

Lecture Notes in Mathematics 2173
CIME Foundation Subseries

Michele Benzi · Dario Bini
Daniel Kressner · Hans Munthe-Kaas
Charles Van Loan

Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications

Cetraro, Italy 2015

Michele Benzi
Valeria Simoncini *Editors*



Editors-in-Chief:

J.-M. Morel, Cachan

B. Teissier, Paris

Advisory Board:

Camillo De Lellis, Zurich

Mario di Bernardo, Bristol

Michel Brion, Grenoble

Alessio Figalli, Zurich

Davar Khoshnevisan, Salt Lake City

Ioannis Kontoyiannis, Athens

Gabor Lugosi, Barcelona

Mark Podolskij, Aarhus

Sylvia Serfaty, New York

Anna Wienhard, Heidelberg

Fondazione C.I.M.E., Firenze



C.I.M.E. stands for *Centro Internazionale Matematico Estivo*, that is, International Mathematical Summer Centre. Conceived in the early fifties, it was born in 1954 in Florence, Italy, and welcomed by the world mathematical community: it continues successfully, year for year, to this day.

Many mathematicians from all over the world have been involved in a way or another in C.I.M.E.'s activities over the years. The main purpose and mode of functioning of the Centre may be summarised as follows: every year, during the summer, sessions on different themes from pure and applied mathematics are offered by application to mathematicians from all countries. A Session is generally based on three or four main courses given by specialists of international renown, plus a certain number of seminars, and is held in an attractive rural location in Italy.

The aim of a C.I.M.E. session is to bring to the attention of younger researchers the origins, development, and perspectives of some very active branch of mathematical research. The topics of the courses are generally of international resonance. The full immersion atmosphere of the courses and the daily exchange among participants are thus an initiation to international collaboration in mathematical research.

C.I.M.E. Director (2002 – 2014)

Pietro Zecca

Dipartimento di Energetica "S. Stecco"

Università di Firenze

Via S. Marta, 3

50139 Florence

Italy

e-mail: zecca@unifi.it

C.I.M.E. Director (2015 –)

Elvira Mascolo

Dipartimento di Matematica "U. Dini"

Università di Firenze

viale G.B. Morgagni 67/A

50134 Florence

Italy

e-mail: mascolo@math.unifi.it

C.I.M.E. Secretary

Paolo Salani

Dipartimento di Matematica "U. Dini"

Università di Firenze

viale G.B. Morgagni 67/A

50134 Florence

Italy

e-mail: salani@math.unifi.it

CIME activity is carried out with
the collaboration and financial support
of INdAM (Istituto Nazionale di Alta
Matematica)

For more information see CIME's homepage:
<http://www.cime.unifi.it>

Michele Benzi • Dario Bini • Daniel Kressner •
Hans Munthe-Kaas • Charles Van Loan

Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications

Cetraro, Italy 2015

Michele Benzi • Valeria Simoncini

Editors



Springer



FONDAZIONE
CIME
ROBERTO CONTI

CENTRO INTERNAZIONALE MATEMATICO ESTIVO
INTERNATIONAL MATHEMATICAL SUMMER CENTER

Authors

Michele Benzi
Mathematics and Science Center
Emory University
Atlanta, USA

Daniel Kressner
École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland

Charles Van Loan
Department of Computer Science
Cornell University
Ithaca
New York, USA

Dario Bini
Università di Pisa
Pisa, Italy

Hans Munthe-Kaas
Department of Mathematics
University of Bergen
Bergen, Norway

Editors

Michele Benzi
Mathematics and Science Center
Emory University
Atlanta, USA

Valeria Simoncini
Dipartimento di Matematica
Università di Bologna
Bologna, Italy

ISSN 0075-8434

Lecture Notes in Mathematics

ISBN 978-3-319-49886-7

DOI 10.1007/978-3-319-49887-4

ISSN 1617-9692 (electronic)

ISBN 978-3-319-49887-4 (eBook)

Library of Congress Control Number: 2016963344

Mathematics Subject Classification (2010): 65Fxx, 65Nxx

© Springer International Publishing AG 2016

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature

The registered company is Springer International Publishing AG

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume collects the notes of the lectures delivered at the CIME Summer Course “Exploiting Hidden Structure in Matrix Computations. Algorithms and Applications,” held in Cetraro, Italy, from June 22 to June 26, 2015.

The course focused on various types of hidden and approximate structure in matrices and on their key role in various emerging applications. Matrices with special structure arise frequently in scientific and engineering problems and have long been the object of study in numerical linear algebra, as well as in matrix and operator theory. For instance, banded matrices or Toeplitz matrices fall under this category. More recently, however, researchers have begun to investigate classes of matrices for which structural properties, while present, may not be immediately obvious. Important examples include matrices with low-rank off-diagonal block structure, such as hierarchical matrices, quasi-separable matrices, and so forth. Another example is provided by matrices in which the entries follow some type of decay pattern away from the main diagonal. The structural analysis of complex networks leads to matrices that appear at first sight to be completely unstructured; a closer look, however, may reveal a great deal of latent structure, for example, in the distribution of the nonzero entries and of the eigenvalues. Knowledge of these properties can be of great importance in the design of efficient numerical methods for such problems.

In many cases, the matrix is only “approximately” structured; for instance, a matrix could be close in some norm to a matrix that has a desirable structure, such as a banded matrix or a semiseparable matrix. In such cases, it may be possible to develop solution algorithms that exploit the “near-structure” present in the matrix; for example, efficient preconditioners could be developed for the nearby structured problem and applied to the original problem. In other cases, the solution of the nearby problem, for which efficient algorithms exist, may be a sufficiently good approximation to the solution of the original problem, and the main difficulty could be detecting the “nearest” structured problem.

Another very useful kind of structure is represented by various types of symmetries. Symmetries in matrices and tensors are often linked to invariance under some group of transformations. Again, the structure, or near-structure, may not be

immediately obvious in a given problem: uncovering the hidden symmetries (and underlying transformation group) in a problem can potentially lead to very efficient algorithms.

The aim of this course was to present this increasingly important point of view to young researchers by exploiting the expertise of leading figures in this area, with different theoretical and application perspectives. The course was attended by 31 PhD students and young researchers, roughly half of them from Italy and the remaining ones from the USA, Great Britain, Germany, the Netherlands, Spain, Croatia, Czech Republic, and Finland. Many participants (about 50%) were at least partially supported by funding from CIME and the European Mathematical Society; other financial support was provided by the Università di Bologna.

Two evenings (for a total of four hours) were devoted to short 15 minute presentations by 16 of the participants on their current research activities. These contributions were of high quality and very enjoyable, contributing to the success of the meeting.

The notes collected in this volume are a faithful record of the material covered in the 28 hours of lectures comprising the course. Charles Van Loan (Cornell University, USA) discussed structured matrix computations originating from tensor analysis, in particular tensor decompositions, with low-rank and Kronecker structure playing the main role. Dario Bini (Università di Pisa, Italy) covered matrices with Toeplitz and Toeplitz-like structure, as well as rank-structured matrices, with applications to Markov modeling and queueing theory. Daniel Kressner (EPFL, Switzerland) lectured on techniques for matrices with hierarchical low-rank structures, including applications to the numerical solution of elliptic PDEs; Jonas Ballani (EPFL) also contributed to the drafting of the corresponding lecture notes contained in this volume. Michele Benzi (Emory University) discussed matrices with decay, with particular emphasis on localization in matrix functions, with applications to quantum physics and network science. The lectures of Hans Munthe-Kaas (University of Bergen, Norway) concerned the use of group-theoretic methods (including representation theory) in numerical linear algebra, with applications to the fast Fourier transform (“harmonic analysis on finite groups”) and to computational aspects of Lie groups and Lie algebras.

We hope that these lecture notes will be of use to young researchers in numerical linear algebra and scientific computing and that they will stimulate further research in this area.

We would like to express our sincere thanks to Elvira Mascolo (CIME Director) and especially to Paolo Salani (CIME Secretary), whose unfailing support proved crucial for the success of the course. We are also grateful to the European Mathematical Society for their endorsement and financial support, and to the Dipartimento di Matematica of Università di Bologna for additional support for the lecturers.

Atlanta, GA, USA
Bologna, Italy

Michele Benzi
Valeria Simoncini

Acknowledgments

CIME activity is carried out with the collaboration and financial support of INdAM (Istituto Nazionale di Alta Matematica).

Contents

Structured Matrix Problems from Tensors	1
Charles F. Van Loan	
Matrix Structures in Queueing Models	65
Dario A. Bini	
Matrices with Hierarchical Low-Rank Structures	161
Jonas Ballani and Daniel Kressner	
Localization in Matrix Computations: Theory and Applications	211
Michele Benzi	
Groups and Symmetries in Numerical Linear Algebra.....	319
Hans Z. Munthe-Kaas	

Structured Matrix Problems from Tensors

Charles F. Van Loan

Abstract This chapter looks at the structured matrix computations that arise in the context of various “svd-like” tensor decompositions. Kronecker products and low-rank manipulations are central to the theme. Algorithmic details include the exploitation of partial symmetries, componentwise optimization, and how we might beat the “curse of dimensionality.” Order-4 tensors figure heavily in the discussion.

1 Introduction

A tensor is a multi-dimensional array. Instead of just $A(i, j)$ as for matrices we have $A(i, j, k, \ell, \dots)$. High-dimensional modeling, cheap storage, and sensor technology combine to explain why tensor computations are surging in importance. Here is an annotated timeline that helps to put things in perspective:

Scalar-Level Thinking

1960's ↓ The factorization paradigm: LU , LDL^T , QR , $U\Sigma V^T$, etc.

Matrix-Level Thinking

1980's ↓ Memory traffic awareness:, cache, parallel computing, LAPACK, etc.

Block Matrix-Level Thinking

2000's ↓ Matrix-tensor connections: unfoldings, Kronecker product, multilinear optimization, etc.

Tensor-Level Thinking

C.F. Van Loan (✉)

Department of Computer Science, Cornell University, Ithaca, NY, USA
e-mail: cv@cs.cornell.edu

An important subtext is the changing definition of what we mean by a “big problem.” In matrix computations, to say that $A \in \mathbb{R}^{n_1 \times n_2}$ is “big” is to say that both n_1 and n_2 are big. In tensor computations, to say that $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ is “big” is to say that $n_1 n_2 \dots n_d$ is big and this does NOT necessarily require big n_k . For example, no computer in the world (nowadays at least!) can store a tensor with modal dimensions $n_1 = n_2 = \dots = n_{1000} = 2$. What this means is that a significant part of the tensor research community is preoccupied with the development of algorithms that scale with d . Algorithmic innovations must deal with the “curse of dimensionality.” How the transition from *matrix-based* scientific computation to *tensor-based* scientific computation plays out is all about the fate of the “the curse.”

This chapter is designed to give readers who are somewhat familiar with matrix computations an idea about the underlying challenges associated with tensor computations. These include mechanisms by which tensor computations are turned into matrix computations and how various matrix algorithms and decompositions (especially the SVD) turn up all along the way. An important theme throughout is the exploitation of Kronecker product structure.

To set the tone we use Sect. 2 to present an overview of some remarkable “hidden structures” that show up in matrix computations. Each of the chosen examples has a review component and a message about tensor-based matrix computations. The connection between block matrices and order-4 tensors is used in Sect. 3 to introduce the idea of a tensor unfolding and to connect Kronecker products and tensor products. A simple nearest rank-1 tensor problem is used in Sect. 4 to showcase the idea of componentwise optimization, a strategy that is widely used in tensor computations. In Sect. 5 we show how Rayleigh quotients can be used to extend the notion of singular values and vectors to tensors. Transposition and tensor symmetry are discussed in Sect. 6. Extending the singular value decomposition to tensors can be done in a number of ways. We present the Tucker decomposition in Sect. 7, the CP decomposition in Sect. 8, the Kronecker product SVD in Sect. 9, and the tensor train SVD in Sect. 10. Cholesky with column pivoting also has a role to play in tensor computations as we show in Sect. 11.

We want to stress that this chapter is a high-level, informal look at the kind of matrix problems that arise out of tensor computations. Implementation details and rigorous analysis are left to the references. To get started with the literature and for general background we recommend [6, 10, 15, 16, 18, 27].

2 The Exploitation of Structure in Matrix Computations

We survey five interesting matrix examples that showcase the idea of hidden structure. By “hidden” we mean “not obvious”. In each case the exploitation of the hidden structure has important ramifications from the computational point of view.

2.1 Exploiting Data Sparsity

The n -by- n discrete Fourier transform matrix F_n is defined by

$$[F_n]_{kq} = \omega_n^{kq} \quad \omega_n = \cos\left(\frac{2\pi}{n}\right) - i \sin\left(\frac{2\pi}{n}\right)$$

where we are subscripting from zero. Thus,

$$F_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega_4 & \omega_4^2 & \omega_4^3 \\ 1 & \omega_4^2 & \omega_4^4 & \omega_4^6 \\ 1 & \omega_4^3 & \omega_4^6 & \omega_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}.$$

If we carefully reorder the columns of F_{2m} , then copies of F_m magically appear, e.g.,

$$F_4 \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & -i & i \\ 1 & 1 & -1 & -1 \\ 1 & -1 & i & -i \end{bmatrix} = \begin{bmatrix} F_2 & \Omega_2 F_2 \\ F_2 & -\Omega_2 F_2 \end{bmatrix}$$

where

$$\Omega_2 = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}.$$

In general we have

$$F_{2m} \Pi_{2,m} = \begin{bmatrix} F_m & \Omega_m F_m \\ F_m & -\Omega_m F_m \end{bmatrix} \tag{1}$$

where $\Pi_{2,m}$ is a *perfect shuffle* permutation (to be described in Sect. 3.7) and Ω_m is the diagonal matrix

$$\Omega_m = \text{diag}(1, \omega_n, \dots, \omega_n^{m-1}).$$

The DFT matrix is dense, but by exploiting the recursion (1) it can be factored into a product of sparse matrices, e.g.,

$$F_{1024} = A_{10} \cdots A_2 A_1 P^T.$$

Here, P is the bit reversal permutation and each A_k has just two nonzeros per row. It is this structured factorization that makes it possible to have a *fast*, $O(n \log n)$ Fourier transform, e.g.,

```

 $y = P^T x$ 
for  $k = 1 : 10$ 
     $y = A_k y$ 
end
```

See [29] for a detailed “matrix factorization” treatment of the FFT.

The DFT matrix F_n is *data sparse* meaning that it can be represented with many fewer than n^2 numbers. Other examples of data sparsity include matrices that have low-rank and matrices that are Kronecker products. *Many tensor problems lead to matrix problems that are data sparse.*

2.2 Exploiting Structured Eigensystems

Suppose $A, F, G \in \mathbb{R}^{n \times n}$ and that both F and G are symmetric. The matrix M defined by

$$M = \begin{bmatrix} A & F \\ G & -A^T \end{bmatrix} \quad F = F^T, \quad G = G^T$$

is said to be a *Hamiltonian* matrix. The eigenvalues of a Hamiltonian matrix come in plus-minus pairs and the eigenvectors associated with such a pair are related:

$$M \begin{bmatrix} y \\ z \end{bmatrix} = \lambda \begin{bmatrix} y \\ z \end{bmatrix} \Rightarrow M^T \begin{bmatrix} z \\ -y \end{bmatrix} = -\lambda \begin{bmatrix} z \\ -y \end{bmatrix}.$$

Hamiltonian structure can also be defined through a permutation similarity. If

$$J_{2n} = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix}$$

then $M \in \mathbb{R}^{2n \times 2n}$ is Hamiltonian if $J_{2n}^T M J_{2n} = -M^T$. Under mild assumptions we can compute a structured Schur decomposition for a Hamiltonian matrix M :

$$Q^T M Q = \begin{bmatrix} Q_1 & Q_2 \\ -Q_2 & Q_1 \end{bmatrix}^T M \begin{bmatrix} Q_1 & Q_2 \\ -Q_2 & Q_1 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ 0 & -T_{11}^T \end{bmatrix}.$$

Here, Q is orthogonal and symplectic ($J_{2n}^T Q J_{2n} = Q^{-T}$) and T_{11} is upper quasi-triangular. Various Riccati equation problems can be solved by exploiting this structured decomposition. See [22].

Tensors with multiple symmetries can be reshaped into matrices with multiple symmetries and these matrices have structured block factorizations.

2.3 Exploiting the Right Representation

Here is an example of a *Cauchy-like* matrix:

$$A = \begin{bmatrix} \frac{r_1 s_1}{\omega_1 - \lambda_1} & \frac{r_1 s_2}{\omega_1 - \lambda_2} & \frac{r_1 s_3}{\omega_1 - \lambda_3} & \frac{r_1 s_4}{\omega_1 - \lambda_4} \\ \frac{r_2 s_1}{\omega_2 - \lambda_1} & \frac{r_2 s_2}{\omega_2 - \lambda_2} & \frac{r_2 s_3}{\omega_2 - \lambda_3} & \frac{r_2 s_4}{\omega_2 - \lambda_4} \\ \frac{r_3 s_1}{\omega_3 - \lambda_1} & \frac{r_3 s_2}{\omega_3 - \lambda_2} & \frac{r_3 s_3}{\omega_3 - \lambda_3} & \frac{r_3 s_4}{\omega_3 - \lambda_4} \\ \frac{r_4 s_1}{\omega_4 - \lambda_1} & \frac{r_4 s_2}{\omega_4 - \lambda_2} & \frac{r_4 s_3}{\omega_4 - \lambda_3} & \frac{r_4 s_4}{\omega_4 - \lambda_4} \end{bmatrix}.$$

For this to be defined we must have $\{\lambda_1, \dots, \lambda_n\} \cup \{\mu_1, \dots, \mu_n\} = \emptyset$. Cauchy-like matrices are data sparse and a particularly clever characterization of this fact is to note that if $\Omega = \text{diag}(\omega_i)$ and $\Lambda = \text{diag}(\lambda_i)$, then

$$\Omega A - A \Lambda = rs^T \quad (2)$$

where $r, s \in \mathbb{R}^n$. If $\Omega A - A \Lambda$ has rank r , then we say that A has *displacement rank* r (with respect to Ω and Λ .) Thus, a Cauchy-like matrix has unit displacement rank.

Now let us consider the first step of Gaussian elimination. Ignoring pivoting this involves computing a row of U , a column of L , and a rank-1 update:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \ell_{21} & 1 & 0 & 0 \\ \ell_{31} & 0 & 1 & 0 \\ \ell_{41} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & b_{22} & b_{23} & b_{24} \\ 0 & b_{32} & b_{33} & b_{34} \\ 0 & b_{42} & b_{43} & b_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

It is easy to compute the required entries of L and U from the displacement rank representation (2). What about B ? If we represent B conventionally as an array then that would involve $O(n^2)$ flops. Instead we exploit the fact that B also has unit displacement rank:

$$\tilde{\Omega}B - B\tilde{\Lambda} = \tilde{r}\tilde{s}^T.$$

It turns out that we can transition from A 's representation $\{\Omega, \Lambda, r, s\}$ to B 's representation $\{\tilde{\Omega}, \tilde{\Lambda}, \tilde{r}, \tilde{s}\}$ with $O(n)$ work and this enables us to compute the LU factorization of a Cauchy-like matrix with just $O(n^2)$ work. See [10, p. 682].

Being able to work with clever representations is often the key to having a successful solution framework for a tensor problem.

2.4 Exploiting Orthogonality Structures

Assume that the columns of the 2-by-1 block matrix

$$Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$

are orthonormal, i.e., $Q_1^T Q_1 + Q_2^T Q_2 = I$. The *CS decomposition* says that Q_1 and Q_2 have related SVDs:

$$\begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}^T \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}_V = \begin{bmatrix} \text{diag}(c_i) \\ \text{diag}(s_i) \end{bmatrix} \quad c_i^2 + s_i^2 = 1 \quad (3)$$

where U_1 , U_2 , and V are orthogonal. This truly remarkable hidden structure can be used to compute stably the *generalized singular value decomposition* (GSVD) of a $A_1 \in \mathbb{R}^{m_1 \times n}$ and $A_2 \in \mathbb{R}^{m_2 \times n}$. In particular, suppose we compute the QR factorization

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R,$$

and then the CS decomposition (3). By setting $X = R^T V$, we obtain the GSVD

$$A_1 = U_1 \cdot \text{diag}(c_i) \cdot X^T \quad A_2 = U_2 \cdot \text{diag}(s_i) \cdot X^T.$$

For a more detailed discussion about the GSVD and the CS decomposition, see [10, p. 309].

A tensor decomposition can often be regarded as a simultaneous decomposition of its (many) matrix “layers”.

2.5 Exploiting a Structured Data layout

Suppose we have a block matrix

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1} & A_{M2} & \cdots & A_{MN} \end{bmatrix}$$

that is stored in such a way that the data in each A_{ij} is contiguous in memory. Note that there is nothing structured about “A-the-matrix”. However, “A-the-stored-array” does have an exploitable structure that we now illustrate by considering the computation of $C = A^T$. We start by transposing each of A’s blocks:

$$\begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1N} \\ B_{21} & B_{22} & \cdots & B_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{M1} & B_{M2} & \cdots & B_{MN} \end{bmatrix} \leftarrow \begin{bmatrix} A_{11}^T & A_{12}^T & \cdots & A_{1N}^T \\ A_{21}^T & A_{22}^T & \cdots & A_{2N}^T \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1}^T & A_{M2}^T & \cdots & A_{MN}^T \end{bmatrix}.$$

These block transpositions involve “local data” and this is important because moving data around in a large scale matrix computation is typically the dominant cost. Next, we transpose B as a block matrix:

$$\begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1M} \\ C_{21} & C_{22} & \cdots & C_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N1} & C_{2N} & \cdots & C_{NM} \end{bmatrix} \leftarrow \begin{bmatrix} B_{11} & B_{21} & \cdots & B_{M1} \\ B_{12} & B_{22} & \cdots & B_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ B_{1N} & B_{2N} & \cdots & B_{MN} \end{bmatrix}$$

Again, this is a “memory traffic friendly” maneuver because blocks of contiguous data are being moved. It is easy to verify that $C_{ij} = A_{ji}^T$.

What we have sketched is a “2-pass” transposition procedure. By blocking in a way that resonates with cache/local memory size and by breaking the overall transposition process down into a sequence of carefully designed passes, one can effectively manage the underlying dataflow. See [10, p. 711].

Transposition and looping are much more complicated with tensors because there are typically an exponential number of possible data structures and an exponential number of possible loop nestings. Software tools that facilitate reasoning in this space are essential. See [1, 24, 28].

3 Matrix-Tensor Connections

Tensor computations typically get “reshaped” into matrix computations. To operate in this venue we need terminology and mechanisms for matricizing the tensor data. In this section we introduce the notion of a tensor unfolding and we get comfortable with Kronecker products and their properties. For more details see [10, 16, 18, 25, 27].

3.1 Talking About Tensors

An order- d tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a real d -dimensional array

$$\mathcal{A}(1 : n_1, \dots, 1 : n_d)$$

where the index range in the k -th *mode* is from 1 to n_k . Note that

$$\text{a } \begin{cases} \text{scalar} \\ \text{vector} \\ \text{matrix} \end{cases} \text{ is an } \begin{cases} \text{order-0} \\ \text{order-1} \\ \text{order-2} \end{cases} \text{ tensor.}$$

We use calligraphic font to designate tensors e.g., $\mathcal{A}, \mathcal{B}, \mathcal{C}$, etc. Sometimes we will write \mathcal{A} for matrix A if it makes things clear.

One way that tensors arise is through discretization. $\mathcal{A}(i, j, k, \ell)$ might house the value of $f(w, x, y, z)$ at $(w, x, y, z) = (w_i, x_j, y_k, z_\ell)$. In multiway analysis the value of $\mathcal{A}(i, j, k, \ell)$ could measure the interaction between four variables/factors. See [2] and [27].

3.2 Tensor Parts: Fibers and Slices

A *fiber* of a tensor \mathcal{A} is a column vector obtained by fixing all but one \mathcal{A} 's indices. For example, if $\mathcal{A} = \mathcal{A}(1:3, 1:5, 1:4, 1:7) \in \mathbb{R}^{3 \times 5 \times 4 \times 7}$, then

$$\mathcal{A}(2, :, 4, 6) = \mathcal{A}(2, 1:5, 4, 6) = \begin{bmatrix} \mathcal{A}(2, 1, 4, 6) \\ \mathcal{A}(2, 2, 4, 6) \\ \mathcal{A}(2, 3, 4, 6) \\ \mathcal{A}(2, 4, 4, 6) \\ \mathcal{A}(2, 5, 4, 6) \end{bmatrix}$$

is a mode-2 fiber.

A *slice* of a tensor \mathcal{A} is a matrix obtained by fixing all but two of \mathcal{A} 's indices. For example, if $\mathcal{A} = \mathcal{A}(1:3, 1:5, 1:4, 1:7)$, then

$$\mathcal{A}(:, 3, :, 6) = \begin{bmatrix} \mathcal{A}(1, 3, 1, 6) & \mathcal{A}(1, 3, 2, 6) & \mathcal{A}(1, 3, 3, 6) & \mathcal{A}(1, 3, 4, 6) \\ \mathcal{A}(2, 3, 1, 6) & \mathcal{A}(2, 3, 2, 6) & \mathcal{A}(2, 3, 3, 6) & \mathcal{A}(2, 3, 4, 6) \\ \mathcal{A}(3, 3, 1, 6) & \mathcal{A}(3, 3, 2, 6) & \mathcal{A}(3, 3, 3, 6) & \mathcal{A}(3, 3, 4, 6) \end{bmatrix}$$

is a slice.

3.3 Order-4 Tensors and Block Matrices

Block matrices with uniformly-sized blocks are reshaped order-4 tensors. For example, if

$$C = \left[\begin{array}{cc|cc|cc} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\ \hline c_{31} & c_{32} & c_{33} & c_{34} & c_{35} & c_{36} \\ c_{41} & c_{42} & c_{43} & c_{44} & c_{45} & c_{46} \\ \hline c_{51} & c_{52} & c_{53} & c_{54} & c_{55} & c_{56} \\ c_{61} & c_{62} & c_{63} & c_{64} & c_{65} & c_{66} \end{array} \right]$$

then matrix entry c_{45} is entry (2,1) of block (2,3). Thus, we can think of $[C_{ij}]_{kl}$ as the (i, j, k, ℓ) entry of a tensor \mathcal{C} , e.g.,

$$c_{45} = [C_{23}]_{21} = \mathcal{C}(2, 3, 2, 1).$$

Working in the other direction we can *unfold* an order-4 tensor into a block matrix. Suppose $\mathcal{A} \in \mathbb{R}^{n \times n \times n \times n}$. Here is its “[1, 2] × [3, 4]” unfolding:

$$\mathcal{A}_{[1,2] \times [3,4]} = \left[\begin{array}{ccc|ccc|ccc} a_{1111} & a_{1112} & a_{1113} & a_{1121} & a_{1122} & a_{1123} & a_{1131} & a_{1132} & a_{1133} \\ a_{1211} & a_{1212} & a_{1213} & a_{1221} & a_{1222} & a_{1223} & a_{1231} & a_{1232} & a_{1233} \\ a_{1311} & a_{1312} & a_{1313} & a_{1321} & a_{1322} & a_{1323} & a_{1331} & a_{1332} & a_{1333} \\ \hline a_{2111} & a_{2112} & a_{2113} & a_{2121} & a_{2122} & a_{2123} & a_{2131} & a_{2132} & a_{2133} \\ a_{2211} & a_{2212} & a_{2213} & a_{2221} & a_{2222} & a_{2223} & a_{2231} & a_{2232} & a_{2233} \\ a_{2311} & a_{2312} & a_{2313} & a_{2321} & a_{2322} & a_{2323} & a_{2331} & a_{2332} & a_{2333} \\ \hline a_{3111} & a_{3112} & a_{3113} & a_{3121} & a_{3122} & a_{3123} & a_{3131} & a_{3132} & a_{3133} \\ a_{3211} & a_{3212} & a_{3213} & a_{3221} & a_{3222} & a_{3223} & a_{3231} & a_{3232} & a_{3233} \\ a_{3311} & a_{3312} & a_{3313} & a_{3321} & a_{3322} & a_{3323} & a_{3331} & a_{3332} & a_{3333} \end{array} \right].$$

If $A = \mathcal{A}_{[1,2] \times [3,4]}$ then the tensor-to-matrix mapping is given by

$$\mathcal{A}(i_1, i_2, i_3, i_4) \rightarrow A(i_1 + (i_2 - 1)n, i_3 + (i_4 - 1)n).$$

An alternate unfolding results if modes 1 and 3 are associated with rows and modes 2 and 4 are associated with columns:

$$\mathcal{A}_{[1,3] \times [2,4]} = \left[\begin{array}{ccc|ccc|ccc} a_{1111} & a_{1112} & a_{1113} & a_{1211} & a_{1212} & a_{1213} & a_{1311} & a_{1312} & a_{1313} \\ a_{1121} & a_{1122} & a_{1123} & a_{1221} & a_{1222} & a_{1223} & a_{1321} & a_{1322} & a_{1323} \\ a_{1131} & a_{1132} & a_{1133} & a_{1231} & a_{1232} & a_{1233} & a_{1331} & a_{1332} & a_{1333} \\ \hline a_{2111} & a_{2112} & a_{2113} & a_{2211} & a_{2212} & a_{2213} & a_{2311} & a_{2312} & a_{2313} \\ a_{2121} & a_{2122} & a_{2123} & a_{2221} & a_{2222} & a_{2223} & a_{2321} & a_{2322} & a_{2323} \\ a_{2131} & a_{2132} & a_{2133} & a_{2231} & a_{2232} & a_{2233} & a_{2331} & a_{2332} & a_{2333} \\ \hline a_{3111} & a_{3112} & a_{3113} & a_{3211} & a_{3212} & a_{3213} & a_{3311} & a_{3312} & a_{3313} \\ a_{3121} & a_{3122} & a_{3123} & a_{3221} & a_{3222} & a_{3223} & a_{3321} & a_{3322} & a_{3323} \\ a_{3131} & a_{3132} & a_{3133} & a_{3231} & a_{3232} & a_{3233} & a_{3331} & a_{3332} & a_{3333} \end{array} \right].$$

If $A = \mathcal{A}_{[1,3] \times [2,4]}$, then the tensor-to-matrix mapping is given by

$$\mathcal{A}(i_1, i_2, i_3, i_4) \rightarrow A(i_1 + (i_3 - 1)n, i_2 + (i_4 - 1)n).$$

If a tensor \mathcal{A} is structured, then different unfoldings reveal that structure in different ways [31]. The idea of block unfoldings is discussed in [25].

3.4 Modal Unfoldings

A particularly important class of tensor unfoldings are the *modal* unfoldings. An order- d tensor has d modal unfoldings which we designate by $\mathcal{A}_{(1)}, \dots, \mathcal{A}_{(d)}$. Let's look at the tensor-to-matrix mappings for the case $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$:

$$\mathcal{A}(i_1, i_2, i_3, i_4) \rightarrow \mathcal{A}_{(1)}(i_1, i_2 + (i_3 - 1)n_2 + (i_4 - 1)n_2n_3)$$

$$\mathcal{A}(i_1, i_2, i_3, i_4) \rightarrow \mathcal{A}_{(2)}(i_2, i_1 + (i_3 - 1)n_1 + (i_4 - 1)n_1n_3)$$

$$\mathcal{A}(i_1, i_2, i_3, i_4) \rightarrow \mathcal{A}_{(3)}(i_3, i_1 + (i_2 - 1)n_1 + (i_4 - 1)n_1n_2)$$

$$\mathcal{A}(i_1, i_2, i_3, i_4) \rightarrow \mathcal{A}_{(4)}(i_4, i_1 + (i_2 - 1)n_1 + (i_3 - 1)n_1n_2).$$

Note that if $N = n_1 \cdots n_d$, then $\mathcal{A}_{(k)}$ is n_k -by- N/n_k and its columns are mode- k fibers. Thus, if $n_2 = 4$ and $n_1 = n_3 = n_4 = 2$, then

$$\mathcal{A}_{(2)} = \begin{bmatrix} a_{1111} & a_{1121} & a_{1112} & a_{1122} & a_{2111} & a_{2121} & a_{2112} & a_{2122} \\ a_{1211} & a_{1221} & a_{1212} & a_{1222} & a_{2211} & a_{2221} & a_{2212} & a_{2222} \\ a_{1311} & a_{1321} & a_{1312} & a_{1322} & a_{2311} & a_{2321} & a_{2312} & a_{2322} \\ a_{1411} & a_{1421} & a_{1412} & a_{1422} & a_{2411} & a_{2421} & a_{2412} & a_{2422} \end{bmatrix}.$$

Modal unfoldings arise naturally in many multilinear optimization settings.

3.5 The vec Operation

The *vec* operator turns tensors into column vectors. The vec of a matrix is obtained by stacking its columns:

$$A \in \mathbb{R}^{3 \times 2} \Rightarrow \text{vec}(A) = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \end{bmatrix}.$$

The vec of an order-3 tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ stacks the vecs of the slices $\mathcal{A}(:,:,1), \dots, \mathcal{A}(:,:,n_3)$, e.g.,

$$\mathcal{A} \in \mathbb{R}^{2 \times 2 \times 2} \Rightarrow \text{vec}(\mathcal{A}) = \begin{bmatrix} \text{vec}(\mathcal{A}(:,:,1)) \\ \text{vec}(\mathcal{A}(:,:,2)) \end{bmatrix} = \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix}.$$

In general, if $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and the order $d - 1$ tensor \mathcal{A}_k is defined by $\mathcal{A}_k = \mathcal{A}(:,:,\dots,:,k)$, then we have the following recursive definition:

$$\text{vec}(\mathcal{A}) = \begin{bmatrix} \text{vec}(\mathcal{A}_1) \\ \vdots \\ \text{vec}(\mathcal{A}_{n_d}) \end{bmatrix}.$$

Thus, vec unfolds a tensor into a column vector.

3.6 The Kronecker Product

Unfoldings enable us to reshape tensor computations as matrix computations and *Kronecker products* are very often part of the scene. The Kronecker product $A = B \otimes C$ of two matrices B and C is a block matrix (A_{ij}) where $A_{ij} = b_{ij}C$, e.g.,

$$A = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} b_{11}C & b_{12}C & b_{13}C \\ \hline b_{21}C & b_{22}C & b_{23}C \\ \hline b_{31}C & b_{32}C & b_{33}C \end{bmatrix}.$$

Of course, $B \otimes C$ is also a matrix of scalars:

$$A = \begin{bmatrix} b_{11}c_{11} & b_{11}c_{12} & b_{12}c_{11} & b_{12}c_{12} & b_{13}c_{11} & b_{13}c_{12} \\ \hline b_{11}c_{21} & b_{11}c_{22} & b_{12}c_{21} & b_{12}c_{22} & b_{13}c_{21} & b_{13}c_{22} \\ \hline b_{21}c_{11} & b_{21}c_{12} & b_{22}c_{11} & b_{22}c_{12} & b_{23}c_{11} & b_{23}c_{12} \\ \hline b_{21}c_{21} & b_{21}c_{22} & b_{22}c_{21} & b_{22}c_{22} & b_{23}c_{21} & b_{23}c_{22} \\ \hline b_{31}c_{11} & b_{31}c_{12} & b_{32}c_{11} & b_{32}c_{12} & b_{33}c_{11} & b_{33}c_{12} \\ \hline b_{31}c_{21} & b_{31}c_{22} & b_{32}c_{21} & b_{32}c_{22} & b_{33}c_{21} & b_{33}c_{22} \end{bmatrix}.$$

Note that every possible product $b_{ij}c_{kl}$ “shows up” in $B \otimes C$.

In general, if $A_1 \in \mathbb{R}^{m_1 \times n_1}$ and $A_2 \in \mathbb{R}^{m_2 \times n_2}$, then $A = A_1 \otimes A_2$ is an $m_1 m_2$ -by- $n_1 n_2$ matrix of scalars. It is also an m_1 -by- n_1 block matrix with blocks that are m_2 -by- n_2 .

Kronecker products can be applied in succession. Thus, if

$$A = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} \otimes \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \end{bmatrix},$$

then A is a 3-by-2 block matrix whose entries are 4-by-4 block matrices whose entries are 3-by-4 matrices.

It is important to have a facility with the Kronecker product operation because they figure heavily in tensor computations. Here are three critical properties:

$$(A \otimes B)(C \otimes D) = AC \otimes BD$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

$$(A \otimes B)^T = A^T \otimes B^T.$$

Of course, in these expressions the various matrix products and inversions have to be defined.

If the matrices A and B have structure, then their Kronecker product is typically structured in the same way. For example, if $Q_1 \in \mathbb{R}^{m_1 \times n_1}$ and $Q_2 \in \mathbb{R}^{m_2 \times n_2}$ have orthonormal columns, then $Q_1 \otimes Q_2$ has orthonormal columns:

$$\begin{aligned}(Q_1 \otimes Q_2)^T(Q_1 \otimes Q_2) &= (Q_1^T \otimes Q_2^T)(Q_1 \otimes Q_2) \\ &= (Q_1^T Q_1) \otimes (Q_2^T Q_2) = I_{n_1} \otimes I_{n_2} = I_{n_1 n_2}.\end{aligned}$$

Kronecker products often arise through the “vectorization” of a matrix equation, e.g.,

$$C = BXA^T \quad \Leftrightarrow \quad \text{vec}(C) = (A \otimes B) \text{vec}(X).$$

3.7 Perfect Shuffles, Kronecker Products, and Transposition

In general, $A_1 \otimes A_2 \neq A_2 \otimes A_1$. However, very structured permutation matrices P_1 and P_2 exist so that $P_1(A_1 \otimes A_2)P_2 = A_2 \otimes A_1$. Define the (p, q) -perfect shuffle matrix $\Pi_{p,q} \in \mathbb{R}^{pq \times pq}$ by

$$\Pi_{p,q} = [I_{pq}(:, 1 : q : pq) \mid I_{pq}(:, 2 : q : pq) \mid \cdots \mid I_{pq}(:, q : q : pq)].$$

Here is an example:

$$\Pi_{3,2} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right].$$

In general $A_1 \otimes A_2 \neq A_2 \otimes A_1$. However, if $A_1 \in \mathbb{R}^{m_1 \times n_1}$ and $A_2 \in \mathbb{R}^{m_2 \times n_2}$ then

$$\Pi_{m_1, m_2}(A_1 \otimes A_2)\Pi_{n_1, n_2}^T = A_2 \otimes A_1.$$

The perfect shuffle is also “behind the scenes” when the transpose of a matrix is taken, e.g.,

$$\Pi_{3,2} \text{vec}(A) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \\ a_{31} \\ a_{32} \end{bmatrix} = \text{vec}(A^T).$$

In general, if $A \in \mathbb{R}^{m \times n}$ then $\Pi_{m,n}\text{vec}(A) = \text{vec}(A^T)$. See [12]. We return to these interconnections when we discuss tensor transposition in Sect. 6.

3.8 Tensor Notation

It is often perfectly adequate to illustrate a tensor computation idea using order-3 examples. For example, suppose $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $X_1 \in \mathbb{R}^{m_1 \times n_1}$, $X_2 \in \mathbb{R}^{m_2 \times n_2}$, and $X_3 \in \mathbb{R}^{m_3 \times n_3}$ are given and that we wish to compute

$$\mathcal{B}(i_1, i_2, i_3) = \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \sum_{j_3=1}^{n_3} \mathcal{A}(j_1, j_2, j_3) X_1(i_1, j_1) X_2(i_2, j_2) X_3(i_3, j_3)$$

where $1 \leq i_1 \leq m_1$, $1 \leq i_2 \leq m_2$, and $1 \leq i_3 \leq m_3$. Here we are using matrix-like subscript notation to spell out the definition of \mathcal{B} . We could probably use the same notation to describe the order-4 version of this computation. However, for higher-order cases we have to resort to the dot-dot-dot notation and it gets pretty unwieldy:

$$\mathcal{B}(i_1, \dots, i_d) = \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \cdots \sum_{j_d=1}^{n_d} \mathcal{A}(j_1, \dots, j_d) X_1(i_1, j_1) \cdots X_d(i_d, j_d).$$

$$1 \leq i_1 \leq m_1, 1 \leq i_2 \leq m_2, \dots, 1 \leq i_d \leq m_d$$

One way to streamline the presentation of such a calculation is to “vectorize” the notation using bold font to indicate vectors of subscripts. Multiple summations can also be combined through vectorization. Thus, if

$$\mathbf{i} = [i_1, \dots, i_d], \quad \mathbf{j} = [j_1, \dots, j_d], \quad \mathbf{m} = [m_1, \dots, m_d], \quad \mathbf{n} = [n_1, \dots, n_d],$$

then the \mathcal{B} tensor given above can be expressed as follows:

$$\mathcal{B}(\mathbf{i}) = \sum_{\mathbf{j}=1}^{\mathbf{n}} \mathcal{A}(\mathbf{j}) X_1(i_1, j_1) \cdots X_d(i_d, j_d), \quad \mathbf{1} \leq \mathbf{i} \leq \mathbf{m}.$$

Here, $\mathbf{1} = [1, 1, \dots, 1]$. As another example of this notation, if $\mathbf{n} = [n_1, \dots, n_d]$ and $\mathcal{A} \in \mathbb{R}^{\mathbf{n}}$, then

$$\|\mathcal{A}\|_F = \sqrt{\sum_{i=1}^n \mathcal{A}(i)^2}$$

is its Frobenius norm. We shall make use of this vectorized notation whenever it is necessary to hide detail and/or when we are working with tensors of arbitrary order.

Finally, it is handy to have a MATLAB “reshape” notation. Suppose $\mathbf{n} = [n_1, \dots, n_d]$ and $\mathbf{m} = [m_1, \dots, m_e]$. If $\mathcal{A} \in \mathbb{R}^{\mathbf{n}}$ and $n_1 \cdots n_d = m_1 \cdots m_e$, then

$$\mathcal{B} = \text{reshape}(\mathcal{A}, \mathbf{m})$$

is the $m_1 \times \cdots \times m_e$ tensor defined by $\text{vec}(\mathcal{A}) = \text{vec}(\mathcal{B})$.

3.9 The Tensor Product

On occasion it is handy to talk about operations between tensors without recasting the discussion in the language of matrices. Suppose $\mathbf{n} = [n_1, \dots, n_d]$ and that $\mathcal{B}, \mathcal{C} \in \mathbb{R}^{\mathbf{n}}$. We can multiply a tensor by a scalar,

$$\mathcal{A} = \alpha \mathcal{B} \Leftrightarrow \mathcal{A}(i) = \alpha \mathcal{B}(i), \quad 1 \leq i \leq \mathbf{n}$$

and we can add one tensor to another,

$$\mathcal{A} = \mathcal{B} + \mathcal{C} \Leftrightarrow \mathcal{A}(i) = \mathcal{B}(i) + \mathcal{C}(i), \quad 1 \leq i \leq \mathbf{n}.$$

Slightly more complicated is the *tensor product* which is a way of multiplying two tensors together to obtain a new, higher order tensor. For example if $\mathcal{B} \in \mathbb{R}^{m_1 \times m_2 \times m_3} = \mathbb{R}^{\mathbf{m}}$ and $\mathcal{C} \in \mathbb{R}^{n_1 \times n_2} = \mathbb{R}^{\mathbf{n}}$, then the tensor product $\mathcal{A} = \mathcal{B} \circ \mathcal{C}$ is defined by

$$\mathcal{A}(i_1, i_2, i_3, j_1, j_2) = \mathcal{B}(i_1, i_2, i_3)\mathcal{C}(j_1, j_2)$$

i.e., $\mathcal{A}(\mathbf{i}, \mathbf{j}) = \mathcal{B}(\mathbf{i})\mathcal{C}(\mathbf{j})$ for all $1 \leq \mathbf{i} \leq \mathbf{m}$ and $1 \leq \mathbf{j} \leq \mathbf{n}$.

If $\mathcal{B} \in \mathbb{R}^{\mathbf{m}}$ and $\mathcal{C} \in \mathbb{R}^{\mathbf{n}}$, then there is a connection between the tensor product $\mathcal{A} = \mathcal{B} \circ \mathcal{C}$ and its \mathbf{m} -by- \mathbf{n} unfolding:

$$\mathcal{A}_{\mathbf{m} \times \mathbf{n}} = \text{vec}(\mathcal{B})\text{vec}(\mathcal{C})^T.$$

There is also a connection between the tensor product of two vectors and their Kronecker product:

$$\text{vec}(x \circ y) = \text{vec}(xy^T) = y \otimes x.$$

Likewise, if \mathcal{B} and \mathcal{C} are order-2 tensors and $\mathcal{A} = \mathcal{B} \circ \mathcal{C}$, then

$$\mathcal{A}_{[1,3] \times [2,4]} = \mathcal{B} \otimes \mathcal{C}.$$

The point of all this notation-heavy discussion is to stress the importance of flexibility and point of view. Whether we write $\mathcal{A}(\mathbf{i})$ or $\mathcal{A}(i_1, i_2, i_3)$ or $a_{i_1 i_2 i_3}$ depends on the context, what we are trying to communicate, and what typesets nicely! Sometimes it will be handy to regard \mathcal{A} as a vector such as $\text{vec}(\mathcal{A})$ and sometimes as a matrix such as $\mathcal{A}_{(3)}$. Algorithmic insights in tensor computations frequently require an ability to “reshape” how the problem at hand is viewed.

4 A Rank-1 Tensor Problem

Rank-1 matrices have a prominent role to play in matrix computations. For example, one step of Gaussian elimination involves a rank-1 update of a submatrix. The SVD decomposes a matrix into a sum of very special rank-1 matrices. Quasi-Newton methods for nonlinear systems involve rank-1 modifications of the current approximate Jacobian matrix.

In this section we introduce the concept of a rank-1 tensor and consider how we might approximate a given tensor with such an entity. This leads to a discussion (through an example) of multilinear optimization.

4.1 Rank-1 Matrices

If u and v are vectors, then $A = uv^T$ is a rank-1 matrix, e.g.,

$$A = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}^T = \begin{bmatrix} u_1 v_1 & u_1 v_2 \\ u_2 v_1 & u_2 v_2 \\ u_3 v_1 & u_3 v_2 \end{bmatrix}.$$

Note that if $A = uv^T$, then $\text{vec}(A) = v \otimes u$ and so we have

$$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{12} \\ a_{22} \\ a_{32} \end{bmatrix} = \begin{bmatrix} u_1 v_1 \\ u_2 v_1 \\ u_3 v_1 \\ u_1 v_2 \\ u_2 v_2 \\ u_3 v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \otimes \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}.$$

4.2 Rank-1 Tensors

How can we extend the rank-1 idea from matrices to tensors? In matrix computations we think of rank-1 matrices as *outer products*, i.e., $A = uv^T$ where u and v are vectors. Thinking of matrix A as tensor \mathcal{A} , we see that it is just the tensor product of u and v : $\mathcal{A}(i_1, i_2) = u(i_1)v(i_2)$. Thus, we have

$$\mathcal{A} = u \circ v = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \circ \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \Leftrightarrow \text{vec}(\mathcal{A}) = \begin{bmatrix} u_1 v_1 \\ u_2 v_1 \\ u_3 v_1 \\ u_1 v_2 \\ u_2 v_2 \\ u_3 v_2 \end{bmatrix} = v \otimes u.$$

Here is an order-3 example of the same idea:

$$\mathcal{A} = u \circ v \circ w = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \circ \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \circ \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \Leftrightarrow \text{vec}(\mathcal{A}) = \begin{bmatrix} u_1 v_1 w_1 \\ u_2 v_1 w_1 \\ u_3 v_1 w_1 \\ u_1 v_2 w_1 \\ u_2 v_2 w_1 \\ u_3 v_2 w_1 \\ u_1 v_1 w_2 \\ u_2 v_1 w_2 \\ u_3 v_1 w_2 \\ u_1 v_2 w_2 \\ u_2 v_2 w_2 \\ u_3 v_2 w_2 \end{bmatrix} = w \otimes v \otimes u.$$

Each entry in \mathcal{A} is a product of entries from u , v , and w : $\mathcal{A}(p, q, r) = u_p v_q w_r$.

In general, a rank-1 tensor is a tensor product of vectors. To be specific, if $x^{(i)} \in \mathbb{R}^{n_i}$ for $i = 1, \dots, d$, then

$$\mathcal{A} = x^{(1)} \circ \dots \circ x^{(d)}$$

is a rank-1 tensor whose entries are defined by $\mathcal{A}(\mathbf{i}) = x_{i_1}^{(1)} \cdots x_{i_d}^{(d)}$. In terms of the Kronecker product we have $\text{vec}(x^{(1)} \circ \cdots \circ x^{(d)}) = x^{(d)} \otimes \cdots \otimes x^{(1)}$.

4.3 The Nearest Rank-1 Problem for Matrices

Given a matrix $A \in \mathbb{R}^{m \times n}$, consider the minimization of

$$\phi(\sigma, u, v) = \|A - \sigma uv^T\|_F$$

where $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^n$ have unit 2-norm and σ is a nonnegative scalar. This is an SVD problem for if $U^T A V = \Sigma = \text{diag}(\sigma_i)$ is the SVD of A and $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$, then $\phi(\sigma, u, v)$ is minimized by setting $\sigma = \sigma_1$, $u = U(:, 1)$, and $v = V(:, 1)$.

Instead of explicitly computing the entire SVD, we can compute σ_1 and its singular vectors using an *alternating least squares* approach. The starting point is to realize that

$$\|A - \sigma uv^T\|_F^2 = \text{tr}(A^T A) - 2\sigma u^T A v + \sigma^2.$$

where $\text{tr}(M)$ indicates the trace of a matrix M , i.e., the sum of its diagonal entries. Note that

$$\phi_u(y) = \text{tr}(A^T A) - 2y^T A v + \|y\|_2^2$$

is minimized by setting $y = Av$ and that

$$\phi_v(x) = \text{tr}(A^T A) - 2u^T A v + \|x\|_2^2$$

is minimized by setting $x = A^T u$. This suggests the following iterative framework for minimizing $\|A - \sigma uv^T\|_F$:

Nearest Rank-1 Matrix

Given: $A \in \mathbb{R}^{m \times n}$, $v \in \mathbb{R}^n$, $\|v\|_2 = 1$

Repeat:

Fix v and choose σ and u to minimize $\|A - \sigma uv^T\|_F$:

$$y = Av; \quad \sigma = \|y\|; \quad u = y/\sigma$$

Fix u and choose σ and v to minimize $\|A - \sigma uv^T\|_F$:

$$x = A^T u; \quad \sigma = \|x\|; \quad v = x/\sigma$$

$$\sigma_{\text{opt}} = \sigma; \quad u_{\text{opt}} = u; \quad v_{\text{opt}} = v$$

This is basically just the power method applied to the matrix

$$\text{sym}(A) = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}.$$

The reason for bringing up this alternating least squares framework is that it readily extends to tensors.

4.4 A Nearest Rank-1 Tensor Problem

Given $\mathcal{A} \in \mathbb{R}^{m \times n \times p}$, we wish to determine unit vectors $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$, and $w \in \mathbb{R}^p$ and a scalar σ so that the following is minimized:

$$\| \mathcal{A} - \sigma \cdot u \circ v \circ w \|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^p (a_{ijk} - u_i v_j w_k)^2}.$$

Noting that

$$\| \mathcal{A} - \sigma \cdot u \circ v \circ w \|_F = \| \text{vec}(\mathcal{A}) - \sigma \cdot w \otimes v \otimes u \|_2$$

we obtain the following alternating least squares framework:

Nearest Rank-1 Tensor

Given: $\mathcal{A} \in \mathbb{R}^{m \times n \times p}$ and unit vectors $v \in \mathbb{R}^n$ and $w \in \mathbb{R}^p$.

Repeat:

Determine $x \in \mathbb{R}^m$ that minimizes $\| \text{vec}(\mathcal{A}) - w \otimes v \otimes x \|_2$
and set $\sigma = \| x \|$ and $u = x/\sigma$.

Determine $y \in \mathbb{R}^n$ that minimizes $\| \text{vec}(\mathcal{A}) - w \otimes y \otimes u \|_2$
and set $\sigma = \| y \|$ and $v = y/\sigma$.

Determine $z \in \mathbb{R}^p$ that minimizes $\| \text{vec}(\mathcal{A}) - z \otimes v \otimes u \|_2$
and set $\sigma = \| z \|$ and $w = z/\sigma$.

$$\sigma_{\text{opt}} = \sigma, \quad u_{\text{opt}} = u, \quad v_{\text{opt}} = v, \quad w_{\text{opt}} = w$$

It is instructive to examine some of the details associated with this iteration for the case $m = n = p = 2$. The objective function has the form

$$\phi(\sigma, \theta_1, \theta_2, \theta_3) = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \sigma \cdot w \otimes v \otimes u \right\| = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \sigma \cdot \begin{bmatrix} c_3c_2c_1 \\ c_3c_2s_1 \\ c_3s_2c_1 \\ c_3s_2s_1 \\ s_3c_2c_1 \\ s_3c_2s_1 \\ s_3s_2c_1 \\ s_3s_2s_1 \end{bmatrix} \right\|$$

where

$$u = \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix} = \begin{bmatrix} c_1 \\ s_1 \end{bmatrix}, \quad v = \begin{bmatrix} \cos(\theta_2) \\ \sin(\theta_2) \end{bmatrix} = \begin{bmatrix} c_2 \\ s_2 \end{bmatrix}, \quad w = \begin{bmatrix} \cos(\theta_3) \\ \sin(\theta_3) \end{bmatrix} = \begin{bmatrix} c_3 \\ s_3 \end{bmatrix}.$$

Let us look at the three structured *linear* least squares problems that arise during each iteration.

- (1) To improve θ_1 and σ , we fix θ_2 and θ_3 and minimize

$$\left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \sigma \cdot \begin{bmatrix} c_3c_2c_1 \\ c_3c_2s_1 \\ c_3s_2c_1 \\ c_3s_2s_1 \\ s_3c_2c_1 \\ s_3c_2s_1 \\ s_3s_2c_1 \\ s_3s_2s_1 \end{bmatrix} \right\| = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \begin{bmatrix} c_3c_2 & 0 \\ 0 & c_3c_2 \\ c_3s_2 & 0 \\ 0 & c_3s_2 \\ s_3c_2 & 0 \\ 0 & s_3c_2 \\ s_3s_2 & 0 \\ 0 & s_3s_2 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \right\|$$

with respect to x_1 and y_1 . We then set $\sigma = \sqrt{x_1^2 + y_1^2}$ and $u = [x_1 \ y_1]^T / \sigma$.

- (2) To improve θ_2 and σ , we fix θ_1 and θ_3 and minimize

$$\left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \sigma \cdot \begin{bmatrix} c_3c_2c_1 \\ c_3c_2s_1 \\ c_3s_2c_1 \\ c_3s_2s_1 \\ s_3c_2c_1 \\ s_3c_2s_1 \\ s_3s_2c_1 \\ s_3s_2s_1 \end{bmatrix} \right\| = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \begin{bmatrix} c_3c_1 & 0 \\ c_3s_1 & 0 \\ 0 & c_3c_1 \\ 0 & c_3s_1 \\ s_3c_1 & 0 \\ s_3s_1 & 0 \\ 0 & s_3c_1 \\ 0 & s_3s_1 \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \right\|$$

with respect to x_2 and y_2 . We then set $\sigma = \sqrt{x_2^2 + y_2^2}$ and $v = [x_2 \ y_2]^T / \sigma$.

(3) To improve θ_3 and σ , we fix θ_1 and θ_2 and minimize

$$\left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \sigma \cdot \begin{bmatrix} c_3c_2c_1 \\ c_3c_2s_1 \\ c_3s_2c_1 \\ c_3s_2s_1 \\ s_3c_2c_1 \\ s_3c_2s_1 \\ s_3s_2c_1 \\ s_3s_2s_1 \end{bmatrix} \right\| = \left\| \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} - \begin{bmatrix} c_2c_1 & 0 \\ c_2s_1 & 0 \\ s_2c_1 & 0 \\ s_2s_1 & 0 \\ 0 & c_2s_1 \\ 0 & c_2s_1 \\ 0 & s_2c_1 \\ 0 & s_2s_1 \end{bmatrix} \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \right\|$$

with respect to x_3 and y_3 . We then set $\sigma = \sqrt{x_3^2 + y_3^2}$ and $w = [x_3 \ y_3]^T / \sigma$.

Componentwise optimization is a common framework for many tensor-related computations. The basic idea is to choose a subset of the unknowns and (temporarily) freeze their value. This leads to a simplified optimization problem involving the other unknowns. The process is repeated using different subsets of unknowns each iteration until convergence. The framework is frequently successful, but there is a tendency for the iterates to get trapped near an uninteresting local minima.

5 The Variational Approach to Tensor Singular Values

If μ^2 is a zero of the characteristic polynomial $p(\lambda) = \det(A^T A - \lambda I)$, then μ is a singular value of A and the associated left and right singular vectors are eigenvectors for AA^T and $A^T A$ respectively. How can we extend these notions to tensors? Is there a version of the characteristic polynomial that makes sense for tensors? What would be the analog of the matrices AA^T and $A^T A$? These are tough questions. Fortunately, there is a constructive way to avoid these difficulties and that is to take a *variational approach*. Singular values and vectors are solutions to a very tractable optimization problem.

5.1 Rayleigh Quotient/Power Method Ideas: The Matrix Case

The singular values and singular vectors of a general matrix $A \in \mathbb{R}^{m \times n}$ are the *stationary values and vectors* of the *Rayleigh quotient*

$$\frac{y^T A x}{\|x\|_2 \|y\|_2}.$$

It is slightly more convenient to pose this as a constrained optimization problem: singular values and singular vectors of a general matrix $A \in \mathbb{R}^{m \times n}$ are the stationary values and vectors of

$$\psi_A(x, y) = x^T A y = \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_i y_j$$

subject to the constraints $\|x\|_2 = \|y\|_2 = 1$. To connect this “definition” to the SVD we use the method of Lagrange multipliers and that means looking at the gradient of

$$\tilde{\psi}_A(x, y) = \psi(x, y) - \frac{\lambda}{2}(x^T x - 1) - \frac{\mu}{2}(y^T y - 1).$$

Using the rearrangements

$$\psi_A(x, y) = \sum_{i=1}^m x_i \left(\sum_{j=1}^n a_{ij} y_j \right) = \sum_{j=1}^n y_j \left(\sum_{i=1}^m a_{ij} x_i \right),$$

it follows that

$$\nabla \tilde{\psi}_A(x, y) = \begin{bmatrix} Ay & -\lambda x \\ A^T x & -\mu y \end{bmatrix}. \quad (4)$$

From the vector equation $\nabla \tilde{\psi}_A(x, y) = 0$ we conclude that $\lambda = \mu = x^T A y = \psi_A(x, y)$ and that x and y satisfy $Ay = (x^T A y)x$ and $A^T x = (x^T A y)y$. That is to say, x is an eigenvector of $A^T A$ and y is an eigenvector of AA^T and the associated eigenvalue in each case is $(y^T Ax)^2$. These are exactly the conclusions that can be reached by equating columns in the SVD equation $AV = U\Sigma$. Indeed, from

$$A[v_1 | \cdots | v_n] = [u_1 | \cdots | u_n] \text{diag}(\sigma_1, \dots, \sigma_n)$$

we see that $Av_i = \sigma_i u_i$, $A^T u_i = \sigma_i v_i$, and $\sigma_i = u_i^T Av_i$, for $i = 1 : n$.

The power method for matrices can be designed to go after the largest singular value and associated singular vectors:

*Power Method for Matrix Singular
Values and Vectors*

Given $A \in \mathbb{R}^{m \times n}$ and unit vector $y \in \mathbb{R}^n$.

Repeat:

$$\tilde{x} = Ay, \quad x = \tilde{x}/\|\tilde{x}\|$$

$$\tilde{y} = A^T x, \quad y = \tilde{y}/\|\tilde{y}\|$$

$$\sigma = \psi_A(x, y) = y^T A x$$

$$\sigma_{opt} = \sigma, \quad u_{opt} = y, \quad v_{opt} = x$$

This can be viewed as an alternating procedure for finding a zero for the gradient (4). Under mild assumptions, $\{\sigma_{opt}, u_{opt}, v_{opt}\}$ will approximate the largest singular value of A and the corresponding left and right singular vectors. In principle, deflation can be used to find other singular value triplets. Thus, by applying the power method to $A - \sigma_1 u_1 v_1^T$ we could obtain an estimate of $\{\sigma_2, u_2, v_2\}$.

5.2 Rayleigh Quotient/Power Method Ideas: The Tensor Case

Let us extend the Rayleigh quotient characterization for matrix singular values and vectors to tensors. We work out the order-3 situation for simplicity.

If $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $x \in \mathbb{R}^{n_1}$, $y \in \mathbb{R}^{n_2}$, and $z \in \mathbb{R}^{n_3}$, then the singular values and vectors of \mathcal{A} are the stationary values and vectors of

$$\psi_{\mathcal{A}}(x, y, z) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} a_{ijk} x_i y_j z_k \quad (5)$$

subject to the constraints $\|x\|_2 = \|y\|_2 = \|z\|_2 = 1$. Before we start taking gradients we present three alternative formulations of this summation. Each highlights a different unfolding of the tensor \mathcal{A} .

The Mode-1 Formulation:

$$\psi_{\mathcal{A}}(x, y, z) = \sum_{i=1}^{n_1} x_i \left(\sum_{j=1}^{n_2} \sum_{k=1}^{n_3} a_{ijk} y_j z_k \right) = x^T \mathcal{A}_{(1)} z \otimes y \quad (6)$$

where $\mathcal{A}_{(1)}(i, j + (k - 1)n_2) = a_{ijk}$. For the case $\mathbf{n} = [4, 3, 2]$ we have

$$\mathcal{A}_{(1)} = \begin{bmatrix} a_{111} & a_{121} & a_{131} & a_{112} & a_{122} & a_{132} \\ a_{211} & a_{221} & a_{231} & a_{212} & a_{222} & a_{232} \\ a_{311} & a_{321} & a_{331} & a_{312} & a_{322} & a_{332} \\ a_{411} & a_{421} & a_{431} & a_{412} & a_{422} & a_{432} \end{bmatrix}_{(1,1) \quad (2,1) \quad (3,1) \quad (1,2) \quad (2,2) \quad (3,2)}.$$

The Mode-2 Formulation:

$$\psi_{\mathcal{A}}(x, y, z) = \sum_{j=1}^{n_2} y_j \left(\sum_{i=1}^{n_1} \sum_{k=1}^{n_3} a_{ijk} x_i z_k \right) = y^T \mathcal{A}_{(2)} z \otimes x \quad (7)$$

where $\mathcal{A}_{(2)}(j, i + (k - 1)n_1) = a_{ijk}$. For the case $\mathbf{n} = [4, 3, 2]$ we have

$$\mathcal{A}_{(2)} = \begin{bmatrix} a_{111} & a_{211} & a_{311} & a_{411} & a_{112} & a_{212} & a_{312} & a_{412} \\ a_{121} & a_{221} & a_{321} & a_{421} & a_{122} & a_{222} & a_{322} & a_{422} \\ a_{131} & a_{231} & a_{331} & a_{431} & a_{132} & a_{232} & a_{332} & a_{432} \end{bmatrix}_{(1,1) \quad (2,1) \quad (3,1) \quad (4,1) \quad (1,2) \quad (2,2) \quad (3,2) \quad (4,2)}.$$

The Mode-3 Formulation:

$$\psi_{\mathcal{A}}(x, y, z) = \sum_{k=1}^p z_k \left(\sum_{i=1}^m \sum_{j=1}^n a_{ijk} x_i y_j \right) = z^T \mathcal{A}_{(3)} y \otimes x \quad (8)$$

where $\mathcal{A}_{(3)}(k, i + (j - 1)n_1) = a_{ijk}$. For the case $\mathbf{n} = [4, 3, 2]$ we have

$$\mathcal{A}_{(3)} = \begin{bmatrix} a_{111} & a_{211} & a_{311} & a_{411} & a_{121} & a_{221} & a_{321} & a_{421} & a_{131} & a_{231} & a_{331} & a_{431} \\ a_{112} & a_{212} & a_{312} & a_{412} & a_{122} & a_{222} & a_{322} & a_{422} & a_{132} & a_{232} & a_{332} & a_{432} \end{bmatrix}_{(1,1) \quad (2,1) \quad (3,1) \quad (4,1) \quad (1,2) \quad (2,2) \quad (3,2) \quad (4,2) \quad (1,3) \quad (2,3) \quad (3,3) \quad (4,3)}.$$

The matrices $\mathcal{A}_{(1)}$, $\mathcal{A}_{(2)}$, and $\mathcal{A}_{(3)}$ are the mode-1, mode-2, and mode-3 unfoldings of \mathcal{A} that we introduced in Sect. 3.4. It is handy to identify columns with multi-indices as we have shown.

We return to the constrained minimization of the objective function $\psi_{\mathcal{A}}(x, y, z)$) that is defined in (5). Using the method of Lagrange multipliers we set the gradient of

$$\tilde{\psi}_{\mathcal{A}}(x, y, z) = \psi_{\mathcal{A}}(x, y, z) - \frac{\lambda}{2}(x^T x - 1) - \frac{\mu}{2}(y^T y - 1) - \frac{\tau}{2}(z^T z - 1)$$

to zero. Using (6)–(8) we get

$$\nabla \tilde{\psi}_{\mathcal{A}} = \begin{bmatrix} \mathcal{A}_{(1)}(z \otimes y) - \lambda x \\ \mathcal{A}_{(2)}(z \otimes x) - \mu y \\ \mathcal{A}_{(3)}(y \otimes x) - \tau z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (9)$$

Since x , y , and z are unit vectors it follows that $\lambda = \mu = \tau = \psi(x, y, z)$. In this case we say that $\sigma = \psi(x, y, z)$ is a singular value of \mathcal{A} and x , y , and z are the associated singular vectors. How might we solve this (highly structured) system of nonlinear equations? The triplet of matrix-vector products:

$$A_{(1)} \cdot (z \otimes y) = \sigma \cdot x \quad A_{(2)} \cdot (z \otimes x) = \sigma \cdot y \quad A_{(3)} \cdot (y \otimes x) = \sigma \cdot z$$

suggests a componentwise solution strategy:

*Power Method for Tensor Singular
Values and Vectors*

Given: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and unit vectors $y \in \mathbb{R}^{n_2}$ and $z \in \mathbb{R}^{n_3}$.

Repeat:

$$\tilde{x} = \mathcal{A}_{(1)}(z \otimes y), \quad \sigma = \|\tilde{x}\|, \quad x = \tilde{x}/\sigma$$

$$\tilde{y} = \mathcal{A}_{(2)}(z \otimes x), \quad \sigma = \|\tilde{y}\|, \quad y = \tilde{y}/\sigma$$

$$\tilde{z} = \mathcal{A}_{(3)}(y \otimes x), \quad \sigma = \|\tilde{z}\|, \quad z = \tilde{z}/\sigma$$

$$\sigma_{opt} = \sigma, \quad x_{opt} = x, \quad y_{opt} = y, \quad z_{opt} = z$$

See [7, 17] for details.

For matrices, the SVD expansion

$$A = U \Sigma V^T = \sum_{i=1}^{\text{rank}(A)} \sigma_i u_i v_i^T$$

has an important optimality property. In particular, the Eckhart-Young theorem tells us that

$$A_r = \sum_{i=1}^r \sigma_i u_i v_i^T \quad r \leq \text{rank}(A)$$

is the closest rank- r matrix to A in either the 2-norm or Frobenius norm. Moreover, the closest rank-1 matrix to A_{r-1} is $\sigma_r u_r v_r^T$. Thus, it would be possible (in principle) to compute the full SVD by solving a sequence of closest rank-1 matrix problems.

This idea does *not* work for tensors. In other words, if $\sigma_1 u_1 \circ v_1 \circ w_1$ is the closest rank-1 tensor to \mathcal{A} and $\sigma_2 u_2 \circ v_2 \circ w_2$ is the closest rank-1 tensor to

$$\tilde{\mathcal{A}} = \mathcal{A} - \sigma_1 u_1 \circ v_1 \circ w_1,$$

then

$$\mathcal{A}_2 = \sigma_1 u_1 \circ v_1 \circ w_1 + \sigma_2 u_2 \circ v_2 \circ w_2$$

is *not* necessarily the closest rank-2 tensor to \mathcal{A} . We need an alternative approach to formulating a “tensor SVD”.

5.3 A First Look at Tensor Rank

Since matrix rank ideas do not readily extend to the tensor setting, we should look more carefully at tensor rank to appreciate the “degree of difficulty” associated with the formulation of illuminating low-rank tensor expansions.

We start with a definition. Suppose the tensor \mathcal{A} can be written as the sum of r rank-1 tensors and that r is minimal in this regard. In this case we say that $\text{rank}(\mathcal{A}) = r$. Let us explore this concept in the simplest possible setting: $\mathcal{A} \in \mathbb{R}^{2 \times 2 \times 2}$. For this problem the goal is to find three thin-as-possible matrices $X, Y, Z \in \mathbb{R}^{2 \times r}$ so that

$$\mathcal{A} = \sum_{k=1}^r X(:, k) \circ Y(:, k) \circ Z(:, k), \quad (10)$$

i.e.,

$$\text{vec}(\mathcal{A}) = \begin{bmatrix} a_{111} \\ a_{211} \\ a_{121} \\ a_{221} \\ a_{112} \\ a_{212} \\ a_{122} \\ a_{222} \end{bmatrix} = \sum_{k=1}^r Z(:, k) \otimes Y(:, k) \otimes X(:, k).$$

This vector equation can also be written as a pair of matrix equations:

$$\begin{aligned}\mathcal{A}(:, 1) &= \begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix} = \sum_{k=1}^r Z(1, k) X(:, k) Y(:, k)^T \\ \mathcal{A}(:, 2) &= \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix} = \sum_{k=1}^r Z(2, k) X(:, k) Y(:, k)^T.\end{aligned}$$

Readers familiar with the generalized eigenvalue problem should see a connection to our 2-by-2-by-2 rank(\mathcal{A}) problem. Indeed, it can be shown that

$$\det \left(\begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix} - \lambda \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix} \right) = 0$$

has real distinct roots with probability 0.79 and complex conjugate roots with probability 0.21 when the matrix entries are randomly selected using the MATLAB `randn` function. If this 2-by-2 generalized eigenvalue problem has real distinct eigenvalues, then it is possible to find nonsingular matrices S and T so that

$$\begin{aligned}\begin{bmatrix} a_{111} & a_{121} \\ a_{211} & a_{221} \end{bmatrix} &= S \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix} T^T \\ \begin{bmatrix} a_{112} & a_{122} \\ a_{212} & a_{222} \end{bmatrix} &= S \begin{bmatrix} \beta_1 & 0 \\ 0 & \beta_2 \end{bmatrix} T^T.\end{aligned}$$

This shows that the rank-1 expansion (10) for \mathcal{A} holds with $r = 2$, $X = S$, $Y = T$ and

$$Z = \begin{bmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \end{bmatrix}.$$

Thus, for 2-by-2-by-2 tensors, rank equals two with probability about 0.79. A similar generalized eigenvalue analysis shows that the rank is three with probability 0.21. This is a very different situation than with matrices where an n -by- n matrix has rank n with probability 1. The subtleties associated with tensor rank are discussed further in Sect. 8.3.

6 Tensor Symmetry

Symmetry for a matrix is defined through transposition: $A = A^T$. This is a nice shorthand way of saying that $A(i, j) = A(j, i)$ for all possible i and j that satisfy $1 \leq i \leq n$ and $1 \leq j \leq n$.

How do we extend this notion to tensors? Transposition moves indices around so we need a way of talking about what happens when we (say) interchange $\mathcal{A}(i, j, k)$ with $\mathcal{A}(j, i, k)$ or $\mathcal{A}(k, j, i)$ or $\mathcal{A}(i, k, j)$ or $\mathcal{A}(j, k, i)$ or $\mathcal{A}(k, i, j)$. It looks like we will have to contend with an exponential number of transpositions and an exponential number of partial symmetries.

6.1 Tensor Transposition

If $\mathcal{C} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then there are $3! = 6$ possible transpositions identified by the notation $\mathcal{C}^{<[i j k]>}$ where $[i j k]$ is a permutation of $[1 2 3]$:

$$\mathcal{B} = \left\{ \begin{array}{l} \mathcal{C}^{<[1 2 3]>} \\ \mathcal{C}^{<[1 3 2]>} \\ \mathcal{C}^{<[2 1 3]>} \\ \mathcal{C}^{<[2 3 1]>} \\ \mathcal{C}^{<[3 1 2]>} \\ \mathcal{C}^{<[3 2 1]>} \end{array} \right\} \implies \left\{ \begin{array}{l} b_{ijk} \\ b_{ikj} \\ b_{jik} \\ b_{jki} \\ b_{kij} \\ b_{kji} \end{array} \right\} = c_{ijk}$$

for $i = 1 : n_1$, $j = 1 : n_2$, $k = 1 : n_3$.

For order- d tensors there are $d!$ possibilities. Suppose $\mathbf{v} = [v_1, v_2, \dots, v_d]$ is a permutation of the integer vector $1 : d = [1, 2, \dots, d]$. If $\mathcal{C} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, then

$$\mathcal{B} = \mathcal{C}^{<\mathbf{v}>} \quad \Rightarrow \quad \mathcal{B}(\mathbf{i}(\mathbf{v})) = \mathcal{C}(\mathbf{i}) \quad \mathbf{1} \leq \mathbf{i} \leq \mathbf{n}.$$

6.2 Symmetric Tensors

An order- d tensor $\mathcal{C} \in \mathbb{R}^{n \times \dots \times n}$ is *symmetric* if $\mathcal{C}^{<\mathbf{v}>} = \mathcal{C}$ for all permutations \mathbf{v} of $1 : d$. If $d = 3$ this means that

$$c_{ijk} = c_{ikj} = c_{jik} = c_{jki} = c_{kij} = c_{kji}.$$

To get a feel for the level of redundancy in a symmetric tensor, we see that a symmetric $\mathcal{C} \in \mathbb{R}^{3 \times 3 \times 3}$ has at most ten distinct values:

$$\begin{aligned} & c_{111} \\ & c_{112} = c_{121} = c_{211} \\ & c_{113} = c_{131} = c_{311} \\ & c_{222} \\ & c_{221} = c_{212} = c_{122} \\ & c_{223} = c_{232} = c_{322} \\ & c_{333} \\ & c_{331} = c_{313} = c_{133} \\ & c_{332} = c_{323} = c_{233} \\ & c_{123} = c_{132} = c_{213} = c_{231} = c_{312} = c_{321}. \end{aligned}$$

The modal unfoldings of a symmetric tensor are all the same. For example, the (2,3) entry in each of the matrices

$$\begin{aligned} C_{(1)} &= \begin{bmatrix} c_{111} & c_{121} & c_{131} & c_{112} & c_{122} & c_{132} & c_{113} & c_{123} & c_{133} \\ c_{211} & c_{221} & c_{231} & c_{212} & c_{222} & c_{232} & c_{213} & c_{223} & c_{233} \\ c_{311} & c_{321} & c_{331} & c_{312} & c_{322} & c_{332} & c_{113} & c_{323} & c_{333} \end{bmatrix} \\ C_{(2)} &= \begin{bmatrix} c_{111} & c_{211} & c_{311} & c_{112} & c_{212} & c_{312} & c_{113} & c_{213} & c_{313} \\ c_{121} & c_{221} & c_{321} & c_{122} & c_{222} & c_{322} & c_{123} & c_{223} & c_{323} \\ c_{131} & c_{231} & c_{331} & c_{132} & c_{232} & c_{332} & c_{113} & c_{233} & c_{333} \end{bmatrix} \\ C_{(3)} &= \begin{bmatrix} c_{111} & c_{211} & c_{311} & c_{121} & c_{221} & c_{321} & c_{131} & c_{231} & c_{331} \\ c_{112} & c_{212} & c_{312} & c_{122} & c_{222} & c_{322} & c_{132} & c_{232} & c_{332} \\ c_{113} & c_{213} & c_{313} & c_{123} & c_{223} & c_{323} & c_{133} & c_{233} & c_{333} \end{bmatrix} \end{aligned}$$

are equal: $c_{231} = c_{321} = c_{312}$.

6.3 Symmetric Rank

An order- d symmetric rank-1 tensor $\mathcal{C} \in \mathbb{R}^{n \times \dots \times n}$ has the form

$$\mathcal{C} = \underbrace{x \circ \dots \circ x}_{d \text{ times}}$$

where $x \in \mathbb{R}^n$. In this case we clearly have

$$\mathcal{C}(i_1, \dots, i_d) = x_{i_1} x_{i_2} \cdots x_{i_d}$$

and

$$\text{vec}(\mathcal{C}) = \underbrace{x \otimes \cdots \otimes x}_{d \text{ times}}.$$

An order-3 symmetric tensor \mathcal{C} has *symmetric rank* r if there exists $x_1, \dots, x_r \in \mathbb{R}^n$ and $\sigma \in \mathbb{R}^r$ such that

$$\mathcal{C} = \sum_{k=1}^r \sigma_k \cdot x_k \circ x_k \circ x_k$$

and no shorter sum of symmetric rank-1 tensors exists. Symmetric rank is denoted by $\text{rank}_S(\mathcal{C})$. Note, in contrast to what we would expect for matrices, there may be a shorter sum of general rank-1 tensors that add up to \mathcal{C} :

$$\mathcal{C} = \sum_{k=1}^{\tilde{r}} \tilde{\sigma}_k \cdot \tilde{x}_k \circ \tilde{y}_k \circ \tilde{z}_k.$$

The symmetric rank of a symmetric tensor is more tractable than the (general) rank of a general tensor. For example, if $\mathcal{C} \in \mathbb{C}^{n \times \cdots \times n}$ is an order- d symmetric tensor, then with probability one we have

$$\text{rank}_S(\mathcal{C}) = \begin{cases} f(d, m) + 1 & \text{if } (d, n) = (3, 5), (4, 3), (4, 4), \text{ or } (4, 5) \\ f(d, n) & \text{otherwise} \end{cases}$$

where

$$f(d, n) = \text{ceil} \left(\frac{\binom{n+d-1}{d}}{n} \right).$$

See [4, 5] for deeper discussions of symmetric rank.

6.4 The Eigenvalues of a Symmetric Tensor

For a symmetric matrix C the stationary values of $\phi_C(x) = x^T C x$ subject to the constraint that $\|x\|_2 = 1$ are the eigenvalues of C . The associated stationary vectors are eigenvectors. We extend this idea to symmetric tensors and the order-3 case is good enough to illustrate the main ideas.

If $\mathcal{C} \in \mathbb{R}^{n \times n \times n}$ is a symmetric tensor, then we define the stationary values of

$$\phi_{\mathcal{C}}(x) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n c_{ijk} x_i x_j x_k = x^T C_{(1)}(x \otimes x)$$

subject to the constraint that $\|x\|_2 = 1$ to be the eigenvalues of \mathcal{C} . The associated stationary vectors are eigenvectors. Using the method of Lagrange multipliers it can be shown that if x is a stationary vector for $\phi_{\mathcal{C}}$ then

$$x = \phi_{\mathcal{C}}(x) C_{(1)}(x \otimes x)$$

This leads to an iteration of the following form:

*Power Method for Tensor Eigenvalues
and Eigenvectors*

Given: Symmetric $\mathcal{C} \in \mathbb{R}^{n \times n \times n}$ and unit vector $x \in \mathbb{R}^n$.

Repeat:

$$\begin{aligned}\tilde{x} &= \mathcal{C}_{(1)}(x \otimes x) \\ \lambda &= \|\tilde{x}\|_2 \\ x &= \tilde{x}/\lambda \\ \lambda_{opt} &= \lambda, x_{opt} = x.\end{aligned}$$

There are some convergence results for this iteration. For example, it can be shown that if the order of \mathcal{C} is even and M is a square unfolding, then the iteration converges if M is positive definite [14].

6.5 Symmetric Embeddings

In the matrix case there are connections between the singular values and vectors of $A \in \mathbb{R}^{n_1 \times n_2}$ and the eigenvalues and vectors of

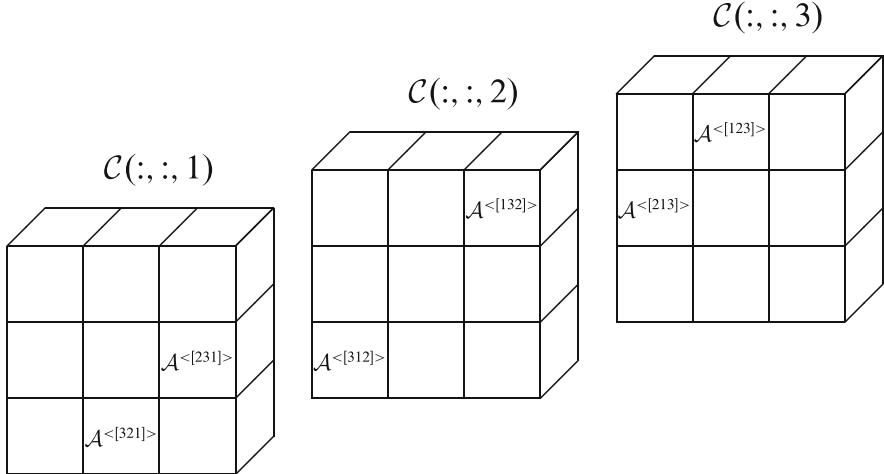
$$\text{sym}(A) = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \in \mathbb{R}^{(n_1+n_2) \times (n_1+n_2)}$$

If $A = U \cdot \text{diag}(\sigma_i) \cdot V^T$ is the SVD of $A \in \mathbb{R}^{n_1 \times n_2}$, then for $k = 1 : \text{rank}(A)$

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} u_k \\ \pm v_k \end{bmatrix} = \pm \sigma_k \begin{bmatrix} u_k \\ \pm v_k \end{bmatrix}$$

where $u_k = U(:, k)$ and $v_k = V(:, k)$.

It turns out that symmetric tensor $\text{sym}(\mathcal{A})$ can be “built” from of a general tensor \mathcal{A} by judiciously positioning \mathcal{A} and all of its transposes. Here is a depiction of the order-3 case:



We can think of $\text{sym}(\mathcal{A})$ as a 3-by-3-by-3 block tensor. As a tensor of scalars, it is N -by- N -by- N where $N = n_1 n_2 n_3$. If

$$\left\{ \sigma, \begin{bmatrix} u \\ v \\ z \end{bmatrix} \right\}$$

is a stationary pair for $\text{sym}(\mathcal{A})$, then so are

$$\left\{ \sigma, \begin{bmatrix} u \\ -v \\ -z \end{bmatrix} \right\}, \quad \left\{ -\sigma, \begin{bmatrix} u \\ -v \\ z \end{bmatrix} \right\}, \quad \left\{ -\sigma, \begin{bmatrix} u \\ v \\ -z \end{bmatrix} \right\}.$$

This is a nice generalization of the result for matrices. There are interesting connections between power methods with \mathcal{A} and power methods with $\text{sym}(\mathcal{A})$. There are also connections between the rank of \mathcal{A} and the symmetric rank of

$\text{sym}(\mathcal{A})$:

$$d! \cdot \text{rank}(\mathcal{A}) \leq \text{rank}_S(\text{sym}(\mathcal{A})).$$

It is not clear if the inequality can be replaced by equality. See [26].

7 The Tucker Decomposition

We have seen that it is possible to extend the definition of singular values and vectors to tensors by using variational principles. However, these insights did not culminate in the production of a tensor SVD and that is disappointing when we consider the power of the matrix SVD:

1. It can turn a given problem into an equivalent easy-to-solve problem. For example, the $\min \|Ax - b\|_2$ problem can be converted into an equivalent diagonal least squares problem using the SVD.
2. It can uncover hidden relationships that exist in matrix-encoded data. For example, the SVD can show that a data matrix has a low rank structure.

In the next several sections we produce various SVD-like tensor decompositions. We start with the Tucker decomposition because it involves orthogonality and has a strong resemblance to the matrix SVD. We use order-3 tensors to illustrate the main ideas.

7.1 Tucker Representations: The Matrix Case

Given a matrix $A \in \mathbb{R}^{n_1 \times n_2}$, the Tucker representation problem involves finding a *core matrix* $S \in \mathbb{R}^{r_1 \times r_2}$ and matrices $U_1 \in \mathbb{R}^{n_1 \times r_1}$ and $U_2 \in \mathbb{R}^{n_2 \times r_2}$ such that

$$A(i_1, i_2) = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} S(j_1, j_2) \cdot U_1(i_1, j_1) \cdot U_2(i_2, j_2).$$

Part of the “deal” involves choosing the integers r_1 and r_2 and perhaps replacing the “=” with “ \approx ”. Regardless, it is possible to reformulate the right hand side as a sum of rank-1 matrices,

$$A = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} S(j_1, j_2) \cdot U_1(:, j_1) \cdot U_2(:, j_2)^T, \quad (11)$$

or as a sum of vector Kronecker products,

$$\text{vec}(A) = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} S(j_1, j_2) U_2(:, j_2) \otimes U_1(:, j_2),$$

or as a single matrix-vector product,

$$\text{vec}(A) = (U_2 \otimes U_1) \cdot \text{vec}(S).$$

The tensor versions of these reformulations get us to think the right way about how we might generalize the matrix SVD.

7.2 Tucker Representations: The Tensor Case

Given a tensor $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the Tucker representation problem involves finding a *core tensor* $S \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ and matrices $U_1 \in \mathbb{R}^{n_1 \times r_1}$, $U_2 \in \mathbb{R}^{n_2 \times r_2}$, and $U_3 \in \mathbb{R}^{n_3 \times r_3}$ such that

$$\mathcal{A}(i_1, i_2, i_3) = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \sum_{j_3=1}^{r_3} S(j_1, j_2, j_3) \cdot U_1(i_1, j_1) \cdot U_2(i_2, j_2) \cdot U_3(i_3, j_3).$$

As in the matrix case above, we can rewrite this as a sum of rank-1 tensors,

$$\mathcal{A} = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \sum_{j_3=1}^{r_3} S(j_1, j_2, j_3) \cdot U_1(:, j_1) \circ U_2(:, j_2) \circ U_3(:, j_3),$$

or as the sum of vector Kronecker products,

$$\text{vec}(\mathcal{A}) = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \sum_{j_3=1}^{r_3} S(j_1, j_2, j_3) \cdot U_3(:, j_3) \otimes U_2(:, j_2) \otimes U_1(:, j_1)$$

or as a single matrix-vector product,

$$\text{vec}(\mathcal{A}) = (U_3 \otimes U_2 \otimes U_1) \cdot \text{vec}(S).$$

The challenge is to design the representation so that it is illuminating and computable.

Before we proceed it is instructive to revisit the matrix case. If we set $r_1 = n_1$ and $r_2 = n_2$ in (11) and assume the U matrices are orthogonal, then the Tucker

representation has the form

$$A = U_1(U_1^T A U_2)U_2^T = \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} S(j_1, j_2) U_1(:, j_1) U_2(:, j_2)^T$$

where $S = U_1^T A U_2$. To make this an “illuminating” representation of A we strive for a diagonal core matrix S and that, of course, leads to the SVD:

$$A = \sum_{k=1}^{\text{rank}(A)} S(k, k) U_1(:, k) U_2(:, k)^T.$$

From there we prove various optimality theorems and conclude that

$$A_r = \sum_{k=1}^r S(k, k) U_1(:, k) U_2(:, k)^T \quad r \leq \text{rank}(A)$$

is the closest rank- r matrix to A in (say) the Frobenius norm.

On the algorithmic front methods typically determine U_1 and U_2 through a sequence of updates. Thus, if $A = U_1 S U_2^T$ is the “current” representation of A we proceed to compute orthogonal Δ_1 and Δ_2 so that $\tilde{S} = \Delta_1^T S \Delta_2$ is “more diagonal” than S . We then update the representation:

$$S \leftarrow \Delta_1^T S \Delta_2, \quad U_1 \leftarrow U_1 \Delta_1, \quad U_2 \leftarrow U_2 \Delta_2.$$

Our plan is to mimic this sequence of events for tensors focusing first on the connection between the core tensor \mathcal{S} and the U matrices.

7.3 The Mode- k Product

Updating a Tucker representation involves updating the current core tensor \mathcal{S} and the associated U -matrices. Regarding the former we anticipate the need to design a relevant tensor-matrix product. The *mode- k product* turns out to be that operation and we motivate the main idea with an example. Suppose $\mathcal{S} \in \mathbb{R}^{4 \times 3 \times 2}$ and consider its mode-2 unfolding:

$$\mathcal{S}_{(2)} = \begin{bmatrix} s_{111} & s_{211} & s_{311} & s_{411} & s_{112} & s_{212} & s_{312} & s_{412} \\ s_{121} & s_{221} & s_{321} & s_{421} & s_{122} & s_{222} & s_{322} & s_{422} \\ s_{131} & s_{231} & s_{331} & s_{431} & s_{132} & s_{232} & s_{332} & s_{432} \end{bmatrix}.$$

Its columns are the mode-2 fibers of \mathcal{S} . Suppose we apply a 5-by-3 matrix M to each of those fibers:

$$\begin{aligned} & \left[\begin{array}{ccccccccc} t_{111} & t_{211} & t_{311} & t_{411} & t_{112} & t_{212} & t_{312} & t_{412} \\ t_{121} & t_{221} & t_{321} & t_{421} & t_{122} & t_{222} & t_{322} & t_{422} \\ t_{131} & t_{231} & t_{331} & t_{431} & t_{132} & t_{232} & t_{332} & t_{432} \\ t_{141} & t_{241} & t_{341} & t_{441} & t_{142} & t_{242} & t_{342} & t_{442} \\ t_{151} & t_{251} & t_{351} & t_{451} & t_{152} & t_{252} & t_{352} & t_{452} \end{array} \right] \\ & = \left[\begin{array}{ccc} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \\ m_{41} & m_{42} & m_{43} \\ m_{51} & m_{52} & m_{53} \end{array} \right] \left[\begin{array}{ccccccccc} s_{111} & s_{211} & s_{311} & s_{411} & s_{112} & s_{212} & s_{312} & s_{412} \\ s_{121} & s_{221} & s_{321} & s_{421} & s_{122} & s_{222} & s_{322} & s_{422} \\ s_{131} & s_{231} & s_{331} & s_{431} & s_{132} & s_{232} & s_{332} & s_{432} \end{array} \right]. \end{aligned}$$

This defines a new tensor $\mathcal{T} \in \mathbb{R}^{4 \times 5 \times 2}$ that is totally specified by the equation

$$\mathcal{T}_{(2)} = M \cdot \mathcal{S}_{(2)}$$

and referred to as the mode-2 product of a tensor \mathcal{S} with a matrix M . In general, if \mathcal{S} is an $n_1 \times \cdots \times n_d$ tensor and $M \in \mathbb{R}^{m_k \times n_k}$ for some k that satisfies $1 \leq k \leq d$, then the mode- k product of \mathcal{S} with M is a new tensor \mathcal{T} defined by

$$\mathcal{T}_{(k)} = M \cdot \mathcal{S}_{(k)}.$$

To indicate this operation we use the notation

$$\mathcal{T} = \mathcal{S} \times_k M.$$

Note that \mathcal{T} is an $n_1 \times \cdots \times n_{k-1} \times m_k \times n_{k+1} \times \cdots \times n_d$ tensor. To illustrate more characterizations of the mode- k product we drop down to the order-3 case.

- *Mode-1 Product.* If $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $M_1 \in \mathbb{R}^{m_1 \times n_1}$, then $\mathcal{T} = \mathcal{S} \times_1 M_1$ is an $m_1 \times n_2 \times n_3$ tensor that is equivalently defined by

$$\mathcal{T}(i_1, i_2, i_3) = \sum_{k=1}^{n_1} M_1(i_1, k) \mathcal{S}(k, i_2, i_3)$$

$$\text{vec}(\mathcal{T}) = (I_{n_3} \otimes I_{n_2} \otimes M_1) \text{vec}(\mathcal{S}). \quad (12)$$

- *Mode-2 Product.* If $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $M_2 \in \mathbb{R}^{m_2 \times n_2}$, then $\mathcal{T} = \mathcal{S} \times_2 M_2$ is an $n_1 \times m_2 \times n_3$ tensor that is equivalently defined by

$$\mathcal{T}(i_1, i_2, i_3) = \sum_{k=1}^{n_2} M_2(i_2, k) \mathcal{S}(i_1, k, i_3)$$

$$\text{vec}(\mathcal{T}) = (I_{n_3} \otimes M_2 \otimes I_{n_1}) \text{vec}(\mathcal{S}) \quad (13)$$

- *Mode-3 Product.* If $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $M_3 \in \mathbb{R}^{m_3 \times n_3}$, then $\mathcal{T} = \mathcal{A} \times_3 M_3$ is an $n_1 \times n_2 \times m_3$ tensor that is equivalently defined by

$$\mathcal{T}(i_1, i_2, i_3) = \sum_{k=1}^{n_3} M_3(i_3, k) \mathcal{S}(i_1, i_2, k)$$

$$\text{vec}(\mathcal{T}) = (M_3 \otimes I_{n_2} \otimes I_{n_1}) \text{vec}(\mathcal{S}) \quad (14)$$

The modal products have two important properties that we will be using later. The first concerns successive products in the *same* mode. If $\mathcal{S} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and $M_1, M_2 \in \mathbb{R}^{n_k \times n_k}$, then

$$(\mathcal{S} \times_k M_1) \times_k M_2 = \mathcal{S} \times_k (M_1 M_2).$$

The second property concerns successive products in *different* modes. If $\mathcal{S} \in \mathbb{R}^{n_1 \times \dots \times n_d}$, $M_k \in \mathbb{R}^{n_k \times n_k}$, $M_j \in \mathbb{R}^{n_j \times n_j}$, and $k \neq j$, then

$$(\mathcal{S} \times_k M_k) \times_j M_j = (\mathcal{S} \times_j M_j) \times_k M_k$$

The order is not important so we just write $\mathcal{S} \times_j M_j \times_k M_k$ or $\mathcal{S} \times_k M_k \times_j M_j$.

7.4 The Core Tensor

Suppose we have a Tucker representation

$$\mathcal{A}(\mathbf{i}) = \sum_{\mathbf{j}=1}^{\mathbf{n}} \mathcal{S}(\mathbf{j}) \cdot U_1(i_1, j_1) \cdot U_2(i_2, j_2) \cdot U_3(i_3, j_3),$$

where $\mathcal{A} \in \mathbb{R}^{\mathbf{n}}$, $\mathbf{n} = [n_1, n_2, n_3]$, and $U_1 \in \mathbb{R}^{n_1 \times n_1}$, $U_2 \in \mathbb{R}^{n_2 \times n_2}$, and $U_3 \in \mathbb{R}^{n_3 \times n_3}$. It follows that

$$\text{vec}(\mathcal{A}) = (U_3 \otimes U_2 \otimes U_1) \text{vec}(\mathcal{S})$$

$$= (U_3 \otimes I_{n_2} \otimes I_{n_1})(I_{n_3} \otimes U_2 \otimes I_{n_1})(I_{n_3} \otimes I_{n_2} \otimes U_1) \text{vec}(\mathcal{S}).$$

This factored product enables us to relate the core tensor \mathcal{S} to \mathcal{A} via a triplet of modal products. Indeed, if

$$\text{vec}(\mathcal{S}^{(1)}) = (I_{n_3} \otimes I_{n_2} \otimes U_1) \text{vec}(\mathcal{S})$$

$$\text{vec}(\mathcal{S}^{(2)}) = (I_{n_3} \otimes U_2 \otimes I_{n_1}) \text{vec}(\mathcal{S}^{(1)})$$

$$\text{vec}(\mathcal{S}^{(3)}) = (U_3 \otimes I_{n_2} \otimes I_{n_1}) \text{vec}(\mathcal{S}^{(2)})$$

then $\mathcal{A} = \mathcal{S}^{(3)}$. But from (12), (13), and (14) this means that

$$\mathcal{S}^{(1)} = \mathcal{S} \times_1 U_1 \quad \mathcal{S}^{(2)} = \mathcal{S}^{(1)} \times_2 U_2 \quad \mathcal{S}^{(3)} = \mathcal{S}^{(2)} \times_3 U_3$$

and so

$$\mathcal{A} = \mathcal{S} \times_1 U_1 \times_2 U_2 \times_3 U_3.$$

If the U 's are nonsingular then

$$\begin{aligned} \mathcal{A} &= \mathcal{A} \times_1 (U_1^{-1} U_1) \times_2 (U_2^{-1} U_2) \times_3 (U_3^{-1} U_3) \\ &= (\mathcal{A} \times_1 U_1^{-1} \times_2 U_2^{-1} \times_3 U_3^{-1}) \times_1 U_1 \times_2 U_2 \times_3 U_3 \end{aligned}$$

and so $\mathcal{S} = \mathcal{A} \times_1 U_1^{-1} \times_2 U_2^{-1} \times_3 U_3^{-1}$.

If the U 's are orthogonal and $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $U_1 \in \mathbb{R}^{n_1 \times n_1}$, $U_2 \in \mathbb{R}^{n_2 \times n_2}$, and $U_3 \in \mathbb{R}^{n_3 \times n_3}$ are orthogonal, then

$$\mathcal{A} = \mathcal{S} \times_1 U_1 \times_2 U_2 \times_3 U_3$$

where

$$\mathcal{S} = \mathcal{A} \times_1 U_1^T \times_2 U_2^T \times_3 U_3^T. \quad (15)$$

or equivalently

$$\mathcal{A}_{(1)} = U_1 \mathcal{S}_{(1)} (U_3 \otimes U_2)^T \quad (16)$$

$$\mathcal{A}_{(2)} = U_2 \mathcal{S}_{(2)} (U_3 \otimes U_1)^T \quad (17)$$

$$\mathcal{A}_{(3)} = U_3 \mathcal{S}_{(3)} (U_2 \otimes U_1)^T \quad (18)$$

With this choice we are representing \mathcal{A} as a Tucker product of a core tensor \mathcal{S} and three orthogonal matrices. Things are beginning to look “SVD-like”.

7.5 The Higher-Order SVD

How do we choose the U matrices in (15) so that the core tensor \mathcal{S} reveals things about the structure of \mathcal{A} ? It is instructive to look at the matrix case where we know the answer to this question. Suppose we have the SVDs

$$U_1^T \mathcal{A}_{(1)} V_1 = \Sigma_1, \quad U_2^T \mathcal{A}_{(2)} V_2 = \Sigma_2.$$

Since $\mathcal{A}_1 = A$ and $\mathcal{A}_{(2)} = A^T$ it follows that we may set $V_1 = U_2$. In other words, we can (in principle) compute the SVD of A by computing the SVD of the modal unfoldings $\mathcal{A}_{(1)}$ and $\mathcal{A}_{(2)}$. The U matrices from the modal SVDs are the “right” choice.

This suggests a strategy for picking good U matrices for the tensor case $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Namely, compute the SVD of the modal unfoldings

$$\mathcal{A}_{(1)} = U_1 \Sigma_1 V_1^T \quad \mathcal{A}_{(2)} = U_2 \Sigma_2 V_2^T \quad \mathcal{A}_{(3)} = U_3 \Sigma_3 V_3^T \quad (19)$$

and set

$$\mathcal{S} = \mathcal{A} \times_1 U_1^T \times_2 U_2^T \times_3 U_3^T.$$

The resulting decomposition

$$\mathcal{A} = \mathcal{S} \times_1 U_1 \times_2 U_2 \times_3 U_3,$$

is the *higher-order SVD* (HOSVD) of \mathcal{A} . If $\mathcal{A} = \mathcal{S} \times_1 U_1 \times_2 U_2 \times_3 U_3$ is the HOSVD of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then

$$\mathcal{A} = \sum_{j=1}^r \mathcal{S}(j) \cdot U_1(:, j_1) \circ U_2(:, j_2) \circ U_3(:, j_3) \quad (20)$$

where $r_1 = \text{rank}(\mathcal{A}_{(1)})$, $r_2 = \text{rank}(\mathcal{A}_{(2)})$, and $r_3 = \text{rank}(\mathcal{A}_{(3)})$. The triplet of modal ranks $[r_1, r_2, r_3]$ is called the *multilinear rank* of \mathcal{A} .

The core tensor in the HOSVD has important properties. By combining (16)–(19) we have

$$\mathcal{S}_{(1)} = \Sigma_1 V_1 (U_3 \otimes U_2)$$

$$\mathcal{S}_{(2)} = \Sigma_2 V_2 (U_3 \otimes U_1)$$

$$\mathcal{S}_{(3)} = \Sigma_3 V_3 (U_2 \otimes U_1)$$

from which we conclude that

$$\| \mathcal{S}(j, :, :) \|_F = \sigma_j(\mathcal{A}_{(1)}) \quad j = 1 : n_1$$

$$\| \mathcal{S}(:, j, :) \|_F = \sigma_j(\mathcal{A}_{(2)}) \quad j = 1 : n_2$$

$$\| \mathcal{S}(:, :, j) \|_F = \sigma_j(\mathcal{A}_{(3)}) \quad j = 1 : n_3.$$

Here, $\sigma_j(C)$ denotes the j th largest singular value of the matrix C . Notice that the norms of the tensor's slices are getting smaller as we "move away" from $\mathcal{A}(1, 1, 1)$.

This suggests that we can use the grading in \mathcal{S} to truncate the HOSVD:

$$\mathcal{A} \approx \mathcal{A}_{\tilde{\mathbf{r}}} = \sum_{\mathbf{j}=1}^{\tilde{\mathbf{r}}} \mathcal{S}(\mathbf{j}) \cdot U_1(:, j_1) \circ U_2(:, j_2) \circ U_3(:, j_3)$$

where $\tilde{\mathbf{r}} \leq \mathbf{r}$, i.e., $\tilde{r}_1 \leq r_1$, $\tilde{r}_2 \leq r_2$, and $\tilde{r}_3 \leq r_3$. As with SVD-based low-rank approximation in the matrix case, we simply need a tolerance to determine how to abbreviate the summation in (20). For a deeper discussion of the HOSVD, see [8].

7.6 The Tucker Nearness Problem

Suppose we are given $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathbf{r} = [r_1, r_2, r_3] \leq [n_1, n_2, n_3] = \mathbf{n}$. In the *Tucker nearness problem* we determine $\mathcal{S} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ and matrices $U_1 \in \mathbb{R}^{n_1 \times r_1}$, $U_2 \in \mathbb{R}^{n_2 \times r_2}$, and $U_3 \in \mathbb{R}^{n_3 \times r_3}$ with orthonormal columns such that

$$\phi(U_1, U_2, U_3) = \left\| \mathcal{A} - \sum_{\mathbf{j}=1}^{\mathbf{r}} \mathcal{S}(\mathbf{j}) \cdot U_1(:, j_1) \circ U_2(:, j_2) \circ U_3(:, j_3) \right\|_F$$

is minimized. It is easy to show that

$$\phi(U_1, U_2, U_3) = \| \text{vec}(\mathcal{A}) - (U_3 \otimes U_2 \otimes U_1) \text{vec}(\mathcal{S}) \|_2$$

and so from using normal equations we see that

$$\mathcal{S} = (U_3^T \otimes U_2^T \otimes U_1^T) \cdot \text{vec}(\mathcal{A}).$$

This is why the objective function ϕ does not have S as an argument: the “best S ” is determined by U_1 , U_2 , and U_3 . The goal is to minimize

$$\phi(U_1, U_2, U_3) = \| (I - (U_3 \otimes U_2 \otimes U_1)(U_3^T \otimes U_2^T \otimes U_1^T)) \text{vec}(\mathcal{A}) \|_2.$$

Since $U_3 \otimes U_2 \otimes U_1$ has orthonormal columns, it follows that minimizing this norm is the same as maximizing

$$\phi(U_1, U_2, U_3) = \| (U_3^T \otimes U_2^T \otimes U_1^T) \cdot \text{vec}(\mathcal{A}) \|_2.$$

The reformulations

$$\phi(U_1, U_2, U_3) = \begin{cases} \| U_1^T \cdot A_{(1)} \cdot (U_3 \otimes U_2) \|_F \\ \| U_2^T \cdot A_{(2)} \cdot (U_3 \otimes U_1) \|_F \\ \| U_3^T \cdot A_{(3)} \cdot (U_2 \otimes U_1) \|_F \end{cases}$$

set the stage for a componentwise optimization approach:

Fix U_2 and U_3 and choose U_1 to maximize $\| U_1^T \cdot A_{(1)} \cdot (U_3 \otimes U_2) \|_F$.

Fix U_1 and U_3 and choose U_2 to maximize $\| U_2^T \cdot A_{(2)} \cdot (U_3 \otimes U_1) \|_F$.

Fix U_1 and U_2 and choose U_3 to maximize $\| U_3^T \cdot A_{(3)} \cdot (U_2 \otimes U_1) \|_F$.

These optimization problems can be solved using the SVD. Consider the problem of maximizing $\| Q^T M \|_F$ where $Q \in \mathbb{R}^{m \times r}$ has orthonormal columns and $M \in \mathbb{R}^{m \times n}$ is given. If

$$M = U \Sigma V^T$$

is the SVD of M , then

$$\| Q^T M \|_F^2 = \| Q^T U \Sigma V^T \|_F^2 = \| Q^T U \Sigma \|_F^2 = \sum_{k=1}^r \sigma_k^2 \| Q^T U(:, k) \|_2^2.$$

It is clear that we can maximize the summation by setting $Q = U(:, 1 : r)$. Putting it all together we obtain the following framework.

The Tucker Nearness Problem

Given $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

Compute the HOSVD of \mathcal{A} and determine $\mathbf{r} = [r_1, r_2, r_3] \leq \mathbf{n}$.

Set $U_1 = U_1(:, 1 : r_1)$, $U_2 = U_2(:, 1 : r_2)$, and $U_3 = U_3(:, 1 : r_3)$.

Repeat:

Compute the SVD $\mathcal{A}_{(1)} \cdot (U_3 \otimes U_2) = \tilde{U}_1 \Sigma_1 V_1^T$

and set $U_1 = \tilde{U}_1(:, 1 : \tilde{r}_1)$.

Compute the SVD $\mathcal{A}_{(2)} \cdot (U_3 \otimes U_1) = \tilde{U}_2 \Sigma_2 V_2^T$

and set $U_2 = \tilde{U}_2(:, 1 : \tilde{r}_2)$.

Compute the SVD $\mathcal{A}_{(3)} \cdot (U_2 \otimes U_1) = \tilde{U}_3 \Sigma_3 V_3^T$

and set $U_3 = \tilde{U}_3(:, 1 : \tilde{r}_3)$.

$$U_1^{(opt)} = U_1, U_2^{(opt)} = U_2, U_3^{(opt)} = U_3$$

Using the HOSVD to generate an initial guess makes sense given the discussion in Sect. 7.5. The matrix-matrix products, e.g., $\mathcal{A}_{(1)} \cdot (U_3 \otimes U_2)$, are rich in exploitable Kronecker structure. See [27] for further details.

7.7 A Jacobi Approach

Solving the Tucker Nearness problem is not equivalent to maximizing the “diagonal mass” of the core tensor \mathcal{S} . We briefly describe a Jacobi-like procedure that does. To motivate the main idea, consider the problem of maximizing $\text{tr}(U_1^T A U_2)$ where $A \in \mathbb{R}^{n_1 \times n_2}$ is given, $U_1 \in \mathbb{R}^{n_1 \times n_1}$ is orthogonal, and $U_2 \in \mathbb{R}^{n_2 \times n_2}$ is orthogonal. It is easy to show that the optimum U_1 and U_2 have the property that $U_1^T A U_2$ is diagonal with nonnegative diagonal entries. Thus, the SVD solves this particular “max trace” problem.

Now suppose \mathcal{C} is n -by- n -by- n and define

$$\psi(\mathcal{C}) = \sum_{i=1}^n c_{iii}.$$

Given $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, our goal is to compute orthogonal $U_1 \in \mathbb{R}^{n_1 \times n_1}$, $U_2 \in \mathbb{R}^{n_2 \times n_2}$, and $U_3 \in \mathbb{R}^{n_3 \times n_3}$ so that if the tensor \mathcal{S} is defined by

$$\text{vec}(\mathcal{S}) = (U_3 \otimes U_2 \otimes U_1)^T \text{vec}(\mathcal{A})$$

then $\phi(\mathcal{S})$ is maximized. Here is a Jacobi-like strategy for updating the “current” orthogonal triplet $\{U_1, U_2, U_3\}$ so that new core tensor has a larger trace.

A Jacobi Framework for Computing a Compressed Tucker Representation

Given: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$

Set $U_1 = I_n$, $U_2 = I_n$, $U_3 = I_n$, and $\mathcal{S} = \mathcal{A}$.

Repeat:

Find “simple” orthogonal \tilde{U}_1 , \tilde{U}_2 , and \tilde{U}_3 so that

$$\text{tr}(\mathcal{S} \times_1 \tilde{U}_1 \times_2 \tilde{U}_2 \times_3 \tilde{U}_3) > \text{tr}(\mathcal{S})$$

Update:

$$\mathcal{S} = \mathcal{S} \times_1 \tilde{U}_1 \times_2 \tilde{U}_2 \times_3 \tilde{U}_3$$

$$U_1 = U_1 \tilde{U}_1, U_2 = U_2 \tilde{U}_2, U_3 = U_3 \tilde{U}_3$$

$$U_1^{(opt)} = U_1, U_2^{(opt)} = U_2, U_3^{(opt)} = U_3.$$

We say that this iteration strives for a “compressed” Tucker representation because there is an explicit attempt to compress the information in \mathcal{A} into a relatively small number of near-the-diagonal entries in \mathcal{S} . One idea for $\tilde{U}_3 \otimes \tilde{U}_2 \otimes \tilde{U}_1$ is to use carefully designed Jacobi rotations, e.g.,

$$\tilde{U}_3 \otimes \tilde{U}_2 \otimes \tilde{U}_1 = \begin{cases} I_n \otimes J_{pq}(\beta) \otimes J_{pq}(\alpha) \\ J_{pq}(\beta) \otimes I_n \otimes J_{pq}(\alpha) \\ J_{pq}(\beta) \otimes J_{pq}(\alpha) \otimes I_n \end{cases}.$$

Here, $J_{pq}(\theta)$ is a Jacobi rotation in planes p and q . These updates modify only two diagonal entries: s_{ppp} and s_{qqq} . Sines and cosines can be chosen to increase the resulting trace and their determination leads to a 2-by-2-by-2 Jacobi rotation subproblem.

For example, determine $c_\alpha = \cos(\alpha)$, $s_\alpha = \sin(\alpha)$, $c_\beta = \cos(\beta)$, and $s_\beta = \sin(\beta)$, so that if

$$\begin{bmatrix} \sigma_{ppp} & \sigma_{pqp} \\ \sigma_{qpp} & \sigma_{qqp} \end{bmatrix} = \begin{bmatrix} c_\alpha & s_\alpha \\ -s_\alpha & c_\alpha \end{bmatrix}^T \begin{bmatrix} s_{ppp} & s_{pqp} \\ s_{qpp} & s_{qqp} \end{bmatrix} \begin{bmatrix} c_\beta & s_\beta \\ -s_\beta & c_\beta \end{bmatrix}$$

and

$$\begin{bmatrix} \sigma_{ppq} & \sigma_{pqq} \\ \sigma_{qpq} & \sigma_{qqq} \end{bmatrix} = \begin{bmatrix} c_\alpha & s_\alpha \\ -s_\alpha & c_\alpha \end{bmatrix}^T \begin{bmatrix} s_{ppq} & s_{pqq} \\ s_{qpq} & s_{qqq} \end{bmatrix} \begin{bmatrix} c_\beta & s_\beta \\ -s_\beta & c_\beta \end{bmatrix}$$

then $\sigma_{ppp} + \sigma_{qqq}$ is maximized. See [19].

8 The CP Decomposition

As with the Tucker representation, the CP representation of a tensor expresses the tensor as a sum-of-rank-one tensors. However, it does not involve orthogonality and the core tensor is truly diagonal, e.g., $s_{ijk} = 0$ unless $i = j = k$.

A note about terminology before we begin. The ideas behind the CP decomposition are very similar to the ideas behind the CANDECOMP (Canonical Decomposition) and the PARAFAC (Parallel Factors Decomposition). Thus, “CP” is an effective way to acknowledge the connections.

8.1 CP Representations: The Matrix Case

For matrices, the SVD

$$A = U_1 \Sigma U_2^T = \sum_i \sigma_i U_1(:, i) U_2(:, i)^T$$

is an example of a CP decomposition. But an eigenvalue decomposition also qualifies. If A is diagonalizable, then we have

$$A = U_1 \text{diag}(\lambda_i) U_2^T = \sum_i \lambda_i U_1(:, i) U_2(:, i)^T$$

where $U_2^T = U_1^{-1}$. Of course orthogonality is part of the SVD and biorthogonality ($U_2^T U_1 = I$) figures in eigenvalue diagonalization. This kind of structure falls by the wayside when we graduate to tensors.

8.2 CP Representation: The Tensor Case

We use the order-3 situation to expose the key ideas. The CP representation for an $n_1 \times n_2 \times n_3$ tensor \mathcal{A} has the form

$$\mathcal{A} = \sum_{k=1}^r \lambda_k U_1(:, k) \circ U_2(:, k) \circ U_3(:, k)$$

where λ 's are real scalars and $U_1 \in \mathbb{R}^{n_1 \times r}$, $U_2 \in \mathbb{R}^{n_2 \times r}$, and $U_3 \in \mathbb{R}^{n_3 \times r}$ have unit 2-norm columns. Alternatively, we have

$$\mathcal{A}(i_1, i_2, i_3) = \sum_{j=1}^r \lambda_j \cdot U_1(i_1, j) \cdot U_2(i_2, j) \cdot U_3(i_3, j) \quad (21)$$

$$\text{vec}(\mathcal{A}) = \sum_{j=1}^r \lambda_j \cdot U_3(:, j) \otimes U_2(:, j) \otimes U_1(:, j) \quad (22)$$

In contrast to the Tucker representation,

$$\mathcal{A} = \sum_{j_1=1}^{r_1} \sum_{j_2=1}^{r_2} \sum_{j_3=1}^{r_3} \mathcal{S}(j_1, j_2, j_3) \cdot U_1(:, j_1) \circ U_2(:, j_2) \circ U_3(:, j_3),$$

we see that the CP representation involves a diagonal core tensor. The Tucker representation gives that up in exchange for orthonormal U matrices.

8.3 More About Tensor Rank

As we mentioned in Sect. 5.3, the rank of a tensor \mathcal{A} is the minimum number of rank-1 tensors that sum to \mathcal{A} . *Thus, the length of the shortest possible CP representation of a tensor is its rank.* Our analysis of the 2-by-2-by-2 situation indicates that there are several fundamental differences between tensor rank and matrix rank. Here are some more anomalies:

Anomaly 1. The largest rank attainable for an n_1 -by- \dots - n_d tensor is called the maximum rank. It is *not* a simple formula that depends on the dimensions n_1, \dots, n_d . Indeed, its precise value is only known for small examples. Maximum rank does not equal $\min\{n_1, \dots, n_d\}$ unless $d \leq 2$.

Anomaly 2. If the set of rank- k tensors in $\mathbb{R}^{n_1 \times \dots \times n_d}$ has positive Lebesgue measure, then k is a typical rank. Here are some examples where this quantity is known:

Size	Typical ranks
$2 \times 2 \times 2$	2,3
$3 \times 3 \times 3$	4
$3 \times 3 \times 4$	4,5
$3 \times 3 \times 5$	5,6

For n_1 -by- n_2 matrices, typical rank and maximal rank are both equal to the smaller of n_1 and n_2 .

Anomaly 3. The rank of a particular tensor over the real field may be different than its rank over the complex field.

Anomaly 4. It is possible for a tensor with a given rank to be arbitrarily close to a tensor with lesser rank. Such a tensor is said to be *degenerate*.

For more on the issue of tensor rank, see [9] and [13].

8.4 The Nearest CP Problem

Suppose $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and r are given. The *nearest CP approximation* problem involves finding a vector $\lambda \in \mathbb{R}^r$ and matrices $U_1 \in \mathbb{R}^{n_1 \times r}$, $U_2 \in \mathbb{R}^{n_2 \times r}$, and $U_3 \in \mathbb{R}^{n_3 \times r}$ (with unit 2-norm columns) so that

$$\phi(U_1, U_2, U_3, \lambda) = \left\| \mathcal{A} - \sum_{j=1}^r \lambda_j \cdot U_1(:,j) \circ U_2(:,j) \circ U_3(:,j) \right\|_F$$

is minimized. The objective function for this multilinear optimization problem has three different formulations:

$$\phi(U_1, U_2, U_3, \lambda) = \begin{cases} \left\| \mathcal{A}_{(1)} - \sum_{j=1}^r \lambda_j \cdot U_1(:,j) \otimes (U_3(:,j) \otimes U_2(:,j))^T \right\|_F \\ \left\| \mathcal{A}_{(2)} - \sum_{j=1}^r \lambda_j \cdot U_2(:,j) \otimes (U_3(:,j) \otimes U_1(:,j))^T \right\|_F \\ \left\| \mathcal{A}_{(3)} - \sum_{j=1}^r \lambda_j \cdot U_3(:,j) \otimes (U_2(:,j) \otimes U_1(:,j))^T \right\|_F \end{cases} \quad (23)$$

The summations in these expressions are highly structured matrix-matrix products. To facilitate the discussion we introduce a special variant of the Kronecker product.

8.5 The Khatri-Rao Product

If $B = [b_1 | \dots | b_r] \in \mathbb{R}^{n_1 \times r}$ and $C = [c_1 | \dots | c_r] \in \mathbb{R}^{n_2 \times r}$, then the *Khatri-Rao product* of B and C is given by

$$B \odot C = [b_1 \otimes c_1 | \dots | b_r \otimes c_r] \in \mathbb{R}^{n_1 n_2 \times r}.$$

Thus, the k th column of $B \odot C$ is $B(:, k) \otimes C(:, k)$. The Khatri-Rao product is a submatrix of the Kronecker product. To see this, observe that

$$\begin{aligned} & [b_1 | b_2 | b_3] \otimes [c_1 | c_2 | c_3] \\ &= [b_1 \otimes c_1 | b_1 \otimes c_2 | b_1 \otimes c_3 | b_2 \otimes c_1 | b_2 \otimes c_2 | b_2 \otimes c_3 | b_3 \otimes c_1 | b_3 \otimes c_2 | b_3 \otimes c_3]. \end{aligned}$$

In general, if $B \in \mathbb{R}^{n_1 \times r}$, $C \in \mathbb{R}^{n_2 \times r}$, and $A = B \odot C$, then $A = \tilde{A}(:, 1:r+1:r^2)$ where $\tilde{A} = B \otimes C$.

The Khatri-Rao least square problem

$$\min \| (B \odot C)x - d \|_2 \quad d \in \mathbb{R}^{n_1 n_2}$$

can be solved very fast if we use the method of normal equations:

$$(B \odot C)^T (B \odot C)x = (B \odot C)^T d.$$

To see this, observe that the matrix of coefficients is a pointwise product of r -by- r matrices:

$$(B \odot C)^T (B \odot C) = (B^T B) .* (C^T C).$$

This is an $O((n_1 + n_2)r^2)$ operation. The structure of the right hand side can also be exploited. Indeed

$$(B \odot C)^T d = \begin{bmatrix} c_1^T D b_1 \\ \vdots \\ c_r^T D b_r \end{bmatrix}$$

where $D = \text{reshape}(z, [n_2, n_1])$. This can be computed with $O((n_1 + n_2)r^2)$ work. Overall it requires $O((n_1 + n_2)r^2)$ work to set up the r -by- r normal equation system and $O(r^3)$ flops to solve it. The naive method would involve $O((n_1 n_2)r^2)$ work.

8.6 Equivalent Formulations

We are now ready to formulate an alternating least squares framework for solving the nearest CP problem. Combining our discussion of the Khatri-Rao product with (23) we see that

$$\phi(U_1, U_2, U_3, \lambda) = \begin{cases} \| \mathcal{A}_{(1)}^T - (U_3 \odot U_2) \cdot (\text{diag}(\lambda_j) \cdot U_1^T) \|_F \\ \| \mathcal{A}_{(2)}^T - (U_3 \odot U_1) \cdot (\text{diag}(\lambda_j) \cdot U_2^T) \|_F \\ \| \mathcal{A}_{(3)}^T - (U_2 \odot U_1) \cdot (\text{diag}(\lambda_j) \cdot U_3^T) \|_F \end{cases}.$$

Repeatedly minimizing these expressions with respect to U_1 , U_2 , and U_3 gives rise to the following framework for solving the nearest CP problem:

The Nearest CP Problem

Given: $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and a positive integer r

Set $U_1 = I_{n_1}(:, 1:r)$, $U_2 = I_{n_2}(:, 1:r)$, $U_3 = I_{n_3}(:, 1:r)$

Repeat:

Let X minimize $\| \mathcal{A}_{(1)}^T - (U_3 \odot U_2)X \|_F$.

for $j = 1:r$

$$\lambda_j = \| X(j,:) \|_2, \quad U_1(:,j) = X(j,:)^T / \lambda_j$$

Let Y minimize $\| \mathcal{A}_{(2)}^T - (U_3 \odot U_1)Y \|_F$

for $j = 1:r$

$$\lambda_j = \| Y(j,:) \|_2, \quad U_2(:,j) = Y(j,:)^T / \lambda_j.$$

Let Z minimize $\| \mathcal{A}_{(3)}^T - (U_2 \odot U_1)Z \|_F$

for $j = 1:r$

$$\lambda_j = \| Z(j,:) \|_2, \quad U_3(:,j) = Z(j,:)^T / \lambda_j.$$

$$U_1^{(opt)} = U_1, \quad U_2^{(opt)} = U_2, \quad U_3^{(opt)} = U_3, \text{ and } \lambda^{(opt)} = \lambda.$$

Notice that the least squares problems for X , Y , and Z are each multiple right hand side Khatri-Rao least squares problems. See [27] for more details.

9 The Kronecker Product SVD

Suppose A is a block matrix with uniformly sized blocks. The Kronecker product SVD expresses A as an “optimal” sum of Kronecker products [23, 30]. Recalling that a block matrix A with uniformly sized blocks is a reshaped order-4 tensor, the KSVD can essentially be used to produce an exact representation for order-4 tensors that is a sum of tensor products between matrices.

9.1 The Nearest Kronecker Product Problem

Suppose $A = (A_{ij})$ is an m_1 -by- n_1 block matrix whose blocks are m_2 -by- n_2 , i.e.,

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1,n_1} \\ \vdots & \ddots & \vdots \\ A_{m_1,1} & \cdots & A_{m_1,n_1} \end{bmatrix}, \quad A_{ij} \in \mathbb{R}^{m_2 \times n_2} \quad (24)$$

The nearest Kronecker product problem with respect to this blocking involves finding $B \in \mathbb{R}^{m_1 \times n_1}$ and $C \in \mathbb{R}^{m_2 \times n_2}$ such that

$$\phi_A(B, C) = \|A - B \otimes C\|_F$$

is minimized. This problem can be reshaped into an equivalent nearest-rank-1 problem. Here is an example:

$$\begin{aligned} \phi_A(B, C) &= \left\| \begin{bmatrix} a_{11} & a_{12} & | & a_{13} & a_{14} \\ \hline a_{21} & a_{22} & | & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & | & a_{33} & a_{34} \\ \hline a_{41} & a_{42} & | & a_{43} & a_{44} \\ \hline a_{51} & a_{52} & | & a_{53} & a_{54} \\ \hline a_{61} & a_{62} & | & a_{63} & a_{64} \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \right\|_F \\ &= \left\| \begin{bmatrix} a_{11} & a_{21} & a_{12} & a_{22} \\ \hline a_{31} & a_{41} & a_{32} & a_{42} \\ \hline a_{51} & a_{61} & a_{52} & a_{62} \\ \hline a_{13} & a_{23} & a_{14} & a_{24} \\ \hline a_{33} & a_{43} & a_{34} & a_{44} \\ \hline a_{53} & a_{63} & a_{54} & a_{64} \end{bmatrix} - \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{12} \\ b_{22} \\ b_{32} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} & c_{12} & c_{22} \end{bmatrix}^T \right\|_F \\ &= \|\tilde{A} - \text{vec}(B) \cdot \text{vec}(C)^T\|_F. \end{aligned}$$

This is a believable result since in both formulations every a_{ij} is uniquely differenced with a product of some B -entry and some C -entry.

There is a method behind the set-up of \tilde{A} . The rows of \tilde{A} are vec's of the blocks stacked in the “vec order”, e.g.,

$$\tilde{A} = \begin{bmatrix} a_{11} & a_{21} & a_{12} & a_{22} \\ \hline a_{31} & a_{41} & a_{32} & a_{42} \\ \hline a_{51} & a_{61} & a_{52} & a_{62} \\ \hline a_{13} & a_{23} & a_{14} & a_{24} \\ \hline a_{33} & a_{43} & a_{34} & a_{44} \\ \hline a_{53} & a_{63} & a_{54} & a_{64} \end{bmatrix} = \begin{bmatrix} \text{vec}(A_{11})^T \\ \text{vec}(A_{21})^T \\ \text{vec}(A_{31})^T \\ \text{vec}(A_{12})^T \\ \text{vec}(A_{22})^T \\ \text{vec}(A_{32})^T \end{bmatrix}.$$

Since the closest rank-1 matrix to $\tilde{A} \in \mathbb{R}^{m_1 n_1 \times m_2 n_2}$ is given by its largest singular value and vectors, we obtain the following solution framework:

The $\min \|A - B \otimes C\|_F$ Problem

Given: $A \in \mathbb{R}^{m_1 m_2 \times n_1 n_2}$.

Compute the SVD $\tilde{A} = U \Sigma V^T = \sum_{k=1}^{r_{KP}} \sigma_k u_k v_k^T$.

Define $B_{opt} \in \mathbb{R}^{m_1 \times n_1}$ by $\text{vec}(B) = \sqrt{\sigma_1} u_1$

Define $C_{opt} \in \mathbb{R}^{m_2 \times n_2}$ by $\text{vec}(C) = \sqrt{\sigma_1} v_1$

There is no need to actually compute the full SVD of \tilde{A} since we only require σ_1 , u_1 , and v_1 . The Lanczos SVD process can be applied to compute these quantities [10, p. 571]. This is a particularly attractive strategy if A (and hence \tilde{A}) is large and sparse.

There are important special cases where the Kronecker factor matrices B and C inherit properties of A . For example, if A is symmetric and positive definite, then the same can be said of both B and C . If A is block banded with uniformly banded blocks, then B and C are banded. If A has positive entries, then B and C have positive entries, etc.

We mention that the same “tilde-matrix technology” can be applied to the minimization of

$$\phi(X) = \|A - X \otimes X\|_F$$

and

$$\phi(X) = \|A - (X \otimes Y + Y \otimes X)\|_F$$

provided \tilde{A} is square. The Schur decomposition is involved in the corresponding “tilde” optimization problem.

9.2 The Kronecker Product SVD (KPSVD)

We can obtain a complete Kronecker product representation of A if we use the complete SVD of $\tilde{A} \in \mathbb{R}^{m_1 n_1 \times m_2 n_2}$:

$$\tilde{A} = U \Sigma V^T = \sum_{k=1}^{r_{KP}} \sigma_k u_k v_k^T.$$

If we define the matrices B_k and C_k by $\text{vec}(B_k) = u_k$ and $\text{vec}(C_k) = v_k$, then

$$A = \sum_{k=1}^{r_{KP}} \sigma_k B_k \otimes C_k.$$

We refer to r_{KP} as the *Kronecker rank* of A with respect to the chosen blocking (24). If $r \leq r_{KP}$, then in the Frobenius norm the matrix

$$A_r = \sum_{k=1}^r \sigma_k B_k \otimes C_k$$

is the nearest matrix to A that has Kronecker rank r .

9.3 Order-4 Tensor Approximation Using the KPSVD

If we unfold $\mathcal{A} \in \mathbb{R}^{n \times n \times n \times n}$ into an n^2 -by- n^2 matrix A and compute its KPSVD, then we obtain an expansion of \mathcal{A} that is a sum of matrix-matrix tensor products. For example, if

$$\mathcal{A}_{[1,3] \times [2,4]} = \sum_{k=1}^{r_{KP}} \sigma_k B_k \otimes C_k \quad B_k, C_k \in \mathbb{R}^{n \times n}$$

then

$$\mathcal{A} = \sum_{k=1}^{r_{KP}} \sigma_k \mathcal{C}_k \circ \mathcal{B}_k,$$

i.e.,

$$\mathcal{A}(i_1, i_2, j_1, j_2) = \sum_{k=1}^{r_{KP}} \sigma_k C_k(i_1, i_2) B_k(j_1, j_2).$$

The summations in the above can be abbreviated to obtain best approximations using the optimality features of the KPSVD.

Is it possible to extend this “order-4 technology” to higher order tensors? Preliminary thinking on this leads to various alternating least squares frameworks. For example, suppose $\mathcal{A} \in \mathbb{R}^n$ where $\mathbf{n} = [n, n, n, n, n, n]$ and that we wish to minimize

$$\phi_A(B, C, D) = \| A - B \otimes C \otimes D \|_F$$

where $B, C, D \in \mathbb{R}^{n \times n}$ and

$$A = \mathcal{A}_{[1 \ 3 \ 5] \times [2 \ 4 \ 6]} \in \mathbb{R}^{n^3 \times n^3}.$$

If we regard $A = (A_{ij})$ as an n -by- n block matrix with n^2 -by- n^2 blocks, then

$$\phi_A(B, C, D)^2 = \sum_{i=1}^n \sum_{j=1}^n \| A_{ij} - b_{ij}(C \otimes D) \|_F^2.$$

If we fix C and D then we can minimize ϕ_A by setting

$$b_{ij} = \frac{\text{tr}((C \otimes D)^T A_{ij})}{\| C \|_F^2 \| D \|_F^2}.$$

Similar expressions can be given for the optimum C given that B and D are fixed and for the optimum D given that B and C are fixed. Thus, we could approach the minimization of ϕ_A with a framework that cycles through these componentwise optimizations.

10 The Tensor Train SVD

The idea behind the tensor train representation is to approximate a high-order tensor with a collection of low-order tensors that are linked together through simple, ‘nearest neighbor’ summations [20, 21]. It is a topic worth discussing because it addresses directly the “curse of dimensionality”, see [2, 3, 11].

A tensor train is a special case of a *tensor network*. In a general tensor network the nodes are low-order tensors and each edge represents a single-index summation between the two nodes that it connects. The notation associated with a general tensor network is a major challenge but is quite tractable for tensor trains.

10.1 Tensor Trains and Data Sparsity

Suppose we are given the following matrices and tensors:

$$\begin{aligned}\mathcal{G}_1 &: n_1 \times r_1 \\ \mathcal{G}_2 &: r_1 \times n_2 \times r_2 \\ \mathcal{G}_3 &: r_2 \times n_3 \times r_3 \\ \mathcal{G}_4 &: r_3 \times n_4 \times r_4 \\ \mathcal{G}_5 &: r_4 \times n_5.\end{aligned}$$

Define the integer vectors \mathbf{n} and \mathbf{r} by

$$\mathbf{n} = [n_1, n_2, n_3, n_4, n_5]$$

and

$$\mathbf{r} = [r_1, r_2, r_3, r_4].$$

The tensor $\mathcal{T} \in \mathbb{R}^{\mathbf{n}}$ defined by

$$\mathcal{T}(\mathbf{i}) = \sum_{k=1}^r \mathcal{G}_1(i_1, k_1) \cdot \mathcal{G}_2(k_1, i_2, k_2) \cdot \mathcal{G}_3(k_2, i_3, k_3) \cdot \mathcal{G}_4(k_3, i_4, k_4) \cdot \mathcal{G}_5(k_4, i_5)$$

is a *tensor train* with *carriages* $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \mathcal{G}_4$, and \mathcal{G}_5 . Note that if $\bar{n} = \max\{n_1, n_2, n_3, n_4, n_5\}$ and

$$(r_1 + r_1 r_2 + r_2 r_3 + r_3 r_4 + r_4) \bar{n} \ll n_1 n_2 n_3 n_4 n_5$$

then \mathcal{T} is data sparse. Under what circumstances can we approximate a given tensor \mathcal{A} with a data sparse tensor train? We need a mechanism that exposes the redundancies in \mathcal{A} and which determines the parameters r_1, \dots, r_4 along the way. The procedure involves a sequence of matrix SVDs and careful unfoldings. The carriages turn out to be reshapings of SVD U -matrices.

10.2 Computing an SVD-Based Tensor Train Representation

We outline a framework that can be used to construct a data sparse tensor train approximation to a given tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \dots \times n_5}$. There are four steps:

Step 1. Set $M_1 = \text{reshape}(\mathcal{A}, [n_1, n_2 n_3 n_4 n_5])$ and compute the SVD

$$M_1 = U_1 \Sigma_1 V_1^T = U_1 Z_1$$

where $U_1 \in \mathbb{R}^{n_1 \times r_1}$, $Z_1 = \Sigma_1 V_1^T \in \mathbb{R}^{r_1 \times n_2 n_3 n_4 n_5}$ and $r_1 = \text{rank}(M_1)$. Define

$$\mathcal{G}_1 = U_1.$$

Step 2. Set $M_2 = \text{reshape}(Z_1, [r_1 n_2, n_3 n_4 n_5])$ and compute the SVD

$$M_2 = U_2 \Sigma_2 V_2^T = U_2 Z_2$$

where $U_2 \in \mathbb{R}^{r_1 n_2 \times r_2}$, $Z_2 = \Sigma_2 V_2^T \in \mathbb{R}^{r_2 \times n_3 n_4 n_5}$ and $r_2 = \text{rank}(M_2)$. Define

$$\mathcal{G}_2 = \text{reshape}(U_2, [r_1, n_2, r_2])$$

Step 3. Set $M_3 = \text{reshape}(Z_2, [r_2 n_3, n_4 n_5])$ and compute the SVD

$$M_3 = U_3 \Sigma_3 V_3^T = U_3 Z_3$$

where $U_3 \in \mathbb{R}^{r_2 n_3 \times r_3}$, $Z_3 = \Sigma_3 V_3^T \in \mathbb{R}^{r_3 \times n_4 n_5}$ and $r_3 = \text{rank}(M_3)$. Define

$$\mathcal{G}_3 = \text{reshape}(U_3, [r_2, n_3, r_3])$$

Step 4. Set $M_4 = \text{reshape}(Z_3, [r_3 n_4, n_5])$ and compute the SVD

$$M_4 = U_4 \Sigma_4 V_4^T = U_4 Z_4$$

where $U_4 \in \mathbb{R}^{r_3 n_4 \times r_4}$, $Z_4 \in \mathbb{R}^{r_4 \times n_5}$ and $r_4 = \text{rank}(M_4)$. Define

$$\mathcal{G}_4 = \text{reshape}(U_4, [r_3, n_4, r_4]) \quad \mathcal{G}_5 = Z_4$$

Verification that the \mathcal{G} 's form a tensor train for \mathcal{A} is somewhat involved and we refer the reader to [10, p.742]. However, to acquire some insight, let us assume that $n_1 = \dots = n_5 = n$ and tabulate the sizes of the various matrices that arise in the tensor train computation:

i	$\text{Size}(M_i)$	$r_i = \text{rank}(M_i)$	$\text{Size}(U_i)$	$\text{Size}(Z_i)$	$\text{Size}(\mathcal{G}_i)$
1	$n\text{-by- } n^4$	$r_1 \leq n$	$n\text{-by- } r_1$	$r_1\text{-by- } n^4$	$n\text{-by- } r_1$
2	$r_1 n\text{-by- } n^3$	$r_2 \leq r_1 n \leq n^2$	$n^2\text{-by- } r_2$	$r_2\text{-by- } n^3$	$r_1\text{-by- } n\text{-by- } r_2$
3	$r_2 n\text{-by- } n^2$	$r_3 \leq \min\{r_2 n, n^2\}$	$n^3\text{-by- } r_3$	$r_3\text{-by- } n^2$	$r_2\text{-by- } n\text{-by- } r_3$
4	$r_3 n\text{-by- } n$	$r_4 \leq n$	$n^4\text{-by- } r_4$	$r_4\text{-by- } n$	$r_3\text{-by- } n\text{-by- } r_4$
5	—	—	—	—	$r_4\text{-by- } n$

Notice that the “rate” at which the M_i get thinner and thinner depends upon the rank deficiencies that the SVDs discover along the way. This is important since the amount of work in step i depends upon the dimensions of M_i . The amount of data in the M_i depend upon the r_i :

$$\text{If } N = n^5 \text{ then the amount of data in } \begin{pmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{pmatrix} \text{ is } \begin{pmatrix} N \\ (r_1/n)N \\ (r_2/n^2)N \\ (r_3/n^3)N \end{pmatrix}.$$

There are important applications for which the factors r_i/n^i are very small.

11 Tensor Problems with Multiple Symmetries

In dense matrix computations, the presence of symmetry ($A = A^T$) usually means that work and storage requirements are halved. Further economies can be realized if additional symmetries are around. For example, a centrosymmetric matrix is symmetric about both its diagonal *and* antidiagonal. It turns out that this can reduce work and storage requirements by a factor of four.

Matrix problems with multiple symmetries arise in tensor problems when the tensor in question has multiple symmetries. For example, if $\mathcal{A} \in \mathbb{R}^{n \times n \times n \times n}$ and

$$\mathcal{A}(i_1, i_2, i_3, i_4) = \mathcal{A}(i_2, i_1, i_3, i_4) = \mathcal{A}(i_1, i_2, i_4, i_3) = \mathcal{A}(i_3, i_4, i_1, i_2),$$

then certain unfoldings give rise to $n^2\text{-by- } n^2$ matrices that possess multiple symmetries. This creates interesting challenges. For example, can we efficiently compute structured low-rank approximations to \mathcal{A} by computing structured low-rank approximations to its structured unfoldings? The answer is “yes”.

11.1 A First Look at Multiple Symmetries

A matrix $A \in \mathbb{R}^{n \times n}$ is *centrosymmetric* if $A = A^T$ and $A = E_n A E_n$ where $E_n = I_n$ ($: , n : -1 : 1$). For example,

$$E_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad A = \begin{bmatrix} a & b & c & d \\ b & e & f & c \\ c & f & e & b \\ d & c & b & a \end{bmatrix}.$$

Suppose $n = 2m$. It can be shown that

$$Q_E = \frac{1}{\sqrt{2}} \left[\begin{array}{c|c} I_m & I_m \\ \hline E_m & -E_m \end{array} \right]$$

is orthogonal and

$$Q_E^T \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} Q_E = \begin{bmatrix} A_{11} + A_{12}E_m & 0 \\ 0 & A_{11} - A_{12}E_m \end{bmatrix}.$$

This kind of “free” block diagonalization is at the heart of all structure-exploiting algorithms for centrosymmetric matrix problems. The original problem is basically replaced by a pair of half-sized problems, one for the (1,1) block $A_{11} + A_{12}E_m$ and the other for the (2,2) block $A_{11} - A_{12}E_m$. Thus, we could compute the Schur decomposition of A by computing two half-sized Schur decompositions. Since the complexity of such a calculation is cubic, work will be reduced by a factor of four.

