let $\hat{\Sigma}_{n,m}$ be the matrix obtained from $\Sigma_{n,m}$ by setting to 0 all the singular values that are less than or equal to $1/\ell_m$, and let $\tilde{\Sigma}_{n,m} = \Sigma_{n,m} - \hat{\Sigma}_{n,m}$ be the matrix obtained from $\Sigma_{n,m}$ by setting to 0 all the singular values that exceed $1/\ell_m$. Then we can write $C_{n,m} = R_{n,m} + N_{n,m}$, where $R_{n,m} = U_{n,m}\hat{\Sigma}_{n,m}V_{n,m}^*$ and $N_{n,m} = U_{n,m}\tilde{\Sigma}_{n,m}V_{n,m}^*$. By definition, $\|N_{n,m}\| \le 1/\ell_m$. Moreover, (9) yields $\limsup_{n\to\infty}(\mathrm{rank}(R_{n,m})/n) \le c(m,\ell_m)$, implying the existence of a n_m such that, for $n \ge n_m$, $\mathrm{rank}(R_{n,m}) \le (c(m,\ell_m) + 1/m)\,n$. This shows that $\{C_{n,m}\}_n \overset{\text{a.c.s.}}{\longrightarrow} \{O_{sn}\}_n$, i.e., $\{B_{n,m}\}_n \overset{\text{a.c.s.}}{\longrightarrow} \{A_n\}_n$. □

4 Block Locally Toeplitz Sequences

In this section we summarize the theory of block LT sequences, which has been developed in [30]. Needless to say, this theory is the basis of the theory of block GLT sequences. A block LT sequence $\{A_n\}_n$ is a special matrix-sequence equipped with a function of the form $a(x) \circ f(\theta)$, where $a : [0,1] \to \mathbb{C}^{s\times s}$ is Riemann-integrable and $f \in L^1([-\pi,\pi], s)$. The function $a(x) \circ f(\theta)$ is referred to as the symbol of $\{A_n\}_n$. In what follows, we write $\{A_n\}_n \sim_{\mathrm{LT}} a(x) \circ f(\theta)$ to indicate that $\{A_n\}_n$ is a block LT sequence with symbol $a(x) \circ f(\theta)$; it is understood that $a : [0,1] \to \mathbb{C}^{s\times s}$ is Riemann-integrable and $f \in L^1([-\pi,\pi], s)$. For $n \in \mathbb{N}$ and

$a : [0,1] \to \mathbb{C}^{s\times s}$, we define the block diagonal sampling matrix $D_n(a)$ as the following block diagonal matrix of size $sn \times sn$:

$$D_n(a) = \underset{i=1,\dots,n}{\text{diag}}\ a\left(\frac{i}{n}\right).$$

LT 1. If $\{A_n^{(i,j)}\}_n \sim_{\text{LT}} a^{(i,j)}(x) \circ f^{(i,j)}(\theta)$ for $i = 1,\dots,p$ and $j = 1,\dots,q_i$ then:

- $\{\sum_{i=1}^{p}\prod_{j=1}^{q_i} A_n^{(i,j)}\}_n \sim_\sigma \sum_{i=1}^{p}\prod_{j=1}^{q_i} a^{(i,j)}(x) \circ f^{(i,j)}(\theta)$;
- $\{\Re(\sum_{i=1}^{p}\prod_{j=1}^{q_i} A_n^{(i,j)})\}_n \sim_\lambda \Re(\sum_{i=1}^{p}\prod_{j=1}^{q_i} a^{(i,j)}(x) \circ f^{(i,j)}(\theta))$.

LT 2. We have:

- $\{T_n(f)\}_n \sim_{\text{LT}} \mathbf{1}_s \circ f(\theta)$ if $f \in L^1([-\pi,\pi], s)$;
- $\{D_n(a)\}_n \sim_{\text{LT}} a(x) \circ \mathbf{1}_s$ if $a : [0,1] \to \mathbb{C}^{s\times s}$ is Riemann-integrable;
- $\{Z_n\}_n \sim_{\text{LT}} O_s$ if and only if $\{Z_n\}_n \sim_\sigma 0$.

LT 3. If $\{A_n\}_n \sim_{\text{LT}} a(x) \circ f(\theta)$ then:

- $\{A_n^*\}_n \sim_{\text{LT}} a(x)^* \circ f(\theta)^* = (a(x) \circ f(\theta))^*$;
- $\{\alpha A_n\}_n \sim_{\text{LT}} \alpha a(x) \circ f(\theta) = a(x) \circ \alpha f(\theta)$ for all $\alpha \in \mathbb{C}$.

5 Block Generalized Locally Toeplitz Sequences

In this section we develop the theory of block GLT sequences, by correcting and extending the results in [40, Section 3.3].

5.1 Equivalent Definitions of Block GLT Sequences

Block GLT sequences can be defined in several different ways. We begin with what we may call the 'classical definition' (though, actually, a definition of this kind has never been formulated before).

Definition 5.1 (Block Generalized Locally Toeplitz Sequence) Let $\{A_n\}_n$ be a matrix-sequence and let $\kappa : [0,1] \times [-\pi,\pi] \to \mathbb{C}^{s\times s}$ be measurable. We say that $\{A_n\}_n$ is a block generalized locally Toeplitz (GLT) sequence with symbol κ, and we write $\{A_n\}_n \sim_{\text{GLT}} \kappa$, if the following condition is met.

For every $m \in \mathbb{N}$ there exists a finite number of block LT sequences $\{A_{n,m}^{(i,j)}\}_n \sim_{\text{LT}} a_m^{(i,j)}(x) \circ f_m^{(i,j)}(\theta)$, $i = 1, \ldots, N_m$, $j = 1, \ldots, M_{m,i}$, such that:

- $\sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} (a_m^{(i,j)}(x) \circ f_m^{(i,j)}(\theta)) \to \kappa(x, \theta)$ in measure;
- $\left\{ \sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} A_{n,m}^{(i,j)} \right\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$.

In the case $s = 1$, it can be shown that Definition 5.1 is equivalent to the definition of (scalar) GLT sequences given in [24, Chapter 8]. Whenever we write a relation such as $\{A_n\}_n \sim_{\text{GLT}} \kappa$, it is understood that $\kappa : [0, 1] \times [-\pi, \pi] \to \mathbb{C}^{s \times s}$ is measurable, as in Definition 5.1.

Remark 5.1 It is clear that any sum of products of block LT sequences is a block GLT sequence. More precisely, if $\{A_n^{(i,j)}\}_n \sim_{\text{LT}} a^{(i,j)}(x) \circ f^{(i,j)}(\theta)$ for $i = 1, \ldots, p$ and $j = 1, \ldots, q_i$ then

$$\left\{ \sum_{i=1}^{p} \prod_{j=1}^{q_i} A_n^{(i,j)} \right\}_n \sim_{\text{GLT}} \sum_{i=1}^{p} \prod_{j=1}^{q_i} (a^{(i,j)}(x) \circ f^{(i,j)}(\theta)).$$

Remark 5.2 Let $\{A_n\}_n \sim_{\text{GLT}} \kappa$ and $\{B_n\}_n \sim_{\text{GLT}} \xi$. Then, $\{A_n^*\}_n \sim_{\text{GLT}} \kappa^*$ and $\{\alpha A_n + \beta B_n\}_n \sim_{\text{GLT}} \alpha\kappa + \beta\xi$ for all $\alpha, \beta \in \mathbb{C}$. This follows immediately from Definition 5.1, **LT 3** and **ACS 3**.

In the remainder of this section, we present an alternative definition of block GLT sequences, which is illuminating for many purposes. Let

$$\begin{aligned} \mathscr{E} &= \{\{A_n\}_n : \{A_n\}_n \text{ is a matrix-sequence}\}, \\ \mathfrak{M} &= \{\kappa : [0, 1] \times [-\pi, \pi] \to \mathbb{C}^{s \times s} : \kappa \text{ is measurable}\}, \\ \mathscr{E} \times \mathfrak{M} &= \{(\{A_n\}_n, \kappa) : \{A_n\}_n \in \mathscr{E}, \kappa \in \mathfrak{M}\}. \end{aligned}$$

We make the following observations.

- $\mathscr{E}$ is a *-algebra with respect to the natural pointwise operations (namely, $\{A_n\}_n^* = \{A_n^*\}_n$, $\alpha\{A_n\}_n + \beta\{B_n\}_n = \{\alpha A_n + \beta B_n\}_n$, $\{A_n\}_n\{B_n\}_n = \{A_n B_n\}_n$), and it is also a pseudometric space with respect to the pseudometric $d_{\text{a.c.s.}}$ inducing the a.c.s. convergence.
- $\mathfrak{M}$ is a *-algebra with respect to the natural pointwise operations, and it is also a pseudometric space with respect one of the equivalent pseudometrics d_{measure} inducing the convergence in measure.
- $\mathscr{E} \times \mathfrak{M}$ is a *-algebra with respect to the natural pointwise operations (namely, $(\{A_n\}_n, \kappa)^* = (\{A_n^*\}_n, \kappa^*)$, $\alpha(\{A_n\}_n, \kappa) + \beta(\{B_n\}_n, \xi) = (\{\alpha A_n + \beta B_n\}_n, \alpha\kappa + \beta\xi)$, $(\{A_n\}_n, \kappa)(\{B_n\}_n, \xi) = (\{A_n B_n\}_n, \kappa\xi)$), and it is also a pseudometric space with respect to the product pseudometric

$$(d_{\text{a.c.s.}} \times d_{\text{measure}})((\{A_n\}_n, \kappa), (\{B_n\}_n, \xi)) = d_{\text{a.c.s.}}(\{A_n\}_n, \{B_n\}_n) + d_{\text{measure}}(\kappa, \xi).$$

Let $\mathcal{A}$ be the *-subalgebra of $\mathscr{E} \times \mathfrak{M}$ generated by the 'block LT pairs'

$$\mathcal{L} = \{(\{A_n\}_n, a(x) \circ f(\theta)) \in \mathscr{E} \times \mathfrak{M} : \ \{A_n\}_n \sim_{\text{LT}} a(x) \circ f(\theta)\}.$$

Using **LT 3**, it is not difficult to see that

$$\mathcal{A} = \left\{ \left(\sum_{i=1}^{p} \prod_{j=1}^{q_i} A_n^{(i,j)}, \sum_{i=1}^{p} \prod_{j=1}^{q_i} (a^{(i,j)}(x) \circ f^{(i,j)}(\theta)) \right) : \right.$$
$$\left. p, q_1, \ldots, q_p \in \mathbb{N}, \quad \{A_n^{(i,j)}\}_n \sim_{\text{LT}} a^{(i,j)}(x) \circ f^{(i,j)}(\theta) \text{ for all } i, j \right\}.$$

We can now reformulate Definition 5.1 as follows.

Definition 5.2 (Block Generalized Locally Toeplitz Sequence) Let $\{A_n\}_n$ be a matrix-sequence and let $\kappa : [0, 1] \times [-\pi, \pi] \to \mathbb{C}^{s \times s}$ be measurable. We say that $\{A_n\}_n$ is a block generalized locally Toeplitz (GLT) sequence with symbol κ, and we write $\{A_n\}_n \sim_{\text{GLT}} \kappa$, if the pair $(\{A_n\}_n, \kappa)$ belongs to the topological closure of $\mathcal{A}$ in $(\mathscr{E} \times \mathfrak{M}, d_{\text{a.c.s.}} \times d_{\text{measure}})$. In other words, the set of 'block GLT pairs'

$$\mathcal{G} = \{(\{A_n\}_n, \kappa) \in \mathscr{E} \times \mathfrak{M} : \ \{A_n\}_n \sim_{\text{GLT}} \kappa\}$$

is defined as the topological closure of $\mathcal{A}$ in $(\mathscr{E} \times \mathfrak{M}, d_{\text{a.c.s.}} \times d_{\text{measure}})$.

In the light of this algebraic-topological definition of block GLT sequences, the following theorem is obvious.

Theorem 5.1 *Let $\{A_n\}_n$ be a matrix-sequence and let $\kappa : [0, 1] \times [-\pi, \pi] \to \mathbb{C}^{s \times s}$ be a measurable matrix-valued function. Suppose that:*

1. *$\{B_{n,m}\}_n \sim_{\text{GLT}} \kappa_m$ for every m;*
2. *$\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$;*
3. *$\kappa_m \to \kappa$ in measure.*

Then $\{A_n\}_n \sim_{\text{GLT}} \kappa$.

5.2 Singular Value and Spectral Distribution of Block GLT Sequences

In this section we prove the main singular value and eigenvalue distribution results for block GLT sequences.

Theorem 5.2 *If $\{A_n\}_n \sim_{\text{GLT}} \kappa$ then $\{A_n\}_n \sim_\sigma \kappa$.*

Proof By definition, for every $m \in \mathbb{N}$ there exist block LT sequences $\{A_{n,m}^{(i,j)}\}_n \sim_{\text{LT}} a_m^{(i,j)}(x) \circ f_m^{(i,j)}(\theta)$, $i = 1, \ldots, N_m$, $j = 1, \ldots, M_{m,i}$, such that:

- $\sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} (a_m^{(i,j)}(x) \circ f_m^{(i,j)}(\theta)) \to \kappa(x,\theta)$ in measure;
- $\left\{ \sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} A_{n,m}^{(i,j)} \right\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$.

Moreover, by **LT 1**,

- $\left\{ \sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} A_{n,m}^{(i,j)} \right\}_n \sim_\sigma \sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} (a_m^{(i,j)}(x) \circ f_m^{(i,j)}(\theta))$.

We conclude that $\{A_n\}_n \sim_\sigma \kappa$ by **ACS 1**. □

Remark 5.3 Any block GLT sequence $\{A_n\}_n$ is s.u. This follows from Theorem 5.2 and Proposition 2.2.

Using Theorem 5.2 we now show that the symbol of a block GLT sequence is essentially unique and that the symbol of a block GLT sequence formed by Hermitian matrices is Hermitian a.e.

Proposition 5.1 *If $\{A_n\}_n \sim_{\text{GLT}} \kappa$ and $\{A_n\}_n \sim_{\text{GLT}} \xi$ then $\kappa = \xi$ a.e.*

Proof By Remark 5.2 we have $\{O_{sn}\}_n = \{A_n - A_n\}_n \sim_{\text{GLT}} \kappa - \xi$. Hence, by Theorem 5.2, we also have $\{O_{sn}\}_n \sim_\sigma \kappa - \xi$, i.e.,

$$F(0) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \int_0^1 \frac{\sum_{j=1}^{s} F(\sigma_j(\kappa(x,\theta) - \xi(x,\theta)))}{s} \mathrm{d}x\mathrm{d}\theta, \quad \forall\, F \in C_c(\mathbb{R}).$$

We conclude that $\kappa - \xi = O_s$ a.e. by Lemma 2.4. □

Proposition 5.2 *If $\{A_n\}_n \sim_{\text{GLT}} \kappa$ and the A_n are Hermitian then κ is Hermitian a.e.*

Proof Since the matrices A_n are Hermitian, by Remark 5.2 we have $\{A_n\}_n \sim_{\text{GLT}} \kappa$ and $\{A_n\}_n \sim_{\text{GLT}} \kappa^*$. Thus, by Proposition 5.1, $\kappa = \kappa^*$ a.e. □

Theorem 5.3 *If $\{A_n\}_n \sim_{\text{GLT}} \kappa$ and the A_n are Hermitian then $\{A_n\}_n \sim_\lambda \kappa$.*

Proof By definition, for every $m \in \mathbb{N}$ there exist block LT sequences $\{A_{n,m}^{(i,j)}\}_n \sim_{\text{LT}} a_m^{(i,j)}(x) \circ f_m^{(i,j)}(\theta)$, $i = 1, \ldots, N_m$, $j = 1, \ldots, M_{m,i}$, such that:

- $\sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} (a_m^{(i,j)}(x) \circ f_m^{(i,j)}(\theta)) \to \kappa(x,\theta)$ in measure;
- $\left\{ \sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} A_{n,m}^{(i,j)} \right\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$.

Thus:

- $\left\{ \Re\left(\sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} A_{n,m}^{(i,j)} \right) \right\}_n \xrightarrow{\text{a.c.s.}} \{\Re(A_n)\}_n$ by **ACS 3**;
- $\left\{ \Re\left(\sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} A_{n,m}^{(i,j)} \right) \right\}_n \sim_\lambda \Re\left(\sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} (a_m^{(i,j)}(x) \circ f_m^{(i,j)}(\theta)) \right)$ by **LT 1**;
- $\Re\left(\sum_{i=1}^{N_m} \prod_{j=1}^{M_{m,i}} (a_m^{(i,j)}(x) \circ f_m^{(i,j)}(\theta)) \right) \to \Re(\kappa(x,\theta))$ in measure.

We conclude that $\{\Re(A_n)\}_n \sim_\lambda \Re(\kappa)$ by **ACS 2**. Since the matrices A_n are Hermitian, we have $\Re(A_n) = A_n$ and $\Re(\kappa) = \kappa$ a.e. by Proposition 5.2. Hence, the spectral distribution $\{\Re(A_n)\}_n \sim_\lambda \Re(\kappa)$ yields $\{A_n\}_n \sim_\lambda \kappa$. □

5.3 The GLT Algebra

The next theorems are of fundamental importance. In particular, the first one shows that the set of block GLT pairs $\mathcal{G}$ is a *-subalgebra of $\mathscr{E} \times \mathfrak{M}$.

Theorem 5.4 *Let $\{A_n\}_n \sim_{\text{GLT}} \kappa$ and $\{B_n\}_n \sim_{\text{GLT}} \xi$. Then:*

1. $\{A_n^*\}_n \sim_{\text{GLT}} \kappa^*$;
2. $\{\alpha A_n + \beta B_n\}_n \sim_{\text{GLT}} \alpha\kappa + \beta\xi$ *for all* $\alpha, \beta \in \mathbb{C}$;
3. $\{A_n B_n\}_n \sim_{\text{GLT}} \kappa\xi$.

Proof The first two statements have already been settled before (see Remark 5.2). We prove the third one. By definition, there exist $(\{A_{n,m}\}_n, \kappa_m), (\{B_{n,m}\}_n, \xi_m) \in \mathcal{A}$ such that $(\{A_{n,m}\}_n, \kappa_m) \to (\{A_n\}_n, \kappa)$ and $(\{B_{n,m}\}_n, \xi_m) \to (\{B_n\}_n, \xi)$ in the space $(\mathscr{E} \times \mathfrak{M}, d_{\text{a.c.s.}} \times d_{\text{measure}})$, i.e.:

- $\{A_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ and $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{B_n\}_n$;
- $\kappa_m \to \kappa$ in measure and $\xi_m \to \xi$ in measure.

Considering that every block GLT sequence is s.u. (see Remark 5.3), from **ACS 3** and Lemma 2.2 we obtain:

- $\{A_{n,m}B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n B_n\}_n$;
- $\kappa_m\xi_m \to \kappa\xi$ in measure.

Since $(\{A_{n,m}B_{n,m}\}_n, \kappa_m\xi_m) \in \mathcal{A}$, by definition we have $\{A_n B_n\}_n \sim_{\text{GLT}} \kappa\xi$. □

Lemma 5.1 *Let $\kappa : [0,1] \times [-\pi,\pi] \to \mathbb{C}^{s\times s}$ be any measurable function. Then, there exists a sequence of block GLT pairs $(\{A_{n,m}\}_m, \kappa_m)$ such that $\kappa_m \to \kappa$ in measure.*

Proof By [24, Lemma 2.8], there exists a sequence of measurable functions $\kappa_m : [0,1] \times [-\pi,\pi] \to \mathbb{C}^{s\times s}$ such that κ_m is of the form

$$\kappa_m(x,\theta) = \sum_{j=-N_m}^{N_m} a_{j,m}(x)\mathrm{e}^{\mathrm{i}j\theta}, \quad a_{j,m} \in C^\infty([0,1]), \quad N_m \in \mathbb{N},$$

and $\kappa_m \to \kappa$ a.e. (and hence also in measure). Take

$$A_{n,m} = \sum_{j=-N_m}^{N_m} D_n(a_{j,m})T_n(I_s\mathrm{e}^{\mathrm{i}j\theta})$$

and note that $\{A_{n,m}\}_n \sim_{\text{GLT}} \kappa_m$ by **LT 2** and Remark 5.1. □

Theorem 5.5 *If* $\{A_n\}_n \sim_{\text{GLT}} \kappa$ *and* κ *is invertible a.e. then* $\{A_n^\dagger\}_n \sim_{\text{GLT}} \kappa^{-1}$.

Proof Take any sequence of block GLT pairs $(\{B_{n,m}\}_n, \xi_m)$ such that $\xi_m \to \kappa^{-1}$ in measure. Note that such a sequence exists by Lemma 5.1. We show that

$$\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n^\dagger\}_n. \tag{10}$$

Once this is done, the thesis follows from Theorem 5.1. By Theorem 5.4 we have $\{B_{n,m}A_n - I_{sn}\}_n \sim_{\text{GLT}} \xi_m\kappa - I_s$, which implies that $\{B_{n,m}A_n - I_{sn}\}_n \sim_\sigma \xi_m\kappa - I_s$ by Theorem 5.2. Moreover, $\xi_m\kappa - I_s \to O_s$ in measure by Lemma 2.2 and so, by **ACS 4**,

$$\{B_{n,m}A_n\}_n \xrightarrow{\text{a.c.s.}} \{I_{sn}\}_n.$$

Since κ is invertible a.e. by hypothesis, $\{A_n\}_n$ is s.v. by Theorem 5.2 and Proposition 2.3. It follows that $A_n^\dagger$ is s.u. (see Remark 2.1) and hence, by **ACS 3**,

$$\{B_{n,m}A_nA_n^\dagger\}_n \xrightarrow{\text{a.c.s.}} \{A_n^\dagger\}_n. \tag{11}$$

Now we observe that, by definition of $A_n^\dagger$,

$$A_nA_n^\dagger = I_{sn} + S_n, \quad \operatorname{rank}(S_n) = \#\{i \in \{1,\dots,sn\} : \sigma_i(A_n) = 0\}.$$

Considering that $\{A_n\}_n$ is s.v., we have

$$\lim_{n\to\infty} \frac{\operatorname{rank}(S_n)}{n} = 0.$$

Thus, from (11) we obtain

$$\{B_{n,m} + Z_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n^\dagger\}_n, \tag{12}$$

where $Z_{n,m} = B_{n,m}S_n$ is such that, for every m, $\lim_{n\to\infty}(\operatorname{rank}(Z_{n,m})/n) = 0$. It follows that $\{Z_{n,m}\}_n$ is zero-distributed for every m by Theorem 2.1, and so (12) and (7) immediately imply (10). □

6 Summary of the Theory

A block GLT sequence is a special matrix-sequence $\{A_n\}_n$ equipped with a measurable function $\kappa : [0,1] \times [-\pi,\pi] \to \mathbb{C}^{s\times s}$, the so-called symbol. The notation $\{A_n\}_n \sim_{\text{GLT}} \kappa$ is used to indicate that $\{A_n\}_n$ is a block GLT sequence with symbol κ. The symbol of a block GLT sequence is unique in the sense that if $\{A_n\}_n \sim_{\text{GLT}} \kappa$ and $\{A_n\}_n \sim_{\text{GLT}} \xi$ then $\kappa = \xi$ a.e. in $[0,1] \times [-\pi,\pi]$. The

main properties of block GLT sequences proved in this paper are summarized in the following list.

GLT 1. If $\{A_n\}_n \sim_{\text{GLT}} \kappa$ then $\{A_n\}_n \sim_\sigma \kappa$. If moreover the matrices A_n are Hermitian then $\{A_n\}_n \sim_\lambda \kappa$.

GLT 2. We have:

- $\{T_n(f)\}_n \sim_{\text{GLT}} \kappa(x,\theta) = f(\theta)$ if $f \in L^1([-\pi,\pi],s)$;
- $\{D_n(a)\}_n \sim_{\text{GLT}} \kappa(x,\theta) = a(x)$ if $a : [0,1] \to \mathbb{C}^{s\times s}$ is Riemann-integrable;
- $\{Z_n\}_n \sim_{\text{GLT}} \kappa(x,\theta) = O_s$ if and only if $\{Z_n\}_n \sim_\sigma 0$.

GLT 3. If $\{A_n\}_n \sim_{\text{GLT}} \kappa$ and $\{B_n\}_n \sim_{\text{GLT}} \xi$ then:

- $\{A_n^*\}_n \sim_{\text{GLT}} \kappa^*$;
- $\{\alpha A_n + \beta B_n\}_n \sim_{\text{GLT}} \alpha\kappa + \beta\xi$ for all $\alpha,\beta \in \mathbb{C}$;
- $\{A_n B_n\}_n \sim_{\text{GLT}} \kappa\xi$;
- $\{A_n^\dagger\}_n \sim_{\text{GLT}} \kappa^{-1}$ provided that κ is invertible a.e.

GLT 4. We have $\{A_n\}_n \sim_{\text{GLT}} \kappa$ if and only if there exist block GLT sequences $\{B_{n,m}\}_n \sim_{\text{GLT}} \kappa_m$ such that $\{B_{n,m}\}_n \xrightarrow{\text{a.c.s.}} \{A_n\}_n$ and $\kappa_m \to \kappa$ in measure.

7 Final Remarks

By comparing the way in which the theory of block GLT sequences has been developed in this paper with the way in which the theory of GLT sequences has been developed in [24, Chapter 8], one immediately realizes that the two ways are different (though, of course, the set of block GLT sequences defined herein coincides in the scalar case $s = 1$ with the set of GLT sequences defined in [24]). The reason of this difference resides in the fact that we here adopted a more appropriate definition than [24, Definition 8.1]. Even though the two definitions turn out to be equivalent in the scalar case $s = 1$, the one employed herein should be considered as the 'right' definition; in particular, it allows us to simplify the proofs of several important theorems (compare, e.g., Theorems 5.1 and 5.4 with [24, Theorems 8.4 and 8.8]). A future edition of [24] should take into account this issue by correcting the definition of GLT sequences; once this is done, any formal difference between the theories of GLT and block GLT sequences will be removed and the latter will be obtained from the former through a straightforward adaptation process, with the only difference that the latter will involve more technicalities (just as the theory of block LT sequences involves more technicalities than the theory of LT sequences; see the discussion in Section 5 of [30]).

We conclude this work by just mentioning some future lines of research. First of all, as pointed out in the introduction, a forthcoming paper [29] will illustrate several applications of the theory of block GLT sequences developed herein. After

that paper, it will be necessary to develop the multivariate version of the theory of block GLT sequences and also the theory of reduced GLT sequences, as explained in [24, Chapter 11].

Acknowledgements Carlo Garoni is a Marie-Curie fellow of the Italian INdAM under grant agreement PCOFUND-GA-2012-600198. The work of the authors has been supported by the INdAM GNCS (Gruppo Nazionale per il Calcolo Scientifico). The authors wish to thank Giovanni Barbarino for useful discussions.

References

1. Al-Fhaid, A.S., Serra-Capizzano, S., Sesana, D., Ullah, M.Z.: Singular-value (and eigenvalue) distribution and Krylov preconditioning of sequences of sampling matrices approximating integral operators. Numer. Linear Algebra Appl. **21**, 722–743 (2014)
2. Avram, F.: On bilinear forms in Gaussian random variables and Toeplitz matrices. Probab. Theory Related Fields **79**, 37–45 (1988)
3. Barbarino, G.: Equivalence between GLT sequences and measurable functions. Linear Algebra Appl. **529**, 397–412 (2017)
4. Beckermann, B., Serra-Capizzano, S.: On the asymptotic spectrum of finite element matrix sequences. SIAM J. Numer. Anal. **45**, 746–769 (2007)
5. Benedusi, P., Garoni, C., Krause, R., Li, X., Serra-Capizzano, S.: Space-time FE-DG discretization of the anisotropic diffusion equation in any dimension: the spectral symbol. SIAM J. Matrix Anal. Appl. **39**, 1383–1420 (2018)
6. Bertaccini, D., Donatelli, M., Durastante, F., Serra-Capizzano, S.: Optimizing a multigrid Runge-Kutta smoother for variable-coefficient convection-diffusion equations. Linear Algebra Appl. **533**, 507–535 (2017)
7. Bhatia, R.: Matrix Analysis. Springer, New York (1997)
8. Bini, D., Latouche, G., Meini, B.: Numerical Methods for Structured Markov Chains. Oxford University Press, Oxford (2005)
9. Bini, D., Iannazzo, B., Meini, B.: Numerical Solution of Algebraic Riccati Equations. SIAM, Philadelphia (2012)
10. Böttcher, A., Grudsky, S.M.: Toeplitz Matrices, Asymptotic Linear Algebra, and Functional Analysis. Birkhäuser, Basel (2000)
11. Böttcher, A., Grudsky, S.M.: Spectral Properties of Banded Toeplitz Matrices. SIAM, Philadelphia (2005)
12. Böttcher, A., Silbermann, B.: Introduction to Large Truncated Toeplitz Matrices. Springer, New York (1999)
13. Böttcher, A., Silbermann, B.: Analysis of Toeplitz Operators, 2nd edn. Springer, Berlin (2006)
14. Böttcher, A., Garoni, C., Serra-Capizzano, S.: Exploration of Toeplitz-like matrices with unbounded symbols is not a purely academic journey. Sb. Math. **208**, 1602–1627 (2017)
15. Del Prete, V., Di Benedetto, F., Donatelli, M., Serra-Capizzano, S.: Symbol approach in a signal-restoration problem involving block Toeplitz matrices. J. Comput. Appl. Math. **272**, 399–416 (2014)
16. Donatelli, M., Garoni, C., Mazza, M., Serra-Capizzano, S., Sesana, D.: Spectral behavior of preconditioned non-Hermitian multilevel block Toeplitz matrices with matrix-valued symbol. Appl. Math. Comput. **245**, 158–173 (2014)
17. Donatelli, M., Garoni, C., Manni, C., Serra-Capizzano, S., Speleers, H.: Spectral analysis and spectral symbol of matrices in isogeometric collocation methods. Math. Comput. **85**, 1639–1680 (2016)

18. Donatelli, M., Garoni, C., Mazza, M., Serra-Capizzano, S., Sesana, D.: Preconditioned HSS method for large multilevel block Toeplitz linear systems via the notion of matrix-valued symbol. Numer. Linear Algebra Appl. **23**, 83–119 (2016)
19. Donatelli, M., Mazza, M., Serra-Capizzano, S.: Spectral analysis and structure preserving preconditioners for fractional diffusion equations. J. Comput. Phys. **307**, 262–279 (2016)
20. Donatelli, M., Dorostkar, A., Mazza, M., Neytcheva, M., Serra-Capizzano, S.: Function-based block multigrid strategy for a two-dimensional linear elasticity-type problem. Comput. Math. Appl. **74**, 1015–1028 (2017)
21. Dumbser, M., Fambri, F., Furci, I., Mazza, M., Serra-Capizzano, S., Tavelli, M.: Staggered discontinuous Galerkin methods for the incompressible Navier-Stokes equations: spectral analysis and computational results. Numer. Linear Algebra Appl. **25**, e2151 (2018)
22. Fasino, D., Serra-Capizzano, S.: From Toeplitz matrix sequences to zero distribution of orthogonal polynomials. Contemp. Math. **323**, 329–340 (2003)
23. Garoni, C.: Spectral distribution of PDE discretization matrices from isogeometric analysis: the case of L^1 coefficients and non-regular geometry. J. Spectral Theory **8**, 297–313 (2018)
24. Garoni, C., Serra-Capizzano, S.: Generalized Locally Toeplitz Sequences: Theory and Applications, vol. I. Springer, Cham (2017)
25. Garoni, C., Serra-Capizzano, S.: Generalized Locally Toeplitz Sequences: Theory and Applications, vol. II. Springer, Cham (2018)
26. Garoni, C., Serra-Capizzano, S., Sesana, D.: Spectral analysis and spectral symbol of d-variate $\mathbb{Q}_p$ Lagrangian FEM stiffness matrices. SIAM J. Matrix Anal. Appl. **36**, 1100–1128 (2015)
27. Garoni, C., Manni, C., Serra-Capizzano, S., Sesana, D., Speleers, H.: Spectral analysis and spectral symbol of matrices in isogeometric Galerkin methods. Math. Comput. **86**, 1343–1373 (2017)
28. Garoni, C., Manni, C., Serra-Capizzano, S., Sesana, D., Speleers, H.: Lusin theorem, GLT sequences and matrix computations: an application to the spectral analysis of PDE discretization matrices. J. Math. Anal. Appl. **446**, 365–382 (2017)
29. Garoni, C., Mazza, M., Serra-Capizzano, S.: Block generalized locally Toeplitz sequences: from the theory to the applications. Axioms **7**, 49 (2018)
30. Garoni, C., Serra-Capizzano, S., Sesana, D.: Block locally Toeplitz sequences: construction and properties. In: Bini, D.A., et al. (eds.) Structured Matrices in Numerical Linear Algebra. Springer INdAM Series, vol. 30, pp. 25–58. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-04088-8_2
31. Garoni, C., Speleers, H., Ekström, S.-E., Reali, A., Serra-Capizzano, S., Hughes, T.J.R.: Symbol-based analysis of finite element and isogeometric B-spline discretizations of eigenvalue problems: exposition and review. Arch. Comput. Meth. Eng. (in press). https://doi.org/10.1007/s11831-018-9295-y
32. Grenander, U., Szegő, G.: Toeplitz Forms and Their Applications, 2nd edn. AMS Chelsea Publishing, New York (1984)
33. Parter, S.V.: On the distribution of the singular values of Toeplitz matrices. Linear Algebra Appl. **80**, 115–130 (1986)
34. Peters, J., Reif, U.: Subdivision Surfaces. Springer, Berlin (2008)
35. Roman, F., Manni, C., Speleers, H.: Spectral analysis of matrices in Galerkin methods based on generalized B-splines with high smoothness. Numer. Math. **135**, 169–216 (2017)
36. Royden, H.L., Fitzpatrick, P.M.: Real Analysis, 4th edn. Pearson Education Asia Limited and China Machine Press, Hong Kong (2010)
37. Salinelli, E., Serra-Capizzano, S., Sesana, D.: Eigenvalue-eigenvector structure of Schoenmakers–Coffey matrices via Toeplitz technology and applications. Linear Algebra Appl. **491**, 138–160 (2016)
38. Serra-Capizzano, S.: Distribution results on the algebra generated by Toeplitz sequences: a finite dimensional approach. Linear Algebra Appl. **328**, 121–130 (2001)
39. Serra-Capizzano, S.: Generalized locally Toeplitz sequences: spectral analysis and applications to discretized partial differential equations. Linear Algebra Appl. **366**, 371–402 (2003)

40. Serra-Capizzano, S.: The GLT class as a generalized Fourier analysis and applications. Linear Algebra Appl. **419**, 180–233 (2006)
41. Tilli, P.: A note on the spectral distribution of Toeplitz matrices. Linear Multilinear Algebra **45**, 147–159 (1998)
42. Tilli, P.: Locally Toeplitz sequences: spectral properties and applications. Linear Algebra Appl. **278**, 91–120 (1998)
43. Tyrtyshnikov, E.E.: A unifying approach to some old and new theorems on distribution and clustering. Linear Algebra Appl. **232**, 1–43 (1996)
44. Tyrtyshnikov, E.E., Zamarashkin, N.L.: Spectra of multilevel Toeplitz matrices: advanced theory via simple matrix relationships. Linear Algebra Appl. **270**, 15–27 (1998)
45. Van Assche, W.: Zero distribution of orthogonal polynomials with asymptotically periodic varying recurrence coefficients. In: Priezzhev, V.B., Spiridonov, V.P. (eds.), Self-Similar Systems, pp. 392–402. Joint Institute for Nuclear Research, Dubna (1999)
46. Zamarashkin, N.L., Tyrtyshnikov, E.E.: Distribution of eigenvalues and singular values of Toeplitz matrices under weakened conditions on the generating function. Sb. Math. **188**, 1191–1201 (1997)

On Matrix Subspaces with Trivial Quadratic Kernels

Alexey Tretyakov, Eugene Tyrtyshnikov, and Alexey Ustimenko

Abstract Some subspaces of real matrices of the same order may contain nonsingular matrices, some may not. We prove that if the maximal rank matrix in the given subspace with trivial quadratic kernel is symmetric, then it must be nonsingular. It immediately follows that any subspace of symmetric matrices with trivial quadratic kernel contains a nonsingular matrix. We present some particular cases when this holds true without the assumption about symmetry. Whether this remains valid in the general case of real nonsymmetric matrices we still do not know.

Keywords Matrix subspaces · Quadratic kernels

1 Introduction and Preliminaries

The topic of this paper has been naturally initiated by some problems arising when one considers a system of nonlinear equations with a singular Jacoby matrix and tries to modify a system in order to make this matrix nonsingular and then be able to apply the Newton method [1, 3]. In particular circumstances, the success of this enterprise is guaranteed when a certain linear subspace of symmetric matrices contains a nonsingular matrix [3].

A. Tretyakov
Faculty of Sciences, Siedlce University, Siedlce, Poland

System Research Institute, Polish Academy of Sciences, Warsaw, Poland

E. Tyrtyshnikov (✉)
Marchuk Institute of Numerical Mathematics of Russian Academy of Sciences, Moscow, Russia

Lomonosov Moscow State University, Moscow, Russia

Faculty of Sciences, Siedlce University, Siedlce, Poland

A. Ustimenko
Marchuk Institute of Numerical Mathematics of Russian Academy of Sciences, Moscow, Russia

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_4

Consider an arbitrary field $\mathbb{F}$ and $n \times n$ matrices with the entries from $\mathbb{F}$. *Quadratic kernel* of a matrix $A \in V_n = \mathbb{F}^{n\times n}$ is defined as a set of all vectors $x \in \mathbb{F}^n$ with the property

$$x^\top Ax = 0.$$

We denote this set by $\ker^2(A)$ and remark that

$$\ker^2(A) \supseteq \ker(A).$$

It might happen that the kernel is trivial (i.e. consists of a single zero vector) while the quadratic kernel is not. For instance, consider a diagonal matrix

$$A = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Now, let $\mathcal{V}$ be a subspace in V_n. Then its quadratic kernel $\ker^2 \mathcal{V}$ is defined as the intersection of quadratic kernels for all matrices in $\mathcal{V}$. Similarly, the subspace kernel $\ker \mathcal{V}$ is the intersection of all the matrix kernels. We are interested to know if a subspace contains a nonsingular matrix and to which extent this property is related to the triviality of the quadratic kernel of this subspace. Note that the kernel triviality $\ker \mathcal{V} = 0$ does not mean that $\mathcal{V}$ includes a nonsingular matrix, e.g. consider $\mathcal{V}$ as a linear span of

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

However, we propose and investigate the following

Conjecture If $\ker^2 \mathcal{V} = 0$ then $\mathcal{V}$ contains a nonsingular matrix.

Let P be a nonsingular matrix of order n. Then a set

$$P^\top \mathcal{V} P := \{P^\top A P \,:\, A \in \mathcal{V}\}$$

is as well a subspace that will be called a congruent (in more detail, P-congruent) subspace to $\mathcal{V}$. Now take any k from 1 to n. For a matrix $A \in V_n$, denote by $A_k \in \mathbb{F}^{k\times k}$ the leading submatrix located in the left upper corner of A. Denote by $\mathcal{V}_k \subseteq V_k = \mathbb{F}^{k\times k}$ the collection of all leading submatrices matrices A_k for all matrices $A \in \mathcal{V}$. The following are some simple but useful observations we base on in what follows.

Statement 1 *If* $\ker^2 \mathcal{V} = 0$, *then* $\ker^2 P^\top \mathcal{V} P = 0$ *for any* P*-congruent subspace.*

Statement 2 *If* $\ker^2 \mathcal{V} = 0$, *then* $\ker^2 \mathcal{V}_k = 0$ *for any* k.

The congruence transformation allows us to simplify the structure of matrices in a subspace, at least for some of them. Assume that $A \in \mathcal{V}$ is singular but has a nonsingular leading submatrix A_{n-1}:

$$A = \begin{bmatrix} A_{n-1} & u \\ v^\top & \alpha \end{bmatrix}. \tag{1}$$

Then the last row of A is a linear combination of the first rows, and setting

$$P^\top = \begin{bmatrix} I_{n-1} & 0 \\ -v^\top A_{n-1}^{-1} & 1 \end{bmatrix}, \tag{2}$$

we obtain

$$P^\top A P = \begin{bmatrix} A_{n-1} & u - A_{n-1} A_{n-1}^{-\top} v \\ 0 & 0 \end{bmatrix}. \tag{3}$$

From this equation we straightforwardly deduce the following.

Statement 3 *Let matrices A and P be defined by formulas (1) and (2), and assume that A is symmetric. Then*

$$P^\top A P = \begin{bmatrix} A_{n-1} & 0 \\ 0 & 0 \end{bmatrix}.$$

The next observation is crucial in the constructions proposed in [3].

Statement 4 *Let $\ker^2 \mathcal{V} = 0$ and k be any integer from 1 to n. Then there exists a matrix $A = [a_{ij}] \in \mathcal{V}$ with the entry $a_{kk} = 1$.*

Proof On the contrary, assume that $a_{kk} = 0$ for any matrix $A \in \mathcal{V}$. Let e_k signify the kth column of the identity matrix. Then

$$e_k^\top A e_k = 0$$

for any $A \in \mathcal{V}$, which contradicts the quadratic kernel triviality. □

Statement 5 *Assume that a subspace $\mathcal{V}$ of $n \times n$ matrices has trivial quadratic kernel and contains a singular matrix with nonsingular leading submatrix of order $n - 1$. Then there is a congruent subspace that contains matrices A and B of the following form:*

$$A = \begin{bmatrix} \hat{A} & p \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} \hat{B} & q \\ 0 & 1 \end{bmatrix}, \tag{4}$$

where

$$\operatorname{rank} \hat{A} \geq n - 2. \tag{5}$$

Proof Assume that $A \in \mathcal{V}$ is singular and its leading submatrix A_{n-1} is nonsingular, and, instead of $\mathcal{V}$, consider a P-congruent subspace, where P is of the form (2). Allowing for (3), among those congruent matrices one is of the form

$$\tilde{A} = \begin{bmatrix} \tilde{A}_{n-1} & p \\ 0 & 0 \end{bmatrix} \quad (\text{note that } \tilde{A}_{n-1} = A_{n-1}),$$

and, by Statement 4, there is another one, say $\tilde{B}$, of the form

$$\tilde{B} = \begin{bmatrix} \tilde{B}_{n-1} & x \\ y^\top & 1 \end{bmatrix}.$$

With a nonsingular matrix

$$Q = \begin{bmatrix} I_{n-1} & 0 \\ -y^\top & 1 \end{bmatrix}$$

we obtain

$$Q^\top \tilde{B} Q = \begin{bmatrix} \tilde{B}_{n-1} - xy^\top & x - y \\ 0 & 1 \end{bmatrix}.$$

By the same congruence, $\tilde{A}$ is transformed into

$$Q^\top \tilde{A} Q = \begin{bmatrix} A_{n-1} - py^\top & p \\ 0 & 0 \end{bmatrix}.$$

Thus, the existence of two matrices with required structure is established, the corresponding leading submatrices being

$$\hat{A} = \tilde{A}_{n-1} - py^\top, \quad \hat{B} = \tilde{B}_{n-1} - xy^\top$$

and

$$q = x - y.$$

□

The next theorem was proposed and proved in [3]. However, we give it here a bit different proof explicitly using all considered above preliminaries.

Theorem 1 ([3]) *If a matrix subspace has trivial quadratic kernel and consists only of symmetric matrices, then it contains a nonsingular matrix.*

Proof We prove the claim by induction in the matrix order n. The case $n = 1$ is trivial. If $n \geq 2$, then, by Statement 2, the subspace of all leading submatrices of order $n - 1$ has nontrivial quadratic kernel and, by the induction assumption, in the given subspace there is a matrix with nonsingular leading submatrix of order $n - 1$. If this matrix is itself nonsingular, then the claim is already proved. If not, we construct a congruent subspace containing matrices A and B of the form (4), according to Statement 5. Now, consider linear combinations

$$tA + B = \begin{bmatrix} t\hat{A} + \hat{B} & tp + q \\ 0 & 1 \end{bmatrix}.$$

By the symmetry and Statement 3, $p = 0$ and hence the block $\hat{A}$ is nonsingular. Thus, a function

$$f(t) = \det(tA + B) = \det(t\hat{A} + \hat{B}) = \frac{\det(tI_{n-1} + \hat{B}\hat{A}^{-1})}{\det(\hat{A})}$$

is a polynomial of degree n and therefore for at most n values of t the linear combinations as above can be singular. □

The symmetric case is important for applications in nonlinear analysis and optimization [1, 3].

2 Maximal Rank Consequences

Further on we assume that the field $\mathbb{F}$ is infinite, although most results are valid as well for finite but sufficiently large fields.

Consider now a subspace $\mathcal{V} \subseteq V_n$ that is allowed to include nonsingular matrices, and let A be a matrix of maximal rank among all matrices belonging to $\mathcal{V}$. We are going to prove the following result.

Theorem 2 *Suppose that $A \in \mathcal{V}$ is a maximal rank matrix in the subspace $\mathcal{V}$. If A is symmetric, then*

$$\ker A \subseteq \ker^2 \mathcal{V}. \tag{6}$$

In order to prove this theorem we will have recourse to some properties of matrix pencils. Besides A, take an arbitrary matrix $B \in \mathcal{V}$ and consider matrices $A + \lambda B$ as the ones over the field of rational functions of λ with coefficients from the field $\mathbb{F}$. Thus,

$$\mathcal{A} = A + \lambda B \in (\mathbb{F}(\lambda))^{n \times n}.$$

We may consider the rank and defect of $\mathcal{A}$ over the field $\mathbb{F}(\lambda)$. Denote them by

$$\operatorname{Rank}(\mathcal{A}) := \operatorname{rank}_{\mathbb{F}(\lambda)}\mathcal{A},$$

$$\operatorname{Def}(\mathcal{A}) := \operatorname{def}_{\mathbb{F}(\lambda)}\mathcal{A} = n - \operatorname{Rank}\mathcal{A}.$$

Lemma 1 *If A is of maximal rank in $\mathcal{V}$, then*

$$\operatorname{Rank}(A + \lambda B) = \operatorname{rank} A \tag{7}$$

for any $B \in \mathcal{V}$.

Proof Since A is of maximal rank, for any $t \in \mathbb{F}$ we have the inequality

$$\operatorname{rank}(A + tB) \leq \operatorname{rank} A.$$

A minor of $\mathcal{A} = A + \lambda B$ is a polynomial in λ. If it is nonzero, then for some t the corresponding minor of $A + tB$ is a nonzero element of the field $\mathbb{F}$, provided that the number of elements in $\mathbb{F}$ exceeds the degree of this polynomial. This is the very place where we demand of the field to be sufficiently large. Consequently,

$$\operatorname{Rank}(A + \lambda B) \leq \max_{t\in\mathbb{F}} \operatorname{rank}(A + tB) \leq \operatorname{rank} A.$$

Moreover, if a minor in $\mathcal{A} = A+\lambda B$ is zero, then the corresponding minor in $A+tB$ is zero for any $t \in \mathbb{F}$, and hence, for an arbitrary field concerning this particular fact,

$$\operatorname{rank} A \leq \max_{t\in\mathbb{F}} \operatorname{rank}(A + tB) \leq \operatorname{Rank}(A + \lambda B).$$

All in all, we come up with the equality

$$\operatorname{Rank}(A + \lambda B) = \max_{t\in\mathbb{F}} \operatorname{rank}(A + tB) = \operatorname{rank} A.$$

□

Now consider the polynomial ring $K = \mathbb{F}[\lambda]$ and the set K^n of all vectors with n elements all belonging to K. Clearly, K^n is a finitely generated free module over K (also called K-module). Denote by M the set of all vectors $X(\lambda) \in K^n$ satisfying the equation

$$(A + \lambda B)X(\lambda) = 0.$$

As is readily seen, M is a submodule in K^n, and since K is the ring of principal ideals, M is a finitely generated free module as well.

Lemma 2 *A basis of M consists of $m = n - r$ vectors, where $r = \operatorname{rank} A$. Written in a polynomial form*

$$X_1(\lambda) = \sum_{i=0}^{d_1} x_{i1}\lambda^i, \quad \ldots, \quad X_m(\lambda) = \sum_{i=0}^{d_m} x_{im}\lambda^i, \tag{8}$$

where

$$x_{ij} \in \mathbb{F}^n,$$

this basis can be chosen in such a way that the free-term vectors $x_{01}, \ldots, x_{0m}$ comprise a basis of the kernel of A.

Proof From the theory of λ-matrices and recalling constructions related with the Smith form it emanates, that the matrix $A + \lambda B$ is diagonalized by some unimodular λ-matrices $U, V \in K^{n\times n}$ (for example, see [4]):

$$U(A + \lambda B)V = D(\lambda) = \begin{bmatrix} D_r(\lambda) & 0 \\ 0 & 0_{m\times m} \end{bmatrix},$$

where $D_r(\lambda) \in K^{r\times r}$ is a diagonal λ-matrix with nonzero polynomials on the diagonal. Obviously,

$$\operatorname{Rank}(A + \lambda B) = \operatorname{Rank} D(\lambda) = r,$$

and the equation

$$(A + \lambda B)X(\lambda) = 0$$

is equivalent to

$$D(\lambda)Y(\lambda) = 0, \quad X(\lambda) = VY(\lambda).$$

Thus, the basis with m vectors for M is directly constructed from the basis with m vectors for the kernel of $D(\lambda)$.

Now, let us assume that the basis (8) is selected to provide the minimal possible value for the sum $d = d_1 + \ldots + d_m$. Such a basis obviously exists and is called a *minimal basis* [2]. If the free-term vectors are linearly dependent, then at least one of them, say x_{0k}, is clearly expressed through the others. Upon the substraction from $X_k(\lambda)$ the corresponding linear combination of vectors $X_l(\lambda)$ for $l \neq k$, we arrive at a new system that remains a basis. All components of the kth vector of this new basis can be devided by λ. When this is done, we get a vector from K^n which can replace the previous one in the basis of the module M. In the result we obtain a new basis in which the sum of degrees is less than d. □

Other useful properties of minimal bases are established in [2]. In particular, it was proved therein that the senior vector coefficients are linearly independent. In addition to this, we have proved above the linear independence of the free term vector coefficients.

Proof of Theorem 2 Let $x_0 \in \ker A$. Then, in line with Lemma 2, x_0 is a linear combination of the vectors $x_{01}, \ldots, x_{0m}$, and therefore there exists a vector

$$X(\lambda) = x_0 + x_1\lambda + \ldots + x_s\lambda^s$$

such that

$$(A + \lambda B)X(\lambda) = Ax_0 + (Ax_1 + Bx_0)\lambda + \ldots = 0.$$

Hence, $Ax_1 + Bx_0 = 0$ and, consequently,

$$x_0^\top Ax_1 + x_0^\top Bx_0 = 0.$$

If A is symmetric, then

$$x_0^\top Ax_1 = (Ax_0)^\top x_1 = 0,$$

from which it directly stems that

$$x_0^\top Bx_0 = 0$$

for any $B \in \mathcal{V}$. Therefore, $x_0 \in \ker^2 \mathcal{V}$. □

Remark 1 In fact, without any reference to the symmetry of the maximal rank matrix A, we proved the inclusion

$$\ker A \cap \ker A^\top \subseteq \ker^2 \mathcal{V}.$$

Remark 2 Theorem 1 can be obtained as a direct corollary of Theorem 2.

3 Particular Cases Without Symmetry

Here we collect our attempts to remove the symmetry assumption from Theorem 1.

Theorem 3 *Let $\mathcal{V} \subseteq V_2$ be an arbitrary subspace with trivial quadratic kernel. Then $\mathcal{V}$ contains a nonsingular matrix.*

Proof On the contrary, suppose that all matrices in $\mathcal{V}$ are singular. In accordance with Statements 4 and 5, there is a congruent subspace to $\mathcal{V}$ in which we can find matrices of the following structure:

$$A = \begin{bmatrix} a & p \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b & q \\ 0 & 1 \end{bmatrix}.$$

If $a \neq 0$ or $b \neq 0$, a nonsingular matrix appears easily as a linear combination of A and B. Assume that $a = b = 0$. Then $p \neq 0$, and we readily conclude that the subspace contains two linearly independent matrices

$$U = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad V = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

If $\dim \mathcal{V} = 2$, then $\mathcal{V}$ is the span of these two matrices, and in this case

$$e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \ker \mathcal{V} \subseteq \ker^2 \mathcal{V},$$

which contradicts the triviality of quadratic kernel. Consequently, $\mathcal{V}$ must contain a nonzero matrix of the form

$$C = \begin{bmatrix} c & 0 \\ d & 0 \end{bmatrix}.$$

In this case the determinant

$$f(x, y) := \det(C + xU + yV) = cy - dx$$

is a nonzero polynomial of x and y, and hence, for some values of x and y we have $f(x, y) \neq 0$. □

Theorem 4 *Let $\mathcal{V} \subseteq V_n$ be a subspace with trivial quadratic kernel and* $\dim \mathcal{V} = 2$. *Then $\mathcal{V}$ contains a nonsingular matrix.*

Proof We will prove the claim by induction in n. First of all, let us acknowledge that the case $n = 2$ is covered by Theorem 3. Now take any $n \geq 3$.

Note that the dimension is the same for all congruent subspaces. Since the matrices A and B of the form (4) are linearly independent, they form a basis in the corresponding congruent subspace. Thus, the subspace of all linear combinations $\alpha A + \beta B$ possesses trivial quadratic kernel, and hence, the same holds true for the linear combinations $\alpha \hat{A} + \beta \hat{B}$. By the induction assumption, among the latter combinations we can pick up a nonsingular matrix. Let α and β be such that

$$\det(\alpha \hat{A} + \beta \hat{B}) \neq 0.$$

Thus, the corresponding linear combination of A and B is an upper triangular matrix of the form

$$\alpha A + \beta B = \begin{bmatrix} \alpha\hat{A} + \beta\hat{B} & \alpha p + \beta q \\ 0 & \beta \end{bmatrix}.$$

If $\beta = 0$, then $\hat{A}$ is a nonsingular block, and thence $t\hat{A} + \hat{B}$ is a nonsingular matrix for all t save for at most n values, and the same is valid for $tA + B$. If $\beta \neq 0$, then

$$\det(\alpha A + \beta B) = \beta \det(\alpha\hat{A} + \beta\hat{B}) \neq 0.$$

□

Acknowledgements This work is supported by the Program of the Presidium of the Russian Academy of Sciences no. 01 "Fundamental Mathematics and its Applications" under grant PRAS-18-01.

References

1. Brezhneva, O.A., Tretyakov, A.A.: On the choice of a method for solving a general system of nonlinear equations. Comput. Math. Math. Phys. **41**(5), 633–637 (2001)
2. Forney, G.D.: Minimal bases of rational vector space, with applications to multivariable linear systems. SIAM J. Control **13**(3), 493–420 (1975)
3. Tretyakov, A., Tyrtyshnikov, E., Ustimenko, A.: The triviality condition for kernels of quadratic mappings and its application in optimization methods. Russ. J. Numer. Anal. Math. Model. **32**(4), 1–9 (2017)
4. Tyrtyshnikov, E.E.: Foundations of Algebra. FIZMATLIT, Moscow (2017, in Russian)

Error Analysis of TT-Format Tensor Algorithms

Dario Fasino and Eugene E. Tyrtyshnikov

Abstract The tensor train (TT) decomposition is a representation technique for arbitrary tensors, which allows efficient storage and computations. For a d-dimensional tensor with $d \geq 2$, that decomposition consists of two ordinary matrices and $d - 2$ third-order tensors. In this paper we prove that the TT decomposition of an arbitrary tensor can be computed (or approximated, for data compression purposes) by means of a backward stable algorithm based on computations with Householder matrices. Moreover, multilinear forms with tensors represented in TT format can be computed efficiently with a small backward error.

Keywords TT-format · Backward stability · Tensor compression · Multilinear algebra

1 Introduction

The *tensor train decomposition* is a representation technique which allows compact storage and efficient computations with arbitrary tensors. The origins of this representation can be traced back to a brief paper by Oseledets and Tyrtyshnikov dating 2009 [9], while its popularization is mainly due to the subsequent papers [7, 10]. Nowadays, the tensor train decomposition is a computationally powerful tool that offers viable and convenient alternatives to classical (e.g., Tucker, CP) tensor representations [2, 5], in particular for the approximation of solutions of high dimensional problems. As shown in, e.g., [1, 6, 8], certain computations with large-scale structured matrices and vectors can be conveniently recast in terms of tensor train representations.

D. Fasino (✉)
Department of Mathematics, Computer Science and Physics, University of Udine, Udine, Italy
e-mail: dario.fasino@uniud.it

E. E. Tyrtyshnikov
Institute of Numerical Mathematics of Russian Academy of Sciences, Moscow, Russia

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_5

Basically, a tensor train (TT) decomposition of a d-dimensional tensor A with size $n_1 \times n_2 \times \cdots \times n_d$ is a sequence $\mathsf{G}_1, \ldots, \mathsf{G}_d$ of tensors of order 2 or 3; the size of G_i is $r_{i-1} \times n_i \times r_i$ with $r_0 = r_d = 1$ (that is, G_1 and G_d are ordinary matrices) and

$$\mathsf{A}(i_1, i_2, \ldots, i_d) = \sum_{j_1=1}^{r_1} \cdots \sum_{j_{d-1}=1}^{r_{d-1}} \mathsf{G}_1(i_1, j_1)\mathsf{G}_2(j_1, i_2, j_2) \cdots \mathsf{G}_d(j_{d-1}, i_d). \quad (1)$$

We will denote by $\mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$ the tensor identified by the right hand side of (1). Conditioning and numerical properties of the representation map $\mathrm{TT} : (\mathsf{G}_1, \ldots, \mathsf{G}_d) \mapsto \mathsf{A}$ are examined in [1].

In this paper, we present a backward error analysis of two algorithms, originally devised in [10], which perform computations with tensors in TT-format. The first algorithm produces an exact or approximate TT decomposition $\mathsf{G}_1, \ldots, \mathsf{G}_d$ of an arbitrary d-dimensional tensor A given in functional form, depending on a tolerance parameter ε. If $\varepsilon = 0$ then the output of the algorithm is an exact TT decomposition, that is, $\mathsf{A} = \mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$. If $\varepsilon > 0$ then $\mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$ is an $\mathscr{O}(\varepsilon)$-approximation of A which can realize significant savings in memory space. The computational core of the algorithm is a suitable (approximate) matrix factorization that, in the original paper, relies on SVD computations. We prove that analogous performances and backward stability can be obtained by means of QR factorizations based on Householder transformations.

The second algorithm computes the *contraction* (i.e., multilinear form) of a given d-dimensional tensor A and vectors $v^{(1)}, \ldots, v^{(d)}$,

$$\alpha = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \mathsf{A}(i_1, i_2, \ldots, i_d) v_{i_1}^{(1)} \cdots v_{i_d}^{(d)}, \quad (2)$$

where A is known in TT format. By means of known error bounds for inner products in floating point arithmetic [4], we prove backward stability of the proposed algorithm under very general hypotheses on the evaluation order of the summations. More precisely, if $\mathsf{A} = \mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$ and no underflows or overflows are encountered then the output computed by the algorithm in floating point arithmetic is the exact contraction of $\widehat{\mathsf{A}} = \mathrm{TT}(\mathsf{G}_1 + \Delta\mathsf{G}_1, \ldots, \mathsf{G}_d + \Delta\mathsf{G}_d)$ and $v^{(1)}, \ldots, v^{(d)}$ where $|\Delta\mathsf{G}_i| \le (n_i + r_{i-1})u|G_i| + \mathscr{O}(u^2)$ and u is the machine precision.

After setting up some basic notations and concepts, in Sect. 3 we present the algorithm for computing the TT-representation of a tensor and analyze its numerical stability. Next, we discuss in Sect. 4 the computation in computer arithmetic of the multilinear form (2) with a tensor in TT-format. A final appendix contains a complementary result.

2 Notations and Preliminaries

We refer to [2, Ch. 12] and [5] for notations and fundamental concepts on tensors and basic multilinear operations. Vectors and matrices are denoted by lower-case and upper-case italic letters, respectively, and higher order tensors by upper-case sans-serif letters: x, X, and X. A tensor X of order d is a multiway array of size $n_1 \times n_2 \times \cdots \times n_d$, where n_k is the size of the kth dimension or mode. A vector is a first-order tensor, and a matrix is a second-order tensor. The $(i_1, i_2, \ldots, i_d)$th entry of $\mathsf{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ is denoted by $\mathsf{X}_{i_1,i_2,\ldots,i_d}$ or, alternatively, $\mathsf{X}(i_1, i_2, \ldots, i_d)$ to avoid long, multiple subscripts. Lower case greek letters denote real numbers.

Throughout this paper, the prototypical tensor is a d-dimensional array A with dimensions $n_1 \times n_2 \times \cdots \times n_d$. In particular, the scalar d is used exclusively for the order of A, and the scalars $n_1, n_2, \ldots, n_d$ are reserved for A's dimensions. The *size* of A is the number of entries, $N(\mathsf{A}) = n_1 n_2 \cdots n_d$. The *vectorization* of A is the vector $\mathrm{vec}(\mathsf{A}) \in \mathbb{R}^{N(\mathsf{A})}$ whose ith entry is the ith entry of A according to the lexicographic ordering of the indices. The Matlab-style function `reshape` is defined in terms of the vectorization operator as follows. Let $m_1, m_2, \ldots, m_k$ be integers such that $N(\mathsf{A}) = m_1 m_2 \cdots m_k$. Then,

$$\mathsf{B} = \texttt{reshape}(\mathsf{A}, [m_1, m_2, \ldots, m_k])$$

is the tensor $\mathsf{B} \in \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_k}$ such that $\mathrm{vec}(\mathsf{A}) = \mathrm{vec}(\mathsf{B})$. In particular, for $k = 1, \ldots, d-1$,

$$A_k = \texttt{reshape}(\mathsf{A}, \left[\textstyle\prod_{i=1}^{k} n_i, \prod_{i=k+1}^{d} n_i\right])$$

is the *k-th unfolding matrix* of A.

The *k-mode product* of a tensor $\mathsf{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ by a matrix $M \in \mathbb{R}^{n_k \times m}$, denoted by $\mathsf{A} \times_k M$, is an $(n_1 \times \cdots \times n_{k-1} \times m \times n_{k+1} \times \cdots \times n_d)$-tensor of which the entries are given by

$$(\mathsf{A} \times_k M)(i_1, \ldots, i_d) = \sum_{j=1}^{n_k} \mathsf{A}(i_1, \ldots, i_{k-1}, j, i_{k+1}, \ldots, i_d) M_{j,i_k}.$$

The latter definition extends trivially to the case where M is a vector, by treating it as a $n_k \times 1$ matrix. The k-mode product satisfies the property

$$(\mathsf{A} \times_i B) \times_j C = (\mathsf{A} \times_j C) \times_i B\ , \tag{3}$$

so that notations like $\mathsf{A} \times_i B \times_j C$ can be used without ambiguity.

If A is a vector, a matrix or a tensor, we denote by $|A|$ the componentwise absolute value of A. Inequalities between tensors hold componentwise. Finally, the

Frobenius inner product of two matrices $A, B \in \mathbb{R}^{m\times n}$ is $\langle A, B\rangle = \mathrm{trace}(A^T B) = \mathrm{vec}(A)^T \mathrm{vec}(B)$, and the associated matrix norm is $\|A\|_{\mathrm{F}} = \|\mathrm{vec}(A)\|_2$. The latter is extended in an obvious way to arbitrary tensors.

2.1 The Tensor Train Format for Multidimensional Arrays

A tensor train decomposition of $\mathsf{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is a sequence $\mathsf{G}_1, \ldots, \mathsf{G}_d$ of tensors, called *carriages* (or *cores* [7, 8]), such that the size of G_i is $r_{i-1} \times n_i \times r_i$ with $r_0 = r_d = 1$ (that is, G_1 and G_d can be understood as ordinary matrices) and fulfilling the identity (1), that is, $\mathsf{A} = \mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$.

Example 1 Let A be a $10\times 20\times 30\times 40$ tensor with TT-ranks 5, 6, 7. The TT-format of A is made of four carriages, whose dimensions are as follows:

G_1 : $1 \times 10 \times 5$
G_2 : $5 \times 20 \times 6$
G_3 : $6 \times 30 \times 7$
G_4 : $7 \times 40 \times 1$.

An alternative viewpoint on the decomposition (1) is

$$\mathsf{A}(i_1, i_2, \ldots, i_d) = G_1'(i_1)G_2'(i_2)\cdots G_d'(i_d),$$

where now $G_k'(i_k)$ is an $r_{k-1} \times r_k$ matrix depending on the integer parameter i_k. The numbers $r_1, \ldots, r_{d-1}$ are called *TT-ranks* of A. As shown in, e.g., [7, 10], r_k is bounded from below by $\mathrm{rank}(A_k)$. Moreover, a TT decomposition with $r_k = \mathrm{rank}(A_k)$ always exists and can be computed by the algorithm shown in [10], which is recalled in the next section. It is often the case that an exact or approximate TT-format of a given tensor yields considerable savings in terms of memory space with respect to other representation techniques.

Remark 1 It is sometimes convenient to assume that G_1 and G_d are not three-dimensional but two-dimensional with sizes $n_1 \times r_1$ and $r_{d-1} \times n_d$, respectively. For that reason, in what follows we will use indifferently the notations G_1 and G_1 to denote the first carriage.

3 Full-to-TT Compression

In this section we address backward stability properties of the compression algorithm from [10], which is recalled hereafter as Algorithm 3.1. This algorithm produces an exact or approximate TT decomposition $\mathsf{G}_1, \ldots, \mathsf{G}_d$ of a given tensor A with $d \geq 2$, depending on a tolerance ε. If $\varepsilon = 0$ then the output of the algorithm is an exact TT decomposition, that is $\mathsf{A} = \mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$ with $r_k = \mathrm{rank}(A_k)$

for $k = 1, \ldots, d-1$, see [10, Thm. 2.1]. If $\varepsilon > 0$ then $r_k \leq \operatorname{rank}(A_k)$ and $\mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$ is an $\mathscr{O}(\varepsilon)$-approximation of A which can realize significant savings in memory space with respect to the exact TT-decomposition.

As a basic compression device we suppose to have at our disposal a generic, black-box algorithm having the interface

$$[M, N, r] = \texttt{compress}(A, \varepsilon) \tag{4}$$

which, for any matrix $A \in \mathbb{R}^{m\times n}$ and tolerance $\varepsilon \geq 0$ returns an integer r and two matrices $M \in \mathbb{R}^{m\times r}$ and $N \in \mathbb{R}^{r\times n}$ such that

$$\operatorname{rank}(M) = \operatorname{rank}(N) = r, \qquad \|MN - A\|_{\mathrm{F}} \leq \varepsilon \|A\|_{\mathrm{F}} \,. \tag{5}$$

In particular, if $\varepsilon = 0$ then $A = MN$ is a rank decomposition of A. In [10] this algorithm is realized by means of a truncated SVD of A. In that case, one has

$$\|A - MN\|_{\mathrm{F}} = \min_{\operatorname{rank}(X)\leq r} \|A - X\|_{\mathrm{F}}, \qquad r = \min_{\|A-X\|_{\mathrm{F}}\leq\varepsilon\|A\|_{\mathrm{F}}} \operatorname{rank}(X).$$

Consequently, if $\varepsilon > 0$ then TT-ranks $r_1, \ldots, r_{d-1}$ computed by the resulting procedure are in some sense optimal. However, other rank-revealing factorizations can be usefully adopted for the purpose of computing the TT-format of (an approximation of) the given tensor.

Observe that Algorithm 3.1 is based on a finite iteration. Each iteration is entirely based on matrix computations. In particular, if the argument A is a matrix then the output consists of the two matrices computed by `compress`. In the pseudocode here below, the intermediate matrices $B_1, \ldots, B_{d-1}$ are introduced to improve readability; in a more implementation-oriented description, these variables can share the same name and memory space.

Algorithm 3.1 Full-to-TT compression, iterative version

Input: tensor A of size $n_1 \times n_2 \times \cdots \times n_d$ and local accuracy bound ε
Output: tensor carriages $\mathsf{G}_1, \ldots, \mathsf{G}_d$

function $[\mathsf{G}_1, \ldots, \mathsf{G}_d] = \textsc{Full\_to\_TT}(\mathsf{A}, \varepsilon)$
  $n := N(\mathsf{A})$
  $n_R := n/n_1$
  $A_1 := \texttt{reshape}(\mathsf{A}, [n_1, n_R])$
  $[\mathsf{G}_1, B_1, r_1] := \texttt{compress}(A_1, \varepsilon)$
  for $k = 2 \ldots d-1$ **do**
    $n_R := n_R/n_k$
    $B_{k-1} := \texttt{reshape}(B_{k-1}, [r_{k-1}n_k, n_R])$
    $[\mathsf{G}_k, B_k, r_k] := \texttt{compress}(B_{k-1}, \varepsilon)$
    $\mathsf{G}_k := \texttt{reshape}(\mathsf{G}_k, [r_{k-1}, n_k, r_k])$
  end for
  $\mathsf{G}_d := B_{d-1}$
end function

Example 2 Suppose that Algorithm 3.1 is applied to the fourth order tensor A in Example 1. Then Algorithm 3.1 performs three for-loops; the dimensions of the matrices B_1 and B_2 before and after being reshaped in step 8 are as follows:

B_1 : 5×24000 and 100×1200
B_2 : 6×1200 and 180×40

At the third iteration the matrix B_3 is 7×40 and is renamed G_4 without being reshaped.

Algorithm 3.2 is a recursive version of Algorithm 3.1, namely, the two algorithms compute the same function, but Algorithm 3.2 calls itself internally. Termination is ensured by the fact that the order of the tensor argument diminishes by one at each recursive call. In fact, if $d = 2$ then the sought decomposition is obtained immediately by one call to `compress`, as in the iterative version. When $d > 2$, the matrix B_1 computed in step 4 together with the carriage G_1 is the first unfolding matrix of the $(d-1)$-dimensional tensor B, whose TT decomposition is computed by the recursive call in step 9. In that step the carriage G_2 is obtained as a matrix of dimension $(r_1 n_2) \times r_2$. In the subsequent steps that matrix is reshaped into a tensor of dimension $r_1 \times n_2 \times r_2$ and the computation is complete.

Remark 2 Apart of reshaping, the matrix B_k computed in Algorithm 3.1 coincides with the matrix B_1 computed in the $(k-1)$-th recursive call of Algorithm 3.2.

In the subsequent analysis we assume that, in exact arithmetic, the function `compress` satisfies the following property: For any input matrix A, the output matrices M and N fulfil the identities

$$M^T M = I, \qquad M^T (A - MN) = O. \tag{6}$$

Algorithm 3.2 Full-to-TT compression, recursive version

Input: tensor A of size $n_1 \times n_2 \times \cdots \times n_d$ and local accuracy bound ε
Output: tensor carriages $\mathsf{G}_1, \ldots, \mathsf{G}_d$

1: **function** $[\mathsf{G}_1, \ldots, \mathsf{G}_d] = \text{FULL\_TO\_TT}(A, \varepsilon)$
2: $n := N(\mathsf{A})$
3: $A_1 := \texttt{reshape}(\mathsf{A}, [n_1, n/n_1])$
4: $[\mathsf{G}_1, B_1, r_1] := \texttt{compress}(A_1, \varepsilon)$
5: **If** $d = 2$ **then**
6: $\mathsf{G}_2 := B_1$
7: **else**
8: $\mathsf{B} := \texttt{reshape}(B_1, [r_1 n_2, n_3, \ldots, n_d])$ ▷ the order of B is $d-1$
9: $[\mathsf{G}_2, \ldots, \mathsf{G}_d] := \texttt{Full\_to\_TT}(\mathsf{B}, \varepsilon)$ ▷ recursive call
10: $r_2 := N(\mathsf{G}_2)/(r_1 n_2)$
11: $\mathsf{G}_2 := \texttt{reshape}(\mathsf{G}_2, [r_1, n_2, r_2])$ ▷ G_2 is tensorized
12: **end if**
13: **end function**

Equations (5) and (6) can be met if `compress` is obtained from a truncated SVD as in [10] or from a (truncated) QR factorization. Indeed, the equations in (6) are fulfilled if and only if the matrix A admits a factorization

$$A = (M\ M') \begin{pmatrix} N \\ N' \end{pmatrix}$$

where the matrix $Q = (M\ M')$ has orthonormal columns and N has full rank, in which case we have $\|A - MN\| = \|M'N'\| = \|N'\|$ and $\|N\| = \|M^T A\| \leq \|A\|$ in any unitarily invariant norm. The sufficiency of that condition is obvious. To prove necessity, consider the residual matrix $R = A - MN$ and let $R = M'N'$ be a factorization where the columns of M' are an orthonormal basis for the column space of R and N' has full rank. Since $M^T M'N' = O$ the columns of M' must belong to the kernel of M^T, hence the matrix $Q = (M\ M')$ has orthonormal columns.

The following theorem is a reworking of Theorem 2.2 from [10] to emphasize the role of the hypotheses placed on `compress`. A shorter proof is included for later reference.

Theorem 1 *Suppose that the function* `compress` *in* (4) *fulfills the conditions* (5) *and* (6). *Let* $\mathsf{T} = \mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$ *where the tensor carriages* $\mathsf{G}_1, \ldots, \mathsf{G}_d$ *are computed from Algorithm 3.2 in exact arithmetic. Then* $\|\mathsf{A} - \mathsf{T}\|_{\mathrm{F}} \leq \varepsilon\sqrt{d-1}\,\|\mathsf{A}\|_{\mathrm{F}}$.

Proof We proceed by induction on d. When $d = 2$ the claim follows immediately from (5). For $d > 2$, consider the matrices G_1 and B_1 computed in step 4. By construction, the first unfolding matrix of T admits the factorization $T_1 = G_1 U_1$, for some $r_1 \times (n_2 \cdots n_d)$ matrix U_1. Since $G_1^T(A_1 - G_1 B_1) = O$ and $G_1^T G_1 = I$ by hypothesis, we have

$$\begin{aligned} \|\mathsf{A} - \mathsf{T}\|_{\mathrm{F}}^2 &= \|A_1 - G_1 B_1 + G_1 B_1 - G_1 U_1\|_{\mathrm{F}}^2 \\ &= \|A_1 - G_1 B_1\|_{\mathrm{F}}^2 + \|G_1(B_1 - U_1)\|_{\mathrm{F}}^2 + 2\langle G_1^T(A_1 - G_1 B_1), B_1 - U_1\rangle \\ &= \|A_1 - G_1 B_1\|_{\mathrm{F}}^2 + \|B_1 - U_1\|_{\mathrm{F}}^2 \,. \end{aligned}$$

Consider the $(d-1)$-dimensional tensor

$$\mathsf{U} = \mathtt{reshape}(U_1, [r_1 n_2, n_3, \ldots, n_d]) \,,$$

whose first unfolding matrix is U_1. Then,

$$\|\mathsf{A} - \mathsf{T}\|_{\mathrm{F}}^2 \leq \varepsilon^2 \|\mathsf{A}\|_{\mathrm{F}}^2 + \|\mathsf{B} - \mathsf{U}\|_{\mathrm{F}}^2 \,,$$

where B is obtained in step 8 of Algorithm 3.2. By construction $\mathsf{U} = \mathrm{TT}(\mathsf{G}_2, \ldots, \mathsf{G}_d)$, that is, U is the tensor whose exact TT-decomposition is given

by the tensor carriages obtained from Algorithm 3.2 with input B. By inductive hypothesis we have

$$\|B_1 - U_1\|_{\mathrm{F}}^2 = \|\mathsf{B} - \mathsf{U}\|_{\mathrm{F}}^2 \leq (d-2)\varepsilon^2 \|\mathsf{B}\|_{\mathrm{F}}^2 .$$

Moreover,

$$\|\mathsf{B}\|_{\mathrm{F}} = \|B_1\|_{\mathrm{F}} = \|G_1^T A_1\|_{\mathrm{F}} \leq \|A_1\|_{\mathrm{F}} = \|\mathsf{A}\|_{\mathrm{F}} ,$$

and the claim follows. □

Remark 3 It is not difficult to prove that the hypothesis $M^T M = I$ in (6) can be replaced by $\|M\|_2 \|M^+\|_2 = 1$, where M^+ is the Moore–Penrose inverse of M, without affecting the claim of Theorem 1. In the proof, one has simply to replace $\|G_1(B_1 - U_1)\|_{\mathrm{F}} = \|B_1 - U_1\|_{\mathrm{F}}$ by $\|G_1(B_1 - U_1)\|_{\mathrm{F}} \leq \|G_1\|_2 \|B_1 - U_1\|_{\mathrm{F}}$ and $\|B_1\|_{\mathrm{F}} = \|G_1^T A_1\|_{\mathrm{F}} \leq \|A_1\|_{\mathrm{F}}$ by $\|B_1\|_{\mathrm{F}} = \|G_1^+ A_1\|_{\mathrm{F}} \leq \|G_1^+\|_2 \|A_1\|_{\mathrm{F}}$.

This fact allows to introduce scaling factors in the computed carriages, e.g., in order to balance their norms, with no changes in the error estimate.

3.1 Backward Stability Analysis

The forthcoming Theorem 2 provides a backward stability estimate for Algorithm 3.2 which, in the exact arithmetic case, reduces to the error estimate given in Theorem 1 and, in the floating point arithmetic case, outlines the effects of the tolerance ε, the loss of orthogonality of the matrix M in (4) and the numerical stability of the function `compress` on the backward error of the computed TT decomposition.

Actually, the following analysis is mainly aimed to deal with two issues arising in the practical usage of Algorithm 3.2:

1. We compute an "exact" ($\varepsilon = 0$) TT decomposition in computer arithmetics and we desire a bound on the backward error of the computed result.
2. We want to compute a "low rank" approximation ($\varepsilon > 0$) of the given tensor but the function `compress` does not meet the conditions (6) as it happens if, e.g., it is based not on (pivoted) QR but on a different rank-revealing factorization where the spectral conditioning of M is greater than 1, see e.g., [2, §5.4.5]. In that case, we would like to quantify to what extent the approximation bound in Theorem 1 is degraded.

These two issues can be tackled together by assuming that the function `compress` fulfills the following hypotheses in place of (6): For any $A \in \mathbb{R}^{m \times n}$ there exists an exact decomposition $A + \Delta A = \widehat{Q}\widehat{R}$ with

$$\widehat{Q} = (M \; M') \in \mathbb{R}^{m \times m}, \qquad \widehat{R} = \begin{pmatrix} N \\ N' \end{pmatrix} \in \mathbb{R}^{m \times n} , \tag{7}$$

such that $\|M'N'\|_F \leq \varepsilon \|A\|_F$ where ε is the user-specified tolerance; and there exist two functions $\eta_1 = \eta_1(m, n, r)$ and $\eta_2 = \eta_2(m, r)$ such that

$$\|\Delta A\|_F \leq \eta_1 \|A\|_F, \qquad \|Q - \widehat{Q}\|_2 \leq \eta_2 \tag{8}$$

for some orthogonal matrix $Q \in \mathbb{R}^{m\times m}$.

If computations are performed in the usual IEEE standard computer arithmetic with machine precision u, the existence of two functions η_1 and η_2 being $\mathcal{O}(u)$, fulfilling (8) and having a moderate growth in m, n, r can be inferred from known facts concerning the error analysis of Householder QR factorization [2, Ch. 5], [3, §19.3]. For example, if the matrix $\widehat{Q}$ in (7) is computed by means of $r \leq \min\{m, n\}$ Householder transformations then [3, pp. 359–361] brings in the estimates

$$\eta_1(m, n, r) = \mathcal{O}(mru), \qquad \eta_2(m, r) = \mathcal{O}(mr^{3/2}u).$$

These estimates are deemed as rather conservative, and practical experience suggests e.g., that η_1 is a linear polynomial in m and r, see [3, p. 368]. In any case, all quantities η_1 and η_2 occurring in what follows are assumed to be "sufficiently small" so that quadratic and higher order terms can be neglected. We stress that conditions (7) and (8) reduce to (6) when $\eta_1 = \eta_2 = 0$.

Lemma 1 *In the preceding notations and hypotheses, neglecting higher order terms in ε, η_1, and η_2 we have*

1. $\|A - MN\|_F \leq (\varepsilon + \eta_1)\|A\|_F$
2. $\|M^T(A - MN)\|_F \leq \eta_1 \|A\|_F$
3. $\|N\|_F \leq (1 + \eta_1 + \eta_2)\|A\|_F$.

Proof Firstly, we have

$$\|A - MN\|_F = \|A + \Delta A - \Delta A - MN\|_F = \|M'N' - \Delta A\|_F \leq (\varepsilon + \eta_1)\|A\|_F \,,$$

and the first part follows.

Let $Q, \widehat{Q} \in \mathbb{R}^{m\times m}$ be the matrices appearing in (7) and (8). Let $Q = (Q_1\ Q_2)$ and $\Delta Q = \widehat{Q} - Q = (\Delta_1\ \Delta_2)$ be partitioned consistently with $\widehat{Q}$ as in (7). Then, $\|\Delta_{1,2}\|_2 \leq \eta_2$ and

$$\begin{aligned}
\|M^T M'\|_2 &= \|(Q_1 + \Delta_1)^T (Q_2 + \Delta_2)\|_2 \\
&\leq \|\Delta_1^T Q_2\|_2 + \|Q_1^T \Delta_2\|_2 + \|\Delta_1^T \Delta_2\|_2 \\
&\leq \|\Delta_1\|_2 + \|\Delta_2\|_2 + \|\Delta_1^T \Delta_2\|_2 \leq \eta_2(2 + \eta_2) \,.
\end{aligned}$$

Neglecting quadratic terms we get $\|M^T M'\|_2 \lesssim 2\eta_2$, $\|N'\|_{\mathrm{F}} \lesssim \varepsilon\|A\|_{\mathrm{F}}$ and

$$\begin{aligned}\|M^T(A - MN)\|_{\mathrm{F}} &= \|M^T(M'N' - \Delta A)\|_{\mathrm{F}} \\ &\leq \|M^T M'\|_2\|N'\|_{\mathrm{F}} + \|M\|_2\|\Delta A\|_{\mathrm{F}} \lesssim (2\eta_2\varepsilon + \eta_1)\|A\|_{\mathrm{F}} .\end{aligned}$$

This proves the second inequality in the claim.

Since $\widehat{Q} = Q + \Delta Q$ and $\|\Delta Q\|_2 \leq \eta_2$, standard perturbation theory yields that the smallest singular value of $\widehat{Q}$ is not smaller than $1 - \eta_2$, see e.g., [2, Corollary 2.4.4]. Hence $\|\widehat{Q}^{-1}\|_2 \leq 1/(1 - \eta_2) \approx 1 + \eta_2$. Finally,

$$\begin{aligned}\|N\|_{\mathrm{F}} \leq \|\widehat{R}\|_{\mathrm{F}} &= \|\widehat{Q}^{-1}(A + \Delta A)\|_{\mathrm{F}} \\ &\leq \|\widehat{Q}^{-1}\|_2\|A + \Delta A\|_{\mathrm{F}} \lesssim (1 + \eta_1)(1 + \eta_2)\|A\|_{\mathrm{F}} ,\end{aligned}$$

and the proof is complete. □

Theorem 2 *Let* $\mathsf{T} = \mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$ *where the carriages* $\mathsf{G}_1, \ldots, \mathsf{G}_d$ *are computed under the aforementioned hypotheses. Let* η_1 *and* η_2 *denote the largest values of the like named coefficients occurring in all recursive calls of Algorithm 3.2 on the specific input* A*. Moreover, let* $\hat{\varepsilon} = \varepsilon + \eta_1$*. Neglecting higher order terms in* ε*,* η_1*, and* η_2*,*

$$\|\mathsf{A} - \mathsf{T}\|_{\mathrm{F}} \leq (1 + \eta_1 + 2\eta_2)^{d-2}\big((d-2)\eta_1 + \sqrt{d-1}\,\hat{\varepsilon}\big)\|\mathsf{A}\|_{\mathrm{F}} .$$

Proof We proceed by induction, as in the the proof of Theorem 1. The $d = 2$ case is an immediate consequence of Lemma 1(1):

$$\|\mathsf{A} - \mathsf{T}\|_{\mathrm{F}} = \|A_1 - G_1 B_1\|_{\mathrm{F}} \leq (\varepsilon + \eta_1)\|A_1\|_{\mathrm{F}} = \hat{\varepsilon}\|\mathsf{A}\|_{\mathrm{F}}.$$

For $d > 2$, the inductive argument begins similarly to the proof of Theorem 1. Indeed, by hypotheses and Lemma 1 we have $\|G_1\|_2 \leq 1 + \eta_2$ and

$$\begin{aligned}\|\mathsf{A} - \mathsf{T}\|_{\mathrm{F}}^2 &= \|A_1 - G_1 B_1 + G_1 B_1 - G_1 U_1\|_{\mathrm{F}}^2 \\ &= \|A_1 - G_1 B_1\|_{\mathrm{F}}^2 + \|G_1(B_1 - U_1)\|_{\mathrm{F}}^2 + 2\langle G_1^T(A_1 - G_1 B_1), B_1 - U_1\rangle \\ &\leq \hat{\varepsilon}^2\|\mathsf{A}\|_{\mathrm{F}}^2 + (1 + \eta_2)^2\|B_1 - U_1\|_{\mathrm{F}}^2 + 2\eta_1\|\mathsf{A}\|_{\mathrm{F}}\|B_1 - U_1\|_{\mathrm{F}} .\end{aligned}$$

The last inequality follows by Cauchy–Schwartz inequality and Lemma 1(2).

Let $\mathsf{A} = \mathsf{A}_d$ and $\mathsf{T} = \mathsf{T}_d$. For $k = 2, \ldots, d-1$, let $\mathsf{A}_k, \mathsf{T}_k$ be the k-dimensional tensors defined as follows: A_k and T_k are the argument and the result of the $(d-k)$-th recursive call of the algorithm,[1]

$$(\mathsf{G}_{d-k+1}, \ldots, \mathsf{G}_d) = \texttt{Full\_to\_TT}(\mathsf{A}_k, \varepsilon) , \qquad \mathsf{T}_k = \mathrm{TT}(\mathsf{G}_{d-k+1}, \ldots, \mathsf{G}_d) .$$

[1] With a little abuse of notation, in the ensuing equation we identify G_{d-k+1} with its first unfolding matrix.

For example, in the proof of Theorem 1 A_{d-1} is denoted by B and T_{d-1} by U. With these notations, the preceding inequality yields

$$\|\mathsf{A}_{k+1} - \mathsf{T}_{k+1}\|_{\mathrm{F}}^2 \le \hat{\varepsilon}^2 \|\mathsf{A}_{k+1}\|_{\mathrm{F}}^2 + (1+\eta_2)^2 \|\mathsf{A}_k - \mathsf{T}_k\|_{\mathrm{F}}^2 + 2\eta_1 \|\mathsf{A}_{k+1}\|_{\mathrm{F}} \|\mathsf{A}_k - \mathsf{T}_k\|_{\mathrm{F}} .$$

Let $\varrho = 1 + \eta_1 + \eta_2$. From Lemma 1(3) we have $\|\mathsf{A}_k\|_{\mathrm{F}} \le \varrho \|\mathsf{A}_{k+1}\|_{\mathrm{F}}$. For $k = 1, \ldots, d-1$ let $e_k = \|\mathsf{A}_{k+1} - \mathsf{T}_{k+1}\| / \|\mathsf{A}_{k+1}\|_{\mathrm{F}}$. Neglecting product terms in η_1, η_2 and other small quantities we arrive at the recurrence

$$\begin{aligned} e_k^2 &\le \hat{\varepsilon}^2 + \varrho^2 (1+\eta_2)^2 e_{k-1}^2 + 2\eta_1 \varrho e_{k-1} \\ &\le (\alpha e_{k-1} + \eta_1)^2 + \hat{\varepsilon}^2 \end{aligned}$$

where $\alpha = 1 + \eta_1 + 2\eta_2$ and $e_1 \le \hat{\varepsilon}$. From Lemma 5 in Appendix we obtain

$$e_k \le \alpha^{k-1} \big((k-1)\eta_1 + \sqrt{k}\, \hat{\varepsilon} \big) ,$$

and the claim follows by setting $k = d - 1$. □

It is worth noting that in the exact case (that is, when $\eta_1 = \eta_2 = 0$) the inequality in the previous claim reduces to that of Theorem 1.

4 Computing Multilinear Forms

The computation of the multilinear form (or contraction)

$$\alpha = \mathsf{A} \times_1 v^{(1)} \times_2 v^{(2)} \times_3 \cdots \times_d v^{(d)}$$

where $v^{(i)} \in \mathbb{R}^{n_i}$, occurs e.g., in the computation of multidimensional integrals on cartesian product grids [10]. If $\mathsf{A} = \mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$ then the preceding expression can be rewritten as

$$\alpha = \sum_{i_1, \ldots, i_d} \sum_{j_1, \ldots, j_{d-1}} \mathsf{G}_1(i_1, j_1) \mathsf{G}_2(j_1, i_2, j_2) \cdots \mathsf{G}_d(j_{d-1}, i_d) v_{i_1}^{(1)} \cdots v_{i_d}^{(d)} .$$

Assuming $n_1 = \ldots = n_d = n$ and $r_1 = \ldots = r_{d-1} = r$, the right hand side can be computed in $\mathcal{O}(dnr^2)$ floating point operations using Algorithm 3 from [10], which is described hereafter as Algorithm 4.1.

