Thèse n° 9967

Fast deterministic and randomized algorithms for low-rank approximation, matrix functions, and trace estimation

Présentée le 3 juin 2022

Faculté des sciences de base
Algorithmes numériques et calcul haute performance - Chaire CADMOS
Programme doctoral en mathématiques

pour l'obtention du grade de Docteur ès Sciences

par

Alice CORTINOVIS

Acceptée sur proposition du jury

Prof. F. Eisenbrand, président du jury
Prof. D. Kressner, directeur de thèse
Prof. M. Benzi, rapporteur
Prof. I. Ipsen, rapporteuse
Prof. F. Nobile, rapporteur

École
polytechnique
fédérale
de Lausanne

2022

Acknowledgements

First and foremost, I would like to sincerely thank my supervisor, Prof. Daniel Kressner, for giving me the opportunity to pursue a PhD in his group and for proposing many stimulating problems to work on. I am grateful for the time, the support, and the encouragement that he gave me. I have benefited a lot from Daniel's vast numerical linear algebra expertise, as well as from his advice on mathematical writing and research presentations.

I am grateful to the Jury members of my private defense – Prof. Michele Benzi, Prof. Ilse Ipsen, and Prof. Fabio Nobile – for their detailed feedback and useful comments that helped me clarifying some parts of this thesis. I would also like to thank Prof. Friedrich Eisenbrand for agreeing to preside the private defense.

I would like to thank the current and past members of the ANCHP group at EPFL, with whom I enjoyed many lunches, talks, and jokes: Axel, Christoph, David, Francesco, Gianluca, Haoze, Hysan, Ivana, Junli, Kathryn, Lana, Margherita, and Yuxin. A very special thanks goes to Stefano Massei: Part of the research contained in this thesis is based on joint work with him, and he has been an incredible mentor and friend during these years. I am also grateful to the present and past members of MATHICSE for making the second floor such a nice place to stay.

In Fall 2021 I had the opportunity to spend two months at the University of Oxford: Prof. Yuji Nakatsukasa has been a great host and I appreciated the many discussion on "randomized stuff" we had. I would also like to thank Maike for making me feel so welcomed in Oxford.

Outside EPFL, I enjoyed many beautiful hikes and dinners with my friends, among whom Ale, Andrea, Cate, Ceci, Eli, Leo, Luca C., Luca P., Nicolino, Riccardo, Silja, and

Stefano S. A special thanks goes to Ondine, my wonderful flatmate, and Giada, with whom I have been sharing my maths journey for 10+ years, for all the emotional support in the good and bad days during these years.

Finally, I would like to express my warmest thanks to my family – mum and dad, Irene, Enrico, and my grandmothers Rita and Sira – for always supporting me even when my life choices bring me away from home.

My research activity was supported by the SNSF research project "Fast algorithms from low-rank updates", with grant number 200020_178806. This support is gratefully acknowledged.

Lausanne, 3 June 2022 Alice Cortinovis

Abstract

In this thesis we propose and analyze algorithms for some numerical linear algebra tasks: finding low-rank approximations of matrices, computing matrix functions, and estimating the trace of matrices.

In the first part, we consider algorithms for building low-rank approximations of a matrix from some rows or columns of the matrix itself. We prove a priori error bounds for a greedy algorithm for *cross approximation* and we develop a faster and more efficient variant of an existing algorithm for column subset selection. Moreover, we present a new deterministic polynomial-time algorithm that gives a cross approximation which is quasi-optimal in the Frobenius norm.

The second part of the thesis is concerned with matrix functions. We develop a divide-and-conquer algorithm for computing functions of matrices that are banded, hierarchically semiseparable, or have some other off-diagonal low-rank structure. An important building block of our approach is an existing algorithm for updating the function of a matrix that undergoes a low-rank modification (update), for which we present new convergence results. The convergence analysis of our divide-and-conquer algorithm is related to polynomial or rational approximation of the function.