11.2 A Tensor Problem with Multiple Symmetries

We now consider a quantum chemistry problem that gives rise to an order-4 tensor that has several different symmetries. Given a basis $\{\phi_i(\mathbf{r})\}_{i=1}^n$ of atomic orbital functions, we consider the following order-4 tensor:

$$\mathcal{A}(i_1, i_2, i_3, i_4) = \int_{\mathbf{R}^3} \int_{\mathbf{R}^3} \frac{\phi_{i_1}(\mathbf{r}_1)\phi_{i_2}(\mathbf{r}_1)\phi_{i_3}(\mathbf{r}_2)\phi_{i_4}(\mathbf{r}_2)}{\|\mathbf{r}_1 - \mathbf{r}_2\|} d\mathbf{r}_1 d\mathbf{r}_2. \quad (25)$$

This is called the *TEI tensor* and it plays an important role in electronic structure theory and ab initio quantum chemistry. By looking at the integrand it is easy to

show that

$$\mathcal{A}(i_1, i_2, i_3, i_4) = \begin{cases} \mathcal{A}(i_2, i_1, i_3, i_4) \\ \mathcal{A}(i_1, i_2, i_4, i_3) \\ \mathcal{A}(i_3, i_4, i_1, i_2) \end{cases}.$$

We say that \mathcal{A} is ((12)(34))-symmetric.

A common calculation involves switching from the atomic orbital basis to a molecular orbital basis $\{\psi_i(\mathbf{r})\}_{i=1}^n$. If

$$\psi_i(\mathbf{r}) = \sum_{k=1}^n X(i, k) \phi_k(\mathbf{r}) \quad i = 1, 2, \dots, n$$

then the molecular orbital basis tensor

$$\mathcal{B}(j_1, j_2, j_3, j_4) = \int_{\mathbf{R}^3} \int_{\mathbf{R}^3} \frac{\psi_{j_1}(\mathbf{r}_1) \psi_{j_2}(\mathbf{r}_1) \psi_{j_3}(\mathbf{r}_2) \psi_{j_4}(\mathbf{r}_2)}{\|\mathbf{r}_1 - \mathbf{r}_2\|} d\mathbf{r}_1 d\mathbf{r}_2$$

is given by

$$\mathcal{B}(\mathbf{j}) = \sum_{i_1=1}^n \sum_{i_2=1}^n \sum_{i_3=1}^n \sum_{i_4=1}^n \mathcal{A}(\mathbf{i}) \cdot X(i_1, j_1) \cdot X(i_2, j_2) \cdot X(i_3, j_3) \cdot X(i_4, j_4).$$

It can be shown that \mathcal{B} is also ((12)(34))-symmetric.

The computation of \mathcal{B} from \mathcal{A} is neatly expressed in terms of the $[1 \ 3] \times [2 \ 4]$ unfolding:

$$\mathcal{B}_{[1,3] \times [2,4]} = (X \otimes X)^T \mathcal{A}_{[1,3] \times [2,4]} (X \otimes X).$$

This unfolding is based on the tensor-to-matrix mapping

$$\mathcal{A}(i_1, i_2, i_3, i_4) \rightarrow A(i_1 + (i_3 - 1)n, i_2 + (i_4 - 1)n)$$

and has a nice block-level interpretation. If we regard $A = \mathcal{A}_{[1,3] \times [2,4]}$ as an n -by- n block matrix (A_{rs}) with n -by- n blocks, then

$$\mathcal{A}(p, q, r, s) \leftrightarrow [A_{rs}]_{pq}.$$

It follows from the symmetries in tensor \mathcal{A} that the blocks of matrix A are symmetric ($A_{rs}^T = A_{rs}$) and that A is block-symmetric ($A_{rs} = A_{sr}$). Less obvious is that the submatrices

$$\tilde{A}_{ij} = A(i : n : n^2, j : n : n^2)$$

are also symmetric. Here is an $n = 3$ example that showcases the three symmetries:

$$A = \begin{bmatrix} 11 & 12 & 13 & | & 12 & 17 & 18 & | & 13 & 18 & 22 \\ 12 & 14 & \mathbf{15} & | & 17 & 19 & \mathbf{20} & | & 18 & 23 & \mathbf{24} \\ 13 & 15 & 16 & | & 18 & 20 & 21 & | & 22 & 24 & 25 \\ \hline 12 & 17 & 18 & | & 14 & 19 & 23 & | & 15 & 20 & 24 \\ 17 & 19 & \mathbf{20} & | & 19 & 26 & \mathbf{27} & | & 20 & 27 & \mathbf{29} \\ 18 & 20 & 21 & | & 23 & 27 & 28 & | & 24 & 29 & 30 \\ \hline 13 & 18 & 22 & | & 15 & 20 & 24 & | & 16 & 21 & 25 \\ 18 & 23 & \mathbf{24} & | & 20 & 27 & \mathbf{29} & | & 21 & 28 & \mathbf{30} \\ 22 & 24 & 25 & | & 24 & 29 & 30 & | & 25 & 30 & 31 \end{bmatrix}.$$

Also of interest is the $[1, 2] \times [3, 4]$ unfolding $A = \mathcal{A}_{[1,2] \times [3,4]}$ defined by the mapping

$$\mathcal{A}(i_1, i_2, i_3, i_4) \rightarrow A(i_1 + (i_2 - 1)n, i_3 + (i_4 - 1)n).$$

Here is an example:

$$A = \begin{bmatrix} 11 & 12 & 13 & | & 12 & 14 & 15 & | & 13 & 15 & 16 \\ 12 & 17 & 18 & | & 17 & 19 & 20 & | & 18 & 20 & 21 \\ 13 & 18 & 22 & | & 18 & 23 & 24 & | & 22 & 24 & 25 \\ \hline 12 & 17 & 18 & | & 17 & 19 & 20 & | & 18 & 20 & 21 \\ 14 & 19 & 23 & | & 19 & 26 & 27 & | & 23 & 27 & 28 \\ 15 & 20 & 24 & | & 20 & 27 & 29 & | & 24 & 29 & 30 \\ \hline 13 & 18 & 22 & | & 18 & 23 & 24 & | & 22 & 24 & 25 \\ 15 & 20 & 24 & | & 20 & 27 & 29 & | & 24 & 29 & 30 \\ 16 & 21 & 25 & | & 21 & 28 & 30 & | & 25 & 30 & 31 \end{bmatrix}.$$

This unfolding of a $((1, 2), (3, 4))$ symmetric tensor is symmetric and has the property that each column reshapes to a symmetric matrix, e.g.,

$$\text{reshape}(A(:, 1), [3 3]) = \begin{bmatrix} 11 & 12 & 13 \\ 12 & 14 & 15 \\ 13 & 15 & 16 \end{bmatrix}$$

We call this *perfect shuffle symmetry*.

11.3 Perfect Shuffle Symmetry

An n^2 -by- n^2 matrix A is *PS-symmetric* if it is symmetric and satisfies

$$A = \Pi_{n,n} A \Pi_{n,n}$$

where $\Pi_{n,n}$ is the perfect shuffle permutation

$$\Pi_{n,n} = I_{n^2}(:, \mathbf{v}), \quad \mathbf{v} = [1 : n : n^2 | 2 : n : n^2 | \cdots | n : n : n^2].$$

Here is an example:

$$\Pi_{3,3} = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right].$$

Because $\Pi_{n,n}$ is symmetric it has just two eigenvalues: +1 and -1. Consider the eigenvector equation

$$\Pi_{3,3}x = \left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ \mathbf{x}_{12} \\ x_{22} \\ x_{32} \\ x_{13} \\ x_{23} \\ x_{33} \end{bmatrix} = \pm \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ \mathbf{x}_{21} \\ x_{22} \\ x_{23} \\ x_{31} \\ x_{32} \\ x_{33} \end{bmatrix}.$$

If $\Pi_{n,n}x = x$, then `reshape(x, [n, n])` is symmetric. If $\Pi_{n,n}x = -x$, then `reshape(x, [n, n])` is skew-symmetric.

11.4 Block Diagonalization

Suppose $A \in \mathbb{R}^{n^2 \times n^2}$ is PS-symmetric and λ is a distinct eigenvalue. Thus,

$$Ax = \lambda x \quad \Rightarrow \quad A(\Pi_{n,n}x) = \lambda(\Pi_{n,n}x)$$

from which we may conclude that either $\Pi_{n,n}x = x$ or $\Pi_{n,n}x = -x$. The first case says that x reshapes to an n -by- n symmetric matrix while the second case says that x reshapes to an n -by- n skew-symmetric matrix. From this we may conclude that the subspaces

$$S_{\text{sym}} = \{x \in \mathbb{R}^{n^2} \mid \text{reshape}(x, [n n]) \text{ is symmetric}\}$$

$$S_{\text{skew}} = \{x \in \mathbb{R}^{n^2} \mid \text{reshape}(x, [n n]) \text{ is skew-symmetric}\}$$

are invariant for A . Moreover, $S_{\text{sym}} = S_{\text{skew}}^\perp$. Here is an orthogonal matrix whose columns span these subspaces:

$$Q_{3,3} = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & -1 \\ 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} = [Q_{\text{sym}} \mid Q_{\text{skew}}] \quad (26)$$

Here are a pair of column reshapings taken from this matrix:

$$Q_{3,3}(:, 4) \equiv \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad Q_{3,3}(:, 7) \equiv \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

It follows that if $A \in \mathbb{R}^{n^2 \times n^2}$ is PS-symmetric, then

$$Q_{n,n}^T A Q_{n,n} = \begin{bmatrix} \times & \times & \times & \times & \times & \times & | & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times & | & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times & | & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times & | & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times & | & 0 & 0 & 0 \\ \times & \times & \times & \times & \times & \times & | & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & | & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & | & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & | & \times & \times & \times \end{bmatrix} = \begin{bmatrix} A_{\text{sym}} & 0 \\ 0 & A_{\text{skew}} \end{bmatrix}. \quad (27)$$

This “free” block diagonalization can be effectively exploited as we now show.

11.5 Low-Rank PS-Symmetric Approximation

In certain important quantum chemistry applications, the n^2 -by- n^2 matrix A in (27) is positive definite and very near a rank- n matrix. Our plan is to approximate the diagonal blocks A_{sym} and A_{skew} using Cholesky with diagonal pivoting.

Recall that pivoted LDL can be used to compute an approximate rank- r approximation to a positive definite matrix:

$$PAP^T \approx LDL^T \quad \begin{cases} P \text{ is a permutation} \\ L \in \mathbb{R}^{n^2 \times r} \text{ is unit lower triangular} \\ D = \text{diag}(d_i), d_1 \geq d_2 \geq \dots \geq d_r > 0 \end{cases}$$

e.g.,

$$PAP^T \approx \begin{bmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & \times \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \\ 0 & 0 & d_3 \end{bmatrix} \begin{bmatrix} \times & \times \\ 0 & \times \\ 0 & 0 & \times & \times & \times & \times & \times & \times \end{bmatrix}$$

See [10, p.167].

It follows from (26) and (27) that if we compute the low-rank LDL decompositions

$$A_{\text{sym}} \approx P_{\text{sym}}^T L_{\text{sym}} D_{\text{sym}} L_{\text{sym}}^T P_{\text{sym}}$$

$$A_{\text{skew}} \approx P_{\text{skew}}^T L_{\text{skew}} D_{\text{skew}} L_{\text{skew}}^T P_{\text{skew}}$$

then

$$A \approx V_{\text{sym}} D_{\text{sym}} V_{\text{sym}}^T + V_{\text{skew}} D_{\text{skew}} V_{\text{skew}}^T \quad (28)$$

where

$$V_{\text{sym}} = Q_{\text{sym}} P_{\text{sym}}^T L_{\text{sym}}$$

and

$$V_{\text{skew}} = Q_{\text{skew}} P_{\text{skew}}^T L_{\text{skew}}$$

It is easy to verify that these low-rank matrices are also PS-symmetric. When the structured approximation (28) to $A = \mathcal{A}_{[1,2] \times [3,4]}$ is substituted into

$$\mathcal{B}_{[1,2] \times [3,4]} = (X \otimes X)^T \mathcal{A}_{[1,2] \times [3,4]} (X \otimes X).$$

then the volume of work is greatly reduced. See [31].

References

1. G. Baumgartner, A. Auer, D. Bernholdt, A. Bibireata, V. Choppella, D. Cociorva, X. Gao, R. Harrison, S. Hirata, S. Krishnamoorthy, S. Krishnan, C. Lam, Q. Lu, M. Nooijen, R. Pitzer, J. Ramanujam, P. Sadayappan, A. Sibiryakov , Synthesis of high-performance parallel programs for a class of ab initio quantum chemistry models. Proc. IEEE **93**(2), 276–292 (2005)
2. G. Beylkin, M.J. Mohlenkamp, Numerical operator calculus in higher dimensions. Proc. Natl. Acad. Sci. **99**(16), 10246–10251 (2002)
3. G. Beylkin, M.J. Mohlenkamp, Algorithms for numerical analysis in high dimensions. SIAM J. Sci. Comp. **26**, 2133–2159 (2005)
4. P. Comon, G. Golub, L.-H. Lim, B. Mourrain, Genericity and rank deficiency of high order symmetric tensors. Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP '06) **31**(3), 125–128 (2006)
5. P. Comon, G. Golub, L.-H. Lim, B. Mourrain, Symmetric tensors and symmetric tensor rank. SIAM J. Matrix Anal. Appl. **30**, 1254–1279 (2008)
6. L. de Lathauwer, Signal Processing Based on Multilinear Algebra. Ph.D. thesis, K.U. Leuven, 1997
7. L. De Lathauwer, P. Comon, B. De Moor, J. Vandewalle, Higher-order power method—application in independent component analysis, in *Proceedings of the International Symposium on Nonlinear Theory and Its Applications (NOLTA '95)*, Las Vegas, NV (1995), pp. 91–96
8. L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**, 1253–1278 (2000)
9. V. De Silva, L.-H. Lim, Tensor rank and the ill-posedness of the best low-rank approximation problem. SIAM J. Matrix Anal. Appl. **30**, 1084–1127 (2008)
10. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 4th edn. (Johns Hopkins University Press, Baltimore, MD, 2013)
11. W. Hackbusch, B.N. Khoromskij, Tensor-product approximation to operators and functions in high dimensions. J. Complexity **23**, 697–714 (2007)
12. H.V. Henderson, S.R. Searle, The vec-permutation matrix, the vec operator and Kronecker products: a review. Linear Multilinear Algebra **9**, 271–288 (1981)
13. C.J. Hillar, L.-H. Lim, Most tensor problems are NP-hard. J. ACM **60**(6), Art. 33–47 (2013)
14. E. Kofidis, P.A. Regalia, On the best rank-1 approximation of higher-order supersymmetric tensors. SIAM J. Matrix Anal. Appl. **23**, 863–884 (2002)
15. T.G. Kolda, B.W. Bader, Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. ACM Trans. Math. Softw. **32**, 635–653 (2006)
16. T.G. Kolda, B.W. Bader, Tensor decompositions and applications. SIAM Rev. **51**, 455–500 (2009)
17. L.-H. Lim, Singular values and eigenvalues of tensors: a variational approach, in *Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP '05)*, vol. 1 (2005), pp. 129–132
18. L.-H. Lim, Tensors and hypermatrices, in *Handbook of Linear Algebra*, Chap. 15, 2nd edn., ed. by L. Hogben (CRC Press, Boca Raton, FL, 2013), 30 pp.

19. C. Martin, C. Van Loan, A Jacobi-type method for computing orthogonal tensor decompositions. *SIAM J. Matrix Anal. Appl.* **30**, 1219–1232 (2008)
20. I.V. Oseledets, E.E. Tyrtyshnikov, Breaking the curse of dimensionality, or how to use SVD in many dimensions. *SIAM J. Sci. Comput.* **31**, 3744–3759 (2008)
21. I. Oseledets, E. Tyrtyshnikov, TT-cross approximation for multidimensional arrays, *Linear Algebra Appl.* **432**, 70–88 (2010)
22. C.C. Paige, C.F. Van Loan, A Schur decomposition for Hamiltonian matrices. *Linear Algebra Appl.* **41**, 11–32 (1981)
23. N. Pitsianis, C. Van Loan, Approximations with Kronecker products, in *Linear Algebra for Large Scale and Real-Time Applications*, ed. by M.S. Moonen, G.H. Golub (Kluwer, Dordrecht, 1993), pp. 293–314
24. J. Poulson, B. Marker, R.A. van de Geijn, J.R. Hammond, N.A. Romero, Elemental: A new framework for distributed memory dense matrix computations. *ACM Trans. Math. Softw.* **39**(2), 13:1–13:24, February 2013 (2014). arXiv:1301.7744
25. S. Ragnarsson, C.F. Van Loan, Block tensor unfoldings. *SIAM J. Matrix Anal. Appl.* **33**(1), 149–169 (2012)
26. S. Ragnarsson, C.F. Van Loan, Block tensors and symmetric embeddings. *Linear Algebra Appl.* **438**, 853–874 (2013)
27. A. Smilde, R. Bro, P. Geladi, *Multi-way Analysis with Applications in the Chemical Sciences* (Wiley, Chichester, 2004)
28. E. Solomonik, D. Matthews, J. Hammond, J. Demmel, Cyclops Tensor Framework: reducing communication and eliminating load imbalance in massively parallel contractions, Berkeley Technical Report No. UCB/EECS-2013-1 (2013)
29. C.F. Van Loan, *Computational Frameworks for the Fast Fourier Transform* (SIAM, Philadelphia, PA, 1992)
30. C. Van Loan, The ubiquitous Kronecker product. *J. Comput. Appl. Math.* **123**, 85–100 (2000)
31. C.F. Van Loan, J.P. Vokt, Approximating matrices with multiple symmetries. *SIAM J. Matrix Anal. Appl.* **36**(3), 974–993 (2015)

Matrix Structures in Queueing Models

Dario A. Bini

Abstract Matrix structures are ubiquitous in linear algebra problems stemming from scientific computing, engineering and from any mathematical models of the real world. They translate, in matrix language, the specific properties of the physical problem. Often, structured matrices reveal themselves in a clear form and apparently seem to show immediately all their properties. Very often, structures are hidden, difficult to discover, and their properties seem to be hardly exploitable. In this note, we rely on the research area of queueing models and Markov chains to present, motivate, and analyze from different points of view the main matrix structures encountered in the applications. We give an overview of the main tools from the *structured matrix technology* including Toeplitz matrices—with their asymptotic spectral properties and their interplay with polynomials—circulant matrices and other trigonometric algebras, preconditioning techniques, the properties of displacement operators and of displacement rank, fast and superfast Toeplitz solvers, Cauchy-like matrices, and more. Among the hidden structures, besides the class of Toeplitz-like matrices, we just recall some properties of rank-structured matrices like quasi-separable matrices. Then we focus our attention to finite dimensional and to infinite dimensional Markov chains which model queueing problems. These chains, which involve block Hessenberg block Toeplitz matrices of finite and infinite size, are efficiently solved by using the tools of the structured matrix technology introduced in the first part. In this note we also provide pointers to some related recent results and to the current research.

1 Introduction to Matrix Structures

In these lecture notes, we will play with matrix structures from different view points, present through several examples the main structures encountered in queueing models, describe the most important computational tools for their algorithmic

D.A. Bini (✉)

Dipartimento di Matematica, Università di Pisa, Pisa, Italy
e-mail: bini@dm.unipi.it

analysis, and provide the application of these tools to designing effective numerical algorithms for the solution of wide classes of queuing models.

One of the first issues that we wish to clarify in this presentation is to figure out why and how matrix structures are ubiquitous in mathematical models of the real world, and how they translate in the language of linear algebra the main underlying properties of these models.

In this presentation, we wish also to collect the most meaningful and useful tools for the detection and the analysis of some apparently hidden matrix structures of relevant interest. Another fundamental aspect concerns the analysis of the main methodologies for exploiting structures in order to design effective algorithms for solving the associated computational problems.

We will follow this line in the framework of queuing models and Markov chains since this research area provides a rich variety of clear examples of structured matrices in applied mathematics.

In this course we would like to let people discover and enjoy the beauty and the variety of the mathematical concepts that we will encounter together with their interplay with algorithms and applications. In this regard, it is relevant to cite the following sentence by Alexander Grothendieck, who received the Fields medal in 1966 [42, p. 48]: *“If there is one thing in mathematics that fascinates me more than anything else (and doubtless always has), it is neither number nor size, but always form. And among the thousand-and-one faces whereby form chooses to reveal itself to us, the one that fascinates me more than any other and continues to fascinate me, is the structure hidden in mathematical things”*.

In linear algebra, structures reveal themselves in terms of matrix properties. Their analysis and exploitation is not just a challenge but it is also a mandatory step which is needed to design *highly effective ad hoc algorithms* for solving 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 huge sizes. It should be also said that large scale problems are usually accompanied by strong matrix structures and that these structures are the algebraic counterpart of the *peculiarity* of the mathematical model that the matrix itself describes. In our point of view, matrix structures are the synonymous of the peculiarity of the object represented by the matrix.

Analyzing structures from the theoretical point of view, turning them in effective solution algorithms, constructing software which implements them and verifying its effectiveness by direct computation is one of the most exciting challenge that covers abstract theory, design and analysis of algorithm, software implementation and applications.

Often, structured matrices reveal themselves in a clear form and apparently seem to show immediately all their properties. Very often, structures are hidden, difficult to discover, and their properties are hardly exploitable, they show themselves slowly and their analysis requires long time and efforts.

This note is divided in three parts. In the first part, formed by this section and by Sect. 2, we present and motivate the main structures of interest encountered in Markov chains with the description of some meaningful queueing models in which they originate. Here, we state the computational problems that we will treat next.

The second part, formed by Sect. 3, contains the analysis of the main matrix structures encountered in stochastic processes, with the description of the main computational tools and the algorithms for their treatment.

The third part, formed by Sects. 4–6, is devoted to algorithms for solving structured Markov chains.

Concerning the second part, we address the attention more specifically to Toeplitz matrices which are ubiquitous in applications and play a relevant role in the analysis of queues. Here, after recalling some classical results, we examine the interplay between Toeplitz matrices, polynomials and power series, discuss on the role of certain matrix algebras, like circulants, and fast trigonometric transforms; we present the concept of displacement operators, and outline fast and super-fast algorithms for Toeplitz-like matrix inversion; finally, we recall the asymptotic spectral properties of Toeplitz matrices with applications to preconditioning.

Concerning part 3, we first consider finite Markov chains where matrices are block tridiagonal almost block Toeplitz (the so called Quasi-Birth-Death processes). Then we consider (generalized) block Hessenberg almost block Toeplitz matrices which model a wide range of processes like M/G/1-type, G/M/1-type, and Non-Skip-Free Markov chains. Then we move to the case of infinite matrices where the Wiener-Hopf canonical factorization provides a fundamental tool.

Finally, in the last section, we analyze some related structures stemming from tree-like processes and from Markovian binary trees, and then apply the Toeplitz matrix machinery for the design and analysis of algorithms for computing the exponential of a block triangular block Toeplitz matrix. This problem is encountered in the Erlangian approximation of Markovian fluid queues.

In this analysis we will encounter general blocks, blocks which are Toeplitz themselves, blocks which are tridiagonal and show how the technology of structured matrices is fundamental to design effective algorithms for the solution of these problems.

1.1 Some Examples of Matrix Structures

Here we report some examples of structures, with the properties of the physical model that they represent, the associated algorithms and their complexities. Structure and computational properties, together with the main concepts on which they rely and the computational tools, will be examined in the next section.

We start with clear structures and then consider hidden structures.

Definition 1 An $n \times n$ matrix $A = (a_{i,j})$ is Toeplitz if there exist $\alpha_k \in \mathbb{C}$, $k = -n + 1, \dots, n - 1$ such that $a_{i,j} = \alpha_{j-i}$ for $i, j = 1, \dots, n$.

An example of Toeplitz matrix is given below

$$\begin{bmatrix} a & b & c & d \\ e & a & b & c \\ f & e & a & b \\ g & f & e & a \end{bmatrix}.$$

Toeplitz matrices are ubiquitous in mathematical models, they are encountered when some mathematical object involved in the model is shift invariant. There are many situations in theory and in the applications where this property occurs. Typical examples come from digital signal processing where the response to signal is time invariant, image restoration when the point-spread function which determines the blur of a point is space invariant, from Markov chains when the probability that the system changes from state i to state j depends only on the difference $i - j$; other problems where Toeplitz matrices and related structures are encountered are finite differences approximation, polynomial and power series computations, integral equations with a shift-invariant kernel, and more.

Bidimensional problems lead to block Toeplitz matrices with Toeplitz blocks. Multidimensional problems lead to block Toeplitz matrices whose blocks are block Toeplitz matrices themselves with Toeplitz blocks.

The main algorithms for manipulating Toeplitz matrices rely on FFT. In particular, as we will see later on,

- multiplying an $n \times n$ Toeplitz matrix and a vector costs $O(n \log n)$ arithmetic operations (ops);
- solving a Toeplitz system costs $O(n^2)$ ops, by using *fast* algorithms; the asymptotic cost is reduced to $O(n \log^2 n)$ ops by using *super-fast* algorithms
- approximating the solution of a Toeplitz system by means of the preconditioned conjugate gradient method costs $O(n \log n)$ ops.

Definition 2 Given a fixed integer $k > 0$, an $n \times n$ *band matrix* $A = (a_{i,j})$ with $2k + 1$ diagonals is such that $a_{|i-j|} = 0$ for $|i - j| > k$, where $n > k$.

Here is an example of tridiagonal matrix obtained with $k = 1$ where we adopt the convention that blanks mean zero entries.

$$\begin{bmatrix} a & b & & & \\ c & d & e & & \\ & f & g & h & \\ & & i & l & \end{bmatrix}.$$

Band matrices are encountered when some mathematical object has locality properties. Typical examples are point-spread functions, where the effect of the blur of a single pixel in an image is localized in a small neighborhood of the

pixel itself; the input response function in a dynamical system; the finite difference approximation to a derivative. Often, band matrices are combined with the Toeplitz structure.

Bidimensional problems lead to block band matrices with banded blocks; multidimensional problems lead to block banded matrices where the blocks are recursively block banded matrices themselves.

The main algorithms for manipulating band matrices have a cost which is linear in n and quadratic in k . In particular for a $(2k - 1)$ -diagonal matrix A of size $n \times n$

- the LU and the QR factorizations cost $O(nk^2)$ ops;
- solving linear systems costs $O(nk^2)$ ops;
- the QR iteration for symmetric tridiagonal matrices costs $O(n)$ operations per step.

Toeplitz and band matrices are examples where the structure is expressed as a linear space. For this reason, it is almost easily detectable. There are situations where the whole class of matrices is hard to detect looking at a single instance of the class.

For instance, the set of matrices formed by the inverses of all the nonsingular $n \times n$ Toeplitz matrices is a matrix class endowed with a structure. However, they are not a linear space and are apparently hard to detect. Another example is the class formed by the products of a lower triangular and an upper triangular Toeplitz matrix. This is a matrix class depending quadratically on $2n - 1$ parameters. The following matrix is an instance in this class.

$$\begin{bmatrix} 20 & 15 & 10 & 5 \\ 4 & 23 & 17 & 11 \\ 8 & 10 & 27 & 19 \\ 4 & 11 & 12 & 28 \end{bmatrix}.$$

Its structure is quite hidden and hardly detectable if one does not have the appropriate tools.

It is interesting to observe that the inverse of a Toeplitz matrix is not generally Toeplitz. However, as we will see next, inverses of Toeplitz matrices share a structure of the form $L_1 U_1 + L_2 U_2$ where L_i, U_i^T are lower triangular Toeplitz matrices, for $i = 1, 2$.

In certain applications, it is useful to introduce more general classes like the one formed by matrices of the kind $A = \sum_{i=1}^k L_i U_i$ where $k \ll n$ and L_i, U_i^T are lower triangular Toeplitz matrices. We will see an example in the framework of Markov chains and will introduce very simple tools to detect this hidden generalized Toeplitz structure called *displacement structure*.

Another hidden structure of great interest is the one formed by quasi-separable matrices. It is interesting to observe that the inverse of an invertible tridiagonal matrix A is not generally banded. However, one can prove that the submatrices of A^{-1} contained in the upper (or lower) triangular part have rank at most 1. Matrices

sharing this property are said *quasi-separable*. If A is also irreducible then

$$A^{-1} = \text{tril}(\mathbf{u}\mathbf{v}^T) + \text{triu}(\mathbf{w}\mathbf{z}^T), \quad \mathbf{u}, \mathbf{v}, \mathbf{w}, \mathbf{z} \in \mathbb{R}^n,$$

where $\text{tril}(A)$ and $\text{triu}(A)$ denote the lower triangular and the upper triangular matrix formed by the corresponding entries of A .

According to this definition, tridiagonal matrices as well as their inverses are quasi-separable.

One can define more general classes where the submatrices contained in the upper/lower triangular part have rank bounded by a positive integer $k << n$. They form the class of *rank structured matrices*.

An example of quasi-separable matrix is displayed below. It is easy to check that all the submatrices contained in the upper triangular part or in the lower triangular part have rank 1.

$$\begin{bmatrix} 7 & 1 & 3 & 4 & 2 \\ 1 & 2 & 6 & 8 & 4 \\ 6 & 12 & -3 & -4 & -2 \\ 4 & 8 & -2 & 6 & 3 \\ 2 & 4 & -1 & 3 & 8 \end{bmatrix}.$$

2 Structures Encountered in Queuing Models

A research area where structured matrices play an important role is the analysis of Markov chains and queuing models. Here, we recall few notions about Markov chains and give some meaningful examples which explain why and how (block) Toeplitz matrices as well as (block) banded or (block) Hessenberg matrices model wide classes of queuing problems. The computational issues encountered in this section will be treated in Sects. 4, 5, and 7.

2.1 Markov Chains

Let us start with some basic definitions. For more details we refer the reader to the books [14, 62, 72].

A *stochastic process*: is a family $\{X_t \in E : t \in T\}$ of random variables X_t taking values in a *state space* E , which is denumerable. The entries of the family are parametrized for $t \in T$ where T is the *time space* which is denumerable as well.

For instance, X_t can be the number of customers in a line at time t waiting to be served. This number depends on some random event, like the arrival of new

customers, and on the service time. If we do not put upper bounds on the length of the queue, then we may choose $E = \mathbb{N}$, the set of natural numbers, and the time set T can be identified with \mathbb{N} as well.

Given random variables X and Y and $a, b \in E$, the expression $\mathcal{P}(X = a | Y = b)$ denotes the conditional probability that $X = a$ given that $Y = b$.

A *Markov chain* is a stochastic process $\{X_n\}_{n \in T}$ such that

$$\mathcal{P}(X_{n+1} = j | X_0 = i_0, X_1 = i_1, \dots, X_n = i_n) = \mathcal{P}(X_{n+1} = j | X_n = i_n).$$

That is, the state X_{n+1} of the system at time $n + 1$ depends only on the state X_n at time n . It does not depend on the past history of the system.

The *homogeneity assumption* states that the probability that the system changes from state i to state j is *independent* of time. That is,

$$\mathcal{P}(X_{n+1} = j | X_n = i) = \mathcal{P}(X_1 = j | X_0 = i) \quad \forall n.$$

The matrix of probabilities

$$P = (p_{i,j})_{i,j \in T}, \quad p_{i,j} = \mathcal{P}(X_1 = j | X_0 = i)$$

is said *transition matrix* of the Markov chain. In particular, P is *row-stochastic*, that is $p_{i,j} \geq 0$, $\sum_{j \in T} p_{i,j} = 1$. Equivalently, $P\mathbf{e} = \mathbf{e}$, where $\mathbf{e} = (1, 1, \dots)^T$ is the vector with all components 1.

We recall some elementary but fundamental properties. If $\mathbf{x}^{(k)} = (x_i^{(k)})$ denotes the probability vector of the Markov chain at time k , that is, $x_i^{(k)} = \mathcal{P}(X_k = i)$, then

$$\mathbf{x}^{(k+1)T} = \mathbf{x}^{(k)T} P, \quad k \geq 0.$$

Moreover, if the limit $\boldsymbol{\pi} = \lim_k \mathbf{x}^{(k)}$ exists, then by continuity $\boldsymbol{\pi}^T = \boldsymbol{\pi}^T P$.

The vector $\boldsymbol{\pi}$ is said the *stationary probability vector* and represents the asymptotic behavior of the Markov chain. Existence, uniqueness, convergence of the sequence $\mathbf{x}^{(k)}$ to $\boldsymbol{\pi}$ whatever is $\mathbf{x}^{(0)}$, together with the design and analysis of algorithms for computing $\boldsymbol{\pi}$, are the main issues concerning the vector $\boldsymbol{\pi}$.

If P is finite then the Perron-Frobenius theorem provides the answers to many questions. Define the spectral radius of an $n \times n$ matrix A as

$$\rho(A) = \max_i |\lambda_i(A)|, \quad \lambda_i(A) \text{ eigenvalue of } A.$$

Observe that if $A = P$ then $\rho(A) = 1$ and \mathbf{e} is an eigenvector corresponding to the eigenvalue $\lambda = 1$ since $P\mathbf{e} = \mathbf{e}$. In general we have the following classical result [80].

Theorem 1 (Perron-Frobenius) Let $A = (a_{i,j})$ be an $n \times n$ matrix such that $a_{i,j} \geq 0$, let A be irreducible. Then

- the spectral radius $\rho(A)$ is a positive simple eigenvalue;
- the corresponding right and left eigenvectors are positive;
- if $B \geq A$ and $B \neq A$ then $\rho(B) > \rho(A)$.

The positive eigenvectors are said *Perron vectors*. In the case of the transition matrix P , if any other eigenvalue of P has modulus less than 1, then $\lim_k P^k = \mathbf{e}\boldsymbol{\pi}^T$ and for any $\mathbf{x}^{(0)} \geq 0$ such that $\mathbf{x}^{(0)T}\mathbf{e} = 1$ it holds

$$\lim_k \mathbf{x}^{(k)T} = \boldsymbol{\pi}^T, \quad \boldsymbol{\pi}^T P = \boldsymbol{\pi}^T, \quad \boldsymbol{\pi}^T \mathbf{e} = 1.$$

This property can be easily deduced from the Jordan form of P and is true if P has positive entries.

In the case where $P = (p_{i,j})_{i,j \in \mathbb{N}}$, that is, P is infinite, the situation is more complicated. In fact, the existence of $\boldsymbol{\pi} > 0$ such that $\boldsymbol{\pi}^T \mathbf{e} = 1$ is not guaranteed. Here are some examples where blanks in the matrix denote zero entries.

$$P = \begin{bmatrix} 0 & 1 & & \\ 3/4 & 0 & 1/4 & \\ & 3/4 & 0 & 1/4 \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \quad \boldsymbol{\pi}^T = [\frac{1}{2}, \frac{2}{3}, \frac{2}{9}, \frac{2}{27}, \dots] \in \ell^1 \text{ positive recurrent,}$$

$$P = \begin{bmatrix} 0 & 1 & & \\ 1/4 & 0 & 3/4 & \\ 1/4 & 0 & 3/4 & \\ & \ddots & \ddots & \ddots \end{bmatrix}, \quad \boldsymbol{\pi}^T = [1, 4, 12, 16, \dots] \notin \ell^\infty \text{ transient,}$$

$$P = \begin{bmatrix} 0 & 1 & & \\ 1/2 & 0 & 1/2 & \\ 1/2 & 0 & 1/2 & \\ & \ddots & \ddots & \ddots \end{bmatrix}, \quad \boldsymbol{\pi}^T = [1/2, 1, 1, \dots] \notin \ell^1 \text{ null recurrent.}$$

Here, ℓ^1 denotes the set of sequences $\{x_i\}$ such that $\sum_{i \in \mathbb{N}} |x_i|$ is finite, while ℓ^∞ denotes the set of sequences $\{x_i\}$ such that $\sup_{i \in \mathbb{N}} |x_i|$ is finite.

Intuitively, positive recurrence means that the global probability that the state changes into a “forward” state is less than the global probability of a change into a backward state. In this way, the probabilities π_i of the stationary probability vector get smaller and smaller as long as i grows. Transience means that it is more likely to move forward. Null recurrence means that the probabilities to move forward/backward are equal. Positive recurrence, plus additional properties, guarantees that even in the infinite case $\lim_k \mathbf{x}^{(k)} = \boldsymbol{\pi}$.

For wide classes of Markov chains, positive/null recurrence and transience can be detected by means of the sign of a computable parameter called *drift*.

In the framework of Markov chains, the most important computational problem is designing efficient algorithms for computing $\pi_1, \pi_2, \dots, \pi_k$ for any given integer k .

2.2 Some Examples

Here, we provide some examples of Markov chains which show how matrix structures reflect specific properties of the physical model. We refer the reader to [4, 14, 53, 54, 58, 60, 61].

The most simple example governed by a tridiagonal almost Toeplitz matrix is the case of the *random walk* model. At each instant a particle Q can move along a line by a unit step. The movement is random and is determined by two numbers $p, q \in (0, 1)$. More precisely, p is the probability that the particle moves right, q is the probability that it moves left, and $1 - p - q$ is the probability that it does not move. Clearly, the position of Q at time $n + 1$ depends only on the position of Q at time n and this system can be modeled by a Markov chain where the set of states E is the set of the coordinates which the particle Q can take, $T = \mathbb{N}$ is the set of time steps, and X_n is the random variable which provides the coordinate of Q at time n . The set E can be \mathbb{Z} , that is, the particle moves along a straight line; it can be \mathbb{N} , in this case the particle moves along a half line; or it can be the finite set $\{1, 2, \dots, n\}$. In the former and in the latter case the probabilities in the boundary states should be assigned in addition. The random walk model along an infinite straight-line is depicted in Fig. 1.

Clearly,

$$X_{n+1} = \begin{cases} X_n + 1 & \text{with probability } p \\ X_n - 1 & \text{with probability } q \\ X_n & \text{with probability } 1 - p - q \end{cases}$$

and,

$$\mathcal{P}(X_{n+1} = j | X_n = i) = \begin{cases} p & \text{if } j = i + 1 \\ q & \text{if } j = i - 1 \\ 1 - p - q & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$$

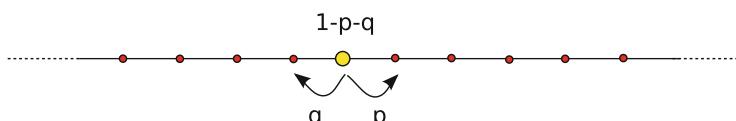


Fig. 1 Random walk along a discrete straight line

so that

$$P = \begin{bmatrix} \ddots & \ddots & \ddots & & \\ q & 1-p-q & p & & \\ & q & 1-p-q & p & \\ & & q & 1-p-q & p \\ & & & \ddots & \ddots & \ddots \end{bmatrix},$$

where, once again, blanks denote null entries. In this case, the transition matrix is bi-infinite (the set of states is \mathbb{N}), it is tridiagonal (the particle can move of at most one step to the right or to the left), it is Toeplitz (the probability to move right or left is independent of the state, that is, the probability is shift invariant).

If $E = \mathbb{Z}^+$, assuming \hat{p} the probability to move right from state 1 and $1 - \hat{p}$ the probability to remain in state 1, the transition matrix takes the form

$$P = \begin{bmatrix} 1-\hat{p} & \hat{p} & & & \\ q & 1-q-p & p & & \\ & q & 1-q-p & p & \\ & & \ddots & \ddots & \ddots \end{bmatrix}.$$

Observe that P is semi-infinite, tridiagonal and Toeplitz except for the entries in position (1, 1) and (1, 2) due to the boundary condition.

In the case where $E = \{1, 2, \dots, n\}$ one obtains the transition matrix

$$P = \begin{bmatrix} 1-\hat{p} & \hat{p} & & & & \\ q & 1-q-p & p & & & \\ & q & 1-q-p & p & & \\ & & \ddots & \ddots & \ddots & \\ & & & q & 1-q-p & p \\ & & & & \hat{q} & 1-\hat{q} \end{bmatrix},$$

where we have assumed that the particle can move left from state n with probability \hat{q} and stay in the state n with probability $1 - \hat{q}$. Once again, we have a tridiagonal almost Toeplitz $n \times n$ matrix.

More general models can be introduced, say, by assuming that the particle can move to the right of k unit steps with probability p_k , where k is any positive integer, and to the left of only one step with probability p_{-1} , where $p_i \geq 0$ and $\sum_{i=-1}^{\infty} p_i = 1$. In the model related to a half line, where the leftmost state has coordinate 0, we have to assign the probabilities that the system moves from state 0 to state $j \geq 0$. Let us denote \hat{p}_i , $i = 1, 2, \dots$, these probabilities so that $\sum_{i=0}^{\infty} \hat{p}_i = 1$. Then the set

of states is $E = \mathbb{N}$, and we have the following semi-infinite matrix

$$P = \begin{bmatrix} \hat{p}_0 & \hat{p}_1 & \hat{p}_2 & \hat{p}_3 & \hat{p}_4 & \dots \\ p_{-1} & p_0 & p_1 & p_2 & p_3 & \ddots \\ & p_{-1} & p_0 & p_1 & p_2 & \ddots \\ & & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

which is Toeplitz, except for the entries in the first row, and has null entries in row i and column j if $i > j + 1$. A matrix with the latter property is said to be in Hessenberg form.

But we may create even more general models, for instance by considering a bidimensional random walk where the particle can move in the plane. At each instant it can move right, left, up, down, up-right, up-left, down-right, down-left, with assigned probabilities $p_i^{(j)}$, $i, j = -1, 0, 1$, such that $\sum_{i,j} p_i^{(j)} = 1$, see Fig. 2.

Consider the semi-infinite case where $E = \mathbb{N} \times \mathbb{N}$. Ordering the coordinates row-wise as

$$(0, 0), (1, 0), (2, 0), \dots, (0, 1), (1, 1), (2, 1), \dots, (0, 2), (1, 2), (2, 2), \dots$$

one finds that the matrix P has the following block structure

$$P = \begin{bmatrix} \widehat{A}_0 & \widehat{A}_1 & & & \\ A_{-1} & A_0 & A_1 & & \\ & A_{-1} & A_0 & A_1 & \\ & & \ddots & \ddots & \ddots \end{bmatrix},$$

$$A_i = \begin{bmatrix} \widetilde{p}_0^{(i)} & \widetilde{p}_1^{(i)} & & & \\ p_{-1}^{(i)} & p_0^{(i)} & p_1^{(i)} & & \\ & p_{-1}^{(i)} & p_0^{(i)} & p_1^{(i)} & \\ & & \ddots & \ddots & \ddots \end{bmatrix}, \quad \widehat{A}_0 = \begin{bmatrix} \widetilde{\widehat{p}}_0^{(0)} & \widetilde{\widehat{p}}_1^{(0)} & & & \\ \widehat{p}_{-1}^{(0)} & \widehat{p}_0^{(0)} & \widehat{p}_1^{(0)} & & \\ & \widehat{p}_{-1}^{(0)} & \widehat{p}_0^{(0)} & \widehat{p}_1^{(0)} & \\ & & \ddots & \ddots & \ddots \end{bmatrix},$$

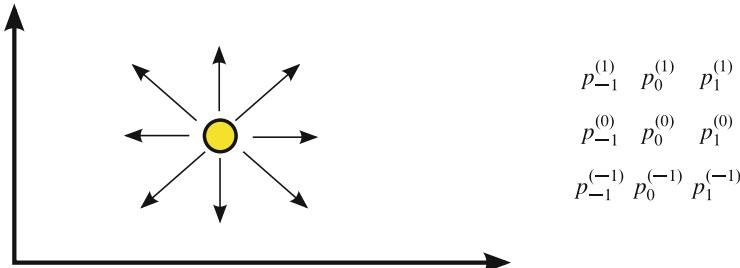


Fig. 2 Bidimensional random walk

where we have denoted with $\hat{p}_j^{(0)}, \hat{p}_0^{(i)}, \hat{p}_1^{(i)}$ and $\tilde{\hat{p}}_0^{(0)}, \tilde{\hat{p}}_1^{(0)}$ the probabilities of changing state when the particle occupies a boundary position. Thus we get a block tridiagonal almost block Toeplitz matrix with tridiagonal almost Toeplitz blocks. Indeed, according to the choice of the set of states, the block matrix can be bi-infinite, semi-infinite, or finite. Similarly, the blocks can be bi-infinite, semi-infinite or finite.

For $E = \mathbb{Z}^d$ one obtains a *multilevel structure with d levels*, that is, a block Toeplitz, block tridiagonal matrix where the blocks have a multilevel structure with $d - 1$ levels.

Another interesting case is the *shortest queue model* which analyzes, for instance, the probability of encountering queues at the toll gate of a highway. In this model, we assume to have m servers with m queues; at each time step, each server serves one customer; α new customers arrive with probability q_α ; each customer joins the shortest queue. Finally, denote X_n the overall number of customers in the lines.

Clearly we have

$$X_{n+1} = \begin{cases} X_n + \alpha - m & \text{if } X_n + \alpha - m \geq 0, \\ 0 & \text{if } X_n + \alpha - m < 0, \end{cases}$$

and

$$p_{i,j} = \begin{cases} q_0 + q_1 + \cdots + q_{m-i} & \text{if } j = 0, \quad 0 \leq i \leq m-1, \\ q_{j-i+m} & \text{if } j - i + m \geq 0, \\ 0 & \text{if } j - i + m < 0. \end{cases}$$

In fact, observe that if $i < m$ then the transition $i \rightarrow 0$ occurs if the number of arrivals is at most $m-i$; moreover, the transition $i \rightarrow j$ occurs if there are $j-i+m \geq 0$ arrivals; finally the transition $i \rightarrow j$ is impossible if $j < i-m$ since at most m customers are served in a unit of time.

Therefore the transition matrix of the model is

$$P = \begin{bmatrix} b_0 & q_{m+1} & q_{m+2} & q_{m+3} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{m-1} & q_2 & q_3 & q_4 & \ddots & \ddots \\ q_0 & q_1 & q_2 & q_3 & \ddots & \ddots \\ q_0 & q_1 & q_2 & q_3 & \ddots & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix},$$

where $b_i = q_0 + q_1 + \cdots + q_{m-i}$, $0 \leq i \leq m-1$. This matrix is in generalized Hessenberg form, that is $p_{i,j} = 0$ for $i-j > m+1$, and is Toeplitz except for the first m entries in the first column.

By reblocking P into $m \times m$ blocks one can view P as a block (almost) Toeplitz matrix in block Hessenberg form. The block Toeplitz structure is lost only in the block in position $(1, 1)$ due to the boundary conditions.

This block Hessenberg almost block Toeplitz structure is shared by a wide class of models which, in the Kendall notation [52] for classifying queueing models, are called M/G/1-type Markov chains.

In an M/G/1-type Markov chain, the set of states, given by $E = \mathbb{N} \times \{1, 2, \dots, m\}$, is bidimensional so that the random variable X_n is given by a pair: $X_n = (\psi_n, \varphi_n) \in E$, where ψ_n is called *level*, and φ_n *phase*. Moreover, $\mathcal{P}(X_{n+1} = (i', j') | X_n = (i, j)) = (A_{i'-i})_{j,j'}$, for $i' - i \geq -1$, $1 \leq j, j' \leq m$, $\mathcal{P}(X_{n+1} = (i', j') | X_n = (0, j)) = (B_{i'})_{j,j'}$, for $i' - i \geq -1$.

Observe that the probability to switch from level i to level i' depends on $i' - i$ and that the transition matrix is given by

$$P = \begin{bmatrix} B_0 & B_1 & B_2 & B_3 & \dots \\ A_{-1} & A_0 & A_1 & A_2 & A_3 & \dots \\ A_{-1} & A_0 & A_1 & A_2 & A_3 & \ddots \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix},$$

that is, an almost block Toeplitz, block Hessenberg matrix.

Another important class of models is given by G/M/1-Type Markov chains. Here, the set of states is the same and the transition matrix is still almost block Toeplitz in block lower Hessenberg form. More precisely,

$$P = \begin{bmatrix} B_0 & A_1 & & & \\ B_{-1} & A_0 & A_1 & & \\ B_{-2} & A_{-1} & A_0 & A_1 & \\ B_{-3} & A_{-2} & A_{-1} & A_0 & A_1 \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

A Markov chain which is at the same time M/G/1-type and G/M/1-type is called Quasi-Birth-Death Markov chain. Here the level can move up or down only by one step while the phase can move anywhere. Thus, the transition matrix is block tridiagonal, block Toeplitz except for the boundary block:

$$P = \begin{bmatrix} B_0 & A_1 & & & \\ A_{-1} & A_0 & A_1 & & \\ & A_{-1} & A_0 & A_1 & \\ & & A_{-1} & A_0 & A_1 \\ & & & \ddots & \ddots \end{bmatrix}.$$

Similar interesting structures are originated by continuous time Markov chains where the time t ranges in a continuous set T , say $T = \mathbb{R}$. In this case the transition matrix is replaced by the *rate* matrix. An important example of this chain is given by the *Tandem Jackson model* where there are two queues in tandem; customers arrive at the first queue according to a Poisson process with rate λ , then they are served with an exponential service time with parameter μ_1 . On leaving the first queue, customers enter the second queue and are served with an exponential service time with parameter μ_2 . See Fig. 3.

The model is described by a Markov chain where the set of states is formed by the pairs (α, β) , where α is the number of customers in the first queue, β is the number of customers in the second queue.

The rate matrix is

$$P = \begin{bmatrix} \tilde{A}_0 & A_1 \\ A_{-1} & A_0 & A_1 \\ & \ddots & \ddots & \ddots \end{bmatrix}$$

where

$$A_{-1} = \begin{bmatrix} 0 & & & \\ \mu_1 & 0 & & \\ & \mu_1 & 0 & \\ & & \ddots & \ddots \end{bmatrix}, \quad A_0 = \begin{bmatrix} 1 - \lambda - \mu_2 & \lambda & & \\ & 1 - \lambda - \mu_1 - \mu_2 & \lambda & \\ & & \ddots & \ddots \end{bmatrix},$$

$$A_1 = \mu_2 I, \quad \tilde{A}_0 = \begin{bmatrix} 1 - \lambda & \lambda & & \\ & 1 - \lambda - \mu_1 & \lambda & \\ & & \ddots & \ddots \end{bmatrix}.$$

There are other situations, modeled by discrete time Markov chains, where the transition to lower levels is limited by a positive constant N . That is, the Toeplitz

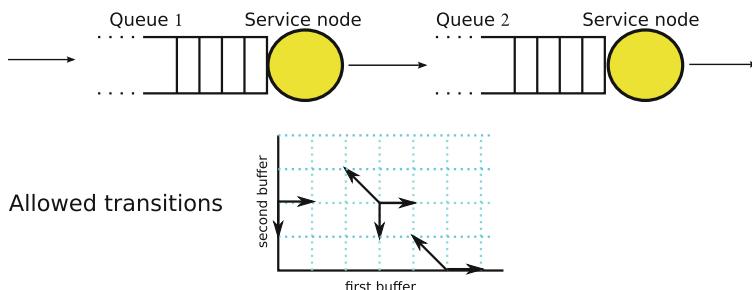


Fig. 3 The Tandem Jackson model

part of the matrix P has the generalized block Hessenberg form

$$\begin{bmatrix} A_0 & A_1 & A_2 & A_3 & \dots & \dots & \dots \\ A_{-1} & A_0 & A_1 & A_2 & A_3 & \ddots & \ddots & \ddots \\ \vdots & A_{-1} & A_0 & A_1 & A_2 & A_3 & \ddots & \ddots \\ A_{-N} & \ddots & A_{-1} & A_0 & A_1 & A_2 & A_3 & \ddots \\ A_{-N} & \ddots & A_{-1} & A_0 & A_1 & A_2 & \ddots & \ddots \\ \ddots & \ddots \end{bmatrix}.$$

These models are said *Non-Skip-Free* Markov chains. Reblocking the above matrix into $N \times N$ blocks yields a block Toeplitz matrix in block Hessenberg form where the blocks are block Toeplitz matrices. With this trick, Non-Skip-Free Markov chains can be reduced to the M/G/1-type model.

Many problems of the Real World are modeled by structured Markov chains. A meaningful instance is given by the IEEE 802.10 wireless protocol: for queuing and communication systems. This protocol is adopted by the wi-fi devices commonly used in the home LAN networks. The model is governed by a finite Markov chain of the M/G/1-type where the transition matrix is given by

$$P = \begin{bmatrix} \widehat{A}_0 & \widehat{A}_1 & \dots & \dots & \widehat{A}_{n-1} & \widehat{A}_n \\ A_{-1} & A_0 & A_1 & \dots & A_{n-2} & \widetilde{A}_{n-1} \\ & A_{-1} & A_0 & A_1 & \vdots & \vdots \\ & \ddots & \ddots & \ddots & \vdots & \vdots \\ & A_{-1} & A_0 & \widetilde{A}_1 & & \\ & & A_{-1} & \widetilde{A}_0 & & \end{bmatrix}.$$

Other situations are encountered in risk and insurance problems, state-dependent susceptible-infected-susceptible epidemic models, inventory systems and more [4].

2.3 Other Structures

In some cases, the blocks A_i in the QBD, M/G/1-type or G/M/1-type processes, have a tensor structure, that is they can be written as Kronecker products of matrices, in other cases the blocks A_i for $i \neq 0$ have low rank [67]. In other situations, transition matrices with a recursive structure are encountered. This is the case of *tree-like stochastic processes* which are defined by bivariate Markov chains over a d -ary tree [13].

The set of states is given by the pairs $(J; i)$, where $J = (j_1, \dots, j_\ell)$, with $1 \leq j_1, \dots, j_\ell \leq d$, while $1 \leq i \leq m$ and ℓ ranges in \mathbb{N} .

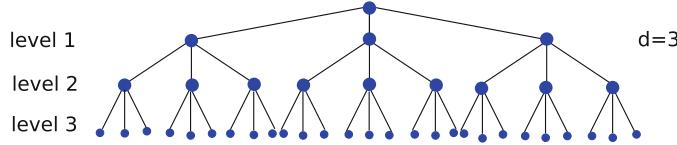


Fig. 4 Tree-like stochastic process

The ℓ -tuple $J = (j_1, \dots, j_\ell)$ denotes the generic node at the level ℓ of the tree, where j_i is the number of the edge at level i of the path joining the root of the tree to the generic node. See Fig. 4.

The allowed transitions in this Markov chain are

- within a node $(J; i) \rightarrow (J; i')$ with probability $(B_j)_{i,i'}$;
- within the root node $i \rightarrow i'$ with probability $(B_0)_{i,i'}$;
- between a node and one of its children $(J; i) \rightarrow ([J, k]; i')$ with probability $(A_k)_{i,i'}$;
- between a node and its parent $([J, k]; i) \rightarrow (J; i')$ with probability $(D_k)_{i,i'}$.

Over an infinite tree, with a lexicographical ordering of the states according to the leftmost integer of the node J , the states are given by $(\ell + 1)$ -tuples $(j_1, \dots, j_\ell; i)$, where $1 \leq j_1, \dots, j_\ell \leq d$, $1 \leq i \leq m$ and ℓ ranges in \mathbb{N} . The Markov chain is governed by the transition matrix

$$P = \begin{bmatrix} C_0 & \Lambda_1 & \Lambda_2 & \dots & \Lambda_d \\ V_1 & W & 0 & \dots & 0 \\ V_2 & 0 & W & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ V_d & 0 & \dots & 0 & W \end{bmatrix},$$

where $C_0 = B_0 - I$, $\Lambda_i = [A_i \ 0 \ 0 \ \dots]$, $V_i^T = [D_i^T \ 0 \ 0 \ \dots]$, we assume $B_1 = \dots = B_d =: B$, $C = I - B$, and the infinite matrix W is recursively defined by

$$W = \begin{bmatrix} C & \Lambda_1 & \Lambda_2 & \dots & \Lambda_d \\ V_1 & W & 0 & \dots & 0 \\ V_2 & 0 & W & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ V_d & 0 & \dots & 0 & W \end{bmatrix}.$$

This kind of Markov chains model single server queues with LIFO service discipline, medium access control protocol with an underlying stack structure [54].

Another interesting structure comes from *Markovian binary trees* and from the computational side involves the solution of vector equations.

Markovian binary trees (MBT) are a particular family of *branching processes* used to model the growth of populations and networking systems. They are characterized by the following laws [57].

1. At each instant, a finite number of entities, called “individuals”, exist.
2. Each individual can be in any one of N different states (say, age classes, or difference features in a population).
3. Each individual evolves independently from the others. Depending on its state i , it has a fixed probability $b_{i,j,k}$ of being replaced by two new individuals (“children”) in states j and k respectively, and a fixed probability a_i of dying without producing any offspring.

The MBT is characterized by the vector $\mathbf{a} = (a_i) \in \mathbb{R}_+^N$ and by the tensor $B = (b_{i,j,k}) \in \mathbb{R}_+^{N \times N \times N}$.

One important issue related to MBTs is the computation of the extinction probability of the population, given by the minimal nonnegative solution $\mathbf{x} \in \mathbb{R}^N$ of the quadratic vector equation [7]

$$x_k = a_k + \mathbf{x}^T B_k \mathbf{x}, \quad B_k = (b_{i,j,k}),$$

where x_k is the probability that a colony starting from a single individual in state k becomes extinct in a finite time. A compatibility condition is that $\mathbf{e} = \mathbf{a} + B(\mathbf{e} \otimes \mathbf{e})$, for $\mathbf{e} = (1, \dots, 1)^T$, that is, the probabilities of all the possible events that may happen to an individual in state i sum to 1.

The quadratic vector equation can be rewritten as

$$\mathbf{x} = \mathbf{a} + \mathcal{B}(\mathbf{x} \otimes \mathbf{x}), \quad \mathcal{B} = \text{diag}(\text{vec}(B_1^T)^T, \dots, \text{vec}(B_N^T)^T).$$

Another computational problem related to Markov chains is the *Poisson problem*. Given a stochastic matrix P , irreducible, non-periodic, positive recurrent, and given a vector \mathbf{d} , determine all the solutions $\mathbf{x} = (x_1, x_2, \dots)$, z to the following system

$$(I - P)\mathbf{x} = \mathbf{d} - z\mathbf{e}, \quad \mathbf{e} = (1, 1 \dots)^T.$$

This problem is found in many places, for instance, in Markov reward processes, central limit theorem for Markov chains, perturbation analysis, heavy-traffic limit theory, variance constant analysis, asymptotic variance of single-birth process [5].

In the finite case, for $\boldsymbol{\pi}$ such that $\boldsymbol{\pi}^T(I - P) = 0$ one finds that $z = \boldsymbol{\pi}^T \mathbf{d}$ so that it remains to solve $(I - P)\mathbf{x} = \mathbf{c}$ with $\mathbf{c} = \mathbf{d} - z\mathbf{e}$, which computationally presents no particularly hard difficulty.

In the infinite case more complicated situations are encountered [56].

Recently [32], some interest has been addressed to the case where P is block tridiagonal almost block Toeplitz, that is a QBD Markov chain. In this situation, one

has to solve a block tridiagonal almost block Toeplitz system of the kind

$$\begin{bmatrix} B & A_1 & & \\ A_{-1} & A_0 & A_1 & \\ & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \end{bmatrix}.$$

Another interesting computational problem, encountered in the Erlangian approximation of Markovian fluid queues [6] still in the framework of continuous time Markov chains, is given by the computation of the matrix exponential

$$\exp(A) = \sum_{i=0}^{\infty} \frac{1}{i!} A^i,$$

where A is an $n \times n$ block triangular block Toeplitz matrix with $m \times m$ blocks; A is a generator, i.e., $A\mathbf{e} \leq 0$, $a_{i,i} \leq 0$, $a_{i,j} \geq 0$ for $i \neq j$ [31].

3 Fundamentals on Structured Matrices

In this section, we examine the main structures of interest encountered in queuing models. A special role is played by Toeplitz matrices; for this reason, we devote large part of the section to this important class of matrices.

Different aspects will be considered. Among these, the interplay of Toeplitz matrices and polynomials, the role of trigonometric algebras and FFT, the generalization to hidden structures like Toeplitz-like matrices and the use of displacement operators for their detection, asymptotic spectral properties and their application to preconditioning linear systems.

We also deal with rank-structured matrices by providing some basic properties.

The guiding idea is the analysis of structure properties with the goal of providing efficient algorithms for the solution of the related computational problems. The spirit of this section is to present the ideas at the basis of algorithms design rather than to give the algorithmic details of each single method. For the latter goal we refer the reader to the current available literature. At the end of this section we report some bibliographical notes which are far to be exhaustive, and provide some pointers to the classical and to the recent literature.

3.1 Classical Results on Toeplitz Matrices

Let us recall some classical results which give the flavor of the richness of Toeplitz matrices and show their relationship with other fields of mathematics different from

linear algebra. Here we consider a sequence of $n \times n$ matrices parametrized with the size n . Let \mathbb{F} be a field ($\mathbb{F} \in \{\mathbb{R}, \mathbb{C}\}$).

Given a bi-infinite sequence $\{a_i\}_{i \in \mathbb{Z}} \in \mathbb{F}^{\mathbb{Z}}$ and an integer n , we consider the $n \times n$ Toeplitz matrix $T_n = (t_{i,j})_{i,j=1,n}$ such that $t_{i,j} = a_{j-i}$. We can view T_n as the leading principal submatrix of the (semi) infinite Toeplitz matrix $T_{\infty} = (t_{i,j})_{i,j \in \mathbb{N}}$, $t_{i,j} = a_{j-i}$.

$$T_{\infty} = \begin{bmatrix} a_0 & a_1 & a_2 & \dots \\ a_{-1} & a_0 & a_1 & \ddots \\ a_{-2} & a_{-1} & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Formally, a semi-infinite Toeplitz matrix T_{∞} , defines a linear operator which associates $\mathbf{x} = (x_i)_{i \in \mathbb{N}}$ with $\mathbf{y} = T_{\infty}\mathbf{x}$ where $y_i = \sum_{j \in \mathbb{N}} a_{j-i}x_j$, $i \in \mathbb{N}$. The infinite summation can be unbounded, and one can wonder under which conditions on the matrix T_{∞} the implication $\mathbf{x} \in \ell^2(\mathbb{N}) \Rightarrow \mathbf{y} \in \ell^2(\mathbb{N})$ is satisfied, where we denote $\ell^2(\mathbb{N})$ the set of vectors $\mathbf{x} = (x_i)_{i \in \mathbb{N}}$ such that $\sum_{i \in \mathbb{N}} |x_i|^2$ is finite.

We have the following classical result by Otto Toeplitz.

Theorem 2 *The matrix T_{∞} defines a bounded linear operator in $\ell^2(\mathbb{N})$, $\mathbf{x} \rightarrow \mathbf{y} = T_{\infty}\mathbf{x}$, $y_i = \sum_{j \in \mathbb{N}} a_{j-i}x_j$ if and only if a_i are the Fourier coefficients of a function $a(z) \in L^{\infty}(\mathbb{T})$, $\mathbb{T} = \{z \in \mathbb{C} : |z| = 1\}$*

$$a(z) = \sum_{n=-\infty}^{+\infty} a_n z^n, \quad a_n = \frac{1}{2\pi} \int_0^{2\pi} a(e^{i\theta}) e^{-in\theta} d\theta, \quad i^2 = -1.$$

In this case the operator norm of T_{∞} defined by $\|T_{\infty}\| := \sup_{\|\mathbf{x}\|_2=1} \|T_{\infty}\mathbf{x}\|_2$, where $\|\mathbf{x}\|_2 = (\sum_{i \in \mathbb{N}} |x_i|^2)^{\frac{1}{2}}$, is such that

$$\|T_{\infty}\| = \text{ess sup}_{z \in \mathbb{T}} |a(z)|.$$

Here, $\text{ess sup } a(z)$ denotes the essential sup of $a(z)$, that is the sup of $a(z)$ up to a set of measure zero. The function $a(z)$ is called *symbol* associated with T_{∞} , and, as we will see in Sect. 3.6, its properties are strictly related to the properties of the eigenvalues of T_n .

Observe that in the case of a Laurent polynomial $a(z) = \sum_{i=-k}^k a_i z^i$ the matrix T_{∞} is a banded Toeplitz matrix which defines a bounded linear operator. Moreover, since $a(z) = a(\cos \theta + i \sin \theta)$ for $\theta \in [0, 2\pi]$, the function $a(z)$, as function of θ is continuous over a compact set so that the ess sup turns into a max and $\|T_{\infty}\| = \max_{z \in \mathbb{T}} |a(z)|$.

Similar definitions and properties can be given for a block Toeplitz matrix. Given a bi-infinite sequence $\{A_i\}_{i \in \mathbb{Z}}$, $A_i \in \mathbb{F}^{m \times m}$ and an integer n , consider the $n \times n$ block

Toeplitz matrix $T_n = (t_{i,j})_{i,j=1,n}$ with $m \times m$ blocks $t_{i,j} = A_{j-i}$. Observe that T_n is the leading principal submatrix of the (semi) infinite block Toeplitz matrix $T_\infty = (t_{i,j})_{i,j \in \mathbb{N}}$, $t_{i,j} = A_{j-i}$

$$T_\infty = \begin{bmatrix} A_0 & A_1 & A_2 & \dots \\ A_{-1} & A_0 & A_1 & \ddots \\ A_{-2} & A_{-1} & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

In this case, we may associate the matrix valued function $a(z) = \sum_{n=-\infty}^{+\infty} z^n A_i$, that is the symbol, with the sequence T_n . A block version of Theorem 2 holds true.

Theorem 3 *The infinite block Toeplitz matrix T_∞ defines a bounded linear operator in $\ell^2(\mathbb{N})$ if and only if the $m \times m$ blocks $A_k = (a_{i,j}^{(k)})$ are the Fourier coefficients of a matrix-valued function $A(z) : \mathbb{T} \rightarrow \mathbb{C}^{m \times m}$, $A(z) = \sum_{k=-\infty}^{+\infty} z^k A_k = (a_{i,j}(z))_{i,j=1,m}$ such that $a_{i,j}(z) \in L^\infty(\mathbb{T})$.*

If the blocks A_k are Toeplitz themselves, that is $A_k = (a_{k,j-i})_{i,j}$, we have a block Toeplitz matrix with Toeplitz blocks. The entries $a_{h,k}$ for $h, k \in \mathbb{Z}$ define the bivariate symbol $a(z, w) : \mathbb{T} \times \mathbb{T} \rightarrow \mathbb{C}$, $a(z, w) = \sum_{i,j=-\infty}^{+\infty} a_{i,j} z^i w^j$. We may associate with the symbol $a(z, w)$ the infinite block Toeplitz matrix T_∞ with infinite Toeplitz blocks as well as the bivariate sequence $T_{n,m}$ of $n \times n$ block Toeplitz matrices with $m \times m$ Toeplitz blocks $A_k = (a_{k,j-i})$.

Also in this case we can extend Theorem 2 on the boundedness of the linear operator.

Theorem 4 *The infinite block Toeplitz matrix T_∞ with infinite Toeplitz blocks associated with the symbol $a(z, w)$ defines a bounded linear operator in $\ell^2(\mathbb{N}) \times \ell^2(\mathbb{N})$ if and only if $a(z) \in L^\infty(\mathbb{T} \times \mathbb{T})$.*

Observe that given the symbol $a(z, w)$, for any pair of integers n, m we may construct an $n \times n$ Toeplitz matrix $T_{m,n} = (A_{j-i})_{i,j=1,n}$ with $m \times m$ Toeplitz blocks $A_k = (a_{k,j-i})_{i,j=1,m}$.

Similarly, we may associate a symbol with multilevel Toeplitz matrices. A function $a : \mathbb{T}^d \rightarrow \mathbb{C}$ having the Fourier expansion

$$a(z_1, z_2, \dots, z_d) = \sum_{i_1, \dots, i_d=-\infty}^{+\infty} a_{i_1, i_2, \dots, i_d} z_{i_1}^{i_1} z_{i_2}^{i_2} \cdots z_{i_d}^{i_d},$$

defines a d -multilevel Toeplitz matrix: that is a block Toeplitz matrix with blocks that are themselves $(d-1)$ -multilevel Toeplitz matrices.

3.2 Toeplitz Matrices, Polynomials and Power Series

Here we point out the interplay between Toeplitz matrices polynomials and power series. Let us start with the problem of polynomial multiplication.

Consider polynomials $a(x) = \sum_{i=0}^n a_i x^i$, $b(x) = \sum_{i=0}^m b_i x^i$, and define their product $c(x) := a(x)b(x) = \sum_{i=0}^{m+n} c_i x^i$. A direct inspection shows that $c_0 = a_0 b_0$, $c_1 = a_0 b_1 + a_1 b_0$, $c_2 = a_0 b_2 + a_1 b_1 + a_2 b_0$, and so on. In matrix notation, we have

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ \vdots \\ c_{m+n} \end{bmatrix} = \begin{bmatrix} a_0 & & & & \\ a_1 & a_0 & & & \\ \vdots & \ddots & \ddots & & \\ a_n & \ddots & \ddots & a_0 & \\ & \ddots & \ddots & a_1 & \\ & & \ddots & \ddots & \vdots \\ & & & a_n & \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix}.$$

That is, polynomial multiplication can be rewritten as the product of a banded (rectangular) Toeplitz matrix and a vector. Conversely, let A be an $n \times n$ Toeplitz matrix and \mathbf{b} an n -vector, consider the matrix-vector product $\mathbf{c} = A\mathbf{b}$ and rewrite it in the following form

$$\begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \vdots \\ \hat{c}_{n-1} \\ \hline c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ \hline \tilde{c}_1 \\ \tilde{c}_2 \\ \vdots \\ \tilde{c}_{n-1} \end{bmatrix} = \begin{bmatrix} a_{n-1} & & & & \\ a_{n-2} & a_{n-1} & & & \\ \vdots & \ddots & \ddots & & \\ a_1 & \dots & a_{n-2} & a_{n-1} & \\ \hline a_0 & a_1 & \dots & a_{n-2} & a_{n-1} \\ a_{-1} & a_0 & a_1 & \dots & a_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{-n+2} & \dots & a_{-1} & a_0 & a_1 \\ \hline a_{-n+1} & a_{-n+2} & \dots & a_{-1} & a_0 \\ & a_{-n+1} & a_{-n+2} & \dots & a_1 \\ & & \ddots & \ddots & \vdots \\ & & & & a_{-n+1} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

This way, the Toeplitz-vector product can be viewed as part of the product of a polynomial of degree at most $2n - 1$ and a polynomial of degree at most $n - 1$. Observe that the result is a polynomial of degree at most $3n - 2$ whose coefficients can be computed by means of an evaluation-interpolation scheme at $3n - 1$ points.

More precisely we have the following algorithm for computing the coefficients of the product of two polynomials given the coefficients of the factors.

1. Choose $N \geq 3n - 1$ different numbers x_1, \dots, x_N ;
2. evaluate $\alpha_i = a(x_i)$ and $\beta_i = b(x_i)$, for $i = 1, \dots, N$;
3. compute $\gamma_i = \alpha_i \beta_i$, $i = 1, \dots, N$;
4. interpolate $c(x_i) = \gamma_i$ and compute the coefficients of $c(x)$.

If the knots x_1, \dots, x_N are the N -th roots of 1 then the evaluation and the interpolation steps can be executed by means of FFT in time $O(N \log N)$ so that the product of an $n \times n$ Toeplitz matrix and a vector can be computed by means of three FFTs of length $N \geq 3n - 1$.

A similar analysis can be carried out for polynomial division. Let $n \geq m$, $a(x) = \sum_{i=0}^n a_i x^i$, $b(x) = \sum_{i=0}^m b_i x^i$, $b_m \neq 0$ be polynomials and $q(x)$, $r(x)$ quotient and remainder, respectively, of the division of $a(x)$ by $b(x)$. That is,

$$a(x) = b(x)q(x) + r(x), \quad \deg r(x) < m.$$

Using the matrix representation of polynomial product yields

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \\ \vdots \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} b_0 & & & & & & \\ b_1 & b_0 & & & & & \\ \vdots & \ddots & \ddots & & & & \\ b_m & \ddots & \ddots & b_0 & & & \\ & \ddots & \ddots & b_1 & & & \\ & & \ddots & \ddots & & & \\ & & & b_m & & & \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-m} \end{bmatrix} + \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{m-1} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Observe that the last $n - m + 1$ equations form a triangular Toeplitz system.

$$\begin{bmatrix} b_m & b_{m-1} & \dots & b_{2m-n} \\ b_m & \ddots & & \vdots \\ \ddots & b_{m-1} & & \\ & b_m & & \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-m} \end{bmatrix} = \begin{bmatrix} a_m \\ a_{m+1} \\ \vdots \\ a_n \end{bmatrix}.$$

Its solution provides the coefficients of the quotient. The remainder can be computed as a difference.

$$\begin{bmatrix} r_0 \\ \vdots \\ r_{m-1} \end{bmatrix} = \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix} - \begin{bmatrix} b_0 & & & & \\ \vdots & \ddots & & & \\ b_{m-1} & \dots & b_0 \end{bmatrix} \begin{bmatrix} q_0 \\ \vdots \\ q_{n-m} \end{bmatrix}.$$

Algorithms for solving triangular Toeplitz systems provide methods for computing quotient and remainder of polynomial division.

Matrix versions of other polynomial computations can be similarly obtained. As an example, consider the greatest common divisor (gcd) of two polynomials. If $g(x) = \text{gcd}(a(x), b(x))$, $\deg(g(x)) = k$, $\deg(a(x)) = n$, $\deg(b(x)) = m$. Then there exist polynomials $r(x), s(x)$ of degree at most $m-k-1, n-k-1$, respectively, such that (Bézout identity)

$$g(x) = a(x)r(x) + b(x)s(x).$$

In matrix form one has the $(m+n-k) \times (m+n-2k)$ system

$$\left[\begin{array}{cc|cc|cc|c} a_0 & & b_0 & & r_0 & & g_0 \\ a_1 & a_0 & b_1 & b_0 & r_1 & & \vdots \\ \vdots & \ddots & \vdots & \ddots & \vdots & & g_k \\ a_n & \ddots & a_0 & b_m & \ddots & b_0 & r_{m-k-1} \\ & \ddots & a_1 & & \ddots & b_1 & s_0 \\ & & \ddots & \vdots & & \vdots & s_1 \\ & & a_n & & b_m & & \vdots \\ & & & & & & s_{n-k-1} \end{array} \right] = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

where the matrix of the system is known as Sylvester matrix.

The last $m+n-2k$ equations provide a linear system of the kind $S \begin{bmatrix} r \\ s \end{bmatrix} = [g_k, 0, \dots, 0]^T$, where S is the $(m+n-2k) \times (m+n-2k)$ submatrix of the Sylvester matrix formed by the last $m+n-2k$ rows.

As finite Toeplitz matrices are related to polynomials, similarly, infinite Toeplitz matrices are related to power series and infinite banded Toeplitz matrices are related to polynomials. For instance, let $a(x), b(x)$ be polynomials of degree n, m with coefficients a_i, b_j , respectively, define the Laurent polynomial

$$c(x) = a(x)b(x^{-1}) = \sum_{i=-m}^n c_i x^i.$$

Then the following infinite UL factorization holds

$$\left[\begin{array}{cccccc} c_0 & c_1 & \dots & c_n & & & \\ c_{-1} & c_0 & c_1 & \ddots & c_n & & \\ \vdots & c_{-1} & c_0 & c_1 & \ddots & c_n & \\ c_{-m} & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{array} \right] = \left[\begin{array}{cccccc} a_0 & \dots & a_n & & & & \\ a_0 & \ddots & a_n & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \ddots & \ddots & \ddots \end{array} \right] \left[\begin{array}{cccccc} b_0 & & & & & & \\ b_1 & b_0 & & & & & \\ \vdots & \ddots & \ddots & & & & \\ b_m & \ddots & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots \end{array} \right],$$

where an infinite banded Toeplitz matrix is decomposed as the product of an upper triangular Toeplitz and a lower triangular Toeplitz matrix.

If the zeros of $a(x)$ and $b(x)$ lie outside the unit disk, this factorization is called *Wiener-Hopf factorization*. This factorization is encountered in many applications.

Remark 1 In the Wiener-Hopf factorization it is required that $a(x), b(x) \neq 0$ for $|x| \leq 1$. This condition has a great importance. Observe that if $|\gamma| > 1$ and $a(x) = x - \gamma$ then

$$\frac{1}{a(x)} = \frac{1}{x - \gamma} = -\frac{1}{\gamma} \frac{1}{1 - \frac{x}{\gamma}} = -\frac{1}{\gamma} \sum_{i=0}^{\infty} \left(\frac{x}{\gamma}\right)^i$$

and the series $\sum_{i=0}^{\infty} 1/|\gamma|^i$ is convergent since $1/|\gamma| < 1$. This is not true if $|\gamma| \leq 1$. Moreover, in matrix form

$$\begin{bmatrix} -\gamma & 1 & 0 & \dots \\ -\gamma & 1 & 0 & \dots \\ \ddots & \ddots & \ddots & \ddots \end{bmatrix}^{-1} = -\gamma^{-1} \begin{bmatrix} 1 & \gamma^{-1} & \gamma^{-2} & \dots \\ 1 & \gamma^{-1} & \gamma^{-2} & \ddots \\ \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Observe that the above matrix has entries bounded in modulus by a constant, moreover the infinity norm of the matrix is finite. A similar property holds if all the zeros of $a(x)$ have modulus > 1 . Analogous remarks apply to the factor $b(z)$.

The Wiener-Hopf factorization can be defined for matrix-valued functions $C(x) = \sum_{i=-\infty}^{+\infty} C_i x^i$, $C_i \in \mathbb{C}^{m \times m}$, in the *Wiener class* \mathcal{W}_m , formed by all the functions such that $\sum_{i=-\infty}^{+\infty} |C_i|$ is finite. Here, $|A| = (|a_{ij}|)$ for $A = (a_{ij})$.

For $C(x) \in \mathcal{W}_m$ the Wiener-Hopf factorization exists in the form

$$C(x) = A(x)\text{diag}(x^{k_1}, \dots, x^{k_m})B(x^{-1}), \quad A(x) = \sum_{i=0}^{\infty} x^i A_i, \quad B(x) = \sum_{i=0}^{\infty} B_i x^i,$$

provided that $\det C(x) \neq 0$ for $|x| = 1$, see [22]. Here, $A(x), B(x) \in \mathcal{W}_m$ and $\det A(x), \det B(x)$ are nonzero in the open unit disk.

If the *partial indices* $k_i \in \mathbb{Z}$ are zero, the factorization takes the form $C(x) = A(x)B(x^{-1})$ and is said *canonical factorization*. Its matrix representation provides a block UL factorization of the infinite block Toeplitz matrix (C_{j-i}) :

$$\begin{bmatrix} C_0 & C_1 & \dots \\ C_{-1} & C_0 & C_1 & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} A_0 & A_1 & \dots \\ A_0 & A_1 & \ddots \\ \ddots & \ddots & \ddots \\ & & \ddots \end{bmatrix} \begin{bmatrix} B_0 \\ B_{-1} & B_0 \\ \vdots & B_{-1} & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Moreover, the condition $\det A(z), \det B(z) \neq 0$ for $|z| \leq 1$ provides the existence of $A(z)^{-1}, B(z)^{-1}$ in \mathcal{W}_m . Consequently, the two infinite matrices have a block Toeplitz inverse which has bounded infinity norm.

If the condition $\det A(z), \det B(z) \neq 0$ is satisfied for $|z| < 1$, for instance there may exist \hat{z} with $|\hat{z}| = 1$ such that $\det A(\hat{z}) = 0$, then the canonical factorization is said *weak canonical factorization*. In this case $A(z)$ or $B(z)$ may be not invertible in \mathcal{W}_m . For instance, the function $A(z) = (1 - z)I$ has inverse $A(z)^{-1} = \sum_{i=0}^{\infty} z^i I$ which does not belong to \mathcal{W}_m .

We will see that the weak canonical factorization is fundamental for computing the vector $\boldsymbol{\pi}$ for infinite Markov chains of the M/G/1-type, G/M/1-type and QBD encountered in queueing models.

3.3 Toeplitz Matrices, Trigonometric Algebras, and FFT

Toeplitz matrices are closely related to certain matrix algebras formed by all the matrices A which can be simultaneously diagonalized by a similarity transformation, that is, such that $S^{-1}AS$ is diagonal for any matrix of the class, where S is an assigned nonsingular matrix. A relevant interest is addressed to those algebras obtained by choosing as S a discrete trigonometric transform like a Sine transform, a Cosine transform or a Fourier transform.

Let ω_n be a primitive n th root of 1, that is, such that $\omega_n^n = 1$ and the set $\{1, \omega_n, \dots, \omega_n^{n-1}\}$ has cardinality n , say, $\omega_n = \cos \frac{2\pi}{n} + i \sin \frac{2\pi}{n}$, where i denotes the complex unit such that $i^2 = -1$. Define the $n \times n$ matrices $\Omega_n = (\omega_n^{ij})_{i,j=0,n-1}$, $F_n = \frac{1}{\sqrt{n}} \Omega_n$. One can easily verify that $F_n^* F_n = I$ that is, F_n is a unitary matrix.

For $\mathbf{x} \in \mathbb{C}^n$ define $\mathbf{y} = \text{DFT}(\mathbf{x}) = \frac{1}{n} \Omega_n^* \mathbf{x}$ the *Discrete Fourier Transform* (DFT) of \mathbf{x} , similarly define $\mathbf{x} = \text{IDFT}(\mathbf{y}) = \Omega_n \mathbf{y}$ the *Inverse Discrete Fourier Transform* (IDFT) of \mathbf{y} . Here Ω^* denotes the conjugate transposed of the matrix Ω .

Observe that $\text{cond}_2(F_n) = \|F_n\|_2 \|F_n^{-1}\|_2 = 1$, $\text{cond}_2(\Omega_n) = 1$. This shows that the DFT and IDFT are numerically well conditioned when the perturbation errors are measured in the 2-norm.

If n is an integer power of 2 then the IDFT of a vector can be computed with the cost of $\frac{3}{2}n \log_2 n$ arithmetic operations by means of the fast Fourier transform algorithm (FFT). It is well known that FFT is numerically stable in the 2-norm [47]. That is, if $\tilde{\mathbf{x}}$ is the value computed in floating point arithmetic with precision μ in place of $\mathbf{x} = \text{IDFT}(\mathbf{y})$ then

$$\|\mathbf{x} - \tilde{\mathbf{x}}\|_2 \leq \mu \gamma \|\mathbf{x}\|_2 \log_2 n,$$

for a moderate constant γ .

Norm-wise well conditioning of DFT and the norm-wise stability of FFT make this tool very effective for *most numerical computations*. Unfortunately, the norm-

wise stability of FFT *does not imply* the component-wise stability. That is, the inequality $|x_i - \tilde{x}_i| \leq \mu\gamma|x_i| \log_2 n$ is *not generally true* for all the components x_i .

3.3.1 Circulant Matrices

A matrix algebra strictly related to Toeplitz matrices and to FFT is the class of *circulant matrices*.

Given the row vector $[a_0, a_1, \dots, a_{n-1}]$, the $n \times n$ matrix

$$A = (a_{j-i \bmod n})_{i,j=1,n} = \begin{bmatrix} a_0 & a_1 & \dots & a_{n-1} \\ a_{n-1} & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ a_1 & \dots & a_{n-1} & a_0 \end{bmatrix}$$

is called the *circulant* matrix associated with $[a_0, a_1, \dots, a_{n-1}]$ and is denoted by $\text{circ}(a_0, a_1, \dots, a_{n-1})$.

If $a_i = A_i$ are $m \times m$ matrices we have a *block circulant matrix*.

Any circulant matrix A can be viewed as a polynomial with coefficients a_i in the unit circulant matrix S defined by its first row $(0, 1, 0, \dots, 0)$

$$A = \sum_{i=0}^{n-1} a_i S^i, \quad S = \begin{bmatrix} 0 & 1 & & & \\ \vdots & \ddots & \ddots & & \\ 0 & & \ddots & 1 & \\ 1 & 0 & \dots & 0 \end{bmatrix}.$$

Clearly, $S^n - I = 0$ so that circulant matrices form a matrix algebra isomorphic to the algebra of polynomials with the product modulo $x^n - 1$.

If A is a circulant matrix with first row r^T and first column c , then

$$A = \frac{1}{n} \mathcal{Q}_n^* \text{diag}(\mathbf{w}) \mathcal{Q}_n = F^* \text{Diag}(\mathbf{w}) F,$$

where $\mathbf{w} = \mathcal{Q}_n c = \mathcal{Q}_n^* r$. An immediate consequence of the latter property is that

$$Ax = \text{DFT}_n(\text{IDFT}_n(c) \odot \text{IDFT}_n(x)),$$

where “ \odot ” denotes the Hadamard, or component-wise product of vectors.

The product Ax of an $n \times n$ circulant matrix A and a vector x , as well as the product of two circulant matrices can be computed by means of two IDFTs and a DFT of length n in $O(n \log n)$ ops.

The inverse of a circulant matrix can be computed in $O(n \log n)$ ops since

$$A^{-1} = \frac{1}{n} \Omega_n^* \text{diag}(\mathbf{w}^{-1}) \Omega_n \quad \Rightarrow \quad A^{-1} \mathbf{e}_1 = \frac{1}{n} \Omega_n^* \mathbf{w}^{-1},$$

where $\mathbf{w}^{-1} := (\mathbf{w}_i^{-1})$, with $\mathbf{w} = (w_i)$. In fact, in order to compute the first column $A^{-1} \mathbf{e}_1$ of A^{-1} it is sufficient to compute $\mathbf{w} = \Omega_n \mathbf{c}$ by means of an IDFT, to compute \mathbf{w}^{-1} by performing the inversion of n numbers, and to compute the DFT of \mathbf{w}^{-1} .

One can easily verify that the inverse of a block circulant matrix can be computed by means of $2m^2$ IDFTs of length n and n inversions of $m \times m$ matrices for the cost of $O(m^2 n \log n + nm^3)$. Similarly, the product of two block circulant matrices can be computed by means of $2m^2$ IDFTs, m^2 DFTs of length n , and n multiplications of $m \times m$ matrices for the cost of $O(m^2 n \log n + nm^3)$.

3.3.2 *z*-Circulant Matrices

A generalization of the algebra of circulant matrices is given by the class of *z-circulant matrices*.

Given a scalar $z \neq 0$ and the row vector $[a_0, a_1, \dots, a_{n-1}]$, the $n \times n$ matrix

$$A = \begin{bmatrix} a_0 & a_1 & \dots & a_{n-1} \\ za_{n-1} & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ za_1 & \dots & za_{n-1} & a_0 \end{bmatrix}$$

is called the *z-circulant* matrix associated with $[a_0, a_1, \dots, a_{n-1}]$.

Denote by S_z the *z*-circulant matrix whose first row is $[0, 1, 0, \dots, 0]$. We have the following properties:

- Any *z*-circulant matrix can be viewed as a polynomial in S_z , that is, $A = \sum_{i=0}^{n-1} a_i S_z^i$;
- $S_z^n = z D_z S D_z^{-1}$, $D_z = \text{diag}(1, z, z^2, \dots, z^{n-1})$, where $S = S_1$ is the unit circulant matrix;
- if A is the z^n -circulant matrix with first row \mathbf{r}^T and first column \mathbf{c} then

$$A = \frac{1}{n} D_z \Omega_n^* \text{diag}(\mathbf{w}) \Omega_n D_z^{-1},$$

with $\mathbf{w} = \Omega_n^* D_z \mathbf{r} = \Omega_n D_z^{-1} \mathbf{c}$;

- multiplication of *z*-circulants costs two IDFTs, one DFT and a scaling;
- the inversion of a *z*-circulant matrix costs one IDFT, one DFT, n inversions and a scaling;
- the extension to block matrices trivially applies to *z*-circulant matrices.

3.3.3 Matrix Embedding

A general technique which is very effective in many situations is *matrix embedding*. Relying on this technique, we can view any Toeplitz matrix as a submatrix of a suitable circulant matrix. In fact, an $n \times n$ Toeplitz matrix $A = (t_{i,j})$, $t_{i,j} = a_{j-i}$, can be embedded into the $2n \times 2n$ circulant matrix B whose first row is $[a_0, a_1, \dots, a_{n-1}, *, a_{-n+1}, \dots, a_{-1}]$, where $*$ denotes any number. For instance, for $n = 3$ we have

$$B = \left[\begin{array}{ccc|cc} a_0 & a_1 & a_2 & * & a_{-2} & a_{-1} \\ a_{-1} & a_0 & a_1 & a_2 & * & a_{-2} \\ a_{-2} & a_{-1} & a_0 & a_1 & a_2 & * \\ \hline * & a_{-2} & a_{-1} & a_0 & a_1 & a_2 \\ a_2 & * & a_{-2} & a_{-1} & a_0 & a_1 \\ a_1 & a_2 & * & a_{-2} & a_{-1} & a_0 \end{array} \right]$$

which is circulant. More generally, an $n \times n$ Toeplitz matrix can be embedded into a $q \times q$ circulant matrix for any $q \geq 2n - 1$.

An immediate consequence of this fact is that the product $\mathbf{y} = A\mathbf{x}$ of an $n \times n$ Toeplitz matrix A and a vector \mathbf{x} can be computed in $O(n \log n)$ ops. In fact, we may write

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{w} \end{bmatrix} = B \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} = \begin{bmatrix} A & H \\ H & A \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} = \begin{bmatrix} A\mathbf{x} \\ H\mathbf{x} \end{bmatrix}.$$

This equation leads to the following algorithm for computing the product of a Toeplitz matrix and a vector.

- embed the Toeplitz matrix A into the circulant matrix $B = \begin{bmatrix} A & H \\ H & A \end{bmatrix}$;
- embed the vector \mathbf{x} into the vector $\mathbf{v} = \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix}$;
- compute the circulant-vector product $\mathbf{u} = B\mathbf{v}$;
- set $\mathbf{y} = (u_1, \dots, u_n)^T$.

The cost of this algorithm is three FFTs of order $2n$, that is $O(n \log n)$ ops. With respect to the algorithm based on the reduction to polynomial multiplication described in Sect. 3.2, this algorithm requires DFTs of lower length. Similarly, the product $\mathbf{y} = A\mathbf{x}$ of an $n \times n$ block Toeplitz matrix with $m \times m$ blocks and a vector $\mathbf{x} \in \mathbb{C}^{mn}$ can be computed in $O(m^2 n \log n)$ ops.

3.3.4 Triangular Toeplitz Matrices

Another interesting matrix algebra is the one formed by (lower or upper) triangular Toeplitz matrices.

Let $Z = (z_{i,j})_{i,j=1,n}$ be the $n \times n$ matrix

$$Z = \begin{bmatrix} 0 & & & 0 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & 1 & 0 \end{bmatrix}.$$

Clearly Z^k is the matrix with entries equal to 1 in the k th diagonal below the main diagonal, and with zero entries elsewhere, for $k = 1, \dots, n-1$, while $Z^k = 0$ for $k \geq n$. This way, given the polynomial $a(x) = \sum_{i=0}^{n-1} a_i x^i$, the matrix $a(Z) = \sum_{i=0}^{n-1} a_i Z^i$ is a lower triangular Toeplitz matrix defined by its first column $(a_0, a_1, \dots, a_{n-1})^T$

$$a(Z) = \begin{bmatrix} a_0 & & & 0 \\ a_1 & a_0 & & \\ \vdots & \ddots & \ddots & \\ a_{n-1} & \dots & a_1 & a_0 \end{bmatrix}.$$

Since $Z^n = 0$ while $Z^{n-1} \neq 0$, the set of lower triangular Toeplitz matrices forms an algebra isomorphic to the algebra of polynomials with the product modulo x^n . The fact that lower triangular Toeplitz matrices are closed under matrix multiplication, implies that they are closed also under inversion so that the inverse matrix T_n^{-1} of a nonsingular matrix T_n is still a lower triangular Toeplitz matrix defined by its first column \mathbf{v}_n . Clearly, the inverse matrix T_n^{-1} can be computed by solving the system $T_n \mathbf{v}_n = \mathbf{e}_1$.

An algorithm for computing \mathbf{v}_n can be easily designed. Let $n = 2h$, h be a positive integer, and partition T_n into $h \times h$ blocks

$$T_n = \left[\begin{array}{c|c} T_h & 0 \\ \hline W_h & T_h \end{array} \right],$$

where T_h , W_h are $h \times h$ Toeplitz matrices and T_h is lower triangular. By formal inversion we have

$$T_n^{-1} = \left[\begin{array}{c|c} T_h^{-1} & 0 \\ \hline -T_h^{-1} W_h T_h^{-1} & T_h^{-1} \end{array} \right].$$

Thus the first column \mathbf{v}_n of T_n^{-1} is given by

$$\mathbf{v}_n = T_n^{-1} \mathbf{e}_1 = \begin{bmatrix} \mathbf{v}_h \\ -T_h^{-1} W_h \mathbf{v}_h \end{bmatrix} = \begin{bmatrix} \mathbf{v}_h \\ -L(\mathbf{v}_h) W_h \mathbf{v}_h \end{bmatrix},$$

where $L(\mathbf{v}_h) = T_h^{-1}$ denotes the lower triangular Toeplitz matrix whose first column is \mathbf{v}_h .

The same relation holds if T_n is block triangular Toeplitz. In this case, the elements a_0, \dots, a_{n-1} are replaced by the $m \times m$ blocks A_0, \dots, A_{n-1} and \mathbf{v}_n denotes the first block column of T_n^{-1} .

From these arguments we may design the following algorithm for computing \mathbf{v}_n which is given in the block case. Given n , where for simplicity we assume $n = 2^k$, and $m \times m$ matrices A_0, \dots, A_{n-1} , the algorithm computes \mathbf{v}_n as follows.

1. Set $\mathbf{v}_1 = A_0^{-1} \mathbf{e}_1$;
2. for $i = 0, \dots, k-1$, given \mathbf{v}_h , $h = 2^i$:
 - (a) Compute the block Toeplitz matrix-vector products $\mathbf{w} = W_h \mathbf{v}_h$ and $\mathbf{u} = -L(\mathbf{v}_h) \mathbf{w}$.
 - (b) Set

$$\mathbf{v}_{2h} = \begin{bmatrix} \mathbf{v}_h \\ \mathbf{u} \end{bmatrix}.$$

In the scalar case, the algorithm costs $O(n \log n)$ ops, in the block case its cost is $O(m^2 n \log n + m^3 n)$ ops. In fact, inverting an $n \times n$ matrix is reduced to inverting an $\frac{n}{2} \times \frac{n}{2}$ matrix and to performing a Toeplitz-vector matrix product. Therefore, denoting $C(n)$ the complexity of $n \times n$ block triangular block Toeplitz matrix inversion, one has

$$C(n) = C(n/2) + O(mn^2 \log n + m^3)$$

which leads to $c(n) = O(m^2 n \log n + m^3 n)$.

3.3.5 z-Circulant and Triangular Toeplitz Matrices

If $\epsilon = |z|$ is “small” then a z-circulant approximates a triangular Toeplitz matrix:

$$\begin{bmatrix} a_0 & a_1 & \dots & a_{n-1} \\ za_{n-1} & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ za_1 & \dots & za_{n-1} & a_0 \end{bmatrix} \approx \begin{bmatrix} a_0 & a_1 & \dots & a_{n-1} \\ a_0 & \ddots & & \vdots \\ & \ddots & a_1 & \\ & & & a_0 \end{bmatrix}.$$

Inverting a z-circulant is less expensive than inverting a triangular Toeplitz (roughly by a factor of 10/3). The advantage is appreciated in a parallel model of computation, over multithreading architectures.

Numerical algorithms for approximating the inverse of (block) triangular block Toeplitz matrices can be easily designed. Here are their main features.

The total error that they generate is the sum of the approximation errors, proportional to ϵ and the algorithmic errors, due to the floating point arithmetic, which are proportional to ϵ^{-1} . The smaller ϵ the better the approximation, but the larger the rounding errors. A good compromise is to choose ϵ such that $\epsilon = u\epsilon^{-1}$, where u is the machine precision of the floating point arithmetic. This implies that the total error is $O(u^{1/2})$, that is, half digits of precision are lost in the floating point computation. Different strategies have been designed to overcome this drawback. A useful observation in this regard is that the approximation errors are polynomials in z . We assume to work over \mathbb{R} .

- (Interpolation) Since the approximation errors are polynomials in z , approximating twice the inverse with, say $z = \epsilon$ and $z = -\epsilon$ and taking the arithmetic mean of the results, the approximation errors become polynomials in ϵ^2 since the odd powers of ϵ cancel out. This allows to reduce the total error to $O(u^{2/3})$, i.e., only 1/3 of digits are lost.
- (Generalization) Compute k approximations of the inverse with values $z_i = \epsilon\omega_k^i$, $i = 0, \dots, k-1$; take the arithmetic mean of the results so that the power of ϵ which are not multiple of k cancel out and we are left with approximation errors of the order $O(\epsilon^k)$; this implies that the total errors can be reduced to $O(u^{k/(k+1)})$. Observe that since the approximation error is a polynomial of degree less than n , for $k = n$ the approximation error provided by this interpolation scheme is zero.
- (Higham trick [1]) Choose $z = i\epsilon$ then the approximation error affecting the real part of the computed approximation is $O(\epsilon^2)$. This implies that the total errors can be reduced to $O(u^{2/3})$.
- (Combination) Choose $z_1 = \epsilon(1+i)/\sqrt{2}$ and $z_2 = -z_1$; apply the algorithm with $z = z_1$ and $z = z_2$; take the arithmetic mean of the results. The approximation error on the real part turns out to be $O(\epsilon^4)$. The total error is $O(u^{4/5})$. Only 1/5 of digits are lost.
- (Replicating the computation) In general choosing as z_j the k th roots of the imaginary unit i and performing k inversions the error becomes $O(u^{2k/(2k+1)})$, i.e., only 1/2k of digits are lost.

3.3.6 Other Matrix Algebras

Besides, circulant, z-circulant and triangular Toeplitz matrices, there are many other matrix algebras related to Toeplitz matrices. In particular, relevant algebras can be associated with discrete trigonometric transforms as the Discrete Cosine Transforms, the Discrete Sine Transforms and the Discrete Hartley Transform.

If F is the matrix defining anyone of these transforms, then the associated matrix algebra is defined by

$$\{A = FDF^{-1} : D = \text{diag}(d_1, d_2, \dots, d_n), d_i \in \mathbb{F}\}.$$

In particular, the class τ introduced in [8], is associated with the sine transform defined by $F = \sqrt{\frac{2}{n+1}}(\sin \frac{\pi}{n+1}ij)$ which is an orthogonal matrix. This class is formed by symmetric real matrices of the kind $p(H)$, where $p(x)$ is a polynomial of degree at most $n - 1$ and $H = \text{trid}(1, 0, 1)$ is a tridiagonal symmetric Toeplitz matrix.

Algebras associated with cosine transforms—there exist eight different kinds of cosine transforms—have been analyzed in [50]; while the algebra associated with the Hartley transform where $F = \cos \frac{2\pi}{n}ij + \sin \frac{2\pi i}{n}ij$ has been introduced and analyzed in [10]. The Hartley algebra contains real symmetric circulant matrices. These kind of matrices find interesting applications as preconditioners of Toeplitz and Toeplitz-like systems.

3.4 Displacement Operators

The theory of displacement operators, introduced by Kailath in the 1980s, provides useful tools for analyzing structure and computational properties of Toeplitz-like matrices and their inverses. In particular, it yields a simple mean for detecting hidden structures of Toeplitz type and allows to design fast and super-fast algorithms for Toeplitz-like matrix inversion. For more details on displacement operators we refer to the review paper by Kailath and Sayed [51], and to the book [12].

Observe that multiplying a matrix A to the left by S_z moves up each row by one place, and replaces the last row with the first row of A multiplied by z . More precisely, if $B = S_z A$ then the k th row of B is the $(k + 1)$ st row of A for $k = 1, \dots, n - 1$, while the last row of B coincides with the first row of A multiplied by z . A similar effect has the multiplication to the right of A by S_z where the shift is operated on the columns and each column is moved to the right.

We can describe the transformation $A \rightarrow S_{z_1}A - AS_{z_2}$ with the following scheme

$$S_{z_1}A - AS_{z_2} = \begin{bmatrix} \uparrow \\ \end{bmatrix} - \begin{bmatrix} \rightarrow \\ \end{bmatrix}.$$

If the transformation is applied to a Toeplitz matrix, say $A = \begin{bmatrix} a & b & c & d \\ e & a & b & c \\ f & e & a & b \\ g & f & e & a \end{bmatrix}$, then one obtains

$$\begin{aligned} S_{z_1}A - AS_{z_2} &= \begin{bmatrix} e & a & b & c \\ f & e & a & b \\ g & f & e & a \\ z_1a & z_1b & z_1c & z_1d \end{bmatrix} - \begin{bmatrix} z_2d & a & b & c \\ z_2c & e & a & b \\ z_2b & f & e & a \\ z_2a & g & f & e \end{bmatrix} \\ &= \begin{bmatrix} * & & & \\ \vdots & 0 & & \\ & & & \\ * & \dots & * & \end{bmatrix} = e_n u^T + v e_1^T. \end{aligned}$$

That is, if A is an $n \times n$ Toeplitz matrix, then $S_{z_1}A - AS_{z_2}$ is a matrix of rank at most 2 having a special pattern.

Besides the operator $A \rightarrow S_{z_1}A - AS_{z_2}$, of Sylvester type, also the operator $A \rightarrow A - S_{z_1}AS_{z_2}^T$, of Stein type, performs a similar transformation when applied to a Toeplitz matrix. This kind of operators are called *displacement operators*.

If the eigenvalues of S_{z_1} are disjoint from those of S_{z_2} then the operator of Sylvester type is invertible. This holds if $z_1 \neq z_2$. If the eigenvalues of S_{z_1} are different from the reciprocal of those of S_{z_2} then the operator of Stein type is invertible. This holds if $z_1 z_2 \neq 1$.

As an example, consider $Z := S_0^T$ the matrix which generates the algebra of lower triangular Toeplitz matrices, and observe that if A is Toeplitz then the displacement operator of Sylvester type $\Delta(A) = AZ - ZA$ is such that

$$\begin{aligned} \Delta(A) &= \begin{bmatrix} \leftarrow \\ \downarrow \end{bmatrix} - \begin{bmatrix} \downarrow \\ \leftarrow \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \dots & a_{n-1} & 0 \\ & & & & -a_{n-1} \\ & & & & \vdots \\ & & & & -a_2 \\ & & & & -a_1 \end{bmatrix} = VW^T, \\ V &= \begin{bmatrix} 1 & 0 \\ 0 & a_{n-1} \\ \vdots & \vdots \\ 0 & a_1 \end{bmatrix}, \quad W = \begin{bmatrix} a_1 & 0 \\ \vdots & \vdots \\ a_{n-1} & 0 \\ 0 & -1 \end{bmatrix}. \end{aligned}$$

For any matrix A , any pair $V, W \in \mathbb{F}^{n \times k}$ such that $\Delta(A) = VW^T$ is called *displacement generator* of rank k of A , while k is called the *displacement rank* of A . In particular, Toeplitz matrices have displacement rank at most 2.

The following result provides a representation of any matrix A having a displacement generator V, W and first column \mathbf{a} .

Proposition 1 *Let $A \in \mathbb{F}^{n \times n}$, $\Delta(A) = AZ - ZA$, and let a be the first column of A . If $\Delta(A) = VW^T$ with $V, W \in \mathbb{F}^{n \times k}$, then*

$$A = L(\mathbf{a}) + \sum_{i=1}^k L(\mathbf{v}_i)L^T(Z\mathbf{w}_i), \quad L(\mathbf{a}) = \begin{bmatrix} a_1 \\ \vdots & \ddots \\ a_n & \dots & a_1 \end{bmatrix}.$$

The proof of this result can be easily outlined. Consider the case where $k = 1$. Rewrite the system $AZ - ZA = \mathbf{v}\mathbf{w}^T$ in vec form as

$$(Z^T \otimes I - I \otimes Z)\text{vec}(A) = \mathbf{w} \otimes \mathbf{v},$$

where $\text{vec}(A)$ is the n^2 -vector obtained by stacking the columns of A . Solve the block triangular system by substitution and get the representation of A .

In particular, any matrix can be written as a sum of products of lower triangular times upper triangular Toeplitz matrices. Equivalent representations can be given for Stein-type operators A matrix with displacement rank “much less” than n is called Toeplitz-like. The following result states that the inverse of a Toeplitz like matrix of displacement rank k is still Toeplitz like with displacement rank k . In particular the inverse of a Toeplitz matrix is Toeplitz like with displacement rank at most 2.

Proposition 2 *For $\Delta(A) = AZ - ZA$ it holds that $\Delta(AB) = A\Delta(B) + \Delta(A)B$ and, for $\det A \neq 0$, $\Delta(A^{-1}) = -A^{-1}\Delta(A)A^{-1}$. Therefore*

$$A^{-1} = L(A^{-1}\mathbf{e}_1) - \sum_{i=1}^k L(A^{-1}\mathbf{v}_i)L^T(ZA^{-T}\mathbf{w}_i)$$

for $\Delta(A) = VW^T$, where \mathbf{v}_i and \mathbf{w}_i are the i th columns of V and W , respectively.

If $A = (a_{j-i})$ is Toeplitz, since $\Delta(A) = VW^T$ where

$$V = \left[\begin{array}{c|c} 1 & 0 \\ 0 & a_{n-1} \\ \vdots & \vdots \\ 0 & a_1 \end{array} \right], \quad W = \left[\begin{array}{c|c} a_1 & 0 \\ \vdots & \vdots \\ a_{n-1} & 0 \\ 0 & -1 \end{array} \right],$$

the inverse of a Toeplitz matrix is Toeplitz-like and

$$\begin{aligned} A^{-1} &= L(A^{-1}\mathbf{e}_1) - L(A^{-1}\mathbf{e}_1)L^T(ZA^{-1}\mathbf{w}_1) + L(A^{-1}\mathbf{v}_2)L^T(ZA^{-1}\mathbf{e}_n) \\ &= L(A^{-1}\mathbf{e}_1)L^T(\mathbf{e}_1 - ZA^{-1}\mathbf{w}_1) + L(A^{-1}\mathbf{v}_2)L^T(ZA^{-1}\mathbf{e}_n). \end{aligned}$$

With a similar argument one can prove the *Gohberg-Semencul-Trench formula* [38]

$$T^{-1} = \frac{1}{x_1} (L(x)L^T(Jy) - L(Zy)L^T(ZJx)),$$

$$x = T^{-1}e_1, \quad y = T^{-1}e_n, \quad J = \begin{bmatrix} & & 1 \\ & \ddots & \\ 1 & & \end{bmatrix}.$$

As consequence of these properties, the first and the last column of the inverse of a Toeplitz matrix define all the entries, moreover, multiplying a vector by the inverse of a Toeplitz matrix costs only $O(n \log n)$ ops.

3.4.1 Other Operators: Cauchy-Like Matrices

Define $\Delta(X) = D_1X - XD_2$, $D_1 = \text{diag}(d_1^{(1)}, \dots, d_n^{(1)})$, $D_2 = \text{diag}(d_1^{(2)}, \dots, d_n^{(2)})$, where $d_i^{(1)} \neq d_j^{(2)}$ for $i \neq j$. It holds that

$$\Delta(A) = uv^T \Leftrightarrow a_{ij} = \frac{u_i v_j}{d_i^{(1)} - d_j^{(2)}}.$$

Similarly, given $n \times k$ matrices U, V , one finds that

$$\Delta(B) = UV^T \Leftrightarrow b_{ij} = \frac{\sum_{r=1}^k u_{i,r} v_{j,r}}{d_i^{(1)} - d_j^{(2)}}.$$

A is said *Cauchy* matrix, B is said *Cauchy-like* matrix.

A nice feature of Cauchy-like matrices is that their Schur complement is still a Cauchy-like matrix.

Consider the case $k = 1$: partition the Cauchy-like matrix C as

$$C = \left[\begin{array}{c|ccc} \frac{u_1 v_1}{d_1^{(1)} - d_1^{(2)}} & \frac{u_1 v_2}{d_1^{(1)} - d_2^{(2)}} & \cdots & \frac{u_1 v_n}{d_1^{(1)} - d_n^{(2)}} \\ \hline \frac{u_2 v_1}{d_2^{(1)} - d_1^{(2)}} & & & \\ \vdots & & & \\ \frac{u_n v_1}{d_n^{(1)} - d_1^{(2)}} & & & \end{array} \widehat{C} \right],$$

where \widehat{C} is still a Cauchy-like matrix. The Schur complement is given by

$$\widehat{C} - \left[\begin{array}{c} \frac{u_2 v_1}{d_2^{(1)} - d_1^{(2)}} \\ \vdots \\ \frac{u_n v_1}{d_n^{(1)} - d_1^{(2)}} \end{array} \right] \frac{d_1^{(1)} - d_1^{(2)}}{u_1 v_1} \left[\begin{array}{ccc} \frac{u_1 v_2}{d_1^{(1)} - d_2^{(2)}} & \cdots & \frac{u_1 v_n}{d_1^{(1)} - d_n^{(2)}} \end{array} \right].$$

The entries of the Schur complement can be written in the form

$$\frac{\widehat{u}_i \widehat{v}_j}{d_i^{(1)} - d_j^{(2)}}, \quad \widehat{u}_i = u_i \frac{d_1^{(1)} - d_i^{(1)}}{d_i^{(1)} - d_1^{(2)}}, \quad \widehat{v}_j = v_j \frac{d_j^{(2)} - d_1^{(2)}}{d_1^{(1)} - d_j^{(2)}}.$$

Moreover, the values \widehat{u}_i and \widehat{v}_j can be computed in $O(n)$ ops.

The computation of the Schur complement can be repeated until the LU decomposition of the matrix C is obtained. The algorithm obtained in this way is known as *Gohberg-Kailath-Olshevsky (GKO) algorithm* [39]. Its overall cost is $O(n^2)$ ops. There are variants, improvements and implementations which allow pivoting [3, 66].

3.5 Algorithms for Toeplitz Inversion

Consider $\Delta(A) = S_1 A - AS_{-1}$ where S_1 is the unit circulant matrix and S_{-1} is the unit (-1) -circulant matrix. We have observed that the matrix $\Delta(A)$ has rank at most 2. Now, recall that $S_1 = F^* D_1 F$, $S_{-1} = DF^* D_{-1} FD^{-1}$, where $D_1 = \text{diag}(1, \bar{\omega}, \bar{\omega}^2, \dots, \bar{\omega}^{n-1})$, $D_{-1} = \delta D_1$, $D = \text{diag}(1, \delta, \delta^2, \dots, \delta^{n-1})$, $\delta = \omega_n^{1/2} = \omega_{2n}$ so that

$$\Delta(A) = F^* D_1 FA - ADF^* D_{-1} FD^{-1}.$$

Multiply to the left the above equation by F , and to the right by DF^* and discover that the matrix $D_1 B - BD_{-1}$ has rank at most 2, where $B = FADF^*$. That is, B is Cauchy like of rank at most 2.

Relying on this simple transformation, Toeplitz systems can be solved in $O(n^2)$ ops by means of the GKO algorithm.

An interesting software package for solving Toeplitz and Toeplitz-like linear systems based on the reduction to Cauchy-like and relying on the GKO algorithm has been provided by Rodriguez and Aricò [3]. It can be downloaded from bugs.unica.it/~gppe/soft/#smt.

Classical fast algorithms, for solving a Toeplitz linear system requiring $O(n^2)$ ops are the algorithms of Levinson [55], Trench [75], and the modification of Zohar [83].

3.5.1 Super-Fast Toeplitz Solvers

The term “fast Toeplitz solvers” denotes algorithms for solving $n \times n$ Toeplitz systems in $O(n^2)$ ops. The term “super-fast Toeplitz solvers” denotes algorithms for solving $n \times n$ Toeplitz systems in $O(n \log^2 n)$ ops. Displacement operators can be used to design super-fast Toeplitz solvers.

Here we describe the idea of the Bitmead-Anderson super-fast solver [16]. Other super-fast solvers have been designed in the literature by Ammar and Gragg [2], Musicus [59] and de Hoog [30].

Consider the displacement operator $F(A) = A - ZAZ^T$ and observe that

$$F(A) = \begin{bmatrix} & \\ & \end{bmatrix} - \begin{bmatrix} & \\ \searrow & \end{bmatrix} = \begin{bmatrix} * & \dots & * \\ \vdots & & \\ * & & \end{bmatrix}.$$

Partition the matrix A as

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix},$$

and, assuming $\det A_{1,1} \neq 0$, factor it as

$$A = \begin{bmatrix} I & 0 \\ A_{2,1}A_{1,1}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{1,1} & A_{1,2} \\ 0 & B \end{bmatrix}, \quad B = A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}.$$

We have the following *fundamental property*: The Schur complement B is such that $\text{rank } F(A) = \text{rank } F(B)$; the other blocks of the above block LU factorization have almost the same displacement rank of the matrix A .

This way, solving two systems with the matrix A (for computing the displacement representation of A^{-1}) is reduced to solving two systems with the matrix $A_{1,1}$ for computing $A_{1,1}^{-1}$ and two systems with the matrix B which has displacement rank 2, plus performing some Toeplitz-vector products.

Denoting $C(n)$ the cost of this recursive procedure, one finds that $C(n) = 2C(n/2) + O(n \log n)$, whence $C(n) = O(n \log^2 n)$.

3.6 Asymptotic Spectral Properties and Preconditioning

The symbol $a(z)$ associated with a sequence of Toeplitz matrices T_n is closely related to the spectral properties of T_n . In this section, we examine this relationship and describe the use of spectral properties to design preconditioners for the efficient solution of positive definite Toeplitz systems through the conjugate gradient iteration.

To this end, we need to recall the concept of *set of sequences* $\{\lambda_i^{(n)}\}_{i=1,n}$, $n \in \mathbb{N}$, distributed as a function $f(x)$.

Let $f(x) : [0, 2\pi] \rightarrow \mathbb{R}$ be a Lebesgue integrable function. We say that the set of sequences $\{\lambda_i^{(n)}\}_{i=1,n}$, $n \in \mathbb{N}$, $\lambda_i^{(n)} \in \mathbb{R}$ is distributed as $f(x)$ if

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n F(\lambda_i^{(n)}) = \frac{1}{2\pi} \int_0^{2\pi} F(f(x)) dx,$$

for any continuous function $F(x)$ with bounded support.

Observe that the set of sequences $\lambda_i^{(n)} = f(2i\pi/n)$, $i = 1, \dots, n$, $n \in \mathbb{N}$, obtained by sampling $f(x)$ at equidistant nodes in $[0, 2\pi]$ is distributed as $f(x)$; in fact the expression $\frac{2\pi}{n} \sum_{i=1}^n F(\lambda_i^{(n)})$ is an approximation to the integral $\int_0^{2\pi} F(f(x))dx$ through the formula of the rectangles.

Roughly speaking, a sequence $\lambda_i^{(n)}$ distributed as $f(x)$ can be interpreted as a sampling of $f(x)$ at a set of nodes which is dense in $[0, 2\pi]$.

With abuse of notation, given $a(z) : \mathbb{T} \rightarrow \mathbb{R}$ we write $a(\theta)$ in place of $a(z(\theta))$, $z(\theta) = \cos \theta + i \sin \theta \in \mathbb{T}$.

Throughout this section we make the following assumptions

- the symbol $a(\theta) : [0 : 2\pi] \rightarrow \mathbb{R}$ is a real valued function so that $a(\theta) = a_0 + 2 \sum_{k=1}^{\infty} a_k \cos k\theta$;
- the matrices T_n form the sequence of Toeplitz matrices associated with $a(\theta)$, i.e., $T_n = (a_{|j-i|})_{i,j=1,n}$; observe that T_n is real and symmetric;
- the values $m_a = \text{ess inf}_{\theta \in [0, 2\pi]} a(\theta)$, $M_a = \text{ess sup}_{\theta \in [0, 2\pi]} a(\theta)$ denote the essential infimum and the essential supremum of $a(\theta)$ over $[0, 2\pi]$;
- the values $\lambda_1^{(n)} \leq \lambda_2^{(n)} \leq \dots \leq \lambda_n^{(n)}$ denote the eigenvalues of T_n sorted in nondecreasing order; the eigenvalues are real since T_n is real symmetric;

We have the following properties which relate the spectrum of T_n to the symbol $a(\theta)$:

1. if $m_a < M_a$ then $\lambda_i^{(n)} \in (m_a, M_a)$, $i = 1, \dots, n$, for any $n \in \mathbb{N}$; if $m_a = M_a$ then $a(\theta)$ is constant and $T_n(a) = m_a I_n$;
2. $\lim_{n \rightarrow \infty} \lambda_1^{(n)} = m_a$, $\lim_{n \rightarrow \infty} \lambda_n^{(n)} = M_a$;
3. the set of eigenvalues sequences $\{\lambda_1^{(n)}, \dots, \lambda_n^{(n)}\}$, $n \in \mathbb{N}$ is distributed as $a(\theta)$;

Some interesting consequences of the above properties are listed below

4. if $a(x) > 0$ the condition number $\mu^{(n)} = \|T_n\|_2 \|T_n^{-1}\|_2$ of T_n is such that $\lim_{n \rightarrow \infty} \mu^{(n)} = M_a/m_a$; that is, $\mu^{(n)}$ is bounded by a constant independent of n ;
5. $a(\theta) > 0$ implies that T_n is uniformly well conditioned;
6. $a(\theta) = 0$ for some θ implies that $\lim_{n \rightarrow \infty} \mu_n = \infty$.

In Fig. 5, we report the eigenvalues of the Toeplitz matrix T_n associated with the symbol $f(\theta) = 2 - 2 \cos \theta - \frac{1}{2} \cos(2\theta)$ for $n = 10, n = 20$, together with the graph of the symbol. It is clear how the set of eigenvalues gets close to the graph of $a(\theta)$ as n grows.

The same asymptotic properties hold true for the eigenvalues of block Toeplitz matrices generated by a real matrix valued symbol $A(x)$, block Toeplitz matrices with Toeplitz blocks generated by a real bivariate symbol $a(x, y)$, multilevel block Toeplitz matrices generated by a real multivariate symbol $a(x_1, x_2, \dots, x_d)$.

The same results hold for the product $A_n = P_n^{-1} T_n$ where T_n and P_n are associated with symbols $t(\theta), p(\theta)$, respectively. In particular, the set of eigenvalues of A_n is distributed as the function $t(\theta)/p(\theta)$ even though A_n is not Toeplitz. Moreover, given $t(\theta) \geq 0$ such that $t(\theta_0) = 0$ for some $\theta_0 \in [0, 2\pi]$, if there

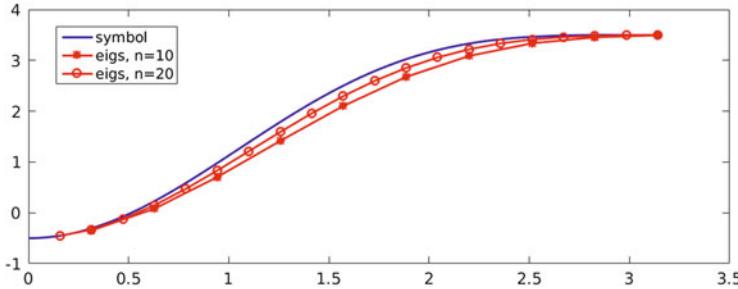


Fig. 5 In red the eigenvalues of the Toeplitz matrix T_n associated with the symbol $f(\theta) = 2 - 2 \cos \theta - \frac{1}{2} \cos(2\theta)$ for $n = 10$, and $n = 20$. In blue the graph of the symbol. As n grows, the values $\lambda_i^{(n)}$ for $i = 1, \dots, n$ tend to be shaped as the graph of the symbol

exists a trigonometric polynomial $p(\theta) = \sum_{k=-K}^K p_k \cos(k\theta)$ such that $p(\theta_0) = 0$, $\lim_{\theta \rightarrow \theta_0} t(\theta)/p(\theta) \neq 0$ then $A_n = P_n^{-1} T_n$ has condition number uniformly bounded by a constant independent of n .

3.6.1 Trigonometric Matrix Algebras and Preconditioning

The solution of a positive definite $n \times n$ Toeplitz system $A_n \mathbf{x} = \mathbf{b}$ can be efficiently approximated with the *preconditioned conjugate gradient (pcg)* method. Below, we recall some features of the *conjugate gradient (cg)* iteration:

- it applies to $n \times n$ positive definite systems $Ax = b$, and generates a sequence of vectors $\{\mathbf{x}_k\}_{k=0,1,2,\dots}$ converging to the solution in n steps;
- computationally, each step requires a matrix-vector product plus some scalar products; thus, for a Toeplitz system the cost is $O(n \log n)$ ops per step;
- the error $r_k = \|\mathbf{x}_k - \mathbf{x}\|_A$ in the A -norm defined by $\|\mathbf{v}\|_A = (\mathbf{v}^T A \mathbf{v})^{1/2}$, is such that $r_k \leq 2\theta^k r_0$, where $\theta = (\sqrt{\mu} - 1)/(\sqrt{\mu} + 1)$, $\mu = \lambda_n^{(n)} / \lambda_1^{(n)}$ is the spectral condition number of A ;
- from the above bound on the residual it follows that convergence is fast for well-conditioned systems, slow otherwise. However:
- according to the Axelsson-Lindskog Theorem, convergence is fast if the eigenvalues are clustered around 1; in fact, the informal statement of this theorem says that if A has all the eigenvalues in the interval $[\alpha, \beta]$ where $0 < \alpha < 1 < \beta$ except for q outliers which stay outside $[\alpha, \beta]$, then after roughly q steps, the error r_k is bounded by $\gamma_1 \theta_1^k$ for $\theta_1 = (\sqrt{\mu_1} - 1)/(\sqrt{\mu_1} + 1)$, where $\mu_1 = \beta/\alpha$ and γ_1 is a constant.

The preconditioned conjugate gradient iteration consists of the conjugate gradient method applied to the system $P^{-1/2} A_n P^{-1/2} P^{1/2} x = P^{-1/2} b$, where the positive definite matrix P is the preconditioner and $P^{-1/2}$ is the matrix square root of P^{-1} . It can be shown that at each step, the pcg iteration requires the solution of a system

with matrix P , the matrix-vector multiplication by the matrix A , and some scalar products. Therefore, in order to be effective, the preconditioner P must be chosen so that:

- solving the system with matrix P is cheap;
- the matrix P mimics the matrix A so that $P^{-1/2}AP^{-1/2}$ has either condition number close to 1, or has eigenvalues in a narrow interval $[\alpha, \beta]$ containing 1, except for *few outliers*. Observe that the matrix $P^{-1/2}AP^{-1/2}$ has the same eigenvalues of $P^{-1}A$ since the two matrices are similar.

For Toeplitz matrices, P can be chosen in a trigonometric algebra so that each step of pcg costs just $O(n \log n)$ ops, that is the cost of a discrete trigonometric transform (Fourier, Sine, Cosine or Hartley). Moreover, it is possible to prove that in this case, under an appropriate choice in the given trigonometric algebra, the spectrum of $P^{-1}A$ is clustered around 1. This implies a *super-linear convergence* of the iteration.

An effective example of preconditioner is the circulant preconditioner of T. Chan where $P_n = C_n$, and C_n is the symmetric circulant matrix which minimizes the Frobenius norm $\|A_n - C_n\|_F$. For this preconditioner, the eigenvalues of $B_n = P_n^{-1}C_n$ are clustered around 1. That is, for any ϵ there exists n_0 such that for any $n \geq n_0$ the eigenvalues of $P_n^{-1}A$ belong to $[1 - \epsilon, 1 + \epsilon]$ except for a few outliers.

Effective preconditioners can be found in the τ and in the Hartley algebras, as well as in the class of banded Toeplitz matrices.

Below, we provide a specific instance of preconditioner. Consider the $n \times n$ matrix A associated with the symbol $a(\theta) = 6 + 2(-4 \cos(\theta) + \cos(2\theta))$:

$$A = \begin{bmatrix} 6 & -4 & 1 & & \\ -4 & 6 & -4 & 1 & \\ 1 & -4 & \ddots & \ddots & \ddots \\ & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Its eigenvalues are distributed as the symbol $a(\theta)$ and its condition number is $O(n^4)$. In Fig. 6 we report the graph of the symbol over $[0, \pi]$; recall that since $a(\theta)$ is an even periodic function then it is symmetric with respect to π .

The eigenvalues of the preconditioned matrix $P^{-1}A$, where P is circulant preconditioner of T. Chan, are clustered around 1 with very few outliers. Figure 7 reports the graph of the logarithms of the eigenvalues of A (in red) and of the logarithm of the eigenvalues of $P^{-1}A$ in blue. It is clear that almost all the eigenvalues of $P^{-1}A$ are clustered around 1.

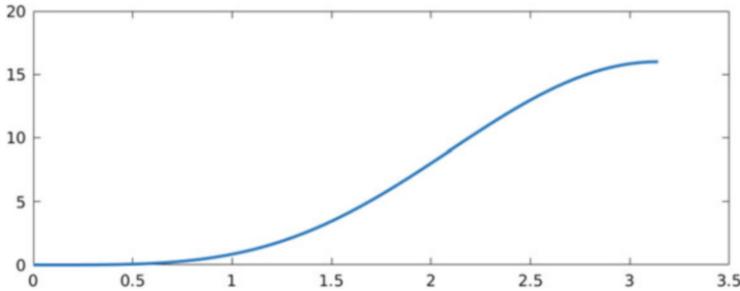


Fig. 6 Plot of the symbol $a(\theta) = 6 + 2(-4 \cos(\theta) + \cos(2\theta))$ over $[0, \pi]$

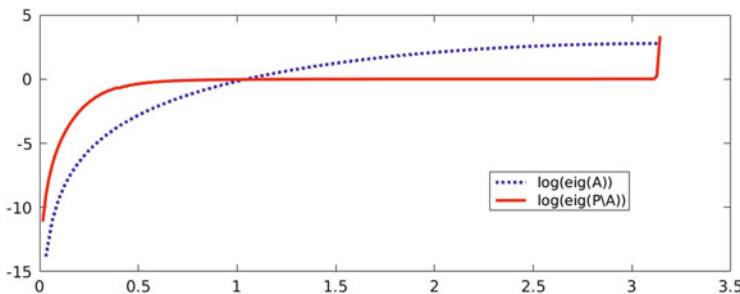


Fig. 7 Logarithm of the eigenvalues of A (in red) and of $P^{-1}A$ in blue. Here, A is the $n \times n$ Toeplitz matrix associated with the symbol $a(\theta) = 6 + 2(-4 \cos \theta + \cos 2\theta)$ and P is the circulant preconditioner of T. Chan

3.7 Rank Structures

An $n \times n$ matrix A is *rank structured* with rank k if all its submatrices contained in the upper triangular part or in the lower triangular part have rank at most k .

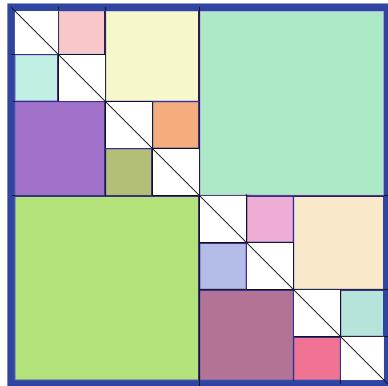
The interest on rank-structured matrices is dated back to the works of Gantmacher and Kreĭn [36]. A wide literature exists, which involves many scholars and research groups, with a blend of theoretical analysis, computational analysis, design of algorithms and applications. For details we refer the reader to the recent books [34, 35, 77, 78], and to the references cited therein.

The terms *semi-separable matrix* and *quasi-separable matrix* are used in the literature to denote this class of matrices with some slightly different meaning.

Here are some examples.

- The sum of a diagonal matrix and a matrix of rank k ;
- a tridiagonal matrix ($k = 1$);
- the inverse of an irreducible tridiagonal matrix ($k = 1$);
- any orthogonal matrix in Hessenberg form ($k = 1$);
- a Frobenius companion matrix ($k = 1$);
- a block Frobenius companion matrix with $m \times m$ blocks ($k = m$).

Fig. 8 Hierarchical representation of a rank structured matrix. Each square represents a matrix of rank 1 which can be stored by means of two vectors



An interesting subclass of rank-structured matrices is the one defined by a *generator*. A rank structured matrix A of rank k has a generator if there exist matrices B, C of rank k such that $\text{triu}(A) = \text{triu}(B)$, $\text{tril}(A) = \text{tril}(C)$.

Observe that an irreducible tridiagonal matrix has no generator; a (block) companion matrix has a generator defining the upper triangular part but no generator concerning the lower triangular part.

Nice properties have been proved concerning rank-structured matrices. Here are some of them.

For a k -rank-structured matrix A

1. the inverse of A is k rank structured;
2. the LU factors of A are k -rank structured;
3. the QR factors of A are rank-structured with rank k and $2k$;
4. solving an $n \times n$ system with a k -rank structured matrix costs $O(nk^2)$ ops;
5. the Hessenberg form H of A is $(2k - 1)$ -rank structured;
6. if A is the sum of a *real diagonal* matrix plus a matrix of rank k then the QR iteration generates a sequence A_i of rank-structured matrices;
7. computing H costs $O(nk^2)$ ops;
8. computing $\det H$ costs $O(nk)$ ops.

A useful representation, with an implementation of specific algorithms is given by Börhm et al. [19] and relies on a hierarchical representation. More precisely, it relies on splitting the matrix into blocks and in representing blocks non-intersecting the diagonal by means of low rank matrices. This strategy is represented in Fig. 8.

3.8 Bibliographical Notes

Concerning references on Toeplitz matrices and linear operators, there is a huge literature. By using MathSciNet, looking for paper with “Toeplitz” in the title

provides a list of more than 5000 items. Here we give some references to books, surveys and to classical papers with more attention to computational issues.

Besides the classical results of Otto Toeplitz, and the pioneering monograph of Grenander and Szegő [41], a systematic work has been carried out by Boettcher and the Chemnitz group, in particular we refer to the books [20–22]. Extensions and generalizations of the original theorem by O. Toeplitz have been given by Tytyshnikov [76].

Many researchers and research groups have given contributions to the analysis of the asymptotic spectral properties of Toeplitz and locally Toeplitz matrices, with applications to preconditioning, among whom Serra Capizzano, Tytyshnikov, Zamarashkin, Tilli, di Benedetto, R. Chan, T. Chan, Strang, Huckle, Noutsos. Here is a very partial list of sample papers [25–27, 49, 63, 68, 70, 73, 74, 81]. For more details and for an extended bibliography, we refer to the self contained survey [69]. A recent review on generalized Toeplitz matrices is given in [37], another important reference is the book by R. Chan and X.-Q. Jin [28]. Extensions of asymptotic properties to singular values have been provided by Widom [81]. Concerning applications to preconditioning Toeplitz systems there is a wide literature including the seminal paper by Strang [73], the preconditioners of R. Chan [27] and of T. Chan [25], the band preconditioner by Serra Capizzano [68], the analysis of matrix algebras by Di Benedetto [9, 33], Favati [10], Kailath and Olshevsky [50], the negative results of Serra and Tytyshnikov [70] and more. See also the already cited book [28].

Concerning rank structured matrices, we refer to the recent books by Vandebril, Van Barel and Mastronardi [77, 78], and to the books by Eidelman, Gohberg, and Haimovici [34, 35], to the works of Börm et al. [19], to the papers by Boito, Chandrasekaran, Dewilde, Gemignani, Eidelman, Gohberg, and Gu, see for instance [17, 18, 71, 82] and to the literature cited therein.

4 Algorithms for Structured Markov Chains: The Finite Case

In this section we deal with algorithms for solving Markov chains having a finite set of states, where the transition matrix has the typical structures derived by the queueing models examined in Sect. 2. More precisely, we will consider block Hessenberg matrices, in particular block tridiagonal, which have the block Toeplitz structure except for the boundary entries. Since the set of states is finite, then the transition matrices that we will examine have finite size. The case of infinite matrices will be treated in Sect. 5.

4.1 The Block Tridiagonal Case

Let us start with a QBD problem. In this case we have to solve a homogeneous linear system with the matrix $I - P$, where the irreducible stochastic matrix P is an $n \times n$ block-tridiagonal almost block-Toeplitz matrix with $m \times m$ blocks. Thus, the equation takes the form

$$[\boldsymbol{\pi}_1^T \ \boldsymbol{\pi}_2^T \ \dots \ \boldsymbol{\pi}_n^T] \begin{bmatrix} I - \widehat{A}_0 & -A_1 & & \\ -A_{-1} & I - A_0 & -A_1 & \\ & \ddots & \ddots & \ddots & \\ & & -A_{-1} & I - A_0 & -A_1 \\ & & & -A_{-1} & I - \widetilde{A}_0 \end{bmatrix} = 0,$$

where the vector $\boldsymbol{\pi}$ has been partitioned into blocks $\boldsymbol{\pi}_i$ of length m . For the sake of simplicity denote

$$\begin{aligned} B_{-1} &= -A_{-1}, & B_0 &= I - A_0, & B_1 &= -A_1, \\ \widehat{B}_0 &= I - \widehat{A}_0, & \widetilde{B}_0 &= I - \widetilde{A}_0, \end{aligned}$$

so that our problem becomes

$$[\boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \dots, \boldsymbol{\pi}_n^T] \mathcal{B} = 0, \quad \mathcal{B} = \begin{bmatrix} \widehat{B}_0 & B_1 & & & \\ B_{-1} & B_0 & B_1 & & \\ & B_{-1} & B_0 & B_1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & B_{-1} & B_0 & B_1 \\ & & & & B_{-1} & \widetilde{B}_0 \end{bmatrix}.$$

The matrix $\mathcal{B} = I - P$ is singular since the vector $\mathbf{e} = (1, \dots, 1)^T$ is in its right null space being $P\mathbf{e} = \mathbf{e}$. By the Perron-Frobenius theorem, if P is irreducible the null space of $I - P$ has dimension 1. The property described in the following remark is useful.

Remark 2 If $\mathcal{B} = I - P$, with P stochastic and irreducible, then all the proper principal submatrices of \mathcal{B} are nonsingular.

This property can be proved by contradiction. Assume that a principal submatrix $\widehat{\mathcal{B}}$ of \mathcal{B} is singular. Then the corresponding submatrix \widehat{P} of P has the eigenvalue 1. This contradicts Lemma 2.6 of Varga [79] which says that $\rho(\widehat{P}) < \rho(P)$ if $P \geq 0$ is irreducible. In particular, since the proper leading principal submatrices are

invertible, there exists unique the LU factorization where only the last diagonal entry of U is zero.

The most immediate possibility to solve the homogeneous system $\pi^T \mathcal{B} = 0$ is to compute the LU factorization $\mathcal{B} = \mathcal{L}\mathcal{U}$ and solve the system $\pi^T \mathcal{L} = (0, \dots, 0, 1)$ since $[0, \dots, 0, 1]\mathcal{U} = 0$.

This method is numerically stable since no cancellation is encountered if the method is applied with the Grassmann-Taksar-Heyman (GTH) trick [40]; its cost is $O(nm^3)$, but it is not the most efficient method. Moreover, the structure of the matrix is only partially exploited by the LU factorization.

A more efficient algorithm relies on the *cyclic reduction (CR)* technique introduced by G.H. Golub and R.W. Hockney in a paper by Hockney [48], where the author writes “*These equations form a tridiagonal system with periodic boundary conditions and a particularly efficient method of solution has been devised in collaboration with Dr. G. Golub. This involves the recursive application of the process of cyclic reduction which follows.*” This algorithm was described with more details in a subsequent paper by Buzbee et al. [24], for the numerical solution of the discrete Poisson equation in a rectangle. Later on, cyclic reduction has been applied to the solution of different computational problems including matrix equations, banded Toeplitz systems and Markov chains. For a comprehensive treatment of CR we refer the reader to the survey [11] and to the books [14, 15].

We give an outline of CR where, for simplicity, we suppose that $n = 2^q$. Apply an even-odd permutation to the block unknowns and block equation of the system $\pi^T \mathcal{B} = 0$ and get

$$\left[\begin{array}{c|c} \pi_{\text{even}}^T & \pi_{\text{odd}}^T \end{array} \right] \left[\begin{array}{c|c} B_0 & B_{-1} \ B_1 \\ \ddots & B_{-1} \ \ddots \\ B_0 & \ddots \ B_1 \\ \hline \widetilde{B}_0 & B_{-1} \\ \hline B_1 & \widehat{B}_0 \\ B_{-1} \ B_1 & B_0 \\ \ddots \ \ddots & \ddots \\ B_{-1} \ B_1 & B_0 \end{array} \right] = 0, \quad (1)$$

where

$$\left[\pi_{\text{even}}^T \ | \ \pi_{\text{odd}}^T \right] = \left[\pi_2^T \ \pi_4^T \ \dots \ \pi_{\frac{n}{2}}^T \ | \ \pi_1^T \ \pi_3^T \ \dots \ \pi_{\frac{n-2}{2}}^T \right].$$

Denote by $\left[\begin{array}{c|c} D_1 & V \\ \hline W & D_2 \end{array} \right]$ the matrix in (1) obtained after the permutation, compute its block LU factorization and get

$$\left[\begin{array}{c|c} D_1 & V \\ \hline W & D_2 \end{array} \right] = \left[\begin{array}{cc} I & 0 \\ WD_2^{-1} & I \end{array} \right] \left[\begin{array}{cc} D_1 & V \\ 0 & \mathcal{S} \end{array} \right], \quad \mathcal{S} = D_2 - WD_1^{-1}V,$$

where the Schur complement \mathcal{S} is singular. Thus, the original problem is reduced to

$$[\boldsymbol{\pi}_{\text{even}}^T : \boldsymbol{\pi}_{\text{odd}}^T] \left[\begin{array}{cc} I & 0 \\ WD_2^{-1} & I \end{array} \right] \left[\begin{array}{cc} D_1 & V \\ 0 & \mathcal{S} \end{array} \right] = [0 : 0],$$

that is

$$\begin{aligned} \boldsymbol{\pi}_{\text{odd}}^T \mathcal{S} &= 0, \\ \boldsymbol{\pi}_{\text{even}}^T &= -\boldsymbol{\pi}_{\text{odd}}^T WD_2^{-1}. \end{aligned} \tag{2}$$

This way, computing $\boldsymbol{\pi}$ is reduced to computing $\boldsymbol{\pi}_{\text{odd}}$ which is a problem of size $mn/2$. It is interesting to discover that *the structure of the Schur complement \mathcal{S} is the same as that of the original matrix \mathcal{B}* . In fact, from the equation

$$\mathcal{S} = \begin{bmatrix} \widehat{B}_0 & & & \\ & B_0 & & \\ & & \ddots & \\ & & & B_0 \end{bmatrix} - \begin{bmatrix} B_1 & & & \\ B_{-1} & B_1 & & \\ & \ddots & \ddots & \\ & & B_{-1} & B_1 \end{bmatrix} \begin{bmatrix} B_0 & & & \\ & \ddots & & \\ & & B_0 & \widetilde{B}_0 \\ & & & \end{bmatrix}^{-1} \begin{bmatrix} B_{-1} & B_1 & & \\ & B_{-1} & \ddots & \\ & & \ddots & B_1 \\ & & & B_{-1} \end{bmatrix}$$

we easily find that

$$\mathcal{S} = \begin{bmatrix} \widehat{B}'_0 & B'_1 & & \\ B'_{-1} & B'_0 & B'_1 & \\ & \ddots & \ddots & \ddots \\ & & B'_{-1} & B'_0 & B'_1 \\ & & & B'_{-1} & \widetilde{B}'_0 \end{bmatrix},$$

where, with $C = B_0^{-1}$, we have

$$\begin{aligned} B'_0 &= B_0 - B_{-1}CB_1 - B_1CB_{-1}, \\ B'_1 &= -B_1CB_1, \\ B'_{-1} &= -B_{-1}CB_{-1}, \\ \widehat{B}'_0 &= \widehat{B}_0 - B_1CB_{-1}, \\ \widetilde{B}'_0 &= \widetilde{B}_0 - B_1\widetilde{B}_0^{-1}B_{-1} - B_{-1}CB_1. \end{aligned} \tag{3}$$

Observe that in (3) only a finite number of matrix multiplications and matrix inversions are involved so that the cost of computing \mathcal{S} is simply $O(m^3)$. The odd-even reduction can be cyclically applied to the new block tridiagonal matrix until we arrive at a Schur complement of size 2×2 . This procedure is synthesized below in a recursive form.