After completion of the k-th cycle of the for-loop, W is an $n_k \times r_k$ matrix and t is an r_k-vector. In particular, when $k = d$ step 4 is a matrix-vector multiplication as $r_d = 1$, so W becomes a vector and step 5 is an inner product of two n_d-vectors. Note that the computation of $W = \mathsf{G}_k \times_1 t$ followed by $t = W^T v^{(k)}$ yields a particular algorithm to compute $\mathsf{G}_k \times_1 t \times_2 v^{(k)}$ which can be implemented using $n_k r_k$ inner products with r_{k-1}-vectors followed by r_k inner products with

Algorithm 4.1 Fast TT contraction algorithm

Input: tensor train $\mathsf{A} = \mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$, vectors $v^{(1)}, \ldots, v^{(d)}$
Output: $\alpha = \mathsf{A} \times_1 v^{(1)} \times_2 \cdots \times_d v^{(d)}$
1: **function** $\alpha = \mathrm{TT\_CONTRACTION}(\mathsf{G}_1, \ldots, \mathsf{G}_d, v^{(1)}, \ldots, v^{(d)})$
2: $t := \mathsf{G}_1 \times_1 v^{(1)}$
3: **for** $k = 2 \ldots d$ **do**
4: $W := \mathsf{G}_k \times_1 t$
5: $t := W^T v^{(k)}$
6: **end for**
7: $\alpha := t$.
8: **end function**

n_k-vectors. Owing to the identity (3), an alternative algorithm for computing the same expression (with almost the same number of floating point operations) is $W := \mathsf{G}_k \times_2 v^{(k)}$ followed by $t := W^T t$.

In what follows, $\mathbb{F}$ denotes a set of (computer) floating point numbers endowed by the usual IEEE standard arithmetics, and u denotes the corresponding unit roundoff. Moreover, we use the notation $\mathrm{fl}(\cdot)$ with an argument that is an expression to denote the computed value of that expression. The next two lemmas are borrowed from [4].

Lemma 2 *Given $x, y \in \mathbb{F}^n$, any order of evaluation of the inner product $x^T y$ produces an approximation α such that $|\alpha - x^T y| \le nu|x|^T|y|$, if no underflows or overflows are encountered.*

Lemma 3 *Given $A \in \mathbb{F}^{m\times n}$ and $x \in \mathbb{F}^n$, let $\widehat{y} = \mathrm{fl}(Ax)$ be the approximation to Ax obtained by computing m inner products of n-vectors, each of them being performed in an arbitrary evaluation order. If no underflow or overflow occurs, then there exists a matrix $\widehat{A} \in \mathbb{R}^{m\times n}$ such that*

$$\widehat{y} = \widehat{A}x, \qquad \widehat{A}_{ij} = (1 + \varepsilon_{ij})A_{ij}, \qquad |\varepsilon_{ij}| \le nu \ .$$

On the basis of these two lemmas it is not hard to obtain the result hereafter.

Theorem 3 *Given $\mathsf{G} \in \mathbb{F}^{\ell\times m\times n}$, $x \in \mathbb{F}^\ell$, and $y \in \mathbb{F}^m$, let $\widehat{z} = \mathrm{fl}(\mathrm{fl}(\mathsf{G} \times_1 x) \times_2 y) \in \mathbb{F}^n$ be the finite precision approximation to $z = \mathsf{G} \times_1 x \times_2 y$ obtained after $mn + n$ inner products, each of them being performed in an arbitrary evaluation order. If no underflow or overflow occurs then there exists a tensor $\Delta\mathsf{G} \in \mathbb{R}^{\ell\times m\times n}$ such that*

$$\widehat{z} = (\mathsf{G} + \Delta\mathsf{G}) \times_1 x \times_2 y, \qquad |\Delta\mathsf{G}| \le (\ell + m)u|\mathsf{G}| + \mathscr{O}(u^2) \ .$$

Proof Introduce the auxiliary matrix $M = \mathsf{G} \times_1 x$ and let $\widehat{M} = \mathrm{fl}(\mathsf{G} \times_1 x)$ be its finite precision counterpart. In exact arithmetic,

$$M_{jk} = \sum_{i=1}^{\ell} \mathsf{G}_{ijk} x_i \ .$$

Owing to Lemma 2, for every j, k there exists ε_{jk} such that

$$\widehat{M}_{jk} - M_{jk} = \varepsilon_{jk} \sum_{i=1}^{\ell} |\mathsf{G}_{ijk} x_i| \,, \qquad |\varepsilon_{jk}| \le \ell u \,.$$

Letting $\eta_{ijk} = \mathrm{sign}(\mathsf{G}_{ijk} x_i)\varepsilon_{jk}$ we obtain

$$\begin{aligned} \widehat{M}_{jk} &= \sum_i \mathsf{G}_{ijk} x_i + \varepsilon_{jk} |\mathsf{G}_{ijk} x_i| \\ &= \sum_i (1 + \eta_{ijk}) \mathsf{G}_{ijk} x_j \,. \end{aligned}$$

Obviously $|\eta_{ijk}| \le \ell u$. By Lemma 3, for $k = 1, \ldots, n$ we have

$$\begin{aligned} \widehat{z}_k = [\mathrm{fl}(\widehat{M}^T y)]_k &= \sum_j (1 + \eta_{jk}) \widehat{M}_{jk} y_j \\ &= \sum_j \sum_i (1 + \eta_{jk})(1 + \eta_{ijk}) \mathsf{G}_{ijk} x_i y_j \end{aligned}$$

for some constants η_{jk} with $|\eta_{jk}| \le mu$. In conclusion, $\widehat{z} = (\mathsf{G} + \Delta\mathsf{G}) \times_1 x \times_2 y$ where

$$\Delta\mathsf{G}_{ijk} = \xi_{ijk} \mathsf{G}_{ijk} \,, \qquad |\xi_{ijk}| = |(1 + \eta_{jk})(1 + \eta_{ijk}) - 1| \le (\ell + m)u + \ell m u^2 \,,$$

and the proof is complete. □

Note that the previous theorem applies also when $\ell = 1$ or $n = 1$, where G reduces to a matrix. Recalling the conventional notation $r_0 = 1$, we obtain immediately the following consequence.

Corollary 1 *Let $\widehat{\alpha}$ be the result of Algorithm 4.1 computed in machine arithmetics from input $\mathsf{A} = \mathrm{TT}(\mathsf{G}_1, \ldots, \mathsf{G}_d)$. Then, there exist tensors $\Delta G_1, \ldots, \Delta G_d$ such that*

$$\widehat{\alpha} = \widehat{\mathsf{A}} \times_1 v^{(1)} \times_2 v^{(2)} \cdots \times_d v^{(d)}, \qquad \widehat{\mathsf{A}} = \mathrm{TT}(\mathsf{G}_1 + \Delta\mathsf{G}_1, \ldots, \mathsf{G}_d + \Delta\mathsf{G}_d) \,,$$

and $|\Delta\mathsf{G}_i| \le (n_i + r_{i-1})u|\mathsf{G}_i| + \mathscr{O}(u^2)$.

The previous result allows us to interpret the rounding errors in the computation of α as due to perturbations in the carriages $\mathsf{G}_1, \ldots, \mathsf{G}_d$, not in A. The forthcoming Theorem 4 provides a backward error bound in terms of a perturbation in A. To that goal, we need the following lemma whose proof derives from an elementary inductive argument and will not be shown.

Lemma 4 *Let $\xi_1, \dots \xi_k$ be numbers such that $s = \sum_{i=1}^{k} |\xi_i| < 1$. Then,*

$$\prod_{i=1}^{k}(1+\xi_i) = 1+\theta, \qquad |\theta| \le \frac{s}{1-s}\ .$$

In particular, if $s \le \frac{1}{2}$ then $|\theta| \le 2s$.

Theorem 4 *Let $\widehat{\alpha}$ be the result of Algorithm 4.1 computed in machine arithmetic from input $\mathsf{A} = \mathrm{TT}(\mathsf{G}_1, \dots, \mathsf{G}_d)$, and let $s = \sum_{i=1}^{d}(n_i + r_{i-1})u$. If $s \le \frac{1}{2}$ then*

$$\widehat{\alpha} = \widehat{\mathsf{A}} \times_1 v^{(1)} \times_2 v^{(2)} \cdots \times_d v^{(d)}$$

where $|\widehat{\mathsf{A}} - \mathsf{A}| \le 2s\,\mathrm{TT}(|\mathsf{G}_1|, \dots, |\mathsf{G}_d|) + \mathscr{O}(u^2)$.

Proof In what follows, we interpret $\mathsf{G}_1(i,j)$ and $\mathsf{G}_d(i,j)$ as $\mathsf{G}_1(1,i,j)$ and $\mathsf{G}_d(i,j,1)$, respectively. Let $\varepsilon_i = (n_i + r_{i-1})u$ for $i = 1, \dots, d$. By Corollary 1, there exist constants $\xi_{ijk}^{(\ell)}$ such that $\widehat{\alpha}$ is the exact contraction of the tensor $\mathrm{TT}(\mathsf{G}_1 + \Delta\mathsf{G}_1, \dots, \mathsf{G}_d + \Delta\mathsf{G}_d)$ and vectors $v^{(1)}, \dots, v^{(d)}$ where

$$\Delta\mathsf{G}_\ell(i,j,k) = \mathsf{G}_\ell(i,j,k)\xi_{ijk}^{(\ell)}, \qquad |\xi_{ijk}^{(\ell)}| \le \varepsilon_\ell + \mathscr{O}(u^2)\ .$$

Assuming $j_0 = j_d = 1$, for every multi-index $i = (i_1, i_2, \dots, i_d)$ we have

$$\begin{aligned}\widehat{\mathsf{A}}(i) &= \sum_{j_1,\dots,j_{d-1}} \mathsf{G}_1(i_1,j_1)\mathsf{G}_2(j_1,i_2,j_2)\cdots G_d(j_{d-1},i_d)\prod_{\ell=1}^{d}(1+\xi_{j_{\ell-1}i_\ell j_\ell}^{(\ell)}) \\ &= \sum_{j_1,\dots,j_{d-1}} \mathsf{G}_1(i_1,j_1)\mathsf{G}_2(j_1,i_2,j_2)\cdots \mathsf{G}_d(j_{d-1},i_d)(1+\theta_x)\end{aligned}$$

with $x = (i_1, \dots, i_d, j_1, \dots, j_{d-1})$ and $|\theta_x| \le 2s$ by Lemma 4. By triangle inequality,

$$|\widehat{\mathsf{A}}(i) - \mathsf{A}(i)| \le \sum_{j_1,\dots,j_{d-1}} |\mathsf{G}_1(i_1,j_1)||\mathsf{G}_2(j_1,i_2,j_2)|\cdots|\mathsf{G}_d(j_{d-1},i_d)||\theta_x|\ ,$$

and the claim follows. □

Appendix

Hereafter, we prove a technical lemma which yields an upper bound for the growth of a sequence occurring within the proof of Theorem 2.

Lemma 5 *Let $\{e_k\}$ be a sequence of nonnegative numbers such that $e_1 \le \gamma$ and for $k \ge 2$*

$$e_k^2 \le (\alpha e_{k-1} + \beta)^2 + \gamma^2$$

for some nonnegative constants α, β, γ with $\alpha \ge 1$. Then for all $k \ge 1$

$$e_k \le \alpha^{k-1}\big((k-1)\beta + \sqrt{k}\,\gamma\big)\,.$$

Proof Define the auxiliary notations $\hat{e}_k = \alpha^{1-k} e_k$, $\hat{\beta}_k = \alpha^{1-k}\beta$ and $\hat{\gamma}_k = \alpha^{1-k}\gamma$. Note that $\hat{\beta}_k \le \beta$ and $\hat{\gamma}_k \le \gamma$ for $k \ge 1$. Then,

$$\begin{aligned}\hat{e}_k^2 = \alpha^{2-2k} e_k^2 &\le (\alpha^{2-k} e_{k-1} + \alpha^{1-k}\beta)^2 + \alpha^{2-2k}\gamma^2 \\ &= (\hat{e}_{k-1} + \hat{\beta}_k)^2 + \hat{\gamma}_k^2\,.\end{aligned}$$

Firstly we prove that for all $k \ge 1$

$$\hat{e}_k \le \textstyle\sum_{j=2}^{k} \hat{\beta}_k + \sqrt{\sum_{j=1}^{k} \hat{\gamma}_j^2}\,.$$

Indeed, the claim is trivially verified when $k = 1$. By an inductive argument, for $k \ge 2$ we have

$$\begin{aligned}\hat{e}_k^2 &\le \Big(\textstyle\sum_{j=2}^{k} \hat{\beta}_j + \sqrt{\sum_{j=1}^{k-1} \hat{\gamma}_j^2}\Big)^2 + \hat{\gamma}_k^2 \\ &= \big(\textstyle\sum_{j=2}^{k} \hat{\beta}_j\big)^2 + \sum_{j=1}^{k} \hat{\gamma}_j^2 + 2\big(\sum_{j=2}^{k} \hat{\beta}_j\big)\sqrt{\sum_{j=1}^{k-1} \hat{\gamma}_j^2} \\ &\le \Big(\textstyle\sum_{j=2}^{k} \hat{\beta}_j + \sqrt{\sum_{j=1}^{k} \hat{\gamma}_j^2}\Big)^2.\end{aligned}$$

Going back to the sequence $\{e_k\}$ we have for all $k \ge 1$

$$\begin{aligned}e_k = \alpha^{k-1}\hat{e}_k &\le \alpha^{k-1}\Big(\textstyle\sum_{j=2}^{k} \hat{\beta}_j + \sqrt{\sum_{j=1}^{k} \hat{\gamma}_j^2}\Big) \\ &\le \alpha^{k-1}\big((k-1)\beta + \sqrt{k}\,\gamma\big)\end{aligned}$$

and we are done. □

Acknowledgements The first author acknowledges the support received by INDAM-GNCS, Italy, for his research. The work of the second author was supported by the Russian Scientific Foundation project 14-11-00806.

References

1. Bachmayr, M., Kazeev, V.: Stability of low-rank tensor representations and structured multilevel preconditioning for elliptic PDEs. ArXiv preprint (2018). http://arxiv.org/pdf/1802.09062.pdf
2. Golub, G.H., Van Loan, C.: Matrix Computations, 4th edn. The John Hopkins University Press, Baltimore (2013)
3. Higham, N.J.: Accuracy and Stability of Numerical Algorithms. SIAM, Philadelphia (2002)
4. Jeannerod, C.-P., Rump, S.M.: Improved error bounds for inner products in floating-point arithmetic. SIAM J. Matrix Anal. Appl. **34**, 338–344 (2013)
5. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
6. Lee, N., Cichocki, A.: Estimating a few extreme singular values and vectors for large-scale matrices in tensor train format. SIAM J. Matrix Anal. Appl. **36**(3), 994–1014 (2015)
7. Oseledets, I.: Tensor-train decomposition. SIAM J. Sci. Comput. **33**(5), 2295–2317 (2011)
8. Oseledets, I., Dolgov, S.V.: Solution of linear systems and matrix inversion in the TT-format. SIAM J. Sci. Comput. **34**(5), A2718–A2739 (2012)
9. Oseledets, I., Tyrtyshnikov, E.: Recursive decomposition of multidimensional tensors. Dokl. Math. **80**(1), 460–462 (2009)
10. Oseledets, I., Tyrtyshnikov, E.: TT-cross approximation for multidimensional arrays. Linear Algebra Appl. **432**(1), 70–88 (2010)

The Derivative of the Matrix Geometric Mean with an Application to the Nonnegative Decomposition of Tensor Grids

Bruno Iannazzo, Ben Jeuris, and Filippo Pompili

Abstract We provide an expression for the derivative of the weighted matrix geometric mean, with respect to both the matrix arguments and the weights, that can be easily translated to an algorithm for its computation. As an application, we consider the problem of the approximate decomposition of a tensor grid M, a matrix whose entries are positive definite matrices. For different geometries on the set of positive definite matrices, we derive an approximate decomposition such that any column of M is a barycentric combination of the columns of a smaller tensor grid. This extends the Euclidean case, already considered in the literature, to the geometry in which the barycenter is the matrix geometric mean and the log-Euclidean geometry.

Keywords Matrix geometric mean · Karcher mean · Tensor grid · Positive definite matrix · Nonnegative factorization

1 Introduction

The geometric mean of positive definite matrices, the so-called Karcher mean, is the minimizer, over the set of positive definite matrices of a fixed size, of the function

$$f(X) = \sum_{\sigma=1}^{p} \delta^2(X, A_\sigma), \tag{1}$$

B. Iannazzo (✉)
Dipartimento di Matematica e Informatica, Università di Perugia, Perugia, Italy
e-mail: bruno.iannazzo@dmi.unipg.it

B. Jeuris
Department of Computer Science, KU Leuven, Leuven, Belgium

F. Pompili
Thomson Reuters, Toronto, ON, Canada
e-mail: filippo.pompili@thomsonreuters.com

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_6

where $A_1, \dots, A_p \in \mathbb{C}^{\mu \times \mu}$ are given positive definite matrices, and δ is a distance on the set of positive definite matrices that we will define later.

The matrix geometric mean can be related to a Riemannian geometry in the set of positive definite matrices of size μ that we will denote by $\mathscr{P}_\mu$. Besides the wide mathematical interest in this geometry (see [2, Ch. 6]), there are also a large number of applications where it is used. In particular, it has been used for averaging tensors [1, 18], for describing the variability of shapes [9], and also in regularization of matrix data [19], appearance tracking [7, 23], and brain–computer interface [8], to cite just some.

A useful generalization of the matrix geometric mean is the weighted matrix geometric mean [16], or weighted Karcher mean, that is the minimizer of

$$\sum_{\sigma=1}^{p} w_\sigma \delta^2(X, A_\sigma), \tag{2}$$

where $w_1, \dots, w_p$ are nonnegative weights, of which some must be nonzero, and δ is the same distance as before. With no loss of generality, we may assume that $w_1 + \cdots + w_p = 1$, so that the weighted matrix geometric mean defines a barycentric combination of $A_1, \dots, A_p$ with respect to the Riemannian structure of $\mathscr{P}_\mu$. We use the notation $K(w_1, \dots, w_p; A_1, \dots, A_p)$ for the weighted matrix geometric mean.

In this chapter, we consider the derivative of the weighted matrix geometric mean with respect to both the matrix arguments and the weights. While there is no explicit expression for the mean K, we are able to find an expression for its derivative that is explicit in terms of K.

The matrix K can be computed by an optimization algorithm or a fixed-point iteration [5, 13, 14, 21]. Once K has been approximated, the derivative can be evaluated directly by means of a numerical algorithm based on the aforementioned expression.

As an application, we consider a generalization of the nonnegative matrix factorization [17] to tensor grids that can be seen as matrices whose entries are positive definite matrices (the physical tensors).

We propose an approach to compute the factorization based on the geometry on $\mathscr{P}_\mu$ mentioned above. This requires the derivative of the matrix geometric mean for its numerical computation.

The chapter is structured as follows: in Sect. 2 we provide some basic material on the Riemannian geometry of $\mathscr{P}_\mu$ that gives rise to the matrix geometric mean; in Sect. 3 we provide an algorithm for computing the weighted matrix geometric mean and its derivative, whose explicit expression is obtained; in Sect. 4 we propose different models for the factorization of tensor grids, with a simple minimization algorithm to obtain it; in Sect. 5 we perform some preliminary numerical experiments proving that the new models have potential in solving factorization problems; in the final section we draw some conclusions.

2 The Geometry of $\mathscr{P}_\mu$ and the Matrix Geometric Mean

The geometry of $\mathscr{P}_\mu$ is very easy, since the positive definite matrices of size μ are an open subset of $\mathbb{H}_\mu$, the set of Hermitian matrices of size μ. This endows $\mathscr{P}_\mu$ with a natural structure of differentiable manifold, where the tangent space at any point $A \in \mathscr{P}_\mu$ is isomorphic to $\mathbb{H}_\mu$. In view of this isomorphism, we will identify tangent vectors with Hermitian matrices.

The vector space $\mathbb{H}_\mu$ is an Euclidean space, with the usual scalar product

$$\langle X, Y \rangle = \operatorname{trace}(XY),$$

whose associated norm is the Euclidean (Frobenius) norm, $\|X\|_F := \left(\operatorname{trace}(X^2)\right)^{1/2}$. Nevertheless, the Riemannian structure on $\mathscr{P}_\mu$, that will allow us to define the distance δ of (1) and (2), is obtained when one considers the scalar product

$$\langle X, Y \rangle_A^{(R)} := \operatorname{trace}(A^{-1} X A^{-1} Y), \qquad X, Y \in \mathbb{H}_\mu, \tag{3}$$

on the tangent space to $\mathscr{P}_\mu$ at the matrix A.

This Riemannian structure has been described, for instance, in [15, Ch. XII], [3, 20], and yields the Riemannian distance δ of (1) on $\mathscr{P}_\mu$, whose explicit expression is known to be

$$\delta(X, Y) = \| \log(X^{-1/2} Y X^{-1/2}) \|_F. \tag{4}$$

The inverse square root and the logarithm in (4) should be understood as *primary matrix functions*. When a matrix X is diagonalizable, say $K^{-1} X K = \operatorname{diag}(\lambda_1, \ldots, \lambda_\mu)$, the primary matrix function $f(X)$ is defined as $K \operatorname{diag}(f(\lambda_1), \ldots, f(\lambda_\mu)) K^{-1}$; this requires that f is well defined on the spectrum of X. The definition of primary matrix function can be extended to non-diagonalizable matrices, making further assumptions on the regularity of f on the spectrum of X (for a detailed description, see [10]). An important property of matrix functions is that they commute with similarities: if $f(X)$ is well defined and M is invertible, then $f(M^{-1} X M)$ is well defined and we have $f(M^{-1} X M) = M^{-1} f(X) M$. We will use, moreover, the fact that $\log(M^{-1}) = -\log(M)$ when M has no nonpositive real eigenvalues.

A primary matrix function $f : \Omega \to \mathbb{C}^{\mu \times \mu}$, with Ω open subset of $\mathbb{C}^{\mu \times \mu}$, is Fréchet differentiable at $X \in \Omega$, if there exists a linear function $Df(X) : \mathbb{C}^{\mu \times \mu} \to \mathbb{C}^{\mu \times \mu}$ such that, for any matrix norm

$$f(X + H) = f(X) + Df(X)[H] + o(\|H\|),$$

as $H \in \mathbb{C}^{\mu \times \mu}$ tends to 0.

For the practical use of derivatives of primary matrix functions, it is useful to consider the vec operator that stacks the columns of a matrix into a long vector and

define a *Kronecker matrix* $K_f(X)$ such that $\operatorname{vec}(Df(X)[H]) = K_f(X)\operatorname{vec}(H)$. The basis of $\mathbb{C}^{\mu\times\mu}$, that is mapped through the vec operator to the standard basis of $\mathbb{C}^{\mu^2}$, is said to be the *vec basis* and $K_f(X)$ is the matrix representing the derivative in this basis. With abuse of notation we will write $Df(X)$ also for $K_f(X)$. A useful equality involving the vec operator and the Kronecker product is $\operatorname{vec}(AXB) = (B^T \otimes A)\operatorname{vec}(X)$ [11, Sec. 4.3].

The Fréchet derivative commutes with similarities: if $Df(X)[H]$ is the Fréchet derivative of f at X in the direction H and M is invertible, then $Df(M^{-1}XM)[M^{-1}HM] = M^{-1}Df(X)[H]M$. This fact can be seen using the commutativity of primary matrix functions with similarities, and observing that if $f(X+H) - f(X) - Df(X)[H] = o(\|H\|)$, then

$$f(M^{-1}XM + M^{-1}HM) - f(M^{-1}XM) - M^{-1}Df(X)[H]M = o(\|M^{-1}HM\|).$$

In the vec basis, we can write

$$Df(M^{-1}XM) = (M^T \otimes M^{-1})Df(X)(M^{-T} \otimes M), \tag{5}$$

where $M^{-T} := (M^{-1})^T$.

We conclude this section, giving a lemma on the derivative of the Euclidean and Riemannian distance with respect to one of its arguments, which will be useful in the following. (The first equality is straightforward, for a proof of the second equality, see [2, Thm. 6.3.3].)

Lemma 1 *Let $A \in \mathbb{H}_\mu$ and let $d_A : \mathbb{H}_\mu \to \mathbb{H}_\mu$ be such that $d_A(X) = \|A - X\|_F$. For any $H \in \mathbb{H}_\mu$, we have*

$$Dd_A^2(X)[H] = 2\operatorname{trace}((X - A)H).$$

Let $A \in \mathscr{P}_\mu$ and let $\delta_A : \mathscr{P}_\mu \to \mathscr{P}_\mu$ be such that $\delta_A(X) = \delta(A, X)$, with δ as in (4). For any $H \in \mathbb{H}_\mu$, we have

$$D\delta_A^2(X)[H] = 2\operatorname{trace}(X^{-1}\log(XA^{-1})H).$$

3 Computing the Weighted Matrix Geometric Mean and Its Derivative

We propose a method for computing the derivative of the weighted matrix geometric mean with respect to both the matrix arguments and the weights.

This method requires the weighted matrix geometric mean: in Sect. 3.1 we adapt the Richardson algorithm [5] from the standard matrix geometric mean to the weighted one; while in Sect. 3.2 we derive explicit expressions (in terms of the

matrix geometric mean itself) for the derivative of the weighted matrix geometric mean that can be evaluated by a numerical algorithm.

3.1 Computing the Weighted Matrix Geometric Mean

The weighted matrix geometric mean of p positive definite matrices $A_1, \ldots, A_p \in \mathscr{P}_\mu$, with nonnegative weights $w_1, \ldots, w_p$, of which some must be nonzero, is the unique minimizer of the function

$$f(w_1, \ldots, w_p; A_1, \ldots, A_p) = \sum_{\sigma=1}^{p} w_\sigma \delta^2(X, A_\sigma), \tag{6}$$

and it is denoted by $K = K(w_1, \ldots, w_p; A_1, \ldots, A_p)$.

Using Lemma 1 one gets that the Euclidean gradient of f and its Riemannian gradient with respect to the inner product (3) are

$$\nabla f(X) = 2 \sum_{\sigma=1}^{p} w_\sigma X^{-1} \log(X A_\sigma^{-1}), \qquad \nabla^{(R)} f(X) = 2 \sum_{\sigma=1}^{p} w_\sigma X \log(A_\sigma^{-1} X), \tag{7}$$

respectively (see [6, Sec. 4.3] for the unweighted case).

For computing the weighted matrix geometric mean, we consider the Riemannian gradient descent iteration

$$X_{\ell+1} = X_\ell \exp\Big(\theta_\ell \sum_{\sigma=1}^{p} w_\sigma \log(X_\ell^{-1} A_\sigma)\Big), \qquad X_0 \in \mathscr{P}_\mu. \tag{8}$$

Using the same idea as the one of the Richardson algorithms [5], we look for the steplength θ_ℓ that gives optimal local convergence, when X_ℓ is seen as an approximation of the matrix geometric mean. This yields $\theta_\ell = 2/\sum_{\sigma=1}^{p} w_\sigma \frac{c_\sigma^{(\ell)}+1}{c_\sigma^{(\ell)}-1} \log c_\sigma^{(\ell)}$, where $c_\sigma^{(\ell)}$ is the ratio between the largest and the smallest eigenvalues of the matrix $X_\ell^{-1/2} A_\sigma X_\ell^{-1/2}$.

As an initial value, a possibility is to use the weighted arithmetic mean or a weighted generalization of the cheap mean [4].

3.2 The Derivative of the Matrix Geometric Mean

Even if an explicit expression for the weighted matrix geometric mean is not known, we are able to find an explicit expression for its derivative with respect to the matrix arguments and the weights, in terms of the matrix geometric mean itself.

We start from Eq. (7) and set the Euclidean gradient to zero. After some simplifications, this yields an equation that defines the weighted matrix geometric mean of the matrices $A_1, \dots, A_p$ with weights $w_1, \dots, w_p$, namely

$$\sum_{\sigma=1}^{p} w_\sigma \log(X A_\sigma^{-1}) = 0, \tag{9}$$

whose unique positive definite solution is the weighted matrix geometric mean $K := K(w_1, \dots, w_p; A_1, \dots, A_p)$.

We thus know that the function

$$\varphi(A_1, \dots, A_p) := \sum_{\sigma=1}^{p} w_\sigma \log(K(w_1, \dots, w_p; A_1, \dots, A_p) A_\sigma^{-1}),$$

as a function from $\mathscr{P}_\mu^p$ to $\mathscr{P}_\mu$ is such that $\varphi \equiv 0$.

From the derivatives of φ we will get the derivatives of K as a function of the matrix arguments. Let us set

$$\Delta_\ell[H_\ell] = DK(w_1, \dots, w_p; A_1, \dots, A_p)[0, \dots, 0, H_\ell, 0, \dots, 0],$$

where the Hermitian matrix H_ℓ is put in position $1 \le \ell \le p$. By the chain rule and since the derivative of the function X^{-1} in the direction H is $-X^{-1} H X^{-1}$, we have

$$\begin{aligned} 0 &= D\varphi(A_1, \dots, A_p)[0, \dots, 0, H_\ell, 0, \dots, 0] \\ &= \sum_{\sigma=1}^{p} w_\sigma D\log(K A_\sigma^{-1})[\Delta_\ell[H_\ell] A_\sigma^{-1} - K A_\ell^{-1} H_\ell A_\ell^{-1} \delta_{\ell\sigma}], \end{aligned}$$

where $\delta_{\ell\sigma}$ is the Kronecker delta function.

In the vec basis we can write (with abuse of notation we denote with $D\log(K A_\sigma^{-1})$ also the $\mu^2 \times \mu^2$ Kronecker matrix representing the derivative in the vec basis)

$$\sum_{\sigma=1}^{p} w_\sigma D\log(K A_\sigma^{-1})(A_\sigma^{-T} \otimes I) \operatorname{vec}(\Delta_\ell[H_\ell]) = w_\ell D\log(K A_\ell^{-1})(A_\ell^{-T} \otimes K A_\ell^{-1}) \operatorname{vec}(H_\ell)$$

and then the matrix representing the derivative of the matrix geometric mean, with respect to its ℓ-th argument, with $\ell = 1, \ldots, p$, is

$$\Delta_\ell = \Big(\sum_{\sigma=1}^{p} Z_\sigma\Big)^{-1} Z_\ell (I \otimes K A_\ell^{-1}), \quad Z_\ell = w_\ell D \log(K A_\ell^{-1})(A_\ell^{-T} \otimes I), \quad \ell = 1, \ldots, p. \tag{10}$$

The matrix Z_ℓ is not necessarily Hermitian. We will obtain another expression of the derivative, involving Hermitian matrices only, and this will make the derivative easier to be computed.

We consider the full rank factorization $K = R^* R$ (for instance, the Cholesky factorization), with $R \in \mathbb{C}^{n\times n}$, and the Schur factorizations

$$R A_\ell^{-1} R^* = U_\ell D_\ell U_\ell^*, \qquad \ell = 1, \ldots, p, \tag{11}$$

from which it follows that, for $\ell = 1, \ldots, p$,

$$K A_\ell^{-1} = R^* U_\ell D_\ell U_\ell^* R^{-*}, \qquad A_\ell^{-T} = \overline{R}^{-1} \overline{U}_\ell D_\ell \overline{U}_\ell^* R^{-T}. \tag{12}$$

Notice that D_ℓ is a diagonal matrix with real positive diagonal entries.

Using (12) and the properties of the derivative of matrix functions (compare (5)), we obtain for Z_ℓ in (10),

$$\begin{aligned} Z_\ell &= w_\ell D \log(R^* U_\ell D_\ell U_\ell^* R^{-*})(A_\ell^{-T} \otimes I) \\ &= w_\ell (\overline{R}^{-1} \overline{U}_\ell \otimes R^* U_\ell) D \log(D_\ell)(\overline{U}_\ell^* \overline{R} \otimes U_\ell^* R^{-*})(\overline{R}^{-1} \overline{U}_\ell D_\ell \overline{U}_\ell^* R^{-T} \otimes I) \\ &= w_\ell (\overline{R}^{-1} \otimes R^*)(\overline{U}_\ell \otimes U_\ell) D \log(D_\ell)(D_\ell \otimes I)(\overline{U}_\ell^* \otimes U_\ell^*)(R^{-T} \otimes R^{-*}). \end{aligned}$$

In order to get an expression for Δ_ℓ where the matrix to be inverted is Hermitian, we define the new matrix

$$S_\ell := w_\ell (R^T \otimes R^*)(\overline{U}_\ell \otimes U_\ell) D \log(D_\ell)(D_\ell \otimes I)(\overline{U}_\ell^* \otimes U_\ell^*)(R^{-T} \otimes R^{-*}), \tag{13}$$

and it is easily seen that

$$\Delta_\ell = \Big(\sum_{\sigma=1}^{p} S_\sigma\Big)^{-1} S_\ell (I \otimes K A_\ell^{-1}). \tag{14}$$

We claim that S_ℓ is positive definite for any $\ell = 1, \ldots, p$, and this can be proved by showing that $D \log(D_\ell)(D_\ell \otimes I)$ is (diagonal and) positive definite. We need the following statement attributed to Daleckii and Krein (see [10, Thm. 3.11]).

Lemma 2 *Let f be analytic on the open set $\Omega \subset \mathbb{C}$. Let $\Delta = \mathrm{diag}(d_1, \dots, d_\mu)$ be a diagonal matrix with $d_i \in \Omega$ for all i. Then, for any $H = (h_{ij})_{i,j=1,\dots,\mu} \in \mathbb{C}^{\mu\times\mu}$, we have*

$$(Df(\Delta)[H])_{ij} = f[d_i, d_j]h_{ij}, \qquad i, j = 1, \dots, \mu,$$

where $f[d_i, d_j] = (f(d_i) - f(d_j))/(d_i - d_j)$ if $d_i \neq d_j$ and $f[d_i, d_j] = f'(d_i)$ if $d_i = d_j$.

Let F be the matrix such that $(F)_{ij} = f[d_i, d_j]$, for $i, j = 1, \dots, \mu$. The matrix that represents $Df(\Delta)$ in the vec basis is $\mathrm{diag}(\mathrm{vec}(F))$.

Lemma 2 shows that $D\log(D_\ell)$ is diagonal and that its diagonal elements are of the type

$$\frac{\log\lambda_i - \log\lambda_j}{\lambda_i - \lambda_j}, \qquad \frac{1}{\lambda_i},$$

where λ_i, λ_j are eigenvalues of D_ℓ. Since the eigenvalues of D_ℓ are positive, also the diagonal elements of $D\log(D_\ell)$ are positive. This shows that S_ℓ is positive definite.

From (12), we get that $U_\ell^* R^{-*} K A_\ell^{-1} = D_\ell U_\ell^* R^{-*}$, and

$$\begin{aligned} T_\ell &:= S_\ell(I \otimes K A_\ell^{-1}) \\ &= w_\ell(R^T \otimes R^*)(\overline{U}_\ell \otimes U_\ell) D\log(D_\ell)(D_\ell \otimes D_\ell)(\overline{U}_\ell^* \otimes U_\ell^*)(R^{-T} \otimes R^{-*}), \end{aligned}$$

that is a positive definite matrix, and we obtain a minor variation of (14), namely

$$\Delta_\ell = \Big(\sum_{\sigma=1}^{p} S_\sigma\Big)^{-1} T_\ell, \tag{15}$$

where the matrix $K A_\ell^{-1}$ (not necessarily Hermitian) does not appear.

The evaluation of the previous formulae (14) or (15) is very expensive, since the matrices, $S_1, \dots, S_p$, have size μ^2 and we do not see a way to compute Δ_ℓ with $O(\mu^3)$ ops, without constructing and inverting (or, more appropriately, solving a multiple right-hand side linear system with coefficient matrix) $S_1 + \cdots + S_p$.

For $p = 2$ there is a much simpler formula for the derivative (see the extended preprint of [12] available at http://arxiv.org/abs/1201.0101).

We also need the derivative of the weighted matrix geometric mean with respect to the weights. As in the derivation with respect to the matrix variables we consider the function

$$\psi(w_1, \dots, w_p) := \sum_{\sigma=1}^{p} w_\sigma \log(K(w_1, \dots, w_p; A_1, \dots, A_p) A_\sigma^{-1}),$$

that is zero for each $w \in \mathbb{R}^p$ such that $w \neq 0$ and $w_j \geq 0$, for $j = 1, \ldots, p$.

Let us set

$$\Gamma_\ell[f_\ell] = DK(w_1, \ldots, w_p; A_1, \ldots, A_p)[0, \ldots, 0, f_\ell, 0, \ldots, 0],$$

where f_ℓ is put in position ℓ. By the chain rule we have

$$\begin{aligned} 0 = D\psi(w_1, \ldots, w_p)[0, \ldots, 0, f_\ell, 0, \ldots, 0] \\ = f_\ell \log(K A_\ell^{-1}) + \sum_{\sigma=1}^{p} w_\sigma D \log(K A_\sigma^{-1})[\Gamma_\ell[f_\ell] A_\sigma^{-1}]. \end{aligned}$$

In the vec basis we can write

$$\begin{aligned} \sum_{\sigma=1}^{p} w_\sigma D \log(K A_\sigma^{-1})(A_\sigma^{-T} \otimes I) \operatorname{vec}(\Gamma_\ell[f_\ell]) \\ = -\operatorname{vec}(f_\ell \log(K A_\ell^{-1})) = \operatorname{vec}(\log(A_\ell K^{-1})) f_\ell, \end{aligned}$$

so that the matrix representing the derivative of the matrix geometric mean, with respect to its ℓ-th weight, is

$$\Gamma_\ell = \Big(\sum_{\sigma=1}^{p} Z_\sigma\Big)^{-1} \operatorname{vec}(\log(A_\ell K^{-1})),$$

with Z_σ defined in (10).

A more symmetric form is obtained by introducing, as before, the Hermitian matrix $S_\ell = (R^T \overline{R} \otimes I) Z_\ell$ that yields

$$\Gamma_\ell = \Big(\sum_{\sigma=1}^{p} S_\sigma\Big)^{-1} \operatorname{vec}(\log(A_\ell K^{-1}) K) = \Big(\sum_{\sigma=1}^{p} S_\sigma\Big)^{-1} \operatorname{vec}(K \log(K^{-1} A_\ell)), \tag{16}$$

where we have used that K is Hermitian and the definition of primary matrix function. Notice that $\sum_{\sigma=1}^{p} \Gamma_\sigma w_\sigma = 0$, because K satisfies $\sum_\sigma w_\sigma \log(A_\sigma K^{-1}) K = 0$, and this is expected since the weighted matrix geometric mean is invariant under positive scaling of the weights.

In summary, let $A_1, \ldots, A_p$ be positive definite matrices of size μ, and $w_1, \ldots, w_p$ nonnegative numbers whose sum is not zero. Let $K = K(w_1, \ldots, w_p; A_1, \ldots, A_p)$ be the weighted matrix geometric mean of $A_1, \ldots, A_p$, then the derivative of the matrix geometric mean with respect to

the ℓ-th matrix variable, in the vec basis, has the following expression:

$$\Delta_\ell = \Big(\sum_{\sigma=1}^{p} S_\sigma\Big)^{-1} S_\ell (I \otimes K A_\ell^{-1}), \tag{17}$$

where

$$S_\ell = w_\ell (R^T \otimes R^*)(\overline{U}_\ell \otimes U_\ell) D \log(D_\ell)(D_\ell \otimes I)(\overline{U}_\ell^* \otimes U_\ell^*)(R^{-T} \otimes R^{-*}),$$

with $K = R^* R$ (full rank factorization) and $R A_\ell^{-1} R^* = U_\ell D_\ell U_\ell^*$ (Schur factorization).

If the weights are positive, then the derivative of the matrix geometric mean with respect to the ℓ-th weight variable, if the vec basis is used on the arrival space, has the following expression:

$$\Gamma_\ell = \Big(\sum_{\sigma=1}^{p} S_\sigma\Big)^{-1} \operatorname{vec}(K \log(K^{-1} A_\ell)). \tag{18}$$

4 Factorization of Tensor Grids

Let $M \in \mathbb{R}^{m\times n}$ be a nonnegative matrix, namely a matrix with nonnegative entries. In its classic formulation, nonnegative matrix factorization (NMF) consists in finding two nonnegative matrices $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{k\times n}$, such that the product UV best approximates the given matrix M with respect to a certain matrix norm.

In the applications, one is mostly interested in cases where k is much smaller than n, since this yields an approximation of the columns of M as linear combinations of the fewer columns of U, through V. This factorization has been extensively used in machine learning and data analysis (see [22] and the references therein).

A more general problem is obtained when one considers instead of an $m \times n$ matrix, a grid of points where a positive definite matrix is attached to each point, namely a set of positive definite matrices M_{ij} of size μ, with $i = 1, \dots, m$ and $j = 1, \dots, n$. Such an object arises when one considers a planar distribution of tensor quantities, for this reason we call it *tensor grid*.

From a tensor grid one can easily construct an $m \times n$ block matrix $M \in \mathbb{R}^{\mu m\times \mu n}$ whose blocks are M_{ij}, but this matrix may have negative entries, so that NMF cannot be applied to M. Xie et al. [24] have proposed to "factorize" M using a set of positive definite matrices $U_{i\ell} \in \mathbb{R}^{\mu\times\mu}$, with $i = 1, \dots, m$ and $\ell = 1, \dots, k$ and a nonnegative matrix $V = (v_{\ell j}) \in \mathbb{R}^{k\times n}$ such that the function

$$\mathscr{E}(U, V) := \sum_{i=1}^{m}\sum_{j=1}^{n} \|M_{ij} - \sum_{\ell=1}^{k} U_{i\ell} v_{\ell j}\|_F^2 \tag{19}$$

achieves the minimum. Each block column of the matrix M is thus approximated by a nonnegative linear combination of the columns of the matrix U (the $n \times k$ block matrix whose blocks are $U_{i\ell}$).

This first model implicitly assumes the Euclidean geometry on the set of positive definite matrices, but the latter set can be provided with a different geometry, such as the one described in Sect. 2. In the following, we propose two new models based on these geometries.

Since we are dealing with objects in the set $\mathscr{P}_\mu^{m\times n}$ that are (mn)-tuples of positive definite matrices, we consider also the product Riemannian manifold structure on $\mathscr{P}_\mu^{m\times n}$, where the tangent space can be identified with $\mathbb{H}_\mu^{m\times n}$, the set of (mn)-tuples of Hermitian matrices, the scalar product at $M \in \mathscr{P}_\mu^{m\times n}$ is

$$\langle X, Y\rangle_M^{(R)} := \sum_{i=1}^{m}\sum_{j=1}^{n}\langle X_{ij}, Y_{ij}\rangle_{M_{ij}}^{(R)}, \qquad X, Y \in \mathbb{H}_\mu^{m\times n}, \tag{20}$$

and the distance is $\Delta^2(M, N) := \sum_{i=1}^{m}\sum_{j=1}^{n}\delta^2(M_{ij}, N_{ij})$.

Now we can give a second model for the decomposition of a tensor grid that is to find the minimum of the cost function

$$\mathscr{R}(U, V) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\delta^2\big(M_{ij}, K(v_{1j}, \ldots, v_{kj}; U_{i1}, \ldots, U_{ik})\big), \tag{21}$$

where, as before, $U_{i\ell} \in \mathbb{R}^{\mu\times\mu}$ is positive definite and $V = (v_{\ell j}) \in \mathbb{R}^{k\times n}$ is nonnegative with no zero columns.

In some sense we are trying to get U and V such that the approximation $M_{ij} \approx K(v_{1j}, \ldots, v_{kj}; U_{i1}, \ldots, U_{ik})$ holds.

This nonlinear decomposition, that we call *matrix geometric mean decomposition*, is much more complicated than its Euclidean counterpart and so it is much more computationally demanding.

For this reason, in some cases, we replace the weighted matrix geometric mean with the weighted log-Euclidean mean, defined as

$$L(w_1, \ldots, w_p; A_1, \ldots, A_p) := \exp\left(\sum_{\ell=1}^{p} w_\ell \log A_\ell\right),$$

that is known to be an approximation of the weighted matrix geometric mean, but cheaper to be computed. The log-Euclidean mean, as the matrix geometric mean, is based on the structure of the set of positive matrices, but it is computed by a direct evaluation of a single expression instead of an iterative process. From this, we obtain a third model as the minimizer of the cost function

$$\mathscr{L}(U, V) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\|\log M_{ij} - \sum_{\ell=1}^{k} v_{\ell j}\log U_{i\ell}\|_F^2. \tag{22}$$

Computing the log-Euclidean decomposition is less expensive than the matrix geometric mean decomposition, while the Riemannian structure of $\mathscr{P}_\mu$ is well approximated.

In the next section we will derive an alternating gradient descent algorithm for the functions (21) and (22).

Remark 1 Let $A_1, \dots, A_k \in \mathscr{P}_\mu$. While the functions $w_1 A_1 + \cdots + w_k A_k$ and the function $\exp(w_1 \log(A_1) + \cdots + w_k \log(A_k))$, for $w_1, \dots, w_k > 0$ describe an object depending on k parameters; the function $K(w_1, \dots, w_k; A_1, \dots, A_k)$ describes an object depending on $k-1$ parameters, since the weighted matrix geometric mean is invariant under positive scaling of the parameters. For $k = 2$, for instance, $w_1 A_1 + w_2 A_2$ describes a surface, while $K(w_1, w_2; A_1, A_2)$ is a curve.

A possible remedy to this issue is to introduce, in the matrix geometric mean model, the values $v_{0j} \geq 0$, for $j = 1, \dots, m$, and define the function

$$\mathscr{R}'(U, V) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} \delta^2\big(M_{ij}, v_{0j} K(v_{1j}, \dots, v_{kj}; U_{i1}, \dots, U_{ik})\big). \tag{23}$$

The new scaling constants reestablish the right number of parameters, marginally affecting the computation.

4.1 A Simple Algorithm for the Matrix Geometric Mean Decomposition

A customary optimization algorithm in NMF is the alternating minimization. In our case we use the alternating minimization of the functions $\mathscr{R}(U, V)$ of (21) or $\mathscr{L}(U, V)$ of (22) with respect to the variables U and V, respectively. We describe the algorithm for $\mathscr{R}$; the one for $\mathscr{L}$ is the same.

Algorithm 1 Start with U_0, V_0 and repeat the cycle:

1. use a gradient descent method, with a backtracking strategy to ensure cost function reduction, on $\mathscr{R}(U_q, V_q)$ to get U_{q+1};
2. use a gradient descent, with a backtracking strategy to ensure cost function reduction, on $\mathscr{R}(U_{q+1}, V_q)$ to get V_{q+1} (project onto the positive orthant, if some of the elements of V_{q+1} are negative);

for $q = 0, 1, 2, \dots$, until the gradients are smaller than a fixed tolerance or a maximum number of iterations has been reached.

In order to use a gradient descent algorithm, it is necessary to derive the gradients of both functions and design an algorithm to compute them. The rest of the section is devoted to the gradients computation.

4.1.1 The Gradients of $\mathscr{R}(U, V)$

We consider the derivatives of the cost function $\mathscr{R}(U, V)$ of (21) as a function of the sole variables U and V, respectively. We first compute the derivative of the cost function with respect to U in the direction $E = (E_{st}) \in \mathbb{H}_{\mu}^{m\times k}$, from which the Riemannian gradient is easily obtained. By linearity, it is sufficient to consider the functions $\varphi_{ij}(U) := \delta^2\big(M_{ij}, K(v_{1j}, \ldots, v_{kj}; U_{i1}, \ldots, U_{ik})\big)$ separately, for each i and j.

Lemma 3 *The Fréchet derivative of the function $\varphi_{ij}(U)$ in the direction $E = (E_{st})$ is*

$$D\varphi_{ij}(U)[E] = 2\sum_{\ell=1}^{k} \operatorname{vec}(K_{ij}^{-1}\log(K_{ij}M_{ij}^{-1}))^* \Delta_{\ell}^{(ij)} \operatorname{vec}(E_{i\ell}),$$

where $K_{ij} := K(v_{1j}, \ldots, v_{kj}; U_{i1}, \ldots, U_{ik})$ and $\Delta_{\ell}^{(ij)}$ is the matrix representing the derivative of the weighted matrix geometric mean in the vec basis, obtained as Δ_{ℓ} of (17) with

$$A_{\ell} := U_{i\ell}, \qquad w_{\ell} := v_{\ell j}, \qquad K := K_{ij}.$$

Proof For a fixed V and for $W \in \mathscr{P}_{\mu}^{n\times k}$, we can define the two functions $\delta_{ij}(X) := \delta_{M_{ij}}(X)$ and $\sigma_{ij}(W) := K(v_{1j}, \ldots, v_{kj}; W_{i1}, \ldots, W_{ik})$ for which $\varphi_{ij}(U) = (\delta_{ij}^2 \circ \sigma_{ij})(U)$. The derivative of σ_{ij} at U, in the direction E, is

$$D\sigma_{ij}(U)[E] = \sum_{s,t} D\sigma_{ij}(U)[E_{st}] = \sum_{\ell=1}^{k} D\sigma_{ij}(U)[E_{i\ell}] = \sum_{\ell=1}^{k} \operatorname{vec}^{-1}(\Delta_{\ell}^{(ij)} \operatorname{vec}(E_{i\ell})),$$

then, by the chain rule and using Lemma 1, we get

$$\begin{aligned} D\varphi_{ij}(U)[E] &= D\delta_{ij}^2(K_{ij})[D\sigma_{ij}(U)[E]] \\ &= 2\sum_{\ell=1}^{k} \operatorname{trace}(K_{ij}^{-1}\log(K_{ij}M_{ij}^{-1}) \operatorname{vec}^{-1}(\Delta_{\ell}^{(ij)} \operatorname{vec}(E_{i\ell}))) \\ &= 2\sum_{\ell=1}^{k} \operatorname{vec}(K_{ij}^{-1}\log(K_{ij}M_{ij}^{-1}))^* \Delta_{\ell}^{(ij)} \operatorname{vec}(E_{i\ell}), \end{aligned}$$

where the latter equality follows from $\operatorname{trace}(AB) = \operatorname{vec}(A)^* \operatorname{vec}(B)$, when A is Hermitian (notice that the matrix $K_{ij}^{-1}\log(K_{ij}M_{ij}^{-1})$ is Hermitian).

Let $\nabla^{(U)}$ denote the Riemannian gradient of the function $\frac{1}{2}\sum_{i,j}\varphi_{ij}(U)$, with respect to the geometry defined in (20). By definition,

$$\langle \nabla^{(U)}, E\rangle_U^{(R)} = D\Big(\frac{1}{2}\sum_{i,j}\varphi_{ij}\Big)(U)[E],$$

for any $E \in {\mathbb{H}_\mu}^{m\times k}$, which, in view of Lemma 3, can be rewritten as

$$\begin{aligned}\langle \nabla^{(U)}, E\rangle_U^{(R)} &= \sum_{i=1}^{m}\sum_{\ell=1}^{k} \operatorname{trace}(U_{i\ell}^{-1}\nabla_{i\ell}^{(U)}U_{i\ell}^{-1}E_{i\ell})\\ = \sum_{i=1}^{m}\sum_{\ell=1}^{k} \operatorname{vec}(U_{i\ell}^{-1}\nabla_{i\ell}^{(U)}U_{i\ell}^{-1})^*\operatorname{vec}(E_{i\ell}) &= \sum_{i=1}^{m}\sum_{\ell=1}^{k} \operatorname{vec}(\nabla_{i\ell}^{(U)})^*(U_{i\ell}^{-T}\otimes U_{i\ell}^{-1})^*\operatorname{vec}(E_{i\ell})\\ &= \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{\ell=1}^{k} \operatorname{vec}(K_{ij}^{-1}\log(K_{ij}M_{ij}^{-1}))^*\Delta_\ell^{(ij)}\operatorname{vec}(E_{i\ell}),\end{aligned}$$

from which we finally get

$$\operatorname{vec}(\nabla_{i\ell}^{(U)}) = \sum_{j=1}^{n}(U_{i\ell}^T\otimes U_{i\ell})\big(\Delta_\ell^{(ij)}\big)^*\operatorname{vec}(K_{ij}^{-1}\log(K_{ij}M_{ij}^{-1})), \tag{24}$$

for $i = 1,\dots,m$ and $\ell = 1,\dots,k$.

On the other hand, we should compute the derivative of the function $\mathscr{R}(U,V)$ of (21) with respect to V in the direction $F = (f_{st}) \in \mathbb{R}^{k\times n}$, from which the Riemannian gradient is easily obtained. By linearity, it is sufficient to consider the functions $\psi_{ij}(V) := \delta^2\big(M_{ij}, K(v_{1j},\dots,v_{kj};U_{i1},\dots,U_{ik})\big)$ separately, for each i and j.

Lemma 4 *The Fréchet derivative of the function $\psi_{ij}(V)$, in the direction $F = (f_{st})$, is*

$$D\psi_{ij}(V)[F] = 2\sum_{\ell=1}^{k}\operatorname{vec}(K_{ij}^{-1}\log(K_{ij}M_{ij}^{-1}))^*\Gamma_\ell^{(ij)}f_{\ell j},$$

where $K_{ij} := K(v_{1j},\dots,v_{kj};U_{i1},\dots,U_{ik})$ and $\Gamma_\ell^{(ij)}$ is the (column) vector representing the derivative of the weighted matrix geometric mean with respect to the weights obtained as Γ_ℓ of (18) with

$$A_\ell := U_{i\ell}, \qquad w_\ell := v_{\ell j}, \qquad K := K_{ij}.$$

Proof The proof is similar to the one of Lemma 3.

Let $\nabla^{(V)}$ be the Riemannian gradient of the function $\frac{1}{2}\sum_{i,j}\psi_{ij}(V)$. By definition

$$\langle \nabla^{(V)}, F\rangle_V = D\Big(\frac{1}{2}\sum_{i,j}\psi_{ij}\Big)(V)[F],$$

for any $F \in \mathbb{R}^{k\times n}$, which, in view of Lemma 4, can be rewritten as

$$\sum_{j=1}^{n}\sum_{\ell=1}^{k}\frac{\nabla_{\ell j}^{(V)} f_{\ell j}}{v_{\ell j}^2} = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{\ell=1}^{k}\operatorname{vec}(K_{ij}^{-1}\log(K_{ij}M_{ij}^{-1}))^* \Gamma_\ell^{(ij)} f_{\ell j},$$

from which we finally get

$$\nabla_{\ell j}{}^{(V)} = \sum_{i=1}^{m} v_{\ell j}^2 \operatorname{vec}(K_{ij}^{-1}\log(K_{ij}M_{ij}^{-1}))^* \Gamma_\ell^{(ij)}, \tag{25}$$

for $\ell = 1,\dots,k$ and $j = 1,\dots,n$.

4.1.2 The Gradients of $\mathscr{L}(U, V)$

We consider the derivatives of the cost function $\mathscr{L}(U, V)$ of (22) as a function of the sole variables U and V, respectively. As before, we compute first the derivative of $\mathscr{L}$ with respect to U in the direction $E = (E_{st}) \in \mathbb{H}_\mu^{m\times k}$, from which we get the Riemannian gradient. By linearity, it is sufficient to consider the functions $\varphi_{ij}(U) = \|\sum_{\ell=1}^{k} v_{\ell j}\log U_{i\ell} - \log M_{ij}\|_F^2$, separately.

Lemma 5 *The Fréchet derivative of the function $\varphi_{ij}(U)$ in the direction $E = (E_{st})$ is*

$$D\varphi_{ij}(U)[E] = 2\sum_{\ell=1}^{k} v_{\ell j}\operatorname{vec}\Big(\sum_{q=1}^{k} v_{qj}\log U_{iq} - \log M_{ij}\Big)^* K_{\log}(U_{i\ell})\operatorname{vec}(E_{i\ell}),$$

where $K_{\log}(U_{i\ell})$ is the $\mu^2\times\mu^2$ Kronecker matrix of the derivative of the matrix logarithm.

Proof For a fixed V, we define $d_{ij}(X) := d_{\log M_{ij}}(X)$ and $\tau_{ij}(U) = \sum_{\ell=1}^{k} v_{\ell j}\log(U_{i\ell})$ for which $\varphi_{ij}(U) = (d_{ij}^2\circ\tau_{ij})(U)$. The derivative of τ_{ij} at U, in the direction E, is

$$D\tau_{ij}(U)[E] = \sum_{s,t} D\tau_{ij}(U)[E_{st}] = \sum_{\ell=1}^{k} D\tau_{ij}(U)[E_{i\ell}] = \sum_{\ell=1}^{k} v_{\ell j}\operatorname{vec}^{-1}(K_{\log}(U_{i\ell})\operatorname{vec}(E_{i\ell})),$$

then, by Lemma 1 and the chain rule, we get

$$
\begin{aligned}
D\varphi_{ij}(U)[E] &= Dd_{ij}^2(\tau_{ij}(U))[D\tau_{ij}(U)[E]] \\
&= 2\sum_{\ell=1}^{k} \operatorname{trace}\Big(\big(\sum_{q=1}^{k} v_{qj}\log U_{iq} - \log M_{ij}\big) v_{\ell j} \operatorname{vec}^{-1}\big(K_{\log}(U_{i\ell})\operatorname{vec}(E_{i\ell})\big)\Big) \\
&= 2\sum_{\ell=1}^{k} v_{\ell j} \operatorname{vec}\big(\sum_{q=1}^{k} v_{qj}\log U_{iq} - \log M_{ij}\big)^* K_{\log}(U_{i\ell})\operatorname{vec}(E_{i\ell}).
\end{aligned}
$$

Let $\nabla^{(U)}$ be the Riemannian gradient of the function $\frac{1}{2}\sum_{i,j}\varphi_{ij}(U)$, then

$$
\langle \nabla^{(U)}, E\rangle_U^{(R)} = D\Big(\frac{1}{2}\sum_{i,j}\varphi_{ij}\Big)(U)[E],
$$

for any $E \in \mathbb{H}_\mu^{m\times k}$, which, by Lemma 3, can be rewritten as

$$
\begin{aligned}
\sum_{i=1}^{m}\sum_{\ell=1}^{k} \operatorname{trace}(U_{i\ell}^{-1}\nabla_{i\ell}^{(U)}U_{i\ell}^{-1}E_{i\ell}) &= \sum_{i=1}^{m}\sum_{\ell=1}^{k} \operatorname{vec}(U_{i\ell}^{-1}\nabla_{i\ell}^{(U)}U_{i\ell}^{-1})^*\operatorname{vec}(E_{i\ell}) \\
&= \sum_{i=1}^{m}\sum_{\ell=1}^{k} \operatorname{vec}(\nabla_{i\ell}^{(U)})^*(U_{i\ell}^{-T}\otimes U_{i\ell}^{-1})^*\operatorname{vec}(E_{i\ell}) \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{\ell=1}^{k} v_{\ell j}\operatorname{vec}\big(\sum_{q=1}^{k} v_{qj}\log U_{iq} - \log M_{ij}\big)^* K_{\log}(U_{i\ell})\operatorname{vec}(E_{i\ell}),
\end{aligned}
$$

that yields

$$
\operatorname{vec}(\nabla_{i\ell}^{(U)}) = \sum_{j=1}^{n} v_{\ell j}(U_{i\ell}^T\otimes U_{i\ell})(K_{\log}(U_{i\ell}))^*\operatorname{vec}\big(\sum_{q=1}^{k} v_{qj}\log U_{iq} - \log M_{ij}\big), \tag{26}
$$

for $i = 1,\dots,m$ and $\ell = 1,\dots,k$.

We should compute also the derivative of $\mathscr{L}(U,V)$ of (22) with respect to V in the direction $F = (f_{st}) \in \mathbb{R}_\mu^{k\times n}$, from which we obtain the Riemannian gradient. By linearity, we consider the functions $\psi_{ij}(V) := \|\sum_{\ell=1}^{k} v_{\ell j}\log U_{i\ell} - \log M_{ij}\|_F^2$ separately, for each i and j.

Lemma 6 *The Fréchet derivative of the function $\psi_{ij}(V)$ in the direction $F = (f_{st})$ is*

$$D\psi_{ij}(V)[F] = 2\sum_{\ell=1}^{k} \operatorname{trace}\Big((\sum_{q=1}^{k} v_{qj} \log U_{iq} - \log M_{ij}) \log U_{i\ell}\Big) f_{\ell j}.$$

Proof The proof is similar to the one of Lemma 5.

Let $\nabla^{(V)}$ denote the Riemannian gradient of $\frac{1}{2}\sum_{i,j} \psi_{ij}(V)$. We have

$$\langle \nabla^{(V)}, F\rangle_V = D\Big(\frac{1}{2}\sum_{i,j} \psi_{ij}\Big)(V)[F],$$

for any $F \in \mathbb{R}^{k\times n}$, which, by Lemma 4, is

$$\sum_{j=1}^{n}\sum_{\ell=1}^{k} \frac{\nabla_{\ell j}^{(V)} f_{\ell j}}{v_{\ell j}^2} = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{\ell=1}^{k} \operatorname{trace}\Big((\sum_{q=1}^{k} v_{qj} \log U_{iq} - \log M_{ij}) \log U_{i\ell}\Big) f_{\ell j},$$

from which we finally get

$$\nabla_{\ell j}{}^{(V)} = \sum_{i=1}^{m} v_{\ell j}^2 \operatorname{trace}\Big((\sum_{q=1}^{k} v_{qj} \log U_{iq} - \log M_{ij}) \log U_{i\ell}\Big), \tag{27}$$

for $\ell = 1, \ldots, k$ and $j = 1, \ldots, n$.

5 Numerical Experiments

In order to compare the performance of all presented models, we examine the speed and accuracy of the algorithms in some numerical tests. The first test will handle a basic example in which the columns of the grid M are exactly one of two possible grid vectors. In the second experiment, a new dataset is constructed based on the natural flow of the Riemannian geometry of positive matrices, and in a final experiment, we perform the decomposition of data without any predefined structure and observe the difference in computational time of the algorithms.

In the following, we will refer to the minimizer of (19) as the Euclidean model, to the minimizer of (21) as the matrix geometric mean model, and to the minimizer of (22) as the log-Euclidean model.

The accuracy of the reconstruction is measured by

$$\text{err} = \left(\frac{\sum_{i,j} \|\widetilde{M}_{ij} - M_{ij}\|^2}{\sum_{i,j} \|M_{ij}\|^2} \right)^{1/2},$$

where $(\widetilde{M}_{ij})_{i,j}$ is the reconstructed tensor grid, while $(M_{ij})_{i,j}$ is the original one.

5.1 Basic Dataset

In this first experiment, we use a dataset M as shown in Fig. 1 and discussed in Xie et al. [24]. Looking at the representation of the data, it is clear that each column in the grid is exactly one of two possibilities, hence we use a decomposition with $k = 2$, indicating two columns in the grid U.

The experiment is conducted by applying noise to the grid in Fig. 1 and performing the decomposition. However, because the columns in the dataset do not combine the given columns in grid U, but rather select one of both, the chosen model (Euclidean, matrix geometric, or log-Euclidean), in this case, does not influence the resulting accuracy of the approximation significantly.

5.2 Geometrically Varying Dataset

As before, we consider a dataset in which the original, underlying grid U consists of two columns. This time, however, the dataset M is not constructed by choosing

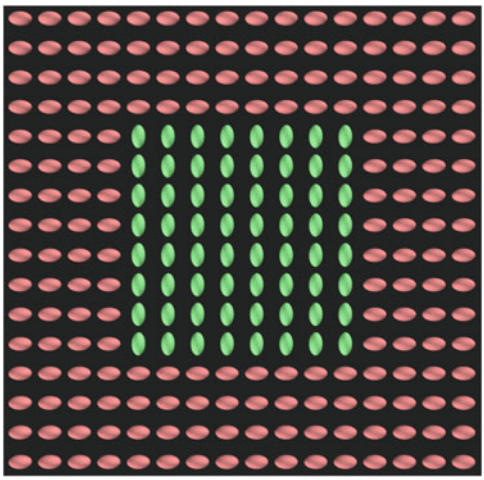


Fig. 1 The dataset presented in Xie et al. [24]. The 3×3 matrix at each gridpoint is represented by an ellipsoid using its eigendecomposition. The color is given by the direction of the principal eigenvector. The picture has been obtained using the program TenVis by R. Parcus (https://github.com/quiuquio/TenVis)

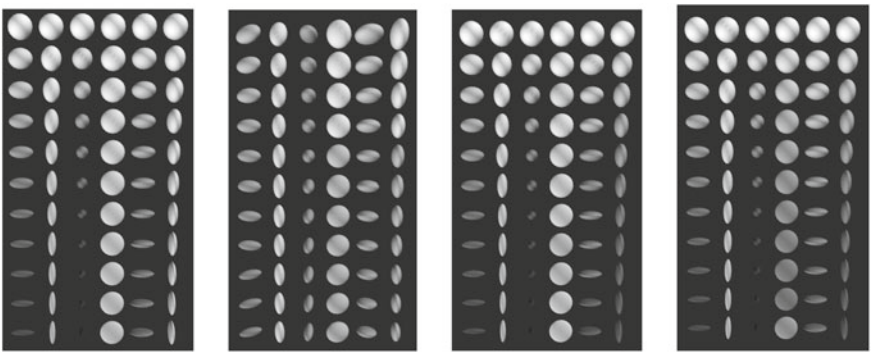


Fig. 2 Accuracy test using data varying according to the Riemannian structure. The 3×3 matrix at each gridpoint is represented by an ellipsoid using its eigendecomposition. (**a**) The original, geometrically varying data. (**b**) The resulting approximation using the Euclidean model. (**c**) The resulting approximation using the matrix geometric model. (**d**) The resulting approximation using the log-Euclidean model

the columns of U separately, but by combining them using the weighted matrix geometric mean with variable weights. This results in a grid M in which the rows are formed by positive matrices varying according to the geometry of $\mathscr{P}_\mu$. Some noise is applied to the matrices in M to prevent the matrix geometric model from being trivial and the resulting dataset is shown in Fig. 2a.

The error in the reconstruction for the Euclidean model is 0.21, while for the matrix geometric and log-Euclidean models is 0.054 and 0.044, respectively. The results for the Euclidean, matrix geometric, and log-Euclidean models are shown in Fig. 2b–d, respectively. The pictures confirm the numerical accuracy of the reconstruction. As expected, the matrix geometric model gives a good approximation of the original dataset, since the data were created based on the underlying geometry of this model. The close connection between the log-Euclidean mean and this geometry causes the log-Euclidean model to give very similar results.

On the other hand, the Euclidean model suffers from the connection between the variation of the data and the geometry of the set $\mathscr{P}_\mu$, especially when the data approach the boundary of $\mathscr{P}_\mu$. A closer look at the iterations in the computation of the Euclidean model reveals that the model attempts to obtain some matrices in the grid U which are no longer positive, violating the assumptions of the nonnegative matrix factorization. Removing this condition on the matrices in U results in a Euclidean model with similar accuracy to the matrix geometric mean and log-Euclidean models, but with less significance since some of the matrices in U would not be positive definite.

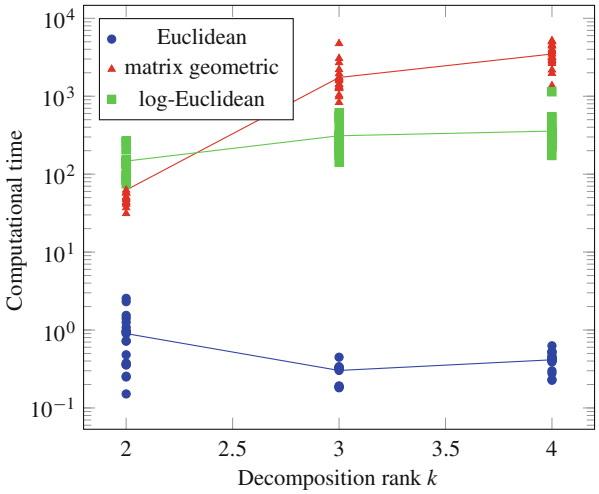


Fig. 3 Computational time for the three decomposition when applied to a 15×15 grid M containing random positive matrices. The experiment is repeated 20 times for each decomposition rank k, with the line connecting the mean computational time

5.3 Speed Test

Finally, we compare the computational time of the different decompositions by repeatedly creating a 15×15 grid M consisting of random positive (3×3) matrices. The results of the experiments are displayed in Fig. 3.

As expected, the matrix geometric and log-Euclidean decompositions require more computational time when compared to the Euclidean model. This is mainly caused by the evaluation of the more involved distance measure and non-trivial matrix functions, and more importantly, by the evaluation of their derivatives. At each iteration step, the current approximation for the grid M also needs to be computed, requiring the evaluation of mn matrix geometric/log-Euclidean means (e.g., 225 in this experiment).

The difference between the matrix geometric and log-Euclidean decompositions becomes clear when examining the evolution of the computational time going from decomposition rank $k = 2$ to $k = 3$. While the matrix geometric mean has an explicit expression for two matrices, it is computed in an iterative process for three or more matrices. This difference greatly influences the amount of work required to evaluate the mean and to compute its derivatives (see also Sect. 3.2). The log-Euclidean mean on the other hand is given by an explicit expression for any number of matrices.

6 Conclusions

We have obtained an expression for the derivative of the weighted matrix geometric mean that can be easily evaluated by a numerical algorithm. As a possible application, we have presented new models for the decomposition of tensor grids based on the non-Euclidean geometry defining the matrix geometric mean.

From some preliminary tests these new models seem to be promising. A future step could be to test these models on real data and to find faster algorithms for computing the decomposition.

Acknowledgements The authors would like to thank the referees for carefully reading the manuscript, providing many insightful comments which improved the presentation of the chapter.

References

1. Batchelor, P.G., Moakher, M., Atkinson, D., Calamante, F., Connelly, A.: A rigorous framework for diffusion tensor calculus. Magn. Reson. Med. **53**(1), 221–225 (2005)
2. Bhatia, R.: Positive Definite Matrices. Princeton Series in Applied Mathematics. Princeton University Press, Princeton (2007)
3. Bhatia, R., Holbrook, J.: Riemannian geometry and matrix geometric means. Linear Algebra Appl. **413**(2–3), 594–618 (2006)
4. Bini, D.A., Iannazzo, B.: A note on computing matrix geometric means. Adv. Comput. Math. **35**(2–4), 175–192 (2011)
5. Bini, D.A., Iannazzo, B.: Computing the Karcher mean of symmetric positive definite matrices. Linear Algebra Appl. **438**(4), 1700–1710 (2013)
6. Bini, D.A., Iannazzo, B., Jeuris, B., Vandebril, R.: Geometric means of structured matrices. BIT **54**(1), 55–83 (2014)
7. Cheng, G., Vemuri, B.C.: A novel dynamic system in the space of SPD matrices with applications to appearance tracking. SIAM J. Imaging Sci. **6**(1), 592–615 (2013)
8. Congedo, M., Barachant, A., Bhatia, R.: Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review. Brain-Comput. Interfaces **4**(3), 155–174 (2017)
9. Fletcher, P.T., Lu, C., Pizer, S.M., Joshi, S.: Principal geodesic analysis for the study of nonlinear statistics of shape. IEEE Trans. Med. Imaging **23**(8), 995–1005 (2004)
10. Higham, N.J.: Functions of Matrices: Theory and Computation. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2008)
11. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, Cambridge (1994)
12. Iannazzo, B.: The geometric mean of two matrices from a computational viewpoint. Numer. Linear Algebra Appl. **23**(2), 208–229 (2016)
13. Iannazzo, B., Porcelli, M.: The Riemannian Barzilai–Borwein method with nonmonotone line search and the matrix geometric mean computation. IMA J. Numer. Anal. **38**(1), 495–517 (2018)
14. Jeuris, B., Vandebril, R., Vandereycken, B.: A survey and comparison of contemporary algorithms for computing the matrix geometric mean. Electron. Trans. Numer. Anal. **39**, 379–402 (2012)
15. Lang, S.: Fundamentals of Differential Geometry, Graduate Texts in Mathematics, vol. 191. Springer, New York (1999)

16. Lawson, J., Lim, Y.: Weighted means and Karcher equations of positive operators. Proc. Natl. Acad. Sci. U. S. A. **110**(39), 15626–15632 (2013)
17. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Leen, T., Dietterich, T., Tresp, V. (eds) Advances in Neural Information Processing Systems, vol. 13, pp. 556–562. MIT Press, Cambridge (2001)
18. Moakher, M.: On the averaging of symmetric positive-definite tensors. J. Elasticity **82**(3), 273–296 (2006)
19. Moakher, M., Zéraï, M.: The Riemannian geometry of the space of positive-definite matrices and its application to the regularization of positive-definite matrix-valued data. J. Math. Imaging Vision **40**(2), 171–187 (2011)
20. Nesterov, Y.E., Todd, M.J.: On the Riemannian geometry defined by self-concordant barriers and interior-point methods. Found. Comput. Math. **2**(4), 333–361 (2002)
21. Rentmeesters, Q., Absil, P.A.: Algorithm comparison for Karcher mean computation of rotation matrices and diffusion tensors. In: 2011 19th European Signal Processing Conference, pp. 2229–2233. IEEE (2011)
22. Sra, S., Dhillon, I.S.: Nonnegative Matrix Approximation: Algorithms and Applications. Technical Report TR-06-27, The University of Texas at Austin, June (2006)
23. Wang, Y., Salehian, H., Cheng, G., Vemuri, B.C.: Tracking on the product manifold of shape and orientation for tractography from diffusion MRI. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3051–3056. IEEE (2014)
24. Xie, Y., Ho, J., Vemuri, B.: Nonnegative factorization of diffusion tensor images and its applications. In: International Conference on Information Processing in Medical Imaging (IPMI) (2011)

Factoring Block Fiedler Companion Matrices

Gianna M. Del Corso, Federico Poloni, Leonardo Robol, and Raf Vandebril

Abstract When Fiedler published his "*A note on Companion matrices*" in 2003 on Linear Algebra and its Applications, he could not have foreseen the significance of this elegant factorization of a companion matrix into essentially two-by-two Gaussian transformations, which we will name *(scalar) elementary Fiedler factors*. Since then, researchers extended these results and studied the various resulting linearizations, the stability of Fiedler companion matrices, factorizations of block companion matrices, Fiedler pencils, and even looked at extensions to non-monomial bases. In this chapter, we introduce a new way to factor block Fiedler companion matrices into the product of scalar elementary Fiedler factors. We use this theory to prove that, e.g. a block (Fiedler) companion matrix can always be written as the product of several scalar (Fiedler) companion matrices. We demonstrate that this factorization in terms of elementary Fiedler factors can be used to construct new linearizations. Some linearizations have notable properties, such as low bandwidth, or allow for factoring the coefficient matrices into unitary-plus-low-rank matrices. Moreover, we will provide bounds on the low-rank parts of the resulting unitary-plus-low-rank decomposition. To present these results in an easy-to-understand manner, we rely on the flow-graph representation for Fiedler matrices recently proposed by Del Corso and Poloni in Linear Algebra and its Applications, 2017.

Keywords Companion matrix · Fiedler linearizations

G. M. Del Corso (✉) · F. Poloni
University of Pisa, Dept. Computer Science, Pisa, Italy
e-mail: gianna.delcorso@unipi.it; federico.poloni@unipi.it

L. Robol
Institute of Information Science and Technologies "A. Faedo", ISTI-CNR, Pisa, Italy
e-mail: leonardo.robol@isti.cnr.it

R. Vandebril
University of Leuven (KU Leuven), Dept. Computer Science, Leuven, Belgium
e-mail: raf.vandebril@cs.kuleuven.be

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_7

1 Introduction

It is well known that, given a monic polynomial $p(z) = z^d + a_{d-1}z^{d-1} + \cdots + a_0$, we can build a *(column) companion matrix*[1] that has the roots of $p(z)$ as eigenvalues and whose entries are just 1, 0, and the coefficients of $p(z)$:

$$\Gamma_p := \begin{bmatrix} & & -a_0 \\ 1 & & -a_1 \\ & 1 & -a_2 \\ & \ddots & \vdots \\ & & 1 \; -a_{d-1} \end{bmatrix}. \tag{1}$$

We remark that constructing a companion matrix is *operation-free*: no arithmetic operations are needed to get Γ_p from p. The pencil $zI - \Gamma_p$ is an example of a *linearization* for $p(z)$. A formal definition of a linearization is the following.

Definition 1 Let $p(z)$ be a $k \times k$ degree d matrix polynomial. Then, the pencil $A - zB$ is a linearization of $p(z)$ if there exist two unimodular matrices $E(z)$ and $F(z)$ (i.e. matrix polynomials with non-zero constant determinant) such that $I_{k(d-1)} \oplus p(z) = E(z)(A - zB)F(z)$.

In the above setting, when $B = I$, we say that A is a *companion matrix*.[2] In the rest of the paper, we will never deal with the matrices $E(z)$ and $F(z)$ directly. For us, it is sufficient to know that the column companion matrix identifies a linearization, and that any matrix similar to it still leads to a linearization (see, for instance, [25]).

The fact that a linearization is operation-free can be formalized as follows:

Definition 2 A companion matrix C of a polynomial $p(z) = a_0 + a_1 z + \cdots + z^d$ is called *operation-free* if each of the elements in C is either 0, 1, or one of the scalars a_j (possibly with a minus sign). Similarly, for a block companion matrix linearizing a matrix polynomial, we say that it is *operation-free* if its entries are either 0, 1, or one entry in the coefficients of the matrix polynomial (possibly with a minus sign).

In 2003, Fiedler showed that Γ_p in (1) can be factored as the product of d *(scalar) elementary Fiedler factors* which are equal to the identity matrix with the only exception of a 1×1 or 2×2 diagonal block [27]. This factorization has a remarkable consequence: the product of these factors in *any order* provides still a linearization for $p(z)$, since its characteristic polynomial remains $p(z)$. Companion

[1] We typically abbreviate column companion matrix and omit the word column, unless we want to emphasize it.

[2] Often, the term companion matrix indicates a matrix obtained from the coefficients of the polynomial without performing arithmetic operations. Here, we have not added this constraint into the definition but—as we will discuss later—all the matrices obtained in our framework satisfy it.

matrices resulting from permuting the elementary Fiedler factors are named *Fiedler companion matrices*.

This theory has then been extended to matrix polynomials, by operating block-wise, and to more general constructions than just permutations of the original factors, which led to Fiedler pencils with repetitions [16, 36], generalized Fiedler pencils [2, 18] and generalized Fiedler pencils with repetitions [20]. These ideas sparked the interest of the numerical linear algebra community: several researchers have tried to find novel linearizations in this class with good numerical properties [26, 28], or which preserve particular structures [17, 23, 26, 33].

The construction of Fiedler companion matrices is connected with permutations of $\{0, \ldots, d-1\}$. In this framework, the development of explicit constructions for palindromic, even–odd, and block-symmetric linearizations in the Fiedler class is investigated in [17, 20, 23, 26]. At the same time, several authors have investigated vector spaces of linearizations with particular structures [29, 30], and linearizations with good numerical properties [11, 21, 35]. Recently, a new graph-based classification of Fiedler pencils has been recently introduced by Poloni and Del Corso [31], and has been used to count the number of Fiedler pencils inside several classes, as well as to describe common results in a simpler way.

The aim of this paper is to extend the theory proposed in [31] by introducing manipulations that operate *inside* the blocks and factor them, but at the same time remain *operation-free*, which is a key property of Fiedler-like pencils.

We show that these tools can be used to construct new factorizations of block Fiedler companion matrices. In particular, we prove that (under reasonable assumptions on the constant coefficient) any block Fiedler companion matrix of a monic $k \times k$ matrix polynomial can be factored into k Fiedler companion matrices of *scalar* polynomials. This approach extends a similar factorization for column companion matrices by Aurentz, Mach, Robol, Vandebril, and Watkins [7]. The graph-based representation makes it easy to visualize the unitary-plus-low-rank structure of (block) Fiedler companion matrices; it also provides upper bounds on the rank in the low-rank correction of the unitary-plus-low-rank matrix.

Aurentz et al. [7] developed a fast method to compute eigenvalues of matrix polynomials by factoring the column block companion matrix into scalar companion matrices and then solving a product eigenvalue problem exploiting the unitary-plus-rank-1 structure of the factors. As we will show in Theorem 8, some block Fiedler companion matrices can be similarly factored as a product of row and column companion matrices (appropriately padded with identities). This makes the algorithm in [7] applicable to devise a fast solver; in fact, this idea is exploited in [6] to describe an algorithm for computing the eigenvalues of unitary-plus-rank-k matrices. As an alternative, after a preliminary reduction of the matrix to Hessenberg form we can employ the algorithm proposed in [10] for generic unitary-plus-rank-k matrices.

As a final application, to illustrate the power of the new representation, we show by an example that these techniques can easily produce novel companion matrices such as thin band or factorizations in which symmetry of the original problem is reflected.

Throughout the chapter, we adopt the following notation: I_n denotes the identity matrix of size $n \times n$; its subscript is dropped whenever its size is clear from the context; e_j denotes the j-th vector of the canonical basis of $\mathbb{C}^n$; and Z denotes the downshift matrix, having ones on the first subdiagonal and zeros elsewhere.

2 Fiedler Graphs and (Block) Fiedler Companion Matrices

As mentioned in the introduction, Fiedler [27] showed that the column companion matrix Γ_p in (1) can be factored as $\Gamma_p = F_0 F_1 \cdots F_{d-1}$, where the F_i, named *elementary Fiedler factors*, are matrices which are equal to the identity except for a diagonal block of size at most 2×2. More precisely:

$$F_0 = F_0(a_0) = (-a_0) \oplus I_{d-1}, \qquad F_i = F_i(a_i) = I_{i-1} \oplus \begin{bmatrix} 0 & 1 \\ 1 & -a_i \end{bmatrix} \oplus I_{d-i-1}. \tag{2}$$

We omit the parameters in parentheses if they are clear from context.

The key result due to Fiedler [27] is that any permutation of the factors in the above factorization $C = F_{\sigma(0)} \cdots F_{\sigma(d-1)}$, where σ is a permutation of $\{0, \ldots, d-1\}$, is still a companion matrix for $p(z)$. We call linearizations obtained in this way *Fiedler linearizations* (and the associated matrices *Fiedler companion matrices*). Throughout the chapter, we use the letter Γ (with various subscripts) to denote a column (or, occasionally, row) companion matrix, possibly padded with identities, i.e. $I_{h_1} \oplus \Gamma_p \oplus I_{h_2}$, and the letter C to denote Fiedler companion matrices. We will heavily rely on the flow-graph representation established by Poloni and Del Corso [31], so we introduce it immediately.

The *elementary flow graph* associated to each elementary Fiedler factor is the graph shown in Fig. 1.

To construct the *Fiedler graph* associated to a product of Fiedler elementary factors $P = F_{i_1}(a_1) \cdots F_{i_k}(a_k)$, each of size $d \times d$, we first draw d horizontal lines labelled with the integers $1, \ldots, d$ (with 1 at the top); this label is called the *height* of a line. Then, we stack horizontally (in the same order in which they appear in P) the graphs corresponding to the elementary factors. These must be properly aligned vertically so that F_{i_j} touches the lines at heights i_j and $i_j + 1$ (or $i_j + 1$ only

Fig. 1 Elementary flow graphs corresponding to $F_i(a)$ (for $i > 0$) and to $F_0(a)$

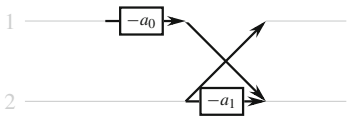


Fig. 2 Flow graph for $P = F_0(a_0)F_1(a_1)$

if $i_j = 0$). Straight horizontal edges are drawn in grey to distinguish them from the edges of the flow graph.

A Fiedler graph is a representation of multiplying a row vector v, having components v_i, with P. This vector–matrix multiplication can be seen as a composition of additions and multiplications by scalars, and the flow graph depicts these operations, as follows. We imagine that for each i the entry v_i of a row vector v enters the graph from the left along the edge at height i and moves towards right. A scalar traveling from left to right through an edge with a box carrying the label a is multiplied by a before it proceeds; an element following a straight edge (with no box) is left unchanged; a scalar arriving at a node with two outgoing edges is duplicated; and finally when two edges meet the corresponding values are added. If one carries on this process, the result at the right end of the graph are the entries of vP, with the jth entry appearing at height j.

Example 1 Consider the simple case in which $d = 2$, and $P = F_0(a_0)F_1(a_1)$. The flow graph associated to P is shown in Fig. 2.

The element v_1 enters from the left at the first row, hits $-a_0$ resulting in a multiplication $-v_1a_0$, and then moves down to the second row. The element v_2 enters at the second row and is duplicated. Its first clone moves to the top row, and its second clone gets multiplied with $-a_1$ and then also ends up in the second row. Since both $-v_1a_0$ and $-v_2a_1$ end up in the bottom row, we add them together. As a result, we obtain $[v_2,\ -v_2a_1 - v_1a_0]$ at the right end of the graph, which is exactly $[v_1,\ v_2]P$.

The power of this framework lies in the fact that representing a matrix by a Fiedler graph enables us to easily draw conclusions on the structure of the matrix and its associated elementary Fiedler factors. For instance, to determine the content of the $(i,\ j)$-th entry of a product of Fiedler factors it is sufficient to inspect the paths on the graph that start from the left at height i and end on the right at height j.

Consider for instance the column companion matrix (1) of degree $d = 4$. The associated Fiedler graph is depicted in Fig. 3. Indeed, entering the graph from the left on row $i > 1$ yields two elements: one on column $i - 1$ (the edge pointing one row up), and the other is $-a_{i-1}$, which follows a descending path until the very last column. This implies that

$$e_i^T \Gamma_p = \begin{bmatrix} 0_{i-2} & 1 & 0_{d-i} & -a_{i-1} \end{bmatrix}, \qquad i > 1,$$

which is exactly the i-th row of Γ_p. The case $i = 1$ produces the row vector

$$e_1^T \Gamma_p = \begin{bmatrix} 0_{d-1} & -a_0 \end{bmatrix}$$

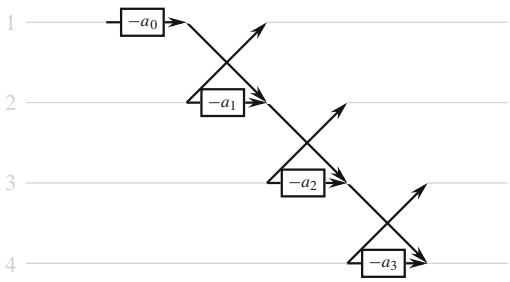


Fig. 3 Fiedler graph associated to a column companion matrix of a degree 4 monic polynomial

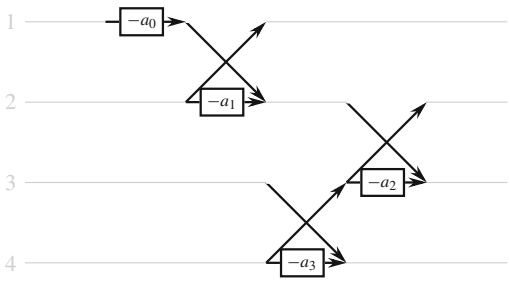


Fig. 4 Fiedler graph associated to the matrix $F = F_0F_1F_3F_2$

Some Fiedler companion matrices are simpler, some are more complex. For instance, we have already considered $F_0 \cdots F_{d-1}$, which is the usual column companion matrix. The transpose of this matrix is $F_{d-1} \cdots F_0$ (all the Fiedler factors are symmetric), which is a Fiedler companion matrix with all the coefficients on the last row:

$$\Gamma_p^T = F_{d-1} \cdots F_0 = \begin{bmatrix} & 1 & & \\ & & \ddots & \\ & & & 1 \\ -a_0 & -a_1 & \dots & -a_{d-1} \end{bmatrix}. \tag{3}$$

We refer to (3) as a *row companion* matrix.[3]

The flow graphs help us to visualize the structure of the factorization in elementary Fiedler factors. For example, the Fiedler companion matrix with $\sigma = (0, 1, 3, 2)$ is associated with the graph in Fig. 4. Note that the order of the elementary flow graphs coincides with the order of the elementary Fiedler factors.

Since F_i and F_j commute whenever $|i - j| > 1$, *different* permutations σ may correspond to the *same* matrix. For example, in Fig. 4, F_1 and F_3 commute, so $F_0F_1F_3F_2 = F_0F_3F_1F_2$. In terms of graphs, we can "compress" a Fiedler graph by drawing the elements F_i and F_j one on top of the other whenever $|i - j| > 1$,

[3]This is a variation on the usual construction of a row companion matrix having the elements a_i in its first row.

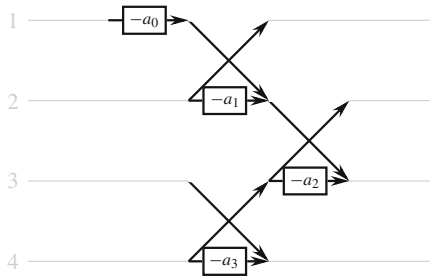


Fig. 5 A more compact representation of the graph in Fig. 4

or swap them; these are operations that do not alter the topological structure of the graph nor the height of each edge.

For example, we can draw the graph in Fig. 4 in a compact way as in Fig. 5. Moreover, we can immediately read off the following equivalences $F_0F_3F_1F_2 = F_0F_1F_3F_2 = F_3F_0F_1F_2$, since the factor F_3 is free to slide to the left of the diagram.

If we allow for repositioning factors in this way, two Fiedler companion matrices coincide if and only if their graphs do (see [31] for a detailed analysis of this characterization).

Remark 1 There are a number of different "standard forms" [31, 36], i.e. canonical ways to order the factors in a Fiedler product or draw the corresponding graph. In this chapter, we do not follow any of them in particular. Rather, when drawing the graph associated to a Fiedler companion matrix C, we try to draw them so that the elements form a connected twisted line. (In practice, this can be obtained by drawing first the elementary factor F_{d-1} at the bottom of the graph, and then F_{d-2}, F_{d-3}, ... each immediately at the left or right of the last drawn element.) This choice gives a better visual interpretation of some of our results; See for instance the discussion after Theorem 6.