In the third part we consider the problem of approximating the trace of a matrix which is available indirectly, through matrix-vector multiplications. We analyze a stochastic algorithm, the Hutchinson trace estimator, for which we prove tail bounds for symmetric (indefinite) matrices. Then we apply our results to the computation of the (log)determinants of symmetric positive definite matrices.

Keywords. Low-rank approximation, column subset selection, low-rank updates, Krylov subspace methods, matrix functions, banded matrices, hierarchically semiseparable matrices, trace estimation, determinant.

Résumé

Dans cette thèse, nous proposons et analysons des algorithmes pour les problèmes d'algèbre linéaire numérique suivants : trouver des approximations de rang faible de matrices, calculer des fonctions matricielles et estimer la trace de matrices.

Dans la première partie, nous considérons des algorithmes pour construire des approximations de rang faible d'une matrice à partir de certaines lignes ou colonnes de cette matrice. Nous prouvons des bornes de l'erreur a priori d'un algorithme *greedy* pour la *cross approximation*, et nous développons une variante plus rapide et plus efficace d'un algorithme existant pour la sélection de sous-ensembles de colonnes. De plus, nous présentons un nouvel algorithme déterministe qui donne en temps polynomial une *cross approximation* quasi-optimale dans la norme de Frobenius.

La deuxième partie de la thèse concerne les fonctions matricielles. Nous développons un algorithme "diviser pour mieux régner" pour calculer les fonctions de matrices à bandes, hiérarchiquement semi-séparables ou qui ont une autre structure de rang faible hors diagonale. Notre approche est basée sur un algorithme existant mettant à jour la fonction d'une matrice qui subit une modification de rang faible, pour lequel nous présentons de nouveaux résultats de convergence. L'analyse de convergence de notre algorithme "diviser pour mieux régner" est liée à une approximation polynomiale ou rationnelle de la fonction.

Dans la troisième partie nous considérons le problème de l'approximation de la trace des matrices dont on ne connaît que leur action sur la multiplication avec un vecteur. Nous analysons un algorithme randomisé, l'estimateur de trace de Hutchinson, et nous prouvons des résultats de convergence pour les matrices symétriques (indéfinies). Ensuite, nous appliquons nos résultats au calcul du (logarithme du) déterminant de matrices symétriques définies positives.

Mots cléfs. Approximation de rang faible, sélection de sous-ensembles de colonnes, mises à jour de rang faible, méthodes de sous-espace de Krylov, fonctions matricielles, matrices à bandes, matrices hiérarchiquement semi-séparables, estimation de trace, déterminant.

Sommario

In questa tesi proponiamo e analizziamo algoritmi per alcuni problemi di algebra lineare numerica: trovare approssimazioni di rango basso di matrici, calcolare funzioni di matrici, e stimare la traccia di funzioni di matrici.

Nella prima parte, consideriamo algoritmi per costruire un'approssimazione di rango basso di una matrice a partire da alcune righe o colonne della matrice stessa. Dimostriamo dei risultati a priori sull'errore di un algoritmo greedy per la *cross approximation* e sviluppiamo una variante più veloce ed efficiente di un algoritmo già esistente per la selezione di un sottoinsieme di colonne. Inoltre presentiamo un nuovo algoritmo deterministico che dà, in tempo polinomiale, una *cross approximation* quasi ottimale nella norma di Frobenius.

La seconda parte della tesi è dedicata alle funzioni di matrici. Sviluppiamo un algoritmo *divide-et-impera* per calcolare funzioni di matrici a banda, gerarchiche, o che abbiano in generale una struttura con dei blocchi di rango basso. Un ingrediente importante per il nostro approccio è un algoritmo esistente per aggiornare la funzione di una matrice che subisce una modifica di rango basso, per il quale presentiamo nuovi risultati di convergenza. L'analisi della convergenza del nostro algoritmo *divide-et-impera* è legata all'approssimazione della funzione tramite polinomi o funzioni razionali.