1. if $n = 2$ compute $\boldsymbol{\pi}$ such that $\boldsymbol{\pi}^T \mathcal{B} = 0$ by using the LU decomposition;
2. otherwise perform an odd-even permutation and compute the Schur complement \mathcal{S} by means of (3);
3. compute $\boldsymbol{\pi}_{\text{odd}}$ such that $\boldsymbol{\pi}_{\text{odd}}^T \mathcal{S} = 0$ by recursively applying this algorithm to \mathcal{S} ;
4. compute $\boldsymbol{\pi}_{\text{even}}$ by means of (2) and output $\boldsymbol{\pi}$ normalized so that $\sum_i \pi_i = 1$.

We may easily give an asymptotic estimate of the overall computational cost of this procedure to compute $\boldsymbol{\pi}$. Denoting c_n the computational cost of performing the CR step at size n , we have $c_2 = O(m^3)$, and for $n > 2$ we have $c_n = c_{n/2} + O(m^3) + O(nm^2)$ where $O(m^3)$ is the cost of computing the Schur complement, while $O(nm^2)$ is the cost of computing $\boldsymbol{\pi}_{\text{odd}}$.

Thus, the overall cost of CR turns out $O(m^3 \log_2 n)$ ops for the Schur complementation and $O(m^2 n + m^2 \frac{n}{2} + m^2 \frac{n}{4} + \dots + m^2) = O(m^2 n)$, for the computation of $\boldsymbol{\pi}$, that is for the back substitution part. The total cost is $O(m^3 \log n + m^2 n)$ ops vs. $O(m^3 n)$ ops of the standard LU approach.

Remark 3 If n is odd, then the even-odd permutation applied to \mathcal{B} yields a matrix with the same shape as before, i.e., $\left[\begin{array}{c|c} D_1 & V \\ \hline W & D_2 \end{array} \right]$, where D_1 is a block diagonal matrix with B_0 in the main diagonal, while $D_2 = \text{diag}(\widehat{B}_0, B_0, \dots, B_0, \widetilde{B}_0)$. Also in this case, the Schur complement is still block tridiagonal, block Toeplitz except for the first and last diagonal entry. The main computational difference is that in this case

the inversion of only one block, i.e., B_0 , is required. Choosing $n = 2^q + 1$, after one step the size becomes $2^{q-1} + 1$ so that CR can be cyclically applied. The advantage is that only one matrix inversion must be performed at each step of CR.

4.2 The Case of a Block Hessenberg Matrix

Cyclic reduction can be applied to the block Hessenberg case, where the problem is stated in the following form

$$\pi^T \mathcal{B} = 0, \quad \mathcal{B} = \begin{bmatrix} \widehat{B}_0 & \widehat{B}_1 & \widehat{B}_2 & \dots & \widehat{B}_{n-2} & \widehat{B}_{n-1} \\ B_{-1} & B_0 & B_1 & \ddots & B_{n-3} & \widetilde{B}_{n-2} \\ & B_{-1} & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & B_1 & \\ & & & B_{-1} & B_0 & \widetilde{B}_1 \\ & & & & B_{-1} & \widetilde{B}_0 \end{bmatrix},$$

$B_0 = I - A_0$, $\widehat{B}_0 = I - \widehat{A}_0$, $\widetilde{B}_0 = I - \widetilde{A}_0$, and $B_j = -A_j$, for $j \geq -1$, $\widehat{B}_j = -\widehat{A}_j$, $\widetilde{B}_j = -\widetilde{A}_j$, for $j > 0$.

Assume $n = 2m + 1$, apply an even-odd block permutation to block-rows and block-columns and get a matrix partitioned into four blocks where each block is a suitable submatrix of the original block Hessenberg matrix. In order to better explain the structure of the four blocks, we represent the original matrix \mathcal{B} in the following form

$$\begin{bmatrix} \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} \\ \cdot & \mathbf{x} \\ \cdot & \cdot & \mathbf{x} \\ \cdot & \cdot & \cdot & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \cdot & \cdot & \cdot & \cdot & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \cdot & \mathbf{x} & \mathbf{x} \end{bmatrix},$$

where in boldface we have denoted the boundary values where the Toeplitz structure is not satisfied, and we circle the entries of the submatrix which form each block.

We get the following four blocks whose entries are denoted by a circle

$$\begin{array}{ll}
 (1, 1) & \left[\begin{array}{cccccccc} x & x & x & x & x & x & x & x & x \\ x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x \\ \cdot & x & x & x & x & x & x & x & x \\ \cdot & \textcircled{\circ} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x \\ \cdot & \cdot & \cdot & x & x & x & x & x & x \\ \cdot & \textcircled{\circ} & \cdot & \textcircled{\circ} & x & \textcircled{x} & x & \textcircled{x} & x \\ \cdot & \cdot & \cdot & \cdot & x & x & x & x & x \\ \cdot & \textcircled{\circ} & \cdot & \textcircled{\circ} & \cdot & \textcircled{\circ} & x & \textcircled{x} & x \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & x & x & x \end{array} \right], \quad (1, 2) & \left[\begin{array}{cccccccc} x & x & x & x & x & x & x & x & x \\ \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} \\ \cdot & x & x & x & x & x & x & x & x \\ \textcircled{\circ} & \cdot & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} \\ \cdot & \cdot & \cdot & x & x & x & x & x & x \\ \textcircled{\circ} & \cdot & \textcircled{\circ} & \cdot & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} \\ \cdot & \cdot & \cdot & \cdot & \cdot & x & x & x & x \\ \textcircled{\circ} & \cdot & \textcircled{\circ} & \cdot & \textcircled{\circ} & \cdot & \textcircled{x} & x & \textcircled{x} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & x & x & x \end{array} \right], \\
 (2, 1) & \left[\begin{array}{cccccccc} x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x \\ x & x & x & x & x & x & x & x & x \\ \cdot & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x \\ \cdot & \cdot & x & x & x & x & x & x & x \\ \cdot & \textcircled{\circ} & \cdot & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x \\ \cdot & \cdot & \cdot & x & x & x & x & x & x \\ \cdot & \textcircled{\circ} & \cdot & \textcircled{\circ} & \cdot & \textcircled{x} & x & \textcircled{x} & x \\ \cdot & \cdot & \cdot & \cdot & \cdot & x & x & x & x \\ \cdot & \textcircled{\circ} & \cdot & \textcircled{\circ} & \cdot & \textcircled{\circ} & \cdot & \textcircled{x} & x \end{array} \right], \quad (2, 2) & \left[\begin{array}{cccccccc} \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} \\ x & x & x & x & x & x & x & x & x \\ \textcircled{\circ} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} \\ \cdot & x & x & x & x & x & x & x & x \\ \textcircled{\circ} & \cdot & \textcircled{\circ} & x & \textcircled{x} & x & \textcircled{x} & x & \textcircled{x} \\ \cdot & \cdot & \cdot & x & x & x & x & x & x \\ \textcircled{\circ} & \cdot & \textcircled{\circ} & \cdot & \textcircled{\circ} & x & \textcircled{x} & x & \textcircled{x} \\ \cdot & \cdot & \cdot & \cdot & \cdot & x & x & x & x \\ \textcircled{\circ} & \cdot & \textcircled{\circ} & \cdot & \textcircled{\circ} & \cdot & \textcircled{x} & x & \textcircled{x} \end{array} \right]. \end{array}$$

It is clear that the (1, 1) block is upper triangular Toeplitz, the (1, 2) block is upper triangular and Toeplitz except for the last column, the (2, 1) block is upper Hessenberg and Toeplitz except for the first row, finally, the (2,2) block is upper triangular and Toeplitz except for its first row and last column. More precisely, denoting D_1 , V , W and D_2 these four blocks, respectively, we have

$$D_1 = \begin{bmatrix} B_0 & B_2 & \dots & B_{2m-4} \\ & \ddots & & \vdots \\ B_0 & & & B_2 \\ & \ddots & & B_2 \\ & & B_0 \end{bmatrix}, \quad D_2 = \begin{bmatrix} \widehat{B}_0 & \widehat{B}_2 & \widehat{B}_4 & \dots & \widehat{B}_{2m-4} & \widehat{B}_{2m-2} \\ B_0 & B_2 & \dots & B_{2m-6} & \widetilde{B}_{2m-4} \\ & \ddots & \ddots & & \vdots & \vdots \\ & & B_0 & B_2 & \widetilde{B}_4 \\ & & & B_0 & \widetilde{B}_2 \\ & & & & \widetilde{B}_0 \end{bmatrix}, \\
 W = \begin{bmatrix} \widehat{B}_1 & \widehat{B}_3 & \dots & \widehat{B}_{2m-3} \\ B_{-1} & B_1 & \dots & B_{2m-5} \\ & \ddots & \ddots & \vdots \\ & & B_{-1} & B_1 \\ & & & B_{-1} \end{bmatrix}, \quad V = \begin{bmatrix} B_{-1} & B_1 & \dots & B_{2m-5} & \widetilde{B}_{2m-3} \\ B_{-1} & \ddots & & \vdots & \vdots \\ & \ddots & B_1 & \widetilde{B}_3 \\ & & B_{-1} & \widetilde{B}_1 \end{bmatrix}$$

and the vector $\boldsymbol{\pi}$ satisfies the equation

$$[\boldsymbol{\pi}_{\text{even}}^T : \boldsymbol{\pi}_{\text{odd}}^T] \begin{bmatrix} D_1 & V \\ W & D_2 \end{bmatrix} = [0 : 0].$$

Computing the LU factorization of $\begin{bmatrix} D_1 & V \\ W & D_2 \end{bmatrix}$ yields

$$[\boldsymbol{\pi}_{\text{even}}^T : \boldsymbol{\pi}_{\text{odd}}^T] \begin{bmatrix} I & 0 \\ WD_1^{-1} & I \end{bmatrix} \begin{bmatrix} D_1 & V \\ 0 & \mathcal{S} \end{bmatrix} = 0, \quad \mathcal{S} = D_2 - WD_1^{-1}V,$$

where the Schur complement \mathcal{S} is singular. Thus, the problem is reduced to computing

$$\begin{aligned} \boldsymbol{\pi}_{\text{odd}}^T \mathcal{S} &= 0, \\ \boldsymbol{\pi}_{\text{even}}^T &= -\boldsymbol{\pi}_{\text{odd}}^T WD_1^{-1}. \end{aligned}$$

It is easy to show that the Schur complement \mathcal{S} preserves the original structure of the matrix, that is, \mathcal{S} is upper block Hessenberg and block Toeplitz except in the first row and in the last column. This property is clearly satisfied by D_2 . Concerning $WD_1^{-1}V$, the property follows from the fact that block triangular block Toeplitz matrices are closed under inversion and multiplication. In fact, D_1^{-1} is upper block triangular block Toeplitz since it is the inverse of an upper block triangular block Toeplitz matrix. The product $D_1^{-1}V$ is a rectangular matrix given by an upper block triangular block Toeplitz matrix with the addition of one more column at the end. Finally $W(D_1^{-1}V)$ is block Hessenberg and block Toeplitz except for its first row and last column, by construction.

Now assume that $n = 2^q + 1$, where q is a positive integer. This way, the Schur complement has odd size $2^{q-1} + 1$ and the process can be repeated recursively to \mathcal{S} until a 3×3 block matrix is obtained. We can describe this procedure by means of the following recursive algorithm.

1. If $n = 3$ compute $\boldsymbol{\pi}$ such that $\boldsymbol{\pi}^T \mathcal{B} = 0$ by using the LU decomposition;
2. otherwise apply the block even-odd permutation and compute the Schur complement \mathcal{S} ;
3. compute $\boldsymbol{\pi}_{\text{odd}}$ such that $\boldsymbol{\pi}_{\text{odd}}^T \mathcal{S} = 0$ by recursively applying this algorithm to the Schur complement \mathcal{S} ;
4. compute $\boldsymbol{\pi}_{\text{even}}^T = -(\boldsymbol{\pi}_{\text{odd}}^T W)D_1^{-1}$;
5. output $\boldsymbol{\pi}$.

It is easy to perform the asymptotic cost analysis of this algorithm. Concerning the Schur complementation, we have to invert a block triangular block Toeplitz matrix, relying on the algorithm shown in Sect. 3.3.4, for the cost of $O(m^3n + m^2n \log n)$ ops. We have to multiply block triangular block Toeplitz matrices, relying on

on the algorithms shown in Sect. 3.3.1, for the cost of $O(m^3n + m^2n \log n)$ ops. Thus, the cost of Schur complementation at a single step of CR is $O(m^3n + m^2n \log n)$ ops.

Computing π_{even} , given π_{odd} amounts to computing the product of a block triangular Toeplitz matrix and a vector for the cost of $O(m^2n + m^2n \log n)$.

Thus, denoting c_n the overall cost for a cyclic reduction step applied to an $n \times n$ block Hessenberg almost block Toeplitz matrix with $m \times m$ blocks, we have

$$c_n = c_{\frac{n+1}{2}} + O(m^3n + m^2n \log n), \quad c_3 = O(m^3),$$

so that the overall computational cost to carry out CR is $O(m^3n + m^2n \log n)$ ops vs. $O(m^3n^2)$ ops of the standard LU factorization.

4.3 Applicability of CR

An important issue concerns the applicability of CR, or equivalently, the nonsingularity of the matrices that must be inverted at each step of CR. In fact, in order to be applied, cyclic reduction requires the nonsingularity of certain matrices at all the steps of the iteration.

Consider the case of a block tridiagonal matrix \mathcal{B} and examine the first step of CR. Here, we need the nonsingularity of B_0 , that is a principal submatrix of \mathcal{A} . After the first step, we have a new block tridiagonal matrix \mathcal{S} defined by the blocks B'_i , for $i = -1, 0, 1$, and by the boundary blocks \widetilde{B}'_0 and \widehat{B}'_0 defined in (3).

To perform the second step of CR it is needed that $\det B'_0 \neq 0$, where $B'_0 = B_0 - B_{-1}B_0^{-1}B_1 - B_1B_0^{-1}B_{-1}$. Now, observe that B'_0 is the Schur complement of B_0 in

$$\begin{bmatrix} B_0 & 0 & B_1 \\ 0 & B_0 & B_{-1} \\ B_{-1} & B_1 & B_0 \end{bmatrix},$$

which is similar by permutation to $\text{trid}_3(B_{-1}, B_0, B_1)$, where we denote by $\text{trid}_n(A, B, C)$ the $n \times n$ block tridiagonal and block Toeplitz matrix whose blocks are A, B, C . By the properties of the Schur complement we have

$$\det \text{trid}_3(B_{-1}, B_0, B_1) = \det B_0 \det B'_0,$$

so that the second step of CR can be applied if $\det \text{trid}_3(B_{-1}, B_0, B_1) \neq 0$.

Inductively, we can prove that $\det \text{trid}_{2i-1}(B_{-1}, B_0, B_1) \neq 0$, $i = 1, 2, \dots, k$ if and only if CR can be applied for the first k steps with no breakdown.

Recall that, since $\mathcal{B} = I - P$, with P stochastic and irreducible, then in view of Remark 2 all the principal submatrices of \mathcal{B} are nonsingular. Thus, in the block tridiagonal case, CR can be carried out with no breakdown if applied to a matrix of

the kind $\mathcal{B} = I - P$, where P is stochastic and irreducible. A similar argument can be used to prove the applicability of CR in the block Hessenberg case.

It is interesting to point out that CR can be applied also in case of breakdown where singular or ill-conditioned matrices are encountered, just by skipping the singular steps. To show this, denote $B_{-1}^{(k)}, B_0^{(k)}, B_1^{(k)}$ the matrices generated by CR at step k . Assume $\det \text{trid}_{2^k-1}(B_{-1}, B_0, B_1) \neq 0$, set $R^{(k)} = \text{trid}_{2^k-1}(B_{-1}, B_0, B_1)^{-1}$, $R^{(k)} = (R_{i,j}^{(k)})$. Then, playing with Schur complements one finds that

$$\begin{aligned} B_{-1}^{(k)} &= -B_{-1} R_{n,1}^{(k)} B_{-1}, \\ B_0^{(k)} &= B_0 - B_{-1} R_{n,n}^{(k)} B_1 - B_1 R_{1,1}^{(k)} B_{-1}, \\ B_1^{(k)} &= -B_1 R_{1,n}^{(k)} B_1. \end{aligned}$$

Matrices $B_i^{(k)}$ are well defined if $\det \text{trid}_{2^k-1}(B_{-1}, B_0, B_1) \neq 0$, no matter if $\det \text{trid}_{2^h-1}(B_{-1}, B_0, B_1) = 0$, for some $h < k$, i.e., if CR encounters breakdown.

4.4 A Functional Interpretation of Cyclic Reduction

Cyclic reduction has a nice functional interpretation which relates it to the Graeffe-Lobachevsky-Dandelin iteration analyzed by Ostrowski in [64, 65]. This interpretation enables us to apply CR to infinite matrices and to solve QBD and M/G/1-type Markov chains as well.

Let us recall the Graeffe-iteration. Let $p(z)$ be a polynomial of degree n having q zeros of modulus less than 1 and $n - q$ zeros of modulus greater than 1. Observe that in the product $p(z)p(-z)$ the odd powers of z cancel out. This way, $p(z)p(-z) = p_1(z^2)$ is a polynomial of degree n in z^2 . Therefore, the zeros of $p_1(z)$ are the squares of the zeros of $p(z)$.

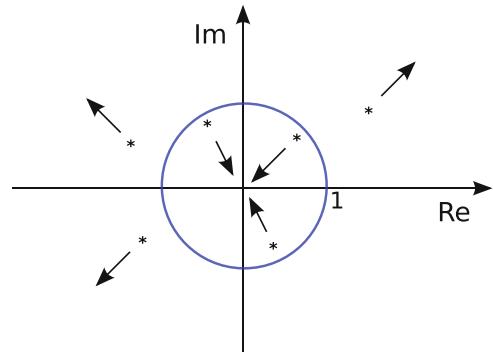
The sequence defined by

$$\begin{aligned} p_0(z) &= p(z), \\ p_{k+1}(z^2) &= p_k(z)p_k(-z), \quad k = 0, 1, \dots \end{aligned}$$

is such that the zeros of $p_k(z)$ are the 2^k powers of the zeros of $p_0(z) = p(z)$. Thus, the zeros of $p_k(z)$ inside the unit disk converge quadratically to zero, while the zeros outside the unit disk converge quadratically to infinity as shown in Fig. 9. Whence, the sequence $p_k(z)/\|p_k\|_\infty$ obtained by normalizing $p_k(z)$ with the coefficient of largest modulus converges to z^q .

Can we do something similar with matrix polynomials or with matrix Laurent polynomials? For simplicity, we limit our analysis to the block tridiagonal case. Let B_{-1}, B_0, B_1 be $m \times m$ matrices, B_0 nonsingular, and define the Laurent matrix polynomial $\varphi(z) = B_{-1}z^{-1} + B_0 + B_1z$.

Fig. 9 Dynamics of the zeros of the polynomials generated by the Graeffe sequence



Consider $\varphi(z)B_0^{-1}\varphi(-z)$ and discover that in the latter matrix polynomial, the odd powers of z cancel out, that is,

$$\varphi_1(z^2) = \varphi(z)B_0^{-1}\varphi(-z), \quad \varphi_1(z) = B_{-1}^{(1)}z^{-1} + B_0^{(1)} + B_1^{(1)}z.$$

The above equations provide a generalization of the Graeffe iteration to Laurent matrix polynomials of the kind $z^{-1}B_{-1} + B_0 + zB_1$. The coefficients of $\varphi_1(z)$ can be easily related to those of $\varphi(z)$ and are given by

$$\begin{aligned} B_0^{(1)} &= B_0 - B_{-1}CB_1 - B_1CB_{-1}, & C &= B_0^{-1}, \\ B_1^{(1)} &= -B_1CB_1, \\ B_{-1}^{(1)} &= -B_{-1}CB_{-1}. \end{aligned}$$

Surprisingly, these are part of the same equations (3) which define cyclic reduction! Cyclic reduction applied to a block tridiagonal Toeplitz matrix generates the coefficients of the Graeffe iteration applied to a matrix Laurent polynomial. Can we deduce nice asymptotic properties from this observation? We provide soon an answer to this question.

With $\varphi_0(z) = \varphi(z)$, define the sequence $\varphi_k(z) = B_{-1}^{(k)}z^{-1} + B_0^{(k)} + B_1^{(k)}z$ by means of

$$\varphi_{k+1}(z^2) = \varphi_k(z)(B_0^{(k)})^{-1}\varphi_k(-z), \quad k = 0, 1, \dots,$$

where we assume that $\det B_0^{(k)} \neq 0$. Moreover, define $\psi_k(z) = \varphi_k(z)^{-1}$. Observe that $B_0^{(k)} = \frac{1}{2}(\varphi_k(-z) + \varphi_k(z))$ so that

$$\begin{aligned} \varphi_{k+1}(z^2) &= \varphi_k(z)2(\varphi_k(-z) + \varphi_k(z))^{-1}\varphi_k(-z) \\ &= 2(\varphi_k(z)^{-1} + \varphi_k(-z)^{-1})^{-1}. \end{aligned}$$

Whence we get

$$\psi_{k+1}(z^2) = \frac{1}{2}(\psi_k(z) + \psi_k(-z)).$$

Thus, $\psi_1(z)$ is the even part of $\psi_0(z)$, $\psi_2(z)$ is the even part of $\psi_1(z)$, and so on. This is a crucial property that, together with the decay of the coefficients of analytic functions, provides very interesting convergence properties of cyclic reduction.

If $\psi(z)$ is analytic in the annulus $\mathbb{A} = \{z \in \mathbb{C} : r_1 < |z| < r_2\}$ then it can be represented as a Laurent series

$$\psi(z) = \sum_{i=-\infty}^{+\infty} z^i H_i, \quad z \in \mathbb{A}.$$

Moreover, for the analyticity of ψ in \mathbb{A} one has, see for instance the book by Henrici [46], that for any $\epsilon > 0$, such that $r_1 + \epsilon < 1$, $r_2 - \epsilon > 1$, and for any norm $\|\cdot\|$ there exists $\theta > 0$ such that

$$\|H_i\| \leq \begin{cases} \theta(r_1 + \epsilon)^i, & i > 0, \\ \theta(r_2 - \epsilon)^i, & i < 0. \end{cases}$$

Thus,

$$\psi_0(z) = \dots + H_{-2}z^{-2} + H_{-1}z^{-1} + H_0 + H_1z^1 + H_2z^2 + \dots$$

$$\psi_1(z) = \dots + H_{-4}z^{-2} + H_{-2}z^{-1} + H_0 + H_2z^1 + H_4z^2 + \dots$$

$$\psi_2(z) = \dots + H_{-8}z^{-2} + H_{-4}z^{-1} + H_0 + H_4z^1 + H_8z^2 + \dots$$

.....

$$\psi_k(z) = \dots + H_{-3 \cdot 2^k}z^3 + H_{-2 \cdot 2^k}z^{-2} + H_{-2^k}z^{-1} + H_0 + H_{2^k}z^1 + H_{2 \cdot 2^k}z^2 + H_{3 \cdot 2^k}z^3 + \dots$$

That is, if $\psi(z)$ is analytic on \mathbb{A} then for any z in a compact set contained in \mathbb{A} the sequence $\psi_k(z)$ converges to H_0 *double exponentially*. Consequently, if $\det H_0 \neq 0$ for any z in any compact set contained in \mathbb{A} the sequence $\varphi_k(z)$ converges to H_0^{-1} *double exponentially*.

Practically, the sequence of block tridiagonal matrices generated by CR converges very quickly to a block diagonal matrix. The speed of convergence is faster the larger the width of the annulus \mathbb{A} . This convergence property makes it easier to solve $n \times n$ block tridiagonal block Toeplitz systems. In fact, it is not needed to perform $\log_2 n$ iteration steps. It is enough to iterate until the Schur complement is numerically block diagonal.

Moreover, convergence of CR enables us to compute any number of components of the solution of an infinite block tridiagonal block Toeplitz system. This goal

is achieved by continuing the iteration until a numerical block diagonal matrix is obtained; a finite number of block components is computed by solving the truncated block diagonal system; back substitution is applied.

We can provide a functional interpretation of CR also in the block Hessenberg case. But for this goal we have to carry out the analysis in the framework of infinite matrices. Therefore we postpone this analysis to the next section on infinite Markov chains.

An important question is to figure out if the function $\psi(z)$ is analytic over some annulus \mathbb{A} . Recall that $\psi(z) = \varphi(z)^{-1}$, and that $\varphi(z)$ is a Laurent polynomial. Thus, if $\det \varphi(z) \neq 0$ for $z \in \mathbb{A}$ then $\psi(z)$ is analytic in \mathbb{A} .

The equation $\det(B_{-1} + zB_0 + z^2B_1) = 0$ plays an important role. Denote ξ_1, \dots, ξ_{2m} the roots of the polynomial $b(z) = \det(B_{-1} + zB_0 + z^2B_1)$, ordered so that $|\xi_i| \leq |\xi_{i+1}|$, where we have added $2m - \deg b(z)$ roots at the infinity if $\deg b(z) < 2m$.

Assume that $|\xi_q| < 1 < |\xi_{q+1}|$ for some integer q . Then $\psi(z)$ is analytic on the open annulus \mathbb{A} of radii $r_1 = |\xi_q|$ and $r_2 = |\xi_{q+1}|$.

In the case of Markov chains the following scenario, depicted in Fig. 10, is encountered according to the feature of the stochastic process:

- positive recurrent: $\xi_m = 1 < \xi_{m+1}$;
- transient: $\xi_m < 1 = \xi_{m+1}$;
- null recurrent: $\xi_m = 1 = \xi_{m+1}$.

4.4.1 Convergence Properties When $1 \in \{\xi_m, \xi_{m+1}\}$

In the case of Markov chains, the annulus \mathbb{A} where $\psi(z)$ is analytic has at least one of its radii equal to 1. Therefore the convergence properties derived from the analyticity of $\psi(z)$ cannot be deduced with the same argument even though CR can be still applied. A simple trick enables us to reduce the problem to the case where the inner part of the annulus \mathbb{A} contains the unit circle. Consider $\tilde{\varphi}(z) := \varphi(\alpha z)$ so that the

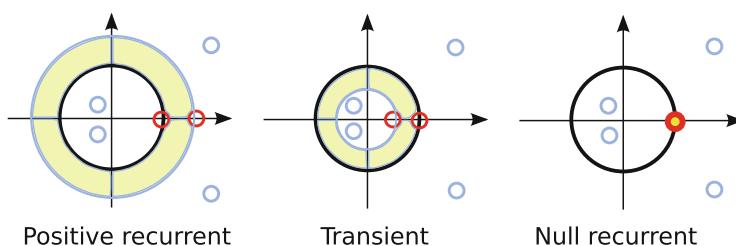


Fig. 10 Location of the zeros of the polynomial $b(z) = \det(B_{-1} + zB_0 + z^2B_1)$ for positive recurrent, transient and null recurrent Markov chains. In black the unit circle, in yellow the annulus of analyticity of $\psi(z) = \varphi(z)^{-1}$

roots of $\det \widetilde{\varphi}(z)$ are $\xi_i \alpha^{-1}$. Choose α so that $\xi_m < \alpha < \xi_{m+1}$, this way the function $\widetilde{\psi}(z) = \widetilde{\varphi}(z)^{-1}$ is analytic in the annulus $\widetilde{\mathbb{A}} = \{z \in \mathbb{C} : \xi_m \alpha^{-1} < |z| < \xi_{m+1} \alpha^{-1}\}$ which contains the unit circle.

With this scaling of the variable one can prove that

- in the positive recurrent case where the analyticity annulus has radii $\xi_m = 1$, $\xi_{m+1} > 1$, the blocks $B_{-1}^{(k)}$ converge to zero with rate $1/d^{2^k}$, for any $1 < d < \xi_{m+1}$, the blocks $B_1^{(k)}$ have a finite nonzero limit;
- in the transient case where the analyticity annulus has radii $\xi_m < 1$, $\xi_{m+1} = 1$, the blocks $B_1^{(k)}$ converge to zero with rate d^{2^k} for any $\xi_m < d < 1$, the blocks $B_{-1}^{(k)}$ have a finite nonzero limit.

However, this trick does not work in the null recurrent case where $\xi_m = \xi_{m+1} = 1$. In this situation, convergence of CR turns to linear with factor $1/2$. In order to restore the quadratic convergence we have to use a more sophisticated technique which will be described next.

Here, we report some convergence results available in the literature, where we denote $\mathbb{A}(r_1, r_2) = \{z \in \mathbb{C} : r_1 < |z| < r_2\}$.

Theorem 5 Assume we are given a function $\varphi(z) = z^{-1}B_1 + B_0 + zB_1$ and positive numbers $r_1 < 1 < r_2$ such that

1. for any $z \in \mathbb{A}(r_1, r_2)$ the matrix $\varphi(z)$ is analytic and nonsingular,
2. the function $\psi(z) = \varphi(z)^{-1}$, analytic in $\mathbb{A}(r_1, r_2)$, is such that $\det H_0 \neq 0$ where $\psi(z) = \sum_{i=-\infty}^{+\infty} z^i H_i$.

Then

1. the sequence $\varphi^{(k)}(z)$ converges uniformly to H_0^{-1} over any compact set in $\mathbb{A}(r_1, r_2)$,
2. for any $\epsilon > 0$ such that $r_1 + \epsilon < 1$ and for any norm there exist constants $c_i > 0$ such that

$$\begin{aligned} \|B_{-1}^{(k)}\| &\leq c_{-1}(r_1 + \epsilon)^{2^k}, \\ \|B_1^{(k)}\| &\leq c_1(r_2 - \epsilon)^{-2^k}, \\ \|B_0^{(k)} - H_0^{-1}\| &\leq c_0 \left(\frac{r_1 + \epsilon}{r_2 - \epsilon} \right)^{2^k}. \end{aligned}$$

Theorem 6 Given $\varphi(z) = z^{-1}B_{-1} + B_0 + zB_1$. If the two matrix equations

$$\begin{aligned} B_{-1} + B_0 X + B_1 X^2 &= 0 \\ B_{-1} Y^2 + B_0 Y + B_1 &= 0 \end{aligned}$$

have solutions X and Y such that $\rho(X) < 1$ and $\rho(Y) < 1$ then $\det H_0 \neq 0$, the roots ξ_i , $i = 1, \dots, 2m$ of $\det \varphi(z)$ are such that $|\xi_m| < 1 < |\xi_{m+1}|$, and $\rho(X) = |\xi_m|$,

$\rho(Y) = 1/|\xi_{m+1}|$. Moreover, $\psi(z) = \varphi(z)^{-1}$ is analytic in $\mathbb{A}(\rho(X), 1/\rho(Y))$ where

$$\psi(z) = \sum_{i=-\infty}^{+\infty} z^i H_i, \quad H_i = \begin{cases} X^{-i} H_0, & i < 0, \\ H_0, & i = 0, \\ H_0 Y^i, & i > 0. \end{cases}$$

In the critical case where $\xi_m = \xi_{m+1}$, Chiang et al. [29] have given the following convergence result

Theorem 7 Let $\varphi(z) = z^{-1}B_{-1} + B_0 + zB_1$ be the function associated with a null recurrent QBD so that its roots ξ_i satisfy the condition $\xi_m = 1 = \xi_{m+1}$. Then cyclic reduction can be applied and there exists a constant γ such that

$$\begin{aligned} \|B_{-1}^{(k)}\| &\leq \gamma 2^{-k}, \\ \|B_0^{(k)} - H_0^{-1}\| &\leq \gamma 2^{-k}, \\ \|B_1^{(k)}\| &\leq \gamma 2^{-k}. \end{aligned}$$

4.5 A Special Case: Non-skip-Free Markov Chain

An interesting problem is the case where the QBD comes from the reblocking of a banded Toeplitz matrix, or more generally from a generalized Hessenberg matrix with the almost Toeplitz structure like in the shortest queue model examined in Sect. 2. In this case, the blocks obtained after reblocking are Toeplitz themselves.

Here the main issue is to exploit the Toeplitz structure of the blocks. This goal can be achieved by relying on the functional interpretation of CR, on the properties of z-circulant matrices introduced in Sect. 3.3.2, and on the properties of the displacement operators introduced in Sect. 3.4.

Assume we are given an $n \times n$ banded Toeplitz matrix, up to some boundary corrections, having $2m + 1$ diagonals. Say, with $m = 2$,

$$\left[\begin{array}{cccccc} \widehat{b}_0 & b_1 & b_2 & & & & \\ \widehat{b}_{-1} & b_0 & b_1 & b_2 & & & \\ b_{-2} & b_{-1} & b_0 & b_1 & b_2 & & \\ b_{-2} & b_{-1} & b_0 & b_1 & b_2 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & & & & \\ b_{-2} & b_{-1} & b_0 & b_1 & b_2 & & \\ b_{-2} & b_{-1} & b_0 & \widetilde{b}_1 & & & \\ b_{-2} & b_{-1} & \widetilde{b}_0 & & & & \end{array} \right].$$

Reblock it into $m \times m$ blocks and get a block tridiagonal matrix with $m \times m$ blocks. The matrix is almost block Toeplitz and its blocks are almost Toeplitz. In our example with $m = 2$ we have

$$\left[\begin{array}{cc|cc|c} \widehat{b}_0 & b_1 & b_2 & & \\ \widehat{b}_{-1} & b_0 & b_1 & b_2 & \\ \hline b_{-2} & b_{-1} & b_0 & b_1 & \ddots \\ b_{-2} & b_{-1} & b_0 & & \ddots \\ \hline & \ddots & \ddots & \ddots & b_2 \\ & & \ddots & \ddots & b_1 & b_2 \\ \hline & & b_{-2} & b_{-1} & b_0 & b_1 \\ & & & b_{-2} & b_{-1} & b_0 \end{array} \right].$$

A very interesting property is that the Laurent polynomial $\varphi(z) = B_{-1}z^{-1} + B_0 + B_1z$ obtained this way is a z -circulant matrix. For instance, for $m = 4$ we have

$$\varphi(z) = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \\ b_{-1} & b_0 & b_1 & b_2 \\ b_{-2} & b_{-1} & b_0 & b_1 \\ b_{-3} & b_{-2} & b_{-1} & b_0 \end{bmatrix} + z \begin{bmatrix} b_4 \\ b_3 & b_4 \\ b_2 & b_3 & b_4 \\ b_1 & b_2 & b_3 & b_4 \end{bmatrix} + z^{-1} \begin{bmatrix} b_{-4} & b_{-3} & b_{-2} & b_{-1} \\ b_{-4} & b_{-3} & b_{-2} & \\ b_{-4} & b_{-3} & & \\ & & & b_{-4} \end{bmatrix}.$$

In fact, multiplying the upper triangular part of $\varphi(z)$ by z we get a circulant matrix.

Now, recall that z -circulant matrices form a matrix algebra, i.e., they are a linear space closed under multiplication and inversion. Therefore the rational function $\psi(z) = \varphi(z)^{-1}$ is z -circulant, in particular, $\psi(z)$ is Toeplitz since z -circulant matrices are also Toeplitz. Since Toeplitz matrices form a vector space, then $\psi_1(z^2) = \frac{1}{2}(\psi(z) + \psi(-z))$ is Toeplitz as well as $\psi_2(z), \psi_3(z), \dots$

Recall that the inverse of a Toeplitz matrix is Toeplitz-like in view of the Gohberg-Semencul formula, or in view of the properties of the displacement operator that we have seen in Sect. 3.4. Therefore $\varphi_k(z) = \psi_k(z)^{-1}$ is Toeplitz-like with displacement rank at most 2 for any value of $z \neq 0$. It follows that also the coefficients of $\varphi_k(z)$ are Toeplitz-like. Moreover, the relations between the matrix coefficients B_{-1}, B_0, B_1 at two subsequent steps of CR can be rewritten in terms of the displacement generators.

We have just to play with the properties of the displacement operators in order to state explicitly these relations. In fact, by using displacement operators we can

prove that

$$\Delta(\varphi^{(k)}(z)) = -z^{-1}\varphi^{(k)}(z) (\mathbf{e}_n \mathbf{e}_n^T \psi^{(k)}(z) Z^T - Z^T \psi^{(k)}(z) \mathbf{e}_1 \mathbf{e}_1^T) \varphi^{(k)}(z),$$

where $\Delta(X) = XZ^T - Z^T X$, $Z = \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}$. This property implies that

$$\Delta(B_{-1}^{(k)}) = \mathbf{a}_{-1}^{(k)} \mathbf{u}_{-1}^{(k)T} - \mathbf{v}_{-1}^{(k)} \mathbf{c}_{-1}^{(k)T},$$

$$\Delta(B_0^{(k)}) = \mathbf{a}_{-1}^{(k)} \mathbf{u}_0^{(k)T} + \mathbf{a}_0^{(k)} \mathbf{u}_{-1}^{(k)T} - \mathbf{v}_{-1}^{(k)} \mathbf{c}_0^{(k)T} - \mathbf{v}_0^{(k)} \mathbf{c}_{-1}^{(k)T},$$

$$\Delta(B_1^{(k)}) = \mathbf{r}_1^{(k)} \mathbf{u}_0^{(k)T} - \mathbf{v}_0^{(k)} \hat{\mathbf{c}}^{(k)T},$$

where the vectors $\mathbf{a}_{-1}^{(k)}, \mathbf{a}_0^{(k)}, \mathbf{u}_0^{(k)}, \mathbf{u}_{-1}^{(k)}, \mathbf{c}_0^{(k)}, \mathbf{c}_{-1}^{(k)}, \mathbf{r}^{(k)}, \hat{\mathbf{c}}^{(k)}$ can be updated by suitable formulas.

Moreover, the matrices $B_{-1}^{(k)}, B_0^{(k)}, B_1^{(k)}$ can be represented as Toeplitz-like matrices through their displacement generators.

Relying on these properties, we find that the computation of the Schur complement has the asymptotic cost of $t_m + m \log m$ ops, where t_m is the cost of solving an $m \times m$ Toeplitz-like system, and $m \log m$ is the cost of multiplying a Toeplitz-like matrix and a vector. One step of back substitution stage can be performed in $O(nm \log m)$ ops, that is, $O(n)$ multiplications of $m \times m$ Toeplitz-like matrices and vectors. The overall asymptotic cost $C(n, m)$ of this computation is given by $C(m, n) = t_m \log n + nm \log m$. Recall that, according to the algorithm used, $t_m \in \{m^2, m \log^2 m, m \log m\}$.

The same algorithm applies to solving a *non-homogeneous* $n \times n$ banded Toeplitz system with $2m+1$ diagonals. The cost is $m \log^2 m \log(n/m)$ for the LU factorization generated by CR, and $O(n \log m)$ for the back substitution.

4.6 A Special Case: QBD with Tridiagonal Blocks

A challenging problem is to solve efficiently a block tridiagonal block Toeplitz system where the blocks are tridiagonal, like in the versions of the bidimensional random walk or in the case of the Jackson tandem queue described in Sect. 2. A similar challenge concerns solving block tridiagonal or block Hessenberg Toeplitz systems where the blocks are banded matrices, not necessarily Toeplitz. Cyclic reduction can be applied once again but the initial band structure of the blocks apparently is destroyed during the CR iterations.

The analysis of this problem is still work in place, the results obtained so far are very promising. We give just an outline of the main properties.

Consider $m \times m$ blocks B_i , $i = -1, 0, 1$ which are tridiagonal, that is, $B_i = \text{trid}_m(b_{-1}^{(i)}, b_0^{(i)}, b_1^{(i)})$. Denote $B_{-1}^{(k)}, B_0^{(k)}, B_1^{(k)}$ the blocks generated after k steps of CR. Recall that, denoting $C^{(k)} = (B_0^{(k)})^{-1}$, we have

$$\begin{aligned} B_0^{(k+1)} &= B_0^{(k)} - B_{-1}^{(k)} C^{(k)} B_1^{(k)} - B_1^{(k)} C^{(k)} B_{-1}^{(k)}, \\ B_1^{(k+1)} &= -B_1^{(k)} C^{(k)} B_1^{(k)}, \quad B_{-1}^{(k+1)} = -B_{-1}^{(k)} C^{(k)} B_{-1}^{(k)} \end{aligned}$$

for $k = 0, 1, \dots$, where $B_i^{(0)} = B_i$, $i = -1, 0, 1$.

Indeed, after k iterations, the tridiagonal structure of B_{-1}, B_0, B_1 is lost, and unfortunately the more general quasi-separable structure is not preserved. In fact the rank of the off-diagonal submatrices of $B_i^{(k)}$, that is the submatrices contained in the upper or in the lower triangular part of $B_i^{(k)}$, grows as 2^k up to saturation. However, from the numerical experiments we discover that the “numerical rank” does not grow much.

Given $t > 1$ and $\gamma, \ell > 0$, consider the class $\mathcal{F}_m(t, \gamma, \ell)$ of $m \times m$ matrix functions $\varphi(z) = z^{-1}B_{-1} + B_0 + zB_1$, such that

- B_{-1}, B_0, B_1 are $m \times m$ tridiagonal matrices such that $\|B_i\| \leq \ell$;
- $\varphi(z)$ is nonsingular in the annulus $\mathbb{A} = \{z \in \mathbb{C} : t^{-1} \leq |z| \leq t\}$;
- $\|\varphi(z)^{-1}\| \leq \gamma$ for $z \in \mathbb{A}$, moreover the entries of B_{-1}, B_1 and the off-diagonal entries of B_0 are nonpositive.

We can prove the following

Theorem 8 *Under the above assumptions, there exists a constant θ depending on t, γ and ℓ such that for any m and for any function $\varphi(z) \in \mathcal{F}_m(t, \gamma, \ell)$ the singular values σ_i of any off-diagonal submatrix of $B_{-1}^{(k)}, B_0^{(k)}, B_1^{(k)}$ are such that $\sigma_i \leq \theta t^{-\frac{1}{2}i}$.*

In other words, the singular values have an *exponential decay* depending on the width of the analyticity domain of $\varphi(z)$. In Fig. 11 we report the graph of the largest singular values of a submatrix of size 599×800 contained in the upper triangular part of $B_0^{(k)}$ for $k = 1, 2, \dots, 10$. Here $\varphi(z)$ has been randomly chosen tridiagonal and irreducible, of size $m = 1600$, in such a way that $I - \varphi(1)$ is stochastic. Even though the number of singular values grows at each step of CR, the number of singular values above the machine precision is bounded by a moderate constant.

Cyclic reduction has been implemented by using the package H2Lib, a library for hierarchical matrices, of Börm et al. [19], and run with some test problems. In Fig. 12 we report the CPU time of cyclic reduction implemented without exploiting the rank structure (blue line), and implemented by using the rank structure with two different values of the cut level ϵ , (red and green lines). In Table 1 we report the CPU time and the residual errors in the solution computed with the different implementation of CR, for different sizes of the blocks.

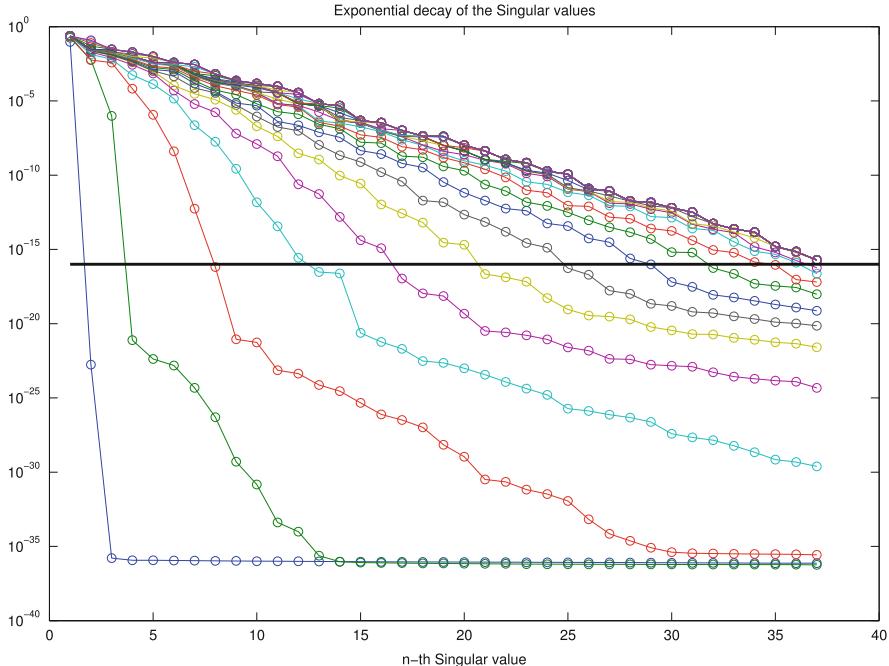


Fig. 11 Log-scale plot of the largest singular values of a submatrix of $B_0^{(k)}$ of size 799×800 contained in the upper triangular part at each step of cyclic reduction for $m = 1600$. The number of nonzero singular values grows at each step. However, the number of the singular values above the machine precision seems to be bounded by a moderate constant. The machine precision is indicated by a *horizontal line*. The singular values tend to be aligned along a straight line, this shows their exponential decay

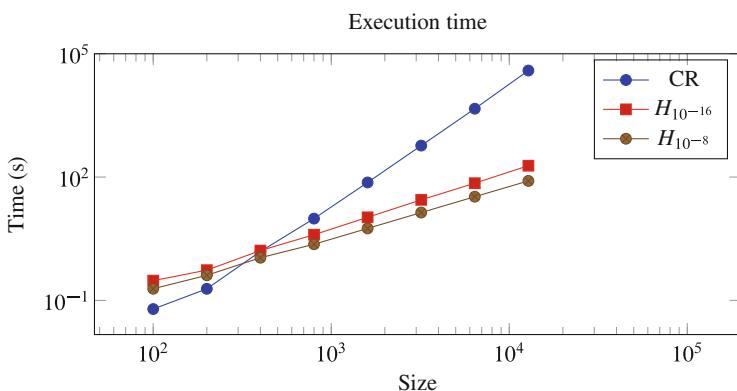


Fig. 12 CPU time of cyclic reduction implemented without exploiting the rank structure (blue line), and implemented by using the rank structure with two different values of the cut level ϵ , (red and green lines)

Table 1 Residual errors in the solution obtained with different implementations of CR

Size	CR	$H_{10^{-16}}$			$H_{10^{-12}}$			$H_{10^{-8}}$		
		Time (s)	Residue	Time (s)	Residue	Time (s)	Residue	Time (s)	Residue	Time (s)
100	$6.04e-02$	$1.91e-16$	$2.21e-01$	$1.79e-15$	$2.04e-01$	$8.26e-14$	$1.92e-01$	$7.40e-10$		
200	$1.88e-01$	$2.51e-16$	$5.78e-01$	$1.39e-14$	$5.03e-01$	$1.01e-13$	$4.29e-01$	$2.29e-09$		
400	$1.61e+01$	$2.00e-16$	$3.32e+00$	$1.41e-14$	$2.60e+00$	$1.33e-13$	$1.98e+00$	$1.99e-09$		
800	$2.63e+01$	$2.74e-16$	$4.55e+00$	$1.94e-14$	$3.49e+00$	$2.71e-13$	$2.63e+00$	$2.69e-09$		
1600	$8.12e+01$	$3.82e-12$	$1.18e+01$	$3.82e-12$	$8.78e+00$	$3.82e-12$	$6.24e+00$	$3.39e-09$		
3200	$6.35e+02$	$5.46e-08$	$3.12e+01$	$5.46e-08$	$2.21e+01$	$5.46e-08$	$1.51e+01$	$5.43e-08$		
6400	$5.03e+03$	$3.89e-08$	$7.83e+01$	$3.89e-08$	$5.38e+01$	$3.89e-08$	$3.58e+01$	$3.87e-08$		
12,800	$4.06e+04$	$1.99e-08$	$1.94e+02$	$1.99e-08$	$1.29e+02$	$1.99e-08$	$8.37e+01$	$1.97e-08$		

5 Algorithms for Structured Markov Chains: The Infinite Case

In this section we deal with the more difficult situation of Markov chains endowed with an infinite number of states. In this case the transition matrix P is infinite and we are interested in computing a finite number of components of the vector π . Cases of interest are M/G/1-type Markov chains where the transition matrix P is block upper Hessenberg and almost block Toeplitz, G/M/1-type Markov chains where P is block lower Hessenberg and almost block Toeplitz, and QBD processes characterized by a block tridiagonal almost block Toeplitz matrix.

As usual, m denotes the size of the blocks and the infinite vector π is partitioned into an infinite number of subvectors π_k , $k \in \mathbb{N}$ of length m . More precisely, we set

$$\pi^T = [\pi_0^T, \pi_1^T, \dots], \quad \pi_i^T = [\pi_1^{(i)}, \dots, \pi_m^{(i)}],$$

with $\pi \geq 0$, $\pi^T e = 1$, $e = [1, 1, \dots]^T$ and $\pi^T(I - P) = 0$, where $P \geq 0$, $Pe = e$. Thus, our problem turns into computing a finite number of block components π_0, \dots, π_k of π , for a given integer k .

Here, the transition matrix P falls in one of the three classes

- M/G/1-type, where

$$P = \begin{bmatrix} \widehat{A}_0 & \widehat{A}_1 & \widehat{A}_2 & \dots & \dots \\ A_{-1} & A_0 & A_1 & A_2 & \dots \\ & A_{-1} & A_0 & A_1 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix},$$

with $A_i, \widehat{A}_i \geq 0$, and $\sum_{i=-1}^{\infty} A_i, \sum_{i=0}^{\infty} \widehat{A}_i$ stochastic and irreducible.

- G/M/1-type, where

$$P = \begin{bmatrix} \widehat{A}_0 & A_1 & & & \\ \widehat{A}_{-1} & A_0 & A_1 & & \\ \widehat{A}_{-2} & A_{-1} & A_0 & A_1 & \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix},$$

with $A_i, \widehat{A}_i \geq 0$, and $\sum_{i=-1}^{\infty} A_{-i}, \widehat{A}_{-k} + \sum_{i=-1}^{k-1} A_{-i}$ stochastic and irreducible.

- QBD, where

$$P = \begin{bmatrix} \widehat{A}_0 & \widehat{A}_1 \\ \widehat{A}_{-1} & A_0 & A_1 \\ & A_{-1} & A_0 & A_1 \\ & & \ddots & \ddots & \ddots \end{bmatrix},$$

with $A_i, \widehat{A}_i \geq 0$, and $A_{-1} + A_0 + A_1, \widehat{A}_0 + \widehat{A}_1, \widehat{A}_{-1} + A_0 + A_1$, stochastic and irreducible.

Other cases like Non-Skip-Free Markov chains can be reduced to the previous classes by means of reblocking as in the shortest queue model examined in Sect. 2.

In this context, a fundamental role is played by the UL factorization (if it exists) of the infinite block Hessenberg block Toeplitz matrix H forming the Toeplitz part of $\mathcal{B} = I - P$. More precisely, denoting for simplicity $B_0 = I - A_0$, and $B_i = -A_i$ for $i \neq 0$, we are interested in the following matrix factorizations of the UL type:

$$\begin{bmatrix} B_0 & B_1 & B_2 & \dots & \dots \\ B_{-1} & B_0 & B_1 & B_2 & \ddots \\ & B_{-1} & B_0 & B_1 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} U_0 & U_1 & U_2 & \dots \\ & U_0 & U_1 & U_2 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} I & & & \\ -G & I & & \\ -G & & I & \\ & & & \ddots \end{bmatrix}, \quad \text{M/G/1}$$

$$\begin{bmatrix} B_0 & B_1 \\ B_{-1} & B_0 & B_1 \\ B_{-2} & B_{-1} & B_0 & B_1 \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} I & -R \\ & I & -R \\ & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} L_0 \\ L_1 & L_0 \\ L_2 & L_1 & L_0 \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}, \quad \text{G/M/1}$$

where $\det U_0, \det L_0 \neq 0$, $\rho(G), \rho(R) \leq 1$. In the case of a QBD, the M/G/1-type and G/M/1-type structures can be combined together yielding

$$\begin{bmatrix} B_0 & B_1 \\ B_{-1} & B_0 & B_1 \\ B_{-2} & B_0 & B_1 \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} I & -R \\ & I & -R \\ & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} U_0 & & \\ & U_0 & \\ & & U_0 & \\ & & & \ddots \end{bmatrix} \begin{bmatrix} I & & \\ -G & I & \\ G & & I \\ & & & \ddots \end{bmatrix},$$

where $\det U_0 \neq 0$, $\rho(R) \leq 1$, $\rho(G) \leq 1$.

We recall from Sect. 3.2 that the UL factorization has a functional counterpart which is the weak canonical (Wiener-Hopf) factorization of the matrix function $B(x) = \sum_{i=-\infty}^{+\infty} z^i B_i$ that is,

$$B(z) = U(z)L(z^{-1}),$$

where $B(z)$, $U(z) = \sum_{i=0}^{\infty} z^i U_i$, $L(z^{-1}) = \sum_{i=0}^{\infty} z^{-i} L_{-i}$ belong to the Wiener class, i.e., $\sum_i |B_i|$, $\sum_i |L_i|$, $\sum_i |U_i|$ are finite, and $\det U(z) \neq 0$, $\det L(z) \neq 0$ for $|z| \geq 1$.

Observe that, if $\rho(G) < 1$ then $\lim_k G^k = 0$, while if $\rho(G) = 1$ and 1 is the only eigenvalue of modulus 1 and is simple, then $\lim_k G^k = \mathbf{u}\mathbf{v}^T$, where \mathbf{u} and \mathbf{v}^T are right and left eigenvectors of G , respectively, corresponding to the eigenvalue 1, normalized so that $\mathbf{v}^T \mathbf{u} = 1$. This way, $|G^k|$ is bounded from above by a constant independent of k . A matrix G with this property is said *power bounded*. For the canonical factorizations of M/G/1-type, G/M/1-type and Q/B/D Markov chains the power boundedness of matrices G and R is generally satisfied.

Observe that the L factor in the canonical factorization of an M/G/1-type Markov chain has inverse given by the block lower triangular block Toeplitz matrix with entries G^{i-j} . The power boundedness of G implies that these blocks are bounded.

5.1 M/G/1-Type Markov Chains

Consider the problem of computing $\boldsymbol{\pi}^T = [\boldsymbol{\pi}_0^T, \boldsymbol{\pi}_1^T, \dots]$ such that

$$[\boldsymbol{\pi}_0^T, \boldsymbol{\pi}_1^T, \dots] \mathcal{B} = 0, \quad \mathcal{B} = \begin{bmatrix} \widehat{B}_0 & \widehat{B}_1 & \widehat{B}_2 & \dots & \dots \\ B_{-1} & B_0 & B_1 & B_2 & \dots \\ & B_{-1} & B_0 & B_1 & \ddots \\ & & B_{-1} & B_0 & \ddots \\ & & & \ddots & \ddots \end{bmatrix}.$$

The above equation splits into

$$\boldsymbol{\pi}_0^T \widehat{B}_0 + \boldsymbol{\pi}_1^T B_{-1} = 0,$$

$$[\boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \dots] \begin{bmatrix} B_0 & B_1 & B_2 & \dots \\ B_{-1} & B_0 & B_1 & \ddots \\ & B_{-1} & B_0 & \ddots \\ & & \ddots & \ddots \end{bmatrix} = -\boldsymbol{\pi}_0^T [\widehat{B}_1, \widehat{B}_2, \dots].$$

Assume that the vector $\boldsymbol{\pi}_0$ is known. Later on we will give a way to compute $\boldsymbol{\pi}_0$. Under this assumption, to compute the components $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \dots$, it is enough to solve the infinite block Hessenberg block Toeplitz system

$$[\boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \dots] \mathcal{H} = -\boldsymbol{\pi}_0^T [\widehat{B}_1, \widehat{B}_2, \dots], \quad \mathcal{H} = \begin{bmatrix} B_0 & B_1 & B_2 & \dots \\ B_{-1} & B_0 & B_1 & \ddots \\ B_{-1} & B_0 & B_1 & \ddots \\ & \ddots & \ddots & \ddots \end{bmatrix}.$$

Now, assume that there exists the UL factorization of \mathcal{H}

$$\mathcal{H} = \begin{bmatrix} U_0 & U_1 & U_2 & \dots \\ & U_0 & U_1 & U_2 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} I & & & \\ -G & I & & \\ & -G & I & \\ & & \ddots & \ddots \end{bmatrix} =: \mathcal{U}\mathcal{L},$$

where $\det U_0 \neq 0$ and G is power bounded. Then the infinite block Hessenberg system turns into $[\boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \dots] \mathcal{U}\mathcal{L} = -\boldsymbol{\pi}_0^T [\hat{B}_1, \hat{B}_2, \dots]$ which formally reduces to

$$[\boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \dots] \mathcal{U} = -\boldsymbol{\pi}_0^T [\hat{B}_1, \hat{B}_2, \dots] \mathcal{L}^{-1}, \quad \mathcal{L}^{-1} = \begin{bmatrix} I & & & \\ G & I & & \\ G^2 & G & I & \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Observe that the right-hand side $\mathbf{b}^T = -\boldsymbol{\pi}_0^T [\hat{B}_1, \hat{B}_2, \dots] \mathcal{L}^{-1}$ of the above equation is well defined since the series $\mathbf{b}_k = -\boldsymbol{\pi}_0^T \sum_{i=k-1}^{\infty} \hat{B}_{i+1} G^{i-k+1}$, $k = 1, 2, \dots$, is convergent, being G power bounded and $-\sum_{i=0}^{\infty} \hat{B}_i = \sum_{i=0}^{\infty} |\hat{B}_i|$ convergent.

Thus, if the UL factorization of \mathcal{H} exists the problem of computing $\boldsymbol{\pi}_0, \dots, \boldsymbol{\pi}_k$ is reduced to computing

- the UL factorization of \mathcal{H} ;
- $\boldsymbol{\pi}_0$;
- the first $k-1$ block components $\mathbf{b}_1^T, \dots, \mathbf{b}_{k-1}^T$ of $\mathbf{b}^T = -\boldsymbol{\pi}_0^T [\hat{B}_1, \hat{B}_2, \dots] \mathcal{L}^{-1}$;
- the solution of the block triangular block Toeplitz system $[\boldsymbol{\pi}_1^T, \dots, \boldsymbol{\pi}_k^T] \mathcal{U}_k = [\mathbf{b}_1^T, \dots, \mathbf{b}_k^T]$, where \mathcal{U}_k is the $k \times k$ leading principal submatrix of \mathcal{U} .

To complete the algorithm, it remains to compute the vector $\boldsymbol{\pi}_0$ and the UL factorization. The vector $\boldsymbol{\pi}_0$ can be computed as follows. The condition

$$[\boldsymbol{\pi}_0, \boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \dots] \left[\begin{array}{c|cccc} \hat{B}_0 & \hat{B}_1 & \hat{B}_2 & \dots & \dots \\ \hline B_{-1} & B_0 & B_1 & B_2 & \ddots \\ & B_{-1} & B_0 & B_1 & \ddots \\ & & \ddots & \ddots & \ddots \end{array} \right] = 0$$

is rewritten as

$$[\boldsymbol{\pi}_0^T, \boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \dots] \left[\begin{array}{c|c} \hat{B}_0 & [\hat{B}_1, \hat{B}_2, \dots] \\ \hline \tilde{B}_{-1} & \mathcal{U}\mathcal{L} \end{array} \right] = 0,$$

where \tilde{B}_{-1} is the block column vector with first component B_{-1} and with null components elsewhere. Multiplying on the right the above equation by $\text{diag}(I, \mathcal{L}^{-1})$ yields

$$[\boldsymbol{\pi}_0^T, \boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \dots] \left[\begin{array}{c|c} \widehat{B}_0 & [\widehat{B}_1, \widehat{B}_2, \dots] \mathcal{L}^{-1} \\ \hline \widehat{B}_{-1} & \mathcal{U} \end{array} \right] = 0,$$

that is

$$[\boldsymbol{\pi}_0^T, \boldsymbol{\pi}_1^T, \boldsymbol{\pi}_2^T, \dots] \left[\begin{array}{c|cccccc} \widehat{B}_0 & [\widehat{B}_1 & \widehat{B}_2 & \dots] \mathcal{L}^{-1} & & & & \\ \hline B_{-1} & U_0 & U_1 & U_2 & \dots & & & \\ 0 & & U_0 & U_1 & \ddots & & & \\ \vdots & & & & \ddots & \ddots & & \end{array} \right] = 0.$$

Thus the first two equations of the latter system yield

$$[\boldsymbol{\pi}_0^T, \boldsymbol{\pi}_1^T] \left[\begin{array}{c|c} \widehat{B}_0 & B_1^* \\ \hline B_{-1} & U_0 \end{array} \right] = 0, \quad B_1^* = \sum_{i=0}^{\infty} \widehat{B}_{i+1} G^i,$$

which provide an algorithm for computing $\boldsymbol{\pi}_0$. Observe once again that the boundedness of $\sum_{i=0}^{\infty} |\widehat{B}_i|$ and the power boundedness of G imply the convergence of the series B_1^* , so that the algorithm is consistent. Concerning the computation of the UL factorization, We will postpone this issue to a next section. Here we take a look at the complexity analysis of this algorithmic approach. In the M/G/1 case, the computation of $\boldsymbol{\pi}_i$, $i = 0, 1, \dots, k$ is reduced to

- computing the UL factorization of \mathcal{H} , more precisely computing the matrices G and U_0 (we will see this next);
- computing $\boldsymbol{\pi}_0$;
- computing the first k block components of $\mathbf{b}^T = -\boldsymbol{\pi}_0^T [\widehat{B}_1, \widehat{B}_2, \dots] \mathcal{L}^{-1}$;
- solving the block triangular Toeplitz system $[\boldsymbol{\pi}_1^T, \dots, \boldsymbol{\pi}_k^T] \mathcal{U}_k = [\mathbf{b}_1^T, \dots, \mathbf{b}_k^T]$.

Relying on the Toeplitz matrix technology, we may easily perform an asymptotic complexity analysis of the above computation. In fact, the computation of $\boldsymbol{\pi}_0$ costs $O(m^3 d)$ ops, where d is an integer such that $\sum_{i=d}^{\infty} \|\widehat{B}_i\| < \epsilon$, where ϵ is a positive constant related to the approximation error; the computation of \mathbf{b} costs $O(m^3 d)$ ops; the solution of the block triangular Toeplitz system costs $O(m^3 k + m^2 k \log k)$ ops.

Thus, the overall asymptotic cost is given by the cost of computing the UL factorization plus $O(m^3(k + d) + m^2 k \log k)$ ops.

5.2 G/M/1-Type Markov Chains

Concerning G/M/1-type Markov chains we can follow a similar approach based on the UL factorization of a block lower Hessenberg block Toeplitz matrix. The problem is given by

$$[\boldsymbol{\pi}_0^T, \boldsymbol{\pi}_1^T, \dots] \begin{bmatrix} \widehat{B}_0 & B_1 & & \\ \widehat{B}_{-1} & B_0 & B_1 & \\ \widehat{B}_{-2} & B_{-1} & B_0 & B_1 \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix} = 0.$$

Denote by \mathcal{H} the principal submatrix forming the Toeplitz part of the above matrix and consider the UL factorization of \mathcal{H} , provided it exists,

$$\mathcal{H} = \begin{bmatrix} B_0 & B_1 & & \\ B_{-1} & B_0 & B_1 & \\ B_{-2} & B_{-1} & B_0 & B_1 \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} I & -R & & \\ & I & -R & \\ & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} L_0 & & & \\ L_1 & L_0 & & \\ L_2 & L_1 & L_0 & \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix},$$

where $\rho(R) \leq 1$, R is power bounded, and $\det L_0 \neq 0$. Multiply the above equation to the left by $[R, R^2, R^3, \dots]$ and, since $RL_0 = -B_1$, find that

$$[R, R^2, R^3, \dots] \mathcal{H} = [RL_0, 0, 0, \dots] = [-B_1, 0, \dots].$$

Now, multiply $I - P$ to the left by $[I, R, R^2, \dots]$ and get

$$[I, R, R^2, \dots] \begin{bmatrix} \widehat{B}_0 & B_1 & & \\ \widehat{B}_{-1} & B_0 & B_1 & \\ \widehat{B}_{-2} & B_{-1} & B_0 & B_1 \\ \vdots & \vdots & \ddots & \ddots & \ddots \end{bmatrix} = \left[\sum_{i=0}^{\infty} R^i \widehat{B}_{-i}, 0, 0, \dots \right].$$

Observe that, since R is power bounded, $\widehat{R}_i \geq 0$ and the series $\sum_{i=0}^{\infty} \widehat{B}_{-i}$ is convergent, then the matrix $\sum_{i=0}^{\infty} R^i \widehat{B}_{-i}$ exists and is finite. Moreover, since $(I - P)\mathbf{e} = 0$ then $[I, R, R^2, \dots](I - P)\mathbf{e} = 0$, that is,

$$\sum_{i=0}^{\infty} R^i \widehat{B}_{-i} [1, 1, \dots, 1]^T = 0,$$

i.e., $\sum_{i=0}^{\infty} R^i \widehat{B}_{-i}$ is singular and there exists $\boldsymbol{\pi}_0$ such that $\boldsymbol{\pi}_0^T \sum_{i=0}^{\infty} R^i \widehat{B}_{-i} = 0$. We deduce that $\boldsymbol{\pi}_0^T [I, R, R^2, R^3, \dots](I - P) = 0$, that is, $\boldsymbol{\pi}_i^T = \boldsymbol{\pi}_0^T R^i$. One can prove that, in the positive recurrent case where $\rho(R) < 1$, if $\boldsymbol{\pi}_0$ is normalized so that $\boldsymbol{\pi}_0^T (I - R)^{-1} [1, \dots, 1]^T = 1$ then $\|\boldsymbol{\pi}\|_1 = 1$.

For a G/M/1-type Markov chain, the computation of any number k of components of $\boldsymbol{\pi}$ is reduced to

- computing the matrix R in the UL factorization of \mathcal{H} ;
- computing the matrix $\sum_{i=0}^{\infty} R^i \widehat{B}_{-i}$;
- solving the $m \times m$ system $\boldsymbol{\pi}_0^T (\sum_{i=0}^{\infty} R^i \widehat{B}_{-i}) = 0$;
- computing $\boldsymbol{\pi}_i^T = \boldsymbol{\pi}_0^T R^i$, $i = 1, \dots, k$.

The overall cost is given by the cost of computing the UL factorization plus $dm^3 + m^2k$ ops, where d is such that $\|\sum_{i=d}^{\infty} \widehat{B}_i\| \leq \epsilon$.

The largest computational cost is the one of computing the UL factorization of the block lower Hessenberg block Toeplitz matrix \mathcal{H} , or equivalently, of the block upper Hessenberg block Toeplitz matrix \mathcal{H}^T .

Our next efforts are addressed to investigate this kind of factorization of block Hessenberg block Toeplitz matrices. More precisely, our next goals are

- prove that there exists the UL factorization of \mathcal{H} ;
- design an algorithm for its computation;
- prove that G and R are power bounded;
- extend the approach to more general situations.

Before doing that, we synthesize the case of QBD processes.

5.3 QBD Markov Chains

In a QBD Markov chain the transition matrix is block tridiagonal with the almost block Toeplitz structure so that the problem is reduced to computing the components of the infinite vector $\boldsymbol{\pi}$ such that

$$\boldsymbol{\pi}^T \begin{bmatrix} \widehat{B}_0 & B_1 \\ B_{-1} & B_0 & B_1 \\ & B_{-1} & B_0 & B_1 \\ & & \ddots & \ddots & \ddots \end{bmatrix} = 0, \quad \mathbf{e}^T \boldsymbol{\pi} = 1.$$

This way, we can treat the problem either as an M/G/1-type Markov chain or as a G/M/1-type Markov chain. The second approach seems better suited and more convenient from the algorithmic point of view. Both approaches rely on the UL factorization of the matrix \mathcal{H} which in this case is block tridiagonal block Toeplitz:

$$\mathcal{H} = \begin{bmatrix} B_0 & B_1 & & \\ B_{-1} & B_0 & B_1 & \\ & \ddots & \ddots & \ddots \end{bmatrix}.$$

This factorization takes the form

$$\mathcal{H} = \begin{bmatrix} I - R & & \\ & I - R & \\ & & \ddots \end{bmatrix} \begin{bmatrix} U & & \\ & U & \\ & & \ddots \end{bmatrix} \begin{bmatrix} I - G & & \\ & I - G & \\ & & \ddots \end{bmatrix}. \quad (4)$$

By following the same arguments of the previous section on G/M/1-type Markov chains, one finds that

$$\boldsymbol{\pi}_0^T(\widehat{B}_0 + RB_{-1}) = 0, \quad \boldsymbol{\pi}_i^T = \boldsymbol{\pi}_0^T R^i, \quad i \geq 0,$$

where the vector $\boldsymbol{\pi}_0$ is normalized so that $\boldsymbol{\pi}_0^T(I - R)^{-1}[1, \dots, 1] = 1$. Recall that in the positive recurrent case one has $\rho(R) < 1$ so that the matrix $I - R$ is invertible. The above equations reduce the computation of $\boldsymbol{\pi}_0, \dots, \boldsymbol{\pi}_k$ to computing the matrix R such that the UL factorization (4) holds, the matrix $\widehat{B}_0 + RB_{-1}$ and a vector $\boldsymbol{\pi}_0^T$ in its left kernel normalized so that $\boldsymbol{\pi}_0^T(I - R)^{-1}[1, \dots, 1]^T = 1$, then the vector-matrix products $\boldsymbol{\pi}_i^T = \boldsymbol{\pi}_{i-1}^T R$ for $i = 1, \dots, k-1$. The most expensive computational effort is computing the matrix R . This issue is treated in the next section.

5.4 Computing Matrices G and R : The Weak Canonical Factorization

In this section we show that there exist matrices G and R which are power bounded and provide the UL factorization of the block upper (lower) Hessenberg block Toeplitz matrices modeling M/G/1-type (G/M/1-type) Markov chains and QBD processes. That is, we show that in the framework of Markov chains there exists the weak canonical factorization of the matrix function $B(z)$ associated with the block Hessenberg block Toeplitz matrix \mathcal{H} .

Then we will provide algorithms for the computation of G and R .

Consider the M/G/1-type case and recall that the block UL factorization of an infinite block Hessenberg block Toeplitz matrix $\mathcal{H} = (B_{i-j})$ can be formally rewritten in matrix power series form as

$$B(z) := \sum_{i=-1}^{\infty} B_i z^i = \left(\sum_{i=0}^{\infty} U_i z^i \right) (I - z^{-1} G) =: U(z)L(z).$$

Taking determinants on both sides of the factorization $B(z) = U(z)L(z)$ yields $\det B(z) = \det U(z) \det L(z)$ so that, if ξ is a zero of $\det B(z)$, then either $\det L(\xi) = 0$ or $\det U(\xi) = 0$, or both determinants are zero.

Since $U(z)$ is nonsingular for $|z| \leq 1$ and $L(z)$ is nonsingular for $|z| \geq 1$ then, ξ is a zero of $\det L(z)$ if $|\xi| < 1$, while ξ is zero of $\det U(z)$ if $|\xi| > 1$.

In the case of an M/G/1-type Markov chain, where $L(z) = I - z^{-1}G$, the zeros of $\det L(z)$ are the zeros of $\det(zI - G)$, that is the eigenvalues of G . Therefore a necessary condition for the existence of a canonical factorization in the M/G/1-type case is that $\det B(z)$ has exactly m zeros of modulus at most 1.

In the framework of Markov chains the $m \times m$ matrix function $B(z)$, of which we are looking for the weak canonical factorization, is in the Wiener class since $B_0 = I - A_0$, $B_i = -A_i$ for $i \neq 0$, and $\sum_{i=-1}^{\infty} |A_i| = \sum_{i=1}^{\infty} A_i$ exists finite since $A_i \geq 0$ and $\sum_{i=1}^{\infty} A_i$ is stochastic.

Moreover, assuming for simplicity that $B(z)$ is analytic in an open disk containing the closed unit disk, it can be proved that $\det(B(z))$ has zeros ξ_1, ξ_2, \dots , ordered such that $|\xi_i| \leq |\xi_{i+1}|$, where

$$\begin{aligned}\xi_m &= 1 < \xi_{m+1} && \text{for positive recurrent processes,} \\ \xi_m &< \xi_{m+1} = 1 && \text{for transient processes,} \\ \xi_m &= \xi_{m+1} = 1 && \text{for null recurrent processes,}\end{aligned}\tag{5}$$

and the zero $\xi_m = 1$ in the positive recurrent case, or $\xi_{m+1} = 1$ in the transient case, is simple. This way, the matrix G is such that $\rho(G) = 1$ and $\xi_m = 1$ is a simple eigenvalue, in the positive and in the null recurrent case, or $\rho(G) < 1$ in the transient case. Thus, in all the cases the matrix G is power bounded.

Assume that there exists the UL factorization of \mathcal{H} :

$$\mathcal{H} = \begin{bmatrix} U_0 & U_1 & U_2 & \dots \\ & U_0 & U_1 & U_2 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} I & & & \\ -G & I & & \\ -G & -G & I & \\ & & \ddots & \ddots & \ddots \end{bmatrix} = \mathcal{U}\mathcal{L},$$

multiply to the right by \mathcal{L}^{-1} and get

$$\begin{bmatrix} B_0 & B_1 & \dots & & \\ B_{-1} & B_0 & B_1 & \ddots & \\ & B_{-1} & B_0 & B_1 & \ddots \end{bmatrix} \begin{bmatrix} I & & & & \\ G & I & & & \\ G^2 & G & I & & \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} U_0 & U_1 & U_2 & \dots & \\ U_0 & U_1 & U_2 & \ddots & \\ & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Reading this equation in the second block-row yields

$$\sum_{i=0}^{\infty} B_{i-1} G^i = 0, \quad U_k = \sum_{i=0}^{\infty} B_{i+k} G^i, \quad k = 0, 1, 2, \dots$$

That is, the matrix G solves the matrix equation $\sum_{i=0}^{\infty} B_{i-1} G^i = 0$.

The existence of the canonical factorization, implies the existence of the solution G of minimal spectral radius $\rho(G) < 1$ of the matrix equation $\sum_{i=0}^{\infty} B_{i-1} X^i = 0$. Conversely, if G solves the above equation where $\rho(G) < 1$, if $\det B(z)$ has exactly m roots of modulus less than 1 and no roots of modulus 1, then there exists a canonical factorization $B(z) = U(z)(I - z^{-1}G)$.

More generally, for a function $B(z) = \sum_{i=-1}^{\infty} z^i B_i \in \mathcal{W}_m$ we can prove that if there exists a weak canonical factorization $B(z) = U(z)(I - z^{-1}G)$ then G solves the above matrix equation, $\|G^k\|$ is uniformly bounded from above and is the solution with minimal spectral radius $\rho(G) \leq 1$.

Conversely, if

- $\sum_{i=0}^{\infty} (i+1)|B_i| < \infty$,
- there exists a solution G of the matrix equation such that $\rho(G) \leq 1$, $\|G^k\|$ is uniformly bounded from above for some norm $\|\cdot\|$,
- all the zeros of $\det B(z)$ of modulus less than 1 are eigenvalues of G ,

then there exists a (weak) canonical factorization $B(z) = U(z)(I - z^{-1}G)$.

In fact, the proof of the existence of the solution G is given in a constructive way by showing that the sequence generated by $G_{k+1} = G_k - \sum_{i=-1}^{\infty} B_i G_k^{i+1}$ with $G_0 = 0$ has non-decreasing non-negative entries which are bounded from above by a constant. Thus it has a limit. This sequence provides also a way for numerically computing G .

Similar arguments hold for G/M/1-type Markov chains. A special situation is given by QBD processes where the matrix H is block tridiagonal block Toeplitz.

For a QBD process where $B(z) = z^{-1}B_{-1} + B_0 + zB_1$ the following canonical factorizations hold

$$B(z) = (zI - R)U_0(I - z^{-1}G), \quad B(z^{-1}) = (zI - \widehat{R})\widehat{U}_0(I - z^{-1}\widehat{G}).$$

Moreover, the roots ξ_i , $i = 1, \dots, 2m$ of the polynomial $\det zB(z)$ are such that

$$|\xi_1| \leq \dots \leq |\xi_m| = \xi_m \leq 1 \leq \xi_{m+1} = |\xi_{m+1}| \leq \dots |\xi_{2m}|,$$

where (5) holds true.

The matrices $G, R, \widehat{G}, \widehat{R}$ solve the equations

$$\begin{aligned} B_{-1} + B_0G + B_1G^2 &= 0, & R^2B_{-1} + RB_0 + B_1 &= 0, \\ B_{-1}\widehat{G}^2 + B_0\widehat{G} + B_1 &= 0, & B_{-1} + \widehat{R}B_0 + \widehat{R}^2B_1 &= 0. \end{aligned} \tag{6}$$

For a general function $B(z) = z^{-1}B_{-1} + B_0 + zB_1$, not necessarily related to Markov chains, we can provide existence conditions for the solutions of the above equations which relate them to the Laurent series of $H(z) := B(z)^{-1} = \sum_{i=-\infty}^{+\infty} z^i H_i$ in its domain of analyticity.

Theorem 9 If $|\xi_m| < 1 < |\xi_{m+1}|$ and if there exists a solution G with $\rho(G) = |\xi_m|$ to the first equation in (6) then

1. the matrix $K = B_0 + B_1 G$ is invertible, there exists the solution $R = -B_1 K^{-1}$, and $B(z)$ has the canonical factorization

$$B(z) = (I - zR)K(I - z^{-1}G)$$

2. $B(z)$ is invertible in the annulus $\mathbb{A} = \{z \in \mathbb{C} : |\xi_m| < |z| < |\xi_{m+1}|\}$ and $H(z) = B(z)^{-1} = \sum_{i=-\infty}^{+\infty} z^i H_i$ is convergent for $z \in \mathbb{A}$, where

$$H_i = \begin{cases} G^{-i} H_0, & i < 0, \\ \sum_{j=0}^{+\infty} G^j K^{-1} R^j, & i = 0, \\ H_0 R^i, & i > 0. \end{cases}$$

3. If H_0 is nonsingular, then there exist the solutions $\widehat{G} = H_0 R H_0^{-1}$, $\widehat{R} = H_0^{-1} G H_0$ and $\widehat{B}(z) = B(z^{-1})$ has the canonical factorization

$$\widehat{B}(z) = (I - z\widehat{R})\widehat{K}(I - z^{-1}\widehat{G}), \quad \widehat{K} = B_0 + B_{-1}\widehat{G} = B_0 + \widehat{R}B_1$$

Theorem 10 Let $|\xi_n| \leq 1 \leq |\xi_{n+1}|$. Assume that there exist solutions G and \widehat{G} to the corresponding equations in (6). Then

1. $B(z)$ has the (weak) canonical factorization

$$B(z) = (I - zR)K(I - z^{-1}G),$$

2. $\widehat{B}(z) = B(z^{-1})$ has the (weak) canonical factorization

$$\widehat{B}(z) = (I - z\widehat{R})\widehat{K}(I - z^{-1}\widehat{G}),$$

3. if $|\xi_n| < |\xi_{n+1}|$, then the series

$$W = \sum_{i=0}^{\infty} G^i K^{-1} R^i, \quad (W = H_0)$$

is convergent, W is the unique solution of the equation

$$X - GXR = K^{-1},$$

W is nonsingular and $\widehat{G} = WRW^{-1}$, $\widehat{R} = W^{-1}GW$.

5.4.1 Solving Matrix Equations

Here we examine some algorithms for solving matrix equations of the kind $X = A_{-1} + A_0X + A_1X^2$, equivalently $B_{-1} + B_0X + B_1X^2 = 0$, or, more generally, of the kind $X = \sum_{i=-1}^{\infty} A_iX^{i+1}$, equivalently $\sum_{i=-1}^{\infty} B_iX^{i+1} = 0$.

The most natural and immediate approach relies on fixed point iterations which generate sequences X_k by means of $X_{k+1} = F(X_k)$ where $F(X)$ is a suitable function. For instance,

$$\begin{aligned} F(X) &= \sum_{i=-1}^{\infty} A_iX^{i+1}, \\ F(X) &= (I - A_0)^{-1}(A_{-1} + \sum_{i=1}^{\infty} A_iX^{i+1}), \\ F(X) &= (I - \sum_{i=0}^{\infty} A_iX^i)^{-1}A_{-1}. \end{aligned}$$

In the framework of Markov chains, these iterations provide

- local linear convergence for $X_0 = 0$ or $X_0 = I$;
- global monotonic convergence for $X_0 = 0$, and in the case of positive recurrence, global convergence for $X_0 = I$;
- generally very slow convergence which turns to sublinear in the null recurrent case.

On the other hand, these algorithms are easy to implement: in the QBD case only few matrix multiplications per step are needed. One can define Newton's iteration which provides quadratic convergence in the non null-recurrent case, but at each step of this iteration a linear matrix equation must be solved. This requires a higher cost.

A more efficient technique relies on cyclic reduction; it combines a low computational cost per step and quadratic convergence in the non null-recurrent case which turns to linear with factor $1/2$ for null-recurrent processes.

5.4.2 Solving Matrix Equations by Means of CR: The QBD Case

Consider for simplicity a quadratic matrix equation $B_{-1} + B_0X + B_1X^2 = 0$, and rewrite it in matrix form as

$$\begin{bmatrix} B_0 & B_1 & & \\ B_{-1} & B_0 & B_1 & \\ & B_{-1} & B_0 & B_1 \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} X \\ X^2 \\ X^3 \\ \vdots \end{bmatrix} = \begin{bmatrix} -B_{-1} \\ 0 \\ 0 \\ \vdots \end{bmatrix}.$$

Apply one step of cyclic reduction and get

$$\begin{bmatrix} \widehat{B}_0^{(1)} & B_1^{(1)} & & & \\ B_{-1}^{(1)} & B_0^{(1)} & B_1^{(1)} & & \\ & B_{-1}^{(1)} & B_0^{(1)} & B_1^{(1)} & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} X \\ X^3 \\ X^5 \\ \vdots \end{bmatrix} = \begin{bmatrix} -B_{-1} \\ 0 \\ 0 \\ \vdots \end{bmatrix}.$$

Cyclically repeating the same reduction yields

$$\begin{bmatrix} \widehat{B}_0^{(k)} & B_1^{(k)} & & & \\ B_{-1}^{(k)} & B_0^{(k)} & B_1^{(k)} & & \\ & B_{-1}^{(k)} & B_0^{(k)} & B_1^{(k)} & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} X \\ X^{2^k-1} \\ X^{2 \cdot 2^k-1} \\ \vdots \end{bmatrix} = \begin{bmatrix} -B_{-1} \\ 0 \\ 0 \\ \vdots \end{bmatrix},$$

where

$$B_{-1}^{(k+1)} = -B_{-1}^{(k)} C^{(k)} B_{-1}^{(k)}, \quad C^{(k)} = (B_0^{(k)})^{-1},$$

$$B_1^{(k+1)} = -B_1^{(k)} C^{(k)} B_1^{(k)},$$

$$B_0^{(k+1)} = B_0^{(k)} - B_1^{(k)} C^{(k)} B_{-1}^{(k)} - B_{-1}^{(k)} C^{(k)} B_1^{(k)},$$

$$\widehat{B}_0^{(k+1)} = \widehat{B}_0^{(k)} - B_1^{(k)} C^{(k)} B_{-1}^{(k)}.$$

Observe that $X = (\widehat{B}_0^{(k)})^{-1}(-B_{-1} - B_1^{(k)}X^{2^k-1})$, moreover, in view of the results of Sect. 4.4.1, one can prove that $\|B_1^{(k)}X^{2^k-1}\| = O(d^{2^k})$, for any $\frac{\xi_m}{\xi_{m+1}} < d < 1$, and that $\widehat{B}_0^{(k)}$ is nonsingular with bounded inverse. Therefore X can be approximated by $-(\widehat{B}_0^{(k)})^{-1}B_{-1}$ with an $O(d^{2^k})$ error.

5.4.3 Solving Matrix Equations by Means of CR: The M/G/1 and G/M/1 Cases

The same procedure can be successfully applied to equations of the kind $\sum_{i=-1}^{\infty} B_i X^i = 0$. In order to see this, we have first to describe and analyze the cyclic reduction algorithm applied to the infinite almost block Toeplitz block Hessenberg matrix

$$\mathcal{B} = \begin{bmatrix} \widehat{B}_0 & \widehat{B}_1 & \widehat{B}_2 & \widehat{B}_3 & \dots & \dots & \dots \\ B_{-1} & B_0 & B_1 & B_2 & B_3 & \ddots & \ddots \\ B_{-1} & B_0 & B_1 & B_2 & B_3 & \ddots & \ddots \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Even-odd permutation applied to block rows and block columns of \mathcal{B} leads to the matrix

$$\left[\begin{array}{ccc|ccccc} B_0 & B_2 & B_4 & \dots & B_{-1} & B_1 & B_3 & \dots \\ B_0 & B_2 & \ddots & & B_{-1} & B_3 & \ddots & \\ \ddots & \ddots & & & \ddots & \ddots & \ddots & \\ \hline \widehat{B}_1 & \widehat{B}_3 & \widehat{B}_5 & \dots & \widehat{B}_0 & \widehat{B}_2 & \widehat{B}_4 & \dots \\ B_{-1} & B_1 & B_3 & \dots & B_0 & B_2 & B_4 & \dots \\ B_{-1} & B_1 & \ddots & & B_0 & \ddots & & \\ \ddots & \ddots & & & \ddots & & & \end{array} \right] =: \left[\begin{array}{c|c} D_1 & V \\ \hline W & D_2 \end{array} \right].$$

The Schur complement $\mathcal{S} = D_2 - WD_1^{-1}V$ is given by the following matrix

$$\left[\begin{array}{ccc|ccccc} \widehat{B}_0 & \widehat{B}_2 & \widehat{B}_4 & \dots \\ B_0 & B_2 & \ddots \\ B_0 & \ddots \\ \ddots & & & \end{array} \right] - \left[\begin{array}{ccc|ccccc} \widehat{B}_1 & \widehat{B}_3 & \widehat{B}_5 & \dots \\ B_{-1} & B_1 & B_3 & \ddots \\ B_{-1} & B_1 & \ddots \\ \ddots & \ddots & & \end{array} \right] \left[\begin{array}{ccc|ccccc} B_0 & B_2 & B_4 & \dots \\ B_0 & B_2 & \ddots \\ B_0 & \ddots \\ \ddots & & & \end{array} \right]^{-1} \left[\begin{array}{ccc|ccccc} B_{-1} & B_1 & B_3 & \dots \\ B_{-1} & B_1 & \ddots \\ B_{-1} & \ddots \\ \ddots & & & \end{array} \right] \quad (7)$$

From the structure properties of block Toeplitz matrices it follows that \mathcal{S} still has the structure of \mathcal{B} , being block Toeplitz except for the first block row, and in block Hessenberg form, more specifically

$$\mathcal{S} = \left[\begin{array}{cccc|cccc} \widehat{B}'_0 & \widehat{B}'_1 & \widehat{B}'_2 & \widehat{B}'_3 & \dots & \dots & \dots & \dots \\ B'_{-1} & B'_0 & B'_1 & B'_2 & B'_3 & \ddots & \ddots & \ddots \\ B'_{-1} & B'_0 & B'_1 & B'_2 & B'_3 & \ddots & \ddots & \ddots \\ \ddots & \ddots \end{array} \right]$$

for suitable blocks \widehat{B}'_i and B'_i . Reading the expression of \mathcal{S} along the first and the second block rows yields the equations for the \widehat{B}'_i and B_i :

$$\begin{aligned} [\widehat{B}'_0 \ \widehat{B}'_1 \ \widehat{B}'_2 \ \dots] &= [\widehat{B}_0 \ \widehat{B}_2 \ \widehat{B}_4 \ \dots] - [\widehat{B}_1 \ \widehat{B}_3 \ \widehat{B}_5 \ \dots] N, \\ [B'_{-1} \ B'_0 \ B'_1 \ \widehat{B}'_2 \ \dots] &= [0 \ B_0 \ B_2 \ B_4 \ \dots] - [B_{-1} \ B_1 \ B_3 \ B_5 \ \dots] N. \end{aligned} \quad (8)$$

for $N = D_1^{-1}V$ being a block upper triangular block Toeplitz matrix since product of two block upper triangular block Toeplitz matrices. Since the structure of \mathcal{B} is preserved, we may cyclically apply the same procedure of odd-even permutation and Schur complementation.

From the computational point of view the following operations must be performed in order to execute one step of CR:

- inverting the infinite block-triangular block-Toeplitz matrix D_1 : this can be performed by means of the doubling algorithm until the (far enough) off-diagonal entries are sufficiently small; due to the exponential decay, the stop condition is verified in few iterations;
- multiplication $N = D_1^{-1}W$ of the block triangular block Toeplitz matrices D_1^{-1} and W : this can be performed by means of FFT.

The overall computational cost is $O(Nm^2 \log m + Nm^3)$ where N is the maximum number of the non-negligible blocks.

It is not difficult to give a functional interpretation to CR applied to infinite almost block Toeplitz matrices in block Hessenberg form. It is sufficient to associate with the block Hessenberg almost block Toeplitz matrix \mathcal{B} the matrix functions $\varphi(z) = \sum_{i=-1}^{\infty} z^i B_i$, and $\widehat{\varphi}(z) = \sum_{i=0}^{\infty} z^i \widehat{B}_i$. Similarly, associate with the structured matrix \mathcal{B}_k obtained at the step k of CR the functions $\varphi(z)^{(k)} = \sum_{i=-1}^{\infty} z^i B_i^{(k)}$, and $\widehat{\varphi}^{(k)}(z) = \sum_{i=0}^{\infty} z^i \widehat{B}_i^{(k)}$. Thus, using the interplay between infinite Toeplitz matrices and power series, in view of (8), the Schur complement at step $k+1$ can be written as

$$\begin{aligned}\varphi^{(k+1)}(z) &= \varphi_{\text{even}}^{(k)}(z) - z\varphi_{\text{odd}}^{(k)}(z) (\varphi_{\text{even}}^{(k)}(z))^{-1} \varphi_{\text{odd}}^{(k)}(z) \\ \widehat{\varphi}^{(k+1)}(z) &= \widehat{\varphi}_{\text{even}}^{(k)}(z) - \widehat{\varphi}_{\text{odd}}^{(k)}(z) (\varphi_{\text{even}}^{(k)}(z))^{-1} \varphi_{\text{odd}}^{(k)}(z)\end{aligned}\quad (9)$$

where for a matrix function $F(z)$ we denote

$$F_{\text{even}}(z^2) = \frac{1}{2}(F(z) + F(-z)), \quad F_{\text{odd}}(z^2) = \frac{1}{2z}(F(z) - F(-z)).$$

By means of formal manipulation, relying on the identity $a - z^2ba^{-1}b = (a + zb)a^{-1}(a - zb)$ we find that

$$\begin{aligned}\varphi^{(k+1)}(z^2) &= \varphi_{\text{even}}^{(k)}(z^2) - z^2\varphi_{\text{odd}}^{(k)}(z^2)\varphi_{\text{even}}^{(k)}(z^2)^{-1}\varphi_{\text{odd}}^{(k)}(z^2) \\ &= (\varphi_{\text{even}}^{(k)}(z^2) + z\varphi_{\text{odd}}^{(k)}(z^2))\varphi_{\text{even}}^{(k)}(z^2)^{-1}(\varphi_{\text{even}}^{(k)}(z^2) - z\varphi_{\text{odd}}^{(k)}(z^2)).\end{aligned}$$

On the other hand, for a function $\varphi(z)$ one has

$$\varphi_{\text{even}}(z^2) + z\varphi_{\text{odd}}(z^2) = \varphi(z), \quad \varphi_{\text{even}}(z^2) - z\varphi_{\text{odd}}(z^2) = \varphi(-z),$$

so that

$$\varphi^{(k+1)}(z^2) = \varphi^{(k)}(z)\varphi_{\text{even}}^{(k)}(z)^{-1}\varphi^{(k)}(-z), \quad (10)$$

which extends the Graeffe iteration to matrix power series. Thus, the functional iteration for $\varphi^{(k)}(z)$ can be rewritten in simpler form as

$$\varphi^{(k+1)}(z^2) = \varphi^{(k)}(z) \left(\frac{1}{2}(\varphi^{(k)}(z) + \varphi^{(k)}(-z)) \right)^{-1} \varphi^{(k)}(-z).$$

Define $\psi^{(k)}(z) = \varphi^{(k)}(z)^{-1}$ and find that $\psi^{(k+1)}(z^2) = \frac{1}{2}(\psi^{(k)}(z) + \psi^{(k)}(-z))$. This property enables us to prove the following convergence result.

Theorem 11 Assume we are given a function $\varphi(z) = \sum_{i=-1}^{+\infty} z^i B_i$ and positive numbers $r_1 < 1 < r_2$ such that

1. for any $z \in \mathbb{A}(r_1, r_2)$ the matrix $\varphi(z)$ is analytic and nonsingular
2. the function $\psi(z) = \varphi(z)^{-1}$, analytic in $\mathbb{A}(r_1, r_2)$, is such that $\det H_0 \neq 0$ where $\psi(z) = \sum_{i=-\infty}^{+\infty} z^i H_i$

Then

1. the sequence $\varphi^{(k)}(z)$ converges uniformly to H_0^{-1} over any compact set in $\mathbb{A}(r_1, r_2)$
2. for any ϵ and for any norm there exist constants $c_i > 0$ such that

$$\begin{aligned} \|B_{-1}^{(k)}\| &\leq c_{-1}(r_1 + \epsilon)^{2^k}, \\ \|B_i^{(k)}\| &\leq c_i(r_2 - \epsilon)^{-i2^k}, \quad \text{for } i \geq 1, \\ \|B_0^{(k)} - H_0^{-1}\| &\leq c_0 \left(\frac{r_1 + \epsilon}{r_2 - \epsilon} \right)^{2^k}. \end{aligned}$$

In the case where $\xi_m = 1 < \xi_{m+1}$ (positive recurrent case) we have the following

Theorem 12 Assume we are given a function $\varphi(z) = \sum_{i=-1}^{+\infty} z^i B_i$ and positive numbers $r_1 = 1 < r_2$ such that

1. for any $z \in \mathbb{A}(r_1, r_2)$ the matrix $\varphi(z)$ is analytic and nonsingular
2. the function $\psi(z) = \varphi(z)^{-1}$, analytic in $\mathbb{A}(r_1, r_2)$, is such that $\det H_0 \neq 0$ where $\psi(z) = \sum_{i=-\infty}^{+\infty} z^i H_i$

Then

1. the sequence $\varphi^{(k)}(z)$ converges uniformly to H_0^{-1} over any compact set in $\mathbb{A}(r_1, r_2)$

2. for any ϵ and for any norm there exist constants $c_i > 0$ such that

$$\begin{aligned} \lim_k B_{-1}^{(k)} &= B_{-1}^{(\infty)}, \\ \|B_i^{(k)}\| &\leq c_i(r_2 - \epsilon)^{-i2^k}, \quad \text{for } i \geq 1, \\ \|B_0^{(k)} - H_0^{-1}\| &\leq c_0 \left(\frac{r + \epsilon}{r_2 - \epsilon} \right)^{2^k}. \end{aligned}$$

Observe that, in principle, cyclic reduction in functional form can be applied to any function having a Laurent series of the kind $\varphi(z) = \sum_{i=-\infty}^{\infty} z^i B_i$, provided it is analytic over an annulus including the unit circle.

In the matrix framework, CR can be applied to the associated block Toeplitz matrix, no matter if it is not Hessenberg. The computational difficulty for a non-Hessenberg matrix is that the block corresponding to $\varphi_{\text{even}}(z)$ is not triangular. Therefore its inversion, required by CR is not cheap.

Now we are ready to provide a fast algorithm for the solution of the matrix equation $\sum_{i=-1}^{\infty} B_i X^{i+1} = 0$. Rewrite the above equation in matrix form as

$$\begin{bmatrix} B_0 & B_1 & B_2 & B_3 & B_4 & \dots \\ B_{-1} & B_0 & B_1 & B_2 & B_3 & \ddots & \ddots \\ & B_{-1} & B_0 & B_1 & B_2 & \ddots & \ddots \\ & & B_{-1} & B_0 & B_1 & B_2 & \ddots & \ddots \\ & & & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} X \\ X^2 \\ X^3 \\ \vdots \end{bmatrix} = \begin{bmatrix} -B_{-1} \\ 0 \\ 0 \\ \vdots \end{bmatrix}.$$

Apply cyclic reduction and get

$$\begin{bmatrix} \widehat{B}_0^{(k)} & \widehat{B}_1^{(k)} & \widehat{B}_2^{(k)} & \widehat{B}_3^{(k)} & \widehat{B}_4^{(k)} & \dots \\ B_{-1}^{(k)} & B_0^{(k)} & B_1^{(k)} & B_2^{(k)} & B_3^{(k)} & \ddots & \ddots \\ & B_{-1}^{(k)} & B_0^{(k)} & B_1^{(k)} & B_2^{(k)} & \ddots & \ddots \\ & & B_{-1}^{(k)} & B_0^{(k)} & B_1^{(k)} & B_2^{(k)} & \ddots & \ddots \\ & & & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} X \\ X^{2^k-1} \\ X^{2 \cdot 2^k-1} \\ X^{3 \cdot 2^k-1} \\ \vdots \end{bmatrix} = \begin{bmatrix} -B_{-1} \\ 0 \\ 0 \\ \vdots \end{bmatrix}, \quad (11)$$

where $B_i^{(k)}$ and $\widehat{B}_i^{(k)}$ are defined in functional form by (9) with $\varphi^{(k)}(z) = \sum_{i=-1}^{\infty} B_i^{(k)} z^{i+1}$, $\widehat{\varphi}^{(k)}(z) = \sum_{i=0}^{\infty} \widehat{B}_i^{(k)} z^i$, where $B_i^{(0)} = B_i$, $\widehat{B}_i^{(0)} = B_i$.

From the first equation of (11) we find that

$$\begin{aligned} X &= (\widehat{B}_0^{(k)})^{-1} (-B_{-1} - \sum_{i=1}^{\infty} \widehat{B}_i^{(k)} X^{i \cdot 2^k - 1}) \\ &= -(\widehat{B}_0^{(k)})^{-1} B_{-1} - (\widehat{B}_0^{(k)})^{-1} \sum_{i=1}^{\infty} \widehat{B}_i^{(k)} X^{i \cdot 2^k - 1}. \end{aligned}$$

We have the following properties:

- $0 \leq (\widehat{B}_0^{(k)})^{-1}$ is bounded from above;
- $B_i^{(k)}$ converges to zero as $k \rightarrow \infty$;
- convergence is double exponential for positive recurrent or transient Markov chains;
- convergence is linear for null recurrent Markov chains.

A natural question is how to implement cyclic reduction for infinite M/G/1-type or G/M/1-type Markov chains. One has to compute the coefficients of the matrix Laurent series $\varphi^{(k)}(z) = \sum_{i=-1}^{\infty} z^{i+1} B_i^{(k)}$ according to (10). A first approach relies on the following steps:

1. Interpret the CR step as the Schur complementation (7) in terms of infinite block Toeplitz matrices;
2. truncate matrices to finite size N for a sufficiently large N ;
3. apply the Toeplitz matrix technology to perform each single operation in the CR step for computing the Schur complement.

A second approach relies on the technique of *evaluation interpolation* applied to (9). That is, remain in the framework of analytic functions and implement the iteration point-wise as follows:

1. Choose the N th roots of the unity, for $N = 2^p$, as knots;
2. apply CR point-wise at the current knots, more specifically, concerning $\varphi^{(k+1)}(z)$, evaluate this function at the knots by using (10) and the values of $\varphi^{(k)}(z)$ at the knots; similarly do for $\widehat{\varphi}^{(k+1)}(z)$ relying on (9);
3. compute the block coefficients of the matrix polynomial which interpolates these values and approximates $\varphi^{(k+1)}(z)$, check if the error in the approximation is small enough;
4. if not, set $p = 2p$ and return to step 2 (using the already computed quantities);
5. if the error is small enough then exit the cycle.

This kind of implementation, which relies on FFT and on a suitable test for estimating the error, is more efficient in practice and allows an easier control on the number of interpolation points. Moreover, in the process of convergence, the number of interpolation points gets smaller.

A similar approach can be applied for the G/M/1 case.

5.5 Shifting Techniques

We have seen that in the solution of QBD Markov chains one needs to compute the minimal nonnegative solution G of the $m \times m$ matrix equation

$$B_{-1} + B_0 X + B_1 X^2 = 0.$$

Moreover the roots ξ_i of the polynomial $\det(B_{-1} + B_0z + B_1z^2)$ are such that

$$|\xi_1| \leq \dots \leq \xi_m \leq 1 \leq \xi_{m+1} \leq \dots \leq |\xi_{2m}|.$$

In the null recurrent case where $\xi_m = \xi_{m+1} = 1$ the convergence of algorithms for computing G deteriorates and the problem of computing G becomes ill-conditioned.

Here we provide a tool for getting rid of these drawbacks. The idea is an elaboration of a result introduced by Brauer [23] in 1952 and extended to matrix polynomials by He et al. [45] in 2001. It relies on transforming the polynomial $B(z) = B_{-1} + zB_0 + z^2B_1$ into a new one $\tilde{B}(z) = \tilde{B}_{-1} + z\tilde{B}_0 + z^2\tilde{B}_1$ in such a way that $\tilde{b}(z) = \det \tilde{B}(z)$ has the same roots of $b(z) = \det B(z)$ except for $\xi_m = 1$ which is shifted to 0, and $\xi_{m+1} = 1$ which is shifted to infinity.

This way, the roots of $\tilde{b}(z)$ are

$$0, \xi_1, \dots, \xi_{m-1}, \xi_{m+2}, \dots, \xi_{2m}, \infty.$$

Now, we give an elementary description of the Brauer idea. Let A be an $n \times n$ matrix, let u be an eigenvector corresponding to the eigenvalue λ , that is $Au = \lambda u$, let v be any vector such that $v^T u = 1$. Then $B := A - \lambda uv^T$ has the same eigenvalues of A except for λ which is replaced by 0. In fact, $Bu = Au - \lambda uv^T u = \lambda u - \lambda u = 0$. Moreover, if $w^T A = \mu w^T$ for $\mu \neq \lambda$, then $w^T u = 0$ so that $w^T B = w^T A = \mu w^T$.

Can we extend the same argument to matrix polynomials and to the polynomial eigenvalue problem?

Recall our assumptions: $B(z) = B_{-1} + zB_0 + z^2B_1$, where $B_{-1} = -A_{-1}$, $B_0 = I - A_0$, $B_1 = -A_1$, A_{-1}, A_0, A_1 are $m \times m$ matrices, such that $A_{-1}, A_0, A_1 \geq 0$, $(A_{-1} + A_0 + A_1)\mathbf{e} = \mathbf{e}$, $\mathbf{e} = (1, \dots, 1)^T$, $A_{-1} + A_0 + A_1$ is irreducible. For the zeros ξ_i of $b(z) = \det B(z)$ we have $\xi_m = 1 = \xi_{m+1}$, so that $z = 1$ is zero of multiplicity 2.

The following equations have solution

$$\begin{aligned} B_{-1} + B_0G + B_1G^2 &= 0, \quad \rho(G) = \xi_m, \\ R^2B_{-1} + RB_0 + B_1 &= 0, \quad \rho(R) = \xi_{m+1}^{-1}, \\ B_{-1}\widehat{G}^2 + B_0\widehat{G} + B_1 &= 0, \quad \rho(\widehat{G}) = \xi_{m+1}^{-1}, \\ B_{-1} + \widehat{R}B_0 + \widehat{R}^2B_1 &= 0, \quad \rho(\widehat{R}) = \xi_m. \end{aligned}$$

Recall that the existence of these four solutions is equivalent to the existence of the canonical factorizations of $\varphi(z) = z^{-1}B(z)$ and of $\varphi(z^{-1})$ where $B(z) = B_{-1} + zB_0 + z^2B_1$, that is,

$$\varphi(z) = (I - zR)K(I - z^{-1}G), \quad R, K, G \in \mathbb{R}^{m \times m}, \quad \det K \neq 0,$$

$$\varphi(z^{-1}) = (I - z\widehat{R})\widehat{K}(I - z^{-1}\widehat{G}), \quad \widehat{R}, \widehat{K}, \widehat{G} \in \mathbb{R}^{m \times m}, \quad \det \widehat{K} \neq 0.$$

5.5.1 Shift to the Right

Here, we construct a new matrix polynomial $\tilde{B}(z)$ having the same roots as $B(z)$ except for the root ξ_m which is shifted to zero.

Recall that G has eigenvalues ξ_1, \dots, ξ_m , denote u_G an eigenvector of G such that $Gu_G = \xi_m u_G$, denote v any vector such that $v^T u_G = 1$ and define

$$\tilde{B}(z) = B(z) \left(I + \frac{\xi_m}{z - \xi_m} Q \right), \quad Q = u_G v^T.$$

Theorem 13 *The function $\tilde{B}(z)$ coincides with the quadratic matrix polynomial $\tilde{B}(z) = \tilde{B}_{-1} + z\tilde{B}_0 + z^2\tilde{B}_1$ with matrix coefficients*

$$\tilde{B}_{-1} = B_{-1}(I - Q), \quad \tilde{B}_0 = B_0 + \xi_m B_1 Q, \quad \tilde{B}_1 = B_1.$$

Moreover, the roots of $\tilde{B}(z)$ are $0, \xi_1, \dots, \xi_{m-1}, \xi_{m+1}, \dots, \xi_{2m}$.

We give an outline of the proof. Since $B(\xi_m)u_G = 0$, and $Q = u_G v^T$, then $B(\xi_m)Q = 0$ so that $B_{-1}Q = -\xi_m B_0 Q - \xi_m^2 B_1 Q$, and we have

$$\begin{aligned} B(z)Q &= -\xi_m B_0 Q - \xi_m^2 B_1 Q + B_0 Q z + B_1 Q z^2 \\ &= (z^2 - \xi_m^2) B_1 Q + (z - \xi_m) B_0 Q. \end{aligned}$$

This way $\frac{\xi_m}{z - \xi_m} B(z)Q = \xi_m(z + \xi_m)B_1 Q + \xi_m B_0 Q$, therefore

$$\tilde{B}(z) = B(z) + \frac{\xi_m}{z - \xi_m} B(z)Q = \tilde{B}_{-1} + \tilde{B}_0 z + \tilde{B}_1 z^2$$

so that

$$\tilde{B}_{-1} = B_{-1}(I - Q), \quad \tilde{B}_0 = B_0 + \xi_m B_1 Q, \quad \tilde{B}_1 = B_1.$$

Since $\det(I + \frac{\xi_m}{z - \xi_m} Q) = \frac{z}{z - \xi_m}$ then from the definition of $\tilde{B}(z)$ we have $\det \tilde{B}(z) = \frac{z}{z - \xi_m} \det B(z)$. This means that the roots of the polynomial $\det \tilde{B}(z)$ coincide with the roots of $\det B(z)$ except the root equal to ξ_m which is replaced with 0.

5.6 Shift to the Left

Here, we construct a new matrix polynomial $\tilde{B}(z)$ having the same roots as $B(z)$ except for the root ξ_m which is shifted to infinity.

Recall that R has eigenvalues $\xi_{m+1}^{-1}, \dots, \xi_{2m}^{-1}$, denote v_R a left eigenvector of R such that $v_R^T R = \xi_{m+1}^{-1} v_R^T$, denote w any vector such that $w^T v_R = 1$ and define

$$\widetilde{B}(z) = \left(I - \frac{z}{z - \xi_{m+1}} S \right) B(z), \quad S = w v_R^T.$$

Theorem 14 *The function $\widetilde{B}(z)$ coincides with the quadratic matrix polynomial $\widetilde{B}(z) = \widetilde{B}_{-1} + z\widetilde{B}_0 + z^2\widetilde{B}_1$ with matrix coefficients*

$$\widetilde{B}_{-1} = B_{-1}, \quad \widetilde{B}_0 = B_0 + \xi_{m+1}^{-1} S B_{-1}, \quad \widetilde{B}_1 = (I - S) B_1.$$

Moreover, the roots of $\widetilde{B}(z)$ are $\xi_1, \dots, \xi_m, \xi_{m+2}, \dots, \xi_{2m}, \infty$.

5.7 Double Shift

The right and left shifts can be combined together yielding a new quadratic matrix polynomial $\widetilde{B}(z)$ with the same roots of $B(z)$, except for ξ_m and ξ_{m+1} , which are shifted to 0 and to infinity, respectively.

Define the matrix function

$$\widetilde{B}(z) = \left(I - \frac{z}{z - \xi_{m+1}} S \right) B(z) \left(I + \frac{\xi_m}{z - \xi_m} Q \right),$$

and find that $\widetilde{B}(z) = \widetilde{B}_{-1} + z\widetilde{B}_0 + z^2\widetilde{B}_1$, with matrix coefficients

$$\begin{aligned} \widetilde{B}_{-1} &= B_{-1}(I - Q), \\ \widetilde{B}_0 &= B_0 + \xi_m B_1 Q + \xi_{m+1}^{-1} S B_{-1} - \xi_{m+1}^{-1} S B_{-1} Q \\ &= B_0 + \xi_m B_1 Q + \xi_{m+1}^{-1} S B_{-1} - \xi_m S B_1 Q \\ \widetilde{B}_1 &= (I - S) B_1. \end{aligned}$$

The matrix polynomial $\widetilde{B}(z)$ has roots $0, \xi_1, \dots, \xi_{m-1}, \xi_{m+2}, \dots, \xi_{2m}, \infty$. In particular, $\widetilde{B}(z)$ is nonsingular on the unit circle and on the annulus $|\xi_{m-1}| < |z| < |\xi_{m+2}|$.

5.8 Shifts and Canonical Factorizations

Here, we provide an answer to the following question. Under which conditions both the functions $\widetilde{\varphi}(z)$ and $\widetilde{\varphi}(z^{-1})$ obtained after applying the shift have a (weak)

canonical factorization? In different words: Under which conditions there exist the four minimal solutions to the equations obtained after applying the shift where the matrices B_i are replaced by \tilde{B}_i , $i = -1, 0, 1$?

These matrix solutions will be denoted by $\tilde{G}, \tilde{R}, \tilde{\tilde{G}}, \tilde{\tilde{R}}$. They are the analogous of the solutions $G, R, \tilde{G}, \tilde{R}$ to the original equations. We examine the case of the shift to the right. The shift to the left can be treated similarly. We will examine separately the case of the double shift.

Independently of the recurrent or transient case, the canonical factorization of $\tilde{\varphi}(z)$ always exists. We have the following theorem concerning $\tilde{B}(z) = B(z)(I + \frac{\xi_n}{z - \xi_n}Q)$, $Q = u_G v^T$.

Theorem 15 *The function $\tilde{\varphi}(z) = z^{-1}\tilde{B}(z)$, has the following factorization*

$$\tilde{\varphi}(z) = (I - zR)K(I - z^{-1}\tilde{G}), \quad \tilde{G} = G - \xi_n Q.$$

This factorization is canonical in the positive recurrent case, and weak canonical otherwise. Moreover, the eigenvalues of \tilde{G} are those of G , except for the eigenvalue ξ_n which is replaced by zero. Finally, $X = \tilde{G}$ and $Y = R$ are the solutions with minimal spectral radius of the equations

$$\tilde{B}_{-1} + \tilde{B}_0 X + \tilde{B}_1 X^2 = 0, \quad Y^2 \tilde{B}_{-1} + Y \tilde{B}_0 + \tilde{B}_1 = 0.$$

Now we examine the case of $\tilde{\varphi}(z^{-1})$. In the positive recurrent case, the matrix polynomial $\tilde{B}(z)$ is nonsingular on the unit circle, so that the function $\tilde{\varphi}(z^{-1})$ has a canonical factorization

Theorem 16 (Positive Recurrent) *If $\xi_m = 1 < \xi_{m+1}$ then the Laurent matrix polynomial $\tilde{\varphi}(z^{-1}) = z\tilde{B}(z^{-1})$, has the canonical factorization*

$$\tilde{\varphi}(z^{-1}) = (I - z\tilde{R})(\tilde{U} - I)(I - z^{-1}\tilde{\tilde{G}})$$

with $\tilde{\tilde{R}} = \tilde{W}^{-1}\tilde{G}\tilde{W}$, $\tilde{G} = G - Q$, $\tilde{\tilde{G}} = \tilde{W}R\tilde{W}^{-1}$, $\tilde{U} = \tilde{B}_0 + \tilde{B}_{-1}\tilde{\tilde{G}} = \tilde{B}_0 + \tilde{\tilde{R}}\tilde{B}_1$, where $\tilde{W} = W - QWR$ and $W = \sum_{i=0}^{\infty} G^i K^{-1} R^i$. Moreover, $X = \tilde{\tilde{G}}$ and $Y = \tilde{R}$ are the solutions with minimal spectral radius of the matrix equations

$$\tilde{B}_{-1}X^2 + \tilde{B}_0X + \tilde{B}_1 = 0, \quad \tilde{B}_{-1} + X\tilde{B}_0 + X^2\tilde{B}_1 = 0.$$

Now we examine the double shift in the null recurrent case. Consider the matrix polynomial obtained with the double shift

$$\tilde{B}(z) = (I - \frac{z}{z - \xi_{m+1}}S)B(z)(I + \frac{\xi_m}{z - \xi_m}Q), \quad Q = u_G v^T, \quad S = wv_R^T$$

Theorem 17 *The function $\tilde{\varphi}(z) = z^{-1}\tilde{B}(z)$ has the canonical factorization*

$$\tilde{\varphi}(z) = (I - z\tilde{R})K(I - z^{-1}\tilde{G}), \quad \tilde{R} = R - \xi_{m+1}S, \quad \tilde{G} = G - \xi_mQ$$

The matrices \tilde{G} and \tilde{R} are the solutions with minimal spectral radius of the equations $\tilde{B}_{-1} + \tilde{B}_0X + \tilde{B}_1X^2 = 0$ and $X^2\tilde{B}_{-1} + X\tilde{B}_0 + \tilde{B}_1 = 0$, respectively.

Theorem 18 *Let $Q = u_G v_G^T$ and $S = u_{\hat{R}} v_{\hat{R}}^T$, with $u_G^T v_{\hat{G}} = 1$ and $v_{\hat{R}}^T u_{\hat{R}} = 1$. Then*

$$\tilde{\varphi}(z^{-1}) = (I - z^{-1}\tilde{R})\tilde{K}(I - z\tilde{G}),$$

where

$$\begin{aligned} \tilde{R} &= \hat{R} - \gamma u_{\hat{R}} v_{\hat{G}}^T \hat{K}^{-1}, & \tilde{G} &= \hat{G} - \gamma \hat{K}^{-1} u_{\hat{R}} v_{\hat{G}}^T, \\ \gamma &= 1/(v_{\hat{G}}^T \hat{K}^{-1} u_{\hat{R}}), \\ \hat{K} &= A_{-1}\hat{G} + A_0 - I, & \tilde{K} &= \tilde{A}_{-1}\tilde{G} + \tilde{A}_0 - I. \end{aligned}$$

An immediate application of the above results concerns the Poisson problem for a QBD. It consists in solving the equation

$$(I - P)x = q + ze, \quad e = (1, 1 \dots)^T$$

where q is a given infinite vector, $x \in \mathbb{R}^N$ and $z \in \mathbb{R}$ are the unknowns and

$$P = \begin{bmatrix} A_0 + A_1 & A_1 & & & \\ A_{-1} & A_0 & A_1 & & \\ & & A_{-1} & A_0 & A_1 \\ & & & \ddots & \ddots & \ddots \end{bmatrix},$$

where A_{-1}, A_0, A_1 are nonnegative and $A_{-1} + A_0 + A_1$ is stochastic.

If $\xi_m < 1 < \xi_{m+1}$ then the general solution of this equation can be explicitly expressed in terms of the solutions of a suitable matrix difference equation. The shift technique provides a way to represent the solution by means of the solution of a suitable matrix difference equation even in the case of null recurrent models.

Some generalizations are possible. The shift technique can be generalized in order to shift to zero or to infinity a set of selected eigenvalues, leaving unchanged the remaining eigenvalues. This generalization is particularly useful when one has to move a pair of conjugate complex eigenvalues to zero or to infinity still maintaining real arithmetic.

There are potential applications which have to be investigated. For instance, the shift technique could be used to deflate already approximated roots within a polynomial root-finder which treats polynomial roots in terms of eigenvalues of

some companion matrix. A potential use is in MPSolve <http://numpi.dm.unipi.it/mpsolve> a package for high precision computation of roots of polynomials of large degree.

5.9 Available Software

The package SMCSolver provides a tool for solving M/G/1, G/M/1, QBD, and Non-Skip-Free type Markov chains. There is a Matlab Toolbox at <http://win.ua.ac.be/~vanhoudt/tools/> and a Fortran 95 version with a GUI interface at <http://bezout.dm.unipi.it/SMCSolver>.

6 Other Problems

Here we examine some other structures encountered in stochastic processes. Namely, a recursive matrix structure modeling tree-like processes and a sort of vector equation stemming from Markovian binary trees.

6.1 Tree-Like Processes

An interesting matrix structure encountered in Markov chains concerns Tree-like processes introduced in Sect. 2.3. In fact, in this case the infinite transition matrix has the following form

$$P = \begin{bmatrix} C_0 & \Lambda_1 & \Lambda_2 & \dots & \Lambda_d \\ V_1 & W & 0 & \dots & 0 \\ V_2 & 0 & W & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ V_d & 0 & \dots & 0 & W \end{bmatrix},$$

where C_0 is $m \times m$, $\Lambda_i = [A_i \ 0 \ 0 \ \dots]$, $V_i^T = [D_i^T \ 0 \ 0 \ \dots]$ and the matrix W is recursively defined by

$$W = \begin{bmatrix} C & \Lambda_1 & \Lambda_2 & \dots & \Lambda_d \\ V_1 & W & 0 & \dots & 0 \\ V_2 & 0 & W & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ V_d & 0 & \dots & 0 & W \end{bmatrix}.$$

The matrix W can be factorized as $W = UL$ where

$$U = \begin{bmatrix} S & A_1 & A_2 & \dots & A_d \\ 0 & U & 0 & \dots & 0 \\ 0 & 0 & U & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & o & U \end{bmatrix}, \quad L = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ Y_1 & L & 0 & \dots & 0 \\ Y_2 & 0 & L & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ Y_d & 0 & \dots & o & L \end{bmatrix}$$

and S is the minimal solution of $X + \sum_{i=1}^d A_i X^{-1} D_i = C$. Once the matrix S is known, the vector π can be computed by using the UL factorization of W . In order to solve the above equation, multiply $X + \sum_{i=1}^d A_i X^{-1} D_i = C$ to the right by $X^{-1} D_i$ for $i = 1, \dots, d$ and get

$$D_i + (C + \sum_{j=1, j \neq i}^d A_j X^{-1} D_j) X^{-1} D_i + A_i (X^{-1} D_i)^2 = 0,$$

that is, $X_i := X^{-1} D_i$ solves

$$D_i + (C + \sum_{j=1, j \neq i}^d A_j X_j) X_i + A_i X_i^2 = 0.$$

We can prove that X_i is the minimal solution. It can be computed by means of a fixed point iteration. Below we report different algorithms based on fixed point iterations analyzed in [13].

Algorithm Simple iteration

Set $X_{i,0} = 0$, $i = 1, \dots, d$

For $k = 0, 1, 2, \dots$

compute $S_k = C + \sum_{i=1}^d A_i X_{i,k}$;

set $X_{i,k+1} = -S_k^{-1} D_i$, for $i = 1, \dots, d$.

One can prove that the sequences $\{S_k\}_k$ and $\{X_{i,k}\}_k$ converge monotonically to S and to X_i , respectively.

Algorithm Fixed point + CR

Set $X_{i,0} = 0$, $i = 1, \dots, d$;

For $k = 0, 1, 2, \dots$

For $i = 1, \dots, d$

set $F_{i,k} = C + \sum_{j=1}^{i-1} A_j X_{j,k} + \sum_{j=i+1}^d A_j X_{j,k-1}$;

compute by means of CR the minimal solution $X_{i,k}$ of

$$D_i + F_{i,k} X + A_i X^2 = 0.$$

One can prove that the sequence $\{X_{i,k}\}_k$ converges monotonically to X_i for $i = 1, \dots, d$.

Algorithm Newton's iteration

Set $S_0 = C$;

For $k = 0, 1, 2, \dots$

compute $L_k = S_k - C + \sum_{i=1}^d A_i S_k^{-1} D_i$;

compute the solution Y_k of $X - \sum_{i=1}^d A_i S_k^{-1} X S_k^{-1} D_i = L_k$;

set $S_{k+1} = S_k - Y_k$.

The sequence $\{S_k\}_k$ converges quadratically to S . It is an open issue to design efficient algorithms for the solution of the above matrix equation.

6.2 Vector Equations

As shown in Sect. 2.3, the extinction probability vector in a Markovian binary tree is given by the minimal nonnegative solution \mathbf{x}^* of the vector equation $\mathbf{x} = \mathbf{a} + \mathbf{b}(\mathbf{x}, \mathbf{x})$, where $\mathbf{a} = (a_i)$ is a probability vector, and $w = \mathbf{b}(\mathbf{u}, \mathbf{v})$ is a bilinear form defined by $w_k = \sum_{i=1}^n \sum_{j=1}^n u_i v_j b_{i,j,k}$, for given $b_{i,j,k}$. Besides the minimal nonnegative solution \mathbf{x}^* , this equation has the vector $\mathbf{e} = (1, \dots, 1)$ as solution.

Some algorithms are reported below, see Bean et al. [7], and Hautphenne et al. [43, 44]

1. $\mathbf{x}_{k+1} = \mathbf{a} + \mathbf{b}(\mathbf{x}_k, \mathbf{x}_k)$, *depth algorithm*;
2. $\mathbf{x}_{k+1} = \mathbf{a} + \mathbf{b}(\mathbf{x}_k, \mathbf{x}_{k+1})$, *order algorithm*;
3. $\mathbf{x}_{k+1} = \mathbf{a} + \mathbf{b}(\mathbf{x}_k, \mathbf{x}_{k+1}), \mathbf{x}_{k+2} = \mathbf{a} + \mathbf{b}(\mathbf{x}_{k+2}, \mathbf{x}_{k+1})$, *thickness algorithm*;
4. $\mathbf{x}_{k+1} = (I - \mathbf{b}(\mathbf{x}_k, \cdot) - \mathbf{b}(\cdot, \mathbf{x}_k))^{-1}(\mathbf{a} - \mathbf{b}(\mathbf{x}_k, \mathbf{x}_k))$, *Newton's iteration*.

Convergence is monotonic with $\mathbf{x}_0 = 0$; iterations 1,2,3 have linear convergence, iteration 4 has quadratic convergence. Convergence turns to sublinear/linear when the problem is critical, i.e., if $\rho(R) = 1$, $R = B(I \otimes \mathbf{e} + \mathbf{e} \otimes I)$, and B is the $n \times n^2$ matrix associated with the bilinear form $\mathbf{b}(\mathbf{u}, \mathbf{v})$.

6.2.1 An Optimistic Approach

An “optimistic approach” has been given by Meini and Poloni [57]. Define $\mathbf{y} = \mathbf{e} - \mathbf{x}$ the vector of survival probability. Then the vector equation becomes

$$\mathbf{y} = \mathbf{b}(\mathbf{y}, \mathbf{e}) + \mathbf{b}(\mathbf{e}, \mathbf{y}) - \mathbf{b}(\mathbf{y}, \mathbf{y}),$$

where we are interested in the solution \mathbf{y}^* such that $0 \leq \mathbf{y}^* \leq \mathbf{x}^*$. The equation can be rewritten as

$$\mathbf{y} = H_h \mathbf{y}, \quad H_h = \mathbf{b}(\cdot, \mathbf{e}) + \mathbf{b}(\mathbf{e}, \cdot) - \mathbf{b}(\mathbf{y}, \cdot).$$

Observe that for $0 \leq \mathbf{y} < \mathbf{e}$, H_h is a nonnegative irreducible matrix. The Perron-Frobenius theorem insures that there exists a positive eigenvector. This observation

leads to the following iteration

$$\mathbf{y}_{k+1} = \text{PerronVector}(H_{\mathbf{y}_k}).$$

Local convergence of this iteration can be proved. Convergence is linear in the noncritical case and super-linear in the critical case.

7 Exponential of a Block Triangular Block Toeplitz Matrix

A nice application of the Toeplitz matrix machinery concerns the computation of the exponential of a block triangular block Toeplitz matrix encountered in the Erlangian approximation of Markovian fluid queues described in Sect. 2.3. In this case, one has to compute

$$\mathcal{Y} = e^{\mathcal{X}} = \sum_{i=0}^{\infty} \frac{1}{i!} \mathcal{X}^i$$

where \mathcal{X} is the $(\ell + 1) \times (\ell + 1)$ block triangular block Toeplitz matrix defined by the $m \times m$ blocks X_0, \dots, X_ℓ defined below

$$\mathcal{X} = \begin{bmatrix} X_0 & X_1 & \dots & X_\ell \\ \ddots & \ddots & & \vdots \\ & X_0 & X_1 & \\ & & X_0 & \end{bmatrix}.$$

In our assumptions, \mathcal{X} has negative diagonal entries, nonnegative off-diagonal entries, moreover the sum of the entries in each row is nonpositive. Clearly, since block triangular Toeplitz matrices form a matrix algebra then \mathcal{Y} is still block triangular Toeplitz.

Here the question is: What is the most convenient way to compute \mathcal{Y} in terms of CPU time and error?

Embed \mathcal{X} into an infinite block triangular block Toeplitz matrix \mathcal{X}_∞ obtained by completing the sequence X_0, X_1, \dots, X_ℓ with zeros:

$$\mathcal{X}_\infty = \begin{bmatrix} X_0 & \dots & X_\ell & 0 & \dots & \dots \\ & X_0 & \ddots & X_\ell & 0 & \ddots \\ & & \ddots & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & \ddots \end{bmatrix}.$$

Denote Y_0, Y_1, \dots the blocks defining $\mathcal{Y}_\infty = e^{\mathcal{X}_\infty}$. Then \mathcal{Y} is the $(\ell + 1) \times (\ell + 1)$ principal submatrix of \mathcal{Y}_∞ .

We can prove the following decay property

$$\|Y_i\|_\infty \leq e^{\alpha(\sigma^{\ell-1}-1)}\sigma^{-i}, \quad \forall \sigma > 1$$

where $\alpha = \max_j(-(X_0)_{j,j})$. This property is fundamental to prove error bounds of the following different algorithms.

7.1 Using ϵ -Circulant Matrices

Approximate \mathcal{X} with an ϵ -circulant matrix $\mathcal{X}^{(\epsilon)}$ and approximate Y with $\mathcal{Y}^{(\epsilon)} = e^{\mathcal{X}^{(\epsilon)}}$. We can prove that if, $\beta = \| [X_1, \dots, X_\ell] \|_\infty$ then

$$\|\mathcal{Y} - \mathcal{Y}^{(\epsilon)}\|_\infty \leq e^{|\epsilon|\beta} - 1 = |\epsilon|\beta + O(|\epsilon|^2)$$

and, if ϵ is purely imaginary then

$$\|\mathcal{Y} - \mathcal{Y}^{(\epsilon)}\|_\infty \leq e^{|\epsilon|^2\beta} - 1 = |\epsilon|^2\beta + O(|\epsilon|^4).$$

7.2 Using Circulant Matrices

Embed \mathcal{X} into a $K \times K$ block circulant matrix $\mathcal{X}^{(K)}$ for $K > \ell$ large, and approximate \mathcal{Y} with the $(\ell+1) \times (\ell+1)$ submatrix $\mathcal{Y}^{(K)}$ of $e^{\mathcal{X}^{(K)}}$.

We can prove the following bound

$$\|[Y_0 - Y_0^{(K)}, \dots, Y_\ell - Y_\ell^{(K)}]\|_\infty \leq (e^\beta - 1)e^{\alpha(\sigma^{\ell-1}-1)} \frac{\sigma^{-K+\ell}}{1-\sigma^{-1}}, \quad \sigma > 1.$$

7.3 Method Based on Taylor Expansion

The matrix \mathcal{Y} is approximated by truncating the series expansion to r terms

$$\mathcal{Y}^{(r)} = \sum_{i=0}^r \frac{1}{i!} \mathcal{X}^i.$$

In all the three approaches, the computation remains inside a matrix algebra, more specifically: block ϵ -circulant matrices of fixed size ℓ , in the first case; block circulant matrices of variable size $K > \ell$, in the second case; block triangular

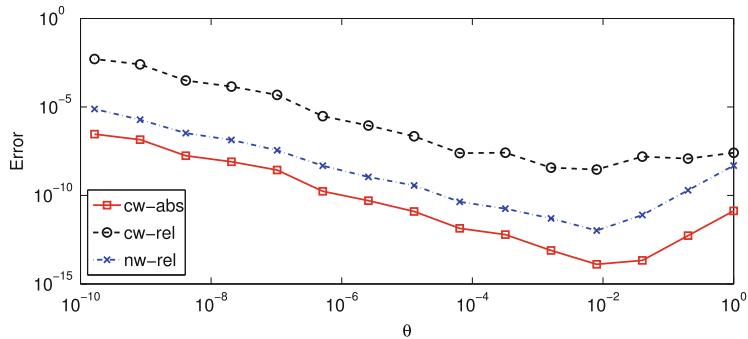


Fig. 13 Norm-wise error, component-wise relative and absolute errors for the solution obtained with the algorithm based on ϵ -circulant matrices with $\epsilon = i\theta$

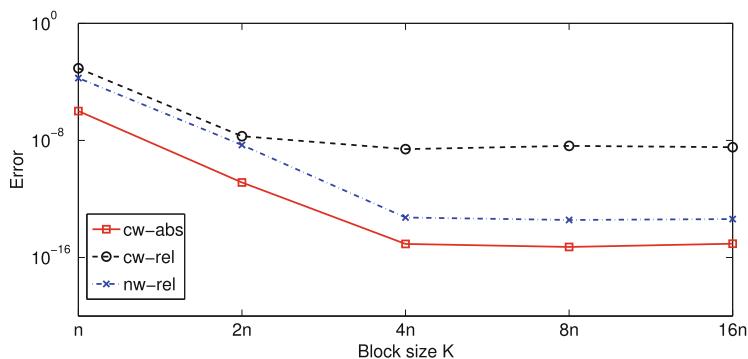


Fig. 14 Norm-wise error, component-wise relative and absolute errors for the solution obtained with the algorithm based on circulant embedding for different values of the embedding size K

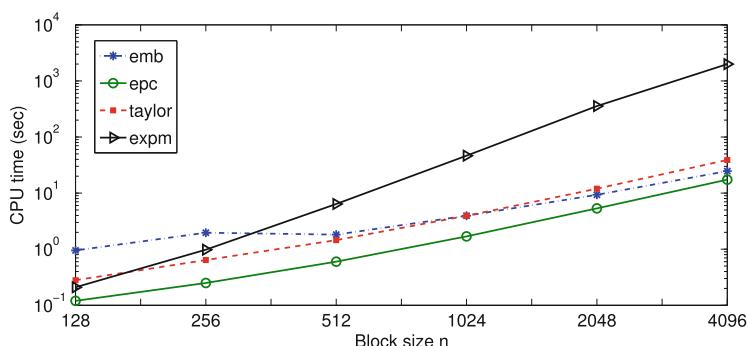


Fig. 15 CPU time of the Matlab function `expm`, and of the algorithms based on ϵ -circulant, circulant embedding, power series expansion

Toeplitz matrices of fixed size ℓ in the third case. Figs. 13, 14, and 15 report errors and CPU time of the above approaches.