We now generalize this construction to monic matrix polynomials. Given a degree-d matrix polynomial with $k \times k$ coefficients

$$P(z) = Iz^d + A_{d-1}z^{d-1} + \cdots + A_0 \in \mathbb{C}^{k\times k}[z],$$

we can factor its column companion matrix as:

$$\Gamma_P = \begin{bmatrix} 0 & 0 & \cdots & 0 & -A_0 \\ I_k & 0 & & 0 & -A_1 \\ & I_k & & 0 & -A_2 \\ & & \ddots & \vdots & \vdots \\ & & & I_k & -A_{d-1} \end{bmatrix} = \mathscr{F}_0\mathscr{F}_1\cdots\mathscr{F}_{d-1},$$

Fig. 6 Graph representing the active part of the block elementary Fiedler factor $\mathscr{F}_i(A)$, for $i > 0$ and of $\mathscr{F}_0(A)$

where all $\mathscr{F}_i$ are *block elementary Fiedler factors*, that is

$$\mathscr{F}_0 = \mathscr{F}_0(A_0) = (-A_0) \oplus I_{k(d-1)}, \qquad \mathscr{F}_i = \mathscr{F}_i(A_i) = I_{(i-1)k} \oplus \begin{bmatrix} 0 & I \\ I & -A_i \end{bmatrix} \oplus I_{(d-i-1)k},$$

for all $i = 0, \ldots, d-1$. Again, each permutation of these factors gives a *(block) Fiedler companion matrix*. We can construct graphs associated to their products in the same way; the entities we operate on are now matrices instead of scalar entries. For instance, for $A \in \mathbb{C}^{k\times k}$, the active part of a block elementary Fiedler factor, which is the diagonal block differing from the identity, can be represented as in Fig. 6. All the results of this section concerning the reordering of the block elementary Fiedler factors remain valid also in the block case. In particular, block Fiedler flow graphs represent the multiplication of a "block row vector" $[V_1, V_2, \ldots, V_d] \in \mathbb{C}^{k\times kd}$ by a product of block Fiedler matrices.

This construction can be thought of as the "blocked" version of the one we had in the scalar case: we treat the blocks as atomic elements, which we cannot inspect nor separate. However, it does not have to be that way, and in the next section we will explore the idea of splitting these blocks into smaller pieces.

In particular, we will show in Sect. 3.1 how each of the block elementary Fiedler factors can be decomposed as a product of (scalar) elementary Fiedler factors. So, we have a coarse (block) level factorization and graph, and a fine (entry) level factorization and corresponding graph.

3 Factoring Elementary Block Fiedler Factors

We discuss the block Fiedler factors $\mathscr{F}_i$ for $i > 0$ and $i = 0$ in different subsections because they require different treatments.

3.1 Block Factors $\mathscr{F}_i$, for $i > 0$

To ease readability and avoid additional notation we will use, in this section, $\mathscr{F}(A)$ to denote the active part (the one different from the identity) of an arbitrary block

Fiedler factor $\mathscr{F}_i(A)$, for $i > 0$. In particular, we have

$$\mathscr{F}(A) := \begin{bmatrix} 0_{k\times k} & I_k \\ I_k & -A \end{bmatrix} \in \mathbb{C}^{2k\times 2k}. \tag{4}$$

Consider the graph of a single Fiedler factor $\mathscr{F}(A)$ given in the left of Fig. 6. This graph represents the multiplication of $\mathscr{F}(A)$ by a block row vector $[V_1, V_2]$, so the two horizontal levels in the graph correspond to the blocks $1 : k$ and $k + 1 : 2k$ (in Fortran/Matlab notation).

We show that it can be converted into a more fine-grained graph in which each line represents a single index in $\{1, 2, \ldots, 2k\}$. We call this construction a *scalar-level graph* (as opposed to the *block-level* graph appearing in Fig. 6).

We first show the result of this construction using flow graphs, to get a feeling of what we are trying to build.

Example 2 Let $k = 3$, and $A = (a_{ij})$, with $1 \le i, j \le 3$. In this case, the elementary block factor $\mathscr{F}(A)$ in Fig. 6 has size 6×6. A scalar-level graph associated to $\mathscr{F}(A)$ is depicted in Fig. 7.

Theorem 1 *Let $\mathscr{F}(A) \in \mathbb{C}^{2k\times 2k}$ be a block elementary Fiedler factor as defined by Eq. (4). Then, $\mathscr{F}(A)$ can be factored into k^2 scalar elementary Fiedler factors associated to the elements of the matrix $A = (a_{ij})$, as follows*

$$\mathscr{F}(A) = \Gamma_k \Gamma_{k-1} \cdots \Gamma_1,$$
$$\Gamma_j = F_j(a_{1j}) F_{j+1}(a_{2j}) \cdots F_{j+k-1}(a_{kj}), \quad j = 1, 2, \ldots, k.$$

Proof From a linear algebra viewpoint, the proof can be obtained simply multiplying the various factors together. Alternatively, one can construct the $2k \times 2k$

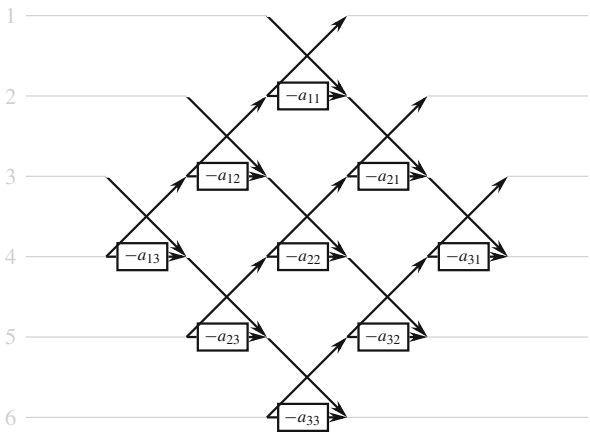


Fig. 7 Scalar-level graph associated to $\mathscr{F}(A)$ where A is a 3×3 matrix with entries a_{ij}

analogue of Fig. 7, and follow the edges of the graph to check the value of each matrix element. □

Remark 2 Each Γ_j is a (scalar) column companion matrix padded with identities, i.e. it has the form $I_{j-1} \oplus \Gamma_{a_j} \oplus I_{k-j}$, where Γ_{a_j} is a particular column companion matrix of size $k+1$. Indeed

$$\Gamma_j = \begin{bmatrix} I_{j-1} & & \\ & Z + a_j e_{k+1}^T & \\ & & I_{k-j} \end{bmatrix}, \qquad a_j := \begin{bmatrix} 1 \\ -Ae_j \end{bmatrix},$$

where Z is the downshift matrix, with ones on the first subdiagonal and zero elsewhere. In the following, we call *column (resp., row) companion matrices* also matrices with this form, ignoring the additional leading and trailing identities.

We could have proved that $\mathscr{F}(A)$ is the product of k column companion matrices also by inspecting the associated scalar-level graph. Indeed, for simplicity let us restrict ourselves to the running example in Fig. 7 of size 6×6. We replot the graph in Fig. 7 inserting gaps between some elements. Hence, the graph is the concatenation of three sequences of factors that can be arranged in a descending diagonal line each. These correspond precisely to Γ_3, Γ_2, Γ_1; indeed, any descending line of diagonal factors forms a column companion matrix (padded with identities) (Fig. 8).

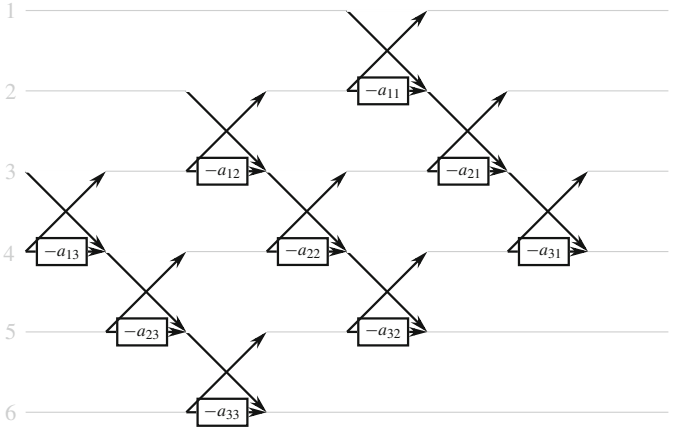


Fig. 8 Replot of the graph in Fig. 7 with gaps between descending diagonal lines. This reveals the factorization into companion matrices

Example 3 Written explicitly, the factors that compose the 6×6 matrix $\mathscr{F}(A)$ of our running example are

$$\mathscr{F}(A) = \left[\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & -a_{11} & -a_{12} & -a_{13} \\ 0 & 1 & 0 & -a_{21} & -a_{22} & -a_{23} \\ 0 & 0 & 1 & -a_{31} & -a_{32} & -a_{33} \end{array}\right] = \begin{bmatrix} 0 & I \\ I & -A \end{bmatrix} = \Gamma_3 \Gamma_2 \Gamma_1$$

$$= \left[\begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -a_{13} \\ 0 & 0 & 0 & 1 & -a_{23} \\ 0 & 0 & 0 & 0 & -a_{33} \end{array}\right] \left[\begin{array}{c|ccccc|c} 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -a_{12} & 0 \\ 0 & 0 & 1 & 0 & -a_{22} & 0 \\ 0 & 0 & 0 & 1 & -a_{32} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \left[\begin{array}{ccc|cc} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -a_{11} & 0 & 0 \\ 0 & 1 & 0 & -a_{21} & 0 & 0 \\ 0 & 0 & 1 & -a_{31} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

Remark 3 We can also factor $\mathscr{F}(A)$ into row companion matrices. If we group the elements in Fig. 7 following ascending diagonals, we obtain an analogous factorization, shown in Fig. 9, into row companion matrices, each containing entries from one row of A. This new decomposition can be identified from the graph in Fig. 9. This result is immediately obtained by applying Theorem 1 to $\mathscr{F}(A)^T$.

Remark 4 Poloni and Del Corso [31], only consider elementary blocks of the form $\mathscr{F}_i = \mathscr{F}_i(A_i)$, where A_i is a coefficient of the matrix polynomial $P(z) = Iz^d + A_{d-1}z^{d-1} + \cdots + zA_1 + A_0$. Here, we allow for a more general case, where

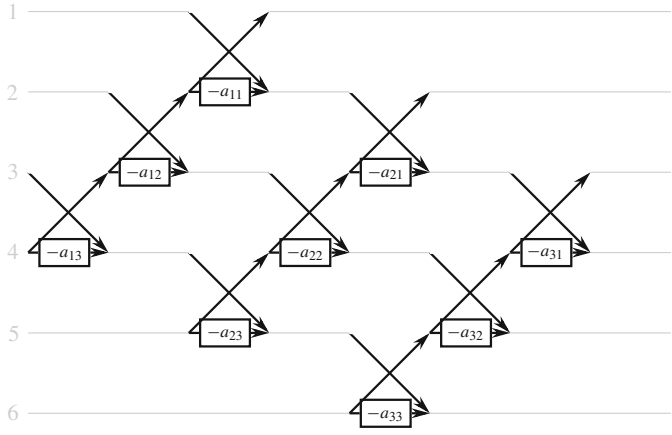


Fig. 9 Replot of the graph in Fig. 7 adding gaps between ascending diagonal lines

two elementary blocks $F_i(a)$ and $F_i(b)$ with the same index i can have different parameters $a \neq b$.

3.2 The Block Fiedler Factor $\mathscr{F}_0$

In this section, we provide a factorization for the block $\mathscr{F}_0$ into the product of scalar companion matrices. Note that the active part of the elementary Fiedler factor $\mathscr{F}_0$ is confined to the first k rows instead of $2k$ rows like the factors $\mathscr{F}_i$ for $i > 0$.

Since our goal is to build linearizations of matrix polynomials, we can perform a preliminary transformation that does not alter the spectrum. If there exist two invertible matrices E, G, such that $EP(\lambda)G = Q(\lambda)$, then the matrix polynomials $P(\lambda)$ and $Q(\lambda)$ are said to be *strictly equivalent* [25]. When this happens, their spectra (both finite and infinite) coincide. If the matrices E and G are also unitary, then the condition number of their eigenvalues also matches,[4] hence we need not worry about instabilities resulting from using this factorization.

In particular, we can choose an orthogonal (resp., unitary) matrix E and let $G = E^T\Pi$ (resp., $G = E^H\Pi$), where Π is the counter-identity matrix, so that the monic matrix polynomial $P(\lambda)$ is transformed into a monic polynomial $Q(\lambda)$ with A_0 lower anti-triangular (i.e. $(A_0)_{i,j} = 0$ whenever $i + j \leq k$). These unitary matrices can be obtained by computing the Schur form of $A_0 = QTQ^T$, and then setting $E = Q^T$. For these reasons, we may assume that A_0 is lower anti-triangular.

Theorem 2 *Let $A \in \mathbb{C}^{k\times k}$ be a lower anti-triangular matrix. Then, $\mathscr{F}_0(A)$ can be factored as the product of $\frac{k(k+1)}{2}$ scalar elementary Fiedler factors as follows*

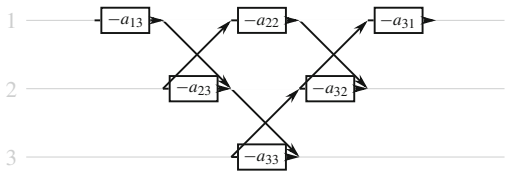
$$\mathscr{F}_0(A) = \Gamma_k\Gamma_{k-1}\cdots\Gamma_1,$$
$$\Gamma_j = F_0(a_{k-j+1,j})F_1(a_{k-j+2,j})\cdots F_{j-1}(a_{k,j}).$$

Moreover, each Γ_j is a scalar column companion matrix (padded with identities).

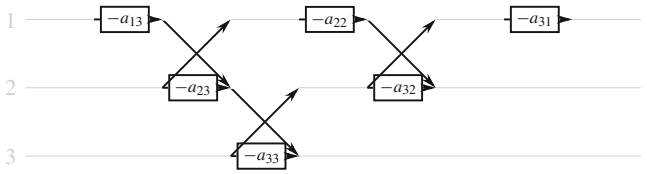
The proof is analogous to the one of Theorem 1. Again, we can consider the flow graph associated with $\mathscr{F}_0$.

Example 4 Consider again $k = 3$. Then, the flow graph associated with $\mathscr{F}_0$ is

[4] Here, by condition number we mean the non-homogeneous absolute or relative condition number defined in [34] (see also [1] where several definitions are compared). It is easy to verify that substituting the change of basis in the formula for the non-homogeneous condition number in [34] does not change the result.



Separating the elementary factors into three descending diagonals, we get the decompositions into three column companion matrices.



The explicit matrices are

$$\mathscr{F}_0 = \begin{bmatrix} 0 & 0 & -a_{13} \\ 0 & -a_{22} & -a_{23} \\ -a_{31} & -a_{32} & -a_{33} \end{bmatrix} = \Gamma_3\Gamma_2\Gamma_1$$

$$= \left[\begin{array}{cc|c} 0 & 0 & -a_{13} \\ 1 & 0 & -a_{23} \\ \hline 0 & 1 & -a_{33} \end{array}\right] \left[\begin{array}{cc|c} 0 & -a_{22} & 0 \\ 1 & -a_{32} & 0 \\ \hline 0 & 0 & 1 \end{array}\right] \left[\begin{array}{c|cc} -a_{31} & 0 & 0 \\ \hline 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right].$$

We can adapt this decomposition to work with lower triangular matrices, but the result is more complicated.

Theorem 3 *Let $A \in \mathbb{C}^{k\times k}$ be a lower triangular matrix. Then, $\mathscr{F}_0(A)$ can be factored as the product of k^2 scalar elementary Fiedler factors as follows*

$$\mathscr{F}_0(A) = \Gamma_1\Gamma_2\cdots\Gamma_k, \tag{5}$$

$$\Gamma_j = F_0(a_{j,j})F_1(a_{j+1,j})\cdots F_{k-j}(a_{k,j})F_{k-j+1}(0)F_{k-j+2}(0)\cdots F_{k-1}(0).$$

Moreover, each Γ_j is a scalar column companion matrix (padded with identities).

Again, we can prove this factorization either algebraically or by following the edges along the associated Fiedler graph, which is shown in Fig. 10.

The additional blocks with zeros are in fact permutations necessary for positioning each element correctly. Even though this factorization is still operation-free, meaning that there are no arithmetic operations involving the a_{ij}, we see that this is only because the trailing elementary Fiedler factors have 0. Indeed, if one replaces the zeros appearing in Fig. 10 with different quantities, the resulting product

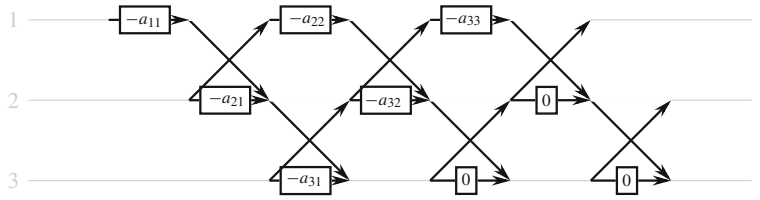


Fig. 10 Fiedler graph associated to $\mathscr{F}_0(A)$ where A is a lower triangular 3×3 matrix

requires arithmetic operations. This is an instance of a more general result, linked to *operation-free* linearizations, as in Definition 2.

Theorem 4 ([31, 36]) *Consider the product $\mathscr{P} = M_1 M_2 \cdots M_\ell$, where for each $k = 1, 2, \ldots, \ell$ the factor M_k is an elementary Fiedler factor $F_{i_k}(a_{j_k})$. Then, $\mathscr{P}$ is operation-free for each choice of the scalars (or matrices) a_{j_k} if and only if between every pair of factors $M_{i_k}, M_{i_{k'}}$ with the same index $i_k = i_{k'} = i$ there is a factor with index $i + 1$. In terms of diagrams, this means that between every two factors at height i there must appear a factor at height $i + 1$.*

Again, this theorem holds for the block version as well. While all the other products of Fiedler factors that we consider in this paper are operation-free a priori because of this theorem, the one in (5) does not satisfy this criterion. It is only operation-free because of the zeros.

Remark 5 It is impossible to find an operation-free factorization of $\mathscr{F}_0(A)$ for an unstructured A. Indeed, if there existed a factorization $\mathscr{F}_0(A) = M_1 M_2 \cdots M_{k^2}$, where each M_i is a scalar elementary Fiedler factor, then by writing $\mathscr{F}_0(A)^{-1} = M_{k^2}^{-1} \cdots M_2^{-1} M_1^{-1}$ one could solve any linear system $Ax = b$ in $O(k^2)$ flops, which is known to be impossible [32].

4 Factoring Block Companion Matrices

In this section, we use the previous results to show that any block Fiedler companion matrix can be factored as $\mathscr{C} = C_1 C_2 \cdots C_k$, where each C_j is a scalar Fiedler companion. This generalizes the results for block column companion matrices from Aurentz et al. [7].

These factorizations have a nice property: they are low-rank perturbations of unitary matrices. This allows the design of fast algorithms for computing the eigenvalues of $\mathscr{C}$, by working on the factored form [7].

This novel factorization allows to build even more linearizations. When all factors C_j are invertible, all the cyclic permutations of the factors provide again linearizations for the same matrix polynomial, since they are all similar.

For column block companion matrices, we have the following (see also Aurentz, et al. [7]).

Theorem 5 *Let $P(z) \in \mathbb{C}[z]^{k\times k}$ be a monic matrix polynomial of degree d with lower anti-triangular constant term A_0. Then, the associated block column companion matrix can be factored as a product of k scalar companion matrices of size $dk \times dk$.*

A formal proof will follow as a special case of Theorem 6. Here, we only point out that this factorization is again easy to detect using the graph representation.

Example 5 Let $d = k = 3$ and $P(z) = Iz^3 + A_2z^2 + A_1z + A_0$, with A_0 lower anti-triangular. Then, using the scalar-level factorizations of each Fiedler block, the column companion matrix of $P(z)$ links to the flow graph in Fig. 11.

It is easy to decompose this graph as the product of the three factors drawn in the figure in different colours. Moreover, each of these three factors is a column

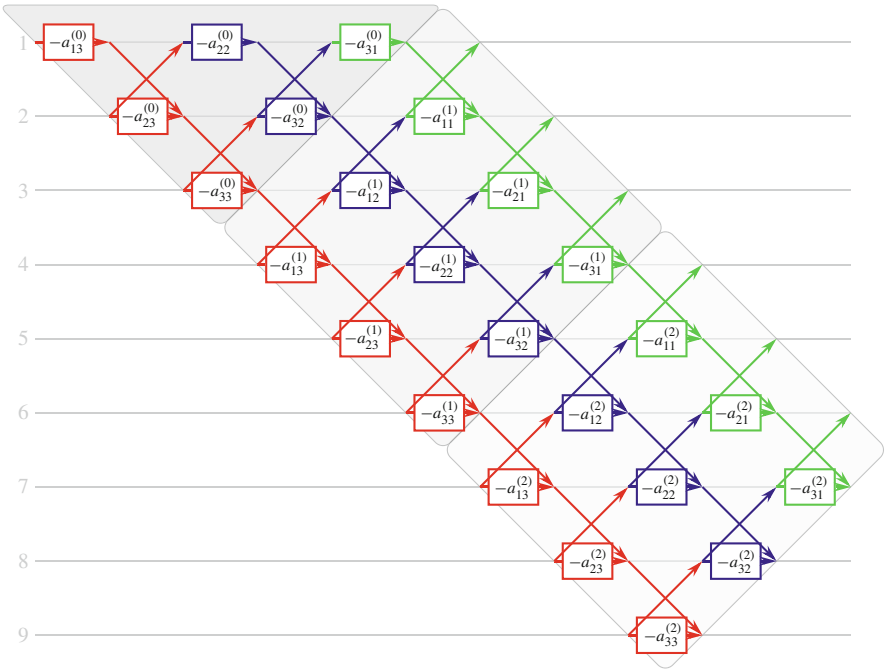


Fig. 11 Graph of the block column companion matrix associated to the monic matrix polynomial $P(z)$. The constant coefficient A is lower anti-triangular. To simplify the notation, we used $a_{ij}^{(i)}$ to denote the entries of the matrices A_i

companion matrix[5] constructed from a polynomial whose coefficients are a column of $[A_0^T \;\; A_1^T \;\; A_2^T]^T$.

This construction can be generalized to a generic block Fiedler companion. To prove the theorem formally, we need some additional definitions [22, 31] and a lemma which is a variation of [19, Proposition 2.12].

Definition 3 Let $P = F_{i_1} F_{i_2} \cdots F_{i_\ell}$ be a product of ℓ Fiedler elementary factors. For each $i = 0, \ldots, d-1$, the *layer* $\mathscr{L}_{i:i+1}(P)$ is the sequence formed by the factors of the form $F_i(a)$ and $F_{i+1}(b)$, for any values of a, b, taken in the order in which they appear in P.

Definition 4 Let $C = F_{\sigma(0)} F_{\sigma(1)} \cdots F_{\sigma(d-1)}$ be a Fiedler companion matrix, where σ is a permutation of $\{0, 1, \ldots, d-1\}$. We say that C has

- A *consecution* at i, $0 \le i \le d-2$, if $\mathscr{L}_{i:i+1}(C) = (F_i, F_{i+1})$;
- An *inversion* at i, $0 \le i \le d-2$, if $\mathscr{L}_{i:i+1}(C) = (F_{i+1}, F_i)$.

For instance, the Fiedler companion matrix whose associated graph is depicted in Fig. 4 has two consecutions at 0 and 1, and an inversion at 2. Note that in the flow graph a consecution corresponds to the subgraph of F_i being to the left of the subgraph of F_{i+1}, and *vice versa* for an inversion. The definition extends readily to the block case.

The layers of a factorization in elementary Fiedler factors uniquely define the resulting product as stated in the next lemma.

Lemma 1 *Let F and G be two products of (scalar or block) elementary Fiedler factors of size $d \times d$. If $\mathscr{L}_{i:i+1}(F) = \mathscr{L}_{i:i+1}(G)$ for all $i = 0, \ldots, d-2$, then the two products can be reordered one into the other by only swapping commuting factors, and hence $F = G$ (as matrices).*

Proof See [19, Proposition 2.12]. □

Theorem 6 *Let $\mathscr{C} = \mathscr{F}_{\sigma(0)} \cdots \mathscr{F}_{\sigma(d-1)}$ be a block Fiedler companion matrix of the monic matrix polynomial $P(z) = Iz^d + A_{d-1}z^{d-1} + \cdots + A_1 z + A_0 \in \mathbb{C}[z]^{k\times k}$, with the matrix A_0 lower anti-triangular. Then, $\mathscr{C} = C_1 C_2 \cdots C_k$, where each of the matrices C_j is a scalar Fiedler companion matrix.*

Proof In the following, we use the notation $a_{ij}^{(i)}$ to denote the (i, j) entry of A_k.

For all $i = 0, 1, \ldots, d-2$ and $j = 1, 2, \ldots, k$, we use j' as a shorthand for $k - j + 1$; let the matrix $M_{i,j}$ be defined as

$$M_{i,j} = F_{ki}(a_{j,j'}^{(i)}) F_{ki+1}(a_{j+1,j'}^{(i)}) \cdots F_{ki+j'-1}(a_{k,j'}^{(i)}) \\ F_{ki+j'}(a_{1,j'}^{(i+1)}) F_{ki+j'+1}(a_{2,j'}^{(i+1)}) \cdots F_{ki+k-1}(a_{j-1,j'}^{(i+1)}) \qquad (6)$$

[5] Similarly, one could factor it into dk row companion matrices linked to polynomials of degree 3.

if $\mathscr{C}$ has a (block) consecution at i, or:

$$M_{i,j} = F_{ki+k-1}(a^{(i+1)}_{j,j'-1})F_{ki+k-2}(a^{(i+1)}_{j,j'-2}) \cdots F_{ki+j}(a^{(i+1)}_{j,1}) \\ F_{ki+j-1}(a^{(i)}_{j,k})F_{ki+j-2}(a^{(i)}_{j,k-1}) \cdots F_{ki}(a^{(i)}_{j,j'}) \tag{7}$$

if $\mathscr{C}$ has a (block) inversion at i.

When $i = d-1$, we can take either of (6) or (7) as $M_{i,j}$, omitting all terms containing entries of $a^{(d)}$. (Hence, in particular, one can find two different factorizations for $\mathscr{C}$.)

We will prove that $\mathscr{C} = C_1 C_2 \cdots C_k$, where

$$C_j = (M_{\sigma(0),j} M_{\sigma(1),j} \cdots M_{\sigma(d-1),j}). \tag{8}$$

Each of the C_j contains exactly one factor of the form $F_0, F_1, \ldots, F_{dk-j}$, hence it is a scalar Fiedler companion matrix, linked to a polynomial whose coefficients are elements out of the original block coefficients.

To show that $\mathscr{C} = C_1 C_2 \cdots C_k$, we rely on Lemma 1. Note that $\mathscr{C} = \mathscr{F}_{\sigma(0)} \mathscr{F}_{\sigma(1)} \cdots \mathscr{F}_{\sigma(d-1)}$ can be seen as a product of scalar elementary Fiedler factors of size $kd \times kd$ using the factorizations in Theorems 1 and 2. Relying on Lemma 1, we simply have to verify that its layers coincide with those of $C_1 C_2 \cdots C_k$. Indeed, let $0 \le i < d$, and $1 \le \ell \le k$; if $\mathscr{C}$ has a block consecution at i, then

$$\mathscr{L}_{ki+\ell-1:ki+\ell}(\mathscr{C}) = \mathscr{L}_{ki+\ell-1:ki+\ell}(\mathscr{F}_i \mathscr{F}_{i+1}) = \\ \Big(F_{ki+\ell-1}(a^{(i)}_{\ell,k}), F_{ki+\ell}(a^{(i)}_{\ell+1,k}), F_{ki+\ell-1}(a^{(i)}_{\ell+1,k-1}), F_{ki+\ell}(a^{(i)}_{\ell+2,k-1}), \ldots, F_{ki+\ell-1}(a^{(i)}_{k,\ell}), \\ F_{ki+\ell}(a^{(i+1)}_{1,\ell}), F_{ki+\ell-1}(a^{(i+1)}_{1,\ell-1}), F_{ki+\ell}(a^{(i+1)}_{2,\ell-1}), F_{ki+\ell-1}(a^{(i+1)}_{2,\ell-2}), \ldots, F_{ki+\ell}(a^{(i+1)}_{\ell,1}) \Big) \\ = \mathscr{L}_{ki+\ell-1:ki+\ell}(M_{i,1} M_{i+1,1} M_{i,2} M_{i+1,2} \cdots M_{i,k} M_{i+1,k}) = \mathscr{L}_{ki+\ell-1:ki+\ell}(C_1 C_2 \cdots C_k).$$

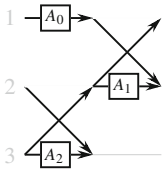
Similarly, if $\mathscr{C}$ has an inversion in i, then:

$$\mathscr{L}_{ki+\ell-1:ki+\ell}(\mathscr{C}) = \mathscr{L}_{ki+\ell-1:ki+\ell}(\mathscr{F}_{i+1} \mathscr{F}_i) = \\ \Big(F_{ki+\ell}(a^{(i+1)}_{1,\ell}), F_{ki+\ell-1}(a^{(i+1)}_{1,\ell-1}), F_{ki+\ell}(a^{(i+1)}_{2,\ell-1}), F_{ki+\ell-1}(a^{(i+1)}_{2,\ell-2}), \ldots, F_{ki+\ell}(a^{(i+1)}_{\ell,1}), \\ F_{ki+\ell-1}(a^{(i)}_{\ell,k}), F_{ki+\ell}(a^{(i)}_{\ell+1,k}), F_{ki+\ell-1}(a^{(i)}_{\ell+1,k-1}), F_{ki+\ell}(a^{(i)}_{\ell+2,k-1}), \ldots, F_{ki+\ell-1}(a^{(i)}_{k,\ell}) \Big) \\ = \mathscr{L}_{ki+\ell-1:ki+\ell}(M_{i+1,1} M_{i,1} M_{i+1,2} M_{i,2} \cdots M_{i+1,k} M_{i,k}) = \mathscr{L}_{ki+\ell-1:ki+\ell}(C_1 C_2 \cdots C_k).$$

□

This tedious algebraic proof hides a simple structure that is revealed by the associated graphs: the scalar elementary Fiedler factors appearing in the graph of $\mathscr{C}$ can be split into k twisted lines that run diagonally, parallel one to the other. Moreover, it is interesting to remark that all resulting Fiedler companion matrices have the same structure, with consecutions and inversions in the same positions. This is illustrated clearly in the next example.

Example 6 Consider the block Fiedler companion matrix of the matrix polynomial $P(z) = Iz^3 + A_2z^2 + A_1z + A_0$, with $d = k = 3$, defined as $\mathscr{C} = \mathscr{F}_2\mathscr{F}_0\mathscr{F}_1$. Its block-level graph is



and has a consecution at block level 0 and an inversion at block level 1. Its scalar-level diagram is presented in Fig. 12. The elements belonging to the three factors C_1, C_2, C_3 are drawn in three different colours. Formally, we have

$$M_{0,1} = F_0(a_{13}^{(0)})F_1(a_{23}^{(0)})F_2(a_{33}^{(0)}), \qquad M_{1,1} = F_5(a_{12}^{(2)})F_4(a_{11}^{(2)})F_3(a_{13}^{(1)}),$$
$$M_{0,2} = F_0(a_{22}^{(0)})F_1(a_{32}^{(0)})F_2(a_{12}^{(1)}), \qquad M_{1,2} = F_5(a_{21}^{(2)})F_4(a_{23}^{(1)})F_3(a_{22}^{(1)}),$$
$$M_{0,3} = F_0(a_{31}^{(0)})F_1(a_{11}^{(1)})F_2(a_{21}^{(1)}), \qquad M_{1,3} = F_5(a_{33}^{(1)})F_4(a_{32}^{(1)})F_3(a_{31}^{(1)})$$

and finally, using (7) we have

$$M_{2,1} = F_6(a_{13}^{(2)}), \quad M_{2,2} = F_7(a_{23}^{(2)})F_6(a_{22}^{(2)}), \quad M_{2,3} = F_8(a_{33}^{(2)})F_7(a_{32}^{(2)})F_6(a_{31}^{(2)}).$$

In accordance with (8), we get $C_1 = M_{2,1}M_{0,1}M_{1,1}$, $C_2 = M_{2,2}M_{0,2}M_{1,2}$, and $C_3 = M_{2,3}M_{0,3}M_{1,3}$.

Note that the factorization is not unique since we can additionally incorporate the terms $F_7(a_{23}^{(2)})F_8(a_{33}^{(2)})$ in C_1, thereby defining the matrix $M_{2,1}$ as in (6) rather than (7). In that case also $M_{2,2}$ and $M_{2,3}$ should be defined in accordance with (6).

The corresponding matrices are

$$\mathscr{C} = \begin{bmatrix} 0 & A_0 & 0 \\ 0 & 0 & I \\ I & A_1 & A_2 \end{bmatrix} = C_1C_2C_3,$$

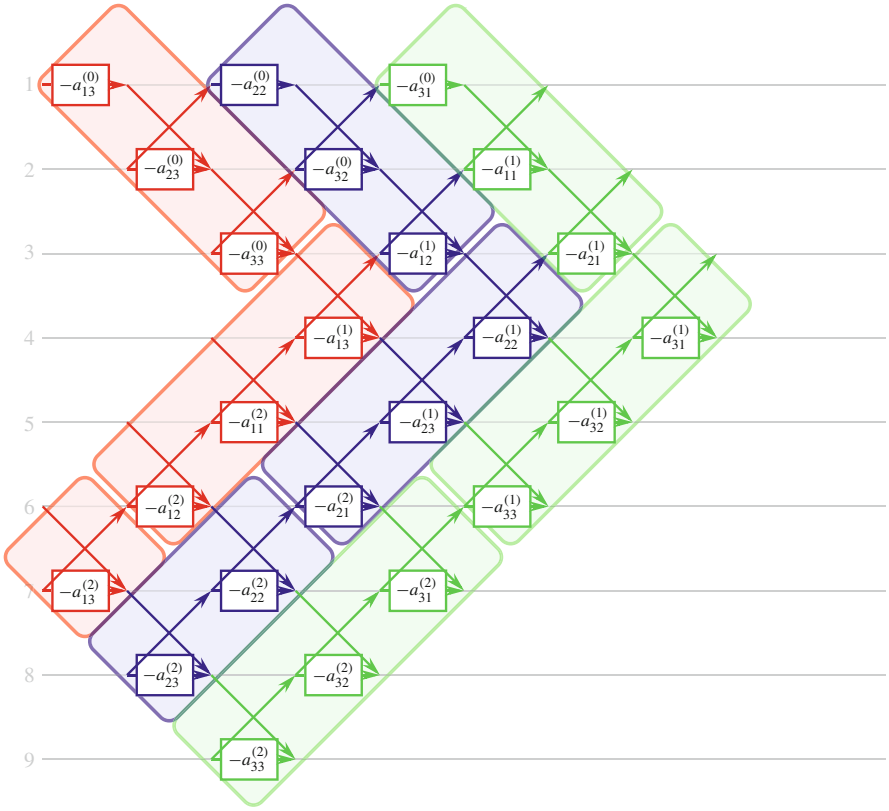


Fig. 12 The scalar-level diagram associated to Example 6. Each diagonal segment in a box represents a term $M_{i,j}$. We use the notation $a_{ij}^{(k)}$ to denote the entry in position (i, j) of A_k

with:

$$C_1 = \begin{bmatrix} & -a_{13}^{(0)} & & & & & \\ 1 & -a_{23}^{(0)} & & & & & \\ & 1 & -a_{33}^{(0)} & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & 1 & -a_{13}^{(1)} & -a_{11}^{(2)} & -a_{12}^{(2)} & -a_{13}^{(2)} & \\ & & & & & & 1 \\ & & & & & & & 1 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} & -a_{22}^{(0)} & & & & & & \\ 1 & -a_{32}^{(0)} & & & & & & \\ & 1 & -a_{12}^{(1)} & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & 1 & -a_{22}^{(1)} & -a_{23}^{(1)} & -a_{21}^{(2)} & -a_{22}^{(2)} & -a_{23}^{(2)} & \\ & & & & & & & 1 \end{bmatrix},$$

$$C_3 = \begin{bmatrix} & -a_{31}^{(0)} & & & & & & \\ 1 & -a_{11}^{(1)} & & & & & & \\ & 1 & -a_{21}^{(1)} & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \\ & 1 & -a_{31}^{(1)} & -a_{32}^{(1)} & -a_{33}^{(1)} & -a_{31}^{(2)} & -a_{32}^{(2)} & -a_{33}^{(2)} \end{bmatrix}.$$

5 Unitary-Plus-Low-Rank Structure

Unitary- (or orthogonal) plus-low-rank matrices appear frequently in fast algorithms for polynomial root-finding [3–5, 7–9, 12–15]. To the best of our knowledge, these techniques have been applied to row and column companion matrices (either block or scalar ones), but never to general Fiedler linearizations, even though recent advances [10] point towards novel algorithms for eigenvalue computations of unitary-plus-low-rank matrices.

In this section, we show that (block) Fiedler companion matrices are unitary-plus-low-rank, and that an upper bound on the rank of the correction is easily determined from their structure. This result is not new, as it is already present in a very similar form in [24, Section 6]; however, we report an alternative proof making use of the Fiedler graphs.

Lemma 2 *Let A_1 be unitary plus rank t_1, and A_2 be unitary plus rank t_2. Then, $A_1 A_2$ is unitary plus rank (at most) $t_1 + t_2$.*

Proof It is sufficient to write $A_i = Q_i + u_i v_i^T$, with Q_i unitary and $u_i, v_i \in \mathbb{C}^{n \times t_i}$, for $i = 1, 2$. Then

$$A_1 A_2 = (Q_1 + u_1 v_1^T) A_2 = Q_1 A_2 + u_1 v_1^T A_2 = Q_1 Q_2 + Q_1 u_2 v_2^T + u_1 v_1^T A_2. \quad \square$$

We introduce the concept of *segment decomposition* of a Fiedler companion matrix, which groups together elementary Fiedler factors with consecutive indices.

Definition 5 Let $C = F_{\sigma(0)} F_{\sigma(1)} \cdots F_{\sigma(d-1)}$ be a scalar Fiedler companion matrix. We say that C has t *segments* (or, equivalently, that its graph has t segments) if t is the minimal positive integer such that $C = \Gamma_1 \cdots \Gamma_t$, for a certain set of indices i_j satisfying

$$\Gamma_j = F_{\sigma(i_j)} \cdots F_{\sigma(i_{j+1}-1)}, \qquad 0 = i_1 < i_2 < \cdots < i_{t+1} = d,$$

and such that the integers $\sigma(i_j), \ldots, \sigma(i_{j+1} - 1)$ are consecutive (either in increasing or decreasing order).

Note that each Γ_j is either a column or row companion matrix possibly padded with identities. Segments are easily identified in the Fiedler graph, as they correspond to sequences of diagonally aligned elementary graphs. For instance, Fig. 13 depicts, on the left, the graph of the Fiedler companion matrix of $C = F_0 F_1 F_5 F_4 F_2 F_6 F_7 F_3$. Swapping commuting blocks we can rearrange the elementary Fiedler factors as follows $C = (F_0 F_1 F_2)(F_5 F_4 F_3)(F_6 F_7)$ identifying the three column and row companion matrices and hence the three segments. Similarly, the graph on the right has two segments.

Remark 6 The paper [22] defines a sequence of integers called the *consecution-inversion structure sequence* (CISS) of a Fiedler companion. The number of segments can be deduced from the CISS: namely, it is the length of CISS (excluding

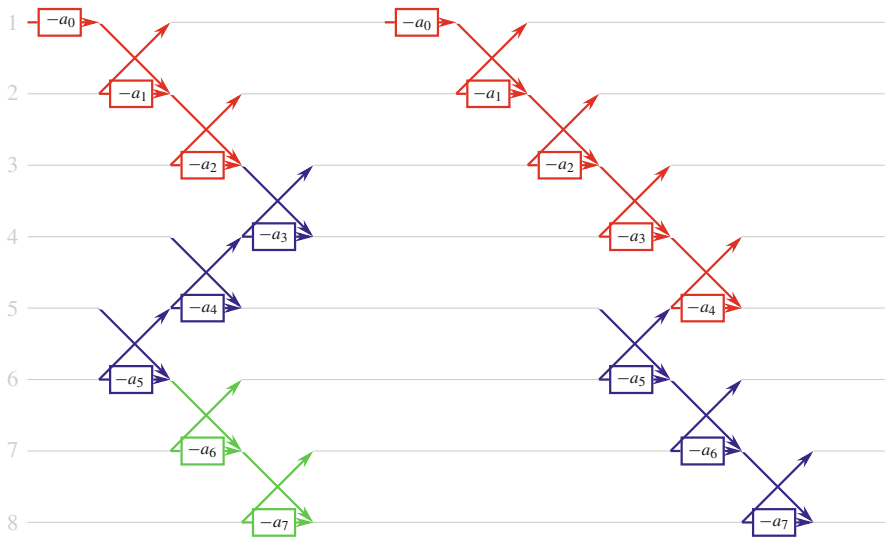


Fig. 13 Two graphs associated to scalar Fiedler companion matrices. The example on the left is composed of three segments, while the one on the right of only two

a leading or trailing zero if there is one) minus the number of distinct pairs of consecutive 1's appearing in it.

Theorem 7 *Let C be a scalar Fiedler companion matrix with t segments. Then, C is unitary plus rank (at most) t.*

Proof If C has t segments, then by definition $C = \Gamma_1\Gamma_2\cdots\Gamma_t$, where each Γ_j is either a column or a row companion matrix (possibly padded with identities). In fact, if $\Gamma_j = F_{\sigma(i_j)}\cdots F_{\sigma(i_{j+1}-1)}$ and the integers $\sigma(i_j),\ldots,\sigma(i_{j+1}-1)$ are consecutive in increasing order, we obtain a column companion matrix; if instead they are consecutive in decreasing order, we obtain a row companion matrix. Each row or column companion matrix is unitary plus rank 1 (since it is sufficient to alter the last row or column to turn it into the unitary cyclic shift matrix $Z + e_1e_n^T$). Hence, C is the product of t unitary-plus-rank-1 matrices, which is unitary plus rank (at most) t by Lemma 2. □

The above result can be used to prove another interesting fact.

Theorem 8 *Let $\mathscr{C}$ be a block Fiedler companion matrix with a block-level graph composed of t segments. Then, $\mathscr{C}$ is unitary plus rank (at most) kt.*

Proof The result follows by generalizing the proof of Theorem 7 to block Fiedler companion matrices, noticing that each block Fiedler companion matrix is unitary plus rank k. □

Remark 7 Given a Fiedler companion matrix $\mathscr{C}$ with t segments and its factorization $\mathscr{C} = C_1C_2\cdots C_k$ obtained through Theorem 6, each C_j has the same number of segments, but it may happen that this number is larger than t. An example is given by the block version of the pencil on the right of Fig. 13, i.e. $\mathscr{C} = \mathscr{F}_5\mathscr{F}_6\mathscr{F}_7\mathscr{F}_0\mathscr{F}_1\mathscr{F}_2\mathscr{F}_3\mathscr{F}_4$. Indeed, each of its scalar Fiedler companion factors C_j has three segments rather than two.

Remark 8 Theorem 8 shows that we can apply to $\mathscr{C}$ structured methods for fast eigenvalues computation, provided that the number of segments in the graph associated with the Fiedler companion matrix is limited.

In addition, it gives an explicit factorization of $\mathscr{C}$ into unitary-plus-rank-1 matrices, therefore providing all the tools required to develop a fast method similar to the one presented in [7] for column block companion matrices.

6 A Thin-Banded Linearization

Another interesting use of scalar-level factorizations of block Fiedler companion matrices is to construct new companion matrices by rearranging factors. We present an example using the flow graphs.

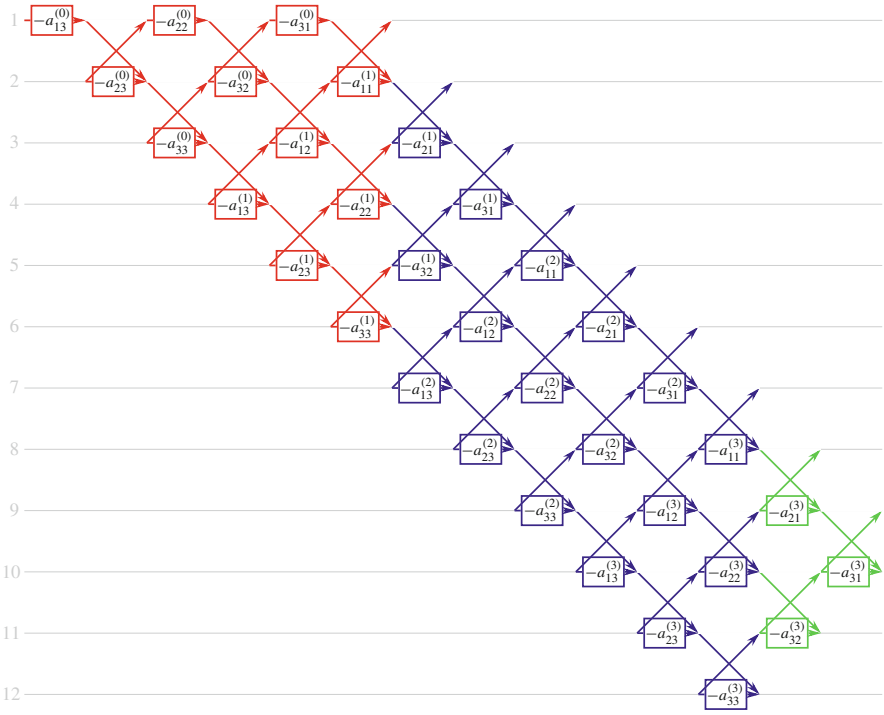


Fig. 14 The flow graph of the column companion matrix in Example 7

Example 7 We consider the matrix polynomial $P(z) = Iz^4 + A_3z^3 + A_2z^2 + A_1z + A_0$, with $d = 4$, $k = 3$. Assume for simplicity that A is already anti-triangular. The graph associated to its column companion matrix Γ is shown in Fig. 14.

We can factor this matrix as the product of three factors $\Gamma = RST$, which we have drawn in different colours in Fig. 14. Note that R and T commute, and that S, T are invertible, being products of non-singular Fiedler factors. Hence, RST is similar to $TRS = RTS$, which is in turn similar to TSR. This proves that $\mathscr{C} = TRS$ is also a companion matrix for $P(z)$. The graph of $\mathscr{C}$ is depicted in Fig. 15.

Note that $\mathscr{C}$ is not a Fiedler companion matrix, as it cannot be obtained by permuting block-level factors; "breaking the blocks" is required to construct it. This construction can be generalized to arbitrary d and k, and it has a couple of nice features.

- $\mathscr{C}$ is a banded matrix. Since we have drawn its diagram inside six columns in Fig. 15, there is no path from the left to the right of the diagram that moves up or down more than five times; this means that $\mathscr{C}_{i,j} = 0$ whenever $|j - i| \geq 6$. Generalizing this construction to arbitrary k and d, one gets $\mathscr{C}_{i,j} = 0$ whenever $|j - i| \geq 2k$. Finding low-bandwidth linearizations and companion matrices has attracted quite some interest in the past: for instance, [2, 27] present a

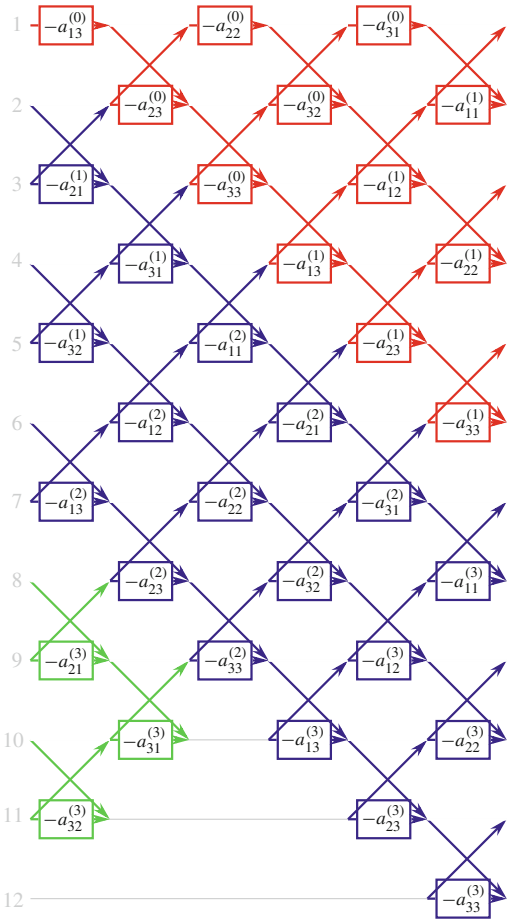


Fig. 15 The graph of $\mathscr{C}$ in Example 7

(block) pentadiagonal companion matrix (which can also be expressed as a block tridiagonal linearizing pencil). The new companion matrix $\mathscr{C}$ has the same bandwidth as this classical example.

- Whenever the coefficients of $P(z)$ are symmetric matrices, we can factor $\mathscr{C}$ into the product of two symmetric matrices $\mathscr{C} = \mathscr{C}_1\mathscr{C}_2$: it is sufficient to take $\mathscr{C}_1$ as the product of all factors appearing in the first five columns of Fig. 15, and $\mathscr{C}_2$ as the product of all factors appearing in the sixth and last one, i.e. $\mathscr{C}_2 = F_1(a_{11}^{(1)})F_3(a_{22}^{(1)})F_5(a_{33}^{(1)})F_7(a_{11}^{(3)})F_9(a_{22}^{(3)})F_{11}(a_{33}^{(3)})$. This means that we can construct a symmetric pencil $\mathscr{C}_1 - \mathscr{C}_2^{-1}z$ which is a linearization of $P(z)$. (Note that $\mathscr{C}_2^{-1}$ is operation-free.) Finding symmetric linearizations for symmetric matrix polynomials is another problem that has attracted research interest in the past [19, 31].

Remark 9 We remark that it is not clear that thin-banded linearizations provide practical advantages in numerical computation. Commonly used eigenvalue algorithms (namely, QZ and QR) cannot exploit this structure, unless the matrix at hand is also symmetric (or the pencil is symmetric/positive definite).

7 Conclusions

We have presented an extension of the graph-based approach by Poloni and Del Corso [31] that allows to produce scalar-level factorizations of block Fiedler companion matrices.

We have shown that this framework can be used for several purposes, such as identifying new factorizations of products of Fiedler matrices, revealing their structures (such as the unitary-plus-rank-t structure), and combining them to build new linearizations.

Once the reader is familiar with reading the Fiedler graphs, there are many more factorizations that could be devised. Every time a diagonal line of factors appears in a graph, it can be transformed into a factorization that involves row or column companion matrices.

The presented approach allows a more general and particularly easy manipulation of these linearizations. It might lead to the development of efficient algorithms for the computation of eigenvalues of matrix polynomials using the product form with the unitary-plus-rank-1 structure.

Acknowledgements The research of the first three authors was partially supported by GNCS projects "Metodi numerici avanzati per equazioni e funzioni di matrici con struttura" and "Tecniche innovative per problemi di algebra lineare"; the research of the first two was also supported by University of Pisa under the grant PRA-2017-05. The research of the fourth author was supported by the Research Council KU Leuven, project C14/16/056 (Inverse-free Rational Krylov Methods: Theory and Applications).

References

1. Anguas, L.M., Bueno, M.I., Dopico, F.M.: A comparison of eigenvalue condition numbers for matrix polynomials, arXiv preprint arXiv:1804.09825 (2018)
2. Antoniou, E.N., Vologiannidis, S.: A new family of companion forms of polynomial matrices. Electron. J. Linear Algebra **11**, 78–87 (2004)
3. Aurentz, J.L., Vandebril, R., Watkins, D.S.: Fast computation of the zeros of a polynomial via factorization of the companion matrix. SIAM J. Sci. Comput. **35**, A255–A269 (2013)
4. Aurentz, J.L., Mach, T., Vandebril, R., Watkins, D.S.: Fast and backward stable computation of roots of polynomials. SIAM J. Matrix Anal. Appl. **36**, 942–973 (2015)
5. Aurentz, J.L., Mach, T., Vandebril, R., Watkins, D.S.: Fast and stable unitary QR algorithm. Electron. Trans. Numer. Anal. **44**, 327–341 (2015)

6. Aurentz, J.L., Mach, T., Robol, L., Vandebril, R., Watkins, D.S.: Core-Chasing Algorithms for the Eigenvalue Problem, vol. 13 of Fundamentals of Algorithms. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2018)
7. Aurentz, J.L., Mach, T., Robol, L., Vandebril, R., Watkins, D.S.: Fast and backward stable computation of the eigenvalues of matrix polynomials. Math. Comput. **88**, 313–247 (2019)
8. Bevilacqua, R., Del Corso, G.M., Gemignani, L.: A CMV-based eigensolver for companion matrices. SIAM J. Matrix Anal. Appl. **36**, 1046–1068 (2015)
9. Bevilacqua, R., Del Corso, G.M., Gemignani, L.: Compression of unitary rank-structured matrices to CMV-like shape with an application to polynomial rootfinding. J. Comput. Appl. Math. **278**, 326–335 (2015)
10. Bevilacqua, R., Del Corso, G.M., Gemignani, L.: Fast QR iterations for unitary plus low-rank matrices, arXiv preprint arXiv:1810.0270 (2018)
11. Bini, D.A., Robol, L.: On a class of matrix pencils and ℓ-ifications equivalent to a given matrix polynomial. Linear Algebra Appl. **502**, 275–298 (2016)
12. Bini, D.A., Eidelman, Y., Gemignani, L., Gohberg, I.: Fast QR eigenvalue algorithms for Hessenberg matrices which are rank-one perturbations of unitary matrices. SIAM J. Matrix Anal. Appl. **29**, 566–585 (2007)
13. Bini, D.A., Boito, P., Eidelman, Y., Gemignani, L., Gohberg, I.: A fast implicit QR eigenvalue algorithm for companion matrices. Linear Algebra Appl. **432**, 2006–2031 (2010)
14. Boito, P., Eidelman, Y., Gemignani, L.: Implicit QR for rank-structured matrix pencils. BIT Numer. Math. **54**, 85–111 (2014)
15. Boito, P., Eidelman, Y., Gemignani, L.: A real QZ algorithm for structured companion pencils, arXiv preprint arXiv:1608.05395 (2016)
16. Bueno, M.I., De Terán, F.: Eigenvectors and minimal bases for some families of Fiedler-like linearizations. Linear Multilinear Algebra **62**, 39–62 (2014)
17. Bueno, M.I., Furtado, S.: Palindromic linearizations of a matrix polynomial of odd degree obtained from Fiedler pencils with repetition. Electron. J. Linear Algebra **23**, 562–577 (2012)
18. Bueno, M.I., de Terán, F., Dopico, F.M.: Recovery of eigenvectors and minimal bases of matrix polynomials from generalized Fiedler linearizations. SIAM J. Matrix Anal. Appl. **32**, 463–483 (2011)
19. Bueno, M.I., Curlett, K., Furtado, S.: Structured strong linearizations from Fiedler pencils with repetition I. Linear Algebra Appl. **460**, 51–80 (2014)
20. Bueno, M.I., Dopico, F.M., Furtado, S., Rychnovsky, M.: Large vector spaces of block-symmetric strong linearizations of matrix polynomials. Linear Algebra Appl. **477**, 165–210 (2015)
21. Bueno, M.I., Dopico, F.M., Furtado, S., Medina, L.: A block-symmetric linearization of odd-degree matrix polynomials with optimal eigenvalue condition number and backward error. Calcolo **55**(3), 32 (2018)
22. De Terán, F., Dopico, F.M., Mackey, D.S.: Fiedler companion linearizations and the recovery of minimal indices. SIAM J. Matrix Anal. Appl. **31**, 2181–2204 (2010)
23. De Terán, F., Dopico, F.M., Mackey, D.S.: Palindromic companion forms for matrix polynomials of odd degree. J. Comput. Math. **236**, 1464–1480 (2011)
24. De Terán, F., Dopico, F.M., Pérez, J.: Condition numbers for inversion of Fiedler companion matrices. Linear Algebra Appl. **439**, 944–981 (2013)
25. De Terán, F., Dopico, F.M., Mackey, D.S.: Spectral equivalence of matrix polynomials and the Index Sum Theorem. Linear Algebra Appl. **459**, 264–333 (2014)
26. Dopico, F.M., Lawrence, P.W., Pérez, J., Van Dooren, P.: Block Kronecker linearizations of matrix polynomials and their backward errors, arXiv preprint arXiv:1707.04843 (2017)
27. Fiedler, M.: A note on companion matrices. Linear Algebra Appl. **372**, 325–331 (2003)
28. Hammarling, S., Munro, C.J., Tisseur, F.: An algorithm for the complete solution of quadratic eigenvalue problems. ACM Trans. Math. Softw. **39**, 18 (2013)
29. Higham, N.J., Mackey, D.S., Mackey, N., Tisseur, F.: Symmetric linearizations for matrix polynomials. SIAM J. Matrix Anal. Appl. **29**, 143–159 (2006)

30. Mackey, D.S., Mackey, N., Mehl, C., Mehrmann, V.: Vector spaces of linearizations for matrix polynomials. SIAM J. Matrix Anal. Appl. **28**, 971–1004 (2006)
31. Poloni, F., Del Corso, G.M.: Counting Fiedler pencils with repetitions. Linear Algebra Appl. **532**, 463–499 (2017)
32. Raz, R.: On the complexity of matrix product. In: Proceedings of the Thirty-fourth Annual ACM Symposium on Theory of Computing, STOC '02, New York, NY, USA, ACM, pp. 144–151 (2002)
33. Robol, L., Vandebril, R., Dooren, P.V.: A framework for structured linearizations of matrix polynomials in various bases. SIAM J. Matrix Anal. Appl. **38**, 188–216 (2017)
34. Tisseur, F.: Backward error and condition of polynomial eigenvalue problems. Linear Algebra Appl. **309**, 339–361 (2000)
35. Van Barel, M., Tisseur., F.: Polynomial eigenvalue solver based on tropically scaled Lagrange linearization. Linear Algebra Appl. **542**, 186–208 (2018)
36. Vologiannidis, S., Antoniou, E.N.: A permuted factors approach for the linearization of polynomial matrices. Math. Control Signals Syst. **22**, 317–342 (2011)

A Class of Quasi-Sparse Companion Pencils

Fernando De Terán and Carla Hernando

Abstract In this paper, we introduce a general class of quasi-sparse potential companion pencils for arbitrary square matrix polynomials over an arbitrary field, which extends the class introduced in [B. Eastman, I.-J. Kim, B. L. Shader, K.N. Vander Meulen, *Companion matrix patterns*. Linear Algebra Appl. 436 (2014) 255–272] for monic scalar polynomials. We provide a canonical form, up to permutation, for companion pencils in this class. We also relate these companion pencils with other relevant families of companion linearizations known so far. Finally, we determine the number of different sparse companion pencils in the class, up to permutation.

Keywords Companion matrix · Companion pencil · Linearization · Sparsity · Scalar polynomial · Matrix polynomial · arbitrary field · Permutation

1 Introduction

The standard way to compute the eigenvalues and eigenvectors of a matrix polynomial

$$Q(\lambda) = \sum_{i=0}^{k} \lambda^i A_i, \qquad A_i \in \mathbb{F}^{n\times n}, \qquad i = 0, 1, \ldots, k, \qquad A_k \neq 0, \tag{1}$$

(with $\mathbb{F}$ being an arbitrary field) is by means of a *linearization*, which is a matrix pencil (that is, a matrix polynomial of degree 1) whose eigenvalues (together with their multiplicities) coincide with the ones of the polynomial (1). Any matrix polynomial has infinitely many linearizations, but in order for them to be useful in practice, it is important to know in advance that they are linearizations. One way to

F. De Terán (✉) · C. Hernando
Departamento de Matemáticas, Universidad Carlos III de Madrid, Leganés, Spain
e-mail: fteran@math.uc3m.es; cahernan@math.uc3m.es

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_8

create such linearizations is by means of symbolic constructions consisting of block-partitioned pencils whose blocks contain the coefficients of (1). *Companion pencils* are particular cases of these constructions (see Definition 2.2). They present several advantages, besides being linearizations for any matrix polynomial. Among the most relevant ones are: (a) they are *strong* linearizations (that is, they also preserve the infinite eigenvalue of the polynomial, and its multiplicities), and (b) they present a template involving no arithmetic operations at all. The only information needed to build companion pencils is the selection and placement of the blocks.

Several families of companion pencils have been introduced in the literature, including the Fiedler-like families [1, 4, 8, 10, 23] and the block-Kronecker linearizations [15]. They contain, as particular cases, the classical Frobenius linearizations, and extend the notion of *companion matrix*, which has been extensively used to compute roots of scalar polynomials (like in the MATLAB command `roots`). In the recent years, some effort has been devoted to introduce new families of companion pencils which preserve some of the structures of matrix polynomials usually encountered in applications [3, 5, 6, 9], or to companion pencils in other polynomial bases than the monomial basis [20–22]. Some recent works have also analyzed particular features or applications of Fiedler-like pencils [2, 12–14]. In particular, it is proved in [7] that the families of Fiedler and generalized Fiedler pencils are particular cases of block-Kronecker linearizations. As for companion matrices of scalar polynomials, we refer to [18] and [19] for a more general notion than the one considered in this paper, and to [16] for some pentadiagonal constructions.

Companion matrices are valid only for monic scalar polynomials. They have been studied in several recent papers [17–19] from a theoretical point of view, providing canonical expressions up to permutation. Some interest has also been paid to *sparse* companion matrices, namely those with the smallest number of nonzero entries, motivated by the simplicity of the constructions. However, from the numerical point of view, it may be desirable to work with non-monic polynomials. This is one of the motivations to introduce the more general notion of companion pencils.

In this paper, we are mainly interested in sparse companion pencils. Our main goal is to extend the results in [17] to such kind of constructions. In particular, we first introduce a general class of pencils (denoted by $\mathscr{R}_{n,k}$) associated with symbolic matrix polynomials as in (1). As with all families of companion pencils mentioned above, the pencils in $\mathscr{R}_{n,k}$ contain $k-1$ identity blocks, plus another $k-1$ blocks equal to λI, together with some other nonzero blocks involving the coefficients of the polynomial. However, its generality relies on the fact that these blocks can be located anywhere in the pencil. This aims to introduce a class of potential linearizations that keeps all the essential structural properties of the previous families of linearizations (namely, the identity blocks), and having a small number of nonzero entries (or blocks). We refer to them as *quasi-sparse* because of this small number of nonzero entries. Some pencils in $\mathscr{R}_{n,k}$ can be either not companion (that is, linearizations) or not sparse, and our interest focuses on those which are companion (firstly) and those which are companion and sparse (secondly).

Our goal is to provide a canonical expression, up to permutation, for companion pencils in this family, resembling the one provided in [17] for companion matrices, and to determine, up to permutation as well, how many different sparse companion pencils are in this family. To achieve this goal, we introduce a new class of pencils, denoted by $\mathscr{QC}_{n,k}$ (plus an intermediate class $\mathscr{Q}_{n,k}$), which comprises, up to permutation, all companion pencils in $\mathscr{R}_{n,k}$, and we count the number of different sparse pencils in $\mathscr{QC}_{n,k}$.

The paper is organized as follows. In Sect. 2 we present the basic notions (including the families $\mathscr{R}_{n,k}$, $\mathscr{Q}_{n,k}$, and $\mathscr{QC}_{n,k}$), together with some structural properties. Section 3 is devoted to prove that any companion pencil in $\mathscr{R}_{n,k}$ is permutationally equivalent to a pencil in $\mathscr{Q}_{n,k}$, and that companion pencils in $\mathscr{Q}_{n,k}$ must belong to $\mathscr{QC}_{n,k}$. We also prove that all pencils in $\mathscr{QC}_{n,k}$ are companion pencils. In Sect. 4 we get the number of different sparse (companion) pencils in $\mathscr{QC}_{n,k}$.

2 Preliminaries

Throughout this paper, we use calligraphic letters with two subindices, like $\mathscr{A}_{n,k}$, to denote a class of $nk \times nk$ block-partitioned matrix pencils, which are viewed as block $k \times k$ matrices with blocks of size $n \times n$.

In order to define the notion of companion pencil for matrix polynomials, we first recall the following notions. For more information about them we refer to [11].

In what follows, the *reversal* of $Q(\lambda)$ in (1) is the polynomial $\mathrm{rev}Q(\lambda) := \sum_{j=0}^{k} \lambda^j A_{k-j}$, obtained by reversing the order of the coefficients of $Q(\lambda)$.

Definition 2.1 A matrix pencil $L(\lambda) = \lambda X + Y$ with $X, Y \in \mathbb{F}^{nk \times nk}$ is a linearization of an $n \times n$ matrix polynomial $Q(\lambda)$ of degree k if there exist two unimodular $nk \times nk$ matrix polynomials $U(\lambda)$ and $V(\lambda)$ such that

$$U(\lambda)L(\lambda)V(\lambda) = \begin{bmatrix} I_{(k-1)n} & 0 \\ 0 & Q(\lambda) \end{bmatrix},$$

(that is, $L(\lambda)$ is unimodularly equivalent to $\mathrm{diag}(I_{(k-1)n}, Q(\lambda))$. The linearization is called a strong linearization if $\mathrm{rev}L(\lambda)$ is also a linearization of $\mathrm{rev}Q(\lambda)$.

Definition 2.2 A companion pencil for general $n \times n$ matrix polynomials $\sum_{i=0}^{k} \lambda^i A_i$ of degree k is an $nk \times nk$ matrix pencil $L(\lambda) = \lambda X + Y$ such that if X and Y are viewed as block $k \times k$ matrices with blocks of size $n \times n$, then:

(i) each nonzero block of X and Y is either I_n or A_i (up to constants), for some $i = 0, \ldots, k$, and
(ii) $L(\lambda)$ is a strong linearization for every $n \times n$ matrix polynomial of degree k.

Note, in particular, that if $L(\lambda)$ is a companion pencil for $Q(\lambda)$, then $\det(L(\lambda)) = \alpha \det(Q(\lambda))$ (for some $\alpha \neq 0$). When $n = 1$, $Q(\lambda)$ is just a scalar polynomial. In

this case, we will use lowercase letters and this determinant condition reduces to $\det(L(\lambda)) = \alpha\, q(\lambda) = \alpha \sum_{i=0}^{k} \lambda^i a_i$, with $0 \neq \alpha \in \mathbb{F}$.

2.1 New Classes of Block-Partitioned Pencils

The most general family of "potential" companion pencils in this work, $\mathscr{R}_{n,k}$, is introduced in Definition 2.3. This family contains all the sparse companion pencils introduced so far in the literature (in the monomial basis). In particular both Fiedler and generalized Fiedler pencils [4, 8, 10], as well as the sparse block-Kronecker pencils introduced in [15, Def 5.1]. The motivation for introducing this family is precisely to create a general family containing all these companion pencils, and also to extend the family of companion matrices introduced in [17].

Definition 2.3 We denote by $\mathscr{R}_{n,k}$ the set of block-partitioned matrix pencils with block entries in $\mathbb{F}[A_0, \ldots, A_k]$ and whose only nonzero blocks are of the form:

- $k-1$ blocks equal to $-I$, together with $k-1$ blocks of the form λI, and
- at most k nonzero blocks, denoted by $B_0(\lambda), \ldots, B_{k-1}(\lambda)$, such that each coefficient A_i, for $i = 0, \ldots, k$, appears only in one B_j, for $j = 0, \ldots, k-1$. These blocks are of the form

$$B_j(\lambda) = B_j^0 + \lambda B_j^1, \tag{2}$$

for $j = 0, \ldots, k-1$, with $B_j(\lambda)$ being either 0, A_i, λA_{i+1} or $A_i + \lambda A_{i+1}$, for some $0 \leq i \leq k-1$.

The generality of the family $\mathscr{R}_{n,k}$ relies on the fact that nothing is said about the location of the nonzero blocks in Definition 2.3. Because of this generality, not all pencils in $\mathscr{R}_{n,k}$ are companion pencils, as we are going to see. The following subclass of $\mathscr{R}_{n,k}$ will comprise, up to permutation, all companion pencils of $\mathscr{R}_{n,k}$ (see Theorem 3.1).

Definition 2.4 $\mathscr{Q}_{n,k}$ is the class of block-partitioned pencils in $\mathscr{R}_{n,k}$ where:

- the blocks equal to $-I$ are in all super-diagonal positions (i.e., the block entries $(i, i+1)$, for $i = 1, \ldots, k-1$),
- the blocks equal to λI, together with a nonzero block B_d, for some $0 \leq d \leq k-1$, are on the main diagonal, and
- the remaining nonzero blocks B_j, for $j \neq d$, are below the main diagonal.

However, it is not difficult to see that not every matrix pencil in $\mathscr{Q}_{n,k}$ is a companion pencil [17, p. 261–262], since these pencils do not necessarily satisfy condition (ii) of Definition 2.2. The following result provides some necessary conditions in order for a pencil in $\mathscr{Q}_{n,k}$ to be companion.

Theorem 2.5 *Let $L(\lambda) \in \mathscr{Q}_{n,k}$ be a companion pencil. Then,*

(i) If B_j, for $1 \leq j \leq k-2$, is located in the ith subdiagonal, for $1 \leq i \leq k-2$, then B_j is either 0, A_{k-i-1}, λA_{k-i}, or $A_{k-i-1} + \lambda A_{k-i}$.
(ii) If B_j is located in the $(k-1)$th subdiagonal, then B_j is either A_0 or $A_0 + \lambda A_1$.
(iii) If B_j is located on the main diagonal, then B_j is either λA_k or $\lambda A_k + A_{k-1}$.

In order to prove Theorem 2.5 we use the following result.

Lemma 2.6 *Let $L(\lambda) = [l_{ij}] \in \mathscr{Q}_{1,k}$. For any nonzero l_{st} with $s - t \geq 0$, the determinant of $L(\lambda)$ contains a nonzero summand of the form:*

$$l_{11} \cdots l_{t-1,t-1} l_{st} l_{s+1,s+1} \cdots l_{kk}. \tag{3}$$

Proof Spanning either across the row or the column containing l_{st}, for some $s-t \geq 0$, we obtain that the only term in $\det(L(\lambda))$ containing l_{st} is $l_{st}C_{st}$, where C_{st} is the cofactor of the block entry l_{st}. This cofactor is of the form:

$$C_{st} = (-1)^{s+t} \det \left(\left[\begin{array}{c|c|c} \begin{matrix} l_{11} & -1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & \\ * & & & -1 \\ & & & l_{t-1,t-1} \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \\ & & \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \\ & & \end{matrix} \\ \hline \begin{matrix} & & \\ & * & \\ & & \\ & & \end{matrix} & \begin{matrix} -1 & 0 & \\ & \ddots & \\ * & & -1 \\ & & \end{matrix} & \begin{matrix} & & \\ & 0 & \\ & & \\ & & \end{matrix} \\ \hline \begin{matrix} & & \\ & * & \\ & & \\ & & \end{matrix} & \begin{matrix} & & \\ & * & \\ & & \\ & & \end{matrix} & \begin{matrix} l_{s+1,s+1} & -1 & 0 \\ & \ddots & \ddots \\ & & \ddots \\ * & & -1 \\ & & l_{kk} \end{matrix} \end{array} \right] \right), \tag{4}$$

that is, $C_{st} = l_{11} \cdots l_{t-1,t-1} l_{s+1,s+1} \cdots l_{kk} + \widetilde{C}_{st}$. Recall that below the main diagonal of $L(\lambda)$ there can be nonzero entries. Since $L(\lambda) \in \mathscr{Q}_{1,k}$, the first summand in C_{st} has degree $k - (s-t) - 1$. It suffices to prove that $\widetilde{C}_{st}$ has, at most, degree $k - (s-t) - 2$.

First, the matrix in (4) is partitioned in six big nonzero blocks. Note that each summand in $\widetilde{C}_{st}$ contains a term below the main diagonal multiplied by its cofactor, in particular, this term can be in one of the $(1,1)$, $(2,1)$, $(2,2)$, $(3,1)$, $(3,2)$, or $(3,3)$ blocks. If it is in any of the blocks $(2,1)$, $(2,2)$, or $(3,2)$, its cofactor is 0. However, if it is in the remaining blocks $(1,1)$, $(3,1)$, or $(3,3)$, its cofactor is obtained by removing two terms on the main diagonal, which are of degree 1 in λ, and the cofactor is multiplied by the term on the subdiagonal, which is, at most, of degree 1 in λ. Then, $\widetilde{C}_{st}$ is, at most, of degree $k - (s-t) - 2$ in λ.

Finally, by Definition 2.3, l_{st} contains some coefficient a_r, which does not appear in any other entry, so (3) cannot cancel out with any other term in $\det(L(\lambda))$. ■

Proof (Theorem 2.5) We focus on the case $n = 1$. Let $L(\lambda) \in \mathscr{Q}_{1,k}$. If a_i, for $0 \leq i \leq k$, is in l_{st} (in the rth subdiagonal, with $r := s - t$), then the exponent of λ which appears multiplied by a_i in $\det(L(\lambda))$ is, by (3), equal to $k - r - 1 + \deg(l_{st})$ (note that $l_{11}, \ldots, l_{t-1,t-1}, l_{s+1,s+1}, \ldots, l_{kk}$ all have degree 1). Then, $i = k - r - 1 + \deg(l_{st})$, so $r = k - i - 1 + \deg(l_{st})$, but $\deg(l_{st})$ is either 0 or 1, and a_i must be either in the $(k - i - 1)$th subdiagonal (without λ) or in the $(k - i)$th subdiagonal (multiplied by λ). In particular, when $i = 0$, the only possibility is $\deg(l_{st}) = 0$ and $r = k - 1$ (otherwise, if $\deg(l_{st}) = 1$, we would have $r = k$, which is not possible, since there are no k subdiagonals), and similarly when $i = k$, the only possibility is $\deg(l_{st}) = 1$ and $r = 0$. This means that a_0 can only be in the $(k - 1)$th subdiagonal (without λ) and a_k can only be in the 0th subdiagonal (multiplied by λ).

As a consequence, if $b_j = b_j^0 + \lambda b_j^1$ is in the ith subdiagonal, then b_j^0 can be either 0 or a_{k-i-1}, and b_j^1 can be either 0 or a_{k-i}. ∎

Now, we introduce the following class of block-partitioned pencils, where part (i) is motivated by Theorem 2.5.

Definition 2.7 $\mathscr{QC}_{n,k}$ is the class of block-partitioned pencils in $\mathscr{Q}_{n,k}$ satisfying the following conditions:

(i) The coefficient A_i is either in the $(k - i - 1)$th subdiagonal or in the $(k - i)$th subdiagonal. In the first case it appears without λ, and in the second one it appears multiplied by λ.

(ii) (Rectangle condition). All possible nonzero blocks B_j, for $j = 0, 1, \ldots, k - 1$, lie on the rectangular block-partitioned submatrix whose upper right corner is the position containing A_k, which is on the main diagonal (denoted as B_{k-1}), and whose lower left corner is the position containing A_0 (denoted by B_0), namely the $(k, 1)$ position.

The following example illustrates the difference between Definitions 2.4 and 2.7.

Example 2.8 Let $Q(\lambda) = \sum_{i=0}^{4} \lambda^i A_i$ be an $n \times n$ matrix polynomial of degree 4. Let us consider the following block-partitioned matrix pencils $L_1(\lambda)$ and $L_2(\lambda)$.

$$L_1(\lambda) = \begin{bmatrix} \lambda I & -I & 0 & 0 \\ \boxed{\begin{matrix} 0 & A_3 + \lambda A_4 \\ 0 & 0 \end{matrix}} & & -I & 0 \\ A_0 & A_1 & \lambda I & -I \\ & & A_2 & \lambda I \end{bmatrix} \quad \text{and} \quad L_2(\lambda) = \begin{bmatrix} \lambda I & -I & 0 & 0 \\ \boxed{\begin{matrix} A_2 + \lambda A_3 & \lambda A_4 \\ 0 & 0 \end{matrix}} & & -I & 0 \\ A_0 + \lambda A_1 & 0 & \lambda I & -I \\ & & 0 & \lambda I \end{bmatrix}.$$

In $L_1(\lambda)$, the coefficient A_2 is not inside the rectangular block-partitioned submatrix indicated with a box. In $L_2(\lambda)$, instead, all nonzero blocks below the main diagonal are in this rectangle. Then, $L_1(\lambda) \in \mathscr{Q}_{n,4} \backslash \mathscr{QC}_{n,4}$, and $L_2(\lambda) \in \mathscr{QC}_{n,4}$.

Figure 1 illustrates the relationship between the classes $\mathscr{R}_{n,k}$, $\mathscr{Q}_{n,k}$, and $\mathscr{QC}_{n,k}$.

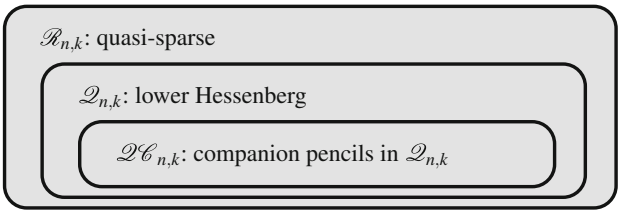


Fig. 1 An illustrative diagram clarifying the relations among the three classes of matrix pencils introduced in Sect. 2.1

3 Companion Pencils in $\mathscr{Q}_{n,k}$

Our first result shows that any companion pencil in $\mathscr{R}_{n,k}$ can be taken to the form $\mathscr{Q}_{n,k}$ by block permutations of rows and columns.

Theorem 3.1 *Any companion pencil in $\mathscr{R}_{n,k}$ is block permutationally equivalent to a pencil in $\mathscr{Q}_{n,k}$.*

Proof We can focus on the case $n = 1$ for simplicity. All developments are also true for arbitrary n.

Let $S(\lambda) \in \mathscr{R}_{1,k}$ be a companion pencil for the scalar polynomial $q(\lambda) = \sum_{i=0}^{k} \lambda^i a_i$, which has $k-1$ entries equal to -1, together with $k-1$ entries equal to λ and, at most, k nonzero entries that we order as $b_0, \ldots, b_{k-1}$. The polynomials b_j, for $j = 0, \ldots, k-1$, are equal to either 0, a_i, λa_{i+1}, or $a_i + \lambda a_{i+1}$, for some $0 \leq i \leq k-1$, as in Definition 2.3. Suppose, b_{k-1} is the entry containing a_k and b_0 is the one containing a_0. Then, b_{k-1} and b_0 must be of the form

$$b_{k-1}(\lambda) = \begin{cases} \lambda a_k, \text{ or} \\ a_{k-1} + \lambda a_k, \end{cases} \text{ and } b_0(\lambda) = \begin{cases} a_0, \text{ or} \\ a_0 + \lambda a_1. \end{cases}$$

Since $S(\lambda)$ is a companion pencil, $\det(S(\lambda)) = \alpha\, q(\lambda) = \alpha \sum_{i=0}^{k} \lambda^i a_i$, with $0 \neq \alpha \in \mathbb{F}$ (note that, since the leading term of $\det(S(\lambda))$ comes from the product of the $k-1$ entries equal to λ, together with b_{k-1}, it must be $\alpha = \pm 1$). This identity is satisfied for all values of the coefficients a_i. Then, we can shrink to zero some coefficients a_i or give them some specific values and the identity, for these particular values, must be true as well.

In the first place, we shrink to zero all the coefficients a_i of $q(\lambda)$ which are not in b_{k-1}, that is, we assume that all entries b_j are zero except b_{k-1}. In this case, $\det(S(\lambda)) = \alpha(\lambda^{k-1} b_{k-1})$. This implies that all entries equal to λ, together with b_{k-1}, are in different rows and columns of $S(\lambda)$.

Similarly, by shrinking to zero all the coefficients a_i of $q(\lambda)$ which are not in b_0, we conclude that all entries equal to -1, together with b_0, are in different rows and columns of $S(\lambda)$.

Now, we can find two permutation matrices P_1, P_2 such that

$$\widetilde{S}(\lambda) := P_1 S(\lambda) P_2 = \begin{bmatrix} \lambda & * & * & * & * & * & * \\ & \ddots & & & & & \\ * & * & \ddots & * & * & * & * \\ * & * & \lambda & * & * & * & * \\ * & * & * & b_{k-1} & * & * & * \\ * & * & * & * & \lambda & * & * \\ & & & & & \ddots & \\ * & * & * & * & * & * & \ddots \\ * & * & * & * & * & * & \lambda \end{bmatrix}.$$

To be precise, P_2 is built up as follows: $S(\lambda)$ has, in each row, only one element equal to either λ or b_{k-1}. Then, we can define P_2 as the matrix that takes this element, which is in the position (i, j_i), to the position (i, i), for each $i = 1, \ldots, k$.

Similarly, we can proceed with columns instead of rows to define P_1. Then, we only need one of P_1 or P_2, depending on whether we perform row or column permutations. Therefore, up to permutational equivalence, we get the pencil $\widetilde{S}(\lambda)$, with the same entries as $S(\lambda)$, but with the $k-1$ entries equal to λ, together with b_{k-1}, on the main diagonal.

There are, at most, $2(k-1)$ nonzero entries $(*)$ in $\widetilde{S}(\lambda)$, which are the $k-1$ entries equal to -1 together with the polynomials b_j, for $j = 0, \ldots, k-2$. Note that b_{k-1} can be in any position on the main diagonal, and that the $k-1$ entries equal to -1, together with b_0, are also in different rows and columns. We are going to show that there is a permutation matrix $\widetilde{P}_1$ such that

$$\widehat{S}(\lambda) := \widetilde{P}_1 \widetilde{S}(\lambda) \widetilde{P}_1^T = \begin{bmatrix} \lambda & -1 & * & * & * & * & * \\ & \ddots & \ddots & & & & \\ * & \ddots & \ddots & * & * & * & * \\ * & * & \lambda & -1 & * & * & * \\ * & * & * & b_{k-1} & -1 & * & * \\ & & & & & \ddots & \\ * & * & * & * & \lambda & \ddots & * \\ & & & & & \ddots & \\ * & * & * & * & * & \ddots & -1 \\ * & * & * & * & * & * & \lambda \end{bmatrix}.$$

Note that the entries equal to λ and b_{k-1} remain on the main diagonal in $\widehat{S}(\lambda)$.

It suffices to prove that the k entries (-1's and b_0) form a k-cycle. For this, let us shrink to zero all coefficients a_i of $q(\lambda)$ other than a_0 and a_k, set $a_k = 1$, and denote by $\widetilde{S}_1(\lambda)$ the pencil obtained from $\widetilde{S}(\lambda)$ after this replacement. Then $\det(\widetilde{S}_1(\lambda)) = \lambda^k + a_0$. Moreover, $\widetilde{S}_1(\lambda)$ does not contain any other terms with degree 1 in λ than the ones on the main diagonal, so $\widetilde{S}_1(\lambda) = \lambda I - A$, with A being a companion matrix for the polynomial $\lambda^k + a_0$.

As a consequence of Lemma 2.1 in [17], we conclude that the -1 entries, together with the one containing a_0, must be in a cycle of length k.

Therefore, up to permutational similarity, we arrive at $\widehat{S}(\lambda)$, having the same entries as $\widetilde{S}(\lambda)$, but the $k-1$ entries equal to -1 being on the super-diagonal and the entry b_0 being in the position $(k, 1)$.

Finally, let us assume, by contradiction, that $\widehat{S}(\lambda)$ has, at least, one entry b_t, for some $1 \leq t \leq k-2$, above the super-diagonal, that is, in the position (i_t, j_t), with $i_t < j_t - 1$.

Let us shrink to zero all the coefficients a_i of $q(\lambda)$ which are not in b_t and denote by $\widehat{S}_1(\lambda)$ the pencil obtained after this replacement. Note that either $b_t(\lambda) = a_r$, $b_t(\lambda) = \lambda a_{r+1}$, or $b_t(\lambda) = a_r + \lambda a_{r+1}$, for some $1 \leq r \leq k-2$. We suppose that $b_t(\lambda) = a_r + \lambda a_{r+1}$ (the proof is analogous for the other two cases). In this case, since $\widehat{S}(\lambda)$ is a companion pencil, it must be $\det(\widehat{S}_1(\lambda)) = \lambda^r a_r + \lambda^{r+1} a_{r+1}$, where

$$\widehat{S}_1(\lambda) := \widehat{S}(0, \ldots, 0, a_r, a_{r+1}, 0, \ldots, 0; \lambda) = \begin{bmatrix} \lambda & -1 & & & & \\ & \ddots & \ddots & & & \\ & & \lambda & \ddots & b_t & \\ & & & 0 & \ddots & \\ & & & & \lambda & \ddots \\ & & & & & \ddots & -1 \\ & & & & & & \lambda \end{bmatrix}.$$

This matrix is upper triangular and its determinant is the product of the entries on the main diagonal, that is, $\det(\widehat{S}_1) = 0 \neq \lambda^r a_r + \lambda^{r+1} a_{r+1}$, which is a contradiction. Therefore, all nonzero entries in $\widehat{S}(\lambda)$ are on or below the super-diagonal. ■

Next we prove that any companion pencil in $\mathscr{Q}_{n,k}$ belongs to $\mathscr{QC}_{n,k}$.

Theorem 3.2 *Let $L(\lambda) \in \mathscr{Q}_{n,k}$ be a companion pencil. Then, $L(\lambda) \in \mathscr{QC}_{n,k}$.*

Proof As in the proof of Theorem 3.1, we can focus on the case $n = 1$. Then, let $L(\lambda) \in \mathscr{Q}_{1,k}$. By Theorem 2.5, we know that if the polynomial b_j, for $j = 1, \ldots, k-2$, is in the ith subdiagonal of $L(\lambda)$, for $i = 1, \ldots, k-2$, then b_j is either 0, a_{k-i-1}, λa_{k-i}, or $a_{k-i-1} + \lambda a_{k-i}$. Therefore, it satisfies condition (i) in Definition 2.7. It remains to prove that $L(\lambda)$ satisfies condition (ii) in Definition 2.7.

By contradiction, let us assume that there is an entry b_t, for some $1 \leq t \leq k-2$, outside the rectangle. Then it is located in the position (i_t, j_t), with either $j_t < i_t < i_0$ or $i_t > j_t > i_0$, where $l_{i_0,i_0}(\lambda) = b_{k-1}(\lambda)$. Let us shrink to zero all the coefficients a_i of the polynomial $q(\lambda)$ which are not in b_t or b_{k-1} and let us denote by $L_1(\lambda)$ the pencil obtained after doing this. Then, $L_1(\lambda)$ is of the form

$$L_1(\lambda) = \begin{bmatrix} \lambda & -1 & & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ & b_t & & \ddots & & \\ 0 & \cdots & 0 & b_{k-1} & \ddots & \\ \vdots & & \vdots & & \ddots & \\ \vdots & & \ddots & 0 & \ddots & \ddots \\ \vdots & & & \vdots & \ddots & -1 \\ 0 & \cdots & \cdots & 0 & & \lambda \end{bmatrix} \text{ or } \begin{bmatrix} \lambda & -1 & & & & \\ & \ddots & \ddots & & & \\ & & \ddots & \ddots & & \\ 0 & \cdots & 0 & b_{k-1} & \ddots & \\ \vdots & & \vdots & & \ddots & \\ \vdots & & \ddots & 0 & \ddots & \ddots \\ \vdots & & & \vdots & \ddots & -1 \\ 0 & \cdots & \cdots & 0 & b_t & \lambda \end{bmatrix},$$

depending on whether $j_t < i_t < i_0$ or $i_t > j_t > i_0$. In both cases, the entries not explicitly indicated in the matrix are zero. Let us first assume that $j_t < i_t < i_0$.

By Lemma 2.6, the determinant of $L_1(\lambda)$ will contain a nonzero term of the form:

$$\underbrace{l_{11}\cdots l_{j_t-1,j_t-1}}_{\lambda' s}\cdot\underbrace{l_{i_t,j_t}}_{b_t}\cdot\underbrace{l_{i_t+1,i_t+1}\cdots l_{kk}}_{\lambda' s \text{ and } b_{k-1}} = \lambda^{k-(i_t-j_t)-2}b_t b_{k-1}.$$

Therefore, $\det(L_1(\lambda))$ contains a term involving the product $b_t b_{k-1}$, which involves in turn a product of coefficients a_i of $q(\lambda)$, and this is in contradiction with the fact that $L(\lambda)$ is a companion pencil.

It remains to analyze the case where $i_0 < j_t < i_t$. The determinant of $L_1(\lambda)$ for this case contains a nonzero term of the form:

$$\underbrace{l_{11}\cdots l_{j_t-1,j_t-1}}_{\lambda' s \text{ and } b_{k-1}}\cdot\underbrace{l_{i_t,j_t}}_{b_t}\cdot\underbrace{l_{i_t+1,i_t+1}\cdots l_{kk}}_{\lambda' s} = \lambda^{k-(i_t-j_t)-2}b_t b_{k-1}.$$

As above, this is a contradiction with the fact that $L(\lambda)$ is a companion pencil. ■

Theorem 3.2 tells us that a matrix pencil in $\mathscr{Q}_{n,k}$ must belong to $\mathscr{QC}_{n,k}$ in order to be a companion pencil. The following result shows that, moreover, all pencils in $\mathscr{QC}_{n,k}$ are companion.