Nella terza parte consideriamo il problema dell'approssimazione della traccia di matrici che sono disponibili solo tramite moltiplicazioni con un vettore. Analizziamo un algoritmo randomizzato, lo stimatore di Hutchinson, per il quale dimostriamo risultati di convergenza per matrici simmetriche, non necessariamente definite. In seguito applichiamo i nostri risultati al calcolo del (logaritmo del) determinante di matrici simmetriche definite positive.

Parole Chiave. Approssimazione di rango basso, selezione di un sottoinsieme di colonne, aggiornamenti di rango basso, metodi di Krylov, funzioni di matrici, matrici a banda, matrici gerarchiche, stima della traccia, determinante.

Contents

Notation

- $\mathbb{R}$ denotes the set of real numbers;
- $\mathbb{C}$ denotes the set of complex numbers;
- I_n denotes the identity matrix of size $n \times n$;
- "log" denotes the natural logarithm.

For a matrix $A \in \mathbb{R}^{n \times n}$:

- The entry in row i and column j is denoted by a_{ij} or $A(i,j)$;
- $\|A\|_2$ is the spectral norm of the matrix A;
- $\|A\|_F$ is the Frobenius norm of the matrix A;
- $\|A\|_{\max} := \max_{i,j=1,\ldots,n} |a_{ij}|$ is the Chebyshev norm of A;
- $\|A\|_*$ is the nuclear norm of the matrix A;
- $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_n(A) \geq 0$ are the singular values of A; when there is no confusion, they may be denoted by $\sigma_1, \ldots, \sigma_n$;
- $\delta_{k+1}(A)$ denotes the error of a best rank-k approximation of A in the Chebyshev norm;
- $W(A) := \{z^* A z \mid z \in \mathbb{C}^n, \|z\|_2 = 1\}$ is the numerical range of A;
- $\kappa(A) := \|A\|_2 \cdot \|A^{-1}\|_2$ is the condition number of A;
- $A^\dagger$ denotes the Moore-Penrose pseudoinverse of A.

Abbreviations:

- ACA = Adaptive Cross Approximation;
- DD = diagonally dominant;
- D&C = divide-and-conquer;
- HSS = hierarchically semiseparable;
- SPD = symmetric positive definite;
- SPSD = symmetric positive semidefinite.

1 Introduction

This thesis is concerned with three topics in numerical linear algebra: the low-rank approximation of matrices, the computation of matrix functions, and stochastic trace estimation. This chapter is a brief introduction to these problems.

Finding a low-rank approximation of a matrix $A \in \mathbb{R}^{m \times n}$ means finding rectangular factors $B \in \mathbb{R}^{m \times k}$ and $C \in \mathbb{R}^{n \times k}$ such that A is approximately equal to BC^T, for some target rank $k \ll \min\{n, m\}$. Almost all matrices have full rank [184], but in many applications there are matrices that have a low *numerical rank*, that is, they can be well approximated by a low-rank matrix. For example, matrices with low numerical rank arise in the discretization of PDEs [14], in statistical machine learning [80], in social network analysis [139], and in text document analysis [156]. Having a low-rank representation of a matrix $A \in \mathbb{R}^{m \times n}$ yields advantages in terms of storage and computational efficiency. For example, multiplying a dense matrix A by a vector $v \in \mathbb{R}^n$ costs $\mathcal{O}(mn)$ operations; if a rank-k factorization $A = BC^T$ is known, the cost of a matrix-vector multiplication reduces to $\mathcal{O}\left((n+m)k\right)$. In some applications, it is useful to consider low-rank approximations in which the factors B and C are made of, or constructed from, rows and columns of A, as this provides enhanced interpretability.