References

1. A.H. Al-Mohy, N.J. Higham, The complex step approximation to the Fréchet derivative of a matrix function. *Numer. Algorithms* **53**(1), 113–148 (2010)
2. G.S. Ammar, W.B. Gragg, Superfast solution of real positive definite Toeplitz systems. *SIAM J. Matrix Anal. Appl.* **9**(1), 61–76 (1988). SIAM Conference on Linear Algebra in Signals, Systems, and Control (Boston, MA, 1986)
3. A. Aricò, G. Rodriguez, A fast solver for linear systems with displacement structure. *Numer. Algorithms* **55**(4), 529–556 (2010)
4. J.R. Artalejo, A. Gómez-Corral, Q. He, Markovian arrivals in stochastic modelling: a survey and some new results. *Stat. Oper. Res. Trans.* **34**(2), 101–144 (2010)
5. S. Asmussen, M. Bladt, Poisson’s equation for queues driven by a Markovian marked point process. *Queueing Syst.* **17**(1–2), 235–274 (1994)
6. S. Asmussen, F. Avram, M. Usábel, Erlangian approximations for finite-horizon ruin probabilities. *ASTIN Bull.* **32**, 267–281 (2002)
7. N.G. Bean, N. Kontoleon, P.G. Taylor, Markovian trees: properties and algorithms. *Ann. Oper. Res.* **160**, 31–50 (2008)
8. D. Bini, M. Capovani, Spectral and computational properties of band symmetric Toeplitz matrices. *Linear Algebra Appl.* **52/53**, 99–126 (1983)
9. D. Bini, F. Di Benedetto, New preconditioner for the parallel solution of positive definite Toeplitz systems, in *Proceedings of the Second Annual ACM Symposium on Parallel Algorithms and Architectures*, Island of Crete, 02–06 July 1990 (ACM, New York, 1990), pp. 220–223. doi:10.1145/97444.97688
10. D. Bini, P. Favati, On a matrix algebra related to the discrete Hartley transform. *SIAM J. Matrix Anal. Appl.* **14**(2), 500–507 (1993)
11. D.A. Bini, B. Meini, The cyclic reduction algorithm: from Poisson equation to stochastic processes and beyond. In memoriam of Gene H. Golub. *Numer. Algorithms* **51**(1), 23–60 (2009)
12. D. Bini, V.Y. Pan, *Polynomial and Matrix Computations: Fundamental Algorithms*, vol. 1. Progress in Theoretical Computer Science (Birkhäuser, Boston, MA, 1994)
13. D.A. Bini, G. Latouche, B. Meini, Solving nonlinear matrix equations arising in tree-like stochastic processes. *Linear Algebra Appl.* **366**, 39–64 (2003). Special issue on structured matrices: analysis, algorithms and applications (Cortona, 2000)
14. D.A. Bini, G. Latouche, B. Meini, *Numerical Methods for Structured Markov Chains*. Numerical Mathematics and Scientific Computation. Oxford Science Publications (Oxford University Press, New York, 2005)
15. D.A. Bini, B. Iannazzo, B. Meini, Numerical solution of algebraic Riccati equations, in *Fundamentals of Algorithms*, vol. 9 (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2012)
16. R.R. Bitmead, B.D.O. Anderson, Asymptotically fast solution of Toeplitz and related systems of linear equations. *Linear Algebra Appl.* **34**, 103–116 (1980)
17. P. Boito, Y. Eidelman, L. Gemignani, I. Gohberg, Implicit QR with compression. *Indag. Math. (NS)* **23**(4), 733–761 (2012)
18. P. Boito, Y. Eidelman, L. Gemignani, Implicit QR for rank-structured matrix pencils. *BIT* **54**(1), 85–111 (2014)
19. S. Börm, L. Grasedyck, W. Hackbusch, Hierarchical matrices. Technical Report 21, Max Plank Institute für Mathematik (2003)

20. A. Böttcher, S.M. Grudsky, *Toeplitz Matrices, Asymptotic Linear Algebra, and Functional Analysis* (Birkhäuser, Basel, 2000)
21. A. Böttcher, S.M. Grudsky, *Spectral Properties of Banded Toeplitz Matrices* (Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005)
22. A. Böttcher, B. Silbermann, *Introduction to Large Truncated Toeplitz Matrices*. Universitext (Springer, New York, 1999)
23. A. Brauer, Limits for the characteristic roots of a matrix. IV. Applications to stochastic matrices. *Duke Math. J.* **19**, 75–91 (1952)
24. B.L. Buzbee, G.H. Golub, C.W. Nielson, On direct methods for solving Poisson's equations. *SIAM J. Numer. Anal.* **7**, 627–656 (1970)
25. T.F. Chan, An optimal circulant preconditioner for Toeplitz systems. *SIAM J. Sci. Stat. Comput.* **9**(4), 766–771 (1988)
26. R.H. Chan, The spectrum of a family of circulant preconditioned Toeplitz systems. *SIAM J. Numer. Anal.* **26**(2), 503–506 (1989)
27. R.H. Chan, Toeplitz preconditioners for Toeplitz systems with nonnegative generating functions. *IMA J. Numer. Anal.* **11**(3), 333–345 (1991)
28. R.H.-F. Chan, X.-Q. Jin, *An Introduction to Iterative Toeplitz Solvers*. Fundamentals of Algorithms, vol. 5 (Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007)
29. C.-Y. Chiang, E.K.-W. Chu, C.-H. Guo, T.-M. Huang, W.-W. Lin, S.-F. Xu, Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case. *SIAM J. Matrix Anal. Appl.* **31**(2), 227–247 (2009)
30. F. de Hoog, A new algorithm for solving Toeplitz systems of equations. *Linear Algebra Appl.* **88/89**, 123–138 (1987)
31. S. Dendievel, G. Latouche, Approximation for time-dependent distributions in Markovian fluid models. *Methodol. Comput. Appl. Probab.* (2016). doi:10.1007/s11009-016-9480
32. S. Dendievel, G. Latouche, Y. Liu, Poisson's equation for discrete-time quasi-birth-and-death processes. *Perform. Eval.* **70**(9), 564–577 (2013)
33. F. Di Benedetto, Analysis of preconditioning techniques for ill-conditioned Toeplitz matrices. *SIAM J. Sci. Comput.* **16**(3), 682–697 (1995)
34. Y. Eidelman, I. Gohberg, I. Haimovici, *Separable Type Representations of Matrices and Fast Algorithms: Basics. Completion Problems. Multiplication and Inversion Algorithms*, vol. 1. Operator Theory: Advances and Applications, vol. 234 (Birkhäuser/Springer, Basel, 2014)
35. Y. Eidelman, I. Gohberg, I. Haimovici, *Separable Type Representations of Matrices and Fast Algorithms: Eigenvalue Method*, vol. 2. Operator Theory: Advances and Applications, vol. 235 (Birkhäuser/Springer, Basel AG, Basel, 2014)
36. F.R. Gantmacher, M.G. Krein, Sur les matrices oscillatoires et complètement non négatives. *Compos. Math.* **4**, 445–476 (1937)
37. C. Garoni, S. Serra Capizzano, The theory of generalized locally Toeplitz sequences: a review, an extension, and a few representative applications. Technical Report, Università dell'Insubria (2015)
38. I.C. Gohberg, A.A. Semencul, The inversion of finite Toeplitz matrices and their continual analogues. *Mat. Issled.* **7**(2(24)), 201–223, 290 (1972)
39. I. Gohberg, T. Kailath, V. Olshevsky, Fast Gaussian elimination with partial pivoting for matrices with displacement structure. *Math. Comput.* **64**(212), 1557–1576 (1995)
40. W.K. Grassmann, M.I. Taksar, D.P. Heyman, Regenerative analysis and steady state distributions for Markov chains. *Oper. Res.* **33**(5), 1107–1116 (1985)
41. U. Grenander, G. Szegő, *Toeplitz Forms and Their Applications*, 2nd edn. (Chelsea Publishing, New York, 1984)
42. A. Grothendieck, Recoltes et semailles, réflexions et témoignage sur un passé de mathématicien (1985). <http://lipn.univ-paris13.fr/~duchamp/Books&more/Grothendieck/RS/pdf/RetS.pdf>
43. S. Hautphenne, G. Latouche, M.-A. Remiche, Newton's iteration for the extinction probability of a Markovian binary tree. *Linear Algebra Appl.* **428**(11–12), 2791–2804 (2008)

44. S. Hautphenne, G. Latouche, M.-A. Remiche, Algorithmic approach to the extinction probability of branching processes. *Methodol. Comput. Appl. Probab.* **13**(1), 171–192 (2011)
45. C. He, B. Meini, N.H. Rhee, A shifted cyclic reduction algorithm for quasi-birth-death problems. *SIAM J. Matrix Anal. Appl.* **23**(3), 673–691 (2001/2002) (electronic)
46. P. Henrici, *Applied and Computational Complex Analysis: Power Series—Integration—Conformal Mapping—Location of Zeros*, vol. 1. Wiley Classics Library (Wiley, New York, 1988). Reprint of the 1974 original. A Wiley-Interscience Publication
47. N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd edn. (Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002)
48. R.W. Hockney, A fast direct solution of Poisson’s equation using Fourier analysis. *J. Assoc. Comput. Mach.* **12**, 95–113 (1965)
49. T. Huckle, S. Serra Capizzano, C. Tablino-Possio, Preconditioning strategies for non-Hermitian Toeplitz linear systems. *Numer. Linear Algebra Appl.* **12**(2–3), 211–220 (2005)
50. T. Kailath, V. Olshevsky, Displacement structure approach to discrete-trigonometric-transform based preconditioners of G. Strang type and of T. Chan type. *SIAM J. Matrix Anal. Appl.* **26**(3), 706–734 (2005) (electronic)
51. T. Kailath, A.H. Sayed, Displacement structure: theory and applications. *SIAM Rev.* **37**(3), 297–386 (1995)
52. D.G. Kendall, Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded Markov chain. *Ann. Math. Stat.* **24**, 338–354 (1953)
53. D.P. Kroese, W.R.W. Scheinhardt, P.G. Taylor, Spectral properties of the tandem Jackson network, seen as a quasi-birth-and-death process. *Ann. Appl. Probab.* **14**(4), 2057–2089 (2004)
54. G. Latouche, V. Ramaswami, *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability (Society for Industrial and Applied Mathematics (SIAM)/American Statistical Association, Philadelphia, PA/Alexandria, VA, 1999)
55. N. Levinson, The Wiener rms error criterion in filter design and prediction. *J. Math. Phys.* **25**, 261–278 (1947)
56. A.M. Makowski, A. Shwartz, The Poisson equation for countable Markov chains: probabilistic methods and interpretations, in *Handbook of Markov Decision Processes* (Springer, Berlin, 2002)
57. B. Meini, F. Poloni, A Perron iteration for the solution of a quadratic vector equation arising in Markovian binary trees. *SIAM J. Matrix Anal. Appl.* **32**(1), 248–261 (2011)
58. M. Miyazawa, Tail decay rates in double QBD processes and related reflected random walks. *Math. Oper. Res.* **34**(3), 547–575 (2009)
59. B.R. Musicus, Levinson and fast Choleski algorithms for Toeplitz and almost Toeplitz matrices. Technical Report 538, Research Laboratory of Electronics Massachusetts Institute of Technology, Cambridge, MA (1988). <https://dspace.mit.edu/bitstream/handle/1721.1/4954/RLE-TR-538-20174000.pdf?sequence=1>
60. M.F. Neuts, *Structured Stochastic Matrices of M/G/1 Type and Their Applications. Probability: Pure and Applied*, vol. 5 (Dekker, New York, 1989)
61. M.F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach* (Dover Publications, New York, 1994). Corrected reprint of the 1981 original
62. J.R. Norris, *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics, vol. 2 (Cambridge University Press, Cambridge, 1998). Reprint of 1997 original
63. D. Noutsos, S. Serra Capizzano, P. Vassalos, Matrix algebra preconditioners for multilevel Toeplitz systems do not insure optimal convergence rate. *Theor. Comput. Sci.* **315**(2–3), 557–579 (2004)
64. A. Ostrowski, Recherches sur la méthode de Graeffe et les zéros des polynomes et des séries de Laurent. *Acta Math.* **72**, 99–155 (1940)
65. A. Ostrowski, Recherches sur la méthode de Graeffe et les zéros des polynomes et des séries de Laurent. Chapitres III et IV. *Acta Math.* **72**, 157–257 (1940)
66. F. Poloni, A note on the $O(n)$ -storage implementation of the GKO algorithm and its adaptation to Trummer-like matrices. *Numer. Algorithms* **55**(1), 115–139 (2010)

67. V. Ramaswami, G. Latouche, A general class of Markov processes with explicit matrix-geometric solutions. *OR Spektrum* **8**(4), 209–218 (1986)
68. S. Serra, Optimal, quasi-optimal and superlinear band-Toeplitz preconditioners for asymptotically ill-conditioned positive definite Toeplitz systems. *Math. Comput.* **66**(218), 651–665 (1997)
69. S. Serra Capizzano, Toeplitz:matrices: spectral properties and preconditioning in the CG method. Technical report, Università dell'Insubria (2007)
70. S. Serra Capizzano, E. Tyrtyshnikov, How to prove that a preconditioner cannot be superlinear. *Math. Comput.* **72**(243), 1305–1316 (2003)
71. Z. Sheng, P. Dewilde, S. Chandrasekaran, Algorithms to solve hierarchically semi-separable systems, in *System Theory, The Schur Algorithm and Multidimensional Analysis*. Operator Theory Advances and Applications, vol. 176 (Birkhäuser, Basel, 2007), pp. 255–294
72. W.J. Stewart, *Introduction to the Numerical Solution of Markov Chains* (Princeton University Press, Princeton, NJ, 1994)
73. G. Strang, A proposal for Toeplitz matrix calculations. *Stud. Appl. Math.* **74**(2), 171–176 (1986)
74. P. Tilli, Locally Toeplitz sequences: spectral properties and applications. *Linear Algebra Appl.* **278**(1–3), 91–120 (1998)
75. W.F. Trench, An algorithm for the inversion of finite Toeplitz matrices. *J. Soc. Ind. Appl. Math.* **12**, 515–522 (1964)
76. E.E. Tyrtyshnikov, A unifying approach to some old and new theorems on distribution and clustering. *Linear Algebra Appl.* **232**, 1–43 (1996)
77. R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices: Linear Systems*, vol. 1 (Johns Hopkins University Press, Baltimore, MD, 2008)
78. R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices: Eigenvalue and Singular Value Methods*, vol. II (Johns Hopkins University Press, Baltimore, MD, 2008)
79. R.S. Varga, *Matrix Iterative Analysis* (Prentice-Hall, Englewood Cliffs, NJ, 1962)
80. R.S. Varga, *Matrix Iterative Analysis*. Springer Series in Computational Mathematics, vol. 27, Expanded edition (Springer, Berlin, 2000)
81. H. Widom, On the singular values of Toeplitz matrices. *Z. Anal. Anwendungen* **8**(3), 221–229 (1989)
82. J. Xia, S. Chandrasekaran, M. Gu, X.S. Li, Fast algorithms for hierarchically semiseparable matrices. *Numer. Linear Algebra Appl.* **17**(6), 953–976 (2010)
83. S. Zohar, Toeplitz matrix inversion: the algorithm of W. F. Trench. *J. Assoc. Comput. Mach.* **16**, 592–601 (1969)

Matrices with Hierarchical Low-Rank Structures

Jonas Ballani and Daniel Kressner

Abstract Matrices with low-rank off-diagonal blocks are a versatile tool to perform matrix compression and to speed up various matrix operations, such as the solution of linear systems. Often, the underlying block partitioning is described by a hierarchical partitioning of the row and column indices, thus giving rise to hierarchical low-rank structures. The goal of this chapter is to provide a brief introduction to these techniques, with an emphasis on linear algebra aspects.

1 Introduction

1.1 Sparsity Versus Data-Sparsity

An $n \times n$ matrix A is called *sparse* if $\text{nnz}(A)$, the number of its nonzero entries, is much smaller than n^2 . Sparsity reduces the storage requirements to $O(\text{nnz}(A))$ and, likewise, the complexity of matrix-vector multiplications. Unfortunately, sparsity is fragile; it is easily destroyed by operations with the matrix. For example, the factors of an LU factorization of A usually have (many) more nonzeros than A itself. This so called fill-in can be reduced, sometimes significantly, by reordering the matrix prior to the factorization [25]. Based on existing numerical evidence, it is probably safe to say that such sparse factorizations constitute the methods of choice for (low-order) finite element or finite difference discretizations of two-dimensional partial differential equations (PDEs). They are less effective for three-dimensional PDEs. More severely, sparse factorizations are of limited use when the inverse of A is explicitly needed, as for example in inverse covariance matrix estimation and matrix iterations for computing matrix functions. For nearly all sparsity patterns of practical relevance, A^{-1} is a completely dense matrix. In certain situations, this issue can be addressed effectively by considering approximate sparsity: A^{-1} is replaced by a

J. Ballani • D. Kressner (✉)

ANCHP, EPF Lausanne, Station 8, CH-1015 Lausanne, Switzerland
e-mail: jonas.ballani@epfl.ch; daniel.kressner@epfl.ch

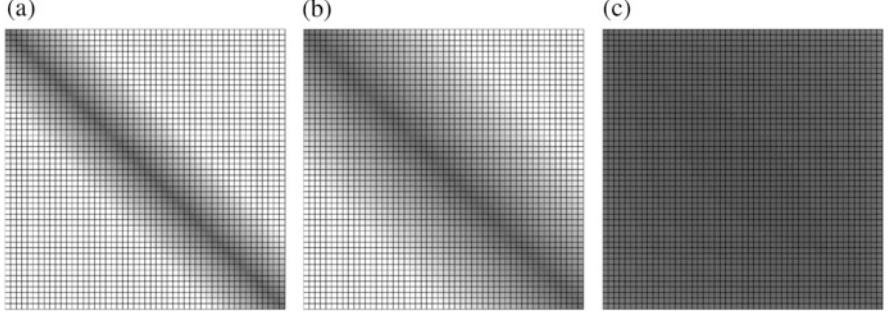


Fig. 1 Approximate sparsity of A^{-1} for the matrix A from (1) with $n = 50$ and different values of σ . (a) $\sigma = 4, \kappa(A) \approx 2$. (b) $\sigma = 1, \kappa(A) \approx 5$. (c) $\sigma = 0, \kappa(A) \approx 10^3$

sparse matrix M such that

$$\|A^{-1} - M\| \leq \text{tol}$$

for some prescribed tolerance tol and an appropriate matrix norm $\|\cdot\|$.

To illustrate the concepts mentioned above, let us consider the tridiagonal matrix

$$A = (n+1)^2 \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} + \sigma(n+1)^2 I_n, \quad (1)$$

where the shift $\sigma > 0$ controls the condition number $\kappa(A) = \|A\|_2 \|A^{-1}\|_2$. The intensity plots in Fig. 1 reveal the magnitude of the entries of A^{-1} ; entries of magnitude below 10^{-15} are white. It can be concluded that A^{-1} can be very well approximated by a banded matrix if its condition number is small. This is in accordance with a classical result [26, Proposition 2.1], which states for any symmetric positive definite tridiagonal matrix A that

$$|[A^{-1}]_{ij}| \leq C \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{|i-j|}, \quad C = \max \{ \lambda_{\min}^{-1}, (2\lambda_{\max})^{-1}(1 + \sqrt{\kappa(A)})^2 \}, \quad (2)$$

where $\lambda_{\min}, \lambda_{\max}$ denote the smallest/largest eigenvalues of A ; see also [15] for recent extensions. This implies that the entries of A^{-1} decay exponentially away from the diagonal, but this decay may deteriorate as $\kappa(A)$ increases. Although the bound (2) can be pessimistic, it correctly predicts that approximate sparsity is of limited use for ill-conditioned matrices as they arise from the discretization of (one-dimensional) PDEs.

An $n \times n$ matrix is called *data-sparse* if it can be represented by much less than n^2 parameters with respect to a certain format. Data-sparsity is much more general than sparsity. For example, any matrix of rank $r \ll n$ can be represented with $2nr$ (or even less) parameters by storing its low-rank factors. The inverse of the tridiagonal matrix (1) is obviously not a low-rank matrix, simply because it is invertible. However, any of its off-diagonal blocks has rank 1. One of the many ways to see is to use the Sherman-Morrison formula. Consider the following decomposition of a general symmetric positive definite tridiagonal matrix:

$$A = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix} - a_{n_1, n_1+1} \begin{pmatrix} e_{n_1} \\ -e_1 \end{pmatrix} \begin{pmatrix} e_{n_1} \\ -e_1 \end{pmatrix}^T, \quad A_{11} \in \mathbb{R}^{n_1 \times n_1}, A_{22} \in \mathbb{R}^{n_2 \times n_2},$$

where e_j denotes a j th unit vector of appropriate length. Then

$$A^{-1} = \begin{pmatrix} A_{11}^{-1} & 0 \\ 0 & A_{22}^{-1} \end{pmatrix} + \frac{a_{n_1, n_1+1}}{1 + e_{n_1}^T A_{11}^{-1} e_{n_1} + e_1^T A_{22}^{-1} e_1} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}^T, \quad (3)$$

where $w_1 = A_{11}^{-1} e_{n_1}$ and $w_2 = -A_{22}^{-1} e_1$. Hence, the off-diagonal blocks of A^{-1} have rank at most 1. Assuming that n is an integer multiple of 4, let us partition

$$A = \left(\begin{array}{c|cc|c} \begin{array}{c|c} A_{11}^{(2)} & A_{12}^{(2)} \\ \hline A_{21}^{(2)} & A_{22}^{(2)} \end{array} & A_{12} \\ \hline A_{21} & \begin{array}{c|c} A_{33}^{(2)} & A_{34}^{(2)} \\ \hline A_{43}^{(2)} & A_{44}^{(2)} \end{array} \end{array} \right), \quad A^{-1} = \left(\begin{array}{c|cc|c} \begin{array}{c|c} B_{11}^{(2)} & B_{12}^{(2)} \\ \hline B_{21}^{(2)} & B_{22}^{(2)} \end{array} & B_{12} \\ \hline B_{34} & \begin{array}{c|c} B_{33}^{(2)} & B_{34}^{(2)} \\ \hline B_{43}^{(2)} & B_{44}^{(2)} \end{array} \end{array} \right), \quad (4)$$

such that $A_{ij}^{(2)}, B_{ij}^{(2)} \in \mathbb{R}^{n/4 \times n/4}$. By storing only the factors of the rank-1 off-diagonal blocks of A^{-1} and exploiting symmetry, it follows that – additionally to the 4 diagonal blocks $B_{jj}^{(2)}$ of size $n/4 \times n/4$ – only $2n/2 + 4n/4 = 2n$ entries are needed to represent A^{-1} . Assuming that $n = 2^k$ and partitioning recursively, it follows that the whole matrix A^{-1} can be stored with

$$2n/2 + 4n/4 + \cdots + 2^k n/2^k + n = n(\log_2 n + 1) \quad (5)$$

parameters.

The logarithmic factor in (5) can actually be removed by exploiting the nestedness of the low-rank factors. To see this, we again consider the partitioning (4) and let $U_j^{(2)} \in \mathbb{R}^{n/4 \times 2}, j = 1, \dots, 4$, be orthonormal bases such that

$$\text{span} \left\{ (A_{jj}^{(2)})^{-1} e_1, (A_{jj}^{(2)})^{-1} e_{n/4} \right\} \subseteq \text{range}(U_j^{(2)}).$$

Applying the Sherman-Morrison formula (3) to $A_{11} = \begin{pmatrix} A_{11}^{(2)} & A_{12}^{(2)} \\ A_{21}^{(2)} & A_{22}^{(2)} \end{pmatrix}$ shows

$$A_{11}^{-1} e_1 \in \text{range} \begin{pmatrix} U_1^{(2)} & 0 \\ 0 & U_2^{(2)} \end{pmatrix}, \quad A_{11}^{-1} e_{n/2} \in \text{range} \begin{pmatrix} U_1^{(2)} & 0 \\ 0 & U_2^{(2)} \end{pmatrix}.$$

Similarly,

$$A_{22}^{-1} e_1 \in \text{range} \begin{pmatrix} U_3^{(2)} & 0 \\ 0 & U_4^{(2)} \end{pmatrix}, \quad A_{22}^{-1} e_{n/2} \in \text{range} \begin{pmatrix} U_3^{(2)} & 0 \\ 0 & U_4^{(2)} \end{pmatrix}.$$

If we let $U_j \in \mathbb{R}^{n/2 \times 2}$, $j = 1, 2$, be orthonormal basis such that

$$\text{span} \{A_{jj}^{-1} e_1, A_{jj}^{-1} e_{n/2}\} \subseteq \text{range}(U_j),$$

then there exists matrices $X_j \in \mathbb{R}^{4 \times 2}$ such that

$$U_j = \begin{pmatrix} U_{2j-1}^{(2)} & 0 \\ 0 & U_{2j}^{(2)} \end{pmatrix} X_j. \quad (6)$$

Hence, there is no need to store the bases $U_1, U_2 \in \mathbb{R}^{n/2 \times 2}$ explicitly; the availability of $U_j^{(2)}$ and the small matrices X_1, X_2 suffices. In summary, we can represent A^{-1} as

$$\left(\begin{array}{c|c} \frac{B_{11}^{(2)}}{U_2^{(2)}(S_{12}^{(2)})^T(U_1^{(2)})^T} & \frac{|U_1^{(2)} S_{12}^{(2)} (U_2^{(2)})^T|}{B_{22}^{(2)}} \\ \hline U_2 S_{12}^T U_1^T & \frac{B_{33}^{(2)}}{|U_4^{(2)}(S_{34}^{(2)})^T(U_3^{(2)})^T|} \end{array} \begin{array}{c} U_1 S_{12} U_2^T \\ \hline \frac{|U_3^{(2)} S_{34}^{(2)} (U_4^{(2)})^T|}{B_{44}^{(2)}} \end{array} \right) \quad (7)$$

for some matrices $S_{12}, S_{ij}^{(2)} \in \mathbb{R}^{2 \times 2}$. The number of entries needed for representing the off-diagonal blocks A^{-1} is thus

$$\underbrace{4 \times 2n/4}_{\text{for } U_j^{(2)}} + \underbrace{2 \times 8}_{\text{for } X_j} + \underbrace{(2+1) \times 4}_{\text{for } S_{12}, S_{12}^{(2)}, S_{34}^{(2)}}.$$

Assuming that $n = 2^k$ and recursively repeating the described procedure $k-1$ times, we find that at most $O(n)$ total storage is needed for representing A^{-1} . Thus we have removed the logarithmic factor from (5), at the expense of a larger constant due to the fact that we work with rank 2 instead of rank 1.

Remark 1 The format (7) together with (6) is a special case of the HSS matrices that will be discussed in Sect. 4.1. The related but different concepts of *semi-separable*

matrices [65] and *sequentially semi-separable* matrices [21] allow to represent the inverse of a tridiagonal matrix A with nested rank-1 factors and consequently reduce the required storage to two vectors of length n . \diamond

For a number of reasons, the situation discussed above is simplistic and not fully representative for the type of applications that can be addressed with the hierarchical low-rank structures covered in this chapter. First, sparsity of the original matrix can be beneficial but it is not needed in the context of low-rank techniques. Second, exact low rank is a rare phenomenon and we will nearly always work with low-rank *approximations* instead. Third, the recursive partitioning (4) is closely tied to banded matrices and related structures, which arise, for example, from the discretization of one-dimensional PDEs. More general partitionings need to be used when dealing with structures that arise from the discretization of higher-dimensional PDEs.

1.2 Applications of Hierarchical Low-Rank Structures

The hierarchical low-rank structures discussed in this chapter (HODLR, \mathcal{H} , HSS, \mathcal{H}^2) have been used in a wide variety of applications. The purpose of the following selection is to give a taste for the variety of applications; it is by no means complete:

- Discretization of (boundary) integral operators [5, 6, 8, 9, 17, 18, 42, 57];
- HSS techniques for integral equations [24, 32, 39, 53];
- Hierarchical low rank approximation combined with (sparse) direct factorizations [2, 31, 54, 56, 69, 72];
- Preconditioning with low-rank Schur complements and HSS techniques [29];
- \mathcal{H} -matrix based preconditioning for finite element discretization of Maxwell equations [55];
- Matrix sign function iteration in \mathcal{H} -matrix arithmetic for solving matrix Lyapunov and Riccati equations [4, 37];
- Combination of contour integrals and \mathcal{H} -matrices for computing matrix functions [30];
- Eigenvalue computation [12–14, 68];
- Kernel approximation in machine learning [61, 70];
- Sparse covariance matrix estimation [1, 3] and Kalman filtering [50];
- Clustered low-rank approximation of graphs [58].

1.3 Outline

The rest of this chapter is structured as follows. In Sect. 2, we spend considerable attention to finding low-rank approximations of matrices. In particular, we discuss various algorithms used in practice and give a priori error estimates that offer insight

into the ability of performing such approximations. Section 3 combines low-rank approximations with block partitionings, significantly extending the construction in (4). Finally, in Sect. 4, we additionally impose nestedness on the low-rank factors, analogous to (6).

There are numerous excellent monographs [6, 17, 42, 66, 67], survey papers, and lecture notes on matrices with hierarchical low-rank structures. This chapter possibly complements the existing literature by serving as a broad but compact introduction to the field.

2 Low-Rank Approximation

2.1 The SVD and Best Low-Rank Approximation

The basis of everything in this chapter is, in one way or the other, the singular value decomposition (SVD).

Theorem 1 (SVD) *Let $A \in \mathbb{R}^{m \times n}$ with $m \geq n$. Then there are orthogonal matrices $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ such that*

$$A = U \Sigma V^T, \quad \text{with} \quad \Sigma = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & 0 & \end{pmatrix} \in \mathbb{R}^{m \times n}$$

and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$.

The restriction $m \geq n$ in Theorem 1 is for notational convenience only. An analogous result holds for $m < n$, which can be seen by applying Theorem 1 to A^T . Given an SVD $A = U \Sigma V^T$, the nonnegative scalars $\sigma_1, \dots, \sigma_n$ are called the **singular values**, the columns v_1, \dots, v_n of V **right singular vectors**, and the columns u_1, \dots, u_m of U **left singular vectors** of A .

In MATLAB, the SVD of a matrix A is computed by typing `[U, D, V] = svd(A)`. If A is not square, say $m > n$, a reduced or “economic” SVD is computed when typing `[U, D, V] = svd(A, 'econ')`. This yields a matrix $U \in \mathbb{R}^{m \times n}$ with orthonormal columns and an orthogonal matrix $V \in \mathbb{R}^{n \times n}$ such that

$$A = U \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{pmatrix} V^T.$$

For $r \leq n$, let

$$U_k := (u_1 \cdots u_k), \quad \Sigma_k := \text{diag}(\sigma_1, \dots, \sigma_k), \quad V_k := (v_1 \cdots v_k).$$

Then the matrix

$$\mathcal{T}_k(A) := U_k \Sigma_k V_k \quad (8)$$

clearly has rank at most k . For any class of unitarily invariant norms $\|\cdot\|$, we have

$$\|\mathcal{T}_k(A) - A\| = \|\text{diag}(\sigma_{k+1}, \dots, \sigma_n)\|$$

In particular, the spectral norm and the Frobenius norm give

$$\|A - \mathcal{T}_k(A)\|_2 = \sigma_{k+1}, \quad \|A - \mathcal{T}_k(A)\|_F = \sqrt{\sigma_{k+1}^2 + \dots + \sigma_n^2}. \quad (9)$$

This turns out to be optimal.

Theorem 2 (Schmidt-Mirsky Theorem) *For $A \in \mathbb{R}^{m \times n}$, let $\mathcal{T}_k(A)$ be defined as in (8). Then*

$$\|A - \mathcal{T}_k(A)\| = \min \left\{ \|A - B\| : B \in \mathbb{R}^{m \times n} \text{ has rank at most } k \right\}$$

holds for any unitarily invariant norm $\|\cdot\|$.

Proof Because of its simplicity and beauty, we include the proof for the spectral norm. The proof for the general case, which goes back to Mirsky, can be found in, e.g., [48, p. 468].

For any matrix $B \in \mathbb{R}^{m \times n}$ of rank at most k , the null space $\text{kernel}(A)$ has dimension at least $n - r$. Hence, the intersection between $\text{kernel}(A)$ and the $(k+1)$ -dimensional space $\text{range}(V_{k+1})$ is nontrivial. Let w be a vector in this intersection with $\|w\|_2 = 1$. Then

$$\begin{aligned} \|A - B\|_2^2 &\geq \|(A - B)w\|_2^2 = \|Aw\|_2^2 = \|AV_{k+1}V_{k+1}^T w\|_2^2 = \|U_{k+1}\Sigma_{k+1}V_{k+1}^T w\|_2^2 \\ &= \sum_{j=1}^{r+1} \sigma_j |v_j^T w|^2 \geq \sigma_{k+1} \sum_{j=1}^{r+1} |v_j^T w|^2 = \sigma_{k+1}. \end{aligned}$$

Together with (9), this shows the result. \square

Remark 2 The singular values are uniquely determined by the matrix A . However, even when ignoring their sign indeterminacy, the singular vectors are only uniquely determined for simple singular values [48, Theorem 2.6.5]. In particular, the subspaces $\text{range}(U_r)$, $\text{range}(V_r)$, and the operator $\mathcal{T}_k(A)$ are uniquely determined if and only if $\sigma_k > \sigma_{k+1}$. However, as we will see in Sect. 2.2 below, there is nothing pathological about the situation $\sigma_k = \sigma_{k+1}$ for the purpose of low-rank approximation. \diamond

A practically important truncation strategy is to choose the rank such that the approximation error is below a certain threshold $\varepsilon > 0$. With slight abuse of

notation, we write

$$\mathcal{T}_\varepsilon(A) := \mathcal{T}_{k_\varepsilon(A)}(A),$$

where $k_\varepsilon(A)$ is the **ε -rank** of A defined as the smallest integer k such that $\|\mathcal{T}_k(A) - A\| \leq \varepsilon$. By entering type rank into MATLAB, one observes that the MATLAB function rank actually computes the ε -rank with respect to the spectral norm and with $\varepsilon = u \max\{m, n\} \|A\|_2$, where $u \approx 2 \times 10^{-16}$ is the unit roundoff.

2.2 Stability of SVD and Low-Rank Approximation

The singular values of a matrix A are perfectly stable under a perturbation $A \mapsto A + E$. Letting $\sigma_j(B)$ denote the j th singular value of a matrix B , Weyl's inequality [48, p. 454] states that

$$\sigma_{i+j-1}(A + E) \leq \sigma_i(A) + \sigma_j(E), \quad 1 \leq i, j \leq \min\{m, n\}, \quad i + j \leq q + 1.$$

Setting $j = 1$, this implies

$$\sigma_i(A + E) \leq \sigma_i(A) + \|E\|_2, \quad i = 1, \dots, \min\{m, n\}. \quad (10)$$

In view of Remark 2, it is probably not surprising that the superb normwise stability of singular values does *not* carry over to singular vectors. For example [63], consider

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 1 + \varepsilon \end{pmatrix}, \quad E = \begin{pmatrix} 0 & \varepsilon \\ \varepsilon & -\varepsilon \end{pmatrix}.$$

Then A has right singular vectors $\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ while $A + E$ has the completely different right singular vectors $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ no matter how small $\varepsilon > 0$ is chosen. Considering the low-rank approximation (8), it is particularly interesting to study the perturbation behavior of the subspace spanned by the first k singular vectors.

Theorem 3 (Wedin [71]) *Let $r < \min\{m, n\}$ and assume that $\delta := \sigma_k(A + E) - \sigma_{k+1}(A) > 0$. Let \mathcal{U}_k and $\widetilde{\mathcal{U}}_k / \mathcal{V}_k$ and $\widetilde{\mathcal{V}}_k$ denote the subspaces spanned by the first k left/right singular vectors of A and $A + E$, respectively. Then*

$$\sqrt{\|\sin \Theta(\mathcal{U}_k, \widetilde{\mathcal{U}}_k)\|_F^2 + \|\sin \Theta(\mathcal{V}_k, \widetilde{\mathcal{V}}_k)\|_F^2} \leq \sqrt{2} \frac{\|E\|_F}{\delta},$$

where Θ is a diagonal matrix containing the canonical angles between two subspaces on its diagonal.

Theorem 3 predicts that the accuracy of the singular subspaces is determined by the perturbation level in A multiplied with $\delta^{-1} \approx (\sigma_k - \sigma_{k+1})^{-1}$. This result appears to be rather discouraging because we will primarily work with matrices for which the singular values decay rather quickly and hence $\sigma_k - \sigma_{k+1}$ is very small.

Fortunately, a simple argument shows that low-rank approximations are rather robust under perturbations.

Lemma 1 *Let $A \in \mathbb{R}^{m \times n}$ have rank at most k . Then*

$$\|\mathcal{T}_k(A + E) - A\| \leq C\|E\|$$

holds with $C = 2$ for any unitarily invariant norm $\|\cdot\|$. For the Frobenius norm, the constant can be improved to $C = (1 + \sqrt{5})/2$.

Proof By Theorem 2, $\|\mathcal{T}_k(A + E) - (A + E)\| \leq \|E\|$. Hence, the triangle inequality implies $\|\mathcal{T}_k(A + E) - (A + E) + (A + E) - A\| \leq 2\|E\|$. The proof of the second part can be found in [43, Theorem 4.5]. \square

For the practically more relevant case that the original matrix A has rank larger than r , Lemma 1 implies

$$\begin{aligned} \|\mathcal{T}_k(A + E) - \mathcal{T}_k(A)\| &= \|\mathcal{T}_k(\mathcal{T}_k(A) + (A - \mathcal{T}_k(A)) + E) - \mathcal{T}_k(A)\| \\ &\leq C\|(A - \mathcal{T}_k(A)) + E\| \leq C(\|A - \mathcal{T}_k(A)\| + \|E\|). \end{aligned} \quad (11)$$

Thus, as long as the perturbation level is not larger than the desired truncation error, a perturbation will at most have a mild effect on our ability to approximate A by a low-rank matrix.

Example 1 Consider a partitioned matrix

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad A_{ij} \in \mathbb{R}^{m_i \times n_j},$$

and a desired rank $k \leq m_i, n_j$. Let $\varepsilon := \|\mathcal{T}_k(A) - A\|$ and $E_{ij} := \mathcal{T}_k(A_{ij}) - A_{ij}$. Because of the min-max characterization of singular values, we have $\|E_{ij}\| \leq \varepsilon$. Setting $\|\cdot\| \equiv \|\cdot\|_F$ and using (11), we obtain

$$\left\| \mathcal{T}_k \begin{pmatrix} \mathcal{T}_k(A_{11}) & \mathcal{T}_k(A_{12}) \\ \mathcal{T}_k(A_{21}) & \mathcal{T}_k(A_{22}) \end{pmatrix} - A \right\|_F = \left\| \mathcal{T}_k \left(A + \begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix} \right) - A \right\|_F \leq C\varepsilon,$$

with $C = \frac{3}{2}(1 + \sqrt{5})$. Consequently, first truncating the matrix blocks and then the entire matrix results in a quasi-optimal approximation with a very reasonable constant. \diamond

2.3 Algorithms for Low-Rank Approximation

The choice of algorithm for computing a good low-rank matrix approximation to a matrix $A \in \mathbb{R}^{m \times n}$ critically depends on the size of the matrix A how its entries can be accessed by the algorithm:

1. If all entries of A are readily available or can be cheaply evaluated and $\min\{m, n\}$ is small, we can simply compute $\mathcal{T}_k(A)$ according to its definition via the SVD. As we will explain below, the SVD can also be useful when $\min\{m, n\}$ is large but A is given in factorized form.
2. For large m, n , the Lanczos-based method and randomized algorithms are typically the methods of choice, provided that matrix-vector multiplications with A and A^T can be performed.
3. When A is not readily available and the computation of its entries is expensive, it becomes mandatory to use an approach that only requires the computation of a selected set of entries, such as adaptive cross approximation.

2.3.1 SVD-Based Algorithms

We refer to [33, Sect. 8.6] and [28] for a description of classical algorithms for computing an (economic) SVD $A = U\Sigma V^T$ of a dense matrix $A \in \mathbb{R}^{m \times n}$. Assuming $m \geq n$, the complexity of these algorithms is $O(mn^2)$. Unlike the algorithms discussed below, none of these classical algorithms is capable of attaining lower complexity when it is known a priori that the target rank k is much smaller than m, n and only the truncated SVD $U_k \Sigma_k V_k^T$ needs to be computed to define $\mathcal{T}_k(A)$.

In practice, one of course never stores $\mathcal{T}_k(A)$ explicitly but always in terms of its factors by storing, for example, the $(m + n)k$ entries of $U_k \Sigma_k$ and V_k . In principle, this can be reduced further to $(m + n)k - k^2 + O(k)$ by expressing U_k, V_k in terms of k Householder reflectors [33, Sect. 5.1.6], but in practice this is rarely done because it becomes too cumbersome to work with such a representation and the benefit for $k \ll m, n$ is marginal at best.

One major advantage of the SVD is that one obtains complete knowledge about the singular values, which conveniently allows for choosing the truncation rank k to attain a certain accuracy. However, care needs to be applied when interpreting small *computed* singular values. Performing a backward stable algorithms in double-precision arithmetic introduces a backward error E with $\|E\|_2 \leq 10^{-16} \times C\sigma_1$ for some constant C mildly growing with m, n . According to (10), one can therefore expect that singular values near or below $10^{-16}\sigma_1$ are completely corrupted by roundoff error. Only in exceptional cases it is possible to compute singular values to high relative accuracy [27]. The stagnation of the singular values observed for the Hilbert matrix in the left plot of Fig. 4 below is typical and entirely due to roundoff error. It is well-known that, in exact arithmetic, these singular values decay exponentially, see [11]; they certainly do not stagnate at 10^{-15} .

The SVD is frequently used for recompression. Suppose that

$$A = BC^T, \quad \text{with } B \in \mathbb{R}^{m \times K}, C \in \mathbb{R}^{n \times K}, \quad (12)$$

where K is larger than the target rank k but still (much) smaller than m, n . For example, such a situation arises when summing up J matrices of rank k :

$$A = \sum_{j=1}^J \underbrace{B_i}_{\in \mathbb{R}^{m \times k}} \underbrace{C_i}_{\in \mathbb{R}^{n \times k}}^T = \underbrace{\left(B_1 \cdots B_J \right)}_{\mathbb{R}^{m \times Jk}} \underbrace{\left(C_1 \cdots C_J \right)}_{\mathbb{R}^{n \times Jk}}^T. \quad (13)$$

In this and similar situations, the matrices B, C in (12) cannot be expected to have orthogonal columns. The following algorithm is then used to recompress A back to rank k :

1. Compute (economic) QR decompositions $B = Q_B R_B$ and $C = Q_C R_C$.
2. Compute truncated SVD $\tilde{\mathcal{T}}_k(R_B R_C^T) = \tilde{U}_k \Sigma_k \tilde{V}_k$.
3. Set $U_k = Q_B \tilde{U}_k$, $V_k = Q_C \tilde{V}_k$ and return $\mathcal{T}_k(A) := U_k \Sigma_k V_k^T$.

This algorithm returns a best rank- k approximation of A and has complexity $O((m+n)K^2)$.

2.3.2 Lanczos-Based Algorithms

In the following, we discuss algorithms that makes use of the following Krylov subspaces for extracting a low-rank approximation to A :

$$\begin{aligned} \mathcal{K}_{K+1}(AA^T, u_1) &= \text{span} \{u_1, AA^T u_1, \dots, (AA^T)^K u_1\}, \\ \mathcal{K}_{K+1}(A^T A, v_1) &= \text{span} \{v_1, A^T A v_1, \dots, (A^T A)^K v_1\}, \end{aligned}$$

with a (random) starting vector $u_1 \in \mathbb{R}^m$ and $v_1 = A^T u_1 / \|A^T u_1\|_2$. The bidiagonal Lanczos process, which is shown in lines 1–5 of Algorithm 1, produces matrices $U_{K+1} \in \mathbb{R}^{m \times (K+1)}$, $V_{K+1} \in \mathbb{R}^{n \times (K+1)}$ such that their columns form orthonormal bases of $\mathcal{K}_{K+1}(AA^T, u_1)$ and $\mathcal{K}_{K+1}(A^T A, v_1)$, respectively. The scalars α_j, β_j generated by the Gram-Schmidt process in Algorithm 1 can be arranged in a bidiagonal matrix

$$B_K = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_K & \alpha_K \end{pmatrix}, \quad (14)$$

leading to the (two-sided) **Lanczos decomposition**

$$A^T U_K = V_K B_K^T, \quad A V_K = U_K B_K + \beta_{K+1} u_{K+1} e_K^T, \quad (15)$$

where e_K denotes the K th unit vector of length K . In view of (8), it might be tempting to approximate $\mathcal{T}_k(A)$ by extracting approximations to the k dominant singular triplets to A from (15); see, e.g., [16]. For this purpose, one could use existing software packages like ARPACK [49], which is behind the MATLAB functions `eigs` and `svds`. Unfortunately, this may turn out to be a rather bad idea, especially if restarting is used. The small gaps between small singular values lead to poor convergence behavior of the Lanczos method for approximating singular vectors, basically because of their ill-conditioning; see Theorem 3. This could mislead the algorithm and trigger unnecessary restarting. Following [62], we propose instead to use the approximation

$$\mathcal{T}_k(A) \approx A_K := U_K \mathcal{T}_k(B_K) V_K^T.$$

This leads to Algorithm 1. Note that, in order to avoid loss orthogonality, one needs to reorthogonalize the vectors \tilde{u} and \tilde{v} in lines 3 and 4 versus the previously computed vectors u_1, \dots, u_j and v_1, \dots, v_j , respectively. The approximation error of

Algorithm 1 Lanczos method for low-rank approximation

Input: Matrix $A \in \mathbb{R}^{m \times n}$, starting vector $u_1 \in \mathbb{R}^m$ with $\|u_1\|_2 = 1$, integer $K \leq \min\{m, n\}$, desired rank $k \leq K$.
Output: Matrices $\tilde{U}_k \in \mathbb{R}^{m \times k}$, $\tilde{V}_k \in \mathbb{R}^{n \times k}$ with orthonormal columns and diagonal matrix $\tilde{\Sigma}_k \in \mathbb{R}^{k \times k}$ such that $A \approx A_K := \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^T$.

- 1: $\tilde{v} \leftarrow A^T u_1$, $\alpha_1 \leftarrow \|\tilde{v}\|_2$, $v_1 \leftarrow \tilde{v}/\alpha_1$.
- 2: **for** $j = 1, \dots, K$ **do**
- 3: $\tilde{u} \leftarrow A v_j - \alpha_j u_j$, $\beta_{j+1} \leftarrow \|\tilde{u}\|_2$, $u_{j+1} \leftarrow \tilde{u}/\beta_{j+1}$.
- 4: $\tilde{v} \leftarrow A^T u_{j+1} - \beta_{j+1} v_j$, $\alpha_{j+1} \leftarrow \|\tilde{v}\|_2$, $v_{j+1} \leftarrow \tilde{v}/\alpha_{j+1}$.
- 5: **end for**
- 6: Set $U_K \leftarrow (u_1, \dots, u_K)$ and $V_K \leftarrow (v_1, \dots, v_K)$.
- 7: Construct bidiagonal matrix B_K according to (14).
- 8: Compute singular value decomposition $B_K = \widehat{U} \widehat{\Sigma} \widehat{V}^T$.
- 9: $\tilde{U}_k \leftarrow U_K \widehat{U}(:, 1:k)$, $\tilde{\Sigma}_k \leftarrow \widehat{\Sigma}(1:k, 1:k)$, $\tilde{V}_k \leftarrow V_K \widehat{V}(:, 1:k)$.

Algorithm 1 can be cheaply monitored, *assuming* that the exact value of $\|A\|_F$ or a highly accurate approximation is known.

Lemma 2 *The approximation A_K returned by Algorithm 1 satisfies*

$$\|A_K - A\|_F \leq \sqrt{\sigma_{k+1}(B_K)^2 + \dots + \sigma_K(B_K)^2} + \omega_K, \quad (16)$$

where $\omega_K^2 = \|A\|_F^2 - \alpha_1^2 \sum_{j=2}^K (\alpha_j^2 + \beta_j^2)$.

Proof By the triangular inequality

$$\begin{aligned} \|A_K - A\|_F &\leq \|U_K(\mathcal{T}_k(B_K) - B_K)V_K^T + U_K B_K V_K^T - A\|_F \\ &\leq \sqrt{\sigma_{k+1}(B_K)^2 + \dots + \sigma_K(B_K)^2} + \|U_K B_K V_K^T - A\|_F. \end{aligned}$$

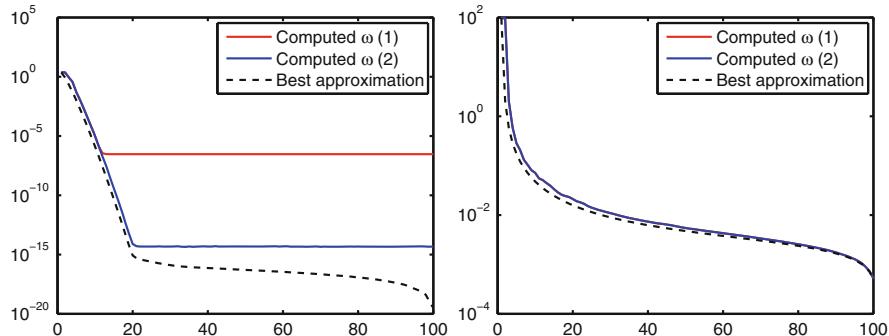


Fig. 2 Behavior of ω_K computed using (1) $\omega_K^2 = \|A\|_F^2 - \alpha_1^2 \sum_{j=2}^K (\alpha_j^2 + \beta_j^2)$ and (2) $\omega_K = \|A - U_K B_K V_K^T\|_F$, compared to the best approximation error $\sqrt{\sigma_{k+1}^2 + \dots + \sigma_n^2}$ for the Hilbert matrix (left figure) and the exponential matrix (right figure)

To bound the second term, we note that $\|A\|_F^2 = \|B_K\|_F^2 + \|U_K B_K V_K^T - A\|_F^2$ holds because of orthogonality. \square

Example 2 We apply Algorithm 1 to obtain the low-rank approximation of two 100×100 matrices with a qualitatively different singular value decay. Since the error bound (16) is dominated by the second term, we only report on the values of ω_K as K increases. We consider two examples with a rather different singular value behavior:

- The Hilbert matrix A defined by $A(i,j) = 1/(i+j-1)$.
- The “exponential” matrix A defined by $A(i,j) = \exp(-\gamma|i-j|/n)$ with $\gamma = 0.1$.

From Fig. 2, it can be seen that ω_K follows the best approximation error quite closely, until it levels off due to round off error. Due to cancellation, this happens much earlier when we compute ω_K using the expression from Lemma 2. Unfortunately, the more accurate formula $\omega_K = \|A - U_K B_K V_K^T\|_F$ is generally too expensive. In practice, one can avoid this by using the much cheaper approximation $\|U_{K+1} B_{K+1} V_{K+1}^T - U_K B_K V_K^T\|_F$, which also does not require knowledge of $\|A\|_F$. \diamond

The excellent behavior of Algorithm 1 observed in Example 2 indicates that not much more than k iterations are needed to attain an approximation error close to $\sqrt{\sigma_{k+1}^2 + \dots + \sigma_{\min\{m,n\}}^2}$, say, within a factor of $\sqrt{2}$. For $k \ll \min\{m, n\}$, it thus pays off to use Algorithm 1 even for a dense matrix A , requiring a complexity of $O(mnk + (m+n)k^2)$.

Example 3 One situation that calls for the use of iterative methods is the recompression of a matrix A that is represented by a longer sum (13). The algorithm based on QR decompositions described in the beginning of this section, see (13), has a complexity that grows quadratically with J , the number of terms in the sum. Assuming that it does not require more than $O(k)$ iterations, Algorithm 1 scales linearly with J simply because the number of matrix-vector multiplications with A and A^T scale linearly with J .

In principle, one could alternatively use successive truncation to attain linear scaling in J . However, this comes at the risk of cancellation significantly distorting the results. For example, we have

$$\mathcal{T}_1 \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 2\varepsilon - 1 & 0 \\ 0 & \varepsilon - 1 \end{pmatrix} \right) = \mathcal{T}_1 \begin{pmatrix} 2\varepsilon & 0 \\ 0 & \varepsilon \end{pmatrix} = \begin{pmatrix} 2\varepsilon & 0 \\ 0 & 0 \end{pmatrix}$$

for any $0 \leq \varepsilon < 1/4$, but successive truncation of this sum leads to

$$\begin{aligned} & \mathcal{T}_1 \left(\mathcal{T}_1 \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \right) + \begin{pmatrix} \varepsilon - 1 & 0 \\ 0 & \varepsilon - 1 \end{pmatrix} \right) \\ &= \mathcal{T}_1 \left(\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 2\varepsilon - 1 & 0 \\ 0 & \varepsilon - 1 \end{pmatrix} \right) = \begin{pmatrix} 0 & 0 \\ 0 & \varepsilon - 1 \end{pmatrix}, \end{aligned}$$

which incurs an error of $O(1)$ instead of $O(\varepsilon)$. \diamond

2.3.3 Randomized Algorithms

Algorithm 1 is based on subsequent matrix-vector multiplications, which perform poorly on a computer because of the need for fetching the whole matrix A from memory for *each* multiplication. One way to address this problem is to use a block Lanczos method [34]. The multiplication of A with a block of k vectors can be easily arranged such that A needs to be read only once from memory. A simpler and (more popular) alternative to the block Lanczos method is to make use of randomized algorithms. The following discussion of randomized algorithms is very brief; we refer to the excellent survey paper by Halko et al. [45] for more details and references.

Algorithm 2 Randomized algorithm for low-rank approximation

Input: Matrix $A \in \mathbb{R}^{m \times n}$, desired rank $k \leq \min\{m, n\}$, oversampling parameter $p \geq 0$.
Output: Matrices $\widetilde{U}_k \in \mathbb{R}^{m \times k}$, $\widetilde{V}_k \in \mathbb{R}^{n \times k}$ with orthonormal columns and diagonal matrix $\widetilde{\Sigma}_k \in \mathbb{R}^{k \times k}$ such that $A \approx \widehat{A} = \widetilde{U}_k \widetilde{\Sigma}_k \widetilde{V}_k^T$.

- 1: Choose standard Gaussian random matrix $\Omega \in \mathbb{R}^{n \times (k+p)}$.
- 2: Perform block matrix vector-multiplication $Y = A\Omega$.
- 3: Compute (economic) QR decomposition $Y = QR$.
- 4: Form $Z = Q^T A$.
- 5: Compute singular value decomposition $Z = \widehat{U} \widehat{\Sigma} \widehat{V}^T$.
- 6: $\widetilde{U}_k \leftarrow Q \widehat{U}(:, 1:k)$, $\widetilde{\Sigma}_k \leftarrow \widehat{\Sigma}(1:k, 1:k)$, $\widetilde{V}_k \leftarrow \widehat{V}(:, 1:k)$.

Algorithm 2 is a simple but effective example for a randomized algorithm. To gain some intuition, let us assume for the moment that A has rank k , that is, it can be factorized as $A = BC^T$ with $B \in \mathbb{R}^{m \times k}$, $C \in \mathbb{R}^{n \times k}$. It then suffices to choose $p = 0$. With probability one, the matrix $C^T \Omega \in \mathbb{R}^{k \times k}$ is invertible, which implies that the matrices $Y = A\Omega = B(C^T \Omega)$ and have the same column space. In turn,

$\widehat{A} = QQ^T A = QQ^T BC^T = BC^T = A$ and, hence, Algorithm 2 almost surely recovers the low-rank factorization of A exactly.

When A does not have rank k (but can be well approximated by a rank- k matrix), it is advisable to choose the oversampling parameter p larger than zero in Algorithm 2, say, $p = 5$ or $p = 10$. The following result shows that the resulting error will not be far away from the best approximation error σ_{k+1} in expectation, with respect to the random matrix Ω .

Theorem 4 ([45, Theorem 1.1]) *Let $k \geq 2$ and $p \geq 2$ be chosen such that $k + p \leq \min\{m, n\}$. Then the rank- k approximation \widehat{A} produced by Algorithm 2 satisfies*

$$\mathbb{E}\|A - \widehat{A}\|_2 \leq \left(2 + \frac{4\sqrt{(k+p)\min\{m,n\}}}{p-1}\right) \sigma_{k+1}. \quad (17)$$

It is shown in [45] that a slightly higher bound than (17) holds with very high probability. The bound (17) improves significantly when performing a few steps of subspace iteration after Step 2 in Algorithm 2, which requires a few additional block matrix-vector multiplications with A^T and A . In practice, this may not be needed. As the following example shows, the observed approximation error is much better than predicted by Theorem 4.

Example 4 We apply Algorithm 2 to the 100×100 matrices from Example 2.

- (a) For the Hilbert matrix A defined by $A(i,j) = 1/(i+j-1)$, we choose $k = 5$ and obtain the following approximation errors as p varies, compared to the exact rank-5 approximation error:

Exact	$p = 0$	$p = 1$	$p = 5$
1.88×10^{-3}	2.82×10^{-3}	1.89×10^{-3}	1.88×10^{-3}

- (b) For the exponential matrix A defined by $A(i,j) = \exp(-\gamma|i-j|/n)$ with $\gamma = 0.1$, we choose $k = 40$ and obtain:

Exact	$p = 0$	$p = 10$	$p = 40$	$p = 80$
1.45×10^{-3}	5×10^{-3}	4×10^{-3}	1.6×10^{-3}	1.45×10^{-3}

These results indicate that small values of p suffice to obtain an approximation not far away from σ_{k+1} . Interestingly, the results are worse for the matrix with slower singular value decay. This is not the case for the Lanczos method (see Fig. 2), for which equally good results are attained for both matrices. \diamond

2.3.4 Adaptive Cross Approximation (ACA)

The aim of *Adaptive Cross Approximation (ACA)* is to construct a low-rank approximation of a matrix A directly from the information provided by well-chosen rows and columns of A . An existence result for such an approach is given in the following theorem. Note that we use MATLAB notation for denoting submatrices.

Theorem 5 ([36]) Let $A \in \mathbb{R}^{m \times n}$. Then there exist row indices $r \subset \{1, \dots, m\}$ and column indices $c \subset \{1, \dots, n\}$ and a matrix $S \in \mathbb{R}^{k \times k}$ such that $|c| = |r| = k$ and

$$\|A - A(:, c)S A(r, :) \|_2 \leq (1 + 2\sqrt{k}(\sqrt{m} + \sqrt{n}))\sigma_{k+1}(A).$$

Theorem 5 shows that it is, in principle, always possible to build a quasi-optimal low-rank approximation from the rows and columns of A .

The construction [36] of the low-rank approximation in Theorem 5 proceeds in two steps:

1. Select c and r by selecting k “good” rows from the singular vector matrices $U_k \in \mathbb{R}^{m \times k}$ and $V_k \in \mathbb{R}^{n \times k}$, respectively.
2. Choose a suitable matrix S .

Step 1 requires to choose c, r with $|c| = |r| = k$ such that $\|U_k(c, :)^{-1}\|_2$ and $\|V_k(r, :)^{-1}\|_2$ are small. This is not practical, primarily because it involves the singular vectors, which we want to avoid computing. Moreover, it is NP hard to choose c and r optimally even when the singular vectors are known [23]. Recently, it was shown [22] that random column and row selections can be expected to result in good approximations, provided that A admits an approximate low-rank factorization with factors satisfying an incoherence condition. However, it is hard to know a priori whether this incoherence condition holds and, in the described scenario of sampling only a few entries of A , it cannot be enforced in a preprocessing step that randomly transforms the columns and rows of A .

Step 2 in the construction of [36] involves the full matrix A , which is not available to us. A simpler alternative is given by $S = (A(r, c))^{-1}$, provided that $A(r, c)$ is invertible. This has a direct consequence: Letting

$$R := A - A(:, c)(A(r, c))^{-1}A(r, :) \quad (18)$$

denote the remainder of the low-rank approximation, we have

$$R(r, :) = 0, \quad R(:, c) = 0.$$

In other words, the rows r and the columns c of A are interpolated exactly, giving rise to the name *cross approximation*, cf. Fig. 3.

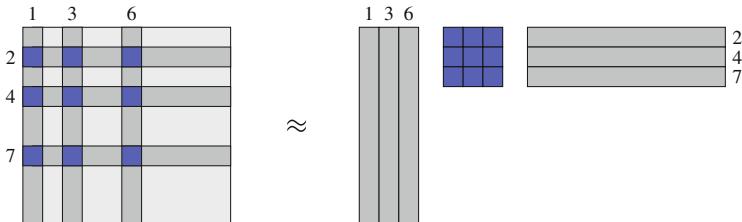


Fig. 3 Low-rank approximation of a matrix by cross approximation

We still need to discuss a cheaper alternative for Step 1. For the approximation (18), the *maximum volume principle* offers a recipe for choosing c and r . To motivate this principle, consider a $(k+1) \times (k+1)$ matrix

$$A = \begin{pmatrix} A_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad A_{11} \in \mathbb{R}^{k \times k}, a_{12} \in \mathbb{R}^{k \times 1}, a_{21} \in \mathbb{R}^{1 \times k}, a_{22} \in \mathbb{R},$$

and suppose that A_{11} is invertible. Then, using the Schur complement,

$$|(A^{-1})_{k+1,k+1}| = \frac{1}{|a_{22} - a_{12}A_{11}^{-1}a_{21}|} = \frac{|\det A|}{|\det A_{11}|}$$

If $|\det A_{11}|$ is maximal (that is, A_{11} has maximal volume) among all possible selections of $k \times k$ submatrices of A , this shows that $|(A^{-1})_{k+1,k+1}|$ is maximal among all entries of A^{-1} :

$$|(A^{-1})_{k+1,k+1}| = \|A^{-1}\|_C := \max_{i,j} |(A^{-1})_{ij}|.$$

On the other hand, using [47, Sect. 6.2], we have

$$\begin{aligned} \sigma_{k+1}(A)^{-1} &= \|A^{-1}\|_2 = \max_x \frac{\|A^{-1}x\|_2}{\|x\|_2} \\ &\geq \frac{1}{k+1} \max_x \frac{\|A^{-1}x\|_\infty}{\|x\|_1} = \frac{1}{k+1} \|A^{-1}\|_C. \end{aligned}$$

When performing cross approximation with the first k rows and columns, the remainder defined in (18) thus satisfies

$$\|R\|_C = |a_{22} - a_{12}A_{11}^{-1}a_{21}| = \frac{1}{\|A^{-1}\|_C} \leq (k+1)\sigma_{k+1}(A).$$

By bordering A_{11} with other rows and columns of A , this result can be extended to $m \times n$ matrices.

Theorem 6 ([35]) *Let $A \in \mathbb{R}^{m \times n}$ and consider a partitioning*

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

such that $A_{11} \in \mathbb{R}^{k \times k}$ is non-singular and has maximal volume among all $k \times k$ submatrices of A . Then

$$\|A_{22} - A_{21}A_{11}^{-1}A_{12}\|_C \leq (k+1)\sigma_{k+1}(A).$$

The quasi-optimality shown in Theorem 6 is good enough for all practical purposes. It requires to choose $S = (A(r, c))^{-1}$ such that the “volume” $|\det A(r, c)|$

is as large as possible. Unfortunately, it turns out, once again, that the determination of a submatrix $A(r, c)$ of maximal volume is NP-hard [20] and thus computationally infeasible.

Greedy approaches try to incrementally construct the row and column index sets r, c by maximizing the volume of the submatrix $A(r, c)$ in each step [5, 9, 64]. A basic algorithm for a greedy strategy is given in Algorithm 3.

Algorithm 3 ACA with full pivoting

```

1: Set  $R_0 := A$ ,  $r := \{\}$ ,  $c := \{\}$ ,  $k := 0$ 
2: repeat
3:    $k := k + 1$ 
4:    $(i^*, j^*) := \arg \max_{i,j} |R_{k-1}(i, j)|$ 
5:    $r := r \cup \{i^*\}$ ,  $c := c \cup \{j^*\}$ 
6:    $\delta_k := R_{k-1}(i^*, j^*)$ 
7:    $u_k := R_{k-1}(:, j^*)$ ,  $v_k := R_{k-1}(i^*, :)^T / \delta_k$ 
8:    $R_k := R_{k-1} - u_k v_k^T$ 
9: until  $\|R_k\|_F \leq \varepsilon \|A\|_F$ 
```

If we had $\text{rank}(A) = k$, Algorithm 3 would stop after exactly k steps with $\|R_k\|_F = 0$ and hence A would be fully recovered.

It turns out that the submatrix $A(r, c)$ constructed in Algorithm 3 has a straightforward characterization.

Lemma 3 ([5]) *Let $r_k = \{i_1, \dots, i_k\}$ and $c_k = \{j_1, \dots, j_k\}$ be the row and column index sets constructed in step k of Algorithm 3. Then*

$$\det(A(r_k, c_k)) = R_0(i_1, j_1) \cdots R_{k-1}(i_k, j_k).$$

Proof From lines 7 and 8 in Algorithm 3, we can see that the last column of $A(r_k, c_k)$ is a linear combination of the columns of the matrix

$$\widetilde{A}_k := [A(r_k, c_{k-1}), R_{k-1}(r_k, j_k)] \in \mathbb{R}^{k \times k},$$

which implies $\det(\widetilde{A}_k) = \det(A(r_k, c_k))$. However, $n_k(i, j_k) = 0$ for all $i = i_1, \dots, i_{k-1}$ and hence

$$\det(\widetilde{A}_k) = R_{k-1}(i_k, j_k) \det(A(r_{k-1}, c_{k-1})).$$

Noting that $\det A(r_1, c_1) = A(i_1, j_1) = R_0(i_1, j_1)$ thus proves the claim by induction. \square

In line 4 of Algorithm 3, a maximization step over all entries of the remainder is performed. Clearly, this is only possible if all matrix entries are cheaply available. In this case, it is reasonable to assume that the norm of A can be used in the stopping criterion in line 9. For larger matrices A with a possibly expensive access to the entries $A(i, j)$, the choice of the pivots and the stopping criterion need to be adapted.

The inspection of all matrix entries in the pivot search can be avoided by a partial pivoting strategy as described in Algorithm 4. In line 3 of Algorithm 4, the new pivot index is found by fixing a particular row index and optimizing the column index only. Note also that there is no need to form the remainder R_k explicitly. Any entry of R_k can always be computed from

$$R_k(i,j) = A(i,j) - \sum_{\ell=1}^k u_\ell(i)v_\ell(j).$$

This means that in each step of Algorithm 4 only a particular row and column of the matrix A needs to be accessed. The overall number of accessed entries is hence bounded by $k(m + n)$.

Algorithm 4 ACA with partial pivoting

```

1: Set  $R_0 := A$ ,  $r := \{\}$ ,  $c := \{\}$ ,  $k := 1$ ,  $i^* := 1$ 
2: repeat
3:    $j^* := \arg \max_j |R_{k-1}(i^*, j)|$                                 % choose column index in given row
4:    $\delta_k := R_{k-1}(i^*, j^*)$ 
5:   if  $\delta_k = 0$  then
6:     if  $\#r = \min\{m, n\} - 1$  then
7:       Stop                                         % the matrix has been recovered
8:     end if
9:   else
10:     $u_k := R_{k-1}(:, j^*)$ ,  $v_k := R_{k-1}(i^*, :)^T / \delta_k$           % update low-rank approximation
11:     $R_k := R_{k-1} - u_k v_k^T$ 
12:     $k := k + 1$ 
13:  end if
14:   $r := r \cup \{i^*\}$ ,  $c := c \cup \{j^*\}$ 
15:   $i^* := \arg \max_{i, i \notin r} u_k(i)$                                 % choose new row index
16: until stopping criterion is satisfied

```

If the norm of A is not available, also the stopping criterion from line 9 in Algorithm 3 needs to be adjusted. A possible alternative is to replace $\|A\|_F$ by $\|A_k\|_F$ from the already computed low-rank approximation $A_k := \sum_{j=1}^k u_j v_j^T$:

$$\|A_k\|_F^2 = \|A_{k-1}\|_F^2 + \sum_{j=1}^{k-1} u_k^T u_j v_j^T v_k + \|u_k\|_2^2 \|v_k\|_2^2,$$

see also [9]. Replacing the residual norm $\|R_k\|_F$ by $\|u_k\|_2 \|v_k\|_2$, Algorithm 4 is stopped if

$$\|u_k\|_2 \|v_k\|_2 \leq \varepsilon \|A_k\|_F.$$

The construction of a low-rank approximation of A from selected knowledge however comes with some cost. Since Algorithm 4 takes only partial information

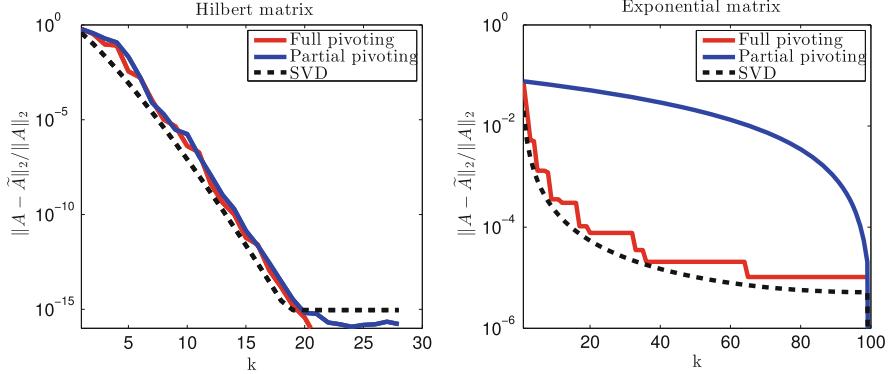


Fig. 4 Approximation of matrices from Example 5 by ACA

of the matrix A into account, it is known to fail to produce reliable low-rank approximations in certain cases, see [18] for a discussion.

Example 5 We apply ACA with full and partial pivoting to the two matrices from Example 2, and compare it to the optimal approximation obtained from the SVD.

- (a) For the Hilbert matrix, it can be seen from Fig. 4 (left) that both ACA strategies yield quasi-optimal low-rank approximations. Recall that the singular values for this example decay very fast.
- (b) For the other matrix, Fig. 4 (right) shows that ACA with full pivoting still yields a quasi-optimal low-rank approximation. In contrast, the partial pivoting strategy suffers a lot from the slow decay of the singular values.

The observed relation between quasi-optimality and decay of the best approximation error has been studied in a related context in [52]. \diamond

The cross approximation technique has a direct correspondence to the process of Gaussian elimination [5]. After a suitable reordering of the rows and columns of A , we can assume that $r = \{1, \dots, k\}$, $c = \{1, \dots, k\}$. If we denote the k th unit vector by e_k , we can write line 8 in Algorithm 3 as

$$R_k = R_{k-1} - \delta_k R_{k-1} e_k e_k^T R_{k-1} = (I - \delta_k R_{k-1} e_k e_k^T) R_{k-1} = L_k R_{k-1},$$

where $L_k \in \mathbb{R}^{m \times m}$ is given by

$$L_k = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & 0 & \\ & & & & \ell_{k+1,k} \\ & & & & \\ & & & & \vdots & \ddots \\ & & & & & 1 \end{bmatrix}, \quad \ell_{i,k} = -\frac{e_i^T R_{k-1} e_k}{e_k^T R_{k-1} e_k}, \quad i = k+1, \dots, m.$$

The matrix L_k differs only in position (k, k) from the usual lower triangular factor in Gaussian elimination. This means that the cross approximation technique can be considered as a column-pivoted LU decomposition for the approximation of a matrix.

The relations between ACA and the Empirical Interpolation Method, which plays an important role in reduced order modeling, are discussed in [10].

If the matrix A happens to be symmetric positive semi-definite, the entry of largest magnitude resides on the diagonal of A . Moreover, one can show that all remainders R_k in Algorithm 3 also remain symmetric positive semi-definite. This means that we can restrict the full pivot search in line 4 to the diagonal without sacrificing the reliability of the low-rank approximation. The resulting algorithm can be interpreted as a pivoted Cholesky decomposition [46, 51] with a computational cost of $O(nk^2)$ instead of $O(n^2k)$ for the general ACA algorithm with full pivoting. See also [47] for an analysis.

2.4 A Priori Approximation Results

None of the techniques discussed in Sect. 2.3 will produce satisfactory approximations if A cannot be well approximated by a low rank matrix in the first place or, in other words, if the singular values of A do not decay sufficiently fast. It is therefore essential to gain some a priori understanding on properties that promote low-rank approximability.

As a rule of thumb, smoothness generally helps. This section illustrates this principle for the simplest setting of matrices associated with bivariate functions; we refer to [17, 42, 57] for detailed technical expositions.

2.4.1 Separability and Low Rank

Let us evaluate a bivariate function

$$f(x, y) : [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \rightarrow \mathbb{R},$$

on a tensor grid $[x_1, \dots, x_n] \times [y_1, \dots, y_m]$ with $x_{\min} \leq x_1 < \dots < x_n \leq x_{\max}$, $y_{\min} \leq y_1 < \dots < y_n \leq y_{\max}$. The obtained function values can be arranged into the $m \times n$ matrix

$$F = \begin{bmatrix} f(x_1, y_1) & f(x_1, y_2) & \cdots & f(x_1, y_n) \\ f(x_2, y_1) & f(x_2, y_2) & \cdots & f(x_2, y_n) \\ \vdots & \vdots & & \vdots \\ f(x_m, y_1) & f(x_m, y_2) & \cdots & f(x_m, y_n) \end{bmatrix}. \quad (19)$$

A basic but crucial observation: When f is separable, that is, it can be written in the form $f(x, y) = g(x)h(y)$ for some functions $g : [x_{\min}, x_{\max}] \rightarrow \mathbb{R}$, $f : [y_{\min}, y_{\max}] \rightarrow \mathbb{R}$ then

$$F = \begin{bmatrix} g(x_1)h(y_1) & \cdots & g(x_1)h(y_n) \\ \vdots & & \vdots \\ g(x_m)h(y_1) & \cdots & g(x_m)h(y_n) \end{bmatrix} = \begin{bmatrix} g(x_1) \\ \vdots \\ g(x_m) \end{bmatrix} \begin{bmatrix} h(y_1) & \cdots & h(y_n) \end{bmatrix}.$$

In other words, separability implies rank at most 1.

More generally, let us suppose that f can be well approximated by a sum of separable functions:

$$f(x, y) = f_k(x, y) + \epsilon_k(x, y), \quad (20)$$

where

$$f_k(x, y) = g_1(x)h_1(y) + \cdots + g_k(x)h_k(y), \quad (21)$$

and the error $|\epsilon_k(x, y)|$ decays quickly as k increases.

Remark 3 Any function in the seemingly more general form $\sum_{i=1}^k \sum_{j=1}^k s_{ij}g_i(x)\tilde{h}_j(y)$ can be written in the form (21), by simply setting $h_i(y) := \sum_{j=1}^k s_{ij}\tilde{h}_j(y)$. \diamond

By (21), the matrix

$$F_k := \begin{bmatrix} f_k(x_1, y_1) & \cdots & f_k(x_1, y_n) \\ \vdots & & \vdots \\ f_k(x_m, y_1) & \cdots & f_k(x_m, y_n) \end{bmatrix}.$$

has rank at most k and, hence,

$$\sigma_{k+1}(F) \leq \|F - F_k\|_2 \leq \|F - F_k\|_F \leq \left(\sum_{i,j=1}^{m,n} \epsilon_k(x_i, y_j)^2 \right)^{1/2} \leq \sqrt{mn} \|\epsilon_k\|_\infty, \quad (22)$$

where $\|\epsilon_k\|_\infty := \max_{\substack{x \in [x_{\min}, x_{\max}] \\ y \in [y_{\min}, y_{\max}]}} |f(x, y) - f_k(x, y)|$. In other words, a good approximation by a short sum of separable functions implies a good low-rank approximation.

2.4.2 Low Rank Approximation via Semi-separable Approximation

There are various approaches to obtain a semi-separable approximation (20).

Exponential Sums

In many cases of practical interest, the function f takes the form $f(x, y) = \tilde{f}(z)$ with $z = \tilde{g}(x) + \tilde{h}(y)$ for certain functions \tilde{g}, \tilde{h} . Notably, this is the case for $f(x, y) = 1/(x - y)$. Any exponential sum approximation $\tilde{f}(z) \approx \sum_{i=1}^k \omega_i \exp(\alpha_i z)$ yields a semi-separable approximation for $f(x, y)$:

$$f(x, y) \approx \sum_{i=1}^k \omega_i \exp(\alpha_i \tilde{g}(x)) \exp(\alpha_i \tilde{h}(y)),$$

For specific functions, such as $1/z$ and $1/\sqrt{z}$, (quasi-)best exponential sum approximations are well understood; see [41]. Sinc quadrature is a general approach to obtain (not necessarily optimal) exponential sum approximations [42, Sect. D.5].

Taylor Expansion

The truncated Taylor expansion of $f(x, y)$ in x around x_0 ,

$$f_k(x, y) := \sum_{i=0}^{k-1} \frac{1}{i!} \frac{\partial^i f(x_0, y)}{\partial x^i} (x - x_0)^i, \quad (23)$$

immediately gives a semi-separable function. One can of course also use the Taylor expansion in y for the same purpose. The approximation error is governed by the remainder of the Taylor series.

To illustrate this construction, let us apply it to the following function, which features prominently in kernel functions of one-dimensional integral operators:

$$f(x, y) := \begin{cases} \log(x - y) & \text{if } x > y, \\ \log(y - x) & \text{if } y > x, \\ 0 & \text{otherwise.} \end{cases}$$

Assuming $x_{\max} > x_{\min} > y_{\max} > y_{\min}$ and setting $x_0 = (x_{\min} + x_{\max})/2$, the Lagrange representation of the error for the truncated Taylor series (23) yields

$$|f(x, y) - f_k(x, y)| \leq \max_{\xi \in [x_{\min}, x_{\max}]} \left| \frac{1}{k!} \frac{\partial^k f(x_0, y)}{\partial x^k} (\xi - x_0)^k \right| \leq \frac{1}{k} \left(\frac{x_{\max} - x_{\min}}{2(x_{\min} - y)} \right)^k. \quad (24)$$

For general sets $\Omega_x, \Omega_y \in \mathbb{R}^d$, let us define

$$\begin{aligned} \text{diam}(\Omega_x) &:= \max\{\|x - y\|_2 : x, y \in \Omega_x\}, \\ \text{dist}(\Omega_x, \Omega_y) &:= \min\{\|x - y\|_2 : x \in \Omega_x, y \in \Omega_y\}. \end{aligned}$$

For our particular case, we set $\Omega_x = [x_{\min}, x_{\max}]$ and $\Omega_y = [y_{\min}, y_{\max}]$ and assume that

$$\min \{\text{diam}(\Omega_x), \text{diam}(\Omega_y)\} \leq \eta \cdot \text{dist}(\Omega_x, \Omega_y) \quad (25)$$

is satisfied for some $0 < \eta < 2$. Such a condition is called *admissibility condition* and the particular form of (25) is typical for functions with a diagonal singularity at $x = y$. We use Taylor expansion in x if $\text{diam}(\Omega_x) \leq \text{diam}(\Omega_y)$ and Taylor expansion in y otherwise. The combination of inequalities (24) and (25) yields

$$\|f(x, y) - f_k(x, y)\|_\infty \leq \frac{1}{k} \left(\frac{\eta}{2}\right)^k.$$

As a consequence of (22), the singular values of the matrix F decay exponentially, provided that (25) holds. By a refined error analysis of the Taylor remainder, the decay factor can be improved to $\frac{\eta}{2+\eta}$ [17, 42], which also removes the restriction $\eta < 2$.

Polynomial Interpolation

Instead of Taylor expansion, Lagrange interpolation in any of the two variables x, y can be used to obtain a semi-separable approximation. This sometimes yields tighter bounds than truncated Taylor expansion, but the error analysis is often harder.

For later purposes, it is interesting to observe the result of performing Lagrange interpolation in *both* variables x, y . Given $f(x, y)$ with $x \in [x_{\min}, x_{\max}]$ and $y \in [y_{\min}, y_{\max}]$, we first perform interpolation in y . For arbitrary interpolation nodes $y_{\min} \leq \theta_1 < \dots < \theta_k \leq y_{\max}$, the Lagrange polynomials $L_{\theta,1}, \dots, L_{\theta,k}$ of degree $k-1$ are defined as

$$L_{\theta,j}(y) := \prod_{\substack{1 \leq i \leq k \\ i \neq j}} \frac{y - \theta_i}{\theta_j - \theta_i}, \quad 1 \leq j \leq k.$$

The corresponding Lagrange interpolation of f in y is given by

$$\mathcal{I}_y[f](x, y) := \sum_{j=1}^k f(x, \theta_j) L_{\theta,j}(y).$$

Analogously, we perform interpolation in x for each $f(x, \theta_j)$ on arbitrary interpolation nodes $x_{\min} \leq \xi_1 < \dots < \xi_k \leq x_{\max}$. Using the linearity of the interpolation operator, this yields

$$f_k(x, y) := \mathcal{I}_x[\mathcal{I}_y[f]](x, y) = \sum_{i,j=1}^k f(\xi_i, \theta_j) L_{\xi,i}(x) L_{\theta,j}(y).$$

By Remark 3, this function is semi-separable. Moreover, in contrast to the result obtained with expanding or interpolating in one variable only, f_k is a bivariate polynomial.

The approximation error can be bounded by

$$\begin{aligned}\|\epsilon_k\|_\infty &\leq \|f - \mathcal{I}_x[\mathcal{I}_y[f]]\|_\infty = \|f - \mathcal{I}_x[f] + \mathcal{I}_x[f] - \mathcal{I}_x[\mathcal{I}_y[f]]\|_\infty \\ &\leq \|f - \mathcal{I}_x[f]\|_\infty + \|\mathcal{I}_x\|_\infty \|f - \mathcal{I}_y[f]\|_\infty,\end{aligned}$$

where the so called Lebesgue constant $\|\mathcal{I}_x\|_\infty$ satisfies $\|\mathcal{I}_x\|_\infty \sim \log r$ when using Chebyshev interpolation nodes. In this case, the one-dimensional interpolation errors $\|f - \mathcal{I}_x[f]\|_\infty$ and $\|f - \mathcal{I}_y[f]\|_\infty$ decay exponentially for an analytic function; see [17, 42]. Because the interpolation is performed in *both* variables, the admissibility condition (25) needs to be replaced by

$$\max \{\text{diam}(\Omega_x), \text{diam}(\Omega_y)\} \leq \eta \cdot \text{dist}(\Omega_x, \Omega_y).$$

Remarks on Smoothness

The techniques presented above are adequate if f is arbitrarily smooth. It is, however, easy to see that the discontinuous function $f : [0, 1] \times [0, 1] \rightarrow \mathbb{R}$ defined by

$$f(x, y) := \begin{cases} 0 & \text{if } 1/4 < x < 3/4, 1/4 < y < 3/4, \\ 1 & \text{otherwise,} \end{cases}$$

yields a matrix F of rank exactly 2. More generally, low-rank approximability is not severely affected by singularities aligned with the coordinate axes. For the case that f only admits finitely many mixed derivatives, see [59] and the references therein.

3 Partitioned Low-Rank Structures

We now proceed to the situation that not the matrix A itself but only certain blocks of A admit good low-rank approximations. The selection of such blocks is usually driven by an admissibility criterion, see (25) for an example.

3.1 HODLR Matrices

Hierarchically Off-Diagonal Low Rank (HODLR) matrices are block matrices with a multi-level structure such that off-diagonal blocks are low-rank. Let $A \in \mathbb{R}^{n \times n}$ and assume for simplicity that $n = 2^p n_p$ for some maximal level $p \in \mathbb{N}$ with $n_p \geq 1$. Moreover, we denote by $I = \{1, \dots, n\}$ the row (and column) index set of A .

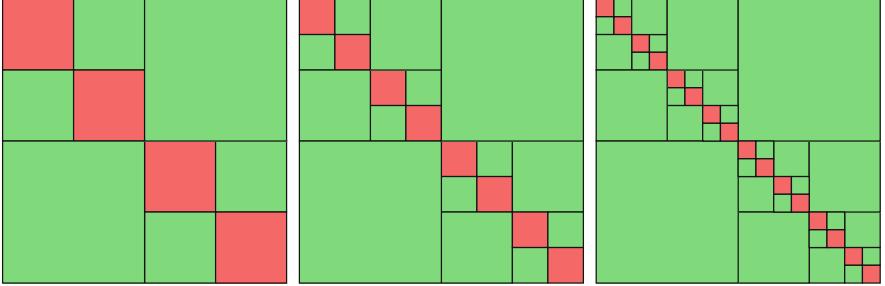


Fig. 5 HODLR matrices with level $p = 2, 3, 4$

As a starting point, we first subdivide the matrix A into blocks. One level $\ell = 0$, we partition the index set $I_1^0 := I$ into $I_1^0 = I_1^1 \cup I_2^1$ with $I_1^1 = \{1, \dots, \frac{n}{2}\}$ and $I_2^1 = \{\frac{n}{2} + 1, \dots, n\}$. This corresponds to a subdivision of A into four subblocks $A(I_i^1, I_j^1)$ with $i, j = 1, 2$. On each level $\ell = 1, \dots, p - 1$, we further partition all index sets I_i^ℓ into equally sized sets $I_{2i-1}^{\ell+1}$ and $I_{2i}^{\ell+1}$. In accordance to this partition, the diagonal blocks $A(I_i^\ell, I_j^\ell)$, $i = 1, \dots, 2^\ell$, are then subdivided into quadratic blocks of size $2^{p-\ell}n_p$. This results in a block structure for A as shown for $p = 2, 3, 4$ in Fig. 5.

Each off-diagonal block $A(I_i^\ell, I_j^\ell)$, $i \neq j$, is assumed to have rank at most k and is kept in the low-rank representation

$$A(I_i^\ell, I_j^\ell) = U_i^{(\ell)}(V_j^{(\ell)})^T, \quad U_i^{(\ell)}, V_j^{(\ell)} \in \mathbb{R}^{n_\ell \times k}, \quad (26)$$

where $n_\ell := \#I_i^\ell = \#I_j^\ell = 2^{p-\ell}n_p$. All diagonal blocks of A are represented on the finest level $\ell = p$ as dense $n_p \times n_p$ matrices.

From Fig. 5, we can easily see that there are 2^ℓ off-diagonal blocks on level $\ell > 0$. If we sum up the storage requirements for the off-diagonal blocks A on the different levels, we get

$$2k \sum_{\ell=1}^p 2^\ell n_\ell = 2kn_p \sum_{\ell=1}^p 2^\ell 2^{p-\ell} = 2kn_p 2^p = 2knp = 2kn \log_2(n/n_p).$$

The diagonal blocks of A on the finest level require $2^p n_p^2 = nn_p$ units of storage. Assuming that $n_p \leq k$, the storage complexity for the HODLR format is hence given by $O(kn \log n)$.

3.1.1 Matrix-Vector Multiplication

A matrix vector multiplication $y = Ax$ for a HODLR matrix A and a vector $x \in \mathbb{R}^n$ can be carried out *exactly* (up to numerical accuracy) by a block recursion. On level

$\ell = 1$, we compute

$$\begin{aligned} y(I_1^1) &= A(I_1^1, I_1^1)x(I_1^1) + A(I_1^1, I_2^1)x(I_2^1), \\ y(I_2^1) &= A(I_2^1, I_1^1)x(I_1^1) + A(I_2^1, I_2^1)x(I_2^1). \end{aligned}$$

In the off-diagonal blocks $A(I_1^1, I_2^1)$ and $A(I_2^1, I_1^1)$, we need to multiply a low-rank matrix of size $n/2$ with a vector of size $n/2$ with cost $c_{LR \cdot v}(n/2)$ where

$$c_{LR \cdot v}(n) = 4nk.$$

In the diagonal blocks, the cost for the matrix-vector multiplication $c_{H \cdot v}$ is the same as for the original matrix with the matrix size reduced to $n/2$. If we add the cost n for the vector addition, we get the recursive relation

$$c_{H \cdot v}(n) = 2c_{H \cdot v}(n/2) + 4kn + n.$$

from which we conclude, using the Master theorem, that

$$c_{H \cdot v}(n) = (4k + 1) \log_2(n)n. \quad (27)$$

3.1.2 Matrix Addition

Let us now consider the computation of $C = A + B$ for HODLR matrices A, B, C . In principle, the addition increases the rank in all off-diagonal blocks of C up to $2k$, cf. (13). We apply truncation to ensure that all off-diagonals have again rank at most k , and thus the result of the addition will in general not be exact. The truncated addition is performed blockwise by

$$C(I_i^\ell, I_j^\ell) := \mathcal{T}_k(A(I_i^\ell, I_j^\ell) + B(I_i^\ell, I_j^\ell))$$

for an off-diagonal block $I_i^\ell \times I_j^\ell$ with the truncation operator \mathcal{T}_k from (8). The cost for two rank- k matrices of size n is

$$c_{LR+LR}(n) = c_{SVD}(nk^2 + k^3),$$

where the generic constant c_{SVD} depends on the cost for the QR factorization and the SVD. As there are 2^ℓ off-diagonal blocks on each level ℓ , the cost for truncated addition sums up to

$$\sum_{\ell=1}^p 2^\ell c_{LR+LR}(n_\ell) = c_{SVD} \sum_{\ell=1}^p 2^\ell (k^3 + n_\ell k^2)$$

$$\begin{aligned} &\leq c_{SVD}(2^{p+1}k^3 + \sum_{\ell=1}^p 2^\ell 2^{p-\ell} n_p k^2) \\ &\leq c_{SVD}(2nk^3 + n \log_2(n)k^2). \end{aligned}$$

The dense diagonal blocks can be added exactly by $2^p n_p^2 = nn_p$ operations. Hence, the overall complexity for the truncated addition of two HODLR matrices is given by

$$c_{H+H}(n) = c_{SVD}(nk^3 + n \log(n)k^2).$$

3.1.3 Matrix Multiplication

As for the addition, the matrix-matrix multiplication $C = AB$ of two HODLR matrices A and B is combined with truncation to ensure that all off-diagonal blocks of C have rank at most k . Again, this implies that the result of multiplication will in general not be exact.

Let $A_{i,j}^{(\ell)} = A(I_i^\ell, I_j^\ell)$, $B_{i,j}^{(\ell)} = B(I_i^\ell, I_j^\ell)$, and consider the blockwise multiplication

$$AB = \begin{bmatrix} A_{1,1}^{(1)} & A_{1,2}^{(1)} \\ A_{2,1}^{(1)} & A_{2,2}^{(1)} \end{bmatrix} \begin{bmatrix} B_{1,1}^{(1)} & B_{1,2}^{(1)} \\ B_{2,1}^{(1)} & B_{2,2}^{(1)} \end{bmatrix} = \begin{bmatrix} A_{1,1}^{(1)}B_{1,1}^{(1)} + A_{1,2}^{(1)}B_{2,1}^{(1)} & A_{1,1}^{(1)}B_{1,2}^{(1)} + A_{1,2}^{(1)}B_{2,2}^{(1)} \\ A_{2,1}^{(1)}B_{1,1}^{(1)} + A_{2,2}^{(1)}B_{2,1}^{(1)} & A_{2,1}^{(1)}B_{1,2}^{(1)} + A_{2,2}^{(1)}B_{2,2}^{(1)} \end{bmatrix}. \quad (28)$$

The block structure of (28) can be illustrated as

$$\begin{array}{c} \text{[Diagram showing a 2x2 grid of blocks where each block is a 2x2 matrix with red corners and green sides, multiplied by a 2x2 grid of low-rank blocks (green with red corners). The result is a 2x2 grid of terms involving the multiplication of these blocks.]}\end{array} = \begin{bmatrix} \text{[Diagram showing the first term of the multiplication: a 2x2 matrix with red corners and green sides multiplied by a 2x2 matrix with red corners and green sides, resulting in a 2x2 matrix with red corners and green sides plus a 2x2 matrix with green sides and red corners.]} & \text{[Diagram showing the second term of the multiplication: a 2x2 matrix with red corners and green sides multiplied by a 2x2 matrix with green sides and red corners, resulting in a 2x2 matrix with red corners and green sides plus a 2x2 matrix with green sides and red corners.]} \\ \text{[Diagram showing the third term of the multiplication: a 2x2 matrix with green sides and red corners multiplied by a 2x2 matrix with red corners and green sides, resulting in a 2x2 matrix with red corners and green sides plus a 2x2 matrix with green sides and red corners.]} & \text{[Diagram showing the fourth term of the multiplication: a 2x2 matrix with green sides and red corners multiplied by a 2x2 matrix with green sides and red corners, resulting in a 2x2 matrix with red corners and green sides plus a 2x2 matrix with green sides and red corners.]} \end{bmatrix}$$

where is again a HODLR matrix of size $n/2$ and is a low-rank block. There are structurally four different types of multiplications in (28):

1. · : the multiplication of two HODLR matrices of size $n/2$,
2. · : the multiplication of two low-rank blocks,
3. · : the multiplication of a HODLR matrix with a low-rank block,
4. · : the multiplication of a low-rank block with a HODLR matrix.

First note that in the cases 2, 3, 4 the multiplication is exact and does not require a truncation step. For case 1, we need to recursively apply low-rank truncation. Let $c_{H \cdot H}$, $c_{LR \cdot LR}$, $c_{H \cdot LR}$, $c_{LR \cdot H}$ denote the cost for each of the four cases, respectively. Moreover, let $c_{H+LR}(n)$ be the cost for the truncated addition of a HODLR matrix

with a low-rank matrix of size n . From (28) we conclude that

$$\begin{aligned} c_{H \cdot H}(n) &= 2(c_{H \cdot H}(n/2) + c_{LR \cdot LR}(n/2) + c_{H \cdot LR}(n/2) + c_{LR \cdot H}(n/2) \\ &\quad + c_{H+LR}(n/2) + c_{LR+LR}(n/2)). \end{aligned}$$

If the result is kept in low-rank form, the cost for the multiplication of two low-rank blocks is

$$c_{LR \cdot LR}(n) = 4nk^2.$$

The multiplication of a HODLR matrix of size n with a rank-one matrix requires $c_{H \cdot v}(n)$ operations. According to (27), we thus get

$$c_{H \cdot LR}(n) = c_{LR \cdot H}(n) = kc_{H \cdot v}(n) = k(4k + 1) \log_2(n)n.$$

The cost for the truncated addition of a HODLR matrix with a low-rank matrix is the same as the cost for adding two HODLR matrices, that is,

$$c_{H+LR}(n) = c_{H+H}(n) = c_{SVD}(nk^3 + n \log(n)k^2).$$

Summing up all terms, we finally obtain

$$c_{H \cdot H}(n) \in O(k^3 n \log n + k^2 n \log^2 n).$$

3.1.4 Matrix Factorization and Linear Systems

A linear system $Ax = b$ can be addressed by first computing the LU factorization $A = LU$ where L is lower triangular with unit diagonal and U is upper triangular. The solution x is then obtained by first solving $Ly = b$ forward substitution and then $Ux = y$ by backward substitution. For a dense $n \times n$ matrix A , the LU factorization requires $O(n^3)$ and constitutes the most expensive step.

In the following, we discuss how an inexact LU factorization $A \approx LU$ can be cheaply computed for a HODLR matrix A , such that L, U are again HODLR matrices:

Such an inexact factorization can be used as a preconditioner in an iterative method for solving $Ax = b$.

Let us first discuss the solution of $Ly = b$ where L is a lower triangular HODLR matrix. On level $\ell = 1$, L , y and b are partitioned as

$$L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

with a low-rank matrix L_{21} and HODLR matrices L_{11}, L_{22} . The corresponding block forward substitution consists of first solving

$$L_{11}y_1 = b_1, \tag{29}$$

then we computing the vector

$$\tilde{b}_2 := b_2 - L_{21}y_1 \tag{30}$$

and finally solving

$$L_{22}y_2 = \tilde{b}_2. \tag{31}$$

The computation of \tilde{b}_2 in (30) requires a matrix-vector product with a low-rank matrix and a vector addition with total cost $(2k + 1)n$. The solutions in (29) and (31) are performed recursively, by forward substitution with the matrix size reduced to $n/2$. Hence, the cost for forward substitution satisfies

$$c_{\text{forw}}(n) = 2c_{\text{forw}}(n/2) + (2k + 1)n.$$

On level $\ell = p$, each of the $2^p = n/n_p$ linear systems of size n_p is solved directly, requiring $O(n_p^3)$ operations. The overall cost for forward substitution is thus given by

$$c_{\text{forw}}(n) \in O(nn_p^2 + kn \log(n)).$$

An analogous estimate holds for the backward substitution $c_{\text{backw}}(n) = c_{\text{forw}}(n)$. Due to the properties of the matrix-vector for HODLR matrices, also the forward and backward algorithms can be carried out exactly (up to numerical accuracy).

Both algorithms can be extended to the matrix level. Assume we want to solve $LY = B$ where

$$L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}, \quad Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}.$$

To this end, we first solve

$$L_{11}Y_{11} = B_{11}, \quad L_{11}Y_{12} = B_{12}$$

to find Y_{11}, Y_{12} . Afterwards, we get Y_{21}, Y_{22} from

$$L_{22}Y_{21} = B_{21} - L_{21}Y_{11}, \quad L_{22}Y_{22} = B_{22} - L_{21}Y_{12}.$$

We observe that this step involves matrix additions and multiplications in the HODLR format which in general require a rank truncation.

The actual computation of the LU factorization can now be performed recursively. On level $\ell = 1$, the matrices A, L, U have the block structure

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}, \quad U = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix},$$

which leads to the equations

$$A_{11} = L_{11}U_{11}, \quad A_{12} = L_{11}U_{12}, \quad A_{21} = L_{21}U_{11}, \quad A_{22} = L_{21}U_{12} + L_{22}U_{22}.$$

We can therefore proceed in the following way:

1. compute the LU factors L_{11}, U_{11} of A_{11} ,
2. compute $U_{12} = L_{11}^{-1}A_{12}$ by forward substitution,
3. compute $L_{21} = A_{21}U_{11}^{-1}$ by backward substitution,
4. compute the LU factors L_{22}, U_{22} of $A_{22} - L_{21}U_{12}$.

Each of these four steps can be addressed using LU factorization or forward and backward substitution for matrices of size $n/2$. On level $\ell = p$ we use dense routines to solve the subproblems of size n_p . By similar arguments as above one can show that the cost c_{LU} for the approximate LU factorization is not larger than the cost for the matrix-matrix multiplication of two HODLR matrices, i.e.

$$c_{LU}(n) \lesssim c_{H\cdot H}(n) = O(k^3 n \log n + k^2 n \log^2 n).$$

If the matrix A happens to be symmetric positive definite, an approximate Cholesky factorization $A \approx LL^T$ can be obtained by the same techniques with a reduction of the computational cost.

3.2 General Clustering Strategies

In many applications, the particular block partitioning of HODLR matrices is not appropriate and leads either to a poor approximation or to large ranks. Obtaining a block partitioning that is adapted to the situation at hand is by no means trivial and referred to as *clustering*. The first step of *clustering* typically consists of ordering the index set $I = \{1, \dots, n\}$ of $A \in \mathbb{R}^{n \times n}$. The second step consists of recursively subdividing the blocks of the reordered matrix A . Our main goals in the clustering

process are the following:

- (a) The ranks of all blocks should be small relative to n .
- (b) The total number of blocks should be small.

Clearly, we need to balance both goals in some way. The reordering of A and the subdivision of the blocks are often driven by some notion of locality between the indices. This locality could be induced by the geometry of the underlying problem or by the incidence graph of A (if A is sparse). In the following, we illustrate both strategies. First, we consider a geometrically motivated clustering strategy related to the concepts from Sect. 2.4.

3.2.1 Geometrical Clustering

Suppose we aim to find $u : \Omega \rightarrow \mathbb{R}$ satisfying a one-dimensional integral equation of the form

$$\int_0^1 \log|x - y| u(y) dy = f(x), \quad x \in \Omega = [0, 1],$$

with a given right-hand side $f : \Omega \rightarrow \mathbb{R}$. For $n = 2^p$, $p \in \mathbb{N}$, let the interval $[0, 1]$ be subdivided into subintervals

$$\tau_i := [(i-1)h, ih], \quad 1 \leq i \leq n,$$

where $h := 1/n$. A Galerkin discretization with piecewise constant basis function

$$\varphi_i(x) := \begin{cases} 1, & x \in \tau_i, \\ 0, & \text{else,} \end{cases}$$

$i = 1, \dots, n$, then leads to a matrix $A \in \mathbb{R}^{n \times n}$ defined by

$$A(i, j) := \int_0^1 \int_0^1 \varphi_i(x) \log|x - y| \varphi_j(y) dy dx = \int_{\tau_i} \int_{\tau_j} \log|x - y| dy dx.$$

This is a variation of the simplified setting discussed in Sect. 2.4, cf. (19), and the results of Sect. 2.4 can be easily adjusted. In particular, a semi-separable approximation $\log|x - y| \approx \sum_{\ell=1}^k g_\ell(x) h_\ell(y)$ yields a rank- k approximation UV^T by setting the entries of the matrices U, V to $U(i, \ell) = \int_{\tau_i} g_\ell(x) dx$ and $V(j, \ell) = \int_{\tau_j} h_\ell(y) dy$.

We recall from Sect. 2.4 that the diagonal singularity of $\log|x - y|$ requires us to impose the admissibility condition (25). To translate this into a condition on the

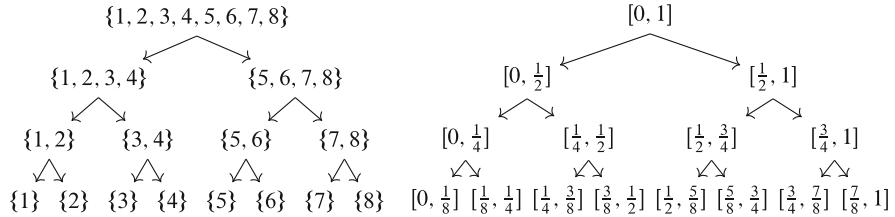


Fig. 6 Left: cluster tree T_I . Right: partition of $\Omega = [0, 1]$ with respect to T_I

blocks of A , let us define

$$\Omega_t := \bigcup_{i \in t} \tau_i \subset \Omega$$

for any index set $t \subset I = \{1, \dots, n\}$.

Definition 1 Let $\eta > 0$ and let $s, t \subset I$. We call a matrix block (s, t) , associated with the row indices s and column indices t , *admissible* if

$$\min\{\text{diam}(\Omega_s), \text{diam}(\Omega_t)\} \leq \eta \text{dist}(\Omega_s, \Omega_t).$$

Thus, a block is admissible if all its entries stay sufficiently far away from the diagonal, relative to the size of the block. To systematically generate a collection of admissible blocks, we first subdivide the index set I by a binary tree T_I such that neighboring indices are hierarchically grouped together. An example for $n = 8$ is depicted in Fig. 6. Each node $t \in T_I$ in the tree is an index set $t \subset I$ that is the union of its sons whenever t is not a leaf node. The tree T_I is then used to recursively define a block subdivision of the matrix A , by applying Algorithm 5 starting with $s, t = I$.

Algorithm 5 BuildBlocks(s, t)

```

1: if  $(s, t)$  is admissible or both  $s$  and  $t$  have no sons then
2:   fix the block  $(s, t)$ 
3: else
4:   for all sons  $s'$  of  $s$  and all sons  $t'$  of  $t$  do
5:     BuildBlocks( $s', t'$ )
6:   end for
7: end if

```

With $\eta := 1$ in the admissibility condition, Algorithm 5 applied to the one-dimensional example from above yields the block structure shown in Fig. 7.

The structure is very similar to the HODLR format except that now also parts close to subdiagonal are stored as dense matrices. In fact, a slightly weaker

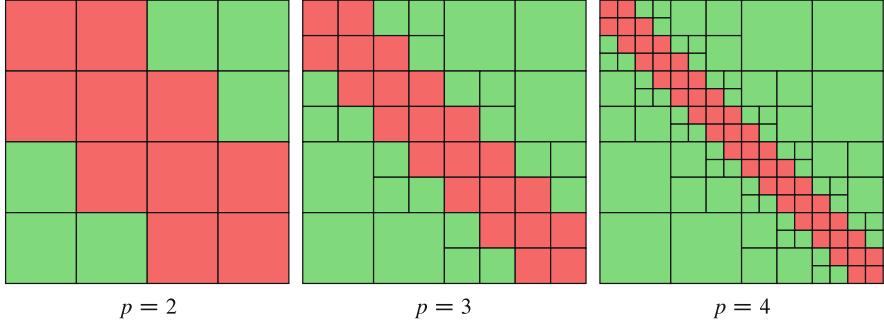


Fig. 7 Block subdivision with admissibility constant $\eta = 1$

admissibility condition than in Definition 1 would have resulted in the block structure from Fig. 5, see [44].

Clearly, the concentration of dense blocks towards the diagonal is only typical for one-dimensional problems. Fortunately, the construction above easily generalizes to problems in higher dimensions with $\Omega \subset \mathbb{R}^d$, $d > 1$. This requires a suitable adaptation of the construction of the cluster tree T_I , from which the block structure can be derived using Algorithm 5, see [19, 42] for details.

3.2.2 Algebraic Clustering

When $A \in \mathbb{R}^{n \times n}$ is sparse, a suitable partition of the blocks of A can sometimes successfully be obtained solely by the sparsity of A , i.e., without having to take the geometry of the original problem into account. We briefly illustrate the basic ideas of such an algebraic clustering strategy and refer to [7] for details. Let $G = (V, E)$ be the incidence graph of A , where $V = I = \{1, \dots, n\}$ denotes the set of vertices and E denotes the set of edges. For simplicity, let us assume that A is symmetric and we thus obtain an undirected graph with

$$(i, j) \in E \Leftrightarrow A(i, j) \neq 0.$$

Any subdivision of the graph G into subgraphs corresponds to a partition of the index set I . By the very basic property that a matrix with k nonzero entries has rank at most k , we aim to determine subgraphs such that the number of connecting edges between the subgraphs remains small. For this purpose, we partition

$$I = I_1 \dot{\cup} I_2$$

with $\#I_1 \approx \#I_2$ such that $\#\{(i, j) \in E : i \in I_1, j \in I_2\}$ is small. This process is continued recursively by subdividing I_1 and I_2 in the same way.

In general, the minimization of the number of connecting edges between subgraphs is an NP-hard problem. However, efficient approximations to this problem are well known and can be found, e.g., in the software packages METIS or Scotch. The subdivision process can be represented by a cluster tree T_I .

We still need to define a suitable admissibility condition in order to turn T_I into a subdivision of the block index set $I \times I$. Given vertices $i, j \in I$, we denote by $\text{dist}(i, j)$ the length of a shortest path from i to j in the graph G . Moreover, for $s, t \subset I$ let

$$\text{diam}(s) := \max_{i, j \in s} \text{dist}(i, j), \quad \text{dist}(s, t) := \min_{i \in s, j \in t} \text{dist}(i, j).$$

In analogy to Definition 1, we call the block $b = s \times t$ admissible if

$$\min\{\text{diam}(s), \text{diam}(t)\} \leq \eta \text{dist}(s, t) \quad (32)$$

for some constant $\eta > 0$. Based on the cluster tree T_I and the *algebraic* admissibility condition (32), we can then construct the partition P by Algorithm 5.

A variant of this partition strategy is to combine the clustering process with nested dissection. To this end, we subdivide I into

$$I = I_1 \dot{\cup} I_2 \dot{\cup} S$$

such that $(i, j) \notin E$ for all $i \in I_1, j \in I_2$. The set S is called a *separator* of G . This leads to a permuted matrix A of the form

$$A = \begin{pmatrix} A(I_1, I_1) & 0 & A(I_1, S) \\ 0 & A(I_2, I_2) & A(I_2, S) \\ A(S, I_1) & A(S, I_2) & A(S, S) \end{pmatrix}. \quad (33)$$

Note that we typically have $\#S \ll \#I$ such that we obtain large off-diagonal blocks that completely vanish. As before, we can recursively continue the subdivision process with the index sets I_1 and I_2 which in turn moves a large number of zeros to the off-diagonal blocks.

For a matrix A admitting an LU factorization $A = LU$, a particularly nice feature of this strategy is that the zero blocks (33) are inherited by L and U . This means that we can incorporate this information a priori in the construction of a block-wise data-sparse approximation of the factors L and U . For an existence result on low-rank approximations to admissible blocks in the factors L and U and in the inverse matrix A^{-1} see [7].

3.3 \mathcal{H} -Matrices

Hierarchical (\mathcal{H} -) matrices represent a generalization of HODLR matrices from Sect. 3.1, see [6, 19, 40, 42]. In the HODLR format, the block structure of a matrix $A \in \mathbb{R}^{n \times n}$ was derived from a perfect binary tree subdividing the index set $I = \{1, \dots, n\}$. In the \mathcal{H} -matrix format, much more general block structures appear. On the one hand, this allows for more freedom and increases the applicability of hierarchical matrices. On the other hand, the design of efficient algorithms can become more complicated.

We start with a general partition P of the block index set $I \times I$ of A . In accordance with Algorithm 5, this partition is often constructed from a cluster tree T_I by means of an admissibility condition as in Definition 1 such that each block $b \in P$ is of the form $b = s \times t$ with $s, t \in T_I$. The partition can be split into two sets $P = P^+ \dot{\cup} P^-$ where

$$P^+ := \{b \in P : b \text{ is admissible}\}, \quad P^- := P \setminus P^+.$$

For fixed P and a given maximal rank $k \in \mathbb{N}$, the set of hierarchical matrices is now defined as

$$\mathcal{H}(P, k) = \{A \in \mathbb{R}^{I \times I} : \text{rank}(A(b)) \leq k \ \forall b \in P^+\}.$$

All blocks $b \in P^+$ are stored by a low-rank representation, whereas all blocks $b \in P^-$ are kept as dense matrices. As for the HODLR case, each block $b = s \times t \in P^+$ of $A \in \mathcal{H}(P, k)$ is represented independently as

$$A(b) = U_b V_b^T, \quad U_b \in \mathbb{R}^{s \times k_b}, \quad V_b \in \mathbb{R}^{t \times k_b}$$

with a possibly different rank $k_b \leq k$ in each block.

Since a low-rank block $b = s \times t$ has storage complexity $k_b(\#s + \#t)$, the storage cost for a hierarchical matrix is given by

$$\text{stor}_{\mathcal{H}} = \sum_{b=s \times t \in P^+} k_b(\#s + \#t) + \sum_{b=s \times t \in P^-} \#s \#t.$$

If we assume that all non-admissible blocks $s \times t \in P^-$ are subject to $\min\{\#s, \#t\} \leq n_{\min}$ for some minimal cluster size $n_{\min} \in \mathbb{N}$, we readily estimate $\#s \#t \leq n_{\min}(\#s + \#t)$ which implies

$$\text{stor}_{\mathcal{H}} \leq \max\{n_{\min}, k\} \sum_{s \times t \in P} (\#s + \#t).$$

To proceed from here, we need to count the number of blocks $b \in P$ and weigh them according to their size. For each $s \in T_I$, the number of blocks in the partition

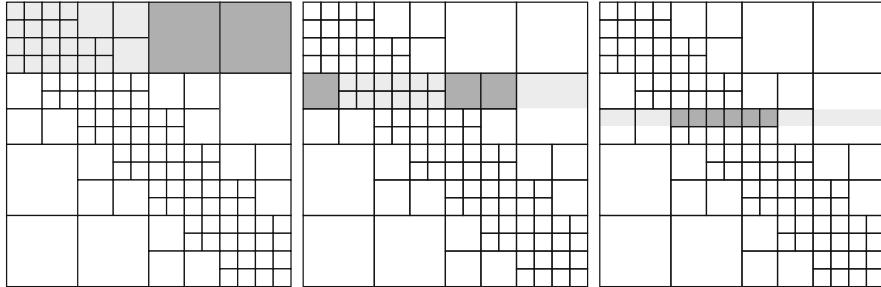


Fig. 8 Number of blocks in the partition P for different row index sets $s \in T_I$

P with row index set s is given by $\#\{t \in T_I : s \times t \in P\}$. Accordingly, the number of blocks in P with column index set t is given by $\#\{s \in T_I : s \times t \in P\}$. The maximum of both quantities over all $s, t \in T_I$ is called the *sparsity constant*

$$C_{\text{sp}}(P) := \max \left\{ \max_{s \in T_I} \#\{t \in T_I : s \times t \in P\}, \max_{t \in T_I} \#\{s \in T_I : s \times t \in P\} \right\}$$

of the partition P .

Example 6 Consider the block structure from Fig. 7 for $p = 4$. In Fig. 8 we can see that for this particular case $C_{\text{sp}}(P) = 6$. Interestingly, this number does not increase for higher values of p .

For each fixed $s \in T_I$, the term $\#s$ can hence only appear at most $C_{\text{sp}}(P)$ times in the sum $\sum_{s \times t \in P} \#s$, i.e.

$$\sum_{s \times t \in P} \#s \leq C_{\text{sp}}(P) \sum_{s \in T_I} \#s.$$

Moreover, we immediately observe that on all levels of the tree T_I an index set $s \subset I$ appears at most once such that

$$\sum_{s \in T_I} \#s \leq \#I(1 + \text{depth}(T_I)).$$

Assuming $n_{\min} \leq k$, the final complexity estimate is hence given by

$$\text{stor}_{\mathcal{H}} \leq 2C_{\text{sp}}(P)kn(1 + \text{depth}(T_I)).$$

For bounded $C_{\text{sp}}(P)$ and $\text{depth}(T_I) \in O(\log n)$ this corresponds to the storage complexity $\text{stor}_{\mathcal{H}} \in O(kn \log n)$ for HODLR matrices.

3.4 Approximation of Elliptic Boundary Value Problems by \mathcal{H} -Matrices

Let $\Omega \subset \mathbb{R}^d$ be a domain and consider the Poisson problem

$$\begin{aligned} Lu &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

for some given right-hand side f and $Lu = -\Delta u$. Using Green's function $G(x, y)$, we can formally write the solution u as

$$u(x) = (L^{-1}f)(x) = \int_{\Omega} G(x, y)f(y)dy. \quad (34)$$

For sufficiently smooth boundary $\partial\Omega$, one can show that $G(x, y)$ can be well approximated by a polynomial $p(x, y)$ for $x \in \sigma \subset \Omega$, $y \in \tau \subset \Omega$ whenever σ and τ are sufficiently far apart. If we split the polynomial p into its components depending on x and y , we hence find that

$$G(x, y) \approx p(x, y) = \sum_{j=1}^k u_j(x)v_j(y), \quad (x, y) \in \sigma \times \tau, \quad (35)$$

for univariate functions u_j, v_j .

Assume now that (34) is discretized by a Galerkin approximation with finite element basis functions φ_i , $i = 1, \dots, n$. This leads to a matrix $M \in \mathbb{R}^{n \times n}$ defined by

$$M(i, j) = \int_{\Omega} \int_{\Omega} \varphi_i(x)G(x, y)\varphi_j(y)dydx.$$

For $s, t \subset \{1, \dots, n\}$, let $\sigma_s := \bigcup_{i \in s} \text{supp } \varphi_i$, $\tau_t := \bigcup_{i \in t} \text{supp } \varphi_i$. Due to (35), we can approximate the matrix block $b = s \times t$ by a low-rank representation

$$M(b) \approx U_b V_b^T, \quad U_b \in \mathbb{R}^{s \times k}, V_b \in \mathbb{R}^{t \times k},$$

where

$$U_b(i, j) = \int_{\Omega} \varphi_i(x)u_j(x)dx, \quad V_b(i, j) = \int_{\Omega} \varphi_i(y)v_j(y)dy,$$

whenever the distance of σ_s and τ_t is large enough. Combined with a clustering strategy as in Sect. 3.2.1, this ensures the existence of \mathcal{H} -matrix approximations to discretizations of the inverse operator L^{-1} . Note however that the Green's

function $G(x, y)$ is usually not explicitly available and hence the polynomial approximation (35) remains a theoretical tool.

If the boundary $\partial\Omega$ is not smooth enough or the operator L depends on non-smooth coefficients, we cannot expect that a good polynomial approximation of $G(x, y)$ exists. Remarkably, for a general uniformly elliptic operator of the form

$$Lu = - \sum_{i,j=1}^d \partial_j(c_{ij}\partial_i u)$$

with $c_{ij} \in L^\infty(\Omega)$, one can still prove the existence of \mathcal{H} -matrix approximations to the inverse of finite-element matrices stiffness and mass matrices, cf. [8].

For the solution of boundary value problems, an explicit construction of an approximate inverse is often not required. As the next example illustrates, an approximate factorization of the discretized operator suffices and can easily be combined with standard iterative schemes.

Example 7 Let $\Omega = (0, 1)^3$ be the unit cube and consider the Poisson problem

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned}$$

with $f \equiv 1$. A finite difference discretization with $m + 2$ points in each spatial direction leads to a linear system

$$Ax = b \tag{36}$$

with $A \in \mathbb{R}^{n \times n}$, $n := m^3$, defined by

$$A = B \otimes E \otimes E + E \otimes B \otimes E + E \otimes E \otimes B,$$

where $B = \text{tridiag}(-1, 2, -1) \in \mathbb{R}^{m \times m}$ and E denotes the identity matrix of size m .

In order to solve (36) by a direct method, we could compute a Cholesky factorization $A = LL^T$. As Fig. 9 illustrates, the number of non-zeros in L increases rapidly with the matrix size. For the simple three-dimensional setting of this example one can show $\text{nnz}(L) = O(m^5) = O(n^{5/3})$.

Alternatively, we can use an iterative solver such as CG which immediately calls for a good preconditioner. In the ideal case, this means that the number of preconditioned CG iterations should be bounded independently of the matrix size n . On the other hand, the complexity for the application (or storage) of the preconditioner should not deteriorate with the matrix size. It turns out that an approximate Cholesky factorization $A \approx L_{\mathcal{H}} L_{\mathcal{H}}^T$ by hierarchical matrices actually fulfills both criteria.

In Fig. 10 we can see that the singular values in the off-diagonal blocks of the Cholesky factor $L_{\mathcal{H}}$ decay rapidly. We now iteratively compute a solution

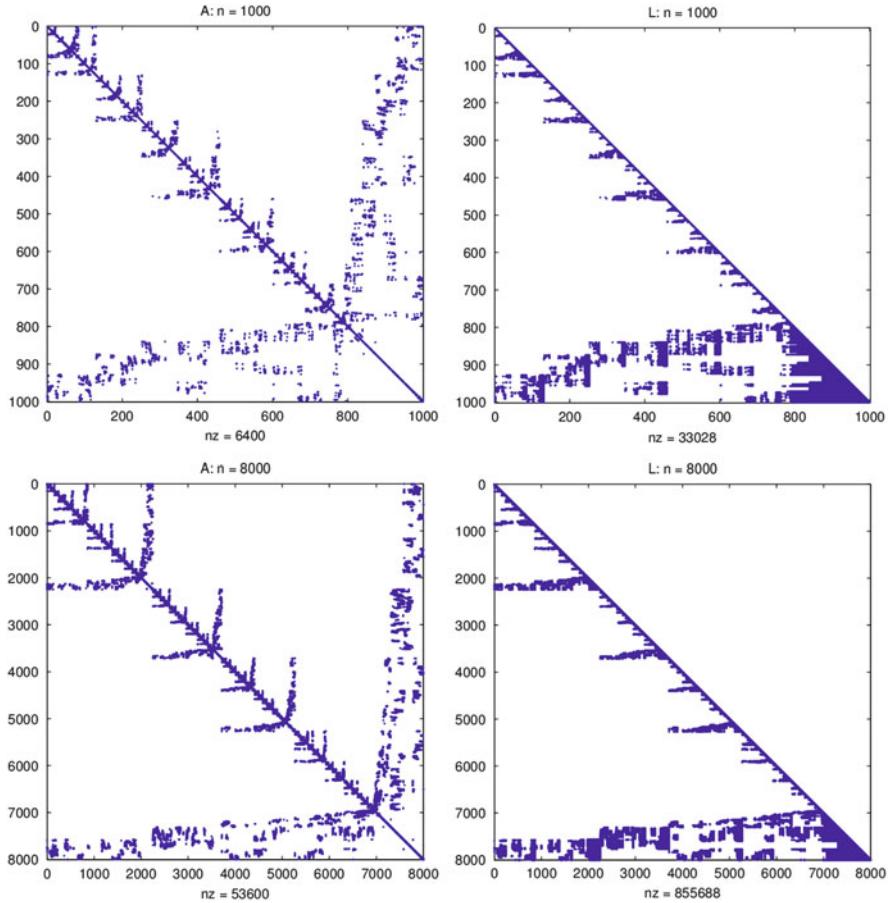


Fig. 9 Cholesky factorization of A from Example 7 with $n = 1000$ (top) and $n = 8000$ (bottom). *Left:* A after reordering with `symamd`. *Right:* Cholesky factor L with $\text{nnz}(L) > 33 \cdot 10^3$ (top) and $\text{nnz}(L) > 855 \cdot 10^3$ (bottom)

of (36) by preconditioned CG using different accuracies for the hierarchical matrix factorization. In Table 1 we can observe that the number of CG iterations stays constant for different sizes of the spatial discretization. Moreover, both the storage requirements for $L_{\mathcal{H}}$ and the time for the factorization and for the solution tend to grow as $O(n \log^* n)$. However, we also recognize an important drawback: the time for the factorization actually exceeds the solution time by far. Still, the computed preconditioner will be useful if the solution for multiple right-hand sides needs to be found.

Remark 4 Example 7 is for illustration only. There are much more efficient methods to solve problems of this class by, e.g., fast multipole or boundary element methods.

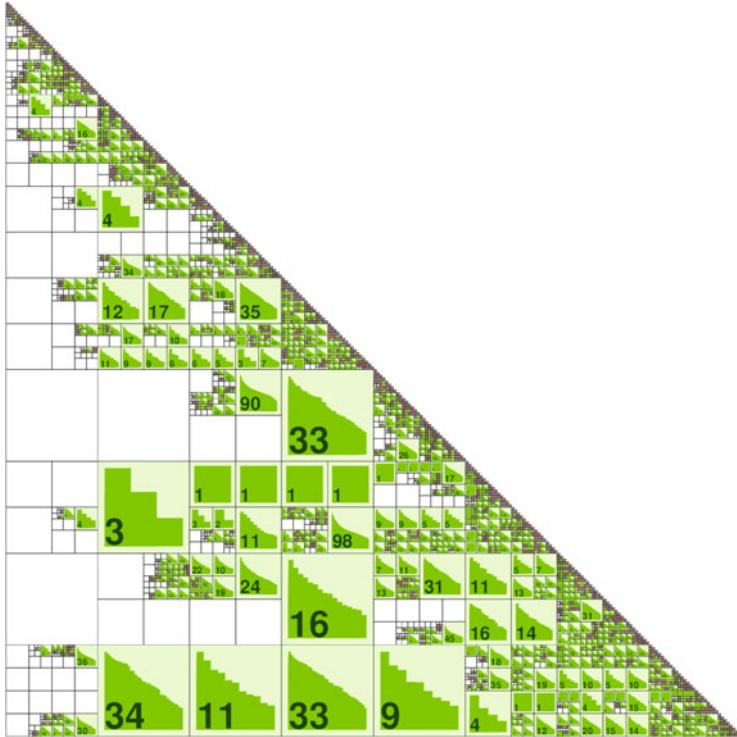


Fig. 10 Cholesky factor $L_{\mathcal{H}}$ of A from Example 7 with $n = 8000$ for $\varepsilon_{LU} = 10^{-4}$

4 Partitioned Low-Rank Structures with Nested Low-Rank Representations

4.1 HSS Matrices

Hierarchically Semi-Separable (HSS) or *Hierarchically Block-Separable (HBS)* matrices represent special cases of the HODLR matrices from Sect. 3.1, see, e.g., [60]. In the HODLR format, each off-diagonal block is expressed *independently* by a low-rank representation of the form (26). In the HSS format, the low-rank representations on different levels are nested and depend on each other in a hierarchical way.

Consider an off-diagonal block of a matrix A in HODLR format on level $0 < \ell < p$ with the low-rank representation $A(I_i^\ell, I_j^\ell) = U_i^{(\ell)} S_{i,j}^{(\ell)} (V_j^{(\ell)})^T$, where now $S_{i,j}^{(\ell)} \in \mathbb{R}^{k \times k}$. On level $\ell + 1$, the row index set is subdivided into $I_i^\ell = I_{2i-1}^{\ell+1} \dot{\cup} I_{2i}^{\ell+1}$ and the column index set into $I_j^\ell = I_{2j-1}^{\ell+1} \dot{\cup} I_{2j}^{\ell+1}$. The crucial assumption in the HSS

Table 1 Iterative solution of (36) by preconditioned CG with approximate Cholesky factorization $A \approx L_{\mathcal{H}} L_{\mathcal{H}}^T$ with accuracy ε_{LU}

n	ε_{LU}	$\ I - (L_{\mathcal{H}} L_{\mathcal{H}}^T)^{-1}A\ _2$	CG steps	Stor($L_{\mathcal{H}}$) [MB]	Time(chol) [s]	Time(solve) [s]
27,000	1e-01	7.82e-01	8	64	1.93	0.35
	1e-02	5.54e-02	3	85	3.28	0.21
	1e-03	3.88e-03	2	107	4.61	0.18
	1e-04	2.98e-04	2	137	6.65	0.22
	1e-05	2.32e-05	1	172	10.31	0.17
64,000	1e-01	9.11e-01	8	174	5.82	0.88
	1e-02	9.66e-02	4	255	10.22	0.62
	1e-03	6.56e-03	2	330	15.15	0.48
	1e-04	5.51e-04	2	428	23.78	0.54
	1e-05	4.53e-05	1	533	34.81	0.46
125,000	1e-01	1.15e+00	9	373	14.26	2.09
	1e-02	1.57e-01	4	542	25.21	1.32
	1e-03	1.19e-02	3	764	44.33	1.33
	1e-04	9.12e-04	2	991	65.86	1.19
	1e-05	7.37e-05	1	1210	97.62	1.01

format then states that there exist matrices $X_i^{(\ell)} \in \mathbb{R}^{2k \times k}$, $Y_j^{(\ell)} \in \mathbb{R}^{2k \times k}$ such that

$$U_i^{(\ell)} = \begin{bmatrix} U_{2i-1}^{(\ell+1)} & 0 \\ 0 & U_{2i}^{(\ell+1)} \end{bmatrix} X_i^{(\ell)}, \quad V_j^{(\ell)} = \begin{bmatrix} V_{2j-1}^{(\ell+1)} & 0 \\ 0 & V_{2j}^{(\ell+1)} \end{bmatrix} Y_j^{(\ell)}. \quad (37)$$

Given the nestedness condition (37), we can construct the row and column bases $U^{(\ell)}$ and $V^{(\ell)}$ for any $\ell = 1, \dots, p-1$ recursively from the bases $U^{(p)}$ and $V^{(p)}$ at the highest level p using the matrices $X^{(\ell)}$ and $Y^{(\ell)}$.

To estimate the storage requirements for HSS matrices, we need a more refined estimate than the one used for HODLR matrices. As before, we denote by $n_\ell = 2^{p-\ell} n_p$ the size of a block on level ℓ . It is clear that whenever $n_\ell \leq k$, all blocks on level ℓ can be represented as dense $n_\ell \times n_\ell$ matrices. The first level where this happens is

$$\ell_c := \min\{\ell : n_\ell \leq k\} = \lfloor p - \log_2(k/n_p) \rfloor.$$

To represent the bases on level ℓ , we need $2^{\ell-1}$ matrices $X^{(\ell)}, Y^{(\ell)}$ of size $2k \times k$ which sums up to

$$2 \sum_{\ell=1}^{\ell_c-1} 2^{\ell-1} 2k^2 \leq 4k^2 2^{\ell_c} \leq 8k^2 2^p n_p / k = 8kn$$

units of storage. As there are 2^ℓ low-rank blocks on level ℓ , an analogous estimate shows that we need $2kn$ storage units to represent all matrices $S^{(\ell)}$ for $\ell = 1, \dots, \ell_c - 1$. The cost for storing all off-diagonal blocks on level $\ell = \ell_c, \dots, p$ amounts to

$$\sum_{\ell=\ell_c}^p 2^\ell n_\ell^2 = \sum_{\ell=\ell_c}^{p-1} 2^\ell 2^{2p-2\ell} n_p = 2^{2p} \sum_{\ell=\ell_c}^{p-1} 2^{-\ell} \leq 2 \cdot 2^{2p} 2^{-\ell_c} \leq 2 \cdot 2^{2p} kn_p / 2^p = 2kn.$$

Note that the storage cost for the diagonal blocks of A is again given by last term of this sum. As a final result, the storage complexity for a matrix in the HSS format sums up to $O(kn)$.

The matrix-vector multiplication $y = Ax$ for $x \in \mathbb{R}^n$ in the HSS format is carried out in three steps. In the first step (*forward transformation*), the restriction of the vector x to an index set $I_j^\ell \subset I$ is transformed via the basis $V_j^{(\ell)}$. The resulting vectors x_j^ℓ are then multiplied with the coupling matrices $S_{i,j}^{(\ell)}$ (*multiplication phase*). In a third step (*backward transformation*), the results y_i^ℓ from this multiplication are multiplied with the bases $U_i^{(\ell)}$ and added up. Due to the nestedness (37), Algorithm 6 has a bottom-to-top-to-bottom structure.

Algorithm 6 Compute $y = Ax$ in HSS format

```

1: On level  $\ell = p$  compute
    $x_i^p = (V_i^{(p)})^T x(I_i^p), \quad i = 1, \dots, 2^p$  % forward substitution
2: for level  $\ell = p-1, \dots, 1$  do
3:   Compute
      
$$x_i^\ell = (Y_i^{(\ell)})^T \begin{bmatrix} x_{2i-1}^{\ell+1} \\ x_{2i}^{\ell+1} \end{bmatrix}, \quad i = 1, \dots, 2^\ell$$

4: end for
5: for level  $\ell = 1, \dots, p-1$  do
6:   Compute
      
$$\begin{bmatrix} y_{2i-1}^{(\ell)} \\ y_{2i}^{(\ell)} \end{bmatrix} = \begin{bmatrix} 0 & S_{2i-1, 2i}^{(\ell)} \\ S_{2i, 2i-1}^{(\ell)} & 0 \end{bmatrix} \begin{bmatrix} x_{2i-1}^\ell \\ x_{2i}^\ell \end{bmatrix}, \quad i = 1, \dots, 2^{\ell-1}$$
 % multiplication
7: end for
8: for level  $\ell = 1, \dots, p-1$  do
9:   Compute
      
$$\begin{bmatrix} y_{2i-1}^{\ell+1} \\ y_{2i}^{\ell+1} \end{bmatrix} = \begin{bmatrix} y_{2i-1}^{\ell+1} \\ y_{2i}^{\ell+1} \end{bmatrix} + X_i^{(\ell)} y_i^\ell, \quad i = 1, \dots, 2^\ell$$
 % backward substitution
10: end for
11: On level  $\ell = p$  compute
     $y(I_i^p) = U_i^{(p)} y_i^p + A(I_i^p, I_i^p) x(I_i^p), \quad i = 1, \dots, 2^p$ 

```

Except for level $\ell = p$, only the matrices $X^{(\ell)}, Y^{(\ell)}, S^{(\ell)}$ are used for the matrix-vector multiplication. On level $\ell = p$, the last line in Algorithm 6 takes into account the contributions from the diagonal blocks of A . Since all matrices from

the HSS representation are touched only once, the complexity for the matrix-vector multiplication in the HSS format is the same as the storage complexity, i.e. $O(kn)$. Interestingly, the HSS matrix-vector multiplication has a direct correspondence to the fast multipole method [38], see [73] and the references therein for a discussion.

Analogous to HODLR matrices, the HSS format allows for approximate arithmetic operations such as matrix-matrix additions and multiplications, see [60] for an overview. As a consequence, approximate factorizations and an approximate inverse of a matrix are computable fully in the HSS format. This is exploited in the development of efficient solvers and preconditioners that scale linearly in the matrix size n , see [2, 24, 29, 31, 32, 39, 53, 54, 69, 72] for examples.

4.2 \mathcal{H}^2 -Matrices

\mathcal{H}^2 -matrices are a generalization of HSS matrices from Sect. 4.1 and a special case of \mathcal{H} -matrices from Sect. 3.3, see [17] and the references therein. We start again with a partition P of the block index set $I \times I$ such that all $b \in P$ have the form $b = s \times t$ with $s, t \in T_I$ for a cluster tree T_I . For $A \in \mathcal{H}(P, k)$, we require that each admissible block $b = s \times t \in P^+$ possesses a low-rank representation of the form

$$A(b) = U_s S_b V_t^T, \quad U_s \in \mathbb{R}^{s \times k_s}, \quad V_t \in \mathbb{R}^{t \times K_t}, \quad S_b \in \mathbb{R}^{k_s \times K_t}. \quad (38)$$

With this condition, the low-rank representations in each block $b \in P^+$ are no longer independent since we have fixed a particular basis (or frame) U_s for each row index set $s \in T_I$ and a particular V_t for each column index set $t \in T_I$. \mathcal{H} -matrices with the additional property (38) are called *uniform \mathcal{H}* -matrices.

For each admissible block b of a uniform \mathcal{H} -matrix, we do not store its low-rank representation separately but only the coupling matrix S_b . In addition, the bases U_s, V_t need only to be stored once for each $s, t \in T_I$. This leads to a storage cost of

$$\text{stor}_{\mathcal{H}\text{uniform}} = \sum_{s \in T_I} k_s \#s + \sum_{t \in T_I} K_t \#t + \sum_{b=s \times t \in P^+} k_s K_t + \sum_{b=s \times t \in P^-} \#s \#t.$$

A further reduction in the storage cost can be obtained if we impose an additional restriction on the matrices U_s, V_t . Let $\mathcal{L}(T_I)$ denote the set of leaves of the cluster tree T_I . We call the family of matrices $(U_s)_{s \in T_I}$ *nested* if for each non-leaf index set $s \in T_I \setminus \mathcal{L}(T_I)$ with sons $s' \in T_I$ there exists a matrix $X_{s,s'} \in \mathbb{R}^{k_{s'} \times k_s}$ such that

$$U_s(s', :) = U_{s'} X_{s,s'}. \quad (39)$$

Accordingly, the family $(V_t)_{t \in T_I}$ is nested if for $t \in T_I \setminus \mathcal{L}(T_I)$ with sons $t' \in T_I$ there exists a matrix $Y_{t,t'} \in \mathbb{R}^{K_{t'} \times K_t}$ such that

$$V_t(t', :) = V_{t'} Y_{t,t'}. \quad (40)$$

A uniform \mathcal{H} -matrix with nested families $(U_s)_{s \in T_I}, (V_t)_{t \in T_I}$ is called an \mathcal{H}^2 -matrix.

Due to the nestedness conditions (39), (40), we only have to store the matrices U_s, V_t in the leaves $s, t \in \mathcal{L}(T_I)$ which requires at most $2kn$ units of storage. For all $s' \in T_I \setminus \{I\}$ with father $s \in T_I$, we can store the transfer matrices $X_{s,s'}$ in a complexity of $k^2 \#T_I$. The number of nodes in the tree T_I can be estimated as $\#T_I \leq 2\#\mathcal{L}(T_I)$. Including the cost for the transfer matrices $Y_{t,t'}$ and for the coupling matrices S_b for all blocks, we arrive at a storage cost of

$$\text{stor}_{\mathcal{H}^2} \leq 2kn + 4k^2\#\mathcal{L}(T_I) + \sum_{b=s \times t \in P^+} k_s K_t + \sum_{b=s \times t \in P^-} \#s \#t.$$

Note that if $\#\mathcal{L}(T_I) \sim n/n_{\min}$ and $n_{\min} \sim k$, the second term in this sum is of order $O(kn)$.

Example 8 To illustrate an explicit construction of an \mathcal{H}^2 -matrix, we recall the one-dimensional problem from Sect. 3.2.1 with $A \in \mathbb{R}^{n \times n}$ defined by

$$A(i,j) = \int_{\tau_i} \int_{\tau_j} \kappa(x,y) dy dx$$

for $\kappa(x,y) := \log|x-y|$. Given an admissible block $b = s \times t$, we let $\Omega_s := \bigcup_{i \in s} \tau_i$, $\Omega_t := \bigcup_{j \in t} \tau_j$ and apply the interpolation technique from Sect. 2.4 to approximate κ on $\Omega_s \times \Omega_t$ by

$$\kappa_k(x,y) := \sum_{\mu=1}^k \sum_{v=1}^k \kappa(\xi_\mu^{(s)}, \theta_v^{(t)}) L_{\xi_\mu^{(s)}, \mu}(x) L_{\theta_v^{(t)}, v}(y), \quad x \in \Omega_s, y \in \Omega_t,$$

with Lagrange polynomials $L_{\xi_\mu^{(s)}, \mu} : \Omega_s \rightarrow \mathbb{R}$, $L_{\theta_v^{(t)}, v} : \Omega_t \rightarrow \mathbb{R}$, and interpolation points $\xi_\mu^{(s)} \in \Omega_s$, $\theta_v^{(t)} \in \Omega_t$. Every entry $(i,j) \in s \times t$ of the submatrix $A(s,t)$ can then be approximated by

$$\begin{aligned} A(i,j) &\approx \int_{\tau_i} \int_{\tau_j} \kappa_k(x,y) dy dx \\ &= \sum_{\mu=1}^k \sum_{v=1}^k \kappa(\xi_\mu^{(s)}, \theta_v^{(t)}) \int_{\tau_i} L_{\xi_\mu^{(s)}, \mu}(x) dx \int_{\tau_j} L_{\theta_v^{(t)}, v}(y) dy = (U_s S_b V_t^T)_{ij}, \end{aligned}$$

where

$$U_s(i,\mu) := \int_{\tau_i} L_{\xi_\mu^{(s)}, \mu}(x) dx, \quad V_t(j,v) := \int_{\tau_j} L_{\theta_v^{(t)}, v}(y) dy, \quad S_b(\mu, v) := \kappa(\xi_\mu^{(s)}, \theta_v^{(t)}).$$

Since the matrices U_s and V_t do only depend on the clusters s, t , respectively, this approximation gives already rise to a uniform \mathcal{H} -matrix. Provided that the

interpolation nodes are mutually different, the first k Lagrange polynomials form a basis for the space of polynomials of order at most $k-1$. This allows us to express the Lagrange polynomial $L_{\xi^{(s)}, \mu}$ for the cluster s as a linear combination of the Lagrange polynomials for another cluster s' :

$$L_{\xi^{(s)}, \mu} = \sum_{v=1}^k X_{s, s'}(v, \mu) L_{\xi^{(s')}, v}$$

with a matrix $X_{s, s'} \in \mathbb{R}^{k \times k}$ defined by $X_{s, s'}(v, \mu) = L_{\xi^{(s)}, \mu}(\xi_v^{(s')})$. In particular, if s' is a son of $s \in T_I$ and $i \in s'$, we get

$$U_s(i, \mu) = \int_{\tau_i} L_{\xi^{(s)}, \mu}(x) dx = \sum_{v=1}^k X_{s, s'}(v, \mu) \int_{\tau_i} L_{\xi^{(s')}, v}(x) dx = (U_{s'} X_{s, s'})_{i, \mu}.$$

We immediately conclude that the family of matrices $(U_s)_{s \in T_I}$ is nested. Analogously, the same holds for $(V_t)_{t \in T_I}$. In summary, (tensorized) Lagrange interpolation allows us to obtain an approximation of A in the \mathcal{H}^2 -matrix format.

Analogous to HSS and \mathcal{H} -matrices, the matrix-vector product for an \mathcal{H}^2 -matrix is carried out within $O(nk)$ operations. However, the implementation of approximate algebraic operations such as matrix-matrix additions and multiplications in the \mathcal{H}^2 -matrix format is significantly more difficult than for HODLR or \mathcal{H} -matrices as one needs to maintain the nestedness structure (39)–(40). Note that this also makes the computation of approximate factorizations and an approximate inverse in the \mathcal{H}^2 -matrix format more involved, see [17] for details.

Acknowledgements The first author has been supported by an EPFL fellowship through the European Union's Seventh Framework Programme under grant agreement no. 291771.