Theorem 3.3 *Any pencil in $\mathscr{QC}_{n,k}$ is a companion pencil.*

Proof Let $L(\lambda) \in \mathscr{QC}_{n,k}$ be an $nk \times nk$ matrix pencil. If B_{k-1} is in the entry $(p+1, p+1)$, for some $0 \le p \le k-1$, then we can write $L(\lambda)$ as the following block-partitioned matrix pencil:

$$L(\lambda) = \left[\begin{array}{ccccc|c}
 & & \lambda I & -I & & \\
 & & & \ddots & \ddots & \\
 & & & & \lambda I & -I \\
\hline
M_{p+1,1} & \cdots & M_{p+1,p} & M_{p+1,p+1} & & -I \\
\vdots & & & & M_{p+2,p+1} & \lambda I \\
 & & & & & \ddots \\
M_{k-1,1} & & & & & \\
M_{k1} & M_{k2} & \cdots & & M_{k,p+1} & \ddots \\
 & & & & & -I \\
 & & & & & \lambda I
\end{array}\right]
\begin{array}{l} \left.\begin{array}{l}\\\\\\\end{array}\right\} pn \\ \left.\begin{array}{l}\\\\\\\\\end{array}\right\} (q+1)n \end{array},$$

with $p+q+1 = k$. Note that $M_{k1} = B_0$, $M_{p+1,p+1} = B_{k-1}$, and the remaining blocks M_{st}, for $s = p+1, \ldots, k$ and $t = 1, \ldots, p+1$, are either 0 or B_j, for $j = 1, \ldots, k-2$. Note that, by Theorem 2.5, if B_j, for $j = 1, \ldots, k-2$, is in the rth subdiagonal of $L(\lambda)$, for $r = 1, \ldots, k-2$, then it is either 0, A_{k-r-1}, λA_{k-r}, or $A_{k-r-1} + \lambda A_{k-r}$.

Now, we consider the following two block permutations P_{row} and P_{col}:

- P_{row} permutes the rows of $L(\lambda)$. Note that $L(\lambda)$ is partitioned in two big blocks of block-partitioned matrices by rows; the first block-partitioned matrix includes rows from 1 to p and the second one includes rows from $p+1$ to k. We define P_{row} as the matrix taking: $s \mapsto k-s+1$, for $s = 1, \ldots, p$, and $s \mapsto s-p$, for $s = p+1, \ldots, k$.

- P_{col} permutes only the first $p+1$ columns of the matrix $L(\lambda)$. We define P_{col} as the matrix taking: $t \to p+2-t$, for $t = 1, \ldots, p+1$.

It is straightforward to see that:

$$\widetilde{L}(\lambda) := P_{\text{row}} L(\lambda) P_{\text{col}} = \left[\begin{array}{ccccc|c} M_{p+1,p+1} & M_{p+1,p} & \cdots & M_{p+1,1} & & -I & \\ & \ddots & & \vdots & & \lambda I & \ddots \\ M_{p+2,p+1} & \ddots & \ddots & & & & \ddots \\ \vdots & \ddots & \ddots & M_{k-1,1} & & & -I \\ M_{k,p+1} & \cdots & M_{k2} & M_{k1} & & & \lambda I \\ \hline & -I & \lambda I & & & & \\ & \ddots & \ddots & & & & \\ & & -I & \lambda I & & & \end{array}\right] \begin{array}{l} \Big\} (q+1)n \\ \Big\} pn \end{array} \quad (5)$$

$$\underbrace{\qquad\qquad}_{(p+1)n} \qquad \underbrace{\quad}_{qn}$$

In particular, the (block) diagonals in the $(2,1)$ big block of $L(\lambda)$ become the (block) anti-diagonals of the $(1,1)$ big block in $\widetilde{L}(\lambda)$. Now, let us consider a general pencil like in the right-hand side of (5). In particular, the blocks M_{st} are matrix pencils. It is shown in [15, Thm. 5.4] that if the sum of the trailing coefficients of all M_{st} blocks in the $(k-i-1)$th anti-diagonal plus the sum of the leading coefficients of all M_{st} blocks in the $(k-i)$th anti-diagonal equals A_i, for all $i = 0, 1, \ldots, k$, then $\widetilde{L}(\lambda)$ is a strong linearization of $Q(\lambda)$. By condition (i) in Definition 2.7, this condition on the leading and trailing coefficients of the M_{st} blocks is satisfied for the particular $\widetilde{L}(\lambda)$ coming from $L(\lambda)$ as in (5). Therefore, $\widetilde{L}(\lambda)$ is a strong linearization of $Q(\lambda)$. Since $L(\lambda)$ is permutationally equivalent to $\widetilde{L}(\lambda)$, $L(\lambda)$ is also a strong linearization of $Q(\lambda)$. Then, $L(\lambda)$ satisfies condition (ii) in Definition 2.2 and, by definition, $L(\lambda)$ satisfies condition (i) as well, so it is a companion pencil. ■

The proof of Theorem 3.3 shows that the family of block-Kronecker pencils introduced in [15, Def. 5.1] comprise, up to block permutation, all companion pencils in $\mathscr{R}_{n,k}$ (in other words, it contains, up to block permutation, all pencils in $\mathscr{DC}_{n,k}$).

4 Number of Different Sparse Companion Pencils in $\mathscr{R}_{n,k}$

We say that companion pencils in $\mathscr{R}_{n,k}$ are quasi-sparse companion pencils, since they have a small number of nonzero block entries. However, not all companion pencils in $\mathscr{R}_{n,k}$ have the same number of nonzero block entries. We first give a lower bound on the number of nonzero block entries of a companion pencil in $\mathscr{R}_{n,k}$. In the following, $\lfloor r \rfloor$ and $\lceil r \rceil$ denote, respectively, the floor and the ceiling of $r \in \mathbb{R}$. Note that, if r is an integer, then $r = \lfloor r \rfloor = \lceil r \rceil$.

Lemma 4.1 *Any companion pencil in $\mathscr{R}_{n,k}$ has, at least, $2k - 1 + \left\lfloor \frac{k}{2} \right\rfloor$ nonzero block entries.*

Proof By Theorems 3.1 and 3.2, any companion pencil in $\mathscr{R}_{n,k}$ is permutationally equivalent to a pencil in $\mathscr{DC}_{n,k}$, so we focus on pencils in $\mathscr{DC}_{n,k}$. Recall

(Theorem 2.5) that the blocks B_j, for $j = 0, \ldots, k-1$, are equal to either 0, A_i, λA_{i+1}, or $A_i + \lambda A_{i+1}$, for some $0 \leq i \leq k-1$. Note that the entries equal to $-I$ and λI add up to $2(k-1)$ nonzero block entries. Regarding the $k+1$ coefficients A_i ($i = 0, \ldots, k$), they can be grouped in the pencils $B_j = B_j^0 + \lambda B_j^1$ ($j = 0, \ldots, k-1$). If k is odd, then $\frac{k+1}{2} = \lceil \frac{k+1}{2} \rceil$ is the smallest number of nonzero blocks B_j, which are of the form $A_{2i} + \lambda A_{2i+1}$, for $i = 0, 1, \ldots, \frac{k-1}{2}$. However, if k is even, different groupings are possible, but in all cases the smallest number of nonzero blocks B_j is $\left\lfloor \frac{k+1}{2} \right\rfloor + 1 = \left\lceil \frac{k+1}{2} \right\rceil$. Adding up, we get $2(k-1) + \left\lceil \frac{k+1}{2} \right\rceil = 2k - 1 + \lfloor \frac{k}{2} \rfloor$ nonzero block entries. □

Lemma 4.1 motivates the following definition.

Definition 4.2 A sparse pencil in $\mathscr{R}_{n,k}$ is a pencil with exactly $2k-1+\lfloor \frac{k}{2} \rfloor$ nonzero block entries.

Since any companion pencil in $\mathscr{R}_{n,k}$ is permutationally equivalent to a pencil in $\mathscr{DC}_{n,k}$, to count the number of sparse companion pencils in $\mathscr{R}_{n,k}$, up to permutation, we can just count the number of non-permutationally equivalent pencils in $\mathscr{DC}_{n,k}$. First, Theorem 4.3 guarantees that no two of them are permutationally equivalent.

Theorem 4.3 *Two different matrix pencils in $\mathscr{DC}_{n,k}$ are not block permutationally equivalent.*

Proof Let $L_1(\lambda), L_2(\lambda) \in \mathscr{DC}_{n,k}$. If $L_1(\lambda)$ is block permutationally equivalent to $L_2(\lambda)$ there exists two block-partitioned permutation matrices P, P' with

$$PL_1(\lambda)P' = L_2(\lambda). \tag{6}$$

Let us shrink to zero all coefficients A_i, for $i = 0, \ldots, k-1$, and let $A_k = I$. Looking only at the leading terms in (6), we get $P \cdot I \cdot P' = I$, so $P = (P')^{-1}$. Then, $L_1(\lambda)$ and $L_2(\lambda)$ are block permutationally similar.

It suffices to prove that two different matrix pencils in $\mathscr{DC}_{n,k}$ are not block permutationally similar. $L_1(\lambda)$ is block permutationally similar to $L_2(\lambda)$ if there exists a block-partitioned permutation matrix P such that

$$PL_1(\lambda)P^B = L_2(\lambda) \tag{7}$$

(where $(\cdot)^B$ stands for block transposition). We prove that the only permutation matrix P satisfying (7) is the identity matrix, which implies $L_1(\lambda) = L_2(\lambda)$. For this, we can focus on the case $n = 1$.

Again, by shrinking to zero all entries a_i, for $i = 0, \ldots, k$, and equating the trailing coefficients in (7) we get $PN = NP$, with

$$N = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ & & & 0 \end{bmatrix}.$$

It is straightforward to check that the only permutation matrix P with $PN = NP$ is $P = I$. ■

The following lemmas will help us to get the number of sparse pencils in $\mathscr{DC}_{n,k}$.

Lemma 4.4 *If $L(\lambda) \in \mathscr{DC}_{n,k}$, then $L(\lambda)$ cannot have two consecutive null subdiagonals.*

Proof Just recall, by condition (i) in Definition 2.7, that A_{k-j} is either in the $(j-1)$th subdiagonal or in the jth subdiagonal (as λA_{k-j}), for $j = 1, \ldots, k-2$. ■

Lemma 4.5 *If $L(\lambda) \in \mathscr{DC}_{n,k}$ is sparse, then it cannot have two nonzero block entries in the same subdiagonal.*

Proof If $L(\lambda) \in \mathscr{DC}_{n,k}$ is sparse and it has two nonzero block entries in the jth subdiagonal, for some $1 \leq j \leq k-2$, these must be A_{k-j-1} and λA_{k-j} (by Definitions 2.3 and 2.7). Then, by joining $A_{k-j-1} + \lambda A_{k-j}$ in the same entry (either the one containing A_{k-j-1} or the one containing λA_{k-j}) we arrive at a new pencil in $\mathscr{DC}_{n,k}$ having less nonzero block entries than $L(\lambda)$, which is a contradiction with the fact that $L(\lambda)$ is sparse. ■

The following result gives us the exact number of zero subdiagonals (including the main diagonal as the 0th subdiagonal) of any sparse pencil in $\mathscr{DC}_{n,k}$.

Lemma 4.6 *Let $L(\lambda) \in \mathscr{DC}_{n,k}$ be sparse. Then $L(\lambda)$ has exactly $k - 1 - \lfloor \frac{k}{2} \rfloor$ null subdiagonals.*

Proof If $L(\lambda) \in \mathscr{DC}_{n,k}$ is sparse, it has $\lfloor \frac{k}{2} \rfloor$ nonzero entries below the diagonal, by Lemma 4.1. By Lemma 4.5, no two nonzero entries of $L(\lambda)$ are in the same subdiagonal, so $L(\lambda)$ has $\lfloor \frac{k}{2} \rfloor$ nonzero subdiagonals and, as a consequence, $k - 1 - \lfloor \frac{k}{2} \rfloor$ null subdiagonals. ■

As a consequence of the previous results, we can explicitly identify which are the only null subdiagonals of any sparse pencil in $\mathscr{DC}_{n,k}$. This is the first step in determining the number of sparse pencils in $\mathscr{DC}_{n,k}$. We start with the case k odd.

Lemma 4.7 *Let $L(\lambda) \in \mathscr{DC}_{n,k}$ be a sparse pencil with k odd. Then the only nonzero jth subdiagonals of $L(\lambda)$ are the ones with indices $j = 0, 2, 4, \ldots, k-1$.*

Proof By Lemma 4.6, $L(\lambda)$ has exactly $\frac{k-1}{2}$ null subdiagonals and, consequently, $\frac{k+1}{2}$ nonzero subdiagonals. Since the main diagonal and the $(k-1)$st subdiagonal (the entry $(k, 1)$) are nonzero, among the remaining $k-2$ subdiagonals there are $\frac{k-1}{2}$ null ones, together with another $\frac{k-3}{2}$ nonzero ones. Since, by Lemma 4.4, there cannot be two consecutive null subdiagonals in $L(\lambda)$, the first and the $(k-2)$nd subdiagonal must be zero, and the zero/nonzero subdiagonals must alternate. □

For k even, the situation is more involved. As there are $k+1$ coefficients A_i, for $i = 0, \ldots, k$, there must be $\frac{k}{2}$ nonzero blocks of the form $A_{k-j-1} + \lambda A_{k-j}$, together with another nonzero block of the form $B_\ell = \lambda A_{k-j}$ or $B_\ell = A_{k-j-1}$, for $0 \leq j \leq k-1$. In particular, the pattern of zero/nonzero subdiagonals in $\mathscr{DC}_{n,k}$

depends on the position of this monomial B_ℓ. This is stated in Lemma 4.8, which also establishes some restrictions about B_ℓ.

Lemma 4.8 *Let $L(\lambda) \in \mathscr{DC}_{n,k}$ be a sparse pencil with k even. Then the monomial B_ℓ, located in the jth subdiagonal (for $0 \leq j \leq k-1$), and the indices of the nonzero rth subdiagonals of $L(\lambda)$ are the following:*

(a) If j is even: $B_\ell = \lambda A_{k-j}$, and $r = 0, 2, 4, \ldots, j, j+1, j+3, \ldots, k-3, k-1$;
(b) If j is odd: $B_\ell = A_{k-j-1}$, and $r = 0, 2, 4, \ldots, j-1, j, j+2, \ldots, k-3, k-1$.

Proof By Lemma 4.6, $L(\lambda)$ has exactly $\frac{k}{2} - 1$ zero subdiagonals and, consequently, $\frac{k}{2} + 1$ nonzero subdiagonals. Recall that the main diagonal and the $(k-1)$th subdiagonal (the entry $(k, 1)$) are nonzero. As mentioned above, there is only one nonzero block of the form $B_\ell = \lambda A_{k-j}$ or A_{k-j-1}, which is located in the jth subdiagonal, for some $0 \leq j \leq k-1$. The remaining nonzero blocks are of the form $A_{k-s-1} + \lambda A_{k-s}$, for $s \neq j$. As a consequence, there are four possible situations:

Case 1: $B_\ell = \lambda A_{k-j}$ is in the jth subdiagonal, for $0 \leq j \leq k-1$, with j even.
Case 2: $B_\ell = \lambda A_{k-j}$ is in the jth subdiagonal, for $0 \leq j \leq k-1$, with j odd.
Case 3: $B_\ell = A_{k-j-1}$ is in the jth subdiagonal, for $0 \leq j \leq k-1$, with j even.
Case 4: $B_\ell = A_{k-j-1}$ is in the jth subdiagonal, for $0 \leq j \leq k-1$, with j odd.

By Lemma 4.5, B_ℓ is the only nonzero block entry in the jth subdiagonal. Moreover, (see Theorem 2.5), the block entries in the ith subdiagonal, for $i = 0, \ldots, k-1$, with $i \neq j$, are either: (i) λA_{k-i}, (ii) A_{k-i-1}, (iii) $A_{k-i-1} + \lambda A_{k-i}$, or (iv) 0.

Case 2: In this case, the j coefficients $A_k, A_{k-1}, \ldots, A_{k-j+1}$ are located in the subdiagonals from 0th to $(j-1)$th. Since j is odd, at least another coefficient must be unpaired, so the pencil is not sparse.

Case 3: Now there are $j+1$ coefficients $A_k, A_{k-1}, \ldots, A_{k-j}$ among the subdiagonals from 0th to $(j-1)$th. Since j is even, the pencil is, again, not sparse.

This proves the first part of the statement. The second part follows from Theorem 2.5, together with Lemma 4.4 and Lemma 4.6. In particular, conditions (i)–(iv) above determine the pattern of zero/nonzero subdiagonals, taking into account Lemma 4.4. This is summarized in Table 1.

■

Remark 4.9 Note that if $L_1(\lambda)$ and $L_2(\lambda)$ are two sparse pencils in $\mathscr{DC}_{n,k}$ of even degree k such that $L_1(\lambda)$ has a monomial in the jth subdiagonal (with j even) of the

Table 1 Possible forms of the block entries in the subdiagonals for Cases 1 and 4

Forms of the block entries	Subdiagonal								
	0th	1st	$\cdots$	$(j-1)$th	jth	$(j+1)$th	$\cdots$	$(k-2)$th	$(k-1)$th
Case 1	$A_{k-1} + \lambda A_k$	0	$\cdots$	0	λA_{k-j}	$A_{k-j-2} + \lambda A_{k-j-1}$	$\cdots$	0	$A_0 + \lambda A_1$
Case 4	$A_{k-1} + \lambda A_k$	0	$\cdots$	$A_{k-j} + \lambda A_{k-j+1}$	A_{k-j-1}	0	$\cdots$	0	$A_0 + \lambda A_1$

Table 2 All possible patterns for sparse pencils in $\mathscr{DC}_{n,4}$

Pattern	Nonzero blocks	
$j = 0, 1$		
$\begin{bmatrix} \bigstar & -I & 0 & 0 \\ \bullet & \bigstar & -I & 0 \\ 0 & \bullet & \bigstar & -I \\ \blacktriangle & 0 & \bullet & \bigstar \end{bmatrix}$	$\boxed{j=0}$ $\lambda A_4 \in \bigstar$ $A_2 + \lambda A_3 \in \bullet$ $A_0 + \lambda A_1 = \blacktriangle$	$\boxed{j=1}$ $A_3 + \lambda A_4 \in \bigstar$ $A_2 \in \bullet$ $A_0 + \lambda A_1 = \blacktriangle$
$j = 2, 3$		
$\begin{bmatrix} \bigstar & -I & 0 & 0 \\ 0 & \bigstar & -I & 0 \\ \bullet & 0 & \bigstar & -I \\ \blacktriangle & \bullet & 0 & \bigstar \end{bmatrix}$	$\boxed{j=2}$ $A_3 + \lambda A_4 \in \bigstar$ $\lambda A_2 \in \bullet$ $A_0 + \lambda A_1 = \blacktriangle$	$\boxed{j=3}$ $A_3 + \lambda A_4 \in \bigstar$ $A_1 + \lambda A_2 \in \bullet$ $A_0 = \blacktriangle$

form $B_\ell = \lambda A_{k-j}$ and $L_2(\lambda)$ has a monomial in the $(j+1)$th subdiagonal of the form $B_\ell = A_{k-j-2}$, for $j = 0, 2, \ldots, k-2$, then $L_1(\lambda)$ and $L_2(\lambda)$ have the same nonzero subdiagonals. This is straightforward to see looking at the indices of the nonzero subdiagonals in Lemma 4.8, just replacing, for the case j odd, j by $j+1$.

Example 4.10 In this example we show all possible patterns for sparse pencils in $\mathscr{DC}_{n,4}$ (that is, for quartic $n \times n$ matrix polynomials). Following Lemma 4.8, the zero/nonzero pattern of the subdiagonals depends on the subdiagonal containing the monomial B_ℓ. Let this subdiagonal be the jth one, for $j = 0, 1, 2, 3$. Then:

(a) For $j = 0, 1$, the monomial is $B_\ell = \lambda A_4$ (for $j = 0$) or $B_\ell = A_2$ (for $j = 1$), and the nonzero subdiagonals are the ones with indices 0, 1, and 3 in both cases.
(b) For $j = 2, 3$, the monomial is $B_\ell = \lambda A_2$ (for $j = 2$) or $B_\ell = A_0$ (for $j = 3$). The nonzero subdiagonals are the ones with indices 0, 2, and 3 in both cases.

The patterns are shown in Table 2.

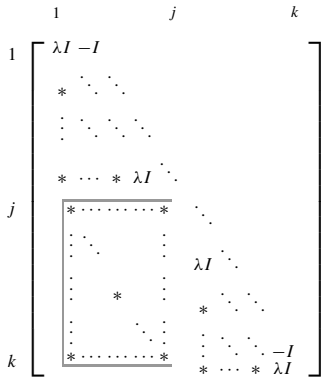
Since we know which are exactly the zero and nonzero subdiagonals in $\mathscr{DC}_{n,k}$, for k odd and even, we can determine the number of different sparse pencils in $\mathscr{DC}_{n,k}$. We use, for a given $k \in \mathbb{N} \bigcup \{0\}$, the *double factorial of* k, defined by the recurrence relation: $k!! := (k-2)!! \cdot k$, if $k \geq 2$, and $k!! := 1$ if $k \leq 1$.

Theorem 4.11 *The number of different sparse pencils in $\mathscr{DC}_{n,k}$, for k odd, is:*

$$2\left(\sum_{j=1}^{\left\lfloor \frac{k}{4} \right\rfloor} ((2j-1)!!)^2 \left((2j-1)^{\left\lceil \frac{k-4j}{2} \right\rceil} + (2j)^{\left\lceil \frac{k-4j}{2} \right\rceil}\right)\right) + \left(\left(\frac{k-3}{2}\right)!!\right)^2 \left(\frac{k+1}{2}\right), \text{ if } \left\lfloor \frac{k}{2} \right\rfloor \text{ is even, and}$$

$$2\left(\sum_{j=1}^{\left\lfloor \left\lfloor \frac{k}{2} \right\rfloor /2 \right\rfloor} ((2j-1)!!)^2 \left((2j-1)^{\left\lceil \frac{k-4j}{2} \right\rceil} + (2j)^{\left\lceil \frac{k-4j}{2} \right\rceil}\right)\right) + 3\left(\left(\frac{k-1}{2}\right)!!\right)^2, \text{ if } \left\lfloor \frac{k}{2} \right\rfloor \text{ is odd.} \tag{8}$$

Proof Let k be an odd integer and let $1 \leq j \leq \left\lfloor \frac{k}{2} \right\rfloor$. We consider the rectangle R_j of an $nk \times nk$ matrix pencil in $\mathscr{DC}_{n,k}$, whose vertices are $(j, 1), (j, j), (k, 1), (k, j)$.



There are $\frac{k+1}{2}$ nonzero blocks among the asterisks $*$. By Lemma 4.7, they are of the form $A_{k-i-1} + \lambda A_{k-i}$, located in the ith subdiagonal, for $i = 0, 2, \ldots, k-1$. We look for the number of different ways to place them in R_j.

This is summarized in Table 3, where we indicate the number of possible positions of each nonzero block in R_j. We need to take into account the parity of j.

Table 3 Number of positions where each nonzero block entry can be in R_j

Coefficient	Can be placed in ...
$\lambda A_k + A_{k-1}$	1 *position* (0*th subdiagonal*)
$\lambda A_{k-2} + A_{k-3}$	3 *positions* (2*nd subdiagonal*)
$\vdots$	$\vdots$
$\begin{cases} \lambda A_{k-(j-2)} + A_{k-(j-1)}, \text{ or} \\ \lambda A_{k-(j-3)} + A_{k-(j-2)} \end{cases}$	$\begin{cases} j-1 \text{ positions } ((j-2)\text{th subdiagonal}) \text{ if } j \text{ even, or} \\ j-2 \text{ positions } ((j-3)\text{th subdiagonal}) \text{ if } j \text{ odd} \end{cases}$
$\begin{cases} \lambda A_{k-j} + A_{k-(j+1)}, \text{ or} \\ \lambda A_{k-(j-1)} + A_{k-j} \end{cases}$	$\begin{cases} j \text{ positions } (j\text{th subdiagonal}) \text{ if } j \text{ even, or} \\ j \text{ positions } ((j-1)\text{th subdiagonal}) \text{ if } j \text{ odd} \end{cases}$
$\vdots$	$\vdots$
$\begin{cases} \lambda A_{j+1} + A_j, \text{ or} \\ \lambda A_j + A_{j-1} \end{cases}$	$\begin{cases} j \text{ positions } ((k-j-1)\text{th subdiagonal}) \text{ if } j \text{ even, or} \\ j \text{ positions } ((k-j)\text{th subdiagonal}) \text{ if } j \text{ odd} \end{cases}$
$\begin{cases} \lambda A_{j-1} + A_{j-2}, \text{ or} \\ \lambda A_{j-2} + A_{j-3} \end{cases}$	$\begin{cases} j-1 \text{ positions } ((k-j+1)\text{th subdiagonal}) \text{ if } j \text{ even, or} \\ j-2 \text{ positions } ((k-j+2)\text{th subdiagonal}) \text{ if } j \text{ odd} \end{cases}$
$\vdots$	$\vdots$
$\lambda A_3 + A_2$	3 *positions* ((k − 3)*th subdiagonal*)
$\lambda A_1 + A_0$	1 *position* ((k − 1)*th subdiagonal*)

Then, the number of possible sparse pencils in $\mathscr{QC}_{n,k}$, for k odd, with all entries in R_j is determined by counting all possible locations for the coefficients in Table 3.

In particular, if $\eta_{n,k}^{(j)}$ denotes the number of possible sparse pencils in $\mathscr{QC}_{n,k}$ with all entries in the rectangle R_j, then:

$$\eta_{n,k}^{(j)} = \begin{cases} 1^2 \cdot 3^2 \cdot 5^2 \cdots (j-1)^2 \cdot (j)^{\left\lceil \frac{k-2j}{2} \right\rceil} = ((j-1)!!)^2 (j)^{\left\lceil \frac{k-2j}{2} \right\rceil}, & \text{if } j \text{ is even, and} \\ 1^2 \cdot 3^2 \cdot 5^2 \cdots (j-2)^2 \cdot (j)^{\left\lceil \frac{k-2j+2}{2} \right\rceil} = ((j-2)!!)^2 (j)^{\left\lceil \frac{k-2j+2}{2} \right\rceil}, & \text{if } j \text{ is odd.} \end{cases} \tag{9}$$

Now we consider all possible rectangles R_j, for $1 \leq j \leq \left\lfloor \frac{k}{2} \right\rfloor$. We only need to look at $j = 1, \ldots, \left\lfloor \frac{k}{2} \right\rfloor$, and also at $j = \frac{k+1}{2}$, since for $j = \frac{k+3}{2}, \ldots, k$, the patterns are symmetric (with respect to the anti-diagonal) to the first $\left\lfloor \frac{k}{2} \right\rfloor$ patterns. In the case $j = \frac{k+1}{2}$, we can obtain the number of possible sparse pencils in $\mathscr{QC}_{n,k}$ with all its entries in the rectangle $R_{\frac{k+1}{2}}$ just replacing j by $\frac{k+1}{2}$ in (9), and this number is equal to:

$$\eta_{n,k}^{\left(\frac{k+1}{2}\right)} = \begin{cases} \left(\left(\frac{k-1}{2}\right)!!\right)^2, & \text{if } \frac{k+1}{2} \text{ is even, and} \\ \left(\left(\frac{k-3}{2}\right)!!\right)^2 \left(\frac{k+1}{2}\right), & \text{if } \frac{k+1}{2} \text{ is odd.} \end{cases} \tag{10}$$

Adding up, the total number of sparse pencils in $\mathscr{QC}_{n,k}$ is:

$$\begin{cases} 2\left(\sum_{j=1}^{\left\lfloor \frac{k}{2} \right\rfloor/2} \left(\eta_{n,k}^{(2j-1)} + \eta_{n,k}^{(2j)}\right)\right) + \eta_{n,k}^{\left(\frac{k+1}{2}\right)}, & \text{if } \left\lfloor \frac{k}{2} \right\rfloor \text{ is even, and} \\ 2\left(\sum_{j=1}^{\left\lceil \left\lfloor \frac{k}{2} \right\rfloor/2 \right\rceil} \eta_{n,k}^{(2j-1)} + \sum_{j=1}^{\left\lfloor \left\lfloor \frac{k}{2} \right\rfloor/2 \right\rfloor} \eta_{n,k}^{(2j)}\right) + \eta_{n,k}^{\left(\frac{k+1}{2}\right)}, & \text{if } \left\lfloor \frac{k}{2} \right\rfloor \text{ is odd,} \end{cases} \tag{11}$$

Using (9) and (10), and grouping summands appropriately in (11), we get (8). ■

Theorem 4.12 *The number of different sparse pencils in $\mathscr{DC}_{n,k}$, for k even, is:*

$$\frac{2}{3}\left(\sum_{j=1}^{\left\lfloor\frac{k}{4}\right\rfloor}((2j-1)!!)^2(3k-4j+4)\left((2j-1)^{\frac{k-4j+2}{2}}+(2j)^{\frac{k-4j+2}{2}}\right)\right), \text{ if } \left\lfloor\frac{k}{2}\right\rfloor \text{ is even, and}$$

$$\frac{2}{3}\left(\sum_{j=1}^{\left\lfloor\frac{k}{4}\right\rfloor}((2j-1)!!)^2(3k-4j+4)\left((2j-1)^{\frac{k-4j+2}{2}}+(2j)^{\frac{k-4j+2}{2}}\right)\right)+$$

$$+\frac{2}{3}\left((2k+2)\left(\left(\frac{k}{2}\right)!!\right)^2\right), \text{ if } \left\lfloor\frac{k}{2}\right\rfloor \text{ is odd.} \tag{12}$$

We will use the following lemma in the proof of Theorem 4.12.

Lemma 4.13 *Let j be an integer. The following identities hold:*

(a) $\dfrac{(j-2)!!}{(j-1)!!}\displaystyle\sum_{i=0}^{\frac{j-4}{2}}\frac{(2i+1)!!}{(2i)!!}=\frac{j-2}{3}$, *if $j\geq 4$ is an even number.*

(b) $\dfrac{(j-1)!!}{(j-2)!!}\displaystyle\sum_{i=0}^{\frac{j-3}{2}}\frac{(2i+1)!!}{(2i)!!}=\frac{j(j-1)}{3}$, *if $j\geq 3$ is an odd number.*

Proof We divide the proof into two cases, depending on the parity of j.

(a) If j is an even number:

$$\frac{(j-2)!!}{(j-1)!!}\sum_{i=0}^{\frac{j-4}{2}}\frac{(2i+1)!!}{(2i)!!}=\frac{2\cdot 4\cdot 6\cdots(j-2)}{3\cdot 5\cdot 7\cdots(j-1)}+\frac{4\cdot 6\cdots(j-2)}{5\cdot 7\cdots(j-1)}+\cdots+\frac{(j-4)(j-2)}{(j-3)(j-1)}+\frac{j-2}{j-1}=$$

$$=\frac{j-2}{(j-1)!!}\Big(2\cdot 4\cdot 6\cdots(j-4)+3\cdot 4\cdot 6\cdots(j-4)+3\cdot 5\cdot 6\cdots(j-4)+\cdots+$$

$$3\cdot 5\cdots(j-5)(j-4)+3\cdot 5\cdots(j-5)(j-3)\Big)=\frac{j-2}{(j-1)!!}\Big(5\cdot 4\cdot 6\cdots(j-4)+3\cdot 5\cdot 6\cdots(j-4)$$

$$+\cdots+3\cdot 5\cdots(j-5)(j-4)+3\cdot 5\cdots(j-5)(j-3)\Big)=\frac{j-2}{(j-1)!!}\Big(5\cdot 7\cdot 6\cdots(j-4)$$

$$+\cdots+$$

$$3\cdot 5\cdots(j-5)(j-4)+3\cdot 5\cdots(j-5)(j-3)\Big)=\cdots+\frac{j-2}{(j-1)!!}\Big(5\cdot 7\cdots(j-1)\Big)=\frac{j-2}{3}.$$

(b) If j is an odd number:

$$\frac{(j-1)!!}{(j-2)!!}\sum_{i=0}^{\frac{j-3}{2}}\frac{(2i+1)!!}{(2i)!!}=\frac{2\cdot 4\cdot 6\cdots(j-1)}{3\cdot 5\cdot 7\cdots(j-2)}+\frac{4\cdot 6\cdots(j-1)}{5\cdot 7\cdots(j-2)}+\cdots+\frac{(j-3)(j-1)}{(j-4)(j-2)}+\frac{j-1}{j-2}=$$

$$=\frac{j-1}{(j-2)!!}\Big(2\cdot 4\cdot 6\cdots(j-3)+3\cdot 4\cdot 6\cdots(j-3)+3\cdot 5\cdot 6\cdots(j-3)+\cdots+$$

$$3\cdot 5\cdots(j-4)(j-3)+3\cdot 5\cdots(j-4)(j-2)\Big)=\frac{j-1}{(j-2)!!}\Big(5\cdot 4\cdot 6\cdots(j-3)+3\cdot 5\cdot 6\cdots(j-3)$$

$$+\cdots+3\cdot 5\cdots(j-4)(j-3)+3\cdot 5\cdots(j-4)(j-2)\Big)=\frac{j-1}{(j-2)!!}\Big(5\cdot 7\cdot 6\cdots(j-3)+\cdots+$$

$$3\cdot 5\cdots(j-4)(j-3)+3\cdot 5\cdots(j-4)(j-2)\Big)=\cdots=\frac{j-1}{(j-2)!!}\Big(5\cdot 7\cdots(j-2)j\Big)=\frac{j(j-1)}{3}.$$

■

Proof (Theorem 4.12) Let k be an even integer and let $1 \leq j \leq \frac{k}{2}$. We consider the rectangle R_j with vertices $(j,1)$, (j,j), $(k,1)$, (k,j), as in the proof of Theorem 4.11. First, we locate the nonzero block entry with only one coefficient, B_ℓ, and, according to Lemma 4.8, the remaining nonzero subdiagonals are uniquely determined.

We assume that $B_\ell = \lambda A_{k-i}$, which is in the ith subdiagonal, for i being an even number (case (a) in Lemma 4.8). For the other case ($B_\ell = A_{k-i-1}$), the nonzero subdiagonals are exactly the same, by Remark 4.9.

As in Theorem 4.11, we have to take into account the parity of the integer j and the position of the monomial B_ℓ. The procedure consists of counting, for each B_ℓ, and for each i even, the number of possible locations of each nonzero block entry inside R_j. It is also important to note that, in R_j,

$$\text{the } i\text{th subdiagonal has } \begin{cases} i+1 \text{ positions, if } 0 \leq i \leq j-2, \\ j \text{ positions,} \qquad \text{if } j-1 \leq i \leq k-j, \text{ and} \\ k-i \text{ positions, if } k-j+1 \leq i \leq k-1. \end{cases} \tag{13}$$

First, let us assume j even. Then, depending on i (i even), we obtain:

- If $i \leq j-4$: The nonzero subdiagonals, by Lemma 4.8, have indices $0, 2, 4, \ldots, i, i+1, \ldots, j-3, j-1, \ldots, k-j-1, k-j+1, \ldots, k-3, k-1$, so, using (13), the number of possible locations for the nonzero blocks inside R_j is:

$$[1\cdot 3\cdot 5\cdots(i+1)(i+2)\cdots(j-2)]\,[j\cdots j]\,[(j-1)\cdots 3\cdot 1]$$
$$=\left[(i+1)!!\frac{(j-2)!!}{(i)!!}\right]\left[j^{\frac{k-2j+2}{2}}\right][(j-1)!!]\,.$$

- If $j-2 \leq i \leq k-j$: The nonzero subdiagonals have indices $0, 2, \ldots, j-2, j, \ldots, i, i+1, \ldots, k-j-1, k-j+1, \ldots, k-3, k-1$, so the number of

possible locations for the nonzero blocks inside R_j is:

$$[1\cdot 3\cdots(j-1)]\,[j\cdots j\cdot j\cdots j]\,[(j-1)\cdots 3\cdot 1] = [(j-1)!!]\left[j^{\frac{k-2j+2}{2}}\right][(j-1)!!]\,.$$

- If $i \geq k-j+2$: The nonzero subdiagonals have indices $0, 2, \ldots, j-2, j, \ldots, k-j, k-j+2, \ldots, i, i+1, \ldots, k-3, k-1$, so, using (13), the number of possible locations for the nonzero block entries inside R_j is:

$$\begin{aligned}&[1\cdot 3\cdots(j-1)]\,[j\cdots j]\,[(j-2)\cdots(k-i)(k-i-1)\cdots 3\cdot 1] = \\ &= [(j-1)!!]\left[j^{\frac{k-2j+2}{2}}\right]\left[(k-i-1)!!\,\frac{(j-2)!!}{(k-i-2)!!}\right].\end{aligned}$$

Finally, if we denote by $E\eta_{n,k}^{(j)}$ the number of possible sparse pencils in $\mathscr{DC}_{n,k}$ with all entries in the rectangle R_j, for j even, then, adding up all the above quantities, we get:

$$\begin{aligned}E\eta_{n,k}^{(j)} = 2\Bigg(&\sum_{i=0}^{\frac{j-4}{2}}\left((2i+1)!!\,\frac{(j-2)!!}{(2i)!!}\right)\left(j^{\frac{k-2j+2}{2}}\right)((j-1)!!) + \sum_{i=\frac{j-2}{2}}^{\frac{k-j}{2}}((j-1)!!)^2\left(j^{\frac{k-2j+2}{2}}\right) + \\ &+ \sum_{i=\frac{k-j+2}{2}}^{\frac{k-2}{2}}((j-1)!!)\left(j^{\frac{k-2j+2}{2}}\right)\left((k-2i-1)!!\,\frac{(j-2)!!}{(k-2i-2)!!}\right)\Bigg).\end{aligned}$$

Note that the first and third summands in the last sum add up to the same number (just replace i by $\frac{k-2}{2} - i$ in the sum of the third term). Moreover, the second summand does not depend on the index i. Then, $E\eta_{n,k}^{(j)}$ is equal to:

$$\begin{aligned}E\eta_{n,k}^{(j)} &= 4\left(j^{\frac{k-2j+2}{2}}\right)((j-1)!!)\,((j-2)!!)\sum_{i=0}^{\frac{j-4}{2}}\frac{(2i+1)!!}{(2i)!!} + 2\,((j-1)!!)^2\left(j^{\frac{k-2j+2}{2}}\right)\frac{k-2j+4}{2} = \\ &= ((j-1)!!)^2\left(j^{\frac{k-2j+2}{2}}\right)\left[4\cdot\frac{(j-2)!!}{(j-1)!!}\sum_{i=0}^{\frac{j-4}{2}}\frac{(2i+1)!!}{(2i)!!} + k - 2j + 4\right] \text{ (by Lemma 4.13 (a))} \\ &= ((j-1)!!)^2\left(j^{\frac{k-2j+2}{2}}\right)\left[4\cdot\frac{j-2}{3} + k - 2j + 4\right] = ((j-1)!!)^2\left(j^{\frac{k-2j+2}{2}}\right)\left(\frac{3k-2j+4}{3}\right).\end{aligned}$$

Repeating this procedure for j odd, if we denote by $O\eta_{n,k}^{(j)}$ the number of all possible

sparse pencils in $\mathscr{DC}_{n,k}$ with all entries in the rectangle R_j, for j odd, then:

$$O\eta_{n,k}^{(j)} = 2\Bigg(\sum_{i=0}^{\frac{j-3}{2}}\left((2i+1)!!\frac{(j-1)!!}{(2i)!!}\right)\left(j^{\frac{k-2j+2}{2}}\right)((j-2)!!) + \sum_{i=\frac{j-1}{2}}^{\frac{k-j-1}{2}}((j-2)!!)^2\left(j^{\frac{k-2j+4}{2}}\right) + \\ + \sum_{i=\frac{k-j+1}{2}}^{\frac{k-2}{2}}((j-2)!!)\left(j^{\frac{k-2j+2}{2}}\right)\left((k-2i-1)!!\frac{(j-1)!!}{(k-2i-2)!!}\right)\Bigg).$$

As above, we can simplify $O\eta_{n,k}^{(j)}$ as:

$$O\eta_{n,k}^{(j)} = 4\left(j^{\frac{k-2j+2}{2}}\right)((j-2)!!)((j-1)!!)\sum_{i=0}^{\frac{j-3}{2}}\frac{(2i+1)!!}{(2i)!!} + 2((j-2)!!)^2\left(j^{\frac{k-2j+4}{2}}\right)\frac{k-2j+2}{2} = \\ = ((j-2)!!)^2\left(j^{\frac{k-2j+2}{2}}\right)\left[4\cdot\frac{(j-1)!!}{(j-2)!!}\sum_{i=0}^{\frac{j-3}{2}}\frac{(2i+1)!!}{(2i)!!} + j(k-2j+2)\right] \text{ (by Lemma 4.13 (b))} \\ = ((j-2)!!)^2\left(j^{\frac{k-2j+2}{2}}\right)\left[4\cdot\frac{j(j-1)}{3} + j(k-2j+2)\right] = \\ = ((j-2)!!)^2\left(j^{\frac{k-2j+2}{2}+1}\right)\left(\frac{3k-2j+2}{3}\right) = (j!!)^2\left(j^{\frac{k-2j}{2}}\right)\left(\frac{3k-2j+2}{3}\right).$$

In summary,

$$E\eta_{n,k}^{(j)} = ((j-1)!!)^2\left(j^{\frac{k-2j+2}{2}}\right)\left(\frac{3k-2j+4}{3}\right), \quad O\eta_{n,k}^{(j)} = (j!!)^2\left(j^{\frac{k-2j}{2}}\right)\left(\frac{3k-2j+2}{3}\right). \tag{14}$$

Now, we consider all possible rectangles R_j, for $1 \le j \le \frac{k}{2}$. We just look at $j = 1, \ldots, \frac{k}{2}$, since for $j = \frac{k}{2}+1, \ldots, k$, the patterns are symmetric (with respect to the anti-diagonal) to the first ones. Adding up, the number of sparse pencils in $\mathscr{DC}_{n,k}$ is:

$$\begin{cases} 2\left(\sum_{j=1}^{\frac{k}{4}}\left(O\eta_{n,k}^{(2j-1)} + E\eta_{n,k}^{(2j)}\right)\right), & \text{if } \frac{k}{2} \text{ is even, and} \\ 2\left(\sum_{j=1}^{\lceil\frac{k}{4}\rceil}O\eta_{n,k}^{(2j-1)} + \sum_{j=1}^{\lfloor\frac{k}{4}\rfloor}E\eta_{n,k}^{(2j)}\right), & \text{if } \frac{k}{2} \text{ is odd.} \end{cases} \tag{15}$$

Using (14) and grouping summands appropriately in (15), we arrive at (12). ■

Remark 4.14 Note that the number of sparse companion pencils in $\mathscr{QC}_{n,k}$ for k even becomes much larger than the one for k odd as k increases. This is due to the fact that the nonzero subdiagonals in the case k odd are determined (they are the ones with indices $0, 2, \ldots, k-1$), but the case k even allows for more flexibility, depending on which is the nonzero subdiagonal containing the block B_ℓ (see Lemma 4.8).

5 Conclusions

In this work, we have first introduced a family of companion pencils for $n \times n$ matrix polynomials of degree k over an arbitrary field, $\mathscr{R}_{n,k}$, which extends the one in [17] for companion matrices of monic scalar polynomials. This family contains all companion pencils in most of the families of companion linearizations introduced so far in the literature, expressed in the monomial basis, and having a small number of nonzero entries. In particular, $\mathscr{R}_{n,k}$ contains both Fiedler and generalized Fiedler pencils, as well as all sparse pencils in the block-Kronecker linearizations presented in [15]. We have provided a "canonical" expression for companion pencils in $\mathscr{R}_{n,k}$, up to block permutation. This expression, which leads to the class $\mathscr{Q}_{n,k}$, is block upper Hessenberg and resembles the one provided in [17] for companion matrices of monic scalar polynomials, We have provided a characterization for a pencil in $\mathscr{Q}_{n,k}$ to be a companion pencil (namely, they are those in the class denoted by $\mathscr{QC}_{n,k}$). Finally, we have obtained the number of different sparse companion pencils in $\mathscr{R}_{n,k}$, up to block permutation. We want to emphasize that there could be other sparse companion pencils for $n \times n$ matrix polynomials of degree k not included in $\mathscr{R}_{n,k}$. Therefore, describing all sparse companion pencils for $n \times n$ matrix polynomials of degree k is still an open field of research.

Acknowledgements We wish to thank two anonymous referees for the careful reading of this paper and for their comments that allowed us to improve the presentation.

This work has been partially supported by the *Ministerio de Economía y Competitividad* of Spain through grants MTM2015-68805-REDT and MTM2015-65798-P.

References

1. Antoniou, E.N., Vologiannidis, S.: A new family of companion forms for polynomial matrices. Electron. J. Linear Algebra **11**, 78–87 (2004)
2. Bueno, M.I., De Terán, F.: Eigenvectors and minimal bases for some families of Fiedler-like linearizations. Linear Multilinear Algebra **62**, 39–62 (2014)
3. Bueno, M.I., Furtado, S.: Palindromic linearizations of a matrix polynomial of odd degree obtained from Fiedler pencils with repetition. Electron. J. Linear Algebra **23**, 562–577 (2012)
4. Bueno, M.I., De Terán, F., Dopico, F.M.: Recovery of eigenvectors and minimal bases of matrix polynomials from generalized Fiedler linearizations. SIAM J. Matrix Anal. Appl. **32**, 463–483 (2011)

5. Bueno, M.I., Curlett, K., Furtado, S.: Structured strong linearizations from Fiedler pencils with repetition I. Linear Algebra Appl. **460**, 51–80 (2014)
6. Bueno, M.I, Dopico, F.M., Furtado, S., Rychnovsky, M.: Large vector spaces of block-symmetric strong linearizations of matrix polynomials. Linear Algebra Appl. **477**, 165–210 (2015)
7. Bueno, M.I, Dopico, F.M., Pérez, J., Saavedra, R., Zykovsi, B.: A unified approach to Fiedler-like pencils via strong block minimal bases pencils. Linear Algebra Appl. **547**, 45–104 (2018)
8. De Terán, F., Dopico, F.M., Mackey, D.S.: Fiedler companion linearizations and the recovery of minimal indices. SIAM J. Matrix Anal. Appl. **31**, 2181–2204 (2010)
9. De Terán, F., Dopico, F.M., Mackey, D.S.: Palindromic companion forms for matrix polynomials of odd degree. J. Comput. Appl. Math. **236**, 1464–1480 (2011)
10. De Terán, F., Dopico, F.M., Mackey, D.S.: Fiedler companion linearizations for rectangular matrix polynomials. Linear Algebra Appl. **437**, 957–991 (2012)
11. De Terán, F., Dopico, F.M., Mackey, D.S.: Spectral Equivalence of matrix polynomials and the Index Sum Theorem. Linear Algebra Appl. **459**, 264–333 (2014)
12. De Terán, F., Dopico, F.M., Pérez, J.: Backward stability of polynomial root-finding using Fiedler companion matrices. IMA J. Numer. Anal. **36**, 133–173 (2016)
13. De Terán, F., Dopico, F.M., Pérez, J.: Eigenvalue condition number and pseudospectra of Fiedler matrices. Calcolo **54**, 319–365 (2017)
14. Del Corso, G., Poloni, F.: Counting Fiedler pencils with repetition. Linear Algebra Appl. **532**, 463–499 (2017)
15. Dopico, F.M., Lawrence, P., Pérez, J., Van Dooren, P.: Block Kronecker linearizations of matrix polynomials and their backward errors. Numer. Math. **140**(2), 373–426 (2018). Available as MIMS Eprint 2016.34. The University of Manchester, UK
16. Eastman, B., Vander Meulen, K.N.: Pentadiagonal companion matrices. Spec. Matrices **4**,13–30 (2016)
17. Eastman, B., Kim, I.-J., Shader, B.L., Vander Meulen, K.N.: Companion matrix patterns. Linear Algebra Appl. **436**, 255–272 (2014)
18. Garnett, C., Shader, B.L., Shader, C.L., van den Driessche, P.: Characterization of a family of generalized companion matrices. Linear Algebra Appl. **498**, 360–365 (2016)
19. Ma, C., Zhan, X.: Extremal sparsity of the companion matrix of a polynomial. Linear Algebra Appl. **438**, 621–625 (2013)
20. Noferini, V., Pérez, J.: Fiedler-comrade and Fiedler–Chebyshev pencils. SIAM J. Matrix Anal. Appl. **37**, 1600–1624 (2016)
21. Noferini, V., Pérez, J.: Chebyshev rootfinding via computing eigenvalues of colleague matrices: when is it stable? Math. Comput. **86**, 1741–1767 (2017)
22. Robol, L., Valdebril, R., Van Dooren, P.: A framework for structured linearizations of matrix polynomials in various bases. SIAM J. Matrix Anal. Appl. **38**, 188–216 (2017)
23. Vologiannidis, S., Antoniou, E.N.: A permuted factors approach for the linearization of polynomial matrices. Math. Control Signals Syst. **22**, 317–342 (2011)

On Computing Eigenvectors of Symmetric Tridiagonal Matrices

Nicola Mastronardi, Harold Taeter, and Paul Van Dooren

Abstract The computation of the eigenvalue decomposition of symmetric matrices is one of the most investigated problems in numerical linear algebra. For a matrix of moderate size, the customary procedure is to reduce it to a symmetric tridiagonal one by means of an orthogonal similarity transformation and then compute the eigendecomposition of the tridiagonal matrix.

Recently, Malyshev and Dhillon have proposed an algorithm for deflating the tridiagonal matrix, once an eigenvalue has been computed. Starting from the aforementioned algorithm, in this manuscript we develop a procedure for computing an eigenvector of a symmetric tridiagonal matrix, once its associate eigenvalue is known.

We illustrate the behavior of the proposed method with a number of numerical examples.

Keywords Tridiagonal matrices · Eigenvalue computation · QR method

The author "Nicola Mastronardi" is a member of the INdAM Research group GNCS.
The scientific responsibility rests with its authors.

N. Mastronardi (✉)
Istituto per le Applicazioni del Calcolo "M. Picone", Consiglio Nazionale delle Ricerche, sede di Bari, Italy
e-mail: n.mastronardi@ba.iac.cnr.it

H. Taeter
Dipartimento di matematica, Università degli Studi di Bari, Bari, Italy
e-mail: harold.taeter@uniba.it

P. Van Dooren
Department of Mathematical Engineering, Catholic University of Louvain, Louvain-la-Neuve, Belgium
e-mail: paul.vandooren@uclouvain.be

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_9

1 Introduction

Computing the eigenvalue decomposition of symmetric matrices is one of the most investigated problems in numerical linear algebra [6, 11]. For a matrix of moderate size, having reduced the symmetric matrix into a symmetric tridiagonal one by means of a similarity orthogonal transformation, the problem reduces to the computation of the eigendecomposition of a tridiagonal matrix.

There are different methods to compute the eigenvalues of symmetric tridiagonal matrices, such as the bisection method [14], the QR method [14] and divide & conquer methods [2, 7]. For computing the eigenvectors one can use inverse iteration [14], the QR method [14] and the multiple relatively robust representations algorithm [5, 13, 15]. The latter algorithm is based on the twisted factorization of the involved tridiagonal matrix to determine the position where the sought eigenvector has a large entry [5, 15, 16].

Once an eigenvalue is computed, a deflation algorithm was proposed in [4] in order to remove it from the tridiagonal matrix and reduce the dimension of the problem by one. Such an algorithm can also be used to compute the eigenvector associated to the computed eigenvalue and it is based on the twisted factorization used in [5, 15].

In this manuscript we consider a modified version of the aforementioned algorithm to compute an eigenvector of a symmetric tridiagonal matrix, supposing the corresponding eigenvalue is known.

Without loss of generality, we consider only the real case. The complex Hermitian one can be handled in the same way.

We illustrate the behavior of the proposed method with some numerical examples. The manuscript is organized as follows. In Sect. 2 the notation used in the manuscript is given. In Sect. 3 the main features of the QR method are described. The proposed algorithm is described in Sect. 4, followed by the section of numerical examples and by the conclusions.

2 Notations and Definitions

Matrices are denoted with upper case letters and their entries with lower case letters, i.e., the element (i, j) of the matrix T is denoted by $t_{i,j}$.

The submatrix of the matrix B made by the rows $i, i+1, i+2, \ldots, i+k$, with $1 \leq i \leq n-k$, $0 \leq k \leq n-i$, and columns $j, j+1, j+2, \ldots, j+l$, with $1 \leq j \leq n-l$, $0 \leq l \leq n-j$, is denoted by $B_{i:i+k,j:j+l}$. If the matrix T is symmetric, the submatrix made by the rows and columns $i, i+1, i+2, \ldots, i+k$, with $1 \leq i \leq n-k$, $0 \leq k \leq n-i$, is simply denoted by $T_{i:i+k}$.

The identity matrix of order n is denoted by I_n or by I if there is no ambiguity.

The matrix $T - \kappa I$, with $\kappa \in \mathbb{R}$, is denoted by $T(\kappa)$.

The principal diagonal of a matrix $B \in \mathbb{R}^{m \times n}$ is denoted by $\mathrm{diag}(B)$.

The machine precision is denoted by ε.

The ith vector of the canonical basis of $\mathbb{R}^n$ is denoted by $\mathbf{e}_i^{(n)}$, or simply by $\mathbf{e}_i$, if there is no ambiguity.

Definition 1 Given $B \in \mathbb{R}^{m\times n}, m \geq n$, let $B = U\Sigma V^T$ be its singular value decomposition, with $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ orthogonal and $\Sigma \in \mathbb{R}^{m\times n}$ diagonal, with $\text{diag}(\Sigma) = \left[\sigma_1,\ \sigma_2,\ \cdots \sigma_n\right]^T$, and $\sigma_i \geq \sigma_{i+1},\ i = 1, \ldots, n-1$.

The columns of B are said ε-linear dependent if $\sigma_n \leq \varepsilon \|B\|_2$.

The columns of B are said *strongly* linear independent if $\sigma_n \gg \varepsilon \|B\|_2 > 0$.

3 Implicit QR Method

Let $T \in \mathbb{R}^{n\times n}$ be the symmetric tridiagonal matrix

$$T = \begin{bmatrix} t_{1,1} & t_{1,2} & & & \\ t_{2,1} & t_{2,2} & \ddots & & \\ & \ddots & \ddots & t_{n-1,n} \\ & & t_{n,n-1} & t_{n,n} \end{bmatrix},$$

with $t_{i,i+1} = t_{i+1,i},\ i = 1, \ldots, n-1$.

Let us suppose that T is irreducible, i.e., $t_{i,i+1} \neq 0,\ i = 1, \ldots, n-1$ and let $T = X\Lambda X^T$ be its eigenvalue decomposition, with $X \in \mathbb{R}^{n\times n}$ orthogonal, and $\Lambda \in \mathbb{R}^{n\times n}$ diagonal, with $\text{diag}(\Lambda) = [\lambda_1, \ldots, \lambda_n]^T$. Since T is irreducible, then $\lambda_i \neq \lambda_j$, with $i \neq j,\ i, j = 1, \ldots, n$.

The Implicit QR (IQR) method is the standard method for computing the eigenvalue decomposition of matrices of moderate size [6]. In particular, `MATLAB` uses the `LAPACK` routine `DSYEV`, based on the QR method, to compute eigenvalues and eigenvectors of a real symmetric matrix [1].

Given a symmetric irreducible tridiagonal matrix $T \in \mathbb{R}^{n\times n}$, and $\kappa \in \mathbb{R}$, one sweep of IQR with shift κ consists of computing the similarity transformation

$$\hat{T}^{(n)} = \hat{G}_{n-1}\hat{G}_{n-2}\cdots\hat{G}_1\hat{T}^{(1)}\hat{G}_1^T\cdots\hat{G}_{n-2}^T\hat{G}_{n-1}^T,$$

where $\hat{T}^{(1)} = T$ and $\hat{G}_i,\ i = 1, \ldots, n-1$, are Givens rotations

$$\hat{G}_i = \begin{bmatrix} I_{i-1} & & & \\ & \hat{c}_i & \hat{s}_i & \\ & -\hat{s}_i & \hat{c}_i & \\ & & & I_{n-i-1} \end{bmatrix}, \quad i = 1, \ldots, n-1.$$

with $\hat{c}_i^2 + \hat{s}_i^2 = 1$. Without loss of generality, we assume that $\hat{c}_i \geq 0$. Hence the matrix $\hat{Q}$ in (1) is uniquely defined.

In particular, $\hat{G}_1$ is the Givens rotation acting on the first two rows of $\hat{T}^{(1)}$, whose coefficients $\hat{c}_1$ and $\hat{s}_1$ are such that

$$\begin{bmatrix} \hat{c}_1 & \hat{s}_1 \\ -\hat{s}_1 & \hat{c}_1 \end{bmatrix} \begin{bmatrix} \hat{t}^{(1)}_{1,1} - \kappa \\ \hat{t}^{(1)}_{2,1} \end{bmatrix} = \begin{bmatrix} \hat{\alpha}_1 \\ 0 \end{bmatrix},$$

with $\hat{\alpha}_1 = \| \begin{bmatrix} \hat{t}^{(1)}_{1,1} - \kappa, \hat{t}^{(1)}_{2,1} \end{bmatrix} \|_2$. The structure of the matrix $\hat{T}^{(2)} = \hat{G}_1 \hat{T}^{(1)} \hat{G}_1^T$ differs from the one of a tridiagonal matrix for an entry different from 0 in position (3, 1) (and, symmetrically, in position (1, 3)), called "*bulge*".

Each of the other Givens rotations $\hat{G}_i$ are applied to move the bulge one position downward along the second subdiagonal/superdiagonal and eventually remove it [11], i.e, the matrix

$$\hat{T}^{(i)} = \hat{G}_{i-1} \hat{G}_{i-2} \cdots \hat{G}_1 \hat{T}^{(1)} \hat{G}_1^T \cdots \hat{G}_{i-2}^T \hat{G}_{i-1}^T,$$

has the bulge in position $(i-1, i+1)$ (and, symmetrically, in position $(i+1, i-1)$), $\hat{T}^{(i+1)} = \hat{G}_i \hat{T}^{(i)} \hat{G}_i^T$ has the bulge in position $(i, i+2)$ (and, symmetrically, in position $(i+2, i)$), and so on. The matrix

$$\hat{Q} = \hat{G}_{n-1} \hat{G}_{n-2} \cdots \hat{G}_1 \tag{1}$$

is orthogonal Hessenberg.

In the sequel, we call the sweep of the IQR method described above a "*forward*" IQR (FIQR) sweep because it starts from the top-left corner of $\hat{T}_1$ and ends in the bottom-right one.

The IQR method can also be implemented in a "*backward*" fashion, i.e., starting from the bottom-right corner of T and ending in the top-left corner [9]. We will refer to one sweep of this procedure as a backward IQR (BIQR) sweep.

Let $\tilde{T}^{(1)} = T$. In a BIQR sweep with shift κ, a sequence of Givens rotations

$$\tilde{G}_i = \begin{bmatrix} I_{n-i-1} & & & \\ & \tilde{c}_i & \tilde{s}_i & \\ & -\tilde{s}_i & \tilde{c}_i & \\ & & & I_{i-1} \end{bmatrix}, \quad i = 1, \ldots, n-1,$$

with $\tilde{c}_i^2 + \tilde{s}_i^2 = 1$, is determined in the following way.

The coefficients $\tilde{c}_1$ and $\tilde{s}_1$ of $\tilde{G}_1$ are computed such that

$$\begin{bmatrix} \tilde{t}^{(1)}_{n,n-1}, \tilde{t}^{(1)}_{n,n} - \kappa \end{bmatrix} \begin{bmatrix} \tilde{c}_1 & \tilde{s}_1 \\ -\tilde{s}_1 & \tilde{c}_1 \end{bmatrix} = \begin{bmatrix} 0, \tilde{\alpha}_n \end{bmatrix},$$

with $\tilde{\alpha}_n = \| \begin{bmatrix} \tilde{t}^{(1)}_{n,n-1}, \tilde{t}^{(1)}_{n,n} - \kappa \end{bmatrix} \|_2$. The matrix $\tilde{T}^{(2)} = \tilde{G}_1^T \tilde{T}^{(1)} \tilde{G}_1$ has a bulge in position $(n, n-2)$ (and, symmetrically, in position $(n-2, n)$).

The Givens rotations $\tilde{G}_i,\ i = 2, \ldots, n-1$, are sequentially applied to $\tilde{T}^{(2)}$ to move the bulge upward along the second subdiagonal and eventually remove it in the matrix $\tilde{T}^{(n)} = \tilde{G}_{n-1}^T \tilde{G}_{n-2}^T \cdots \tilde{G}_1^T \tilde{T}^{(1)} \tilde{G}_1 \cdots \tilde{G}_{n-2} \tilde{G}_{n-1}$.

Let $\tilde{Q} = \tilde{G}_1 \cdots \tilde{G}_{n-2} \tilde{G}_{n-1}$. Without loss of generality, we assume $\tilde{c}_i \geq 0$, which makes the matrix $\tilde{Q}$ uniquely defined.

Let λ be an eigenvalue of T with corresponding eigenvector $\mathbf{x}$. In infinite precision arithmetic, if λ is chosen as shift κ in the FIQR sweep, λ shows up in position (n, n) of $\hat{T}^{(n)}$. Moreover, $\hat{t}_{n-1,n}^{(n)} = \hat{t}_{n,n-1}^{(n)} = 0$, and $\mathbf{x} = \hat{Q}(:, n)$. In particular, since

$$\hat{Q} = \begin{bmatrix} \hat{c}_1 & -\hat{s}_1\hat{c}_2 & \hat{s}_1\hat{s}_2\hat{c}_3 & \ddots & -1^n\hat{c}_{n-1}\prod_{i=1}^{n-2}\hat{s}_i & -1^{n+1}\prod_{i=1}^{n-1}\hat{s}_i \\ \hat{s}_1 & \hat{c}_1\hat{c}_2 & -\hat{c}_1\hat{s}_2\hat{c}_3 & \ddots & -1^{n-1}\hat{c}_1\hat{c}_{n-1}\prod_{i=2}^{n-2}\hat{s}_i & -1^n\hat{c}_1\prod_{i=2}^{n-1}\hat{s}_i \\ & \hat{s}_1 & \hat{c}_1\hat{c}_2 & \ddots & \vdots & \vdots \\ & & \ddots & \ddots & -\hat{c}_{n-3}\hat{s}_{n-2}\hat{c}_{n-1} & \hat{c}_{n-3}\hat{s}_{n-2}\hat{s}_{n-1} \\ & & & \hat{s}_{n-2} & \hat{c}_{n-2}\hat{c}_{n-1} & -\hat{c}_{n-2}\hat{s}_{n-1} \\ & & & & \hat{s}_{n-1} & \hat{c}_{n-1} \end{bmatrix},$$

then

$$\mathbf{x} = \hat{Q}(:, n) = \begin{bmatrix} -1^{n+1}\prod_{i=1}^{n-1}\hat{s}_i \\ -1^n\hat{c}_1\prod_{i=2}^{n-1}\hat{s}_i \\ -1^{n-1}\hat{c}_2\prod_{i=3}^{n-1}\hat{s}_i \\ \vdots \\ \hat{c}_{n-3}\hat{s}_{n-2}\hat{s}_{n-1} \\ -\hat{c}_{n-2}\hat{s}_{n-1} \\ \hat{c}_{n-1} \end{bmatrix}. \tag{2}$$

Analogously, in infinite precision arithmetic, if λ is chosen as shift κ in the BIQR sweep, λ shows up in position $(1, 1)$ of $\tilde{T}^{(n)}$. Moreover, $\tilde{t}_{1,2}^{(n)} = \tilde{t}_{2,1}^{(n)} = 0$, and $\mathbf{x} = \tilde{Q}(1, :)^T$,

$$\mathbf{x} = \begin{bmatrix} \tilde{c}_{n-1} \\ -\tilde{c}_{n-2}\tilde{s}_{n-1} \\ \tilde{c}_{n-3}\tilde{s}_{n-2}\tilde{s}_{n-1} \\ \vdots \\ -1^{n-1}\tilde{c}_2\prod_{i=3}^{n-1}\tilde{s}_i \\ -1^n\tilde{c}_1\prod_{i=2}^{n-1}\tilde{s}_i \\ -1^{n+1}\prod_{i=1}^{n-1}\tilde{s}_i \end{bmatrix}. \tag{3}$$

Therefore, for a given eigenvalue λ, it is suggested in [11] to apply one sweep of either forward or backward IQR with shift λ to compute the corresponding eigenvector with $O(n)$ floating point operations [11].

Unfortunately, forward instability can occur in floating point arithmetic in one forward/backward IQR sweep with shift λ and the last column of $\hat{Q}$ (the first row of $\tilde{Q}$) may be far from the sought eigenvector [12].

In particular, forward instability occurs at step j of one sweep of FIQR if and only if the shift κ is very close to one of the eigenvalues of $\hat{T}^{(j)}_{1:j,1:j}$ and the last entry of the corresponding eigenvector is tiny [12]. As a consequence, the entries $t^{(j)}_{j,j-1}$ and $t^{(j)}_{j,j+1}$ are *"sufficiently"* small[1] [12]. By (2), the last component of the eigenvector is given by $\hat{c}_j$. Hence, forward instability happens if κ is very close to one of the eigenvalues of $\hat{T}_{1:j,1:j}$ and $\hat{c}_j \sim O(\varepsilon)$. This means that the first j columns of $\hat{T}_{1:j,1:j}$ are ε-linear dependent.

The same phenomenon can occur in a BIQR sweep.

To examine in which step of a IQR sweep forward instability can occur, let us consider the following Corollary [8, p.149].

Corollary 1 *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix and let B be a submatrix obtained by deleting r rows from A. Then*

$$\sigma_k(A) \geq \sigma_k(B) \geq \sigma_{k+r}(A), \quad k = 1, \ldots, n,$$

where $\sigma_\ell(A) \equiv 0$, if $\ell > n$.

Let us suppose that $\sigma_{n-1}(T(\lambda)) \gg \varepsilon > \sigma_n(T(\lambda)) = 0$ and $\sigma_{j-1}(T_{1:j,:}(\lambda)) \gg \sigma_j(T_{1:j,:}(\lambda)) \sim O(\varepsilon)$, for a $j \in \{2, \ldots, n\}$. By Corollary 1, all the submatrices $T_{1:j+\ell,:}(\lambda)$ are ε-singular, $\ell = 1, \ldots, n-j$, and

$$\sigma_{j+\ell}(T_{1:j+\ell,:}(\lambda)) \geq \sigma_{j+\ell}(T_{1:j+\ell,1:j+\ell}(\lambda)), \quad \ell = 1, \ldots, n-j,$$

i.e., the submatrices $T_{1:j+\ell,1:j+\ell}(\lambda)$ are ε-singular as well.

On the other hand,

$$\begin{gathered}\sigma_{j-1}(T_{n-j+1:n,:}(\lambda)) \geq \sigma_{n-1}(T(\lambda)) \gg \varepsilon,\\ \sigma_{j-1}(T_{n-j+1:n,:}(\lambda)) \geq \sigma_j(T_{n-j+1:n,n-j+1:n}(\lambda)) \geq \sigma_j(T_{n-j+1:n,:}(\lambda)).\end{gathered}$$

This means that forward instability is not encountered in the first $n-j-1$ steps of FIQR and in the first j steps of BIQR.

In the following example it is shown how the sequences $\{\hat{c}_j\}_{j=1}^{n-1}$ and $\{\tilde{c}_j\}_{j=1}^{n-1}$, computed in floating point arithmetic, differ from those computed in infinite precision arithmetic.

[1] If one of the indices i, j in $t_{i,j}$ is either 0 or n, we set $t_{i,j} \equiv 0$.

Example 1 Let $T \in \mathbb{R}^{n\times n}$, $n = 100$, be a symmetric irreducible tridiagonal matrix with its entries generated by the `MATLAB` function `randn`.[2]

Let $T = X^{(M)}\Lambda^{(M)}X^{(M)^T}$ be the eigenvalue decomposition of T computed by using the `MATLAB` function `eig`, with $\Lambda^{(M)} = \mathrm{diag}(\lambda_1^{(M)}\lambda_2^{(M)}, \ldots, \lambda_n^{(M)})$, with $\lambda_i^{(M)} \geq \lambda_{i+1}^{(M)}$ $i = 1, \ldots, n-1$.

We report the behaviour of one sweep of F/B IQR with shift $\lambda_{19}^{(M)}$, although a similar behavior can be observed if we choose almost any $\lambda_i^{(M)}$, $i = 1, \ldots, n$, as a shift.

Let $(\bar{\lambda}, \bar{\mathbf{x}})$ be the eigenpair computed by a few steps of inverse iteration with initial guess $(\lambda_{19}^{(M)}, X^{(M)}(:, 19))$. In this way, $\bar{\mathbf{x}}$ is computed with higher accuracy with respect to $X^{(M)}(:, 19))$.

Let $\{\check{c}_i\}_{i=1}^{n-1}$ and $\{\bar{c}_i\}_{i=1}^{n-1}$ be the sequence of the cosines of the Givens matrices $\{\check{G}_i\}_{i=1}^{n-1}$ and $\{\bar{G}_i\}_{i=1}^{n-1}$, determined in order to transform $\bar{\mathbf{x}}$ to $\mathbf{e}_n$ and $\mathbf{e}_1$, respectively, i.e.,

$$\check{G}_i = \begin{bmatrix} I_{n-i-1} & & & \\ & \check{c}_i & \check{s}_i & \\ & -\check{s}_i & \check{c}_i & \\ & & & I_{i-1} \end{bmatrix}, \quad \text{such that } \check{G}_{n-1}\check{G}_{n-2}\cdots\check{G}_1\bar{\mathbf{x}} = \mathbf{e}_n,$$

$$\bar{G}_i = \begin{bmatrix} I_{i-1} & & & \\ & \bar{c}_i & \bar{s}_i & \\ & -\bar{s}_i & \bar{c}_i & \\ & & & I_{n-i-1} \end{bmatrix}, \quad \text{such that } \bar{G}_1\cdots\bar{G}_{n-2}\bar{G}_{n-1}\bar{\mathbf{x}} = \mathbf{e}_1.$$

Without loss of generality, we assume $\check{c}_i \geq 0$ and $\bar{c}_i \geq 0$, $i = 1, \ldots, n-1$.

Since $\bar{\mathbf{x}}$ is computed with high accuracy, the sequences $\{\check{c}_i\}_{i=1}^{n-1}$ and $\{\bar{c}_i\}_{i=1}^{n-1}$, are computed with high accuracy, too [10].

In infinite precision arithmetic, the sequences $\{\check{c}_i\}_{i=1}^{n-1}$ and $\{\hat{c}_i\}_{i=1}^{n-1}$ should be the same, while in floating point arithmetic the sequence $\{\hat{c}_i\}_{i=1}^{n-1}$ can depart from the sequence $\{\check{c}_i\}_{i=1}^{n-1}$ due to the forward instability [10]. The same holds for the sequences $\{\bar{c}_i\}_{i=1}^{n-1}$ and $\{\tilde{c}_i\}_{i=1}^{n-1}$.

The sequences $\{\hat{c}_i\}_{i=1}^{n-1}$, $\{\check{c}_i\}_{i=1}^{n-1}$, $\{|\hat{t}_{i-1,i}^{(i)}| + |\hat{t}_{i,i+1}^{(i)}|\}_{i=1}^{n-1}$, $\{\check{\sigma}_i\}_{i=1}^{n-1}$, and $\{\hat{\sigma}_i\}_{i=1}^{n-1}$, with $\check{\sigma}_i = \min(\mathrm{svd}(T_{:,i:n}(\lambda_{19}^{(M)})))$ and $\hat{\sigma}_i = \min(\mathrm{svd}(T_{i:n,i:n}(\lambda_{19}^{(M)})))$, denoted respectively by "$*$", "$+$", "$\circ$", "$\diamond$" and "∇", are displayed in Fig. 1 on a logarithmic scale.

We can observe that the first and the third sequence have a similar behaviour. The same can be said for the second and the fifth sequence. Moreover, the two sequences of cosines $\{\hat{c}_i\}_{i=1}^{n-1}$ and $\{\check{c}_i\}_{i=1}^{n-1}$ are similar until forward instability occurs, i.e., until $\hat{c}_i$ and $|\hat{t}_{i-1,i}^{(i)}| + |\hat{t}_{i,i+1}^{(i)}|$ are both greater than $O(\sqrt{\varepsilon})$.

[2]The matrix T can be downloaded at users.ba.cnr.it/iac/irmanm21/TRID_SYM.

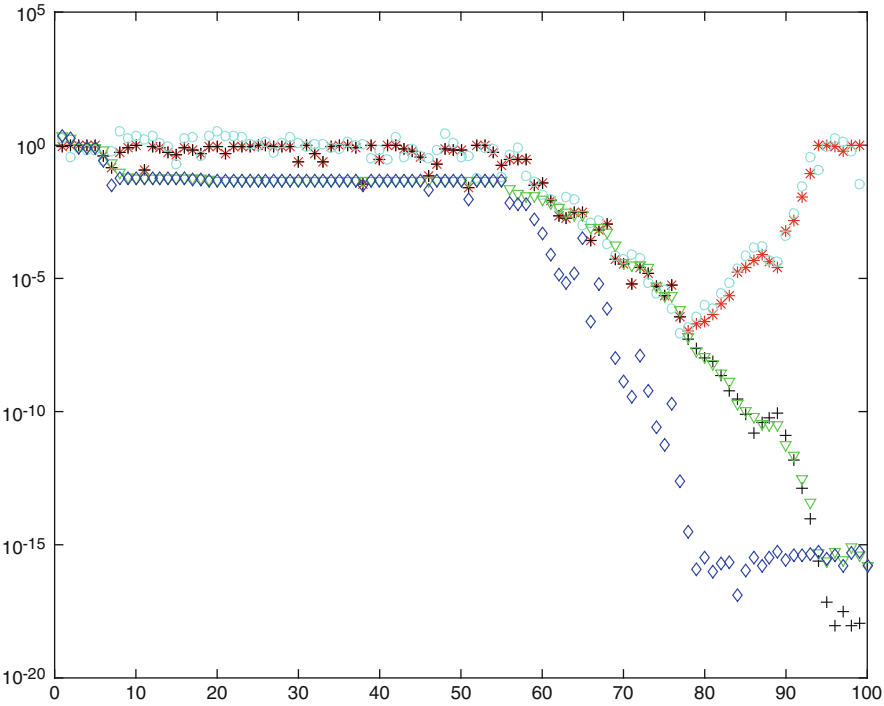


Fig. 1 Sequences $\{\hat{c}_i\}_{i=1}^{n-1}$, $\{\check{c}_i\}_{i=1}^{n-1}$, $\{|\hat{t}_{i-1,i}^{(i)}| + |\hat{t}_{i,i+1}^{(i)}|\}_{i=1}^{n-1}$, $\{\check{\sigma}_i\}_{i=1}^{n-1}$, and $\{\hat{\sigma}_i\}_{i=1}^{n-1}$, denoted respectively by "asterisk", "plus", "circle", "diamond" and "triangledown", related to $T(\lambda_{19}^{(M)})$, with T the matrix of Example 1 and $\lambda_{19}^{(M)}$ the 19th ordered eigenvalue computed by `eig` of `MATLAB`

The sequences $\{\tilde{c}_i\}_{i=1}^{n-1}$, $\{\bar{c}_i\}_{i=1}^{n-1}$, $\{|\tilde{t}_{n-i-1,n-i-2}^{(i)}| + |\tilde{t}_{n-i,n-i-1}^{(i)}|\}_{i=1}^{n-1}$, $\{\check{\sigma}_i\}_{i=1}^{n-1}$, and $\{\hat{\sigma}_i\}_{i=1}^{n-1}$, with $\check{\sigma}_i = \min(\mathrm{svd}(T_{:,1:i}(\lambda_{19}^{(M)})))$, $\hat{\sigma}_i = \min(\mathrm{svd}(T_{1:i,1:i}(\lambda_{19}^{(M)})))$, denoted respectively by "$*$", "$+$", "$\circ$", "$\diamond$" and "∇", are displayed in Fig. 2 in logarithmic scale.

Also in this case, the first and the third sequence have a similar behaviour and the same can be said for the second and the fifth sequence. Moreover, the two sequences of cosines $\{\tilde{c}_i\}_{i=1}^{n-1}$ and $\{\bar{c}_i\}_{i=1}^{n-1}$ are similar until forward instability occurs, i.e., until $\tilde{c}_i$ and $|\tilde{t}_{n-i-1,n-i-2}^{(i)}| + |\tilde{t}_{n-i,n-i-1}^{(i)}|$ are both greater than $O(\sqrt{\varepsilon})$.

Summarizing, forward instability occurs if the smallest singular value $\sigma_j^{(j)}$ of $T_{1:j,1:j}(\lambda)$ is close to the machine precision ε, for a certain $j \in \{1, \ldots, n\}$. As a consequence, the elements of the last column of $\hat{Q}$ of index greater than j begin to depart from the elements of the eigenvector $\bar{\mathbf{x}}$. Moreover, $\hat{t}_{j,j-1}^{(j)} \approx \hat{t}_{j+1,j}^{(j)} \approx O(\sqrt{\varepsilon})$, where $\hat{t}_{i,k}^{(j)}$ is the (i,k) entry of the matrix obtained after having applied j Givens rotations in the forward IQR sweep with shift $\bar{\lambda}$ to $T_{1:n}(\bar{\lambda})$ [12]. The same holds to one sweep of BIQR, i.e., the first row of the upper Hessenberg matrix $\tilde{Q}$ is accurately

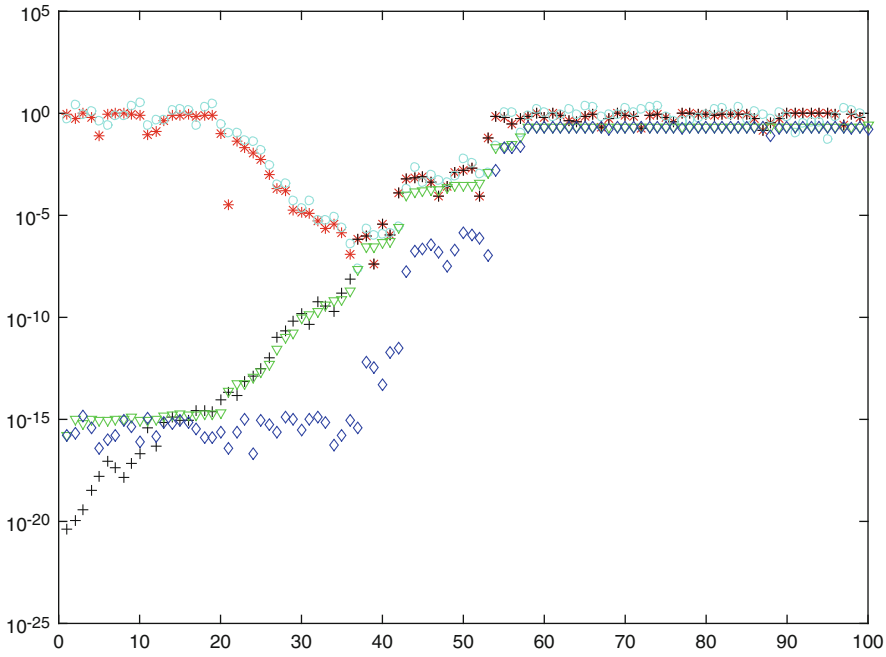


Fig. 2 Sequences $\{\check{c}_i\}_{i=1}^{n-1}$, $\{\bar{c}_i\}_{i=1}^{n-1}$, $\{|\tilde{t}^{(i)}_{n-i-1,n-i-2}| + |\tilde{t}^{(i)}_{n-i,n-i-1}|\}_{i=1}^{n-1}$, $\{\bar{\sigma}_i\}_{i=1}^{n-1}$, and $\{\tilde{\sigma}_i\}_{i=1}^{n-1}$, denoted respectively by "asterisk", "plus", "circle", "diamond" and "triangledown", related to $T(\lambda_{19}^{(M)})$, with T the matrix of Example 1 and $\lambda_{19}^{(M)}$ the 19th ordered eigenvalue computed by `eig` of `MATLAB`

computed as far as the smallest singular value of $T_{j:n}(\lambda)$ is large enough, for a certain $j \in \{1, \ldots, n\}$.

Hence, the main issue is to determine the index j.

In the next section we consider the problem of determining the index j such that the computed eigenvector will be obtained by $\hat{Q}(1 : j, n)$ and $\tilde{Q}(1, j + 1 : n)^T$, i.e., gluing together the first j entries of $\hat{Q}$ and the last $n - j$ entries of the first row of $\tilde{Q}$.

4 Computation of the Eigenvector

In this section we describe a technique to determine the index j used for constructing the sought eigenvector by fetching the first j entries of the last column of $\hat{Q}$ and the last $n - j$ entries of the first row of $\tilde{Q}$.

If $\sigma_{n-1}(T(\lambda)) \gg \sigma_n(T(\lambda))$ and forward instability occurs at step j of one sweep of FIQR with shift λ, the sequence $\{\hat{c}_i\}_{i=1}^{n}$ begins to depart from the sequence $\{\check{c}_i\}_{i=1}^{n}$ around the index j. Analogously, the sequence $\{\tilde{c}_i\}_{i=1}^{n}$ begins to depart from

the sequence $\{\bar{c}_i\}_{i=1}^{n}$ around the index $n-j$. Therefore, the sought index j can be computed in the following way.

The sequence $\{\hat{c}_i\}_{i=1}^{n-1}$ generated by one FIQR sweep, is computed until $\hat{c}_{\hat{\jmath}} < tol_1$ and $|\hat{t}^{(\hat{\jmath})}_{\hat{\jmath}-1,\hat{\jmath}}| + |\hat{t}^{(\hat{\jmath})}_{\hat{\jmath},\hat{\jmath}+1}| < tol_2$, with tol_1 and tol_2 fixed tolerances and $1 \le \hat{\jmath} \le n-1$.

The sequence $\{\tilde{c}_i\}_{i=1}^{n-1}$, generated by one BIQR sweep, is thus computed until $\tilde{c}_{\tilde{\jmath}} < tol_1$ and $|\tilde{t}^{(\tilde{\jmath})}_{\tilde{\jmath}-1,\tilde{\jmath}}| + |\tilde{t}^{(\tilde{\jmath})}_{\tilde{\jmath},\tilde{\jmath}+1}| < tol_2$.

Hence, the sought index $\bar{\jmath}$ is computed as the index $\bar{\jmath}$ such that

$$\hat{c}_{\bar{\jmath}} + \tilde{c}_{\bar{\jmath}} \ge \hat{c}_i + \tilde{c}_i, \quad i, \bar{\jmath} \in [\tilde{\jmath}, \hat{\jmath}], i \ne \bar{\jmath},$$

i.e., the index $\bar{\jmath}$ is chosen so that the columns of $T_{:,1:\bar{\jmath}}$ and $T_{:,\bar{\jmath}:n}$ are strongly linear independent.

The last column of $\hat{Q}$ in (2) depends on all the Givens coefficients $\hat{c}_i$ and $\hat{s}_i$, $i = 1, \ldots, n-1$, while the first row of $\tilde{Q}$ in (3) depends on all the Givens coefficients $\tilde{c}_i$ and $\tilde{s}_i$, $i = 1, \ldots, n-1$.

Therefore, at the first sight one can say that both the last column of $\hat{Q}$ and the first row of $\tilde{Q}$ must be computed in order to construct the sought eigenvector even though the "splitting" index j is already determined.

In the sequel we show that the sought approximation of the eigenvector can be computed relying only on the knowledge of $\hat{c}_i$ and $\hat{s}_i$, $i = 1, \ldots, j-1$, and $\tilde{c}_i$ and $\tilde{s}_i$, $i = 1, \ldots, n-j+1$. In fact, once the index j is determined, we observe that the "good" part of the vector (2) can be written as

$$\hat{\mathbf{x}}_{1:j} = \begin{bmatrix} -1^{n+1} \prod_{i=1}^{n-1} \hat{s}_i \\ -1^{n} \hat{c}_1 \prod_{i=2}^{n-1} \hat{s}_i \\ -1^{n-1} \hat{c}_2 \prod_{i=3}^{n-1} \hat{s}_i \\ \vdots \\ -1^{j+1} \hat{c}_{j-2} \prod_{i=j-1}^{n-1} \hat{s}_i \\ -1^{j} \hat{c}_{j-1} \prod_{i=j}^{n-1} \hat{s}_i \\ -1^{j-1} \hat{c}_{j} \prod_{i=j+1}^{n-1} \hat{s}_i \end{bmatrix} = \gamma^{(u)} \hat{\mathbf{x}}^{(u)},$$

while the "good" part of the vector (2) can be written as

$$\tilde{\mathbf{x}}_{n-j:n} = \begin{bmatrix} -1^{n-j-1} \tilde{c}_{n-j} \prod_{i=n-j+1}^{n-1} \tilde{s}_i \\ -1^{n-j} \tilde{c}_{n-j-1} \prod_{i=n-j}^{n-1} \tilde{s}_i \\ -1^{n-j+1} \tilde{c}_{n-j-2} \prod_{i=n-j-1}^{n-1} \tilde{s}_i \\ \vdots \\ -1^{n-1} \tilde{c}_2 \prod_{i=3}^{n-1} \tilde{s}_i \\ -1^{n} \tilde{c}_1 \prod_{i=2}^{n-1} \tilde{s}_i \\ -1^{n+1} \prod_{i=1}^{n-1} \tilde{s}_i \end{bmatrix} = \gamma^{(b)} \tilde{\mathbf{x}}^{(b)},$$

where $\gamma^{(u)} = \prod_{i=j+1}^{n-1} \hat{s}_i$, $\gamma^{(b)} = \prod_{i=n-j+1}^{n-1} \tilde{s}_i$,

$$\hat{\mathbf{x}}^{(u)} = \begin{bmatrix} -1^{n+1} \prod_{i=1}^{j} \hat{s}_i \\ -1^{n} \hat{c}_1 \prod_{i=2}^{j} \hat{s}_i \\ -1^{n-1} \hat{c}_2 \prod_{i=3}^{j} \hat{s}_i \\ \vdots \\ -1^{j+1} \hat{c}_{j-2} \prod_{i=j-1}^{j} \hat{s}_i \\ -1^{j} \hat{c}_{j-1} \hat{s}_j \\ -1^{j-1} \hat{c}_j \end{bmatrix}, \quad \tilde{\mathbf{x}}^{(b)} = \begin{bmatrix} -1^{n-j-1} \tilde{c}_{n-j} \\ -1^{n-j} \tilde{c}_{n-j-1} \tilde{s}_{n-j} \\ -1^{n-j+1} \tilde{c}_{n-j-2} \prod_{i=n-j-1}^{n-j} \tilde{s}_i \\ \vdots \\ -1^{n-1} \tilde{c}_2 \prod_{i=3}^{n-j} \tilde{s}_i \\ -1^{n} \tilde{c}_1 \prod_{i=2}^{n-j} \tilde{s}_i \\ -1^{n+1} \prod_{i=1}^{n-j} \tilde{s}_i \end{bmatrix}.$$

Hence, we first normalize both vectors in this way,

$$\check{\mathbf{x}}_{1:j} = \frac{\hat{\mathbf{x}}^{(u)}_{1:j}}{\hat{\mathbf{x}}^{(u)}_{j}}, \quad \check{\mathbf{x}}_{j+1:n} = \frac{\tilde{\mathbf{x}}^{(b)}_{2:n-j+1}}{\tilde{\mathbf{x}}^{(b)}_{1}},$$

i.e., we divide the first vector by its last component and the second one by its first one in order to have 1 as the j-th entry of the first vector and as the first entry of the second one, and finally we normalize $\check{\mathbf{x}}$ such that $\|\check{\mathbf{x}}\|_2 = 1$.

The corresponding `MATLAB` code to compute the eigenvector associated to a given eigenvalue of a symmetric tridiagonal matrix is freely available and can be downloaded at users.ba.cnr.it/iac/irmanm21/TRID_SYM.

5 Deflation

Once the eigenvector $\check{\mathbf{x}}$ has been computed, we can apply to it a sequence of $n-1$ Givens rotations G_i in order to transform it either to $\pm\mathbf{e}_1^{(n)}$ or to $\pm\mathbf{e}_n^{(n)}$, where $\pm\mathbf{e}_i^{(n)}$, $i = 1, \ldots, n$, is the ith vector of the canonical basis of $\mathbb{R}^n$.

The same Givens rotations G_i are applied to the matrix T obtaining

$$\check{T} = G_{n-1} G_{n-2} \cdots G_1 T G_1^T \cdots G_{n-2}^T G_{n-1}^T. \tag{4}$$

If the eigenvector $\check{\mathbf{x}}$ is computed in an accurate way and satisfies particular properties, it has been shown in [9, 10] that $\check{T}$ in (4) is still tridiagonal with the entry $(2, 1)$ equal to zero if the Givens rotations are applied in a backward fashion or the entry $(n, n-1)$ is equal to zero if the Givens rotations are applied in a forward manner. In the first case the last row and column can be dropped and the other eigenvalues to be computed are the eigenvalues of $\check{T}_{1:n-1}$. In the other case, the first row and column are removed and the other eigenvalues are the eigenvalues of $\check{T}_{2:n}$.

6 Numerical Examples

All the numerical experiments of this section are performed in MATLAB Ver. 2014b, with machine precision $\varepsilon \sim 2.22 \times 10^{-16}$. We have compared the results obtained computing the eigenvector matrix with the following techniques: eig of MATLAB, MR^3 and the proposed method, denoted by MTV.[3] For the second and the third method, the eigenvalues are computed by the bisection method [14].

For all the experiments, the tolerances tol_1 and tol_2 were chosen equal to $n\sqrt{\varepsilon}$, with n the order of the matrix.

Example 2 In this example we consider symmetric tridiagonal matrices $T_n \in \mathbb{R}^{n\times n}$, $n = 128, 256, 512, 1024$ whose elements are generated by the MATLAB function randn.