In the first part of this thesis, we consider three different types of low-rank approximations: *cross approximation*, which interpolates A in k rows and k columns [88]; *column subset selection*, which aims at selecting k columns that well approximate the range of A [58]; and *CUR approximation*, which uses k rows and columns and then minimizes the low-rank approximation error [63]. We prove a priori error bounds for an existing algorithm [13] for cross approximation. For column subset selection, we improve an existing algorithm [57] that is guaranteed to achieve a quasi-optimal low-rank approxima-

tion in the Frobenius norm, and extend the technique to CUR and cross approximation. Although we mostly deal with matrices, we include the analysis of the cross approximation algorithm from [13] applied to bivariate functions [168] and an extension of the column subset selection algorithm from [57] to low-rank approximation of (low-order) tensors.

The second part of the thesis deals with the computation of functions of a matrix $A \in \mathbb{R}^{n\times n}$. Examples of matrix functions include the inverse of A, the square root of a symmetric positive semidefinite (SPSD) A, and the matrix exponential. More generally, one can define a matrix function $f(A)$ whenever the function f is analytic on the spectrum of A [110]. Applications include PDEs [65, 128], social network analysis [70], and electronic structure calculations [19, 84]. Here we consider matrices which have a specific low-rank structure, that is, they have off-diagonal blocks with low rank. Examples include banded matrices and hierarchically semiseparable (HSS) matrices [100]. It is well known that, in many cases, functions of such matrices retain some (approximate) low-rank structure [23, 24, 55, 158], allowing for a memory-efficient representation of $f(A)$. In this thesis we exploit this fact, combined with the observation that if A undergoes a low-rank modification $R \in \mathbb{R}^{n\times n}$ then $f(A+R) - f(A)$ is often numerically low-rank [18, 17], to develop a fast divide-and-conquer (D&C) algorithm to compute matrix functions.

In some applications, only specific quantities associated to a matrix function are required. For example, the logarithm of the determinant of a symmetric positive definite (SPD) matrix can be expressed as the trace of the matrix logarithm of A. For a general dense matrix A, computing $f(A)$ is infeasible in a large-scale setting and cheaper methods are needed to approximate its trace. A stochastic algorithm, the Hutchinson trace estimator [114], provides a way of approximating the trace of a symmetric matrix $B \in \mathbb{R}^{n\times n}$ using a few quadratic forms involving suitable random vectors. In the setting in which $B = f(A)$ is a matrix function, quadratic forms with $f(A)$ can be computed – approximately – via quadrature [85] much more cheaply than the computation of the whole $f(A)$. In the third part of this thesis we analyze the convergence properties of the Hutchinson trace estimator when it is used on a symmetric but indefinite matrix B and we apply the results to the approximation of the determinant of SPD matrices.

Organization of the thesis. The thesis is divided into three parts corresponding to the three topics mentioned above. Each part contains a more detailed introductory chapter. Our contributions are presented in Chapters 3, 4, 6, 7, and 9, which are based on the papers [47, 45, 17, 48, 46]. Chapter 10 serves as the conclusion of the whole manuscript.

Part I

Low-rank approximation based on rows and columns

2 Introduction to low-rank approximation

The first part of this thesis is concerned with low-rank approximation. For a matrix $A \in \mathbb{R}^{m\times n}$, we fix a rank $k \in \{1, 2, \ldots, \min\{m, n\}\}$ and consider the problem of finding a matrix $B \in \mathbb{R}^{m\times n}$ of rank at most k such that $A - B$ is "small". We assume without loss of generality that $m \leq n$. We recall that the singular value decomposition (SVD) of A always exists (see, e.g., [112, Theorem 2.6.3]) and is defined as follows.

Definition 2.1. *An (economy-sized) SVD of A is a factorization of the form*

$$A = U\Sigma V^T,$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times m}$ are matrices with orthonormal columns and $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_m)$ is the $m \times m$ diagonal matrix containing the singular values $\sigma_1 \geq \ldots \geq \sigma_m \geq 0$.

The problem of minimizing $\|A-B\|$ for a unitarily invariant norm (such as the spectral norm $\|\cdot\|_2$ or the Frobenius norm $\|\cdot\|_F$) has an elegant and well-known solution given by the Eckart-Young-Mirsky theorem.