References

1. S. Ambikasaran, D. Foreman-Mackey, L. Greengard, D.W. Hogg, M. O'Neil, Fast direct methods for Gaussian processes. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(2), 252–265 (2016)
2. A. Aminfar, E. Darve, A fast sparse solver for finite-element matrices. Preprint arXiv:1410.2697 (2014)
3. J. Ballani, D. Kressner, Sparse inverse covariance estimation with hierarchical matrices. Technical Report, EPFL-MATHICSE-ANCHP, Oct (2014)
4. U. Baur, P. Benner, Factorized solution of Lyapunov equations based on hierarchical matrix arithmetic. *Computing* **78**(3), 211–234 (2006)
5. M. Bebendorf, Approximation of boundary element matrices. *Numer. Math.* **86**(4), 565–589 (2000)
6. M. Bebendorf, *Hierarchical Matrices*. Lecture Notes in Computational Science and Engineering, vol. 63 (Springer, Berlin, 2008)

7. M. Bebendorf, T. Fischer, On the purely algebraic data-sparse approximation of the inverse and the triangular factors of sparse matrices. *Numer. Linear Algebra Appl.* **18**(1), 105–122 (2011)
8. M. Bebendorf, W. Hackbusch, Existence of \mathcal{H} -matrix approximants to the inverse FE-matrix of elliptic operators with L^∞ -coefficients. *Numer. Math.* **95**(1), 1–28 (2003)
9. M. Bebendorf, S. Rjasanow, Adaptive low-rank approximation of collocation matrices. *Computing* **70**(1), 1–24 (2003)
10. M. Bebendorf, Y. Maday, B. Stamm, Comparison of some reduced representation approximations, in *Reduced Order Methods for Modeling and Computational Reduction*. MS & A Modelling, Simulation and Applications, vol. 9 (Springer, Cham, 2014), pp. 67–100
11. B. Beckermann, The condition number of real Vandermonde, Krylov and positive definite Hankel matrices. *Numer. Math.* **85**(4), 553–577 (2000)
12. P. Benner, T. Mach, Computing all or some eigenvalues of symmetric \mathcal{H}_ℓ -matrices. *SIAM J. Sci. Comput.* **34**(1), A485–A496 (2012)
13. P. Benner, T. Mach, The preconditioned inverse iteration for hierarchical matrices. *Numer. Linear Algebra Appl.* **20**, 150–166 (2013)
14. P. Benner, S. Börm, T. Mach, K. Reimer, Computing the eigenvalues of symmetric \mathcal{H}^2 -matrices by slicing the spectrum. *Comput. Visual. Sci.* **16**, 271–282 (2013)
15. M. Benzi, N. Razouk, Decay bounds and $O(n)$ algorithms for approximating functions of sparse matrices. *Electron. Trans. Numer. Anal.* **28**, 16–39 (2007/2008)
16. M.W. Berry, Large scale singular value decomposition. *Internat. J. Supercomput. Appl.* **6**, 13–49 (1992)
17. S. Börm, Efficient numerical methods for non-local operators, in *EMS Tracts in Mathematics*, vol. 14 (European Mathematical Society (EMS), Zürich, 2010)
18. S. Börm, L. Grasedyck, Hybrid cross approximation of integral operators. *Numer. Math.* **101**(2), 221–249 (2005)
19. S. Börm, L. Grasedyck, W. Hackbusch, *Hierarchical Matrices*. Lecture Notes 21 (MPI for Mathematics in the Sciences, Leipzig, 2003)
20. A. Çivril, M. Magdon-Ismail, Exponential inapproximability of selecting a maximum volume sub-matrix. *Algorithmica* **65**(1), 159–176 (2013)
21. S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A.-J. van der Veen, D. White, Some fast algorithms for sequentially semiseparable representations. *SIAM J. Matrix Anal. Appl.* **27**(2), 341–364 (2005)
22. J. Chiu, L. Demanet, Sublinear randomized algorithms for skeleton decompositions. *SIAM J. Matrix Anal. Appl.* **34**(3), 1361–1383 (2013)
23. A. Çivril, M. Magdon-Ismail, On selecting a maximum volume sub-matrix of a matrix and related problems. *Theor. Comput. Sci.* **410**(47–49), 4801–4811 (2009)
24. E. Corona, P.-G. Martinsson, D. Zorin, An $O(N)$ direct solver for integral equations on the plane. *Appl. Comput. Harmon. Anal.* **38**(2), 284–317 (2015)
25. T.A. Davis, *Direct Methods for Sparse Linear Systems*. Fundamentals of Algorithms, vol. 2 (Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006)
26. S. Demko, W.F. Moss, P.W. Smith, Decay rates for inverses of band matrices. *Math. Comput.* **43**(168), 491–499 (1984)
27. J.W. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, Z. Drmač, Computing the singular value decomposition with high relative accuracy. *Linear Algebra Appl.* **299**(1–3), 21–80 (1999)
28. Z. Drmač, K. Veselić, New fast and accurate Jacobi SVD algorithm. I. *SIAM J. Matrix Anal. Appl.* **29**(4), 1322–1342 (2007)
29. P. Gatto, J.S. Hesthaven, A preconditioner based on low-rank approximation of Schur complements. Technical Report, EPFL-MATHICSE-MCSS (2015)
30. I.P. Gavrilyuk, W. Hackbusch, B.N. Khoromskij, \mathcal{H} -matrix approximation for the operator exponential with applications. *Numer. Math.* **92**(1), 83–111 (2002)
31. P. Ghysels, X.S. Li, F.-H. Rouet, S. Williams, A. Napov, An efficient multi-core implementation of a novel HSS-structured multifrontal solver using randomized sampling. *SIAM J. Sci. Comput.* **38**(5), S358–S384 (2016)

32. A. Gillman, P.M. Young, P.G. Martinsson, A direct solver with $O(N)$ complexity for integral equations on one-dimensional domains. *Front. Math. China* **7**(2), 217–247 (2012)
33. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 4th edn. (Johns Hopkins University Press, Baltimore, MD, 2013)
34. G.H. Golub, F.T. Luk, M.L. Overton, A block Lánczos method for computing the singular values of corresponding singular vectors of a matrix. *ACM Trans. Math. Softw.* **7**(2), 149–169 (1981)
35. S.A. Goreinov, E.E. Tyrtyshnikov, The maximal-volume concept in approximation by low-rank matrices, in *Structured Matrices in Mathematics, Computer Science, and Engineering, I (Boulder, CO, 1999)*. Contemporary Mathematics, vol. 280 (American Mathematical Society, Providence, RI, 2001), pp. 47–51
36. S.A. Goreinov, E.E. Tyrtyshnikov, N.L. Zamarashkin, A theory of pseudoskeleton approximations. *Linear Algebra Appl.* **261**, 1–21 (1997)
37. L. Grasedyck, W. Hackbusch, B.N. Khoromskij, Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices. *Computing* **70**(2), 121–165 (2003)
38. L. Greengard, V. Rokhlin, A fast algorithm for particle simulations. *J. Comput. Phys.* **73**(2), 325–348 (1987)
39. L. Greengard, D. Gueyffier, P.G. Martinsson, V. Rokhlin, Fast direct solvers for integral equations in complex three-dimensional domains. *Acta Numer.* **18**, 243–275 (2009)
40. W. Hackbusch, A sparse matrix arithmetic based on \mathcal{H} -matrices. Part I: Introduction to \mathcal{H} -matrices. *Computing* **62**(2), 89–108 (1999)
41. W. Hackbusch, Entwicklungen nach Exponentialsummen. Technical Report 4/2005, MPI MIS Leipzig (2010) Revised version Sept (2010)
42. W. Hackbusch, *Hierarchical Matrices: Algorithms and Analysis* (Springer, Berlin, 2015)
43. W. Hackbusch, New estimates for the recursive low-rank truncation of block-structured matrices. *Numer. Math.* **132**(2), 303–328 (2015)
44. W. Hackbusch, B.N. Khoromskij, R. Kriemann, Hierarchical matrices based on a weak admissibility criterion. *Computing* **73**(3), 207–243 (2004)
45. N. Halko, P.G. Martinsson, J.A. Tropp, Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* **53**(2), 217–288 (2011)
46. H. Harbrecht, M. Peters, R. Schneider, On the low-rank approximation by the pivoted Cholesky decomposition. *Appl. Numer. Math.* **62**(4), 428–440 (2012)
47. N.J. Higham, *Accuracy and Stability of Numerical Algorithms* (SIAM, Philadelphia, PA, 1996)
48. R.A. Horn, C.R. Johnson, *Matrix Analysis*, 2nd edn. (Cambridge University Press, Cambridge, 2013)
49. R.B. Lehoucq, D.C. Sorensen, C. Yang, *ARPACK Users' Guide* (SIAM, Philadelphia, PA, 1998)
50. J.Y. Li, S. Ambikasaran, E.F. Darve, P.K. Kitanidis, A Kalman filter powered by H^2 -matrices for quasi-continuous data assimilation problems. *Water Resour. Res.* **50**, 3734–3749 (2014)
51. D. Liu, H.G. Matthies, Pivoted Cholesky decomposition by cross approximation for efficient solution of kernel systems. Preprint arXiv:1505.06195 (2015)
52. Y. Maday, O. Mula, A.T. Patera, M. Yano, The generalized empirical interpolation method: stability theory on Hilbert spaces with an application to the Stokes equation. *Comput. Methods Appl. Mech. Eng.* **287**, 310–334 (2015)
53. P.G. Martinsson, V. Rokhlin, A fast direct solver for boundary integral equations in two dimensions. *J. Comput. Phys.* **205**(1), 1–23 (2005)
54. A. Napov, X.S. Li, An algebraic multifrontal preconditioner that exploits the low-rank property. *Numer. Linear Algebra Appl.* **23**(1), 61–82 (2016)
55. J. Ostrowski, M. Bebendorf, R. Hiptmair, and F. Krämer. H -matrix based operator preconditioning for full Maxwell at low frequencies. *IEEE Trans. Magn.* **46**(8), 3193–3196 (2010)
56. H. Pouransari, P. Coulier, E. Darve, Fast hierarchical solvers for sparse matrices. Preprint arXiv:1510.07363 (2015)

57. S.A. Sauter, C. Schwab, *Boundary Element Methods*. Springer Series in Computational Mathematics (Springer, Heidelberg, 2010)
58. B. Savas, I.S. Dhillon, Clustered low rank approximation of graphs in information science applications, in *SDM*, pp. 164–175 (2011)
59. R. Schneider and A. Uschmajew. Approximation rates for the hierarchical tensor format in periodic Sobolev spaces. *J. Complex.* **30**(2), 56–71 (2014)
60. Z. Sheng, P. Dewilde, S. Chandrasekaran, Algorithms to solve hierarchically semi-separable systems, in *System Theory, the Schur Algorithm and Multidimensional Analysis*, ed. by D. Alpay, V. Vinnikov. *Operator Theory: Advances and Applications*, vol. 176 (Birkhäuser Basel, 2007), pp. 255–294
61. S. Si, C.-J. Hsieh, I.S. Dhillon, Memory efficient kernel approximation, in *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pp. 701–709 (2014)
62. H.D. Simon, H. Zha, Low-rank matrix approximation using the Lanczos bidiagonalization process with applications. *SIAM J. Sci. Comput.* **21**(6), 2257–2274 (2000)
63. G.W. Stewart, Perturbation theory for the singular value decomposition, in *SVD and Signal Processing II: Algorithms, Analysis and Applications*, ed. by R.J. Vaccaro (Elsevier, Amsterdam, 1991)
64. E.E. Tyrtyshnikov, Incomplete cross approximation in the mosaic-skeleton method. *Computing* **64**(4), 367–380 (2000)
65. R. Vandebril, M. Van Barel, N. Mastronardi, A note on the representation and definition of semiseparable matrices. *Numer. Linear Algebra Appl.* **12**(8), 839–858 (2005)
66. R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices*, vol. 1 (Johns Hopkins University Press, Baltimore, MD, 2008)
67. R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices*, vol. II (Johns Hopkins University Press, Baltimore, MD, 2008)
68. J. Vogel, J. Xia, S. Cauley, B. Venkataraman, Superfast divide-and-conquer method and perturbation analysis for structured eigenvalue solutions. *SIAM J. Sci. Comput.* **38**(3), A1358–A1382 (2016)
69. S. Wang, X.S. Li, F. Rouet, J. Xia, M.V. de Hoop, A parallel geometric multifrontal solver using hierarchically semiseparable structure. *ACM Trans. Math. Softw.* **42**(3), 21:1–21:21 (2016)
70. R. Wang, R. Li, M.W. Mahoney, E. Darve, Structured block basis factorization for scalable kernel matrix evaluation. Preprint arXiv:1505.00398 (2015)
71. P.-Å. Wedin, Perturbation bounds in connection with singular value decomposition. *Nordisk Tidskr. Informationsbehandling (BIT)* **12**, 99–111 (1972)
72. J. Xia, S. Chandrasekaran, M. Gu, X.S. Li, Fast algorithms for hierarchically semiseparable matrices. *Numer. Linear Algebra Appl.* **17**(6), 953–976 (2010)
73. R. Yokota, G. Turkiiyah, D. Keyes, Communication complexity of the fast multipole method and its algebraic variants. Preprint arXiv:1406.1974 (2014)

Localization in Matrix Computations: Theory and Applications

Michele Benzi

Abstract Many important problems in mathematics and physics lead to (non-sparse) functions, vectors, or matrices in which the fraction of nonnegligible entries is vanishingly small compared to the total number of entries as the size of the system tends to infinity. In other words, the nonnegligible entries tend to be *localized*, or concentrated, around a small region within the computational domain, with rapid decay away from this region (uniformly as the system size grows). When present, localization opens up the possibility of developing fast approximation algorithms, the complexity of which scales linearly in the size of the problem. While localization already plays an important role in various areas of quantum physics and chemistry, it has received until recently relatively little attention by researchers in numerical linear algebra. In this chapter we survey localization phenomena arising in various fields, and we provide unified theoretical explanations for such phenomena using general results on the decay behavior of matrix functions. We also discuss computational implications for a range of applications.

1 Introduction

In numerical linear algebra, it is common to distinguish between *sparse* and *dense* matrix computations. An $n \times n$ sparse matrix A is one in which the number of nonzero entries is much smaller than n^2 for n large. It is generally understood that a matrix is dense if it is not sparse.¹ These are not, of course, formal definitions. A more precise definition of a sparse $n \times n$ matrix, used by some authors, requires that the number of nonzeros in A is $O(n)$ as $n \rightarrow \infty$. That is, the average number of nonzeros per row must remain bounded by a constant for large n . Note that this definition

¹Note that we do not discuss here the case of *data-sparse matrices*, which are thoroughly treated elsewhere in this book.

M. Benzi (✉)

Department of Mathematics and Computer Science, Emory University, Atlanta, GA 30322, USA
e-mail: benzi@mathcs.emory.edu

does not apply to a single matrix, but to a family of matrices parameterized by the dimension, n . The definition can be easily adapted to the case of non-square matrices, in particular to vectors.

The latter definition, while useful, is rather arbitrary. For instance, suppose we have a family of $n \times n$ matrices in which the number of nonzero entries behaves like $O(n^{1+\varepsilon})$ as $n \rightarrow \infty$, for some $\varepsilon \in (0, 1)$. Clearly, for such matrix family the fraction of nonzero entries vanishes as $n \rightarrow \infty$, and yet such matrices would not be regarded as sparse according to this definition.²

Another limitation of the usual definition of sparsity is that it does not take into account the *size* of the nonzeros. All nonzeros are treated as equals: a matrix is either sparse or not sparse (dense). As we shall see, there are many situations in computational practice where one encounters vectors or matrices in which virtually every entry is nonzero, but only a very small fraction of the entries has nonnegligible magnitude. A matrix of this kind is close to being sparse: it would become truly sparse (according to most definitions) upon *thresholding*, or *truncation* (i.e., the setting to zero of matrix elements smaller than a prescribed, sufficiently small quantity in absolute value). However, this assumes that entries are first computed, then set to zero if small enough, which could be an expensive and wasteful task. Failing to recognize this may lead to algorithms with typical $O(n^2)$ or $O(n^3)$ scaling for most matrix computation tasks. In contrast, careful exploitation of this property can lead to *linear scaling algorithms*, i.e., approximation algorithms with $O(n)$ computational complexity (in some cases even sublinear complexity may be possible). One way to accomplish this is to derive a priori bounds on the size of the elements, so as to know in advance which ones *not* to compute.

Matrices with the above-mentioned property are often referred to as being *localized*, or to exhibit *decay*.³ These terms are no more precise than the term “sparse” previously discussed, and one of the goals of these lectures is to provide precise formalizations of these notions. While the literature on sparse matrix computations is enormous, much less attention has been devoted by the numerical linear algebra community to the exploitation of localization in computational problems; it is our hope that these lectures will attract some interest in this interesting and important property, which is well known to computational physicists and chemists.

Just as sparse matrices are often structured, in the sense that the nonzeros in them are usually not distributed at random, so are localized matrices and vectors. The entries in them typically fit some type of decay behavior, such as exponential decay, away from certain clearly defined positions, for example the main diagonal. Many important computational problems admit localized solutions, and identifying this *hidden structure* (i.e., being able to predict the decay properties of the solution) can lead to efficient approximation algorithms. The aim of these lectures is to provide

²Perhaps a better definition is the one given in [72, p. 1]: “A matrix is **sparse** if there is an advantage in exploiting its zeros.”

³Occasionally, the term *pseudosparse* is used; see, e.g., [34].

the reader with the mathematical background and tools needed to understand and exploit localization in matrix computations.

We now proceed to give a brief (and by no means complete) overview of localization in physics and in numerical mathematics. Some of these examples will be discussed in greater detail in later sections.

1.1 Localization in Physics

Generally speaking, the term *locality* is used in physics to describe situations where the strength of interactions between the different parts of a system decay rapidly with the distance: in other words, correlations are *short-ranged*. Mathematically, this fact is expressed by saying that some function $\phi(\mathbf{r}, \mathbf{r}')$ decays rapidly to zero as the spatial separation $\|\mathbf{r} - \mathbf{r}'\|$ increases. The opposite of localization is *delocalization*: a function is delocalized if its values are nonnegligible on an extended region. In other words, if non-local (long-range) interactions are important, a system is delocalized. Locality (or lack of it) is of special importance in quantum chemistry and solid state physics, since the properties of molecules and the behavior of materials are strongly dependent on the presence (or absence) of localization.

Recall that in quantum mechanics the stationary states of a system of N particles are described by wave functions, $\Psi_n \in L^2(\mathbb{R}^{3N})$, $n = 0, 1, \dots$, normalized so that $\|\Psi_n\|_{L^2} = 1$. These states are stationary in the sense that a system initially in state Ψ_n will remain in it if left unperturbed. The probability that a system in the stationary state corresponding to Ψ_n is in a configuration \mathbf{x} belonging to a given region $\Omega \subseteq \mathbb{R}^{3N}$ is given by

$$\Pr(\text{system configuration } \mathbf{x} \in \Omega) = \int_{\Omega} |\Psi_n(\mathbf{x})|^2 d\mathbf{x}.$$

As an example, consider the electron in a hydrogen atom. We let $\mathbf{r} = (x, y, z) \in \mathbb{R}^3$ be the position of the electron with respect to the nucleus (supposed to be at the origin) and $r = \sqrt{x^2 + y^2 + z^2}$. The radial part $\psi_0(r)$ of the first atomic orbital, the wave function $\Psi_0(\mathbf{r}) \in L^2(\mathbb{R}^3)$ corresponding to the lowest energy (ground state), is a decaying exponential:

$$\psi_0(r) = \frac{1}{\sqrt{\pi} a_0^{3/2}} e^{-r/a_0}, \quad r \geq a_0,$$

where (using Gaussian units) $a_0 = \frac{\hbar^2}{me^2} = 0.0529$ nm is the Bohr radius. Thus, the wave function is strongly localized in space (see Fig. 1, left). Localization of the wave function Ψ_0 expresses the fact that in the hydrogen atom at ground state, the electron is bound to a small region around the nucleus, and the probability of finding the electron at a distance r decreases rapidly as r increases.

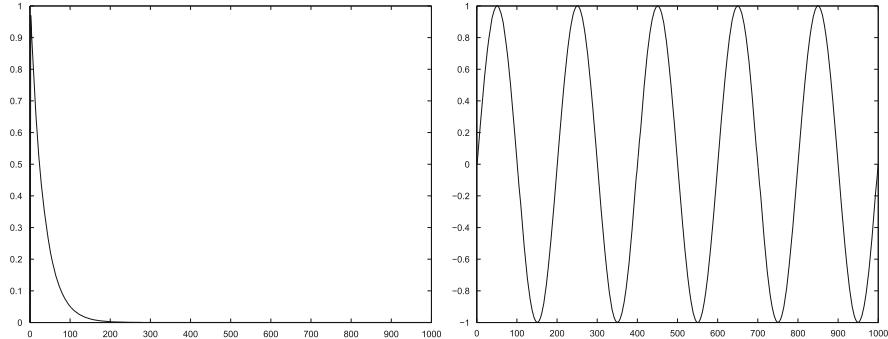


Fig. 1 *Left:* Localized eigenfunction. *Right:* Delocalized eigenfunction

The wave function Ψ_0 satisfies the (stationary) Schrödinger equation:

$$H \Psi_0 = E_0 \Psi_0$$

where the operator H (using now atomic units) is given by

$$H = -\frac{1}{2} \Delta - \frac{1}{r} \quad (\Delta = \text{Laplacian})$$

is the *Hamiltonian*, or energy, operator, and E_0 is the *ground state energy*. That is, the ground state Ψ_0 is the eigenfunction of the Hamiltonian corresponding to the lowest eigenvalue E_0 .

Note that the Hamiltonian is of the form $H = T + V$ where

$$T = -\frac{1}{2} \Delta = \text{kinetic energy}$$

and

$$V = -\frac{1}{r} = \text{(Coulomb) potential.}$$

What happens if the Coulomb potential is absent? In this case there is no force binding the electron to the nucleus: the electron is “free.” This implies *delocalization*: there are no eigenvalues (the spectrum is purely continuous) and therefore no eigenfunctions in $L^2(\mathbb{R}^3)$. Another example is the following. Consider a particle confined to the interval $[0, L]$, then the eigenfunction corresponding to the smallest eigenvalue of the Hamiltonian $H = -\frac{d^2}{dx^2}$ (with zero Dirichlet boundary conditions) is given (up to a normalization factor) by $\Psi_0(x) = \sin\left(\frac{2\pi}{L}x\right)$, which is delocalized (see Fig. 1, right).

Consider now an extended system consisting of a large number of atoms, assumed to be in the ground state. Suppose the system is perturbed at one point

space, for example by slightly changing the value of the potential V near some point \mathbf{x} . If the system is an insulator, then the effect of the perturbation will only be felt locally: it will not be felt outside of a small region. This “absence of diffusion” is also known as localization. Kohn [122, 166] called this behavior the “nearsightedness” of electronic matter. In insulators, and also in semi-conductors and in metallic systems under suitable conditions (such as room temperature), the electrons tend to stay put.

Localization is a phenomenon of major importance in quantum chemistry and in solid state physics. We will return on this in Sect. 4.2, when we discuss applications to the electronic structure problem. Another important example is *Anderson localization*, which refers to the localization in systems described by Hamiltonians of the form $H = T + \gamma V$ where V is a random potential and $\gamma > 0$ a parameter that controls the “disorder strength” in the system [4]. Loosely speaking, once γ exceeds a certain threshold γ_0 the eigenfunctions of H abruptly undergo a transition from extended to localized with very high probability. Anderson localization is beyond the scope of the techniques discussed in these lectures. The interested reader is referred to [183] for a survey.

Locality (or lack thereof) is also of central importance in quantum information theory and quantum computing, in connection with the notion of entanglement of states [76].

1.2 Localization in Numerical Mathematics

In contrast to the situation in physics, the recognition of localization as an important property in numerical mathematics is relatively recent. It began to slowly emerge in the late 1970s and early 1980s as a result of various trends in numerical analysis, particularly in approximation theory (convergence properties of splines) and in numerical linear algebra. Researchers in these areas were the first to investigate the decay properties of inverses and eigenvectors of certain classes of banded matrices; see [67, 68] and [59]. By the late 1990s, the decay behavior of the entries of fairly general functions of banded matrices had been analyzed [20, 115], and numerous papers on the subject have appeared since then. Figures 2, 3 and 4 provide examples of localization for different matrix functions.

From a rather different direction, Banach algebras of infinite matrices with off-diagonal decay arising in computational harmonic analysis and other problems of a numerical nature were being investigated in the 1990s by Jaffard in France [117] and by Baskakov and others [12, 13, 34] in the former Soviet Union. In particular, much effort has been devoted to the study of classes of inverse-closed algebras of infinite matrices with off-diagonal decay.⁴ This is now a well-developed area of

⁴Let $\mathcal{A} \subseteq \mathcal{B}$ be two algebras with common identity. Then \mathcal{A} is said to be *inverse-closed* in \mathcal{B} if $A^{-1} \in \mathcal{A}$ for all $A \in \mathcal{A}$ that are invertible in \mathcal{B} [96].

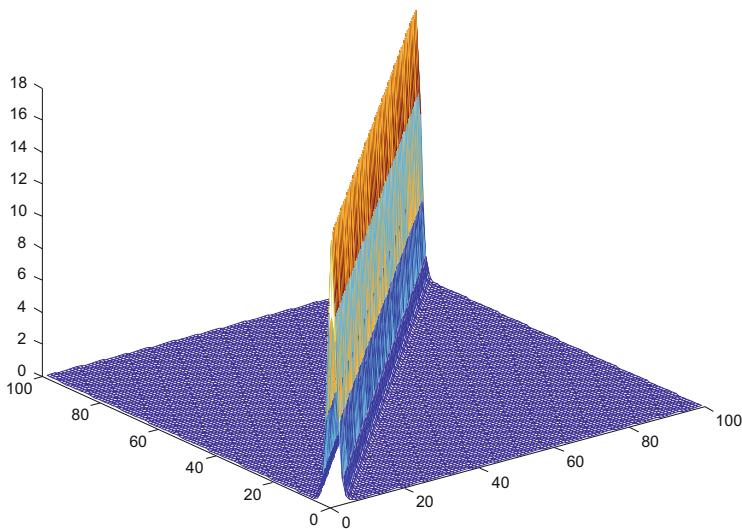


Fig. 2 Plot of $|[e^A]_{ij}|$ for A tridiagonal (discrete 1D Laplacian)

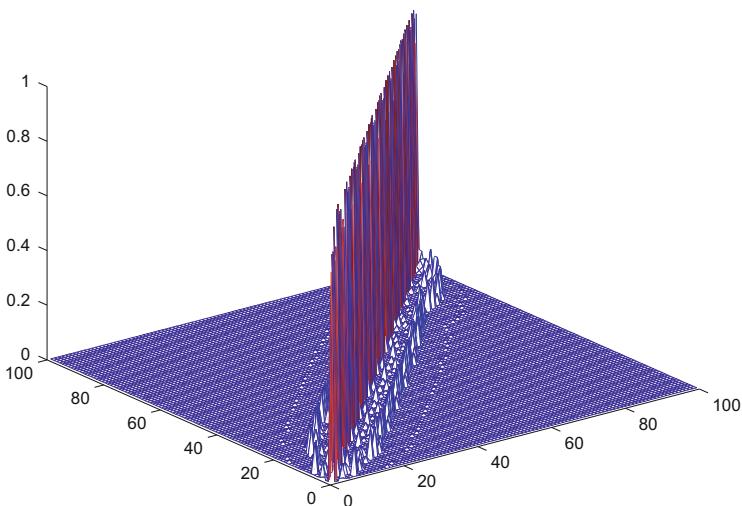


Fig. 3 Plot of $|[A^{1/2}]_{ij}|$ for matrix nos4 from the University of Florida Sparse Matrix Collection [64] (scaled and reordered with reverse Cuthill–McKee)

mathematics; see, e.g., [96, 97, 186] as well as [139, 170]. We will return on this topic in Sect. 3.8.

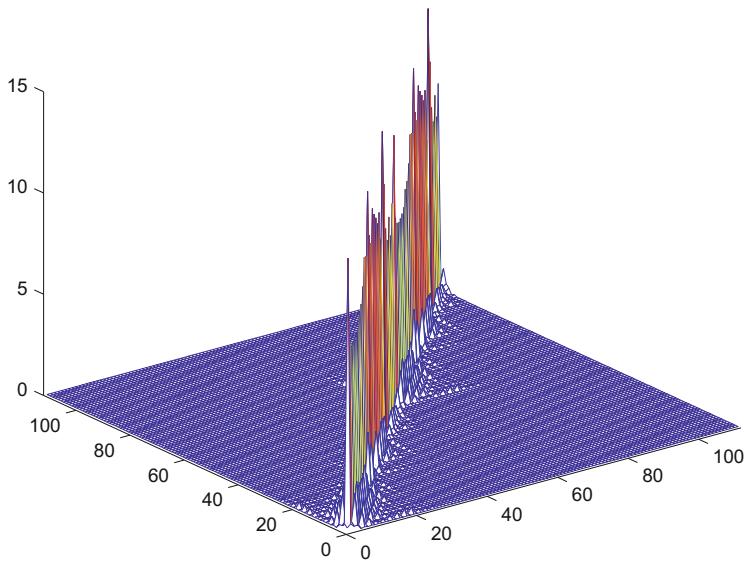


Fig. 4 Plot of $|[\log(A)]_{ij}|$ for matrix `bcsstk03` from the University of Florida Sparse Matrix Collection [64] (scaled and reordered with reverse Cuthill–McKee)

Locality in numerical linear algebra is related to, but should not be confused with, sparsity. A matrix can be localized even if it is a full matrix, although it will be close to a sparse matrix (in some norm).

Perhaps less obviously, a (discrete) system could well be described by a highly sparse matrix but be strongly delocalized. This happens when all the different parts comprising the system are “close together” in some sense. Network science provides striking examples of this: small diameter graphs, and particularly small-world networks, such as Facebook, and other online social networks, are highly sparse but delocalized, in the sense that there is no clear distinction between “short-range” and “long-range” interactions between the components of the system. Even if, on average, each component of such a system is directly connected to only a few other components, the system is strongly delocalized, since every node is only a few steps away from every other node. Hence, a “disturbance” at one node propagates quickly to the entire system. Every short range interaction is also long-range: locality is almost absent in such systems. We shall return to this topic in Sect. 4.3.

Intuitively speaking, localization makes sense (for a system of N parts embedded in some n -dimensional space) when it is possible to let the system size N grow to infinity while keeping the density (number of parts per unit volume) constant. This situation is sometimes referred to as the *thermodynamic limit* (or *bulk limit* in solid state physics). We will provide a more formal discussion of this in a later section of the paper using notions from graph theory.

It is interesting to observe that both localization and delocalization can be advantageous from a computational perspective. Computing approximations to vectors or matrices that are strongly localized can be very efficient in terms of both storage and arithmetic complexity, but computations with systems that are both sparse and delocalized (in the sense just discussed) can also be very efficient, since information propagates very quickly in such systems. As a result, iterative methods based on matrix vector products for solving linear systems, computing eigenvalues and evaluating matrix functions tend to converge very quickly for sparse problems corresponding to small-diameter graphs; see, e.g., [5].

2 Notation and Background in Linear Algebra and Graph Theory

In this chapter we provide the necessary background in linear algebra and graph theory. Excellent general references for linear algebra and matrix analysis are the two volumes by Horn and Johnson [111, 112]. For a thorough treatment of matrix functions, see the monograph by Higham [107]. A good general introduction to graph theory is Diestel [71].

We will be dealing primarily with matrices and vectors with entries in \mathbb{R} or \mathbb{C} . The (i,j) entry of matrix A will be denoted either by a_{ij} or by $[A]_{ij}$. Throughout this chapter, I will denote the identity matrix (or operator); the dimension should be clear from the context.

Recall that a matrix $A \in \mathbb{C}^{n \times n}$ is *Hermitian* if $A^* = A$, *skew-Hermitian* if $A^* = -A$, *unitary* if $A^* = A^{-1}$, *symmetric* if $A^T = A$, *skew-symmetric* if $A^T = -A$, and *orthogonal* if $A^T = A^{-1}$. A matrix A is *diagonalizable* if it is similar to a diagonal matrix: there exist a diagonal matrix D and a nonsingular matrix X such that $A = XDX^{-1}$. The diagonal entries of D are the eigenvalues of A , denoted by λ_i , and they constitute the *spectrum* of A , denoted by $\sigma(A)$. The columns of X are the corresponding eigenvectors. A matrix A is *unitarily diagonalizable* if $A = UDU^*$ with D diagonal and U unitary. The *spectral theorem* states that a necessary and sufficient condition for a matrix A to be unitarily diagonalizable is that A is *normal*: $AA^* = A^*A$. Hermitian, skew-Hermitian and unitary matrices are examples of normal matrices.

Any matrix $A \in \mathbb{C}^{n \times n}$ can be reduced to *Jordan form*. Let $\lambda_1, \dots, \lambda_s \in \mathbb{C}$ be the distinct eigenvalues of A . Then there exists a nonsingular $Z \in \mathbb{C}^{n \times n}$ such that $Z^{-1}AZ = J = \text{diag}(J_1, J_2, \dots, J_s)$, where each diagonal block J_1, J_2, \dots, J_s is block diagonal and has the form $J_i = \text{diag}(J_i^{(1)}, J_i^{(2)}, \dots, J_i^{(g_i)})$, where g_i is the geometric

multiplicity of the λ_i ,

$$J_i^{(j)} = \begin{bmatrix} \lambda_i & 1 & 0 & \dots & 0 \\ 0 & \lambda_i & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_i & 1 \\ 0 & 0 & \dots & 0 & \lambda_i \end{bmatrix} \in \mathbb{C}^{v_i^{(j)} \times v_i^{(j)}},$$

and $\sum_{i=1}^s \sum_{j=1}^{g_i} v_i^{(j)} = n$. The Jordan matrix J is unique up to the ordering of the blocks, but Z is not. The order n_i of the largest Jordan block in which the eigenvalue λ_i appears is called the *index* of λ_i . If the blocks J_i are ordered from largest to smallest, then $\text{index}(\lambda_i) = v_i^{(1)}$. A matrix A is diagonalizable if and only if all the Jordan blocks in J are 1×1 .

From the Jordan decomposition of a matrix $A \in \mathbb{C}^{n \times n}$ we obtain the following “coordinate-free” form of the Jordan decomposition of A :

$$A = \sum_{i=1}^s [\lambda_i G_i + N_i] \quad (1)$$

where $\lambda_1, \dots, \lambda_s$ are the distinct eigenvalues of A , G_i is the projector onto the generalized eigenspace $\text{Ker}((A - \lambda_i I)^{n_i})$ along $\text{Ran}((A - \lambda_i I)^{n_i})$ with $n_i = \text{index}(\lambda_i)$, and $N_i = (A - \lambda_i I)G_i = G_i(A - \lambda_i I)$ is nilpotent of index n_i . The G_i ’s are the *Frobenius covariants* of A .

If A is diagonalizable ($A = XDX^{-1}$) then $N_i = 0$ and the expression above can be written

$$A = \sum_{i=1}^n \lambda_i \mathbf{x}_i \mathbf{y}_i^*$$

where $\lambda_1, \dots, \lambda_n$ are not necessarily distinct eigenvalues, and $\mathbf{x}_i, \mathbf{y}_i$ are right and left eigenvectors of A corresponding to λ_i . Hence, A is a weighted sum of at most n rank-one matrices (oblique projectors).

If A is normal then the spectral theorem yields

$$A = \sum_{i=1}^n \lambda_i \mathbf{u}_i \mathbf{u}_i^*$$

where \mathbf{u}_i is eigenvector corresponding to λ_i . Hence, A is a weighted sum of at most n rank-one orthogonal projectors.

From these expressions one readily obtains for any matrix $A \in \mathbb{C}^{n \times n}$ that

$$\text{Tr}(A) := \sum_{i=1}^n a_{ii} = \sum_{i=1}^n \lambda_i$$

and, more generally,

$$\mathrm{Tr}(A^k) = \sum_{i=1}^n \lambda_i^k, \quad \forall k = 1, 2, \dots$$

Next, we recall the *singular value decomposition* (SVD) of a matrix. For any $A \in \mathbb{C}^{m \times n}$ there exist unitary matrices $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$ and a “diagonal” matrix $\Sigma \in \mathbb{R}^{m \times n}$ such that

$$U^*AV = \Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_p)$$

where $p = \min\{m, n\}$. The σ_i are the *singular values* of A and satisfy (for $A \neq 0$)

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_p = 0,$$

where $r = \mathrm{rank}(A)$. The matrix Σ is uniquely determined by A , but U and V are not. The columns \mathbf{u}_i and \mathbf{v}_i of U and V are left and right singular vectors of A corresponding to the singular value σ_i . From $AA^* = U\Sigma\Sigma^TU^*$ and $A^*A = V\Sigma^T\Sigma V^*$ we deduce that the singular values of A are the (positive) square roots of the eigenvalues of the matrices AA^* and A^*A ; the left singular vectors of A are eigenvectors of AA^* , and the right ones are eigenvectors of A^*A . Moreover,

$$A = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^*,$$

showing that any matrix A of rank r is the sum of exactly r rank-one matrices.

The notion of a *norm* on a vector space (over \mathbb{R} or \mathbb{C}) is well known. A *matrix norm* on the matrix spaces $\mathbb{R}^{n \times n}$ or $\mathbb{C}^{n \times n}$ is just a vector norm $\|\cdot\|$ which satisfies the additional requirement of being *submultiplicative*:

$$\|AB\| \leq \|A\| \|B\|, \quad \forall A, B.$$

Important examples of matrix norms include the *Frobenius norm*

$$\|A\|_F := \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2}$$

as well as the norms

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|, \quad \|A\|_\infty = \|A^*\|_1 = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$$

and the *spectral norm* $\|A\|_2 = \sigma_1$. Note that $\|A\|_F = \sqrt{\sum_{i=1}^n \sigma_i^2}$ and therefore $\|A\|_2 \leq \|A\|_F$ for all A . The inequality

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty} \quad (2)$$

is often useful; note that for $A = A^*$ it implies that $\|A\|_2 \leq \|A\|_1 = \|A\|_\infty$.

The *spectral radius* $\varrho(A) := \max\{|\lambda| : \lambda \in \sigma(A)\}$ satisfies $\varrho(A) \leq \|A\|$ for all A and all matrix norms. For a normal matrix, $\varrho(A) = \|A\|_2$. But if A is nonnormal, $\|A\|_2 - \varrho(A)$ can be arbitrarily large. Also note that if A is diagonalizable with $A = XDX^{-1}$, then

$$\|A\|_2 = \|XDX^{-1}\|_2 \leq \|X\|_2 \|X^{-1}\|_2 \|D\|_2 = \kappa_2(X) \varrho(A),$$

where $\kappa_2(X) = \|X\|_2 \|X^{-1}\|_2$ is defined as the infimum of the spectral condition numbers of X taken over the set of *all* matrices X which diagonalize A .

Clearly, the spectrum $\sigma(A)$ is entirely contained in the closed disk in the complex plane centered at the origin with radius $\varrho(A)$. Much effort has been devoted to finding better “inclusion regions,” i.e., subsets of \mathbb{C} containing all the eigenvalues of a given matrix. We review some of these next.

Let $A \in \mathbb{C}^{n \times n}$. For all $i = 1, \dots, n$, let

$$r_i := \sum_{j \neq i} |a_{ij}|, \quad D_i = D_i(a_{ii}, r_i) := \{z \in \mathbb{C} : |z - a_{ii}| \leq r_i\}.$$

The set D_i is called the *i*th Geršgorin disk of A . *Geršgorin’s Theorem* (1931) states that $\sigma(A) \subset \cup_{i=1}^n D_i$. Moreover, each connected component of $\cup_{i=1}^n D_i$ consisting of p Geršgorin disks contains exactly p eigenvalues of A , counted with their multiplicities. Of course, the same result holds replacing the off-diagonal row-sums with off-diagonal column-sums. The spectrum is then contained in the intersection of the two resulting regions.

Also of great importance is the *field of values* (or *numerical range*) of $A \in \mathbb{C}^{n \times n}$, defined as the set

$$\mathcal{W}(A) := \{z = \langle Ax, x \rangle : x^*x = 1\}. \quad (3)$$

This set is a compact subset of \mathbb{C} containing the eigenvalues of A ; it is also convex. This last statement is known as the *Hausdorff–Toeplitz Theorem*, and is highly nontrivial. If A is normal, the field of values is the convex hull of the eigenvalues; the converse is true if $n \leq 4$, but not in general. The eigenvalues and the field of values of a random 10×10 matrix are shown in Fig. 5.

For a matrix $A \in \mathbb{C}^{n \times n}$, let

$$H_1 = \frac{1}{2}(A + A^*), \quad H_2 = \frac{1}{2i}(A - A^*). \quad (4)$$

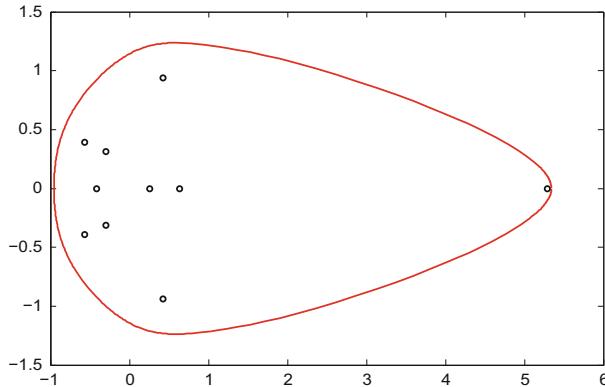


Fig. 5 Eigenvalues and field of values of a random 10×10 matrix

Note that H_1, H_2 are both Hermitian. Let $a = \min \lambda(H_1)$, $b = \max \lambda(H_1)$, $c = \min \lambda(H_2)$, and $d = \max \lambda(H_2)$. Then for every eigenvalue $\lambda(A)$ of A we have that

$$a \leq \Re(\lambda(A)) \leq b, \quad c \leq \Im(\lambda(A)) \leq d.$$

This is sometimes referred to as the *Bendixson–Hirsch Theorem*; see, e.g., [14, p. 224]. Moreover, the field of values of A is entirely contained in the rectangle $[a, b] \times [c, d]$ in the complex plane [111, p. 9]. Note that if $A \in \mathbb{R}^{n \times n}$, then $c = -d$.

The definition of field of values (3) also applies to *bounded linear operators* on a Hilbert space \mathcal{H} ; however, $\mathcal{W}(A)$ may not be closed if $\dim(\mathcal{H}) = \infty$.

For a matrix $A \in \mathbb{C}^{n \times n}$ and a scalar polynomial

$$p(\lambda) = c_0 + c_1\lambda + c_2\lambda^2 + \cdots + c_k\lambda^k,$$

define

$$p(A) = c_0I + c_1A + c_2A^2 + \cdots + c_kA^k.$$

Let $A = ZJZ^{-1}$ where J is the Jordan form of A . Then $p(A) = Zp(J)Z^{-1}$. Hence, the eigenvalues of $p(A)$ are given by $p(\lambda_i)$, for $i = 1, \dots, n$. In particular, if A is diagonalizable with $A = XDX^{-1}$ then $p(A) = Xp(D)X^{-1}$. Hence, A and $p(A)$ have the same eigenvectors.

The *Cayley–Hamilton Theorem* states that for any matrix $A \in \mathbb{C}^{n \times n}$ it holds that $p_A(A) = 0$, where $p_A(\lambda) := \det(A - \lambda I)$ is the characteristic polynomial of A . Perhaps an even more important polynomial is the *minimum polynomial* of A , which is defined as the monic polynomial $q_A(\lambda)$ of least degree such that $q_A(A) = 0$. Note that $q_A|p_A$, hence $\deg(q_A) \leq \deg(p_A) = n$. It easily follows from this that for any nonsingular $A \in \mathbb{C}^{n \times n}$, the inverse A^{-1} can be expressed as a polynomial in A of

degree at most $n - 1$:

$$A^{-1} = c_0I + c_1A + c_2A^2 + \cdots + c_kA^k, \quad k \leq n - 1.$$

Note, however, that the coefficients c_i depend on A . It also follows that powers A^p with $p \geq n$ can be expressed as linear combinations of powers A^k with $0 \leq k \leq n - 1$. The same result holds more generally for matrix functions $f(A)$ that can be represented as power series in A (see below).

Indeed, let $\lambda_1, \dots, \lambda_s$ be the distinct eigenvalues of $A \in \mathbb{C}^{n \times n}$ and let n_i be the index of λ_i . If f is a given function, we *define* the matrix function

$$f(A) := r(A),$$

where r is the unique Lagrange–Hermite interpolating polynomial of degree $< \sum_{i=1}^s n_i$ satisfying

$$r^{(j)}(\lambda_i) = f^{(j)}(\lambda_i) \quad j = 0, \dots, n_i - 1, \quad i = 1, \dots, s.$$

Here $f^{(j)}$ denotes the j th derivative of f , with $f^{(0)} \equiv f$. Note that for the definition to make sense we must require that the values $f^{(j)}(\lambda_i)$ with $0 \leq j \leq n_i - 1$ and $1 \leq i \leq s$ exist. We say that f is *defined on the spectrum of A* . When all the eigenvalues are distinct, the interpolation polynomial has degree $n - 1$. In this case, the minimum polynomial and the characteristic polynomial of A coincide.

There are several other ways to define $f(A)$, all equivalent to the definition just given [107]. One such definition is through the Jordan canonical form. Let $A \in \mathbb{C}^{n \times n}$ have Jordan form $Z^{-1}AZ = J$ with $J = \text{diag}(J_1, \dots, J_s)$. We define

$$f(A) := Zf(J)Z^{-1} = Z \text{diag}(f(J_1), f(J_2), \dots, f(J_s)) Z^{-1},$$

where $f(J_i) = \text{diag}(f(J_i^{(1)}), f(J_i^{(2)}), \dots, f(J_i^{(g_i)}))$ and

$$f(J_i^{(j)}) = \begin{bmatrix} f(\lambda_i) & f'(\lambda_i) & \cdots & \frac{f^{(v_i^{(j)}-1)}(\lambda_i)}{(v_i^{(j)}-1)!} \\ & f(\lambda_i) & \ddots & \vdots \\ & & \ddots & f'(\lambda_i) \\ & & & f(\lambda_i) \end{bmatrix}.$$

An equivalent expression is the following:

$$f(A) = \sum_{i=1}^s \sum_{j=0}^{n_i-1} \frac{f^{(j)}(\lambda_i)}{j!} (A - \lambda_i I)^j G_i,$$

where $n_i = \text{index}(\lambda_i)$ and G_i is the Frobenius covariant associated with λ_i (see (1)). The usefulness of this definition is primarily theoretical, given the difficulty of determining the Jordan structure of a matrix numerically. If $A = XDX^{-1}$ with D diagonal, then $f(A) := Xf(D)X^{-1} = X\text{diag}(f(\lambda_i))X^{-1}$. Denoting with \mathbf{x}_i the i th column of X and with \mathbf{y}_i^T the i th row of X^{-1} we obtain the expression

$$f(A) = \sum_{i=1}^n f(\lambda_i) \mathbf{x}_i \mathbf{y}_i^T.$$

If in addition $A = UDU^*$ is normal then

$$f(A) = \sum_{i=1}^n f(\lambda_i) \mathbf{u}_i \mathbf{u}_i^*.$$

If f is analytic in a domain $\Omega \subseteq \mathbb{C}$ containing the spectrum of $A \in \mathbb{C}^{n \times n}$, then

$$f(A) = \frac{1}{2\pi i} \int_{\Gamma} f(z)(zI - A)^{-1} dz, \quad (5)$$

where $i = \sqrt{-1}$ is the imaginary unit and Γ is any simple closed curve surrounding the eigenvalues of A and entirely contained in Ω , oriented counterclockwise. This definition has the advantage of being easily generalized to functions of bounded operators on Banach spaces, and it is also the basis for some of the currently most efficient computational methods for the evaluation of matrix functions.

Another widely used definition of $f(A)$ when f is analytic is through power series. Suppose f has a Taylor series expansion

$$f(z) = \sum_{k=0}^{\infty} a_k (z - z_0)^k \quad \left(a_k = \frac{f^{(k)}(z_0)}{k!} \right)$$

with radius of convergence R . If $A \in \mathbb{C}^{n \times n}$ and each of the distinct eigenvalues $\lambda_1, \dots, \lambda_s$ of A satisfies

$$|\lambda_i - z_0| < R,$$

then

$$f(A) := \sum_{k=0}^{\infty} a_k (A - z_0 I)^k.$$

If $A \in \mathbb{C}^{n \times n}$ and f is defined on $\sigma(A)$, the following facts hold (see [107]):

- (i) $f(A)A = Af(A);$
- (ii) $f(A^T) = f(A)^T;$

- (iii) $f(XAX^{-1}) = Xf(A)X^{-1}$;
- (iv) $\sigma(f(A)) = f(\sigma(A))$;
- (v) (λ, x) eigenpair of $A \Rightarrow (f(\lambda), x)$ eigenpair of $f(A)$;
- (vi) A is block triangular $\Rightarrow F = f(A)$ is block triangular with the same block structure as A , and $F_{ii} = f(A_{ii})$ where A_{ii} is the i th diagonal block of A ;
- (vii) In particular, $f(\text{diag}(A_{11}, \dots, A_{pp})) = \text{diag}(f(A_{11}), \dots, f(A_{pp}))$;
- (viii) $f(I_m \otimes A) = I_m \otimes f(A)$, where \otimes is the Kronecker product;
- (ix) $f(A \otimes I_m) = f(A) \otimes I_m$.

Another useful result is the following:

Theorem 1 ([108]) *Let f be analytic on an open set $\Omega \subseteq \mathbb{C}$ such that each connected component of Ω is closed under conjugation. Consider the corresponding matrix function f on the set $\mathcal{D} = \{A \in \mathbb{C}^{n \times n} : \sigma(A) \subseteq \Omega\}$. Then the following are equivalent:*

- (a) $f(A^*) = f(A)^*$ for all $A \in \mathcal{D}$.
- (b) $f(\bar{A}) = \bar{f}(A)$ for all $A \in \mathcal{D}$.
- (c) $f(\mathbb{R}^{n \times n} \cap \mathcal{D}) \subseteq \mathbb{R}^{n \times n}$.
- (d) $f(\mathbb{R} \cap \Omega) \subseteq \mathbb{R}$.

In particular, if $f(x) \in \mathbb{R}$ for $x \in \mathbb{R}$ and A is Hermitian, so is $f(A)$.

Important examples of matrix functions are the resolvent and the matrix exponential. Let $A \in \mathbb{C}^{n \times n}$, and let $z \notin \sigma(A)$. The *resolvent* of A at z is defined as

$$R(A; z) = (zI - A)^{-1}.$$

The resolvent is central to the definition of matrix functions via the contour integral approach (5). The resolvent also plays a fundamental role in spectral theory. For example, it can be used to define the spectral projector onto the eigenspace of a matrix or operator corresponding to an isolated eigenvalue $\lambda_0 \in \sigma(A)$:

$$P_{\lambda_0} := \frac{1}{2\pi i} \int_{|z-\lambda_0|=\varepsilon} (zI - A)^{-1} dz,$$

where $\varepsilon > 0$ is small enough so that no other eigenvalue of A falls within ε of λ_0 . It can be shown that $P_{\lambda_0}^2 = P_{\lambda_0}$ and that the range of P_{λ_0} is the one-dimensional subspace spanned by the eigenvector associated with λ_0 . More generally, one can define the spectral projector onto the invariant subspace of A corresponding to a set of selected eigenvalues by integrating $R(A; z)$ along a contour surrounding those eigenvalues and excluding the others. It should be noted that the spectral projector is an orthogonal projector ($P = P^*$) if and only if A is normal. If A is diagonalizable, a spectral projector P is a simple function of A : if f is any function taking the value 1 at the eigenvalues of interest and 0 on the remaining ones, then $P = f(A)$.

The *matrix exponential* can be defined via the Maclaurin expansion

$$e^A = I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots = \sum_{k=0}^{\infty} \frac{1}{k!}A^k,$$

which converges for arbitrary $A \in \mathbb{C}^{n \times n}$. Just as the resolvent is central to spectral theory, the matrix exponential is fundamental to the solution of differential equations. For example, the solution to the inhomogeneous system

$$\frac{dy}{dt} = Ay + f(t, y), \quad y(0) = y_0, \quad y \in \mathbb{C}^n, \quad A \in \mathbb{C}^{n \times n}$$

is given (implicitly!) by

$$y(t) = e^{tA}y_0 + \int_0^t e^{A(t-s)}f(s, y(s))ds.$$

In particular, $y(t) = e^{tA}y_0$ when $f = 0$. It is worth recalling that $\lim_{t \rightarrow \infty} e^{tA} = 0$ if and only if A is a *stable* matrix: $\Re(\lambda) < 0$ for all $\lambda \in \sigma(A)$.

When $f(t, y) = \mathbf{b} \in \mathbb{C}^n$ (=const.), the solution can also be expressed as

$$y(t) = t\psi_1(tA)(\mathbf{b} + Ay_0) + y_0,$$

where

$$\psi_1(z) = \frac{e^z - 1}{z} = 1 + \frac{z}{2!} + \frac{z^2}{3!} + \cdots$$

The matrix exponential plays an especially important role in quantum theory. Consider for instance the time-dependent Schrödinger equation:

$$i\frac{\partial\Psi}{\partial t} = H\Psi, \quad t \in \mathbb{R}, \quad \Psi(0) = \Psi_0, \tag{6}$$

where $\Psi_0 \in L^2$ is a prescribed initial state with $\|\Psi_0\|_2 = 1$. Here $H = H^*$ is the Hamiltonian, or energy operator (which we assume to be time-independent). The solution of (6) is given explicitly by $\Psi(t) = e^{-itH}\Psi_0$, for all $t \in \mathbb{R}$; note that since itH is skew-Hermitian, the *propagator* $U(t) = e^{-itH}$ is unitary, which guarantees that the solution has unit norm for all t :

$$\|\Psi(t)\|_2 = \|U(t)\Psi_0\|_2 = \|\Psi_0\|_2 = 1, \quad \forall t \in \mathbb{R}.$$

Also very important in many-body quantum mechanics is the *Fermi–Dirac operator*, defined as

$$f(H) := (I + \exp(\beta(H - \mu I)))^{-1},$$

where $\beta = (\kappa_B T)^{-1}$ is the *inverse temperature*, κ_B the Boltzmann constant, and μ is the *Fermi level*, separating the eigenvalues of H corresponding to the first n_e eigenvectors from the rest, where n_e is the number of particles (electrons) comprising the system under study. This matrix function will be discussed in Sect. 4.2.

Finally, in statistical quantum mechanics the state of a system is completely described (statistically) by the *density operator*:

$$\rho := \frac{e^{-\beta H}}{Z}, \quad \text{where } Z = \text{Tr}(e^{-\beta H}).$$

The quantity $Z = Z(\beta)$ is known as the *partition function* of the system.

Trigonometric functions and square roots of matrices are also important in applications to differential equations. For example, the solution to the second-order system

$$\frac{d^2\mathbf{y}}{dt^2} + A\mathbf{y} = 0, \quad \mathbf{y}(0) = \mathbf{y}_0, \quad \mathbf{y}'(0) = \mathbf{y}'_0$$

(where A is SPD) can be expressed as

$$\mathbf{y}(t) = \cos(\sqrt{A}t)\mathbf{y}_0 + (\sqrt{A})^{-1}\sin(\sqrt{A}t)\mathbf{y}'_0.$$

Apart from the contour integration formula, the matrix exponential and the resolvent are also related through the *Laplace transform*: there exists an $\omega \in \mathbb{R}$ such that $z \notin \sigma(A)$ for $\Re(z) > \omega$ and

$$(zI - A)^{-1} = \int_0^\infty e^{-zt} e^{tA} dt = \int_0^\infty e^{-t(zI - A)} dt.$$

Also note that if $|z| > \varrho(A)$, the following *Neumann series expansion* of the resolvent is valid:

$$(zI - A)^{-1} = z^{-1}(I + z^{-1}A + z^{-2}A^2 + \dots) = z^{-1} \sum_{k=0}^{\infty} z^{-k}A^k.$$

Next, we recall a few definitions and notations associated with graphs. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with $n = |\mathcal{V}|$ nodes (or vertices) and $m = |\mathcal{E}|$ edges (or links). The elements of \mathcal{V} will be denoted simply by $1, \dots, n$. If for all $i, j \in \mathcal{V}$ such that $(i, j) \in \mathcal{E}$ then also $(j, i) \in \mathcal{E}$, the graph is said to be *undirected*. On the other hand, if this condition does not hold, namely if there exists $(i, j) \in \mathcal{E}$ such that $(j, i) \notin \mathcal{E}$, then the network is said to be *directed*. A directed graph is commonly referred to as a *digraph*. If $(i, j) \in \mathcal{E}$ in a digraph, we will write $i \rightarrow j$. A graph is *simple* if it is unweighted, contains no *loops* (edges of the form (i, i)) and there are no multiple

edges with the same orientation between any two nodes. A simple graph can be represented by means of its *adjacency matrix* $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, where

$$a_{ij} = \begin{cases} 1, & \text{if } (i, j) \in \mathcal{E}, \\ 0, & \text{else.} \end{cases}$$

Note that $A = A^T$ if, and only if, \mathcal{G} is undirected. If the graph is weighted, then a_{ij} will be equal to the weight of the corresponding edge (i, j) .

If \mathcal{G} is undirected, the *degree* $\deg(i)$ of node i is the number of edges incident to i in \mathcal{G} . That is, $\deg(i)$ is the number of “immediate neighbors” of i in \mathcal{G} . Note that in terms of the adjacency matrix, $\deg(i) = \sum_{j=1}^n a_{ij}$. A *d-regular graph* is a graph where every node has the same degree d .

For an undirected graph we also define the *graph Laplacian* as the matrix

$$L := D - A, \quad \text{where } D := \text{diag}(\deg(1), \deg(2), \dots, \deg(n))$$

and, assuming $\deg(i) \neq 0$ for all i , the *normalized Laplacian*

$$\hat{L} := I - D^{-1/2}AD^{-1/2}.$$

Both of these matrices play an important role in the structural analysis of networks and in the study of diffusion-type processes on graphs, and matrix exponentials of the form e^{-tL} and $e^{-t\hat{L}}$, where $t > 0$ denotes time, are widely used in applications. Note that L and \hat{L} are both symmetric positive semidefinite matrices. Moreover, if \mathcal{G} is a d -regular graph, then the eigenvalues of L are given by $d - \lambda_i(A)$ (where $\lambda_i(A)$ are the eigenvalues of the adjacency matrix A of \mathcal{G}) and L and A have the same eigenvectors. For more general graphs, however, there is no simple relationship between the spectra of L and A .

A *walk* of length k in \mathcal{G} is a set of nodes $\{i_1, i_2, \dots, i_k, i_{k+1}\}$ such that for all $1 \leq j \leq k$, there is an edge between i_j and i_{j+1} (a directed edge $i_j \rightarrow i_{j+1}$ for a digraph). A *closed walk* is a walk where $i_1 = i_{k+1}$. A *path* is a walk with no repeated nodes.

There is a close connection between the walks in \mathcal{G} and the entries of the powers of the adjacency matrix A . Indeed, let $k \geq 1$. For any simple graph \mathcal{G} , the following holds:

$$\begin{aligned} [A^k]_{ii} &= \text{number of closed walks of length } k \text{ starting and ending at node } i; \\ [A^k]_{ij} &= \text{number of walks of length } k \text{ starting at node } i \text{ and ending at node } j. \end{aligned}$$

Let now i and j be any two nodes in \mathcal{G} . In many situations in network science it is desirable to have a measure of how “well connected” nodes i and j are. Estrada and Hatano [79] have proposed to quantify the strength of connection between nodes in terms of the number of walks joining i and j , assigning more weight to shorter walks (i.e., penalizing longer ones). If walks of length k are downweighted by a factor $\frac{1}{k!}$, this leads [79] to the following definition of *communicability* between node i and

node j :

$$C(i,j) := [\mathbf{e}^A]_{ij} = \sum_{k=0}^{\infty} \frac{[A^k]_{ij}}{k!}, \quad (7)$$

where by convention we assign the value 1 to the number of “walks of length 0.” Of course, other matrix functions can also be used to define the communicability between nodes [80], but the matrix exponential has a natural physical interpretation (see [81]).

The *geodesic distance* $d(i,j)$ between two nodes i and j is the length of the shortest path connecting i and j . We let $d(i,j) = \infty$ if no such path exists. We note that $d(\cdot, \cdot)$ is a true distance function (i.e., a *metric* on \mathcal{G}) if the graph is undirected, but not in general, since only in an undirected graph the condition $d(i,j) = d(j,i)$ is satisfied for all $i \in \mathcal{V}$.

The *diameter* of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as

$$\text{diam}(\mathcal{G}) := \max_{i,j \in \mathcal{V}} d(i,j).$$

A digraph \mathcal{G} is *strongly connected* (or, in short, *connected*) if for every pair of nodes i and j there is a path in \mathcal{G} that starts at i and ends at j ; i.e., $\text{diam}(\mathcal{G}) < \infty$. We say that \mathcal{G} is *weakly connected* if it is connected as an undirected graph (i.e., when the orientation of the edges is disregarded). Clearly, for an undirected graph the two notions coincide. It can be shown that for an undirected graph the number of connected components is equal to the dimension of $\text{Ker}(L)$, the null space of the graph Laplacian.

Just as we have associated matrices to graphs, graphs can also be associated to matrices. In particular, to any matrix $A \in \mathbb{C}^{n \times n}$ we can associate a digraph $\mathcal{G}(A) = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, 2, \dots, n\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where $(i,j) \in \mathcal{E}$ if and only if $a_{ij} \neq 0$. Diagonal entries in A are usually ignored, so that there are no loops in $\mathcal{G}(A)$. We also note that for *structurally symmetric* matrices ($a_{ij} \neq 0 \Leftrightarrow a_{ji} \neq 0$) the associated graph $\mathcal{G}(A)$ is undirected.

Let $|A| := [|a_{ij}|]$, then the digraph $\mathcal{G}(|A|^2)$ is given by $(\mathcal{V}, \hat{\mathcal{E}})$ where $\hat{\mathcal{E}}$ is obtained by including all directed edges (i,k) such that there exists $j \in V$ with $(i,j) \in \mathcal{E}$ and $(j,k) \in \mathcal{E}$. (The reason for the absolute value is to disregard the effect of possible cancellations in A^2 .) For higher powers ℓ , the digraph $\mathcal{G}(|A|^\ell)$ is defined similarly: its edge set consists of all pairs (i,k) such that there is a directed path of length at most ℓ joining node i with node k in $\mathcal{G}(A)$.

Thus, for any square matrix A it is possible to predict the *structural nonzero pattern* of the powers A^ℓ for $\ell = 2, 3, \dots$ from the connectivity of the graphs $\mathcal{G}(|A|^\ell)$. One of the first observations that can be made is that powers of narrow-banded matrices (corresponding to graphs with large diameter, for example paths) take large values of ℓ to fill, whereas the opposite happens with matrices that correspond to small-diameter graphs. Figures 6 and 7 illustrate this fact by displaying the graph Laplacian L and the fifth power of L for two highly sparse undirected graphs,

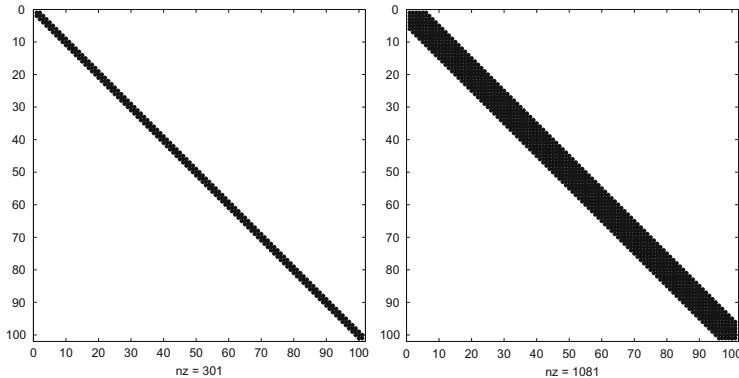


Fig. 6 Path graph. *Left:* nonzero pattern of Laplacian matrix L . *Right:* pattern of fifth power of L

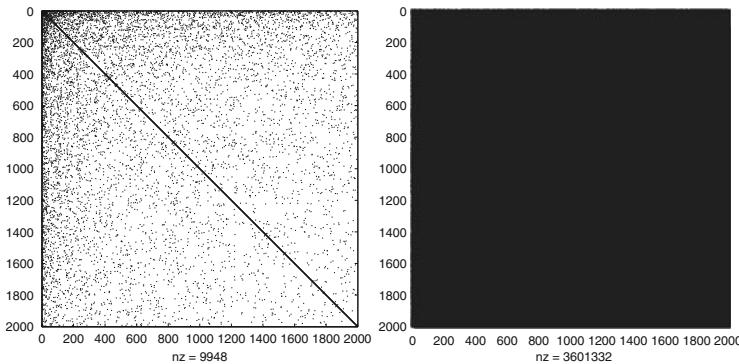


Fig. 7 Scale-free graph. *Left:* nonzero pattern of Laplacian matrix L . *Right:* pattern of fifth power of L

a path graph with $n = 100$ nodes and a scale-free graph on $n = 2000$ nodes built according to preferential attachment scheme (see, e.g., [78]). Graphs of this type are examples of *small-world graphs*, in particular they can be expected to have small diameter. It can be seen that in the case of the scale-free graph the fifth power of the Laplacian, L^5 , is almost completely full (the number of nonzeros is 3,601,332 out of a possible 4,000,000), implying that in this graph most pairs of nodes are less than five degrees of separation away from one another.

The *transitive closure* of \mathcal{G} is the graph $\bar{\mathcal{G}} = (\mathcal{V}, \bar{\mathcal{E}})$ where $(i, j) \in \bar{\mathcal{E}}$ if and only if there is a directed path from i to j in $\mathcal{G}(A)$. A matrix $A \in \mathbb{C}^{n \times n}$ is *reducible* if there exists a permutation matrix P such that

$$P^T AP = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

with A_{11} and A_{22} square submatrices. If no such P exists, A is said to be *irreducible*. Denote by K_n the *complete graph* on n nodes, i.e., the graph where every edge (i,j) is present (with $i \neq j$). The following statements are equivalent:

- (i) the matrix A is irreducible;
- (ii) the digraph $\mathcal{G}(A)$ is strongly connected;
- (iii) the transitive closure $\tilde{\mathcal{G}}(A)$ of $\mathcal{G}(A)$ is K_n .

Note that (iii) and the Cayley–Hamilton Theorem imply that the powers $(I + |A|)^k$ are completely full for $k \geq n - 1$. This has important implications for matrix functions, since it implies that for an irreducible matrix A a matrix function of the form

$$f(A) = \sum_{k=0}^{\infty} a_k (A - z_0 I)^k$$

is completely full, if no cancellation occurs and $a_k \neq 0$ for sufficiently many k . This is precisely formulated in the following result.

Theorem 2 ([21]) *Let f be an analytic function of the form*

$$f(z) = \sum_{k=0}^{\infty} a_k (z - z_0)^k \quad \left(a_k = \frac{f^{(k)}(z_0)}{k!} \right),$$

where $z_0 \in \mathbb{C}$ and the power series expansion has radius of convergence $R > 0$. Let A have an irreducible sparsity pattern and let l ($1 \leq l \leq n - 1$) be the diameter of $\mathcal{G}(A)$. Assume further that there exists $k \geq l$ such that $f^{(k)}(z_0) \neq 0$. Then it is possible to assign values to the nonzero entries of A in such a way that $f(A)$ is defined and $[f(A)]_{ij} \neq 0$ for all $i \neq j$.

This result applies, in particular, to banded A and to such functions as the inverse (resolvent) and the matrix exponential.

3 Localization in Matrix Functions

We have just seen that if A is irreducible and f is a “generic” analytic function defined on the spectrum of A then we should expect $f(A)$ to be completely full (barring fortuitous cancellation). For A large, this seems to make the explicit computation of $f(A)$ impossible, and this is certainly the case if all entries of $f(A)$ need to be accurately approximated.

As we have already mentioned in the Introduction, however, numerical experiments show that when A is a banded matrix and $f(z)$ is a smooth function for which $f(A)$ is defined, the entries of $f(A)$ often decay rapidly as one moves away from the diagonal. The same property is often (but not always!) satisfied by more general

sparse matrices: in this case the decay is away from the support (nonzero pattern) of A . In other words, nonnegligible entries of $f(A)$ tend to be concentrated near the positions (i, j) for which $a_{ij} \neq 0$.

This observation opens up the possibility of approximating functions of sparse matrices, by neglecting “sufficiently small” matrix elements in $f(A)$. Depending on the rate of decay and on the accuracy requirements, it may be possible to develop approximation algorithms that exhibit optimal computational complexity, i.e., $O(n)$ (or *linear scaling*) methods.

In this section we review our current knowledge on localization in functions of large and sparse matrices. In particular, we consider the following questions:

1. Under which conditions can we expect decay in $f(A)$?
2. Can we obtain sharp bounds on the entries of $f(A)$?
3. Can we characterize the rate of decay in $f(A)$ in terms of
 - the bandwidth/sparsity of A ?
 - the spectral properties of A ?
 - the location of singularities of $f(z)$ in relation to the spectrum of A ?
4. What if $f(z)$ is an entire⁵ function?
5. When is the rate of decay independent of the matrix size n ?

The last point is especially crucial if we want to develop $O(n)$ algorithms for approximating functions of sparse matrices.

3.1 Matrices with Decay

A matrix $A \in \mathbb{C}^{n \times n}$ is said to have the *off-diagonal decay property* if its entries $[A]_{ij}$ satisfy a bound of the form

$$|[A]_{ij}| \leq K\phi(|i - j|), \quad \forall i, j, \tag{8}$$

where $K > 0$ is a constant and ϕ is a function defined and positive for $x \geq 0$ and such that $\phi(x) \rightarrow 0$ as $x \rightarrow \infty$. Important examples of decay include exponential decay, corresponding to $\phi(x) = e^{-\alpha x}$ for some $\alpha > 0$, and algebraic (or power-law) decay, corresponding to $\phi(x) = (1 + |i - j|^p)^{-1}$ for some $p \geq 1$.

As it stands, however, this definition is meaningless, since for any fixed matrix $A \in \mathbb{C}^{n \times n}$ the bound can always be achieved with an arbitrary choice of ϕ just by taking K sufficiently large. To give a meaningful definition we need to consider either infinite matrices (for example, bounded linear operators on some sequence

⁵Recall that an *entire function* is a function of a complex variable that is analytic everywhere on the complex plane.

space ℓ^p), or sequences of matrices of increasing dimension. The latter situation being the more familiar one in numerical analysis, we give the following definition.

Definition 1 Let $\{A_n\}$ be a sequence of $n \times n$ matrices with entries in \mathbb{C} , where $n \rightarrow \infty$. We say that the matrix sequence $\{A_n\}$ has the off-diagonal decay property if

$$|[A_n]_{ij}| \leq K\phi(|i-j|), \quad \forall i, j = 1, \dots, n, \quad (9)$$

where the constant $K > 0$ and the function $\phi(x)$, defined for $x \geq 0$ and such that $\phi(x) \rightarrow 0$ as $x \rightarrow \infty$, do not depend on n .

Note that if A is an infinite matrix that satisfies (8) then its finite $n \times n$ sections (leading principal submatrices, see [139]) A_n form a matrix sequence that satisfies Definition 1. The definition can also be extended to block matrices in a natural way.

When dealing with non-Hermitian matrices, it is sometimes required to allow for different decay rates on either side of the main diagonal. For instance, one could have exponential decay on either side but with different rates:

$$|[A_n]_{ij}| \leq K_1 e^{-\alpha(i-j)} \quad \text{for } i > j,$$

and

$$|[A_n]_{ij}| \leq K_2 e^{-\beta(j-i)} \quad \text{for } j > i.$$

Here K_1, K_2 and α, β are all positive constants. It is also possible to have matrices where decay is present on only one side of the main diagonal (see [21, Theorem 3.5]). For simplicity, in the rest of the paper we will primarily focus on the case where the decay bound has the same form for $i > j$ and for $j > i$. However, most of the results can be extended easily to the more general case.

Also, in multidimensional problems it is important to be able to describe decay behavior not just away from the main diagonal but with a more complicated pattern. To this end, we can use any distance function (metric) d (with $d(i,j) = d(j,i)$ for simplicity) with the property that

$$\forall \varepsilon > 0 \exists c = c(\varepsilon) \text{ such that } \sup_j \sum_i e^{-\varepsilon d(i,j)} \leq c(\varepsilon), \quad (10)$$

see [117]. Again, condition (10) is trivially satisfied for any distance function on a finite set $S = \{1, 2, \dots, n\}$, but here we allow infinite ($S = \mathbb{N}$) or bi-infinite matrices ($S = \mathbb{Z}$). In practice, we will consider sequences of matrices of increasing size n and we will define for each n a distance d_n on the set $S = \{1, 2, \dots, n\}$ and assume that each d_n satisfies condition (10) with respect to a constant $c = c(\varepsilon)$ independent of n .

We will be mostly concerned with decay away from a sparsity pattern. For banded sparsity patterns, this is just off-diagonal decay. For more general sparsity

patterns, we assume that we are given a sequence of sparse graphs $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ with $|\mathcal{V}_n| = n$ and $|\mathcal{E}_n| = O(n)$ and a distance function d_n satisfying (10) uniformly with respect to n . In practice we will take d_n to be the geodesic distance on \mathcal{G}_n and we will impose the following *bounded maximum degree* condition:

$$\sup_n \{\deg(i) \mid i \in \mathcal{G}_n\} < \infty. \quad (11)$$

This condition guarantees that the distance $d_n(i, j)$ grows unboundedly as $|i - j|$ does, at a rate independent of n for $n \rightarrow \infty$. In particular, we have that $\lim_{n \rightarrow \infty} \text{diam}(\mathcal{G}_n) = \infty$. This is necessary if we want the entries of matrices with decay to actually go to zero with the distance as $n \rightarrow \infty$.

Let us now consider a sequence of $n \times n$ matrices A_n with associated graphs \mathcal{G}_n and graph distances $d_n(i, j)$. We will say that A_n has the *exponential decay property relative to the graph \mathcal{G}_n* if there are constants $K > 0$ and $\alpha > 0$ independent of n such that

$$|[A_n]_{ij}| \leq K e^{-\alpha d_n(i,j)}, \quad \text{for all } i, j = 1, \dots, n, \quad \forall n \in \mathbb{N}. \quad (12)$$

The following two results says that matrices with decay can be “uniformly well approximated” by sparse matrices.

Theorem 3 ([26]) *Let $\{A_n\}$ be a sequence of $n \times n$ matrices satisfying the exponential decay property (12) relative to a sequence of graphs $\{\mathcal{G}_n\}$ having uniformly bounded maximal degree. Then, for any given $0 < \delta < K$, each A_n contains at most $O(n)$ entries greater than δ in magnitude.*

Theorem 4 ([21]) *Let the matrix sequence $\{A_n\}$ satisfy the assumptions of Theorem 3. Then, for all $\varepsilon > 0$ and for all n there exists an $n \times n$ matrix \tilde{A}_n containing only $O(n)$ nonzeros such that*

$$\|A_n - \tilde{A}_n\|_1 < \varepsilon. \quad (13)$$

For example, suppose the each matrix in the sequence $\{A_n\}$ satisfies the following exponential decay property: there exist $K, \alpha > 0$ independent of n such that

$$|[A_n]_{ij}| \leq K e^{-\alpha|i-j|}, \quad \forall i, j = 1, \dots, n, \quad \forall n \in \mathbb{N}.$$

Then, for any $\varepsilon > 0$, there is a sequence of p -banded matrices \tilde{A}_n , with p independent of n , such that $\|A_n - \tilde{A}_n\|_1 < \varepsilon$. The matrices \tilde{A}_n can be defined as follows:

$$[\tilde{A}_n]_{ij} = \begin{cases} [A_n]_{ij} & \text{if } |i - j| \leq p; \\ 0 & \text{otherwise,} \end{cases}$$

where p satisfies

$$p \geq \left\lceil \frac{1}{\alpha} \log \left(\frac{2K}{1 - e^{-\alpha}} \varepsilon^{-1} \right) \right\rceil. \quad (14)$$

Note that \tilde{A}_n is the orthogonal projection of A_n , with respect to the inner product associated with the Frobenius norm, onto the linear subspace of $\mathbb{C}^{n \times n}$ of p -banded matrices.

Similar approximation results hold for other matrix norms. For instance, using the inequality (2) one can easily satisfy error bounds in the matrix 2-norm.

Remark 1 As mentioned in [26], similar results also hold for other types of decay; for instance, it suffices to have algebraic decay of the form

$$|[A_n]_{ij}| \leq K (|i - j|^p + 1)^{-1} \quad \forall i, j, \quad \forall n \in \mathbb{N},$$

with $p > 1$. However, this type of decay is often too slow to be useful in practice, in the sense that any sparse approximation \tilde{A}_n to A_n would have to have $O(n)$ nonzeros with a huge prefactor in order to satisfy (13) for even moderately small values of ε .

3.2 Decay Bounds for the Inverse

It has long been known that the entries in the inverse of banded matrices are bounded in a decaying manner away from the main diagonal, with the decay being faster for more diagonally dominant matrices [67]. In 1984, Demko et al. [68] proved that the entries of A^{-1} , where A is Hermitian positive definite and m -banded ($[A]_{ij} = 0$ if $|i - j| > m$), satisfy the following exponential off-diagonal decay bound:

$$|[A^{-1}]_{ij}| \leq K \rho^{|i-j|}, \quad \forall i, j. \quad (15)$$

Here we have set

$$K = \max\{a^{-1}, K_0\}, \quad K_0 = (1 + \sqrt{\kappa})/2b, \quad \kappa = \frac{b}{a}, \quad (16)$$

where $[a, b]$ is the smallest interval containing the spectrum $\sigma(A)$ of A , and

$$\rho = q^{1/m}, \quad q = q(\kappa) = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}. \quad (17)$$

Hence, the decay bound deteriorates as the relative distance between the spectrum of A and the singularity at zero of the function $f(x) = x^{-1}$ tends to zero (i.e., as $\kappa \rightarrow \infty$) and/or if the bandwidth m increases. The bound is sharp (being attained for certain tridiagonal Toeplitz matrices). The result holds for $n \times n$ matrices as well

as for bounded, infinite matrices acting on the Hilbert space ℓ^2 . We also note that the bound (15) can be rewritten as

$$|[A^{-1}]_{ij}| \leq K e^{-\alpha|i-j|}, \quad \forall i, j, \quad (18)$$

where we have set $\alpha = -\log(\rho)$.

It should be emphasized that (15) is just a bound: the off-diagonal decay in A^{-1} is in general not monotonic. Furthermore the bound, although sharp, may be pessimistic in practice.

The result of Demko et al. implies that if we are given a sequence of $n \times n$ matrices $\{A_n\}$ of increasing size, all Hermitian, positive definite, m -banded (with $m < n_0$) and such that

$$\sigma(A_n) \subset [a, b] \quad \forall n \geq n_0, \quad (19)$$

then the bound (15) holds for all matrices of the sequence; in other words, if the spectra $\sigma(A_n)$ are bounded away from zero and infinity uniformly in n , the entries of A_n^{-1} are uniformly bounded in an exponentially decaying manner (i.e., the decay rates are independent of n). Note that it is not necessary that all matrices have exactly the same bandwidth m , as long as they are banded with bandwidth less than or equal to a constant m .

The requirement that the matrices A_n have uniformly bounded condition number as $n \rightarrow \infty$ is restrictive. For example, it does not apply to banded or sparse matrices that arise from the discretization of differential operators, or in fact of any unbounded operator. Consider for example the sequence of tridiagonal matrices

$$A_n = (n+1)^2 \operatorname{tridiag}(-1, 2, -1)$$

which arise from the three-point finite difference approximation with mesh spacing $h = \frac{1}{n+1}$ of the operator $T = -\frac{d^2}{dx^2}$ with zero Dirichlet conditions at $x = 0$ and $x = 1$. For $n \rightarrow \infty$ the condition number of A_n grows like $O(n^2)$, and although the entries of each inverse A_n^{-1} satisfy a bound of the type (15), the spectral condition number $\kappa_2(A_n)$ is unbounded and therefore the bound deteriorates since $K = K(n) \rightarrow \frac{1}{\pi^2}$ and $\rho = \rho(n) \rightarrow 1$ as $n \rightarrow \infty$. Moreover, in this particular example the actual decay in A_n^{-1} (and not just the bound) slows down as $h \rightarrow 0$. This is to be expected since A_n^{-1} is trying to approximate the Green's function of T , which does not fall off exponentially.

Nevertheless, this result is important for several reasons. First of all, families of banded or sparse matrices (parameterized by the dimension n) exhibiting bounded condition numbers do occur in applications. For example, under mild conditions, mass matrices in finite element analysis and overlap matrices in quantum chemistry satisfy such conditions (these matrices represent the identity operator with respect to some non-orthogonal basis set $\{\phi_i\}_{i=1}^n$, where the ϕ_i are strongly localized in space). Second, the result is important because it suggests a possible *sufficient* condition for

the existence of a uniform exponential decay bound in more general situations: the relative distance of the spectra $\sigma(A_n)$ from the singularities of the function must remain strictly positive as $n \rightarrow \infty$. Third, it turns out that the method of proof used in [68] works with minor changes also for more general functions and matrix classes, as we shall see. The proof of (15) is based on a classical result of Chebyshev on the uniform approximation error

$$\min \max_{a \leq x \leq b} |p_k(x) - x^{-1}|$$

(where the minimum is taken over all polynomials p_k of degree $\leq k$), according to which the error decays exponentially in the degree k as $k \rightarrow \infty$. Combined with the spectral theorem (which allows to go from scalar functions to matrix functions, with the $\|\cdot\|_2$ matrix norm replacing the $\|\cdot\|_\infty$ norm), this result gives the exponential decay bound for $[A^{-1}]_{ij}$. A crucial ingredient of the proof is the fact that if A is m -banded, then A^k is km -banded, for all $k = 0, 1, 2, \dots$.

The paper of Demko et al. also contains some extensions to the case of non-Hermitian matrices and to matrices with a general sparsity pattern. Invertible, non-Hermitian matrices are dealt with by observing that for any $A \in \mathbb{C}^{n \times n}$ one can write

$$A^{-1} = A^*(AA^*)^{-1} \quad (20)$$

and that if A is banded, then the Hermitian positive definite matrix AA^* is also banded (albeit with a larger bandwidth). It is not difficult to see that the product of two matrices, one of which is banded and the other has entries that satisfy an exponential decay bound, is also a matrix with entries that satisfy an exponential decay bound.

For a general sparse matrix, the authors of [68] observe that the entries of A^{-1} are bounded in an exponentially decaying manner away from the support (nonzero pattern) of A . This fact can be expressed in the form

$$|[A^{-1}]_{ij}| \leq K e^{-\alpha d(i,j)}, \quad \forall i, j, \quad (21)$$

where $d(i, j)$ is the geodesic distance between nodes i and j in the undirected graph $\mathcal{G}(A)$ associated with A .

Results similar to those in [68] where independently obtained by Jaffard [117], motivated by problems concerning wavelet expansions. In this paper Jaffard proves exponential decay bounds for the entries of A^{-1} and mentions that similar bounds can be obtained for other matrix functions, such as $A^{-1/2}$ for A positive definite. Moreover, the bounds are formulated for (in general, infinite) matrices the entries of which are indexed by the elements of a suitable metric space, allowing the author to obtain decay results for the inverses of matrices with arbitrary nonzero pattern and even of dense matrices with decaying entries (we will return to this topic in Sect. 3.8).

The exponential decay bound (15) together with Theorem 4 implies the following (asymptotic) uniform approximation result.

Theorem 5 *Let $\{A_n\}$ be a sequence of $n \times n$ matrices, all Hermitian positive definite and m -banded. Assume that there exists an interval $[a, b]$, $0 < a < b < \infty$, such that $\sigma(A_n) \subset [a, b]$, for all n . Then, for all $\varepsilon > 0$ and for all n there exist an integer $p = p(\varepsilon, m, a, b)$ (independent of n) and a matrix $B_n = B_n^*$ with bandwidth p such that $\|A_n^{-1} - B_n\|_2 < \varepsilon$.*

The smallest value of the bandwidth p needed to satisfy the prescribed accuracy can be easily computed via (14). As an example, for tridiagonal matrices A_n ($m = 1$), $K = 10$, $\alpha = 0.6$ (which corresponds to $\rho \approx 0.5488$) we find $\|A_n^{-1} - B_n\|_2 < 10^{-6}$ for all $p \geq 29$, regardless of n . In practice, of course, this result is of interest only for $n > p$ (in fact, for $n \gg p$).

We note that a similar result holds for sparse matrix sequences $\{A_n\}$ corresponding to a sequence of graphs $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ satisfying the assumption (11) of bounded maximum degree. In this case the matrices B_n will be sparse rather than banded, with a maximum number p of nonzeros per row which does not depend on n ; in other words, the graph sequence $\mathcal{G}(B_n)$ associated with the matrix sequence $\{B_n\}$ will also satisfy a condition like (11).

The proof of the decay bound (15) shows that for any prescribed value of $\varepsilon > 0$, each inverse matrix A_n^{-1} can be approximated within ε (in the 2-norm) by a polynomial $p_k(A_n)$ of degree k in A_n , with k independent of n . To this end, it suffices to take the (unique) polynomial of best approximation of degree k of the function $f(x) = x^{-1}$, with k large enough that the error satisfies

$$\max_{a \leq x \leq b} |p_k(x) - x^{-1}| < \varepsilon.$$

In this very special case an exact, closed form expression for the approximation error is known ; see [148, pp. 33–34]. This expression yields an upper bound for the error $\|p_k(A_n) - A_n^{-1}\|_2$, uniform in n . Provided that the assumptions of Theorem 5 are satisfied, the degree k of this polynomial does not depend on n , but only on ε . This shows that it is in principle possible to approximate A_n^{-1} using only $O(n)$ arithmetic operations and storage.

Remark 2 The polynomial of best approximation to the function $f(x) = x^{-1}$ found by Chebyshev does not yield a practically useful expression for the explicit approximation of A^{-1} . However, observing that for any invertible matrix A and any polynomial p

$$\frac{\|A^{-1} - p(A)\|_2}{\|A^{-1}\|_2} \leq \|I - p(A)A\|_2,$$

we can obtain an upper bound on the *relative* approximation error by finding the polynomial of smallest degree k for which

$$\max_{a \leq x \leq b} |1 - p_k(x)x| = \min . \quad (22)$$

Problem (22) admits an explicit solution in terms of shifted and scaled Chebyshev polynomials; see, e.g., [174, p. 381]. Other procedures for approximating the inverse will be briefly mentioned in Sect. 4.1.

A number of improvements, extensions, and refinements of the basic decay results by Demko et al. have been obtained by various authors, largely motivated by applications in numerical analysis, mathematical physics and signal processing, and the topic continues to be actively researched. Decay bounds for the inverses of M -matrices that are near to Toeplitz matrices (a structure that arises frequently in the numerical solution of partial differential equations) can be found in Eijkhout and Polman [75]. Freund [84] obtains an exponential decay bound for the entries of the inverse of a banded matrix A of the form

$$A = cI + dT, \quad T = T^*, \quad c, d \in \mathbb{C}.$$

Exponential decay bounds for resolvents and eigenvectors of infinite banded matrices were obtained by Smith [182]. Decay bounds for the inverses of nonsymmetric band matrices can be found in a paper by Nabben [154]. The paper by Meurant [151] provides an extensive treatment of the tridiagonal and block tridiagonal cases. Inverses of triangular Toeplitz matrices arising from the solution of integral equations also exhibit interesting decay properties; see [83].

A recent development is the derivation of bounds that accurately capture the oscillatory decay behavior observed in the inverses of sparse matrices arising from the discretization of multidimensional partial differential equations. In [47], Canuto et al. obtain bounds for the inverse of matrices in *Kronecker sum* form, i.e., matrices of the type

$$A = T_1 \oplus T_2 := T_1 \otimes I + I \otimes T_2, \quad (23)$$

with T_1 and T_2 banded (for example, tridiagonal). For instance, the 5-point finite difference scheme for the discretization of the Laplacian on a rectangle produces matrices of this form. Generalization to higher-dimensional cases (where A is the Kronecker sum of three or more banded matrices) is also possible.

3.3 Decay Bounds for the Matrix Exponential

As we have seen in the Introduction, the entries in the exponential of banded matrices can exhibit rapid off-diagonal decay (see Fig. 2). As it turns out, the actual decay rate is faster than exponential (the term *superexponential* is often used), a phenomenon common to all entire functions of a matrix. More precisely, we have the following definition.

Definition 2 A matrix A has the *superexponential off-diagonal decay property* if for any $\alpha > 0$ there exists a $K > 0$ such that

$$|[A]_{ij}| \leq K e^{-\alpha|i-j|} \quad \forall i, j.$$

As usual, in this definition A is either infinite or a member of a sequence of matrices of increasing order, in which case K and α do not depend on the order. The definition can be readily extended to decay with respect to a general nonzero pattern, in which case $|i-j|$ must be replaced by the geodesic distance on the corresponding graph.

A superexponential decay bound on the entries of the exponential of a *tridiagonal* matrix has been obtained by Iserles [115]. The bound takes the form

$$|[\exp^A]_{ij}| \leq e^\rho I_{|i-j|}(2\rho), \quad i, j = 1, \dots, n \quad (24)$$

where $\rho = \max_{i,j} |[A]_{ij}|$ and $I_\nu(z)$ is the *modified Bessel function of the first kind*:

$$I_\nu(z) = \left(\frac{1}{2}z\right)^\nu \sum_{k=0}^{\infty} \frac{\left(\frac{1}{4}z^2\right)^k}{k! \Gamma(\nu + k + 1)},$$

where $\nu \in \mathbb{R}$ and Γ is the gamma function; see [1]. For any fixed value of $z \in \mathbb{C}$, the values of $|I_\nu(z)|$ decay faster than exponentially for $\nu \rightarrow \infty$. The paper by Iserles also presents superexponential decay bounds for the exponential of more general banded matrices, but the bounds only apply at sufficiently large distances from the main diagonal. None of these bounds require A to be Hermitian.

In [22], new decay bounds for the entries of the exponential of a banded, Hermitian, positive semidefinite matrix A have been presented. The bounds are a consequence of fundamental error bounds for Krylov subspace approximations to the matrix exponential due to Hochbruch and Lubich [109]. The decay bounds are as follows.

Theorem 6 ([22]) *Let A be a Hermitian positive semidefinite matrix with eigenvalues in the interval $[0, 4\rho]$ and let $\tau > 0$. Assume in addition that A is m -banded. For $i \neq j$, let $\xi = \lceil |i-j|/m \rceil$. Then*

i) *For $\rho\tau \geq 1$ and $\sqrt{4\rho\tau} \leq \xi \leq 2\rho\tau$,*

$$|[\exp(-\tau A)]_{ij}| \leq 10 \exp\left(-\frac{1}{5\rho\tau}\xi^2\right);$$

ii) *For $\xi \geq 2\rho\tau$,*

$$|[\exp(-\tau A)]_{ij}| \leq 10 \frac{\exp(-\rho\tau)}{\rho\tau} \left(\frac{e\rho\tau}{\xi}\right)^\xi.$$

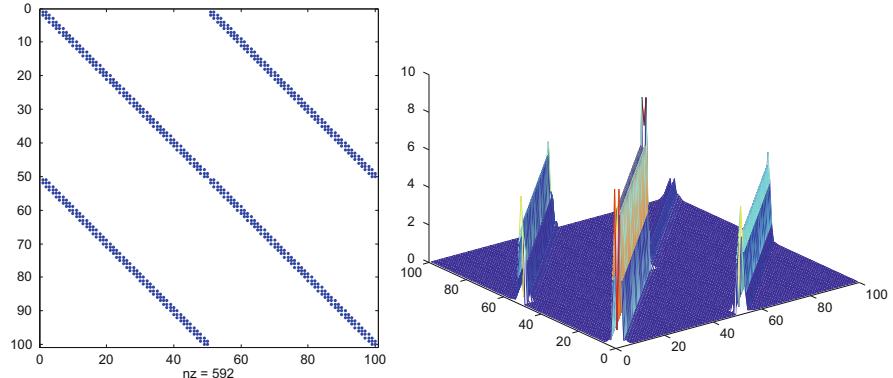


Fig. 8 Sparsity pattern of multi-banded matrix A and decay in e^A

As shown in [22], these bounds are quite tight and capture the actual superexponential decay behavior very well. Similar bounds can be derived for the skew-Hermitian case ($A = -A^*$). See also [177], where decay bounds are derived for the exponential of a class of *unbounded* infinite skew-Hermitian tridiagonal matrices arising in quantum mechanical problems, and [199].

These bounds can also be adapted to describe the decay behavior of the exponential of matrices with a general sparsity pattern. See Fig. 8 for an example.

Bounds for the matrix exponential in the nonnormal case will be discussed in Sect. 3.4.2 below, as special cases of bounds for general analytic functions of matrices.

We note that exploiting the well known identity

$$\exp(A \oplus B) = \exp(A) \otimes \exp(B) \quad (25)$$

(see [107, Theorem 10.9]), it is possible to use Theorem 6 to obtain bounds for the exponential of a matrix that is the Kronecker sum of two (or more) banded matrices; these bounds succeed in capturing the oscillatory decay behavior in the exponential of such matrices (see [22]).

3.4 Decay Bounds for General Analytic Functions

In this section we present decay bounds for the entries of matrix functions of the form $f(A)$ where f is analytic on an open connected set $\Omega \subseteq \mathbb{C}$ with $\sigma(A) \subset \Omega$ and A is banded or sparse. These bounds are obtained combining classical results on the approximation of analytic functions by polynomials with the spectral theorem, similar to the approach used by Demko et al. in [68] to prove exponential decay in the inverses of banded matrices. The classical Chebyshev expression for the error

incurred by the polynomials of best approximation (in the infinity norm) of $f(x) = x^{-1}$ will be replaced by an equally classical bound (due to S.N. Bernstein) valid for arbitrary analytic functions. The greater generality of Bernstein's result comes at a price: instead of having an exact expression for the approximation error, it provides only an upper bound. This is sufficient, however, for our purposes.

We begin with the Hermitian case.⁶ If $[a, b] \subset \mathbb{R}$ denotes any interval containing the spectrum of a (possibly infinite) matrix $A = A^*$, the shifted and scaled matrix

$$\hat{A} = \frac{2}{b-a}A - \frac{a+b}{a-b}I \quad (26)$$

has spectrum contained in $[-1, 1]$. Since decay bounds are simpler to express for functions of matrices with spectrum contained in $[-1, 1]$ than in a general interval $[a, b]$, we will make the assumption that A has already been scaled and shifted so that $\sigma(A) \subseteq [-1, 1]$. It is in general not difficult to translate the decay bounds in terms of the original matrix, if required. In practice it is desirable that $[-1, 1]$ is the smallest interval containing the spectrum of the scaled and shifted matrix.

Given a function f continuous on $[-1, 1]$ and a positive integer k , the k th best approximation error for f by polynomials is the quantity

$$E_k(f) = \inf \left\{ \max_{-1 \leq x \leq 1} |f(x) - p(x)| : p \in \mathbb{P}_k \right\},$$

where \mathbb{P}_k is the set of all polynomials of degree less than or equal to k . Bernstein's Theorem describes the asymptotic behavior of the best polynomial approximation error for a function f analytic on a domain containing the interval $[-1, 1]$.

Consider now the family of ellipses in the complex plane with foci in -1 and 1 . Any ellipse in this family is completely determined by the sum $\chi > 1$ of the lengths of its half-axes; if these are denoted by $\kappa_1 > 1$ and $\kappa_2 > 0$, it is well known that

$$\sqrt{\kappa_1^2 - \kappa_2^2} = 1, \quad \kappa_1 - \kappa_2 = 1/(\kappa_1 + \kappa_2) = 1/\chi.$$

We will denote the ellipse characterized by $\chi > 1$ by \mathcal{E}_χ .

If f is analytic on a region (open simply connected subset) of \mathbb{C} containing $[-1, 1]$, then there exists an infinite family of ellipses \mathcal{E}_χ with $1 < \chi < \bar{\chi}$ such that f is analytic in the interior of \mathcal{E}_χ and continuous on \mathcal{E}_χ . Moreover, $\bar{\chi} = \infty$ if and only if f is entire.

The following fundamental result is known as *Bernstein's Theorem*.

⁶The treatment is essentially the same for any normal matrix with eigenvalues lying on a line segment in the complex plane, in particular if A is skew-Hermitian.

Theorem 7 Let the function f be analytic in the interior of the ellipse \mathcal{E}_χ and continuous on \mathcal{E}_χ , for $\chi > 1$. Then

$$E_k(f) \leq \frac{2M(\chi)}{\chi^k(\chi - 1)},$$

where $M(\chi) = \max_{z \in \mathcal{E}_\chi} |f(z)|$.

Proof See, e.g., [143, Sect. 3.15].

Hence, if f is analytic, the error corresponding to polynomials of best approximation in the uniform convergence norm decays exponentially with the degree of the polynomial. As a consequence, we obtain the following exponential decay bounds on the entries of $f(A)$. We include the proof (modeled after the one in [68]) as it is instructive.

Theorem 8 ([20]) Let $A = A^*$ be m -banded with spectrum $\sigma(A)$ contained in $[-1, 1]$ and let f be analytic in the interior of \mathcal{E}_χ and continuous on \mathcal{E}_χ for $1 < \chi < \bar{\chi}$. Let

$$\rho := \chi^{-\frac{1}{m}}, \quad M(\chi) = \max_{z \in \mathcal{E}_\chi} |f(z)|, \quad \text{and} \quad K = \frac{2\chi M(\chi)}{\chi - 1}.$$

Then

$$|[f(A)]_{ij}| \leq K \rho^{|i-j|}, \quad \forall i, j. \quad (27)$$

Proof Let p_k be the polynomial of degree k of best uniform approximation for f on $[-1, 1]$. First, observe that if A is m -banded then A^k (and therefore $p_k(A)$) is km -banded: $[p_k(A)]_{ij} = 0$ if $|i-j| > km$. For $i \neq j$ write $|i-j| = km + l$, $l = 1, 2, \dots, m$, hence $k < |i-j|/m$ and $\chi^{-k} < \chi^{-\frac{|i-j|}{m}} = \rho^{|i-j|}$. Therefore, for all $i \neq j$ we have

$$|[f(A)]_{ij}| = |[f(A)]_{ij} - [p_k(A)]_{ij}| \leq \|f(A) - p_k(A)\|_2 \leq \|f - p_k\|_\infty \leq K\rho^{|i-j|}.$$

The last inequality follows from Theorem 7. For $i = j$ we have $|[f(A)]_{ii}| \leq \|f(A)\|_2 < K$ (since $2\chi/(\chi - 1) > 1$ and $\|f(A)\|_2 \leq M(\chi)$ for all $\chi > 1$ by the maximum principle). Therefore the bound (27) holds for all i, j .

Note that the bound can also be expressed as

$$|[f(A)]_{ij}| \leq K e^{-\alpha|i-j|}, \quad \forall i, j,$$

by introducing $\alpha = -\log(\rho) > 0$.

Remark 3 An important difference between (27) and bound (15) is that (27) actually represents an infinite family of bounds, one for every $\chi \in (1, \bar{\chi})$. Hence, we cannot expect (27) to be sharp for any fixed value of χ . There is a clear trade-off

involved in the choice of χ ; larger values of χ result in faster exponential decay (smaller ρ) and smaller values of $2\chi/(\chi - 1) > 1$ (which is a monotonically decreasing function of χ for $\chi > 1$), but potentially much larger values of $M(\chi)$. In particular, as χ approaches $\bar{\chi}$ from below, we must have $M(\chi) \rightarrow \infty$. As noted in [26, pp. 27–28] and [177, p. 70], for any entry (i,j) of interest the bound (27) can be optimized by finding the value of $\chi \in (1, \bar{\chi})$ that minimizes the right-hand side of (27); for many functions of practical interest there is a unique minimizer which can be found numerically if necessary.

Remark 4 Theorem 8 can be applied to both finite matrices and bounded infinite matrices on ℓ^2 . Note that infinite matrices may have continuous spectrum, and indeed it can be $\sigma(A) = [-1, 1]$. The result is most usefully applied to matrix sequences $\{A_n\}$ of increasing size, all m -banded (or with bandwidth $\leq m$ for all n) and such that

$$\bigcup_{n=1}^{\infty} \sigma(A_n) \subset [-1, 1],$$

assuming f is analytic on a region $\Omega \subseteq \mathbb{C}$ containing $[-1, 1]$ in its interior. For instance, each A_n could be a finite section of a bounded infinite matrix A on ℓ^2 with $\sigma(A) \subseteq [-1, 1]$. The bound (27) then becomes

$$|[f(A_n)]_{ij}| \leq K \rho^{|i-j|} \quad \forall i, j, \quad \forall n \in \mathbb{N}. \quad (28)$$

In other words, the bounds (28) are uniform in n . Analogous to Theorem 5, it follows that under the conditions of Theorem 8, for any prescribed $\varepsilon > 0$ there exists a positive integer p and a sequence of p -banded matrices $B_n = B_n^*$ such that

$$\|f(A_n) - B_n\|_2 < \varepsilon.$$

Moreover, the proof of Theorem 8 shows that each B_n can be taken to be a polynomial in A_n , which does not depend on n . Therefore, it is possible in principle to approximate $f(A_n)$ with arbitrary accuracy in $O(n)$ work and storage.

We emphasize again that the restriction to the interval $[-1, 1]$ is done for ease of exposition only; in practice, it suffices that there exists a bounded interval $\mathcal{I} = [a, b] \subset \mathbb{R}$ such that $\sigma(A_n) \subset [a, b]$ for all $n \in \mathbb{N}$. In this case we require f to be analytic on a region of \mathbb{C} containing $[a, b]$ in its interior. The result can then be applied to the corresponding shifted and scaled matrices \hat{A}_n with spectrum in $[-1, 1]$, see (26). The following example illustrates how to obtain the decay bounds expressed in terms of the original matrices in a special case.

Example 1 The following example is taken from [20]. Assume that $A = A^*$ is m -banded and has spectrum in $[a, b]$ where $b > a > 0$, and suppose we want to obtain decay bounds on the entries of $A^{-1/2}$. Note that there is an infinite family of ellipses $\{\mathcal{E}_{\xi}\}$ entirely contained in the open half plane with foci in a and b , such that the

function $F(z) = z^{-1/2}$ is analytic on the interior of each \mathcal{E}_ξ and continuous on it. If ψ denotes the linear affine mapping

$$\psi(z) = \frac{2z - (a + b)}{b - a}$$

which maps $[a, b]$ to $[-1, 1]$, we can apply Theorem 8 to the function $f = F \circ \psi^{-1}$, where

$$\psi^{-1}(w) = \frac{(b - a)w + a + b}{2}.$$

Obviously, f is analytic on the interior of a family \mathcal{E}_χ of ellipses (images via ψ of the \mathcal{E}_ξ) with foci in $[-1, 1]$ and continuous on each \mathcal{E}_χ , with $1 < \chi < \bar{\chi}$. An easy calculation shows that

$$\bar{\chi} = \frac{b + a}{b - a} + \sqrt{\left(\frac{b + a}{b - a}\right)^2 - 1} = \frac{(\sqrt{\kappa + 1})^2}{\kappa - 1},$$

where $\kappa = \frac{b}{a}$. Finally, for any $\chi \in (1, \bar{\chi})$ we easily find (recalling that $\chi = \kappa_1 + \kappa_2$)

$$M(\chi) = \max_{z \in \mathcal{E}_\chi} |f(z)| = |f(-\kappa_1)| = \frac{\sqrt{2}}{\sqrt{(a - b)\kappa_1 + a + b}} = \frac{\sqrt{2}}{\sqrt{\frac{(a - b)(\chi^2 + 1)}{2\chi} + a + b}}.$$

It is now possible to compute the bounds (27) for any $\chi \in (1, \bar{\chi})$ and for all i, j . Note that if b is fixed and $a \rightarrow 0+$, $M(\chi)$ grows without bound and $\rho \rightarrow 1-$, showing that the decay bound deteriorates as A becomes nearly singular. Conversely, for well-conditioned A decay can be very fast, since $\bar{\chi}$ will be large for small conditioned numbers κ . This is analogous to the situation for A^{-1} .

More generally, the decay rate in the bound (27) depends on the distance between the singularities of f (if any) and the interval $[a, b]$ (and, of course, on the bandwidth m). If f has any singularities near $[a, b]$ then $\bar{\chi}$ will be necessarily close to 1, and the bound (27) will decay very slowly. Conversely, if they are far from $[a, b]$ then χ can be taken large and decay will be fast.

In the case of an entire function, χ can be taken arbitrarily large, so that the exponential decay part of the bound decays arbitrarily fast; note, however, that this will cause K to increase. Thus, it is clear that for f entire and A banded, the entries of $f(A)$ are bounded in a superexponentially decay manner according to Definition 2; see [177]. As a special case, we have an alternative proof of the superexponential decay for the matrix exponential. Note, however, that in the case of the matrix exponential the specialized bounds given in Theorem 6 are generally tighter.

Remark 5 Let now $\{A_n\}$ be a sequence of m -banded matrices of increasing size. It is clear that if $\sigma(A_n)$ is not bounded away from the singularities of f for all n ,

then we cannot expect to have uniform decay bounds like (27) valid for all n . The same happens in the case of a (non-constant) entire function f if the smallest interval containing $\sigma(A_n)$ is unbounded as $n \rightarrow \infty$. Hence, the bounds (27) cannot be expected to hold uniformly for matrices A_n arising from the discretization of unbounded differential operators if the size n is related to the mesh size h (in the sense that $n \rightarrow \infty$ if $h \rightarrow 0$). Nevertheless, we will see that there are important applications where the decay bounds (8) hold uniformly in n .

As in the case of the inverse, the bounds (8) can be extended, with some caution, from the banded case to the case of matrices with more general sparsity patterns. We formally state this result as follows.

Theorem 9 ([26]) *Let $\{A_n\}$ be a sequence of sparse Hermitian matrices of increasing size. Assume that there exists a bounded interval $[a, b] \subset \mathbb{R}$ such that $\sigma(A_n) \subset [a, b]$ for all $n \in \mathbb{N}$, and that the sequence of graphs $\mathcal{G}(A_n)$ has bounded maximum degree. If f is analytic on a region containing $[a, b]$ in its interior, there exist constants $K > 0$ and $\alpha > 0$ such that*

$$|[f(A_n)]_{ij}| \leq K e^{-\alpha d_n(i,j)}, \quad \forall i, j, \quad \forall n \in \mathbb{N}, \quad (29)$$

where d_n denotes the geodesic distance on $\mathcal{G}(A_n)$. The constants K and α depend on $[a, b]$ and on the maximum degree of any node in the graph sequence $\{\mathcal{G}(A_n)\}$, but not on n .

As before, (29) is actually an infinite family of bounds parameterized by χ , the sum of the semi-axes of the infinitely many ellipses with foci in a and b entirely contained in the largest simply connected region Ω on which f is analytic. The expressions for K and α (equivalently, ρ) are exactly as in Theorem 8 when $a = -1$ and $b = -1$, otherwise they can be found as shown in Example 1.

Theorem 9 also holds if the sequence $\{A_n\}$ is replaced by a single bounded infinite matrix acting on ℓ^2 such that $\sigma(A) \subseteq [a, b]$ and such that the infinite graph $\mathcal{G}(A)$ has finite maximum degree.

Remark 6 The proof of Theorem 8 shows that the bounds (27) and (29) are in general pessimistic. Indeed, much can be lost when bounding $|[f(A)]_{ij} - [p_k(A)]_{ij}|$ with $\|f(A) - p_k(A)\|_2$ and the latter quantity with $\|f - p_k\|_\infty$. In particular, these bounds completely ignore the distribution of the eigenvalues of A in the interval $[-1, 1]$; in this sense, the situation is completely analogous to the well known error estimate for the A -norm of the error in the conjugate gradient method based on the condition number of A , see [95, p. 636]. It is well known that this bound, while sharp, can be overly conservative. The same holds for (29): the bound on the rate of decay depends on a single essential quantity, the distance between the spectrum of A and the singularities of f , and thus cannot be expected to be very accurate. More accurate bounds can likely be obtained given more information on the spectral distribution of A ; for example, if most of the eigenvalues of A are tightly clustered, then the decay rate in $f(A)$ should be much faster than if the eigenvalues of A are distributed more or less evenly in the spectral interval. In the limiting case where

A has only $k \ll n$ distinct eigenvalues (so that the minimum polynomial of A has degree k), then $f(A)$ can be represented exactly by a low degree polynomial, and many of the entries of A will be exactly zero as long as $\text{diam}(\mathcal{G}(A)) \gg k$.

3.4.1 Bounds for the Normal Case

Owing to the fact that the spectral theorem applies not just to Hermitian matrices but more generally to normal matrices, it is not surprising that results completely analogous to Theorems 8 and 9 can be stated and proved assuming that $\{A_n\}$ is a sequence of normal matrices (banded or sparse) with eigenvalues lying on a line segment $[z_1, z_2] \subset \mathbb{C}$ entirely contained in a region Ω on which f is analytic. For instance, decay bounds for the entries of functions of banded skew-symmetric matrices have been given in [66, 141].

More generally, suppose A is normal and m -banded. Let $\mathcal{F} \subset \mathbb{C}$ be a compact, connected region containing $\sigma(A)$, and denote by \mathbb{P}_k the set of complex polynomials of degree at most k . Then the argument in the proof of Theorem 8 still holds, except that now polynomial approximation is no longer applied on an interval, but on the complex region \mathcal{F} . Therefore, the following bound holds for all indices i, j such that $|i - j| > km$:

$$|[f(A)]_{ij}| \leq \max_{\lambda \in \sigma(A)} |f(\lambda) - p(\lambda)| \leq E_k(f, \mathcal{F}), \quad (30)$$

where

$$E_k(f, \mathcal{F}) := \min_{p \in \mathbb{P}_k} \max_{z \in \mathcal{F}} |f(z) - p(z)| =: \min_{p \in \mathbb{P}_k} \|f - p\|_{\infty, \mathcal{F}}.$$

Unless more accurate estimates for $\sigma(A)$ are available, a possible choice for \mathcal{F} is the disk of center 0 and radius $\varrho(A)$.

If f is analytic on \mathcal{F} , bounds for $E_k(f, \mathcal{F})$ that decay exponentially with k are available through the use of *Faber polynomials*: see [21, Theorem 3.3] and the next subsection for more details. More precisely, there exist constants $\tilde{c} > 0$ and $0 < \tilde{\rho} < 1$ such that $E_k(f, \mathcal{F}) \leq \tilde{c} \tilde{\rho}^k$ for all $k \in \mathbb{N}$. This result, together with (30), yields for all i and j the bound

$$|[f(A)]_{ij}| \leq K \rho^{|i-j|} = K e^{-\alpha|i-j|} \quad (31)$$

(where $\alpha = -\log(\rho)$) for suitable constants $K > 0$ and $0 < \rho < 1$, which do not depend on the size of the matrix n , although they generally depend on f and \mathcal{F} (and of course on the bandwidth, m , in the case of ρ).

The extension of these bounds to sparse matrices with more general sparsity patterns is entirely straightforward; note, however, that unless A is structurally symmetric (in which case the graph $\mathcal{G}(A)$ is undirected), the distance $d(i, j)$, defined as the length of the shortest directed path starting at node i and ending at node j ,

will be different, in general, from $d(j, i)$. Hence, different rates of decay may be observed on either side of the main diagonal.

3.4.2 Bounds for the Nonnormal Case

As can be expected, the derivation of decay bounds for the entries of $f(A)$ when A is banded or sparse and nonnormal is more challenging than in the normal case, since in this case the spectral theorem can no longer be relied upon.

It is easy to give examples where decay fails to occur. The simplest one is probably the following. Let A_n be the $n \times n$ upper bidiagonal matrix

$$A_n = \begin{bmatrix} 1 & -\alpha & 0 & \dots & 0 \\ 0 & 1 & -\alpha & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -\alpha \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}, \quad (32)$$

where $\alpha \in \mathbb{R}$. Then

$$A_n^{-1} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{n-1} \\ 0 & 1 & \alpha & \dots & \alpha^{n-2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & \alpha \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}.$$

Hence, for $\alpha \geq 1$ no decay is present⁷ in the upper triangular part of A_n^{-1} .

Nevertheless, useful bounds can still be obtained in many cases. In order to proceed, we need some additional background in approximation theory, namely, some notions about Faber polynomials and their use in the approximation of analytic functions on certain regions of the complex plane. In a nutshell, Faber polynomials play for these regions the same role played by Taylor polynomials for disks. The following discussion is taken from [21] and follows closely the treatment in [143].

A *continuum* in \mathbb{C} is a nonempty, compact and connected subset of \mathbb{C} . Let F be a continuum consisting of more than one point. Let G_∞ denote the component of the complement of F containing the point at infinity. Note that G_∞ is a simply connected domain in the extended complex plane $\bar{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$. By the Riemann Mapping Theorem there exists a function $w = \Phi(z)$ which maps G_∞ conformally

⁷At first sight, this example seems to contradict the result by Demko et al. [68] based on the identity (20). However, the result of Demko et al. assumes that the condition number of AA^* (equivalently, of A itself) remains bounded as $n \rightarrow \infty$, which is not the case for this example.

onto a domain of the form $|w| > \rho > 0$ satisfying the normalization conditions

$$\Phi(\infty) = \infty, \quad \lim_{z \rightarrow \infty} \frac{\Phi(z)}{z} = 1; \quad (33)$$

ρ is the *logarithmic capacity* of F . Given any integer $N > 0$, the function $[\Phi(z)]^N$ has a Laurent series expansion of the form

$$[\Phi(z)]^N = z^N + \alpha_{N-1}^{(N)} z^{N-1} + \cdots + \alpha_0^{(N)} + \frac{\alpha_{-1}^{(N)}}{z} + \cdots \quad (34)$$

at infinity. The polynomials

$$\Phi_N(z) = z^N + \alpha_{N-1}^{(N)} z^{N-1} + \cdots + \alpha_0^{(N)}$$

consisting of the terms with nonnegative powers of z in the expansion (34) are called the *Faber polynomials generated by the continuum F* .

Let Ψ be the inverse of Φ . By C_R we denote the image under Ψ of a circle $|w| = R > \rho$. The (Jordan) region with boundary C_R is denoted by $I(C_R)$. By Theorem 3.17, p. 109 of [143], every function f analytic on $I(C_{R_0})$ with $R_0 > \rho$ can be expanded in a series of Faber polynomials:

$$f(z) = \sum_{k=0}^{\infty} \alpha_k \Phi_k(z), \quad (35)$$

where the series converges uniformly inside $I(C_{R_0})$. The coefficients are given by

$$\alpha_k = \frac{1}{2\pi i} \int_{|w|=R} \frac{f(\Psi(w))}{w^{k+1}} dw$$

where $\rho < R < R_0$. We denote the partial sums of the series in (35) by

$$\Pi_N(z) := \sum_{k=0}^N \alpha_k \Phi_k(z). \quad (36)$$

Each $\Pi_N(z)$ is a polynomial of degree at most N , since each $\Phi_k(z)$ is of degree k . We are now ready to state the following generalization of Theorem 7, also due to Bernstein, which will be instrumental for what follows.

Theorem 10 *Let f be a function defined on F . Then given any $\varepsilon > 0$ and any integer $N \geq 0$, there exists a polynomial Π_N of degree at most N and a positive constant $c(\varepsilon)$ such that*

$$|f(z) - \Pi_N(z)| < c(\varepsilon)(q + \varepsilon)^N \quad (0 < q < 1) \quad (37)$$

for all $z \in F$ if and only iff f is analytic on the domain $I(C_{R_0})$, where $R_0 = \rho/q$. In this case, the sequence $\{\Pi_N\}$ converges uniformly to f inside $I(C_{R_0})$ as $N \rightarrow \infty$.

In the special case where F is a disk of radius ρ centered at z_0 , Theorem 10 states that for any function f analytic on the disk of radius ρ/q centered at z_0 , where $0 < q < 1$, there exists a polynomial Π_N of degree at most N and a positive constant $c(\varepsilon)$ such that for any $\varepsilon > 0$

$$|f(z) - \Pi_N(z)| < c(\varepsilon)(q + \varepsilon)^N, \quad (38)$$

for all $z \in F$. We are primarily concerned with the sufficiency part of Theorem 10. Note that the choice of q (with $0 < q < 1$) depends on the region where the function f is analytic. If f is defined on a continuum F with logarithmic capacity ρ then we can pick q bounded away from 1 as long as the function is analytic on $I(C_{\rho/q})$. Therefore, the rate of convergence is directly related to the properties of the function f , such as the location of its poles (if there are any). Following [143], the constant $c(\varepsilon)$ can be estimated as follows. Let R_0, q and ε be given as in Theorem 10. Furthermore, let R' and R be chosen such that $\rho < R' < R < R_0$ and

$$\frac{R'}{R} = q + \varepsilon,$$

then we define

$$M(R) = \max_{z \in C_R} |f(z)|.$$

An estimate for the value of $c(\varepsilon)$ is asymptotically (i.e., for sufficiently large N) given by

$$c(\varepsilon) \approx \frac{3}{2} M(R) \frac{q + \varepsilon}{1 - (q + \varepsilon)}.$$

It may be necessary to replace the above expression for $c(\varepsilon)$ by a larger one to obtain validity of the bound (37) for all N . However, for certain regions (and in particular for convex F) it is possible to obtain an explicit constant valid for all $N \geq 0$; see [77] and [21, Sect. 3.7]. Based on this theorem, we can state the following result.

Theorem 11 *Let $\{A_n\}$ be a sequence of $n \times n$ diagonalizable matrices and assume that $\sigma(A_n)$ is contained in a continuum F , for all n . Assume further that the matrices A_n are sparse and that each graph $\mathcal{G}(A_n)$ satisfies the maximum bounded degree assumption. Let $\kappa_2(X_n)$ be the spectral condition number of the eigenvector matrix of A_n . Let f be a function defined on F . Furthermore, assume that f is analytic on $I(C_{R_0}) (\supset \sigma(A_n))$, where $R_0 = \frac{\rho}{q}$ with $0 < q < 1$ and ρ is the logarithmic capacity of F . Then there are positive constants K and α , independent of n , such that*

$$|[f(A_n)]_{ij}| < \kappa(X_n) K e^{-\alpha d_n(i,j)}, \quad \forall i, j, \quad \forall n \in \mathbb{N}, \quad (39)$$

where d_n denotes the geodesic distance on $\mathcal{G}(A_n)$.

Proof From Theorem 10 we know that for any $\varepsilon > 0$ there exists a sequence of polynomials Π_k of degree k which satisfies for all $z \in F$

$$|f(z) - \Pi_k(z)| < c(\varepsilon)(q + \varepsilon)^k, \quad \text{where } 0 < q < 1.$$

Therefore, since $A_n = X_n D_n X_n^{-1}$ with D_n diagonal, we have

$$\|f(A_n) - \Pi_k(A_n)\|_2 \leq \kappa_2(X_n) \max_{z \in \sigma(A_n)} |f(z) - \Pi_k(z)| < \kappa_2(X_n) c(\varepsilon)(q + \varepsilon)^k,$$

where $0 < q < 1$. For $i \neq j$ we can write

$$d_n(i,j) = k + 1$$

and therefore, observing that $[\Pi_k(A_n)]_{ij} = 0$ for $d_n(i,j) > k$, we have

$$|[f(A_n)]_{ij}| = |[f(A_n)]_{ij} - [\Pi_k(A_n)]_{ij}| \leq \|f(A_n) - \Pi_k(A_n)\|_2 < \kappa_2(X_n) c(\varepsilon)(q + \varepsilon)^{d_n(i,j)-1}.$$

Hence, choosing $\varepsilon > 0$ such that $\rho_0 := q + \varepsilon < 1$ and letting $K_0 = c(\varepsilon)/(q + \varepsilon)$ we obtain

$$|[f(A_n)]_{ij}| < \kappa_2(X_n) K_0 \rho_0^{d_n(i,j)}.$$

If $i = j$ then $|[f(A_n)]_{ii}| \leq \|f(A_n)\|_2$ and therefore letting $K = \max\{K_0, \|f(A)\|_2\}$ and $\alpha = -\log(\rho_0)$ we see that inequality (39) holds for all i, j .

It is worth noting that in the normal case we have $\kappa_2(X_n) = 1$ in (39), and therefore the bound (31) is proved. Bound (39) may also prove useful if the spectral condition numbers $\kappa_2(X_n)$ are uniformly bounded by a (moderate) constant. However, in the case of a highly nonnormal sequence (for which the $\kappa_2(X_n)$ are necessarily very large, see [95, P7.2.3]) the bound is virtually useless as it can be a severe overestimate of the actual size of the elements of $f(A_n)$; see [21, p. 25] for an example. The situation is entirely analogous to the standard residual bound for GMRES applied to diagonalizable matrices; see, e.g., [174, Proposition 6.32].

In this case a different approach, based on the field of values and not requiring diagonalizability, is often found to give better bounds. The following discussion is based on [19]. The field of values of a complex matrix appears in the context of bounds for functions of matrices thanks to a fundamental result by Crouzeix (see [57]):

Theorem 12 (Crouzeix) *There is a universal constant $2 \leq \mathcal{Q} \leq 11.08$ such that, given $A \in \mathbb{C}^{n \times n}$, \mathcal{F} a convex compact set containing the field of values $\mathcal{W}(A)$, a function g continuous on \mathcal{F} and analytic in its interior, then the following inequality holds:*

$$\|g(A)\|_2 \leq \mathcal{Q} \sup_{z \in \mathcal{F}} |g(z)|.$$

We mention that Crouzeix has conjectured that \mathcal{Q} can be replaced by 2, but so far this has been proved only in some special cases. Combining Theorem 10 with Theorem 12 we obtain the following result from [19].

Theorem 13 ([19]) *Let $\{A_n\}$ be a sequence of banded $n \times n$ matrices, with bandwidths uniformly bounded by m . Let the complex function f be analytic on a neighborhood of a connected compact set $\mathcal{C} \subset \mathbb{C}$ containing $\mathcal{W}(A_n)$ for all n . Then there exist explicitly computable constants $K > 0$ and $\alpha > 0$, independent of n , such that*

$$|[f(A_n)]_{ij}| \leq K e^{-\alpha|i-j|} \quad (40)$$

for all indices i, j , and for all $n \in \mathbb{N}$.

This result is similar to the one for the normal case, with the field of values $\mathcal{W}(A_n)$ now playing the roles of the spectra $\sigma(A_n)$. As long as the singularities of f (if any) stay bounded away from a fixed compact set \mathcal{C} containing the union of all the fields of values $\mathcal{W}(A_n)$, and as long as the matrices A_n have bandwidths less than a fixed integer m , the entries of $f(A_n)$ are bounded in an exponentially decaying manner away from the main diagonal, at a rate bounded below by a fixed positive constant as $n \rightarrow \infty$. The larger the distance between the singularities of f and the compact \mathcal{C} , the larger this constant is (and the faster the bound decays).

As usual, the same result holds for sequences of sparse matrices A_n such that the graphs $\mathcal{G}(A_n)$ satisfy the bounded maximum degree assumption, in which case $|i-j|$ in (40) is replaced by the geodesic distance $d_n(i, j)$.

In Fig. 9 we show three plots which correspond to the first row of e^A where A is the 100×100 nonnormal tridiagonal Toeplitz matrix generated by the symbol $\phi(t) = 2t^{-1} + 1 + 3t$, see [41]. This matrix is diagonalizable with eigenvector matrix X such that $\kappa_2(X) \approx 5.26 \cdot 10^8$. The lowest curve is the actual magnitude of the entries $[e^A]_{lj}$ for $j = 1, \dots, 100$ (drawn as a continuous curve). The top curve is the bound (39), and the curve in the middle is the bound (40) obtained from Crouzeix's Theorem (with $\mathcal{C} = \mathcal{W}(A)$). Note the logarithmic scale on the vertical axis. Clearly, for this example the Crouzeix-type bound is a significant improvement over the earlier bound from [21].

A practical limitation of bound (40) is that it is in general difficult to find the constants K and α . This requires knowledge of the compact set \mathcal{C} in the statement of Theorem 13. If such a set is known, a simple approach to the computation of constants K and α goes as follows [19, 147]. Suppose there is a disk of center 0 and radius $r > 0$ that contains \mathcal{C} , and such that f is analytic on an open neighborhood of some disk of center 0 and radius $R > r$. Define

$$E_k(f, \mathcal{C}) = \inf \max_{z \in \mathcal{C}} |f(z) - p_k(z)|,$$

where the infimum is taken over all polynomials with complex coefficients of degree $\leq k$. Then the standard theory of complex Taylor series gives the following estimate

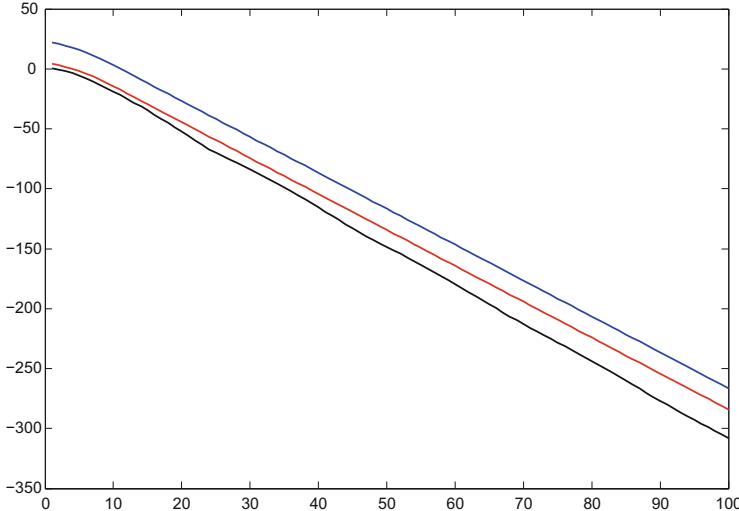


Fig. 9 Black: first row of e^A . Blue: bound (39). Red: bound (40)

for the Taylor approximation error [77, Corollary 2.2]:

$$E_k(f, \mathcal{C}) \leq \frac{M(R)}{1 - \frac{r}{R}} \left(\frac{r}{R} \right)^{k+1}, \quad (41)$$

where $M(R) = \max_{|z|=R} |f(z)|$. Therefore we can choose

$$K = \max \left\{ \|f(A)\|, Q M(R) \frac{r}{R-r} \right\}, \quad \hat{\rho} = \left(\frac{r}{R} \right)^{1/m}, \quad \alpha = -\log(\hat{\rho}).$$

The choice of the parameter R in (41) is somewhat arbitrary: any value of R will do, as long as $r < R < \min |\zeta|$, where ζ varies over the poles of f (if f is entire, we let $\min |\zeta| = \infty$). As discussed earlier, there is a trade-off involved in the choice of R : choosing as large a value of R as possible gives a better asymptotic decay rate, but also a potentially large constant K . It is also clear that in the entire case the bound decays superexponentially.

We also mention that the choice of a disk can result in poor bounds, as it can give a crude estimate of the field of values. Better estimates can sometimes be obtained by replacing disks with rectangles: for instance, if good estimates for the smallest and largest eigenvalues of the real and imaginary parts H_1 and H_2 of A are known (see (4)), then one can approximate the rectangle of the Bendixson–Hirsch Theorem. This is a compact set containing $\mathcal{W}(A)$ and may be a much better estimate of $\mathcal{W}(A)$ than some disk containing the field of values. In [199] the authors show how these rectangles, combined with certain conformal mappings, may be useful in obtaining improved decay bounds in the case of the exponential of a matrix in

upper Hessenberg form, which in turn provides accurate error estimates for Krylov subspace approximations of the action of the matrix exponential on a given vector in the nonnormal case. We shall return to this topic in Sect. 4.1.

Further decay bounds for the entries of analytic functions of general nonnormal, nondiagonalizable band matrices based on Faber polynomials can be found in Sect. 3.7.

3.5 Bounds for Matrix Functions Defined by Integral Transforms

In the Hermitian positive definite case, the available decay bounds (see (15) and Theorem 6) for the inverse and the exponential of a band matrix are generally better than the general bounds (27) based on Bernstein's Theorem. This is not surprising: the bound (15) of Demko et al. exploits the fact that the best approximation error of $f(x) = x^{-1}$ is known exactly, and similarly very good error bounds are known for the polynomial approximation of $f(x) = e^{-x}$. On the other hand, Bernstein's Theorem is much more general since it applies to any analytic function and thus the bounds on the entries of $f(A)$ obtained from it are likely to be less sharp.

This observation suggests that improved bounds should be obtainable for those matrix functions that are related in some manner to the exponential and the inverse function. As it turns out, many of the most important functions that arise in applications can be expressed as integral transforms involving either the exponential or the resolvent (and as we know, these two functions are related through the Laplace transform).

Here we focus on two (overlapping) classes of functions: the *strictly completely monotonic functions* (associated with the *Laplace–Stieltjes transform*) and the *Markov functions* (associated with the *Cauchy–Stieltjes transform*). We first review some basic properties of these functions and the relationship between the two classes, following closely the treatment in [22].

Definition 3 Let f be defined in the interval (a, b) where $-\infty \leq a < b \leq +\infty$. Then, f is said to be *completely monotonic* in (a, b) if

$$(-1)^k f^{(k)}(x) \geq 0 \quad \text{for all } a < x < b \quad \text{and all } k = 0, 1, 2, \dots$$

Moreover, f is said to be *strictly completely monotonic* in (a, b) if

$$(-1)^k f^{(k)}(x) > 0 \quad \text{for all } a < x < b \quad \text{and all } k = 0, 1, 2, \dots$$

Here $f^{(k)}$ denotes the k th derivative of f , with $f^{(0)} \equiv f$. A classical result of Bernstein (see [201]) states that a function f is completely monotonic in $(0, \infty)$ if

and only if f is a Laplace–Stieltjes transform:

$$f(x) = \int_0^\infty e^{-\tau x} d\alpha(\tau), \quad (42)$$

where $\alpha(\tau)$ is nondecreasing and the integral in (42) converges for all $x > 0$. Furthermore, f is strictly completely monotonic in $(0, \infty)$ if it is completely monotonic there and the function $\alpha(\tau)$ has at least one positive point of increase, that is, there exists a $\tau_0 > 0$ such that $\alpha(\tau_0 + \delta) > \alpha(\tau_0)$ for any $\delta > 0$. Here we assume that $\alpha(\tau)$ is nonnegative and that the integral in (42) is a Riemann–Stieltjes integral.

Important examples of strictly completely monotonic functions include (see for instance [196]):

1. $f_1(x) = x^{-1} = \int_0^\infty e^{-x\tau} d\alpha(\tau)$ for $x > 0$, where $\alpha(\tau) = \tau$ for $\tau \geq 0$.
2. $f_2(x) = e^{-x} = \int_0^\infty e^{-x\tau} d\alpha(\tau)$ for $x > 0$, where $\alpha(\tau) = 0$ for $0 \leq \tau < 1$ and $\alpha(\tau) = 1$ for $\tau \geq 1$.
3. $f_3(x) = (1 - e^{-x})/x = \int_0^\infty e^{-x\tau} d\alpha(\tau)$ for $x > 0$, where $\alpha(\tau) = \tau$ for $0 \leq \tau \leq 1$, and $\alpha(\tau) = 1$ for $\tau \geq 1$.

Other examples include the functions $x^{-\sigma}$ (for any $\sigma > 0$), $\log(1 + 1/x)$ and $\exp(1/x)$, all strictly completely monotonic on $(0, \infty)$. Also, it is clear that products and positive linear combinations of strictly completely monotonic functions are strictly completely monotonic.

A closely related class of functions is given by the Cauchy–Stieltjes (or Markov-type) functions, which can be written as

$$f(z) = \int_\Gamma \frac{d\gamma(\omega)}{z - \omega}, \quad z \in \mathbb{C} \setminus \Gamma, \quad (43)$$

where γ is a (complex) measure supported on a closed set $\Gamma \subset \mathbb{C}$ and the integral is absolutely convergent. In this paper we are especially interested in the special case $\Gamma = (-\infty, 0]$ so that

$$f(x) = \int_{-\infty}^0 \frac{d\gamma(\omega)}{x - \omega}, \quad x \in \mathbb{C} \setminus (-\infty, 0], \quad (44)$$

where γ is now a (possibly signed) real measure. The following functions, which occur in various applications (see, e.g., [101] and references therein), fall into this class:

$$\begin{aligned} x^{-\frac{1}{2}} &= \int_{-\infty}^0 \frac{1}{x - \omega} \frac{1}{\pi \sqrt{-\omega}} d\omega, \\ \frac{e^{-t\sqrt{x}} - 1}{x} &= \int_{-\infty}^0 \frac{1}{x - \omega} \frac{\sin(t\sqrt{-\omega})}{-\pi \omega} d\omega, \\ \frac{\log(1 + x)}{x} &= \int_{-\infty}^{-1} \frac{1}{x - \omega} \frac{1}{(-\omega)} d\omega. \end{aligned}$$

The two classes of functions just introduced overlap. Indeed, it is easy to see (e.g., [150]) that functions of the form

$$f(x) = \int_0^\infty \frac{d\mu(\omega)}{x + \omega},$$

with μ a positive measure, are strictly completely monotonic on $(0, \infty)$; but every such function can also be written in the form

$$f(x) = \int_{-\infty}^0 \frac{d\gamma(\omega)}{x - \omega}, \quad \gamma(\omega) = -\mu(-\omega),$$

and therefore it is a Cauchy–Stieltjes function. We note, however, that the two classes do not coincide: e.g., $f(x) = \exp(-x)$ is strictly completely monotonic but is not a Cauchy–Stieltjes function. In the following, the term *Laplace–Stieltjes function* will be used to denote a function that is strictly completely monotonic on $(0, \infty)$.

If A is Hermitian and positive definite and f is a Laplace–Stieltjes function given by (42), we can write

$$f(A) = \int_0^\infty e^{-\tau A} d\alpha(\tau)$$

and therefore

$$|[f(A)]_{ij}| \leq \int_0^\infty |[e^{-\tau A}]_{ij}| d\alpha(\tau), \quad \forall i, j = 1, \dots, n.$$

We can now apply Theorem 6 on the off-diagonal decay behavior of $[e^{-\tau A}]_{ij}$ to bound the entries of $f(A)$. We thus obtain the following result.

Theorem 14 ([22]) *Let $A = A^*$ be m -banded and positive definite, and let $\widehat{A} = A - \lambda_{\min}(A)I$. Let $[0, 4\rho]$ be the smallest interval containing $\sigma(\widehat{A})$, and assume f is a Laplace–Stieltjes function. Then, with the notation and assumptions of Theorem 6 and for $\xi = \lceil |i-j|/m \rceil \geq 2$:*

$$\begin{aligned} |[f(A)]_{ij}| &\leq \int_0^\infty \exp(-\tau \lambda_{\min}(A)) |[\exp(-\tau \widehat{A})]_{ij}| d\alpha(\tau) \\ &\leq 10 \int_0^{\frac{\xi}{2\rho}} \exp(-\tau \lambda_{\min}(A)) \frac{\exp(-\rho\tau)}{\rho\tau} \left(\frac{e\rho\tau}{\xi} \right)^{\xi} d\alpha(\tau) \\ &\quad + 10 \int_{\frac{\xi}{2\rho}}^{\frac{\xi^2}{4\rho}} \exp(-\tau \lambda_{\min}(A)) \exp\left(-\frac{\xi^2}{5\rho\tau}\right) d\alpha(\tau) \\ &\quad + \int_{\frac{\xi^2}{4\rho}}^\infty \exp(-\tau \lambda_{\min}(A)) |[\exp(-\tau \widehat{A})]_{ij}| d\alpha(\tau) = I + II + III. \end{aligned} \tag{45}$$

The nature of these bounds (45) is quite different from the ones previously seen, since they are given only implicitly by the integrals *I*, *II* and *III*. Note that the latter integral does not significantly contribute provided that $|i - j|$ is sufficiently large while ρ and m are not too large. In general, it will be necessary to evaluate these integrals numerically; in some special cases it may be possible to find explicit bounds for the various terms in (45). On the other hand, these bounds are much more accurate, in general, than those based on Bernstein's Theorem. We refer to [22] for numerical examples illustrating the tightness of these bounds.

Suppose now that f is a Cauchy–Stieltjes function of the form (44), then for any Hermitian positive definite matrix A we can write

$$f(A) = \int_{-\infty}^0 (A - \omega I)^{-1} d\gamma(\omega),$$

Since $A - \omega I$ is Hermitian positive definite, if A is banded (or sparse) we can apply the bounds (15) of Demko et al. to the resolvent $(A - \omega I)^{-1}$ in order to derive bounds for the entries of $f(A)$.

For a given $\omega \in \Gamma = (-\infty, 0)$, let $\kappa = \kappa(\omega) = (\lambda_{\max}(A) - \omega)/(\lambda_{\min}(A) - \omega)$, $q = q(\omega) = (\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$, $C = C(-\omega) = \max\{1/(\lambda_{\min}(A) - \omega), C_0\}$, with $C_0 = C_0(-\omega) = (1 + \sqrt{\kappa})^2/(2(\lambda_{\max}(A) - \omega))$. We have the following result.

Theorem 15 ([22]) *Let $A = A^*$ be positive definite and let f be a Cauchy–Stieltjes function of the form (44), where γ is of bounded variation on $(-\infty, 0]$. Then for all i and j we have*

$$|[f(A)]_{ij}| \leq \int_{-\infty}^0 |[(A - \omega I)^{-1}]_{ij}| |\mathrm{d}\gamma(\omega)| \leq \int_{-\infty}^0 C(\omega)q(\omega)^{\frac{|i-j|}{m}} |\mathrm{d}\gamma(\omega)|. \quad (46)$$

We remark that the first inequality in (46) is a consequence of the assumptions made on γ ; see [201, Chap. I].

A natural question is the following: if f is both a Laplace–Stieltjes and a Cauchy–Stieltjes function, how do the bounds (45) and (46) compare? Is one better than the other? The answer is likely to depend on the particular function considered. However, (46) tends to lead to more explicit and more accurate bounds for some important functions, like $f(x) = x^{-1/2}$ (which is both a Laplace–Stieltjes and a Cauchy–Stieltjes function); see [22].

Remark 7 As always, all the decay bounds discussed in this section can be formulated for a sequence $\{A_n\}$ of Hermitian positive definite matrices of increasing size n as long as they are all banded (with bandwidth uniformly bounded by m) and such that the norms $\|A_n\|_2$ are uniformly bounded with respect to n . In this case, the decay bounds (45) and (46) hold uniformly in n . Moreover, the bounds can be modified to accommodate more general sparsity patterns, and can be applied to sequences of sparse matrices of increasing size under the bounded maximum degree assumption (11).

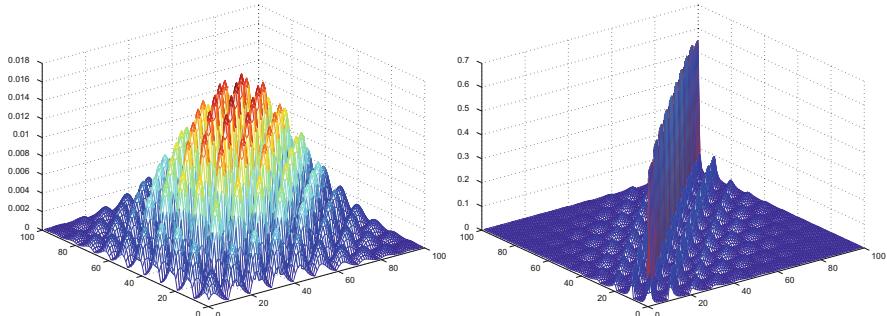


Fig. 10 Oscillatory behavior of $\exp(-5A)$ and $A^{-1/2}$ where A is a discrete Laplacian on a 10×10 grid

Next, we briefly discuss the case of matrices that are Kronecker sums of banded or sparse matrices, again following the treatment in [22]. Recall that the Kronecker sum of two matrices $T_1 \in \mathbb{C}^{n \times n}$ and $T_2 \in \mathbb{C}^{m \times m}$ is defined as the $nm \times nm$ matrix

$$A = T_1 \oplus T_2 = T_1 \otimes I_m + I_n \otimes T_2.$$

A familiar example is given by the 5-point finite difference discretization of the Dirichlet Laplacian on a rectangular region. In this case T_1 and T_2 are tridiagonal, and A is block tridiagonal.

Suppose now that $f(A)$ is defined, then numerical experiments reveal that the entries of $f(A)$ decay in an oscillatory manner. In Fig. 10 we illustrate this behavior for two different matrix functions, the exponential e^{-tA} for $t = 5$ and the inverse square root $A^{-1/2}$. Here A is 100×100 and represents the 5-point finite difference discretization of the Dirichlet Laplacian on the unit square.

The question arises of whether such oscillatory behavior can be accurately captured in the form of non-monotonic decay bounds. One possibility is to treat A as a general sparse matrix, and to use the graph distance to measure decay. However, this approach does not exploit the relationship (25) between the exponential of A and the exponentials of T_1 and T_2 ; since we have very good bounds for the decay in the exponential of a banded matrix, it should be possible to derive similarly good bounds on the entries of the exponential of A by making use of (25). Moreover, suppose that f is a Laplace–Stieltjes functions and that $A = T_1 \oplus T_2$ (with T_1, T_2 Hermitian positive definite), such as $f(A) = A^{-1/2}$. Then we can write

$$f(A) = \int_0^\infty \exp(-\tau A) d\alpha(\tau) = \int_0^\infty \exp(-\tau T_1) \otimes \exp(-\tau T_2) d\alpha(\tau) \quad (47)$$

for a suitable choice of $\alpha(\tau)$. Now (47) can be used to obtain bounds on the entries of $f(A)$ in terms of integrals involving the entries of $\exp(-\tau T_1)$ and $\exp(-\tau T_2)$, for which accurate bounds are available. It has been shown in [22] that these bounds are

generally better than the ones obtained when A is treated as a general sparse matrix. A similar approach can also be used in the case of Cauchy–Stieltjes functions; see [22] for details. These results can be extended to the case where A is the Kronecker sum of an arbitrary number of terms.

3.6 Functions of Structured Matrices

Up to now we have considered rather general classes of banded or sparse matrices, and (apart from the case where A is a Kronecker sum) we have not taken into account possible additional structure present in A . The question arises whether more can be said about the structural and decay properties of $f(A)$ when A is restricted to a specific class of structured matrices.

The simplest nontrivial example is perhaps that of circulant matrices [63]. Since any circulant $n \times n$ matrix with complex entries is of the form $A = F^* \Lambda F$, where Λ is diagonal and F is the (unitary) discrete Fourier transform matrix, we have that $f(A) = F^* f(\Lambda) F$ is also circulant. More generally, if A belongs to a subalgebra $\mathcal{A} \subseteq \mathbb{C}^{n \times n}$, then so does $f(A)$. Clearly, this poses strong constraints on the decay pattern of $f(A)$.

What if A is not circulant, but Toeplitz? Since Toeplitz matrices do not form an algebra, it is not generally true that $f(A)$ is Toeplitz if A is. Banded Toeplitz and block Toeplitz matrices arise in many important applications (see for example [40]), but relatively little has been done in the study of functions of banded Toeplitz and block Toeplitz matrices. To our knowledge, most of the results in this area are found in [100], in which the authors study the structure of functions of banded Hermitian block Toeplitz matrices $A_n \in \mathbb{C}^{nN \times nN}$ in the limit as $n \rightarrow \infty$ (with N , the block size, being fixed). In this limit, $f(A_n)$ is asymptotically “close” to being a block Toeplitz matrix, in a precise sense. However, there is no explicit discussion of decay in [40].

Another very interesting example is that of functions of finite difference matrices (approximations of differential operators), which in [184] are shown to have a “Toeplitz-plus-Hankel” structure. Again, this fact imposes constraints on the decay behavior of the entries of $f(A)$.

Finally, it can happen that A and $f(A)$ belong to different, but related structures. For example, if A is skew-Hermitian, the exponential e^A is unitary. Hence, the exponential map takes elements of the *Lie algebra* of skew-Hermitian matrices to elements of the corresponding *Lie group*, the unitary matrices. Many more such examples are given in [107, Sect. 14.1.1]. Exploiting these structural properties may lead to improved bounds for the entries of $f(A)$; to our knowledge, however, this possibility has not been explored so far.

3.7 Some Generalizations

So far we have only considered matrices over \mathbb{R} or \mathbb{C} . In applications (especially in physics) it is sometimes necessary to consider functions of matrices over more general algebraic and topological structures. Depending on the problem, these could be non-commutative division algebras, algebras of operators on a Hilbert space (finite or infinite dimensional), algebras of continuous functions, etc.

The question arises then whether decay bounds such as those discussed up to now can be extended to these more general situations. The answer, as shown in [19], is largely affirmative. The most natural tool for carrying out the desired extension is the general theory of *complex C^* -algebras*. In particular, the *holomorphic functional calculus* allows one to define the notion of analytic function on such algebras, and to develop a theory of functions of matrices over such algebras. In this setting, almost all⁸ the decay bounds described so far can be extended *verbatim*, the only difference being that the absolute value of $[f(A)]_{ij}$ must now be replaced by $\|[f(A)]_{ij}\|$, where $\|\cdot\|$ is the norm in the underlying C^* -algebra.

We proceed now to sketch the generalization of some of the decay bounds. The following discussion is based on [19]; for an excellent introduction to the basic theory of C^* -algebras, see [120].