The latter matrices can be downloaded at users.ba.cnr.it/iac/irmanm21/TRID$_$SYM.

In Table 1 the orthogonality of the computed eigenvectors with the considered three methods are displayed. In column 5, the average lengths of the computed intervals in which to search the index j are reported.

In Table 2 the accuracy of the residuals of the computed eigenvectors with the considered three methods are displayed.

We can conclude that the eigenvectors obtained with the proposed procedure are computed in an accurate way.

Example 3 In this example $T_n \in \mathbb{R}^{n\times n}$, $n = 128, 256, 512, 1024$ are the Jacobi matrices associated to the Chebyshev polynomials of first kind [3], whose eigenvalues are

$$\cos\left(\frac{i\pi}{n+1}\right), \quad i = 1\ldots, n.$$

In Table 3 the orthogonality of the computed eigenvectors with the considered three methods are displayed. We do not report the average lengths of the computed intervals in which to search the index j in this case, since there is no premature deflation for such matrices.

In Table 4 the accuracy of the residuals of the computed eigenvectors with the considered three methods are displayed.

We can conclude that the eigenvectors obtained with the proposed procedure are computed in an accurate way.

[3] We have used a MATLAB implementation of the MR^3 algorithm written by Petschow [13].

Table 1 Accuracy of the orthogonality of the computed eigenvectors computed by `eig` of `MATLAB` (second column), by `MR`^{3} (third column) and by the proposed method (fourth column)

$\max_i \frac{\|X^T\mathbf{x}_i - \mathbf{e}_i^{(n)}\|_2}{n\varepsilon}$				
n	`eig`	`MR`^{3}	`MTV`	$\frac{\sum_{i=1}^{n}(\tilde{j}-\hat{j}+1)}{n}$
128	1.14×10^{-1}	2.01×10^{1}	7.05×10^{1}	26
256	5.72×10^{-2}	1.00×10^{1}	3.52×10^{1}	24
512	2.86×10^{-2}	5.02×10^{0}	1.97×10^{1}	24
1024	1.43×10^{-2}	3.40×10^{0}	3.83×10^{1}	24

Average lengths of the computed intervals in which the index j is sought (fifth column)

Table 2 Accuracy of the residuals of the eigenvectors computed by `eig` of `MATLAB` (second column), by `MR`^{3} (third column) and by the proposed method (fourth column)

$\max_i \frac{\|T\bar{x}_i - \lambda_i\bar{x}_i\|_2}{n\varepsilon\|T\|_2}$			
n	`eig`	`MR`^{3}	`MTV`
128	4.96×10^{-2}	5.33×10^{0}	1.87×10^{1}
256	1.73×10^{-2}	1.00×10^{1}	3.52×10^{1}
512	1.06×10^{-2}	5.02×10^{0}	1.76×10^{1}
1024	6.72×10^{-3}	5.94×10^{-1}	5.96×10^{0}

Table 3 Accuracy of the orthogonality of the computed eigenvectors computed by `eig` of `MATLAB` (second column), by `MR`^{3} (third column) and by the proposed method (fourth column)

$\max_i \frac{\|X^T\mathbf{x}_i - \mathbf{e}_i^{(n)}\|_2}{n\varepsilon}$			
n	`eig`	`MR`^{3}	`MTV`
128	2.36×10^{-1}	2.901×10^{0}	2.16×10^{2}
256	1.18×10^{-1}	2.69×10^{0}	6.35×10^{2}
512	5.92×10^{-2}	7.26×10^{1}	1.03×10^{1}
1024	1.43×10^{-2}	2.99×10^{0}	3.83×10^{1}

Table 4 Accuracy of the residuals of the eigenvectors computed by `eig` of `MATLAB` (second column), by `MR`^{3} (third column) and by the proposed method (fourth column)

$\max_i \frac{\|T\bar{x}_i - \lambda_i\bar{x}_i\|_2}{n\varepsilon\|T\|_2}$			
n	`eig`	`MR`^{3}	`MTV`
128	1.67×10^{-1}	2.05×10^{0}	1.52×10^{2}
256	8.37×10^{-2}	1.45×10^{1}	1.08×10^{1}
512	4.18×10^{-2}	7.26×10^{0}	1.05×10^{2}
1024	6.72×10^{-3}	9.48×10^{-1}	5.21×10^{0}

7 Conclusions

Recently, Malyshev and Dhillon have proposed an algorithm for deflating the tridiagonal matrix, once an eigenvalue has been computed. Starting from the above mentioned algorithm, a method for computing the eigenvectors of a symmetric tridiagonal matrix T has been proposed. It requires the computation of an index j which determines the premature deflation in the implicit QR method. The index j is computed considering the two sequences of cosines generated by a sweep of forward and backward QR method with shift the computed eigenvalue. The sought eigenvector is obtained form the first j Givens coefficients generated by the forward implicit QR method and from the last $n - j$ Givens coefficients generated by the backward implicit QR method.

The overall complexity for computing an eigenvector depends linearly on the size of the matrix.

The numerical tests show the reliability of the proposed technique.

Acknowledgements The authors wish to thank the anonymous reviewers for their constructive remarks that helped improving the proposed algorithm and the presentation of the results.

The authors would like to thank Paolo Bientinesi and Matthias Petschow for providing their `MATLAB` implementation of the MR3 algorithm, written by Matthias Petschow.

The work of the author "Nicola Mastronardi" is partly supported by GNCS–INdAM and by CNR under the Short Term Mobility Program. The work of the author "Harold Taeter" is supported by INdAM-DP-COFUND-2015, grant number: 713485.

References

1. Anderson, E., Bai, Z., Bischof, C., Blackford, L., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK Users' Guide, 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia (1999)
2. Cuppen, J.J.M.: A divide and conquer method for the symmetric tridiagonal eigenproblem. Numer. Math. **36**, 177–195 (1981)
3. Davis, P., Rabinowitz, P.: Methods of Numerical Integration, 2nd edn. Academic Press, Cambridge (1984)
4. Dhillon, I., Malyshev, A.: Inner deflation for symmetric tridiagonal matrices. Linear Algebra Appl. **358**, 139–144 (2003)
5. Dhillon, I., Parlett, B.: Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices. Linear Algebra Appl. **387**, 1–28 (2004)
6. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore (2013)
7. Gu, M., Eisenstat, S.: A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem. SIAM J. Matrix Anal. Appl. **16**(1), 172–191 (1995)
8. Horn, R., Johnson, C.: Topics in Matrix Analysis. Cambridge University Press, New York (1991)
9. Mastronardi, N., Van Dooren, P.: Computing the Jordan structure of an eigenvalue. SIAM J. Matrix Anal. Appl. **38**, 949–966 (2017)
10. Mastronardi, N., Van Dooren, P.: The QR–steps with perfect shifts. SIAM J. Matrix Anal. Appl. **39**, 1591–1615 (2018)

11. Parlett, B.N.: The Symmetric Eigenvalue Problem. Society for Industrial and Applied Mathematics, Philadelplhia (1997)
12. Parlett, B.N., Le, J.: Forward instability of tridiagonal QR. SIAM J. Matrix Anal. Appl. **14**(1), 279–316 (1993)
13. Petschow, M., Quintana-Ortí, E., Bientinesi, P.: Improved accuracy and parallelism for MRRR–based eigensolvers–a mixed precision approach. SIAM J. Sci. Comput. **36**(2), C240–C263 (2014)
14. Wilkinson, J., Bauer, F., Reinsch, C.: Linear Algebra. Handbook for Automatic Computation. Springer, Berlin (2013)
15. Willems, P.R., Lang, B.: Twisted factorizations and qd–type transformations for the MR3 algorithm–new representations and analysis. SIAM J. Matrix Anal. Appl. **33**(2), 523–553 (2012)
16. Willems, P.R., Lang, B.: A framework for the MR3 algorithm: theory and implementation. SIAM J. Sci. Stat. Comput. **35**(2), A740–A766 (2013)

A Krylov Subspace Method for the Approximation of Bivariate Matrix Functions

Daniel Kressner

Abstract Bivariate matrix functions provide a unified framework for various tasks in numerical linear algebra, including the solution of linear matrix equations and the application of the Fréchet derivative. In this work, we propose a novel tensorized Krylov subspace method for approximating such bivariate matrix functions and analyze its convergence. While this method is already known for some instances, our analysis appears to result in new convergence estimates and insights for all but one instance, Sylvester matrix equations.

Keywords Matrix function · Krylov subspace method · Bivariate polynomial · Fréchet derivative · Sylvester equation

1 Introduction

Given a univariate function $f(z)$ defined in the neighborhood of the spectrum $\Lambda(A)$ of a matrix $A \in \mathbb{C}^{n\times n}$, the numerical computation of the *matrix function* $f(A) \in \mathbb{C}^{n\times n}$ has been studied intensively during the last decades; see [10, 15, 18] for surveys. The extension of the notion of matrix functions to bivariate or, more generally, multivariate functions f has a long history as well, notably in the context of holomorphic functional calculus and operator theory; see [23, Sec. 3] for a detailed discussion and references. In the numerical analysis literature, however, bivariate matrix functions have been discussed mostly for special cases only.

Given two matrices $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$ and a bivariate function $f(x, y)$ defined in a neighborhood of $\Lambda(A) \times \Lambda(B)$, the *bivariate matrix function* $f\{A, B\}$ is a linear operator on $\mathbb{C}^{m\times n}$. We will recall the formal definition of $f\{A, B\}$ in

D. Kressner (✉)
Institute of Mathematics, EPF Lausanne, Lausanne, Switzerland
e-mail: daniel.kressner@epfl.ch

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_10

Sect. 2 below. Linear matrix equations and Fréchet derivatives constitute the most widely known instances of bivariate matrix functions:

1. For $f(x, y) = 1/(x + y)$ the matrix $X = f\{A, B\}(C)$ is the solution of the Sylvester matrix equation

$$AX + XB^T = C, \qquad C \in \mathbb{C}^{m\times n}, \tag{1}$$

where B^T denotes the complex transpose of B and C is often of low rank. When B equals $\bar{A}$ (denoting the complex conjugate of A) and C is Hermitian, (1) is called Lyapunov matrix equation. Such matrix equations play an important role in control, e.g., for computing the Gramians in balanced truncation model reduction of linear time-invariant control systems. They also arise from structured discretizations of partial differential equations. See [5, 35] for references.
2. There are several variants of (1) that fit the framework of bivariate matrix functions. The solution of the Stein equation $AXB^T - X = C$ is given by $X = f\{A, B\}(C)$ with $f(x, y) = 1/(1 - xy)$. More generally, for $f(x, y) = 1/p(x, y)$, with a bivariate polynomial $p(x, y) = \sum_{i=0}^{k}\sum_{j=0}^{\ell} p_{ij}x^i y^j$, the matrix $X = f\{A, B\}(C)$ is the solution of the matrix equation

$$\sum_{i=0}^{k}\sum_{j=0}^{\ell} p_{ij}A^i X(B^T)^j = C,$$

which has been considered, e.g., in [9, 27].

Time-limited and frequency-limited balanced truncation model reduction [6, 12] give rise to matrix equations that involve matrix exponentials and logarithms. For example, the reachability Gramian corresponding to a time interval $0 \le t_s < t_e \le \infty$ satisfies an equation of the form

$$AX + XA^* = -\exp(t_s A)C\exp(t_s A^*) + \exp(t_e A)C\exp(t_e A^*), \tag{2}$$

where $A^* = \bar{A}^T$ denotes the Hermitian transpose and, once again, C is often of low rank. The solution of (2) can be expressed as $X = f\{A, \bar{A}\}(C)$ with

$$f(x, y) = \frac{\exp(t_e(x + y)) - \exp(t_s(x + y))}{x + y}. \tag{3}$$

In the analogous situation for frequency-limited balanced truncation, the corresponding function takes the form

$$f(x, y) = -\frac{g(x) + g(y)}{x + y}, \quad g(z) = \mathsf{Re}\Big(\frac{\mathrm{i}}{\pi}\ln\Big(\frac{z + \mathrm{i}\omega_2}{z + \mathrm{i}\omega_1}\Big)\Big), \quad 0 \le \omega_1 < \omega_2 \le \infty, \tag{4}$$

where Re denotes the real part of a complex number.

3. Given a (univariate) matrix function $f(A)$ and the finite difference quotient

$$f^{[1]}(x, y) := f[x, y] = \begin{cases} \frac{f(x)-f(y)}{x-y}, & \text{for } x \neq y, \\ f'(x), & \text{for } x = y, \end{cases} \tag{5}$$

the matrix $X = f^{[1]}\{A, A^T\}(C)$ is the Fréchet derivative of f at A in direction C; see [23, Thm. 5.1].

In this work, we consider the numerical approximation of $f\{A, B\}(C)$ for large matrices A and B. As the size of the involved matrices grows, it becomes necessary to impose additional structure before attempting this task. We assume that matrix-vector multiplications with A and B are feasible because, for example, A and B are sparse. Moreover, C is assumed to have low rank. The latter is a common assumption in numerical solvers for large-scale matrix equations (1), but we also refer to [14, 16, 25, 30] for works that consider other types of data-sparsity for C.

Given a rank-one matrix $C = cd^T$, the method proposed in this paper makes use of the two Krylov subspaces generated by the matrices A, B with starting vectors c, d. An approximation to $f\{A, B\}(C)$ is then selected from the tensor product of these two subspaces. Our method already exists for several of the instances mentioned above. For $f(x, y) = 1/(x + y)$, it corresponds to a widely known Krylov subspace method for Lyapunov and Sylvester equations [20, 32]. For the functions (3) and (4), our method corresponds to the Krylov subspace methods presented in [26] and [6], respectively. For the Fréchet derivative, the algorithm presented in this paper has been proposed independently in [21]. For Lyapunov and Sylvester equations, the convergence of these methods has been analyzed in detail; see, e.g., [2, 36]. For all other instances, the general theory presented in this work appear to result in previously unknown convergence estimates.

We note in passing that the algorithm proposed in this paper shares similarities with a recently proposed Krylov subspace method for performing low-rank updates of matrix functions [4].

2 Preliminaries

We first recall the definition of bivariate matrix functions and their basic properties from [23]. Let $\Pi_{k,\ell}$ denote the set of all bivariate polynomials of degree at most (k, ℓ), that is, for $p \in \Pi_{k,\ell}$ we have that $p(x, y)$ has degree at most k in x and degree at most ℓ in y. Every such polynomial takes the form

$$p(x, y) = \sum_{i=0}^{k} \sum_{j=0}^{\ell} p_{ij} x^i y^j, \qquad p_{ij} \in \mathbb{C}.$$

The bivariate matrix function corresponding to p and evaluated at $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$ is defined as

$$p\{A, B\} : \mathbb{C}^{m\times n} \to \mathbb{C}^{m\times n}, \qquad p\{A, B\}(C) := \sum_{i=0}^{k}\sum_{j=0}^{\ell} p_{ij} A^i C (B^T)^j. \tag{6}$$

This definition extends via Hermite interpolation to general functions f that are sufficiently often differentiable at the eigenvalues of A and B; see [23, Def. 2.3] for details. A more compact and direct definition is possible when f is analytic.

Assumption 1 *There exist domains $\Omega_A, \Omega_B \subset \mathbb{C}$ containing the eigenvalues of A and B, respectively, such that $f_y(x) := f(x, y)$ is analytic in Ω_A for every $y \in \Omega_B$ and $f_x(y) := f(x, y)$ is analytic in Ω_B for every $x \in \Omega_A$.*

By Hartog's theorem [22], Assumption 1 implies that f is analytic in $\Omega_A \times \Omega_B$. Moreover, we have

$$f\{A, B\}(C) = -\frac{1}{4\pi^2}\oint_{\Gamma_A}\oint_{\Gamma_B} f(x, y)(xI - A)^{-1} C (yI - B^T)^{-1}\,\mathrm{d}y\,\mathrm{d}x, \tag{7}$$

where $\Gamma_A \subset \Omega_A$ and $\Gamma_B \subset \Omega_B$ are closed contours enclosing the eigenvalues of A and B, respectively.

Diagonalizing one of the two matrices A, B relates bivariate matrix functions to (univariate) matrix functions of the other matrix. A similar result has already been presented in [23, Sec. 6]; we include its proof for the sake of completeness.

Lemma 1 *Suppose that Assumption 1 holds and that there is an invertible matrix Q such that $Q^{-1}BQ = \mathrm{diag}(\mu_1, \ldots, \mu_n)$. Then*

$$f\{A, B\}(C) = \begin{bmatrix} f_{\mu_1}(A)\tilde{c}_1 & f_{\mu_2}(A)\tilde{c}_2 & \cdots & f_{\mu_n}(A)\tilde{c}_n \end{bmatrix} Q^T,$$

with $CQ^{-T} =: \tilde{C} = \begin{bmatrix}\tilde{c}_1 & \cdots & \tilde{c}_n\end{bmatrix}$ and $f_\mu := f(x, \mu)$.

Proof Setting $\Lambda_B = \mathrm{diag}(\mu_1, \ldots, \mu_n)$, we obtain from (7) that

$$\begin{aligned} f\{A, B\}(C) &= -\frac{1}{4\pi^2}\oint_{\Gamma_A} (xI - A)^{-1}\tilde{C}\Big[\oint_{\Gamma_B} f(x, y)(yI - \Lambda_B)^{-1}\,\mathrm{d}y\Big] Q^T\,\mathrm{d}x \\ &= \frac{1}{2\pi\mathrm{i}}\oint_{\Gamma_A} (xI - A)^{-1}\tilde{C}\cdot \mathrm{diag}(f_{\mu_1}(x), \ldots, f_{\mu_n}(x)) Q^T\,\mathrm{d}x \\ &= \frac{1}{2\pi\mathrm{i}}\oint_{\Gamma_A} \begin{bmatrix} f_{\mu_1}(x)(xI - A)^{-1}\tilde{c}_1 & \cdots & f_{\mu_n}(x)(xI - A)^{-1}\tilde{c}_n \end{bmatrix} Q^T\,\mathrm{d}x, \end{aligned}$$

which concludes the proof, using the contour integral representation of $f_\mu(A)$. □

For the case $f(x, y) = 1/(x + y)$, the result of Lemma 1 is related to algorithms for Sylvester equation with large m but relatively small n; see [34].

If both A, B are diagonalizable, that is, additionally to the assumption of Lemma 1 there exists an invertible matrix P such that $P^{-1}AP = \operatorname{diag}(\lambda_1, \ldots, \lambda_m)$ then the result of the lemma implies

$$f\{A, B\}(C) = P \left(\begin{bmatrix} f(\lambda_1, \mu_1) & \cdots & f(\lambda_1, \mu_n) \\ \vdots & & \vdots \\ f(\lambda_m, \mu_1) & \cdots & f(\lambda_m, \mu_n) \end{bmatrix} \circ C \right) Q^T, \qquad \tilde{C} := P^{-1} C Q^{-T},$$

where $\circ$ denotes the elementwise (or Hadamard) product.

3 Algorithm

For the sake of simplifying the presentation, we assume that C has rank 1 and can thus be written as $C = cd^T$ for nonzero vectors $c, d \in \mathbb{C}^n$. We comment on the extension to (small) ranks larger than 1 below.

Our method proceeds by constructing orthonormal bases for the Krylov subspaces

$$\mathscr{K}_k(A, b) = \operatorname{span}\{c, Ac, \ldots, A^{k-1}c\}, \qquad \mathscr{K}_\ell(B, d) = \operatorname{span}\{d, Bd, \ldots, B^{\ell-1}d\},$$

When $k \leq m$ and $\ell \leq n$, these subspaces are generically of dimension k and ℓ, which will be assumed in the following. The Arnoldi method [38] applied to $\mathscr{K}_k(A, b)$, $\mathscr{K}_\ell(B, d)$ not only produces orthonormal bases $U_k \in \mathbb{C}^{m\times k}$, $V_\ell \in \mathbb{C}^{n\times \ell}$ but also yields Arnoldi decompositions

$$AU_k = U_k G_k + g_{k+1,k} u_{k+1} e_k^T, \tag{8}$$

$$BV_\ell = V_\ell H_\ell + h_{\ell+1,\ell} v_{\ell+1} e_\ell^T, \tag{9}$$

where $G_k = U_k^* A U_k$ and $H_\ell = V_\ell^* B V_\ell$ are upper Hessenberg matrices, e_k and e_ℓ denote the kth and ℓth unit vectors of suitable length, $g_{k+1,k}$ and $h_{\ell+1,\ell}$ are complex scalars. If $k < m$ and $\ell < n$ then $[U_k, u_{k+1}]$ and $[V_\ell, u_{\ell+1}]$ form orthonormal bases of $\mathscr{K}_{k+1}(A, b)$ and $\mathscr{K}_{\ell+1}(B, d)$, respectively.

We search for an approximation to $f\{A, B\}(C)$ in $\mathscr{K}_k(A, b) \times \mathscr{K}_\ell(B, d)$. Every such approximation takes the form $U_k X_{k,\ell} V_\ell^T$ with some matrix $X_{k,\ell} \in \mathbb{C}^{k\times \ell}$. For reasons that become clear in Sect. 4 below, a suitable (but possibly not the only) choice for this matrix is obtained by evaluating the compressed function:

$$X_{k,\ell} = f\{U_k^* A U_k, V_k^* B V_k\}(U_k^* C \overline{V}_\ell) = f\{G_k, H_\ell\}(\tilde{c}\tilde{d}^T),$$

with $\tilde{c} = U_k^* c$, $\tilde{d} = V_\ell^* d$.

Algorithm 1 Arnoldi method for approximating $f\{A, B\}(C)$ with $C = cd^T$

1: Perform k steps of the Arnoldi method to compute an orthonormal basis U_k of $\mathscr{K}_k(A, c)$ and $G_k = U_k^* A U_k$, $\tilde{c} = U_k^* c$.
2: Perform ℓ steps of the Arnoldi method to compute an orthonormal basis V_ℓ of $\mathscr{K}_\ell(B, d)$ and $H_\ell = V_\ell^* B V_\ell$, $\tilde{d} = V_k^* d$.
3: Compute bivariate matrix function $X_{k,\ell} = f\{G_k, H_\ell\}(\tilde{c}\tilde{d}^T)$.
4: Return $U_k X_{k,\ell} V_\ell^T$.

The described procedure is summarized in Algorithm 1. We conclude this section with several remarks:

1. For the compressed function in Line 3, one requires that f is defined on $\Lambda(G_k) \times \Lambda(H_\ell)$. Considering the numerical ranges

$$\mathscr{W}(A) = \left\{w^* A w : w \in \mathbb{C}^m, \ \|w\|_2 = 1\right\}, \quad \mathscr{W}(B) = \left\{w^* B w : w \in \mathbb{C}^m, \ \|w\|_2 = 1\right\},$$

the following assumption guarantees that this requirement is met; it is also needed in the convergence analysis of Sect. 4.

Assumption 2 *Assumption 1 is satisfied with domains Ω_A, Ω_B satisfying $\mathscr{W}(A) \subset \Omega_A$ and $\mathscr{W}(B) \subset \Omega_B$.*

Because of $\Lambda(G_k) \subset \mathscr{W}(G_k) \subset \mathscr{W}(A)$ and $\Lambda(H_k) \subset \mathscr{W}(H_k) \subset \mathscr{W}(B)$, Assumption 2 implies that $f\{G_k, H_\ell\}$ is well defined.

General-purpose approaches to evaluating the small and dense bivariate matrix function $f\{G_k, H_\ell\}(\tilde{c}\tilde{d}^T)$ in Line 3 are discussed in [23, Sec. 6]. However, let us stress that it is generally advisable to use an approach that is tailored to the function f at hand. For example, for $f(x, y) = 1/(x + y)$ this amounts to solving a small linear matrix equation, for which the Bartels-Stewart algorithm [1] should be used. For the finite difference quotient (5), a suitable method is discussed in Sect. 5 below.

2. As in the case of univariate functions, there is no reliable stopping criterion for general f that would allow to choose k, ℓ such that Algorithm 1 is guaranteed to return an approximation with a prescribed accuracy. In the spirit of existing heuristic criteria, we propose to use the approximation

$$\|f\{A, B\}(cd^T) - U_k X_{k,\ell} V_\ell^T\|_F \approx \|U_{k+h} X_{k+h,\ell+h} V_{\ell+h}^T - U_k X_{k,\ell} V_\ell^T\|_F := e_{k,\ell,h}$$

for some small integer h, say $h = 2$. As already explained in, e.g., [4, Sec. 2.3], the quantity $e_{k,\ell,h}$ is inexpensive to check because

$$e_{k,\ell,h} = \left\| U_{k+h} \left(X_{k+h,\ell+h} - \begin{bmatrix} X_{k,\ell} & 0 \\ 0 & 0 \end{bmatrix} \right) V_{\ell+h}^T \right\|_F = \left\| X_{k+h,\ell+h} - \begin{bmatrix} X_{k,\ell} & 0 \\ 0 & 0 \end{bmatrix} \right\|_F.$$

If $e_{k,\ell,h}$ is smaller than a user-specified tolerance, the output of Algorithm 1 is accepted. Otherwise, k and ℓ are increased, the orthonormal bases U_k, V_ℓ are extended and Step 3 is repeated. It may be desirable to increase k and ℓ separately. For example, one could increase k if

$$\left\| X_{k+h,\ell} - \begin{bmatrix} X_{k,\ell} \\ 0 \end{bmatrix} \right\|_F \geq \left\| X_{k,\ell+h} - \begin{bmatrix} X_{k,\ell} & 0 \end{bmatrix} \right\|_F$$

and increase ℓ otherwise.

Again, we emphasize that better stopping criteria may exist for specific choices of f. This is particularly true for linear matrix equations; see [31] and the references therein.

3. Algorithm 1 extends to matrices C of rank $r > 1$ by replacing the Arnoldi method in Steps 1 and 2 by a block Arnoldi method, by a global Arnoldi method, or by splitting C into r rank-1 terms; see [11] for a comparison of these approaches in a related setting.

4 Exactness Properties and Convergence Analysis

In this section, we analyze the convergence of Algorithm 1 following a strategy commonly used for matrix functions; see, in particular, [4]. First, we establish that Algorithm 1 is exact (that is, it returns $f\{A, B\}(cd^T)$) for polynomials of bounded degree. This then allows us to relate its error for general functions to a bivariate polynomial approximation problem on the numerical ranges.

Lemma 2 *Algorithm 1 is exact if* $f \in \Pi_{(k-1,\ell-1)}$.

Proof The following well-known exactness property of the Arnoldi method (see, e.g., [33]) follows by induction from (8)–(9):

$$A^i c = U_k (G_k)^i U_k^* c, \quad i = 0, \ldots, k-1, \qquad B^j d = V_\ell (H_\ell)^j V_\ell^* d, \quad j = 0, \ldots, \ell-1.$$

By writing $f(x, y) = \sum_{i=0}^{k-1} \sum_{j=0}^{\ell-1} p_{ij} x^i y^j$ and using (6), this gives

$$\begin{aligned} f\{A, B\}(cd^T) &= \sum_{i=0}^{k-1} \sum_{j=0}^{\ell-1} A^i c (B^j d)^T = U_k \Big(\sum_{i=0}^{k-1} \sum_{j=0}^{\ell-1} G_k^i U_k^* cd^T \, \overline{V}_\ell (H_\ell^T)^j \Big) V_\ell^T \\ &= U_k \cdot f\{G_k, H_\ell\}\big(U_k^* cd^T \, \overline{V}_\ell\big) \cdot V_\ell^T, \end{aligned}$$

which corresponds to what is returned by Algorithm 1. □

To treat general functions, we will need to estimate the norm of $f\{A, B\}$ induced by the Frobenius norm on $\mathbb{C}^{m\times n}$:

$$\|f\{A, B\}\| := \max\left\{\|f\{A, B\}(C)\|_F : C \in \mathbb{C}^{m\times n},\ \|C\|_F = 1\right\}.$$

For a (univariate) matrix function $f(A)$, the seminal result by Crouzeix and Palencia [8] states that $\|f(A)\|_2 \leq (1+\sqrt{2})\max_{x\in\mathscr{W}(A)}|f(x)|$. Theorem 1.1 in [13] appears to be the only result in the literature that aims at establishing norm bounds for general bivariate functions. This result provides an upper bound in terms of Henrici's departure from normality for A and B [17] as well as the maximal absolute value of f and its derivatives on convex hulls of $\Lambda(A)$, $\Lambda(B)$. The following lemma provides an upper bound in terms of the maximal absolute value of f on the numerical ranges, which is better suited for our purposes.

Lemma 3 *Suppose that Assumption 2 holds and let $\mathbb{E}_A, \mathbb{E}_B$ be compact connected sets such that $\mathscr{W}(A) \subset \mathbb{E}_A \subset \Omega_A$ and $\mathscr{W}(B) \subset \mathbb{E}_B \subset \Omega_B$. Let $\operatorname{len}(\partial\mathbb{E}_A)$ denote the length of the boundary curve $\partial\mathbb{E}_A$ of $\mathbb{E}_A$, let $d_A(\cdot)$ denote the distance between a subset of $\mathbb{C}$ and $\mathscr{W}(A)$, and define analogous quantities for B. Then*

$$\|f\{A, B\}\| \leq M \cdot \max_{x\in\mathbb{E}_A, y\in\mathbb{E}_B} |f(x, y)|,$$

where

(a) $M = 1$ if both A and B are normal;
(b) $M = 1+\sqrt{2}$ if A or B are normal;
(c) $M = \frac{1+\sqrt{2}}{2\pi}\min\left\{\frac{\operatorname{len}(\partial\mathbb{E}_A)}{d_A(\partial\mathbb{E}_A)}, \frac{\operatorname{len}(\partial\mathbb{E}_B)}{d_B(\partial\mathbb{E}_B)}\right\}$ otherwise, under the additional assumption that $d_A(\partial\mathbb{E}_A) > 0$ or $d_B(\partial\mathbb{E}_B) > 0$.

Proof (a) and (b) Assume that B is normal. The result of Lemma 1, with Q chosen unitary, implies

$$\begin{aligned}\|f\{A, B\}(C)\|_F^2 &= \sum_{j=1}^{m} \|f_{\mu_j}(A)\tilde{c}_j\|_2^2 \leq \sum_{j=1}^{m} \|f_{\mu_j}(A)\|_2^2\|\tilde{c}_j\|_2^2 \\ &= M^2\sum_{j=1}^{m}\max_{x\in\mathbb{E}_A}|f_{\mu_j}(x)|^2\cdot\|\tilde{c}_j\|_2^2 \leq M^2\max_{x\in\mathbb{E}_A, y\in\mathbb{E}_B}|f(x, y)|^2\cdot\|C\|_F^2,\end{aligned}$$

with $M = 1$ if A is also normal and $M = 1+\sqrt{2}$ otherwise [8]. The proof is analogous when B is normal and A is not.

(c) Starting from the representation (7), we obtain

$$\begin{aligned}f\{A, B\}(C) &= -\frac{1}{4\pi^2}\oint_{\partial\mathbb{E}_B}\Big[\oint_{\partial\mathbb{E}_A} f(x, y)(xI - A)^{-1}\mathrm{d}x\Big]C(yI - B^T)^{-1}\,\mathrm{d}y \\ &= \frac{1}{2\pi\mathrm{i}}\oint_{\partial\mathbb{E}_B} f_y(A)C(yI - B^T)^{-1}\,\mathrm{d}y\end{aligned}$$

and, in turn,

$$\|f\{A,B\}(C)\|_F \le \frac{1}{2\pi}\max_{y\in\mathbb{E}_B}\|f_y(A)\|_2\|C\|_F\cdot\oint_{\partial\mathbb{E}_B}\left\|(yI-B^T)^{-1}\right\|_2\,\mathrm{d}y$$

Combined with $\|(yI-B^T)^{-1}\|_2 \le 1/d_B(y)$, this shows $\|f\{A,B\}\| \le \frac{1+\sqrt{2}}{2\pi}\frac{\mathrm{len}(\partial\mathbb{E}_B)}{d_B(\partial\mathbb{E}_B)}$. Analogously, one establishes the same inequality with B replaced by A. □

Remark 1 The result of Lemma 3 can be strengthened in the special case that $f(x,y) = g(x+y)$ for a univariate function g. This class of functions covers the matrix equations discussed in the introduction and also features prominently in [7]. Using that $W(I\otimes A+B\otimes I) = W(A)+W(B)$ (see, e.g., the proof of [37, Corollary 3.2]), we obtain

$$\begin{aligned}\|f\{A,B\}\| &= \|g(I\otimes A+B\otimes I)\|_2 \le (1+\sqrt{2})\|g\|_{W(I\otimes A+B\otimes I)}\\ &= (1+\sqrt{2})\|g\|_{W(A)+W(B)} = (1+\sqrt{2})\max_{x\in W(A),y\in W(B)}|f(x,y)|.\end{aligned}$$

It remains an open and interesting problem to study whether a similar bound holds for a general bivariate function f.

Theorem 1 *Let $\mathbb{E}_A$, $\mathbb{E}_B$, and M be defined as in Lemma 3 and suppose that the assumptions of the lemma hold. Then the output of Algorithm 1 satisfies the error bound*

$$\|f\{A,B\}(cd^T) - U_k X_{k,\ell} V_\ell^T\|_F \le 2M\|c\|_2\|d\|_2\cdot\inf_{p\in\Pi_{k-1,\ell-1}}\max_{x\in\mathbb{E}_A,y\in\mathbb{E}_B}|f(x,y)-p(x,y)|.$$

Proof Let $p\in\Pi_{k-1,\ell-1}$. By Lemma 2, we have

$$p\{A,B\}(cd^T) = U_k\cdot p\{G_k,H_\ell\}\big(\tilde{c}\tilde{d}^T\big)\cdot V_\ell^T,\quad \tilde{c} = U_k^*c,\quad \tilde{d} = V_\ell^*c.$$

Hence,

$$\begin{aligned}&f\{A,B\}(cd^T) - U_kX_{k,\ell}V_\ell^T\\ &= f\{A,B\}(cd^T) - p\{A,B\}(cd^T) - U_k\Big(f\{G_k,H_\ell\}\big(\tilde{c}\tilde{d}^T\big) - p\{G_k,H_\ell\}\big(\tilde{c}\tilde{d}^T\big)\Big)V_\ell^T\\ &= e\{A,B\}(cd^T) - U_k\cdot e\{G_k,H_\ell\}\big(\tilde{c}\tilde{d}^T\big)\cdot V_\ell^T \qquad (10)\end{aligned}$$

with $e = f - p$. Applying Lemma 3 and using that the numerical ranges of G_k and H_k are contained in A and B, respectively, we have

$$\max\{\|e\{A, B\}\|_F, \|e\{G_k, H_\ell\}\|_F\} \leq M \cdot \max_{x \in \mathbb{E}_A, y \in \mathbb{E}_B} |e(x, y)|$$

Inserted into (10), this gives

$$\|f\{A, B\}(cd^T) - U_k X_{k,\ell} V_\ell^T\|_F \leq 2M\|c\|_2\|d\|_2 \cdot \max_{x \in \mathbb{E}_A, y \in \mathbb{E}_B} |e(x, y)|,$$

Because p was chosen arbitrary, the result of the theorem follows. □

Combining Lemma 3 with existing results on polynomial multivariate approximation yields concrete convergence estimates. For example, let us consider the case of Hermitian matrices A and B. By a suitable reparametrization, we may assume without loss of generality that $\mathscr{W}(A) = \mathscr{W}(B) = [-1, 1]$. By Assumption 2, there is $\rho > 1$ such that f is analytic on $E_\rho \times E_\rho$, with the Bernstein ellipse $E_\rho = \{z \in \mathbb{C} : |z - 1| + |z + 1| \leq \rho + \rho^{-1}\}$. Then for any $\tilde{\rho} \in (1, \rho)$ it holds that

$$\inf_{p \in \Pi_{k-1,k-1}} \max_{x,y \in [-1,1]} |f(x, y) - p(x, y)| = \mathscr{O}(\tilde{\rho}^{-k}), \quad k \to \infty, \tag{11}$$

see, e.g., [40]. Hence, Algorithm 1 converges linearly as $\ell = k \to \infty$ with a rate arbitrarily close to ρ.

For $f(x, y) = 1/(\alpha + x + y)$, a specification of (11) can be found in [24, Lemma A.1], resulting in a convergence bound for Sylvester equation that matches the asymptotics of [36]. This is also an example for a function of the form $f(x, y) = g(x + y)$. By choosing an approximating polynomial of the same form and using Remark 1, the convergence estimate of Theorem 1 simplifies for any such function f to

$$\begin{aligned} &\|f\{A, B\}(cd^T) - U_k X_{k,\ell} V_\ell^T\|_F \\ &\leq 2(1 + \sqrt{2})\|c\|_2\|d\|_2 \cdot \min_{p \in \Pi_{k-1}} \max_{z \in W(A)+W(B)} |g(z) - p(z)|, \end{aligned} \tag{12}$$

where Π_{k-1} is the set of all (univariate) polynomials of degree at most $k - 1$.

We now use (12) to analyze the Krylov subspace method for the time-limited Gramian (2) for a symmetric negative definite matrix A with eigenvalues contained in the interval $[-\beta, -\alpha]$, $0 < \alpha < \beta < \infty$, and a rank-one matrix $C = cc^T$. By combining (3) and (12), convergence estimates can be obtained by studying the polynomial approximation of $g(z) = z^{-1}(\exp(t_e z) - \exp(t_s z))$ on the interval $[-\beta, -\alpha]$. For $t_e = \infty$, g always has a singularity at $z = 0$. In turn, the asymptotic linear convergence rate ρ predicted by polynomial approximation is independent of $t_s \geq 0$. In other words, for $t_e = \infty$ the convergence behavior for time-limited Gramians ($t_s > 0$) and Lyapunov equations ($t_s = 0$) are expected to be similar. for $t_e < \infty$, the situation is dramatically different: g is an entire function, yielding

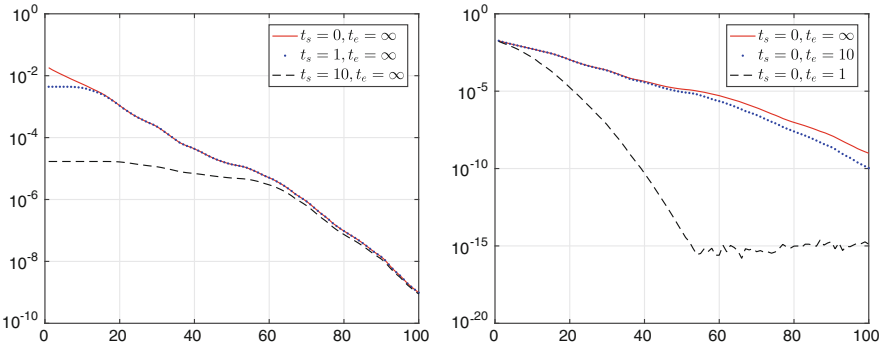


Fig. 1 Convergence of Algorithm 1 applied to the time-limited Gramians from Example 1 for different choices of t_s, t_e

superlinear convergence. For $t_s = 0$, $g(z) = z^{-1}(\exp(t_e z) - 1) = t_e \varphi(t_e z)$ and Lemma 5 in the appendix can be applied to obtain quantitative convergence estimates.

Example 1 To illustrate the convergence of Algorithm 1 for approximating time-limited Gramians, we consider a 500×500 diagonal matrix A with eigenvalues uniformly distributed in $[-100, -0.1]$ and a random vector c of norm 1. Figure 1 reports the error $\|X - \tilde{X}_k\|_2$ (vs. k) of the approximation $\tilde{X}_k$ returned by Algorithm 1 with $\ell = k$. The left plot displays the effect of varying t_s while keeping $t_e = \infty$ fixed. While there is a pronounced difference initially, probably due to the different norms of X, the convergence eventually settles at the same curve. The right plot displays the effect of choosing t_e finite, clearly exhibiting superlinear convergence for $t_e = 1$.

5 Application to Fréchet Derivatives

Given a univariate function f analytic in a neighborhood of the eigenvalues of A, the Fréchet derivative of f at A is a linear map $Df\{A\} : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ uniquely defined by the property $f(A + E) = f(A) + Df\{A\}(E) + \mathcal{O}(\|E\|_2^2)$. In [23, Thm 5.1] it was shown that $Df\{A\} = f^{[1]}\{A, A^T\}$ for the function $f^{[1]}$ defined in (5). In turn, this enables us to use Algorithm 1 for approximating the application of $Df\{A\}$ to rank-one or, more generally, to low-rank matrices. This may be, for example, of interest when approximating gradients in the solution of optimization problems that involve matrix functions; see [39] for an example.

When applying Algorithm 1 to $f^{[1]}\{A, A^T\}$ with $\ell = k$, the reduced problem $f^{[1]}\{G_k, H_k\}$ does, in general, not satisfy $H_k = G_k^T$ and can therefore not be related to a Fréchet derivative of f (unless A is Hermitian and d is a scalar multiple of $\overline{c}$).

The following lemma shows that a well-known formula for the Fréchet derivative (see, e.g., [29, Thm. 2.1]) carries over to this situation.

Lemma 4 *Let f be analytic on a domain Ω containing the eigenvalues of $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$. Then*

$$f\left(\begin{bmatrix} A & C \\ 0 & B \end{bmatrix}\right) = \begin{bmatrix} f(A) & f^{[1]}\{A, B^T\}(C) \\ 0 & f(B) \end{bmatrix}.$$

Proof The assumption of the lemma implies that Assumption 1 is satisfied for $f^{[1]}\{A, B^T\}$ with domains Ω_A, Ω_B satisfying $\overline{\Omega_A \cup \Omega_B} \subset \Omega$. Let $\Gamma \subset \Omega$ be a closed contour enclosing Ω_A and Ω_B. Combining the contour integral representation (7) with

$$f^{[1]}(x, y) = \frac{1}{2\pi \mathrm{i}} \oint_\Gamma \frac{f(z)}{(z-x)(z-y)}\,\mathrm{d}z, \qquad \forall x \in \Omega_A, y \in \Omega_B,$$

gives

$$\begin{aligned} f^{[1]}\{A, B^T\}(C) &= -\frac{1}{8\pi^3 \mathrm{i}} \oint_{\Gamma_A} \oint_{\Gamma_B} \Big[\oint_\Gamma \frac{f(z)}{(z-x)(z-y)}\,\mathrm{d}z \Big] (xI - A)^{-1} C (yI - B)^{-1}\,\mathrm{d}y\,\mathrm{d}x \\ &= -\frac{1}{8\pi^3 \mathrm{i}} \oint_\Gamma f(z) \Big[\oint_{\Gamma_A} \frac{(xI - A)^{-1}}{z-x}\mathrm{d}x \Big] C \Big[\oint_{\Gamma_B} \frac{(yI - B)^{-1}}{z-y}\mathrm{d}y \Big]\,\mathrm{d}z \\ &= \frac{1}{2\pi \mathrm{i}} \oint_\Gamma f(z)(zI - A)^{-1} C (zI - B)^{-1}\,\mathrm{d}z. \end{aligned}$$

On the other hand, we have

$$\begin{aligned} f\left(\begin{bmatrix} A & C \\ 0 & B \end{bmatrix}\right) &= \frac{1}{2\pi \mathrm{i}} \oint_\Gamma f(z) \begin{bmatrix} zI - A & -C \\ 0 & zI - B \end{bmatrix}^{-1} \mathrm{d}z \\ &= \begin{bmatrix} f(A) & \frac{1}{2\pi \mathrm{i}} \oint_\Gamma f(z)(zI - A)^{-1} C (zI - B)^{-1}\,\mathrm{d}z \\ 0 & f(B) \end{bmatrix}, \end{aligned}$$

which completes the proof. □

When applying Algorithm 1 to $f^{[1]}$, we can now use Lemma 4 to address the reduced problem with a standard method for evaluating small and dense matrix functions. This yields Algorithm 2.

The following convergence result is a consequence of Theorem 1; the particular structure of $f^{[1]}$ allows us to reduce the bivariate to a univariate polynomial approximation problem.

Algorithm 2 Arnoldi method for approximating $Df\{A\}(cd^T)$

1: Perform k steps of the Arnoldi method to compute an orthonormal basis U_k of $\mathscr{K}_k(A, c)$ and $G_k = U_k^* A U_k, \tilde{c} = U_k^* c$.
2: Perform k steps of the Arnoldi method to compute an orthonormal basis V_k of $\mathscr{K}_k(A^T, d)$ and $H_k = V_k^* A^T V_k, \tilde{d} = V_k^* d$.
3: Compute $F = f\left(\begin{bmatrix} G_k & \tilde{c}\tilde{d}^T \\ 0 & H_k^T \end{bmatrix}\right)$ and set $X_k = F(1:k, k+1:2k)$.
4: Return $U_k X_k V_k^T$.

Corollary 1 *Let f be analytic on a domain Ω_A containing $\mathscr{W}(A)$ and let $\mathbb{E}_A$ be a compact convex set such that $\mathscr{W}(A) \subset \mathbb{E}_A \subset \Omega_A$. Then the output of Algorithm 2 satisfies the error bound*

$$\|Df\{A\}(cd^T) - U_k X_k V_k^T\|_F \leq 2M\|c\|_2\|d\|_2 \min_{p\in\Pi_{k-1}} \max_{x\in\mathbb{E}_A} |f'(x) - p(x)|,$$

where $M = 1$ if A is normal and $M = \frac{1+\sqrt{2}}{2\pi}\frac{\mathrm{len}(\partial\mathbb{E}_A)}{d_A(\partial\mathbb{E}_A)}$ otherwise.

Proof The conditions of the corollary imply that the conditions of Theorem 1 are satisfied for $f^{[1]}\{A, A^T\}$, which in turn yields

$$\|f^{[1]}\{A, A^T\}(cd^T) - U_k X_k V_k^T\|_F \leq 2M\|c\|_2\|d\|_2 \cdot \inf_{p\in\Pi_{k-1,k-1}} \max_{x,y\in\mathbb{E}_A} |f^{[1]}(x, y) - p(x, y)|.$$

For arbitrary $q \in \Pi_k$, we let $\tilde{p}(x, y) := q^{[1]}(x, y) \in \Pi_{k-1,k-1}$ and set $e := f - q$. By the mean value theorem and convexity of $\mathbb{E}_A$, for every $x, y \in \mathbb{E}_A$ with $x \neq y$ there is $\xi \in \mathbb{E}_A$ such that

$$e'(\xi) = \frac{e(x) - e(y)}{x - y} = f^{[1]}(x, y) - \tilde{p}(x, y).$$

Hence,

$$\max_{x,y\in\mathbb{E}_A} |f^{[1]}(x, y) - \tilde{p}(x, y)| \leq \max_{\xi\in\mathbb{E}_A} |e'(\xi)| = \max_{\xi\in\mathbb{E}_A} |f'(\xi) - q'(\xi)|.$$

Setting $p = q' \in \Pi_{k-1}$ completes the proof. □

Corollary 1 indicates that the convergence of Algorithm 2 is similar to the convergence of the standard Arnoldi method for approximating $f'(A)c$ and $f'(A^T)d$. Moreover, Corollary 1 allows us to directly apply existing polynomial approximation results derived for studying the convergence of the latter method, such as the ones from [3, 19].

Example 2 We consider the matrix A and the vector c from Example 1 and measure the error $\|Df\{A\}(cc^T) - F_k\|_2$ of the approximation $F_k = U_k X_k U_k^T$ returned

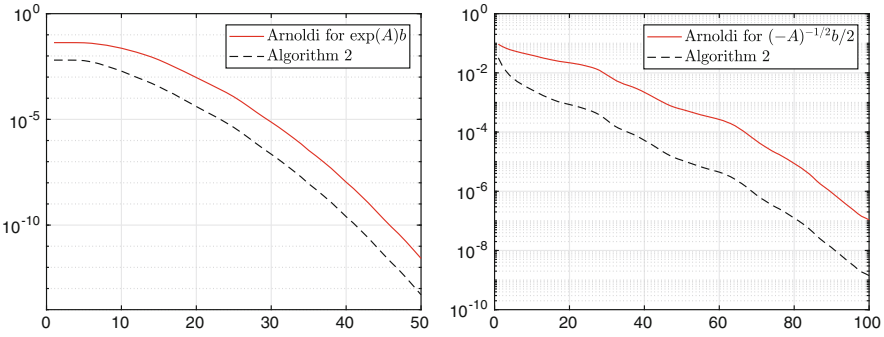


Fig. 2 Convergence of Algorithm 2 for approximating $Df\{A\}(cc^T)$ and convergence of Arnoldi method for approximating $f'(A)c$ for $f(z) = \exp(z)$ (left plot) and $f(z) = \sqrt{-z}$ (right plot)

by Algorithm 2. This is compared with the error $\|f'(A)c - U_k f'(G_k)\tilde{c}\|_2$ of the standard Arnoldi approximation for $f'(A)c$. Figure 2 demonstrates that both algorithms exhibit the same qualitative convergence behavior.

6 Outlook

This work offers numerous opportunities for future work. Most notably, it remains an open problem whether the result of Lemma 3 can be established with a constant independent of A, B. All experiments reported in this paper are of academic nature, their purpose is to illustrate convergence properties. Although the algorithms are, in principle, designed to tackle large-scale matrices, the detailed implementation in a large-scale setting has not been discussed and will be reported elsewhere.

Appendix: Polynomial Approximation of the ϕ Function

The φ function, which plays an important role in exponential integrators, is given by $\varphi(z) = (\exp(z) - 1)/z$. As φ is an entire function, we expect polynomial approximations to converge superlinearly. The following lemma derives such an error bound when considering approximations on an interval $[-4\rho, 0]$.

Lemma 5 *Let* $\rho > 0$ *and* $\varepsilon_k = \min_{p\in\Pi_{k-1}} \max_{z\in[-4\rho,0]} |\varphi(z) - p(z)|$. *Then*

$$\varepsilon_k \le 40\frac{\rho^2}{k^3}\exp\left(-\frac{k^2}{5\rho}\right) \qquad \text{for } \sqrt{4\rho} \le k \le 2\rho, \tag{13}$$

$$\varepsilon_k \leq \frac{8}{3k-5\rho}\left(\frac{e\rho}{k+2\rho}\right)^k \qquad \text{for } k \geq 2\rho. \tag{14}$$

Proof We use $x \mapsto (2x-2)\rho$ to map $[-1,1]$ to $[-4\rho, 0]$, yielding the equivalent polynomial optimization problem

$$\varepsilon_k = \min_{p\in\Pi_{k-1}} \max_{x\in[-1,1]} |\tilde{\varphi}(x) - p(x)|,$$

with $\tilde{\varphi}(x) := \varphi((2x-2)\rho)$. By [28, Theorem 2.2], we have for any $r > 1$ that

$$\varepsilon_k \leq 2\mu(\tilde{\varphi}, r)\frac{r^{-k}}{1-r^{-1}},$$

where

$$\begin{aligned}\mu(\tilde{\varphi}, r) &\leq \max_{\substack{w\in\mathbb{C}\\ |w|=r}} \left|\tilde{\varphi}\left(\left(w+w^{-1}\right)/2\right)\right| = \max_{\substack{w\in\mathbb{C}\\ |w|=r}} \left|\varphi\left(\left(w+w^{-1}-2\right)\rho\right)\right| \\ &= \left|\varphi\left(\left(r+r^{-1}-2\right)\rho\right)\right| \leq \frac{\exp\left(\left(r+r^{-1}-2\right)\rho\right)}{\left(r+r^{-1}-2\right)\rho}.\end{aligned}$$

The expression $\exp((r+r^{-1}-2)\rho)r^{-k}$ is minimized by setting $r := \frac{k}{2\rho} + \sqrt{\frac{k^2}{4\rho^2}+1}$. Note that $r^{-1} = \sqrt{\frac{k^2}{4\rho^2}+1} - \frac{k}{2\rho}$ and $(r+r^{-1}-2)\rho = \sqrt{k^2+4\rho^2} - 2\rho$.

We first discuss the case $\sqrt{4\rho} \leq k \leq 2\rho$, which in particular implies $\rho \geq 1$. The inequality

$$\frac{\sqrt{k^2+4\rho^2}-2\rho}{k} + \log\left(\sqrt{\frac{k^2}{4\rho^2}+1} - \frac{k}{2\rho}\right) \leq -\frac{k}{5\rho} \tag{15}$$

is shown for $k = \sqrt{4\rho}$ by direct calculation. By differentiating, it is shown that the difference between both sides of (15) is monotonically decreasing for $k \in [\sqrt{4\rho}, 2\rho]$ and hence the inequality holds for all such k. Using also

$$\sqrt{k^2+4\rho^2} - 2\rho \geq \frac{k^2}{5\rho}, \qquad 1 - r^{-1} \geq \frac{k}{4\rho},$$

we obtain from (15) that

$$\mu(\tilde{\varphi}, r)\frac{r^{-k}}{1-r^{-1}} \le \frac{\exp(\sqrt{k^2+4\rho^2}-2\rho)}{\sqrt{k^2+4\rho^2}-2\rho} \cdot \frac{\exp\left(k\log\left(\sqrt{\frac{k^2}{4\rho^2}+1}-\frac{k}{2\rho}\right)\right)}{1-r^{-1}}$$
$$\le 20\frac{\rho^2}{k^3}\exp\left(-\frac{k^2}{5\rho}\right),$$

which completes the proof of (13).

Similarly, the inequality (14) follows from combining

$$\frac{\sqrt{k^2+4\rho^2}-2\rho}{k} + \log\left(\sqrt{\frac{k^2}{4\rho^2}+1}-\frac{k}{2\rho}\right) \le \log(e\rho) - \log(k+2\rho)$$

with

$$(\sqrt{k^2+4\rho^2}-2\rho)(1-r^{-1}) \ge \frac{3}{4}k - \frac{5}{4}\rho,$$

which hold for $k \ge 2\rho$. □

Compared to the corresponding bounds for the exponential [19, Theorem 2], the bounds of Lemma 5 are lower for larger k, primarily because they benefit from the additional factor $\mathcal{O}(1/k)$ due to the slower growth of the φ function. Additionally, the factor $\left(\frac{e\rho}{k+2\rho}\right)^k$ in (14) seems to be better than the corresponding factor $\exp(-\rho)\left(\frac{e\rho}{k}\right)^k$ [19, Eqn. (14)]. This improvement can probably be carried over to the exponential. Figure 3 illustrates the differences between the bounds.

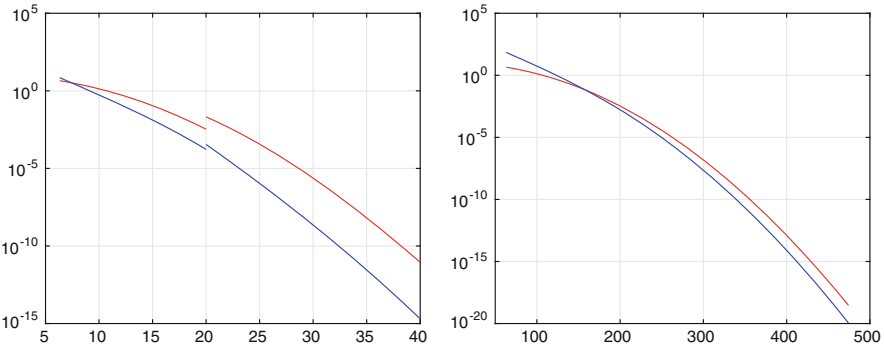


Fig. 3 Bounds of Lemma 5 for the polynomial approximation of the φ function (in blue) and bounds of [19, Theorem 2] for the polynomial approximation of the exponential function (in red). Left plot: $\rho = 10$. Right plot: $\rho = 1000$

Acknowledgements The author thanks Marcel Schweitzer for inspiring discussions on the topic of this work and Christian Lubich for the idea of the proof for Lemma 5. He also thanks the referees for their constructive comments, which improved the presentation of the paper.

References

1. Bartels, R.H., Stewart, G.W.: Algorithm 432: the solution of the matrix equation $AX + XB = C$. Commun. ACM **15**, 820–826 (1972)
2. Beckermann, B.: An error analysis for rational Galerkin projection applied to the Sylvester equation. SIAM J. Numer. Anal. **49**, 2430–2450 (2011)
3. Beckermann, B., Reichel, L.: Error estimates and evaluation of matrix functions via the Faber transform. SIAM J. Numer. Anal. **47**, 3849–3883 (2009)
4. Beckermann, B., Kressner, D., Schweitzer, M.: Low-rank updates of matrix functions. SIAM J. Matrix Anal. Appl. (2017, to appear). arXiv:1707.03045
5. Benner, P., Saak, J.: Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey. GAMM-Mitt. **36**, 32–52 (2013)
6. Benner, P., Kürschner, P., Saak, J.: Frequency-limited balanced truncation with low-rank approximations. SIAM J. Sci. Comput. **38**, A471–A499 (2016)
7. Benzi, M., Simoncini, V.: Approximation of functions of large matrices with Kronecker structure. Numer. Math. **135**, 1–26 (2017)
8. Crouzeix, M., Palencia, C.: The numerical range is a $(1 + \sqrt{2})$-spectral set. SIAM J. Matrix Anal. Appl. **38**, 649–655 (2017)
9. Daleckiĭ, J.L., Kreĭn, S.G.: Stability of Solutions of Differential Equations in Banach Space. American Mathematical Society, Providence (1974)
10. Frommer, A., Simoncini, V.: Matrix functions. In: Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 275–303. Springer, Berlin (2008)
11. Frommer, A., Lund, K., Szyld, D.B.: Block Krylov subspace methods for functions of matrices. Electron. Trans. Numer. Anal. **47**, 100–126 (2017)
12. Gawronski, W., Juang, J.-N.: Model reduction in limited time and frequency intervals. Int. J. Syst. Sci. **21**, 349–376 (1990)
13. Gil', M.: Norm estimates for functions of two non-commuting matrices. Electron. J. Linear Algebra **22**, 504–512 (2011)
14. Grasedyck, L., Hackbusch, W., Khoromskij, B.N.: Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices. Computing **70**, 121–165 (2003)
15. Güttel, S.: Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. GAMM-Mitt. **36**, 8–31 (2013)
16. Haber, A., Verhaegen, M.: Sparse solution of the Lyapunov equation for large-scale interconnected systems. Automatica J. IFAC **73**, 256–268 (2016)
17. Henrici, P.: Bounds for iterates, inverses, spectral variation and fields of values of non-normal matrices. Numer. Math. **4**, 24–40 (1962)
18. Higham, N. J.: Functions of Matrices. SIAM, Philadelphia (2008)
19. Hochbruck, M., Lubich, C.: On Krylov subspace approximations to the matrix exponential operator. SIAM J. Numer. Anal. **34**, 1911–1925 (1997)
20. Jaimoukha, I., Kasenally, E.: Oblique projection methods for large scale model reduction. SIAM J. Matrix Anal. Appl. **16**, 602–627 (1995)
21. Kandolf, P., Koskela, A., Relton, S.D., Schweitzer, M.: Computing low-rank approximations of the Fréchet derivative of a matrix function using Krylov subspace methods, Personal communication
22. Krantz, S.G.: Function Theory of Several Complex Variables. Wiley, New York (1982)
23. Kressner, D.: Bivariate matrix functions. Oper. Matrices **8**, 449–466 (2014)

24. Kressner, D., Tobler, C.: Krylov subspace methods for linear systems with tensor product structure. SIAM J. Matrix Anal. Appl. **31**, 1688–1714 (2010)
25. Kressner, D., Massei, S., Robol, L.: Low-rank updates and a divide-and-conquer method for linear matrix equations (2017). arXiv:1712.04349
26. Kürschner, P.: Balanced truncation model order reduction in limited time intervals for large systems (2017). arXiv:1707.02839
27. Lancaster, P.: Explicit solutions of linear matrix equations. SIAM Rev. **12**, 544–566 (1970)
28. Lubich, C.: From quantum to classical molecular dynamics: reduced models and numerical analysis. In: Zurich Lectures in Advanced Mathematics. European Mathematical Society (EMS), Zürich (2008)
29. Mathias, R.: A chain rule for matrix functions and applications. SIAM J. Matrix Anal. Appl. **17**, 610–620 (1996)
30. Palitta, D., Simoncini, V.: Numerical methods for large-scale Lyapunov equations with symmetric banded data (2017). arXiv 1711.04187
31. Palitta, D., Simoncini, V.: Computationally enhanced projection methods for symmetric Sylvester and Lyapunov matrix equations. J. Comput. Appl. Math. **330**, 648–659 (2018)
32. Saad, Y.: Numerical solution of large Lyapunov equations. In: Signal Processing, Scattering and Operator Theory, and Numerical Methods (Amsterdam, 1989). Program Systems Control Theory, vol. 5, pp. 503–511. Birkhäuser, Boston (1990)
33. Saad, Y.: Numerical Methods for Large Eigenvalue Problems: Theory and Algorithms. Wiley, New York (1992)
34. Simoncini, V.: On the numerical solution of $AX - XB = C$. BIT **36**, 814–830 (1996)
35. Simoncini, V.: Computational methods for linear matrix equations. SIAM Rev. **58**, 377–441 (2016)
36. Simoncini, V., Druskin, V.: Convergence analysis of projection methods for the numerical solution of large Lyapunov equations. SIAM J. Numer. Anal. **47**, 828–843 (2009)
37. Starke, G.: Fields of values and the ADI method for nonnormal matrices. Linear Algebra Appl. **180**, 199–218 (1993)
38. Stewart, G.W.: Matrix Algorithms, vol. II. SIAM, Philadelphia (2001). Eigensystems
39. Thanou, D., Dong, X., Kressner, D., Frossard, P.: Learning heat diffusion graphs. IEEE Trans. Signal Inform. Process. Netw. **3**, 484–499 (2017)
40. Trefethen, L.N.: Multivariate polynomial approximation in the hypercube. Proc. Am. Math. Soc. **145**, 4837–4844 (2017)

Uzawa-Type and Augmented Lagrangian Methods for Double Saddle Point Systems

Michele Benzi and Fatemeh Panjeh Ali Beik

Abstract We study different types of stationary iterative methods for solving a class of large, sparse linear systems with double saddle point structure. In particular, we propose a class of Uzawa-like methods including a generalized (block) Gauss-Seidel (GGS) scheme and a generalized (block) successive overrelaxation (GSOR) method. Both schemes rely on a relaxation parameter, and we establish convergence intervals for these parameters. Additionally, we investigate the performance of these methods in combination with an augmented Lagrangian approach. Numerical experiments are reported for test problems from two different applications, a mixed-hybrid discretization of the potential fluid flow problem and finite element modeling of liquid crystal directors. Our results show that fast convergence can be achieved with a suitable choice of parameters.

Keywords Uzawa-like methods · Double saddle point problems · Augmented Lagrangian method · Finite elements · Potential fluid flow · Liquid crystals

AMS Subject Classifications: 65F10, 65F08, 65F50

M. Benzi (✉)
Department of Mathematics and Computer Science, Emory University, Atlanta, GA, USA

Classe di Scienze, Scuola Normale Superiore, Pisa, Italy
e-mail: michele.benzi@sns.it; benzi@mathcs.emory.edu

F. P. A. Beik
Department of Mathematics, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran
e-mail: f.beik@vru.ac.ir

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_11

1 Introduction

Consider the following linear system of equations:

$$\mathscr{A}u \equiv \begin{bmatrix} A & B^T & C^T \\ B & 0 & 0 \\ C & 0 & -D \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \equiv b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite (SPD), $B \in \mathbb{R}^{m\times n}$ has full row rank, $C \in \mathbb{R}^{p\times n}$ and the matrix $D \in \mathbb{R}^{p\times p}$ is symmetric positive semidefinite (SPS). In this paper we focus primarily on two cases: D is either SPD, or the zero matrix. When D is zero, $C \in \mathbb{R}^{p\times n}$ is assumed to have full row rank. Throughout the paper we assume that $n \geq m + p$.

Linear systems of this type arise, e.g., from finite element models of liquid crystals (case $D \neq 0$) and from mixed finite element approximation of potential fluid flow problems (case $D = 0$); see [1, 10, 13] and the references therein for detailed descriptions of these problems.

The following two propositions give necessary and sufficient conditions for the invertibility of the coefficient matrix $\mathscr{A}$ in (1).

Proposition 1 ([1, Proposition 2.3]) *Let A be SPD and assume that B and C have full row rank. Consider the linear system (1) with $D = 0$. Then range$(B^T) \cap$ range$(C^T) = \{0\}$ is a necessary and sufficient condition for the coefficient matrix $\mathscr{A}$ to be invertible.*

Proposition 2 ([1, Proposition 2.1]) *Assume that A and D are respectively SPD and SPS matrices. Then matrix $\mathscr{A}$ is invertible if and only if B has full row rank.*

As is well known, stationary iterative schemes for solving $\mathscr{A}x = b$ are uniquely associated with a given splitting $\mathscr{A} = \mathscr{M} - \mathscr{N}$ where $\mathscr{M}$ is nonsingular. More precisely, an iterative scheme produces a sequence of approximate solutions as follows:

$$u_{k+1} = \mathscr{G}u_k + \mathscr{M}^{-1}b, \quad k = 0, 1, 2, \ldots, \tag{2}$$

where $\mathscr{G} = \mathscr{M}^{-1}\mathscr{N}$ and u_0 is given. It is well-known that (2) is convergent for any initial guess if and only if $\rho(\mathscr{G}) < 1$, see [14].

In practice, stationary methods may fail to converge or converge too slowly. For this reason they are usually combined with acceleration techniques, such as Chebyshev or Krylov subspace methods [14]. These acceleration schemes, while very successful, have some limitations. For instance, the use of Chebyshev acceleration may require spectral information that is not always available, while Krylov acceleration necessitates the computation of an orthonormal basis for the Krylov subspace. For methods like GMRES the latter operation is known to have an adverse impact on the parallel efficiency, especially on emerging multicore and

hybrid architectures [7, 17]. On future-generation exascale architectures, resilience is also expected to be an issue with these methods [4, 15]. This realization has spurred renewed interest in classical fixed point iterations of the form (2), which do not require any orthogonalization steps. Alternative acceleration techniques, such as Monte Carlo and Anderson-type acceleration, are currently being investigated by researchers [4, 9, 11, 12, 16]. Acceleration is only needed, of course, if the basic stationary scheme (2) converges slowly. There are, however, situations where fast convergence of (2) can be obtained, for example through the use of suitable relaxation parameters and, in the case of saddle point problems, augmented Lagrangian techniques. In this paper we show that it is possible to have fast convergence of stationary methods for linear systems of the form (2), without the need for Krylov acceleration.

The remainder of this paper is organized as follows. Before ending this section, we present some notations that are used throughout the paper. In Sect. 2 we investigate a class of Uzawa-like methods, which can also be interpreted as generalized (block) Gauss–Seidel method. In this section, we also consider the use of an augmented Lagrangian technique to improve the performance of the iteration. In Sect. 3 we propose the GSOR method to solve (1) in the case that its (3, 3)-block is SPD. The convergence properties of the GSOR method are also studied. Illustrative examples are reported in Sect. 4 for test problems appearing in groundwater flow and liquid crystal modeling. Finally, we briefly state our conclusions in Sect. 5.

Notations For a given arbitrary square matrix W, its spectrum is denoted by $\sigma(W)$. If all eigenvalues of W are real, we use $\lambda_{\min}(W)$ and $\lambda_{\max}(W)$ to denote the minimum and maximum eigenvalues of W, respectively. Moreover, the notation $\rho(W)$ stands for the spectral radius of W. If W is symmetric positive (semi)definite we write $W \succ 0$ ($W \succcurlyeq 0$). Furthermore for two given matrices W_1 and W_2, by $W_1 \succ W_2$ ($W_1 \succcurlyeq W_2$) we mean $W_1 - W_2 \succ 0$ ($W_1 - W_2 \succcurlyeq 0$). For given vectors x, y and z of dimensions n, m and p, $(x; y; z)$ will denote a column vector of dimension $n + m + p$.

2 Uzawa-Like Iterative Schemes

Uzawa's method (see, e.g., [3]) has long been a popular technique for solving saddle point problems. In this section, we investigate possible extensions of Uzawa's method to the double saddle point problem (1). Since this involves a (lower) block triangular splitting of the coefficient matrix, these schemes can also be regarded as a generalization of the classical (block) Gauss–Seidel scheme. To this end, first we split $\mathscr{A}$ as follows:

$$\mathscr{A} = \mathscr{M}_{GGS} - \mathscr{N}_{GGS}, \tag{3}$$

where

$$\mathscr{M}_{GGS} = \begin{bmatrix} A & 0 & 0 \\ B & -\frac{1}{\alpha}Q & 0 \\ C & 0 & M \end{bmatrix} \quad \text{and} \quad \mathscr{N}_{GGS} = \begin{bmatrix} 0 & -B^T & -C^T \\ 0 & -\frac{1}{\alpha}Q & 0 \\ 0 & 0 & N \end{bmatrix},$$

in which the parameter $\alpha > 0$ and the matrix $Q \succ 0$ are given and $D = N - M$ where M is a negative definite matrix.

2.1 Double Saddle Point Problems with Zero (3,3)-Block

Here we assume that the matrix D in $\mathscr{A}$ is zero. Substituting $M = N$ into the splitting (3), we consider the following iterative method for solving (1),

$$u_{k+1} = \bar{\mathscr{G}}_{GGS} u_k + \mathscr{M}_{GGS}^{-1} b, \qquad k = 0, 1, 2 \ldots, \tag{4}$$

in which the arbitrary initial guess u_0 is given and

$$\bar{\mathscr{G}}_{GGS} = \begin{bmatrix} 0 & -A^{-1}B^T & -A^{-1}C^T \\ 0 & I - \alpha Q^{-1} S_B & -\alpha Q^{-1} B A^{-1} C^T \\ 0 & M^{-1} C A^{-1} B^T & I + M^{-1} S_C \end{bmatrix}, \tag{5}$$

where $S_B = BA^{-1}B^T$ and $S_C = CA^{-1}C^T$.

We recall next the following theorem and lemma, which we need to prove the convergence of iterative method (4) under appropriate conditions. The lemma is an immediate consequence of Weyl's Theorem, see [8, Theorem 4.3.1].

Theorem 1 ([8, Theorem 7.7.3]) *Let A and B be two $n \times n$ real symmetric matrices such that A is positive definite and B is positive semidefinite. Then $A \succcurlyeq B$ if and only if $\rho(A^{-1}B) \leq 1$, and $A \succ B$ if and only if $\rho(A^{-1}B) < 1$.*

Lemma 1 *Let A and B be two Hermitian matrices. Then,*

$$\lambda_{max}(A + B) \leq \lambda_{max}(A) + \lambda_{max}(B),$$
$$\lambda_{min}(A + B) \geq \lambda_{min}(A) + \lambda_{min}(B).$$

Theorem 2 *Let $A \succ 0$, $Q \succ 0$ and $M \prec 0$. Assume that the matrices B and C have full row rank and that range$(B^T) \cap$ range$(C^T) = \{0\}$. If $-M \succcurlyeq CA^{-1}C^T$ and*

$$0 < \alpha \leq \frac{1}{\lambda_{max}(Q^{-1} S_B)}, \tag{6}$$

the iterative scheme (4) converges to the solution of (1).

Proof Let $\lambda \in \sigma(\bar{\mathscr{G}}_{GGS})$ and $(x; y; z)$ be a corresponding eigenvector which is equivalent to say that

$$-B^T y - C^T z = \lambda A x, \tag{7}$$

$$-\frac{1}{\alpha} Q y \qquad = \lambda(Bx - \frac{1}{\alpha} Q y), \tag{8}$$

$$M z = \lambda(Cx + Mz). \tag{9}$$

First we observe that $\lambda \neq 1$. Otherwise, From (8) and (9), we respectively conclude that $Bx = 0$ and $Cx = 0$ which together with (7) and the positive definiteness of A imply that $x = 0$. Now using the assumption range$(B^T) \cap$ range$(C^T) = \{0\}$, we can deduce that y and z are both zero vectors. Consequently, it must be $(x; y; z) = (0; 0; 0)$, which contradicts our assumption that $(x; y; z)$ is an eigenvector. Assuming $\lambda \neq 1$, from (8) and (9), we have

$$y = \frac{\lambda}{\lambda - 1} \alpha Q^{-1} B x \quad \text{and} \quad z = \frac{\lambda}{1 - \lambda} M^{-1} C x.$$

We observe that x cannot be zero. Substituting y and z from the above relation into (7), it can be found that λ is either zero or it satisfies the following relation:

$$1 - \lambda = \alpha \tilde{p} - \tilde{q}, \tag{10}$$

where

$$\tilde{p} = \frac{x^* B^T Q^{-1} B x}{x^* A x} \quad \text{and} \quad \tilde{q} = \frac{x^* C^T M^{-1} C x}{x^* A x}.$$

By the assumptions it must be $\tilde{p} \geq 0$ and $\tilde{q} \leq 0$, therefore $1 - \lambda \geq 0$. In view of the fact that $\lambda \neq 1$, we conclude that $\lambda < 1$. We observe that

$$\tilde{p} \leq \max_{x \neq 0} \frac{x^* B^T Q^{-1} B x}{x^* A x} = \lambda_{\max}(A^{-1} B^T Q^{-1} B) = \lambda_{\max}(Q^{-1} S_B),$$

and

$$-\tilde{q} \leq \max_{x \neq 0} \frac{-x^* C^T M^{-1} C x}{x^* A x} = \lambda_{\max}(-A^{-1} C^T M^{-1} C) = \lambda_{\max}(-M^{-1} S_C).$$

The assumption $-M \succcurlyeq C A^{-1} C^T$ ensures that $-1 \leq \tilde{q}$. Hence, we conclude that

$$1 - \lambda \leq \alpha \lambda_{\max}(Q^{-1} S_B) + 1.$$

Since $\alpha\lambda_{\max}(Q^{-1}S_B) \leq 1$, we have $1 - \lambda \leq 2$ which implies that $-1 \leq \lambda$. To complete the proof, we only need to show that $\lambda \neq -1$. Let $\lambda = -1$, from (10), we have

$$\frac{\alpha v^* A^{-\frac{1}{2}} B^T Q^{-1} B A^{-\frac{1}{2}} v}{v^* v} - \frac{v^* A^{-\frac{1}{2}} C^T M^{-1} C A^{-\frac{1}{2}} v}{v^* v} = 2,$$

where $v = A^{\frac{1}{2}} x$. The above relation is equivalent to

$$\frac{v^*(I - \alpha A^{-\frac{1}{2}} B^T Q^{-1} B A^{-\frac{1}{2}})v}{v^* v} + \frac{v^*(I + A^{-\frac{1}{2}} C^T M^{-1} C A^{-\frac{1}{2}})v}{v^* v} = 0.$$

In view of (6) and $-M \succcurlyeq C A^{-1} C^T$, the following two matrices

$$I - \alpha A^{-\frac{1}{2}} B^T Q^{-1} B A^{-\frac{1}{2}} \quad \text{and} \quad I + A^{-\frac{1}{2}} C^T M^{-1} C A^{-\frac{1}{2}},$$

are both symmetric positive semidefinite. Consequently, we deduce that

$$\alpha A^{-\frac{1}{2}} B^T Q^{-1} B A^{-\frac{1}{2}} v = v \quad \text{and} \quad - A^{-\frac{1}{2}} C^T M^{-1} C A^{-\frac{1}{2}} v = v.$$

The preceding two relations imply that

$$\alpha B^T Q^{-1} B A^{-\frac{1}{2}} v = -C^T M^{-1} C A^{-\frac{1}{2}} v.$$

Since $\text{range}(B^T) \cap \text{range}(C^T) = \{0\}$, the above equality implies that $BA^{-\frac{1}{2}} v = 0$ and $CA^{-\frac{1}{2}} v = 0$, which is equivalent to say that $Bx = 0$ and $Cx = 0$. Notice that $Bx = 0$ and $Cx = 0$ implies that $(x; y; z) = 0$, which is a contradiction. This completes the proof.

2.2 *Double Saddle Point Problems with SPD (3,3)-Block*

Here we assume that $D \succ 0$ and $A \succ C^T D^{-1} C$. The latter assumption warrants some discussion. In the case of linear systems of the form (1) arising from liquid crystal modeling, we have been able to verify numerically that the condition holds true for problems of small or moderate size, and numerical tests suggest that it may hold for larger problems as well. In some cases, it may be possible to enforce the condition by a suitable modification of the (1, 1) block A (augmented Lagrangian technique). We first consider the case where the assumption is satisfied, then in the next subsection we briefly discuss the augmented Lagrangian approach.

Substituting $M = -D$ into (3) results in the splitting

$$\mathscr{A} = \mathscr{M}_{GGS} - \mathscr{N}_{GGS},$$

with

$$\mathscr{M}_{GGS} = \begin{bmatrix} A & 0 & 0 \\ B & -\frac{1}{\alpha}Q & 0 \\ C & 0 & -D \end{bmatrix},$$

where the parameter $\alpha > 0$ and $Q \succ 0$ are given. The corresponding GGS iterative scheme is given by

$$u_{k+1} = \mathscr{G}_{GGS} u_k + \mathscr{M}_{GGS}^{-1} b, \quad k = 0, 1, 2, \ldots, \tag{11}$$

where $\mathscr{G}_{GGS} = \mathscr{M}_{GGS}^{-1} \mathscr{N}_{GGS}$ and u_0 is given.

In the following, we obtain a sufficient condition for the convergence of the GGS iterative scheme. To this end, we first recall the following useful lemma.

Lemma 2 ([18, Section 6.2]) *Consider the quadratic equation $x^2 - bx + c = 0$, where b and c are real numbers. Both roots of the equation are less than one in modulus if and only if $|c| < 1$ and $|b| < 1 + c$.*

Theorem 3 *Let $A \succ 0$ and $D \succ 0$ and let B be a full row rank matrix. Suppose that $A \succ C^T D^{-1} C$. If*

$$0 < \alpha < \frac{2 - 2\lambda_{\,max}(\mathscr{G})}{\lambda_{\,max}(A^{-1} B^T Q^{-1} B)}, \tag{12}$$

where $\mathscr{G} = A^{-1} C^T D^{-1} C$, then $\rho(\mathscr{G}_{GGS}) < 1$.

Proof Note that $\lambda = 0$ is an eigenvalue of $\mathscr{G}_{GGS}$ with the possible eigenvector $(x; 0; z)$ where x is an arbitrary nonzero vector and z can be either a zero vector or, if C^T does not have full column rank, a nonzero vector in $\ker(C^T)$.

Now let $\lambda \neq 0$ be an arbitrary eigenvalue of $\mathscr{G}_{GGS} = \mathscr{M}_{GGS}^{-1} \mathscr{N}_{GGS}$. Therefore, there exists a nonzero vector $(x; y; z)$ such that

$$-B^T y - C^T z = \lambda A x, \tag{13}$$

$$-\frac{1}{\alpha} Q y \qquad = \lambda (B x - \frac{1}{\alpha} Q y), \tag{14}$$

$$0 = \lambda (C x - D z). \tag{15}$$

Evidently $x \neq 0$, otherwise in view of the positive definiteness of D and the fact that B has full row rank we would have that y and z are both zero vectors, contradicting the assumption that $(x; y; z)$ is an eigenvector.

From (14) we derive

$$\frac{\lambda-1}{\alpha}y=\lambda Q^{-1}Bx, \tag{16}$$

The vector z can be computed from (15) by

$$z=D^{-1}Cx.$$

Premultiplying (13) by $\frac{\lambda-1}{\alpha}$ and then substituting z from the above relation, we have

$$\frac{\lambda(\lambda-1)}{\alpha}Ax=-\frac{(\lambda-1)}{\alpha}B^Ty-\frac{(\lambda-1)}{\alpha}C^TD^{-1}Cx.$$

In view of (16), the above equation can be rewritten as follows:

$$\frac{\lambda(\lambda-1)}{\alpha}Ax=-\lambda B^TQ^{-1}Bx-\frac{(\lambda-1)}{\alpha}C^TD^{-1}Cx.$$

Multiplying the above equation by αx^* on the left side, we obtain the following quadratic equation:

$$\lambda^2+(-1+\alpha p+q)\lambda-q=0, \tag{17}$$

where

$$p=\frac{x^*B^TQ^{-1}Bx}{x^*Ax}\quad\text{and}\quad q=\frac{x^*C^TD^{-1}Cx}{x^*Ax}. \tag{18}$$

It is not difficult to verify that

$$p\le\max_{x\neq0}\frac{x^*B^TQ^{-1}Bx}{x^*Ax}=\lambda_{\max}(A^{-1}B^TQ^{-1}B),$$

and

$$q\le\max_{x\neq0}\frac{x^*C^TD^{-1}Cx}{x^*Ax}=\lambda_{\max}(\mathscr{G}).$$

Notice that the assumption $A\succ C^TD^{-1}C$ implies that $\rho(\mathscr{G})<1$. Invoking the above inequality, $\rho(\mathscr{G})<1$ implies $q<1$. Also, $q<1$ together with (12) ensures that $|-1+\alpha p+q|<1-q$. The result now follows from Lemma 2.

We end this section by the following remark on the parameter α in the GGS method.

Remark 1 Under the assumptions of Theorem 3, the roots of the quadratic equation (17) are given by

$$\frac{(1-\alpha p-q)\pm\sqrt{(-1+\alpha p+q)^2+4q}}{2},$$

where p and q are in the forms (18), respectively. Therefore, the eigenvalues of $\mathscr{G}_{GGS}=\mathscr{M}_{GGS}^{-1}\mathscr{N}_{GGS}$ are all real. Assume that α is chosen such that

$$-1+\alpha\lambda_{\max}(A^{-1}B^TQ^{-1}B)+\lambda_{\max}(\mathscr{G})=0,$$

which is equivalent to say that

$$\alpha=\frac{1-\lambda_{\max}(\mathscr{G})}{\lambda_{\max}(A^{-1}B^TQ^{-1}B)}.$$

Note that for the choice $Q=BA^{-1}B^T$ then $\lambda_{\max}(A^{-1}B^TQ^{-1}B)=1$. Our numerical tests show that in this case $\bar{\alpha}=1-\lambda_{\max}(\mathscr{G})$ is a very good approximation of the optimum value of α. In particular, for the test problems arising from the liquid crystal model the value of $\bar{\alpha}$ remains roughly constant as the dimension of the problem increases.

2.3 Augmenting the (1,1)-Block of Double Saddle Point Problems

The result in the previous subsection relies on the assumption that $A\succ C^TD^{-1}C$. Although this condition appears to be satisfied in some cases of practical interest, it is rather restrictive and one cannot expect it to always hold. In some cases, it may be possible to enforce the condition by applying the iterative scheme (11) to an equivalent linear system of equations, rather than directly to the original system (1). Indeed, the double saddle point system (1) is equivalent to the following linear system of equations:

$$\mathscr{A}u\equiv\begin{bmatrix}\hat{A}&B^T&C^T\\B&0&0\\C&0&-D\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}b_1+rB^Tb_2\\b_2\\b_3\end{bmatrix}\equiv b,\tag{19}$$

where $\hat{A}=A+rB^TB$ for a given $r>0$. Note that if we wish to apply the proposed iterative scheme (11) for solving (19), we may be able to choose r large enough so that the assumption $\hat{A}\succ C^TD^{-1}C$ holds, since $\hat{A}\succeq A$ for all $r>0$.

Table 1 Numerical results with iterative method (4), potential fluid flow problem ($\alpha = 1$ and $\epsilon = 10^{-7}$)

	$r = 0$	$r = 20$				$r = 200$			
Size	Iter	Iter	CPU	Iter_{pcg}	Err	Iter	CPU	Iter_{pcg}	Err
2125	†	6	0.0148	102	0.3183e−07	4	0.0103	68	0.3462e−06
17,000	†	8	0.1218	240	0.2985e−07	6	0.0912	180	0.6884e−08
57,375	†	8	0.6007	342	0.1608e−05	6	0.4548	257	0.6967e−07
136,000	†	10	2.6079	555	0.1443e−05	6	1.6174	335	0.5211e−06
265,625	†	12	8.1918	805	0.2172e−05	6	4.1609	413	0.2498e−05
459,000	†	16	24.084	1248	0.7639e−06	8	12.602	648	0.1809e−06

For example, consider the case $A = I_2$, $B = C = e_1 = [1, 0]$, and $D = [\frac{1}{2}]$. Then $C^T D^{-1} C = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}$ and the condition $A \succ C^T D^{-1} C$ fails to hold. However, $A + rB^T B = \begin{bmatrix} 1+r & 0 \\ 0 & 1 \end{bmatrix}$ and thus the condition $A + rB^T B \succ C^T D^{-1} C$ holds for all $r > 1$.

Augmentation can be beneficial also for the case $D = 0$, where it leads to faster convergence of the GGS iterative scheme. However, generally speaking, there may be a price to pay for this faster convergence: solving linear systems with $\hat{A}$ is often more expensive than solving systems with A. In particular, augmentation often leads to loss of sparsity in A. Note, however, that this need not always be the case. Using the well-known Sherman–Morrison formula, we obtain $B\hat{A}^{-1}B^T = S_B - rS_B(I + rS_B)^{-1}S_B$. In the case of the potential fluid flow problem studied in [10], it turns out that S_B is diagonal. In view of the preceding relation, we conclude that $\hat{S}_B = B\hat{A}^{-1}B^T$ is also a diagonal matrix. Hence, in this case there is virtually no increase in costs associated with the augmented Lagrangian approach. Moreover, with the choice $Q = \hat{S}_B$, it can be shown (and is numerically observed) that the convergence rate of the iterative scheme (4) for solving (19), asymptotically, becomes faster for increasing values of r; see the results in Table 1.

As already mentioned, for the double saddle point problems arising from liquid crystal problem, the condition $A \succ C^T D^{-1} C$ seems to be satisfied. However, our numerical experiments show that augmentation can be beneficial here as well. Indeed, increasing the value of r and applying iterative method (11) to the equivalent system (19) provides better results than those obtained from applying the method to (1), both in terms of number of iterations and total solution time; see Table 4 for further details.

We emphasize again that augmenting the (1, 1)-block does not always lead to any excessive fill-in problems. For the sake of illustration, in Fig. 1 we compare the sparsity patterns of the (1, 1)-blocks of the coefficient matrix in (1) and (19) corresponding to potential fluid flow and liquid crystal problems, respectively. It is clear that the loss of sparsity is very modest for the first problem. For the second problem, it may seem at first sight that augmentation destroys the sparsity of the

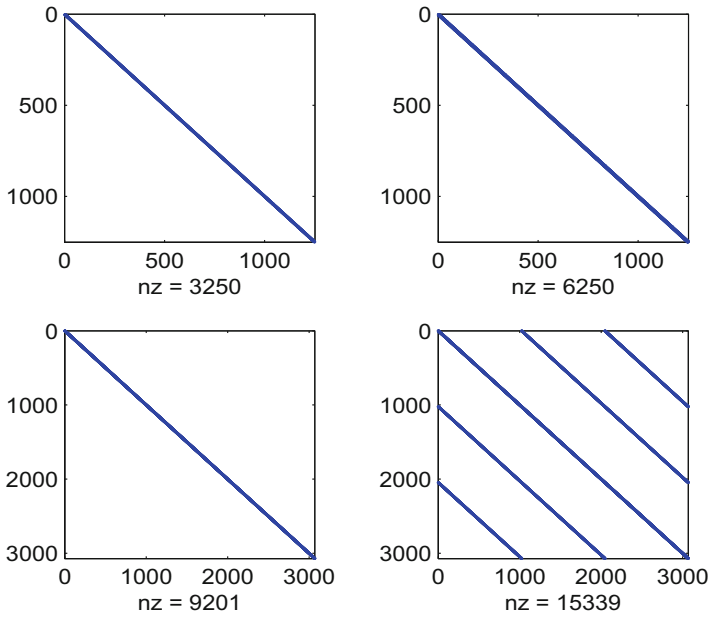


Fig. 1 Sparsity pattern of A (left) versus sparsity pattern of $A + rB^TB$ (right). Top: potential fluid flow problem ($n = 1250$ corresponds to problem size 2125) Bottom: liquid crystal problem ($n = 3069$ corresponds to problem size 5115)

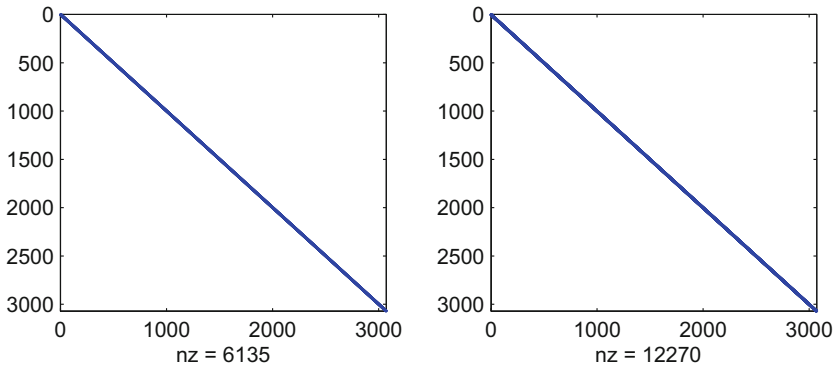


Fig. 2 Sparsity patterns. Left: factor of A obtained by Cholesky factorization; Right: factor of $A + rB^TB$ obtained by Cholesky factorization with SYMAMD reordering ($n = 3069$ corresponds to problem size 5115, liquid crystal problem)

(1, 1) block. However, it turns out that using an appropriate reordering (symmetric AMD) leads to a very sparse Cholesky factor with a modest increase in fill-in, see Fig. 2.