Recall that a *Banach algebra* is a complex algebra \mathbb{A} with a norm making \mathbb{A} into a Banach space and satisfying

$$\|ab\| \leq \|a\| \|b\|$$

for all $a, b \in \mathbb{A}$. In this paper we consider only *unital* Banach algebras, i.e., algebras with a multiplicative unit I with $\|I\| = 1$.

An *involution* on a Banach algebra \mathbb{A} is a map $a \mapsto a^*$ of \mathbb{A} into itself satisfying

- (i) $(a^*)^* = a$
- (ii) $(ab)^* = b^*a^*$
- (iii) $(\lambda a + b)^* = \bar{\lambda}a^* + b^*$

for all $a, b \in \mathbb{A}$ and $\lambda \in \mathbb{C}$. A *C^* -algebra* is a Banach algebra with an involution such that the *C^* -identity*

$$\|a^*a\| = \|a\|^2$$

holds for all $a \in \mathbb{A}$. Note that we do not make any assumption on whether \mathbb{A} is commutative or not. Basic examples of C^* -algebras are:

1. The commutative algebra $\mathcal{C}(\mathcal{X})$ of all continuous complex-valued functions on a compact Hausdorff space \mathcal{X} . Here the addition and multiplication operations

⁸The exception is given by the bounds involving the condition number of the eigenvector matrix, see Theorem 11.

are defined pointwise, and the norm is given by $\|f\|_\infty = \max_{t \in \mathcal{X}} |f(t)|$. The involution on $\mathcal{C}(\mathcal{X})$ maps each function f to its complex conjugate f^* , defined by $f^*(t) = \overline{f(t)}$ for all $t \in \mathcal{X}$.

2. The algebra $\mathcal{B}(\mathcal{H})$ of all bounded linear operators on a complex Hilbert space \mathcal{H} , with the operator norm $\|T\| = \sup \|T\mathbf{x}\|_{\mathcal{H}} / \|\mathbf{x}\|_{\mathcal{H}}$, where the supremum is taken over all nonzero $\mathbf{x} \in \mathcal{H}$. The involution on $\mathcal{B}(\mathcal{H})$ maps each bounded linear operator T on \mathcal{H} to its adjoint, T^* .

Note that the second example contains as a special case the algebra $M_k(\mathbb{C}) (= \mathbb{C}^{k \times k})$ of all $k \times k$ matrices with complex entries, with the norm being the usual spectral norm and the involution mapping each matrix $A = [a_{ij}]$ to its Hermitian conjugate $A^* = [\overline{a_{ji}}]$. This algebra is noncommutative for $k \geq 2$.

Examples 1 and 2 above provide, in a precise sense, the “only” examples of C^* -algebras. Indeed, every (unital) commutative C^* -algebra admits a faithful representation onto an algebra of the form $\mathcal{C}(\mathcal{X})$ for a suitable (and essentially unique) compact Hausdorff space \mathcal{X} ; and, similarly, every unital (possibly noncommutative) C^* -algebra can be faithfully represented as a norm-closed subalgebra of $\mathcal{B}(\mathcal{H})$ for a suitable complex Hilbert space \mathcal{H} .

More precisely, a map ϕ between two C^* -algebras is a *$*$ -homomorphism* if ϕ is linear, multiplicative, and such that $\phi(a^*) = \phi(a)^*$; a *$*$ -isomorphism* is a bijective $*$ -homomorphism. Two C^* -algebras are said to be *isometrically $*$ -isomorphic* if there is a norm-preserving $*$ -isomorphism between them, in which case they are indistinguishable as C^* -algebras. A *$*$ -subalgebra* \mathbb{B} of a C^* -algebra \mathbb{A} is a subalgebra that is $*$ -closed, i.e., $a \in \mathbb{B}$ implies $a^* \in \mathbb{B}$. Finally, a C^* -subalgebra is a norm-closed $*$ -subalgebra of a C^* -algebra. The following two results are classical [85, 86].

Theorem 16 (Gelfand) *Let \mathbb{A} be a commutative C^* -algebra. Then there is a compact Hausdorff space \mathcal{X} such that \mathbb{A} is isometrically $*$ -isomorphic to $\mathcal{C}(\mathcal{X})$. If \mathcal{Y} is another compact Hausdorff space such that \mathbb{A} is isometrically $*$ -isomorphic to $\mathcal{C}(\mathcal{Y})$, then \mathcal{X} and \mathcal{Y} are necessarily homeomorphic.*

Theorem 17 (Gelfand–Naimark) *Let \mathbb{A} be a C^* -algebra. Then there is a complex Hilbert space \mathcal{H} such that \mathbb{A} is isometrically $*$ -isomorphic to a C^* -subalgebra of $\mathcal{B}(\mathcal{H})$.*

We will also need the following definitions and facts. An element a of a C^* -algebra is *unitary* if $aa^* = a^*a = I$, *Hermitian* (or *self-adjoint*) if $a^* = a$, *skew-Hermitian* if $a^* = -a$, *normal* if $aa^* = a^*a$. Clearly, unitary, Hermitian and skew-Hermitian elements are all normal. Any element a in a C^* -algebra can be written uniquely as $a = h_1 + i h_2$ with h_1, h_2 Hermitian and $i = \sqrt{-1}$.

For any (complex) Banach algebra \mathbb{A} , the *spectrum* of an element $a \in \mathbb{A}$ is the set of all $\lambda \in \mathbb{C}$ such that $\lambda I - a$ is not invertible in \mathbb{A} . We denote the spectrum of a by $\sigma(a)$. For any $a \in \mathbb{A}$, the spectrum $\sigma(a)$ is a non-empty compact subset of \mathbb{C} contained in the closed disk of radius $r = \|a\|$ centered at 0. The *spectral radius* of a is defined as $\varrho(a) = \max\{|\lambda| : \lambda \in \sigma(A)\}$. *Gelfand’s formula* for the spectral

radius [85] states that

$$\varrho(a) = \lim_{m \rightarrow \infty} \|a^m\|^{\frac{1}{m}}. \quad (48)$$

Note that this identity contains the statement that the above limit exists.

If $a \in \mathbb{A}$ (a C*-algebra) is Hermitian, $\sigma(a)$ is a subset of \mathbb{R} . If $a \in \mathbb{A}$ is normal (in particular, Hermitian), then $\varrho(a) = \|a\|$. This implies that if a is Hermitian, then either $-\|a\| \in \sigma(a)$ or $\|a\| \in \sigma(a)$. The spectrum of a skew-Hermitian element is purely imaginary, and the spectrum of a unitary element is contained in the unit circle $\mathbb{T} = \{z \in \mathbb{C} : |z| = 1\}$.

An element $a \in \mathbb{A}$ is *nonnegative* if $a = a^*$ and the spectrum of a is contained in \mathbb{R}_+ , the nonnegative real axis; a is *positive* if $\sigma(a) \subset (0, \infty)$. Any linear combination with real nonnegative coefficients of nonnegative elements of a C*-algebra is nonnegative; in other words, the set of all nonnegative elements in a C*-algebra \mathbb{A} form a (nonnegative) *cone* in \mathbb{A} . For any $a \in \mathbb{A}$, aa^* is nonnegative, and $I + aa^*$ is invertible in \mathbb{A} . Furthermore, $\|a\| = \sqrt{\varrho(a^*a)} = \sqrt{\varrho(aa^*)}$, for any $a \in \mathbb{A}$.

Note that if $\|\cdot\|_*$ and $\|\cdot\|_{**}$ are two norms with respect to which \mathbb{A} is a C*-algebra, then $\|\cdot\|_* = \|\cdot\|_{**}$.

Let \mathbb{A} be a C*-algebra. Given a positive integer n , let $\mathbb{A}^{n \times n} = \mathcal{M}_n(\mathbb{A})$ be the set of $n \times n$ matrices with entries in \mathbb{A} . Observe that $\mathbb{A}^{n \times n}$ has a natural C*-algebra structure, with matrix addition and multiplication defined in the usual way (in terms, of course, of the corresponding operations on \mathbb{A}). The involution is naturally defined as follows: given a matrix $A = [a_{ij}] \in \mathbb{A}^{n \times n}$, the adjoint of A is given by $A^* = [a_{ji}^*]$. The algebra $\mathbb{A}^{n \times n}$ is obviously unital, with unit

$$I_n = \begin{bmatrix} I & 0 & \dots & \dots & 0 \\ 0 & I & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & I & 0 \\ 0 & \dots & \dots & 0 & I \end{bmatrix}$$

where I is the unit of \mathbb{A} . The definition of unitary, Hermitian, skew-Hermitian and normal matrix are the obvious ones.

It follows from the Gelfand–Naimark Representation Theorem (Theorem 17 above) that each $A \in \mathbb{A}^{n \times n}$ can be represented as a matrix T_A of bounded linear operators, where T_A acts on the direct sum $\mathcal{H} = \mathcal{H} \oplus \cdots \oplus \mathcal{H}$ of n copies of a suitable complex Hilbert space \mathcal{H} . This fact allows us to introduce an operator norm on $\mathbb{A}^{n \times n}$, defined as follows:

$$\|A\| := \sup_{\|\mathbf{x}\|_{\mathcal{H}}=1} \|T_A \mathbf{x}\|_{\mathcal{H}}, \quad (49)$$

where

$$\|\mathbf{x}\|_{\mathcal{H}} := \sqrt{\|\mathbf{x}_1\|_{\mathcal{H}}^2 + \cdots + \|\mathbf{x}_n\|_{\mathcal{H}}^2}$$

is the norm of an element $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathcal{H}$. Relative to this norm, $\mathbb{A}^{n \times n}$ is a C*-algebra. Note that $\mathbb{A}^{n \times n}$ can also be identified with the tensor product of C*-algebras $\mathbb{A} \otimes \mathcal{M}_n(\mathbb{C})$.

Similarly, Gelfand's Theorem (Theorem 16 above) implies that if \mathbb{A} is commutative, there is a compact Hausdorff space \mathcal{X} such that any $A \in \mathbb{A}^{n \times n}$ can be identified with a continuous matrix-valued function

$$A : \mathcal{X} \longrightarrow \mathcal{M}_n(\mathbb{C}).$$

In other words, A can be represented as an $n \times n$ matrix of continuous, complex-valued functions: $A = [a_{ij}(t)]$, with domain \mathcal{X} . The natural C*-algebra norm on $\mathbb{A}^{n \times n}$, which can be identified with $\mathcal{C}(\mathcal{X}) \otimes \mathcal{M}_n(\mathbb{C})$, is now the operator norm

$$\|A\| := \sup_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|, \quad (50)$$

where $\mathbf{x} = (x_1, \dots, x_n) \in [\mathcal{C}(\mathcal{X})]^n$ has norm $\|\mathbf{x}\| = \sqrt{\|x_1\|_\infty^2 + \cdots + \|x_n\|_\infty^2}$ with $\|x_i\|_\infty = \max_{t \in \mathcal{X}} |x_i(t)|$, for $1 \leq i \leq n$.

Since $\mathbb{A}^{n \times n}$ is a C*-algebra, all the definitions and basic facts about the spectrum remain valid for any matrix A with entries in \mathbb{A} . Thus, the spectrum $\sigma(A)$ of $A \in \mathbb{A}^{n \times n}$ is the set of all $\lambda \in \mathbb{C}$ such that $\lambda I_n - A$ is not invertible in $\mathbb{A}^{n \times n}$. The set $\sigma(A)$ is a nonempty compact subset of \mathbb{C} completely contained in the disk of radius $\|A\|$ centered at 0. The definition of spectral radius and Gelfand's formula (48) remain valid. Hermitian matrices have real spectra, skew-Hermitian matrices have purely imaginary spectra, unitary matrices have spectra contained in \mathbb{T} , and so forth. Note, however, that it is not true in general that a normal matrix A over a C*-algebra can be unitarily diagonalized [119].

The standard way to define the notion of an analytic function $f(a)$ of an element a of a C*-algebra \mathbb{A} is via contour integration. In particular, we can use this approach to define functions of a matrix A with elements in \mathbb{A} .

Let $f(z)$ be a complex function which is analytic in an open neighborhood U of $\sigma(a)$. Since $\sigma(a)$ is compact, we can always find a finite collection $\Gamma = \cup_{j=1}^\ell \gamma_j$ of smooth simple closed curves whose interior parts contain $\sigma(a)$ and entirely contained in U . The curves γ_j are assumed to be oriented counterclockwise.

Then $f(a) \in \mathbb{A}$ can be defined as

$$f(a) = \frac{1}{2\pi i} \int_{\Gamma} f(z)(zI - a)^{-1} dz, \quad (51)$$

where the line integral of a Banach-space-valued function g defined on a smooth curve $\gamma : t \mapsto z(t)$ for $t \in [0, 1]$ is given by the norm limit of Riemann sums of the

form

$$\sum_{j=1}^v g(z(\theta_j)) [z(t_j) - z(t_{j-1})], \quad t_{j-1} \leq \theta_j \leq t_j,$$

where $0 = t_0 < t_1 < \dots < t_{v-1} < t_v = 1$.

Denote by $\mathcal{H}(a)$ the algebra of analytic functions whose domain contains an open neighborhood of $\sigma(a)$. The following well known result is the basis for the *holomorphic functional calculus*; see, e.g., [120, p. 206].

Theorem 18 *The mapping $\mathcal{H}(a) \rightarrow \mathbb{A}$ defined by $f \mapsto f(a)$ is an algebra homomorphism, which maps the constant function 1 to $I \in \mathbb{A}$ and maps the identity function to a . If $f(z) = \sum_{j=0}^{\infty} c_j z^j$ is the power series representation of $f \in \mathcal{H}(a)$ over an open neighborhood of $\sigma(a)$, then we have*

$$f(a) = \sum_{j=0}^{\infty} c_j a^j.$$

Moreover, the following version of the *spectral theorem* holds:

$$\sigma(f(a)) = f(\sigma(a)). \quad (52)$$

If a is normal, the following properties also hold:

1. $\|f(a)\| = \|f\|_{\infty, \sigma(a)} := \max_{\lambda \in \sigma(a)} |f(\lambda)|$;
2. $\overline{f(a)} = [f(a)]^*$; in particular, if a is Hermitian then $f(a)$ is also Hermitian if and only if $f(\sigma(a)) \subset \mathbb{R}$;
3. $f(a)$ is normal;
4. $f(a)b = bf(a)$ whenever $b \in \mathbb{A}$ and $ab = ba$.

Obviously, these definitions and results apply in the case where a is a matrix A with entries in a C^* -algebra \mathbb{A} . In particular, if $f(z)$ is analytic on a neighborhood of $\sigma(A)$, we define $f(A)$ via

$$f(A) = \frac{1}{2\pi i} \int_{\Gamma} f(z)(zI_n - A)^{-1} dz, \quad (53)$$

with the obvious meaning of Γ .

The holomorphic functional calculus allows us to generalize most of the decay results that hold for analytic functions of matrices with entries in \mathbb{C} to functions of matrices with entries in an arbitrary C^* -algebra \mathbb{A} almost without changes. The fact that finite matrices over \mathbb{C} have finite spectra whereas matrices over a general C^* -algebra \mathbb{A} have in general continuous spectra makes no difference whatsoever; note that we have already encountered this situation when we discussed the case of functions of bounded infinite matrices. To see that the same arguments used for

matrices over \mathbb{C} carry over almost *verbatim* to this more general setting, consider for example an m -banded Hermitian matrix $A \in \mathbb{A}^{n \times n}$. Then $\sigma(A) \subseteq [a, b] \subset \mathbb{R}$ for some a, b with $-\infty < a < b < \infty$. Up to scaling and shift, we can assume that $[a, b] = \mathcal{I} = [-1, 1]$. Let f be analytic on a region $\Omega \subseteq \mathbb{C}$ containing \mathcal{I} and let \mathbb{P}_k denote the set of all complex polynomials of degree at most k on \mathcal{I} . Given $p \in \mathbb{P}_k$, the matrix $p(A) \in \mathbb{A}^{n \times n}$ is well defined and it is banded with bandwidth at most km . So for any polynomial $p \in \mathbb{P}_k$ and any pair of indices i, j such that $|i - j| > km$ we have

$$\|[f(A)]_{ij}\| = \|[f(A) - p(A)]_{ij}\| \quad (54)$$

$$\leq \|f(A) - p(A)\| \quad (55)$$

$$= \varrho(f(A) - p(A)) \quad (56)$$

$$= \max(\sigma(f(A) - p(A))) = \max(\sigma((f - p)(A))) \quad (57)$$

$$= \max((f - p)(\sigma(A))) \leq E_k(f, \mathcal{I}), \quad (58)$$

where $E_k(f, \mathcal{I})$ is the best uniform approximation error for the function f on the interval \mathcal{I} using polynomials of degree at most k :

$$E_k(f, \mathcal{I}) := \min_{p \in \mathbb{P}_k} \max_{x \in \mathcal{I}} |f(x) - p(x)|.$$

In the above derivation we made use of the definition (49), of the fact that $A = A^*$, and of the spectral theorem (52), valid for normal elements of any C*-algebra. We can now apply Bernstein's Theorem to bound $E_k(f, \mathcal{I})$ in terms of ellipses contained in Ω having foci at $-1, 1$. From this we can deduce exponentially decaying bounds for $\|[f(A)]_{ij}\|$ with respect to $|i - j|$ in the usual manner:

$$\|[f(A)]_{ij}\| \leq K \chi^{-\frac{|i-j|}{m}} = K \rho^{|i-j|}, \quad \rho = \chi^{-\frac{1}{m}}, \quad (59)$$

where $K = 2M(\chi)/(\chi - 1)$, $M(\chi) = \max_{z \in \mathcal{E}_\chi} |f(z)|$.

Analogous decay bounds can be derived in the normal case, without any changes to the proofs. The same is true for the general, nonnormal case, with one caveat: the notion of field of values is not well-defined, in general, for an element of a C*-algebra. One can attempt to define the field of values of an element a of a C*-algebra \mathbb{A} by making use of the Gelfand–Naimark Representation Theorem (Theorem 17): since there is an isometric $*$ -isomorphism ϕ from \mathbb{A} into the algebra $\mathcal{B}(\mathcal{H})$ for some complex Hilbert space \mathcal{H} , we could define the field of values of a as the field of values of the bounded linear operator $T_a = \phi(a)$, i.e.,

$$\mathcal{W}(a) = \mathcal{W}(T_a) = \{\langle T_a \mathbf{x}, \mathbf{x} \rangle \mid \mathbf{x} \in \mathcal{H}, \|\mathbf{x}\| = 1\}.$$

Unfortunately, ϕ is not unique and it turns out that different choices of ϕ may give rise to different fields of values. Fortunately, however, the *closure* of the field of

values is independent of the choice of representation [28]. Hence, if we replace the field of values $\mathcal{W}(T_a)$ with the closure $\overline{\mathcal{W}(T_a)}$, everything works as in the “classical” case.⁹

In order to achieve the desired generalization, we make use of the following theorem of Crouzeix, which is an extension of Theorem 12. Given a set $E \subset \mathbb{C}$, denote by $\mathcal{H}_b(E)$ the algebra of continuous and bounded functions in \overline{E} which are analytic in the interior of E . Furthermore, for $T \in \mathcal{B}(\mathcal{H})$ let $\|p\|_{\infty,T} := \max_{z \in \overline{\mathcal{W}(T)}} |p(z)|$. Then we have [57, Theorem 2]:

Theorem 19 *For any bounded linear operator $T \in \mathcal{B}(\mathcal{H})$ the homomorphism $p \mapsto p(T)$ from the algebra $\mathbb{C}[z]$, with norm $\|\cdot\|_{\infty,T}$, into the algebra $\mathcal{B}(\mathcal{H})$, is bounded with constant \mathcal{Q} . It admits a unique bounded extension from $\mathcal{H}_b(\mathcal{W}(T))$ into $\mathcal{B}(\mathcal{H})$. This extension is also bounded with constant \mathcal{Q} .*

Making use of the notion of field of values for elements of a C^* -algebra, we obtain the following corollary.

Corollary 1 *Given $A \in \mathbb{A}^{n \times n}$, the following bound holds for any complex function g analytic on a neighborhood of $\overline{\mathcal{W}(A)}$:*

$$\|g(A)\| \leq \mathcal{Q} \|g\|_{\infty,A} = \mathcal{Q} \max_{z \in \overline{\mathcal{W}(A)}} |g(z)|.$$

In order to obtain bounds on $\|[f(A)]_{ij}\|$, where the function $f(z)$ can be assumed to be analytic on an open set $S \supset \overline{\mathcal{W}(A)}$, we can choose $g(z)$ in Corollary 1 as $f(z) - p_k(z)$, where $p_k(z)$ is any complex polynomial of degree bounded by k . The argument in (54)–(58) can then be adapted as follows:

$$\|[f(A)]_{ij}\| = \|[f(A) - p_k(A)]_{ij}\| \quad (60)$$

$$\leq \|f(A) - p_k(A)\| \quad (61)$$

$$\leq \mathcal{Q} \|f - p_k\|_{\infty,A} \quad (62)$$

$$= \mathcal{Q} \max_{z \in \overline{\mathcal{W}(A)}} |f(z) - p_k(z)| \quad (63)$$

$$\leq \mathcal{Q} E_k(f, \overline{\mathcal{W}(A)}), \quad (64)$$

where $E_k(f, \overline{\mathcal{W}(A)})$ is the degree k best approximation error for f on the compact set $\overline{\mathcal{W}(A)}$. In order to make explicit computations easier, we may of course replace $\mathcal{W}(A)$ with a larger but more manageable set in the above argument.

Putting everything together, we obtain the following generalization of Theorem 13:

⁹Recall that for $A \in \mathbb{C}^{n \times n}$ the field of values $\mathcal{W}(A)$ is compact and therefore always closed.

Theorem 20 ([19]) *Let $\{A_n\}$ be a sequence of matrices of increasing size over a complex C^* -algebra \mathbb{A} with bandwidths uniformly bounded by m . Let the complex function f be analytic on a neighborhood of a connected compact set $\mathcal{C} \subset \mathbb{C}$ containing $\overline{\mathcal{W}(A_n)}$ for all n . Then there exist explicitly computable constants $K > 0$ and $\alpha > 0$, independent of n , such that*

$$\|[f(A_n)]_{ij}\| \leq K e^{-\alpha|i-j|}$$

for all indices i, j and for all $n \in \mathbb{N}$.

As always, analogous results hold for more general sparse matrices, with the geodesic distance on the matrix graphs $\mathcal{G}(A_n)$ replacing the distance from the main diagonal, as long as the bounded maximum degree condition (11) holds. Also, if f is entire, then the entries of $f(A_n)$ are bounded in a superexponentially decaying manner.

As a consequence of this extension, the decay bounds apply directly to functions of block-banded matrices (with blocks all of the same size), to functions of banded or sparse matrices of operators on a complex Hilbert space, and to functions of matrices the entries of which are complex-valued continuous functions. In [19] one can also find the results of numerical and symbolic computations illustrating the decay in $f(A)$ where A is a banded (in particular, tridiagonal) matrix over the function algebra $C[0, 1]$ endowed with the infinity norm, showing the rapid decay of $\|[f(A)]_{ij}\|_\infty$ for increasing $|i - j|$.

In [19] it is further shown that the theory can be extended to cover analytic functions of matrices with entries in the *real* C^* -algebra \mathbb{H} of quaternions, as long as these functions have power series expansions with real coefficients.

3.7.1 The Time-Ordered Exponential

In mathematical physics, the *time-ordered exponential* $OE[A]$ associated with a given time-dependent matrix $A = A(t) = [A(t)]_{ij}$, $t \in [a, b] \subset \mathbb{R}$, is defined as the unique solution to the system of ordinary differential equations

$$\frac{d}{dt'} OE[A](t', t) = A(t') OE[A](t', t)$$

such that $OE[A](t, t) = I$ for all $t \in [a, b]$. Hence, the time-ordered exponential, also denoted by

$$OE[A](t', t) = \mathcal{T} \exp \left(\int_t^{t'} A(\tau) d\tau \right),$$

with \mathcal{T} the *time-ordering operator* (see, e.g., [129]), provides a way to express the solution of a linear first-order system of ordinary differential equations with variable

coefficients. In the case of a constant A , the time-ordered exponential reduces to the usual matrix exponential: $OE[A](t', t) = e^{(t'-t)A}$, where we assume $t' > t$.

When $A(t) \neq A(t')$ for $t \neq t'$, no simple, explicit expression is available for the time-ordered exponential. Techniques for evaluating $OE[A](t', t)$ (or its action) have been studied in, e.g., [88]. In that paper the authors also study the decay properties of the time-ordered exponential for the case of a sparse $A(t)$. Note that $OE[A](t', t)$ cannot be expressed in terms of contour integration, power series expansion, or other such device, therefore techniques different from those employed so far must be employed. The authors of [88] assume that $A = A(t)$ is a possibly infinite sparse matrix satisfying

$$M := \sup_{t \in [a,b]} \max_{i,j} |[A(t)]_{ij}| < \infty,$$

and that the nonzero pattern of $A(t)$ does not depend on t . The following result holds.

Theorem 21 ([88]) *If $d = d(i,j)$ is the geodesic distance between nodes i and j in the graph $\mathcal{G}(A)$, then the following bound holds for all i and j and for $t' > t$:*

$$\left| [OE[A](t', t)]_{ij} \right| \leq \sum_{k=d}^{\infty} \frac{M^k (t' - t)^k}{k!} W_{i,j;k}, \quad (65)$$

where $W_{i,j;k}$ is the number of walks of length k in $\mathcal{G}(A)$ between node i and node j . If Δ , the maximum degree of any node in $\mathcal{G}(A)$, is finite, then we also have the weaker bound

$$\left| [OE[A](t', t)]_{ij} \right| \leq e^{\Delta M(t' - t)} \frac{(\Delta M(t' - t))^d}{d!}. \quad (66)$$

The bound (66) decays superexponentially with the distance $d(i,j)$. The same result can be restated for a sequence of sparse time-dependent matrices $\{A_n(t)\}$ of increasing size such that $\sup_n \sup_{t \in [a,b]} \max_{i,j} |[A_n(t)]_{ij}| < \infty$, as long as the corresponding graphs $\mathcal{G}(A_n)$ satisfy the bounded maximum degree assumption. In this case a bound of the type (66) holds uniformly in n . In [88] it is also shown by example that the superexponential decay fails, in general, if $\Delta = \infty$.

3.8 Decal Algebras

Although so far our main emphasis has been on exponential decay, other types of decay occur frequently in applications. These different decay rates lead to the definition of various *decay algebras*, which are Banach algebras of infinite matrices, the entries of which satisfy different decay conditions.

Seminal works on decay algebras are the already cited paper by Jaffard [117] and papers by Baskakov [12, 13]. A key question addressed in these papers is that of inverse-closedness (see footnote 4). This question is highly relevant for us in view of the fact that the techniques covered so far all assume bandedness or sparsity of A in order to make statements about the decay behavior in A^{-1} or in more general matrix functions $f(A)$, but they are not applicable if A is a full matrix satisfying a decay condition. As noted in [97], if \mathcal{A} and \mathcal{B} are Banach algebras with $\mathcal{A} \subseteq \mathcal{B}$ and \mathcal{A} is inverse-closed in \mathcal{B} , then using the contour integral definition of a matrix function

$$f(A) = \frac{1}{2\pi i} \int_{\Gamma} f(z)(zI - A)^{-1} dz \quad (67)$$

(where the integral of a Banach-space-valued function has been defined in the previous section) we immediately obtain that $f(A) \in \mathcal{A}$ if $A \in \mathcal{A}$. Therefore, the entries of $f(A)$ must satisfy the same decay bound as those of A itself. Hence, inverse-closedness provides a powerful tool to establish the decay properties in the entries of $f(A)$ when A is not just a sparse or banded matrix, but more generally a matrix with certain types of decay. We emphasize that this approach is completely different from the techniques reviewed earlier, which are largely based on classical results on the approximation of analytic functions with polynomials.

Results on inverse-closedness of decay algebras can be regarded as noncommutative variants of *Wiener's Lemma*: if a periodic function f has an absolutely convergent Fourier series and is never zero, then $1/f$ has an absolutely convergent Fourier series.¹⁰ There is a strong analogy between Wiener's Lemma and the inverse-closedness of matrix algebras. Just as a function with rapidly decaying Fourier coefficients can be well approximated by trigonometric polynomials, so a matrix with rapidly decaying off-diagonal entries can be well approximated by banded matrices. We refer the reader to [96] for details.

In this section we limit ourselves to a brief description of some of the most important decay algebras, and we refer to the original papers for details and applications. For simplicity we focus on matrices (bounded linear operators) of the form $A = [A_{ij}]_{i,j \in S}$ with $S = \mathbb{Z}$ or $S = \mathbb{N}$ and on off-diagonal decay measured in terms of the distance $d(i,j) = |i-j|$, although the same results hold more generally for matrices indexed by a set $T \times T$ where (T,d) is a metric space such that the distance function d on T satisfies condition (10).

The first two examples are due to Jaffard [117]:

¹⁰As is well known, Gelfand was able to give a short proof of Wiener's Lemma using his general theory of *commutative* Banach algebras; see, e.g., [87, p. 33]. Wiener's Lemma is simply the statement that the *Wiener algebra* $\mathcal{A}(\mathbb{T})$ of all functions on the unit circle having an absolutely convergent Fourier expansion is inverse-closed in the algebra $\mathcal{C}(\mathbb{T})$ of continuous functions on \mathbb{T} .

Definition 4 Let $\gamma > 0$. A matrix A belongs to the class \mathcal{E}_γ if for all i, j :

$$\forall \gamma' < \gamma, \quad |[A]_{ij}| \leq K(\gamma') \exp(-\gamma'|i - j|) \quad (68)$$

for some constant $K = K(\gamma') > 0$.

Next, suppose that

$$\sup_{i \in S} \sum_{j \in S} (1 + |i - j|)^{-p} < \infty;$$

for example, $p > 1$ (for $S = \mathbb{N}$ or \mathbb{Z}). For such p we have the following definition.

Definition 5 Let $\alpha > p$. A matrix A belongs to the class Q_α if for all i, j :

$$|[A]_{ij}| \leq K(1 + |i - j|)^{-\alpha}, \quad (69)$$

for some constant $K > 0$.

Any matrix A in \mathcal{E}_γ or in Q_α is a bounded linear operator on $\ell^2(S)$ (this is a consequence of *Schur's Lemma*, see [117]). Moreover, in [117] it is also shown that both \mathcal{E}_γ and Q_α are algebras; Q_α is called the *Jaffard algebra*.

In [117], Jaffard proved the following theorems (the first straightforward, the second not).

Theorem 22 ([117]) *Let $A \in \mathcal{E}_\gamma$ and assume that A is invertible as an operator on $\ell^2(S)$. Then $A^{-1} \in \mathcal{E}_{\gamma'}$ for some $0 < \gamma' < \gamma$.*

Hence, if A has an exponential off-diagonal decay property and A is invertible in $\mathcal{B}(\ell^2)$, the entries of A^{-1} are also bounded in an exponentially decaying manner away from the main diagonal but with a different decay rate, the decay being generally slower. Note that this result generalizes many of the results known in the literature about the exponential decay in the inverses of band matrices.

A deeper, and a priori unexpected, result is the following.

Theorem 23 ([117]) *Let $A \in Q_\alpha$ and assume that A is invertible as an operator on $\ell^2(S)$. Then $A^{-1} \in Q_\alpha$*

Thus, the *Jaffard algebra* Q_α is inverse-closed in $\mathcal{B}(\ell^2)$: if A satisfies the off-diagonal algebraic decay property (69) and is invertible in $\mathcal{B}(\ell^2)$, the inverse A^{-1} satisfies exactly the same decay property. Similar results were obtained by Baskakov in [12, 13].

Jaffard's and Baskakov's results have attracted considerable interest and have been generalized in various directions. Extensions to different types of decay can be found in [97] and [186]; the former paper, in particular, makes use of Banach algebra techniques (not mentioned in Jaffard's original paper) and points out the implications for the functional calculus.

Although concerned with infinite matrices, there is no lack of applications of the theory to concrete, finite-dimensional problems from numerical analysis. A connection is provided by the finite section method for the solution of operator equations of the form $Ax = \mathbf{b}$, where A is assumed to be boundedly invertible and $\mathbf{b} \in \ell^2$. In a nutshell, this method consists in considering the n -dimensional sections $A_n = P_n A P_n$ (where P_n is the orthogonal projector onto the subspace spanned by $\mathbf{e}_1, \dots, \mathbf{e}_n$) of the infinite matrix A and the truncated vectors $\mathbf{b}_n = P_n \mathbf{b}$, solving the finite-dimensional problems $A_n \mathbf{x}_n = \mathbf{b}_n$, and letting $n \rightarrow \infty$. The component-wise convergence of the approximate solutions \mathbf{x}_n to the solution $\mathbf{x} = A^{-1} \mathbf{b}$ of the original, infinite-dimensional problem requires that the sequence $\{A_n\}$ be *stable*, i.e., the inverses A_n^{-1} exist and have uniformly bounded norm with respect to n . These conditions are essentially those that guarantee off-diagonal decay in A^{-1} ; hence, decay algebras play a key role in the analysis, see for example [98], or [139] for a systematic treatment. Another approach, based on the notion of *nearest neighbor approximation*, is described in [60]; here again decay algebras play the main role. In the opposite direction, the authors of [171] develop an algorithm for solving large $n \times n$ Toeplitz systems by embedding the coefficient matrix A_n into a semi-infinite Toeplitz matrix A and making use of the (canonical) Wiener–Hopf factorization of the inverse of the symbol of A to obtain the solution, which is then truncated and corrected (via the solution of a much smaller Toeplitz system by conventional techniques) to yield the solution of the original problem. Ultimately, this approach works because of the exponential decay of the entries of the inverse of the infinite matrix A .

The finite section method, when applicable, can also be used to establish decay properties for functions of $n \times n$ matrices A_n with off-diagonal decay (with $n \rightarrow \infty$) by thinking of the A_n 's as the finite sections of an infinite matrix $A \in \mathcal{A}$ for a suitable decay algebra \mathcal{A} , assumed to be inverse-closed in $\mathcal{B}(\ell^2)$. Suppose that the spectra of all the A_n are contained in a compact subset \mathcal{C} of \mathbb{C} and that the contour Γ in (67) surrounds \mathcal{C} . If the norms of the resolvents $(zI_n - A_n)^{-1}$ are bounded uniformly in n and in $z \in \Gamma$, then the entries of $(zI_n - A_n)^{-1}$ converge to those of $(zI - A)^{-1}$ as $n \rightarrow \infty$, and therefore the entries of $f(A_n)$ must converge to those of $f(A)$ as $n \rightarrow \infty$. This implies that, at least for n sufficiently large, the off-diagonal entries of $f(A_n)$ must decay like those of $f(A)$, therefore the decay is that of the algebra \mathcal{A} . Note that this approach does not work unless \mathcal{A} is inverse-closed in $\mathcal{B}(\ell^2)$; thus, it cannot be used to prove exponential decay (or superexponential decay when f is entire), since the algebra \mathcal{E}_γ is *not* inverse-closed in $\mathcal{B}(\ell^2)$.

3.9 Localization in Matrix Factorizations

So far we have focused on the decay properties of functions of matrices, including the inverse. In numerical linear algebra, however, matrix factorizations (LU, Cholesky, QR, and so forth) are even more fundamental. What can be said about

the localization properties of the factors of a matrix which is itself localized? By *localized* here we mean banded, sparse, or satisfying an off-diagonal decay property.

We say that a matrix A is (m, p) -*banded* if $[A]_{ij} = 0$ for $i - j > m$ and for $j - i > p$. If A is (m, p) -banded and has the LU factorization $A = LU$ with L unit lower triangular and U upper triangular (without pivoting), it is clear that the triangular factors L and U have, respectively, lower bandwidth m and upper bandwidth p . A similar observation applies to the Cholesky and QR factors.

For more general sparse matrices the situation is more complicated, because of the fill-in that usually occurs in the factors of a sparse matrix and due to the fact that reorderings (row and column permutations) are usually applied in an attempt to preserve sparsity. Nevertheless, much is known about the nonzero structure of the triangular factors, especially in the case of Cholesky and QR factorizations.

The decay properties of the *inverse* factors of infinite banded matrices have been studied by a few authors. Existence and bounded invertibility results for triangular factorizations and block factorizations of have been obtained, e.g., in [194, 195], in particular for Toeplitz and block Toeplitz matrices, where the decay properties of the inverse factors were also considered.

For a banded $n \times n$ matrix A_n , example (32) shows that in general we cannot expect decay in the inverse triangular factors as $n \rightarrow \infty$ unless some uniform boundedness condition is imposed on the condition number of A_n . One such result (from [24]) goes as follows. Recall the bound of Demko et al. for the entries of the inverse of an m -banded Hermitian and positive definite A :

$$|[A^{-1}]_{ij}| \leq K \rho^{|i-j|}, \quad \forall i, j$$

where $[a, b]$ is the smallest interval containing the spectrum $\sigma(A)$ of A , $K = \max\{a^{-1}, K_0\}$, $K_0 = (1 + \sqrt{\kappa_2(A)})/2b$, $\kappa = \frac{b}{a} = \|A\|_2 \|A^{-1}\|_2$, $\rho = q^{1/m}$, and $q = q(\kappa_2(A)) = \frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}$. With these definitions of K and ρ , we have:

Theorem 24 ([24]) *Let $A = A^* \in \mathbb{C}^{n \times n}$ be positive definite and m -banded, and suppose A has been scaled so that $\max_{1 \leq i \leq n} [A]_{ii} = 1$. Let $A = LL^*$ denote the Cholesky factorization of A . Then*

$$|[L^{-1}]_{ij}| \leq K_1 \rho^{|i-j|}, \quad \forall i > j, \tag{70}$$

where $K_1 = K \frac{1-\rho^m}{1-\rho}$.

In view of the identity $L^{-1} = L^T A^{-1}$, the decay in the inverse factor is a consequence of the fact that the product of a banded matrix times a matrix with exponential decay must necessarily decay as well. Since $K_1 > K$, the bound (70) indicates a potentially slower decay in L^{-1} than in the corresponding entries of A^{-1} , but this is not always the case. For instance, as noted in [24], the entries of L^{-1} must actually be smaller than the corresponding entries of A^{-1} when A is an M -matrix.

As usual, Theorem 24 can be applied to a sequence $\{A_n\}$ of m -banded, Hermitian positive definite matrices of increasing order, such that $\sigma(A_n) \subset [a, b]$ for all

n , normalized so that their largest entry is equal to 1. The theorem then gives a uniform (in n) exponential decay bound on the entries of the inverse Cholesky factors L_n^{-1} , as $n \rightarrow \infty$. If, on the other hand, the condition numbers $\kappa_2(A_n)$ grow unboundedly for $n \rightarrow \infty$, the bounds (70) depend on n , and will deteriorate as $n \rightarrow \infty$. This is the case, for instance, of matrices arising from the discretization of partial differential equations. Nevertheless, sharp decay bounds on the elements of the inverse Cholesky factor of sparse matrices arising from the discretization of certain PDEs have been recently obtained in [48].

What about the case of matrices with decay, which may be full rather than banded or sparse? When are the factors of a localized matrix themselves localized? A wealth of results for matrices belonging to different decay algebras have been obtained by Blatov [33, 34] and more recently by Kryshnal et al. [127]. Roughly speaking, these papers show that for the most frequently encountered decay algebras \mathcal{A} , if a matrix $A \in \mathcal{A}$ admits the LU factorization in $\mathcal{B}(\ell^2)$, then the factors belong to \mathcal{A} , hence they satisfy the same decay condition as A ; if, moreover, the algebra \mathcal{A} is inverse-closed in $\mathcal{B}(\ell^2)$, then obviously the inverse factors L^{-1}, U^{-1} must satisfy the same decay bounds. Analogous results, under the appropriate technical conditions, apply to the Cholesky, QR, and polar factorizations. We refer to [33, 34] and [127] for details.

3.10 Localization in the Unbounded Case

Throughout this paper, we have limited our discussion of localization to the following situations:

- finite matrices of fixed size;
- sequences of matrices of increasing size, with uniformly bounded spectra;
- bounded infinite matrices on ℓ^2 .

A natural question is, to what extent can the decay results for matrix functions (especially the inverse, the exponential, and spectral projectors) be extended to unbounded operators¹¹ or to sequences of matrices of increasing size not having uniformly bounded spectra? We know from simple examples that in general we cannot hope to find straightforward extensions of (say) exponential decay bounds of the type (15) or (28) without the boundedness assumption on the spectra. On the other hand, classical exponential decay results for the eigenfunctions of certain elliptic operators (e.g., [2, 53, 74, 180]) and for the Green's function of parabolic operators (like the *heat kernel* for a fixed time t , see e.g. [200, p. 328]) show that exponential and even superexponential decay with respect to space do occur for certain functions of unbounded operators. It is reasonable to expect that similar results should hold for discretizations of these operators that lead to sparse matrices;

¹¹For the sake of simplicity, we only consider the self-adjoint case here.

ideally, one would like to obtain decay rates that do not depend on discretization parameters, and unfortunately decay bounds like the ones in Theorem 8 fail to meet this requirement.

In more detail, suppose A is a self-adjoint, unbounded operator defined on a dense subspace of a Hilbert space \mathcal{H} , and that f is a function defined on the spectrum of $A = A^*$. If f is an essentially bounded function defined on $\sigma(A)$, the spectral theorem for self-adjoint operators (see, e.g., [173]) allows one to define the function $f(A)$ via the integral representation

$$f(A) = \int_{-\infty}^{\infty} f(\lambda) dE(\lambda),$$

where E is the spectral family (resolution of the identity) associated with A which maps Lebesgue-measurable subsets of $\sigma(A)$ to the algebra $\mathcal{B}(\mathcal{H})$, such that $E(\sigma(A)) = I$. Note that $\sigma(A) \subseteq \mathbb{R}$ is now unbounded. Since $f \in L^\infty(\sigma(A))$, clearly $f(A) \in \mathcal{B}(\mathcal{H})$. Thus, bounded functions of unbounded operators are bounded operators. Non-trivial examples include the Cayley transform,

$$\Psi(A) = (A - iI)(A + iI)^{-1},$$

a unitary (and therefore bounded) operator, and the exponential e^{itA} , also unitary. Furthermore, the exponential e^{-tA} (when A is positive definite) and the closely related resolvent $(A - zI)^{-1}$ (with $z \notin \sigma(A)$) are compact, and therefore bounded, for some important classes of unbounded operators. Another obvious example is given by the spectral projectors, since in this case the range of f is just the set $\{0, 1\}$. In such cases it is sometimes possible to obtain exponential decay results for certain (analytic) functions of banded, unbounded operators.

Remark 8 The case in which $f(A)$ is compact is especially favorable: since every compact operator on a separable Hilbert space is the norm-limit of finite rank operators, the entries of $f(A)$ (represented by an infinite matrix with respect to an arbitrary orthonormal basis on \mathcal{H}) must have rapid decay away from a *finite* set of positions, including down the main diagonal. If in addition $f(A)$ is *trace class* ($\sum_{i=1}^{\infty} |[f(A)]_{ii}| < \infty$) and in particular *Hilbert–Schmidt* ($\sum_{i,j=1}^{\infty} |[f(A)]_{ij}|^2 < \infty$), decay must be quite fast, though not necessarily exponential.

To the best of our knowledge, only isolated results are available in the literature. An example, already mentioned, is that of e^{itA} for a specific class of unbounded tridiagonal matrices A on $\ell^2(\mathbb{Z})$. Additional examples can be found in [182] and [118]. In these papers one can find exponential localization results for the resolvent and eigenfunctions (and thus spectral projectors) of certain infinite banded matrices of physical interest. Very recently, sharp decay estimates of discretized Green's functions for a broad class of Schrödinger operators have been obtained in [136]. These decay results are established for finite difference and pseudo-spectral discretizations, using methods similar to those used to establish the decay properties of the continuous Green's function (see, e.g., [167]). The advantage of these bounds

is that they do not deteriorate as the mesh parameter h tends to zero, and thus they are able to capture the exponential decay in the Green's function, when present.

It would be desirable to investigate to what extent one can derive general decay results for bounded analytic functions of unbounded banded (or sparse) infinite matrices, analogous to those available in the bounded case.

4 Applications

In this section we discuss a few selected applications of the theory developed so far. We focus on two broad areas: numerical linear algebra, and electronic structure computations. Algorithmic aspects are also briefly mentioned. The following is not intended as an exhaustive discussion, but rather as a sampling of current and potential applications with pointers to the literature for the interested reader.

4.1 *Applications in Numerical Linear Algebra*

The decay properties of inverses and other matrix functions have been found useful in various problems of numerical linear algebra, from solving linear systems and eigenvalue problems to matrix function evaluation. Below we discuss a few of these problems.

4.1.1 Linear Systems with Localized Solutions

A possible application of inverse decay occurs when solving linear systems $\mathbf{Ax} = \mathbf{b}$ with a localized right-hand side \mathbf{b} . For example, if $\mathbf{b} = \alpha \mathbf{e}_i$ where \mathbf{e}_i is the i th standard basis vector, the solution is given by $\mathbf{x} = \alpha A^{-1} \mathbf{e}_i$, a multiple of the i th column of A^{-1} . If it is known that A^{-1} decays rapidly away from certain positions, the solution vector will be localized around the corresponding positions in \mathbf{x} . The same holds if \mathbf{b} contains not just one but $k \ll n$ nonzero entries, or if it is a dense but localized vector. Problems of this kind arise frequently in applications, where the right-hand side \mathbf{b} often corresponds to a localized forcing term such as a point load or a source (of heat, of neutrons, etc.) located in a small subregion of the computational domain. In each of these cases, bounds on the entries of A^{-1} can be used to determine a priori an “envelope” containing those parts of the solution vector \mathbf{x} in which the nonnegligible entries are concentrated. Even if the bounds are pessimistic, this can lead to worthwhile computational savings.

Of course, this approach requires the use of algorithms for solving linear systems that are capable of computing only selected parts of the solution vector. Available methods include variants of Gaussian elimination [72, Sect. 7.9], Monte Carlo

linear solvers [27], and quadrature rule-based methods for evaluating bilinear forms $\mathbf{u}^T A^{-1} \mathbf{v}$ (in our case $\mathbf{u} = \mathbf{e}_j$ and $\mathbf{v} = \mathbf{b}$, since $x_j = \mathbf{e}_j^T A^{-1} \mathbf{b}$), see [39, 94].

It is worth mentioning that this problem is somewhat different from that arising in compressed sensing, where one looks for solutions that have (near-)maximal sparsity in a non-standard basis, leading to the problem of finding the “sparsest” solution among the infinitely many solutions of an underdetermined system [45, 46].

4.1.2 Construction of Preconditioners

The results on the exponential decay in the inverses of band and sparse matrices, originally motivated by the converge analysis of spline approximations [67], have been applied to the construction of preconditioners for large, sparse systems of linear equations. Specifically, such results have been used, either implicitly or explicitly, in the development of block incomplete factorizations and of sparse approximate inverse preconditioners.

A pioneering paper on block incomplete factorizations is [54], where various block incomplete Cholesky preconditioners are developed for solving large, symmetric positive definite block tridiagonal linear systems with the preconditioned conjugate gradient method. This paper has inspired many other authors to develop similar techniques, including preconditioners for nonsymmetric problems; see, e.g., [7, Chap. 7], and [8, 11, 33, 35, 197] among others.

Consider a large, sparse, block tridiagonal matrix (assumed to be symmetric positive definite for simplicity):

$$A = \begin{bmatrix} A_1 & B_2^T \\ B_2 & A_2 & B_3^T \\ & B_3 & A_3 & B_4^T \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & \ddots \\ & & & & B_{p-1} & A_{p-1} & B_p^T \\ & & & & B_p & A_p & \end{bmatrix},$$

where the blocks A_i and B_i are typically banded; for example, in [54] the diagonal blocks A_i are all tridiagonal, and the off-diagonal nonzero blocks B_i are diagonal. Then A has a block Cholesky factorization of the form

$$A = (L + D)D^{-1}(L + D)^T$$

where L is block strictly lower triangular and D is block diagonal with blocks

$$\Delta_1 = A_1, \quad \Delta_i := A_i - B_i \Delta_{i-1}^{-1} B_i^T, \quad i = 2, \dots, p. \quad (71)$$

The successive Schur complements Δ_i in (71) are the *pivot blocks* of the incomplete block Cholesky (more precisely, block LDL^T) factorization. They are dense matrices for $i = 2, \dots, p$. An incomplete block factorization can be obtained by approximating them with sparse matrices:

$$\Delta_1^{-1} \approx \Sigma_1, \quad \Delta_i \approx A_i - B_i \Sigma_{i-1} B_i^T, \quad i = 2, \dots, p,$$

where $\Sigma_i \approx \Delta_i^{-1}$ for $1 \leq i \leq p$ is typically a banded approximation. Estimates of the decay rates of the inverses of band matrices can then be used to determine the bandwidth of the successive approximations to the pivot blocks. We refer to the above-given references for details on how these banded approximations can be obtained.

Unfortunately, unless the pivot blocks are sufficiently diagonally dominant they cannot be well-approximated by banded matrices. For these reasons, more sophisticated (albeit generally more expensive) approximations have been developed based on hierarchical matrix techniques in recent years [11]. Nevertheless, cheap approximations to Schur complements using banded or sparse approximations to the inverses of the blocks may be sufficient in some applications; see, e.g., [142, 179].

Preconditioners for general sparse matrices based on sparse approximate inverses, the first examples of which date back to the 1970s, have been intensively developed beginning in the 1990s; see, e.g., [17] for a survey, and [174, Sect. 10.5] for a self-contained discussion. More recently, interest in these inherently parallel preconditioning methods has been revived, due in part to the widespread diffusion of Graphic Processing Units (GPUs). In these methods, the inverse of the coefficient matrix is approximated directly and explicitly by a sparse matrix $M \approx A^{-1}$; in some cases M is the product of two sparse triangular matrices which approximate the inverse triangular factors L, U of A . Applying the preconditioner only requires matrix-vector products, which are much easier to parallelize than triangular solves [99]. The main challenge in the construction of these preconditioners is the determination of a suitable sparsity pattern for M . Indeed, if a “good” sparsity pattern can be estimated in advance, the task of computing a sparse approximate inverse with a nonzero pattern that is a subset of the given one is greatly facilitated [52, 99, 113]. If A is banded, a banded approximate inverse may suffice. If A is not banded but strongly diagonally dominant, a sparse approximate inverse with the same nonzero pattern as A will usually do. In other cases, using the sparsity pattern of A^2 will give better results, although at a higher cost since A^2 may be considerably less sparse than A . The rationale for considering the patterns of successive powers of A is the following. Suppose A is diagonally dominant. Then, up to a diagonal scaling, it can be written as $A = I - B$ for some matrix $B \in \mathbb{C}^{n \times n}$ with $\varrho(B) < 1$. Therefore

$$A^{-1} = (I - B)^{-1} = I + B + B^2 + \dots, \quad (72)$$

where the entries of B^k must decay rapidly to zero as $k \rightarrow \infty$ since A is diagonally dominant. Since B and A have the same pattern, (72) suggests that using the sparsity

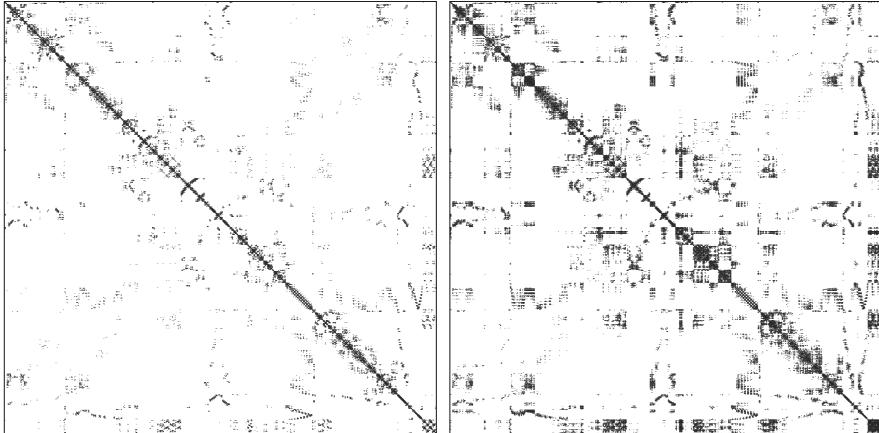


Fig. 11 *Left:* nonzero pattern of A . *Right:* pattern of approximate inverse of A

pattern of A for M may be sufficient, especially if A exhibits very strong diagonal dominance; if not, higher order terms in (72) may have to be considered. Note that considering higher powers of A means looking beyond the immediate neighbors of each node in the sparsity graph $\mathcal{G}(A)$ of A ; in view of bounds like (21), according to which the entries in A^{-1} decay rapidly with the distance from the nonzeros in A , it shouldn't be necessary to look at high powers, and indeed in practice one rarely goes beyond A^2 . For highly nonsymmetric matrices, Huckle [113] has shown that the nonzero pattern of A^T and its powers may have to be considered as well. We refer to [3, 52] for further details.

As an example, in Fig. 11 we show on the left the nonzero pattern of a complex symmetric matrix A arising from an application to computational electromagnetics (see [3] for details), and on the right the sparsity pattern of an approximation M to A^{-1} corresponding to the nonzero pattern of A^2 . The approximate inverse M was computed by minimizing the Frobenius norm $\|I - AM\|_F$ over all matrices with the same sparsity pattern of A^2 . With this preconditioner, GMRES requires 80 iterations to reduce the relative residual norm below 10^{-8} (the method stagnates without preconditioning). It is worth noting that computing the exact inverse A^{-1} and dropping all entries with $|[A^{-1}]_{ij}| < \varepsilon \|A\|_F$ with $\varepsilon = 0.075$ produces a sparse matrix with a nonzero pattern virtually identical to the one in Fig. 11 (right).

In the construction of *factorized* approximate inverse preconditioners (like FSAI [125] and AINV [23, 25]), it is useful to know something about the decay in the inverse triangular factors of A . Hence, bounds like (70) provide some insight into the choice of a good approximate sparsity pattern and on the choice of ordering [24]. Similar remarks apply to more traditional incomplete LU and QR factorization using the results in Sect. 3.9 on the localization in the factors of matrices with decay. See also [48] and [137] for other examples of situations where knowledge of decay bounds in the inverse Cholesky factor is useful for numerical purposes.

In recent years, much attention has been devoted to the numerical solution of fractional differential equations. Discretization of these non-local equations leads to dense matrices which are usually not formed explicitly. Matrix-vector multiplications (needed in the course of Krylov subspace iterations) can be efficiently computed in $O(n \log n)$ work using Fast Fourier Transforms (FFTs) and diagonal scalings. In [160], preconditioning techniques for linear systems arising from the discretization of certain initial boundary value problems for a fractional diffusion equation of order $\alpha \in (1, 2)$ are introduced and analyzed. At each time step, a nonsymmetric linear system $\mathbf{Ax} = \mathbf{b}$ must be solved, where A is of the form

$$A = \eta I + DT + WT^T,$$

with $\eta > 0$, D , W diagonal and nonnegative, and T a lower Hessenberg Toeplitz matrix. As the matrices D and W change at each time step, A also changes, and it is therefore important to develop preconditioners that are easy to construct and apply, while at the same time resulting in fast convergence rates of the preconditioned Krylov iteration. The preconditioners studied in [160] are based on circulant approximations, FFTs and interpolation and the authors show both theoretically and numerically that they are effective. The theoretical analysis in [160], which shows that the preconditioned matrices have spectra clustered around unity, makes crucial use of inverse-closed decay algebras. In particular, the authors show that T , and therefore A and the circulant approximations used in constructing the preconditioners, all belong to the Jaffard algebra $Q_{\alpha+1}$, where α is the fractional order of the spatial derivatives in the differential equation (see Definition 5). This fact is used in establishing the spectral properties of the preconditioned matrices.

4.1.3 Localization and Eigenvalue Problems

Parlett [161, 162] and Vömel and Parlett [198] have observed that the eigenvectors corresponding to isolated groups of eigenvalues of symmetric tridiagonal matrices are often localized—an observation already made by Cuppen [59].¹² In [198], Vömel and Parlett develop heuristics for estimating envelopes corresponding to nonnegligible entries of eigenvectors of tridiagonals, and show that knowledge of these envelopes can lead to substantial savings when the eigenvectors are localized and when solvers that are able to compute only prescribed components of eigenvectors are used, such as inverse iteration and the MRRR algorithm [70, 144, 163]. They also observe that eigenvectors corresponding to isolated eigenvalue clusters are not always localized, and that a priori detection of localization of the eigenvectors poses a challenge.

To see how the theory of decay in matrix functions can help address this challenge, we recast the problem in terms of spectral projectors instead of eigenvectors.

¹²See also Exercise 30.7 in Trefethen and Bau [191].

Let $A = A^* \in \mathbb{C}^{n \times n}$ have eigenvalues

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$$

(in the tridiagonal case we can assume that A is irreducible so that its eigenvalues are all simple, see [95, p. 467]). Suppose now that eigenvalue λ_{k+1} is well-separated from λ_k , and that eigenvalue λ_{k+p} is well-separated from λ_{k+p+1} . If \mathbf{v}_i denotes an eigenvector corresponding to λ_i with $\|\mathbf{v}_i\|_2 = 1$, the spectral projector associated with the group of eigenvalues $\{\lambda_{k+1}, \dots, \lambda_{k+p}\}$ can be written as

$$P = \mathbf{v}_{k+1}\mathbf{v}_{k+1}^* + \cdots + \mathbf{v}_{k+p}\mathbf{v}_{k+p}^* = VV^*, \quad (73)$$

where

$$V = [\mathbf{v}_{k+1}, \dots, \mathbf{v}_{k+p}] \in \mathbb{C}^{n \times p} \quad (74)$$

is a matrix with orthonormal columns. Note that P is the orthogonal projector onto the A -invariant subspace $\mathcal{V} = \text{span}\{\mathbf{v}_{k+1}, \dots, \mathbf{v}_{k+p}\}$.

Let us first consider the case of a single, simple eigenvalue, $p = 1$. If \mathbf{v} is the corresponding normalized eigenvector, the spectral projector is of the form

$$P = \mathbf{v}\mathbf{v}^* = \begin{bmatrix} |v_1|^2 & v_1\bar{v}_2 & \dots & v_1\bar{v}_n \\ v_2\bar{v}_1 & |v_2|^2 & \dots & v_2\bar{v}_n \\ \vdots & \vdots & \ddots & \vdots \\ v_n\bar{v}_1 & v_n\bar{v}_2 & \dots & |v_n|^2 \end{bmatrix}.$$

Conversely, given a rank-1 projector P , the eigenvector \mathbf{v} is uniquely determined (up to a constant). It is also clear that if $\mathbf{v} = (v_i)$ is a vector such that $|v_i| \ll |v_j|$ for $i \neq j$, then the entries of P must decay rapidly away from $[P]_{ii} = |v_i|^2$ (not only away from the main diagonal, but also along the main diagonal). More generally, if most of the “mass” of \mathbf{v} is concentrated in a few components, the entries of P must decay rapidly away from the corresponding diagonal entries. Conversely, it is evident that rapid decay in P implies that \mathbf{v} must itself be localized. Hence, in the rank-1 case $P = \mathbf{v}\mathbf{v}^*$ is localized if and only if \mathbf{v} is. An example is shown in Fig. 12.

On the other hand, in the case of spectral projectors of the form (73) with $p > 1$, localization in the eigenvectors $\mathbf{v}_{k+1}, \dots, \mathbf{v}_{k+p}$ is a *sufficient* condition for localization of P , but not a necessary one. This is due to the possible (near) cancellation in the off-diagonal entries when adding up the rank-1 projectors $\mathbf{v}_j\mathbf{v}_j^*$. This fact becomes actually obvious if one observes that summing *all* the projectors $\mathbf{v}_j\mathbf{v}_j^*$ for $j = 1, \dots, n$ must result in the identity matrix, which is maximally localized even though the eigenvectors may be strongly delocalized. Less trivial examples (with $1 < p \ll n$) can be easily constructed. Real-world instances of this phenomenon are actually well known in physics; see, e.g., [44] and Sect. 4.2 below.

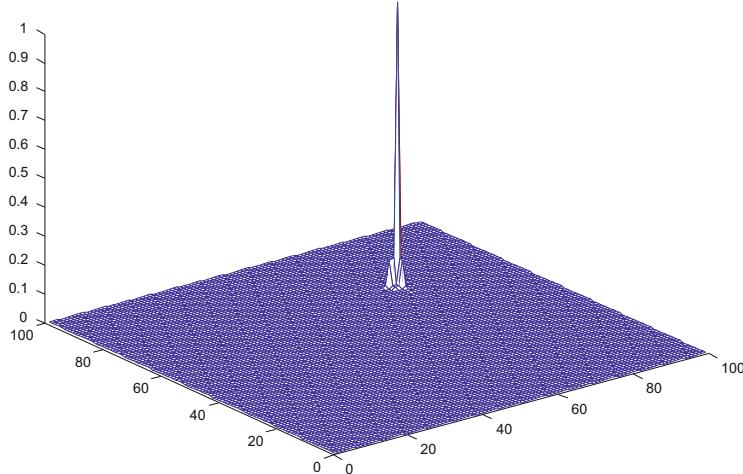


Fig. 12 Plot of $|[P]_{ij}|$ where P is the spectral projector onto the eigenspace corresponding to an isolated eigenvalue of a tridiagonal matrix of order 100

We also remark that if P is localized, there may well be another orthonormal basis $\{\mathbf{u}_{k+1}, \dots, \mathbf{u}_{k+p}\}$ of \mathcal{V} , different from the eigenvector basis, which is localized. When $p > 1$, P does not determine the basis vectors uniquely. Indeed, if $\Theta \in \mathbb{C}^{p \times n}$ is any matrix with orthonormal rows, we have that

$$P = VV^* = V\Theta\Theta^*V^* = UU^*, \quad U = V\Theta,$$

which shows how $U = [\mathbf{u}_{k+1}, \dots, \mathbf{u}_{k+p}]$ is related to V . Even if the columns of V are not localized, those of U may well be, for a suitable choice of Θ . We note, on the other hand, that if P is delocalized then there can be no strongly localized basis vectors $\{\mathbf{u}_{k+1}, \dots, \mathbf{u}_{k+p}\}$ for \mathcal{V} . Nevertheless, searching for an orthonormal basis that is “as localized as possible” is an important problem in certain physics applications; see, e.g., [61, 62, 92].

Now that we have recast the problem in terms of spectral projectors, we can apply the theory of decay in matrix functions. Indeed, the spectral projector is a function of A : if $A = A^*$ has the spectral decomposition $A = Q\Lambda Q^*$, where Q is unitary and $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, then

$$P = \phi(A) = Q\phi(\Lambda)Q^*, \tag{75}$$

where ϕ is any function such that

$$\phi(\lambda_i) = \begin{cases} 1, & \text{if } k+1 \leq i \leq k+p, \\ 0, & \text{else.} \end{cases}$$

Hence, any analytic function that interpolates ϕ at the eigenvalues of A will do; in practice, it is sufficient to use an analytic function that approximates ϕ on the spectrum of A . For example, any function f such that $f(\lambda) \approx 1$ for $\lambda \in [\lambda_{k+1}, \lambda_{k+p}]$ which drops rapidly to zero outside this interval will yield an excellent approximation of P . It is easy to see that the wider the gaps $(\lambda_k, \lambda_{k+1})$ and $(\lambda_{k+p}, \lambda_{k+p+1})$, the easier it is to construct such an analytic approximation of $\phi(\lambda)$, and the faster the off-diagonal decay is in $f(A)$ and therefore in P , assuming of course that A is banded or sparse.

As an illustration, consider the case of an isolated eigenvalue $\lambda \in \sigma(A)$. Since an eigenvector of A associated with λ is an eigenvector of $A - \lambda I$ associated with 0, we can assume without any loss of generality that $\lambda = 0$. Let \mathbf{v} be this eigenvector (with $\|\mathbf{v}\|_2 = 1$) and let $P = \mathbf{v}\mathbf{v}^*$ be the corresponding spectral projector. The function ϕ such that $P = \phi(A)$ can be approximated to within arbitrary accuracy by a Gaussian

$$f(x) = e^{-x^2/\xi}, \quad \text{where } \xi > 0. \quad (76)$$

The choice of ξ , which controls the rate of decay to zero of the Gaussian, will depend on the desired accuracy and thus on the distance between the eigenvalue $\lambda = 0$ and its nearest neighbor in the spectrum of A ; we denote this distance by η . To determine ξ , suppose we wish to have $f(\pm\eta) \leq \varepsilon$ for a prescribed $\varepsilon > 0$. Thus, we require that

$$e^{-\eta^2/\xi} \leq \varepsilon,$$

which yields

$$0 < \xi \leq -\eta^2 / \log(\varepsilon).$$

For instance, given $\varepsilon > 0$ we can approximate P by

$$P \approx f(A) = \exp(-A^2/\xi), \quad \xi = -\eta^2 / \log(\varepsilon).$$

It is shown in [168] that this approximation works very well. For instance, for $\varepsilon = 10^{-8}$ and $\eta = 0.1$, choosing $\xi = 3 \cdot 10^{-4}$ yields $\|P - f(A)\|_F = 7 \cdot 10^{-14}$. Moreover, specializing Theorem 8 to the Gaussian (76) leads to the following off-diagonal decay result (Theorem 4.2 in [168]):

Theorem 25 ([168]) *Let $\varepsilon > 0$ and let f be given by (76) with $\xi = -c\eta^2$, $c = 1/\log(\varepsilon)$. Let $A = A^*$ be m -banded and assume that $[-1, 1]$ is the smallest interval containing $\sigma(A)$. Then, for $i \neq j$ we have*

$$|[\exp(-A^2/\xi)]_{ij}| \leq K \rho^{|i-j|}, \quad (77)$$

where

$$K = \frac{2\chi e^{c\alpha^2/\eta^2}}{\chi - 1}, \quad \alpha > 1, \quad \chi = \alpha + \sqrt{\alpha^2 - 1}, \quad \rho = \chi^{-1/m}.$$

Note that the assumption that $\sigma(A) \subset [-1, 1]$ leads to no loss of generality, since spectral projectors are invariant under shifting and scaling of A . Recalling that for any projector $|[P]_{ij}| \leq 1$ for all i, j , it is clear that the bound (77) is only informative if the quantity on the right-hand side is less than 1, which may require taking $|i - j|$ sufficiently large. This theorem provides an infinite family of bounds parameterized by $\alpha > 1$ (equivalently, by $\chi > 1$). Hence, the entries of $f(A)$, and thus of P , satisfy a superexponential off-diagonal decay (this is expected since f is entire). Note that there is fast decay also along the main diagonal: this is obvious since for an orthogonal projector,

$$\text{Tr}(P) = \text{rank}(P), \quad (78)$$

and since the diagonal entries of P are all positive, they must decrease rapidly away from the $(1, 1)$ position for the trace of P to be equal to 1. With this, the localization of the spectral projector (and thus of a normalized eigenvector) corresponding to isolated eigenvalues of banded matrices (in particular, tridiagonal matrices) is rigorously established.

The above construction can be extended to approximate the spectral projector P corresponding to a group of k well-separated eigenvalues: in this case P can be well-approximated by a sum of rapidly decaying Gaussians centered at the eigenvalues in the given group, hence P will again exhibit superexponential off-diagonal decay, with k spikes appearing on the main diagonal.

The use of a single shifted Gaussian (centered at a prescribed value μ) with a suitable choice of the parameter ξ can also be used to approximate the spectral projector corresponding to a tight cluster of several eigenvalues falling in a small interval around μ . Combined with a divide-and-conquer approach, this observation is at the basis of the recently proposed *localized spectrum slicing* (LSS) technique for computing interior eigenpairs of large, sparse, Hermitian matrices, see [135]. Unlike most current methods, this technique does not require the solution of highly indefinite, shifted linear systems. The decay theory for analytic functions of sparse matrices plays a central role in the development and analysis of this algorithm, which is shown in [135] to have linear cost in n . It should also be noted that the LSS algorithm has controllable error.

On the other hand, a different function f must be used if the eigenvalues of interest form an isolated *band*, i.e., they densely fill an interval which is well separated from the rest of the spectrum. This situation, which is of importance in physical applications, will be discussed in Sect. 4.2 below. As we shall see, in this case only exponential off-diagonal decay should be expected. Nevertheless, this gives a partial answer to the problem posed by Vömel and Parlett in [198]: even though the *eigenvectors* corresponding to an isolated cluster of eigenvalues of a tridiagonal A may fail to be localized, the corresponding spectral projector will be localized, the more so the larger the relative gap between the cluster and the rest of the spectrum.

We emphasize, however, that the gap assumption is only a *sufficient condition*, not a necessary one. If A is a large tridiagonal matrix without any discernible gap in

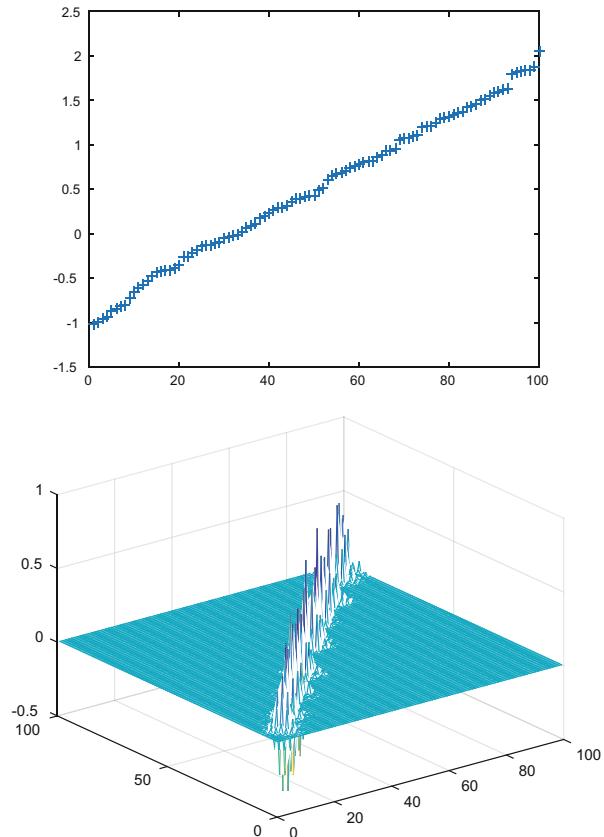
the spectrum, its eigenvectors may or may not be localized. For instance, the $n \times n$ tridiagonal matrix

$$A_n = \text{tridiag}(-1, 2, -1)$$

presents no gaps in the spectrum as $n \rightarrow \infty$, and in fact the corresponding infinite tridiagonal matrix A , viewed as a bounded operator on ℓ^2 , has purely continuous spectrum: $\sigma(A) = [0, 4]$. As is well known, the eigenvectors of A_n are delocalized, and the orthogonal projectors corresponding to individual eigenvalues or to groups of eigenvalues of A_n are also delocalized with very slow decay as $n \rightarrow \infty$ (see [26, Sect. 10] for a detailed analysis). On the other hand, with very high probability, the eigenvectors (and therefore the spectral projectors) of a randomly generated symmetric tridiagonal matrix will exhibit a high degree of localization, even in the absence of any clearly defined spectral gap between (groups of) eigenvalues.

An instance of this behavior is shown in Fig. 13. At the top we show a plot of the eigenvalues of a random symmetric tridiagonal matrix A of order $n = 100$,

Fig. 13 *Top:* eigenvalues of a random tridiagonal. *Bottom:* spectral projector corresponding to 10 smallest eigenvalues



and at the bottom we display the spectral projector P onto the invariant subspace spanned by the eigenvectors associated with the 10 smallest eigenvalues of A . Note that there is no clear gap separating the eigenvalues $\lambda_1, \dots, \lambda_{10}$ from the rest of the spectrum, and yet P exhibits rapid off-diagonal decay. The eigenvectors themselves are also strongly localized. See also the already referenced Exercise 30.7 in [191] and the examples discussed in [40, pp. 374–375], where a connection with Anderson localization [4] is made. Hence, the challenge posed by Vömel and Parlett in [198] remains in part open, both because localization can occur even in the absence of gaps in the spectrum, and because the presence of gaps may be difficult to determine a priori.

Another interesting application of off-diagonal decay is to eigenvalue perturbation theory. It turns out that for certain structured matrices, such as tridiagonal or block tridiagonal matrices, the effect of small perturbations in the matrix entries on some of the eigenvalues is much smaller than can be expected from the “standard” theory based on Weyl’s Theorem.¹³ It has been observed (see, e.g., [155]) that for tridiagonal matrices an eigenvalue is insensitive to perturbations in A if the corresponding eigenvector components are small. In [155], generalizations of this fact are established for block tridiagonal A under suitable assumptions. Hence, eigenvector localization plays an important role in proving much tighter perturbation results when A is block tridiagonal (including the special cases of tridiagonal and general m -banded matrices).

Here we show how localization results like Theorem 25 can shed some light on perturbation theory. Assume $A = A^* \in \mathbb{C}^{n \times n}$ is banded and consider the eigenvalue problem $A\mathbf{v} = \lambda\mathbf{v}$. For simplicity we assume that λ is a simple eigenvalue. If \mathbf{v} is normalized ($\|\mathbf{v}\|_2 = 1$), then

$$\lambda = \mathbf{v}^* A \mathbf{v} = \sum_{i=1}^n \sum_{j=1}^n a_{ij} \bar{v}_i v_j.$$

The sensitivity of an eigenvalue λ to small changes in the entries of A can be estimated, to first order, by the partial derivative

$$\frac{\partial \lambda}{\partial a_{ij}} = \bar{v}_i v_j + v_i \bar{v}_j \quad \forall i, j.$$

Now, the (i, j) entry of the spectral projector $P = \mathbf{v}\mathbf{v}^*$ on the eigenspace associated with λ is $[P]_{ij} = v_i \bar{v}_j$. Therefore,

$$|[P]_{ij}| \approx 0 \Rightarrow \lambda \text{ is insensitive to small changes in } a_{ij}.$$

¹³Weyl’s Theorem implies that the eigenvalues of A and $A + E$ (both Hermitian) can differ by a quantity as large as $\|E\|_2$. See [112, Sect. 4.3] for precise statements.

But we know from Theorem 25 that $|[P]_{ij}| \approx 0$ if $|i - j|$ is sufficiently large, since the entries of $|[P]_{ij}|$ satisfy a superexponential decay bound. Thus, perturbing entries of A at some distance from the main diagonal by a small amount $\delta \neq 0$ will cause a change in λ much smaller than δ . The change can be expected to be comparable to δ , on the other hand, if the perturbation occurs in a position (i, j) where $[P]_{ij}$ is not small.

Example 2 Consider the symmetric tridiagonal matrix

$$A = \begin{bmatrix} 5 & 1 & & & \\ 1 & 0 & 1 & & \\ & 1 & 0 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & \ddots \\ & & & & 1 & 0 & 1 \\ & & & & 1 & 0 & \end{bmatrix}$$

of order 100. The spectrum of A consists of the eigenvalues $\lambda_1, \dots, \lambda_{99}$, all falling in the interval $[-1.99902, 1.99902]$, plus the eigenvalue $\lambda = \lambda_{100} = 5.2$. As expected from Theorem 25, the (normalized) eigenvector \mathbf{v} associated with λ is strongly localized:

$$\mathbf{v} = \begin{bmatrix} 0.979795897113271 \\ 0.195959179422654 \\ 0.039191835884531 \\ 0.007838367176906 \\ 0.001567673435381 \\ 0.000313534687076 \\ 0.000062706937415 \\ 0.000012541387483 \\ 0.000002508277497 \\ 0.000000501655499 \\ \vdots \end{bmatrix}.$$

The entries of \mathbf{v} decay monotonically; they are all smaller than the double-precision machine epsilon from the 22nd one on.¹⁴ Hence, $P = \mathbf{v}\mathbf{v}^T$ is strongly localized, and in fact its entries decay very fast away from the $(1, 1)$ position. See also Fig. 12 for a similar case corresponding to a well-separated interior eigenvalue.

¹⁴We mention in passing reference [140], where an alternative justification is given for the observed exponential decay in \mathbf{v} .

Let \tilde{A} be the perturbed matrix obtained by replacing the 5 in position $(1, 1)$ with the value 5.001. Clearly, $\|A - \tilde{A}\|_2 = 10^{-3}$. We find that the change in the largest eigenvalue is $|\lambda(\tilde{A}) - \lambda(A)| = 9.6 \cdot 10^{-4}$. Hence, the change in the isolated eigenvalue is essentially as large as the change in the matrix; note that the $(1, 1)$ entry of the spectral projector, P , is equal to $0.96 \approx 1$.

On the other hand, suppose that the perturbed matrix \tilde{A} is obtained from A by replacing the zero in positions $(10, 1)$ and $(1, 10)$ of A by $\delta = 10^{-3}$. Again, we have that $\|A - \tilde{A}\|_2 = 10^{-3}$, but the largest eigenvalue of the modified matrix is now $\lambda(\tilde{A}) = 5.2000002$. Hence, in this case a perturbation of size 10^{-3} in A only produces a change of $O(10^{-7})$ in the isolated eigenvalue; note that the $(10, 1)$ entry of P is $\approx 4.9152 \cdot 10^{-7}$.

As we have mentioned, rapid decay in P is not limited to the off-diagonal entries: the diagonal entries $[P]_{ii}$ of P also decay superexponentially fast for increasing i . Perturbing the $(2, 2)$ entry of A by 0.001 causes a change equal to $3.84 \cdot 10^{-5}$ in the largest eigenvalue, consistent with the fact that $[P]_{2,2} = 3.84 \cdot 10^{-2}$; perturbing the $(5, 5)$ entry of A again by 0.001 causes a change equal to $2.458 \cdot 10^{-9}$, consistent with the fact that $[P]_{2,2} = 2.458 \cdot 10^{-6}$. After a perturbation by 0.001 in the (i, i) of A for $i \geq 12$, we find that the largest computed eigenvalue is numerically unchanged at 5.2.

Incidentally, we note that in this example the presence of an isolated eigenvalue can be determined a priori from Geršgorin's Theorem. More generally, this theorem can sometimes be used to determine the presence of groups or clusters of eigenvalues well-separated from the rest of the spectrum.

More generally, suppose we are interested in computing the quantity

$$\text{Tr}(PA) = \lambda_1 + \lambda_2 + \cdots + \lambda_k, \quad (79)$$

where P is the orthogonal projector onto the invariant subspace spanned by the eigenvectors corresponding to the k smallest eigenvalues of A , assumed to be banded or sparse. This is a problem that occurs frequently in applications, especially in physics.

If the relative gap $\gamma = (\lambda_{k+1} - \lambda_k)/(\lambda_n - \lambda_1)$ is “large”, then the entries of P can be shown to decay exponentially away from the sparsity pattern of A , with larger γ leading to faster decay (see Sect. 4.2). Differentiating (79) with respect to a_{ij} shows again that the quantity in (79) is insensitive to small perturbations in positions of A that are far from the nonzero pattern of A . This fact has important consequences in quantum chemistry and solid state physics.

Although we have limited our discussion to Hermitian eigenvalue problems, an identical treatment applies more generally to normal matrices. In the nonnormal case, it is an open question whether decay results (for oblique spectral projectors) can be used to gain insight into the stability of isolated components of the spectrum, or of the pseudo-spectra [192], of a matrix. For a study of localization in the case of random nonnormal matrices, see [193].

4.1.4 Approximation of Matrix Functions

In most applications involving functions of large, sparse matrices, it is required to compute the vector $\mathbf{x} = f(A)\mathbf{b}$ for given $A \in \mathbb{C}^{n \times n}$ and $\mathbf{b} \in \mathbb{C}^{n \times 1}$. When $f(A) = A^{-1}$, this reduces to approximating the solution of a linear system. If \mathbf{b} is localized, for example $\mathbf{b} = \mathbf{e}_i$ (or a linear combination of a few standard basis vectors), then the decay in $f(A)$ leads to a localized solution vector \mathbf{x} . In this case, similar observations to the ones we made earlier about localized linear system solutions apply.

Suppose now that we want to compute a sparse approximation to $f(A_n)$, where $\{A_n\}$ is a sequence of banded or sparse matrices of increasing size. If the conditions of Theorems 8, 11 or 13 are satisfied, the entries of $f(A_n)$ are bounded in an exponentially decaying manner, with decay rates independent of n ; if f is entire, decay is superexponential. In all these cases Theorem 4 ensures that we can find a banded (or sparse) approximation to $f(A_n)$ to within an arbitrary accuracy $\varepsilon > 0$ in $O(n)$ work.

The question remains of *how* to compute these approximations. The above-mentioned theorems are based on the existence of best approximation polynomials $p_k(x)$, such that the error $\|p_k(A_n) - f(A_n)\|_2$ decays exponentially fast with the degree k . Under the assumptions of those theorems, for every $\varepsilon > 0$ one can determine a value of k , independent of n , such that $\|p_k(A_n) - f(A_n)\|_2 < \varepsilon$. Unfortunately, the form of the polynomial p_k is not known, except in very special cases. However, it is not necessary to make use of the polynomial of best approximation: there may well be other polynomials, also exponentially convergent to f , which can be easily constructed explicitly.

From here on we drop the subscript n and we work with a fixed matrix A , but the question of (in-)dependence of n should always be kept in mind. Suppose first that $A = A^*$ is banded or sparse, with spectrum in $[-1, 1]$; shifting and scaling A so that $\sigma(A) \subset [-1, 1]$ requires bounds on the extreme eigenvalues of A , which can usually be obtained in $O(n)$ work, for instance by carrying out a few Lanczos iterations. Let f be a function defined on a region containing $[-1, 1]$. A popular approach is polynomial approximation of $f(A)$ based on Chebyshev polynomials; see, e.g., [10, 91]. For many analytic functions, Chebyshev polynomials are known to converge very fast; for example, convergence is superexponential for $f(x) = e^x$ and other entire functions.

The following discussion is based on [21] (see also [168]). We start by recalling the matrix version of the classical three-term recurrence relation for the Chebyshev polynomials:

$$T_{k+1}(A) = 2AT_k(A) - T_{k-1}(A), \quad k = 1, 2, \dots \quad (80)$$

(with $T_0(A) = I$, $T_1(A) = A$). These matrices can be used to obtain an approximation

$$f(A) = \sum_{k=1}^{\infty} c_k T_k(A) - \frac{c_1}{2} I \approx \sum_{k=1}^N c_k T_k(A) - \frac{c_1}{2} I =: p_N(A)$$

to $f(A)$ by truncating the Chebyshev series expansion after N terms. The coefficients c_k in the expansion only depend on f (not on A) and can be easily computed numerically at a cost independent of n using the approximation

$$c_k \approx \frac{2}{M} \sum_{j=1}^M f(\cos(\theta_j)) \cos((k-1)\theta_j),$$

where $\theta_j = \pi(j - \frac{1}{2})/M$ with a sufficiently large value of M . Thus, most of the computational work is performed in (80). The basic operation in (80) is the matrix–matrix multiply. If the initial matrix A is m -banded, then after k iterations the matrix $T_{k+1}(A)$ will be km -banded. The *Paterson–Stockmeyer algorithm* can be used to evaluate polynomials in a matrix A with minimal arithmetic complexity, see [164] and [107, pp. 73–74]. We also mention [36], where sophisticated algorithms for matrix–matrix multiplication that take decay into account are developed.

In order to have a linear scaling algorithm, it is essential to fix a maximum bandwidth for the approximation $P_N(A)$, which must not depend on n . Then the cost is dominated by the matrix–matrix multiplies, and this is an $O(n)$ operation provided that the maximum bandwidth remains bounded as $n \rightarrow \infty$. Similar conclusions apply for more general sparsity patterns, which can be determined by using the structure of successive powers A^k of A . In alternative, dropping elements by size using a drop tolerance is often used, although rigorous justification of this procedure is more difficult.

Let us now consider the error incurred by the series truncation:

$$\|e_N(A)\|_2 = \|f(A) - P_N(A)\|_2, \quad (81)$$

where $P_N(A) = \sum_{k=1}^N c_k T_k(A) - \frac{c_1}{2} I$. We limit our discussion to the banded case, but the same arguments apply in the case of general sparsity patterns as well. Since $|T_k(x)| \leq 1$ for all $x \in [-1, 1]$ and $k = 1, 2, \dots$, we have that $\|T_k(A)\|_2 \leq 1$ for all k , since $\sigma(A) \subset [-1, 1]$. Using this well known property to bound the error in (81), we obtain that

$$\|e_N(A)\|_2 = \left\| \sum_{k=N+1}^{\infty} c_k T_k(A) \right\|_2 \leq \sum_{k=N+1}^{\infty} |c_k|.$$

The last inequality shows that the error defined by (81) only depends on the sum of the absolute values of the coefficients c_k for $k = N + 1, N + 2, \dots$, but these in turn do not depend on n , the dimension of the matrix we are approximating. Hence if we have a sequence of $n \times n$ matrices $\{A_n\}$ with $\sigma(A_n) \subset [-1, 1]$ for all n , we can use an estimate of the quantity $\sum_{k=N+1}^{\infty} |c_k|$ (see, for instance, [29, Eqs. (2.2)–(2.3)]) and use that to prescribe a sufficiently large bandwidth (sparsity pattern) to ensure a prescribed accuracy of the approximation. As long as the bandwidth of the approximation does not exceed the maximum prescribed bandwidth, the

error is guaranteed to be n -independent. In practice, however, this strategy is too conservative. Because of the rapid decay outside of the bandwidth of the original matrix, it is usually sufficient to prescribe a much smaller maximum bandwidth than the one predicted by the truncation error. This means that numerical dropping is necessary (see below for a brief discussion), since the bandwidth of $P_N(A)$ rapidly exceeds the maximum allowed bandwidth. Because of dropping, the simple error estimate given above is no longer rigorously valid. The numerical experiments reported in [21], however, suggest that n -independence (and therefore linearly scaling complexity and storage requirements) is maintained.

We now turn to the problem of approximating $f(A)$ for a general A with spectrum contained in an arbitrary continuum $\mathcal{F} \subset \mathbb{C}$; for a more detailed description of the technique we use, see [188]. In this case we can use a (Newton) interpolation polynomial of the form

$$P_N(A) = c_0I + c_1(A - z_0I) + c_2(A - z_0I)(A - z_1I) + \cdots + c_N(A - z_0I) \dots (A - z_{N-1}I)$$

where c_k is the divided difference of order k , i.e.,

$$c_k = f[z_0, \dots, z_k], \quad k = 0, \dots, N.$$

For $k = 0, \dots, N-1$, the interpolation points are chosen as $z_k = \Psi(\omega_k)$, where ω_k are the $N-1$ roots of the equation $\omega^{N-1} = \rho$ and $\Psi(z)$ is the inverse of the map $\Phi(z)$ that maps the complement of \mathcal{F} to the outside of a disk with radius ρ and satisfies the normalization conditions (33). This method does not require the computation of Faber polynomials and their coefficients. However, it does require knowledge of the map $\Psi(z)$. For specific domains \mathcal{F} this map can be determined analytically, see for example [30, 188]. In addition, $\Psi(z)$ may require information on the convex hull of the eigenvalues of A . For more general domains one may have to resort to numerical approximations to compute $\Psi(z)$; see [190]. Once again, the approximation algorithm requires mostly matrix–matrix multiplies with banded (or sparse) matrices and appropriate sparsification is generally required to keep the cost within $O(n)$ as the problem size n grows. A rigorous error analysis that takes dropping as well as truncation into account is however still lacking.

We briefly discuss now numerical dropping. The idea applies to more general sparsity patterns, but we restrict our discussion to the case where A is a banded matrix with bandwidth m . In this case we only keep elements inside a prescribed bandwidth \hat{m} in every iteration. For given ρ and R (see (41)) we choose \hat{m} a priori so as to guarantee that

$$(\rho/R)^{\hat{m}} \approx \varepsilon/K$$

where $K > 0$ is the constant for the bounds on $|[f(A)]_{ij}|$ (with $i \neq j$) appearing in Theorem 13 (for instance) and $\varepsilon > 0$ is a prescribed tolerance. As already noted, if A is banded with bandwidth m , then A^k has bandwidth km . This means that if we want the approximation to have a fixed bandwidth \hat{m} , where \hat{m} is (say) an integer multiple

of m corresponding to a prescribed approximation error ε , then we ought to truncate the expansion at the N^* th term, with $N^* = \hat{m}/m$. It may happen, however, that this value of N is actually too small to reduce the error below the prescribed threshold. In this case it is necessary to add extra terms to the Chebyshev expansion; but this would lead to an increase of the bandwidth beyond the prescribed limit. A solution that has been used by physicists is simply to continue the recurrence but ignoring all entries in positions outside the prescribed bandwidth. By restricting all the terms in the three-term recurrence (80) to have a fixed bandwidth (independent of n and N) we obtain an approximation scheme whose cost scales linearly in the size n of the problem. This, however, leaves open the problem of controlling the approximation error.

4.1.5 Approximation Based on Quadrature Rules

Another approach that can sometimes be used to find a banded or sparse approximation of a rapidly decaying matrix $f(A)$ is based on Gaussian quadrature [94]. With this approach it is possible to compute or estimate individual entries in $f(A)$. There exist also block versions of these techniques which allow the computation of several entries of $f(A)$ at once [94, 169]. Thus, if we know that only the entries of $f(A)$ within a certain bandwidth or block structure are nonnegligible, one can use Gaussian quadrature rules to estimate the entries within this bandwidth or blocks. This approach has been used, e.g., in [20] to construct simple banded preconditioners for Toeplitz matrices with decay, using the function $f(A) = A^{-1}$.

Here we briefly sketch this technique. Suppose f is strictly completely monotonic on an interval (a, b) (see Definition 3). For instance, the function $f(x) = x^{-\sigma}$ is strictly completely monotonic on $(0, \infty)$ for any $\sigma > 0$, and $f(x) = e^{-x}$ is strictly completely monotonic on \mathbb{R} .

Now, let $A = A^T \in \mathbb{R}^{n \times n}$. Consider the eigendecompositions $A = Q\Lambda Q^T$ and $f(A) = Qf(\Lambda)Q^T$. For $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ we have

$$\mathbf{u}^T f(A) \mathbf{v} = \mathbf{u}^T Q f(\Lambda) Q^T \mathbf{v} = \mathbf{p}^T f(\Lambda) \mathbf{q} = \sum_{i=1}^n f(\lambda_i) p_i q_i, \quad (82)$$

where $\mathbf{p} = Q^T \mathbf{u}$ and $\mathbf{q} = Q^T \mathbf{v}$. In particular, we have that $[f(A)]_{ij} = \mathbf{e}_i^T f(A) \mathbf{e}_j$.

Next, we rewrite the expression in (82) as a Riemann–Stieltjes integral with respect to the spectral measure:

$$\mathbf{u}^T f(A) \mathbf{v} = \int_a^b f(\lambda) d\mu(\lambda), \quad \mu(\lambda) = \begin{cases} 0, & \text{if } \lambda < a = \lambda_1, \\ \sum_{j=1}^i p_j q_j, & \text{if } \lambda_i \leq \lambda < \lambda_{i+1}, \\ \sum_{j=1}^n p_j q_j, & \text{if } b = \lambda_n \leq \lambda. \end{cases}$$

The general Gauss-type quadrature rule gives in this case:

$$\int_a^b f(\lambda) d\mu(\lambda) = \sum_{j=1}^N w_j f(t_j) + \sum_{k=1}^M v_k f(z_k) + R[f], \quad (83)$$

where the nodes $\{t_j\}_{j=1}^N$ and the weights $\{w_j\}_{j=1}^N$ are unknown, whereas the nodes $\{z_k\}_{k=1}^M$ are prescribed. We have

- $M = 0$ for the Gauss rule,
- $M = 1, z_1 = a$ or $z_1 = b$ for the Gauss–Radau rule,
- $M = 2, z_1 = a$ and $z_2 = b$ for the Gauss–Lobatto rule.

Also, for the case $\mathbf{u} = \mathbf{v}$, the remainder in (83) can be written as

$$R[f] = \frac{f^{(2N+M)}(\eta)}{(2N+M)!} \int_a^b \prod_{k=1}^M (\lambda - z_k) \left[\prod_{j=1}^N (\lambda - t_j) \right]^2 d\mu(\lambda), \quad (84)$$

for some $a < \eta < b$. This expression shows that, if $f(x)$ is strictly completely monotonic on an interval containing the spectrum of A , then quadrature rules applied to (83) give bounds on $\mathbf{u}^T f(A) \mathbf{v}$. More precisely, the Gauss rule gives a lower bound, the Gauss–Lobatto rule gives an upper bound, whereas the Gauss–Radau rule can be used to obtain both a lower and an upper bound. In particular, they can be used to obtain bounds on $[f(A)]_{ij}$. The evaluation of these quadrature rules is reduced to the computation of orthogonal polynomials via three-term recurrence, or, equivalently, to the computation of entries and spectral information on a certain tridiagonal matrix via the Lanczos algorithm. We refer to [20, 94] for details. Here we limit ourselves to observe that the conditions under which one can expect rapid decay of the off-diagonal entries of $f(A)$ also guarantee fast convergence of the Lanczos process. In practice, this means that under such conditions a small number N of quadrature nodes (equivalently, Lanczos steps) are sufficient to obtain very good estimates of the entries of $f(A)$. In numerical experiments, this number is usually between 5 and 10, see [18].

4.1.6 Error Bounds for Krylov Subspace Approximations

Another situation where the decay bounds for $f(A)$ have found application is in the derivation of error bounds for Krylov subspace approximations of $f(A)\mathbf{b}$, and in particular for the important case of the matrix exponential $f(A) = e^{-tA}\mathbf{b}$ [199, 203]. Recall that Krylov subspace methods are examples of polynomial approximation methods, where $f(A)\mathbf{b}$ is approximated by $p(A)\mathbf{b}$ for some (low-degree) polynomial p . Since every matrix function $f(A)$ is a polynomial in A , this is appropriate. The k th Krylov subspace of $A \in \mathbb{C}^{n \times n}$ and a nonzero vector $\mathbf{b} \in \mathbb{C}^n$ is defined by

$$\mathcal{K}_k(A, \mathbf{b}) = \text{span } \{\mathbf{b}, A\mathbf{b}, \dots, A^{k-1}\mathbf{b}\},$$

and it can be written as

$$\mathcal{K}_k(A, \mathbf{b}) = \{q(A)\mathbf{b} \mid q \text{ is a polynomial of degree } \leq k - 1\}.$$

The successive Krylov subspaces form a nested sequence:

$$\mathcal{K}_1(A, \mathbf{b}) \subset \mathcal{K}_2(A, \mathbf{b}) \subset \cdots \subset \mathcal{K}_d(A, \mathbf{b}) = \cdots = \mathcal{K}_n(A, \mathbf{b}).$$

Here d is the degree of the minimum polynomial of A with respect to \mathbf{b} . This is just the monic polynomial p of least degree such that $p(A)\mathbf{b} = \mathbf{0}$.

The basic idea behind Krylov methods is to project the given problem onto the successive Krylov subspaces, solving the (low-dimensional) projected problems, and expand the solution back to n -dimensional space to yield the next approximation. An orthonormal basis for a Krylov subspace can be efficiently constructed using the *Arnoldi process*; in the Hermitian case, this reduces to the Lanczos process (see [95, 174, 191]). Both of these algorithms are efficient implementations of the classical Gram–Schmidt process. In Arnoldi’s method, the projected matrix H_k has upper Hessenberg structure, which can be exploited in the computation. In the Hermitian case, H_k is tridiagonal.

Denoting by $\mathcal{Q}_k = [\mathbf{q}_1, \dots, \mathbf{q}_k] \in \mathbb{C}^{n \times k}$, with $\mathbf{q}_1 = \mathbf{b}/\|\mathbf{b}\|_2$, the orthonormal basis for the k th Krylov subspace produced by the Arnoldi process, the k th approximation to the solution vector $f(A)\mathbf{b}$ is computed as

$$\mathbf{x}_k := \|\mathbf{v}\|_2 Q_k f(H_k) \mathbf{e}_1 = Q_k f(H_k) Q_k^* \mathbf{v}. \quad (85)$$

Typically, $k \ll n$ and computing $f(H_k)$ is inexpensive, and can be carried out in a number of ways. For instance, when $H_k = H_k^* = T_k$ (a tridiagonal matrix), it can be computed via explicit diagonalization of T_k . More generally, methods based on the Schur form of H_k can be used [95, Sect. 9.1.4].

The main remaining issue is to decide when to stop the iteration. Much effort has been devoted in recent years to obtained bounds for the error $\|\mathbf{x}_k - f(A)\mathbf{b}\|_2$. As it turns out, in the case of the matrix exponential $f(A) = e^{-tA}$ the approximation error is mainly governed by the quantity

$$h(t) = \mathbf{e}_k^T e^{-tH_k} \mathbf{e}_1, \quad (86)$$

i.e., by the last entry in the first column of e^{-tH_k} . Since H_k is upper Hessenberg (in particular, tridiagonal if $A = A^*$), the bottom entry in the first column of e^{-tH_k} should be expected to decay rapidly to zero as k increases. In [199] the authors show how the decay bounds in Theorem 13, combined with estimates of the field of values of A obtained from a clever use of the Bendixson–Hirsch Theorem, can lead to fairly tight bounds on $|h(t)|$, which in turn leads to an explanation of the superlinear convergence behavior of Krylov methods. These results can be seen as a generalization to the nonnormal case of the bounds obtained in [109] and [203] for the Hermitian case.

4.1.7 Exponentials of Stochastic Matrices

A real $n \times n$ matrix S is *row-stochastic* if it has nonnegative entries and its row sums are all equal to 1. It is *doubly stochastic* if its column sums are also all equal to 1. Such matrices arise as transition matrices of discrete Markov chains, and play an important role in the analysis of large graphs and networks (see, e.g., [130] and [31]).

In the study of diffusion-type and other dynamical processes on graphs, it is often necessary to perform computations involving matrix exponentials of the form e^{tS} , where $t \in \mathbb{R}$; see, e.g., [89, p. 357] and [90]. In some cases, one is interested in approximating selected columns of e^{tS} . Note that this is a special case of the problem of computing a matrix function times a given vector, where the given vector is now of the form \mathbf{e}_i . If the entries in the i th column of e^{tS} are strongly localized, it may be possible to compute reasonable approximations very cheaply. This is crucial given the often huge size of graphs arising from real-world applications, for example in information retrieval.

For sparse matrices corresponding to graphs with maximum degree uniformly bounded in n , the decay theory for matrix functions guarantees superexponential decay in the entries of e^{tS} . This means that each column of e^{tS} only contains $O(1)$ nonnegligible entries (with a prefactor depending on the desired accuracy ε , of course). Localized computations such as those investigated in [90] may be able to achieve the desired linear, or even sublinear, scaling. Note, however, that the bounded maximum degree assumption is not always realistic. Whether strong localization can occur without this assumption is an open question. While it is easy to construct sparse graphs which violate the condition and lead to delocalized exponentials, the condition is only a sufficient one and it may well happen that e^{tS} remains localized even if the maximum degree grows as $n \rightarrow \infty$.

Localization in e^{tS} is also linked to localization in the PageRank vector \mathbf{p} , the unique stationary probability distribution vector (such that $\mathbf{p}^T = \mathbf{p}^T S$) associated with an irreducible row-stochastic matrix S [89, 130, 157]. When S is the “Google matrix” associated with the World Wide Web, however, the decay in the entries of \mathbf{p} does not appear to be exponential, but rather to satisfy a power law of the form $p_k = O(k^{-\gamma})$ for $k \rightarrow \infty$, assuming the entries are sorted in nonincreasing order. The value of γ is estimated to be approximately 2.1; see [130, p. 110]. This fact reflects the power law nature of the degree distribution of the Web. General conditions for strong localization in seeded PageRank vectors are discussed in [157].

A completely different type of stochastic process leading to exponentials of very large, structured matrices is the Markovian analysis of queueing networks, see [31, 32]. In [32] the authors study the exponential of huge block upper triangular, block Toeplitz matrices and show that this matrix function satisfies useful decay properties that can be exploited in the computations, leading to efficient algorithms.

4.1.8 Exponential Integrators

Finally, we mention that the decay properties of the exponential of banded matrices have recently been used in [38] to develop and analyze a class of domain decomposition methods for the integration of time-dependent PDEs [38] and in the analysis of an infinite Arnoldi exponential integrator for systems of ODEs [126].

4.2 Linear Scaling Methods for Electronic Structure Computations

In quantum chemistry and solid state physics, one is interested in determining the *electronic structure* of (possibly large) atomic and molecular systems [145]. The problem amounts to computing the ground state (smallest eigenvalue and corresponding eigenfunction) of the many-body quantum-mechanical Hamiltonian (Schrödinger operator), \mathbf{H} . In variational terms, we want to minimize the Rayleigh quotient:

$$E_0 = \min_{\Psi \neq 0} \frac{\langle \mathbf{H}\Psi, \Psi \rangle}{\langle \Psi, \Psi \rangle} \quad \text{and} \quad \Psi_0 = \operatorname{argmin}_{\Psi \neq 0} \frac{\langle \mathbf{H}\Psi, \Psi \rangle}{\langle \Psi, \Psi \rangle} \quad (87)$$

where $\langle \cdot, \cdot \rangle$ denotes the L^2 inner product. In the Born–Oppenheimer approximation, the many-body Hamiltonian is given (in atomic units) by

$$\mathbf{H} = \sum_{i=1}^{n_e} \left(-\frac{1}{2} \Delta_i - \sum_{j=1}^M \frac{Z_j}{|\mathbf{x}_i - \mathbf{r}_j|} + \sum_{j \neq i}^{n_e} \frac{1}{|\mathbf{x}_i - \mathbf{x}_j|} \right)$$

where n_e = number of electrons and M = number of nuclei in the system. The electron positions are denoted by \mathbf{x}_i , those of the nuclei by \mathbf{r}_j ; as usual, the charges are denoted by Z_j . The operator \mathbf{H} acts on a suitable subspace of $H^1(\mathbb{R}^{3n_e})$ consisting of anti-symmetric functions (as a consequence of Pauli’s Exclusion Principle for fermions). Here, the spin is neglected in order to simplify the presentation.

Unless n_e is very small, the “curse of dimensionality” makes this problem intractable; even storing the wave function Ψ becomes impossible already for moderately-sized systems [123]. In order to make the problem more tractable, various approximations have been devised, most notably:

- Wave function methods (e.g., Hartree–Fock);
- Density Functional Theory (e.g., Kohn–Sham);
- Hybrid methods (e.g., B3LYP).

In these approximations the original, linear eigenproblem $\mathbf{H}\Psi = E\Psi$ for the many-electrons Hamiltonian is replaced by a non-linear one-particle eigenproblem:

$$\mathcal{F}(\psi_i) = \lambda_i \psi_i, \quad \langle \psi_i, \psi_j \rangle = \delta_{ij}, \quad 1 \leq i, j \leq n_e, \quad (88)$$

where $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n_e}$ are the n_e smallest eigenvalues of (88). In the case of Density Functional Theory, (88) are known as the *Kohn–Sham equations*, and the nonlinear operator \mathcal{F} in (88) has the form $\mathcal{F}(\psi_i) = (-\frac{1}{2}\Delta + V(\rho))\psi_i$, where $\rho = \rho(\mathbf{x}) = \sum_{i=1}^{n_e} |\psi_i(\mathbf{x})|^2$ is the *electronic density*, a function of only three variables that alone is sufficient to determine, in principle, all the properties of a system [110, 124]. The Kohn–Sham equations (88) are the Euler–Lagrange equations for the minimization of a functional $J = J[\rho]$ (the *density functional*) such that the ground state energy, E_0 , is the minimum of the functional: $E_0 = \inf_{\rho} J$. While the exact form of this functional is not known explicitly, a number of increasingly accurate approximations have been developed since the original paper [124] appeared. The enormous success of Density Functional Theory (which led to the award of a share of the 1998 Nobel Prize for Chemistry to Kohn) is due to the fact that the high-dimensional, intractable minimization problem (87) with respect to $\Psi \in L^2(\mathbb{R}^{3n_e})$ is replaced by a minimization problem with respect to $\rho \in L^2(\mathbb{R}^3)$.

The nonlinear Kohn–Sham equations (88) can be solved by a “self-consistent field” (SCF) iteration, leading to a sequence of *linear* eigenproblems

$$\mathcal{F}^{(k)}\psi_i^{(k)} = \lambda_i^{(k)}\psi_i^{(k)}, \quad \langle \psi_i^{(k)}, \psi_j^{(k)} \rangle = \delta_{ij}, \quad k = 1, 2, \dots \quad (89)$$

($1 \leq i, j \leq n_e$), where each $\mathcal{F}^{(k)} = -\frac{1}{2}\Delta + V^{(k)}$ is a one-electron Hamiltonian with potential

$$V^{(k)} = V^{(k)}(\rho^{(k-1)}), \quad \rho^{(k-1)} = \sum_{i=1}^{n_e} |\psi_i^{(k-1)}(\mathbf{x})|^2.$$

Solution of each of the (discretized) linear eigenproblems (89) leads to a typical $O(n_e^3)$ cost per SCF iteration. However, the actual eigenpairs $(\psi_i^{(k)}, \lambda_i^{(k)})$ are unnecessary; hence, diagonalization of the (discretized) one-particle Hamiltonians can be avoided. Indeed, all one really needs is the *density matrix*, i.e., the spectral projector P onto the invariant subspace

$$V_{occ} = \text{span}\{\psi_1, \dots, \psi_{n_e}\}$$

corresponding to the n_e lowest eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n_e}$ (“occupied states”). At the k th SCF cycle, an approximation $P^{(k)} \approx P$ to the orthogonal projector onto the occupied subspace V_{occ} needs to be computed. At convergence, all quantities of interest in electronic structure theory can be computed from P .

In practice, operators are replaced by matrices by Galerkin projection onto a finite-dimensional subspace spanned by a set of basis functions $\{\phi_i\}_{i=1}^n$, where n is a multiple of n_e . Typically, $n = n_b \cdot n_e$ where $n_b \geq 2$ is a moderate constant when linear combinations of Gaussian-type orbitals are used; often, $n_b \approx 10 - 25$ (see [132]). We assume that the basis functions are localized, so that the resulting discrete Hamiltonians (denoted by H) are, up to some small truncation tolerance, sparse. Finite difference approximations can also be used, in which case the sparsity pattern

is that of the discrete Laplacian, since the potential is represented by a diagonal matrix.

Non-orthogonal bases are easily accommodated into the theory but they may lead to algorithmic complications. They are often dealt with by a congruence transformation to an orthogonal basis, which can be accomplished via an inverse-Cholesky factorization; the transformed Hamiltonian is $\hat{H} = Z^T H Z$ where $S^{-1} = ZZ^T$ is the inverse Cholesky factorization of the *overlap matrix*,

$$S = [S_{ij}], \quad S_{ij} = \int_{\Omega} \phi_i(\mathbf{r}) \phi_j(\mathbf{r}) d\mathbf{r}.$$

The inverse factor Z can be efficiently approximated by the AINV algorithm [25, 49]. Often S is strongly localized and has condition number independent of n . As we have seen, under these conditions its inverse (and therefore Z) decays exponentially fast, with a rate independent of n ; see Theorem 24. Hence, up to a small truncation tolerance Z is sparse and so is \hat{H} , see [26, p. 49]. In alternative, Z can be replaced by the inverse square root $S^{-1/2}$ of S ; transformation from H to $S^{-1/2} H S^{-1/2}$ is known as *Löwdin orthogonalization*. Again, localization of $S^{-1/2}$ is guaranteed if S is banded, sparse, or localized and well-conditioned, so a sparse approximation to $S^{-1/2}$ is possible. It is important to stress that transformation of H into \hat{H} need not be carried out explicitly in most linear scaling algorithms. Rather, the transformed matrix is kept in factored form, similar to preconditioning. From here on, we assume that the transformation has already been performed and we denote the representation of the discrete Hamiltonian in the orthogonal basis by H instead of \hat{H} .

Thus, the fundamental problem of (zero-temperature) electronic structure theory has been reduced to the approximation of the spectral projector P onto the subspace spanned by the n_e lowest eigenfunctions of H (occupied states):

$$P = \psi_1 \otimes \psi_1 + \cdots + \psi_{n_e} \otimes \psi_{n_e}, \quad (90)$$

where $H\psi_i = \lambda_i \psi_i$, $i = 1, \dots, n_e$. Note that we can write $P = h(H)$, where f is the Heaviside (step) function

$$h(x) = \begin{cases} 1 & \text{if } x < \mu \\ 0 & \text{if } x > \mu \end{cases}$$

with $\lambda_{n_e} < \mu < \lambda_{n_e+1}$ (μ is the “Fermi level”). Alternatively, we can write $P = (I - \text{sign}(H - \mu I))/2$, where sign denotes the sign function ($\text{sign}(x) = 1$ if $x > 0$, $\text{sign}(x) = -1$ if $x < 0$). We will come back to this representation at the end of this section.

As usual, we can assume that H has been scaled and shifted so that $\sigma(H) \subset [-1, 1]$. If the spectral gap $\gamma = \lambda_{n_e+1} - \lambda_{n_e}$ is not too small, h can be well approximated by a smooth function with rapid decrease from 1 to 0 within the gap

$(\lambda_{n_e}, \lambda_{n_e+1})$. A common choice is to replace h by the Fermi–Dirac function

$$f(x) = \frac{1}{1 + e^{\beta(x-\mu)}}, \quad (91)$$

which tends to a step function as the parameter β increases.

Physicists have observed long ago that for “gapped systems” (like insulators and, under certain conditions, semiconductors) the entries of the density matrix P decay exponentially fast away from the main diagonal, reflecting the fact that interaction strengths decrease rapidly with the distance [9, 69, 105, 116, 121, 132, 146, 158, 165, 166]. We recall that, as already discussed, exponential decay in the ψ_i in (90) is sufficient for localization of the density matrix P , but not necessary; and indeed, situations can be found where P decays exponentially but the ψ_i do not, see [44] and the discussion in [26, Sect. 4].

Localization of the density matrix is a manifestation of the “nearsightedness” of electronic matter discussed in Sect. 1.1.¹⁵ Localization is crucial as it provides the basis for so-called *linear scaling* (i.e., $O(n_e)$) methods for electronic structure calculations. These methods have been vigorously developed since the early 1990s, and they are currently able to handle very large systems, see, e.g., [9, 10, 37, 42, 49, 91, 122, 128, 133, 134, 138, 159, 172, 175, 202]. These methods include expansion of the Fermi–Dirac operator $f(H)$ in the Chebyshev basis, constrained optimization methods based on density matrix minimization (possibly with ℓ^1 regularization to enforce localized solutions), methods based on the sign matrix representation of P (such as “McWeeney purification”), multipole expansions, and many others. As we have seen, rapidly decaying matrices can be approximated by sparse matrices, uniformly in n . Hence, rigorously establishing the rate of decay in the density matrix provides a sound mathematical foundation for linear scaling methods in electronic structure calculations. A mathematical analysis of the asymptotic decay properties of spectral projectors associated with large, banded or sparse Hermitian matrices has been presented in [26]. The main result in [26] can be summarized in the following theorem.

Theorem 26 ([26]) *Let $n = n_b \cdot n_e$ where n_b is a fixed positive integer and the integers n_e form a monotonically increasing sequence. Let $\{H_n\}$ be a sequence of Hermitian $n \times n$ matrices with the following properties:*

1. *Each H_n has bandwidth m independent of n ;*

¹⁵As we saw earlier (see (79)), rapid decay in P means that quantities like $\text{Tr}(PH)$, which in electronic structure theory has the interpretation of a single particle-energy [91, 159], are insensitive to small perturbations in the Hamiltonian in positions that correspond to small entries in P . Also, the fact that $\text{Tr}(P) = \text{rank}(P) = n_e \ll n$ implies that many of the diagonal entries of P will be tiny; hence, slight changes in the potential $V(\mathbf{x})$ at a point \mathbf{x} are only felt locally, see Example 2.

2. There exist two (fixed) intervals $I_1 = [a, b]$, $I_2 = [c, d] \subset \mathbb{R}$ with $\gamma = c - b > 0$ such that for all $n = n_b \cdot n_e$, I_1 contains the smallest n_e eigenvalues of H_n (counted with their multiplicities) and I_2 contains the remaining $n - n_e$ eigenvalues.

Let P_n denote the $n \times n$ spectral projector onto the subspace spanned by the eigenvectors associated with the n_e smallest eigenvalues of H_n , for each n . Then there exist constants $K > 0$, $\alpha > 0$ independent of n such that

$$|[P_n]_{ij}| \leq K e^{-\alpha|i-j|}, \quad \forall i \neq j. \quad (92)$$

Moreover, for any $\varepsilon > 0$ there is a matrix \tilde{P}_n of bandwidth p independent of n such that $\|P_n - \tilde{P}_n\| < \varepsilon$, for all n .

As usual, the bandedness assumption can be replaced with the assumption that the Hamiltonians are sparse, with associated graphs that satisfy the bounded maximum degree assumption. In this case the geodesic distance on the graphs should be used to measure decay.

Different proofs of Theorem 26 can be found in [26]. One approach consists in approximating P_n via the (analytic) Fermi–Dirac function (91), and exploiting the fact that the rate of decay in $f(H_n)$ is independent of n thanks to the non-vanishing gap assumption. This approach yields explicit, computable formulas for the constants K and α in (92), see [26, pp. 26–27]. Another approach is based on results on the polynomial approximation of analytic functions on disjoint intervals [58, 104]. Both proofs make crucial use of the general theory of exponential decay in analytic functions of banded and sparse matrices discussed earlier, in particular Theorem 9.

Some comments about the meaning of Theorem 26 are in order. The first thing to observe is that the independence of the decay bounds on n follows from the assumption that $n = n_b \cdot n_e$ where n_b , which controls the accuracy of the discretization, is fixed, whereas n_e , which determines the system size (i.e., the number of electrons in the system), is allowed to increase without bounds. This is sometimes referred to as the *thermodynamic limit*, where the system size grows but the distance between atoms is kept constant. It should not be confused with the limit in which n_e is fixed and $n_b \rightarrow \infty$, or with the case where both n_e and n_b are allowed to grow without bounds. Keeping n_b constant ensures that the spectra of the discrete Hamiltonians remain uniformly bounded, which guarantees (for insulators) that the relative spectral gap γ does not vanish as $n \rightarrow \infty$. In practice a constant n_b is a reasonable assumption since existing basis sets are not very large and they are already highly optimized so as to achieve the desired accuracy.

Another issue that warrants comment is the choice of the parameter β (the “inverse temperature”) in the Fermi–Dirac function. Note that the Fermi–Dirac function has two poles in the complex plane: if we assume, without loss of generality, that the Fermi level is at $\mu = 0$, the poles are on the imaginary axis at $\pm i\pi/\beta$. Since the rate of decay is governed by the distance between these two poles and the smallest intervals I_1, I_2 containing all the spectra $\sigma(H_n)$, a large value of β (corresponding to a small gap γ) means that the poles approach 0 and therefore b and

c. In this case, decay could be slow; indeed, the rate of decay α in the bounds (92) tends to zero as $\gamma \rightarrow 0$ or, equivalently, as $\beta \rightarrow \infty$. On the other hand, a relatively large value of γ means that β can be chosen moderate and this will imply fast decay in the entries of the density matrix P_n , for all n . We refer to [26, pp. 40–42] for details on the dependence of the decay rate in the density matrix as a function of the gap and of the temperature, in particular in the zero-temperature limit ($\beta \rightarrow \infty$), and to [187] for an example of how these results can be used in actual computations.

The case of metallic systems at zero temperature corresponds to $\gamma \rightarrow 0$. In this case the bound (92) becomes useless, since $\alpha \rightarrow 0$. The actual decay in the density matrix in this case can be as slow as $(1 + |i - j|)^{-1}$; see [26, Sect. 10] for a detailed analysis of a model problem.

We conclude this section discussing alternative representations of the density matrix that could potentially lead to better decay bounds. Recall that the step function $h(x)$ can be expressed in terms of the sign function as $h(x) = (1 - \text{sign}(x))/2$. Hence, studying the decay behavior of the spectral projector $h(H)$ amounts to studying the decay behavior in $\text{sign}(H)$. Again, we assume for simplicity that the Fermi level is at $\mu = 0$. Now, the matrix sign function admits a well known integral representation, see [107]:

$$\text{sign}(H) = \frac{2}{\pi} H \int_0^\infty (t^2 I + H^2)^{-1} dt. \quad (93)$$

One can now use available decay bounds on the inverse of the banded (or sparse) Hermitian positive definite matrix $t^2 I + H^2$ together with quadrature rules to obtain bounds on the entries of $\text{sign}(H)$ and therefore of P . Note the similarity of this approach with the one reviewed in Sect. 3.5.

Another approach, which yields more explicit bounds, is based on the identity

$$\text{sign}(H) = H(H^2)^{-1/2}, \quad (94)$$

see again [107]. We can then use bounds for the inverse square root of the banded (or sparse) Hermitian positive definite matrix H^2 to obtain exponential decay bounds for $\text{sign}(H)$, using the fact that the product of a banded (or sparse) matrix times an exponentially decaying one retains the exponential decay property. Preliminary numerical experiments on simple model gapped Hamiltonians (Boito, Private communication, 2015) suggest that the decay bounds obtained via the representations (93) and (94) can be more accurate than those obtained via the Fermi–Dirac representation.

4.3 Further Applications

In this section we briefly mention a few other applications of localization and decay bounds in applied mathematics and physics.

4.3.1 Localized Solutions to Matrix Equations

In the area of control of discrete-time, large-scale dynamical systems, a central role is played by the *Lyapunov equation* associated to a linear, time-varying dynamical system:

$$AX + XA^T = P, \quad (95)$$

with $A, P \in \mathbb{R}^{n \times n}$ given matrices and X unknown. If A is stable, Eq. (95) has a unique solution, which can be expressed as

$$X = - \int_0^\infty e^{tA} P e^{tA} dt; \quad (96)$$

and also as the solution of a linear system in Kronecker sum form:

$$(I \otimes A + A^T \otimes I) \text{vec}(X) = \text{vec}(P), \quad (97)$$

where “ vec ” is the operator that takes an $n \times n$ matrix and forms the vector of length n^2 consisting of the columns of the matrix stacked one on top of one another; see, e.g. [131] or [181].

Several authors (e.g., [51, 103]) have observed that whenever A and P are banded, or sparse, the solution matrix X is localized, with rapid off-diagonal decay if A is well-conditioned. Moreover, the decay is oscillatory (see [103, Fig. 3]). This does not come as a surprise given the relation of X to the matrix exponential of A , see (96), and to the inverse of a matrix in Kronecker sum form, see (97). The decay in the solution has been exploited to develop efficient solution techniques for (95) with A and P banded. When $A = A^T$, an approximate solution to (95) can sometimes be obtained using polynomial expansion in the Chebyshev basis, as outlined in the previous section. We refer again to [103] for details.

Localized matrices (in the form of rapidly decaying inverse Gramians) also arise in another problem in control theory, namely, subspace identification of large-scale interconnected systems. Again, this fact can be exploited to develop fast approximation algorithms; see [102].

For a different application of exponential decay bounds for A^{-1} to the study of the behavior of dynamical systems, see [153].

4.3.2 Localization in Graph and Network Analysis

Recently, several authors have proved that, with high probability, the eigenvectors of the adjacency matrix [65, 73, 189] and of the Laplacian [43] of large sparse undirected random graphs, in particular Erdős–Rényi graphs, are delocalized, a fact that was already known on the basis of empirical observation; see, e.g., [93]. On the other hand, localization in eigenvectors of scale-free networks, particularly

those corresponding to the largest eigenvalues of the adjacency matrix or of the Laplacian, has been reported, for both synthetic [93, 149] and real-world [106, 156, 178] networks. For instance, in [93] the eigenvector corresponding to the largest eigenvalue of the adjacency matrix of a power-law graph was found to be localized at the *hub* (the node of maximum degree).

Power-law graphs are also studied in [106], where a class of graph substructures leading to locally supported eigenvectors is identified. In some cases these eigenvectors are actually *sparse*, not just localized: an extreme example is given by the so-called *Faria vectors* [82], i.e., eigenvectors that have only two nonzero components (one positive, the other necessarily negative). These eigenvectors are associated with eigenvalues of very high multiplicity (sometimes as large as $O(n)$), such as those associated with star graphs—graphs consisting of a central node connected to several peripheral nodes. Many star-like subgraphs, and thus Laplacian eigenvalues of very high multiplicity, are often found in real-world scale-free graphs, an observation that leads to the conjecture that such Laplacians may have a number of locally supported eigenvectors. We refer to [106] for a detailed study, including computational aspects. It should be noted that since power-law graphs do not satisfy the bounded maximum degree assumption, we cannot directly apply the decay theory for matrix functions (and in particular for spectral projectors) to explain the eigenvector localization discussed in [93, 106].

Another class of graphs for which eigenvector localization has been observed is discussed in [156], motivated by the study of dendrites of retinal ganglion cells (RGCs). It turns out the Laplacian eigenvalues of a typical dendritic tree display a peculiar distribution: most of the λ_i are distributed according to a smooth curve in the interval $(0, 4)$, after which a jump occurs in the spectrum and the remaining eigenvalues are clustered around some value larger than 4. Moreover, the eigenvectors associated with eigenvalues less than 4 are delocalized, while those associated with eigenvalues greater than 4 are exponentially localized.

In order to find an explanation to this phenomenon, the authors of [156] consider simplified models of RGCs that are easier to analyze but at the same time capture the main properties of RGCs. The simplest among these models is a *star-like tree*, obtained by connecting one or more path graphs to the central hub of a star graph S_k (consisting of k peripheral nodes connected to a central node), where $k \geq 3$. In the case of a single path graph P_ℓ connected to the central hub of a star graph S_k , the resulting graph is called a *comet* of type (k, ℓ) . The Laplacian of the resulting star-like tree is then obtained by *gluing* the Laplacians of S_k and P_ℓ . For example, in the case $k = 4, \ell = 5$ the corresponding comet graph (with the hub numbered first)

has 10 nodes and the associated Laplacian is given by

$$L = \left[\begin{array}{cc|ccccc} 5 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 0 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{array} \right],$$

where the horizontal and vertical line have been added to more clearly show how the gluing of the Laplacians of S_4 and P_5 is carried out. The resulting L has nine eigenvalues < 4 , all falling between 0 and 3.652007105, with the tenth eigenvalue $\lambda \approx 6.0550$. It is known that a star-like tree can have only one eigenvalue ≥ 4 , with equality if and only if the graph is a *claw* (see [156] for details). In our example the (normalized) eigenvector associated with the largest eigenvalue is of the form

$$\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2],$$

where

$$\mathbf{v}_1 = [0.9012, -0.1783, -0.1783, -0.1783, -0.1783],$$

and

$$\mathbf{v}_2 = [0.2377, -0.0627, 0.0165, -0.0043, 0.0008]$$

(rounding to four decimal places). We see that the dominant eigenvector is mostly concentrated at the hub, is constant on the peripheral nodes of the star S_4 , and decays monotonically in magnitude along the path P_5 away from the hub. It can be shown that this phenomenon is common to all comet graphs and that the decay of the dominant eigenvector along the path is exponential [156]. Moreover, similar behavior is also common to star-like trees in general; if there are multiple paths connected to the central node of a star, the dominant eigenvector will decay exponentially along the paths, away from the central hub, where the eigenvector is concentrated. The remaining eigenvectors, corresponding to eigenvalues less than 4, are delocalized (oscillatory).

The proofs in [156] are direct and are based on previous results on the eigenvalues of star-like trees, together with a careful analysis of certain recurrences that the entries of the dominant eigenvector of L must satisfy. Here we point out that the decay behavior of the dominant eigenvector of L for star-like trees (and also for

more general, less structured graphs obtained by gluing one or more long paths to a graph having a hub of sufficiently high degree) is a byproduct of our general decay results for functions of banded or sparse matrices. To see this, we consider the case of a single path of length ℓ attached to the hub of a graph with order $k + 1$ nodes. The resulting graph has $n = k + \ell + 1$ nodes, and its Laplacian is of the form

$$L_n = \begin{bmatrix} L_{11} & L_{12} \\ L_{12}^T & L_{22} \end{bmatrix},$$

where L_{12} is a $(k + 1) \times \ell$ matrix with all its entries equal to 0 except for a -1 in the upper left corner. For fixed k and increasing ℓ , the sequence $\{L_n\}$ satisfies the assumptions of Theorem 9, therefore for any analytic function f , the entries of $f(L_n)$ must decay at least exponentially fast away from the main diagonal (or nonzero pattern of L_n) with rate independent of n for ℓ sufficiently large. Moreover, assume that the dominant eigenvalue of L_n is well separated from the rest of the spectrum; this happens for example if one of the nodes, say the first, has a significantly higher degree than the remaining ones. Then the corresponding eigenvector is localized, and its entries decay exponentially along the path, away from the hub. To see this we can approximate the corresponding spectral projector, P , by a Gaussian in L_n and use the fact that $\text{Tr}(P) = 1$, as discussed earlier.

Next, we consider the issue of localization in functions of matrices associated with graphs, limited to the undirected case. We are especially interested in the matrix exponential, which is widely used in the study of network structure and dynamics. We discuss first the communicability between nodes, see (7), which is measured by the entries of the exponential of the adjacency matrix. In many applications, see for example [81], it is desirable to identify pairs of nodes within a network having low communicability. For fairly regular sparse graphs with large diameter, communication between neighboring nodes is clearly much easier than communication between pairs of distant nodes, and this fact is well captured by the fact that $[e^A]_{ij}$ decays superexponentially with the geodesic distance $d(i,j)$. The simplest example is that of a path graph P_ℓ , for which A is tridiagonal: as we have shown, in this case the entries of the exponential decay very fast with the distance $|i - j|$. On the other hand, for large connected graphs of small diameter, like many real world complex networks, the distance $d(i,j)$ is small for any i and j , and decay bounds like (29) cannot predict any small entries in functions of the adjacency matrix. This fact is related to the violation of the bounded maximum degree assumption. The simplest example is now that of a star graph S_n , which has diameter 2 and maximum degree n , and for which there is no decay whatsoever in e^A .

Analogous remarks apply to functions of the Laplacian of a connected graph. Consider for instance the heat kernel, e^{-tL} . As is well known, this matrix function occurs in the solution of initial value problems of the form

$$\dot{\mathbf{x}} = -L\mathbf{x} \quad \text{subject to} \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (98)$$

where the dot on a vector denotes differentiation with respect to time and $\mathbf{x}_0 \in \mathbb{R}^n$ is given. The solution of (98) is given, for all $t \geq 0$, by

$$\mathbf{x}(t) = e^{-tL}\mathbf{x}_0.$$

Consider now the special case where $\mathbf{x}_0 = \mathbf{e}_i$, the i th vector of the standard basis. This means that a unit amount of some “substance” is placed at node i at time $t = 0$, the amount at all other nodes being zero. Alternatively, we could think of node i being at 1° temperature, with all other nodes having 0° temperature initially. Then the j th entry of the solution vector $\mathbf{x}(t)$ represents the fraction of the substance that has diffused to node j at time t or, alternatively, the temperature reached by node j at time t . This quantity is given by

$$x_j(t) = [e^{-tL}]_{ij}.$$

Note that

$$\lim_{t \rightarrow \infty} x_j(t) = \frac{1}{n} \quad \forall j = 1, \dots, n,$$

and moreover this limit is independent of i . This fact simply means that asymptotically, the system is at thermal equilibrium, with all the initial “substance” (e.g., heat) being equally distributed among the nodes of the network, regardless of the initial condition.

Another interpretation is possible: observing that e^{-tL} is a (doubly stochastic) matrix for all $t \geq 0$, we can interpret its entries $[e^{-tL}]_{ij}$ as transition probabilities for a Markov chain, namely, for a (continuous-time) random walk on the graph. Then $[e^{-tL}]_{ij}$ has the meaning of the probability of a “walker” being at node j at time t given that it was at node i at time $t = 0$.

No matter what the interpretation is, we see that the entries of e^{-tL} can serve as a measure of *communicability over time* between pairs of nodes in a network. We note that for $t = 1$ and regular graphs, this measure is identical (up to a constant factor) to the earlier notion of communicability (7). Note that for fairly regular, large diameter, sparse, “grid-like” graphs and for fixed $t > 0$ the entries $[e^{-tL}]_{ij}$ decay superexponentially fast as the geodesic distance $d(i, j)$ increases, reflecting the fact that after a finite time only a relatively small fraction of the diffusing substance will have reached the furthest nodes in the network. Clearly, this amount increases with time. The rate of convergence to equilibrium is governed by the spectral gap (i.e., the smallest nonzero eigenvalue of L): the smaller it is, the longer it takes for the system to reach equilibrium. Since for grid-like graphs the smallest eigenvalue goes to zero rapidly as $n \rightarrow \infty$ ($\lambda_2(L) = O(n^{-2})$), convergence to equilibrium is slow. On the other hand, graphs with good expansion properties tend to have large spectral gap and equilibrium is reached much faster, even for large n . This is reflected in the fact that e^{-tL} is usually delocalized for such graphs. Note, however, that things are

more complicated in the case of *weighted graphs*, or if a *normalized Laplacian*

$$\widehat{L} = I - D^{-1/2}AD^{-1/2} \quad \text{or} \quad \widetilde{L} = I - D^{-1}A$$

is used instead of L (see for instance [90]).

Finally, we consider a “quantum” form of network communicability, obtained by replacing the diffusion-type equation (98) with the Schrödinger-type equation:

$$i\dot{\mathbf{x}} = L\mathbf{x}, \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (99)$$

where $i = \sqrt{-1}$, $\mathbf{x}_0 \in \mathbb{C}^n$ is given and such that $\|\mathbf{x}_0\|_2 = 1$. As we have seen, the solution of (99) is given by

$$\mathbf{x}(t) = e^{-itL}\mathbf{x}_0, \quad \forall t \in \mathbb{R}.$$

Note that since $U(t) = e^{-itL}$ is unitary, the solution satisfies $\|\mathbf{x}(t)\|_2 = 1$ for all $t \in \mathbb{R}$. Consider now the matrix family $\{S(t)\}_{t \in \mathbb{R}}$ defined as follows:

$$S(t) = [S(t)]_{ij}, \quad [S(t)]_{ij} = |[U(t)]_{ij}|^2 \quad \forall i, j. \quad (100)$$

Since $U(t)$ is unitary, $S(t)$ is doubly stochastic. Its entries are transition probabilities: they measure how likely a system (say, a particle) governed by Eq. (99), initially in state i , is to be in state j at time t . Here we have identified the nodes of the underlying graph with the states of the system. We see again that the magnitudes (squared) of the entries of e^{-itL} measure the communicability (which we could call “quantum communicability”) between pairs of nodes, or states. Localization in e^{-itL} means low quantum communicability between far away pairs of nodes.

4.3.3 Log-Determinant Evaluation

In statistics, it is frequently necessary to evaluate the expression

$$\log(\det(A)) = \text{Tr}(\log(A)), \quad (101)$$

where A is symmetric positive definite. When A is not too large, one can directly compute the determinant on the left-hand side of (101) via a Cholesky factorization of A . If A is too large to be factored, various alternative methods have been proposed, the most popular of which are randomized methods for trace estimation such as the one proposed by Hutchinson [114]:

$$\text{Tr}(\log(A)) \approx \frac{1}{s} \sum_{j=1}^s \mathbf{v}_j^T \log(A) \mathbf{v}_j,$$

with $\mathbf{v}_1, \dots, \mathbf{v}_s$ suitably chosen “sampling” vectors. This method requires the rapid evaluation of the matrix-vector products $\log(A)\mathbf{v}_j$ for many different vectors \mathbf{v}_j .

A number of methods for approximating $\log(A)\mathbf{v}_j$ are studied in [6]. Some of the techniques in [6] rely on the off-diagonal decay property in $\log(A)$ when A is sparse. The authors of [6] make the observation that the decay in $\log(A)$ will be different in different bases; using the fact that for any nonsingular matrix W the identities

$$\det(f(WAW^{-1})) = \det(Wf(A)W^{-1}) = \det(f(A))$$

hold, it may be possible to find a basis W in which $f(WAW^{-1})$ is highly localized, so that performing the computations in this basis might result in significant speed-ups. In particular, the use of an orthonormal ($W^{-1} = W^T$) wavelet basis is shown to result in considerable off-diagonal compression, as long as the entries in A vary smoothly (as is often the case). We refer to [6] for details. We note that a similar approach, with $f(A) = A^{-1}$, has been used (e.g., in [50]) to construct sparse approximate inverses for use as preconditioners.

4.3.4 Quantum Information Theory

Another area of research where decay bounds for matrix functions have proven useful is the study of many-body systems in quantum information theory; see, e.g., [55, 56, 76, 176]. For instance, relationships between spectral gaps and rates of decay for functions of local Hamiltonian operators have been derived in [55] based on Bernstein’s Theorem, following [20].

As shown in [56], exponential decay bounds for matrix functions can be used to establish so-called *area laws* for the *entanglement entropy* of ground states associated with bosonic systems. In a nutshell, these area laws imply that the entanglement entropy associated with a region of a 3D bosonic lattice is proportional to the surface area, rather than to the volume, of the region. It is noteworthy that such area laws are analogous to those governing the Beckenstein–Hawking black hole entropy. We refer the interested reader to the comprehensive survey paper [76], where implications for computer simulations of quantum states are also discussed.

5 Conclusions and Future Work

The traditional dichotomy between *sparse* and *dense* matrix computations is too restrictive and needs to be revised to allow for additional modes of computation in which other, less-obvious forms of (approximate) sparsity are present, either in the problem data or in the solution, or both.

In recent years there has been strong interest and many important developments in research areas like hierarchical matrices and data-sparse algorithms (discussed by Ballani and Kressner and by Bini in this same volume) and compressed sensing;

a different direction, the exploitation of localization, or decay, has been the subject of this chapter. Localization has long played an important role (both conceptual and computational) in various areas of physics, but until recently it has received less attention from researchers in the field of numerical linear algebra. Here we have reviewed various notions of localization arising in different fields of mathematics and some of its applications in physics. We have attempted to provide a unified view of localization in numerical linear algebra using various types of decay bounds for the entries of matrix functions. Other useful tools include the use of decay algebras and C^* -algebras, and integral representations of matrix functions. We have further indicated how exploitation of localization is being used for developing fast approximate solution algorithms, in some cases having linear complexity in the size of the problem.

There are numerous opportunities for further research in this area. At several points in the chapter we have pointed out a few open problems and challenges, which can be summarized briefly as follows.

1. Concerning functions of matrices, including the important special cases of inverses and spectral projectors, we have discussed several conditions for localization. We have seen that these conditions are sufficient, but not necessary in general. Finding necessary conditions would deepen our understanding of localization considerably. Deriving some lower bounds on the entries of $f(A)$ would be useful in this regard.
2. Many of the decay bounds we have seen are rather pessimistic in practice. Similar to the convergence theory for Krylov subspace methods, it should be possible to obtain improved bounds by making use of more detailed spectral information on the matrix, at least in the Hermitian case.
3. As usual, the case of nonnormal matrices presents challenges and difficulties not present in the normal case. It would be useful to have a better understanding of decay properties in functions of highly nonnormal matrices, for example in oblique spectral projectors. This may have interesting applications in fields like non-Hermitian quantum mechanics [15, 16, 152, 193].
4. It is easy to see with examples that violating the bounded maximum degree assumption leads to failure of exponential decay in the limit $n \rightarrow \infty$; in practice, however, sufficiently rapid decay may persist for finite n to be useful in computation if the maximum degree increases slowly enough. This aspect seems to warrant further investigation, especially in view of applications in network analysis.
5. It would be interesting to develop general conditions under which bounded functions of unbounded, banded operators (or sequences of banded finite matrices without uniformly bounded spectra) exhibit decay behavior.
6. Outside of the broad area of linear scaling methods for electronic structure computations and in the solution of certain types of structured problems (e.g., [32, 185]), relatively little has been done so far to exploit advance knowledge of localization in designing efficient algorithms. It would be especially useful to develop approximation algorithms that can exploit localization in the solution

of large linear systems and in the eigenvectors (or invariant subspaces) of large matrices, when present. The ideas and techniques set forth in [135] for Hermitian matrices are a good starting point.

7. Last but not least, error control techniques in algorithms based on neglecting small matrix or vector entries deserve careful study.

We hope that this chapter will stimulate progress on these and other problems related to localization.

Acknowledgements I would like to express my sincere gratitude to several friends and collaborators without whose contributions these lecture notes would not have been written, namely, Paola Boito, Matt Challacombe, Nader Razouk, Valeria Simoncini, and the late Gene Golub. I am also grateful for financial support to the Fondazione CIME and to the US National Science Foundation (grants DMS-0810862, DMS-1115692 and DMS-1418889).