3 A Generalization of the Block SOR Method

In this section, we further develop the idea of the well-known SOR iterative method to construct an iterative scheme for solving (1) in the case that $D \succ 0$. To this end, first, we need to consider the following splitting

$$\mathscr{A} = \mathscr{M}_{GSOR} - \mathscr{N}_{GSOR}, \tag{20}$$

with

$$\mathscr{M}_{GSOR} = \frac{1}{\omega}\begin{bmatrix} A & B^T & 0 \\ B & 0 & 0 \\ \omega C & 0 & -D \end{bmatrix} \quad \text{and} \quad \mathscr{N}_{GSOR} = \frac{1}{\omega}\begin{bmatrix} (1-\omega)A & (1-\omega)B^T & -\omega C^T \\ (1-\omega)B & 0 & 0 \\ 0 & 0 & -(1-\omega)D \end{bmatrix},$$

where $\omega \neq 0$ is given. Therefore, using the splitting (20), we derive the GSOR method as follows:

$$u_{k+1} = \mathscr{G}_{GSOR} u_k + \mathscr{M}_{GSOR}^{-1} b, \quad k = 0, 1, 2, \ldots, \tag{21}$$

where $\mathscr{G}_{GSOR} = \mathscr{M}_{GSOR}^{-1}\mathscr{N}_{GSOR}$ and the initial guess u_0 is given.

A possible procedure for the computation of $u_{k+1} = (x_{k+1}; y_{k+1}; z_{k+1})$ from $u_k = (x_k; y_k; z_k)$ is detailed in Algorithm 1.

Algorithm 1: Computing $(k+1)$-th approximation in the GSOR method

Compute the vectors r_1 and r_2 as follows:
  $r_1 = (1-\omega)Ax_k + (1-\omega)B^T y_k - \omega C^T z_k + \omega b_1$;
  $r_2 = (1-\omega)Bx_k + \omega b_2$;
Solve the following two systems to find y_{k+1} and x_{k+1}, respectively,
  $(BA^{-1}B^T)y_{k+1} = BA^{-1}r_1 - r_2$;
  $Ax_{k+1} = r_1 - B^T y_{k+1}$;
Compute $r_3 = \omega C x_{k+1} + (1-\omega)Dz_k - \omega b_3$;
Solve $Dz_{k+1} = r_3$ to find z_{k+1}.

Next, we establish a theorem about the eigenvalues of $\mathscr{G}_{GSOR}$. The theorem plays a key role in deriving a sufficient condition for the convergence of the GSOR method.

Theorem 4 *Assume that $A \succ 0$, $D \succ 0$ and B has full row rank. Also suppose that $\lambda \in \sigma(\mathscr{G}_{GSOR})$ where $\mathscr{G}_{GSOR}$ denotes the iteration matrix of the GSOR method. Then either $\lambda = 1-\omega$ or $\lambda \neq 1-\omega$ and there exists a positive constant μ such that the following relation holds:*

$$(\lambda + \omega - 1)^2 = -\omega^2 \lambda \mu. \tag{22}$$

Proof Let $\lambda \in \sigma(\mathscr{G}_{GSOR})$, hence there exists a nonzero vector $v = (x; y; z)$ such that $\mathscr{N}_{GSOR} v = \lambda \mathscr{M}_{GSOR} v$, or equivalently,

$$(1-\omega)Ax + (1-\omega)B^T y - \omega C^T z = \lambda(Ax + B^T y) \tag{23}$$

$$(1-\omega)Bx = \lambda Bx \tag{24}$$

$$-(1-\omega)Dz = \lambda(\omega Cx - Dz). \tag{25}$$

Evidently $\lambda = 1-\omega$ is an eigenvalue of $\mathscr{G}_{GSOR}$ with a corresponding eigenvector of the form $(0; y; 0)$ where $y \neq 0$. In the rest of proof, we assume that $\lambda \neq 1-\omega$.

From (25), we have

$$z = \frac{\lambda\omega}{\lambda + \omega - 1} D^{-1} Cx. \tag{26}$$

Invoking the earlier assumption that $\lambda \neq 1-\omega$ and in view of (24), we get $Bx = 0$. Notice that $x = 0$ with $\lambda \neq 1-\omega$ implies that $(x; y; z) = (0; 0; 0)$, in contradiction with the fact that $(x; y; z)$ is an eigenvector. Consequently, $x \neq 0$. Premultiplying (23) by x^* and then substituting z from (26) into it, we obtain $(\lambda + \omega - 1)^2 = -\omega^2 \lambda \mu$ where

$$\mu = \frac{x^* C^T D^{-1} Cx}{x^* Ax},$$

which completes the proof.

The above theorem can be used to establish that if the parameter ω lies in a certain interval, then the GSOR method for solving (1) is convergent.

Theorem 5 *Assume that $A \succ 0$, $D \succ 0$, B has full row rank and $\mathscr{G} = A^{-1}C^T D^{-1}C$. If $\omega \in (0, \frac{2}{1+\sqrt{\rho(\mathscr{G})}})$, then the GSOR method converges to the exact solution of (1) for any initial guess.*

Proof Let $\lambda \in \sigma(\mathscr{G}_{GSOR})$, we need to show that $|\lambda| < 1$. Note that from the assumption, it is obvious that $\omega \in (0, 2)$ which implies $|1-\omega| < 1$. This ensures that if $\lambda = 1-\omega$, we immediately obtain the result.

In the remaining part of the proof, we assume that $\lambda \neq 1-\omega$. By Theorem 4, there exists $\mu > 0$ such that (22) holds. Simplifying (22), we derive

$$\lambda^2 + (\omega^2\mu + 2\omega - 2)\lambda + (\omega - 1)^2 = 0.$$

Notice that by Lemma 2, one may conclude that $|\lambda| < 1$ if,

$$|\omega - 1| < 1, \tag{27}$$

and,

$$|\omega^2\mu + 2\omega - 2| < 1 + (\omega - 1)^2. \tag{28}$$

As observed earlier, the inequality (27) is equivalent to the fact that $\omega \in (0, 2)$. Note that (28) holds as soon as $\omega^2\mu < (\omega - 2)^2$. From the assumption it is easy to check that $\omega^2\rho(\mathscr{G}) < (\omega - 2)^2$. Now the fact that $\mu < \rho(\mathscr{G})$ implies the desired result.

We end this section with the following remark.

Remark 2 Numerical tests show that if $A \succ C^T D^{-1} C$ and $D \succ 0$, then

$$\bar{\omega} = \frac{2}{1 + \sqrt{1 + \rho(\mathscr{G})}}$$

is a good approximation for the experimentally obtained optimum parameter of the GSOR method to solve (1). We recall here that numerical observations indicate that the condition $A \succ C^T D^{-1} C$ holds for (1) arising from the liquid crystal problem. As pointed out in Remark 1, the value of $\rho(\mathscr{G})$ remains roughly constant for all sizes of (1) when solving the liquid crystal problem.

4 Numerical Experiments

In this section we present the results of numerical tests on two sets of problems of the type (1) arising in the finite element modeling of potential fluid flow problems (with $D = 0$) and liquid crystals (with $D \neq 0$).

All of the reported numerical results were performed on a 64-bit 2.45 GHz core i7 processor and 8.00 GB RAM using MATLAB version 8.3.0532. In all of the experiments, we have used right-hand sides corresponding to random solution vectors, performing ten runs and then averaging the CPU-times. At each iteration of the proposed iterative methods, we need to solve at least two SPD linear systems as subtasks. These are either solved by Cholesky factorization (when feasible) or by the preconditioned conjugate gradient (PCG) method using a strict tolerance; loose tolerances can be employed in the case of *inexact* variants (see [6]). The iteration counts reported in the tables (under "Iter") are also averages (rounded to the nearest integer). Under "Iter$_{pcg}$", we report the total number of inner PCG iterations performed. In all of the following numerical tests, the initial guess is taken to be the zero vector and the iterations are stopped once $\|\mathscr{A}u_k - b\|_2 < \epsilon\|b\|_2$ where u_k is the obtained k-th approximate solution to the exact solution of (1) and ϵ is given. Furthermore, under "Err" we report the relative error $\|u_k - u^*\|_2/\|u^*\|_2$, averaged over the ten runs. Here u_k is the approximate solution obtained with the above described stopping criterion and u^* is the exact solution of (1).

Unless otherwise specified, the preconditioner in the PCG method is a drop tolerance-based incomplete Cholesky factorization [2] computed using the MATLAB function "`ichol(.,opts)`", where

- `opts.type = 'ict'`,
- `opts.droptol = 1e-2`.

Example 1 Here we consider linear systems of equations of the form

$$\begin{bmatrix} A & B^T & C^T \\ B & 0 & 0 \\ C & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, \tag{29}$$

arising from a low-order Raviart–Thomas mixed-hybrid finite element approximation [5] of Darcy's law and continuity equation describing the three-dimensional potential fluid flow problem in porous media. For this problem we have that the conditions of Proposition 1 are satisfied, hence $\mathscr{A}$ is nonsingular. Details on the dimensions of the sub-blocks A, B, and C and further information can be found in [10, Table 1]. For this test problem, the SPD matrix A is block diagonal with small (5×5) dense blocks, and linear systems associated with it can be solved very cheaply by means of Cholesky factorization. Furthermore, it turns out that the Schur complement $S_B = BA^{-1}B^T$ is a scalar multiple of the $m \times m$ identity matrix. First we set $Q = BA^{-1}B^T$ and $M = -CA^{-1}C^T$, then we used iterative method (4) to solve (29). Notice that Theorem 2 shows that the iterative method (4) converges with the specified choices of Q and M. However, the method converges too slowly (after 1000 iterations the stopping criterion is not yet satisfied), denoted by † in Table 1 for the case that $r = 0$ in (19) with zero $(3, 3)$-block. Then we applied iterative method (4) with $Q = B(A + rB^TB)^{-1}B^T$ and $M = -C(A + rB^TB)^{-1}C^T$ for solving the equivalent (augmented Lagrangian) linear system of the form (19) with $D = 0$. We observed that the best results for the GGS method to solve (29) are obtained when $\alpha = 1$, and this is the value used in the numerical tests reported in Tables 1 and 2. To illustrate the sensitivity of the method for different values of α, we also reported the performance of the method for problem size 2125 for two different values of α and three different values of $r > 0$ in Table 3. From the reported results, it can be seen that increasing the value of r speeds up the convergence of iterative method (4). As pointed out in Sect. 2.3, the matrix $B(A + rB^TB)^{-1}B^T$ is also a scalar multiple of the $m \times m$ identity matrix. We comment that M is sparse and inexpensive to form and the linear systems corresponding to it were solved with PCG with the inner-tolerance 10^{-15}.

Example 2 In this example we consider linear systems of equations arising from liquid crystal directors modeling, see [13]. These are double saddle point systems of the form (1) where A is $n \times n$, B is $m \times n$, C is $p \times n$ and D is $p \times p$ with $n = 3k$ and $m = p = k$. Here k is an integer taking up seven values, ranging from 1023 to 61,535. All the nonzero blocks in $\mathscr{A}$ are sparse and structured. A detailed description of the submatrices A, B, C and D is given in [13]. Here we mention that

Table 2 Numerical results with iterative method (4), potential fluid flow problem ($\alpha = 1$ and $\epsilon = 10^{-7}$)

	$r = 500$				$r = 1000$				$r = 2000$			
Size	Iter	CPU	Iter_{pcg}	Err	Iter	CPU	Iter_{pcg}	Err	Iter	CPU	Iter_{pcg}	Err
2125	4	0.0099	68	0.5554e−07			†				†	
17,000	4	0.0611	120	0.8824e−06	4	0.0617	120	0.2317e−07			†	
57,375	6	0.4664	258	0.1012e−07	4	0.3038	172	0.1111e−05			†	
136,000	6	1.5920	336	0.3479e−07	6	0.6725	336	0.3916e−07			†	
265,625	6	4.1979	413	0.1627e−07	6	4.0095	413	0.2930e−07			†	
459,000	6	9.4211	486	0.5995e−06	6	9.5061	492	0.8126e−07	6	9.4341	492	0.5668e−07

Table 3 Numerical results with iterative method (4), potential fluid flow problem (problem size 2125 and $\epsilon = 10^{-7}$)

	$\alpha = 0.8$				$\alpha = 1.2$			
r	Iter	CPU	Iter_{pcg}	Err	Iter	CPU	Iter_{pcg}	Err
20	11	0.0266	187	0.1135e−06	11	0.0264	187	0.1164e−06
200	11	0.0260	187	0.3631e−07	11	0.0277	187	0.3623e−07
500	11	0.0269	187	0.3977e−07	11	0.0265	187	0.2922e−07

A is SPD, B has full row rank, C is rank deficient, and D is tridiagonal and SPD. By Proposition 2, $\mathscr{A}$ is nonsingular.

In applying iterative method (11), we set $Q = BA^{-1}B^T$ and $\alpha \approx 0.825$ is determined using the discussions in Remark 1. For implementing iterative method (11) and the GSOR method, we need to solve linear systems with the coefficient matrices A and D. As already mentioned D is tridiagonal, therefore the solution of linear systems involving D is not an issue. Linear systems with A are also easy to solve, since it turns out that the Cholesky factorization of A (with the original ordering) does not incur any fill-in. Hence, we compute the Cholesky factorization $A = LL^T$ at the outset, and then perform back and forward substitutions each time the action of A^{-1} on a vector is required.

For applying the GSOR method, we further need to solve a saddle point problem of size $(n + m) \times (n + m)$ of the form

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix}. \tag{30}$$

The solution of (30) can be obtained in two steps as follows:

- **Step I.** Solve $(BA^{-1}B^T)w_2 = BA^{-1}r_1 - r_2$, to find w_2.
- **Step II.** Set $w_1 = A^{-1}(r_1 - B^T w_2)$.

As observed, for applying both iterative method (11) and the GSOR method, we further need to solve the linear systems with the coefficient matrix $BA^{-1}B^T$. To this end, the PCG method in conjunction with the approximate inverse preconditioner $BAB^T \approx (BA^{-1}B^T)^{-1}$ is used; see [1], where it is shown that this preconditioner results in very fast convergence independent of problem size. The inner tolerances for the PCG method are set to 10^{-3} and 10^{-5} in applying iterative method (11) and the GSOR method, respectively. Furthermore, we applied the iterative scheme (11) for solving the equivalent linear system of equations (19) in the same manner used for solving (1). The obtained results show that the convergence rate of the GGS method (11) with the augmented Lagrangian approach ($r > 0$) is mesh-independent and can be improved by increasing the value of r. We comment that for $r \neq 0$, we compute a sparse Cholesky factorization of $A + rB^T B$ with the symmetric approximate minimum degree (SYMAMD) reordering and using Remark 1, for $r = 2$ and $r = 2000$ in the GSS method, we obtain $\alpha \approx 0.9808$ and $\alpha \approx 0.9834$, respectively. The numerical results for the GGS method for different values of r are reported in Table 4. In order to show the effect of the value of α on the speed of

Table 4 Numerical results with the iterative method (11), liquid crystal problem ($\alpha = 0.82538$ and $\epsilon = 10^{-10}$)

	$r = 0$				$r = 2$				$r = 2000$			
Size	Iter	CPU	Iter_{pcg}	Err	Iter	CPU	Iter_{pcg}	Err	Iter	CPU	Iter_{pcg}	Err
2555	18	0.0116	44	0.1100e−06	8	0.0042	8	0.8458e−09	7	0.0043	7	0.2685e−08
5115	18	0.0201	44	0.2389e−06	7	0.0072	7	0.5956e−08	7	0.0081	7	0.2619e−08
10,235	18	0.0367	44	0.4794e−06	7	0.0133	7	0.4962e−08	7	0.0132	7	0.2639e−08
20,475	18	0.0725	44	0.9617e−06	7	0.0264	7	0.4757e−08	7	0.0282	7	0.2629e−08
40,955	18	0.1459	44	0.1917e−05	7	0.0536	7	0.4735e−08	6	0.0460	6	0.5672e−07
81,915	18	0.3043	44	0.3836e−05	6	0.1014	6	0.1056e−06	6	0.1011	6	0.5659e−07
163,835	18	0.6876	44	0.7681e−05	6	0.2189	6	0.1057e−06	6	0.2201	6	0.5677e−07
327,675	18	1.8087	44	0.1533e−04	6	0.5293	6	0.1049e−06	6	0.5271	6	0.5670e−07

Table 5 Numerical results with the iterative method (11), potential liquid crystal problem size 2555 ($\epsilon = 10^{-10}$)

	$\alpha = 0.8$				$\alpha = 1$			
r	Iter	CPU	Iter$_{pcg}$	Err	Iter	CPU	Iter$_{pcg}$	Err
0	21	0.0132	50	0.4384e−07	25	0.0144	52	0.5167e−07
2	12	0.0066	12	0.6738e−08	6	0.0039	6	0.1847e−07
2000	11	0.0060	11	0.2407e−07	5	0.0027	5	0.4628e−07

convergence of the GGS method, for the problem size 2555, we present the results of using two other values of α in Table 5.

As seen from the reported results, applying the GGS method for solving (19) with $r > 0$ gives better results than $r = 0$. Our experimental observations show that $r = 2$ leads to better results than $0 < r < 2$. The improvement of the convergence of the GSOR method for solving (19) is not significant with the augmented Lagrangian approach, so we do not report it here. For choosing a suitable value for ω, we used Remark 2, which yields an approximation of the optimal value of ω in terms of the spectral radius of $\mathscr{G}$, and obtained $\omega \approx 0.9597$ for the coarsest grid. As mentioned in Remark 2, the same value was used also for the finer grids. To clarify the performance of the GSOR method in terms of different values of ω, we reported the corresponding results for the GSOR method in Table 6 with respect to three different values of ω including $\omega = 0.9597$.

We conclude this section with some comments on how the performance of these methods compares with that of other possible solution approaches. For the potential fluid flow problem, it is possible to explicitly form the reduced (Schur complement) systems and to use standard PCG methods for their solution; see [10]. In [1], block diagonal and block triangular preconditioners based on Schur complement approximations were used to solve the potential fluid flow problem in conjunction with Krylov methods like MINRES and (F)GMRES. These approaches turn out to be very efficient, and are faster than the methods studied here for the potential fluid flow problem. This is ultimately due to the fact that for this problem the Schur complement matrices remain sparse and it is not difficult to find effective preconditioners for them. For the case of the liquid crystal problems, however, the situation is reversed. Here the Schur complement matrices are completely full and it is not easy to find effective preconditioners for them. Looking at the results reported in [1], Tables 4 and 5 (note that the computer used to obtain those timings is the same one that was used for this paper), we find that the stationary iterative schemes proposed in the present paper can be considerably faster than the more standard Krylov-based methods with approximate Schur complement preconditioners.

Table 6 Numerical results with GSOR iterative method, liquid crystal problem ($r = 0$ and $\epsilon = 10^{-7}$)

	$\omega = 0.8$				$\omega = 0.9597$				$\omega = 1.1$			
Size	Iter	CPU	Iter_{pcg}	Err	Iter	CPU	Iter_{pcg}	Err	Iter	CPU	Iter_{pcg}	Err
2555	11	0.0345	262	0.2541e−06	6	0.0206	132	0.1564e−06	8	0.0231	174	0.5458e−06
5115	11	0.0594	261	0.2506e−06	6	0.0384	142	0.1651e−06	8	0.0427	179	0.3095e−06
10,235	11	0.1173	285	0.2494e−06	6	0.0746	152	0.1601e−06	8	0.0824	197	0.4654e−06
20,475	11	0.2065	252	0.2501e−06	6	0.1428	149	0.1646e−06	8	0.1571	232	0.5394e−06
40,955	11	0.4729	293	0.2504e−06	6	0.2829	147	0.1637e−06	8	0.3246	191	0.5049e−06
81,915	11	1.0545	294	0.2442e−06	6	0.7262	176	0.1672e−06	8	0.8384	233	0.5275e−06
163,835	11	2.2489	260	0.2445e−06	6	1.3814	139	0.1654e−06	8	1.6866	196	0.4982e−06
327,675	11	6.6564	338	0.2496e−06	6	4.3709	167	0.1704e−06	8	4.5301	209	0.2930e−05

5 Conclusions

In this paper we have introduced and analyzed some Uzawa-type and block SOR-type stationary iterative schemes for solving large sparse linear systems in double saddle point form. These methods, possibly in combination with an augmented Lagrangian formulation, are able to achieve fast convergence when applied to linear systems arising in certain applications and therefore can be a valid alternative to Krylov subspace methods, especially on emerging hybrid and multicore architectures.

Acknowledgements We would like to thank Alison Ramage and Miroslav Tůma for providing the test problems used in the numerical experiments. We also express our sincere thank to two anonymous referees for their valuable comments and helpful suggestions. The second author is grateful for the hospitality of the Department of Mathematics and Computer Science at Emory University, where part of this work was completed.

References

1. Beik, F.P.A., Benzi, M.: Iterative methods for double saddle point systems. SIAM J. Matrix Anal. Appl. **39**, 602–621 (2018)
2. Benzi, M.: Preconditioning techniques for large linear systems: a survey. J. Comput. Phys. **182**, 417–477 (2002)
3. Benzi, M., Golub, G.H., Liesen, J.: Numerical solution of saddle point problems. Acta Numer. **14**, 1–137 (2005)
4. Benzi, M., Evans, T.M., Hamilton, S.P., Lupo Pasini, M., Slattery, S.R.: Analysis of Monte Carlo accelerated iterative methods for sparse linear systems. Numer. Linear Algebra Appl. **24**, e2008 (2017)
5. Boffi, D., Brezzi, F., Fortin, M.: Mixed Finite Element Methods and Applications. Springer Series in Computational Mathematics. Springer, New York (2013)
6. Elman, H.C., Golub, G.H.: Inexact and preconditioned Uzawa algorithms for saddle point problems. SIAM J. Numer. Anal. **31**, 1645–1661 (1994)
7. Hoemmen, M.: Communication-avoiding Krylov subspace methods. Doctoral Dissertation, University of California at Berkeley, Berkeley (2010)
8. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1985)
9. Lupo Pasini, M.: Convergence analysis of Anderson-type acceleration of Richardson's iteration, Preprint, Department of Mathematics and Computer Science, Emory University (2017)
10. Maryška, J., Rozložník, M., Tůma, M.: Schur complement systems in the mixed-hybrid finite element approximation of the potential fluid flow problem. SIAM J. Sci. Comput. **22**, 704–723 (2005)
11. Pratapa, P.P., Suryanarayana, P., Pask, J.E.: Anderson acceleration of the Jacobi iterative method: an efficient alternative to preconditioned Krylov methods for large, sparse linear systems. J. Comput. Phys. **306**, 43–54 (2016)
12. Pratapa, P.P., Suryanarayana, P., Pask, J.E.: Alternating Anderson-Richardson method: an efficient alternative to preconditioned Krylov methods for large, sparse linear systems. Comput. Phys. Commun. **234**, 278–285 (2019)
13. Ramage, A., Gartland, E.C., Jr.: A preconditioned nullspace method for liquid crystal director modeling. SIAM J. Sci. Comput. **35**, B226–B247 (2013)

14. Saad, S.: Iterative Methods for Sparse Linear Systems, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2003)
15. Stoyanov, M., Webster, C.: Numerical analysis of fixed point algorithms in the presence of hardware faults. SIAM J. Sci. Comput. **35**, C532–C553 (2015)
16. Walker, H.F., Ni, P.: Anderson acceleration for fixed-point iterations. SIAM J. Numer. Anal. **49**, 1715–1735 (2015)
17. Yamazaki, I., Rajamanickam, S., Boman, E.G., Hoemmen, M., Heroux, M., Tomov, S.: Domain decomposition preconditioners for communication-avoiding Krylov subspace methods on a hybrid CPU/GPU cluster. In: International Conference on High Performance Computing, Networking, Storage and Analysis, SC, pp. 933–944 (2015)
18. Young, D.M.: Iterative Solution or Large Linear Systems. Academic Press, New York (1971)

Generalized Block Tuned Preconditioners for SPD Eigensolvers

Luca Bergamaschi and Ángeles Martínez

Abstract Given an $n \times n$ symmetric positive definite (SPD) matrix A and an SPD preconditioner P, we propose a new class of generalized block tuned (GBT) preconditioners. These are defined as a p-rank correction of P with the property that arbitrary (positive) parameters $\gamma_1, \ldots, \gamma_p$ are eigenvalues of the preconditioned matrix. We propose to employ these GBT preconditioners to accelerate the iterative solution of linear systems like $(A - \theta I)\boldsymbol{s} = \boldsymbol{r}$ in the framework of iterative eigensolvers. We give theoretical evidence that a suitable, and effective, choice of the scalars γ_j is able to shift p eigenvalues of $P(A - \theta I)$ very close to one. Numerical experiments on various matrices of very large size show that the proposed preconditioner is able to yield an almost constant number of iterations, for different eigenpairs, irrespective of the relative separation between consecutive eigenvalues. We also give numerical evidence that the GBT preconditioner is always far superior to the spectral preconditioner (Numer. Linear Algebra Appl. 24(3):1–14, 2017), on matrices with highly clustered eigenvalues.

Keywords Eigenvalues · SPD matrix · Newton method · Tuned preconditioner · Incomplete Cholesky preconditioner

1 Introduction

Let A be a symmetric positive definite (SPD), large and sparse $n \times n$ matrix. We denote as $\lambda_1 \leq \lambda_2 \leq \ldots \lambda_m \ldots \leq \lambda_n$ its positive eigenvalues and $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m, \ldots \boldsymbol{v}_n$ the corresponding eigenvectors. The computation of the

L. Bergamaschi (✉)
Department of Civil, Environmental and Architectural Engineering, University of Padua, Padova, Italy
e-mail: luca.bergamaschi@unipd.it

Á. Martínez
Department of Mathematics "Tullio Levi-Civita", University of Padua, Padova, Italy
e-mail: angeles.martinez@unipd.it

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*, Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_12

$m \ll n$ leftmost eigenpairs of such a matrix is a common task in many scientific applications. Typical examples are offered by the vibrational analysis of mechanical structures [1], and the electronic structure calculations [2]. Computation of a few eigenpairs is also crucial in the approximation of the generalized inverse of the graph Laplacian [3, 4]. We also mention that approximate knowledge of the leftmost eigenpairs can be conveniently employed in the efficient solution of ill-conditioned linear systems [5].

Recently in [6], an efficiently preconditioned Newton method (DACG–Newton) has been developed which has proven to display comparable performances against the well-known Jacobi–Davidson (JD) method [7] and outperforms the implicitly restarted Lanczos (IRL) method with optimal tuned preconditioning [8] if a moderate number of eigenpairs are being sought.

The common feature to all these methods, when approaching the eigenpair $(\lambda_j, \boldsymbol{v}_j)$, is the need to solve a linear system like:

$$(A - \theta_j I)\boldsymbol{s} = \boldsymbol{r} \tag{1}$$

where $\theta_j \approx \lambda_j$, as needed in the shift-invert IRL method and also, implicitly in the DACG method, see [9], or to solve the projection of (1) in a subspace orthogonal to the previous iterate $\boldsymbol{u}_k$ and the previously computed eigenvectors, like:

$$J_k^{(j)}\boldsymbol{s} = -(A - \theta_j I)\boldsymbol{u}_k; \qquad \text{where} \tag{2}$$

$$J_k^{(j)} = (I - QQ^\top)(A - \theta_j I)(I - QQ^\top), \quad Q = \left[\boldsymbol{v}_1 \ldots \boldsymbol{v}_{j-1}\ \boldsymbol{u}_k\right] \tag{3}$$

(Newton and JD methods).

The idea of updating a given preconditioner with a low-rank matrix has been studied in a number of papers such as [10, 11]. In this chapter, we propose and develop a new preconditioning strategy for accelerating the solution of such linear system within the PCG method when accurately computing the eigenpair $(\lambda_j, \boldsymbol{v}_j)$ once a number of subsequent eigenpairs $(\tilde{\lambda}_s, \tilde{\boldsymbol{v}}_s), s = j+1, \ldots, j+p \equiv m$ are known to a (possibly) very rough accuracy. After collecting the approximate eigenpairs in a matrix V_j and choosing a suitable SPD diagonal matrix $\Gamma = \mathrm{diag}(\gamma_{j+1}, \ldots, \gamma_m)$, the *generalized block tuned* (GBT) preconditioner is defined as:

$$\widehat{P} = P + \text{low rank matrix}(V_j, \Gamma)$$

satisfying $\widehat{P}AV_j = V_j\Gamma$. We also develop a symmetric variant of such preconditioners and prove that there is an optimal choice of scalars γ_s, depending only on the approximate eigenvalues $\tilde{\lambda}_s$ which cluster close to one $m - j$ eigenvalues of $\widehat{P}(A - \theta_j I)$, or of $P_Q J_k^{(j)}$, with $P_Q = (I - QQ^\top)\widehat{P}(I - QQ^\top)$. Note that this GBT preconditioner can be viewed as an improvement of the spectral preconditioner proposed in [12], and particularly so in case of matrices with very close eigenvalues.

We experimentally test the proposed preconditioners in accelerating either the DACG method [13] or the Newton method (or simplified Jacobi–Davidson) after computing the inaccurate eigenpairs by the DACG method. In both cases, we obtain fast convergence when computing a small to moderate number of the leftmost eigenpairs of large SPD matrices. The action of the generalized tuned preconditioner is to weaken the dependence of the number of iterations in solving systems (1) or (2) on the relative separation of the eigenvalue being sought and the next higher one. Numerical results onto a number of medium- to large-size SPD matrices arising from Finite Element discretization of PDEs modelling groundwater flow in porous media, geomechanical processes in reservoirs, financial, and thermal processes, show the significant improvement provided by the GBT on all the test problems.

The outline of the chapter is as follows: In Sect. 2, we define the generalized tuned preconditioner and we theoretically prove the clustering of a number of eigenvalues of both $\widehat{P}(A - \theta_j I)$ and $P_Q J_k^{(j)}$. In Sect. 3, we give some implementation details. Section 4 provides numerical results of the proposed preconditioners onto a number of realistic and large-size test problems, while in Sect. 5 we give the main conclusions.

2 The Generalized Tuned Preconditioner

In [14], the *tuned* preconditioner is proposed in the framework of the iterative solution of the inner system within the inverse iteration (and the Rayleigh quotient iteration).

Definition 1 Given a preconditioner P and a vector x, a tuned preconditioner for matrix A is a matrix $\widehat{P}$ obtained by adding a rank-1 correction to P and satisfying

$$\widehat{P}A\boldsymbol{x} = \boldsymbol{x}. \tag{4}$$

An example of a tuned preconditioner can be found in [14]:

$$\widehat{P} = P - \frac{\boldsymbol{u}\boldsymbol{u}^\top}{\boldsymbol{u}^\top A\boldsymbol{x}}, \qquad \text{where} \qquad \boldsymbol{u} = PA\boldsymbol{x} - \boldsymbol{x}$$

This definition can be easily extended to multiple vectors:

Definition 2 Given a preconditioner P and an $n \times p$ matrix V with full column rank, a block tuned preconditioner for matrix A is a matrix $\widehat{P}$ obtained by adding a rank-p correction to P and satisfying

$$\widehat{P}AV = V. \tag{5}$$

The following example of block tuned preconditioner is taken from [8]:

$$\widehat{P} = P - Z\left(Z^\top AV\right)^{-1} Z^\top, \qquad \text{where} \qquad Z = PAV - V. \tag{6}$$

A block tuned preconditioner has the pleasant property that the p columns of matrix V are eigenvectors of the preconditioned matrix corresponding to eigenvalues equal to one. Tuned preconditioners have been used in [12] to accelerate the inner linear systems in the framework of the inexact Newton method (or simplified JD) when seeking the eigenpair $(\boldsymbol{v}_j, \lambda_j)$. In this case, the system to be solved has the form, with $\theta_j \approx \lambda_j$:

$$(A - \theta_j I)\boldsymbol{x} = \boldsymbol{b}, \tag{7}$$

and it has been proved that the tuned preconditioner has the effect to cluster the eigenvalues of the preconditioned matrix $\widehat{P}(A - \theta_j I)$, when the columns of V are orthonormal *approximate* eigenvectors of A:

$$V_j = \left[\tilde{\boldsymbol{v}}_{j+1} \ldots \tilde{\boldsymbol{v}}_m\right] \tag{8}$$

satisfying

$$A\tilde{\boldsymbol{v}}_s = \lambda_s \tilde{\boldsymbol{v}}_s + \mathbf{res}_s, \qquad \|\mathbf{res}_s\| \le \tau\lambda_s, \ s = j+1, \ldots, m. \tag{9}$$

In addition, we assume that $\tilde{\lambda}_s = \tilde{\boldsymbol{v}}_s^\top A\tilde{\boldsymbol{v}}_s > \lambda_s$ (and this is always true for eigensolvers that minimize the Rayleigh Quotient) and also that the same accuracy τ is fulfilled for the relative eigenvalue error, namely

$$\tilde{\lambda}_s - \lambda_s \le \tau\lambda_s,$$

which is again a reasonable hypothesis since for SPD matrices there holds: $\tilde{\lambda}_s = \tilde{\boldsymbol{v}}_s^\top A\tilde{\boldsymbol{v}}_s = \lambda_s + O(\|\tilde{\boldsymbol{v}}_s - \boldsymbol{v}_s\|^2)$.
We recall the following result stated in [12, Lemma 3.1]:

Lemma 1 *Let matrix V_j be as in (8), $\widehat{P}_j$ a block tuned preconditioner satisfying condition (5). In the hypothesis (9), each column of V_j, i.e., $\tilde{\boldsymbol{v}}_s$, $s = j+1, \ldots, m$, is an approximate eigenvector of $\widehat{P}_j(A - \theta_j I)$ corresponding to the approximate eigenvalue $1 - \dfrac{\theta_j}{\lambda_s} \approx 1 - \dfrac{\lambda_j}{\lambda_s}$. In particular, the following relation holds:*

$$\widehat{P}_j(A - \theta_j I)\tilde{\boldsymbol{v}}_s = \left(1 - \frac{\theta_j}{\lambda_s}\right)\tilde{\boldsymbol{v}}_s + \boldsymbol{\varepsilon}_s, \qquad \textit{with} \qquad \|\boldsymbol{\varepsilon}_s\| \le \tau\lambda_{j+1}\|\widehat{P}_j\|.$$

When solving a linear system by an iterative Krylov subspace method, it is clear that the *tuning* property is in some sense optimal as it provides a clustering of a number of eigenvalues of the preconditioned matrix at 1. However, Lemma 1 points

out this is not the case for eigenvalue computation. The effect of applying a tuned preconditioner to $A-\theta_j I$ is to set a number of eigenvalues of $\widehat{P}(A-\theta_j I)$ to a value that is close to one under the conditions that the eigenvalues are well separated, i.e., $\frac{\lambda_j}{\lambda_{j+1}} \ll 1$, which is not always the case on realistic problems.

In order to define a more effective preconditioner for systems like (7), we allow the preconditioned matrix $\widehat{P}A$ to have eigenvalues different from one corresponding to the columns of matrix V. We thus define a **generalized block tuned** (GBT) preconditioner:

Definition 3 Given a preconditioner P, an $n \times p$ matrix V with full column rank, and a diagonal matrix $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_p)$, a generalized block tuned preconditioner for matrix A is a matrix $\widehat{P}$ obtained by adding a rank-p correction to P and satisfying

$$\widehat{P}AV = V\Gamma. \tag{10}$$

As an example of a generalized block tuned preconditioner, we propose the generalization of (6) as:

$$\widehat{P} = P - Z\Pi^{-1}Z^\top, \quad \text{where} \quad Z = PAV - V\Gamma, \quad \text{and} \quad \Pi = Z^\top AV. \tag{11}$$

Note that the above preconditioner is not in general symmetric as small matrix Π is not and hence its use would prevent convergence either of the DACG eigensolver or the inner PCG iteration within the Newton method. However, this drawback can be circumvented when $V \equiv V_j$ represents the matrix of the (approximate) eigenvectors and $\Lambda_j = \mathrm{diag}(\tilde{\lambda}_{j+1}, \tilde{\lambda}_{j+2}, \ldots, \tilde{\lambda}_m)$. In such case, we can approximate Π as:

$$\Pi = V_j^\top APAV_j - \Gamma V_j^\top AV_j \approx V_j^\top APAV_j - \Gamma\Lambda_j = \tilde{\Pi}, \tag{12}$$

so restoring symmetry. This modified preconditioner:

$$\begin{aligned}\widetilde{P}_j &= P - Z\tilde{\Pi}^{-1}Z^\top = \\ &= P - (PAV_j - V_j\Gamma)\left(V_j^\top APAV_j - \Lambda_j\Gamma\right)^{-1}(PAV_j - V_j\Gamma)^\top \end{aligned} \tag{13}$$

does not satisfy exactly the tuning property since

$$\begin{aligned}\widetilde{P}_j AV_j &= PAV_j - Z\tilde{\Pi}^{-1}\left(\tilde{\Pi} + \Gamma\Lambda_j - \Gamma V_j^\top AV_j\right) \\ &= V_j\Gamma - Z\tilde{\Pi}^{-1}\Gamma(\Lambda_j - V_j^\top AV_j) = V_j\Gamma + \mathcal{E}. \end{aligned} \tag{14}$$

Writing (14) column-wise yields: $\widetilde{P}_j A\tilde{\boldsymbol{v}}_s = \gamma_s \tilde{\boldsymbol{v}}_s + \mathcal{E}_s$, having denoted with $\mathcal{E}_s$ the s-th column of $\mathcal{E}$. Finally, in view of (9) we have

$$\mathcal{E}_s = -Z\tilde{\Pi}^{-1}\Gamma(\tilde{\lambda}_s \boldsymbol{e}_s - V_j^\top A\tilde{\boldsymbol{v}}_s) = Z\tilde{\Pi}^{-1}\Gamma V_j^\top \mathbf{res}_s,$$

with $\boldsymbol{e}_s$ the $s-$th vector of the canonical basis, $\|\mathcal{E}_s\| \le \tau\lambda_s\alpha$, and $\alpha = \|Z\tilde{\Pi}^{-1}\Gamma\|$. The approximate generalized block tuned preconditioner therefore satisfies

$$\widetilde{P}_j A\tilde{\boldsymbol{v}}_s = \gamma_s \tilde{\boldsymbol{v}}_s + \mathcal{E}_s, \quad \|\mathcal{E}_s\| \le \tau\lambda_s\alpha, \qquad s = j+1, \ldots, m. \tag{15}$$

The following theorem states that it is possible to have p eigenvalues of the preconditioned matrix $\widetilde{P}_j(A - \theta_j I)$ very close to one depending on how the columns of matrix V approximate the eigenvectors of A. We assume to know $\tilde{\lambda}_j$, an approximation of the wanted eigenvalue λ_j. We also define the reciprocal of the relative separation between pairs of eigenvalues as:

$$\eta_s^{(j)} = \frac{\lambda_s}{\lambda_s - \lambda_j}, \quad s = j+1, \ldots, m; \tag{16}$$

$$\xi_j = \max_{s \ge j+1} \eta_s^{(j)} = \eta_{j+1}^{(j)} = \frac{\lambda_{j+1}}{\lambda_{j+1} - \lambda_j}. \tag{17}$$

Theorem 1 *Let matrix V_j be as in (8), $\widetilde{P}_j$ an approximate GBT preconditioner satisfying condition (15), with $\gamma_s = \tilde{\lambda}_s/(\tilde{\lambda}_s - \tilde{\lambda}_j)$, $s = j+1, \ldots, m$, then each column of V_j, i.e., $\tilde{\boldsymbol{v}}_s$, $s = j+1, \ldots, m$, is an approximate eigenvector of $\widetilde{P}_j(A - \theta_j I)$ corresponding to the approximate eigenvalue:*

$$\mu_s = \frac{\lambda_s - \theta_j}{\lambda_s} \frac{\tilde{\lambda}_s}{\tilde{\lambda}_s - \tilde{\lambda}_j}.$$

In particular, the following relation holds:

$$\widetilde{P}_j(A - \theta_j I)\tilde{\boldsymbol{v}}_s = \mu_s \tilde{\boldsymbol{v}}_s + \boldsymbol{\varepsilon}_s, \qquad \textit{with} \qquad \|\boldsymbol{\varepsilon}_s\| \le \tau \left(\lambda_s\alpha + \lambda_{j+1}\|\widetilde{P}_j\|\right).$$

Proof Since $\widetilde{P}_j$ is a generalized block tuned preconditioner, it satisfies (15). Moreover from (9), we have

$$\tilde{\boldsymbol{v}}_s = \frac{A\tilde{\boldsymbol{v}}_s}{\lambda_s} - \boldsymbol{g}, \quad \text{with} \quad \|\boldsymbol{g}\| \le \tau, \qquad \text{hence}$$

$$\begin{aligned}\widetilde{P}_j(A - \theta_j I)\tilde{\boldsymbol{v}}_s &= \widetilde{P}_j A\tilde{\boldsymbol{v}}_s - \theta_j \widetilde{P}_j \tilde{\boldsymbol{v}}_s \\ &= \widetilde{P}_j A\tilde{\boldsymbol{v}}_s - \theta_j \widetilde{P}_j \left(\frac{A\tilde{\boldsymbol{v}}_s}{\lambda_s} - \boldsymbol{g}\right)\end{aligned}$$

$$
\begin{aligned}
&= \left(1 - \frac{\theta_j}{\lambda_s}\right) \widetilde{P}_j A \tilde{\boldsymbol{v}}_s + \theta_j \widetilde{P}_j \boldsymbol{g} \\
&= \left(1 - \frac{\theta_j}{\lambda_s}\right) (\gamma_s \tilde{\boldsymbol{v}}_s + \mathcal{E}_s) + \theta_j \widetilde{P}_j \boldsymbol{g} \\
&= \frac{\lambda_s - \theta_j}{\lambda_s} \frac{\tilde{\lambda}_s}{\tilde{\lambda}_s - \tilde{\lambda}_j} \tilde{\boldsymbol{v}}_s + \left(1 - \frac{\theta_j}{\lambda_s}\right) \mathcal{E}_s + \theta_j \widetilde{P}_j \boldsymbol{g} \\
&= \mu_s \tilde{\boldsymbol{v}}_s + \boldsymbol{\varepsilon}_s
\end{aligned}
$$

where we have set $\boldsymbol{\varepsilon}_s = \left(1 - \frac{\theta_j}{\lambda_s}\right) \mathcal{E}_s + \theta_j \widetilde{P}_j \boldsymbol{g}$. Noting that

$$\|\boldsymbol{\varepsilon}_s\| \leq \tau \lambda_s \alpha + \tau \lambda_{j+1} \|\widetilde{P}_j\|$$

concludes the proof.

The eigenvalues μ_s are expected to be **very** close to one, depending on the initial tolerance τ. The bounds on the distance of μ_s from one are stated in the Corollary 1, which assumes as additional hypotheses that: (1) θ_j is closer to λ_j than $\tilde{\lambda}_j$ and (2) $\tau < (2\xi_j)^{-1}$. This last assumption implies that $\tilde{\lambda}_j$ is closer to λ_j than to λ_{j+1}. In fact:

$$\tilde{\lambda}_j - \lambda_j \leq \tau \lambda_j < \frac{1}{2} \frac{\lambda_{j+1} - \lambda_j}{\lambda_{j+1}} \lambda_j \leq \frac{\lambda_{j+1} - \lambda_j}{2},$$

and is also needed for the convergence of the simplified JD, see [15].

Corollary 1 *Let $\theta_j \in (\lambda_j, \tilde{\lambda}_j)$ and $\tau < (2\xi_j)^{-1}$, then the following bounds hold:*

$$1 - \tau \left(2\eta_s^{(j)} - 1\right) \leq \mu_s \leq 1 + 2\tau(\xi_j - 1), \qquad s = j+1, \ldots, m.$$

Proof First, the lower bound:

$$
\begin{aligned}
1 - \mu_s &\leq 1 - \frac{\lambda_s - \theta_j}{\tilde{\lambda}_s - \tilde{\lambda}_j} \\
&\leq 1 - \frac{\lambda_s - (1+\tau)\lambda_j}{\lambda_s(1+\tau) - \lambda_j} = \frac{\tau(\lambda_s + \lambda_j)}{\lambda_s(1+\tau) - \lambda_j} \leq \tau \frac{\lambda_s + \lambda_j}{\lambda_s - \lambda_j} = \tau \left(2\eta_s^{(j)} - 1\right).
\end{aligned}
$$

Hence, $\mu_s \geq 1 - \tau(2\eta_s^{(j)} - 1)$. Regarding the upper bound:

$$\begin{aligned}\mu_s &\leq \frac{\tilde{\lambda}_s - \theta_j}{\tilde{\lambda}_s - \tilde{\lambda}_j} = 1 + \frac{\tilde{\lambda}_j - \theta_j}{\tilde{\lambda}_s - \tilde{\lambda}_j} \\ &\leq 1 + \frac{\tau\lambda_j}{\tilde{\lambda}_s - \tilde{\lambda}_j} \leq 1 + \frac{\tau\lambda_j}{\lambda_{j+1} - \tilde{\lambda}_j} \leq 1 + \frac{2\tau\lambda_j}{\lambda_{j+1} - \lambda_j} = 1 + 2\tau(\xi_j - 1).\end{aligned}$$

Remark 1 From Corollary 1, it is clear that μ_s can be made arbitrarily close to one by appropriately reducing the tolerance τ. As an example, if $\xi_j = 10^2$, and $\tau = 10^{-3}$, then all μ_s are expected to be in $(0.8, 1.2)$.

The following theorem states a result, analogous to that of Theorem 1, which characterizes the eigenvalues of $P_Q J_k^{(j)}$, that is the preconditioned system matrix in the Newton phase. The proof of this theorem is not given here, for being quite similar to that of Lemma 3.1 and Theorem 3.1 of [12].

Theorem 2 *Let matrix $V_j = \left[\tilde{\mathbf{v}}_{j+1} \dots \tilde{\mathbf{v}}_m\right]$, $\widetilde{P}_j$ a generalized block tuned preconditioner, and $P_Q = (I - QQ^\top)\widetilde{P}_j(I - QQ^\top)$, then $(\tilde{\mathbf{v}}_s, \mu_s)$, $s = j+1, \dots, m$, is an approximate eigenpair of $P_Q J_k^{(j)}$ satisfying*

$$P_Q J_k^{(j)} \tilde{\mathbf{v}}_s = \mu_s \tilde{\mathbf{v}}_s + \mathbf{err}, \qquad \|\mathbf{err}\| \leq \tau\, C, \tag{18}$$

and $C \equiv C(\alpha, \tau, \|\widetilde{P}_j\|, \lambda_j, \lambda_{j+1})$ is increasing with respect to τ.

3 Algorithmic Issues

Efficient implementation of our generalized tuned preconditioner takes into account the following issues:

1. Limited memory implementation. We fix the maximum number of columns of matrix V_j, parameter $l_{\max}$.
2. Conversely, for assessing an eigenpair whose index j is close to m, the size of matrix V_j is too small, and too few eigenvalues are shifted from around zero to near to 1 and the preconditioner looses efficiency. To avoid this, we propose to compute an additional number (win) of approximated eigenpairs by the DACG procedure.

With these variants, in the computation of the j-th eigenpair we will use $V_j = \left[\tilde{\mathbf{v}}_{j+1} \dots \tilde{\mathbf{v}}_{j_{end}}\right]$ with $j_{end} = \min\{m + \text{ win}, l_{\max} + j\}$ to get the final expression for

our GBT preconditioner:

$$\begin{aligned} P &= (LL^\top)^{-1} \\ \widetilde{P}_j &= P - Z\tilde{\Pi}^{-1}Z^\top, \quad \text{with} \quad Z = PAV_j - V_j\Gamma \\ P_Q &= (I - QQ^\top)\widetilde{P}_j(I - QQ^\top) \end{aligned} \tag{19}$$

being $L = IC(A)$ an incomplete triangular Cholesky factor of A, with parameters LFIL, maximum fill-in of a row in L, and τ_{IC} the threshold for dropping small elements in the factorization. The construction (C) of $\widetilde{P}_j$ and its application (A) as $\widetilde{P}_j\boldsymbol{r}$ are sketched below. MVP = matrix–vector products, $m_j = j_{end} - j$, $Z_j = Z_0(:, j+1, j_{end})$ and $\Pi_j = \Pi_0(j+1 : j_{end}, j+1 : j_{end})$.

Phase	When	What	Relevant cost
C	Once and for all	• $Z_0 = PAV_0$ • $\Pi_0 = Z_0^T AV_0$	m MVP and m applications of P $m^2/2$ dot products.
C	For every eigenpair	• $Z = Z_j - V_j\Gamma$ • $\tilde{\Pi} = \Pi_j - \Gamma\Lambda_j$	m_j daxpys
A	At each iteration	• $\boldsymbol{h} = Z^T\boldsymbol{r}$ • $\boldsymbol{g} = \tilde{\Pi}\backslash\boldsymbol{h}$ • $\boldsymbol{w} = P\boldsymbol{r} - Z\boldsymbol{g}$	m_j dot products 1 system solve of size m_j 1 application of P, m_j daxpys

3.1 Repeated Application of the GBT Preconditioner

In principle, every eigenvalue solver may take advantage of the GBT preconditioner to update a given approximate inverse of A. In this chapter, we embed this preconditioner in the DACG–Newton method [12, Algorithm 2] also allowing to run twice the DACG solver: in the first run, a very rough approximation of the leftmost $m +$ win eigenpairs: $\tilde{\boldsymbol{v}}_1^{(0)}, \tilde{\boldsymbol{v}}_2^{(0)}, \ldots, \tilde{\boldsymbol{v}}_{m+\text{win}}^{(0)}$ is provided, satisfying $\|A\tilde{\boldsymbol{v}}_j^{(0)} - q(\tilde{\boldsymbol{v}}_j^{(0)})\tilde{\boldsymbol{v}}_j^{(0)}\| \leq \tau_1 q(\tilde{\boldsymbol{v}}_j^{(0)})$, with $\tau_1 > \tau$. All these approximate eigenvectors are used to form the GBT preconditioner to accelerate a second DACG run (up to tolerance τ) which provides $\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \ldots, \tilde{\boldsymbol{v}}_m$. These new approximations serve as starting points for the subsequent Newton scheme, while the set $\{\tilde{\boldsymbol{v}}_1, \tilde{\boldsymbol{v}}_2, \ldots, \tilde{\boldsymbol{v}}_m, \tilde{\boldsymbol{v}}_{m+1}^{(0)}, \ldots, \tilde{\boldsymbol{v}}_{m+\text{win}}^{(0)}\}$ is used for the projected GBT preconditioner updating.

4 Numerical Results

In this section, we provide numerical results where both the DACG and the DACG–Newton algorithms are tried for different values of the parameters for the GBT preconditioners.

We tested the proposed algorithm in the computation of the 20 smallest eigenpairs of a number of small to large matrices arising from various realistic applications. The CPU times (in seconds) refer to running a Fortran 90 code on a 2× Intel Xeon CPU E5645 at 2.40 GHz (six core) and with 4-GB RAM for each core.
The iteration is stopped whenever the following exit test on the relative eigenresidual is satisfied:

$$\frac{\|A\boldsymbol{u} - q(\boldsymbol{u})\boldsymbol{u}\|}{q(\boldsymbol{u})} \leq \varepsilon,$$

with $\varepsilon = 10^{-8}$. The parameters for the inner PCG solver within the DACG–Newton method were set to: $\tau_{PCG} = 10^{-2}$, $\text{ITMAX}_{PCG} = 20$.
The list of the selected problems together with their size n, and nonzero number nz is reported in Table 1. Some of the matrices are publicly available in the SuiteSparse Matrix Collection (SMC) at https://sparse.tamu.edu/. We also computed the fill-in σ of the initial preconditioner defined as the ratio between the nonzeros of L and the nonzeros of the lower triangular part of A.

4.1 Matrices with Clustered Small Eigenvalues

We analyze in detail the behavior of the proposed preconditioner in eigensolving two matrices having very clustered small eigenvalues, which represents the most challenging situation.

Matrix FINAN512

For this test case, the 20 smallest eigenvalues are much clustered, thus suggesting that the spectral preconditioner could not accelerate the iterative eigensolvers. We

Table 1 Main characteristics of the matrices used in the tests

Matrix	Source	SMC	n	nz	Initial preconditioner		
					LFIL	τ_{IC}	σ
MONTE-CARLO	Stochastic PDE	NO	77120	384320	20	10^{-3}	2.30
FINAN512	Financial problem	YES	74752	596992	10	10^{-1}	1.40
THERMOMEC	Thermal problem	YES	102158	711558	20	10^{-3}	0.94
MAT268515	Flow in porous media	NO	268515	3 926823	20	10^{-3}	2.67
EMILIA-923	Elasticity problem	YES	923136	41 005206	20	10^{-3}	1.86

Table 2 Eigenvalues λ_j and inverse of the relative separation ξ_j, for matrix FINAN512

j	λ_j	ξ_j	j	λ_j	ξ_j	j	λ_j	ξ_j	j	λ_j	ξ_j
1	0.94746	3.8E+2	6	1.03176	8.7E+2	11	1.05180	1.6E+2	16	1.07829	5.6E+4
2	0.95024	1.6E+1	7	1.03288	1.5E+2	12	1.05779	7.1E+1	17	1.07831	1.3E+3
3	1.01279	1.6E+2	8	1.03943	2.8E+2	13	1.07086	2.5E+2	18	1.07905	4.6E+2
4	1.01895	9.7E+1	9	1.04282	4.5E+3	14	1.07460	4.9E+2	19	1.08108	7.2E+2
5	1.02902	3.5E+2	10	1.04303	1.1E+2	15	1.07652	5.2E+2	20	1.08235	1.5E+2

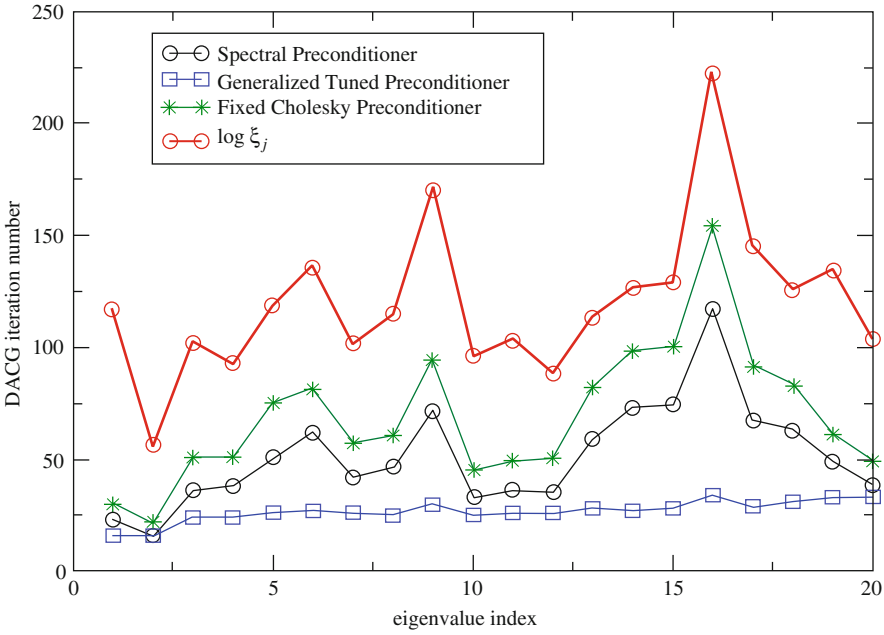


Fig. 1 Number of iterations for the second DACG run with various preconditioners. In red is the (scaled) logarithm of the indicator ξ_j

show in Table 2 the 20 smallest eigenvalues together with the reciprocal of the relative separation between consecutive eigenvalues, ξ_j, see (17).
We run first the DACG method with a tolerance $\tau = 4 \times 10^{-3}$ and then again DACG up to $\varepsilon = 10^{-8}$. In Fig. 1, we compare the number of iterations per eigenvalue index of the second DACG step, using the fixed Cholesky preconditioner, the spectral preconditioner as in [12] and the proposed GBT preconditioner. In the same figure, we also display the (scaled) $\log \xi_j$.

The iteration curves corresponding to the fixed and spectral preconditioners show a clear dependence on $\log \xi_j$. Moreover, the blue curve is almost constant so confirming the weakened dependence on the relative eigenvalue separation. We finally notice the great reduction of the iteration number obtained with the GBT preconditioner, irrespective of the eigenvalue index.

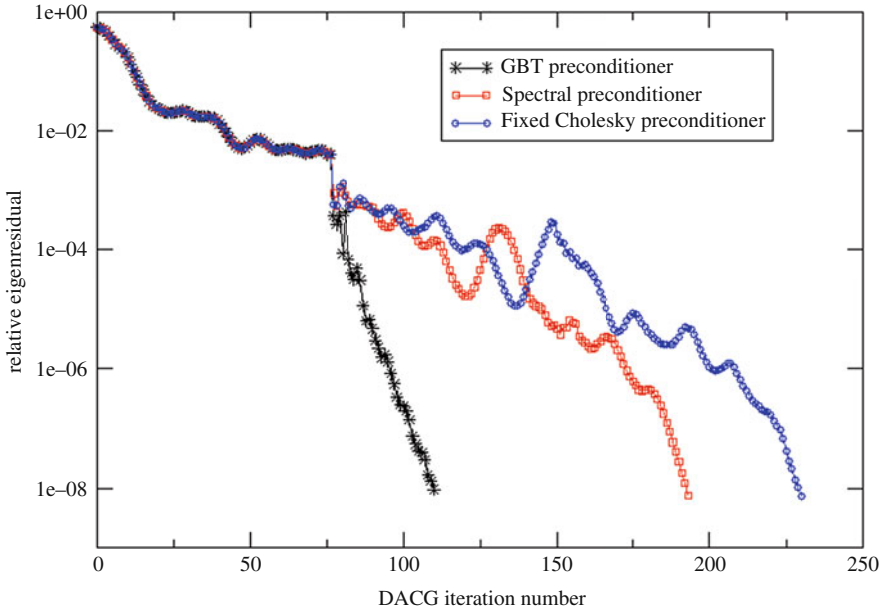


Fig. 2 Convergence profile of DACG using three different preconditioners (fixed, spectral, and GBT) in evaluating eigenpair # 16

We plot in Fig. 2 the convergence profile of the three preconditioners (fixed, spectral GBT) in evaluating eigenpair $(\lambda_{16}, \boldsymbol{v}_{16})$ which corresponds to the smallest relative separation, see Table 2. There is a constant portion of the three graphs which correspond to the first DACG run. Then, the steep convergence profile of the GBT preconditioner reveals the fastest convergence.

A similar acceleration is provided by the GBT preconditioner to the Newton phase, once approximate eigenvectors are evaluated by the DACG method. They will be used both as starting points for the Newton method and to form the GBT preconditioner. In Table 3, we report the results of both DACG and Newton-DACG methods with different preconditioners. The proposed GBT preconditioner is shown to accelerate both methods in terms of iteration number and CPU time, the combined DACG–Newton method being the most efficient method.

Matrix THERMOMEC

In this case also, the smallest eigenvalues are much clustered as accounted for by Table 4.

The results of the runs, summarized in Table 5, show that the GBT preconditioner is the most efficient one. To obtain convergence with the spectral preconditioner within the DACG–Newton method, it was necessary to enlarge the dimension of matrix V_j (win $= 10$) with however larger number of iterations in comparison with the GBT preconditioner. In Fig. 3, we plot the number of iterations taken by the Newton phase with the spectral and GBT preconditioners. In this case, we set the

Table 3 Timings and iterations for the DACG and DACG–Newton methods for the computation of $m = 20$ eigenpairs of matrix FINAN512

			DACG				Newton			Total	
							Iterations				
Prec.	Win	$l_{\max}$	τ_1	τ	$Its.$	CPU	OUT	Inner	CPU	MVP	CPU
Fixed	0	0	0.01	10^{-3}	1134	10.72	339	15372	153.13	16845	164.19
Spectral	5	10	0.01	10^{-3}	1212	12.21	76	1429	16.12	2717	28.52
Spectral	5	10	0.03	3×10^{-3}	953	9.64	[a]	[a]	[a]	[a]	[a]
GBT	5	10	0.01	10^{-3}	1053	10.72	48	528	6.45	1629	17.33
GBT	5	10	0.03	3×10^{-3}	833	8.58	86	634	7.75	**1525**	**16.57**
Fixed	0	0		10^{-8}	3217	29.97	–	–	–	3217	29.97
Spectral	5	10	4×10^{-3}	10^{-8}	2285	23.34	–	–	–	2285	23.34
Spectral	5	10	10^{-3}	10^{-8}	2541	25.82	–	–	–	2541	25.82
GBT	5	10	4×10^{-3}	10^{-8}	1789	18.36	–	–	–	**1789**	**18.36**
GBT	5	10	10^{-3}	10^{-8}	2100	21.77	–	–	–	2100	21.77

[a]No convergence

Table 4 Eigenvalues λ_j (all scaled by a factor 10^3) and inverse of the relative separation ξ_j, for matrix THERMOMEC

j	λ_j	ξ_j	j	λ_j	ξ_j	j	λ_j	ξ_j	j	λ_j	ξ_j
1	0.4540	3.27E+1	6	0.4892	1.65E+2	11	0.5087	1.62E+2	16	0.5201	2.30E+2
2	0.4688	1.57E+3	7	0.4922	1.10E+2	12	0.5119	4.12E+3	17	0.5224	4.24E+1
3	0.4691	4.40E+1	8	0.4967	8.49E+1	13	0.5120	1.36E+2	18	0.5353	2.28E+2
4	0.4802	9.54E+1	9	0.5027	1.54E+2	14	0.5158	1.37E+3	19	0.5376	1.18E+2
5	0.4854	1.29E+2	10	0.5060	1.93E+2	15	0.5162	1.35E+2	20	0.5423	2.23E+2

Table 5 Timings and iterations for the DACG and DACG–Newton methods for the computation of $m = 20$ eigenpairs of matrix THERMOMEC

			DACG				Newton			Total	
							Iterations				
Prec.	Win	$l_{\max}$	τ_1	τ	$Its.$	CPU	OUT	Inner	CPU	MVP	CPU
Fixed	0	0	10^{-3}	10^{-4}	1510	15.86	153	2628	34.12	4291	52.97
Spectral	10	10	0.01	10^{-4}	1533	16.81	51	838	12.67	2422	33.58
Spectral	10	10	0.01	10^{-3}	1335	14.89	137	2187	32.19	3659	51.48
GBT	5	10	0.01	10^{-3}	956	13.17	45	591	9.16	1592	22.50
GBT	5	10	0.03	10^{-3}	777	11.16	44	607	9.42	**1428**	**20.74**
Fixed	0	0		10^{-8}	2876	35.28	–	–	–	2876	35.28
Spectral	5	10	0.01	10^{-8}	2122	28.88	–	–	–	2122	28.88
Spectral	5	10	0.03	10^{-8}	2167	29.87	–	–	–	2167	29.87
GBT	5	10	0.01	10^{-8}	1580	21.87	–	–	–	1580	21.87
GBT	5	10	0.03	10^{-8}	1516	21.77	–	–	–	**1516**	**21.77**

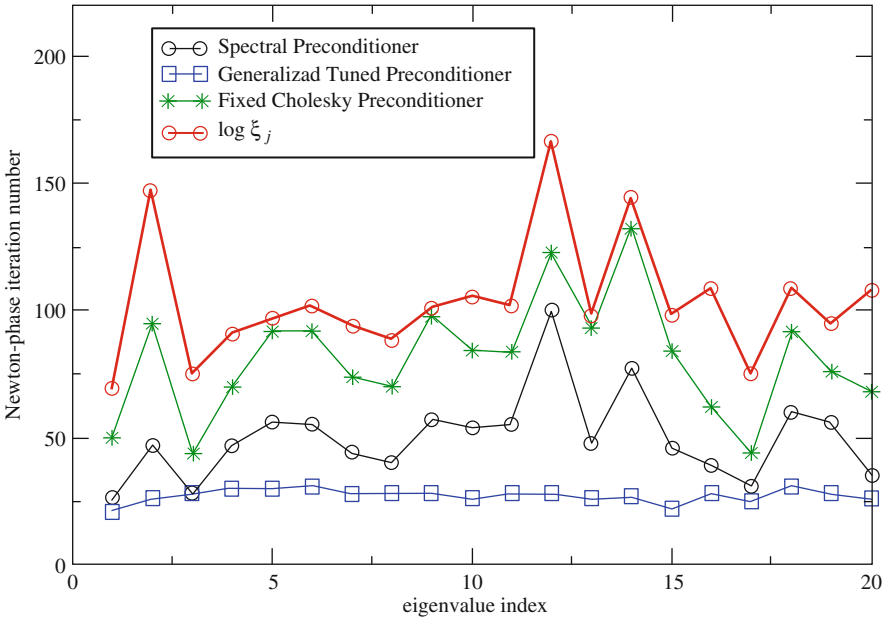


Fig. 3 Number of iterations for the Newton phase with fixed, spectral, and GBT preconditioners. In red is the (scaled) logarithm of the indicator ξ_j

Table 6 Comparisons between GBT–DACG (Newton) and Jacobi–Davidson

FINAN512				THERMOMEC			
Preconditioner	Method	MVP	CPU	Preconditioner	Method	MVP	CPU
GBT	DACG	1789	18.36	GBT	DACG	1580	21.87
GBT	DACG–Newton	1525	16.57	GBT	DACG–Newton	1428	20.74
Fixed	JD	2077	23.54	Fixed	JD	1947	29.38

second DACG tolerance $\tau = 5 \times 10^{-4}$. Larger tolerances prevented fast convergence of the spectral preconditioner. The almost constant GBT curve confirms the property of this preconditioner which makes the number of iterations nearly independent of the relative separation between eigenvalues.

We conclude this section by reporting (Table 6) the comparisons between the above methods and the Jacobi–Davidson method [7] with fixed Cholesky preconditioner and with minimum and maximum dimension of the search subspace set to 15 and 25, respectively, which revealed the most successful combination. Moreover, within the JD eigensolver, the PCG method for the inner linear systems has been efficiently implemented following [15]. Both DACG and DACG–Newton, with GBT preconditioners, prove faster than JD with fixed preconditioner.

Table 7 Timings and iterations for the DACG and DACG–Newton methods for the computation of $m = 20$ eigenpairs of matrix EMILIA-923

				DACG				Newton			Total	
								Iterations				
	Prec.	Win	$l_{\max}$	τ_1	τ	Its.	CPU	OUT	Inner	CPU	MVP	CPU
MONTE-CARLO	Fixed	0	0	0.02	10^{-3}	1565	13.98	135	2403	14.03	4103	36.61
	Spectral	5	20	0.2	10^{-3}	1120	11.05	38	500	5.98	1658	17.11
	GBT	5	10	0.2	10^{-3}	1063	10.75	48	521	6.39	1623	17.29
	Fixed	0	0		10^{-8}	3278	29.18	–	–	–	3278	29.18
	Spectral	5	10	0.1	10^{-8}	1738	17.00	–	–	–	1738	17.00
	GBT	5	10	0.1	10^{-8}	1669	16.58	–	–	–	**1669**	**16.58**
MAT268515	Fixed	0	0	0.02	0.2	655	35.08	98	1337	73.746	2090	110.36
	Spectral	5	10	0.2	0.02	619	33.32	55	588	38.39	1262	73.16
	GBT	5	10	0.2	0.02	588	33.04	60	592	39.24	1240	72.28
	Fixed	0	0		10^{-8}	2285	108.54	–	–	–	2285	108.54
	Spectral	5	10	0.1	10^{-8}	1387	72.30	–	–	–	1387	**72.30**
	GBT	5	10	0.1	10^{-8}	1357	72.65	–	–	–	**1357**	72.65
EMILIA-923	Fixed	0	0		10^{-3}	1761	515.56	182	3345	1044.46	5288	1570.94
	Spectral	5	10	0.2	10^{-3}	1436	438.30	47	641	233.39	2124	694.58
	GBT	5	10	0.2	10^{-3}	1345	409.69	55	566	214.38	1966	636.57
	Fixed	0	0		10^{-8}	3990	1176.64	–	–	–	3990	1176.64
	Spectral	5	10	0.2	10^{-8}	2162	698.33	–	–	–	2162	698.33
	GBT	5	10	0.2	10^{-8}	1921	628.12	–	–	–	**1921**	**628.12**

4.2 Summary of Results on the Remaining Matrices

We now report in Table 7 the results in eigensolving the other test matrices. The results show that either the spectral or the GBT preconditioners provide an important acceleration as compared with a fixed Cholesky preconditioner.

In all cases, the GBT preconditioner requires the smallest number of iterations to converge, as compared to the fixed and the spectral preconditioners. In particular, for the largest matrix EMILIA-923, the (GBT) DACG method reveals the most efficient one, halving the number of iterations required by the (Fixed) DACG variant.

5 Conclusions

A new generalized block tuned preconditioner $\widetilde{P}_j$ has been proposed and analyzed in order to accelerate the iterative (PCG) solution of the shifted linear systems like $(A - \theta_j I)\boldsymbol{u} = \boldsymbol{r}$, being θ_j an approximation of the sought eigenvalue. The action of the proposed preconditioner is theoretically proved to shift a number of the eigenvalues of $\widetilde{P}_j(A - \theta_j I)$ very close to one, by taking advantage of a rough approximation of subsequent eigenvectors. Numerical results onto matrices of large size arising from different models confirm the theoretical findings. Inserted

in the DACG, or in the Newton-DACG, methods, the GBT preconditioner provides a noteworthy acceleration of these iterative eigensolvers.

Acknowledgements This work has been supported by the Italian project CPDA155834/15: "Stable and efficient discretizations of the mechanics of faults" and by the Italian INdAM-GNCS Project *Metodi numerici per problemi di ottimizzazione vincolata di grandi dimensioni e applicazioni* (2017). We wish to thank the anonymous reviewers whose comments and suggestions helped improve the quality of the paper.

References

1. Bathe, K.J.: Finite Element Procedures in Engineering Analysis. Prentice-Hall, Englewood Cliffs (1982)
2. Saad, Y., Stathopoulos, A., Chelikowsky, J., Wu, K., Öğüt, S.: Solution of large eigenvalue problems in electronic structure calculations. BIT **36**(3), 563–578 (1996)
3. Bozzo, E., Franceschet, M.: Approximations of the generalized inverse of the graph Laplacian matrix. Internet Math. **8**, 456–481 (2012)
4. Bergamaschi, L., Bozzo, E.: Computing the smallest eigenpairs of the graph Laplacian. SeMA J. **75**, 1–16 (2018)
5. Bergamaschi, L., Facca, E., Martínez, A., Putti, M.: Spectral preconditioners for the efficient numerical solution of a continuous branched transport model. J. Comput. Appl. Math. (2018). https://doi.org/10.1016/j.cam.2018.01.022
6. Bergamaschi, L., Martínez, A.: Efficiently preconditioned inexact Newton methods for large symmetric eigenvalue problems. Optim. Methods Softw. **30**, 301–322 (2015)
7. Sleijpen, G.L.G., van der Vorst, H.A.: A Jacobi-Davidson method for linear eigenvalue problems. SIAM J. Matrix Anal. **17**(2), 401–425 (1996)
8. Martínez, A.: Tuned preconditioners for the eigensolution of large SPD matrices arising in engineering problems. Numer. Linear Algebra Appl. **23**(3), 427–443 (2016)
9. Bergamaschi, L., Gambolati, G., Pini, G.: Asymptotic convergence of conjugate gradient methods for the partial symmetric eigenproblem. Numer. Linear Algebra Appl. **4**(2), 69–84 (1997)
10. Freitag, M.A., Spence, A.: Rayleigh quotient iteration and simplified Jacobi-Davidson method with preconditioned iterative solves. Linear Algebra Appl. **428**(8–9), 2049–2060 (2008)
11. Carpentieri, B., Duff, I.S., Giraud, L.: A class of spectral two-level preconditioners. SIAM J. Sci. Comput. **25**(2), 749–765 (2003) (electronic)
12. Bergamaschi, L., Martínez, A.: Two-stage spectral preconditioners for iterative eigensolvers. Numer. Linear Algebra Appl. **24**(3), 1–14 (2017)
13. Bergamaschi, L., Putti, M.: Numerical comparison of iterative eigensolvers for large sparse symmetric matrices. Comput. Methods App. Mech. Eng. **191**(45), 5233–5247 (2002)
14. Freitag, M.A., Spence, A.: A tuned preconditioner for inexact inverse iteration applied to Hermitian eigenvalue problems. IMA J. Numer. Anal. **28**(3), 522–551 (2008)
15. Notay, Y.: Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric eigenproblem. Numer. Linear Algebra Appl. **9**(1), 21–44 (2002)

Stability of Gyroscopic Systems with Respect to Perturbations

Nicola Guglielmi and Manuela Manetta

Abstract A linear gyroscopic system is of the form:

$$M\ddot{x} + G\dot{x} + Kx = 0,$$

where the mass matrix M is a symmetric positive definite real matrix, the gyroscopic matrix G is real and skew symmetric, and the stiffness matrix K is real and symmetric. The system is stable if and only if the quadratic eigenvalue problem $\det(\lambda^2 M + \lambda G + K) = 0$ has all eigenvalues on the imaginary axis.

In this chapter, we are interested in evaluating robustness of a given stable gyroscopic system with respect to perturbations. In order to do this, we present an ODE-based methodology which aims to compute the closest unstable gyroscopic system with respect to the Frobenius distance.

A few examples illustrate the effectiveness of the methodology.

Keywords Stability of gyroscopic systems · Robust stability · Structured matrix nearness problems · Matrix ODEs

1 Introduction

Gyroscopic systems play an important role in a wide variety of engineering and physics applications, and vary from the design of urban structures (buildings, highways, and bridges), to aircraft industry, and to the motion of fluids in flexible pipes.

N. Guglielmi (✉)
Gran Sasso Science Institute, L'Aquila, Italy
e-mail: nicola.guglielmi@gssi.it

M. Manetta
Department of Mathematics and Computer Science, Emory University, Atlanta, GA, USA
e-mail: manuela.manetta@emory.edu

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_13

In its most general form, a gyroscopic system is modeled by means of a linear differential system on a finite-dimensional space, as follows:

$$M\ddot{x}(t) + (G + D)\dot{x}(t) + (K + N)x(t) = 0. \tag{1}$$

Here, $x(t)$ corresponds to the generalized coordinates of the system, $M = M^T$ represents the mass matrix, $G = -G^T$ and $K = K^T$ are related to gyroscopic and potential forces, $D = D^T$ and $N = -N^T$ are related to dissipative (damping) and nonconservative positional (circulatory) forces, respectively. Therefore, the gyroscopic system (1) is not conservative when D and N are nonzero matrices.

The stability of the system is determined by its associated quadratic eigenvalue problem:

$$M\lambda^2 + (G + D)\lambda + (K + N) = 0. \tag{2}$$

In particular, the system is said to be *strongly stable* if all eigenvalues of (2) lie in the open left half plane, *weakly stable* if all eigenvalues of (2) lie in the closed left half plane, that is, there is at least one pure imaginary eigenvalue and all such eigenvalues are semi-simple. It is unstable otherwise.

Although nonconservative systems are of great interest, especially in the context of nonlinear mechanics (see [8] for reference), this work is confined to conservative systems. Thus, the equation of motion is given by:

$$M\ddot{x}(t) + G\dot{x}(t) + Kx(t) = 0. \tag{3}$$

In particular, the spectrum of (2) is characterized by Hamiltonian symmetry. We note indeed that for any eigenvalue λ with a corresponding pair of left and right eigenvectors (y, x), that is:

$$(\lambda^2 M + \lambda G + K)x = 0, \qquad y^*(\lambda^2 M + \lambda G + K) = 0 \qquad (x, y \neq 0),$$

also $\overline{\lambda}$, $-\lambda$, $-\overline{\lambda}$ are eigenvalues with corresponding pairs of left and right eigenvectors $(\overline{y}, \overline{x})$, (x, y), $(\overline{x}, \overline{y})$, respectively.

Let us define the matrix pencil $\mathcal{Q}(\lambda) = M\lambda^2 + G\lambda + K$ such that the associated quadratic eigenvalue problem reads

$$\mathcal{Q}(\lambda)x = [M\lambda^2 + G\lambda + K]x = 0. \tag{4}$$

In the absence of gyroscopic forces, it is well known that the system $M\ddot{x}(t) + Kx(t) = 0$ is stable for K positive definite and unstable otherwise. When G is nonzero, then the system is weakly stable (see [11]) if the stiffness matrix K is positive definite, and may be unstable if $K \leq 0$ and K is singular. In the latter case, indeed, the 0 eigenvalue can be either semi-simple (thus the system is stable) or defective (unstable). Indeed, as numbers in the complex plane, the eigenvalues are symmetrically placed with respect to both the real and imaginary axes. This

property has two important consequences. On one hand, the eigenvalues can only move on the axis they belong to unless coalesce occurs; on the other hand, stability of system (3) only holds if all eigenvalues are purely imaginary.

Basically, for a conservative gyroscopic system, strong stability is impossible, since the presence of an eigenvalue on the left half plane would imply the existence of its corresponding symmetric one in the right half plane. The only possibility for the system to be stable is to be *marginally stable* (a particular case of weak stability), which requires that all eigenvalues lie on the imaginary axis, and the only way to lead the system to instability is a strong interaction (coalescence of two or more eigenvalues, necessary for them to leave the imaginary axis). The stiffness matrix K, for which no information about its signature is provided, plays a fundamental role in the stability of the system, and many stability results are available in the literature, based on the mutual relationship of G and K, as reported in [6, 7, 10] and references therein, and summarized in [12]. Given a marginally stable system of the form (3), the aim of this work is to find a measure of robustness of the system, that is the maximal perturbation that retains stability.

The paper is organized as follows. In Sect. 2, we phrase the problem in terms of structured distance to instability and present the methodology we adopt. In Sect. 3, we illustrate the system of ODEs for computing the minimal distance between pairs of eigenvalues. In Sect. 4, we derive a variational formula to compute the distance to instability. In Sect. 5, we present the method, and in Sect. 6, some experiments.

2 Distance to Instability

Distance to instability is the measure of the smallest additive perturbation which leads the system to be unstable. To estimate the robustness of (3), we will use the Frobenius norm. In order to preserve the Hamiltonian symmetry of the system, we will allow specific classes of perturbations. Indeed, gyroscopic forces and potential energy will be subject to additive skew-symmetric and symmetric perturbations, respectively. In [9], such a measure of robustness is called strong stability, which seems to be misleading according to the definitions in Sect. 1. Nevertheless, the author's aim was to find a neighboring system, that is an arbitrarily close system which retains stability and symmetry properties. Interesting results on stability are presented, allowing sufficiently small perturbations. However, our goal is to characterize these perturbations, and give a measure of "how small" they need to be to avoid instability. The distance to instability is related to the ε-pseudospectrum of the system.

In particular, we assume that M is fixed and we allow specific additive perturbations on G and K.

Therefore, let us define the structured ε-pseudospectrum of $[G, K]$ as follows:

$$\sigma_\varepsilon([G,K]) = \{\lambda \in \mathbb{C} : \lambda \in \sigma([G+\Delta G, K+\Delta K]) \text{ with } ||[\Delta G, \Delta K]||_F \le \varepsilon, \\ \text{for some skew-symmetric } \Delta G, \text{ and symmetric } \Delta K\}$$

We will call $\varepsilon^\star$ the sought measure, meaning that for every $\varepsilon < \varepsilon^\star$ the system remains marginally stable. Moreover, as mentioned in Sect. 1, the only way to lead the system to instability is a strong interaction, which means that at least two $\varepsilon^\star$-pseudoeigenvalues coalesce. Exploiting this property, we will compute the distance to instability in two phases: an outer iteration will change the measure ε of the perturbation, and an inner iteration will allow the ε-pseudoeigenvalues to move on the imaginary axis, according to the fixed ε, until determining the candidates for coalescence. The following remark suggests to limit our interest to systems in which the stiffness matrix is not positive definite.

Remark 1 When K is positive definite, the distance to instability of the system coincides with the distance to singularity of the matrix K, which is trivially equal to the absolute value of the smallest eigenvalue of K, because of the Hamiltonian symmetry.

2.1 Methodology

We make use of a two-level methodology.

First, we fix as ε the Frobenius norm of the admitted perturbation $[\Delta G, \Delta K]$. Then, given a pair of (close) eigenvalues λ_1, λ_2 on the imaginary axis, we look for the perturbations associated to a minimum of the distance $|\lambda_1 - \lambda_2|$ on the imaginary axis. This is obtained by integrating a suitable gradient system for the functional $|\lambda_1 - \lambda_2|$, preserving the norm of the perturbation $[\Delta G, \Delta K]$.

The external method controls the perturbation level ε to the aim of finding the minimal value ε^* for which λ_1 and λ_2 coalesce. The method is based on a fast Newton-like iteration.

Two-level iterations of a similar type have previously been used in [4, 5] for other matrix-nearness problems.

To formulate the internal optimization problem, we introduce the functional, for $\varepsilon > 0$:

$$f_\varepsilon(\Delta G, \Delta K) = \left|\lambda_1(\Delta G, \Delta K) - \lambda_2(\Delta G, \Delta K)\right| \tag{5}$$

where $\lambda_{1,2}(\Delta G, \Delta K)$ are the closest eigenvalues on the imaginary axis of the quadratic eigenvalue problem $\det(M\lambda^2 + (G + \Delta G)\lambda + (K + \Delta K)) = 0$.

Thus, we can recast the problem of computing the distance to instability as follows:

(1) For fixed ε, compute

$$[\Delta G(\varepsilon), \Delta K(\varepsilon)] \longrightarrow \min_{\Delta G, \Delta K : \|[\Delta G, \Delta K]\|_F = \varepsilon} f_\varepsilon(\Delta G, \Delta K) := f(\varepsilon) \tag{6}$$

with

$$\Delta G + \Delta G^T \quad \text{and} \quad \Delta K + \Delta K^T = 0. \tag{7}$$

(2) Compute

$$\varepsilon^* \longrightarrow \min_{\varepsilon > 0} \{\varepsilon : f(\varepsilon) = 0\}. \tag{8}$$

that means computing a pair $(\Delta G_*, \Delta K_*)$ of norm ε^* such that $\lambda_1(\varepsilon^*)$ is a double eigenvalue of the quadratic eigenvalue problem $\det(M\lambda^2 + (G + \Delta G_*)\lambda + (K + \Delta K_*)) = 0$.

2.2 Algorithm

In order to perform the internal minimization at (6), we locally minimize the functional $f_\varepsilon(\Delta G, \Delta K)$ over all $[\Delta G, \Delta K]$ of at most unit Frobenius norm, by integrating a steepest-descent differential equation (identifying the gradient system to the functional (5)) until a stationary point. The key instrument to deal with eigenvalue optimization is a classical variational result concerning the derivative of a simple eigenvalue of a quadratic eigenvalue problem.

In order to perform the minimization at (8), instead, denoting the minimum value of $f_\varepsilon(\Delta G, \Delta K)$ by $f(\varepsilon)$, we determine then the smallest perturbation $\varepsilon^\star > 0$ such that $f(\varepsilon^\star) = 0$, by making use of a quadratically convergent iteration.

Remark 2 Given a fixed ε, we compute all the possible distances between the eigenvalues, in order to identify the eigenpair which coalesces first (global optimum).

The whole method is summarized later by Algorithm 2.

3 The Gradient System of ODEs

In this section, the goal is to design a system of differential equations that, for a given ε, will find the closest pair of ε-pseudoeigenvalues on the imaginary axis. Indeed, this turns out to be a gradient system for the considered functional, which allows to obtain a useful monotonicity property along its analytic solutions.

To this intent, let us define the two-parameter operator:

$$Q(\tau, \lambda) = M\lambda^2 + G(\tau)\lambda + K(\tau). \tag{9}$$

Let $\lambda = \lambda(\tau)$, and let λ_0 satisfy the quadratic eigenvalue problem (4).

Assuming that λ_0 is a simple eigenvalue, then by Theorem 3.2 in [1]:

$$y_0^* \frac{\partial Q}{\partial \lambda} x_0 \neq 0,$$

where x_0 and y_0 are the right and left eigenvectors of Q at λ_0, respectively.

Under this assumption, therefore, by the variational result (5.3) in [1], the derivative of λ with respect to τ is well defined and given by:

$$\frac{d\lambda}{d\tau} = -\left(y_0^* \frac{\partial Q}{\partial \tau} x_0\right) \Big/ \left(y_0^* \frac{\partial Q}{\partial \lambda} x_0\right) \tag{10}$$

Next, let us consider the matrix-valued functions $G_\varepsilon(t) = G + \varepsilon \Delta G(t)$ and $K_\varepsilon(t) = K + \varepsilon \Delta K(t)$, where the augmented matrix $[\Delta G, \Delta K]$ satisfies (7), and

$$\|[\Delta G(t), \Delta K(t)]\|_F = 1 \quad \text{for all } t \in \mathbb{R}. \tag{11}$$

The corresponding quadratic eigenvalue problem is $Q_\varepsilon(t, \lambda)x = 0$, where:

$$Q_\varepsilon(t, \lambda) = M\lambda^2 + [G + \varepsilon \Delta G]\lambda + [K + \varepsilon \Delta K].$$

Moreover, let $\lambda_1(t) = \mathbf{i}\theta_1(t)$ and $\lambda_2(t) = \mathbf{i}\theta_2(t)$, with $\theta_1(t) > \theta_2(t)$ be two purely imaginary eigenvalues of $Q_\varepsilon(t, \lambda)x = 0$, corresponding to the eigenvalue of minimal distance of $Q_\varepsilon(t, \lambda)x = 0$.

Let $\lambda_1 = \mathbf{i}\theta_1$ and $\lambda_2 = \mathbf{i}\theta_2$ with $\theta_1, \theta_2 \in \mathbb{R}$.

Conventionally assume $\theta_1 > \theta_2$.

For $i = 1, 2$, let y_i such that

$$\gamma_i := y_i^* \left[2\mathbf{i}\theta_i M + (G + \varepsilon \Delta G)\right] x_i > 0 \tag{12}$$

be real and positive. This is naturally possible by suitably scaling the eigenvectors. Then, applying (10) gives

$$\begin{aligned} \dot{\theta}_1 - \dot{\theta}_2 &= \mathbf{i}\varepsilon \left[\frac{y_1^* \left(\mathbf{i}\theta_1 \dot{\Delta G} + \dot{\Delta K}\right) x_1}{\gamma_1} - \frac{y_2^* \left(\mathbf{i}\theta_2 \dot{\Delta G} + \dot{\Delta K}\right) x_2}{\gamma_2} \right] \\ &= \varepsilon \left\langle -\frac{\theta_1}{\gamma_1} y_1 x_1^* + \frac{\theta_2}{\gamma_2} y_2 x_2^*,\ \dot{\Delta G} \right\rangle + \varepsilon \left\langle -\frac{\mathbf{i}}{\gamma_1} y_1 x_1^* + \frac{\mathbf{i}}{\gamma_2} y_2 x_2^*,\ \dot{\Delta K} \right\rangle. \end{aligned} \tag{13}$$

where—for a pair of matrices A, B—we denote the Frobenius inner product:

$$\langle A, B\rangle = \text{trace}(A^* B).$$

The derivative of $[\Delta G(t), \Delta K(t)]$ must be chosen in the direction that gives the maximum possible decrease of the distance between the two closest eigenvalues, along the manifold of unitary Frobenius norm matrices $[\Delta G, \Delta K]$. Notice that constraint (11) is equivalent to

$$\left\langle [\Delta G, \Delta K], [\dot{\Delta G}, \dot{\Delta K}] \right\rangle = 0.$$

We have the following optimization result, which allows us to determine the constrained gradient of $f_\varepsilon(\Delta G, \Delta K)$.

Theorem 3 *Let $[\Delta G, \Delta K] \in \mathbb{R}^{n,2n}$ a real matrix of unit norm satisfying conditions (7)–(11), x_i and y_i right and left eigenvectors relative to the eigenvalues $\lambda_i = \boldsymbol{i}\theta_i$, for $i = 1, 2$, of $Q_\varepsilon(t, \lambda)x = 0$. Moreover, let γ_i, with $i = 1, 2$, be two real and positive numbers and consider the optimization problem:*

$$\min_{Z\in\Omega} \left\langle -\frac{\theta_1}{\gamma_1} y_1 x_1^* + \frac{\theta_2}{\gamma_2} y_2 x_2^*,\ Z_G \right\rangle + \left\langle -\frac{\boldsymbol{i}}{\gamma_1} y_1 x_1^* + \frac{\boldsymbol{i}}{\gamma_2} y_2 x_2^*,\ Z_K \right\rangle \tag{14}$$

with

$$\Omega = \left\{ \|Z\| = 1,\ \langle [\Delta G, \Delta K], Z\rangle = 0,\ Z_G \in \mathcal{M}_{\text{Skew}},\ Z_K \in \mathcal{M}_{\text{Sym}} \right\},$$

where $\mathcal{M}_{\text{Skew}}$ is the manifold of skew-symmetric matrices and $\mathcal{M}_{\text{Sym}}$ the manifold of symmetric matrices.

The solution $Z^\star = [Z_G^\star, Z_K^\star]$ of (14) is given by:

$$\mu Z^\star = \mu \left[Z_G^\star, Z_K^\star \right] = \left[f_G - \eta\, \Delta G,\ f_K - \eta\, \Delta K \right] \tag{15}$$

where $\mu > 0$ is a suitable scaling factor, and

$$\begin{aligned} \eta &= \Re \left\langle [\Delta G, \Delta K], [f_G, f_K] \right\rangle \\ f_G &= Skew\left(\Re \left[\frac{\theta_1}{\gamma_1} y_1 x_1^* - \frac{\theta_2}{\gamma_2} y_2 x_2^* \right] \right) \\ f_K &= Sym\left(\Im \left[\frac{1}{\gamma_2} y_2 x_2^* - \frac{1}{\gamma_1} y_1 x_1^* \right] \right) \end{aligned} \tag{16}$$

where Skew(B) *denotes the skew-symmetric part of B and* Sym(B) *denotes the symmetric part of B.*

Proof Preliminarily, we observe that for a real matrix:

$$B = \frac{B + B^{\mathrm{T}}}{2} + \frac{B - B^{\mathrm{T}}}{2} = \mathrm{Sym}(B) + \mathrm{Skew}(B)$$

the orthogonal projection (with respect to the Frobenius inner product) onto the manifolds $\mathcal{M}_{\mathrm{Sym}}$ of symmetric matrices and $\mathcal{M}_{\mathrm{Skew}}$ of skew-symmetric matrices are, respectively, $\mathrm{Sym}(B)$ and $\mathrm{Skew}(B)$. In fact:

$$\langle \mathrm{Sym}(B), Z \rangle = 0 \quad \text{for all } Z \in \mathcal{M}_{\mathrm{Skew}}$$

and

$$\langle \mathrm{Skew}(B), Z \rangle = 0 \quad \text{for all } Z \in \mathcal{M}_{\mathrm{Sym}}.$$

Looking at (14), we set the free gradients:

$$\phi_G = -\frac{\theta_1}{\gamma_1} y_1 x_1^* + \frac{\theta_2}{\gamma_2} y_2 x_2^* \quad \text{and} \quad \phi_K = -\frac{\mathbf{i}}{\gamma_1} y_1 x_1^* + \frac{\mathbf{i}}{\gamma_2} y_2 x_2^*.$$

The proof is obtained by considering the orthogonal projection (with respect to the Frobenius inner product) of the matrices (which can be considered as *vectors*) $-\phi_G$ and $-\phi_K$ onto the real manifold $\mathcal{M}_{\mathrm{Skew}}$ of skew-symmetric matrices and onto the real manifold $\mathcal{M}_{\mathrm{Sym}}$ of symmetric matrices, and further projecting the obtained rectangular matrix onto the tangent space to the manifold of real rectangular matrices with unit norm.

3.1 The System of ODEs

Following Theorem 3, we consider the following system of ODEs, where we omit the dependence of t:

$$\begin{cases} \dfrac{d}{dt}\Delta G = f_G - \eta\, \Delta G \\ \dfrac{d}{dt}\Delta K = f_K - \eta\, \Delta K \end{cases} \tag{17}$$

with η, f_G, and f_K as in (16).

This is a gradient system, which implies that the functional $f_\varepsilon(\Delta G(t), \Delta K(t))$ decreases monotonically along solutions of (17), until a stationary point is reached, which is generically associated to a local minimum of the functional.

4 The Computation of the Distance to Instability

As mentioned in Sect. 1, the only way to break the Hamiltonian symmetry is a strong interaction, that is two (or more) eigenvalues coalesce. This property allows us to reformulate the problem of distance to instability in terms of distance to defectivity (see [3]). In particular, since the matrices G and K must preserve their structure, we will consider a structured distance to defectivity. Because of the coalescence, we do not expect the distance between the eigenvalues to be a smooth function with respect to ε when $f_\varepsilon = 0$.

As an illustrative example, consider the gyroscopic system described by the equation:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \ddot{x}(t) + \begin{bmatrix} 0 & 3 \\ -3 & 0 \end{bmatrix} \dot{x}(t) - \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} x(t) = 0. \tag{18}$$

The minimal distance among the eigenvalue of this system is achieved by the conjugate pair closest to the origin, that is, $|\theta_1| = |\theta_2|$, and coalescence occurs at the origin, as shown in Fig. 1 (left).

Let us substitute the stiffness matrix in (18) with $-I$, that is:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \ddot{x}(t) + \begin{bmatrix} 0 & 3 \\ -3 & 0 \end{bmatrix} \dot{x}(t) - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(t) = 0. \tag{19}$$

Although $|\theta_1| = |\theta_2|$ still holds, strong interaction does not occur at the origin. Here, two pairs coalesce at the same time, as shown in Fig. 1 (right).

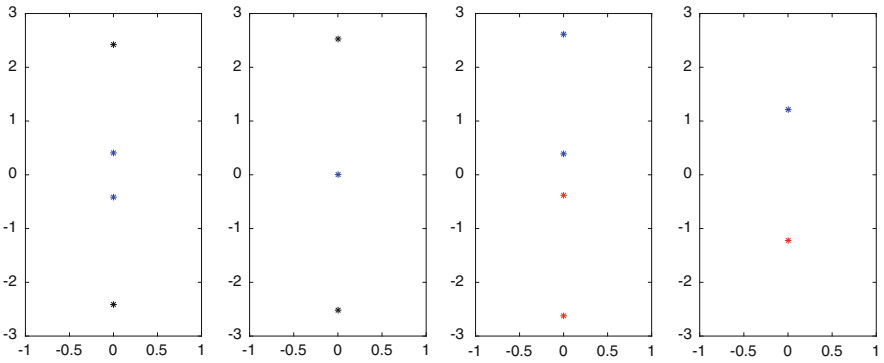


Fig. 1 Eigenvalues of system (18) on the left, before and at the moment of strong interaction at the origin. Eigenvalues of system (19) on the right: two strong interactions occur at the same time

4.1 Variational Formula for the ε-Pseudoeigenvalues with Respect to ε

We consider here the minimizers $\Delta G(\varepsilon)$ and $\Delta K(\varepsilon)$ computed as stationary points of the system of ODEs (17) for a given ε, and the associated eigenvalues $\lambda_i(\varepsilon) = \mathbf{i}\theta_i(\varepsilon)$ of the quadratic eigenvalue problem with $\varepsilon < \varepsilon^*$ (which implies $\theta_1(\varepsilon) \neq \theta_2(\varepsilon)$). We assume that all the abovementioned quantities are smooth functions with respect to ε, which we expect to hold generically.

Formula (10) is useful to compute the derivative of the ε-pseudoeigenvalues with respect to ε. We need the derivative of the operator Q w.r.t. ε, which appears to be given by:

$$\frac{\partial Q}{\partial \varepsilon} = \Delta G\lambda + \Delta K + \varepsilon(\Delta G'\lambda + \Delta K')$$

Here, the notation $A' = \frac{dA}{d\varepsilon}$ is adopted. Assuming that $\lambda = \lambda_0$ is a simple eigenvalue, and x_0 and y_0 are the right and left eigenvectors of Q at λ_0 respectively, then

$$\frac{\partial \lambda}{\partial \varepsilon} = -\frac{y_0^*(\Delta G\lambda + \Delta K + \varepsilon(\Delta G'\lambda + \Delta K'))x_0}{y_0^*(2M\lambda + G + \varepsilon\Delta G)x_0}$$

Claim $y_0^*(\Delta G'\lambda + \Delta K')x_0 = 0$.

The norm conservation $||[\Delta G, \Delta K]||_F = 1$, which is equivalent to $||\Delta G||_F^2 + ||\Delta K]||_F^2 = 1$, implies that $\langle \Delta G, \Delta G' \rangle = 0 = \langle \Delta K, \Delta K' \rangle$. Also:

$$\Re(y_0^*\lambda_0 \Delta G' x_0) = \Re(y_0^*\, \mathbf{i}\theta_0\, \Delta G' x_0) = \langle \Delta G', \Re(y_0\, x_0^*) \rangle = \langle \Delta G', \eta \Delta G \rangle = 0,$$

and

$$\Im(y_0^*\lambda_0 \Delta K' x_0) = \Im(y_0^*\, \Delta K' x_0) = \langle \Delta K', \Im(y_0\, x_0^*) \rangle = \langle \Delta K', \eta \Delta K \rangle = 0.$$

Therefore:

$$\frac{\partial \lambda}{\partial \varepsilon} = -\frac{y_0^*(\Delta G\lambda + \Delta K)x_0}{y_0^*(2M\lambda + G + \varepsilon\Delta G)x_0}$$

and

$$\theta_1' - \theta_2' = \frac{1}{\gamma_2}\Big[\theta_2 \Re(y_2^* \, \Delta G \, x_2) + \Im(y_2^* \, \Delta K \, x_2)\Big] - \frac{1}{\gamma_1}\Big[\theta_1 \Re(y_1^* \, \Delta G \, x_1) + \Im(y_1^* \, \Delta K \, x_1)\Big] \tag{20}$$

The previous expression provides $f'(\varepsilon)$. Hence, for $\varepsilon < \varepsilon^\star$ we can exploit its knowledge. Since generically coalescence gives rise to a defective pair on the imaginary axis, we have that the derivative of $f(\varepsilon)$ is singular at $\varepsilon^\star$.

Our goal is that of approximating $\varepsilon^\star$ by solving $f(\varepsilon) = \delta$ with $\delta > 0$ a sufficiently small number. For ε close to $\varepsilon^\star$, $\varepsilon < \varepsilon^\star$ we have generically (see [3])

$$\begin{cases} f(\varepsilon) = \gamma\sqrt{\varepsilon^\star - \varepsilon} + \mathcal{O}\big((\varepsilon^\star - \varepsilon)^{3/2}\big) \\ f'(\varepsilon) = -\dfrac{\gamma}{2\sqrt{\varepsilon^\star - \varepsilon}} + \mathcal{O}\big((\varepsilon^\star - \varepsilon)^{1/2}\big), \end{cases} \tag{21}$$

which corresponds to the coalescence of two eigenvalues. For an iterative process, given ε_k, we use formula (20) to compute $f'(\varepsilon)$ and estimate γ and $\varepsilon^\star$ by solving (21) with respect to γ and $\varepsilon^\star$. We denote the solution as γ_k and $\varepsilon_k^\star$, that is:

$$\gamma_k = \sqrt{2 f(\varepsilon_k)|f'(\varepsilon_k)|}, \qquad \varepsilon_k^\star = \varepsilon_k + \frac{f(\varepsilon_k)}{2|f'(\varepsilon_k)|} \tag{22}$$

and then compute

$$\varepsilon_{k+1} = \varepsilon_k^\star - \delta^2/\gamma_k^2. \tag{23}$$

An algorithm based on previous formulæ is Algorithm 2, which does not add any additional cost to the algorithm since the computation of $f'(\varepsilon_k)$ is very cheap.

Unfortunately, since the function $f(\varepsilon)$ is not smooth at $\varepsilon^\star$, and vanishes identically for $\varepsilon > \varepsilon^\star$, the fast algorithm has to be complemented by a slower bisection technique to provide a reliable method to approximate $\varepsilon^\star$.

5 The Complete Algorithm

The whole Algorithm 2 follows:

Algorithm 2: Algorithm for computing $\varepsilon^\star$

Data: tol > 0, $\delta > 0$, and $\varepsilon_0, \varepsilon_1, \varepsilon_u$ (such that $f(\varepsilon_0) > f(\varepsilon_1) >$ tol, and $f(\varepsilon_u) <$ tol).
Result: ε_f (approximation of $\varepsilon^\star$).
begin
1 Set Reject = False and $k = 1$.
2 **while** $|\varepsilon_k - \varepsilon_u| \geq tol$ **do**
3 **if** Reject = False. **then**
Store ε_k and $f(\varepsilon_k)$ into the memory.
4 Solve the system (17) and compute $[\Delta G(\varepsilon), \Delta K(\varepsilon)]$ and $f(\varepsilon) = f_\varepsilon(\Delta G(\varepsilon), \Delta K(\varepsilon))$.
5 Compute $\tilde{\varepsilon}_{k+1}$ by the formula (23).
6 **if** $\tilde{\varepsilon}_{k+1} > \varepsilon_u$ **then**
Set $\tilde{\varepsilon}_{k+1} = (\varepsilon_u + \varepsilon_k)/2$.
else
Set $\tilde{\varepsilon}_{k+1} = (\varepsilon_u + \varepsilon_k)/2$.
7 Compute $f(\tilde{\varepsilon}_{k+1})$ by integrating (17) with initial datum $[\Delta G(\varepsilon_k), \Delta K(\varepsilon_k)]$ (the minimizer associated to ε_k).
8 **if** $|f(\tilde{\varepsilon}_{k+1})| <$ tol **then**
Set Reject = True.
Set $\varepsilon_u = \tilde{\varepsilon}_{k+1}$.
else
Set Reject = False.
9 Set $\varepsilon_{k+1} = \tilde{\varepsilon}_{k+1}$.
10 Set $k = k + 1$.
11 Set $\varepsilon_f = \varepsilon_k$.

6 Numerical Experiments

We consider here some illustrative examples with $M = I$, from [2, 10, 13]. In the following, ε_u is chosen as the distance between the largest and the smallest eigenvalues, whereas $\varepsilon_0 = 0$ and ε_1 is obtained by (23).

6.1 Example 1

Let $G = \begin{bmatrix} 0 & -2 & 4 \\ 2 & 0 & -2 \\ -4 & 2 & 0 \end{bmatrix}$ and $K = \begin{bmatrix} 13 & 2 & 1 \\ 2 & 7 & 2 \\ 1 & 2 & 4 \end{bmatrix}$.

Also in this example, the stiffness matrix is positive definite, and the distance to singularity is $\varepsilon^\star = 3$ which coincides with the distance to instability.

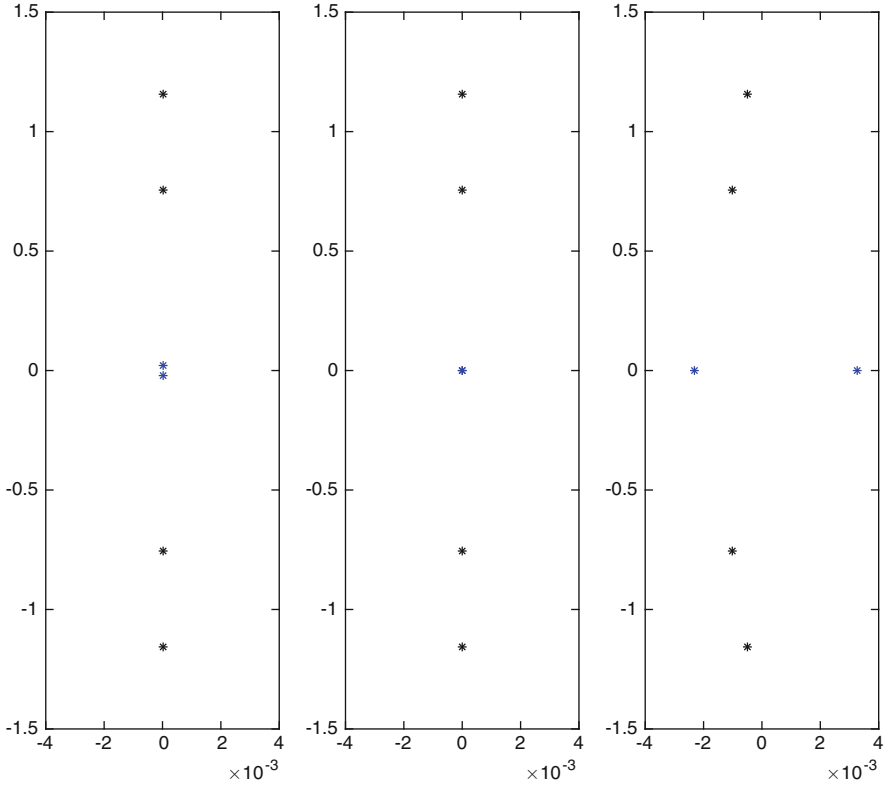


Fig. 2 A zoom-in, before, during, and after strong interaction for system (24)

6.2 Example 2

Let us consider the equation of motion $M\ddot{x}(t) + G\dot{x}(t) + Kx(t) = 0$, with:

$$M = \begin{bmatrix} 8 & -2 & 1 & 0 \\ -2 & 10 & 4 & 4 \\ 1 & 4 & 10 & -1.2 \\ 0 & 4 & -1.2 & 8 \end{bmatrix}, \; G = \begin{bmatrix} 0 & -16 & -8 & -12 \\ 16 & 0 & -40 & -12 \\ 8 & 40 & 0 & 16 \\ 12 & 12 & -16 & 0 \end{bmatrix}, \; K = \begin{bmatrix} 4 & -3 & 2 & 0 \\ -3 & 6 & 1 & -3 \\ 2 & 1 & 5 & -2 \\ 0 & -3 & -2 & 4 \end{bmatrix} \tag{24}$$

Here, the two closest eigenvalues of the system are the complex conjugate $\theta_1 = -\theta_2 = 2.1213e - 02$ and coalescence occurs at the origin, with $\varepsilon^\star = 4.6605e - 01$. Figure 2 illustrates these results. On the left, a zoom-in of the eigenvalues of system (24) near the origin is provided. In the center, coalescence occurs for the perturbed system $M\ddot{x}(t) + (G + \varepsilon^\star \Delta G)\dot{x}(t) + (K + \varepsilon^\star \Delta K)x(t) = 0$. On the right, the two eigenvalues become real after the strong interaction, namely, for $\varepsilon > \varepsilon^\star$, and the positive one leads the system to instability.

6.3 Example 3

This problem arises in the vibration analysis of a wiresaw. Let n be the dimension of the matrices. Let

$$M = I_n/2, \quad K = \mathrm{diag}_{1\le j\le n}(j^2\pi^2(1-v^2)/2)$$

and $G = (g_{jk})$ where $g_{jk} = \frac{4jk}{j^2-k^2}v$ if $j+k$ is odd, and 0 otherwise.
The parameter v is a real nonnegative number representing the speed of the wire. For $v \in (0,1)$, the stiffness matrix is positive definite. Here, we present two cases in which $v > 1$, and K is negative definite.

First, consider $n = 4$ and $v = 1.1$. Then, the system is marginally stable, and the distance to instability is given by $\varepsilon^\star = 4.6739e-02$. The eigenvalues $\mathbf{i}\theta_1 = \mathbf{i}3.4653$ and $\mathbf{i}\theta_2 = \mathbf{i}2.5859$ coalesce, as well as their respective conjugates.

Acknowledgements N. Guglielmi thanks the Italian M.I.U.R. and the INdAM GNCS for financial support and also the Center of Excellence DEWS.

References

1. Andrew, A.L., Chu, K.W.E., Lancaster, P.: Derivatives of eigenvalues and eigenvectors of matrix functions. SIAM. J. Matrix Anal. Appl. **14**(4), 903–926 (1993)
2. Beckte, T., Higham, N.J., Mehrmann, V., Schöder, C., Tisseur, F.: NLEVP: a collection of nonlinear eigenvalue problems. ACM Trans. Math. Softw. **39**(2), Article 7 (2013)
3. Buttà, P., Guglielmi, N., Manetta, M., Noschese, S.: Differential equations for real-structured (and unstructured) defectivity measures. SIAM J. Matrix Anal. Appl. **36**(2), 523–548 (2015)
4. Guglielmi, N., Overton, M.L.: Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix. SIAM J. Matrix Anal. Appl. **32**(4), 1166–1192 (2011)
5. Guglielmi, N., Kressner, D., Lubich, C.: Low rank equations for Hamiltonian nearness problems. Numer. Math. **129**(2), 279–319 (2015)
6. Huseyin, K., Hagedorn, P., Teschner, W.: On the stability of linear conservative gyroscopic systems. J. Appl. Maths. Phys. (ZAMP) **34**(6), 807–815 (1983)
7. Inman, D.J., Saggio III, F.: Stability analysis of gyroscopic systems by matrix methods. AIAA J. Guid. Control Dyn. **8**(1), 150–152 (1985)
8. Kirillov, O.N.: Destabilization paradox due to breaking the Hamiltonian and reversible symmetry. Int. J. Non Linear Mech. **42**(1), 71–87 (2007)
9. Lancaster, P.: Strongly stable gyroscopic systems. Electron. J. Linear Algebra **5**, 53–66 (1999)
10. Lancaster, P.: Stability of linear gyroscopic systems: a review. Linear Algebra Appl. **439**, 686–706 (2013)
11. Merkin, D.R.: Gyroscopic Systems. Gostekhizdat, Moscow (1956)
12. Seyranian, A., Stoustrup, J., Kliem, W.: On gyroscopic stabilization. Z. Angew. Math. Phys. **46**(2), 255–267 (1995)
13. Yuan, Y., Dai, H.: An inverse problem for undamped gyroscopic systems. J. Comput. Appl. Math. **236**, 2574–2581 (2012)

Energetic BEM for the Numerical Solution of 2D Hard Scattering Problems of Damped Waves by Open Arcs

Alessandra Aimi, Mauro Diligenti, and Chiara Guardasoni

Abstract The energetic boundary element method (BEM) is a discretization technique for the numerical solution of wave propagation problems, introduced and applied in the last decade to scalar wave propagation inside bounded domains or outside bounded obstacles, in 1D, 2D, and 3D space dimension.

The differential initial-boundary value problem at hand is converted into a space–time boundary integral equations (BIEs), then written in a weak form through considerations on energy and discretized by a Galerkin approach.

The paper will focus on the extension of 2D wave problems of hard scattering by open arcs to the more involved case of damped waves propagation, taking into account both viscous and material damping.

Details will be given on the algebraic reformulation of Energetic BEM, i.e., on the so-called time-marching procedure that gives rise to a linear system whose matrix has a Toeplitz lower triangular block structure.

Numerical results confirm accuracy and stability of the proposed technique, already proved for the numerical treatment of undamped wave propagation problems in several space dimensions and for the 1D damped case.

Keywords Damped waves · Energetic boundary element method · FFT

1 Introduction

A variety of engineering and physical applications, such as the propagation or the scattering of acoustic or electromagnetic waves, leads to the problem of solving linear hyperbolic partial differential equations (PDEs) in two- or three-dimensional space. These problems are normally considered in an unbounded homogeneous

All the authors are members of the INdAM-GNCS Research Group.

A. Aimi (✉) · M. Diligenti · C. Guardasoni
Dept. of Mathematical, Physical and Computer Sciences, University of Parma, Parma, Italy
e-mail: alessandra.aimi@unipr.it; mauro.diligenti@unipr.it; chiara.guardasoni@unipr.it

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_14

domain, and a method to tackle them is to reformulate the PDE as a boundary integral equation (BIE) on the usually bounded boundary of the domain, which can then be numerically solved using the boundary element method (BEM) [9, 17]. In some applications, the physically relevant data are given not by the solution in the interior of the domain but rather by the boundary values of the solution or its derivatives. These data can be obtained directly from the solution of BIEs, whereas it is well known that boundary values obtained from finite element method (FEM) solutions are in general not so accurate.

In the context of wave propagation, while the elastic forces tend to maintain the oscillatory motion, the transient effect dies out because of energy dissipations. The process of energy dissipation is generally referred to as damping. The analysis of damping phenomena that occur, for example, in mechanics, in fluid dynamics, and in semiconductors, is of particular interest [18, 20]: the dissipation is generated by the interaction between the waves and the propagation medium and it can be also closely related to the dispersion, as in the interactions between water streams and surface waves or in ferromagnetic materials. On the other side, in mechanical systems, in general, damping has the effect of reducing the amplitude of vibrations and, therefore, it is desirable to have some amount of damping in order to faster achieve stability. Hence, damping is whether an unavoidable presence in physical reality or a desired characteristic in industrial design.

The use of advanced numerical techniques to solve the related PDEs, such as FEMs and finite difference methods (FDMs), is well established and it is standard in this framework, while in the context of BEMs the analysis of dissipation through damped wave equation rewritten as a BIE is a relatively new topic, because it has been scarcely investigated until now. For the numerical solution of this kind of problems, one needs consistent approximations and accurate simulations even on large time intervals. Furthermore, as wave propagation phenomena are often observed in semi-infinite media (domain) where Sommerfeld radiation condition holds, a suitable numerical method has to ensure that this condition is not violated. For example, FEMs need the application of special techniques to fulfill this condition that, on the contrary, is implicitly fulfilled by BEM; hence a suitable coupling of both these techniques, when applicable, gives undoubted advantages.

In principle, both frequency-domain [13] and time-domain [7, 8] BEM can be used for hyperbolic initial-boundary value problems. Space–time BEM has the advantage that it directly yields the unknown time-dependent quantities. In this last approach, the construction of the BIEs, via representation formula in terms of single- and double-layer potentials, uses the fundamental solution of the hyperbolic partial differential equation and jump relations [15]. The mathematical background of time-dependent boundary integral equations is summarized by M. Costabel in [14].

For the numerical solution of the damped wave equation in 1D unbounded media, we have already considered in [5] the extension of the so-called space–time energetic BEM, introduced for the undamped wave equation in several space dimensions [1, 3, 4]. Energetic BEM comes from the discretization of a weak formulation based on energy arguments, directly expressed in the space–time domain, thus avoiding the use of the Laplace transform and of its inversion suggested in [15].

The analysis carried out for 1D damped wave propagation problems allowed to fully understand the approximation technique for what concerns marching on time, avoiding space integration with BEM singular kernels and it was considered as a touchstone for the extension to higher space dimensions, which is done here for the 2D case, taking into account hard scattering problems in unbounded domains, being soft scattering already treated in [6]. The chapter is structured as follows: at first, we present the differential model problem on an unbounded 2D domain and its energetic boundary weak formulation, and then we illustrate the consequent BEM discretization, highlighting numerical aspects of its algebraic reformulation. Significant numerical benchmarks are introduced and discussed, showing, from a numerical point of view, stability and accuracy of the obtained approximate solutions.

2 Model Problem and Its Weak Boundary Integral Formulation

We will consider the 2D Neumann problem for the damped wave equation in a bounded time interval $[0, T]$, exterior to an obstacle given by an open arc $\Gamma \subset \mathbf{R}^2$:

$$\left[\Delta u - \frac{1}{c^2} u_{tt} - \frac{2D}{c^2} u_t - \frac{P}{c^2} u\right](\mathbf{x}, t) = 0, \qquad \mathbf{x} \in \mathbf{R}^2 \setminus \Gamma,\ t \in (0, T] \quad (1)$$

$$u(\mathbf{x}, 0) = u_t(\mathbf{x}, 0) = 0, \qquad \mathbf{x} \in \mathbf{R}^2 \setminus \Gamma, \quad (2)$$

$$q(\mathbf{x}, t) := \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}, t) = \bar{q}(\mathbf{x}, t), \qquad \mathbf{x} \in \Gamma,\ t \in (0, T], \quad (3)$$

where c is the propagation velocity of a perturbation inside the domain, D and P are the viscous and material damping coefficients, respectively; $\mathbf{n}$ stands for the normal unit vector to Γ, and the datum $\bar{q}$ represents the opposite of the normal derivative of the incident wave along Γ, i.e., $\bar{q} = -\frac{\partial u^I}{\partial \mathbf{n}}$. In the acoustic framework, the exterior Neumann problem defines the scattering of a plane wave at a hard obstacle [21].

Remark 1 When $D = P = 0$, the given PDE collapses to the classical wave equation and the considered model problem can be also conceived, as stated in [10], as the scattering problem by a crack Γ in an unbounded elastic isotropic medium $\mathbf{R}^2 \setminus \Gamma$. Let Γ^- and Γ^+ denote the lower and upper faces of the crack, respectively, and $\mathbf{n}$ the normal unit vector to Γ oriented from Γ^- to Γ^+. As usual, the total displacement field can be represented as the sum of the incident field (the wave propagating without the crack) and the scattered field. In a 3D elastic isotropic medium, there are three plane waves propagating in a fixed direction: the primary wave P, the shear horizontal wave SH, and the vertical wave SV. The 2D antiplane problem corresponds to an incident SH wave, when all quantities are independent of the third component z (in particular, the crack has to be invariant with respect to z).

The scattered wave satisfies the Neumann problem (1)–(3) for the wave operator, where u stands for the third component of the displacement field.

Since we want to discretize the above problem using BEM, we have to rewrite it in a boundary integral form. This can be done using classical arguments and the knowledge of the fundamental solution of the 2D damped wave operator. Hence, we start writing the double-layer representation of the solution of (1)–(3):

$$u(\mathbf{x},t) = \int_\Gamma \int_0^t \frac{\partial G}{\partial \mathbf{n}_\mathbf{y}}(\mathbf{r}, t-\tau)\varphi(\mathbf{y},\tau)\, d\tau\, d\gamma_\mathbf{y}, \qquad \mathbf{x} \in \mathbf{R}^2 \setminus \Gamma,\ t \in (0,T], \tag{4}$$

where $\mathbf{r} := \mathbf{x} - \mathbf{y}$, the unknown density $\varphi = [u]_\Gamma$ represents the time history of the jump of u along Γ, and

$$G(\mathbf{x},t) = \begin{cases} \frac{c}{2\pi}\, e^{-Dt}\, \dfrac{\cos\left(\frac{\sqrt{P-D^2}}{c}\sqrt{c^2t^2-\|\mathbf{x}\|^2}\right)}{\sqrt{c^2t^2-\|\mathbf{x}\|^2}}\, H[ct-\|\mathbf{x}\|], & P \geq D^2 \\ \frac{c}{2\pi}\, e^{-Dt}\, \dfrac{\cosh\left(\frac{\sqrt{D^2-P}}{c}\sqrt{c^2t^2-\|\mathbf{x}\|^2}\right)}{\sqrt{c^2t^2-\|\mathbf{x}\|^2}}\, H[ct-\|\mathbf{x}\|], & P \leq D^2 \end{cases} \tag{5}$$

is the forward fundamental solution of the 2D damped wave operator, with $H[\cdot]$ the Heaviside distribution and $\|\cdot\|$ the Euclidean vector norm. Definition (5) switches from $\cos(\cdot)$ to $\cosh(\cdot)$ depending on the reciprocal magnitude of P and D^2: when $P > D^2$ we are in the so-called *underdamping* configuration, when $P < D^2$ we are in *overdamping* configuration, while the separation state $P = D^2$, referred to the vanishing of both $\cos(\cdot)$ and $\cosh(\cdot)$ arguments, is called *critical damping*. Note that in the limit for $D,\ P$ tending to 0, $G(\mathbf{x},t)$ tends to the fundamental solution of the 2D undamped wave operator, that is:

$$G_0(\mathbf{x},t) = \frac{c}{2\pi}\, \frac{H[ct-\|\mathbf{x}\|]}{\sqrt{c^2t^2-\|\mathbf{x}\|^2}}. \tag{6}$$

Defining the auxiliary kernel:

$$\tilde{G}(\mathbf{x},t) = \begin{cases} \frac{\sqrt{P-D^2}}{2\pi}\, e^{-Dt}\, \sin\left(\frac{\sqrt{P-D^2}}{c}\sqrt{c^2t^2-\|\mathbf{x}\|^2}\right) H[ct-\|\mathbf{x}\|], & P \geq D^2 \\ -\frac{\sqrt{D^2-P}}{2\pi}\, e^{-Dt}\, \sinh\left(\frac{\sqrt{D^2-P}}{c}\sqrt{c^2t^2-\|\mathbf{x}\|^2}\right) H[ct-\|\mathbf{x}\|], & P \leq D^2 \end{cases} \tag{7}$$

the representation formula (4) can be rewritten explicitly as:

$$u(\mathbf{x},t)=\int_{\Gamma}\int_{0}^{t}\frac{\mathbf{r}\cdot\mathbf{n}_{\mathbf{y}}}{r}\left\{G(\mathbf{r},t-\tau)\left[\frac{\varphi_{\tau}(\mathbf{y},\tau)+D\varphi(\mathbf{y},\tau)}{c}+\frac{\varphi(\mathbf{y},\tau)}{c(t-\tau)+r}\right]+\tilde{G}(\mathbf{r},t-\tau)\frac{\varphi(\mathbf{y},\tau)}{c(t-\tau)+r}\right\}d\tau\,d\gamma_{\mathbf{y}},\quad \mathbf{x}\in\mathbf{R}^2\setminus\Gamma,\ t\in(0,T],\ r:=\|\mathbf{r}\|. \tag{8}$$

Now, it is clear that if we want to recover the solution of the differential problem at any point outside the obstacle and at any time instant, we have to proceed with a post-processing phase provided that we know the density function $\varphi(\mathbf{x},t)$. To this aim, applying a directional (normal) derivative w.r.t. $\mathbf{x}$ in (4), performing a limiting process for $\mathbf{x}$ tending to Γ, and using the assigned Neumann boundary condition (3) we obtain the hypersingular space–time BIE:

$$\int_{\Gamma}\int_{0}^{t}\frac{\partial^2 G}{\partial\mathbf{n}_{\mathbf{x}}\partial\mathbf{n}_{\mathbf{y}}}(\mathbf{r},t-\tau)\varphi(\mathbf{y},\tau)\,d\tau\,d\gamma_{\mathbf{y}}=\bar{q}(\mathbf{x},t),\qquad \mathbf{x}\in\Gamma,\ t\in[0,T], \tag{9}$$

in the unknown $\varphi(\mathbf{x},t)$, which can be written with the compact notation:

$$\mathscr{D}\varphi=\bar{q}\,. \tag{10}$$

Problem (10) has been set in weak form. The so-called energetic weak formulation of (10) is defined similarly as in [3] and it can be deduced observing that, multiplying the PDE (1) by u_t, integrating over $[0,T]\times(\mathbf{R}^2\setminus\Gamma)$, and using integration by parts in space, one obtains that the energy $\mathscr{E}(u,T)$ of the solution u at the final time of analysis T, defined by:

$$\frac{1}{2}\int_{\mathbf{R}^2\setminus\Gamma}\left[\|\nabla_{\mathbf{x}}u(\mathbf{x},T)\|^2+\frac{1}{c^2}u_t^2(\mathbf{x},T)+\frac{P}{c^2}u^2(\mathbf{x},T)+\frac{4D}{c^2}\int_0^T u_t^2(\mathbf{x},t)dt\right]d\gamma_{\mathbf{x}} \tag{11}$$

can be rewritten as:

$$\mathscr{E}(u,T)=\int_{\Gamma}\int_{0}^{T}[u_t]_{\Gamma}(\mathbf{x},t)\,\frac{\partial u}{\partial\mathbf{n}_{\mathbf{x}}}(\mathbf{x},t)\,dt\,d\gamma_{\mathbf{x}}=\int_{\Gamma}\int_{0}^{T}\varphi_t(\mathbf{x},t)\,\mathscr{D}\varphi(\mathbf{x},t)\,dt\,d\gamma_{\mathbf{x}}\,. \tag{12}$$

Hence, projecting (10) by means of test functions ψ, derived w.r.t. time and belonging to the same functional space where we will search for the unknown density φ, we can write the energetic weak problem:

find $\varphi \in H^1([0,T];\, H_0^{1/2}(\Gamma))$ *such that*

$$\int_\Gamma \int_0^T (\mathscr{D}\varphi)(\mathbf{x},t)\psi_t(\mathbf{x},t)\,dt\,d\gamma_{\mathbf{x}} = \int_\Gamma \int_0^T \bar{q}(\mathbf{x},t)\psi_t(\mathbf{x},t)\,dt\,d\gamma_{\mathbf{x}}\,,$$
$$\forall \psi \in H^1([0,T];\, H_0^{1/2}(\Gamma))\,. \qquad (13)$$

Remark 2 The theoretical analysis of the quadratic form coming from the left-hand side of (13) was carried out for $P = D = 0$ in [3] where, under suitable hypothesis, coercivity was proved with some technicalities. This property allowed us to deduce stability and convergence of the related Galerkin approximate solution, which in this paper, for the case of nontrivial damping coefficients, will be verified from a numerical point of view.

3 Energetic BEM Discretization

We consider on the obstacle Γ, a boundary mesh constituted by $M_{\Delta x}$ straight elements $\{e_1, \cdots, e_{M_{\Delta x}}\}$, with $length(e_i) \le \Delta x$, $e_i \cap e_j = \emptyset$ if $i \neq j$ and such that $\bigcup_{i=1}^{M_{\Delta x}} \overline{e}_i$ coincides with $\overline{\Gamma}$, closure of Γ, if the obstacle is (piece-wise) linear, or is a suitable approximation of $\overline{\Gamma}$, otherwise. The functional background compels one to choose space shape functions belonging to $H_0^1(\Gamma)$; hence, we use standard piece-wise linear polynomial boundary element functions $w_j(\mathbf{x})$, $j = 1, \cdots, N_{\Delta x}$, with $N_{\Delta x} := M_{\Delta x} - 1$, suitably defined in relation to the introduced mesh over the obstacle and vanishing at the endpoints of Γ.

For time discretization, we consider a uniform decomposition of the time interval $[0,T]$ with time step $\Delta t = T/N_{\Delta t}$, $N_{\Delta t} \in \mathbf{N}^+$, generated by the $N_{\Delta t} + 1$ instants: $t_k = k\,\Delta t$, $k = 0, \cdots, N_{\Delta t}$, and we choose piece-wise linear time shape functions. Note that, for this particular choice, our shape functions, denoted by $v_k(t)$, $k = 0, \cdots, N_{\Delta t} - 1$, will be defined as:

$$v_k(t) = R[t - t_k] - 2R[t - t_{k+1}] + R[t - t_{k+2}]\,, \qquad (14)$$

where $R(t - t_k) := \frac{t - t_k}{\Delta t}\, H[t - t_k]$ is the ramp function. Hence, the approximate solution of the problem at hand will be expressed as:

$$\varphi(\mathbf{x},t) \simeq \sum_{k=0}^{N_{\Delta t}-1} \sum_{j=1}^{N_{\Delta x}} \alpha_j^{(k)}\, w_j(\mathbf{x})\, v_k(t). \qquad (15)$$

The Galerkin BEM discretization coming from energetic weak formulation (13) produces the linear system:

$$A\,\alpha = b\,, \qquad (16)$$

of order $N_{\Delta x} \cdot N_{\Delta t}$, where matrix A has a block lower triangular Toeplitz structure. Each block has dimension $N_{\Delta x}$. If we indicate with $A^{(\ell)}$ the block obtained when $t_h - t_k = \ell\, \Delta t,\ \ell = 0, \ldots, N_{\Delta t} - 1$, the linear system can be written as:

$$\begin{pmatrix} A^{(0)} & 0 & 0 & \cdots & 0 \\ A^{(1)} & A^{(0)} & 0 & \cdots & 0 \\ A^{(2)} & A^{(1)} & A^{(0)} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & 0 \\ A^{(N_{\Delta t}-1)} & A^{(N_{\Delta t}-2)} & \cdots & A^{(1)} & A^{(0)} \end{pmatrix} \begin{pmatrix} \alpha^{(0)} \\ \alpha^{(1)} \\ \alpha^{(2)} \\ \vdots \\ \alpha^{(N_{\Delta t}-1)} \end{pmatrix} = \begin{pmatrix} b^{(0)} \\ b^{(1)} \\ b^{(2)} \\ \vdots \\ b^{(N_{\Delta t}-1)} \end{pmatrix} \tag{17}$$

where: $\alpha^{(\ell)} = \left(\alpha_j^{(\ell)}\right)$ and $b^{(\ell)} = \left(b_j^{(\ell)}\right)$, $\ell = 0, \ldots, N_{\Delta t} - 1$, $j = 1, \ldots, N_{\Delta x}$.

The solution of (17) is obtained with a block forward substitution, i.e., at every time instant t_ℓ, we solve a reduced linear system of the type:

$$A^{(0)}\alpha^{(\ell)} = b^{(\ell)} - (A^{(1)}\alpha^{(\ell-1)} + \cdots + A^{(\ell)}\alpha^{(0)}). \tag{18}$$

Procedure (18) is a marching-on-time (MoT) technique, where the only matrix to be inverted once and for all is the symmetric, even if dense, non-singular $A^{(0)}$ diagonal block, while all the other blocks are used to update at every time step the right-hand side. Owing to this procedure, we can construct and store only the blocks $A^{(0)}, \cdots, A^{(N_{\Delta t}-1)}$ with a considerable reduction of computational cost and memory requirement. Let us finally note that for 2D problems blocks dimensions are typically low, as well as the condition number of block $A^{(0)}$. On the other side, the only drawback is the necessity of calculating, as discretization matrix elements, double integrals, involving hypersingular kernels in the space variables, as it happens for Galerkin BEM applied in the context of Neumann elliptic problems. Efficient quadrature schemes, used in this work, for numerical evaluation of these types of integrals are based on those described in [2], to which the interested reader is referred.

4 An FFT-Based Algorithm for MoT Computation

A reduction in computational cost, evaluating the block–vector products in the right-hand side of (18), can be obtained using an FFT-based algorithm as suggested in [16] and described here in detail. The interested reader is also referred to [12].

Definition Let $\mathbf{v} = [\mathrm{v}_0 \ldots \mathrm{v}_{m-1}]$, $\mathbf{w} = [\mathrm{w}_0 \ldots \mathrm{w}_{m-1}]$, $m \geq 1$, two vectors with the same length. We define their discrete circular convolution as a vector of components:

$$(\mathbf{v} * \mathbf{w})_q := \sum_{p=0}^{m-1} \mathrm{v}_{mod[m+q-p,m]}\, \mathrm{w}_p\,, \qquad q = 0, \ldots, m-1\,. \tag{19}$$

Note that we can equalize the discrete circular convolution to a matrix–vector product with a circulant matrix associated to the first vector of the convolution:

$$\mathbf{v} * \mathbf{w} = \begin{pmatrix} \mathrm{v}_0 & \mathrm{v}_{m-1} & \mathrm{v}_{m-2} & \cdots & \mathrm{v}_1 \\ \mathrm{v}_1 & \mathrm{v}_0 & \mathrm{v}_{m-1} & \cdots & \mathrm{v}_2 \\ \mathrm{v}_2 & \mathrm{v}_1 & \mathrm{v}_0 & \cdots & \mathrm{v}_3 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathrm{v}_{m-1} & \mathrm{v}_{m-2} & \mathrm{v}_{m-3} & \cdots & \mathrm{v}_0 \end{pmatrix} \begin{pmatrix} \mathrm{w}_0 \\ \mathrm{w}_1 \\ \mathrm{w}_2 \\ \cdots \\ \mathrm{w}_{m-1} \end{pmatrix} . \tag{20}$$

For the discrete circular convolution (19), the following result holds (see [19]):

$$\mathbf{v} * \mathbf{w} = \mathscr{F}^{-1}(\mathscr{F}(\mathbf{v})\mathscr{F}(\mathbf{w})) , \tag{21}$$

where $\mathscr{F}$ is the discrete Fourier transform (DFT). This allows to faster evaluate the convolution of two vectors with an FFT algorithm.

At every time step, for $\ell = 1, \cdots, N_{\Delta t} - 1$, we have to evaluate in (18) the ℓ-th component of the vector resulting from the product:

$$\begin{pmatrix} A^{(1)} & 0 & 0 & \cdots & 0 \\ A^{(2)} & A^{(1)} & 0 & \cdots & 0 \\ A^{(3)} & A^{(2)} & A^{(1)} & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ A^{(N_{\Delta t}-1)} & A^{(N_{\Delta t}-2)} & \cdots & A^{(2)} & A^{(1)} \end{pmatrix} \begin{pmatrix} \alpha^{(0)} \\ \alpha^{(1)} \\ \alpha^{(2)} \\ \vdots \\ \alpha^{(N_{\Delta t}-2)} \end{pmatrix} \tag{22}$$

The algorithm proceeds as follows.

At first, we compute $\alpha^{(0)}$, then we compute $\alpha^{(1)}$ having evaluated, in the right-hand side of (18), $A^{(1)}\alpha^{(0)} =: \mathbf{b}^{(1)}$. Afterwards, we can proceed observing that for the evaluation of the next two unknown vectors $\alpha^{(2)}$ and $\alpha^{(3)}$, we need to prepare the following terms at the right-hand side of (18):

$$\begin{aligned} &A^{(2)}\alpha^{(0)} + A^{(1)}\alpha^{(1)} \\ &A^{(3)}\alpha^{(0)} + A^{(2)}\alpha^{(1)} + A^{(1)}\alpha^{(2)} . \end{aligned} \tag{23}$$

Having at disposal $\alpha^{(0)}$ and $\alpha^{(1)}$, we can compute

$$\begin{pmatrix} \mathbf{b}^{(2)} \\ \mathbf{b}^{(3)} \end{pmatrix} := \begin{pmatrix} A^{(2)} & A^{(1)} \\ A^{(3)} & A^{(2)} \end{pmatrix} \begin{pmatrix} \alpha^{(0)} \\ \alpha^{(1)} \end{pmatrix} = \left[\begin{pmatrix} A^{(1)} & A^{(3)} & A^{(2)} \\ A^{(2)} & A^{(1)} & A^{(3)} \\ A^{(3)} & A^{(2)} & A^{(1)} \end{pmatrix} \begin{pmatrix} \alpha^{(0)} \\ \alpha^{(1)} \\ \mathbf{0} \end{pmatrix} \right]_{q=2,3} \tag{24}$$

where, here and in the following, $\mathbf{b}^{(j)} := \left(b_1^{(j)} \cdots b_{N_{\Delta x}}^{(j)}\right)^\top$, and intending that, in the product with the circulant matrix in the right-hand side, we will consider only

the second and the third block–vectors ($q = 2, 3$) and that $\mathbf{0}$ is a vector of zeros of dimension $N_{\Delta x}$.

The FFT, as suggested in (21), is point-wise performed to faster evaluate (24): for $i = 1, \ldots, N_{\Delta x}$, we get

$$\begin{aligned}\begin{pmatrix} b_i^{(2)} \\ b_i^{(3)} \end{pmatrix} &= \sum_{j=1}^{N_{\Delta x}} \left[\begin{pmatrix} A_{ij}^{(1)} & A_{ij}^{(3)} & A_{ij}^{(2)} \\ A_{ij}^{(2)} & A_{ij}^{(1)} & A_{ij}^{(3)} \\ A_{ij}^{(3)} & A_{ij}^{(2)} & A_{ij}^{(1)} \end{pmatrix} \begin{pmatrix} \alpha_j^{(0)} \\ \alpha_j^{(1)} \\ 0 \end{pmatrix} \right]_{q=2,3} = \sum_{j=1}^{N_{\Delta x}} \left[\left[A_{ij}^{(1)} \;\; A_{ij}^{(2)} \;\; A_{ij}^{(3)} \right] * \left[\alpha_j^{(0)} \;\; \alpha_j^{(1)} \;\; 0 \right] \right]_{q=2,3} \\ &= \sum_{j=1}^{N_{\Delta x}} \left[\mathscr{F}^{-1} \left(\mathscr{F} \left(\left[A_{ij}^{(1)} \;\; A_{ij}^{(2)} \;\; A_{ij}^{(3)} \right] \right) \mathscr{F} \left(\left[\alpha_j^{(0)} \;\; \alpha_j^{(1)} \;\; 0 \right] \right) \right) \right]_{q=2,3}\end{aligned} \tag{25}$$

and, with a suitable reordering, we recover $(\mathbf{b}^{(2)} \;\; \mathbf{b}^{(3)})^\top$; after having obtained $\alpha^{(2)}$, we can complete the computation in (23) in order to finally get $\alpha^{(3)}$.

Then, we can proceed observing that for the evaluation of the next four unknown vectors $\alpha^{(4)}$, $\alpha^{(5)}$, $\alpha^{(6)}$ and $\alpha^{(7)}$, we need to prepare the following terms at the right-hand side of (18):

$$\begin{aligned} &A^{(4)}\alpha^{(0)} + A^{(3)}\alpha^{(1)} + A^{(2)}\alpha^{(2)} + A^{(1)}\alpha^{(3)} \\ &A^{(5)}\alpha^{(0)} + A^{(4)}\alpha^{(1)} + A^{(3)}\alpha^{(2)} + A^{(2)}\alpha^{(3)} + A^{(1)}\alpha^{(4)} \\ &A^{(6)}\alpha^{(0)} + A^{(5)}\alpha^{(1)} + A^{(4)}\alpha^{(2)} + A^{(3)}\alpha^{(3)} + A^{(2)}\alpha^{(4)} + A^{(1)}\alpha^{(5)} \\ &A^{(7)}\alpha^{(0)} + A^{(6)}\alpha^{(1)} + A^{(5)}\alpha^{(2)} + A^{(4)}\alpha^{(3)} + A^{(3)}\alpha^{(4)} + A^{(2)}\alpha^{(5)} + A^{(1)}\alpha^{(6)} \end{aligned} \tag{26}$$

Having at disposal $\alpha^{(0)}$, $\alpha^{(1)}$, $\alpha^{(2)}$ and $\alpha^{(3)}$, we can fastly compute

$$\begin{pmatrix} \mathbf{b}^{(4)} \\ \mathbf{b}^{(5)} \\ \mathbf{b}^{(6)} \\ \mathbf{b}^{(7)} \end{pmatrix} := \begin{pmatrix} A^{(4)} & A^{(3)} & A^{(2)} & A^{(1)} \\ A^{(5)} & A^{(4)} & A^{(3)} & A^{(2)} \\ A^{(6)} & A^{(5)} & A^{(4)} & A^{(3)} \\ A^{(7)} & A^{(6)} & A^{(5)} & A^{(4)} \end{pmatrix} \begin{pmatrix} \alpha^{(0)} \\ \alpha^{(1)} \\ \alpha^{(2)} \\ \alpha^{(3)} \end{pmatrix} = \left[\begin{pmatrix} A^{(1)} & A^{(7)} & \ldots & A^{(2)} \\ A^{(2)} & A^{(1)} & \ldots & A^{(3)} \\ \ldots & \ldots & \ldots & \ldots \\ A^{(7)} & A^{(6)} & \ldots & A^{(1)} \end{pmatrix} \begin{pmatrix} \alpha^{(0)} \\ \vdots \\ \alpha^{(3)} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \right]_{q=4,\ldots,7} \tag{27}$$

using the FFT: for $i = 1 \dots N_{\Delta x}$, we get

$$\begin{aligned}
\begin{pmatrix} b_i^{(4)} \\ b_i^{(5)} \\ b_i^{(6)} \\ b_i^{(7)} \end{pmatrix} &= \sum_{j=1}^{N_{\Delta x}} \left[\begin{pmatrix} A_{ij}^{(1)} & A_{ij}^{(7)} & \dots & A_{ij}^{(2)} \\ A_{ij}^{(2)} & A_{ij}^{(1)} & \dots & A_{ij}^{(3)} \\ \dots & \dots & \dots & \dots \\ A_{ij}^{(7)} & A_{ij}^{(6)} & \dots & A_{ij}^{(1)} \end{pmatrix} \begin{pmatrix} \alpha_j^{(0)} \\ \vdots \\ \alpha_j^{(3)} \\ 0 \\ 0 \\ 0 \end{pmatrix} \right]_{q=4,\dots,7} \\
&= \sum_{j=1}^{N_{\Delta x}} \left[[A_{ij}^{(1)} \dots A_{ij}^{(7)}] * [\alpha_j^{(0)} \dots \alpha_j^{(3)} \ 0 \ 0 \ 0] \right]_{q=4,\dots,7} \\
&= \sum_{j=1}^{N_{\Delta x}} \left[\mathscr{F}^{-1}\left(\mathscr{F}([A_{ij}^{(1)} \dots A_{ij}^{(7)}]) \mathscr{F}([\alpha_j^{(0)} \dots \alpha_j^{(3)} \ 0 \ 0 \ 0]) \right) \right]_{q=4,\dots,7},
\end{aligned} \tag{28}$$

and, with a suitable reordering, we recover $(\mathbf{b}^{(4)} \dots \mathbf{b}^{(7)})^\top$, so that we can complete the computation in (26) using partial results of previous steps, in order to get, step by step, $\alpha^{(4)}$, $\alpha^{(5)}$, $\alpha^{(6)}$, and $\alpha^{(7)}$.

This strategy can be generalized in the following algorithm *AFFT*:
Let $N_{\Delta t}$ be a power of 2 and let $\overline{\nu}$ be such that $2^{\overline{\nu}+1} = N_{\Delta t}$. Then:

- *compute* $\alpha^{(0)}$,
- *for* $\nu = 1, \dots, \overline{\nu} + 1$ *compute:*

  i) *the matrix–vector product:*

$$\begin{aligned}
\begin{pmatrix} \mathbf{b}^{(2^{\nu-1})} \\ \mathbf{b}^{(2^{\nu-1}+1)} \\ \vdots \\ \mathbf{b}^{(2^{\nu}-1)} \end{pmatrix} &:= \begin{pmatrix} A^{(2^{\nu-1})} & \dots & A^{(2)} & A^{(1)} \\ A^{(2^{\nu-1}+1)} & \dots & A^{(3)} & A^{(2)} \\ \dots & \dots\dots & \dots & \\ A^{(2^{\nu}-1)} & \dots & A^{(2^{\nu-1}+1)} & A^{(2^{\nu-1})} \end{pmatrix} \begin{pmatrix} \alpha^{(0)} \\ \alpha^{(1)} \\ \vdots \\ \alpha^{(2^{\nu-1}-1)} \end{pmatrix} \\
&= \left[\begin{pmatrix} A^{(1)} & \dots & A^{(3)} & A^{(2)} \\ A^{(2)} & \dots & A^{(4)} & A^{(3)} \\ \dots & \dots\dots & \dots & \\ A^{(2^{\nu}-2)} & \dots & A^{(1)} & A^{(2^{\nu}-1)} \\ A^{(2^{\nu}-1)} & \dots & A^{(2)} & A^{(1)} \end{pmatrix} \begin{pmatrix} \alpha^{(0)} \\ \vdots \\ \alpha^{(2^{\nu-1}-1)} \\ \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} {\scriptstyle 2^{\nu-1}} \end{pmatrix} \right]_{q=2^{\nu-1},\dots,2^{\nu}-1}
\end{aligned}$$

that is, for $i = 1, \ldots, N_{\Delta x}$, the FFT of the discrete convolutions:

$$\begin{pmatrix} b_i^{(2^{\nu-1})} \\ b_i^{(2^{\nu-1}+1)} \\ \vdots \\ b_i^{(2^{\nu}-1)} \end{pmatrix} = \sum_{j=1}^{N_{\Delta x}} \left[\begin{pmatrix} A_{ij}^{(1)} & \cdots & A_{ij}^{(3)} & A_{ij}^{(2)} \\ A_{ij}^{(2)} & \cdots & A_{ij}^{(4)} & A_{ij}^{(3)} \\ \cdots & \cdots & \cdots & \cdots \\ A_{ij}^{(2^{\nu}-2)} & \cdots & A_{ij}^{(1)} & A_{ij}^{(2^{\nu}-1)} \\ A_{ij}^{(2^{\nu}-1)} & \cdots & A_{ij}^{(2)} & A_{ij}^{(1)} \end{pmatrix} \begin{pmatrix} \alpha_j^{(0)} \\ \vdots \\ \alpha_j^{(2^{\nu-1}-1)} \\ \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} 2^{\nu-1}-1 \end{pmatrix} \right]_{q=2^{\nu-1},\ldots,2^{\nu}-1} =$$

$$= \sum_{j=1}^{N_{\Delta x}} \left[\left[A_{ij}^{(1)} \ \ldots \ A_{ij}^{(2^{\nu}-1)} \right] * \left[\alpha_j^{(0)} \ \ldots \ \alpha_j^{(2^{\nu-1}-1)} \ \underbrace{0 \ldots 0}_{2^{\nu-1}-1} \right] \right]_{q=2^{\nu-1},\ldots,2^{\nu}-1} =$$

$$= \sum_{j=1}^{N_{\Delta x}} \left[\mathscr{F}^{-1} \left(\mathscr{F} \left(\left[A_{ij}^{(1)} \ \ldots \ A_{ij}^{(2^{\nu}-1)} \right] \right) \mathscr{F} \left(\left[\alpha_j^{(0)} \ \ldots \ \alpha_j^{(2^{\nu-1}-1)} \ \underbrace{0 \ldots 0}_{2^{\nu-1}-1} \right] \right) \right) \right]_{q=2^{\nu-1},\ldots,2^{\nu}-1}$$

with final reordering of elements,

ii) *the unknown vectors $\alpha^{(q)}$, for $q = 2^{\nu-1}, \ldots, 2^{\nu} - 1$, successively, using partial FFT results from the previous steps.*

Remark 3 Algorithm *AFFT* can be visualized as shown in Fig. 1, where the dimension of the square blocks:

$$\left[A^{(q-p)} \right]_{q=2^{\nu-1},\cdots,2^{\nu}-1;\ p=0,\cdots,2^{\nu-1}-1},$$

involved in the FFT computation is reported, for $\overline{\nu} = 3$. Note that triangles in the same figure can represent the numerical solution of the linear system in (18), to be done at each time step t_ℓ, for $\ell = 0, \cdots, N_{\Delta t} - 1$.

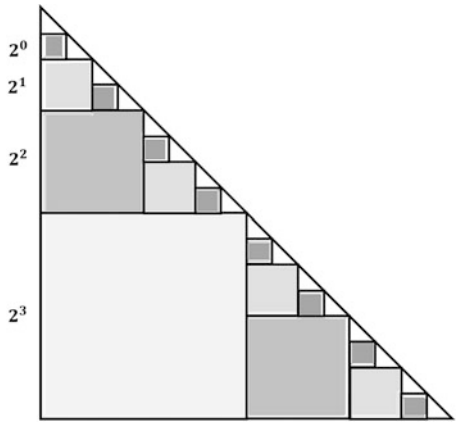


Fig. 1 Scheme of repartition of block-FFT computation in the algorithm *AFFT*, for $\overline{\nu} = 3$

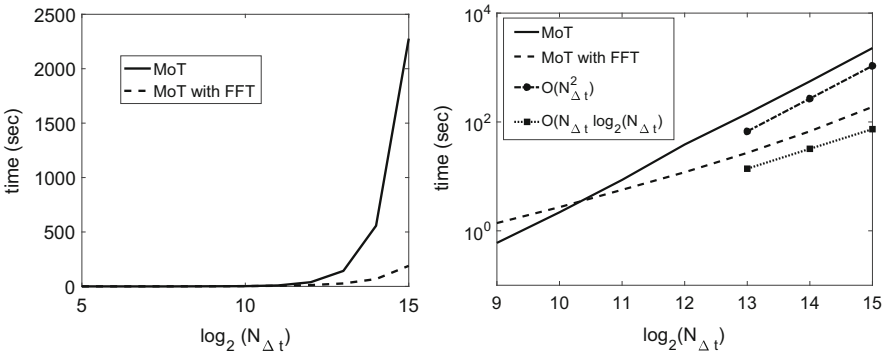


Fig. 2 Comparison between computational times of MoT and *AFFT*-based MoT, for different values of ν and for $N_{\Delta x} = 10$ on the left, with a zoom and theoretical slopes on the right

The above algorithm, which exploits at every iteration FFT results from the previous steps, applied to a simpler lower triangular matrix of order $2^{\overline{\nu}+1}$, has a computational cost of order $O(2^{\overline{\nu}+1}\log(2^{\overline{\nu}+1}))$, but for our problem everything is complicated by the fact that $A^{(q-p)}$ is not a scalar quantity but a matrix of order $N_{\Delta x}$, hence we have to apply the FFT $N_{\Delta x}^2$ times. Summarizing, in our case the number of arithmetical operations can be kept at the order $O(N_{\Delta x}^2 N_{\Delta t}\log(N_{\Delta t}))$ instead of $O(N_{\Delta x}^2 N_{\Delta t}^2)$, and the computational time saving is appreciable in case of large number of time steps, as shown in Fig. 2, where a comparison between computational times, on a standard laptop, of MoT and *AFFT*-based MoT, for different values of ν and for $N_{\Delta x} = 10$, is plotted. Note that the leading term of the computational cost of the proposed FFT-based MoT algorithm is coincident with the one findable for the Algorithm 4 in [11], related to a block Toeplitz matrix–vector product.

Alternatively, one can proceed with the inversion of the whole block lower triangular Toeplitz matrix A in (16), following specific Algorithm 5 proposed in [11], where computational cost is proved to be of order $O(N_{\Delta x}^3 N_{\Delta t} + N_{\Delta x}^2 N_{\Delta t}\log(N_{\Delta t}))$.

Remark 4 In general, there are FFT algorithms that keep their efficiency for any integer dimensions; however, for example, the Cooley–Tukey FFT algorithm [19] is optimized for vector dimensions equal to powers of 2, reason why we have chosen in the above description $N_{\Delta t} = 2^{\overline{\nu}+1}$.

5 Numerical Results

In the following, we present some numerical results obtained by the energetic BEM applied to the analysis of 2D damped waves hard scattering.

We consider the model problem (1)–(3) fixing $\Gamma = \{\mathbf{x} = (x, 0) \,|\, x \in [0, 1]\}$, $c = 1$, $[0, T] = [0, 5]$, and Neumann boundary datum, taken from [3], coming from an incident plane linear wave $u^I(\mathbf{x}, t)$ propagating in direction $\mathbf{k} = (\cos\theta, \sin\theta)$, that is:

$$\bar{q}(\mathbf{x}, t) = -\frac{\partial}{\partial \mathbf{n}_\mathbf{x}} f(t - \mathbf{k} \cdot \mathbf{x})\Big|_\Gamma \quad \text{with} \quad f(t) = 0.5\, t\, H[t]\ . \tag{29}$$

In this case, the Neumann datum (29) tends to the constant value $\bar{q}_\theta = 0.5 \sin\theta$, independent of time, when t tends to infinity, so we expect that the approximate transient solution $\varphi(\mathbf{x}, t)$ of BIE (9) on Γ will tend to the BIE solution related to a simpler elliptic PDE.

When $P = 0$, $D \geq 0$, we can discard in (1) the terms dependent on time and on P; we can therefore consider the following stationary BVP for the Laplace equation:

$$\begin{cases} \Delta u_\infty(\mathbf{x}) = 0, & \mathbf{x} \in \mathbf{R}^2 \setminus \Gamma \\ q_\infty(\mathbf{x}) = \bar{q}_\theta, & \mathbf{x} \in \Gamma \\ u_\infty(\mathbf{x}) = O(\|\mathbf{x}\|_2^{-1}), & \|\mathbf{x}\| \to \infty\,, \end{cases} \tag{30}$$

and the related BIE, whose analytical solution is explicitly known, it reads $\varphi_\infty(x) = \sin\theta \sqrt{x(1-x)}$ and it is shown in Fig. 3. Let's remark that this static solution remains the same for every value of D.

For an incident angle of $\pi/3$ and for discretization parameters fixed as $\Delta x = 0.05$ and $\Delta t = 0.05$, in Fig. 4 we show the approximate solution obtained by energetic BEM, at the final time instant of analysis $T = 5$, for $P = 0$ and different values of the viscous damping parameter $D = 0, 0.25, 1, 4$ (overdamping configuration). The higher the value of D, the higher the gap between transient and steady-state solutions, meaning that when we are in the presence of growing

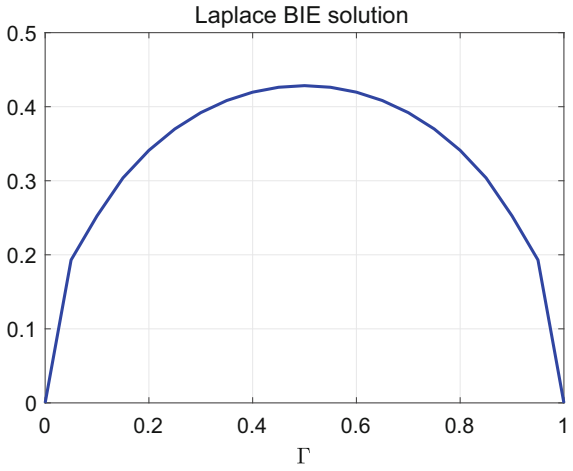


Fig. 3 BIE static solution on Γ related to the Laplace BVP (30)

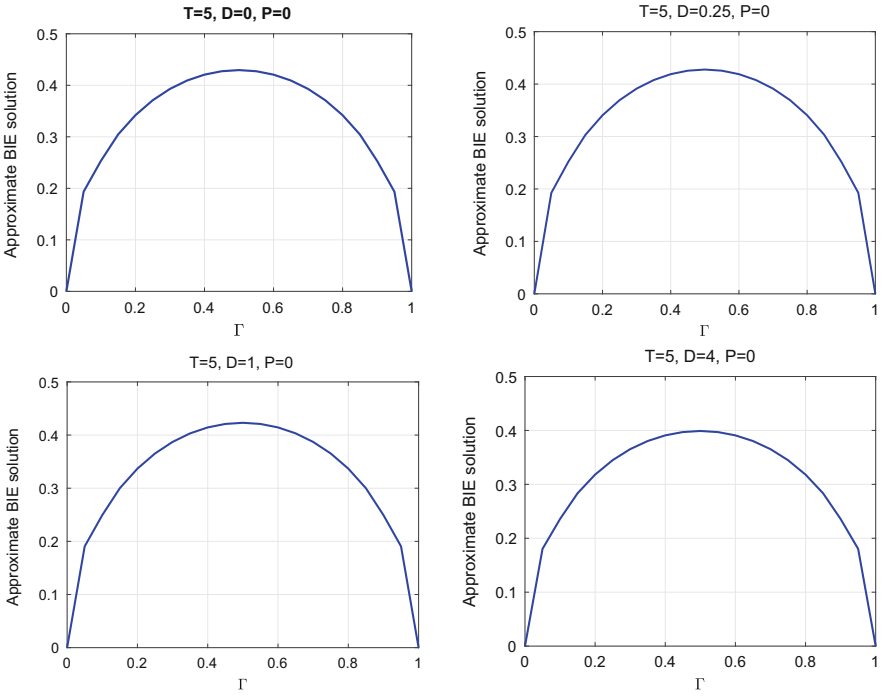


Fig. 4 $\varphi(\mathbf{x}, T)$ on Γ, for $P = 0$ and different values of D

viscosity, we would need more time to see the overlapping between the two corresponding plots.

When $P > 0$, we can discard in (1) the terms dependent on time; hence, for any value of $D \geq 0$, we can consider the following stationary BVP for the Helmholtz equation:

$$\begin{cases} \Delta u_\infty(\mathbf{x}) + k^2 u_\infty(\mathbf{x}) = 0, & \mathbf{x} \in \mathbf{R}^2 \setminus \Gamma \\ q_\infty(\mathbf{x}) = \bar{q}_\theta, & \mathbf{x} \in \Gamma \\ u_\infty(\mathbf{x}) = O(\|\mathbf{x}\|_2^{-1}), & \|\mathbf{x}\| \to \infty, \end{cases} \tag{31}$$

with $k^2 = -P$. The corresponding BIE solution $\varphi_\infty(\mathbf{x})$ assumes the same regularity of the steady-state solution related to the Laplace BVP and, of course, it changes for different values of material damping coefficient P. Again, for an incident angle of $\pi/3$ and for discretization parameters fixed as $\Delta x = 0.05$ and $\Delta t = 0.05$, in Fig. 5 we show the approximate solution obtained by energetic BEM, at the final time instant of analysis $T = 5$, for $D = 0$ and different values of the material damping parameter $P = 0.25, 1, 4$ (underdamping configuration). The higher the value of P, the smaller the maximum value of the transient and steady-state solutions. The accordance between the corresponding plots is perfectly visible.

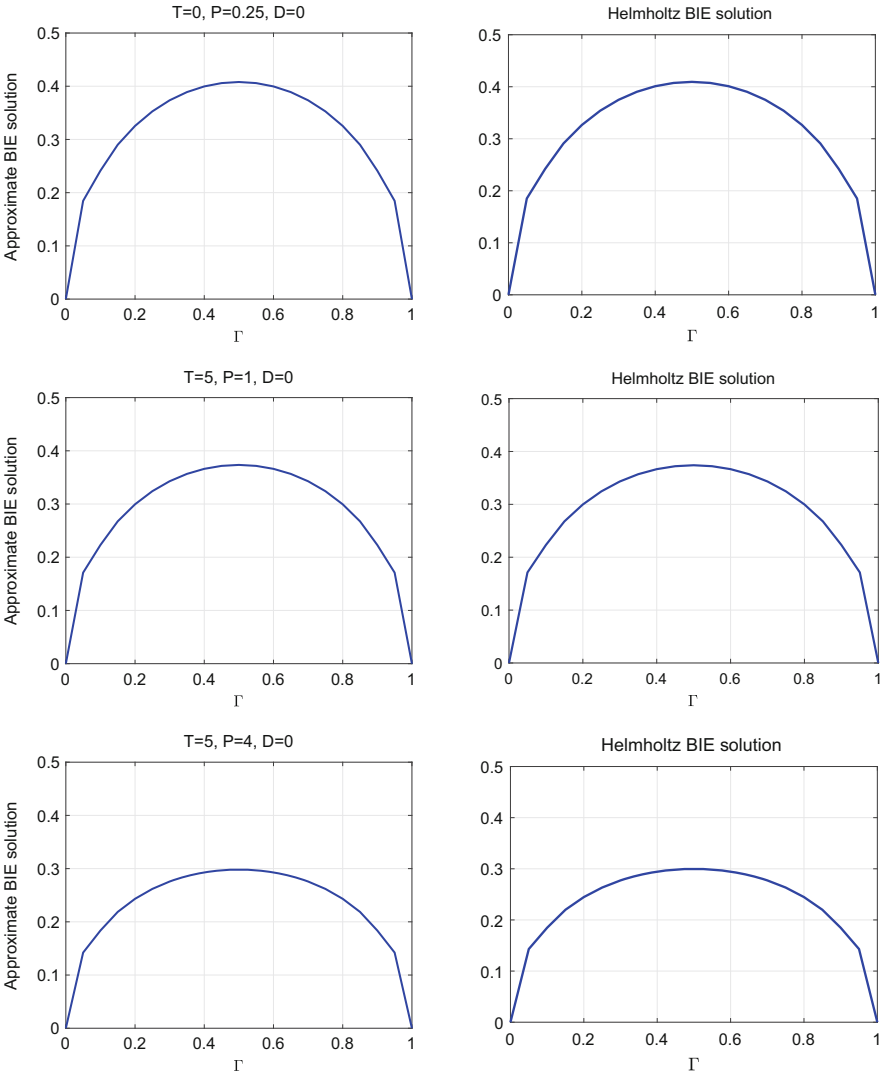


Fig. 5 $\varphi(\mathbf{x}, T)$ on Γ, for $D = 0$ and different values of P (left), together with the corresponding Helmholtz BIE static solution (right)

In Table 1, the condition number $\mu_2(A^{(0)})$ of linear system (17) matrix diagonal block, related to the previous simulations, is shown: the variation w.r.t. damping parameters is almost negligible, and we can note that the discrete problem to be solved at each time step is very well conditioned.

Table 1 Condition number $\mu_2(A^{(0)})$ with $A^{(0)} \in \mathbf{R}^{19\times 19}$ for different values of damping parameters

$P=0$	$\mu_2(A^{(0)})$	$D=0$	$\mu_2(A^{(0)})$
$D=0$	2.4161	$P=0$	2.4161
$D=0.25$	2.4180	$P=0.25$	2.4161
$D=1$	2.4238	$P=1$	2.4161
$D=4$	2.4455	$P=4$	2.4162

6 Conclusions

In this chapter, we have considered the so-called energetic BEM for the numerical solution of 2D damped wave propagation exterior problems equipped with Neumann boundary condition. The method was already considered for the numerical solution of the undamped wave equation in several space dimensions, revealing its accuracy and stability, also coupled with FEM. The presented numerical results confirm that these properties are maintained in the presence of dissipation terms in the model problem, as already highlighted in 1D simulations [5] and for 2D exterior problems equipped by Dirichlet boundary conditions [6].

Acknowledgements The authors are grateful to INdAM-GNCS for its financial support through Research Projects funding.

References

1. Aimi, A., Diligenti, M.: A new space-time energetic formulation for wave propagation analysis in layered media by BEMs. Int. J. Numer. Methods Eng. **75**, 1102–1132 (2008)
2. Aimi, A., Diligenti, M., Guardasoni, C.: Numerical integration schemes for space-time hypersingular integrals in energetic Galerkin BEM. Num. Alg. **55**(2–3), 145–170 (2010)
3. Aimi, A., Diligenti, M., Panizzi, S.: Energetic Galerkin BEM for wave propagation Neumann exterior problems. CMES **58**(2), 185–219 (2010)
4. Aimi, A., Diligenti, M., Frangi, A., Guardasoni, C.: Neumann exterior wave propagation problems: computational aspects of 3D energetic Galerkin BEM. Comp. Mech. **51**(4), 475–493 (2013)
5. Aimi, A., Diligenti, M., Guardasoni, C.: Energetic BEM-FEM coupling for the numerical solution of the damped wave equation. Adv. Comput. Math. **43**, 627–651 (2017)
6. Aimi, A., Diligenti, M., Guardasoni, C.: Energetic BEM for the numerical analysis of 2D Dirichlet damped wave propagation exterior problems. Commun. Appl. Ind. Math. **8(1)**, 103–127 (2017)
7. Bamberger, A., Ha Duong, T.: Formulation variationelle espace-temps pour le calcul par potential retardé de la diffraction d'une onde acoustique (I). Math. Meth. Appl. Sci. **8**, 405–435 (1986)
8. Bamberger, A., Ha Duong, T.: Formulation variationelle pour le calcul de la diffraction d'une onde acoustique par une surface rigide. Math. Meth. Appl. Sci. **8**, 598–608 (1986)
9. Banerjee, P., Butterfield, P.: Boundary Element Methods in Engineering. McGraw-Hill, London (1981)
10. Becache, E.: A variational Boundary Integral Equation method for an elastodynamic antiplane crack. Int. J. Numer. Methods Eng. **36**, 969–984 (1993)

11. Bini, D.: Matrix Structure and Applications. Les cours du CIRM **4**(1), 1–45 (2014)
12. Borodin, A., Munro, I.: The Computational Complexity of Algebraic and Numeric Problems. American Elsevier, New York (1975)
13. Chaillat, S., Desiderio, L., Ciarlet, P.: Theory and implementation of H-matrix based iterative and direct solvers for Helmholtz and elastodynamic oscillatory kernels. J. Comput. Phys. **341**, 429–446 (2017)
14. Costabel, M.: Time-dependent problems with the boundary integral equation method. In: Stein, E., et al. (eds.) Encyclopedia of Computational Mechanics, pp. 1–28. Wiley, New York (2004)
15. Ha Duong, T.: On retarded potential boundary integral equations and their discretization. In: Ainsworth, M., et al. (eds.) Topics in Computational Wave Propagation. Direct and Inverse Problems, pp. 301–336. Springer, Berlin (2003)
16. Hairer, E., Lubich, C., Schlichte, M.: Fast numerical solution of nonlinear Volterra convolution equations. SIAM J. Sci. Stat. Comput. **6**, 532–541 (1985)
17. Hartmann, F.: Introduction to Boundary Element Theory Method in Engineering. McGraw-Hill, London (1981)
18. Mayer, S.: Plasmonics: Fundamentals and Applications. Springer, Berlin (2007)
19. Oran Brigham, E.: The Fast Fourier Transform and Its Applications. Prentice Hall, Englewood Cliffs (1988)
20. Rao, S.: Mechanical Vibrations. Addison-Wesley Publishing, Reading (2010)
21. Stephan, E., Suri, M.: On the convergence of the p-version of the Boundary Element Galerkin Method. Math. Comput. **52**(185), 31–48 (1989)

Efficient Preconditioner Updates for Semilinear Space–Time Fractional Reaction–Diffusion Equations

Daniele Bertaccini and Fabio Durastante

Abstract The numerical solution of fractional partial differential equations poses significant computational challenges in regard to efficiency as a result of the nonlocality of the fractional differential operators. In this work we consider the numerical solution of nonlinear space–time fractional reaction–diffusion equations integrated in time by fractional linear multistep formulas. The Newton step needed to advance in (fractional) time requires the solution of sequences of large and dense linear systems because of the fractional operators in space. A preconditioning updating strategy devised recently is adapted and the spectrum of the underlying operators is briefly analyzed. Because of the quasilinearity of the problem, each Jacobian matrix of the Newton equations can be written as the sum of a multilevel Toeplitz plus a diagonal matrix and produced exactly in the code. Numerical tests with a population dynamics problem show that the proposed approach is fast and reliable with respect to standard direct, unpreconditioned, multilevel circulant/Toeplitz and ILU preconditioned iterative solvers.

Keywords Semilinear fractional diffusion equations · Update of preconditioners · Localized and structured linear systems

D. Bertaccini (✉)
Università di Roma "Tor Vergata", dipartimento di Matematica, Rome, Italy

Istituto per le Applicazioni del Calcolo (IAC) "M. Picone", National Research Council (CNR), Rome, Italy
e-mail: bertaccini@mat.uniroma2.it

F. Durastante
Università di Pisa, dipartimento di Informatica, Pisa, Italy
e-mail: fabio.durastante@di.unipi.it

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_15

1 Introduction and Rationale

Recently there has been a great deal of interest in the scientific community concerning theoretical aspects of the fractional calculus and its applications to modelling anomalous diffusion. Fractional derivatives are becoming widely used and accepted in models of diffusion-type processes where the underlying particle motion deviates from Brownian motion. Here we concentrate on semilinear space–time fractional reaction–diffusion equations, i.e., fractional reaction–diffusion partial differential equations where the nonlinearity of f is only in the reaction term such as

$$\begin{cases} \dfrac{\partial^\alpha u}{\partial t^\alpha} = f(u) \equiv \dfrac{\partial^\beta u}{\partial |x|^\beta} + \dfrac{\partial^\beta u}{\partial |y|^\beta} + g(u), \ \alpha \in (0,1), \ \beta \in (0,2), \\ u(x,y,0) = u_0(x,y), \qquad (x,y) \in \Omega, \\ u(x,y,t) = 0, \qquad (x,y) \in \partial\Omega, \ t > 0, \end{cases} \tag{1}$$

where $\dfrac{\partial^\alpha u}{\partial t^\alpha}$ is the Caputo fractional derivative of order α

$$\frac{\partial^\alpha u}{\partial t^\alpha} = \frac{1}{\Gamma(m-\alpha)} \int_0^t \frac{u^{(m)}(\tau)}{(t-\tau)^{\alpha+1-m}} d\tau, \qquad \alpha \in (m-1, m),$$

with $\Gamma(z)$ the Euler Gamma function, and

$$\frac{\partial^\beta u(x,y)}{\partial |x|^\beta} = -\frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{-i\xi} |\xi|^\beta \left[\int_{-\infty}^{+\infty} u(\eta, y) e^{i\xi\eta} d\eta \right] d\xi \tag{2}$$

is the Riesz fractional derivative in space relative to the space variable x. A similar expression is obtained for y.

To advance the solution in (fractional) time, we use fractional linear multistep formulas proposed in [19].

The solution of the Newton equations at each step, needed to advance the candidate approximate solution in (fractional) time, requires solving sequences of large and dense linear systems because of the fractional operators in space. Because of the quasilinearity of the problem, each Jacobian matrix of the Newton equations can be written as the sum of a multilevel Toeplitz T plus a diagonal matrix D and here is easily computed exactly. The above-mentioned explicit Toeplitz structure is essentially used for the analysis of the spectrum of the eigenvalues of the Jacobian matrices. Note that there exist no fast direct solvers for solving Toeplitz (multilevel or not) plus diagonal $(T + D)\mathbf{x} = \mathbf{b}$ because the displacement rank of the matrix $T + D$ can take any value between 0 and n. Hence, fast Toeplitz solvers that are based on small displacement rank of matrices cannot be applied. Given a vector $\mathbf{v}$ and a Toeplitz matrix T, the product $(T + D)\mathbf{v}$ can be computed in $O(n \log n)$ operations. In fact, T $\mathbf{v}$ can be obtained by FFTs by first embedding T into a $2n$-by-$2n$ circulant matrix; see Strang [27]. Thus Krylov iterative methods such as

the conjugate gradient or BiCGstab can be employed for these linear systems. The convergence rate of the conjugate gradient method depends on the spectrum of the matrix $T + D$, see Golub and van Loan [15]. However, in general, the spectrum of T, and hence of $T + D$, is not clustered, and hence the underlying Krylov method converges slowly without preconditioning. For a short review on preconditioners for Toeplitz plus diagonal matrices, see, e.g., [22] and the references therein.

In this paper we propose using the conjugate gradient and BiCGstab to solve the sequence of the Newton equations with a preconditioner updating strategy for the multilevel Toeplitz plus diagonal Jacobian matrices generated by the discretization of (1) based on incomplete factorizations using inversion and sparsification of incomplete LU factorizations with threshold. Some important computational aspects of the latter have been recently analyzed in [11].

Here we also add comparisons with standard direct, unpreconditioned, fixed multilevel Toeplitz-like and ILU preconditioned iterative solvers. In particular, for experiments performed with multilevel BCCB circulant (BCCB stands for block circulant with circulant blocks; see [21] for more details on multilevel Toeplitz and circulant matrices) and multilevel Toeplitz approximations, our unstructured but localized updates appear faster and more reliable than the structured ones for the linear systems of the Newton equation for problem (1), including the approximate inverse of a multilevel circulant plus diagonal as a preconditioner for $T + D$ proposed in [22].

Among the other papers concerning numerical integration of nonlinear fractional partial differential equations, we mention [12, 20, 26], but they do not consider preconditioner updates and are focused on integer partial derivatives in time only. In [12], a semilinear fractional partial differential equation from optimal control is considered with an approximate inverse preconditioner but with no updates because it is not necessary there.

2 Fractional Linear Multistep Formulas or FLMMs

An elegant and effective strategy for obtaining fractional linear multistep methods from linear multistep methods for ordinary differential equations was proposed in [19]. The key aspect of these schemes is the approximation of the Riemann–Liouville integral in the definition of the Caputo derivatives. This means establishing a convolution quadrature formula

$$I_h^\alpha \mathbf{u}(t_n) = h^\alpha \sum_{j=0}^{n} \omega_{n-j} \mathbf{u}(t_j) + h^\alpha \sum_{j=0}^{s} w_{n,j} \mathbf{u}(t_j), \tag{3}$$

on the uniform grid $\{t_n\}_n = \{n\, h\}_n$ for $h > 0$, where the weights ω_n and $w_{n,j}$ do not depend on h, moreover, the parameter $s \leq n$, i.e., the number of starting weights, is selected to take into account the singular behavior of integrated quantities in the

origin, see [13] . As discussed in [18, 29], the weights $\{\omega_n\}_n$ for the ordinary case with $\alpha = 1$, i.e., the case of a derivative of order one in time, can be computed as the coefficients of the formal power series:

$$\omega(\zeta) = \sum_{n=0}^{+\infty} \omega_n \zeta^n, \qquad \omega(\zeta) = \frac{\sigma(1/\zeta)}{\rho(1/\zeta)},$$

where $\sigma(\zeta)$ and $\rho(\zeta)$ are the characteristic polynomials of the linear multistep method. In [19], the extension for the fractional differential equations is obtained through the use of the new generating function $\omega_\alpha(\zeta)$:

$$\omega_\alpha(\zeta) = \sum_{n=0}^{+\infty} \omega_n \zeta^n, \qquad \omega_\alpha(\zeta) = \left(\frac{\sigma(1/\zeta)}{\rho(1/\zeta)}\right)^\alpha. \tag{4}$$

Methods of this kind are called fractional linear multistep methods (FLMMs), and when applied to fractional problems such as

$$\begin{cases} \dfrac{\partial^\alpha u(\mathbf{x},t)}{\partial t^\alpha} = f(\mathbf{x},t,u), & (\mathbf{x},t) \in Q = \Omega \times (0,T], \\ +\text{Boundary Conditions}, & (\mathbf{x},t) \in \Sigma = \partial\Omega \times (0,T], \\ +m \text{ Initial Conditions } \{u_0^{(k)}(\mathbf{x})\}_{k=0}^{m-1}, \ m \in \mathbb{N}, \ m-1 < \alpha \le m, \end{cases} \tag{5}$$

where among the arguments of the (nonlinear) function f there can be fractional partial differential equations, we get

$$\mathbf{u}^{(n)} = \sum_{k=0}^{m-1} \frac{t_n^k}{k!} \mathbf{u}_0^{(k)} + h^\alpha \sum_{j=0}^{s} w_{n,j} f(\mathbf{x}, t_j, \mathbf{u}^{(j)}) + h^\alpha \sum_{j=0}^{n} \omega_{n-j} f(\mathbf{x}, t_j, \mathbf{u}^{(j)}). \tag{6}$$

Theorem 1 ([19]) *Let (ρ, σ) denote the characteristic polynomials of an implicit linear multistep method which is stable and consistent of order p. Assume that the zeros of $\sigma(\zeta)$ have absolute values less than 1. If $\omega_\alpha(\zeta)$, given by (4), denotes the generating power series, then the corresponding convolution quadrature formula (3) is convergent of order p.*

Now, let us assume that p is the order of convergence of the method. The starting weights $w_{n,j}$ which are needed to deal with the singular behavior of the solution at the left end point of the time-integration interval are selected as in [13] by imposing (3) exact for the function t^ν and $\nu \in \mathscr{A}_{p-1} \cup \{p-1\} = \{\nu \in \mathbb{R} : \nu = i + j\alpha,\ i, j \in \{0, 1, 2, \ldots\},\ \nu < p-1\} \cup \{p-1\}$. Since we focus on methods of order $p = 2$, then we need to solve a system of $s+1$ linear equations at each step n, where s is the cardinality of $\mathscr{A}_1$, i.e., $s = \lceil \frac{1}{\alpha} \rceil$, given by

$$\sum_{j=0}^{s} w_{n,j} j^\nu = -\sum_{j=0}^{n} \omega_{n-j} j^\nu + \frac{\Gamma(\nu+1)}{\Gamma(\nu+1+\alpha)} n^{\nu+\alpha}, \qquad \nu \in \mathscr{A}_1 \cup \{1\}.$$

As discussed in [13, 14], in this case we have a mildly ill-conditioned Vandermonde system of reasonably small size (see [14, Section 6.2]), whose solution can be faced analytically. For more general cases and higher orders p, we refer to the analysis in [13]. The initialization step for (6) requires the knowledge of the first $s+1$ approximations $\mathbf{u}^{(0)}, \ldots, \mathbf{u}^{(s)}$ of the solution. Usually problems (5) come with only the value of $\mathbf{u}^{(0)}$, thus the remaining s values $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(s)}$ need to be evaluated in other ways. By using the same method, one needs to solve the nonlinear system of size sN, where N is the size of the discretized system of nonlinear fractional differential equations (or FDEs for short):

$$\begin{bmatrix} \mathbf{u}^{(1)} \\ \mathbf{u}^{(2)} \\ \vdots \\ \mathbf{u}^{(s)} \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{m-1} \frac{t_1^k}{k!} u_0^{(k)} \\ \sum_{k=0}^{m-1} \frac{t_2^k}{k!} u_0^{(k)} \\ \vdots \\ \sum_{k=0}^{m-1} \frac{t_s^k}{k!} u_0^{(k)} \end{bmatrix} + h^\alpha \begin{bmatrix} (\omega_1 + w_{1,0}) f(\mathbf{x}, t_0, \mathbf{u}^{(0)}) \\ (\omega_2 + w_{2,0}) f(\mathbf{x}, t_0, \mathbf{u}^{(0)}) \\ \vdots \\ (\omega_s + w_{s,0}) f(\mathbf{x}, t_0, \mathbf{u}^{(0)}) \end{bmatrix} + h^\alpha \mathscr{B} \begin{bmatrix} f(\mathbf{x}, t_1, \mathbf{u}^{(1)}) \\ f(\mathbf{x}, t_2, \mathbf{u}^{(2)}) \\ \vdots \\ f(\mathbf{x}, t_s, \mathbf{u}^{(s)}) \end{bmatrix}, \tag{7}$$

where

$$\mathscr{B} = \begin{bmatrix} \omega_0 I & & & \\ \omega_1 I & \omega_0 I & & \\ \vdots & \vdots & \ddots & \\ \omega_{s-1} I & \omega_{s-2} I & \ldots & \omega_0 I \end{bmatrix} + \begin{bmatrix} w_{1,1} I & w_{1,2} I & \ldots & w_{1,s} I \\ w_{2,1} I & w_{2,2} I & \ldots & w_{2,s} I \\ \vdots & \vdots & \ddots & \vdots \\ w_{s,1} I & w_{s,2} I & \ldots & w_{s,s} I \end{bmatrix}.$$

If M is the number of time steps, i.e., $h = T/M$, then the computational tasks we need to face for applying these methods to (5) are

1. M solutions of the Vandermonde linear systems of size $s+1$ for the computation of the starting weights $\{w_{n,j}\}$;
2. one solution of the nonlinear system (7) computing the initial approximations $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(s)}$;
3. M solutions of the nonlinear system given by the quadrature rule (6) for advancing the solution in time; and
4. computing the *lag term* in the quadrature rule (6), i.e., the term

$$\sum_{j=0}^{n-1} \omega_{n-j} f(\mathbf{x}, t_j, \mathbf{u}^{(j)})$$

accounting for the "memory" of the time fractional partial derivative.

The computation of the initial weights, i.e., the solution of the Vandermonde linear systems, does not present any particular challenge in our setting since its dimension

is usually reasonably small ($p = 2$ and $\alpha \in (0, 2)$). In general, the underlying computational cost depends linearly on the cardinality of the set $\mathscr{A}_{p-1}$ and thus on both the order p of the method and the order α of the fractional derivative, we refer to [13] for the treatment of these cases. On the other hand, the computation of the *lag term* can become increasingly expensive if the number of time steps M becomes large: this represents a critical bottleneck for all these methods. Using the implementation in [14] we perform this task by the $O(M \log(M)^2)$ FFT algorithm in [16], thus reaching a reasonable cost. In Sect. 3, we focus on the solution of the sequence of nonlinear systems needed to advance in time (6).

3 Preconditioning the Linear Systems of the Newton Iterations

At each time step we need to solve the nonlinear system (6) that can be restated into the form:

$$\mathbf{u}^{(n)} = \mathbf{z}^{(n)} + h^{\alpha}\omega_0 f(\mathbf{x}, t_n, \mathbf{u}^{(n)}),$$

where both $\mathbf{z}^{(n)}$ and ω_0 do not depend on $\mathbf{u}^{(n)}$. This task is equivalent to find a zero of the function

$$F(\mathbf{u}) = \mathbf{z}^{(n)} + h^{\alpha}\omega_0 f(\mathbf{x}, t_n, \mathbf{u}) - \mathbf{u} \equiv 0$$

and can be faced by means of the usual Newton iteration in the following form:

$$\begin{cases} \mathbf{u}_{k+1}^{(n)} = \mathbf{u}_k^{(n)} + \mathbf{d}_k^{(n)}, & k \geq 0, \\ \mathbf{d}_k^{(n)} = -J_F^{-1}(\mathbf{u}_k^{(n)}) F(\mathbf{u}_k^{(n)}) \\ \qquad = \left(I - h^{\alpha}\omega_0 J_f(\mathbf{x}, t_n, \mathbf{u}_k^{(n)})\right)^{-1} F(\mathbf{u}_k^{(n)}), \\ \mathbf{u}_0^{(n)} = \mathbf{u}^{(n-1)}, \end{cases} \tag{8}$$

until the *stopping criteria* on either the norm of $\|\mathbf{d}_k^{(n)}\|_2$ or $\|F(\mathbf{u}_k^{(n)})\|_2$ are satisfied. To shorten the notation, sometimes we write J_F for $J_F(\mathbf{u}_k^{(n)})$.