References

1. M. Abramowitz, I. Stegun, *Handbook of Mathematical Functions* (Dover, New York, NY, 1965)
2. S. Agmon, *Lectures on Exponential Decay of Solutions of Second-Order Elliptic Equations: Bounds on Eigenfunctions of N-Body Schrödinger Operators*. Mathematical Notes, vol. 29 (Princeton University Press, Princeton, NJ; University of Tokyo Press, Tokyo, 1982)
3. G. Alléon, M. Benzi, L. Giraud, Sparse approximate inverse preconditioning for dense linear systems arising in computational electromagnetics. *Numer. Algorithms* **16**, 1–15 (1997)
4. P.W. Anderson, Absence of diffusion in certain random lattices. *Phys. Rev.* **109**, 1492–1505 (1958)
5. M. Arioli, M. Benzi, A finite element method for quantum graphs. Math/CS Technical Report TR-2015-009, Emory University, Oct 2015
6. E. Aune, D.P. Simpson, J. Eidsvik, Parameter estimation in high dimensional Gaussian distributions. *Stat. Comput.* **24**, 247–263 (2014)
7. O. Axelsson, *Iterative Solution Methods* (Cambridge University Press, Cambridge, 1994)
8. O. Axelsson, B. Polman, On approximate factorization methods for block matrices suitable for vector and parallel processors. *Linear Algebra Appl.* **77**, 3–26 (1986)
9. R. Baer, M. Head-Gordon, Sparsity of the density matrix in Kohn–Sham density functional theory and an assessment of linear system-size scaling methods. *Phys. Rev. Lett.* **79**, 3962–3965 (1997)
10. R. Baer, M. Head-Gordon, Chebyshev expansion methods for electronic structure calculations on large molecular systems. *J. Chem. Phys.* **107**, 10003–10013 (1997)
11. H. Bağci, J.E. Pasciak, K.Y. Sirenko, A convergence analysis for a sweeping preconditioner for block tridiagonal systems of linear equations. *Numer. Linear Algebra Appl.* **22**, 371–392 (2015)
12. A.G. Baskakov, Wiener’s theorem and the asymptotic estimates of the elements of inverse matrices. *Funct. Anal. Appl.* **24**, 222–224 (1990)
13. A.G. Baskakov, Estimates for the entries of inverse matrices and the spectral analysis of linear operators. *Izv. Math.* **61**, 1113–1135 (1997)
14. R. Bellman, *Introduction to Matrix Analysis*, 2nd edn. (McGraw-Hill, New York, NY, 1970)
15. C.M. Bender, S. Boettcher, P.N. Meisinger, PT-symmetric quantum mechanics. *J. Math. Phys.* **40**, 2201–2229 (1999)
16. C.M. Bender, D.C. Brody, H.F. Jones, Must a Hamiltonian be Hermitian? *Am. J. Phys.* **71**, 1095–1102 (2003)

17. M. Benzi, Preconditioning techniques for large linear systems: a survey. *J. Comp. Phys.* **182**, 418–477 (2002)
18. M. Benzi, P. Boito, Quadrature rule-based bounds for functions of adjacency matrices. *Linear Algebra Appl.* **433**, 637–652 (2010)
19. M. Benzi, P. Boito, Decay properties for functions of matrices over C^* -algebras. *Linear Algebra Appl.* **456**, 174–198 (2014)
20. M. Benzi, G.H. Golub, Bounds for the entries of matrix functions with applications to preconditioning. *BIT Numer. Math.* **39**, 417–438 (1999)
21. M. Benzi, N. Razouk, Decay bounds and $O(n)$ algorithms for approximating functions of sparse matrices. *Electron. Trans. Numer. Anal.* **28**, 16–39 (2007)
22. M. Benzi, V. Simoncini, Decay bounds for functions of Hermitian matrices with banded or Kronecker structure. *SIAM J. Matrix Anal. Appl.* **36**, 1263–1282 (2015)
23. M. Benzi, M. Tuma, A sparse approximate inverse preconditioner for nonsymmetric linear systems. *SIAM J. Sci. Comput.* **19**, 968–994 (1998)
24. M. Benzi, M. Tuma, Orderings for factorized approximate inverse preconditioners. *SIAM J. Sci. Comput.* **21**, 1851–1868 (2000)
25. M. Benzi, C.D. Meyer, M. Tuma, A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM J. Sci. Comput.* **17**, 1135–1149 (1996)
26. M. Benzi, P. Boito, N. Razouk, Decay properties of spectral projectors with applications to electronic structure. *SIAM Rev.* **55**, 3–64 (2013)
27. M. Benzi, T. Evans, S. Hamilton, M. Lupo Pasini, S. Slattery, Analysis of Monte Carlo accelerated iterative methods for sparse linear systems. Math/CS Technical Report TR-2016-002, Emory University. *Numer. Linear Algebra Appl.* 2017, to appear
28. S.K. Berberian, G.H. Orland, On the closure of the numerical range of an operator. *Proc. Am. Math. Soc.* **18**, 499–503 (1967)
29. L. Bergamaschi, M. Vianello, Efficient computation of the exponential operator for large, sparse, symmetric matrices. *Numer. Linear Algebra Appl.* **7**, 27–45 (2000)
30. L. Bergamaschi, M. Caliari, M. Vianello, Efficient approximation of the exponential operator for discrete 2D advection-diffusion problems. *Numer. Linear Algebra Appl.* **10**, 271–289 (2003)
31. D.A. Bini, G. Latouche, B. Meini, *Numerical Methods for Structured Markov Chains* (Oxford University Press, Oxford, 2005)
32. D.A. Bini, S. Dendievel, G. Latouche, B. Meini, Computing the exponential of large block-triangular block-Toeplitz matrices encountered in fluid queues. *Linear Algebra Appl.* **502**, 387–419 (2016)
33. I.A. Blatov, Incomplete factorization methods for systems with sparse matrices. *Comput. Math. Math. Phys.* **33**, 727–741 (1993)
34. I.A. Blatov, On algebras and applications of operators with pseudosparse matrices. *Siber. Math. J.* **37**, 32–52 (1996)
35. I.A. Blatov, A.A. Terteryan, Estimates of the elements of the inverse matrices and pivotal condensation methods of incomplete block factorization. *Comput. Math. Math. Phys.* **32**, 1509–1522 (1992)
36. N. Bock, M. Challacombe, An optimized sparse approximate matrix multiply for matrices with decay. *SIAM J. Sci. Comput.* **35**, C72–C98 (2013)
37. N. Bock, M. Challacombe, L.V. Kalé, Solvers for $\mathcal{O}(N)$ electronic structure in the strong scaling limit. *SIAM J. Sci. Comput.* **38**, C1–C21 (2016)
38. L. Bonaventura, Local exponential methods: a domain decomposition approach to exponential time integration of PDEs. arXiv:1505.02248v1, May 2015
39. F. Bonchi, P. Esfandiar, D.F. Gleich, C. Greif, L.V.S. Lakshmanan, Fast matrix computations for pair-wise and column-wise commute times and Katz scores. *Internet Math.* **8**, 73–112 (2012)
40. A. Böttcher, S.M. Grudsky, *Spectral Properties of Banded Toeplitz Matrices* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005)

41. A. Böttcher, B. Silbermann, *Introduction to Large Truncated Toeplitz Matrices* (Springer, New York, NY, 1998)
42. D.R. Bowler, T. Miyazaki, $O(N)$ methods in electronic structure calculations. *Rep. Prog. Phys.* **75**, 036503 (2012)
43. S. Brooks, E. Lindenstrauss, Non-localization of eigenfunctions on large regular graphs. *Isr. J. Math.* **193**, 1–14 (2013)
44. C. Brouder, G. Panati, M. Calandra, C. Mourougane, N. Marzari, Exponential localization of Wannier functions in insulators. *Phys. Rev. Lett.* **98**, 046402 (2007)
45. S. Brucoliaglia, S. Micheletti, S. Perotto, Compressed solving: a numerical approximation technique for elliptic PDEs based on compressed sensing. *Comput. Math. Appl.* **70**, 1306–1335 (2015)
46. K. Bryan, T. Lee, Making do with less: an introduction to compressed sensing. *SIAM Rev.* **55**, 547–566 (2013)
47. C. Canuto, V. Simoncini, M. Verani, On the decay of the inverse of matrices that are sum of Kronecker products. *Linear Algebra Appl.* **452**, 21–39 (2014)
48. C. Canuto, V. Simoncini, M. Verani, Contraction and optimality properties of an adaptive Legendre–Galerkin method: the multi-dimensional case. *J. Sci. Comput.* **63**, 769–798 (2015)
49. M. Challacombe, A simplified density matrix minimization for linear scaling self-consistent field theory. *J. Chem. Phys.* **110**, 2332–2342 (1999)
50. T. Chan, W.-P. Tang, J. Wan, Wavelet sparse approximate inverse preconditioners. *BIT Numer. Math.* **37**, 644–660 (1997)
51. J. Chandrasekar, D.S. Bernstein, Correlation bounds for discrete-time systems with banded dynamics. *Syst. Control Lett.* **56**, 83–86 (2007)
52. E. Chow, A priori sparsity patterns for parallel sparse approximate inverse preconditioners. *SIAM J. Sci. Comput.* **21**, 1804–1822 (2000)
53. J.-M. Combes, L. Thomas, Asymptotic behaviour of eigenfunctions for multiparticle Schrödinger operators. *Commun. Math. Phys.* **34**, 251–270 (1973)
54. P. Concus, G.H. Golub, G. Meurant, Block preconditioning for the conjugate gradient method. *SIAM J. Sci. Stat. Comput.* **6**, 220–252 (1985)
55. M. Cramer, J. Eisert, Correlations, spectral gap and entanglement in harmonic quantum systems on generic lattices. *New J. Phys.* **8**, 71 (2006)
56. M. Cramer, J. Eisert, M.B. Plenio, J. Dreissig, Entanglement-area law for general Bosonic harmonic lattice systems. *Phys. Rev. A* **73**, 012309 (2006)
57. M. Crouzeix, Numerical range and functional calculus in Hilbert space. *J. Funct. Anal.* **244**, 668–690 (2007)
58. C.K. Chui, M. Hasson, Degree of uniform approximation on disjoint intervals. *Pac. J. Math.* **105**, 291–297 (1983)
59. J.J.M. Cuppen, A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numer. Math.* **36**, 177–195 (1981)
60. S. Dahlke, M. Fornasier, K. Gröchenig, Optimal adaptive computations in the Jaffard algebra and localized frames. *J. Approx. Theory* **162**, 153–185 (2010)
61. A. Damle, L. Lin, L. Ying, Compressed representations of Kohn–Sham orbitals via selected columns of the density matrix. *J. Chem. Theory Comput.* **11**, 1463–1469 (2015)
62. A. Damle, L. Lin, L. Ying, Accelerating selected columns of the density matrix computations via approximate column selection. arXiv:1604.06830v1, April 2016
63. P.J. Davis, *Circulant Matrices* (Wiley, New York, 1979)
64. T.A. Davis, Y. Hu, The University of Florida sparse matrix collection. *ACM Trans. Math. Softw.* **38**, 1–25 (2011)
65. Y. Dekel, J.R. Lee, N. Linial, Eigenvectors of random graphs: nodal domains. *Random Struct. Algorithm* **39**, 39–58 (2011)
66. N. Del Buono, L. Lopez, R. Peluso, Computation of the exponential of large sparse skew-symmetric matrices. *SIAM J. Sci. Comput.* **27**, 278–293 (2005)
67. S. Demko, Inverses of band matrices and local convergence of spline projections. *SIAM J. Numer. Anal.* **14**, 616–619 (1977)

68. S. Demko, W.F. Moss, P.W. Smith, Decay rates for inverses of band matrices. *Math. Comput.* **43**, 491–499 (1984)
69. J. des Cloizeaux, Energy bands and projection operators in a crystal: analytic and asymptotic properties. *Phys. Rev.* **135**, A685–A697 (1964)
70. I.S. Dhillon, B.S. Parlett, C. Vömel, The design and implementation of the MRRR algorithm. *ACM Trans. Math. Softw.* **32**, 533–560 (2006)
71. R. Diestel, *Graph Theory* (Springer, Berlin, 2000)
72. I.S. Duff, A.M. Erisman, J.K. Reid, *Direct Methods for Sparse Matrices* (Oxford University Press, Oxford, 1986)
73. I. Dumitriu, S. Pal, Sparse regular random graphs: spectral density and eigenvectors. *Ann. Prob.* **40**, 2197–2235 (2012)
74. W.E. J. Lu, The electronic structure of smoothly deformed crystals: Wannier functions and the Cauchy–Born rule. *Arch. Ration. Mech. Anal.* **199**, 407–433 (2011)
75. V. Eijkhout, B. Polman, Decay rates of inverses of banded M -matrices that are near to Toeplitz matrices. *Linear Algebra Appl.* **109**, 247–277 (1988)
76. J. Eisert, M. Cramer, M.B. Plenio, Colloquium: area laws for the entanglement entropy. *Rev. Modern Phys.* **82**, 277–306 (2010)
77. S.W. Ellacott, Computation of Faber series with application to numerical polynomial approximation in the complex plane. *Math. Comput.* **40**, 575–587 (1983)
78. E. Estrada, *The Structure of Complex Networks: Theory and Applications* (Oxford University Press, Oxford, 2012)
79. E. Estrada, N. Hatano, Communicability in complex networks. *Phys. Rev. E* **77**, 036111 (2008)
80. E. Estrada, D.J. Higham, Network properties revealed by matrix functions. *SIAM Rev.* **52**, 696–714 (2010)
81. E. Estrada, N. Hatano, M. Benzi, The physics of communicability in complex networks. *Phys. Rep.* **514**, 89–119 (2012)
82. I. Faria, Permanent roots and the star degree of a graph. *Linear Algebra Appl.* **64**, 255–265 (1985)
83. N.J. Ford, D.V. Savostyanov, N.L. Zamarashkin, On the decay of the elements of inverse triangular Toeplitz matrices. *SIAM J. Matrix Anal. Appl.* **35**, 1288–1302 (2014)
84. R. Freund, On polynomial approximations to $f_a(z) = (z - a)^{-1}$ with complex a and some applications to certain non-Hermitian matrices. *Approx. Theory Appl.* **5**, 15–31 (1989)
85. I.M. Gelfand, Normierte Ringe. *Mat. Sb.* **9**, 3–23 (1941)
86. I.M. Gelfand, M.A. Neumark, On the imbedding of normed rings in the ring of operators in Hilbert space. *Mat. Sb.* **12**, 197–213 (1943)
87. I.M. Gelfand, D.A. Raikov, G.E. Shilov, *Commutative Normed Rings* (Chelsea Publishing Co., Bronx/New York, 1964)
88. P.-L. Giscard, K. Lui, S.J. Thwaite, D. Jaksch, An exact formulation of the time-ordered exponential using path-sums. *J. Math. Phys.* **56**, 053503 (2015)
89. D.F. Gleich, PageRank beyond the Web. *SIAM Rev.* **57**, 321–363 (2015)
90. D.F. Gleich, K. Kloster, Sublinear column-wise actions of the matrix exponential on social networks. *Internet Math.* **11**, 352–384 (2015)
91. S. Goedecker, Linear scaling electronic structure methods. *Rev. Mod. Phys.* **71**, 1085–1123 (1999)
92. S. Goedecker, O.V. Ivanov, Frequency localization properties of the density matrix and its resulting hypersparsity in a wavelet representation. *Phys. Rev. B* **59**, 7270–7273 (1999)
93. K.-I. Goh, B. Khang, D. Kim, Spectra and eigenvectors of scale-free networks. *Phys. Rev. E* **64**, 051903 (2001)
94. G.H. Golub, G. Meurant, *Matrices, Moments and Quadrature with Applications* (Princeton University Press, Princeton, NJ, 2010)
95. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 4th edn. (Johns Hopkins University Press, Baltimore/London, 2013)

96. K. Gröchenig, A. Klotz, Noncommutative approximation: inverse-closed subalgebras and off-diagonal decay of matrices. *Constr. Approx.* **32**, 429–466 (2010)
97. K. Gröchenig, M. Leinert, Symmetry and inverse-closedness of matrix algebras and functional calculus for infinite matrices. *Trans. Am. Math. Soc.* **358**, 2695–2711 (2006)
98. K. Gröchenig, Z. Rzeszotnik, T. Strohmer, Convergence analysis of the finite section method and Banach algebras of matrices. *Integr. Equ. Oper. Theory* **67**, 183–202 (2010)
99. M. Grote, T. Huckle, Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.* **18**, 838–853 (1997)
100. J. Gutiérrez-Gutiérrez, P.M. Crespo, A. Böttcher, Functions of the banded Hermitian block Toeplitz matrices in signal processing. *Linear Algebra Appl.* **422**, 788–807 (2007)
101. S. Güttel, L. Knizhnerman, A black-box rational Arnoldi variant for Cauchy–Stieltjes matrix functions. *BIT Numer. Math.* **53**, 595–616 (2013)
102. A. Haber, M. Verhaegen, Subspace identification of large-scale interconnected systems. *IEEE Trans. Automat. Control* **59**, 2754–2759 (2014)
103. A. Haber, M. Verhaegen, Sparse solution of the Lyapunov equation for large-scale interconnected systems. *Automatica* **73**, 256–268 (2016)
104. M. Hasson, The degree of approximation by polynomials on some disjoint intervals in the complex plane. *J. Approx. Theory* **144**, 119–132 (2007)
105. L. He, D. Vanderbilt, Exponential decay properties of Wannier functions and related quantities. *Phys. Rev. Lett.* **86**, 5341–5344 (2001)
106. V.E. Henson, G. Sanders, Locally supported eigenvectors of matrices associated with connected and unweighted power-law graphs. *Electron. Trans. Numer. Anal.* **39**, 353–378 (2012)
107. N.J. Higham, *Matrix Functions. Theory and Computation* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2008)
108. N.J. Higham, D.S. Mackey, N. Mackey, F. Tisseur, Functions preserving matrix groups and iterations for the matrix square root. *SIAM J. Matrix Anal. Appl.* **26**, 1178–1192 (2005)
109. M. Hochbruck, Ch. Lubich, On Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.* **34**, 1911–1925 (1997)
110. P. Hohenberg, W. Kohn, Inhomogeneous electron gas. *Phys. Rev.* **136**, B864–871 (1964)
111. R.A. Horn, C.R. Johnson, *Topics in Matrix Analysis* (Cambridge University Press, Cambridge, 1994)
112. R.A. Horn, C.R. Johnson, *Matrix Analysis*, 2nd edn. (Cambridge University Press, Cambridge, 2013)
113. T. Huckle, Approximate sparsity patterns for the inverse of a matrix and preconditioning. *Appl. Numer. Math.* **30**, 291–303 (1999)
114. M. Hutchinson, A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Commun. Stat. Simul. Comput.* **18**, 1059–1076 (1989)
115. A. Iserles, How large is the exponential of a banded matrix? *N. Z. J. Math.* **29**, 177–192 (2000)
116. S. Ismail-Beigi, T.A. Arias, Locality of the density matrix in metals, semiconductors, and insulators. *Phys. Rev. Lett.* **82**, 2127–2130 (1999)
117. S. Jaffard, Propriétés des matrices “bien localisées” près de leur diagonale et quelques applications. *Ann. Inst. Henri Poincaré* **7**, 461–476 (1990)
118. J. Janas, S. Naboko, G. Stolz, Decay bounds on eigenfunctions and the singular spectrum of unbounded Jacobi matrices. *Intern. Math. Res. Notices* **4**, 736–764 (2009)
119. R. Kadison, Diagonalizing matrices. *Am. J. Math.* **106**, 1451–1468 (1984)
120. R. Kadison, J. Ringrose, *Fundamentals of the Theory of Operator Algebras*. Elementary Theory, vol. I (Academic Press, Orlando, FL, 1983)
121. W. Kohn, Analytic properties of Bloch waves and Wannier functions. *Phys. Rev.* **115**, 809–821 (1959)
122. W. Kohn, Density functional and density matrix method scaling linearly with the number of atoms. *Phys. Rev. Lett.* **76**, 3168–3171 (1996)
123. W. Kohn, Nobel lecture: electronic structure of matter—wave functions and density functionals. *Rev. Mod. Phys.* **71**, 1253–1266 (1999)

124. W. Kohn, L.J. Sham, Self-consistent equations including exchange and correlation effects. *Phys. Rev. Lett.* **140**, A1133–1138 (1965)
125. L.Y. Kolotilina, A.Y. Yeremin, Factorized sparse approximate inverse preconditioning I. Theory. *SIAM J. Matrix Anal. Appl.* **14**, 45–58 (1993)
126. A. Koskela, E. Jarlebring, The infinite Arnoldi exponential integrator for linear inhomogeneous ODEs. arXiv:1502.01613v2, Feb 2015
127. I. Kryshnal, T. Strohmer, T. Wertz, Localization of matrix factorizations. *Found. Comput. Math.* **15**, 931–951 (2015)
128. R. Lai, J. Lu, Localized density matrix minimization and linear-scaling algorithms. *J. Comput. Phys.* **315**, 194–210 (2016)
129. C.S. Lam, Decomposition of time-ordered products and path-ordered exponentials. *J. Math. Phys.* **39**, 5543–5558 (1998)
130. A.N. Langville, C.D. Meyer *Google's PageRank and Beyond: The Science of Search Engine Rankings* (Princeton University Press, Princeton, NJ, 2006)
131. A.J. Laub, *Matrix Analysis for Scientists and Engineers* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005)
132. C. Le Bris, Computational chemistry from the perspective of numerical analysis. *Acta Numer.* **14**, 363–444 (2005)
133. X.-P. Li, R.W. Nunes, D. Vanderbilt, Density-matrix electronic structure method with linear system-size scaling. *Phys. Rev. B* **47**, 10891–10894 (1993)
134. W. Liang, C. Saravanan, Y. Shao, R. Baer, A. T. Bell, M. Head-Gordon, Improved Fermi operator expansion methods for fast electronic structure calculations. *J. Chem. Phys.* **119**, 4117–4124 (2003)
135. L. Lin, Localized spectrum slicing. *Math. Comput.* (2016, to appear). DOI:10.1090/mcom/3166
136. L. Lin, J. Lu, Sharp decay estimates of discretized Green's functions for Schrödinger type operators. *Sci. China Math.* **59**, 1561–1578 (2016)
137. F.-R. Lin, M.K. Ng, W.-K. Ching, Factorized banded inverse preconditioners for matrices with Toeplitz structure. *SIAM J. Sci. Comput.* **26**, 1852–1870 (2005)
138. L. Lin, J. Lu, L. Ying, R. Car, E. Weinan, Multipole representation of the Fermi operator with application to the electronic structure analysis of metallic systems. *Phys. Rev. B* **79**, 115133 (2009)
139. M. Lindner, *Infinite Matrices and Their Finite Sections* (Birkhäuser, Basel, 2006)
140. X. Liu, G. Strang, S. Ott, Localized eigenvectors from widely spaced matrix modifications. *SIAM J. Discrete Math.* **16**, 479–498 (2003)
141. L. Lopez, A. Pugliese, Decay behaviour of functions of skew-symmetric matrices, in *Proceedings of HERCMA 2005, 7th Hellenic-European Conference on Computer Mathematics and Applications*, 22–24 Sept 2005, Athens, ed. By E.A. Lipitakis, Electronic Editions (LEA, Athens, 2005)
142. T. Malas, L. Gürel, Schur complement preconditioners for surface integral-equation formulations of dielectric problems solved with the multilevel multipole algorithm. *SIAM J. Sci. Comput.* **33**, 2440–2467 (2011)
143. A.I. Markushevich, *Theory of Functions of a Complex Variable*, vol. III (Prentice-Hall, Englewood Cliffs, NJ, 1967)
144. O.A. Marques, B.N. Parlett, C. Vömel, Computation of eigenpair subsets with the MRRII algorithm. *Numer. Linear Algebra Appl.* **13**, 643–653 (2006)
145. R.M. Martin, *Electronic Structure. Basic Theory and Practical Methods* (Cambridge University Press, Cambridge, 2004)
146. P.E. Maslen, C. Ochsenfeld, C.A. White, M.S. Lee, M. Head-Gordon, Locality and sparsity of ab initio one-particle density matrices and localized orbitals. *J. Phys. Chem. A* **102**, 2215–2222 (1998)
147. N. Mastronardi, M.K. Ng, E.E. Tyrtyshnikov, Decay in functions of multi-band matrices. *SIAM J. Matrix Anal. Appl.* **31**, 2721–2737 (2010)

148. G. Meinardus, *Approximation of Functions: Theory and Numerical Methods*. Springer Tracts in Natural Philosophy, vol. 13 (Springer, New York, 1967)
149. P.N. McGraw, M. Menzinger, Laplacian spectra as a diagnostic tool for network structure and dynamics. *Phys. Rev. E* **77**, 031102 (2008)
150. N. Merkle, *Completely monotone functions—a digest*. arXiv:1211.0900v1, Nov 2012
151. G. Meurant, A review of the inverse of symmetric tridiagonal and block tridiagonal matrices. *SIAM J. Matrix Anal. Appl.* **13**, 707–728 (1992)
152. N. Moiseyev, *Non-Hermitian Quantum Mechanics* (Cambridge University Press, Cambridge, 2011)
153. L. Molinari, Identities and exponential bounds for transfer matrices. *J. Phys. A: Math. Theor.* **46**, 254004 (2013)
154. R. Nabben, Decay rates of the inverse of nonsymmetric tridiagonal and band matrices. *SIAM J. Matrix Anal. Appl.* **20**, 820–837 (1999)
155. Y. Nakatsukasa, Eigenvalue perturbation bounds for Hermitian block tridiagonal matrices. *Appl. Numer. Math.* **62**, 67–78 (2012)
156. Y. Nakatsukasa, N. Saito, E. Woei, Mysteries around the graph Laplacian eigenvalue 4. *Linear Algebra Appl.* **438**, 3231–3246 (2013)
157. H. Nassar, K. Kloster, D.F. Gleich, Strong localization in personalized PageRank vectors, in *Algorithms and Models for the Web Graph*, ed. by D.F. Gleich et al. Lecture Notes in Computer Science, vol. 9479 (Springer, New York, 2015), pp. 190–202
158. G. Nenciu, Existence of the exponentially localised Wannier functions. *Commun. Math. Phys.* **91**, 81–85 (1983)
159. A.M.N. Niklasson, Density matrix methods in linear scaling electronic structure theory, in *Linear-Scaling Techniques in Computational Chemistry and Physics*, ed. by R. Zaleśny et al. (Springer, New York, 2011), pp. 439–473
160. J. Pan, R. Ke, M.K. Ng, H.-W. Sun, Preconditioning techniques for diagonal-times-Toeplitz matrices in fractional diffusion equations. *SIAM J. Sci. Comput.* **36**, A2698–A2719 (2014)
161. B.N. Parlett, Invariant subspaces for tightly clustered eigenvalues of tridiagonals. *BIT Numer. Math.* **36**, 542–562 (1996)
162. B.N. Parlett, A result complementary to Geršgorin's circle theorem. *Linear Algebra Appl.* **432**, 20–27 (2009)
163. B.N. Parlett, I.S. Dhillon, Relatively robust representations of symmetric tridiagonals. *Linear Algebra Appl.* **309**, 121–151 (2000)
164. M.S. Paterson, L.J. Stockmeyer, On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.* **2**, 60–66 (1973)
165. E. Prodan, Nearsightedness of electronic matter in one dimension. *Phys. Rev. B* **73**, 085108 (2006)
166. E. Prodan, W. Kohn, Nearsightedness of electronic matter. *Proc. Nat. Acad. Sci.*, **102**, 11635–11638 (2005)
167. E. Prodan, S.R. Garcia, M. Putinar, Norm estimates of complex symmetric operators applied to quantum systems. *J. Phys. A: Math. Gen.* **39**, 389–400 (2006)
168. N. Razouk, Localization phenomena in matrix functions: theory and algorithms, Ph.D. Thesis, Emory University, 2008
169. L. Reichel, G. Rodriguez, T. Tang, New block quadrature rules for the approximation of matrix functions. *Linear Algebra Appl.* **502**, 299–326 (2016)
170. S. Roch, *Finite Sections of Band-Dominated Operators*, vol. 191, no. 895 (Memoirs of the American Mathematical Society, Providence, RI, 2008)
171. G. Rodriguez, S. Seatzu, D. Theis, An algorithm for solving Toeplitz systems by embedding in infinite systems. *Oper. Theory Adv. Appl.* **160**, 383–401 (2005)
172. E.H. Rubensson, E. Rudberg, P. Salek, Methods for Hartree–Fock and density functional theory electronic structure calculations with linearly scaling processor time and memory usage, in *Linear-Scaling Techniques in Computational Chemistry and Physics*, ed. by R. Zaleśny et al. (Springer, New York, NY, 2011), pp. 269–300
173. W. Rudin, *Functional Analysis* (McGraw-Hill, New York, NY, 1973)

174. Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd edn. (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003)
175. Y. Saad, J.R. Chelikowsky, S.M. Shontz, Numerical methods for electronic structure calculations of materials. *SIAM Rev.* **52**, 3–54 (2010)
176. N. Schuch, J.I. Cirac, M.M. Wolf, Quantum states on harmonic lattices. *Commun. Math. Phys.* **267**, 65–92 (2006)
177. M. Shao, On the finite section method for computing exponentials of doubly-infinite skew-Hermitian matrices. *Linear Algebra Appl.* **451**, 65–96 (2014)
178. D.I. Shuman, B. Ricaud, P. Vandergheynst, Vertex-frequency analysis on graphs. *Appl. Comput. Harmon. Anal.* **40**, 260–291 (2016)
179. C. Siefert, E. de Sturler, Probing methods for saddle-point problems. *Electron. Trans. Numer. Anal.* **22**, 163–183 (2006)
180. B. Simon, Semiclassical analysis of low lying eigenvalues. I. Nondegenerate minima: asymptotic expansions. *Ann. Inst. H. Poincaré Sect. A* **38**, 295–308 (1983)
181. V. Simoncini, Computational methods for linear matrix equations. *SIAM Rev.* **58**, 377–441 (2016)
182. D.T. Smith, Exponential decay of resolvents and discrete eigenfunctions of banded infinite matrices. *J. Approx. Theory* **66**, 83–97 (1991)
183. G. Stoltz, An introduction to the mathematics of Anderson localization, in *Entropy and the Quantum II*, ed. by R. Sims, D. Ueltschi. Contemporary Mathematics, vol. 552 (American Mathematical Society, Providence, RI, 2011), pp. 71–108
184. G. Strang, S. MacNamara, Functions of difference matrices are Toeplitz plus Hankel. *SIAM Rev.* **56**, 525–546 (2014)
185. T. Strohmer, Four short stories about Toeplitz matrix calculations. *Linear Algebra Appl.* **343/344**, 321–344 (2002)
186. Q. Sun, Wiener's lemma for infinite matrices with polynomial off-diagonal decay. *C. R. Acad. Sci. Paris Ser. I* **340**, 567–570 (2005)
187. P. Suryanarayana, On spectral quadrature for linear-scaling density functional theory. *Chem. Phys. Lett.* **584**, 182–187 (2013)
188. H. Tal-Ezer, Polynomial approximation of functions of matrices and applications. *J. Sci. Comput.* **4**, 25–60 (1989)
189. L.V. Tran, V.H. Vu, K. Wang, Sparse random graphs: eigenvalues and eigenvectors. *Random Struct. Algorithm.* **42**, 110–134 (2013)
190. L.N. Trefethen, Numerical computation of the Schwarz–Christoffel transformation. *SIAM J. Sci. Stat. Comput.* **1**, 82–102 (1980)
191. L.N. Trefethen, D. Bau, *Numerical Linear Algebra* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997)
192. L.N. Trefethen, M. Embree, *Spectra and Pseudospectra. The Behavior of Nonnormal Matrices and Operators* (Princeton University Press, Princeton, NJ, 2005)
193. L.N. Trefethen, M. Contedini, M. Embree, Spectra, pseudospectra, and localization for random bidiagonal matrices. *Commun. Pure Appl. Math.* **54**, 595–623 (2001)
194. C.V.M. van der Mee, G. Rodriguez, S. Seatzu, *LDU* factorization results for bi-infinite and semi-infinite scalar and block Toeplitz matrices. *Calcolo* **33**, 307–335 (1998)
195. C.V.M. van der Mee, G. Rodriguez, S. Seatzu, Block Cholesky factorization of infinite matrices and orthonormalization of vectors of functions, in *Advances in Computational Mathematics (Guangzhou, 1997)*. Lecture Notes in Pure and Applied Mathematics (Dekker, New York, 1999), pp. 423–455
196. R.S. Varga, Nonnegatively posed problems and completely monotonic functions. *Linear Algebra Appl.* **1**, 329–347 (1968)
197. P.S. Vassilevski, On some ways of approximating inverses of band matrices in connection with deriving preconditioners based on incomplete block factorizations. *Computing* **43**, 277–296 (1990)
198. C. Vömel, B. N. Parlett, Detecting localization in an invariant subspace. *SIAM J. Sci. Comput.* **33**, 3447–3467 (2011)

199. H. Wang, Q. Ye, Error bounds for the Krylov subspace methods for computations of matrix exponentials. Tech. Rep., Department of Mathematics, University of Kentucky, Lexington, KY, 2016
200. H.F. Weinberger, *A First Course in Partial Differential Equations* (Wiley, New York, 1965)
201. D.V. Widder, *The Laplace Transform* (Princeton University Press, Princeton, 1946)
202. W. Yang, Direct calculation of electron density in density-functional theory. *Phys. Rev. Lett.* **66**, 1438–1441 (1991)
203. Q. Ye, Error bounds for the Lanczos method for approximating matrix exponentials. *SIAM J. Numer. Anal.* **51**, 68–87 (2013)

Groups and Symmetries in Numerical Linear Algebra

Hans Z. Munthe-Kaas

Abstract Groups are fundamental objects of mathematics, describing symmetries of objects and also describing sets of motions moving points in a domain, such as translations in the plane and rotations of a sphere. The topic of these lecture notes is applications of group theory in computational mathematics. We will first cover fundamental properties of groups and continue with an extensive discussion of commutative (abelian) groups and their relationship to computational Fourier analysis. Various numerical algorithms will be discussed in the setting of group theory. Finally we will, more briefly, discuss generalisation of Fourier analysis to non-commutative groups and discuss problems in linear algebra with non-commutative symmetries. The representation theory of non-commutative finite groups is used as a tool to efficiently solve linear algebra problems with symmetries, exemplified by the computation of matrix exponentials.

1 Introduction

‘Symmetry’ is a vaguely defined concept deeply rooted in nature, physics, biology, art, culture and mathematics. In everyday language it refers to a harmonious proportion and balance. In mathematics, the symmetries of an object are more precisely defined as a set of transformations leaving the object invariant. Examples in art are tessellations and mosaics invariant under translations and reflections. In mechanics, symmetry can refer to invariance of a Lagrangian function under transformations such as spatial rotations and translation in time, and the famous theorem of Emmy Noether relates such symmetries to conservation laws. Sophus Lie (1842–1899) revolutionised the theory of differential equations by considering the symmetries sending solution curves to other solutions. A huge part of signal processing and Fourier analysis is based on invariance of linear operators under time or space translations. Classical Fourier analysis extends to non-commutative harmonic analysis and group representation theory when the symmetry transformations do not commute (when $ab \neq ba$).

H.Z. Munthe-Kaas (✉)

Department of Mathematics, University of Bergen, Postbox 7803, N-5020 Bergen, Norway
e-mail: hans.munthe-kaas@uib.no

When designing an algorithm for solving some computational problem, it is usually a good idea to look for the symmetries of the problem. An algorithm which preserves symmetries often results in more accurate or stable numerical computations, and potentially also leads to huge savings in terms of time and space consumption. For these reasons, knowledge of the mathematics of symmetries (group theory) should be very important for students of computational science.

In these lectures we will focus our attention to applications of group theory in numerical linear algebra, Fourier analysis and signal processing. We will in particular focus on a unified treatment of classical Fourier analysis, based on translational invariance in space or time (commutative symmetries), and continue with a treatment of non-commutative groups of reflection symmetries such as the symmetries generated by the mirrors in a kaleidoscope. This is applied to fast solution of boundary value problems, computation of matrix exponentials and applications to sampling theory and numerical computations on regular lattices.

It is our goal that most of the material in these lectures should be accessible to advanced undergraduate students in applied and computational mathematics, requiring only basic knowledge of linear algebra and calculus, and not any prior knowledge of group theory nor abstract algebra.

1.1 Motivation for the Main Topics of the Lectures

Consider the objects below, a thirteenth century mosaic from Alhambra, a tessellation by Maurice Escher, a three-foil knot and a special matrix:



$$A = \begin{pmatrix} a_0 & a_2 & a_1 & a_3 & a_4 & a_5 \\ a_1 & a_0 & a_2 & a_5 & a_3 & a_4 \\ a_2 & a_1 & a_0 & a_4 & a_5 & a_3 \\ a_3 & a_5 & a_4 & a_0 & a_1 & a_2 \\ a_4 & a_3 & a_5 & a_2 & a_0 & a_1 \\ a_5 & a_4 & a_3 & a_1 & a_2 & a_0 \end{pmatrix}$$

Quiz

1. Do any of these objects have the same symmetries?
2. How can we compute the eigenvalues and eigenvectors of A ?

In order to try to answer (1), we must define what we mean by ‘invariance under a set of transformations’. The two tessellations (Alhambra and Escher) can be seen to be invariant under certain Euclidean (rigid) motions of the plane. The exact group of symmetries depends on whether or not one considers the colouring, or just the shapes. Without regarding the colouring, they are both invariant under 120° rotations in certain points and translations in two different directions. In a certain sense, which has not yet been made clear, it seems as the two tessellations have the same symmetries.

The trefoil knot, understood as a curve in \mathbb{R}^3 is invariant under transformation α being a 120° rotation around the centre as well as transformation β being a 180° rotation around the vertical line through the plane of the knot. Any product of α, β and their inverses are also symmetries, so that the total group of symmetries becomes $\{1, \alpha, \alpha^2, \beta, \alpha\beta, \alpha^2\beta\}$, where 1 denotes the identity transformation (do nothing). We can verify that α and β satisfy the relations

$$\alpha^3 = \beta^2 = \alpha\beta\alpha\beta = 1. \quad (1)$$

The symmetries of A are less evident. We can verify that A commutes with some particular (permutation) matrices; we have that $P_i A = AP_i$, or equivalently $A = P_i A P_i^{-1}$ for both

$$P_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad P_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and hence also for any P_i which is given as a product of these and their inverses, $P_i \in \{I, P_1, P_1^2, P_2, P_1 P_2, P_1^2 P_2\}$. It can be shown that P_1 and P_2 satisfy exactly the same relations (1) as α and β . However, to claim that the three-foil and A have the same symmetries, we need to abstract the notion of a symmetry group and the action of a group on a set so that we can discuss the properties of the abstract group independently of the concrete transformations the group performs on a given object. We start Sect. 3 with the modern definition of a group and group actions.

Now, back to Question 2 in the Quiz. Once we understand that A commutes with a set of matrices, we are closer to finding the eigenvectors. From elementary linear algebra we know that matrices with a common complete set of eigenvectors do commute, and conversely, under quite mild conditions (e.g. distinct eigenvalues), commuting matrices share a complete set of common eigenvectors. However, P_1 and

P_2 do not have distinct eigenvalues and furthermore they do not commute among themselves, since $P_1P_2 = P_2P_1^{-1}$, so we cannot possibly find a complete set of common eigenvectors. However, *groups representation theory* provides something almost as good; a complete list of *irreducible representations*, which yields a particular basis change such that A becomes block diagonalised. Applications of this theory is the topic of Sect. 4. A very important and special case of structured matrices appears in classical Fourier analysis, where A commutes with a set of matrices P_i such that also $P_iP_j = P_jP_i$. If the set of such matrices is sufficiently large, we find a complete set of common eigenvectors for all these P_i and these also form a complete set of eigenvectors for A . In the case of finite dimensional A , the mathematical analysis becomes particularly simple, the common eigenvectors are exponential functions and the change of basis is given by the Discrete Fourier Transform. Also Fourier series on the continuous circle and the Fourier transform for functions on the real line can be described within a common group theoretical framework. A detailed study of these cases, with applications, is the topic of Sect. 3.

2 Prelude: Introduction to Group Theory

Before going into a detailed study of abelian (= commutative) groups, we will for future reference introduce some general concepts in group theory. A detailed understanding of this chapter is not needed for the applications in Fourier analysis and sampling theory, where these concepts become somewhat simpler. I suggest that this chapter is read lightly in first pass, and studied more carefully whenever needed later.

2.1 Groups and Actions

Definition 1 (Group) A group is a set G with a binary operation $\cdot: G \times G \rightarrow G$, called the group product, such that

1. The product is associative, $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ for all $x, y, z \in G$.
2. There exists an identity element $\mathbf{1} \in G$ such that $x \cdot \mathbf{1} = \mathbf{1} \cdot x = x$ for all $x \in G$.
3. Every element $x \in G$ has an inverse $x^{-1} \in G$ such that $x \cdot x^{-1} = \mathbf{1}$.

Sometimes we write the group product without the dot, as xy instead of $x \cdot y$. The special groups where $x \cdot y = y \cdot x$ for all $x, y \in G$ are called *commutative* or *abelian groups*. In the case of abelian groups we will often (but not always) write $+$ instead of \cdot , $-x$ instead of x^{-1} and $\mathbf{0}$ instead of $\mathbf{1}$.

Example 1 We list some common groups that we will encounter later.

- **Zero group $\{\mathbf{0}\}$.** This is the trivial additive abelian group consisting of just the identity element. Sometimes we write $\mathbf{0}$ instead of $\{\mathbf{0}\}$.

- **Additive group of reals** $(\mathbb{R}, +)$.
- **Additive circle group** $(T, +)$: This is the real numbers $[0, 1)$ under addition modulo 1. It is also defined as $T = \mathbb{R}/\mathbb{Z}$, a quotient group (see below). The name T refers to this being the 1-dimensional torus.
- **Additive group of integers** $(\mathbb{Z}, +)$.
- **Additive cyclic group** $(\mathbb{Z}_n, + \bmod n)$: This consists of the integers $\{0, 1, \dots, n-1\}$ with group operation being addition modulo n , and is also given as the quotient $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$.
- **Identity group {1}**: This is the trivial multiplicative group consisting of just the identity element. The two trivial groups $\{0\}$ and $\{1\}$ are *isomorphic* (abstractly the same group). Isomorphisms are discussed below.
- **Multiplicative cyclic group** C_n : This consists of the complex n th roots of 1 under multiplication, $\{e^{2\pi ij/n}\}_{j=0}^{n-1}$, and it is isomorphic to \mathbb{Z}_n .
- **Multiplicative circle group** \mathbb{T} : The multiplicative group of complex numbers with modulus 1, $\mathbb{T} = \{e^{2\pi i\theta}\}$ for $\theta \in [0, 1)$, isomorphic to T .
- **Dihedral group** D_n : The symmetries of an regular n -gon form a group called the *Dihedral group*, D_n . In particular D_3 are the six symmetries of an equilateral triangle (three rotations and three reflected rotations). Abstractly, D_n is generated by two elements α and β satisfying the relations

$$\alpha^n = \beta^2 = \alpha\beta\alpha\beta = 1. \quad (2)$$

The last relation is equivalent to $\alpha\beta = \beta\alpha^{-1}$, thus this is our first example of a non-commutative group.

- **General linear group** $GL(V)$: For a vector space V (e.g. $V = \mathbb{R}^n$), this group consists of all invertible linear operators on V (e.g. all invertible real $n \times n$ matrices), and the group product is composition of linear operators (matrix product).
- **Orthogonal group** $O(V)$: The set of all orthogonal matrices ($A^T = A^{-1}$) in $GL(V)$.
- **The symmetric group** S_n : This is the group of all permutations of n objects, thus S_n has $n!$ elements.
- **The alternating group** A_n : The subset of all *even* permutations of n objects, with $n!/2$ elements.
- **Lie groups**: There are two main classes of groups, discrete groups and Lie groups. Lie groups are groups which also has a differentiable structure, so that one can define continuous and smooth families of transformations. Among the examples above, the Lie groups are $(\mathbb{R}, +)$, $(T, +)$, \mathbb{T} , $GL(V)$ and $O(V)$. The remaining groups are discrete.

We want to connect an abstract group to a concrete group of transformations of some ‘object’. This is done by the concept of a *group action*.

Definition 2 (Group Action) A group G acts¹ on a set X if there is a function ('group action') $\cdot : G \times X \rightarrow X$ satisfying

$$\begin{aligned} 1 \cdot x &= x \quad \text{for all } x \in X \\ g \cdot (h \cdot x) &= (g \cdot h) \cdot x \quad \text{for all } g, h \in G, x \in X. \end{aligned}$$

Example 2 We define an action $\cdot : D_n \times \mathbb{C} \rightarrow \mathbb{C}$ on the generators of D_n as $\alpha \cdot z = e^{2\pi i/n}z$ (rotation counterclockwise through the angle $2\pi/n$) and $\beta \cdot z = \bar{z}$ (complex conjugation). These two symmetry operations are compatible with (2), we have $\alpha \cdot (\alpha \cdots (\alpha \cdot z)) = z$ (n times application of α), $\beta \cdot (\beta \cdot z) = z$ and $\alpha \cdot (\beta \cdot (\alpha \cdot (\beta \cdot z))) = z$. Therefore, we can extend this to a group action of D_n on \mathbb{C} . Consider the regular n -gon in \mathbb{C} , with vertices in the n th roots of unity $C_n \subset \mathbb{C}$. It is straightforward to check that the set of vertices of the n -gon is invariant under this action.

Example 3 The dihedral group D_3 acts on the set of all 6×6 matrices as $\alpha \cdot X = P_1XP_1^{-1}$ and $\beta \cdot X = P_2XP_2^{-1}$, where P_1 and P_2 are given above.

Example 4 Any group G can act on itself in various ways. We can let G act on itself by left multiplication $L_g g' := gg'$ or by right multiplication $R_g g' = g'g^{-1}$ or by conjugation $\text{Conj}_g g' := gg'g^{-1}$. Check that all these are well defined actions.

Definition 3 (Types of Actions) An action $\cdot : G \times X \rightarrow X$ is:

- *transitive* if for any pair $x, y \in X$ there exists a $g \in G$ such that $g \cdot x = y$,
- *free* if the identity $\mathbf{1} \in G$ is the only group element which has a fixed point on X , i.e. for $g \in G$ there exists an $x \in X$ such that $g \cdot x = x$ only if $g = \mathbf{1}$,
- *regular* if it is both free and transitive,
- *effective* if whenever $g, h \in G$ and $g \neq h$ there exist an $x \in X$ such that $g \cdot x \neq h \cdot x$.

Exercise 1

1. Show that *free* \Rightarrow *effective*.
2. Is the action of D_n on C_n defined in Example 2 regular?
3. Show that if an action is regular, then there is a 1–1 correspondence between elements of G and X . Find a subset of $2n$ points in \mathbb{C} on which the action of D_n defined in Example 2 is regular.

2.2 Subgroups and Quotients

It is important to understand some basic ways of obtaining groups from other groups, by decompositions (subgroups and quotients) and compositions (direct- and semidirect products).

¹This definition is more precisely called a *left action*.

Definition 4 (Subgroup) A non-empty subset $H \subset G$ which is closed under the group product and inversion is called a subgroup, denoted $H < G$.

A subgroup $H < G$ decomposes G into subsets called *cosets*, these can be defined from *left* or from *right*:

$$\begin{aligned} gH &:= \{gh : g \in G, h \in H\} \\ Hg &:= \{hg : g \in G, h \in H\}. \end{aligned}$$

Note that for $g, g' \in G$ we have either $gH = g'H$ or $gH \cap g'H = \emptyset$, so the collection of all left (or all right) cosets form a disjoint partition of G .

Example 5 The dihedral group $D_3 = \{\mathbf{1}, \alpha, \alpha^2, \beta, \beta\alpha, \beta\alpha^2\}$ has four subgroups. The *trivial* subgroup consists of just the identity $\{\mathbf{1}\} < D_3$, and the *improper* subgroup is the whole group $D_3 < D_3$. The two proper and non-trivial subgroups are $H = \{\mathbf{1}, \alpha, \alpha^2\}$ and $\tilde{H} = \{\mathbf{1}, \beta\}$. The left cosets of H are H and $\beta H = \{\beta, \beta\alpha, \beta\alpha^2\}$, and these form a disjoint partition $D_3 = H \cup \beta H$. The right cosets are H and $H\beta = \{\beta, \alpha\beta, \alpha^2\beta\} = \{\beta, \beta\alpha^2, \beta\alpha\} = \beta H$. The three left cosets of \tilde{H} are \tilde{H} , $\alpha\tilde{H} = \{\alpha, \alpha\beta\} = \{\alpha, \beta\alpha^2\}$ and $\alpha^2\tilde{H} = \{\alpha^2, \beta\alpha\}$. The three right cosets are \tilde{H} , $\tilde{H}\alpha = \{\alpha, \beta\alpha\}$ and $\tilde{H}\alpha^2 = \{\alpha^2, \beta\alpha^2\}$. Note that all left cosets of H are also right cosets, $gH = Hg$ for all $g \in G$. This is *not* the case for \tilde{H} .

Definition 5 (Normal Subgroup) A subgroup $H < G$ is called normal if $gH = Hg$ for every $g \in G$. We write a normal subgroup as $H \triangleleft G$.

The collection of cosets of a subgroup $H < G$ can be turned into a group if and only if H is normal.

Definition 6 (Quotient Group) For a normal subgroup $H \triangleleft G$ we define the quotient group G/H as a group where the elements of G/H are the cosets gH and the product of two cosets are defined as

$$gH \cdot g'H = gg'H,$$

where gg' is the product of g and g' in G .

Example 6 Continuing Example 5, we obtain the quotient group D_3/H with two elements H and βH and the multiplication rule $H \cdot H = H$, $\beta H \cdot H = H \cdot \beta H = \beta H$ and $\beta H \cdot \beta H = H$. The group D_3/H can be identified with the group $C_2 = \{1, -1\} \subset \mathbb{R}$ with multiplication as group product, meaning that if we define the map $\varphi: D_3/H \rightarrow C_2$ as $\varphi(H) = 1$, $\varphi(\beta H) = -1$, we find that $\varphi(g_1g_2) = \varphi(g_1)\varphi(g_2)$ for $g_1, g_2 \in D_3/H$. This is an example of a *group isomorphism*, which identifies the two groups as being abstractly the same.

2.3 Homomorphisms and Exact Sequences

Definition 7 (Group Homomorphism) Let H and G be two groups. A homomorphism is a map $\varphi: H \rightarrow G$ such that $\varphi(h_1 \cdot h_2) = \varphi(h_1) \cdot \varphi(h_2)$ for all $h_1, h_2 \in H$. The set of all such homomorphisms is denoted $\text{hom}(H, G)$.

Definition 8 (Kernel and Image) The *kernel* and *image* of $\varphi \in \text{hom}(H, G)$ are defined as

$$\ker(\varphi) = \{h \in H: \varphi(h) = \mathbf{1}\}$$

$$\text{im}(\varphi) = \{g \in G: g = \varphi(h) \text{ for some } h \in H\}$$

Definition 9 (Epimorphism, Monomorphism and Isomorphism) If $\ker(\varphi) = \mathbf{1}$ then φ is *injective*, meaning that $\varphi(h) = \varphi(h') \Rightarrow h = h'$. If $\text{im}(\varphi) = G$ we say that φ is *surjective* (onto G). A surjective homomorphism is called an *epimorphism*, denoted $\phi \in \text{epi}(G_1, G_2)$ and an injective homomorphism is called a *monomorphism*, denoted $\phi \in \text{mono}(G_1, G_2)$. A homomorphism which is both injective and surjective is called an *isomorphism*, denoted $\phi \in \text{iso}(G_1, G_2)$. If there exists an isomorphism between G_1 and G_2 we write $G_1 \simeq G_2$ and say that G_1 and G_2 are isomorphic groups, meaning that they are structurally identical.

Exercise 2 Show that the additive group of real numbers $(\mathbb{R}, +)$ and the multiplicative group of positive reals (\mathbb{R}^+, \cdot) are isomorphic. Hint: use the exponential map.

Exercise 3 Let $\varphi \in \text{hom}(H, G)$. Show that $\ker(\varphi) \triangleleft H$ and that $\text{im}(\varphi) \triangleleft G$.

Definition 10 (Coimage) Let $\varphi \in \text{hom}(H, G)$. Since $\ker(\varphi) \triangleleft G$ (always normal subgroup), we can form the quotient. This is called the *coimage*

$$\text{coim}(\varphi) := H / \ker(\varphi).$$

Definition 11 (Cokernel) Let $\varphi \in \text{hom}(H, G)$. If $\text{im}(\varphi) \triangleleft G$ we can form the quotient $C = G / \text{im}(\varphi)$. This is called the *cokernel* of φ .

It is very useful to present homomorphisms in terms of *exact sequences*.

Definition 12 (Exact Sequence) A sequence

$$G_0 \xrightarrow{\varphi_1} G_1 \xrightarrow{\varphi_2} G_2 \xrightarrow{\varphi_3} \dots \xrightarrow{\varphi_n} G_n$$

of groups and group homomorphisms is called an *exact sequence* if $\text{im}(\varphi_i) = \ker(\varphi_{i+1})$ for every i .

Let $\mathbf{1}$ denote the trivial group containing just the identity element. An exact sequence

$$\mathbf{1} \longrightarrow H \xrightarrow{\varphi} G$$

indicates that $\varphi \in \text{hom}(H, G)$ is a monomorphism. To see this we note that the only homomorphism in $\text{hom}(\mathbf{1}, H)$ is the trivial map $\mathbf{1} \mapsto \mathbf{1}$, thus $\ker(\varphi) = \mathbf{1}$. We will frequently also use a hooked arrow $H \xhookrightarrow{\phi} G$ to indicate that ϕ is a monomorphism.

Exactness of the sequence

$$H \xrightarrow{\varphi} G \longrightarrow \mathbf{1}$$

means that $\varphi \in \text{hom}(H, G)$ is an epimorphism, since the only homomorphism in $\text{hom}(G, \mathbf{1})$ is the map sending G to $\mathbf{1}$ and hence $\text{im}(\varphi) = G$. We will also use a double arrow $H \xrightarrow[\phi]{} G$ to visualise $\phi \in \text{epi}(H, G)$. The exact sequence

$$\mathbf{1} \longrightarrow H \xrightarrow{\varphi} G \longrightarrow \mathbf{1}$$

means that φ is both epi- and mono- and hence it is an isomorphism and $H \simeq G$.

Definition 13 (Short Exact Sequence) A *short exact sequence* is an exact sequence of the form

$$\mathbf{1} \longrightarrow H \xrightarrow{\varphi_G} G \xrightarrow{\varphi_K} K \longrightarrow \mathbf{1} . \quad (3)$$

This indicates that $H \simeq \text{im}(\varphi_G) \triangleleft G$ and that $K \simeq G / \text{im}(\varphi_G) = \text{coker}(\varphi_G)$, or by a slight abuse of notation (identification by isomorphisms) we write this as $H \triangleleft G$ and $K = G/H$.

We ask the reader to think through the meaning of the short exact sequence carefully! Since φ_G is injective, it must define an isomorphism between H and its image in G . To see that $H \triangleleft G$ is a normal subgroup and that φ_K is a projection of G onto G/H , we compute for $g \in G$ and $h \in \text{im}(\varphi_G) = \ker(\varphi_K)$:

$$\varphi_K(gh) = \varphi_K(g)\varphi_K(h) = \varphi_K(g)\mathbf{1} = \mathbf{1}\varphi_K(g) = \varphi_K(h)\varphi_K(g) = \varphi_K(hg),$$

so all elements of gH and Hg are sent to the same element in C . Furthermore, if $\varphi_K(g) = \varphi_K(g')$, we must have $\mathbf{1} = \varphi_K(g')^{-1}\varphi_K(g) = \varphi_K(g'^{-1}g)$ thus $g'^{-1}g = h \in \ker(\varphi_K)$ and $g = g'h$. We conclude that $\varphi_K(g) = \varphi_K(g')$ if and only if g and g' belong to the same left and right coset. Finally we check that $\varphi_K(gH \cdot g'H) = \varphi_K(gg'H)$ and hence $K \simeq G/H$.

Example 7 Let D_n be the dihedral group and $C_n = \{e^{2\pi ij/n}\}_{j=0}^{n-1} \subset \mathbb{C}$ the cyclic group of n elements, identified with the multiplicative group of complex n th roots of unity. There is a short exact sequence

$$\mathbf{1} \longrightarrow C_n \xrightarrow{\varphi_1} D_n \xrightarrow{\varphi_2} C_2 \longrightarrow \mathbf{1}, \quad (4)$$

where $\varphi_1(e^{2\pi ij/n}) = \alpha^j$ and $\varphi_2(\alpha^j) = 1$, $\varphi_2(\beta\alpha^j) = -1$.

An exact sequence of the form

$$\mathbf{1} \longrightarrow K \xrightarrow{\ker(\varphi)} H \xrightarrow{\varphi} G \xrightarrow{\text{coker}(\varphi)} C \longrightarrow \mathbf{1} \quad (5)$$

indicates that $K \simeq \ker(\varphi)$ and $C \simeq \text{coker}(\varphi)$. Note that we have now called the injection arrow of K into H for the kernel of φ and the projection arrow from G onto C the cokernel of φ . This definition of kernels and cokernels as arrows rather than objects (groups) is standard in category theory language, where any mathematical property of an object is defined in terms of arrows into and out of the object. We will call both the arrows and their images for kernels and cokernels. If we really need to distinguish, we call the arrow ‘(co)kernel homomorphism’ and the group ‘(co)kernel group’.

Definition 14 (Kernel and Cokernel Homomorphisms²) The kernel homomorphism of $\varphi \in \text{hom}(H, G)$ is defined as a monomorphism, denoted $\ker(\varphi) \in \text{mono}(K, H)$, such that the image of $\ker(\varphi)$ is the kernel group of φ . The cokernel homomorphism of $\varphi \in \text{hom}(H, G)$ is defined as an epimorphism, denoted $\text{coker}(\varphi) \in \text{epi}(G, C)$, such that the image of $\text{coker}(\varphi)$ is the cokernel group of φ .

Definition 15 (Image and Coimage Homomorphisms) Let $\varphi \in \text{hom}(G_1, G_2)$ and let $K = G_1/\ker(\varphi)$ be the coimage group. The coimage homomorphism is defined as an epimorphism $\text{coim}(\varphi) \in \text{epi}(G_1, K)$ and the image homomorphism is a monomorphism $\text{im}(\varphi) \in \text{mono}(K, G_2)$ such that

$$\varphi = \text{im}(\varphi) \circ \text{coim}(\varphi).$$

Note that these homomorphisms are defined up to an isomorphism of K , so there is a freedom in how to represent $K = G_1/\ker(\varphi)$. However, the image and coimage homomorphisms must be consistent with this choice. The following example is illustrating this point.

Example 8 This example should make the above discussion more familiar to computational scientists. Consider the set of all abelian groups $(\mathbb{R}^n, +)$ for all $n \in \mathbb{N}$ and the continuous homomorphisms between these. This is an example of

²Check Wikipedia for a proper categorical definition of kernel and cokernel which only refers to properties of arrows.

a category,³ where \mathbb{R}^n are ‘objects’ and homomorphisms are the ‘arrows’. The set $\text{hom}(\mathbb{R}^n, \mathbb{R}^m)$ can be identified with the set of $m \times n$ matrices

$$\text{hom}(\mathbb{R}^n, \mathbb{R}^m) \approx \mathbb{R}^{m \times n}$$

and the composition of homomorphisms is given as matrix products. A monomorphism is a matrix with full column-rank, and an epimorphism a matrix with full row-rank. The isomorphisms are the invertible matrices.

For $A \in \mathbb{R}^{m \times n}$ we want to compute the homomorphisms (matrices) $\text{im}(A)$, $\text{coim}(A)$, $\ker(A)$ and $\text{coker}(A)$. Recall the singular value decomposition

$$A = U\Sigma V,$$

where $\Sigma \in \mathbb{R}^{m \times k}$ is a diagonal matrix with non-negative diagonal elements $\sigma_i = \Sigma_{i,i}$ called singular values. We assume that $\sigma_i \geq \sigma_{i+1}$ and $\sigma_{k+1} = 0$, so there are k positive singular values. The two matrices $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal. We block up the three matrices as

$$U = \begin{pmatrix} U_1 & U_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \quad V = \begin{pmatrix} V_1 \\ V_2 \end{pmatrix},$$

where $U_1 \in \mathbb{R}^{m \times k}$, $U_2 \in \mathbb{R}^{m \times (m-k)}$, $\Sigma_{11} \in \mathbb{R}^{k \times k}$, $\Sigma_{12} \in \mathbb{R}^{k \times (n-k)}$, $\Sigma_{21} \in \mathbb{R}^{(n-k) \times k}$, $\Sigma_{22} \in \mathbb{R}^{(n-k) \times (n-k)}$, $V_1 \in \mathbb{R}^{k \times n}$ and $V_2 \in \mathbb{R}^{(n-k) \times n}$. The matrix Σ_{11} is diagonal with positive diagonal and Σ_{12} , Σ_{21} and Σ_{22} are all zero. Since U and V are orthogonal, their inverses are $U^{-1} = U^T$ and $V^{-1} = V^T$. From $A \in \text{hom}(\mathbb{R}^n, \mathbb{R}^m)$ we get the four homomorphisms

$$\begin{aligned} \ker(A) &= V_2^T \in \text{mono}(\mathbb{R}^{n-k}, \mathbb{R}^n) \\ \text{coker}(A) &= U_2^T \in \text{epi}(\mathbb{R}^m, \mathbb{R}^{m-k}) \\ \text{coim}(A) &= V_1 \in \text{epi}(\mathbb{R}^n, \mathbb{R}^k) \\ \text{im}(A) &= U_1 \Sigma_{11} \in \text{mono}(\mathbb{R}^k, \mathbb{R}^m). \end{aligned}$$

We leave the verification of this to the reader. To check that the kernel and cokernel homomorphisms are correctly defined, you must verify that

$$\mathbf{0} \longrightarrow \mathbb{R}^{n-k} \xrightarrow{V_2^T} \mathbb{R}^n \xrightarrow{A} \mathbb{R}^m \xrightarrow{U_2^T} \mathbb{R}^{m-k} \longrightarrow \mathbf{0}$$

is an exact sequence.

³A category is a collection of objects and arrows between the objects such that the composition of an arrow from A to B and an arrow from B to C yields an arrow from A to C .

To check the image and coimage, you must verify that the diagram

$$\begin{array}{ccccc}
 & & \mathbf{0} & & \\
 & & \downarrow & & \\
 \mathbb{R}^n & \xrightarrow{V_1} & \mathbb{R}^k & \longrightarrow & \mathbf{0} \\
 & \searrow A & \downarrow U_1 \Sigma_{11} & & \\
 & & \mathbb{R}^m & &
 \end{array}$$

commutes (meaning that you get the same result if you follow different paths between two objects) and that the row and the column are exact.

The image- coimage factorisation is $A = (U_1 \Sigma_{11})V_1$, where the left term has full column rank and the right has full row-rank. Such a factorisation is not unique, for any invertible $k \times k$ matrix X , we could instead do the factorisation as $A = (U_1 \Sigma_{11}X)(X^{-1}V_1)$, which is another factorisation of A in a product of a matrix with full column rank and a matrix with full row-rank. The possibility of choosing X is expressed as ‘defined up to isomorphisms’.

Exercise 4 Repeat the example using the *QR*-factorisation instead of SVD.

2.4 Products of Groups and Split Exact Sequences

How can we construct more complicated groups from simpler ones? The two most important operations are called *direct product* and *semidirect product*.

Definition 16 (Direct Product) For two groups G and H we define their direct product $G \times H$ as a group defined on the set of pairs

$$G \times H = \{(g, h) : g \in G, h \in H\}$$

with product defined componentwise

$$(g, h) \cdot (g', h') = (gg', hh').$$

Example 9 For additive abelian groups we write the direct product as \oplus instead of \times . The abelian group $\mathbb{R}^2 = \mathbb{R} \oplus \mathbb{R}$ is defined on pairs of reals with the sum $(x, y) + (x', y') = (x + x', y + y')$ and $\mathbf{0} = (0, 0)$.

The semidirect product is a bit more involved. To motivate the definition, let us look at a particular group of all affine linear mappings on a vector space.

Example 10 (Affine Group) Let $V = \mathbb{R}^n$ be a vector space. Any vector space is also an abelian group (by forgetting scalar multiplication), so we can let V act on itself by translation $v, w \mapsto v + w$. An other action is the linear action of $GL(V)$ on V by

matrix-vector product $A \cdot v = Av$. The affine action for $(A, b) \in \mathrm{GL}(V) \times V$ on V is given as

$$(A, b) \cdot v := Av + b.$$

What is the group structure on $\mathrm{GL}(V) \times V$ compatible with this action? We compute:

$$(A', b') \cdot ((A, b) \cdot v) = (A', b') \cdot (Av + b) = A'Av + A'b + b' = (AA', b' + A'b) \cdot v,$$

thus we obtain the group product

$$(A', b') \cdot (A, b) = (AA', b' + A'b).$$

The identity element is $(I, 0)$, where I is the identity matrix. This is an important example of a semidirect product. We write the affine group as $\mathrm{Aff}(V) := \mathrm{GL}(V) \rtimes V$.

Definition 17 (Semidirect Product) A semidirect product is defined from two groups G and H and an action $\cdot : G \times H \rightarrow H$. We write the products in G and H as gg' and hh' , and the action as $g \cdot h$. The semidirect product of G and H , written $G \rtimes H$ is the set of pairs (g, h) with the product

$$(g, h) \cdot (g', h') := (gg', h(g \cdot h')).$$

The direct product is the special case of semidirect product where $g \cdot h = h$ for all g and h .

Example 11 Let C_2 act on C_n by complex conjugation, $(-1) \cdot z = \bar{z}$ for all $z \in C_n$. We claim that $D_n \simeq C_2 \rtimes C_n$, with respect to this action. We have that $C_2 \times C_n = \{(\pm 1, \omega^j)\}_{j=0}^{n-1}$ where $\omega = e^{2\pi i/n}$. Let $\alpha = (1, \omega)$ and $\beta = (-1, 1)$. We ask the reader to verify that these two elements generate $C_2 \rtimes C_n$ and satisfy the relations (2).

From this example and (4) we might be tempted to believe that (3) implies $G \simeq G/H \rtimes H$. This is, however, NOT true in general.

Example 12 We have a short exact sequence

$$\mathbf{1} \longrightarrow \mathbb{Z}_2 \xrightarrow{\cdot 4} \mathbb{Z}_8 \xrightarrow{\text{mod } 4} \mathbb{Z}_4 \longrightarrow \mathbf{1},$$

however, there is no way \mathbb{Z}_8 can be written as a direct or semidirect product of \mathbb{Z}_2 and \mathbb{Z}_4 . On the other hand, we have

$$\mathbf{1} \longrightarrow \mathbb{Z}_3 \xrightarrow{\cdot 4} \mathbb{Z}_{12} \xrightarrow{\text{mod } 4} \mathbb{Z}_4 \longrightarrow \mathbf{1},$$

corresponding to a decomposition $\mathbb{Z}_{12} \simeq \mathbb{Z}_3 \times \mathbb{Z}_4$. The difference between these two cases is that the latter *splits* in the sense defined below. We return to this example in Sect. 3.1.

Definition 18 (Split Exact Sequence) The short exact sequence

$$\mathbf{1} \longrightarrow H \xrightarrow{\varphi_G} G \xrightleftharpoons[\varphi_s]{\varphi_K} K \longrightarrow \mathbf{1} \quad (6)$$

is called *right split* if there exists a homomorphism $\varphi_s: K \rightarrow G$ such that the composition $\varphi_K \circ \varphi_s = \text{Id}_K$ (the identity map). The exact sequence

$$\mathbf{1} \longrightarrow H \xleftarrow[\varphi_s]{\varphi_G} G \xrightarrow{\varphi_K} K \longrightarrow \mathbf{1} \quad (7)$$

is called *left split* if there exists a homomorphism $\varphi_s: G \rightarrow H$ such that $\varphi_s \circ \varphi_G = \text{Id}_H$.

Theorem 1

- A semidirect product decomposition $G = K \rtimes H$ is equivalent to a right split short exact sequence.
- A direct product decomposition $G = H \times K$ is equivalent to a left split short exact sequence.
- Any left split short exact sequence is also right split (but not vice versa).

Before we prove this theorem, let us discuss decomposition of G with respect to any subgroup $H < G$. As a set, G decomposes into a disjoint union in right cosets $G = \cup_i Hk_i$, where the subset $\{k_i\} \subset G$ consists of exactly one element k_i from each coset Hg . Such k_i are called *coset representatives*. Hence, we have a unique factorisation $g = hk$, $h \in H$, $k \in \{k_i\}$, identifying G and $\{k_i\} \times H$ as sets. An important question is whether or not the coset representatives can be chosen in a canonical (natural) way, so that this identification also carries along a (semidirect product) group structure.

Proof (Theorem 1) Given the right split short exact sequence (6). For any function $\varphi_s: K \rightarrow G$ such that $\varphi_K \circ \varphi_s = \text{Id}_K$ we have that $\text{im}(\varphi_s) \subset G$ is a set of coset representatives. This defines a set-mapping

$$\xi: K \times H \rightarrow G, \quad (k, h) \mapsto g = \varphi_G(h)\varphi_s(k),$$

with inverse given as $k = \varphi_K(g)$, and we find h from $\varphi_G(h) = g\varphi_s(k)^{-1}$. Now let $g = \xi(k, h)$ and $g' = \xi(k', h')$ be arbitrary. If φ_s is a homomorphism

$$gg' = \varphi_G(h)\varphi_s(k)\varphi_G(h')\varphi_s(k') = \varphi_G(h)\varphi_s(k)\varphi_G(h')\varphi_s(k^{-1})\varphi_s(kk').$$

The K -part of gg' is $\varphi_K(gg') = kk'$, hence $\varphi_G(h)\varphi_s(k)\varphi_G(h')\varphi_s(k^{-1}) \in \text{im}(H)$, and we conclude that $k \cdot h: K \times H \rightarrow H$ defined such that

$$\varphi_G(k \cdot h) = \varphi_s(k)\varphi_G(h')\varphi_s(k^{-1})$$

is a well-defined action of K on H . We see that $K \rtimes H$ with the semidirect product $(k, h)(k', h') = (kk', h(k \cdot h'))$ is isomorphic to G .

Conversely, it is straightforward to check that if $G = K \rtimes H$ we have that $\varphi_G(h) = (1, h)$, $\varphi_K(k, h) = k$ and $\varphi_s(k) = (k, 1)$ defines a right split short exact sequence, and we have established the first point in the theorem.

To prove the second point, we assume the existence of a left split short exact sequence (7). We want to factor $g = hk$ for $h \in \text{im}(\varphi_G)$ and k in some set of coset representatives. We find $h = \varphi_G \circ \varphi_s(g)$ and $k = h^{-1}g$, thus $k = \sigma(g)$ where

$$\sigma(g) := (\varphi_G \circ \varphi_s(g))^{-1} g.$$

If φ_s is a homomorphism we can check that $\sigma(hg) = \sigma(g)$ and $H\sigma(g) = Hg$, hence σ picks a unique representative from each coset. We conclude that the mapping $\psi: G \rightarrow K \times H$, $g \mapsto (\varphi_K(g), \varphi_s(g))$ is an invertible set function. It is clearly a group homomorphism and hence also an isomorphism. We conclude that $G \simeq K \times H$. Conversely it is easy to check that any direct product $G = K \times H$ is left split.

Since a direct product is also a semi-direct product, we conclude the third point that left split implies right split. \square

2.5 Domains in Computational Mathematics

It is time to be a bit more philosophical and less technical on the role of groups in computational mathematics. A fundamental question is what do we mean by a ‘Domain’? More specifically, what abstract properties do we require from the ‘domain’ of a differential equation? Over the last century, mathematicians agree that the notion of a (*differentiable*) manifold is a powerful abstract setting for a general theory of differential equations. Manifolds are sets endowed with a ‘differentiable structure’, we have *points* in the domain as well as *tangents* at a given point. Points have a position (coordinates), tangents are velocities, specifying a direction and a speed. The most important property of manifolds is that they support scalar functions (real or complex), and derivations of these in directions specified by tangent vectors. Examples of manifolds are the familiar space \mathbb{R}^n , but also spaces like the surface of a sphere. In \mathbb{R}^n both points and tangents are vectors in \mathbb{R}^n , but the spherical surface is a good example of a space where these two different things should not be confused!

The mathematical definition of a manifold does not have enough structure to be suitable as an abstraction of a computational domain. Manifolds have tangents, but there is no (practical) way of moving in the direction of a given tangent. In pure mathematics motions arise from the concept of the solution operator (flow map) of a tangent vector field (= solution of ordinary differential equations, ODEs), but in computations one cannot assume that differential equations can be solved exactly. For doing computations we need to specify a set of motions which we assume can be computed fast and accurately. Here groups and group actions come in handy! For the

purpose of solving ODEs, it has turned out to be very useful to study algorithms on *homogeneous spaces*, which are domains together with a transitive action of a Lie group. One example is \mathbb{R}^n , which acts on itself by translations, the basic motions are obtained by adding vectors. An other example is the surface of a sphere, under the action of the orthogonal group. A substantial theory of numerical Lie group integration has been developed over the last two decades [14].

Among the homogeneous spaces, abelian Lie groups are the most important for practical computations. Most modelling in physics and engineering take place in \mathbb{R}^n or subdomains of this. As a *domain* (set of positions), the important structure of \mathbb{R}^n is its abelian Lie group structure, where we can move around using translations. As a tangent space (set of velocities), the important structure is the vector space structure (Lie algebra structure) of \mathbb{R}^n . These spaces play very different roles in theory and in computations and should not be confused!

The theory of abelian groups is *much* simpler than general groups and Fourier analysis is a ubiquitous tool which is tightly associated with the group structure of these spaces. The relationship between the continuous and the discrete is a fundamental aspect of computational algorithms, such as the relationship between continuous groups such as \mathbb{R}^n and discrete subgroups (lattices). The Fourier transform can be computed fast on finite abelian groups, but not on T nor \mathbb{R} . Without a good mathematical theory and supporting software, it is, however not trivial to relate the continuous and the discrete Fourier transforms. This is in particular the case for general sampling lattices in \mathbb{R}^n .

The general theory of subgroups, quotients and exact sequences turns out to be a very useful framework for developing and analysing computational algorithms. This is the topic of the next chapter.

3 Abelian Groups, Fourier Analysis, Lattices and Sampling

In this chapter we will introduce abelian groups as domains for computations. We will in particular present a general theory of discretisation lattices, sampling theory and the relationship between discrete and continuous Fourier analysis, and discuss a variety of computational algorithms. Circulant matrices and their multidimensional analogues is also a central theme.

3.1 *Introduction to Abelian Groups*

3.1.1 Definition and Basic Properties

Using the additive notation with $+$ and 0 , we define abelian groups:

Definition 19 (Abelian Group) An abelian group is a set G with a binary operation $+: G \times G \rightarrow G$ called the *sum*, such that

1. The sum is associative, $x + (y + z) = (x + y) + z$ for all $x, y, z \in G$.
2. The sum is commutative, $x + y = y + x$.
3. There exists an identity element $\mathbf{0} \in G$ such that $x + \mathbf{0} = x$ for all $x \in G$.
4. Every element $x \in G$ has an inverse $-x \in G$ such that $x + (-x) = \mathbf{0}$.

For abelian groups the direct product is the same as a *direct sum*,⁴ we write this as $H \oplus K \equiv H \times K$. This means, as before, that $H \oplus K = \{(k, h)\}$ with $(k, h) + (k', h') = (k + k', h + h')$.

Abelian groups are much simpler than general groups, since there is no difference between ‘left’ and ‘right’, they enjoy the following properties:

- Any subgroup $H < G$ is a normal subgroup.
- A short exact sequence is right split if and only if it is left split, thus a split short exact sequence is always of the form

$$\mathbf{0} \longrightarrow H \xrightleftharpoons[\psi_H]{\varphi_G} G \xrightleftharpoons[\psi_G]{\varphi_K} K \longrightarrow \mathbf{0} \quad (8)$$

corresponding to the direct sum decomposition $G = H \oplus K$.

3.1.2 Topology

In order to develop a mathematical theory of Fourier analysis, it is necessary to have some topology (notion of continuity) on the groups. The standard foundation of Fourier analysis on groups are so called *locally compact* groups.

Definition 20 (Locally Compact Group) A group is called locally compact if it has a topology such that every point has a compact neighbourhood, and such that the product and inverse in the group are continuous operations.

Definition 21 (LCA—Locally Compact Abelian) LCA denotes the locally compact abelian groups.

Between topological groups, homomorphisms are always assumed continuous, and when we talk about a subgroup $H < G$, we will always assume that H is a *closed subset*. For example, the rationals $(\mathbb{Q}, +)$ is algebraically a subgroup of $(\mathbb{R}, +)$, but it is not a topologically closed subset.

⁴The category theoretical definition of *products* and *coproducts* are dual of each other, but for abelian groups they coincide.

3.1.3 The Elementary Groups

In these lectures we will mainly focus the *elementary* abelian groups, those that can be obtained from \mathbb{R} and \mathbb{Z} by taking direct sums, (closed) subgroups and quotients. The topology for these is what we are used to, e.g. \mathbb{Z} has the discrete topology where every subset is an open set (and also closed!), and \mathbb{R} has the familiar topology of the real line based on open intervals defining open subsets. The elementary LCAs are isomorphic to one of the following:

Definition 22 The *elementary LCAs* are:

- The *reals* \mathbb{R} under addition, with the standard definition of open sets.
- The *integers* \mathbb{Z} under addition (with the discrete topology). This is also known as the infinite cyclic group.
- The 1-dimensional *torus*, or *circle* $T = \mathbb{R}/\mathbb{Z}$ defined as $[0, 1) \subset \mathbb{R}$ under addition modulo 1, with the circle topology.
- The *cyclic group* of order k , $\mathbb{Z}_k = \mathbb{Z}/k\mathbb{Z}$, which consists of the integers $0, 1, \dots, k - 1$ under addition modulo k (with the discrete topology).
- *Direct sums* of the above spaces, $G \oplus H$, in particular \mathbb{R}^n (real n -space), T^n (the n -torus) and all finitely generated abelian groups.

A set of *generators* for a group is a subset such that any element in the group can be written as a finite sum (or difference) of the generators. The *finitely generated abelian groups* are those having a finite set of generators. These are easy to describe, since they are always isomorphic to a direct sum of \mathbb{Z} and \mathbb{Z}_{n_i} . We take this as a definition, but keep in mind that they may appear in disguise, as e.g. the multiplicative group C_n isomorphic to \mathbb{Z}_n .

Definition 23 (Finitely Generated Abelian Group, FGA) An FGA is a group isomorphic to

$$\mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \cdots \oplus \mathbb{Z}_{n_k} \oplus \mathbb{Z}^d.$$

We represent this as the space of integer column vectors of length $k + d$ under addition mod n_i in first k components and integer addition in the last d components. The canonical generators are $(1, 0, \dots, 0)^T$, $(0, 1, 0, \dots, 0)^T$, ..., $(0, \dots, 0, 1)^T$. Note that $\mathbb{Z}_1 = \mathbf{0}$ and $\mathbf{0} \oplus G \simeq G$, hence we can remove the terms \mathbb{Z}_{n_i} whenever $n_i = 1$.

We will in the sequel use the following compact notation for FGAs

$$\mathbb{Z}_{\mathbf{k}} \oplus \mathbb{Z}^d := \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \cdots \oplus \mathbb{Z}_{n_k} \oplus \mathbb{Z}^d,$$

where $\mathbf{k} = (n_1, n_2, \dots, n_k)$ is a multi-index of length k . The number $k + d$ (number of generators) is called the *rank* of the FGA, but the rank is not an invariant under isomorphisms.

FGAs are similar to vector spaces, but where the ‘scalars’ are the integers \mathbb{Z} instead of the usual fields \mathbb{R} or \mathbb{C} . The generators of the FGA are similar to basis vectors for a vector space. Homomorphisms between FGAs can always be written as integer matrices representing how the homomorphism acts on the canonical generators. Note that we will always assume that the target space knows the periods of its components, e.g. the homomorphism $\cdot 2: \mathbb{Z} \rightarrow \mathbb{Z}_3$ (multiplication by 2), sends $0 \mapsto 0, 1 \mapsto 2, 2 \mapsto 4 \equiv 1(\text{mod } 3)$, etc. We will not write the reduction modulo 3 explicitly.

Note that not every integer matrix (of appropriate dimensions) represents a homomorphism. The obstruction is that every integer vector congruent to $\mathbf{0}$ in the source group must be mapped to an integer vector congruent to $\mathbf{0}$ in the target group. For example $\cdot 2$ does not define a homomorphism from \mathbb{Z}_2 to \mathbb{Z}_3 since $2 \cdot 2 \neq 0(\text{mod } 3)$, however $\cdot 4: \mathbb{Z}_3 \rightarrow \mathbb{Z}_{12}$ is a homomorphism since $4 \cdot 3k = 0(\text{mod } 12)$ for every $k \in \mathbb{Z}$.

The notion of the dimension is less clear in the theory of FGAs compared to standard linear algebra (where the size of a basis is invariant under basis change). In Example 12, we claimed that $\mathbb{Z}_{12} \simeq \mathbb{Z}_3 \oplus \mathbb{Z}_4$, so isomorphic groups can have different numbers of independent generators. In this case $(4, 3): \mathbb{Z}_3 \oplus \mathbb{Z}_4 \rightarrow \mathbb{Z}_{12}$ and $(1, -1)^T: \mathbb{Z}_{12} \rightarrow \mathbb{Z}_3 \oplus \mathbb{Z}_4$ define isomorphisms between the two groups. (Check this!)

In general we have that $\mathbb{Z}_p \oplus \mathbb{Z}_q \simeq \mathbb{Z}_{pq}$ if and only if p and q are relative prime numbers (i.e. if their greatest common divisor, gcd, is 1). To compute the isomorphism between these, we employ the *extended Euclidean algorithm* (matlab function ‘gcd’), which given two positive integers p and q produces two integers a and b such that

$$ap + bq = \gcd(p, q).$$

If $\gcd(p, q) = 1$, we have that $(q, p): \mathbb{Z}_p \oplus \mathbb{Z}_q \rightarrow \mathbb{Z}_{pq}$ and $(b, a)^T: \mathbb{Z}_{pq} \rightarrow \mathbb{Z}_p \oplus \mathbb{Z}_q$ are isomorphisms. We can also illustrate the isomorphism by the split short exact sequence

$$\mathbf{0} \longrightarrow \mathbb{Z}_p \xleftarrow[\cdot b]{\cdot q} \mathbb{Z}_{pq} \xleftarrow[\cdot p]{\cdot a} \mathbb{Z}_q \longrightarrow \mathbf{0}$$

Check yourself that this is split exact!

We have two standard ways of representing FGAs uniquely (up to isomorphisms), the first of these has the largest possible number of generators and the latter the smallest possible:

Theorem 2 (Classification of FGA) *If G is an FGA, and $G \neq \mathbf{0}$, then G is isomorphic to a group of the form called the primary factor decomposition*

$$\mathbb{Z}_{p_1^{n_1}} \oplus \mathbb{Z}_{p_2^{n_2}} \oplus \cdots \oplus \mathbb{Z}_{p_\ell^{n_\ell}} \oplus \mathbb{Z}^n \tag{9}$$

where p_i are primes, $p_1 \leq p_2 \leq \dots \leq p_\ell$, $n_i \in \mathbb{N}$ and $n_i \leq n_{i+1}$ whenever $p_i = p_{i+1}$. Furthermore G is also isomorphic to a group of the form called the invariant factor decomposition

$$\mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \dots \oplus \mathbb{Z}_{n_k} \oplus \mathbb{Z}^n \quad (10)$$

where $n_i > 1$ and $n_i | n_{i+1}$, $1 \leq i < k$ (n_i divides n_{i+1}). In both forms the representation is unique, i.e. two FGAs are isomorphic iff they can be transformed into the same canonical form.

3.2 Computing with FGAs

For applications in lattice sampling algorithms and computational Fourier analysis, it is important to do computations on FGAs and homomorphisms between these. It is our philosophy that software in computational mathematics should closely follow mathematical definitions. *Object oriented programming* is founded on the idea that programming is ‘what’ + ‘how’, i.e. the software is constructed from classes where there is a distinction between the (public) signature (what) of the class and the (private) implementation (how). The signature consists of the functions operating on the structure and the implementation consists of data structures and algorithms. To help finding useful abstractions for defining the ‘what’ part of program design, we have found mathematical category theory very useful. Categories consists of objects and arrows between the objects, just as we have seen in the discussion of exact sequences above. In category theory one does not explicitly describe what is ‘inside’ an object, the only mathematical properties one is interested in are those that can be described in terms of arrows into and out of the object. This philosophy fits very well with object oriented programming design, and a categorical definition of a mathematical object is a good starting point for object oriented software construction.

For example, a split exact sequence

$$\mathbf{0} \longrightarrow G_1 \xrightleftharpoons[\text{proj}_1]{\text{inj}_1} G_1 \oplus G_2 \xrightleftharpoons[\text{inj}_2]{\text{proj}_2} G_2 \longrightarrow \mathbf{0}$$

could be taken as the *definition* of the direct sum $G_1 \oplus G_2$. The ‘object’ $G_1 \oplus G_2$ is defined by the existence of the four morphisms inj_1 , inj_2 , proj_1 and proj_2 such that $\text{proj}_1 \circ \text{inj}_1 = \text{Id}_{G_1}$ (the identity homomorphism), $\text{proj}_2 \circ \text{inj}_2 = \text{Id}_{G_2}$ and exactness of the diagram in both directions. The usual *implementation* (‘how’) of the direct product $G_1 \oplus G_2$ is as pairs (g_1, g_2) , where the arrows are $\text{inj}_1(g_1) = (g_1, 0)$, $\text{proj}_1((g_1, g_2)) = g_1$, and similarly for G_2 . Could there possibly be any other implementation of the direct sum? Yes, for high dimensional n there are different ways of representing \mathbb{R}^n . The most common is just as vectors of length n , but if many of the vectors are sparse, one could instead use a sparse representation where

only the non-zero components are stored. It is important to realise that these two implementations are isomorphic realisations of the ‘specification’ given by the split exact sequence.

3.2.1 Abelian Categories

Category theory gives an important hint on what are the most fundamental properties we should know about when designing software. The collection of all FGAs and homomorphisms between these form an *abelian category*, where each object is an FGA and each arrow is a homomorphism between two FGAs. Abelian categories have the following properties:

- There is a *zero object* $\mathbf{0}$. For any object G there is a unique $\mathbf{0}$ arrow $\mathbf{0} \rightarrow G$ and a unique arrow $G \rightarrow \mathbf{0}$.
- We can form the *product* and *coproduct* of any two objects. In the setting of FGAs, these two are the same, represented by the direct sum of two abelian groups $G_1 \oplus G_2$ and the arrows in and out of the sum.
- The set $\text{hom}(H, G)$ of all morphisms from H to G is an object in the category, i.e. it contains the $\mathbf{0}$ -arrow, any two arrows can be added, and furthermore the composition $\circ: \text{hom}(H, G) \times \text{hom}(G, K) \rightarrow \text{hom}(K, H)$ is bilinear.
- Every homomorphism $\varphi \in \text{hom}(H, G)$ has a kernel $\ker(\varphi) \in \text{hom}(K, H)$ and a cokernel $\text{coker}(\varphi) \in \text{hom}(G, C)$, such that the following is an exact sequence

$$\mathbf{0} \longrightarrow K \xrightarrow{\ker(\varphi)} H \xrightarrow{\varphi} G \xrightarrow{\text{coker}(\varphi)} C \longrightarrow \mathbf{0} .$$

- Every monomorphism is a kernel of some homomorphism and every epimorphism is the cokernel of some homomorphism.
- Every homomorphism $\varphi \in \text{hom}(G_1, G_2)$ factors in the composition of an epimorphism followed by a monomorphism. The epimorphism is called the coimage, and the monomorphism is called the image,

$$\varphi = \text{im}(\varphi) \circ \text{coim}(\varphi).$$

All these properties should be implemented in a software package for computing with FGAs. Furthermore, there are a set of operations which are derived from the addition and composition of homomorphisms. We introduce some short hand notation for these. First three operations which are related to direct sums. For homomorphisms represented as matrices, these operations correspond to creating new matrices from matrix blocks. The matrix interpretation is based on FGAs being column vectors and the sum $G_1 \oplus G_2$ interpreted as putting the two column vectors on top of each other. For Matlab users semicolon notation is familiar. If $x \in G_1$ and $y \in G_2$ are column vectors, then $(x; y) \in G_1 \oplus G_2$ means that we put x and y together in a long column with x on top of y .

Definition 24 (Block Compositions of Homomorphisms)

- For $\phi_1 \in \text{hom}(G_1, H_1)$, $\phi_2 \in \text{hom}(G_2, H_2)$ we define

$$\phi_1 \oplus \phi_2 \in \text{hom}(G_1 \oplus G_2, H_1 \oplus H_2)$$

as

$$(\phi_1 \oplus \phi_2)(x; y) := (\phi_1(x); \phi_2(y)).$$

This corresponds to a diagonal 2×2 block matrix with ϕ_1 in upper left and ϕ_2 in lower right block, or the diagram

$$\begin{array}{ccccc} G_1 & \iff & G_1 \oplus G_2 & \iff & G_2 \\ \downarrow \phi_1 & & \downarrow \phi_1 \oplus \phi_2 & & \downarrow \phi_2 \\ H_1 & \iff & H_1 \oplus H_2 & \iff & H_2. \end{array}$$

- For $\phi_1 \in \text{hom}(G_1, H)$, $\phi_2 \in \text{hom}(G_2, H)$ we define

$$\phi_1 | \phi_2 \in \text{hom}(G_1 \oplus G_2, H)$$

as

$$(\phi_1 | \phi_2)(x; y) := \phi_1(x) + \phi_2(y).$$

This corresponds to putting two matrices horizontally in a 1×2 block matrix, or the diagram

$$\begin{array}{ccccc} G_1 & \iff & G_1 \oplus G_2 & \iff & G_2 \\ & \searrow \phi_1 & \downarrow \phi_1 | \phi_2 & \nearrow \phi_2 & \\ & & H & & . \end{array}$$

- For $\phi_1 \in \text{hom}(H, G_1)$, $\phi_2 \in \text{hom}(H, G_2)$ we define

$$\frac{\phi_1}{\phi_2} \in \text{hom}(H, G_1 \oplus G_2)$$

as

$$\left(\frac{\phi_1}{\phi_2} \right) (x) := (\phi_1(x); \phi_2(x)).$$

This corresponds to putting two matrices vertically in a 2×1 block matrix, or the diagram

$$\begin{array}{ccccc} & & H & & \\ & \swarrow \phi_1 & \downarrow \frac{\phi_1}{\phi_2} & \searrow \phi_2 & \\ G_1 & \iff & G_1 \oplus G_2 & \iff & G_2. \end{array}$$

The next two operations are factorisations of a homomorphism through another, which is similar to solving linear equations.

Definition 25 (Factorisation of a Homomorphism Through Another) We define two ways of solving for an unknown homomorphism x . The solution may not exist, or may not be unique (conditions apply).

- For $\phi_1 \in \text{hom}(G_1, H)$ and $\phi_2 \in \text{hom}(G_2, H)$ we denote $x = \phi_2 \setminus \phi_1$ a homomorphism $x \in \text{hom}(G_1, G_2)$ such that $\phi_2 \circ x = \phi_1$.

$$\begin{array}{ccc} & G_2 & \\ & \nearrow x & \downarrow \phi_2 \\ G_1 & \xrightarrow{\phi_1} & H \end{array}$$

- For $\phi_1 \in \text{hom}(H, G_1)$ and $\phi_2 \in \text{hom}(H, G_2)$ we denote $x = \phi_1 / \phi_2$ a homomorphism $x \in \text{hom}(G_2, G_1)$ such that $x \circ \phi_2 = \phi_1$.

$$\begin{array}{ccc} & G_2 & \\ & \nearrow x & \uparrow \phi_2 \\ G_1 & \xleftarrow{\phi_1} & H \end{array}$$

3.2.2 Free FGAs and Smith's Normal Form

The free finitely generated abelian groups are those which have no relations between the generators, i.e. the abelian groups \mathbb{Z}^n for $n \in \mathbb{N}$. These are particularly simple, since the set of integer matrices are in 1–1 correspondence with homomorphisms

$$\text{hom}(\mathbb{Z}^n, \mathbb{Z}^m) \approx \mathbb{Z}^{m \times n}.$$

The set $\text{hom}(\mathbb{Z}^n, \mathbb{Z}^m)$ is an FGA with addition defined as matrix addition and 0 being the zero matrix. The composition of homomorphisms is given by matrix products. From Cramers rule we realise that a matrix $A \in \mathbb{Z}^{n \times n}$ has an inverse in $\mathbb{Z}^{n \times n}$ if and only if $\det(A) = \pm 1$.

Definition 26 (Unimodular Matrix) A matrix $A \in \mathbb{Z}^{n \times n}$ with $\det(A) = \pm 1$ is called *unimodular* and represents an isomorphism in $\text{iso}(\mathbb{Z}^n, \mathbb{Z}^n)$. The unimodular $n \times n$ integer matrices are denoted $\text{GL}(n, \mathbb{Z})$.

Many fundamental properties of homomorphisms in $\text{hom}(\mathbb{Z}^n, \mathbb{Z}^m)$ are computed from the *Smith normal form* of A , a decomposition quite similar to the SVD. An algorithm for computing this is given in Wikipedia [31].

Theorem 3 An integer matrix $A \in \mathbb{Z}^{n \times n}$ can be decomposed in a product

$$A = U \Sigma V$$

where $U \in \text{GL}(m, \mathbb{Z})$ and $V \in \text{GL}(n, \mathbb{Z})$ are unimodular and $\Sigma \in \mathbb{Z}^{m \times n}$ is diagonal with non-negative diagonal elements. The diagonal elements $n_i = \Sigma_{ii}$ satisfy $n_i | n_{i+1} \forall 1 \leq i < k$ and $n_i = 0 \forall k < i \leq \min(m, n)$.

Theorem 4 Let $A \in \text{hom}(\mathbb{Z}^n, \mathbb{Z}^m)$ with Smith decomposition $A = U \Sigma V$, with matrices partitioned as

$$U = \begin{pmatrix} U_1 & U_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \quad V = \begin{pmatrix} V_1 \\ V_2 \end{pmatrix},$$

where $U_1 \in \mathbb{Z}^{m \times k}$, $U_2 \in \mathbb{Z}^{m \times (m-k)}$, $\Sigma_{11} \in \mathbb{Z}^{k \times k}$, $\Sigma_{12} \in \mathbb{Z}^{k \times (n-k)}$, $\Sigma_{21} \in \mathbb{Z}^{(n-k) \times k}$, $\Sigma_{22} \in \mathbb{Z}^{(n-k) \times (n-k)}$, $V_1 \in \mathbb{Z}^{k \times n}$ and $V_2 \in \mathbb{Z}^{(n-k) \times n}$. The matrix Σ_{11} has diagonal $\mathbf{k} = (n_1, n_2, \dots, n_k)$ such that $n_i | n_{i+1}$ and Σ_{12} , Σ_{21} and Σ_{22} are all zero. Partition U^{-1} and V^{-1} as

$$U^{-1} = \begin{pmatrix} U_1^{-1} \\ U_2^{-1} \end{pmatrix}, \quad V^{-1} = \begin{pmatrix} V_1^{-1} & V_2^{-1} \end{pmatrix},$$

where $U_1^{-1} \in \mathbb{Z}^{k \times m}$, $U_2^{-1} \in \mathbb{Z}^{(m-k) \times m}$, $V_1^{-1} \in \mathbb{Z}^{n \times k}$ and $V_2^{-1} \in \mathbb{Z}^{n \times (n-k)}$. Then

$$\ker(A) = V_2^{-1} \in \text{mono}(\mathbb{Z}^{n-k}, \mathbb{Z}^n)$$

$$\text{coker}(A) = U^{-1} = \frac{U_1^{-1}}{U_2^{-1}} \in \text{epi}(\mathbb{Z}^m, \mathbb{Z}_{\mathbf{k}} \oplus \mathbb{Z}^{m-k})$$

$$\text{coim}(A) = V_1 \in \text{epi}(\mathbb{Z}^n, \mathbb{Z}^k)$$

$$\text{im}(A) = U_1 \Sigma_{11} \in \text{mono}(\mathbb{Z}^k, \mathbb{Z}^m).$$

Proof Check that that the diagrams

$$\mathbf{0} \longrightarrow \mathbb{Z}^{n-k} \xrightarrow{V_2^{-1}} \mathbb{Z}^n \xrightarrow{A} \mathbb{Z}^m \xrightarrow{U^{-1}} \mathbb{Z}_{\mathbf{k}} \oplus \mathbb{Z}^{m-k} \longrightarrow \mathbf{0}$$

and

$$\begin{array}{ccccc} & & \mathbf{0} & & \\ & & \downarrow & & \\ \mathbb{Z}^n & \xrightarrow{V_1} & \mathbb{Z}^k & \longrightarrow & \mathbf{0} \\ & \searrow A & \downarrow U_1 \Sigma_{11} & & \\ & & \mathbb{Z}^m & & \end{array}$$

are commutative with exact rows and columns. \square

Example 13 Given $A \in \text{hom}(\mathbb{Z}^3, \mathbb{Z}^4)$ with Smith normal form

$$A = \begin{pmatrix} -20 & 8 & 16 \\ -6 & 0 & 6 \\ 0 & -12 & 6 \\ 4 & -16 & 4 \end{pmatrix} = \begin{pmatrix} 8 & 6 & 3 & 0 \\ 3 & 2 & 1 & 0 \\ 3 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 6 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & -4 & 1 \\ -1 & 2 & 0 \\ 0 & 1 & -1 \end{pmatrix} = U \Sigma V,$$

where

$$U^{-1} = \begin{pmatrix} -1 & 3 & 0 & 0 \\ 3 & -9 & 1 & 0 \\ -3 & 10 & -2 & 0 \\ 2 & -6 & 0 & 1 \end{pmatrix}, V^{-1} = \begin{pmatrix} -2 & -3 & -2 \\ -1 & -1 & -1 \\ -1 & -1 & -2 \end{pmatrix}.$$

From this we see that A generates a rank-2 subgroup of \mathbb{Z}^4 , spanned by $2 \cdot (8, 3, 3, 2)^T$ and $6 \cdot (6, 2, 1, 0)^T$. The quotient of \mathbb{Z}^4 with this subgroup is $\mathbb{Z}_2 \oplus \mathbb{Z}_6 \oplus \mathbb{Z}$ and the matrix U^{-1} projects onto this quotient. The kernel of A is the rank-1 subgroup of \mathbb{Z}^3 spanned by $(-2, -1, -2)^T$.

Example 14 Let $H < \mathbb{Z}^2$ be the subgroup spanned by $(-1, 3)^T$ and $(2, 2)^T$. Compute \mathbb{Z}^2/H and the projection. We compute the Smith factorisation of the generators

$$A = \begin{pmatrix} -1 & 2 \\ 3 & 2 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 11 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 8 \end{pmatrix} \begin{pmatrix} 1 & -2 \\ -1 & 3 \end{pmatrix} = U \Sigma V,$$

where $U = U^{-1}$. From this we see that $\mathbb{Z}^2/H = \mathbb{Z}_1 \oplus \mathbb{Z}_8 \simeq \mathbb{Z}_8$, and $\text{coker}(A)$ is just the last row of U^{-1} ,

$$\text{coker}(A) = \begin{pmatrix} 11 & 1 \end{pmatrix} \in \text{epi}(\mathbb{Z}^2, \mathbb{Z}_8).$$

3.2.3 General Homomorphisms

We have seen that homomorphisms between the free finitely generated abelian groups are integer matrices. How can we represent and compute homomorphisms between general FGAs?

Lemma 1 Any FGA G of rank m is given as the image of a projection of the free group \mathbb{Z}^m onto G , $\pi_G \in \text{epi}(\mathbb{Z}^m, G)$.

Proof For $G = \mathbb{Z}_{\mathbf{k}} \oplus \mathbb{Z}^d$ let $m = k + d$, $k = |\mathbf{k}|$ and set $\pi_G(z) = z \bmod \mathbf{k}$ in the first k components and $\pi_G(z) = z$ in the last d components. Since any FGA is isomorphic to such a G we can produce a projection on any FGA by composing π_G with an isomorphism. \square

We call the above defined π_G the *canonical projection*. In many situations it is useful to represent G by an other projection, e.g. we can more generally choose some $A \in \mathbb{Z}^{m \times n}$ and let $\pi_G = \text{coker}(A)$. In this case, if $A = U\Sigma V$ we have $\pi_G(z) = U^{-1}z \bmod \mathbf{k}$, where \mathbf{k} is the diagonal of Σ .

Lemma 2 Let G and H be arbitrary FGAs and let $\pi_G \in \text{epi}(\mathbb{Z}^n, G)$ and $\pi_H \in \text{epi}(\mathbb{Z}^m, H)$ be projections onto these. A matrix $A \in \mathbb{Z}^{m \times n} \approx \text{hom}(\mathbb{Z}^n, \mathbb{Z}^m)$ represents a homomorphism $\phi = (\pi_H \circ A) / \pi_G \in \text{hom}(G, H)$ if and only if $\ker(\pi_G) < \ker(\pi_H \circ A)$. Any $\phi \in \text{hom}(G, H)$ can be written this way. The matrix A is generally not unique for a given ϕ .

$$\begin{array}{ccc} \mathbb{Z}^n & \xrightarrow{A} & \mathbb{Z}^m \\ \downarrow \pi_G & & \downarrow \pi_H \\ G & \xrightarrow{\phi} & H \end{array}$$

Proof First we start with a given $\phi \in \text{hom}(G, H)$. Since $\phi \circ \pi_G = \pi_H \circ A$, we have $A = \pi_H \setminus (\phi \circ \pi_G)$. Since π_H is onto H , this equation can always be solved, but the solution is not unique since we can add something in the $\ker(\pi_H)$ to A without affecting the solution. From the diagram it is easy to check that $\ker(\pi_G) < \ker(\pi_H \circ A)$.

Now, assume we are given an A such that $\ker(\pi_G) < \ker(\pi_H \circ A)$. This means that for any $y \in \ker(\pi_G)$ we have $(\pi_H \circ A)(x + y) = (\pi_H \circ A)(x)$ for all x . Hence $\pi_H \circ A$ takes constant values on each of the cosets of $\ker(\pi_G) < \mathbb{Z}^n$, and it defines a function on $G \simeq \mathbb{Z}^n / \ker(\pi_G)$. In other words, this is the necessary condition for solving the equation $\phi = (\pi_H \circ A) / \pi_G \in \text{hom}(G, H)$. \square

Exercise 5 Check that $A = 4$ defines a homomorphism $\phi \in \text{hom}(\mathbb{Z}_2, \mathbb{Z}_8)$. Find an other A representing the same homomorphism.

Definition 27 (Matrix Representation of a General Homomorphism) The notation $A \in \text{hom}(G, H)$ for some matrix $A \in \mathbb{Z}^{m \times n}$ means that A represents the homomorphism $(\pi_H \circ A) / \pi_G \in \text{hom}(G, H)$. Unless otherwise specified, the

projections π_G and π_H are the canonical projections (i.e. \mathbb{Z}^m and \mathbb{Z}^n mod the periods of G and H).

Even if we can represent any homomorphism in terms of an integer matrix, it does not mean that all computations are trivial. Some care must be taken! We illustrate by an example.

Example 15 Let $G = \mathbb{Z}_8 \oplus \mathbb{Z}_8$ and $H = \langle(1; 5)\rangle < G$, meaning that H is the subgroup of G generated by the element $(1; 5) \in G$. We want to compute G/H . Let $\pi \in \text{epi}(\mathbb{Z}^2, G)$ be the natural projection, $\pi(z) = z \bmod (8; 8)$ and $A = (1; 5) \in \text{hom}(\mathbb{Z}, \mathbb{Z}^2)$. We have $H = \text{im}(\phi)$ where $\phi = \pi \circ A$, and our task is to compute $\text{coker}(\phi) \in \text{epi}(G, G/H)$. Note that even if $\phi = \pi \circ A$, it is not so that the pre-image of H in \mathbb{Z}^2 is the image of A . The image of A is just the line of points $\langle(1; 5)\rangle < \mathbb{Z}^2$, while the pre-image of H contains all the points $j \cdot (1; 5) + y$ for all $y \in \ker(\pi)$. To find the pre-image of H we compute

$$\ker(\pi) = \begin{pmatrix} 8 & 0 \\ 0 & 8 \end{pmatrix}$$

and

$$\tilde{A} = A|_{\ker(\pi)} = \begin{pmatrix} 1 & 8 & 0 \\ 5 & 0 & 8 \end{pmatrix}.$$

Thus, the image of \tilde{A} is exactly the pre-image of H in \mathbb{Z}^2 . Smith decomposition yields

$$\tilde{A} = \begin{pmatrix} 1 & 0 \\ -3 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 8 & 0 \end{pmatrix} \begin{pmatrix} 1 & 8 & 0 \\ -1 & -3 & -1 \\ 0 & 1 & 0 \end{pmatrix},$$

from which we see that $\mathbb{Z}^2 / \text{im}(\tilde{A}) = \mathbb{Z}_1 \oplus \mathbb{Z}_8$ and

$$\text{coker}(\tilde{A}) = U^{-1} = \begin{pmatrix} 1 & 0 \\ -3 & -1 \end{pmatrix} \in \text{epi}(\mathbb{Z}^2, \mathbb{Z}_1 \oplus \mathbb{Z}_8).$$

Of course $\mathbb{Z}_1 \oplus \mathbb{Z}_8 \simeq \mathbb{Z}_8$ and we have $\text{coker}(\tilde{A}) = (-3 - 1) \in \text{epi}(\mathbb{Z}^2, \mathbb{Z}_8)$. Since $\ker(\pi) < \text{im}(\tilde{A})$ we have that $\text{coker}(\tilde{A})$ factors through π and we can compute

$$\text{coker}(\phi) = \text{coker}(\tilde{A})/\pi = (-3 - 1) \in \text{epi}(G, \mathbb{Z}_8).$$

We conclude that $G/H = \mathbb{Z}_8$ with this projection.

Example 16 (Computing the Cokernel of a General Homomorphism) The above example generalises to the general problem of computing G/H , where $H < G$ is a

subgroup generated by k elements of G . Let $\pi \in \text{epi}(\mathbb{Z}^n, G)$ be the natural projection and $A \in \text{hom}(\mathbb{Z}^k, \mathbb{Z}^n)$ such that H is the image of $\phi = \pi \circ A$, i.e. the columns of A represent the generators of H . We claim

$$\text{coker}(\phi) = \psi := \pi \setminus \text{coker}(A| \ker(\pi)). \quad (11)$$

To prove this, we must show that the bottom row of

$$\begin{array}{ccccc} & & \mathbb{Z}^n & & \\ & \nearrow A & \downarrow \pi & \searrow \text{coker}(A| \ker(\pi)) & \\ \mathbb{Z}^k & \xrightarrow{\phi} & G & \xrightarrow{\psi} & C \longrightarrow \mathbf{0} \end{array}$$

is exact. First we check that $\psi \circ \phi = 0$ by following the two top diagonal arrows (which by definition compose to zero). Next we see that ψ is an epimorphism (onto C), since $\text{coker}(A| \ker(\pi))$ by definition is onto. Last we pick an $z \in G$ such that $\psi(z) = 0$. This must mean that $z = \pi(y)$ for some $y \in \text{im}(A| \ker(\pi))$, hence $y = Ax + w$ for some $x \in \mathbb{Z}^k$ and $w \in \ker(\pi)$, from which it follows that $\phi(x) = z$. This proves that $\text{im}(\phi) = \ker(\psi)$ and the bottom line is exact. We conclude that $C = G / \text{im}(\phi) = G/H$ and that $\text{coker}(\phi) = \psi \in \text{epi}(G, G/H)$ is the projection.

3.2.4 Hermite's Normal Form

Smith's normal form is perfect for computing the structure of quotients. To compute images (and coimages) of maps into general FGAs, another normal form is sometimes more useful. Whereas Smith's normal form is the integer matrix version of SVD, the *Hermite normal form* is the integer version of LU factorisation. The basic idea is to factorise $A \in \mathbb{Z}^{m \times n}$ as $AV = H$, where $H \in \mathbb{Z}^{m \times k}$ is in *lower echelon form* and $V \in \text{GL}(n, \mathbb{Z})$. If the columns of A are generators of some subgroup then the columns of H constitute a set of generators for the same subgroup. The details of Hermite's normal form differ among different authors. There are row and column versions and some other details that can be done differently. Since we interpret group elements as column vectors we prefer a column version.

We say that an element $h_{i,j}$ in H is a *pivot* if $h_{i,j} \neq 0$ and everything above and to the right of $h_{i,j}$ is zero, i.e. $h_{k,\ell} = 0$ whenever $k \leq i$ and $\ell \geq j$ and $(k, \ell) \neq (i, j)$. The matrix H is in *lower echelon form* if every column i has a pivot $h_{p(i),i}$ and furthermore $p(i) < p(i+1)$ for every $i \in \{1, \dots, k-1\}$.

Definition 28 A matrix $H \in \mathbb{Z}^{m \times k}$ is in Hermite's normal form if

1. H is in lower echelon form.
2. Each pivot $h_{p(i),i} > 0$.
3. All elements to the left of a pivot are nonnegative and smaller than the pivot; for every $j \in \{1, \dots, i-1\}$ we have $0 \leq h_{p(i),j} < h_{p(i),i}$.

Lemma 3 For every non-zero $A \in \mathbb{Z}^{m \times n}$ there exists a $V \in GL(n, \mathbb{Z})$ partitioned as $V = (V_1|V_2)$, where $V_1 \in \mathbb{Z}^{n \times k}$, $V_2 \in \mathbb{Z}^{n \times (n-k)}$ such that

$$H = AV_1 \in \mathbb{Z}^{m \times k}$$

is in Hermite's normal form.

Proof We sketch an algorithm for computing this factorisation by applying elementary unimodular matrices acting on A from the right. A matrix of the form

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

in the upper left, and the identity in the lower right swaps the two first columns of A . More generally, any permutation matrix having exactly one 1 in each column and in each row, and zeros elsewhere permutes the columns and is always unimodular. A less trivial unimodular matrix is obtained as follows. Consider two positive integers r and s . From the Euclidean algorithm we compute integers a and b such that

$$ar + bs = g = \gcd(r, s).$$

Note that

$$(r \ s) \begin{pmatrix} a & -\frac{s}{g} \\ b & \frac{r}{g} \end{pmatrix} = (g \ 0),$$

where the matrix is unimodular. Thus, if $A_{11} = r$ and $A_{12} = s$ we can multiply A by such a matrix from the right to obtain $(g, 0)$ in the first two positions of row 1. We can continue to eliminate all entries to the right of A_{11} , possibly swapping columns if some entries in the first row are 0. We proceed the algorithm by searching for a new pivot in position 2 to m in row 2. If there are no pivots here we go to the next row etc. Whenever we have eliminated everything to the right of a pivot, we subtract multiples of the pivot column from all columns to the left of the pivot to fulfil criterion 3. in the definition of the Hermite normal form. \square

For $A \in \text{hom}(\mathbb{Z}^n, \mathbb{Z}^m)$ it is clear that the columns of H are independent and span the image of A , hence we can find both the kernel, image and coimage of A from the Hermite normal form decomposition:

$$\begin{aligned} \ker(A) &= V_2 \in \text{mono}(\mathbb{Z}^{n-k}, \mathbb{Z}^n) \\ \text{im}(A) &= H \in \text{mono}(\mathbb{Z}^k, \mathbb{Z}^m) \\ \text{coim}(A) &= V_1^{-1} \in \text{epi}(\mathbb{Z}^n, \mathbb{Z}^k), \end{aligned}$$

where V_1^{-1} denotes the upper $k \times n$ block of V^{-1} .

Example 17 The matrix A of Example 13 has Hermite factorisation

$$\begin{pmatrix} -20 & 8 & 16 \\ -6 & 0 & 6 \\ 0 & -12 & 6 \\ 4 & -16 & 4 \end{pmatrix} \begin{pmatrix} -1 & 0 & 2 \\ 0 & -2 & 1 \\ -1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 6 & 0 \\ -6 & 30 & 0 \\ -8 & 36 & 0 \end{pmatrix}$$

from which we find

$$A = \text{im}(A) \circ \text{coim}(A) = HV_1^{-1} = \begin{pmatrix} 4 & 0 \\ 0 & 6 \\ -6 & 30 \\ -8 & 36 \end{pmatrix} \begin{pmatrix} -5 & 2 & 4 \\ -1 & 0 & 1 \end{pmatrix}$$

and $\ker(A) = V_2 = (2; 1; 2)$.

If the matrix A represents a homomorphism in $\text{hom}(\mathbb{Z}^n, G)$ for an arbitrary FGA G , things are less simple. The columns of H still span the image of A , but they need not be independent generators, in which case H is not a monomorphism, so in order to compute kernels and images of general homomorphisms, we must be a bit more sophisticated.

Example 18 (Computing the Image/Coimage of a General Homomorphism) Given a homomorphism in terms of m generators, $A \in \text{hom}(\mathbb{Z}^m, G)$, we want to compute $\text{im}(A) \in \text{mono}(\mathbb{Z}_{p_1} \oplus \mathbb{Z}_{p_2} \oplus \cdots \oplus \mathbb{Z}_{p_k}, G)$ such that

$$\begin{array}{ccc} \mathbb{Z}^m & \xrightarrow{A} & G \\ \text{coim}(A) \downarrow & \swarrow \text{im}(A) & \\ \mathbb{Z}_{p_1} \oplus \cdots \oplus \mathbb{Z}_{p_k} & & . \end{array}$$

The image splits in components $\text{im}(A) = \alpha_1 | \alpha_2 | \cdots | \alpha_k$, where the components $\alpha_i \in \text{mono}(\mathbb{Z}_{p_i}, G)$ are the generators. The numbers p_i are called the *order* of the generator, i.e. the smallest positive integer such that $p_i \alpha_i = 0$. The idea of the algorithm is to compute the generators α_i by recursion in the rank m of A . First we show that we can compute one generator. Then we show that if one generator is known, we can reduce the computation to finding the image of a rank $m-1$ map.

- **Computing one generator.** Start by eliminating the first row of A as in the Hermite normal form algorithm, obtaining

$$AV = (a_1 | \bar{A}),$$

where a_1 is the first column with a non-zero top element and $\bar{A} \in \text{hom}(\mathbb{Z}^{m-1}, G)$ are the remaining columns eliminated to 0 on the top row. Clearly, a_1 is

independent from \bar{A} , so it must be a generator. We compute p_1 , the order of this generator, and find

$$\alpha_1 \in \text{mono}(\mathbb{Z}_{p_1}, G),$$

where a_1 and α_1 are represented by the same column vector.

- **Computing the rest by recursion in the rank.** The following diagram is of help in explaining the recursion step.

$$\begin{array}{ccccc} \mathbb{Z}^m & \xrightarrow{A} & G & \xrightarrow{\pi_1} & G/\alpha_1 \\ \downarrow V^{-1} & \nearrow \alpha_1|\bar{A} & \downarrow \text{im}(A) & \nearrow \psi & \\ \mathbb{Z}_{p_1} \oplus \mathbb{Z}^{m-1} & \xrightarrow{\phi} & \mathbb{Z}_{p_1} \oplus \mathbb{Z}_{p_2} \oplus \cdots \oplus \mathbb{Z}_{p_k} & & , \end{array}$$

where $\pi_1 = \text{coker}(\alpha_1)$ and ψ is defined as

$$\psi = \pi_1 \circ \text{im } A = 0 | (\pi_1 \circ \alpha_2) | (\pi_1 \circ \alpha_3) | \cdots | (\pi_1 \circ \alpha_k),$$

which is a monomorphism. Since ψ is mono, and ϕ is epi, we must have

$$\psi = \text{im}(\pi_1 \circ (\alpha_1|\bar{A})) = \text{im}(0|\pi_1 \circ \bar{A}) = 0 | (\text{im}(\pi_1 \circ \bar{A})),$$

hence

$$\text{im}(\pi_1 \circ \bar{A}) = (\pi_1 \circ \alpha_2) | (\pi_1 \circ \alpha_3) | \cdots | (\pi_1 \circ \alpha_k). \quad (12)$$

Since $\pi_1 \circ \bar{A}$ is of lower rank, we get by recursion the image of this

$$\text{im}(\pi_1 \circ \bar{A}) = \alpha'_2 | \alpha'_3 | \cdots | \alpha'_k,$$

and from (18) we obtain

$$\alpha_i = \pi_1 \setminus \alpha'_i, \quad 2 \leq i \leq k,$$

from which we get the answer

$$\text{im}(A) = \alpha_1 | \alpha_2 | \cdots | \alpha_k.$$

It will often happen that a column of A at some stage of the recursion becomes **0**. In this case the column is swapped out to the right, and V^{-1} is replaced by the upper $(m-1) \times m$ block of V^{-1} .

- **Computing $\text{coim}(A)$.** We have

$$\phi = \text{coim}(\pi_1 \circ (\alpha_1|\bar{A}))$$

Example 19 Let $G = \mathbb{Z}_4 \oplus \mathbb{Z}_{12}$. The matrix

$$A = \begin{pmatrix} 2 & 0 \\ 4 & 8 \end{pmatrix} \in \text{hom}(\mathbb{Z}^2, \mathbb{Z}_4 \oplus \mathbb{Z}_{12})$$

is in Hermite normal form, but the columns are not independent, since

$$A \begin{pmatrix} 2 \\ -1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \in G.$$

Following the above procedure to compute $\text{im}(A)$ we find

$$\alpha_1 = (2; 4) \in \text{mono}(\mathbb{Z}_6, \mathbb{Z}_4 \oplus \mathbb{Z}_{12}),$$

and by the technique of Example 16 we compute

$$\pi_1 = \text{coker}(\alpha_1) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \in \text{epi}(\mathbb{Z}_4 \oplus \mathbb{Z}_{12}, \mathbb{Z}_2 \oplus \mathbb{Z}_4).$$

Since $\pi_1(0; 8) = 0$, we are done (the projection discovered dependence between the generators), and we obtain

$$\text{im}(A) = \alpha_1 = (2; 4) \in \text{mono}(\mathbb{Z}_6, \mathbb{Z}_4 \oplus \mathbb{Z}_{12}).$$

Example 20 Compute $\text{im}(A)$ for

$$A = \begin{pmatrix} 1 & 1 \\ 3 & 5 \end{pmatrix} \in \text{hom}(\mathbb{Z}^2, \mathbb{Z}_4 \oplus \mathbb{Z}_8).$$

- We eliminate first row

$$\begin{pmatrix} 1 & 1 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 3 & 2 \end{pmatrix},$$

yielding

$$\alpha_1 = (1; 3) \in \text{mono}(\mathbb{Z}_8, \mathbb{Z}_4 \oplus \mathbb{Z}_8).$$

- From the Smith normal form decomposition

$$\begin{pmatrix} 1 & 4 & 0 \\ 3 & 0 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \end{pmatrix} \begin{pmatrix} 1 & -1 & 0 \\ 4 & -1 & -1 \\ 0 & -2 & 1 \end{pmatrix}$$

we find $\pi_1 = (-1, -1) \in \text{hom}(\mathbb{Z}_4 \oplus \mathbb{Z}_8, \mathbb{Z}_4)$ and $\bar{A} = \pi_1(0; 2) = 2 \in \text{hom}(\mathbb{Z}, \mathbb{Z}_4)$ and $\text{im}(\bar{A}) = 2 \in \text{mono}(\mathbb{Z}_2, \mathbb{Z}_4)$. Solving $\alpha_2 = \pi_1 \setminus \text{im}(\bar{A})$ yields $\alpha_2 = (2; 4) \in \text{mono}(\mathbb{Z}_2, \mathbb{Z}_4 \oplus \mathbb{Z}_8)$.

- We assemble and find

$$\text{im}(A) = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \in \text{hom}(\mathbb{Z}_8 \oplus \mathbb{Z}_2, \mathbb{Z}_4 \oplus \mathbb{Z}_8).$$

3.2.5 Summary

In this section we have sketched the outline of a software system for doing general computations in the category of finitely generated abelian groups. We have introduced the main operations in such a package and indicated the algorithms behind the construction.

3.3 Circulant Matrices and the Discrete Fourier Transform

To pave the road for later developments, we will start with a discussion of linear operators which are invariant under discrete circular shifts, and generalisations to finite abelian groups. This example has many of the properties of the general theory, but is simpler, since the spaces involved are finite dimensional vector spaces, and there are no questions of convergence. The classical notion of a *circulant matrix* is an $n \times n$ matrix

$$A = \begin{pmatrix} a_0 & a_{n-1} & \cdots & a_2 & a_1 \\ a_1 & a_0 & a_{n-1} & & a_2 \\ \vdots & a_1 & a_0 & a_{n-1} & \\ & & \ddots & \ddots & \ddots \\ a_{n-1} & & & a_1 & a_0 \end{pmatrix},$$

where the ‘wrap-around’ diagonals are constant, $A_{i,j} = a_{i-j \bmod n}$. The special circulant

$$S = \begin{pmatrix} & & & 1 \\ 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix},$$

where $a_1 = 1$ and $a_i = 0$ for $i \neq 1$ is called the *unit shift operator*. Let $\mathbb{C}[\mathbb{Z}_n]$ denote the vector space of all complex valued functions on \mathbb{Z}_n . The shift matrix S can be defined by its action on $\mathbb{C}[\mathbb{Z}_n]$

$$S\mathbf{x}(j) = \mathbf{x}(j-1) \quad \text{for all } j \in \mathbb{Z}_n.$$

Lemma 4 *For a matrix $A \in \mathbb{C}^{n \times n}$ the following are equivalent*

1. *A is circulant.*
2. *A is a polynomial in the shift operator*

$$A = \sum_{j \in \mathbb{Z}_n} a_j S^j.$$

3. *A acts on a vector $\mathbf{x} \in \mathbb{C}[\mathbb{Z}_n]$ through the convolution product $A\mathbf{x} = \mathbf{a} * \mathbf{x}$, defined as*

$$(\mathbf{a} * \mathbf{x})(j) := \sum_{\ell \in \mathbb{Z}_n} \mathbf{a}(\ell) \mathbf{x}(j - \ell).$$

4. *A is a linear translation invariant (LTI) operator on $\mathbb{C}[\mathbb{Z}_n]$, i.e. $AS = SA$.*
5. *The eigenvectors of A are $\{\chi_k\}_{k \in \mathbb{Z}_n}$ given as*

$$\chi_k(j) = e^{2\pi i j k / n}.$$

The reader is encouraged to prove this result for this case of classical circulant matrices (over the cyclic group \mathbb{Z}_n). We generalise to the case of a general finite abelian group.

Definition 29 (Group Ring) Let G be a finite abelian group. The *group ring* $\mathbb{C}[G]$ is the vector space of all complex valued functions $a: G \rightarrow \mathbb{C}$.

Alternatively (since G is finite), we can identify the group ring with the \mathbb{C} -linear combinations of elements of G ,

$$\mathbb{C}[G] = \left\{ \sum_{j \in G} a(j) j \right\}.$$

The structure of the domain G being a group yields important additional structure of $\mathbb{C}[G]$. For any $t \in G$ we define the shift operator $S_t: \mathbb{C}[G] \rightarrow \mathbb{C}[G]$

$$(S_t a)(j) := a(j - t). \tag{13}$$

It is easy to check that the shifts define an *action* of G on $\mathbb{C}[G]$, i.e. we have $S_t S_u = S_{t+u}$, and furthermore this action is by linear transformations on a vector space. Such linear actions are called *group representations* and are fundamental objects in Fourier analysis (both on commutative and non-commutative groups).

Since G forms a basis for $\mathbb{C}[G]$, we can extend the product on G by linearity to a product $*: \mathbb{C}[G] \times \mathbb{C}[G] \rightarrow \mathbb{C}[G]$ called the *convolution*, given as

$$(a * b)(j) := \sum_{\ell \in G} a(\ell)b(j - \ell) = \sum_{\ell \in G} a(j - \ell)b(\ell). \quad (14)$$

The convolution product is associative and commutative, and the *delta-function* $\delta \in \mathbb{C}[G]$, defined such that

$$\delta(j) = \begin{cases} 1 & \text{if } j = 0 \\ 0 & \text{otherwise} \end{cases}$$

is the unit of the convolution product, satisfying $\mathbf{a} * \delta = \delta * \mathbf{a} = \mathbf{a}$ for all $a \in \mathbb{C}[G]$. Convolutions and translation invariant operators go hand-in-hand.

Definition 30 (Linear Translation Invariant Operator (LTI)) A mapping $A: \mathbb{C}[G] \rightarrow \mathbb{C}[G]$ is called LTI if it is linear and commutes with shifts

$$AS_t = S_tA \quad \text{for all } t \in G.$$

For a vector space V let $\text{End}(V)$ denote the linear mappings $A: V \rightarrow V$ (endomorphisms). The elements of G form the natural basis for $\mathbb{C}[G]$, and with respect to this basis any $A \in \text{End}(\mathbb{C}[G])$ is represented by a matrix $A_{i,j}$ for indices $i, j \in G$, such that $(Ax)(i) = \sum_j A_{i,j}x(j)$. From this it is straightforward to verify the following result:

Lemma 5 $A \in \text{End}(\mathbb{C}[G])$ is LTI if and only if

$$A_{i,j} = A_{i-t,j-t}$$

for all $i, j, t \in G$.

We can reconstruct an LTI A from its first column. Let $\mathbf{a} = A\delta \in \mathbb{C}[G]$, in coordinates $\mathbf{a}(i) = A_{i,0}$, then

$$A_{i,j} = \mathbf{a}(i - j).$$

We see that in the case $G = \mathbb{Z}_n$, the LTI operators are exactly the circulant matrices. Writing the matrix-vector product in terms of \mathbf{a} we find

$$(Ax)(i) = \sum_{\ell \in G} A_{i,\ell}x(\ell) = \sum_{\ell \in G} \mathbf{a}(i - \ell)x(\ell) = \mathbf{a} * \mathbf{x}.$$

Conversely, any linear operator defined in terms of a convolution must be LTI. Hence, we conclude

Lemma 6 *For a finite abelian group G a matrix $A \in \text{End}(\mathbb{C}[G])$ is LTI if and only if it is given as a convolution*

$$A\mathbf{x} = \mathbf{a} * \mathbf{x}.$$

We want to understand the eigenvectors of convolutional operators. Recall that \mathbb{T} is the multiplicative group of complex numbers on the unit circle.

Lemma 7 *The eigenvectors of a convolutional operator $A\mathbf{x} = \mathbf{a} * \mathbf{x}$ are exactly the non-zero homomorphisms $\chi \in \text{hom}(G, \mathbb{T})$, i.e. the $\chi \in \mathbb{C}[G]$ such that*

$$\chi(j+k) = \chi(j)\chi(k) \quad \text{for all } j, k \in G.$$

For $\chi \in \text{hom}(G, \mathbb{T})$ we have $A\chi = \widehat{\mathbf{a}}(\chi) \cdot \chi$, where the eigenvalue $\widehat{\mathbf{a}}(\chi)$ is

$$\widehat{\mathbf{a}}(\chi) = \sum_{j \in G} \mathbf{a}(j) \overline{\chi(j)}.$$

Proof We start by picking a $\chi \in \text{hom}(G, \mathbb{T})$. Then, using $\chi(-j) = \overline{\chi(j)}$

$$(\mathbf{a} * \chi)(k) = \sum_{j \in G} \mathbf{a}(j) \chi(k-j) = \left(\sum_j \mathbf{a}(j) \chi(-j) \right) \chi(k) = \widehat{\mathbf{a}}(\chi) \chi(k).$$

When we compute $\text{hom}(G, \mathbb{T})$ explicitly, we will see that $\text{hom}(G, \mathbb{T}) \simeq G$. Since $\dim(\mathbb{C}[G]) = |G|$, this is a complete set of eigenvectors. \square

The mapping $\mathbf{a} \mapsto \widehat{\mathbf{a}}$ is called the *discrete Fourier transform* (DFT), and can be understood as an expansion in the orthogonal basis for $\mathbb{C}[G]$ given by the eigenvectors $\text{hom}(G, \mathbb{T})$, henceforth called the *characters* of G . Let $\langle \cdot, \cdot \rangle: \mathbb{C}[G] \times \mathbb{C}[G] \rightarrow \mathbb{C}$ denote the inner product on $\mathbb{C}[G]$

$$\langle f, g \rangle := \sum_{\ell \in G} \overline{f(\ell)} g(\ell).$$

Theorem 5 (Discrete Fourier Transform (DFT)) *Let $G = \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \cdots \oplus \mathbb{Z}_{n_d}$ be a FAG. The characters $\text{hom}(G, \mathbb{T})$ are in 1–1 correspondence with G ; for every $k \in G$ there is a unique character $\chi_k \in \text{hom}(G, \mathbb{T})$ given at $j \in G$ as*

$$\chi_k(j) = \exp \left(2\pi i \left(\frac{k_1 j_1}{n_1} + \frac{k_2 j_2}{n_2} + \cdots + \frac{k_d j_d}{n_d} \right) \right).$$

The characters are orthogonal

$$\langle \chi_k, \chi_{k'} \rangle = \begin{cases} |G| & \text{if } k = k' \\ 0 & \text{otherwise} \end{cases}.$$

The discrete Fourier transform $\widehat{\cdot}: \mathbb{C}[G] \rightarrow \mathbb{C}[G]$ and its inverse are given as

$$\widehat{\mathbf{a}}(k) = \sum_{j \in G} \mathbf{a}(j) \overline{\chi_k(j)} = \langle \chi_k, \mathbf{a} \rangle \quad (15)$$

$$\mathbf{a}(j) = \sum_{k \in G} \widehat{\mathbf{a}}(k) \chi_k(j). \quad (16)$$

Proof We first compute the characters on the group \mathbb{Z}_n . For any character χ we have that $\chi(0) = 1$. Now, since $n \cdot 1 = 0$ in G , we find that $\chi(1)^n = \chi(0) = 1$, thus $\chi(1) = \exp(2\pi i k/n)$ for some $k \in \{0, 1, \dots, n-1\}$. Let χ_k be the character with $\chi_k(1) = \exp(2\pi i k/n)$. Then $\chi_k(j) = \chi_k(1)^j = \exp(2\pi i j k/n)$. Thus the characters on \mathbb{Z}_n are given as

$$\chi_k(j) = \exp(2\pi i j k/n) \text{ for } k \in \mathbb{Z}_n. \quad (17)$$

By the formula for a geometric sum, it is straightforward to verify the orthogonality

$$\langle \chi_k, \chi_{k'} \rangle = \begin{cases} |G| & \text{if } k = k' \\ 0 & \text{otherwise} \end{cases}.$$

(Orthogonality of characters is proven for general LCAs in the next section.)

For $G = G_1 \oplus G_2$ we check that $\chi^1 \in \text{hom}(G_1, \mathbb{T})$ and $\chi^2 \in \text{hom}(G_2, \mathbb{T})$ produces a character $\chi = \chi^1 \oplus \chi^2 \in \text{hom}(G, \mathbb{T})$, and furthermore

$$\langle \chi^1 \oplus \chi^2, \widetilde{\chi}^1 \oplus \widetilde{\chi}^2 \rangle = \langle \chi^1, \widetilde{\chi}^1 \rangle \cdot \langle \chi^2, \widetilde{\chi}^2 \rangle.$$

From this the characters on $G = \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \dots \oplus \mathbb{Z}_{n_d}$ and their orthogonality relations follow, thus $\text{hom}(G, \mathbb{T})$ forms a complete orthogonal basis for $\mathbb{C}[G]$. The DFT and its inverse follow from the orthogonal expansion in $\mathbb{C}[G]$

$$\mathbf{a} = \sum_{\chi \in \text{hom}(G, \mathbb{T})} \frac{\langle \chi, \mathbf{a} \rangle}{\langle \chi, \chi \rangle} \chi.$$

□

The basic facts that **LTI** \Leftrightarrow **convolutions**, that Fourier transforms *diagonalise convolutions*,

$$(\widehat{\mathbf{a} * \mathbf{b}})(\chi) = \widehat{\mathbf{a}}(\chi) \cdot \widehat{\mathbf{b}}(\chi)$$

and that the DFT can be computed blazingly fast using the Fast Fourier Transform (FFT) explains why the FFT is one of the most important algorithms in computational mathematics. In the sequel we discuss Fourier analysis on more general LCAs. It is only for finite G that we have a direct fast algorithms for computing the Fourier transform. Hence, a detailed understanding of the relationship between the continuous and the discrete Fourier analysis is crucial for computational Fourier analysis. We will detail these relationships using the language of group homomorphisms introduced above.

3.4 Fourier Analysis on General LCAs

In this section we provide a quick survey of the general theory of Fourier analysis on general Locally Compact Abelian groups. The general theory has many of the properties of the finite case, but more care must be taken with respect to analytical properties of function spaces.

3.4.1 Functions on G

For finite G the group ring $\mathbb{C}[G]$ is a well defined space of all functions on G . For infinite G we have to be more careful, due to convergence issues. We will use the following notation:

- \mathbb{C}^G : **Complex valued functions on G .** This space is too large to be useful for mathematical analysis and we use this notation when we want to convey an idea without being very accurate on convergence issues. Read this as “*an appropriate space of functions on G* ”.
- $L^2(G)$: **Square integrable functions,**

$$L^2(G) = \left\{ f \in \mathbb{C}^G : \int_G |f(x)|^2 dx < \infty \right\},$$

where \int_G is defined below.

- $\mathcal{S}(G)$: **Schwartz functions.** Defined below, this space consists of rapidly decreasing, infinitely smooth functions.
- $\mathcal{S}'(G)$: **Tempered distributions.** Defined below, this is the dual space of $\mathcal{S}(G)$ and consists of generalised functions such as the Dirac δ function (point mass).

3.4.2 Shifts, Integrals and Convolutions

Let G be an LCA. Shifts of functions are defined as

$$(S_y f)(x) = f(x - y) \text{ for } x, y \in G \text{ and } f \in \mathbb{C}^G. \quad (18)$$

In [24] it is shown that for any LCA there exists a non negative measure μ which is shift invariant, i.e.

$$\mu(E) = \mu(E + x) \geq 0$$

for all Borel sets E and all $x \in G$, and $\mu(E) > 0$ for some E . This is called the Haar measure, and is unique up to a scaling. From this we obtain a shift invariant integral on G which we will just write as $\int_G \cdot dx$. For any integrable function f it satisfies:

$$\int_G f(x) dx = \int_G S_y f(x) dx \text{ for all } y \in G. \quad (19)$$

Example 21 For the LCAs of Definition 22 the invariant integrals are:

$$\mathbb{R} : \int_{\mathbb{R}} f(x) dx = \int_{-\infty}^{\infty} f(x) dx \text{ (the standard integral)}$$

$$T : \int_T f(x) dx = \int_0^1 f(x) dx \text{ (the standard integral)}$$

$$\mathbb{Z} : \int_{\mathbb{Z}} f(x) dx = \sum_{j=-\infty}^{\infty} f(j)$$

$$\mathbb{Z}_n : \int_{\mathbb{Z}_n} f(x) dx = \sum_{j=0}^{n-1} f(j)$$

For direct products $G = G_1 \times G_2$ it is obtained as a multiple integral

$$\int_G f(x) dx = \int_{G_1} \int_{G_2} f(x, y) dx dy.$$

For any finitely generated group G , the integral notation means the discrete sum

$$\int_G f(x) dx = \sum_{j \in G} f(j).$$

From the integral we get two important products on \mathbb{C}^G , the inner product and the convolution. The inner product $\langle \cdot, \cdot \rangle : \mathbb{C}^G \times \mathbb{C}^G \rightarrow \mathbb{C}$ is defined as

$$\langle f, g \rangle = \int_G \overline{f(x)} g(x) dx \quad (20)$$

where \overline{f} denotes the complex conjugate. We will sometimes write $\langle f, g \rangle_{\mathbb{C}^G}$ to emphasize on which domain we consider the innerproduct.

The convolution product $* : \mathbb{C}^G \times \mathbb{C}^G \rightarrow \mathbb{C}^G$ is defined as

$$(f * g)(y) = \int_G f(x) g(y - x) dx. \quad (21)$$

Note that the convolution can be understood as a weighted linear combination of shifts. This is evident in the finite case, $(f * g)(y) = \sum_{x \in G} f(x)g(y - x)$ thus $f * g = \sum_{x \in G} f(x)S_x g$. The various shifts $S_x g$ are multiplied with the weights $f(x)$. By a change of variables we verify that $f * g = g * f$, so one may also think of g as being the weights and f the function that is shifted.

Convolutions are ubiquitous in computational mathematics, common examples being finite difference approximations and linear digital filters. As a rule of thumb, convolutions are important whenever a problem is invariant under shifts. To be more precise, we say that a *linear* operator $A : \mathbb{C}^G \rightarrow \mathbb{C}^G$ is *translation invariant* (LTI) if $AS_g = S_g A$ for all $g \in G$. Thus linear differential operators with constant coefficients such as d/dx and ∇^2 are examples of LTI operators on $\mathbb{C}^{\mathbb{R}^n}$.

In the case of finite G , we saw that LTI operators are the same as convolutions. This is generally not the case for infinite G . E.g. there exists no (classical) function $f \in \mathbb{C}^{\mathbb{R}}$ such that $f * g = dg/dx$ for all differentiable g , and there is no (classical) function being the identity of convolution, $\delta * g = g$. However, there are various ways of *approximating* LTI operators on $\mathbb{C}^{\mathbb{R}}$ by convolutions, and the convolutional identity exists as a distribution $\delta \in \mathcal{S}'(G)$. So, we think of LTI and convolutional operators as being *essentially the same*, also for infinite G .

3.4.3 The Dual Group

Since all shifts commute, they share a common set of eigenfunctions. Convolutions are linear combinations of shifts, and do hence also share the same eigenfunctions. These are called the *characters* of the group. We will see that the characters form an orthogonal basis for $L^2(G)$, the square integrable functions on G . The Fourier transform is an expansion of functions in this basis. In the Fourier basis all convolutions become diagonal matrices. This diagonalizing property is the most important property of the Fourier transform. We will in this section see that the space of all Fourier coefficients also has the structure of an abelian group. It is called the dual group.

As above, let \mathbb{T} denote the unitary complex numbers

$$\mathbb{T} = \{ z \in \mathbb{C} \mid |z| = 1 \}. \quad (22)$$

As a multiplicative abelian group \mathbb{T} is isomorphic with T , via the map $T \ni x \mapsto \exp(2\pi i x) \in \mathbb{T}$.

Definition 31 (Group Character) A character on a group G is a (continuous⁵) homomorphism $\chi \in \text{hom}(G, \mathbb{T})$ i.e.

$$\chi(x + y) = \chi(x)\chi(y) \text{ for all } x, y \in G. \quad (23)$$

Note that $(S_y\chi)(x) = \chi(x - y) = \chi(-y)\chi(x)$, which shows that the characters are eigenfunctions of shifts. In Theorem 5 we have found that for G finite, the characters are in 1–1 correspondence with G itself.

Example 22 We want to find the characters on \mathbb{R} . We have $\chi(x + t) = \chi(t)\chi(x)$, differentiation with respect to t at $t = 0$ yields

$$\chi'(x) = \chi'(0)\chi(x).$$

Since $|\chi(x)| = 1$ we must have $\chi'(0) = i\omega$ for some $\omega \in \mathbb{R}$. Combined with $\chi(0) = 1$ this yields the complete family of continuous characters

$$\chi_\omega(x) = \exp(i\omega x) \text{ for } \omega \in \mathbb{R}. \quad (24)$$

To complete the argument we have to show that every continuous $\chi(x)$ is differentiable. We can always choose a small $\delta > 0$ such that $\int_0^\delta \chi(t)dt = \alpha > 0$. Thus

$$\alpha \cdot \chi(x) = \chi(x) \int_0^\delta \chi(t)dt = \int_0^\delta \chi(x + t)dt = \int_x^{x+\delta} \chi(t)dt.$$

The right hand side is the integral of a continuous function, and is thus differentiable. Hence $\chi(x)$ is differentiable.

Example 23 Let us compute the continuous characters on the unit circle $T = \mathbb{R}/\mathbb{Z}$. Let $x \in T$ be an irrational number, thus the sequence $x, 2x, 3x, \dots$ fills a dense subset of T . Once we have fixed the value of a character at x , we can derive the value of the character on this dense subset, $\chi(jx) = \chi(x)^j$. If we require $\chi(x)$ to be continuous, we can extend it uniquely to the whole of T . We leave it to the reader to verify that the resulting continuous characters on T are given as

$$\chi_k(x) = \exp(2\pi ikx) \text{ for } k \in \mathbb{Z}. \quad (25)$$

⁵For topological groups $\text{hom}(G, H)$ denotes the continuous homomorphisms.

One may alternatively arrive at the same result using the technique of the previous example. Note that if we did not have the condition that the characters should be continuous functions, we would get an awful lot of them, since we could make a separate choice of χ on each coset of \mathbb{Q} in \mathbb{R} .

We define the product of two characters as

$$(\chi_k \cdot \chi_l)(x) = \chi_k(x) \cdot \chi_l(x). \quad (26)$$

The product is obviously commutative. A simple computation shows that $(\chi_k \cdot \chi_l)(x+y) = (\chi_k \cdot \chi_l)(x) \cdot (\chi_k \cdot \chi_l)(y)$, thus also the product is a character.

Definition 32 (Dual Group) Let G be an LCA. The dual group Γ is defined as the collection of all (continuous) characters on G with the product (26). Γ has a natural topology turning it into an LCA.⁶

A natural question to ask is *what is the dual of the dual group?* For a given $x \in G$ and $\chi \in \Gamma$, let $\psi_x(\chi) = \chi(x)$. Since

$$\psi_x(\chi_k \cdot \chi_l) = \chi_k(x) \chi_l(x) = \psi_x(\chi_k) \cdot \psi_x(\chi_l),$$

we see that ψ_x is a character on Γ . It is also easy to verify that $\psi_x \cdot \psi_y = \psi_{x+y}$, thus G can at least be identified with a subgroup of the group of characters on Γ . If topology is taken into the picture it can be shown that G and the dual of Γ are isomorphic as LCAs, see [24].