The nonlinear system (7) for the initialization phase can be solved similarly. In the cases of interest α is usually either in (0, 1) or in (1, 2), thus the solution of (7) is not computationally expensive and can be faced efficiently with a Newton–Krylov method with a frozen (i.e., computed once and then used for all systems in the underlying sequence of the Newton linear equations) ILUT preconditioner. See, e.g., [25] for details on ILUT preconditioners. Therefore, we do not discuss this issue anymore.

In order to update the Newton steps (8), we need to solve a sequence of linear systems with matrices given by

$$J_F(\mathbf{u}_k^{(n)}) = I - h^\alpha \omega_0 J_f(\mathbf{x}, t_n, \mathbf{u}_k^{(n)}),$$

where, other than $h^\alpha \omega_0$, the only term that can change is a diagonal matrix inside the Jacobian J_f and thus J_F can be written as

$$J_F(\mathbf{u}_k^{(n)}) = I - h^\alpha \omega_0 J_f(\mathbf{x}, t_n, \mathbf{u}_k^{(n)}) = A + D(\mathbf{x}, t_n, \mathbf{u}_k^{(n)}), \tag{9}$$

where A is a constant nonsingular matrix whose spectrum of eigenvalues is in the right half complex plane, in particular, in the case of Equation (1), matrix A contains the discretization of the linear part, i.e., the discrete representation of the fractional operator

$$\frac{\partial^\beta}{\partial |x|^\beta} + \frac{\partial^\beta}{\partial |y|^\beta},$$

while D is the diagonal matrix accounting for the Jacobian of the function $g(\cdot)$.

In the following, we denote with $J_g(u)$ the Jacobian matrix of the function $g(u)$ in (1).

Theorem 2 *Let $J_g(u)$ have eigenvalues with nonpositive real part. Then, the Jacobian matrix $J_F(\cdot)$ in (8) has all eigenvalues in the right half plane and thus it is nonsingular.*

Proof Follows from (9) and by the definition of f in (1) by recalling that h and ω_0 are positive values.

Corollary 1 *Let $J_f(u)$ be symmetric and nonnegative definite. Then, the Jacobian matrix $J_F(\cdot)$ in (8) is positive definite and thus it is nonsingular.*

Proof Follows from (9) by recalling that h and ω_0 are positive values.

In particular, the test problems in Sect. 4, based on the Riesz fractional derivatives (2), generate real symmetric and negative definite matrices J_f and therefore a nonsingular Jacobian matrix by Corollary 1, i.e., we easily find that the problems in Sect. 4 generate real, symmetric, and positive definite Jacobian matrices (and a nonsymmetric one but only in the setup phase (7)).

Remark 1 Under the assumptions in [10, Section 3.5.1], based mostly on the results in [17], in particular that the underlying matrices and their inverses are bounded in $\ell^2(\mathbb{K})$ for $\mathbb{K} = \mathbb{Z}$ or $\mathbb{N}$, we can prove that there is a decay of the entries along the main diagonal for all the Jacobian matrices generated by the discretization of the

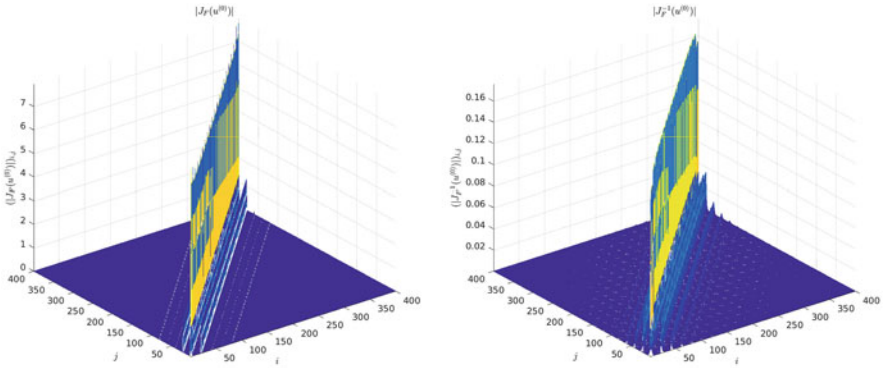


Fig. 1 Decay pattern of the reference Jacobian $J_F(\mathbf{u}^{(0)})$ and of its inverse $J_F^{-1}(\mathbf{u}^{(0)})$ for the problem described in Sect. 4

semilinear fractional partial differential equation (1) and, more interesting, for their inverses, i.e., we can write something like this:

$$\exists\, \lambda \in (0,1), \quad \text{such that} \quad (|J_F^{-1}(\cdot)|)_{i,j} \leq C(h)\lambda^{|i-j|}.$$

The claim is confirmed by numerical experiments, see Fig. 1. Unfortunately, without some restrictive assumptions, the parameter C that is dependent on h can be large and increases by decreasing h, while $\lambda \in (0,1)$ can be very near one in general.

The topic in Remark 1 is actually under investigation for some classes of nonlinear problems and will be considered in a future work.

These properties suggest that a sparse approximate inverse preconditioners in factored form can be appropriate for approximating the (dense) Jacobian matrix J_F.

By some numerical experiments, we can immediately observe that, in order to solve the linear systems of the underlying Newton steps by an iterative method faster than a direct solver, we need preconditioning. In order to build *efficiently* a sequence of preconditioners for (9), we start by considering approximate inverse preconditioners in factored form. The latter have been proved to be fast and reliable and well suited for various parallel computing paradigms; see, e.g., [3] and [11] for some recent notes on the implementation and computational cost analysis.

Let us write a sparse approximation of the inverse of A in factored form

$$A^{-1} \approx Z D^{-1} W^T,$$

where the matrices Z and W are lower triangular. If A is real, symmetric, and definite, then $W = Z$.

Among reliable algorithms for computing an approximate factorization of the inverse of the underlying matrices, we mention:

- the AINV techniques, see, e.g., [3–6, 11], based on a biconjugate Gram–Schmidt process (conjugate with respect to the bilinear form associated with the underlying matrix). Sparsity in the inverse factors is obtained by carrying out biconjugation process incompletely;
- the inversion-and-sparsification technique considered in [10, 11, 28], or INVT for short, based on computing and inverting an incomplete LU decomposition (or ILUT for short, where the final T stands for ILU with threshold) with a process of sparsification during the inversion process for the factors of the ILU.

We recall that there exist stabilized algorithms for computing approximate inverses for nonsymmetric matrices that are less prone to breakdowns than others; see [3, 6]. Indeed, we stress that in general incomplete factorizations (ILUs; see [25]) are not guaranteed to be nonsingular also for positive definite matrices. This issue holds also for efficient inverse ILU techniques, i.e., approximations of the inverse matrix generated by sparsification and inversion of an ILU algorithm; see [11, 28]. However, for our tests, both the above-mentioned approaches are efficient and reliable.

To avoid recomputing a new preconditioner from scratch for each value of the parameters and for each value of the approximate solution in (9), we adapt the *approximate inverse preconditioners updating framework* devised for sparse matrices in [2, 7, 9, 10] even if here the matrices are dense. The property that permits us to produce and use effective sparse approximations for dense matrices (the inverses of the Jacobians J_F) is the *decay* of the entries of the underlying matrices observed in Remark 1.

The overall strategy is the following: once an approximate inverse factorization J_0 for $J_F(\mathbf{u}^{(0)})$ is generated as

$$J_0^{-1} = Z D^{-1} W^T,$$

then by writing each $J_F(\mathbf{u}^{(k)})$, for $k \geq 1$, as

$$\begin{aligned} J_F(\mathbf{u}^{(k)}) &= J_F(\mathbf{u}^{(k)}) - J_F(\mathbf{u}^{(0)}) + J_F(\mathbf{u}^{(0)}) \\ &= J_F(\mathbf{u}^{(0)}) + \Delta_k, \qquad \Delta_k \triangleq J_F(\mathbf{u}^{(k)}) - J_F(\mathbf{u}^{(0)}), \end{aligned}$$

we build the updated preconditioner for $J_F(\mathbf{u}^{(k)})$ at step k, denoted J_k to shorten the notation, by using the one produced for J_0^{-1} as

$$P_k^{-1} = Z(D + E_k)^{-1} W^T \approx J_k^{-1}, \ E_k \triangleq g(W^T \Delta_k Z), \tag{10}$$

where g is a sparsification function, e.g., a function that extracts some banded approximation of its matrix argument. Note that g can also be chosen to produce a structured approximation of its arguments. A convergence analysis can be found

in [10, Section 3.6.1] where there is also a strategy to approximate the matrix value $g(W^T \Delta_k Z)$ without computing the matrix–matrix product $W^T \Delta_k Z$. When A is symmetric, all reasonings above repeat with $W = Z$.

Techniques with more than one *reference matrix* have been devised in [8, 10] by means of matrix interpolation.

4 Numerical Experiments

The preliminary experiments presented here are performed on a laptop running Linux with 8 Gb memory and CPU Intel® Core™ i7-4710HQ CPU with clock 2.50 GHz, while the GPU is a NVIDIA GeForce GTX 860M. The scalar code is written and executed in MATLAB R2016b, while for the GPU we use C++ with Cuda compilation tools, release 7.5, V7.5.17 and the CUSP library v0.5.1 [1]. The code for FLMM from [14] is used as outer framework for the updating strategy.

4.1 A Time-Fractional Biological Population Model

We consider the following space–time fractional partial differential equation [24]:

$$\begin{cases} \dfrac{\partial^\alpha u}{\partial t^\alpha} = \dfrac{\partial^\beta u}{\partial |x|^\beta} + \dfrac{\partial^\beta u}{\partial |y|^\beta} + g(u), & (x, y) \in \Omega,\ t > 0, \\ u(x, y, 0) = u_0(x, y), & (x, y) \in \Omega, \\ u(x, y, t) = 0, & (x, y) \in \partial\Omega,\ t > 0, \end{cases} \tag{11}$$

where $\alpha \in (0, 1)$, $\beta \in (1, 2)$; $g(u) = h\, u^{\tilde{a}}(1 - r\, u)^{\tilde{b}}$ represents the balance between birth and death in the population u on the domain Ω.

We semidiscretize in space equation (11) in the domain $\Omega = [a, b] \times [c, d]$ on the grid

$$\left\{ (x_i, y_j) = (a + \Delta x\, i, c + \Delta y\, j) \ :\ \Delta x = \frac{b-a}{N_x},\ \Delta y = \frac{d-c}{N_y} \right\}_{i,j=0}^{N_x, N_y}$$

with the usual notation $u_{i,j} = u(x_i, y_j)$, $g_{i,j} = g(u_{i,j})$ and using the fractional centered derivatives discretization for the Riesz derivative [23] from equation (2), we find:

$$\begin{aligned} \frac{\partial^\beta u(x, y)}{\partial |x|^\beta} &\approx -\frac{1}{\Delta x^\beta} \sum_{k=-\frac{b-x}{\Delta x}}^{\frac{x-a}{\Delta x}} \varsigma_k u(x - k\Delta x, y) + O(\Delta x^2), \\ \varsigma_k &= \frac{(-1)^k \Gamma(\beta + 1)}{\Gamma(\beta/2 - k + 1)\Gamma(\beta/2 + k + 1)}, \end{aligned} \tag{12}$$

and similarly for $\partial^\beta u(x,y)/\partial|y|^\beta$. To apply the Newton method, we need to compute the Jacobian of the function

$$(f(\mathbf{u}))_{i,j} = \left[\frac{1}{\Delta x^\beta} \sum_{k=-\frac{b-x_i}{\Delta x}}^{\frac{x_i-a}{\Delta x}} \varsigma_k u(x-k\Delta x, y_j) + \frac{1}{\Delta y^\beta} \sum_{k=-\frac{d-y_j}{\Delta y}}^{\frac{y_j-c}{\Delta y}} \varsigma_k u(x_i, y-k\Delta y) \right] + g_{i,j},$$

for $i = 0, \ldots, N_x$ and $j = 0, \ldots, N_y$, that is given by

$$J_f(\mathbf{u}) = \Delta x^{-\beta} T^{(1)} \otimes I + \Delta y^{-\beta} I \otimes T^{(2)} + J_g(\mathbf{u}), \tag{13}$$

where the entries of the Toeplitz matrices $T^{(1)}$ and $T^{(2)}$ come from the coefficients of the discretization in (12), and J_g is the Jacobian of the function $g(\mathbf{u})$ satisfying Theorem 2 as well.

Lemma 1 *If $\Delta x = \Delta y = \Delta$ and $\beta \in (1,2)$, then the sequence of matrices $\{T^{(1)} \otimes I + I \otimes T^{(2)}\}_{N_x}$ is a two-level Toeplitz matrix sequence with generating function*

$$t_\beta(\theta_1, \theta_2) = \Re\left((e^{i\theta_1} - e^{-i\theta_1})^\beta + (e^{i\theta_2} - e^{-i\theta_2})^\beta \right).$$

Proof Matrices $T^{(1)} = T^{(2)}$ are symmetric negative definite Toeplitz matrices whose coefficients are defined by (12). Moreover, $\{\varsigma_k\}_k \in \ell^1$, since

$$|\varsigma_k| \sim \frac{1}{\pi} \left| \frac{\Gamma(\beta+1)}{k^{1+\beta}} \right| \text{ for } k \to +\infty \text{ and } \beta \in (1,2),$$

i.e., we have convergence by comparing asymptotically with an absolutely summable series. Finally, $t_\beta(\theta_1, \theta_2)$ generates the matrix sequence by direct inspection of the coefficients ς_k that are the Fourier coefficients of the function $\Re((e^{i\theta_1} - e^{-i\theta_1})^\beta)$.

Proposition 1 *If $\Delta x = \Delta y = \Delta$ and $\beta \in (1,2)$, then the matrix A in (9) is given by*

$$A = I - \frac{h^\alpha}{\Delta^\beta} \omega_0 (T^{(1)} \otimes I + I \otimes T^{(2)}),$$

and it is symmetric and positive definite.

Proof The proof is straightforward by using Lemma 1 and recalling that all coefficients h, Δ, ω_0 are positive.

For the source term $g(u)$, we select $\tilde{a} = -1$, $\tilde{b} = 1$, $h = 3$, and $r = 0.25$, while the initial data is given by

$$u_0(x,y) = \sqrt{hr\frac{x^2}{4} + hr\frac{y^2}{4} + y + 5},$$

with the same choice of h and r, the fractional order of derivative in space is $\beta = 1.5$, and the domain is $\Omega = [-1, 1]^2$.

In Table 1 we report the total number of matrix–vector products ($A\mathbf{v}$), the number of nonlinear iterations (N^{IT}) (equal to the total number of linear system solved here), the average number of matrix–vector products for each linear system ($A\mathbf{v}^{\mathrm{avg.}}$), and the timings in seconds. The iterative methods considered are *BiCGstab* for the initialization steps of (7) because the related linear systems are nonsymmetric, and the *conjugate gradient* (or CG for short) for solving the Newton linear systems (8) that are symmetric and positive definite for our problems. The updated INVT preconditioners used with CG to solve linear systems in (8) are built by using only a diagonal correction from a reference preconditioner with tolerances $\tau_L = 1e-3$ for the ILUT phase and $\tau_Z = 1e-1$ for the inversion phase, respectively. We compare the performances of the underlying approach with the built-in Matlab's direct solver "\," with the nonpreconditioned BiCGstab/CG and BiCGstab/CG preconditioned by recomputing an ILUT incomplete factorization for all systems from scratch with threshold (drop tolerance) $\tau_L = 1e-2$. It is intended that when the underlying linear systems are symmetric, then the symmetric versions of the incomplete factorizations are used even if we write ILU/ILUT, etc., and that the drop tolerances used are those that give among the best possible performances. Moreover, we report also tests with a block circulant with circulant blocks preconditioner with Strang's approximation (see, e.g., [21] for details on this approach) recomputed for each linear system because keeping the preconditioner fixed delays sensibly the convergence to the prescribed accuracy. A † is reported when the method fails to achieve convergence.

The three fractional quadrature formulas generated by the following second order methods are tested: fractional trapezoidal rule, Newton–Gregory, and BDF2 formulas. We do not show the results with the frozen ILUT/ILUT(l) preconditioners (the ILUT computed once and then reused for all experiments, also with l level of fill-in dropping; see, e.g., [25]) because they give unreliable results due to an erratic convergence behavior for some tests.

From the reported experiments and from others performed and not shown, we can draw some final comments:

- The iterations using preconditioners computed by our updating strategy are always faster (and much less memory consuming) than the built-in Matlab's solver, unpreconditioned and ILU/ILUT- and BCCB-preconditioned iterations.
- We can observe a clustering effect around the unity for the eigenvalues produced by our updated preconditioners such as the sample in Fig. 2.
- The Matlab direct solver cannot be used for large problems because it gives out of memory error quite soon.
- The updated preconditioners based on INVT (inversion and sparsification) are reliable and slightly faster than those based on AINV [4–6, 11]. On the other hand, we observed experimentally that the latter are optimal with respect to the discretization, i.e., the number of the underlying preconditioned iterations does not increase refining the mesh. Moreover, AINV can give better performances when used in a GPU environment, see [11].

Table 1 A space–time fractional biological population model

			Recomputed BCCB Strang				Updated INVT(1e-3,1e-1)				Recomputed ILU				Direct
N	M	α	N^{IT}	$A\mathbf{v}$	$A\mathbf{v}^{avg.}$	T(s)	N^{IT}	$A\mathbf{v}$	$A\mathbf{v}^{avg.}$	T(s)	N^{IT}	$A\mathbf{v}$	$A\mathbf{v}^{avg.}$	T(s)	T(s)
20	200	0.4	255	6794	27	2.22e+00	201	2005	10	2.90e-01	201	1010	5	3.73e-01	7.41e-01
40	400	0.4	839	16212	20	2.68e+01	413	5300	13	2.65e+00	413	2586	6	6.07e+00	1.10e+01
60	600	0.4	1321	28872	22	2.41e+02	612	9182	15	1.53e+01	†	†	†	†	1.23e+02
80	800	0.4	1905	45826	24	1.04e+03	813	13852	17	5.77e+01	†	†	†	†	–
20	200	0.5	202	2853	14	1.83e+00	202	1619	8	2.30e-01	202	1009	5	3.47e-01	1.74e+00
40	400	0.5	425	7298	17	8.09e+00	425	4691	11	2.37e+00	426	2159	5	5.42e+00	9.86e+00
60	600	0.5	674	12486	19	3.39e+01	674	8173	12	1.44e+01	673	4045	6	3.84e+01	1.26e+02
80	800	0.5	998	20416	21	1.09e+02	997	14001	14	5.64e+01	997	6065	6	1.59e+02	–
20	200	0.6	201	2425	12	1.26e+00	201	1208	6	2.15e-01	201	805	4	3.34e-01	3.22e-01
40	400	0.6	408	5756	14	7.57e+00	409	3330	8	2.05e+00	409	2046	5	5.25e+00	9.62e+00
60	600	0.6	608	9202	15	2.97e+01	609	6033	10	1.15e+01	608	3045	5	3.38e+01	1.09e+02
80	800	0.6	815	13147	16	8.64e+01	814	8431	10	4.06e+01	814	4095	5	1.29e+02	–
20	200	0.7	200	2063	10	1.22e+00	200	944	5	2.04e-01	200	798	4	3.31e-01	3.25e-01
40	400	0.7	405	4840	12	7.41e+00	405	2230	6	1.90e+00	405	1621	4	5.09e+00	9.55e+00
60	600	0.7	602	7274	12	2.87e+01	602	3782	6	1.08e+01	602	2412	4	3.36e+01	1.10e+02
80	800	0.7	803	9768	13	8.05e+01	802	5732	7	3.44e+01	802	3235	4	1.24e+02	–

(a) Trapezoidal (Tustin) Method

(continued)

Table 1 (continued)

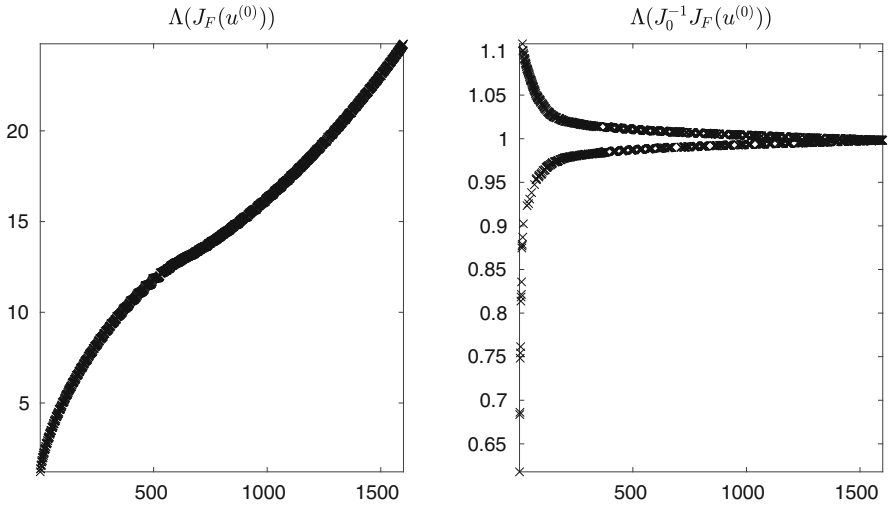
			Recomputed BCCB Strang				Updated INVT(1e-3,1e-1)				Recomputed ILU				Direct
N	M	α	N^{IT}	$A\mathbf{v}$	$A\mathbf{v}^{avg.}$	T(s)	N^{IT}	$A\mathbf{v}$	$A\mathbf{v}^{avg.}$	T(s)	N^{IT}	$A\mathbf{v}$	$A\mathbf{v}^{avg.}$	T(s)	T(s)
20	200	0.4	202	4584	23	1.91e+00	202	2018	10	2.62e-01	201	1010	5	3.65e-01	3.84e-01
40	400	0.4	435	8772	20	2.07e+01	413	5373	13	2.65e+00	413	2986	7	6.08e+00	1.08e+01
60	600	0.4	646	14350	22	1.71e+02	609	8704	14	1.43e+01	†	†	†	†	1.18e+02
80	800	0.4	851	20648	24	7.95e+02	807	13740	17	5.70e+01	†	†	†	†	–
20	200	0.5	203	2885	14	2.74e+00	202	1619	8	2.40e-01	202	1009	5	3.62e-01	2.71e+00
40	400	0.5	412	6890	17	7.99e+00	412	4407	11	2.26e+00	411	2069	5	5.53e+00	9.85e+00
60	600	0.5	608	11076	18	3.10e+01	608	6927	11	1.25e+01	607	3653	6	3.47e+01	1.10e+02
80	800	0.5	805	15960	20	9.46e+01	806	10509	13	4.58e+01	806	4871	6	1.29e+02	–
20	200	0.6	201	2437	12	1.25e+00	201	1214	6	2.11e-01	201	805	4	3.38e-01	3.07e-01
40	400	0.6	408	5768	14	7.80e+00	408	3313	8	2.04e+00	408	2041	5	5.24e+00	9.47e+00
60	600	0.6	605	9724	16	3.06e+01	605	6067	10	1.15e+01	605	3034	5	3.38e+01	1.12e+02
80	800	0.6	804	12967	16	8.69e+01	802	8103	10	4.01e+01	803	4029	5	1.26e+02	–
20	200	0.7	200	2101	11	1.23e+00	200	1001	5	2.08e-01	200	798	4	3.30e-01	3.94e-01
40	400	0.7	405	4884	12	7.38e+00	405	2392	6	1.88e+00	405	1626	4	5.17e+00	1.15e+01
60	600	0.7	603	7310	12	2.86e+01	603	3801	6	1.08e+01	602	2415	4	3.31e+01	1.09e+02
80	800	0.7	802	9798	13	8.05e+01	801	5803	7	3.53e+01	801	3237	4	1.24e+02	–

(b) Generalized Newton–Gregory formula

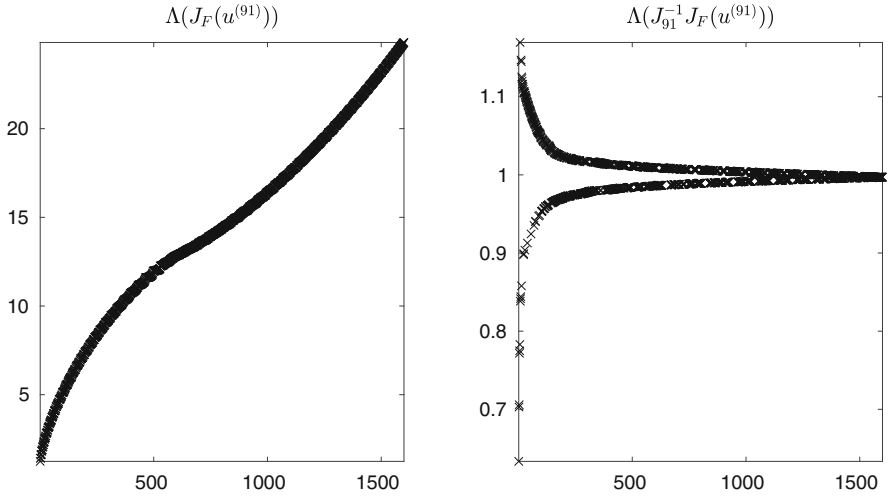
20	200	0.4	202	4845	24	2.09e+00	202	2018	10	3.05e-01	202	2030	10	3.71e-01	3.70e-01
40	400	0.4	441	9032	21	2.52e+01	413	5373	13	2.66e+00	413	5976	14	6.13e+00	1.10e+01
60	600	0.4	651	14528	22	1.64e+02	608	9135	15	1.48e+01	†	†	†	†	1.14e+02
80	800	0.4	859	20972	25	1.68e+03	806	13419	17	5.55e+01	†	†	†	†	–
20	200	0.5	203	2927	15	1.51e+00	203	1647	8	2.82e-01	203	2027	10	3.60e-01	6.38e-01
40	400	0.5	413	7430	18	8.01e+00	413	4589	11	2.34e+00	413	4283	10	5.30e+00	1.04e+01
60	600	0.5	609	11194	19	3.01e+01	609	7364	12	1.29e+01	609	7337	12	3.46e+01	1.06e+02
80	800	0.5	807	16258	20	8.87e+01	808	10531	13	4.55e+01	807	10752	13	1.29e+02	–
20	200	0.6	201	2475	12	1.32e+00	201	1259	6	2.15e-01	201	1611	8	3.40e-01	3.30e-01
40	400	0.6	410	5838	14	7.59e+00	409	3402	8	2.04e+00	409	4091	10	5.23e+00	9.78e+00
60	600	0.6	606	9764	16	2.92e+01	606	5744	9	1.20e+01	606	6082	10	3.47e+01	1.06e+02
80	800	0.6	806	13093	16	8.48e+01	804	8330	10	4.09e+01	805	8096	10	1.26e+02	–
20	200	0.7	200	2395	12	1.22e+00	200	1008	5	1.98e-01	200	1596	8	3.34e-01	3.39e-01
40	400	0.7	406	4918	12	7.37e+00	406	2500	6	1.94e+00	406	3261	8	5.17e+00	9.86e+00
60	600	0.7	604	7408	12	2.82e+01	603	4295	7	1.10e+01	603	4869	8	3.37e+01	1.02e+02
80	800	0.7	803	11282	14	8.24e+01	800	6277	8	3.60e+01	800	6452	8	1.24e+02	–

(c) Backward Differentiation Formula

The above-mentioned order two time-fractional approximation formulas are used



(a) Seed Jacobian



(b) Updated Jacobian

Fig. 2 Comparisons of the spectra for the seed Jacobian, its preconditioned version (**a**) and of the Jacobian for a matrix of the sequence (13) with the updated preconditioner (**b**). The seed preconditioner is computed with a drop tolerance $\tau = 1e-3$ for the problem with $N = 40$, $M = 400$, and $\alpha = 0.4$ using the trapezoidal method. To update the underlying preconditioners, the function g in (10) extracts only the main diagonal of its matrix argument

- Using a multilevel circulant preconditioner gives timings worse than ours and sometimes much worse if the preconditioner is recomputed for each system.
- Adapting the updating circulant plus diagonal preconditioners discussed in [22], even if it is very interesting and fast, is not beneficial here because of two main reasons: The function generating the Jacobian matrix J_F is bivariate and the multilevel circulant preconditioner is not optimal for the differential part of the problem. The interpolation used in the approach in [22] works very well for univariate functions and it is essentially linear while the problems considered here are not.

Acknowledgements We wish to thank two anonymous referees for their constructive comments which have improved the readability of the paper.

This work was supported in part by the 2018 GNCS–INDAM project "Tecniche innovative per problemi di algebra lineare." The first author acknowledges the MIUR Excellence Department Project awarded to the Department of Mathematics, University of Rome Tor Vergata, CUP E83C18000100006.

References

1. Bell, N., Garland, M.: Cusp: Generic Parallel Algorithms for Sparse Matrix and Graph Computations. Version 0.5.1 (2015). http://cusplibrary.github.io/
2. Bellavia, S., Bertaccini, D., Morini, B.: Nonsymmetric preconditioner updates in Newton-Krylov methods for nonlinear systems. SIAM J. Sci. Comput. **33**(5), 2595–2619 (2011)
3. Benzi, M.: Preconditioning techniques for large linear systems: a survey. J. Comput. Phys. **182**, 418–477 (2002)
4. Benzi, M., Tüma, M.: A sparse approximate inverse preconditioner for nonsymmetric linear systems. SIAM J. Sci. Comput. **19**(3), 968–994 (1998)
5. Benzi, M., Meyer, C.D., Tüma, M.: A sparse approximate inverse preconditioner for the conjugate gradient method. SIAM J. Sci. Comput. **17**(5), 1135–1149 (1996)
6. Benzi, M., Cullum, J.K., Tüma, M.: Robust approximate inverse preconditioning for the conjugate gradient method. SIAM J. Sci. Comput. **22**(4), 1318–1332 (2000)
7. Bertaccini, D.: Efficient preconditioning for sequences of parametric complex symmetric linear systems. ETNA **18**, 49–64 (2004)
8. Bertaccini, D., Durastante, F.: Interpolating preconditioners for the solution of sequence of linear systems. Comput. Math. Appl. **72**(4), 1118–1130 (2016)
9. Bertaccini, D., Durastante, F.: Solving mixed classical and fractional partial differential equations using short–memory principle and approximate inverses. Numer. Algorithms **74**(4), 1061–1082 (2017)
10. Bertaccini, D., Durastante, F.: Iterative Methods and Preconditioning for Large and Sparse Linear Systems with Applications. Chapman and Hall/CRC, New York (2018)
11. Bertaccini, D., Filippone, S.: Sparse approximate inverse preconditioners on high performance GPU platforms. Comput. Math. Appl. **71**, 693–711 (2016)
12. Cipolla, S., Durastante, F.: Fractional PDE constrained optimization: An optimize-then-discretize approach with L-BFGS and approximate inverse preconditioning. Appl. Numer. Math. **123**, 43–57 (2018)
13. Diethelm, K., Ford, J.M., Ford, N.J., Weilbeer, M.: Pitfalls in fast numerical solvers for fractional differential equations. J. Comput. Appl. Math. **186**(2), 482–503 (2006)
14. Garrappa, R.: Trapezoidal methods for fractional differential equations: theoretical and computational aspects. Math. Comput. Simul. **110**, 96–112 (2015)

15. Golub, G.H., Van Loan, C.F.: Matrix computations, 4 edn. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore (2012)
16. Hairer, E., Lubich, C., Schlichte, M.: Fast numerical solution of nonlinear Volterra convolution equations. SIAM J. Sci. Stat. Comput. **6**(3), 532–541 (1985)
17. Jaffard, S.: Propriétés des matrices «bien localisées» près de leur diagonale et quelques applications. In: Ann. Henri Poincaré Analyse non linéaire, vol. 7.5, pp. 461–476. Gauthier-Villars (1990)
18. Lambert, J.D.: Numerical Methods for Ordinary Differential Systems: The Initial Value Problem. Wiley, New York, (1991)
19. Lubich, C.: Discretized fractional calculus. SIAM J. Math. Anal. **17**(3), 704–719 (1986)
20. Moroney, T., Yang, Q.: A banded preconditioner for the two–sided, nonlinear space–fractional diffusion equation. Comput. Math. Appl. **66**(5), 659–667 (2013). Fractional Differentiation and its Applications. https://doi.org/10.1016/j.camwa.2013.01.048. http://www.sciencedirect.com/science/article/pii/S0898122113000783
21. Ng, M.K.: Iterative Methods for Toeplitz Systems. Oxford University Press, Oxford (2004)
22. Ng, M.K., Pan, J.: Approximate inverse circulant–plus–diagonal preconditioners for Toeplitz-plus-diagonal matrices. SIAM J. Sci. Comput. **32**(3), 1442–1464 (2010)
23. Ortigueira, M.D.: Riesz potential operators and inverses via fractional centred derivatives. Int. J. Math. Math. Sci. **2006**, 48391 (2006)
24. Prakash, A., Kumar, M.: Numerical solution of two dimensional time fractional-order biological population model. Open Phys. **14**(1), 177–186 (2016)
25. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Philadelphia (2003)
26. Simmons, A., Yang, Q., Moroney, T.: A preconditioned numerical solver for stiff nonlinear reaction–diffusion equations with fractional Laplacians that avoids dense matrices. J. Comput. Phys. **287**, 254–268 (2015). https://doi.org/10.1016/j.jcp.2015.02.012. http://www.sciencedirect.com/science/article/pii/S0021999115000741
27. Strang, G.: A proposal for Toeplitz matrix calculations. Stud. Appl. Math. **74**(2), 171–176 (1986)
28. Van Duin, A.C.: Scalable parallel preconditioning with the sparse approximate inverse of triangular matrices. SIAM J. Matrix Anal. Appl. **20**(4), 987–1006 (1999)
29. Wolkenfelt, P.H.M.: Linear Multistep Methods and the Construction of Quadrature Formulae for Volterra Integral and Integro–Differential Equations. Tech. Rep. NW 76/79, Mathematisch Centrum, Amsterdam (1979)

A Nuclear-Norm Model for Multi-Frame Super-Resolution Reconstruction from Video Clips

Rui Zhao and Raymond HF Chan

Abstract We propose a variational approach to obtain super-resolution images from multiple low-resolution frames extracted from video clips. First the displacement between the low-resolution frames and the reference frame is computed by an optical flow algorithm. Then a low-rank model is used to construct the reference frame in high resolution by incorporating the information of the low-resolution frames. The model has two terms: a 2-norm data fidelity term and a nuclear-norm regularization term. Alternating direction method of multipliers is used to solve the model. Comparison of our methods with other models on synthetic and real video clips shows that our resulting images are more accurate with less artifacts. It also provides much finer and discernable details.

Keywords Image processing · Super-resolution · Low-rank approximation

1 Introduction

Super-resolution (SR) image reconstruction from multiple low-resolution (LR) frames has many applications, such as in remote sensing, surveillance, and medical imaging. After the pioneering work of Tsai and Huang [28], SR image reconstruction has become more and more popular in image processing community, see, for example, [3, 8, 10, 12, 19, 25–27]. SR image reconstruction problems can be classified into two categories: single-frame super-resolution (SFSR) problems and multi-frame super-resolution (MFSR) problems. In this paper, we mainly focus on the multi-frame case, especially the MFSR problems from low-resolution video sequences. Below, we first review some existing work related to MFSR problems.

R. Zhao
Department of Mathematics, The Chinese University of Hong Kong, Shatin, NT, Hong Kong
e-mail: rzhao@math.cuhk.edu.hk

R.H.F. Chan (✉)
Department of Mathematics, City University of Hong Kong, KLN, Hong Kong
e-mail: rchan.sci@cityu.edu.hk

D. A. Bini et al. (eds.), *Structured Matrices in Numerical Linear Algebra*,
Springer INdAM Series 30, https://doi.org/10.1007/978-3-030-04088-8_16

Bose and Boo [3] considered the case where the multiple LR image frames were shifted with affine transformations. They modeled the original high-resolution (HR) image as a stationary Markov–Gaussian random field. Then they made use of the maximum a posteriori scheme to solve their model. However, the affine transformation assumption may not be satisfied in practice, for example, when there are complex motions or illumination changes. Another approach for SR image reconstruction is the one known as patch-based or learning-based. Bishop et al. [2] used a set of learned image patches which capture the information between the middle and high spatial frequency bands. They assumed a priori distribution over such patches and made use of the previous enhanced frame to provide part of the training set. The disadvantage of this patch-based method is that it is usually time consuming and sensitive to the off-line training set. Liu and Sun [18] applied Bayesian approach to estimate simultaneously the underlying motion, the blurring kernel, the noise level, and the HR image. Within each iteration, they estimated the motion, the blurring kernel, and the HR image alternatively by maximizing a posteriori, respectively. Based on this work, Ma et al. [20] tackled motion blur in their paper. An expectation-maximization (EM) framework is applied to the Bayesian approach to guide the estimation of motion blur. These methods used optical flow to model the motion between different frames. However, they are sensitive to the accuracy of flow estimation. The results may fail when the noise is heavy.

In [6], Chan et al. applied wavelet analysis to HR image reconstruction. They decomposed the image from previous iteration into wavelet frequency domain and applied wavelet thresholding to denoise the resulting images. Based on this model, Chan et al. [7] later developed an iterative MFSR approach by using tight-frame wavelet filters. However, because of the number of framelets involved in analyzing the LR images, the algorithm can be extremely time consuming.

Optimization models are one of the most important image processing models. Following the classical ROF model [24], Farsiu et al. [11] proposed a total variation-l_1 model where they used the l_1 norm for the super-resolution data fidelity term. However, it is known that TV regularization enforces a piecewise solution. Therefore, their method will produce some artifacts. Li et al. [16] used l_1 norm of the geometric tight-framelet coefficients as the regularizer and adaptively mimicking l_1 and l_2 norms as the data fidelity term. They also assumed affine motions between different frames. The results are therefore not good when complex motions or illumination changes are involved.

Chen and Qi [9] recently proposed a single-frame HR image reconstruction method via low rank regularization. Jin et al. [14] designed a patch-based low rank matrix completion algorithm from the sparse representation of LR images. The main idea of these two papers is based on the assumption that each LR image is downsampled from a blurred and shifted HR image. However, these works assumed that the original HR image, when considered as a matrix, has a low rank property, which is not convincing in general.

In this paper, we show that the low rank property can in fact be constructed under MFSR framework. The idea is to consider each LR image as a downsampled instance of a *different* blurred and shifted HR image. Then when all these different HR images are properly aligned, they should give a low rank matrix; therefore, we

can use a low-rank prior to obtain a better solution. Many existing works assume that the shift between two consecutive LR frames is small, see, e.g., [1, 11, 22, 30, 31]. In this paper, we allow illumination changes and more complex motions other than affine transformation. They are handled by an optical flow model proposed in [13]. Once the motions are determined, we reconstruct the high-resolution image by minimizing a functional which consists of two terms: the 2-norm data fidelity term to suppress Gaussian noise and a nuclear-norm regularizer to enforce the low-rank prior. Tests on seven synthetic and real video clips show that our resulting images are more accurate with less artifacts. It can also provide much finer and discernable details.

The rest of the paper is organized as follows: Section 2 gives a brief review of a classical model on modeling LR images from HR images. Our model will be based on this model. Section 3 provides the details of our low-rank model, including image registration by optical flow and the solution of our optimization problem by alternating direction method. Section 4 gives experimental results on the test videos. Conclusions are given in Sect. 5.

To simplify our discussion, we now give the notation that we will be using in the rest of the paper. For any integer $m \in \mathbb{Z}$, I_m is the $m \times m$ identity matrix. For any integer $l \in \mathbb{Z}$ and positive integer $n \in \mathbb{Z}^+$, there exists a unique $0 \le \widetilde{l} < n$ such that $\widetilde{l} \equiv l \mod n$. Let $N_n(l)$ denote the $n \times n$ matrix

$$N_n(l) = \begin{bmatrix} 0 & I_{n-\widetilde{l}} \\ I_{\widetilde{l}} & 0 \end{bmatrix}. \tag{1}$$

For a vector $\mathbf{f} \in \mathbb{R}^n$, $N_n(l)\mathbf{f}$ is the vector with entries of $\mathbf{f}$ cyclic-shifted by l.

Define the downsampling matrix D_i and the upsampling matrix D_i^T as

$$D_i(n) = I_n \otimes \mathbf{e}_i^T \text{ and } D_i^T(n) = I_n \otimes \mathbf{e}_i, \ i = 0, 1, \tag{2}$$

where $\mathbf{e}_0 = [1, 0]^T$, $\mathbf{e}_1 = [0, 1]^T$, and $\otimes$ is the Kronecker product. For $0 \le \epsilon \le 1$, define $T_n(\epsilon)$ to be the $n \times n$ circulant matrix

$$T_n(\epsilon) = \begin{bmatrix} 1-\epsilon & \epsilon & \cdots & 0 \\ 0 & 1-\epsilon & \ddots & \vdots \\ \vdots & \ddots & \ddots & \epsilon \\ \epsilon & \cdots & 0 & 1-\epsilon \end{bmatrix}. \tag{3}$$

This matrix performs the effect of linear interpolation shifted by ϵ.

For a matrix $X_{m\times n}$, the *nuclear norm* $\|\cdot\|_*$ of $X_{m\times n}$ is given by

$$\|X_{m\times n}\|_* = \sum_{i=1}^{r} |\sigma_i|,$$

where $\sigma_i, i = 1, 2, \cdots, r$ are *singular values* of $X_{m\times n}$.

2 Low-Resolution Model with Shifts

Consider a LR sensor array recording a video of an object. Then it gives multiple LR images of the object. Unless the object or the sensor array is completely motionless during the recording, the LR images will contain multiple information of the object at different shifted locations (either because of the motion of the object or of the sensor array itself). Our problem is to improve the resolution of one of the LR images (called the reference image) by incorporating information from the other LR images.

Let the sensor array consist of $m \times n$ sensing elements, where the width and the height of each sensing element are L_x and L_y, respectively. Then, the sensor array will produce an $m \times n$ discrete image with mn pixels, where each of these LR pixels is of size $L_x \times L_y$. Let r be the upsampling factor, i.e., we would like to construct an image of resolution $rm \times rn$ of the same scene. Then the size of the HR pixels will be $L_x/r \times L_y/r$. Figure 1a shows an example. The big rectangles with solid edges are the LR pixels and the small rectangles with dashed edges are the HR pixels.

Let $\{g_i \in \mathbb{R}^{m\times n}, 1 \le i \le p\}$ be the sequence of LR images produced by the sensor array at different time points, where p is the number of frames. For simplicity we let g_0 be the reference LR image which can be chosen to be any one of the LR images g_i. The displacement of g_i from the reference image g_0 is denoted by $(\epsilon_i^x L_x, \epsilon_i^y L_y)$, see the solid rectangle in Fig. 1a labeled as g_i. For ease of notation, we will represent the 2D images g_i, $0 \le i \le p$, by vectors $\mathbf{g}_i \in \mathbb{R}^{mn}$ obtained by stacking the columns of g_i. We use $\mathbf{f} \in \mathbb{R}^{r^2mn}$ to denote the HR reconstruction of g_0 that we are seeking.

We model the relationship between $\mathbf{f}$ and $\mathbf{g}_0$ by averaging, see [3, 8]. Figure 1b illustrates that the intensity value of the LR pixel is the weighted average of the

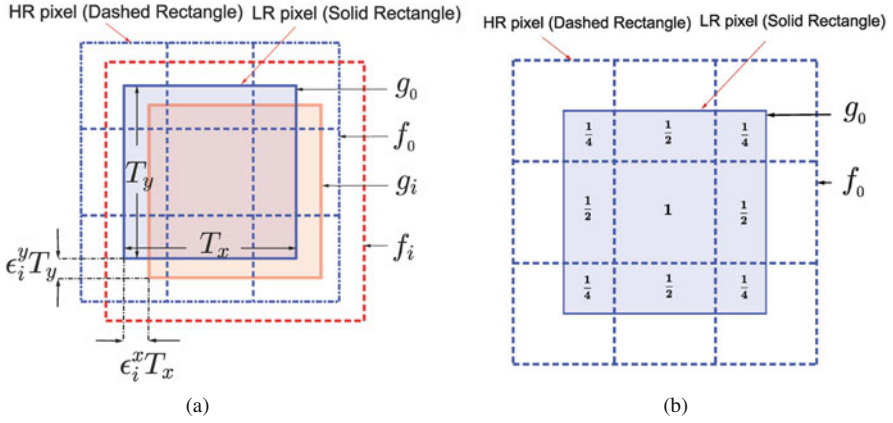


Fig. 1 LR images with displacements. (**a**) Displacements between LR images. (**b**) The averaging process

intensity values of the HR pixels overlapping with it. The weight is precisely the area of overlapping. Thus, the process from $\mathbf{f}$ to each of the LR images g_i can be modeled by [8]

$$\mathbf{g}_i = DKA_i\mathbf{f} + \mathbf{n}_i,\ i = 1, 2, \cdots, p, \tag{4}$$

where $D = D_0(n) \otimes D_0(m) \in \mathbb{R}^{mn \times r^2mn}$ is the downsampling matrix defined by (2); $K \in \mathbb{R}^{r^2mn \times r^2mn}$ is the average operator mentioned above; $A_i \in \mathbb{R}^{r^2mn \times r^2mn}$ is the warping matrix which measures the displacement between g_i and g_0; and $\mathbf{n}_i$ is the additive unknown noise. In this paper, we assume for simplicity that the noise is Gaussian. Other noise models can be handled by choosing suitable data fidelity terms.

The warping matrix A_i, $1 \leq i \leq p$, is to align the LR pixels in $\mathbf{g}_i$ at exactly the middle of the corresponding HR pixels in $\mathbf{f}$, exactly like the $\mathbf{g}_0$ is w.r.t $\mathbf{f}_0$ in Fig. 1b. Once this alignment is done, the average operator K, which is just a blurring operator, can be written out easily. In fact, the 2D kernel (i.e., the point spread function) of K is given by vv^T, where $v = [1/2, 1, \ldots, 1, 1/2]^T$ with $(r-1)$ ones in the middle, see [3]. The A_i are more difficult to obtain. In the most ideal case where the motions are only translation of less than one HR pixel length and width, A_i can be modeled by $A_i = T_n(\epsilon_i^x) \otimes T_m(\epsilon_i^y)$, where $T_n(\epsilon_i^x) and T_m(\epsilon_i^y)$ are the circulant matrices given by (3) with $(\epsilon_i^x L_x, \epsilon_i^y L_y)$ being the horizontal and vertical displacements of g_i, see Fig. 1a and [8]. In reality, the changes between different LR frames are much more complicated. It can involve illumination changes and other complex non-planar motions. We will discuss the formation of A_i in more detail in Sects. 3.1 and 3.3.

3 Nuclear-Norm Model

Given (4), a way to obtain $\mathbf{f}$ is to apply least-squares. However, because D is singular, the problem is ill-posed. Regularization is necessary to make use of some priori information to choose the correct solution. A typical regularizer for solving this problem is *total variation* (TV) [24]. The TV model is well known for edge preserving and can give a reasonable solution for MFSR problems. However, it assumes that the HR image is piecewise constant. This will produce some artifacts.

Instead we will develop a low-rank model for the problem. The main motivation is as follows: We consider each LR image $\mathbf{g}_i$ as a downsampled version of an HR image $\mathbf{f}_i$. If all these HR images $\mathbf{f}_i$ are properly aligned with the HR image $\mathbf{f}$, then they all should be the same exactly (as they are representing the same scene $\mathbf{f}$). W_i is the alignment matrix that aligns $\mathbf{f}_i$ with $\mathbf{f}$. For example, if $p = 2$, and

$$f_1 = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, f_2 = \begin{pmatrix} b \\ c \\ d \end{pmatrix},$$

then we can let

$$W_1 = \begin{pmatrix} 0\,1\,0 \\ 0\,0\,1 \end{pmatrix}, W_2 = \begin{pmatrix} 1\,0\,0 \\ 0\,1\,0 \end{pmatrix},$$

thence

$$W_1\mathbf{f}_1 = \begin{pmatrix} b \\ c \end{pmatrix} = W_2\mathbf{f}_2.$$

In general, $[W_1\mathbf{f}_1, W_2\mathbf{f}_2, \ldots, W_p\mathbf{f}_p]$ should be a low-rank matrix (ideally a rank 1 matrix). Thus, the rank of the matrix can be used as a prior.

In Sect. 3.1, we introduce our low-rank model in the case where the LR images are perturbed only by translations. Then in Sect. 3.2, we explain how to solve the model by the alternating direction method. In Sect. 3.3, we discuss how to modify the model when there are more complex motions or changes between the LR frames.

3.1 Decomposition of the Warping Matrices

In order to introduce our model without too cumbersome notations, we assume first here that the displacements of the LR images from the reference frame are translations only. Let $s_i^x L_x$ and $s_i^y L_y$ be the horizontal and vertical displacements of g_i from g_0. (How to obtain s_i^x and s_i^y will be discussed in Sect. 3.3.) Since the width and height of one HR pixel are L_x/r and L_y/r, respectively, the displacements are equivalent to rs_i^x HR pixel length and rs_i^y HR pixel width. We decompose rs_i^x and rs_i^y into the integral parts and fractional parts:

$$rs_i^x = l_i^x + \epsilon_i^x,\ rs_i^y = l_i^y + \epsilon_i^y, \tag{5}$$

where $l_i^x and l_i^y$ are the integers and $0 \le \epsilon_i^x, \epsilon_i^x < 1$. Then the warping matrix can be decomposed as

$$A_i = C_i B_i, \tag{6}$$

where $B_i = N_n(l_i^x) \otimes N_m(l_i^y)$ is given by (1) and $C_i = T_n(\epsilon_i^x) \otimes T_m(\epsilon_i^y)$ is given by (3) [6]. Thus, by letting $\mathbf{f}_i = B_i\mathbf{f}$, $1 \le i \le p$, (4) can be rewritten as

$$\mathbf{g}_i = DKC_i\mathbf{f}_i + \mathbf{n}_i,\ i = 1, 2, \cdots, p. \tag{7}$$

As mentioned in the motivation above, all these $\mathbf{f}_i$, which are equal to $B_i\mathbf{f}$, are integral shift from $\mathbf{f}$. Hence, if they are aligned correctly by an alignment matrix W_i, the overlapping entries should be the same. Figure 2 is the 1D illustration of this idea. W_i^x is the matrix that aligns $\mathbf{f}_i$ with $\mathbf{f}$ (in the x-direction) and the dark squares

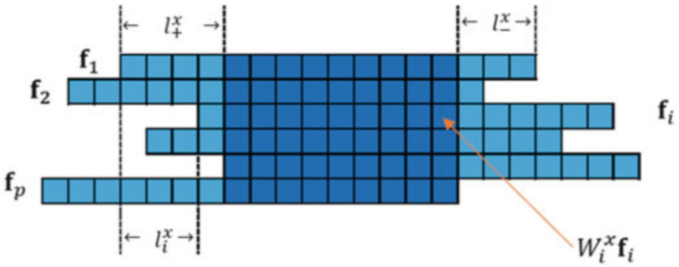


Fig. 2 1D signals with integer displacements

are the overlapping pixels and they should all be the same as the corresponding pixels in $\mathbf{f}$.

Mathematically, W_i is constructed as follows: Given the decomposition of rs_i^x and rs_i^y in (5), let $l_+^x = \max_i\{0, l_i^x\}$, $l_+^y = \max_i\{0, l_i^y\}$ and $l_-^x = \max_i\{0, -l_i^x\}$, $l_-^y = \max_i\{0, -l_i^y\}$. Then

$$W_i = W_i^x \otimes W_i^y, \tag{8}$$

where

$$W_i^x = \begin{bmatrix} 0_{l_+^x - l_i^x} & & \\ & I_{rn-l_+^x-l_-^x} & \\ & & 0_{l_-^x+l_i^x} \end{bmatrix},$$

$$W_i^y = \begin{bmatrix} 0_{l_+^y - l_i^y} & & \\ & I_{rm-l_+^y-l_-^y} & \\ & & 0_{l_-^y+l_i^y} \end{bmatrix}.$$

Note that W_i nullifies the entries outside the overlapping part (i.e., outside the dark squares in Fig. 2).

Ideally, the matrix $[W_1\mathbf{f}_1, W_2\mathbf{f}_2, \cdots, W_p\mathbf{f}_p]$ should be a rank-one matrix as every column should be a replicate of $\mathbf{f}$ in the overlapping region. In practice, it can be of low rank due to various reasons such as errors in measurements and noise in the given video. Since nuclear norm is the convexification of low-rank prior, see [5], this leads to our convex model

$$\min_{\mathbf{f}_1,\cdots,\mathbf{f}_p} \alpha\|W_1\mathbf{f}_1, W_2\mathbf{f}_2, \cdots, W_p\mathbf{f}_p\|_* + \frac{1}{2}\sum_{i=1}^{p}\|\mathbf{g}_i - DKC_i\mathbf{f}_i\|_2^2, \tag{9}$$

where $\|\cdot\|_*$ is the matrix nuclear norm and α is the regularization parameter. We call our model (9) the *nuclear-norm model*. We remark that here we use the 2-norm

data fidelity term because we assume the noise is Gaussian. It can be changed to another norm according to the noise type.

3.2 Algorithm for Solving the Nuclear-Norm Model

We use *alternating direction method of multipliers* (ADMM) [4] to solve the nuclear-norm model. We replace $\{W_i\mathbf{f}_i\}_{i=1}^p$ in the model by variables $\{\mathbf{h}_i\}_{i=1}^p$. Let $H = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_p]$, $F = [\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_p]$, and $WF = [W_1\mathbf{f}_1, W_2\mathbf{f}_2, \cdots, W_p\mathbf{f}_p]$. The *augmented Lagrange* of model (9) is

$$\mathscr{L}_{\alpha\rho}(H, F, \Lambda) = \alpha\|H\|_* + \frac{1}{2}\sum_{i=1}^{p}\|\mathbf{g}_i - DKC_i\mathbf{f}_i\|_2^2 + \sum_{i=1}^{p}\langle\Lambda_i, \mathbf{h}_i - W_i\mathbf{f}_i\rangle + \frac{1}{2\rho}\|H - WF\|_{\mathscr{F}}^2,$$

where $\Lambda = [\Lambda_1, \Lambda_2, \cdots, \Lambda_p]$ is the matrix of Lagrange multipliers, $\|\cdot\|_{\mathscr{F}}$ is the Frobenius norm, and ρ is an algorithm parameter.

To solve the nuclear-norm model, it is equivalent to minimize $\mathscr{L}_{\alpha\rho}$, and we use ADMM [4] to minimize it. The idea of the scheme is to minimize H and F alternatively by fixing the other, i.e., given the initial value F^0, Λ^0, let $H^{k+1} = \arg\min_H \mathscr{L}_{\alpha\rho}(H, F^k, \Lambda^k)$ and $F^{k+1} = \arg\min_F \mathscr{L}_{\alpha\rho}(H^{k+1}, F, \Lambda^k)$, where k is the iteration number. These two problems are convex problems. The *singular value threshold* (SVT) gives the solution of the H-subproblem. The F-subproblem is reduced to solving p linear systems. For a matrix X, the SVT of X is defined to be

$$\mathrm{SVT}_\rho(X) = U\Sigma_\rho^+ V^T,$$

where $X = U\Sigma V^T$ is the singular value decomposition (SVD) of X and $\Sigma_\rho^+ = \max\{\Sigma - \rho, 0\}$. We summarize the algorithm in Algorithm 1. It is well-known that the algorithm is convergent if $\rho > 0$ [4].

In Algorithm 1, the SVT operator involves the SVD of a matrix $WF^k - \Lambda^k$. Its number of column is p, the number of LR frames, which is relatively small. Therefore, the SVT step is not time consuming. For the second subproblem, we need to solve p linear systems. The coefficient matrices contain some structures which help accelerating the calculation. The matrices $D^T D$ and $W_i^T W_i$ are diagonal matrices, while K and C_i can be diagonalized by either FFT or DCT depending on the boundary conditions we choose, see [23]. In our tests, we always use periodic boundary conditions.

In Algorithm 1, within each iteration, we should apply once singular value decomposition to an $r^2mn \times p$ matrix. The complexity of SVD is $O(r^2mnp^2)$. Then, we should solve p linear systems with r^2mn equations. By using FFT, the

Algorithm 1 $\mathbf{f} \leftarrow (\{g_i, W_i, C_i\}, K, \alpha, \rho, \Lambda^0, F^0)$

for $k = 1, 2, 3, \cdots$ **do**
  $H^{k+1} = \mathrm{SVT}_{\alpha\rho}(WF^k - \rho\Lambda^k)$;
  for $i = 1$ to p **do**
    $M_i = (DKC_i)^T DKC_i + \frac{1}{\rho} W_i^T W_i$;
    $\mathbf{f}_i^{k+1} = (M_i)^{-1}\left((DKC_i)^T \mathbf{g}_i + W_i^T \Lambda_i^k + \frac{1}{\rho} W_i^T \mathbf{h}_i^{k+1}\right)$;
  end for
  $\Lambda^{k+1} = \Lambda^k + \frac{1}{\rho}(H^{k+1} - WF^{k+1})$;
end for
Output: $\mathbf{f}$ as the average of the columns of F^k.

complexity for this step is $O(pr^2mn\log(r^2mn))$. Usually, $\log(r^2mn)$ is larger than p. Thence, the overall complexity for Algorithm 1 is $O(pr^2mn\log(r^2mn))$, where $m \times n$ is the size of LR images; r is the upsampling factor; and p is the number of frames.

3.3 Image Registration and Parameter Selection

In Algorithm 1, we assume that there are only translations between different LR frames. However, there can be other complex motions and/or illumination changes in practice. We handle these by using the *local all-pass* (LAP) optical flow algorithm proposed in [13]. Given a set of all-pass filters $\{\phi_j\}_{j=0}^N$ and $\phi := \phi_0 + \sum_{j=1}^{N-1} c_j\phi_j$, the optical flow $\mathscr{M}_i$ of g_i is obtained by solving the following problem:

$$\min_{\{c_1,\cdots,c_{N-1}\}} \sum_{l,k\in R} |(\phi * g_i)(k,l) - (\phi_- * g_0)(k,l)|^2,$$

where $*$ is the convolution operator, R is a window centered at (x, y), and $\phi_-(k,l) = \phi(-k,-l)$. In our experiments, we followed the settings in the paper [13], and let $N = 6$, $R = 16$ and

$$\begin{aligned}
\phi_0(k,l) &= e^{-\frac{k^2+l^2}{2\sigma^2}}, & \phi_1(k,l) &= k\phi_0(k,l),\\
\phi_2(k,l) &= l\phi_0(k,l), & \phi_3(k,l) &= (k^2+l^2-2\sigma^2)\phi_0(k,l),\\
\phi_4(k,l) &= kl\phi_0(k,l), & \phi_5(k,l) &= (k^2-l^2)\phi_0(k,l),
\end{aligned}$$

where $\sigma = \frac{R+2}{4}$ and ϕ is supported in $[-R, R] \times [-R, R]$. The coefficients c_n can be obtained by solving a linear system. The optical flow $\mathscr{M}_i$ at (x, y) is then given by

$$\mathscr{M}_i(x,y) = \left(\frac{2\sum_{k,l} k\phi(k,l)}{\sum_{k,l}\phi(k,l)}, \frac{2\sum_{k,l} l\phi(k,l)}{\sum_{k,l}\phi(k,l)}\right),$$

which can be used to transform g_i back to the grid of g_0. In order to increase the speed by avoiding interpolation, here we consider only the integer part of the flow. Hence, we get the *restored LR images*

$$\tilde{g}_i(x, y) = g_i([\mathscr{M}_i](x, y)),\ i = 1, 2, \cdots, p, \quad \forall (x, y) \in \Omega, \tag{10}$$

where $[\mathscr{M}_i]$ is the integer part of the flow $\mathscr{M}_i$ and Ω is the image domain.

The optical flow method can handle complex motions and illumination changes and will restore the positions of pixels in g_i w.r.t g_0. To enhance the accuracy of the image registration, we further estimate if there are any translations that are unaccounted for after the optical flow. In particular, we assume that $\tilde{g}_i$ may be displaced from g_0 by a simple translation

$$\mathscr{T}(x, y) = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} s_i^x \\ s_i^y \end{bmatrix}. \tag{11}$$

To estimate the displacement vector $[s_i^x, s_i^y]^T$, we use the Levenberg–Marquardt algorithm proposed in [15, 21], which is a well-known method for nonlinear least-squares problems. It aims to minimize the squared error

$$E(\tilde{g}_i, g_0) = \sum_{(x,y)\in\Omega} [\tilde{g}_i(\mathscr{T}(x, y)) - g_0(x, y)]^2. \tag{12}$$

The detailed implementation of this algorithm can be found in [8, Algorithm 3]. After obtaining $[s_i^x, s_i^y]$, then by (6) and (8), we can construct the matrices C_i and W_i for our nuclear-norm model (9).

Before giving out the whole algorithm, there remains the problem about parameters selection. There are two parameters to be determined: α, the regularization parameter, and ρ, the algorithm (ADMM) parameter. We need to tune these two parameters in practice such that the two subproblems can be solved effectively and accurately. Theoretically, ρ will not affect the minimizer of the model but only the convergence of the algorithm [4]. However, in order to get an effective algorithm, it should not be set very small. For α, we use the following empirical formula to approximate it in each iteration [16],

$$\alpha \approx \frac{1/2 \sum_{i=1}^{p} \|\widetilde{\mathbf{g}}_i - DKC_i\mathbf{f}_i^k\|^2}{\|W_1\mathbf{f}_1^k, W_2\mathbf{f}_2^k, \cdots, W_p\mathbf{f}_p^k\|_*}, \tag{13}$$

where $\mathbf{f}_i^k$ is the estimation of $\mathbf{f}_i$ in the kth iteration. The formula may not give the best α but can largely narrow its scope. We then use trial and error to get the best parameter. We give out the full algorithm for our model below.

4 Numerical Experiments

In this section, we illustrate the effectiveness of our algorithm by comparing it with 3 different variational methods on 7 synthetic videos and real videos. Chan et al. [6] applied wavelet analysis to MFSR problem and then developed an iterative approach by using tight-frame wavelet filters [8]. We refer their model as *tight-frame* (TF) model. Li et al. [16] proposed the *sparse directional regularization* (SDR) model where they used l_1 norm of the geometric tight-framelet coefficients as the regularizer and the adaptively mimicking l_1 and l_2 norms as the data fidelity term. Ma et al. [20] introduced an expectation-maximization (EM) framework to the Bayesian approach of Liu and Sun [18]. They also tackled motion blur in their paper. We refer it as the MAP model. We will compare our Algorithm 2 (the nuclear-norm model) with these three methods. The sizes of the videos we used are listed in Table 1. The CPU timing of all methods is also listed. It shows that our method is the fastest, with two exceptions (i.e., the "disk" video when $r = 2$ and the "text" video when $r = 2$). For other instances, our model is the best. We marked the fastest results with bold letters. These data show that, when dealing with small-size images, the SDR model is the fastest. When the size of the images gets larger, our nuclear-norm model is the fastest.

There is one parameter for the TF model—a thresholding parameter η which controls the registration quality of the restored LR images $\tilde{g}_i$ (see (10)). If the PSNR value between $\tilde{g}_i$ and the reference image g_0 is smaller than η, it will discard $\tilde{g}_i$ in the reconstruction. We apply *trial and error* method to choose the best η. For the SDR method, we use the default setting in the paper [16]. Hence, the parameters are selected automatically by the algorithm. The TF model, the SDR model, and the nuclear-norm model are applied to $\tilde{g}_i$, i.e., we use the same optical flow algorithm [13] for these three models. For the MAP model, it utilized an optical flow algorithm from Liu [17]. Following the paper, the optical flow parameter α is very small. We also apply *trial and error* method to tune it.

All the videos used in the tests and the results are available at http://www.math.cuhk.edu.hk/~rchan/paper/super-resolution/experiments.html.

Algorithm 2 $\mathbf{f} \leftarrow (\{g_i\}, i_0, K, \Lambda^0, F^0, \alpha, \rho)$

for $i = 0, 1, 2, \cdots p$ **do**
  Compute $\widetilde{g}_i(x, y)$ from (10);
  Compute s_i^x and s_i^y in (11) by using the Levenberg–Marquardt algorithm in [8, Algorithm 3]
  Compute the warping matrices C_i and W_i, according to (6) and (8);
end for
Apply Algorithm 1 to compute the HR images $\mathbf{f} \leftarrow (\{\widetilde{g}_i, W_i, C_i\}, K, \alpha, \rho, \Lambda^0, F^0)$;
Output **f**.

Table 1 Size of each data set and CPU time for all models

	Size of data			Factor	CPU time (in seconds)			
	Height	Width	Frame	r	TF	MAP	SDR	Nuclear
Boat	240	240	17	2	1251	198	336	**138**
Boat	120	120	17	4	7642	196	282	**94.4**
Bridge	240	240	17	2	3256	202	348	**142**
Bridge	120	120	17	4	9703	189	278	**92.3**
Disk	57	49	19	2	568	**6.4**	28	7.9
Disk	57	49	19	4	5913	21.4	53	**13.6**
Text	57	49	21	2	497	**6.2**	30	**8.5**
Text	57	49	21	4	4517	22.1	56	**14.5**
Alpaca	96	128	21	2	816	26.1	78	**24**
Alpaca	96	128	21	4	6178	172	250	**90.6**
Books	288	352	21	2	3943	1511	818	**689**

4.1 Synthetic Videos

We start from a given HR image $\mathbf{f}^*$, see, e.g., the boat image in Fig. 3f. We translate and rotate $\mathbf{f}^*$ with known parameters and also change their illuminations by different scales. Then we downsample these frames with the given factor $r = 2$ or $r = 4$ to get our LR frames $\{\mathbf{g}_i\}_{i=1}^{p}$. We take $p = 17$, and Gaussian noise of ratio 5% is added to each LR frame.

After we reconstruct the HR image $\mathbf{f}$ by a method, we compare it with the true solution $\mathbf{f}^*$ using two popular error measurements. The first one is *peak signal-to-noise ratio* (PSNR) and the second one is *structural similarity* (SSIM) [29]. For two signals $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ and $\mathbf{y} = (y_1, y_2, \cdots, y_n)^T$, they are defined by

$$\text{PSNR}(\mathbf{x}, \mathbf{y}) = 10 \log_{10} \left(\frac{d^2}{\|\mathbf{x} - \mathbf{y}\|^2 / n} \right),$$

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x \mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)},$$

where d is the dynamic range of $\mathbf{x}, \mathbf{y}$; $\mu_x and \mu_y$ are the mean values of $\mathbf{x}$ and $\mathbf{y}$; $\sigma_x and \sigma_y$ are the variances of $\mathbf{x}$ and $\mathbf{y}$; σ_{xy} is the covariance of $\mathbf{x}$ and $\mathbf{y}$; and $c_i, i = 1, 2$, are the constants related to d, which are typically set to be $c_1 = (0.01d)^2 and c_2 = (0.03d)^2$. Because of the motions, we do not have enough information to reconstruct $\mathbf{f}$ near the boundary; hence, this part of $\mathbf{f}$ will not be accurate. Thus, we restrict the comparison within the overlapping area of all LR images.

Table 2 gives the PSNR values and SSIM values of the reconstructed HR images $\mathbf{f}$ from the boat and the bridge videos. The results show that our model gives much more accurate $\mathbf{f}$ for both upsampling factor $r = 2$ and 4, see the boldfaced values.

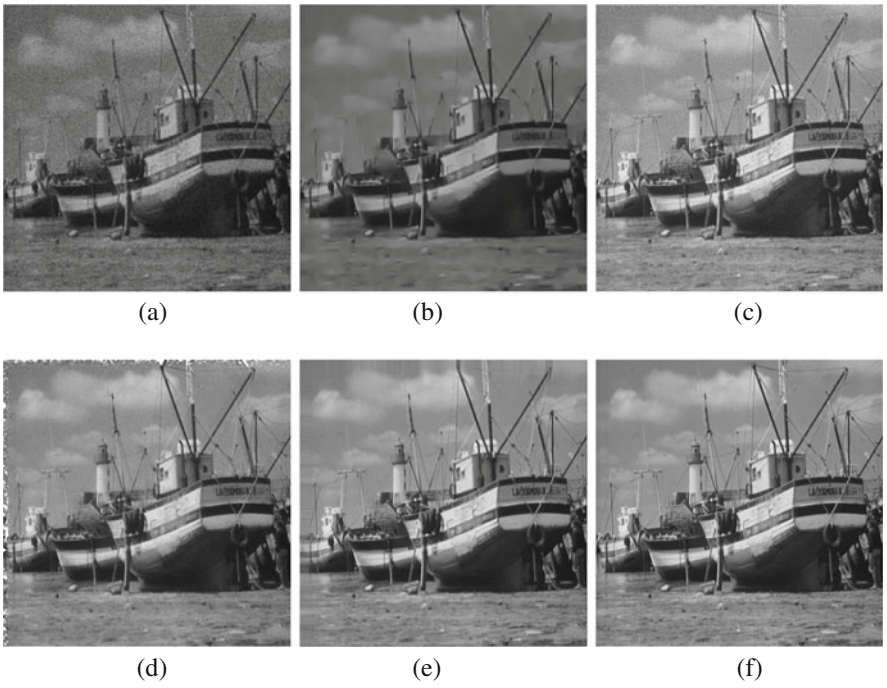


Fig. 3 Comparison of different algorithms on "boat" image with upsampling factor $r = 2$. (**a**) The reference LR image. (**b**) Result of the TF model [8]. (**c**) Result of the MAP model [20]. (**d**) Result of the SDR model [16]. (**e**) Result of our nuclear-norm model ($\alpha = 35.2924$ and $\rho = 3.379 \times 10^4$). (**f**) True HR image

Table 2 PSNR and SSIM values for the "boat" and "bridge" videos

		Upsampling factor $r = 2$				Upsampling factor $r = 4$			
		TF	MAP	SDR	Nuclear	TF	MAP	SDR	Nuclear
Boat	PSNR	18.7	25.3	28.2	**29.8**	20.7	23.6	27.0	**27.1**
	SSIM	0.69	0.70	0.80	**0.82**	0.69	0.67	0.72	**0.76**
Bridge	PSNR	20.7	23.6	**27.0**	26.9	20.1	22.4	24.6	**24.9**
	SSIM	0.69	0.67	0.72	**0.80**	0.53	0.57	0.65	**0.70**

The improvement is significant when comparing to the other three models, e.g., at least 1.6 dB in PSNR for the boat video when $r = 2$. All the PSNR values and SSIM values of our method for boat video are higher than that of other models. All the PSNR values and SSIM values of our method for bridge video are higher than that of other models except the PSNR value when $r = 2$, see the fifth column of the last row. It is comparable with the SDR method. However, the SSIM value is higher. This means the reconstructed structure is better for our method. The major cost of this algorithm is to solve the $\mathbf{f}_i$ subproblems in Algorithm 1. Since the resulting images

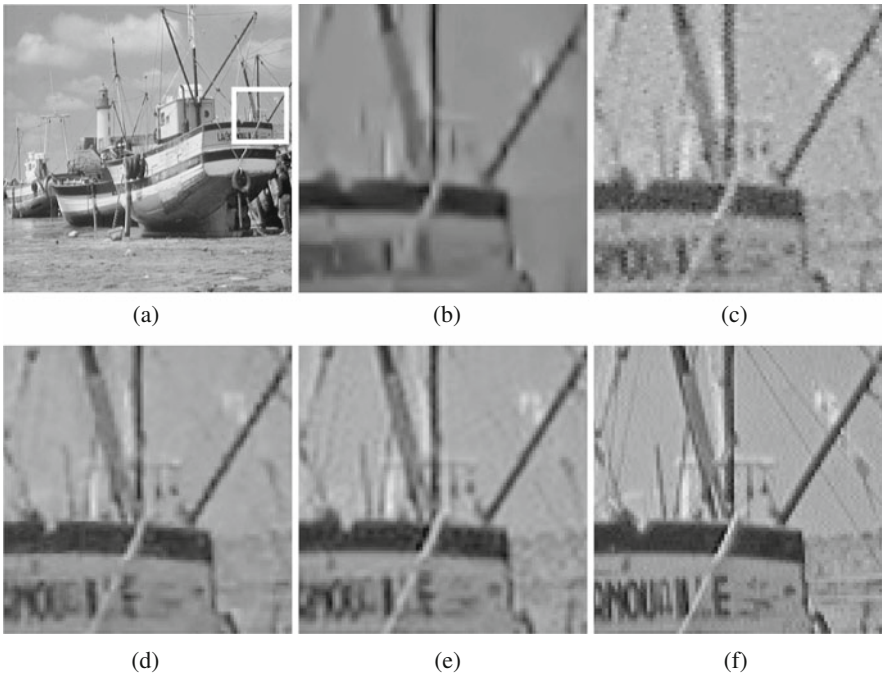


Fig. 4 Zoomed-in comparison of different algorithms on "boat" image for $r = 2$. (**a**) The zoom-in part in the HR image. (**b**) Result of the TF model [8]. (**c**) Result of the MAP model [20]. (**d**) Result of the SDR model [16]. (**e**) Result of our nuclear-norm model ($\alpha = 35.2924$ and $\rho = 3.379 \times 10^4$). (**f**) Zoomed-in original HR image

are with larger sizes, the sizes of coefficients of all subproblems in Algorithm 1 are larger. Thence, when $r = 4$, the cost is larger than that when $r = 2$.

To compare the images visually, we give the results and their zoom-ins for the boat video in Figs. 3, 4, 5. The results for the bridge video are similar and therefore omitted. Figure 3 shows the boat reconstructions for $r = 2$. We notice that the TF model loses many fine details, e.g., the ropes of the mast. The MAP model produces some distortion on the edges and is sensitive to the noise; and the SDR model contains some artifacts along the edges. One can see the difference more clearly from the zoom-in images in Fig. 4. We also give the zoom-in results for $r = 4$ in Fig. 5. We can see that the nuclear-norm model produces more details and less artifacts than the other three models.

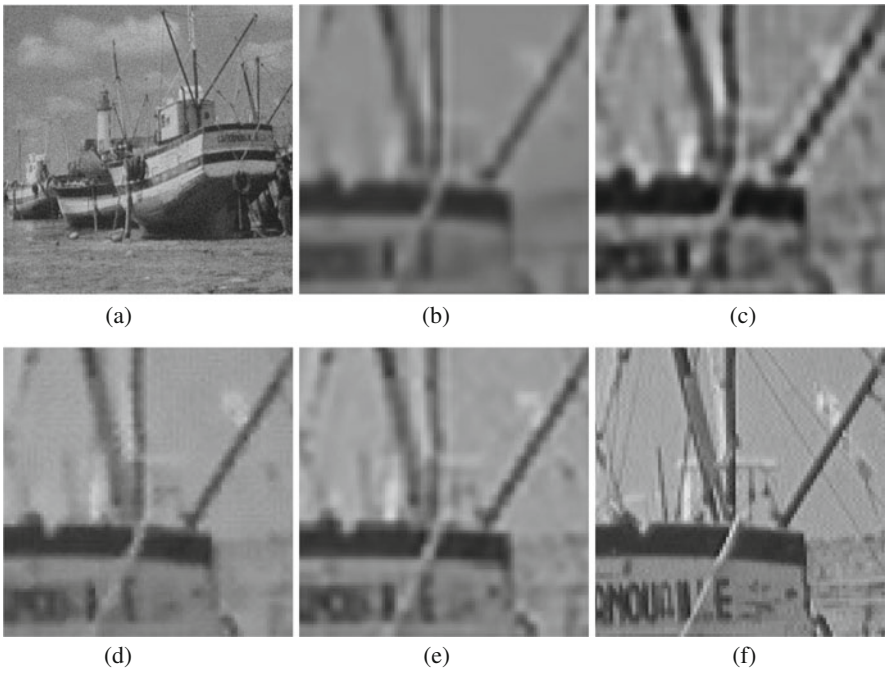


Fig. 5 Zoom-in comparison of different algorithms on "boat" image for $r = 4$. (**a**) The reference LR image. (**b**) Result of the TF model [8]. (**c**) Result of the MAP model [20]. (**d**) Result of the SDR model [16]. (**e**) Result of our nuclear-norm model ($\alpha = 32.0659$ and $\rho = 3.5841 \times 10^4$). (**f**) Zoomed-in original HR image

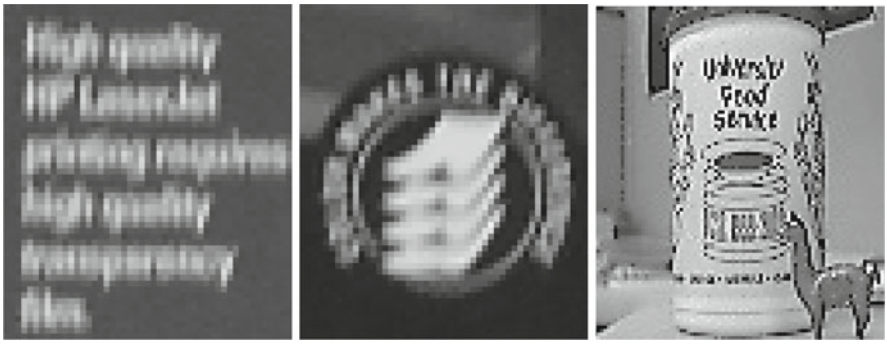
4.2 Real Videos

In the following, experiments on real videos are carried out. Three videos "text," "disk," and "alpaca" are downloaded from the website https://users.soe.ucsc.edu/~milanfar/software/sr-datasets.html.

The basic information of these videos are listed in Table 1. We see that they are very low-resolution videos. Figure 6 shows the reference LR images for these videos. It is difficult to discern most of the letters from the reference images.

The first test video is the "text video." The results are shown in Fig. 7. We see that the TF model produces blurry reconstructions. The images by the MAP model have obvious distortions. We also see that for the SDR model, some of the letters are coalesced, e.g., the word "film." The results of the nuclear-norm model are better. One can easily tell each word and there are no obvious artifacts for the letters.

The second video is the "disk video," which contains 26 gray-scale images with the last 7 ones being zoom-in images. Therefore, we only use the first 19 frames in our experiment. The results are shown in Fig. 8. The TF model again produces blurry reconstructions. The MAP results are better but still blurry. The SDR results



(a) (b) (c)

Fig. 6 The reference LR images of (**a**) "text," (**b**) "disk," and (**c**) "alpaca"

(a) (b) (c) (d)

Fig. 7 Comparison of different algorithms on "text video." Top row with upsampling factor $r = 2$ and second row with $r = 4$. (**a**) Result of the TF model [8]. (**b**) Result of the MAP model [20]. (**c**) Result of the SDR model [16]. (**d**) Result of our nuclear-norm model ($\alpha = 8.368$ and $\rho = 3.6236 \times 10^6$ for $r = 2$; $\alpha = 8.6391$ and $\rho = 4.5618 \times 10^5$ for $r = 4$)

have some artifacts, especially in the word "DIFFERENCE." Our results are the best ones with each letter being well reconstructed, especially when $r = 2$.

The third video is the "alpaca video," and the results are shown in Fig. 9. When $r = 2$, the word "service" is not clear from the TF model, the MAP model, and the SDR model. When $r = 4$, the resulting images from all models are improved and the phrase "university food service" is clear. However, we can see that our nuclear-norm model still gives the best reconstruction.

(a) (b) (c) (d)

Fig. 8 Comparison of different algorithms on "disk video." Top row with upsampling factor $r = 2$ and second row with $r = 4$. (**a**) Result of the TF model [8]. (**b**) Result of the MAP model [20]. (**c**) Result of the SDR model [16]. (**d**) Result of our nuclear-norm model ($\alpha = 6.6802$ and $\rho = 1.0701 \times 10^6$ for $r = 2$; $\alpha = 11.6185$ and $\rho = 8.6404 \times 10^5$ for $r = 4$)

(a) (b) (c) (d)

Fig. 9 Comparison of different algorithms on "alpaca video." Top row with upsampling factor $r = 2$ and second row with $r = 4$. (**a**) Result of the TF model [8]. (**b**) Result of the MAP model [20]. (**c**) Result of the SDR model [16]. (**d**) Result of our nuclear-norm model ($\alpha = 35.3704$ and $\rho = 2.7892 \times 10^4$ for $r = 2$; $\alpha = 45.6486$ and $\rho = 2.9798 \times 10^5$ for $r = 4$)

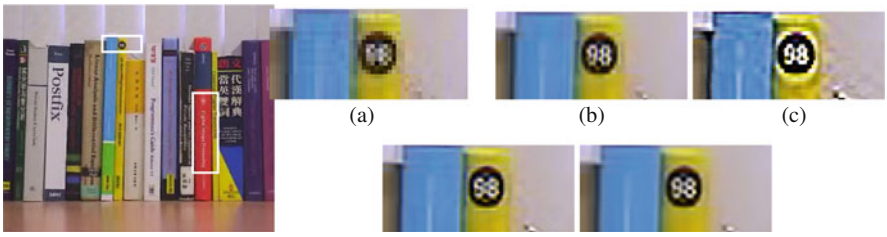


Fig. 10 Zoom-in comparison of different algorithms on "books video" with $r = 2$. Leftmost figure: the LR reference frame with zoom-in areas marked. (**a**) Zoomed-in LR image. (**b**) Result of the TF model [8]. (**c**) Result of the MAP model [20]. (**d**) Result of the SDR model [16]. (**e**) Result of our nuclear-norm model ($\alpha = 15.3958$ and $\rho = 5.6858 \times 10^5$ for $r = 2$)

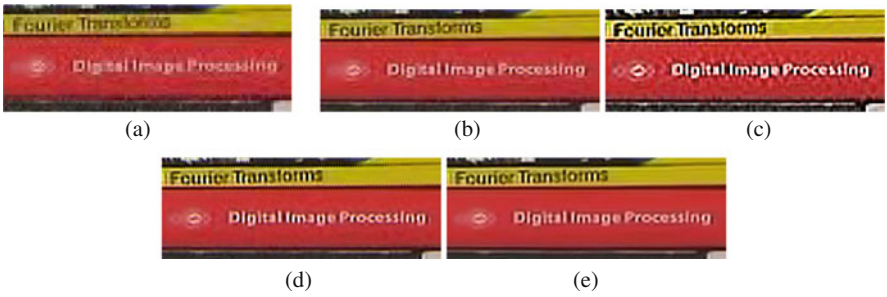


Fig. 11 Another zoom-in comparison on "books video" with $r = 2$. (**a**) Zoomed-in LR image. (**b**) Result of the TF model [8]. (**c**) Result of the MAP model [20]. (**d**) Result of the SDR model [16]. (**e**) Result of our nuclear-norm model ($\alpha = 15.3958$ and $\rho = 5.6858 \times 10^5$ for $r = 2$)

The last video is a color video which is used in the tests in [7, 8]. It contains 257 color frames. We take the 100th frame to be the reference frame, see the leftmost figure in Fig. 10. Frames 90–110 in the video are used as LR images to enhance the reference image. We transform the RGB images into the Ycbcr color space and then apply the algorithms to each color channel. Then we transform the resulting HR images back to the RGB color space. Figures 10 and 11 show the zoom-in patches of the resulting images by different models. In Fig. 10, the patch shows a number "98" on the spine of a book. We see that the TF model gives a reasonable result when compared with MAP and SDR. However, our nuclear-norm model gives the clearest "98" with very clean background. Figure 11 shows the spines of two other books: "Fourier Transforms" and "Digital Image Processing." Again, we see that our nuclear-norm model gives the best reconstruction of the words with much less noisy artifacts.

5 Conclusion

In this paper, we proposed an effective algorithm to reconstruct a high-resolution image using multiple low-resolution images from video clips. The LR images are first registered to the reference frame by using an optical flow. Then a low-rank model is used to reconstruct the high-resolution image by making use of the overlapping information between different LR images. Our model can handle complex motions and illumination changes. Tests on synthetic and real videos show that our model can reconstruct an HR image with much more details and less artifacts.

Acknowledgements This work was supported by HKRGC Grants Nos. CUHK14306316, HKRGC CRF Grant C1007-15G, and HKRGC AoE Grant AoE/M-05/12.

References

1. Altunbasak, Y., Patti, A., Mersereau, R.: Super-resolution still and video reconstruction from mpeg-coded video. IEEE Trans. Circuits Syst. Video Technol. **12**(4), 217–226 (2002)
2. Bishop, C.M., Blake, A., Marthi, B.: Super-resolution enhancement of video. In: Proc. Artificial Intelligence and Statistics, vol. 2. Key West, FL, USA (2003)
3. Bose, N., Boo, K.: High-resolution image reconstruction with multisensors. Int. J. Imaging Syst. Technol. **9**(4), 294–304 (1998)
4. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011)
5. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? J. ACM **58**(3), 11 (2011)
6. Chan, R.H., Chan, T.F., Shen, L., Shen, Z.: Wavelet algorithms for high-resolution image reconstruction. SIAM J. Sci. Comput. **24**(4), 1408–1432 (2003)
7. Chan, R.H., Riemenschneider, S.D., Shen, L., Shen, Z.: Tight frame: an efficient way for high-resolution image reconstruction. Appl. Comput. Harmon. Anal. **17**(1), 91–115 (2004)
8. Chan, R.H., Shen, Z., Xia, T.: A framelet algorithm for enhancing video stills. Appl. Comput. Harmon. Anal. **23**(2), 153–170 (2007)
9. Chen, X., Qi, C.: A single-image super-resolution method via low-rank matrix recovery and nonlinear mappings. In: 20th IEEE International Conference on Image Processing (ICIP), 2013, pp. 635–639 (2013). https://doi.org/10.1109/ICIP.2013.6738131
10. Duponchel, L., Milanfar, P., Ruckebusch, C., Huvenne, J.P.: Super-resolution and Raman chemical imaging: from multiple low resolution images to a high resolution image. Anal. Chim. Acta **607**(2), 168–175 (2008)
11. Farsiu, S., Robinson, M.D., Elad, M., Milanfar, P.: Fast and robust multiframe super resolution. IEEE Trans. Image Process. **13**(10), 1327–1344 (2004)
12. Farsiu, S., Elad, M., Milanfar, P.: Multiframe demosaicing and super-resolution of color images. IEEE Trans. Image Process. **15**(1), 141–159 (2006)
13. Gilliam, C., Blu, T.: Local all-pass filters for optical flow estimation. In: IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 1, IEEE (2015)
14. Jin, C., Nunez-Yanez, J., Achim, A.: Video super-resolution using low rank matrix completion. In: 20th IEEE International Conference on Image Processing (ICIP), 2013, Melbourne, Australia, pp. 1376–1380. (2013)

15. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Q. Appl. Math. **2**, 164–168 (1944)
16. Li, Y.R., Dai, D.Q., Shen, L.: Multiframe super-resolution reconstruction using sparse directional regularization. IEEE Trans. Circuits Syst. Video Technol. **20**(7), 945–956 (2010)
17. Liu, C.: Beyond pixels: exploring new representations and applications for motion analysis. Ph.D. thesis, Citeseer (2009)
18. Liu, C., Sun, D.: On Bayesian adaptive video super resolution. IEEE Trans. Pattern Anal. Mach. Intell. **36**(2), 346–360 (2014)
19. Lu, Y., Shen, L., Xu, Y.: Multi-parameter regularization methods for high-resolution image reconstruction with displacement errors. IEEE Trans. Circuits Syst. Regul. Pap. **54**(8), 1788–1799 (2007)
20. Ma, Z., Liao, R., Tao, X., Xu, L., Jia, J., Wu, E.: Handling motion blur in multi-frame super-resolution. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5224–5232 (2015). http://dx.doi.org/10.1109/CVPR.2015.7299159
21. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. **11(2)**, 431–441 (1962)
22. Narayanan, B., Hardie, R.C., Barner, K.E., Shao, M.: A computationally efficient super-resolution algorithm for video processing using partition filters. IEEE Trans. Circuits Syst. Video Technol. **17**(5), 621–634 (2007)
23. Ng, M.K., Chan, R.H., Tang, W.C.: A fast algorithm for deblurring models with Neumann boundary conditions. SIAM J. Sci. Comput. **21**(3), 851–866 (1999)
24. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**(1), 259–268 (1992)
25. Shankar, P.M., Neifeld, M.A.: Sparsity constrained regularization for multiframe image restoration. JOSA A **25**(5), 1199–1214 (2008)
26. Shen, L., Sun, Q.: Biorthogonal wavelet system for high-resolution image reconstruction. IEEE Trans. Signal Process. **52**(7), 1997–2011 (2004)
27. Takeda, H., Milanfar, P., Protter, M., Elad, M.: Super-resolution without explicit subpixel motion estimation. IEEE Trans. Image Process. **18**(9), 1958–1975 (2009)
28. Tsai, R., Huang, T.: Multiframe image restoration and registration. Adv. Comput. Vis. Image Process. **1**(2), 317–339 (1984)
29. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004). http://dx.doi.org/10.1109/TIP.2003.819861
30. Wang, C., Xue, P., Lin, W.: Improved super-resolution reconstruction from video. IEEE Trans. Circuits Syst. Video Technol. **16**(11), 1411–1422 (2006)
31. Zibetti, M.V.W., Mayer, J.: A robust and computationally efficient simultaneous super-resolution scheme for image sequences. IEEE Trans. Circuits Syst. Video Technol. **17**(10), 1288–1300 (2007)