Theorem 6 (Pontryagin Duality) *The identification of $x \in G$ with the character $\psi_x(\chi) = \chi(x)$ is an isomorphism between G and the dual of Γ .*

Example 23 showed that the dual of T is isomorphic with \mathbb{Z} under the map $\mathbb{Z} \ni k \mapsto \chi_k(\cdot) = \exp(2\pi i k \cdot)$ and Theorem 6 implies that the dual of \mathbb{Z} is naturally isomorphic to T . In order to recover the characters on T and on \mathbb{Z} , we define the function $(\cdot, \cdot) : \mathbb{Z} \times T \rightarrow \mathbb{T}$ as

$$(k, x) \equiv \chi_k(x) = \exp(2\pi i k x).$$

If we fix k then (k, \cdot) gives us all the characters on T , and when x is fixed we get all the characters (\cdot, x) on \mathbb{Z} . Thus we may simply say that T and \mathbb{Z} are dual spaces, where the characters are recovered by *the dual pairing* (\cdot, \cdot) .

Definition 33 (Dual Pair) Two LCAs G and \widehat{G} are called a *dual pair of LCAs* if there exists a continuous function $(\cdot, \cdot) : \widehat{G} \times G \rightarrow \mathbb{T}$ such that the map

$$\widehat{G} \ni k \mapsto (k, \cdot) \in \mathbb{C}^G$$

⁶It is given the weakest topology such that for any $x \in G$, the map $k \mapsto \chi_k(x) : \Gamma \rightarrow \mathbb{T}$ is continuous in k , see [24].

is an LCA isomorphism between \widehat{G} and the dual of G , and the map

$$G \ni x \mapsto (\cdot, x) \in \mathbb{C}^{\widehat{G}}$$

is an LCA isomorphism between G and the dual of \widehat{G} .

In particular the reader is encouraged to verify the following identities:

$$(k + k', x) = (k, x) \cdot (k', x) \quad (27)$$

$$(k, x + x') = (k, x) \cdot (k, x') \quad (28)$$

$$(0, x) = (k, 0) = 1 \quad (29)$$

$$\overline{(k, x)} = (-k, x). \quad (30)$$

Furthermore, if $(k, x) = 1$ for all k then $x = 0$, and if $(k, x) = 1$ for all x then $k = 0$.

Since $T = \mathbb{R}/\mathbb{Z}$ is isomorphic to \mathbb{T} through the exponential mapping $x \mapsto \exp(2\pi i x)$, we will often present the dual pairing in its *bi-additive form* $\langle \cdot, \cdot \rangle : \widehat{G} \times G \rightarrow T$ such that

$$(k, x) = \exp(2\pi i \langle k, x \rangle).$$

This satisfies the following equations

$$\langle k + k', x \rangle = \langle k, x \rangle + \langle k', x \rangle \quad (31)$$

$$\langle k, x + x' \rangle = \langle k, x \rangle + \langle k, x' \rangle \quad (32)$$

$$\langle 0, x \rangle = \langle k, 0 \rangle = 0 \quad (33)$$

$$\langle k, x \rangle = 0 \quad \forall k \Leftrightarrow x = 0 \quad (34)$$

$$\langle k, x \rangle = 0 \quad \forall x \Leftrightarrow k = 0, \quad (35)$$

thus we can think of $\langle \cdot, \cdot \rangle$ as an abelian group version of a non-degenerate bilinear pairing between vector spaces.

3.4.4 The Fourier Transform

The main goal of this section is to study the expansion of functions $f \in \mathbb{C}^G$ in terms of characters (the Fourier basis),

$$f(x) = \int_{\widehat{G}} \widehat{f}(k)(k, x) dx = \int_{\widehat{G}} \widehat{f}(k) e^{2\pi i \langle k, x \rangle} dx \quad \text{for some } \widehat{f}(k) \in \mathbb{C}^{\widehat{G}}. \quad (36)$$

If G is finite, than *any* $f \in \mathbb{C}^G$ can be expanded in this basis, but this is not generally true for infinite G . Necessary and sufficient conditions for functions to be expressible in terms of Fourier series is discussed in many textbooks on Fourier analysis, see e.g. [11].

The following lemma shows that the Fourier basis is orthogonal.

Lemma 8 *The characters are orthogonal under the inner product defined in (20)*

$$\langle (k, \cdot), (l, \cdot) \rangle = \int_G \overline{(k, x)} \cdot (l, x) dx = 0 \text{ when } k \neq l. \quad (37)$$

Proof Assume $k \neq l$.

$$\int_G (-k, x) \cdot (l, x) dx = \int_G (l - k, x) dx = \int_G (m, x) dx$$

where $m \neq 0$. Pick a point $x_0 \in G$ such that $(m, x_0) \neq 1$. Using the invariance of the integral, we find

$$\int_G (m, x) dx = (m, x_0) \int_G (m, x - x_0) dx = (m, x_0) \int_G (m, x) dx dx.$$

Hence, $\int_G (m, x) dx = 0$. □

Definition 34 (Fourier Transform) The Fourier transform is a linear map $\widehat{} : \mathbb{C}^G \rightarrow \mathbb{C}^{\widehat{G}}$ given as

$$\widehat{f}(k) = \langle (k, \cdot), f(x) \rangle_G = \int_G (-k, x) f(x) dx. \quad (38)$$

We also use the alternative notation

$$\mathcal{F}_G[f] := \widehat{f} \quad (39)$$

to specify the domain G explicitly.

Inversion of the Fourier transform is simple due to orthogonality of the characters. If G is compact then \widehat{G} is discrete, Theorem 12, and the integral over \widehat{G} is given as a sum $\int_{\widehat{G}} g(k) dk = \sum_{k \in \widehat{G}} g(k)$.

Lemma 9 *If G is compact, then*

$$f(x) = \frac{1}{C} \int_G \widehat{f}(k)(k, x) dk = \frac{1}{C} \sum_{k \in \widehat{G}} \widehat{f}(k)(k, x) dk, \quad (40)$$

where $C = \int_G 1 dx$.

Proof Given f as in (36). Using the orthogonality of the characters we find

$$\begin{aligned}\langle (k, \cdot), f(\cdot) \rangle_G &= \int_{x \in G} (-k, x) \sum_{\ell \in \widehat{G}} g(\ell)(\ell, x) dx \\ &= \sum_{\ell \in \widehat{G}} g(\ell) \int_{x \in G} (-k, x)(\ell, x) dx = g(k) \int_G 1 dx,\end{aligned}$$

thus $g(k) = \frac{1}{C} \widehat{f}(k)$. \square

A similar result holds also in the general case, see [24] for a proof:

Theorem 7 (Fourier Reconstruction) *Given any LCA G there exists a constant C so that Fourier reconstruction is given as*

$$f(x) = \frac{1}{C} \int_G \widehat{f}(k)(k, x) dk. \quad (41)$$

As stated in the beginning of this section, the most fundamental property of the Fourier transform is the diagonalization of convolutions:

Theorem 8 (Convolution Theorem)

$$\widehat{(f * g)}(k) = \widehat{f}(k)\widehat{g}(k). \quad (42)$$

Proof

$$\begin{aligned}\widehat{(f * g)}(k) &= \int_G (f * g)(x)(-k, x) dx = \int_G \int_G f(x-y)g(y)(k, -x) dxdy \\ &= \int_G f(x-y)(k, -x+y) dx \int_G g(y)(k, -y) dy = \widehat{f}(k)\widehat{g}(k).\end{aligned} \quad \square$$

A very related result is the following, which states that a shift of a function $f \in \mathbb{C}^G$ corresponds to a multiplication of \widehat{f} by a character on \widehat{G} , while a multiplication of f by a character on G corresponds to a shift of \widehat{f} . The proof is a straight forward computation left as an exercise.

Theorem 9 (Shift Formulas) *Let $\chi_k = (k, \cdot)$ and $\chi_x = (\cdot, x)$ be characters on G and \widehat{G} . Let S_x and S_k be shifts on $F(G)$ and $F(\widehat{G})$ defined in (18). Then*

$$\widehat{S_x f} = \chi_{-x} \cdot \widehat{f} \quad (43)$$

$$S_k \widehat{f} = \widehat{\chi_k \cdot f}. \quad (44)$$

The final fundamental result of this section states that the Fourier transform $\widehat{\cdot} : \mathbb{C}^G \rightarrow \mathbb{C}^{\widehat{G}}$ preserves the inner product on the two spaces. It bears the name of Parseval or Plancherel, depending on whether or not f and g are equal.

Theorem 10 (Parseval–Plancherel) Let C be the constant of the Fourier inversion (41). Then

$$\int_G \bar{f}(x)g(x)dx = \frac{1}{C} \int_{\widehat{G}} \bar{\hat{f}}(k)\hat{g}(k)dk. \quad (45)$$

Proof

$$\begin{aligned} \int_G \bar{f}(x)g(x)dx &= \int_G \bar{f}(x) \frac{1}{C} \int_{\widehat{G}} \hat{g}(k)(k, x)dk dx \\ &= \frac{1}{C} \int_{\widehat{G}} \int_G \bar{f}(x)(k, x)dx \hat{g}(k)dk = \frac{1}{C} \int_{\widehat{G}} \bar{\hat{f}}(k)\hat{g}(k)dk. \end{aligned}$$

□

Example 24 (Fourier Analysis on the Classical Groups) The following table presents the group and dual groups, the dual pairing, the Fourier transform and reconstruction for the basic groups \mathbb{R} , T , \mathbb{Z} and \mathbb{Z}_n .

G	\widehat{G}	(\cdot, \cdot)	$\widehat{f}(\cdot)$	$f(\cdot)$
$x \in \mathbb{R}$	$\omega \in \mathbb{R}$	$e^{2\pi i \omega x}$	$\int_{-\infty}^{\infty} e^{-2\pi i \omega x} f(x)dx$	$\int_{-\infty}^{\infty} e^{2\pi i \omega x} \hat{f}(\omega)d\omega$
$x \in T$	$k \in \mathbb{Z}$	$e^{2\pi ikx}$	$\int_0^1 e^{-2\pi ikx} f(x)dx$	$\sum_{k=-\infty}^{\infty} e^{2\pi ikx} \hat{f}(k)$
$j \in \mathbb{Z}_n$	$k \in \mathbb{Z}_n$	$e^{\frac{2\pi i k j}{n}}$	$\sum_{j=0}^{n-1} e^{\frac{-2\pi i k j}{n}} f(j)$	$\frac{1}{n} \sum_{k=0}^{n-1} e^{\frac{2\pi i k j}{n}} \hat{f}(k)$

Multidimensional versions are given by the componentwise formulae:

$$\begin{aligned} x &= (x_1, x_2) \in G = G_1 \oplus G_2 \\ k &= (k_1, k_2) \in \widehat{G} = \widehat{G}_1 \oplus \widehat{G}_2 \\ (k, x) &= (k_1, x_1) \cdot (k_2, x_2) \\ \widehat{f}(k_1, k_2) &= \int_{G_1} \int_{G_2} (-k_1, x_1)(-k_2, x_2) f(x_1, x_2) dx_1 dx_2 \\ f(x_1, x_2) &= \frac{1}{C_1 C_2} \int_{\widehat{G}_1} \int_{\widehat{G}_2} (k_1, x_1)(k_2, x_2) \widehat{f}(k_1, k_2) dk_1 dk_2. \end{aligned}$$

This gives the explicit form of the multidimensional transforms

G	\widehat{G}	$\langle \cdot, \cdot \rangle$	$\widehat{f}(\cdot)$	$f(\cdot)$
$x \in \mathbb{R}^n$	$\omega \in \mathbb{R}^n$	$\sum_{\ell} x_{\ell} \omega_{\ell}$	$\int_{-\infty}^{\infty} e^{-2\pi i \langle \omega, x \rangle} f(x)dx$	$\int_{-\infty}^{\infty} e^{2\pi i \langle \omega, x \rangle} \widehat{f}(\omega)d\omega$
$x \in T^n$	$k \in \mathbb{Z}^n$	$\sum_{\ell} x_{\ell} k_{\ell}$	$\int_{T^n} e^{-2\pi i \langle k, x \rangle} f(x)dx$	$\sum_{k \in \mathbb{Z}^n} e^{2\pi i \langle k, x \rangle} \widehat{f}(k)$
$j \in \mathbb{Z}_{\mathbf{m}}$	$k \in \mathbb{Z}_{\mathbf{m}}$	$\sum_{\ell=1}^n \frac{j_{\ell} k_{\ell}}{m_{\ell}}$	$\sum_{j \in \mathbb{Z}_{\mathbf{m}}} e^{-2\pi i \langle k, j \rangle} f(j)$	$\frac{1}{M} \sum_{k \in \mathbb{Z}_{\mathbf{m}}} e^{2\pi i \langle k, j \rangle} \widehat{f}(k)$

where $\mathbb{Z}_{\mathbf{m}} = \mathbb{Z}_{m_1} \oplus \mathbb{Z}_{m_2} \oplus \cdots \oplus \mathbb{Z}_{m_n}$ and $M = \prod_{\ell=1}^n m_i$ is the number of grid points in $\mathbb{Z}_{\mathbf{m}}$.

3.4.5 Schwartz Space and Tempered Distributions

For a proper discussion of sampling we need to introduce Schwartz functions and tempered distributions. The set $\mathcal{S}(G)$ of Schwartz functions on the elementary groups are defined as follows:

- On a finite G the Schwartz functions are all functions in $\mathbb{C}[G]$.
- On \mathbb{Z}^n the Schwartz functions are the functions that decrease faster than polynomially towards infinity.
- On T^n , the Schwartz functions are $C^\infty(T)$, the smooth functions.
- On \mathbb{R}^n the Schwartz functions are those $f \in C^\infty(\mathbb{R})$ such that both $f(x)$ and $\widehat{f}(\xi)$ decrease fast (faster than polynomially) as $x, \xi \rightarrow \infty$.

On a general LCA G there is also a notion of such functions called *Schwartz–Bruhat functions*. These can be defined as the functions f on G , such that both f and \widehat{f} have rapidly decreasing L^∞ -norms [23]. Important for introducing these functions are the following properties:

- $\mathcal{S}(G)$ is closed under sums, products, translations and convolutions.
- $\mathcal{S}(G)$ is dense in $L^p(G)$ for all $1 \leq p \leq \infty$.
- The space of *bump functions* $C_c^\infty(G)$ (smooth functions with compact support) is dense in $\mathcal{S}(G)$.
- The Fourier transform is a linear isomorphism between $\mathcal{S}(G)$ and $\mathcal{S}(\widehat{G})$.
- If $\phi \in \text{mono}(H, G)$ and $f \in \mathcal{S}(G)$ then $f \circ \phi \in \mathcal{S}(H)$.

The *tempered distributions* $\mathcal{S}'(G)$ is the set of all linear functionals on $\mathcal{S}(G)$, linear mappings from $\mathcal{S}(G)$ to \mathbb{C} . For $T \in \mathcal{S}'(G)$ and $\phi \in \mathcal{S}(G)$ it is convenient to use the notation $\langle T, \phi \rangle$ for the evaluation of T at ϕ . Every measurable function $f: G \rightarrow \mathbb{C}$ growing slowly (not faster than polynomial) defines a distribution $T_f \in \mathcal{S}'(G)$ via the integral

$$\langle T_f, \phi \rangle := \int_{x \in G} f(x)\phi(x)dx \quad \text{for all } \phi \in \mathcal{S}(G).$$

These are called the *regular distributions*. There are, however, also other (singular) distributions which are not given by classical functions, such as the Dirac function $\delta(x)$, which physicists interpret as a unit mass in 0 such that $\int_{x \in G} \delta(x)\phi(x)dx = \phi(0)$. Such a function $\delta(x)$ is not a classical function, and the correct way to think of this is as the functional $\delta \in \mathcal{S}'(G)$ defined such that

$$\langle \delta, \phi \rangle := \phi(0) \quad \text{for all } \phi \in \mathcal{S}(G).$$

We can define Fourier transforms and derivatives of distributions by dualisation. The regular distributions give the hint on the correct definitions. For smooth slowly growing functions $f(x)$ on $G = \mathbb{R}$, integration by parts yields

$$\int f(x) \frac{d\phi(x)}{dx} dx = \int -\frac{df}{dx} \phi(x) dx.$$

For this reason, we *define* the derivative of $T \in \mathcal{S}'(\mathbb{R})$ as

$$\langle \frac{dT}{dx}, \phi(x) \rangle := \langle T, -\frac{d\phi(x)}{dx} \rangle. \quad (46)$$

Similarly, for a nice function $f \in \mathbb{C}^G$ we have, by Plancherel's theorem

$$\langle f, \check{\phi} \rangle_G = \frac{1}{C} \langle \widehat{f}, \phi \rangle_{\widehat{G}},$$

where $\check{\cdot}: \mathbb{C}^{\widehat{G}} \rightarrow \mathbb{C}^G$ denotes the inverse Fourier transform. Thus, we *define* the Fourier transform of a distribution $T \in \mathcal{S}'(G)$ as

$$\langle \widehat{T}, \phi \rangle := C \langle T, \check{\phi} \rangle, \quad (47)$$

where C is the constant from Plancherel's theorem.

Exercise 6 Check that this definition implies $\widehat{\delta} = 1$.

We refer to [11] for more details on tempered distributions.

3.4.6 Pullback and Pushforward of Functions on Groups

The central topic of our lectures are the relationship between functions defined on a group and related functions on a subgroup and the quotient.

Definition 35 (Pullback and Pushforward of Functions) For $\phi \in \text{hom}(H, G)$ we define pullback $\phi^*: \mathbb{C}^G \rightarrow \mathbb{C}^H$ and pushforward $\phi_*: \mathbb{C}^H \rightarrow \mathbb{C}^G$ as adjoint operators with respect to the inner products

$$\phi^*(f) := f \circ \phi \quad (48)$$

$$\langle \phi_*(g), f \rangle_{\mathbb{C}^G} := \langle g, \phi^*(f) \rangle_{\mathbb{C}^H} \quad (49)$$

for $f \in \mathbb{C}^G$ and $g \in \mathbb{C}^H$.

Exercise 7 Show that for $\phi \in \text{hom}(G_1, G_2)$ where G_1, G_2 are finite, we have

$$\begin{aligned}\phi_* f(\phi(j)) &= \sum_{k \in \ker(\phi)} f(j+k) \\ \phi_* f(\ell) &= 0 \quad \text{for } \ell \notin \text{im}(\phi).\end{aligned}$$

The pullback is easy to understand, e.g. if $\phi \in \text{mono}(H, G)$ defines a discrete lattice, then $\phi^* f$ is the sampling of f in the lattice points $\phi(H) \subset G$. The push forward $\phi_* f$ along $\phi \in \text{epi}(G, K)$ is summing up the values of f in all points of G mapping to the same point in K . It can be shown that for $\phi \in \text{epi}(G, K)$ there always exists a constant C (depending on the choice of Haar measures on G and K) such that

$$\phi_* f(\phi(x)) = C \int_{y \in \ker(\phi)} f(x+y) dy.$$

It follows that:

Lemma 10 For $\psi \in \text{mono}(H, G)$ pullback is well-defined for Schwartz functions, $\psi^*: \mathcal{S}(G) \rightarrow \mathcal{S}(H)$. For $\phi \in \text{epi}(G, K)$ pushforward is well-defined for Schwartz functions $\phi_*: \mathcal{S}(G) \rightarrow \mathcal{S}(K)$.

Example 25 Pullback along epimorphisms does not in general send Schwartz functions to Schwartz functions, for example take $\phi \in \text{epi}(\mathbb{R}, T)$ as $\phi(x) = x$. The constant function $f(x) = 1 \in \mathcal{S}(T)$, but $\phi^*(1) = 1 \notin \mathcal{S}(\mathbb{R})$. The result is, however, a distribution in $\mathcal{S}'(\mathbb{R})$. Similarly, pushforward along monomorphisms is in general well-defined for distributions and not for Schwartz functions.

Definition 36 (Pullback and Pushforward of Distributions) For $\psi \in \text{mono}(H, G)$ we define pushforward of distributions $\psi_*: \mathcal{S}'(H) \rightarrow \mathcal{S}'(G)$ as

$$\langle \psi_* T, f \rangle := \langle T, \psi^* f \rangle \quad \text{for all } f \in \mathcal{S}(G).$$

For $\phi \in \text{epi}(G, K)$ we define pullback of distributions $\phi^*: \mathcal{S}'(K) \rightarrow \mathcal{S}'(G)$ as

$$\langle \phi^* T, f \rangle := \langle T, \phi_* f \rangle \quad \text{for all } f \in \mathcal{S}(G).$$

Example 26 Let $\phi \in \text{mono}(\mathbf{0}, \mathbb{R})$ (the **0**-arrow). Since

$$\langle \phi_* 1, f \rangle = \langle 1, \phi^* f \rangle = f(0),$$

we have $\phi_* 1 = \delta$, the Dirac distribution on \mathbb{R} . There is nothing particular about \mathbb{R} here, indeed for any LCA G we have the following equivalent characterisations of the δ -distribution:

$$\delta = \mathbf{0}_* 1 \in \mathcal{S}'(G) \Leftrightarrow \langle \delta, f \rangle = f(0). \tag{50}$$

3.5 Duality of Subgroups and Quotients

3.5.1 Dual Homomorphisms

Recall the discussion above, for an LCA G , the dual group \widehat{G} is isomorphic to $\text{hom}(G, \mathbb{T})$, which contains all eigen functions of the shift operators S_i acting on \mathbb{C}^G , or as mappings to the additive group $T = \mathbb{R}/\mathbb{Z}$ we have $\widehat{G} \cong \text{hom}(G, T)$. Similar to the adjoint of a linear mapping, we define the adjoint of a LCA homomorphism

Definition 37 (Dual Homomorphism) Given $\phi \in \text{hom}(H, G)$ the dual homomorphism $\widehat{\phi} \in \text{hom}(\widehat{G}, \widehat{H})$ is defined for $\widehat{H} = \text{hom}(H, T)$ and $\widehat{G} = \text{hom}(G, T)$, acting on an element $\alpha \in \text{hom}(G, T)$ as

$$\widehat{\phi}(\alpha) = \alpha \circ \phi \in \text{hom}(H, T).$$

Equivalently, for dual pairs G, \widehat{G} and H, \widehat{H} with pairings $\langle \cdot, \cdot \rangle_G: \widehat{G} \times G \rightarrow T$ and $\langle \cdot, \cdot \rangle_H: \widehat{H} \times H \rightarrow T$, we define

$$\langle \widehat{\phi}(\alpha), h \rangle_H = \langle \alpha, \phi(h) \rangle_G,$$

for all $\alpha \in \widehat{G}$ and $h \in H$.

Example 27 Let $H = \mathbb{Z}^n$ and $\widehat{H} = T^n$ with pairings $\langle \xi, \mathbf{j} \rangle_{\mathbb{Z}^n} = \xi^T \mathbf{j} \bmod 1$ and $\langle \xi, \mathbf{x} \rangle_{\mathbb{R}^n} = \xi^T \mathbf{x} \bmod 1$. A non-singular matrix $A \in \mathbb{R}^{n \times n}$ defines a homomorphism $\phi \in \text{hom}(\mathbb{Z}^n, \mathbb{R}^n)$ as $\phi(\mathbf{j}) = A\mathbf{j}$. The dual homomorphism $\widehat{\phi} \in \text{hom}(\mathbb{R}^n, T^n)$ is given as $\widehat{\phi}(\xi) = A^T \xi \bmod 1$. If the columns of A are linearly independent then ϕ is a monomorphism and $\widehat{\phi}$ an epimorphism.

3.5.2 The Fundamental Duality Theorem

The main topic of this section is a theorem relating subgroup and quotient decompositions of a group to decompositions of the dual spaces. To prepare for this we discuss duality of sequences of homomorphisms in general. A *chain complex* is a sequence of groups G_i and homomorphisms $\phi_i \in \text{hom}(G_i, G_{i+1})$ such that $\phi_{i+1} \circ \phi_i = 0$ for all i . A *co-chain complex* is similarly defined, where the indices decrease rather than increase. Recall that the chain complex is *exact* if $\text{im}(\phi_i) = \ker(\phi_{i+1})$ for all i , and similarly for the co-chain. An equivalent way of defining exactness is to say that whenever $x \in G_{i+1}$ such that $\phi_{i+1}(x) = 0$, there exists an $y \in G_i$ such that $x = \phi_i(y)$.

Lemma 11 *If (ϕ_i, G_i) is a chain complex, then the dual $(\widehat{\phi}_i, \widehat{G}_i)$ is a co-chain complex, and if one of them is exact, then also the other is exact.*

Proof Let $\widehat{G}_i = \text{hom}(G_i, T)$. For $\alpha \in G_{i+2}$, we see $(\widehat{\phi}_i \circ \widehat{\phi}_{i+1})(\alpha) = \alpha \circ \phi_{i+1} \circ \phi_i = 0$, hence $(\widehat{\phi}_i, \widehat{G}_i)$ is a co-chain complex.

To prove the statement about exactness, assume (ϕ_i, G_i) exact. We pick a $\chi \in \widehat{G_{i+1}}$ such that $\phi_i^*(\chi) = \chi \circ \phi_i = 0$ and want to show that there exists a $\chi' \in G_{i+2}^*$ such that $\widehat{\phi_{i+1}}(\chi') = \chi$. Pick an $x \in G_{i+1}$ such that $\phi_{i+1}(x) = 0$. Exactness implies that $x = \phi_i(y)$ for some $y \in G_i$, hence $\chi(x) = (\chi \circ \phi_i)(y) = 0$. Thus $\chi \circ \ker(\phi_{i+1}) = 0$, and since it is zero on the kernel we can solve the equation $\chi' = \chi / \phi_{i+1}$ for $\chi' \in \widehat{G_{i+2}}$. This proves exactness of $(\widehat{\phi}_i, \widehat{G}_i)$.

If $(\widehat{\phi}_i, \widehat{G}_i)$ is exact then (ϕ_i, G_i) must be exact because of Pontryagin duality. \square

A very important consequence of this lemma is the following theorem, which is fundamental for the understanding of sampling theory and computational Fourier transforms:

Theorem 11 (Fundamental Duality Theorem of LCAs) *A short sequence*

$$\mathbf{0} \longrightarrow H \xrightarrow{\phi_1} G \xrightarrow{\phi_2} K \longrightarrow \mathbf{0} \quad (51)$$

is exact if and only if the dual sequence

$$\mathbf{0} \longleftarrow \widehat{H} \xleftarrow{\widehat{\phi}_1} \widehat{G} \xleftarrow{\widehat{\phi}_2} \widehat{K} \longleftarrow \mathbf{0} \quad (52)$$

is exact. Furthermore, $\phi_1(H) < G$ is a closed subgroup if and only if $\widehat{\phi}_2(\widehat{K}) < \widehat{G}$ is closed.

A proof of the final statement about closed subgroups is found in [24].

Corollary 1 *Let H be a closed subgroup of G and $K = G/H$. Then \widehat{K} is a closed subgroup of \widehat{G} and $\widehat{H} \approx \widehat{G}/\widehat{K}$.*

Corollary 2 *If $\phi \in \text{mono}(H, G)$ then $\widehat{\phi} \in \text{epi}(\widehat{G}, \widehat{H})$, and if $\phi \in \text{epi}(G, K)$ then $\widehat{\phi} \in \text{mono}(\widehat{K}, \widehat{G})$.*

Definition 38 (Annihilator Subgroup) For a closed subgroup $H < G$ the closed subgroup $\widehat{G}/\widehat{H} < \widehat{G}$ is called *the annihilator subgroup* of \widehat{G} , denoted

$$H^\perp := \widehat{G}/\widehat{H}.$$

The annihilator H^\perp consists exactly of exactly those characters in $\text{hom}(G, \mathbb{T}) \approx \widehat{G}$ (a.k.a. Fourier basis functions) which evaluate to 1 at all points $h \in H$:

Lemma 12 *Referring to diagrams (51)–(52) we have that*

$$(\xi, \phi_1(H))_G \equiv 1$$

if and only if $\xi \in \widehat{\phi}_2(\widehat{K})$.

Proof For $x = \phi_1(h)$ and $\xi = \widehat{\phi}_2(k)$ we get

$$(\xi, x)_G = (\widehat{\phi}_2(k), \phi_1(h))_G = (k, \phi_2 \circ \phi_1(h))_K = (k, 0)_K = 1.$$

On the other hand, if $(\xi, \phi_1(h))_G = 1$ for all $h \in H$, then $(\widehat{\phi}_1(\xi), h)_H = 0$, and hence $\widehat{\phi}_1(\xi) = 0$. Exactness implies the existence of a $k \in \widehat{K}$ such that $\widehat{\phi}_2(k) = \xi$. \square

In terms of the bi-additive pairing $\langle \cdot, \cdot \rangle_G$, we have

$$\phi_1(H)^\perp = \left\{ \xi \in \widehat{G} : \langle \xi, \phi_1(H) \rangle_G \equiv 0 \in T \right\},$$

so the annihilator is the abelian group version of orthogonal complement in linear algebra. We also have $(H^\perp)^\perp = H$.

3.6 Lattices and Sampling

An LCA K is called *compact* if

$$\text{vol}(K) := \int_K dx < \infty.$$

An LCA H is called *discrete* if every point in H (and every subset of H) are open sets. We say that H is *continuous* if it is not discrete. The following result is proven in [24]. We have not discussed enough topology to reproduce the proof.

Theorem 12 *An LCA G is compact if and only if the dual group \widehat{G} is discrete, and G is discrete if and only if the dual \widehat{G} is compact.*

Example 28

- $\mathbb{R} \leftrightarrow \widehat{\mathbb{R}} \approx \mathbb{R}$ (continuous, non-compact \leftrightarrow continuous, non-compact).
- $\mathbb{Z} \leftrightarrow \widehat{\mathbb{Z}} \approx T$ (discrete, non-compact \leftrightarrow compact, continuous).
- $\mathbb{Z}_n \leftrightarrow \widehat{\mathbb{Z}}_n \approx \mathbb{Z}_n$ (discrete, compact \leftrightarrow discrete, compact).

Perhaps it is worth noting that we could choose $G = \mathbb{R}$ with a discrete topology. In this case \widehat{G} is a compact space which is called the Bohr compactification of \mathbb{R} , after Harald Bohr, the brother of Niels Bohr, who studied the Fourier analysis of so-called *almost periodic functions*. A discussion of this topic is interesting, but brings us beyond the scope of these notes.

Definition 39 (Lattice) A lattice is a *discrete* and closed subgroup $H < G$ such that G/H is compact.

Recall that $H^\perp \approx \widehat{G/H}$ hence if H is a lattice, then the annihilator $H^\perp < \widehat{G}$ is also a lattice, called the *reciprocal lattice*. In the sequel we will study sampling theory as movements of functions between the domains in the diagram

$$\begin{array}{ccccccc} \mathbf{0} & \longleftarrow & \widehat{H} & \xleftarrow{\phi_1} & \widehat{G} & \xleftarrow{\phi_2} & H^\perp \longleftarrow \mathbf{0} \\ & & | & & | & & | \\ \mathbf{0} & \longrightarrow & H & \xrightarrow{\phi_1} & G & \xrightarrow{\phi_2} & G/H \longrightarrow \mathbf{0}, \end{array} \quad (53)$$

where the vertical lines indicate dual pairs of groups, $\phi_1(H) < G$ and $\phi_2(H^\perp) < \widehat{G}$ are the reciprocal lattices and both rows are exact. In particular we study the relationship between Fourier transforms on G and on H , and we will even see that there is a relationship between functions on the spaces \widehat{H} and G/H , which explains the Fast Fourier Transform. First, let us give a few concrete examples of this diagram.

Example 29 (Sound Sampling) The classical setting of sampling of sound is the case where $G = \mathbb{R}$, $H = \mathbb{Z}$, $\phi_1(j) = j \cdot h$, where h is the sampling interval. We can set $G/H = h$ with $\phi_2(t) = t/h \bmod 1$, and $H^\perp = \mathbb{Z}$ with $\widehat{\phi}_2(k) = k/h$ and $\widehat{H} = T$ with $\widehat{\phi}_1(\xi) = \xi \cdot h$. The pairings are $\langle \xi, t \rangle_G = \xi \cdot t$, $\langle \xi, j \rangle_H = \xi \cdot j$ and $\langle k, t \rangle_{G/H} = k \cdot t$.

Example 30 (Multidimensional Sampling of \mathbb{R}^n) Let $G = \mathbb{R}^n$ and $H = \mathbb{Z}^n$. A nonsingular matrix $A \in \mathbb{R}^{n \times n}$ defines $\phi_1(\mathbf{j}) = A\mathbf{j}$, where $G/H = T^n$ and $\phi_2(\mathbf{x}) = A^{-1}\mathbf{x}$. On the dual side we have $\widehat{H} = T^n$, $\widehat{G} = \mathbb{R}^n$ and $H^\perp = T^n$ with pairings $\langle \xi, \mathbf{x} \rangle_G = \xi^T \mathbf{x}$, $\langle \xi, \mathbf{j} \rangle_H = \xi^T \mathbf{j}$ and $\langle \mathbf{k}, \mathbf{x} \rangle_{G/H} = \mathbf{k}^T \mathbf{x}$. This yields the dual homomorphisms $\widehat{\phi}_1(\xi) = A^T \xi$ and $\widehat{\phi}_2(\mathbf{j}) = A^{-T} \mathbf{j}$. Note that a matrix of rank lower than n does not define a lattice in \mathbb{R}^n , since the quotient G/H in that case is non-compact.

Example 31 (Splitting for the FFT) Let $G = \mathbb{Z}_{mn}$ and $H = \mathbb{Z}_m$ with $\phi_1(j) = jn$. We have $G/H = \mathbb{Z}_n$ and $\phi_2(j) = j$. On the dual side we have $\widehat{H} = \mathbb{Z}_m$, $\widehat{G} = \mathbb{Z}_{mn}$, $H^\perp = \mathbb{Z}_n$ with pairings $\langle k, j \rangle_G = kj/mn$, $\langle k, j \rangle_H = kj/m$ and $\langle k, j \rangle_{G/H} = kj/n$. This yields the dual homomorphisms $\widehat{\phi}_1(k) = k$, since $\langle \widehat{\phi}_1(k), j \rangle_H = \langle k, j \rangle_H = kj/m = \langle k, nj \rangle_G = \langle k, \phi_1(j) \rangle_G$. Similarly we find $\widehat{\phi}_2(k) = km$.

3.6.1 Pullback and Pushforward on Lattices

For $\phi_1 \in \text{mono}(H, G)$, where H is discrete, we call the operation $\phi_1^* : \mathbb{C}^G \rightarrow \mathbb{C}^H$ (down) *sampling*, defined as $\phi_1^* f := f \circ \phi_1$. For $\phi_2 \in \text{epi}(G, K)$, where $\ker(\phi_2)$ is discrete, we call $\phi_2_* : \mathcal{S}(G) \rightarrow \mathcal{S}(K)$ *periodisation*, given as

$$\phi_2_* f(\phi_2(x)) = \sum_{k \in \ker(\phi_2)} f(x + k).$$

The name ‘periodisation’ reminds us that if we compute $g = \phi_2^* \circ \phi_{2*} f$, we obtain $g \in \mathcal{S}'(G)$ as a function periodic $g(x+k) = g(x)$ for all $k \in \ker(\phi_2)$. If we have a lattice $H < G$ and $K = G/H = \{g + H\}$, as the cosets, we have

$$\phi_{2*} f(g + H) = \sum_{h \in H} f(g + h),$$

which can be interpreted as a H -periodic function in \mathbb{C}^G .

Distributions can be moved in the opposite direction by ϕ_1 and ϕ_2 . We call $\phi_{1*}: \mathcal{S}'(H) \rightarrow \mathcal{S}'(G)$ *up sampling*. This is given as

$$\phi_{1*} f = \sum_{h \in H} f(h) \delta_{\phi_1(h)},$$

where $\delta_{\phi_1(h)} = \delta(x - \phi_1(h))$ is the shifted δ -distribution. Up sampling of a discrete function on a lattice yields a set of point masses in the lattice points. The operation $\phi_{2*}: \mathcal{S}'(K) \rightarrow \mathcal{S}'(G)$ yields an H -periodic distribution on G .

Many important dual relationships can be derived from the following result, which is proven in many texts, see e.g. [22]:

Theorem 13 (Poisson Summation Formula) *Let $\phi_1 \in \text{mono}(H, G)$ be a lattice with dual lattice $\widehat{\phi}_2 \in \text{mono}(H^\perp, G)$, as in (53). For $f \in \mathcal{S}(G)$ we have*

$$\sum_{h \in H} f(\phi_1(h)) = \frac{1}{C} \sum_{k \in H^\perp} \widehat{f}(\widehat{\phi}_2(k)),$$

where the constant $C = \text{vol}(G/\phi_1(H))$ is the volume of the unit-cell of the lattice (if G is discrete C is the number of points in the unit-cell).

3.6.2 Choosing Coset Representatives

For many computational problems it is necessary to choose representative elements from each of the cosets in the quotient groups G/H and $\widehat{H} = \widehat{G}/H^\perp$. E.g. in sampling theory on a lattice $H < G = \mathbb{R}^n$, all characters in a coset $H^\perp + \xi \subset \widehat{G}$ alias on H (i.e. they evaluate to the same on H), but physical relevance is usually given to the character $\xi' \in H^\perp + \xi$ which is closest to 0 (the lowest frequency mode). Similarly, we often represent $K = G/H$ by picking a representative from each coset (e.g. \mathbb{R}/\mathbb{Z} can be represented by $[0, 1) \subset \mathbb{R}$). The projection map $\phi \in \text{epi}(G, K)$ assigns each coset to a unique element in K , and we need to decide on a right inverse of this map.

Definition 40 (Transversal of Quotient $K = G/H$) Given a quotient projection $\phi \in \text{epi}(G, K)$, a function $\sigma: K \rightarrow G$ is called a *transversal* of ϕ if $\phi_1 \circ \sigma = \text{Id}_K$ (this is often also called a *section* of the projection).

Note that in general we cannot choose σ as a group homomorphism (only if $G = H \oplus K$), but it can be chosen as a continuous function. In many applications G has a natural norm (e.g. Euclidean distance on \mathbb{R}^n) and we can choose σ such that the coset representatives are as close to the origin as possible, i.e. such that $||\sigma(k)|| \leq ||\sigma(k) - h||$ for all $h \in H$.

Definition 41 (Voronoi Transversal) Let $G = \mathbb{R}^n$ or $G = \mathbb{T}^n$, and let $H < G$ be a lattice. The transversal $\sigma: G/H \rightarrow G$ such that $||\sigma(k)|| \leq ||\sigma(k) - h||$ for all $h \in H$ is called the *Voronoi transversal*. The image of the Voronoi transversal is a polyhedron around the origin in G , limited by hyperplanes orthogonal to the lines between the origin and the closest lattice points, and dividing these in the middle.

3.6.3 Sampling and Aliasing

Shannon's theory of sampling and reconstruction is a classical topic discussed in any textbook on signal processing, usually presented in the setting of Example 29. We review this in our setting of abelian groups, referring to the general lattice decomposition in (53). Periodisation and sampling are dual operations, in the sense that a function $f \in \mathbb{C}^G$ can be moved to $\widehat{\mathbb{C}^H}$ in two different ways, we can first sample f down to H and then compute the Fourier transform on H , or we can Fourier transform f on G and then periodise \widehat{f} down to \widehat{H} . The result of these two operations is the same!

Theorem 14 For a lattice $\phi_1 \in \text{mono}(H, G)$

$$\mathcal{F}_H [\phi_1^* f] = \widehat{\phi_1}_* \mathcal{F}_G [f] \quad \forall f \in \mathcal{S}(G), \quad (54)$$

where $\mathcal{F}_H[\cdot]$ and $\mathcal{F}_G[\cdot]$ denotes the Fourier transforms on H and G .

Proof Pick an arbitrary $\xi \in \widehat{G}$ and let $\chi_\xi(x) := (\xi, x)_G$ be the corresponding character on G . Using the shift property of the Fourier transform and the Poisson summation formula, we find

$$\begin{aligned} \widehat{\phi_1}_* \mathcal{F}_G [f] (\widehat{\phi_1}(\xi)) &= \sum_{k \in \ker(\widehat{\phi_1})} \mathcal{F}_G [f](\xi + k) = \sum_{k \in \ker(\widehat{\phi_1})} \mathcal{F}_G [\chi_{-\xi} f](k) \\ &= \sum_{h \in H} (-\xi, \phi_1(h))_G f(\phi_1(h)) = \sum_{h \in H} (-\widehat{\phi_1}(\xi), h)_H f(\phi_1(h)) \\ &= \mathcal{F}_H [\phi_1^* f] (\widehat{\phi_1}(\xi)). \end{aligned}$$

□

Let $f \in \mathbb{C}^G$, $\widehat{f} \in \mathbb{C}^{\widehat{G}}$, $f_H := \phi_1^* f$ and $\widehat{f}_H := \mathcal{F}_H(f_H)$. Theorem 14 says:

$$\widehat{f}_H(\widehat{\phi}_1(\xi)) = \sum_{k \in \ker(\widehat{\phi}_1)} \widehat{f}(\xi + k).$$

The *aliasing phenomenon* is the fact that Fourier components of \widehat{f} which belong to the same coset of the reciprocal lattice add up to the same component of \widehat{f}_H . To reconstruct \widehat{f} from \widehat{f}_H we must decide on which of the aliasing components in the coset is the best representative for the coset. Reconstruction of f_H is based on choosing $\sigma: \widehat{H} \rightarrow \widehat{G}$ a transversal of $\widehat{\phi}_1 \in \text{epi}(\widehat{G}, \widehat{H})$. The standard choice if $G = \mathbb{R}^n$ or a $G = \mathbb{T}^n$ is the Voronoi transversal, where σ picks points closest possible to 0 in the Euclidean norm. In the standard setting of Shannon sampling of Example 29 we choose $\sigma: T \rightarrow \mathbb{R}$ as $\sigma(x) = x/h$ for $x \in [0, \frac{1}{2})$ and $\sigma(x) = (x - 1)/h$ for $x \in [\frac{1}{2}, 1)$, but other choices are also used in particular applications, where we want to reconstruct particular parts of the spectrum (e.g. sideband coding). We will always assume that σ is chosen such that the closure of $\text{im}(\sigma) \subset \widehat{G}$ is compact.

Definition 42 (Bandlimited Function) A function $f \in \mathbb{C}^G$ is *bandlimited* with respect to a transversal $\sigma: \widehat{H} \rightarrow \widehat{G}$ if $\text{supp}(\widehat{f}) \subset \text{im}(\sigma)$, where $\text{supp}(\widehat{f})$ is the support of \widehat{f} i.e. the points where it takes non-zero values.

For a given transversal σ we define a corresponding (low-pass) filter $\alpha_\sigma \in \mathbb{C}^{\widehat{G}}$ as the indicator function on the image of σ ,

$$\alpha_\sigma(x) = \begin{cases} 1 & \text{for } x \in \text{im}(\sigma) \\ 0 & \text{else.} \end{cases}$$

Thus, f is band-limited if and only if $\widehat{f} \cdot \alpha_\sigma = \widehat{f}$. Hence we have:

Lemma 13 (Shannon–Nyquist) A band-limited function $f \in \mathcal{S}(G)$ can be reconstructed from its down-sample f_H as

$$\widehat{f} = \alpha_\sigma \cdot (\widehat{\phi}_1^* \widehat{f}_H). \quad (55)$$

Polyhedral Dirichlet Kernels

We henceforth assume that $G = \mathbb{R}^n$ or $G = \mathbb{T}^n$ and the transversal $\sigma: \widehat{H} \rightarrow \widehat{G}$ is the Voronoi transversal, with an image being a polyhedron centered at $\mathbf{0} \in \widehat{G}$. The corresponding low-pass filter is 1 inside this polyhedron and on the boundary (in particular if \widehat{G} is discrete) we give weight $1/n$ on all n points which tie-break on the distance criterion.

Definition 43 (Polyhedral Dirichlet Kernel) Let

$$\Omega = \left\{ \xi \in \widehat{G} : ||\xi|| < ||\xi - k|| \text{ for all } k \in \widehat{\phi}_2(H^\perp) \setminus \mathbf{0} \right\}$$

$$\partial\Omega = \left\{ \xi \in \widehat{G} : ||\xi|| = ||\xi - k|| \text{ for some } k \in \widehat{\phi}_2(H^\perp) \setminus \mathbf{0} \right\}$$

We define the low-pass filter $\widehat{\mathcal{D}}_H \in \mathcal{S}'(\widehat{G})$ as

$$\widehat{\mathcal{D}}_H(\xi) = \begin{cases} 1 & \text{for } \xi \in \Omega \\ \frac{1}{N} & \text{for } \xi \in \partial\Omega, \\ 0 & \text{otherwise} \end{cases},$$

where $N = \#\{k \in \widehat{\phi}_2(H^\perp) : ||\xi|| = ||\xi - k||\}$. The polyhedral Dirichlet kernel $\mathcal{D}_H \in \mathbb{C}^\infty(G) \cap \mathcal{S}'(G)$ is defined⁷ as

$$\mathcal{D}_H = \mathcal{F}_G^{-1}(\widehat{\mathcal{D}}_H).$$

Example 32 Continuing Example 29, where $G = \widehat{G} = \mathbb{R}$, and $\phi_1(j) = h j \in \text{mono}(H, G)$, we find

$$D_H(x) = \int_{-\frac{1}{2h}}^{\frac{1}{2h}} e^{2\pi i \xi x} d\xi = \frac{\sin(\pi x/h)}{\pi x} = \frac{1}{h} \text{sinc}(\pi x/h).$$

Example 33 For $G = T$, $H = \mathbb{Z}$ and $\phi_1(j) = j/N$, we have $\widehat{G} = \mathbb{Z}$, $H^\perp = \mathbb{Z}$ and $\widehat{\phi}_2(k) = Nk$, which gives

$$D_H(x) = \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} e^{2\pi i k x} = \frac{\sin(N\pi x)}{\sin(\pi x)} \quad \text{if } N \text{ is odd}$$

$$D_H(x) = \sum_{k=-\frac{N}{2}-1}^{\frac{N}{2}-1} e^{2\pi i k x} + \frac{1}{2}(e^{\pi i N x} + e^{-\pi i N x})$$

$$= \frac{\sin((N-1)\pi x)}{\sin(\pi x)} + \cos(N\pi x) \quad \text{if } N \text{ is even}$$

We want to reproduce the classical convolutional formula for band-limited reconstruction of a sampled function in our setting. Let $f \in \mathcal{S}(G)$ and let the

⁷Since $\widehat{\mathcal{D}}_H$ has compact support, its inverse Fourier transform is smooth.

Shannon–Nyquist low-pass reconstruction⁸ be given as

$$f \approx \mathcal{F}_G^{-1} \left[\widehat{\mathcal{D}}_H \cdot \left(\widehat{\phi}_1^* \widehat{f}_H \right) \right].$$

We have $\widehat{\phi}_1^* \widehat{f}_H \in \mathcal{S}'(\widehat{G})$. Recall Theorem 14, for Schwartz functions sampling and periodisation are dual operations. Tempered distributions belong to the dual space and move in the opposite direction, so we have in particular

$$\mathcal{F}_G^{-1} \left[\widehat{\phi}_1^* \widehat{f}_H \right] = \phi_{1*} f_H = \sum_{j \in H} f(\phi_1(j)) \delta_{\phi_1(j)},$$

where $\delta_{\phi_1(j)}(x) = \delta(x - \phi_1(j))$. Since $\widehat{\mathcal{D}}_H$ has compact support, there is a convolutional formula for distributions leading to the reconstruction

$$\mathcal{F}_G^{-1} \left[\widehat{\mathcal{D}}_H \cdot \left(\widehat{\phi}_1^* \widehat{f}_H \right) \right] = \frac{1}{C} \mathcal{D}_H * \left(\sum_{j \in H} f(\phi_1(j)) \delta_{\phi_1(j)} \right),$$

where the constant $C = D_H(0)$. This yields:

Theorem 15 (Shannon–Nyquist Convolution Formula) *The band-limited reconstruction off from $f_H = \phi_1^* f = f \circ \phi_1$ can be computed as*

$$f(x) \approx \frac{1}{D_H(0)} \sum_{j \in H} D_H(x - \phi_1(j)) f(\phi_1(j)). \quad (56)$$

This is an exact reconstruction for band-limited f .

We see from band limited f that the formula is interpolating in the lattice points, and we conclude:

Lemma 14 *The normalised polyhedral Dirichlet kernel satisfies for $j \in H$*

$$\frac{D_H(\phi_1(j))}{D_H(0)} = \begin{cases} 1 & \text{for } j = \mathbf{0} \\ 0 & \text{else.} \end{cases}$$

The translates $S_{\phi_1(j)} D_H(x) = D_H(x - \phi_1(j))$ for all $j \in H$ form a complete set of Lagrangian basis functions for band limited trigonometric interpolation in the lattice points.

Analytical properties of polyhedral Dirichlet kernels are important for understanding sampling theory on general lattices. Detailed analysis of these functions

⁸For band limited f this is exact, for other functions it is an interpolating formula.

is done in [27, 30]. In particular it is important that they in the case $G = T^n$ the interpolation operator has a Lebesgue constant scaling like $\mathcal{O}(\log^n(N))$, where N is the number of sampling points in H .

3.7 The Fast Fourier Transform (FFT)

We return once more to the basic splitting diagram (53), but in this section we assume that all involved groups are finite. The aim is to compute the discrete Fourier transform (DFT) on G by expressing \mathcal{F}_G in terms of the DFTs \mathcal{F}_H and \mathcal{F}_K , where $K = G/H$. The simplest situation is when the diagram (53) splits, i.e. the case when $G = H \oplus K$. Then there exists homomorphisms $\sigma_1 \in \text{epi}(G, H)$ and $\sigma_2 \in \text{mono}(K, G)$ such that $\sigma_1 \circ \phi_1 = \text{Id}_H$ and $\phi_2 \circ \sigma_2 = \text{Id}_K$, and an isomorphism $\psi = \frac{\sigma_1}{\phi_2} \in \text{iso}(G, H \oplus K)$. On $\mathbb{C}[H \oplus K]$ the DFT is $\mathcal{F}_H \oplus \mathcal{F}_K$, thus the whole DFT on G factorises as

$$\mathcal{F}_G = \widehat{\psi} \circ (\mathcal{F}_H \oplus \mathcal{F}_K) \circ \psi.$$

We can think of $\mathbb{C}[H \oplus K]$ as a 2D table. The isomorphisms ψ and $\widehat{\psi}$ are just permutations of the data, so the factorisation has three stages; first we use ψ_1 to arrange the data in a 2D table, then we use \mathcal{F}_H on each column, and \mathcal{F}_K on each row of the table, and finally we collect the data back into \widehat{G} . The computation is facilitated by a software package for doing computations of homomorphisms between finite abelian groups. This factorisation of the DFT in the case where $G = H \oplus K$ is in FFT literature called *twiddle-free* FFT decomposition.

In the more general situation we have that $H \oplus K$ is not isomorphic to G . In this case we still try to use H and K as coordinates on G , but we cannot do this in a canonical way. We choose two transversals $\sigma_K: K \rightarrow G$ and $\sigma_H: \widehat{H} \rightarrow \widehat{G}$ and write

$$\begin{aligned} j &= \phi_1(m) + \sigma_K(\ell) \quad \text{for } m \in H, \ell \in K, j \in G \\ k &= \widehat{\phi}_2(p) + \sigma_H(n) \quad \text{for } p \in \widehat{K}, n \in \widehat{H}, k \in \widehat{G}. \end{aligned}$$

Using the properties we have derived for dual pairings we find (exercise!)

$$(k, j)_G = (n, m)_H(p, \ell)_K(\sigma_H(n), \sigma_K(\ell))_G.$$

The last factor $(\sigma_H(n), \sigma_K(\ell))_G$ is called a ‘twiddle factor’ and it reflects the fact that $G \neq H \oplus K$. We find that the Fourier transform on G factorises as

$$\begin{aligned} \mathcal{F}_G[f](k) &= \mathcal{F}_G[f](\widehat{\phi}_2(p) + \sigma_H(n)) \\ &= \sum_{\ell \in K} \left((\sigma_H(n), \sigma_K(\ell))_G \sum_{m \in H} (-n, m)_H f(\phi_1(m) + \sigma_K(\ell)) \right) (-p, \ell)_K. \end{aligned} \tag{57}$$

Again, interpreting f as data in a 2D array, indexed by $m \in H$ and $\ell \in K$, we see that the DFT on G factorises in applying \mathcal{F}_H on each column, then multiplying by the twiddle factors and finally \mathcal{F}_K on the rows. This is the basis for the Cooley–Tukey algorithm, where this factorisation is done recursively to obtain the Fast Fourier Transform. The fact that this can be done with respect to any subgroup $H < G$ is of theoretical importance, and practical importance if we want to design versions of FFTs taking account of symmetries in the data f , see [18].

However, this factorisation is not canonical, there is a choice of transversals and twiddle factors involved. So aesthetically this factorisation formula is not optimal. It is possible to obtain a canonical factorisation of a similar nature. For completeness, I would like to explain also this factorisation. This involves the lifting of f to a larger space than $H \oplus K$, called the Heisenberg group (originating from quantum mechanics). The last part of this section may be skipped without loss of continuity.

3.7.1 Heisenberg Groups and the Weil–Brezin Map

More material on topics related to this section is found in [5, 29].

We can act upon $f \in \mathbb{C}[G]$ with a time-shift $S_x f(t) := f(t + x)$ and with a frequency shift $\chi_\xi f(t) := (\xi, t)f(t)$. These two operations are dual under the Fourier transform, but do not commute:

$$\widehat{S_x f}(\xi) = \chi_x \widehat{f}(\eta) \quad (58)$$

$$\widehat{\chi_\xi f}(\eta) = S_{-\xi} \widehat{f}(\eta) \quad (59)$$

$$(S_x \chi_\xi f)(t) = (\xi, x) \cdot (\chi_\xi S_x f)(t). \quad (60)$$

The full (non-commutative) group generated by time and frequency shifts on $\mathbb{C}[G]$ is called the *Heisenberg group* of G .

The Heisenberg group of \mathbb{R}^n is commonly defined as the multiplicative group of matrices of the form

$$\begin{pmatrix} 1 & x^T & s \\ 0 & I_n & \xi \\ 0 & 0 & 1 \end{pmatrix},$$

where $\xi, x \in \mathbb{R}^n$, $s \in \mathbb{R}$. This group is isomorphic to the semidirect product $\mathbb{R}^n \times \mathbb{R}^n \rtimes \mathbb{R}$ where

$$(\xi', x', s') \cdot (\xi, x, s) = (\xi' + \xi, x' + x, s' + s + x'^T \xi).$$

We prefer to instead consider $\mathbb{R}^n \times \mathbb{R}^n \rtimes \mathbb{T}$ (where \mathbb{T} is the multiplicative group consisting of $z \in \mathbb{C}$ such that $|z| = 1$) with product

$$(\xi', x', z') \cdot (\xi, x, z) = (\xi' + \xi, x' + x, z'ze^{2\pi i x'^T \xi}).$$

More generally:

Definition 44 For an LCA G we define the Heisenberg group $\mathcal{H}_G = \widehat{G} \times G \rtimes \mathbb{T}$ with the semidirect product

$$(\xi', x', z') \cdot (\xi, x, z) = (\xi' + \xi, x' + x, z' \cdot z \cdot (\xi, x')).$$

We define a *left action* $\mathcal{H}_G \times \mathbb{C}[G] \rightarrow \mathbb{C}[G]$ as follows

$$(\xi, x, z) \cdot f = z \cdot \chi_\xi S_x f. \quad (61)$$

To see that this defines a left action, we check that $(0, 0, 1) \cdot f = f$ and

$$(\xi', x', z') \cdot ((\xi, x, z) \cdot f) = ((\xi', x', z') \cdot (\xi, x, z)) \cdot f.$$

Lemma 15 Let $\mathcal{H}_G = \widehat{G} \times G \rtimes \mathbb{T}$ and $\mathcal{H}_{\widehat{G}} = G \times \widehat{G} \rtimes \mathbb{T}$ act upon $f \in \mathbb{C}G$ and $\widehat{f} \in \mathbb{C}\widehat{G}$ as in (61). Then

$$\mathcal{F}((\xi, x, z) \cdot f) = z \cdot (-\xi, x) \cdot \chi_x S_{-\xi} \widehat{f} = (x, -\xi, z \cdot (-\xi, x)) \cdot \widehat{f}$$

Proof This follows from (58)–(60).

We will henceforth assume that H, G and K form a short exact sequence as in (53), with H discrete and $K = G/H$ compact.

Definition 45 (Weil–Brezin Map) The *Weil–Brezin map* \mathcal{W}_G^H is defined for $f \in \mathbb{C}[G]$ and $(\xi, x, z) \in \mathcal{H}_G$ as

$$\mathcal{W}_G^H f(\xi, x, z) = \sum_{j \in H} ((\xi, x, z) \cdot f)_H(j),$$

where $f_H := f \circ \phi_1$ denotes downsampling along $\phi_1 \in \text{mono}(H, G)$.

A direct computation shows that the Weil–Brezin map satisfies the following symmetries for all $(h', h, 1) \in H^\perp \times H \times 1 \subset \mathcal{H}_G$ and all $z \in \mathbb{T}$:

$$\mathcal{W}_G^H f((h', h, 1) \cdot (\xi, x, s)) = \mathcal{W}_G^H f(\xi, x, s) \quad (62)$$

$$\mathcal{W}_G^H f(\xi, x, z) = z \cdot \mathcal{W}_G^H f(\xi, x, 1). \quad (63)$$

Lemma 16 $\Gamma = H^\perp \times H \times 1$ is a subgroup of \mathcal{H}_G . It is not a normal subgroup, so we cannot form the quotient group. However, as a manifold the set of right cosets is

$$\Gamma \backslash \mathcal{H}_G = \widehat{H} \times K \times \mathbb{T}.$$

The Heisenberg group has a right and left invariant volume measure given by the direct product of the invariant measures of \widehat{G} , G and \mathbb{T} . Thus we can define the Hilbert spaces $L^2(\mathcal{H}_G^H)$ and $L^2(\widehat{H} \times K \times \mathbb{T})$. By Fourier decomposition in the last variable (z -transform), $L^2(\widehat{H} \times K \times \mathbb{T})$ splits into an orthogonal sum of subspaces \mathcal{V}_k for $k \in \mathbb{Z}$, consisting of those $g \in L^2(\widehat{H} \times K \times \mathbb{T})$ such that

$$g(\xi, x, z) = z^k g(\xi, x, 1) \quad \text{for all } z = e^{2\pi i \theta}.$$

It can be verified that \mathcal{W}_G^H is unitary with respect to the L^2 inner product. Together with (62)–(63) this implies:

Lemma 17 *The Weil–Brezin map is a unitary transform*

$$\mathcal{W}_G^H: L^2(G) \rightarrow \mathcal{V}_1 \subset L^2(\widehat{H} \times K \times \mathbb{T}).$$

Note that the Weil–Bezin map on \widehat{G} , with respect to the reciprocal lattice H^\perp , is

$$\mathcal{W}_{\widehat{G}}^{H^\perp}: L^2(G) \rightarrow \mathcal{V}_1 \subset L^2(K \times \widehat{H} \times \mathbb{T}).$$

The Poisson summation formula (Theorem 13) together with Lemma 15 implies that these two maps are related via

$$\mathcal{W}_G^H f(\xi, x, z) = \mathcal{W}_{\widehat{G}}^{H^\perp} \widehat{f}(x, -\xi, z \cdot (\xi, x)).$$

Defining the unitary map $J: L^2 \subset L^2(\widehat{H} \times K \rtimes \mathbb{T}) \rightarrow L^2(K \times \widehat{H} \times \mathbb{T})$ as

$$Jf(x, -\xi, z \cdot (\xi, x)) = f(\xi, x, z), \tag{64}$$

we obtain the following fundamental theorem.

Theorem 16 (Weil–Brezin Factorization) *Given an LCA G and a lattice $H < G$. The Fourier transform on G factorizes in a product of three unitary maps*

$$\mathcal{F}_G = \left(\mathcal{W}_{\widehat{G}}^{H^\perp} \right)^{-1} \circ J \circ \mathcal{W}_G^H. \tag{65}$$

The Zak Transform

We want to explain (57) in terms of the Weil–Brezin map. Given a lattice $H < G$ and transversals $\sigma: K \rightarrow G$ and $\widehat{\sigma}: \widehat{H} \rightarrow \widehat{G}$. The *Zak transform* is defined as

$$\mathcal{Z}_G^H f(\xi, x) := \mathcal{W}_G^H f(\xi, x, 1) \quad \text{for } \xi \in \widehat{\sigma}(\widehat{H}), x \in \sigma(K). \quad (66)$$

The Zak transform can be computed as a collection of Fourier transforms on H of f shifted by x , for all $x \in \sigma(K)$. The definition of the Fourier transform yields:

$$\mathcal{Z}_G^H f(-\xi, x) = \mathcal{F}_H((S_x f)_H)(\widehat{\phi}_0(\xi)). \quad (67)$$

We see that the Zak transform is invertible when $\mathcal{Z}_G^H f(-\xi, x)$ is computed for all $\xi \in \widehat{\sigma}(\widehat{H})$ and all $x \in \sigma(K)$. Written in terms of the Zak transform, the Weil–Brezin factorization (65) becomes

$$\mathcal{Z}_{\widehat{G}}^{H^\perp} \widehat{f}(x, \xi) = (\xi, x)_G \mathcal{Z}_G^H f(-\xi, x). \quad (68)$$

This is essentially the same formula as (57), where $(\xi, x)_G$ is the *twiddle factor*.

Due to the symmetries (62)–(63), the Weil–Brezin map is trivially recovered from the Zak transform. The Zak transform is the practical way of computing the Weil–Brezin map and its inverse. However, since the invertible Zak transform cannot be defined canonically, independently of the transversals σ and $\widehat{\sigma}$, the Weil–Brezin formulation is more fundamental.

We end our discussion of the FFT at this point with the remark that the DFT on a finite abelian group G can always be computed with a complexity of $\mathcal{O}(|G| \log |G|)$ floating point operations, although for some cases such as \mathbb{Z}_p , where p is a large prime we must use other techniques than those discussed here. The underlying principles for computing the DFT are based on group theory. The details of state of the art FFT-software is involved, but for most applications in computational science it is sufficient to know that excellent FFT libraries exists. The practical question is then how the Fourier transforms on more general LCAs can be related to the finite groups.

3.8 Lattice Rules

Lattice rules are numerical algorithms for computing in continuous groups by sampling in regular lattices and reducing to computations on finite groups. Most commonly the term refers to numerical integration of multivariate periodic functions in \mathbb{C}^{T^n} . The solution of PDEs by lattice sampling rules is discussed in [21]. In the present general setting, we discuss lattice rules for functions on $G = \mathbb{R}^n$, in which

case we must introduce sampling lattices both in G and in \widehat{G} to obtain a finite group where computations reduce to the FFT.

For general functions $f \in \mathbb{C}^G$, the error between the Fourier transform of the true and the sampled function follows from Theorem 14

$$\mathcal{F}_H(f_H)(\widehat{\phi}_1(\xi)) - \mathcal{F}_G(f)(\xi) = \sum_{k \in \widehat{\phi}_2(\widehat{K}) \setminus \{0\}} \mathcal{F}_G(f)(k + \xi).$$

The game of Lattice rules is, given f with specific properties, to find a lattice $H < G$ such that the error is minimised. We first assume (as is commonly done in the lattice-rule literature) that the original domain is periodic $G = T^n$. Lattice rules are designed such that the nonzero points in H^\perp neighbouring 0 are pushed as far out as possible with respect to a given norm, depending on the properties of f . If f is spherically symmetric, H should be chosen as a *densest lattice packing* (with respect to the 2-norm) [8], e.g. hexagonal lattice in \mathbb{R}^2 and face centred cubic packing in \mathbb{R}^3 (as the orange farmers know well). In dimensions up to 8, these are given by certain root lattices [20]. The savings, compared to standard tensor product lattices, are given by the factors 1.15, 1.4, 2.0, 2.8, 4.6, 8.0 and 16.0 in dimensions $n = 2, 3, \dots, 8$. This is important, but not dramatic, e.g. a camera with 8.7 megapixels arranged in a hexagonal lattice has approximately the same sampling error as a 10 megapixel camera with a standard square pixel distribution. However, these alternative lattices have other attractive features, such as larger spatial symmetry groups, yielding more isotropic discretizations. A hexagonal lattice picture can be rotated more uniformly than a square lattice picture.

Dramatic savings can be obtained for functions belonging to the *Korobov spaces*. This is a common assumption in much work on high dimensional approximation theory. Korobov functions are functions whose Fourier transforms have energy concentrated along the axis directions in \widehat{G} , the so-called hyperbolic cross energy distribution. Whereas the tensor product lattice with $2d$ points in each direction contains $(2d)^n$ lattice points in T^n , the optimal lattice with respect to the Korobov norm contains only $\mathcal{O}(2^n d (\log(d))^{n-1})$ points, removing exponential dependence on d .

The group theoretical understanding of lattice rules makes software implementation very clean and straightforward. In [21], numerical experiments are reported on lattice rules for FFT-based spectral methods for PDEs. Note that whereas the choice of transversal $\widehat{\sigma}: \widehat{H} \rightarrow \widehat{G}$ is irrelevant for lattice integration rules, it is essential for pseudospectral derivation. The Laplacian $\nabla^2 f$ is computed on \widehat{G} as $\widehat{f}(\xi) \mapsto c|\xi|^2 \widehat{f}(\xi)$, whereas the corresponding computation on \widehat{H} must be done as $\mathcal{F}_H(f_H)(\eta) \mapsto c|\widehat{\sigma}(\eta)|^2 \mathcal{F}_H(f_H)(\eta)$ for $\eta \in \widehat{H}$, and we must choose the Voronoi transversal to minimise aliasing errors.

3.8.1 Computational Fourier Analysis on \mathbb{R}^d

A topic which in our opinion has not been fully addressed in the Lattice-rule literature is the computation of Fourier transforms on the non-compact continuous groups \mathbb{R}^d . The problem here is that there are no homomorphisms of a finite abelian group into \mathbb{R}^d , since any lattice in \mathbb{R}^d must be non-compact. In order to move the computation to a finite group $\mathbb{Z}_{\mathbf{n}}$, $\mathbf{n} = (n_1, \dots, n_k)$, we must use *two* homomorphisms

$$\mathbb{Z}_{\mathbf{n}} \xrightarrow{\phi_s} \mathbb{T}^d \xleftarrow[\mathbb{R}]{} \overset{\phi_p}{\mathbb{R}}^d, \quad (69)$$

where ϕ_s is a sampling lattice and ϕ_p defines periodisation of a function with respect to the *periodisation lattice* $\ker(\phi_p) < \mathbb{R}^d$. A function $f \in S(\mathbb{R}^n)$ can be mapped down to a finite $f_{\mathbf{n}} \in \mathcal{S}(\mathbb{Z}_{\mathbf{n}})$ as

$$f_{\mathbf{n}} = (\phi_s^* \circ \phi_p)_*(f). \quad (70)$$

Since $\widehat{\mathbb{Z}_{\mathbf{n}}} = \mathbb{Z}_{\mathbf{n}}$, $\widehat{T}^d = \mathbb{Z}^d$ and $\widehat{\mathbb{R}^d} = \mathbb{R}^d$, the dual of (69) is

$$\mathbb{Z}_{\mathbf{n}} \xleftarrow[\mathbb{Z}]{} \overset{\widehat{\phi}_s}{\mathbb{Z}}^d \xrightarrow{\widehat{\phi}_p} \mathbb{R}^d. \quad (71)$$

The relationship between the discrete and the continuous Fourier transform follows from Theorem 14 (applied twice)

$$\widehat{f}_{\mathbf{n}} = (\widehat{\phi}_{s*} \circ \widehat{\phi}_p^*)(\widehat{f}). \quad (72)$$

Note that sampling in the primary domain becomes periodisation in the Fourier domain and vice versa. The reconstruction of \widehat{f} from $\widehat{f}_{\mathbf{n}}$ can be done as a 2-stage process involving band limited approximation in the Voronoi domain of the dual sampling lattice $\ker(\widehat{\phi}_s) < \mathbb{Z}^d$ and a space limited approximation in the Voronoi domain of the periodisation lattice $\ker(\widehat{\phi}_p) < \mathbb{R}^d$, using the Shannon–Nyquist reconstruction formula twice. In this process no function (except $f = 0$) is perfectly reconstructed, since $f = 0$ is the only function which is both band limited and with compact support.

We want to understand the mappings involved in this two-stage sampling process in more detail. We have two lattices, the periodisation lattice $\ker(\phi_p) < \mathbb{R}^n$ and sampling lattice $\phi_s(\mathbb{Z}_{\mathbf{n}}) < T^d$. Considering the sampling lattice lifted to \mathbb{R}^d as the subgroup $\phi_p^{-1}(\phi_s(\mathbb{Z}_{\mathbf{n}})) < \mathbb{R}^d$, we realise that the periodisation lattice is a sub-lattice of the sampling lattice in \mathbb{R}^d . The two lattices are described by two matrices $S \in \mathbb{R}^{d \times d}$ and $A \in \mathbb{Z}^{d \times d}$ with non-vanishing determinants. These define a sampling lattice $S: \mathbb{Z}^d \hookrightarrow \mathbb{R}^d$ and a periodisation sub lattice defined by $SA: \mathbb{Z}^d \hookrightarrow \mathbb{R}^d$. Consider the following commutative diagram where all rows and all columns are

exact and where $\mathbf{n} = (n_1, \dots, n_k)$ such that $n_i|n_{i+1}$ for $i = 1, \dots, k-1$. The second row describes the sampling lattice and the second column the periodisation lattice in \mathbb{R}^n .

$$\begin{array}{ccccccc}
& 0 & & 0 & & & \\
& \downarrow & & \downarrow & & & \\
0 & \longrightarrow & \mathbb{Z}^d & \xlongequal{\quad} & \mathbb{Z}^d & \longrightarrow & 0 \\
& & \downarrow A & & \downarrow & & \downarrow \\
0 & \longrightarrow & \mathbb{Z}^d & \xrightarrow{S} & \mathbb{R}^d & \longrightarrow & T^d \longrightarrow 0 \\
& & \downarrow & & \downarrow & & \parallel \\
0 & \longrightarrow & \mathbb{Z}_{\mathbf{n}} & \longrightarrow & T^d & \longrightarrow & T^d \longrightarrow 0 \\
& & \downarrow & & \downarrow & & \downarrow \\
& 0 & & 0 & & 0 &
\end{array}$$

Given A and S as above, there is a unique (up to isomorphisms) way of completing this diagram such that all rows and columns are exact. We will explicitly compute all the arrows. Let the Smith normal form of A be

$$A = UNV,$$

where $U \in \mathbb{Z}^{d \times d}$ and $V \in \mathbb{Z}^{d \times d}$ are unimodular and $N \in \mathbb{Z}^{d \times d}$ is diagonal, where the diagonal $n_i = N_{i,i}$ contains positive integers such that $n_i|n_{i+1}$ for all i . Since A has nonzero determinant, none of the n_i are zero, but the first ones could be 1. Let k denote the number of n_i such that $n_i > 1$, and let \mathbf{n} be the vector containing these last k diagonal elements, defining the FAG $\mathbb{Z}_{\mathbf{n}}$. Let $U_k \in \mathbb{Z}^{k \times d}$ denote the last k rows of U^{-1} and let $V_k \in \mathbb{Z}^{d \times k}$ denote the last k columns of V^{-1} . Finally, let $N_k = \text{diag}(\mathbf{n}) \in \mathbb{Z}^{k \times k}$ and $N_k^{-1} \in \mathbb{R}^{k \times k}$. Then the diagram is completed as follows.

$$\begin{array}{ccccccc}
& 0 & & 0 & & & \\
& \downarrow & & \downarrow & & & \\
0 & \longrightarrow & \mathbb{Z}^d & \xlongequal{\quad} & \mathbb{Z}^d & \longrightarrow & 0 \\
& & \downarrow A & & \downarrow SA & & \downarrow \\
0 & \longrightarrow & \mathbb{Z}^d & \xrightarrow{S} & \mathbb{R}^d & \xrightarrow{S^{-1}} & T^d \longrightarrow 0 \\
& & \downarrow U_k & & \downarrow (SA)^{-1} & & \parallel \\
0 & \longrightarrow & \mathbb{Z}_{\mathbf{n}} & \xrightarrow{V_k N_k^{-1}} & T^d & \xrightarrow{A} & T^d \longrightarrow 0 \\
& & \downarrow & & \downarrow & & \downarrow \\
& 0 & & 0 & & 0 &
\end{array}$$

It is straightforward to check that the first two rows and last two columns are short and exact sequences. The first column is a short exact sequence because $U_k A \bmod \mathbf{n} = 0$ and U_k has maximal rank. It is straightforward to check the commutativity of the NW, NE and SE squares. The SW square commutes because $A^{-1} - V_k N_k^{-1} U_k$ is an integer matrix. The last row is exact by the 9-lemma of homological algebra [17].

This defines the periodisation lattice $SA: \mathbb{Z}^d \hookrightarrow \mathbb{R}^d$ with quotient mapping $(SA)^{-1}: \mathbb{R}^d \twoheadrightarrow T^d$. A function $f \in \mathcal{S}(\mathbb{R}^d)$ can be approximated by a function $f_{\mathbf{n}} \in \mathcal{S}(\mathbb{Z}_{\mathbf{n}})$ as

$$f_{\mathbf{n}} = (U_k)_* S^* f = (V_k N_k^{-1})^* (SA)_*^{-1} f.$$

We say that $V_k N_k^{-1}$ is a *rank k lattice rule*. Dualising the above sampling-periodisation diagram yields:

$$\begin{array}{ccccccc} & 0 & & 0 & & & \\ & \uparrow & & \uparrow & & & \\ 0 & \longleftarrow & T^d & = & T^d & \longleftarrow & 0 \\ & A^T \uparrow & & (SA)^T \uparrow & & & \uparrow \\ 0 & \longleftarrow & T^d & \xleftarrow{S^T} & \mathbb{R}^d & \xleftarrow{S^{-T}} & \mathbb{Z}^d \longleftarrow 0 \\ & U_k^T N_k^{-1} \uparrow & & (SA)^{-T} \uparrow & & & \parallel \\ 0 & \longleftarrow & \mathbb{Z} & \xleftarrow{V_k^T} & \mathbb{Z}^d & \xleftarrow{A^T} & \mathbb{Z}^d \longleftarrow 0 \\ & \uparrow & & \uparrow & & & \uparrow \\ & 0 & & 0 & & & 0 \end{array}$$

If we flip the diagram around the SW-NE diagonal, we see that this is nearly identical to the original sampling-periodisation diagram. But here the sampling lattice is given by $\widehat{SA} = (SA)^{-T}: \mathbb{Z}^d \rightarrow \mathbb{R}^d$, while the periodisation lattice is given by $\widehat{S} = S^{-T}: \mathbb{Z}^d \rightarrow \mathbb{R}^d$. Thus, the reciprocal of the primal sampling lattice is the dual periodisation lattice and the reciprocal of the primal periodisation lattice is the dual sampling lattice.

Complete symmetry between primal and dual spaces is obtained by letting the primal sampling lattice be obtained by down-scaling the reciprocal of the primal periodisation lattice in \mathbb{R}^d . Specifically, given a non-singular matrix $L \in \mathbb{R}^{d \times d}$ and an integer m , we let the primal sampling lattice be $S = \frac{1}{m}L$ and $A = mL^{-1}L^{-T}$. The primal periodisation lattice is $SA = L^{-T}$. The dual sampling lattice is $(SA)^{-T} = L = mS$ and the dual periodisation lattice $S^{-T} = mSA$.

3.8.2 Eigenfunctions of the Continuous and Discrete Fourier Transforms

We end this section with a brief remark showing a beautiful and perhaps unexpected property of discretising \mathbb{R}^n in a completely symmetric fashion as discussed above. To understand the analytic properties of the discrete and continuous Fourier

transforms, it is of importance to know the eigenfunctions of the Fourier operator. The eigenvectors of discrete Fourier transforms is a topic of interest both in pure and applied mathematics [5]. Both on \mathbb{R}^n and on \mathbb{Z}_n , the Fourier transform is a linear operator from a space to itself, so we can talk about eigenfunctions η with the property that $\widehat{\eta} = \lambda \cdot \eta$ for some $\lambda \in \mathbb{C}$. The Fourier transform satisfies $\mathcal{F}^4 = I$, hence we have that $\lambda \in \{1, i, -1, -i\}$. Since there are only four invariant subspaces, the eigenfunctions are not uniquely defined. However, for $\mathcal{F}_{\mathbb{R}}$ a particular complete set of eigenfunctions is known. The most famous eigenfunction is an appropriate scaling of the Gaussian $\exp(-x^2/\sigma^2)$, which is the ground state of the quantum harmonic oscillator. The set of all the eigenstates of the quantum harmonic oscillator form a complete set of eigenfunctions of $\mathcal{F}_{\mathbb{R}}$. These are of the form of a Hermite polynomial times a Gaussian. Similarly, we can take the eigenstates of the d -dimensional quantum harmonic oscillator as a basis for the eigen spaces of the d -dimensional Fourier transform $\mathcal{F}_{\mathbb{R}}^d$.

Theorem 17 *If we have a complete symmetry between the primal and dual sampling and periodisation lattices on \mathbb{R}^d , then the discretisation of eigenfunction η of the continuous Fourier transform $\mathcal{F}_{\mathbb{R}^d}$*

$$\eta_n = (U_{k*} \circ S^*)\eta$$

is an eigenfunction of the discrete Fourier transform $\mathcal{F}_{\mathbb{Z}_n}$.

Proof In the symmetric situation we have that the primal and dual discretisations are the same

$$\begin{aligned}\eta_n &= (U_{k*} \circ S^*)\eta \\ \mathcal{F}_{\mathbb{Z}_n}[\eta_n] &= (U_{k*} \circ S^*)\mathcal{F}_{\mathbb{R}^d}[\eta],\end{aligned}$$

hence

$$\mathcal{F}_{\mathbb{Z}_n}[\eta_n] = (U_{k*} \circ S^*)\mathcal{F}_{\mathbb{R}^d}[\eta] = \lambda \cdot (U_{k*} \circ S^*)\eta = \lambda \cdot \eta_n.$$

It might be an interesting research topic to investigate computational reconstruction algorithms which aims at being accurate for down sampled eigenstates of the quantum harmonic oscillator.

3.9 Boundaries, Mirrors and Kaleidoscopes

The Fourier theory is a perfect tool for computing with shift invariant linear operators. This is, however, a very ideal world. Practical computational problems usually involve operators with coefficients varying over space and problems with boundaries. What can we do with such problems? A crucial technique is precondi-

tioning, where real-life computational problems are approximated by problems in the ideal world. E.g. operators with variable coefficients can be approximated by operators where the coefficients are averaged over the domain. This is discussed in Sect. 3.11. For boundaries, it is worth knowing that *certain* special boundaries can be treated exactly within Fourier theory. The classification of such domains is of importance to computational science. One technique, which we will not pursue here, is based on separation of variables for PDEs. This has lead to fast solvers for Poisson problems on domains such as rectangles and circles.

We will instead discuss boundary problems which can be solved by Fourier techniques using mirrors on the boundaries of the domain, leading to fast computational techniques for *certain* triangles (2D) and simplexes in higher dimensions. These techniques also relates to beautiful topics in pure mathematics, such as the classification of reflection groups (kaleidoscopes), and the classification of semi-simple Lie groups.

We provide the basic idea with a well-known example derived in an unusual manner.

Example 34 Find eigenvectors and eigenvalues of the discrete 1-D Laplacian with Dirichlet conditions, the $(n - 1) \times (n - 1)$ matrix

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

Apart from the boundaries, this is a convolution on $\mathbb{C}^{\mathbb{Z}}$ with $\mathbf{a} \in \mathbb{C}^{\mathbb{Z}}$ given as $\mathbf{a}(0) = 2$, $\mathbf{a}(1) = -1$, $\mathbf{a}(-1) = -1$. Now we fix the boundaries by setting up mirrors on the edges $j = 0$ and $j = n$. Since we have Dirichlet conditions, we set up two mirrors which act on a function by flipping it around a boundary point and changing the sign, i.e. we have two reflections acting on $f \in \mathbb{C}^{\mathbb{Z}}$ as

$$\begin{aligned} F_1 f(j) &= -f(-j) \\ F_2 f(j) &= -f(2n - j). \end{aligned}$$

Note that the convolution operator \mathbf{a} commutes with the reflections,

$F_i(\mathbf{a} * f) = \mathbf{a} * (F_i f)$ for every $f \in \mathbb{C}^{\mathbb{Z}}$. We seek a subspace of $\mathbb{C}^{\mathbb{Z}}$ of functions invariant under the action of F_i , the linear subspace $V \subset \mathbb{C}^{\mathbb{Z}}$ such that for all $f_s \in V$ we have $F_1 f = F_2 f = f$. Note that $F_2 \circ F_1 = S_{2n}$, the shift operator $S_{2n} f(j) = f(j - 2n)$. Hence we see that $V \subset \mathbb{C}^{\mathbb{Z}/2n\mathbb{Z}} = \mathbb{C}[\mathbb{Z}_{2n}]$, and furthermore

$$V = \{f \in \mathbb{C}[\mathbb{Z}_{2n}] : F_1 f = f\}.$$

The other symmetry $F_2 f = f$ follows because of 2n-periodicity.

Let $\Pi = \frac{1}{2}(I + F_1): \mathbb{C}[\mathbb{Z}_{2n}] \rightarrow V$ be the projection onto the symmetric subspace. V contains all $2n$ periodic functions of the form

$$(\dots, -f_2, -f_1, 0, f_1, f_2, \dots, f_{n-1}, 0, -f_{n-1}, \dots).$$

Let $\Omega = \{1, 2, \dots, n-1\}$ be the fundamental domain of the symmetric subspace, i.e. any $f \in V$ can be reconstructed from its restriction $f|_{\Omega}$. Note that A acts on the fundamental domain just like the convolution with \mathbf{a} on V ,

$$(\alpha * f)|_{\Omega} = A \cdot f|_{\Omega} \quad \text{for all } f \in V.$$

Since the convolution $\mathbf{a}*$ commutes with F_i , it also commutes with the projection Π and hence for any eigenvector $\alpha * \eta = \lambda \eta$ we have

$$\alpha * (\Pi \eta) = \Pi(\alpha * \eta) = \lambda \Pi \eta,$$

so $\Pi \eta$ is also an eigenvector with the same eigenvalue. Hence

$$A \cdot \Pi \eta|_{\Omega} = (\alpha * \Pi \eta)|_{\Omega} = \lambda \Pi \eta|_{\Omega},$$

so $\Pi \eta|_{\Omega}$ is an eigenvector of A . On $\mathbb{C}[\mathbb{Z}_{2n}]$ the eigenvectors of the convolution $\mathbf{a}*$ are the characters $\chi_k = \exp(\pi ijk/n)$, which yields the eigenvector for A :

$$(\Pi \chi_k)(j) = \frac{1}{2}(e^{\pi ijk/n} + e^{-\pi ijk/n}) = \cos(\pi jk/n).$$

The corresponding eigenvalue is $\lambda_k = \widehat{\mathbf{a}}(k) = 2 - 2 \cos(\pi k/n)$.

The trick in this example works for the following reasons:

- The matrix A acts like a convolution $\mathbf{a}*$ inside of the domain Ω .
- On the boundary of Ω , the boundary conditions can be satisfied by reflection operators, which commute with the convolution.
- A acts on the domain Ω as the convolution $\mathbf{a}*$ acts on the symmetrized extended functions in V .
- The reflections generate translations, so that we can obtain the eigenfunctions of A from the eigenfunctions on the larger periodic domain.

By similar techniques, we can find eigenvectors for a number of different tri-diagonal matrices with combinations of Dirichlet or Neumann conditions at lattice points or between lattice points. More generally, we can ask: On which domains in \mathbb{R}^d can we define boundary conditions by similar mirror techniques and employ symmetric versions of Fourier expansions as a computational tool? To answer this question, we ask first what are the polytopes in \mathbb{R}^d with the property that if we reflect the domain at its boundaries, we eventually generate finite translations in all d directions? When these domains are understood, we can by reflection techniques find the eigenfunctions of the Laplacian ∇^2 on these domains, with

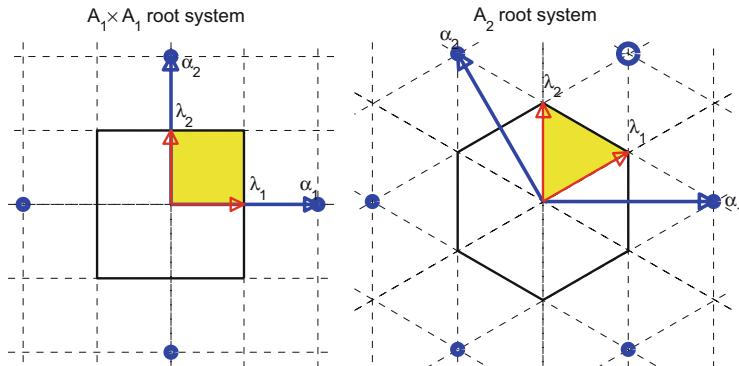


Fig. 1 Reducible root system $A_1 \times A_1$ and irreducible system A_2

various combinations of Dirichlet and Neumann conditions, and finally we can seek lattices which are invariant under the boundary reflections to obtain discretisations which can be computed by FFT techniques.

We will not go into the detail of this topic in these lectures. The interested reader is referred to [7]. We summarise the main results. In \mathbb{R}^2 the only domains with the property that reflections about the boundaries generate (finite) translations in both directions are:

- Any rectangle (Fig. 1, left).
- The equilateral triangle. Reflections of the triangle produce six rotated and reflected triangles inside a hexagon, and continued reflections produce a tiling of \mathbb{R}^2 where this hexagon is shifted in two different directions (Fig. 1, right).
- The 45° - 45° - 90° triangle. Reflections of the triangle produce eight rotated/reflected triangles inside a square, and continued reflections produce a tiling of \mathbb{R}^2 where this square is shifted in two different directions (Fig. 2, left).
- The 30° - 60° - 90° triangle. Reflections of the triangle produce 12 rotated and reflected triangles inside a hexagon, and continued reflections produce a tiling of \mathbb{R}^2 where this hexagon is shifted in two different directions (Fig. 2, right).

The classification of such ‘kaleidoscopic mirror systems’, called ‘root systems’, in all dimensions was completed in the 1890s by Wilhelm Killing and Elie Cartan. They needed this to classify all semisimple Lie groups. There are some domains which decompose in orthogonal directions, such as a rectangle, which decomposes in two mirrors in the horizontal direction and two mirrors in the vertical direction, and there are some irreducible domains which cannot be decomposed into orthogonal directions, such as the three triangles in \mathbb{R}^2 listed above. The fundamental domains for the irreducible cases are always special simplexes (triangles and their higher dimensional analogues). The fundamental domains for the reducible cases are cartesian products of irreducible domains, such as a rectangle in $2D$, which is the cartesian product of two orthogonal lines (1-D simplexes) and an equilateral

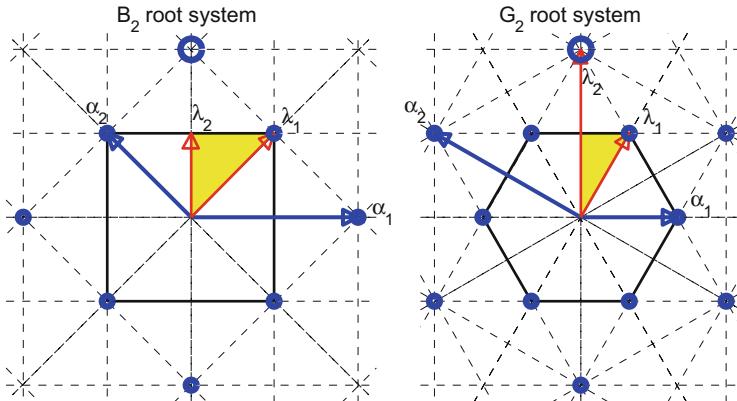


Fig. 2 The irreducible root systems B_2 and G_2

prism in 3D, which is the cartesian product of an equilateral triangle with a line. We summarise the theory:

- There are four infinite families of irreducible root systems, $A_n, n > 0$, $B_n, n > 1$, $C_n, n > 2$ and $D_n, n > 3$. Here n is the dimension of the space. A_2 is the equilateral triangle and B_2 is the 45° - 45° - 90° triangle.
- There are five exceptional root systems which only exist in particular dimensions, E_6, E_7, E_8, F_4 and G_2 , where G_2 is the 30° - 60° - 90° triangle.
- For each root system there corresponds two particularly nice families of lattices which are preserved under the reflection symmetries. These are called the roots lattice and the weights lattice and are straightforward to compute.

We refer the readers to [13, 25] for a discussion of these topics. We also mention that for each root system there corresponds a system of multivariate Chebyshev polynomials which has a number of remarkable approximation properties.

3.10 Cyclic Reduction

The topic of this section is discussed in more detail in [19]. Cyclic reduction is a classical computational technique which has been employed in the design of fast Poisson solvers [28], among other things. As a simple example consider the solution of a linear system on \mathbb{Z}_6 with coefficient matrix A is given as the convolution with \mathbf{a} where $a(0) = 2, a(1) = a(-1) = -1$ and the rest is 0. Let us pre-multiply A with B being convolution with \mathbf{b} where $b(0) = 2, b(1) = b(-1) = 1$ and the rest is 0. After re-arranging the nodes in even-odd order 0, 2, 4, 1, 3, 5, we have the following

matrix presentation of this ‘odd-even’ cyclic reduction step:

$$\begin{aligned} BA &= \left(\begin{array}{c|cc} 2 & 1 & 1 \\ 2 & 1 & 1 \\ \hline 2 & 1 & 1 \\ \hline 1 & 1 & 2 \\ 1 & 1 & 2 \\ \hline 1 & 1 & 2 \end{array} \right) \left(\begin{array}{ccc|cc} 2 & -1 & -1 & -1 & -1 \\ 2 & -1 & -1 & -1 & -1 \\ \hline -1 & -1 & 2 & -1 & -1 \\ -1 & -1 & 2 & 2 & 2 \\ -1 & -1 & 2 & 2 & 2 \end{array} \right) \\ &= \left(\begin{array}{c|ccc} 2 & -1 & -1 & -1 & -1 \\ -1 & 2 & -1 & -1 & -1 \\ \hline -1 & -1 & 2 & 2 & -1 \\ \hline & & & -1 & 2 \\ & & & -1 & -1 \\ & & & -1 & 2 \end{array} \right). \end{aligned}$$

So, we have decoupled odd and even nodes, and can continue with a problem of half the size. As a convolution of ‘stencils’, the reduction is

$$[1, 2, 1] * [-1, 2, -1] = [-1, 0, 2, 0, -1].$$

Remarks:

- If $n = 2^k$ we can apply the procedure recursively for solving $Ax = b$.
- For solving $Ax = b$ it is only necessary to do the reduction to the even points, and back substitute the solution afterwards.
- The procedure also works on those domains with boundaries, that can be represented in terms of symmetric functions as in Sect. 3.9. This includes 1D problems on an interval (Dirichlet or Neumann boundaries), rectangles in 2D, 7-point Laplacian stencil on a hexagonal lattice on an equilateral triangle with Dirichlet or Neumann boundaries, etc.
- In 2D and higher dimensions, classical (1-way) cyclic reduction schemes are unstable, and special caution must be exercised.

Our question is now how this can be generalised to the reduction of a convolutional operator $\mathbf{a} \in \mathbb{C}[G]$ to $\mathbf{b} * \mathbf{a}$ with support on an arbitrary subgroup $H < G$? The answer is a nice exercise applying the duality theory of the Fourier transform.

Theorem 18 *Given a convolutional operator $\mathbf{a} \in \mathbb{C}[G]$ a subgroup $H < G$. Let $\chi_k(j) = (k, j)_G$ and let $\mathbf{b} \in \mathbb{C}[G]$ be given as the repeated convolution*

$$\mathbf{b} = \underset{\substack{k \in H^\perp \\ k \neq 0}}{*} (\chi_k \mathbf{a}) \tag{73}$$

then $\text{supp}(\mathbf{b} * \mathbf{a}) \subset H$, thus the operator $\rho = \mathbf{b} * \mathbf{a}$ is reduced to H . The eigenvalues of ρ are

$$\widehat{\rho}(\xi) = \prod_{k \in H^\perp} \widehat{\mathbf{a}}(\xi + k). \quad (74)$$

Proof We have

$$\rho = \mathbf{b} * \mathbf{a} = \underset{k \in H^\perp}{*} (\chi_k \mathbf{a}).$$

The shift formula $\widehat{\chi_k \mathbf{a}} = S_k \widehat{\mathbf{a}}$ yields

$$\widehat{\rho} = \prod_{k \in H^\perp} S_k \widehat{\mathbf{a}},$$

which proves (74). Since $\rho(\xi + k) = \rho(\xi)$ for all $k \in H^\perp$, it follows that $\text{supp}(\rho) \subset H$, because function is supported on a lattice H if and only if its Fourier transform is periodic with respect to H^\perp , cf. Theorem 14.

Note that for $k \in H^\perp$, the characters χ_k takes constant values on the cosets of H , hence the cyclic reduction is obtained by changing the values of \mathbf{a} on the cosets of H according to χ_k .

Example 35 Consider the standard 5-point Laplacian stencil on the square lattice $\mathbb{Z}_{2n} \oplus \mathbb{Z}_{2n}$. Let $H = \mathbb{Z}_n \oplus \mathbb{Z}_m$ and $K = G/H = \mathbb{Z}_2 \oplus \mathbb{Z}_2$. The characters on K take the following values on the cosets of H :

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix},$$

thus the (2-way) cyclic reduction to H is

$$\begin{aligned} & \begin{bmatrix} 1 \\ 1 & 4 & 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 & 4 & -1 \\ 1 \end{bmatrix} * \begin{bmatrix} -1 \\ 1 & 4 & 1 \\ -1 \end{bmatrix} * \begin{bmatrix} -1 \\ -1 & 4 & -1 \\ -1 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & -32 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -32 & 0 & 132 & 0 & -32 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 0 & -32 & 0 & -2 \\ 0 & 0 & 0 \\ 1 \end{bmatrix}. \end{aligned}$$

The generalised cyclic reduction algorithm presented here is of importance to boundary problems of the form discussed in Sect. 3.9, on triangle and simplexes. Furthermore it turns out to have some other advantages compared to the classical cyclic reduction, which is based on consecutive odd-even deductions in the same direction (1-way cyclic reduction). Whereas the classical reduction is known to be unstable for Poisson problems, the condition number of the reduced operator grows exponentially in the number of reduction steps, on the other hand, with the two way (multi-) we have discussed here, the condition number of the reduced operator is *decreasing*.

Example 36 We consider the 5-point Laplacian stencil on a 32×32 square lattice with Dirichlet boundary conditions. A 1-way reduction to 1/16 of the original size increases the condition number of the reduced operator from 415 to $3 \cdot 10^{11}$. A 2-way reduction decreases the condition number to 20.9. The explanation comes from (74). The 2-way reduction is sampling the eigenvalues of the Laplacian in a much more uniform manner than the 1-way reduction. In the 1-way case small eigenvalues are multiplied by small, and large by large, so the condition number explodes.

3.11 Preconditioning with Convolutional Operators

So far we have discussed algorithms for special matrices; convolution operators on periodic domains and convolution operators on special domains with boundary conditions satisfied by reflection symmetries. In this section we will briefly discuss some applications of group theory to more general matrices.

3.11.1 Matrix Multiplication by Diagonals

Our first topic is the technique of *matrix multiplication by diagonals*. This is a computational technique which has been popular for matrix-matrix and matrix-vector multiplication with sparse structured matrices, such as matrices arising from 5-point or 7-point stencils on rectangular grids, but where (unlike our previous discussions) the coefficients vary over space. The matrices are stored as diagonals together with an off-set indicating the position of the diagonal. We revisit this technique and describe it in the language of groups and want to show that book keeping in matrices stored by diagonals is simplified by the notation of finite abelian groups.

Let G be a finite abelian group and $f \in \mathbb{C}[G]$. Let $D(f)$ denote the diagonal matrix corresponding to f , i.e. the matrix such that $(D(f)g)(j) = f(j)g(j)$ for all $g \in \mathbb{C}[G]$, and as before, for $\ell \in G$, we let S_ℓ is the shift matrix $(S_\ell g)(j) = g(j - \ell)$ for $g \in \mathbb{C}[G]$. We want to develop matrix algebra expressed in terms of sums of products of shift matrices and diagonal matrices, $S_\ell D(f)$. We ask the reader to verify the following result as an exercise:

Lemma 18 *Shift matrices and diagonals can be swapped according to the rule*

$$S_\ell D(f) = D(S_\ell f) S_\ell. \quad (75)$$

Lemma 19 (Matrix Representation by Shifted Diagonals) *Any matrix $A \in \text{End}(\mathbb{C}[G])$ can be written as a sum of products of shift matrices with diagonal matrices,*

$$A = \sum_{\ell \in G} S_\ell D(a_\ell), \quad (76)$$

where and $a_\ell \in \mathbb{C}[G]$ for all $\ell \in G$.

Proof Let δ be the delta-function, which is 1 in $\mathbf{0}$ and 0 elsewhere. Let $\delta_j = S_j \delta$. For a matrix A , and $i, j \in G$ let $A_{i,j} = (A\delta_j)(i)$ be its entries by classical index notation. For A defined in (76) we compute, using Lemma 19 that

$$(A\delta_j)(i) = f_{i-j}(j).$$

(The reader should check this for a simple example such as $G = \mathbb{Z}_4$). From this we see that we can represent any matrix $A \in \text{End}(\mathbb{C}[G])$ by putting $A_{i,j}$ into $f_{i-j}(j)$, thus any A can be written in this form. \square

The following is very easily verified (check an example with $G = \mathbb{Z}_n$):

Lemma 20 (Matrix–Vector Multiplication by Diagonals) *Let $A \in \text{End}(\mathbb{C}[G])$ be represented as in (76), then for $x \in \mathbb{C}[G]$ we have*

$$Ax = \sum_{\ell \in G} S_\ell (a_\ell \bullet x),$$

where $\bullet: \mathbb{C}[G] \times \mathbb{C}[G] \rightarrow \mathbb{C}[G]$ denotes point wise product of vectors.

Theorem 19 (Matrix–Matrix Multiplication by Diagonals) *Let $A, B, C \in \text{End}(\mathbb{C}[G])$ be represented as in (76) and let $C = AB$. Then*

$$c_r = \sum_{\ell \in G} (S_{-\ell} a_{r-\ell}) \bullet b_\ell.$$

Proof We compute, setting $r = k + \ell$:

$$\begin{aligned} AB &= \sum_k S_k D(a_k) \sum_\ell S_\ell D(b_\ell) = \sum_{k,\ell} S_k S_\ell S_{-\ell} D(a_k) S_\ell D(b_\ell) \\ &= \sum_{k,\ell} S_{k+\ell} D((S_{-\ell} a_k) \bullet b_\ell) = \sum_r S_r D \left(\sum_\ell (S_{-\ell} a_{r-\ell}) \bullet b_\ell \right). \end{aligned} \quad \square$$

Multiplication by diagonals is especially attractive for matrices such as 5-point stencils etc, where the number of diagonals is small. In this case we, of course, have to compute only those c_r for which $r = j + k$, where j and k are non-zero diagonals in A and B .

3.11.2 Preconditioning

We end this section with a brief result about matrix approximation using convolutional operators. The goal of preconditioning a linear system $Ax = b$ is to find an approximation $C \approx A$ such that $Cx = b$ can be easily solved. If C is a convolutional operator, we know that this can be easily solved by Fourier analysis or cyclic reduction (or a combination of these). What is the ‘best’ approximation of a general matrix by a convolution? This does depend on the norm we use to measure closeness. The Frobenius norm gives a particularly simple answer.

Definition 46 (Frobenius Norm) For $A, B \in \text{End}(G)$, we define the Frobenius inner product

$$(A, B)_F := \text{trace}(A^h B),$$

where A^h denotes the complex conjugate and transpose of A . The Frobenius norm is

$$\|A, B\|_F = (A, B)_F^{\frac{1}{2}}.$$

Lemma 21 *The shift matrices S_j are orthogonal in the Frobenius inner-product*

$$(S_j, S_k)_F = \begin{cases} |G| & \text{if } j = k \\ 0 & \text{else} \end{cases}.$$

Proof $S_j^h S_k = S_{k-j}$, which has diagonal entries all 1 if $k=j$ and all 0 else. \square

Theorem 20 *Let $A = \sum_j S_j a_j \in \text{End}(\mathbb{C}[G])$ be represented by its diagonals $a_j \in \mathbb{C}[G]$. The best Frobenius norm approximation to A by a convolutional operator is given as*

$$C = \sum_{j \in G} c_j S_j,$$

where

$$c_j = \frac{1}{|G|} \sum_{k \in G} a_j(k) \in \mathbb{C}.$$

Proof The shift matrices $\{S_j\}_{j \in G}$ form an orthogonal basis (with coefficients in \mathbb{C}) for the convolutional operators as a subspace of $\text{End}(\mathbb{C}[G])$. Hence, we find the best approximation of $A \in \text{End}(\mathbb{C}[G])$ in the subspace of convolutional operators by projecting A orthogonally onto the subspace in Frobenius inner product:

$$C = \sum_{j \in G} \frac{(S_j, A)_F}{(S_j, S_j)_F} S_j.$$

The result follows by noting that $(S_j, A)_F = \sum_k a_j(k)$. \square

A lot more could have been said about preconditioning with convolutional operators, but time and space is limited so we leave this interesting topic at this point.

4 Domain Symmetries and Non-commutative Groups

The topic of this chapter is applications of Fourier analysis on non-commutative groups in linear algebra. In particular we will as an example discuss the computation of matrix exponentials for physical problems being symmetric with respect to a discrete non-commutative group acting upon the domain. Assuming that the domain is discretized with a symmetry respecting discretization, we will show that by a change of basis derived from the irreducible representations of the group, the operator is block diagonalized. This simplifies the computation of matrix exponentials, eigenvalue problems and the solution of linear equations. The basic mathematics behind this Chapter is *representation theory of finite groups* [15, 16, 26]. Applications of this theory in scientific computing is discussed by a number of authors, see e.g. [2, 4, 6, 9, 12]. Our exposition, based on the *group algebra* is explained in detail in [1], which is intended to be a self contained introduction to the subject.

4.1 \mathcal{G} -Equivariant Matrices

A *group* is a set \mathcal{G} with a binary operation $g, h \mapsto gh$, inverse $g \mapsto g^{-1}$ and identity element e , such that $g(ht) = (gh)t$, $eg = ge = g$ and $gg^{-1} = g^{-1}g = e$ for all $g, h, t \in \mathcal{G}$. We let $|\mathcal{G}|$ denote the number of elements in the group. Let \mathcal{I} denote the set of indices used to enumerate the nodes in the discretization of a computational domain. We say that a group \mathcal{G} *acts on* a set \mathcal{I} (from the right) if there exists a product $(i, g) \mapsto ig : \mathcal{I} \times \mathcal{G} \rightarrow \mathcal{I}$ such that

$$ie = i \quad \text{for all } i \in \mathcal{I}, \tag{77}$$

$$i(gh) = (ig)h \quad \text{for all } g, h \in \mathcal{G} \text{ and } i \in \mathcal{I}. \tag{78}$$

The map $i \mapsto ig$ is a permutation of the set \mathcal{I} , with the inverse permutation being $i \mapsto ig^{-1}$. An action partitions \mathcal{I} into disjoint *orbits*

$$\mathcal{O}_i = \{j \in \mathcal{I} : j = ig \text{ for some } g \in \mathcal{G}\}, \quad i \in \mathcal{I}.$$

We let $\mathcal{S} \subset \mathcal{I}$ denote a selection of *orbit representatives*, i.e. one element from each orbit. The action is called *transitive* if \mathcal{I} consists of just a single orbit, $|\mathcal{S}| = 1$. For any $i \in \mathcal{I}$ we let the *isotropy subgroup at i* , \mathcal{G}_i be defined as

$$\mathcal{G}_i = \{g \in \mathcal{G} : ig = i\}.$$

The action is *free* if $\mathcal{G}_i = \{e\}$ for every $i \in \mathcal{I}$, i.e., there are no fixed points under the action of \mathcal{G} .

Definition 47 A matrix $A \in \mathbb{C}^{\mathcal{I} \times \mathcal{I}}$, is \mathcal{G} -equivariant if

$$A_{i,j} = A_{ig,jg} \quad \text{for all } i, j \in \mathcal{I} \text{ and all } g \in \mathcal{G}. \quad (79)$$

The definition is motivated by the result that if \mathcal{L} is a linear differential operator commuting with a group of domain symmetries \mathcal{G} , and if we can find a set of discretization nodes \mathcal{I} such that every $g \in \mathcal{G}$ acts on \mathcal{I} as a permutation $i \mapsto ig$, then \mathcal{L} can be discretized as a \mathcal{G} -equivariant matrix A , see [4, 6].

4.2 The Group Algebra

We will establish that \mathcal{G} equivariant matrices are associated with (scalar or block) convolutional operators in the *group algebra*.

Definition 48 The *group algebra* $\mathbb{C}\mathcal{G}$ is the complex vectorspace $\mathbb{C}\mathcal{G}$ where each $g \in \mathcal{G}$ corresponds to a basis vector $\mathbf{g} \in \mathbb{C}\mathcal{G}$. A vector $a \in \mathbb{C}\mathcal{G}$ can be written as

$$a = \sum_{g \in \mathcal{G}} a(g)\mathbf{g} \quad \text{where } a(g) \in \mathbb{C}.$$

The convolution product $* : \mathbb{C}\mathcal{G} \times \mathbb{C}\mathcal{G} \rightarrow \mathbb{C}\mathcal{G}$ is induced from the product in \mathcal{G} as follows. For basis vectors \mathbf{g}, \mathbf{h} , we set $\mathbf{g} * \mathbf{h} \equiv \mathbf{g}\mathbf{h}$, and in general if $a = \sum_{g \in \mathcal{G}} a(g)\mathbf{g}$ and $b = \sum_{h \in \mathcal{G}} b(h)\mathbf{h}$, then

$$a * b = \left(\sum_{g \in \mathcal{G}} a(g)\mathbf{g} \right) * \left(\sum_{h \in \mathcal{G}} b(h)\mathbf{h} \right) = \sum_{g,h \in \mathcal{G}} a(g)b(h)(\mathbf{g}\mathbf{h}) = \sum_{g \in \mathcal{G}} (a * b)(g)\mathbf{g},$$

where

$$(a * b)(g) = \sum_{h \in \mathcal{G}} a(gh^{-1})b(h) = \sum_{h \in \mathcal{G}} a(h)b(h^{-1}g). \quad (80)$$

Consider a \mathcal{G} -equivariant $\mathbf{A} \in \mathbb{C}^{n \times n}$ in the case where \mathcal{G} acts freely and transitively on \mathcal{I} . In this case there is only one orbit of size $|\mathcal{G}|$ and hence \mathcal{I} may be identified with \mathcal{G} . Corresponding to \mathbf{A} there is a unique $A \in \mathbb{C}\mathcal{G}$, given as $A = \sum_{g \in \mathcal{G}} A(g)g$, where A is the first column of \mathbf{A} , i.e.,

$$A(gh^{-1}) = \mathbf{A}_{gh^{-1}, e} = \mathbf{A}_{g, h}. \quad (81)$$

Similarly, any vector $\mathbf{x} \in \mathbb{C}^n$ corresponds uniquely to $x = \sum_{g \in \mathcal{G}} x(g)g \in \mathbb{C}\mathcal{G}$, where $x(g) = \mathbf{x}_g$ for all $g \in \mathcal{G}$. Consider the matrix vector product:

$$(\mathbf{Ax})_g = \sum_{h \in \mathcal{G}} \mathbf{A}_{g, h} \mathbf{x}_h = \sum_{h \in \mathcal{G}} A(gh^{-1})x(h) = (A * x)(g).$$

If \mathbf{A} and \mathbf{B} are two equivariant matrices, then \mathbf{AB} is the equivariant matrix where the first column is given as

$$(\mathbf{AB})_{g, e} = \sum_{h \in \mathcal{G}} \mathbf{A}_{g, h} \mathbf{B}_{h, e} = \sum_{h \in \mathcal{G}} A(gh^{-1})B(h) = (A * B)(g).$$

We have shown that *if \mathcal{G} acts freely and transitively, then the algebra of \mathcal{G} -equivariant matrices acting on \mathbb{C}^n is isomorphic to the group algebra $\mathbb{C}\mathcal{G}$ acting on itself by convolutions from the left.*

In the case where \mathbf{A} is \mathcal{G} -equivariant w.r.t. a free, but not transitive, action of \mathcal{G} on \mathcal{I} , we need a block version of the above theory. Let $\mathbb{C}^{m \times \ell}\mathcal{G} \equiv \mathbb{C}^{m \times \ell} \otimes \mathbb{C}\mathcal{G}$ denote the space of vectors consisting of $|\mathcal{G}|$ matrix blocks, each block of size $m \times \ell$, thus $A \in \mathbb{C}^{m \times \ell}\mathcal{G}$ can be written as

$$A = \sum_{g \in \mathcal{G}} A(g) \otimes g \quad \text{where } A(g) \in \mathbb{C}^{m \times \ell}. \quad (82)$$

The convolution product (80) generalizes to a block convolution $* : \mathbb{C}^{m \times \ell}\mathcal{G} \times \mathbb{C}^{\ell \times k}\mathcal{G} \rightarrow \mathbb{C}^{m \times k}\mathcal{G}$ given as

$$A * B = \left(\sum_{g \in \mathcal{G}} A(g) \otimes g \right) * \left(\sum_{h \in \mathcal{G}} B(h) \otimes h \right) = \sum_{g \in \mathcal{G}} (A * B)(g) \otimes g,$$

where

$$(A * B)(g) = \sum_{h \in \mathcal{G}} A(gh^{-1})B(h) = \sum_{h \in \mathcal{G}} A(h)B(h^{-1}g), \quad (83)$$

and $A(h)B(h^{-1}g)$ denotes a matrix product.

If the action of \mathcal{G} on \mathcal{I} is free, but not transitive, then \mathcal{I} split in m orbits, each of size $|\mathcal{G}|$. We let \mathcal{S} denote a selection of one representative from each orbit. We will establish an isomorphism between the algebra of \mathcal{G} -equivariant matrices acting on \mathbb{C}^n and the block-convolution algebra $\mathbb{C}^{m \times m}\mathcal{G}$ acting on $\mathbb{C}^m\mathcal{G}$. We define the mappings $\mu : \mathbb{C}^n \rightarrow \mathbb{C}^m\mathcal{G}$, $v : \mathbb{C}^{n \times n} \rightarrow \mathbb{C}^{m \times m}\mathcal{G}$ as:

$$\mu(\mathbf{y})_i(g) = y_i(g) = \mathbf{y}_{ig} \quad \forall i \in \mathcal{S}, g \in \mathcal{G}, \quad (84)$$

$$v(\mathbf{A})_{ij}(g) = A_{ij}(g) = \mathbf{A}_{ig,j} \quad \forall i, j \in \mathcal{S}, g \in \mathcal{G}. \quad (85)$$

In [1] we show:

Proposition 1 *Let \mathcal{G} act freely on \mathcal{I} . Then μ is invertible and v is invertible on the subspace of \mathcal{G} -equivariant matrices. Furthermore, if $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n \times n}$ are \mathcal{G} -equivariant, and $\mathbf{y} \in \mathbb{C}^n$, then*

$$\mu(\mathbf{Ay}) = v(\mathbf{A}) * \mu(\mathbf{y}), \quad (86)$$

$$v(\mathbf{AB}) = v(\mathbf{A}) * v(\mathbf{B}). \quad (87)$$

To complete the connection between \mathcal{G} -equivariance and block convolutions, we need to address the general case where the action is not free, hence some of the orbits in \mathcal{I} have reduced size. One way to treat this case is to duplicate the nodes with non-trivial isotropy subgroups, thus a point $j \in \mathcal{I}$ is considered to be $|\mathcal{G}_j|$ identical points, and the action is extended to a free action on this extended space. Equivariant matrices on the original space is extended by duplicating the matrix entries, and scaled according to the size of the isotropy. We define

$$\mu(\mathbf{x})_i(g) = x_i(g) = \mathbf{x}_{ig} \quad \forall i \in \mathcal{S}, g \in \mathcal{G}, \quad (88)$$

$$v(\mathbf{A})_{ij}(g) = A_{ij}(g) = \frac{1}{|\mathcal{G}_j|} \mathbf{A}_{ig,j} \quad \forall i, j \in \mathcal{S}, g \in \mathcal{G}. \quad (89)$$

With these definitions it can be shown that (86)–(87) still hold. It should be noted that μ and v are no longer invertible, and the extended block convolutional operator $v(\mathbf{A})$ becomes singular. This poses no problems for the computation of exponentials since this is a forward computation. Thus we just exponentiate the block convolutional operator and restrict the result back to the original space. However, for inverse computations such as solving linear systems, the characterization of the image of μ and v as a subspaces of $\mathbb{C}^m\mathcal{G}$ and $\mathbb{C}^{m \times m}\mathcal{G}$ is an important issue for finding the correct solution [1, 3].

4.3 The Generalized Fourier Transform (GFT)

So far we have argued that a differential operator with spatial symmetries becomes a \mathcal{G} -equivariant matrix under discretization, which again can be represented as a block convolutional operator. In this section we will show how convolutional operators are block diagonalized by a Fourier transform on \mathcal{G} . This is the central part of Frobenius' theory of group representations from 1897–1899. We recommend the monographs [10, 15, 16, 26] as introductions to representation theory with applications.

Definition 49 A d -dimensional group representation is a map $R : \mathcal{G} \rightarrow \mathbb{C}^{d \times d}$ such that

$$R(gh) = R(g)R(h) \quad \text{for all } g, h \in \mathcal{G}. \quad (90)$$

Generalizing the definition of *Fourier coefficients* we define for any $A \in \mathbb{C}^{m \times k}\mathcal{G}$ and any d -dimensional representation R a matrix $\hat{A}(R) \in \mathbb{C}^{m \times k} \otimes \mathbb{C}^{d \times d}$ as:

$$\hat{A}(R) = \sum_{g \in \mathcal{G}} A(g) \otimes R(g). \quad (91)$$

Proposition 2 (The Convolution Theorem) *For any $A \in \mathbb{C}^{m \times k}\mathcal{G}$, $B \in \mathbb{C}^{k \times \ell}\mathcal{G}$ and any representation R we have*

$$\widehat{(A * B)}(R) = \hat{A}(R)\hat{B}(R). \quad (92)$$

Proof The statement follows from

$$\begin{aligned} \hat{A}(R)\hat{B}(R) &= \left(\sum_{g \in \mathcal{G}} A(g) \otimes R(g) \right) \left(\sum_{h \in \mathcal{G}} B(h) \otimes R(h) \right) \\ &= \sum_{g, h \in \mathcal{G}} A(g)B(h) \otimes R(g)R(h) = \sum_{g, h \in \mathcal{G}} A(g)B(h) \otimes R(gh) \\ &= \sum_{g, h \in \mathcal{G}} A(gh^{-1})B(h) \otimes R(g) = \widehat{(A * B)}(R). \end{aligned}$$

Let d_R denote the dimension of the representation. For use in practical computations, it is important that $A * B$ can be recovered by knowing $\widehat{(A * B)}(R)$ for a suitable selection of representations, and furthermore that their dimensions d_R are as small as possible. Note that if R is a representation and $X \in \mathbb{C}^{d_R \times d_R}$ is non-singular, then also $\tilde{R}(g) = XR(g)X^{-1}$ is a representation. We say that R and \tilde{R} are equivalent representations. If there exists a similarity transform $\tilde{R}(g) = XR(g)X^{-1}$ such that $\tilde{R}(g)$ has a block diagonal structure, independent of $g \in \mathcal{G}$, then R is called *reducible*, otherwise it is *irreducible*.

Theorem 21 (Frobenius) *For any finite group \mathcal{G} there exists a complete list \mathcal{R} of non-equivalent irreducible representations such that*

$$\sum_{R \in \mathcal{R}} d_R^2 = |\mathcal{G}|.$$

Defining the GFT for $a \in \mathcal{G}$ Algebra as

$$\hat{a}(R) = \sum_{g \in \mathcal{G}} a(g)R(g) \quad \text{for every } R \in \mathcal{R}, \quad (93)$$

we may recover a by the inverse GFT (IGFT):

$$a(g) = \frac{1}{|\mathcal{G}|} \sum_{R \in \mathcal{R}} d_R \text{trace}(R(g^{-1})\hat{a}(R)). \quad (94)$$

For the block transform of $A \in \mathbb{C}^{m \times k}\mathcal{G}$ given in (91), the GFT and the IGFT are given componentwise as

$$\hat{A}_{i,j}(R) = \sum_{g \in \mathcal{G}} A_{i,j}(g)R(g) \in \mathbb{C}^{d_R \times d_R}, \quad (95)$$

$$A_{i,j}(g) = \frac{1}{|\mathcal{G}|} \sum_{R \in \mathcal{R}} d_R \text{trace}(R(g^{-1})\hat{A}_{i,j}(R)). \quad (96)$$

Complete lists of irreducible representations for a selection of common groups are found in [16].

4.4 Applications to the Matrix Exponential

We have seen that via the GFT, any \mathcal{G} -equivariant matrix is block diagonalized. Corresponding to an irreducible representation R , we obtain a matrix block $\hat{A}(R)$ of size $md_R \times md_R$, where m is the number of orbits in \mathcal{I} and d_R the size of the representation. Let W_{direct} denote the computational work, in terms of floating point operations, for computing the matrix exponential on the original data A , and let W_{fspace} be the cost of doing the same algorithm on the corresponding block diagonal GFT transformed data \hat{A} . Thus $W_{\text{direct}} = c(m|\mathcal{G}|)^3 = cm^3 (\sum_{R \in \mathcal{R}} d_R^2)^3$, $W_{\text{fspace}} = cm^3 \sum_{R \in \mathcal{R}} d_R^3$ and the ratio becomes

$$\mathcal{O}(n^3) : \quad W_{\text{direct}}/W_{\text{fspace}} = \left(\sum_{R \in \mathcal{R}} d_R^2 \right)^3 / \sum_{R \in \mathcal{R}} d_R^3.$$

Table 1 Gain in computational complexity for matrix exponential via GFT

Domain	\mathcal{G}	$ \mathcal{G} $	$\{d_R\}_{R \in \mathcal{R}}$	$W_{\text{direct}}/W_{\text{fspace}}$
Triangle	\mathcal{D}_3	6	{1, 1, 2}	21.6
Tetrahedron	\mathcal{S}_4	24	{1, 1, 2, 3, 3}	216
Cube	$\mathcal{S}_4 \times \mathcal{C}_2$	48	{1, 1, 1, 1, 2, 2, 3, 3, 3, 3}	864
Icosahedron	$\mathcal{A}_5 \times \mathcal{C}_2$	120	{1, 1, 3, 3, 3, 3, 4, 4, 5, 5}	3541

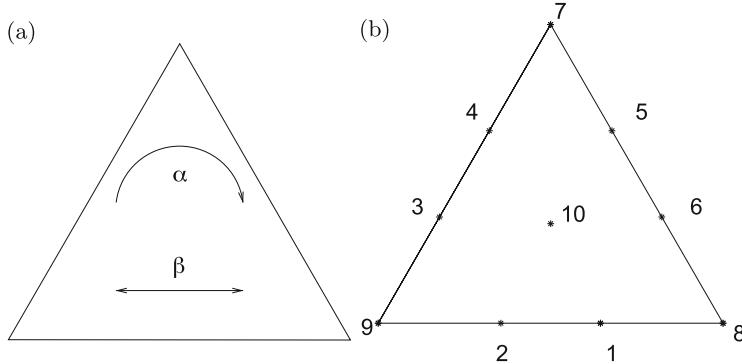
**Fig. 3** Equilateral triangle with a symmetry preserving set of 10 nodes (a) Generator of symmetry group \mathcal{D}_3 (b) Discretisation respecting symmetries

Table 1 lists this factor for the symmetries of the triangle, the tetrahedron, the 3D cube and the maximally symmetric discretization of a 3D sphere (icosahedral symmetry with reflections).

The cost of computing the GFT is not taken into account in this estimate. There exists fast GFT algorithms of complexity $\mathcal{O}(|\mathcal{G}| \log^\ell(|\mathcal{G}|))$ for a number of groups, but even if we use a slow transform of complexity $\mathcal{O}(|\mathcal{G}^2|)$, the total cost of the GFT becomes just $\mathcal{O}(m^2|\mathcal{G}|^2)$, which is much less than W_{fspace} .

4.4.1 Example: Equilateral Triangle

The smallest noncommutative group is \mathcal{D}_3 , the symmetries of an equilateral triangle. There are six linear transformations that map the triangle onto itself, three pure rotations and three rotations combined with reflections. In Fig. 3a we indicate the two generators α (rotation 120° clockwise) and β (right-left reflection). These satisfy the algebraic relations $\alpha^3 = \beta^2 = e$, $\beta\alpha\beta = \alpha^{-1}$, where e denotes the identity transform. The whole group is $\mathcal{D}_3 = \{e, \alpha, \alpha^2, \beta, \alpha\beta, \alpha^2\beta\}$.

Given an elliptic operator \mathcal{L} on the triangle such that $\mathcal{L}(u \circ \alpha) = \mathcal{L}(u) \circ \alpha$ and $\mathcal{L}(u \circ \beta) = \mathcal{L}(u) \circ \beta$ for any u satisfying the appropriate boundary conditions on the triangle, let the domain be discretized with a *symmetry respecting discretization*, see Fig. 3b. In this example we consider a finite difference discretization represented by the nodes $\mathcal{I} = \{1, 2, \dots, 10\}$, such that both α and β map nodes to nodes. In

Table 2 A complete list of irreducible representations for \mathcal{D}_3

	α	β
ρ_0	1	1
ρ_1	1	-1
ρ_2	$\begin{pmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

finite element discretizations one would use basis functions mapped to other basis functions by the symmetries. We define the action of \mathcal{D}_3 on \mathcal{I} as

$$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)\alpha = (5, 6, 1, 2, 3, 4, 9, 7, 8, 10),$$

$$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)\beta = (2, 1, 6, 5, 4, 3, 7, 9, 8, 10),$$

and extend to all of \mathcal{D}_3 using (78). As orbit representatives, we may pick $\mathcal{S} = \{1, 7, 10\}$. The action of the symmetry group is free on the orbit $\mathcal{O}_1 = \{1, 2, 3, 4, 5, 6\}$, while the points in the orbit $\mathcal{O}_7 = \{7, 8, 9\}$ have isotropy subgroups of size 2, and finally $\mathcal{O}_{10} = \{10\}$ has isotropy of size 6.

The operator \mathcal{L} is discretized as a matrix $\mathbf{A} \in \mathbb{C}^{10 \times 10}$ satisfying the equivariances $\mathbf{A}_{ig,jg} = \mathbf{A}_{i,j}$ for $g \in \{\alpha, \beta\}$ and $i, j \in \mathcal{S}$. Thus we have e.g. $\mathbf{A}_{1,6} = \mathbf{A}_{3,2} = \mathbf{A}_{5,4} = \mathbf{A}_{4,5} = \mathbf{A}_{2,3} = \mathbf{A}_{6,1}$.

\mathcal{D}_3 has three irreducible representations given in Table 2 [extended to the whole group using (90)]. To compute $\exp(\mathbf{A})$, we find $\mathbf{A} = v(\mathbf{A}) \in \mathbb{C}^{3 \times 3}\mathcal{G}$ from (89) and find $\hat{\mathbf{A}} = \text{GFT}(\mathbf{A})$ from (95). The transformed matrix $\hat{\mathbf{A}}$ has three blocks, $\hat{\mathbf{A}}(\rho_0), \hat{\mathbf{A}}(\rho_1) \in \mathbb{C}^{m \times m}$ and $\hat{\mathbf{A}}(\rho_2) \in \mathbb{C}^{m \times m} \otimes \mathbb{C}^{2 \times 2} \simeq \mathbb{C}^{2m \times 2m}$, where $m = 3$ is the number of orbits. We exponentiate each of these blocks, and find the components of $\exp(\mathbf{A})$ using the Inverse GFT (96).

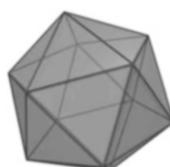
We should remark that in Lie group integrators, it is usually more important to compute $y = \exp(\mathbf{A}) \cdot x$ for some vector x . In this case, we compute $\hat{y}(\rho_i) = \exp(\hat{\mathbf{A}}(\rho_i)) \cdot \hat{x}(\rho_i)$, and recover y by Inverse GFT. Note that $\hat{x}(\rho_2), \hat{y}(\rho_2) \in \mathbb{C}^m \otimes \mathbb{C}^{2 \times 2} \simeq \mathbb{C}^{2m \times 2}$.

4.4.2 Example: Icosahedral Symmetry

As a second example illustrating the general theory, we solve the simple heat equation

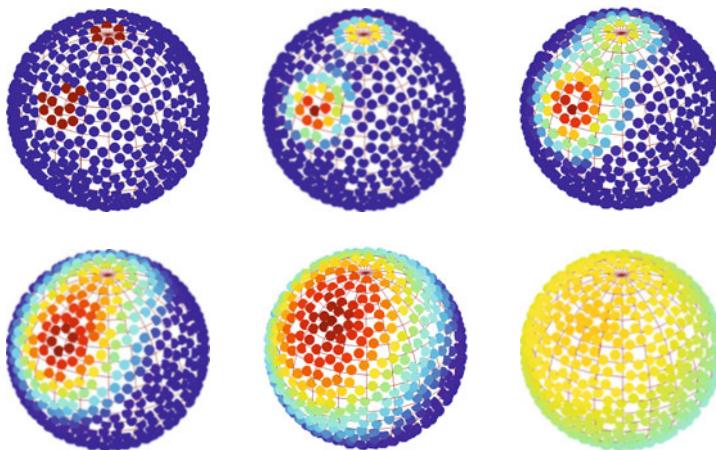
$$u_t = \nabla^2 u$$

on the surface of a unit sphere.



The sphere is divided into 20 equilateral triangles, and each triangle subdivided in a finite difference mesh respecting all the 120 symmetries of the full icosahedral symmetry group (including reflections). To understand this group, it is useful to realize that five tetrahedra can be simultaneously embedded in the icosahedron, so that the 20 triangles correspond to the in total 20 corners of these five tetrahedra. From this one sees that the icosahedral rotation group is isomorphic to A_5 , the group of all 60 even permutations of the five tetrahedra. The 3-D reflection matrix $-I$ obviously commutes with any 3-D rotation, and hence we realize that the full icosahedral group is isomorphic to the direct product $C_2 \times A_5$, where $C_2 = \{1, -1\}$. The irreducible representations of A_5 , listed in Lomont have dimensions $\{1, 3, 3, 4, 5\}$, and the representations of the full icosahedral group are found by taking tensor products of these with the two 1-dimensional representations of C_2 . The fact that the full icosahedral group is a direct product is also utilized in faster computation of the GFT. This is, however, not of major importance, since the cost of the GFT in any case is much less than the cost of the matrix exponential.

The figures below show the solution of the heat equation at times 0, 2, 5, 10, 25 and 100. The initial condition consists of two located heat sources in the northern hemisphere.



5 Concluding Remarks

We have in these lectures presented the basic concepts of group theory in a setting aimed at understanding computational algorithms. Some applications within computational mathematics have been discussed in detail, others in a more sketchy manner and many topics have been omitted altogether. Among the omissions, we would in particular point to the theory of multivariate Chebyshev approximations,

a beautiful application of group theory which originate from the study of kaleidoscopic reflection groups (Coxeter groups), and which has connections to many areas of mathematics, the representation theory of Lie groups in particular. The multivariate Chebyshev polynomials share the excellent approximation properties of the classical univariate case, and the multivariate polynomials are defined on domains that are related to simplexes in higher dimensions.

We have discussed Fourier analysis on abelian groups and on finite non-abelian groups. The next steps along this line is the Fourier analysis on compact Lie groups, where the fundamental Peter–Weyl theorem states that the countably infinite list of non-equivalent irreducible representations gives a complete orthogonal basis for $L^2(G)$. Certain non-compact groups (the unimodular groups) such as the Euclidean group of rigid motions in \mathbb{R}^n can be derived from the compact case and is of significant interest in image and signal processing.

Finally, we would like to mention the topic of time integration of differential equations evolving on manifolds. The so-called *Lie group* integrators advance the solution by computing the action of a Lie group on the domain. This topic has been developed in substantial detail over the last two decades and has lead to many theoretical insights and practical computational algorithms [14].

Acknowledgements I would like to express a deep gratitude towards CIME and the organisers of this summer school for inviting me to present these lectures and for their patience with me during the tortuous process of writing the lecture notes. Also, I would like to thank Ulrich von der Ohe for his careful reading and commenting upon the manuscript.

References

1. K. Åhlander, H. Munthe-Kaas, Applications of the Generalized Fourier Transform in numerical linear algebra. *BIT* **45**(4), 819–850 (2005)
2. E.L. Allgower, K. Böhmer, K. Georg, R. Miranda, Exploiting symmetry in boundary element methods. *SIAM J. Numer. Anal.* **29**, 534–552 (1992)
3. E.L. Allgower, K. Georg, R. Miranda, Exploiting permutation symmetry with fixed points in linear equations, in *Lectures in Applied Mathematics*, vol. 29, ed. by E.L. Allgower, K. Georg, R. Miranda (American Mathematical Society, Providence, RI, 1993), pp. 23–36
4. E.L. Allgower, K. Georg, R. Miranda, J. Tausch, Numerical exploitation of equivariance. *Z. Angew. Math. Mech.* **78**, 185–201 (1998)
5. L. Auslander, R. Tolimieri, Is computing with the finite Fourier transform pure or applied mathematics? *Not. AMS* **1**(6), 847–897 (1979)
6. A. Bossavit, Symmetry, groups, and boundary value problems. A progressive introduction to noncommutative harmonic analysis of partial differential equations in domains with geometrical symmetry. *Comput. Methods Appl. Mech. Eng.* **56**, 167–215 (1986)
7. S.H. Christiansen, H.Z. Munthe-Kaas, B. Owren, Topics in structure-preserving discretization. *Acta Numer.* **20**(1), 1–119 (2011)
8. J.H. Conway, N.J.A. Sloane, E. Bannai, *Sphere Packings, Lattices, and Groups*, vol. 290 (Springer, Berlin, 1999)
9. C.C. Douglas, J. Mandel, Abstract theory for the domain reduction method. *Computing* **48**, 73–96 (1992)

10. A.F. Fässler, E. Stiefel, *Group Theoretical Methods and Their Applications* (Birkhäuser, Boston, 1992)
11. C. Gasquet, P. Witomski, *Fourier Analysis and Applications: Filtering, Numerical Computation, Wavelets*, vol. 30 (Springer Science & Business Media, Berlin, 2013)
12. K. Georg, R. Miranda, Exploiting symmetry in solving linear equations, in *Bifurcation and Symmetry*, vol. 104, ed. by E.L. Allgower, K. Böhmer, M. Golubitsky. International Series of Numerical Mathematics (Birkhäuser, Basel, 1992), pp. 157–168
13. M.E. Hoffman, W.D. Withers, Generalized Chebyshev polynomials associated with affine Weyl groups. *Trans. AMS* **308**(1), 91–104 (1988)
14. A. Iserles, H. Munthe-Kaas, S.P. Nørsett, A. Zanna, *Lie-group methods*. *Acta Numerica*, vol. 9 (Cambridge University Press, Cambridge, 2000), pp. 215–365
15. G. James, M. Liebeck, *Representations and Characters of Groups*, 2nd edn. (Cambridge University Press, Cambridge, 2001). ISBN 052100392X
16. J.S. Lomont, *Applications of Finite Groups* (Academic, New York, 1959)
17. S. Mac Lane, *Categories for the Working Mathematician*, vol. 5 (Springer Science & Business Media, Berlin, 2013)
18. H. Munthe-Kaas, Symmetric FFTs; a general approach. Technical Report, NTNU, Trondheim, 1989. Available at: <http://hans.munthe-kaas.no>
19. H. Munthe-Kaas, Topics in linear algebra for vector- and parallel computers. Ph.D. thesis, Norwegian University of Science and Technology (NTNU), 1989
20. H.Z. Munthe-Kaas, On group Fourier analysis and symmetry preserving discretizations of PDEs. *J. Phys. A Math. Gen.* **39**, 5563 (2006)
21. H. Munthe-Kaas, T. Sørevik, Multidimensional pseudo-spectral methods on lattice grids. *Appl. Numer. Math.* **62**(3), 155–165 (2012)
22. H.Z. Munthe-Kaas, M. Nome, B.N. Ryland, Through the kaleidoscope; symmetries, groups and Chebyshev approximations from a computational point of view, in *Foundations of Computational Mathematics, Budapest 2011*. London Mathematical Society Lecture Notes Series, vol. 403 (Cambridge University Press, Cambridge, 2013), pp. 188–229
23. M.S. Osborne, On the Schwartz-Bruhat space and the Paley-Wiener theorem for locally compact Abelian groups. *J. Funct. Anal.* **19**(1), 40–49 (1975)
24. W. Rudin, *Fourier Analysis on Groups*, vol. 12 (Wiley-Interscience, New York, 1990)
25. B.N. Ryland, H.Z. Munthe-Kaas, On multivariate Chebyshev polynomials and spectral approximations on triangles, in *Spectral and High Order Methods for Partial Differential Equations*, vol. 76, ed. by J.S. Hesthaven, E.M. Rønquist. Lecture Notes in Computer Science and Engineering (Springer, Berlin, 2011), pp. 19–41
26. J.P. Serre, *Linear Representations of Finite Groups* (Springer, Berlin, 1977). ISBN 0387901906
27. R.J. Stanton, P.A. Tomas, Polyhedral summability of Fourier series on compact Lie groups. *Am. J. Math.* **100**(3), 477–493 (1978)
28. P.N. Swarztrauber, The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle. *SIAM Rev.* **19**(3), 490–501 (1977)
29. S. Thangavelu, *Harmonic Analysis on the Heisenberg Group*, vol. 159 (Birkhauser, Basel, 2012)
30. G. Travaglini, Polyhedral summability of multiple Fourier series. *Colloq. Math.* **65**, 103–116 (1993)
31. Wikipedia. Smith normal form — Wikipedia, the free encyclopedia (2015)

Editors in Chief: J.-M. Morel, B. Teissier;

Editorial Policy

1. Lecture Notes aim to report new developments in all areas of mathematics and their applications – quickly, informally and at a high level. Mathematical texts analysing new developments in modelling and numerical simulation are welcome.

Manuscripts should be reasonably self-contained and rounded off. Thus they may, and often will, present not only results of the author but also related work by other people. They may be based on specialised lecture courses. Furthermore, the manuscripts should provide sufficient motivation, examples and applications. This clearly distinguishes Lecture Notes from journal articles or technical reports which normally are very concise. Articles intended for a journal but too long to be accepted by most journals, usually do not have this “lecture notes” character. For similar reasons it is unusual for doctoral theses to be accepted for the Lecture Notes series, though habilitation theses may be appropriate.

2. Besides monographs, multi-author manuscripts resulting from SUMMER SCHOOLS or similar INTENSIVE COURSES are welcome, provided their objective was held to present an active mathematical topic to an audience at the beginning or intermediate graduate level (a list of participants should be provided).

The resulting manuscript should not be just a collection of course notes, but should require advance planning and coordination among the main lecturers. The subject matter should dictate the structure of the book. This structure should be motivated and explained in a scientific introduction, and the notation, references, index and formulation of results should be, if possible, unified by the editors. Each contribution should have an abstract and an introduction referring to the other contributions. In other words, more preparatory work must go into a multi-authored volume than simply assembling a disparate collection of papers, communicated at the event.

3. Manuscripts should be submitted either online at www.editorialmanager.com/lnm to Springer’s mathematics editorial in Heidelberg, or electronically to one of the series editors. Authors should be aware that incomplete or insufficiently close-to-final manuscripts almost always result in longer refereeing times and nevertheless unclear referees’ recommendations, making further refereeing of a final draft necessary. The strict minimum amount of material that will be considered should include a detailed outline describing the planned contents of each chapter, a bibliography and several sample chapters. Parallel submission of a manuscript to another publisher while under consideration for LNM is not acceptable and can lead to rejection.

4. In general, **monographs** will be sent out to at least 2 external referees for evaluation.

A final decision to publish can be made only on the basis of the complete manuscript, however a refereeing process leading to a preliminary decision can be based on a pre-final or incomplete manuscript.

Volume Editors of **multi-author works** are expected to arrange for the refereeing, to the usual scientific standards, of the individual contributions. If the resulting reports can be

forwarded to the LNM Editorial Board, this is very helpful. If no reports are forwarded or if other questions remain unclear in respect of homogeneity etc, the series editors may wish to consult external referees for an overall evaluation of the volume.

5. Manuscripts should in general be submitted in English. Final manuscripts should contain at least 100 pages of mathematical text and should always include
 - a table of contents;
 - an informative introduction, with adequate motivation and perhaps some historical remarks: it should be accessible to a reader not intimately familiar with the topic treated;
 - a subject index: as a rule this is genuinely helpful for the reader.
 - For evaluation purposes, manuscripts should be submitted as pdf files.
6. Careful preparation of the manuscripts will help keep production time short besides ensuring satisfactory appearance of the finished book in print and online. After acceptance of the manuscript authors will be asked to prepare the final LaTeX source files (see LaTeX templates online: <https://www.springer.com/gb/authors-editors/book-authors-editors/manuscriptpreparation/5636>) plus the corresponding pdf- or zipped ps-file. The LaTeX source files are essential for producing the full-text online version of the book, see <http://link.springer.com/bookseries/304> for the existing online volumes of LNM). The technical production of a Lecture Notes volume takes approximately 12 weeks. Additional instructions, if necessary, are available on request from; lnm@springer.com.
7. Authors receive a total of 30 free copies of their volume and free access to their book on SpringerLink, but no royalties. They are entitled to a discount of 33.3 % on the price of Springer books purchased for their personal use, if ordering directly from Springer.
8. Commitment to publish is made by a *Publishing Agreement*; contributing authors of multiauthor books are requested to sign a *Consent to Publish form*. Springer-Verlag registers the copyright for each volume. Authors are free to reuse material contained in their LNM volumes in later publications: a brief written (or e-mail) request for formal permission is sufficient.

Addresses:

Professor Jean-Michel Morel, CMLA, École Normale Supérieure de Cachan, France
E-mail: moreljeanmichel@gmail.com

Professor Bernard Teissier, Equipe Géométrie et Dynamique,
Institut de Mathématiques de Jussieu – Paris Rive Gauche, Paris, France
E-mail: bernard.teissier@imj-prg.fr

Springer: Ute McCrory, Mathematics, Heidelberg, Germany,
E-mail: lnm@springer.com