Theorem 2.2 (Eckart-Young-Mirsky (see, e.g., [112])). *For every unitarily invariant norm $\|\cdot\|$ it holds that*

$$\min_{\substack{B\in\mathbb{R}^{m\times n}\\ \operatorname{rank}(B)\leq k}} \|A - B\| = \|A - U_k\Sigma_k V_k^T\|, \tag{2.1}$$

where U_k and V_k denote the matrices formed by the first k columns of U and V, respectively,

and Σ_k denotes the leading $k \times k$ submatrix of Σ. For the spectral norm the right hand side of (2.1) *equals σ_{k+1} and for the Frobenius norm it equals $\sqrt{\sigma_{k+1}^2 + \sigma_{k+2}^2 + \ldots + \sigma_m^2}$.*

In this thesis we also consider the Chebyshev norm, which is denoted by $\|\cdot\|_{\max}$ and is defined as the maximum absolute value of an entry of the matrix. We let

$$\delta_{k+1}(A) := \min_{\substack{B \in \mathbb{R}^{m \times n} \\ \operatorname{rank}(B) \leq k}} \|A - B\|_{\max}, \qquad k = 1, \ldots, m-1 \tag{2.2}$$

be the error of a best rank-k approximation in the Chebyshev norm. Because of $\|A - B\|_2/\sqrt{nm} \leq \|A - B\|_{\max} \leq \|A - B\|_2$, we have $\sigma_k(A)/\sqrt{nm} \leq \delta_k(A) \leq \sigma_k(A)$. There is no explicit characterization of $\delta_k(A)$ in terms of the singular values of A. Iterative methods have been developed to find (approximately) the best low-rank approximation in the Chebyshev norm [143].

While the truncated SVD (2.1) gives the best possible low-rank approximation for the spectral and Frobenius norms, other low-rank approximation strategies have been developed. First of all, it can be beneficial to construct low-rank approximations using rows and/or columns of the original matrix, as this allows for enhanced interpretability and structure preservation. Examples include unsupervised feature selection [34, 138], sensor selection [116], and data mining [72]. We pursue this direction and in Chapters 3 and 4 we focus our work on cross approximation [13, 88, 168] (see Section 2.1), column subset selection [58] (see Section 2.2), and CUR approximation [63, 166, 172] (see Section 2.3).

A second disadvantage of the truncated SVD is that it costs $\mathcal{O}(nm^2)$ *flops* (floating point operations) for dense matrices: Faster methods are desirable for large-scale matrices. An algorithm for cross approximation that runs faster than the SVD is a greedy algorithm called *Adaptive Cross Approximation (ACA)* [13], which we will analyze in Chapter 3. For column subset selection, greedy and randomized strategies have been developed; see, e.g., [35, 63, 64, 78, 147, 192]. Another (cheap) way to obtain a low-rank approximation of a matrix is via a randomized SVD (see, e.g., [101]).

In the rest of this chapter, we introduce cross approximations, the column subset selection problem, and CUR approximations. We review some results which prescribe ways to choose rows and columns such that a good approximation is obtained, and briefly review existing algorithms. Finally, in Section 2.4 we briefly discuss the generalization of these algorithms to the low-rank approximation of tensors.

2.1 Cross approximation

Cross approximation is a type of low-rank approximation that is often used for matrices A that cannot be stored in memory, because, for example, A has too many nonzero entries or it is too expensive to compute all the entries of A. This situation occurs frequently for discretized integral operators and cross approximation plays an important role in accelerating computations within the boundary element method [13] and uncertainty quantification [106].

To formally define a cross approximation, we denote by $I \in \{1, \ldots, m\}^k$ and $J \in \{1, \ldots, n\}^k$ some ordered sets (*tuples*) corresponding to row and column index sets of cardinality k. We denote by $A(I, :)$ and $A(:, J)$ the corresponding rows and columns of A, and by $A(I, J)$ the $k \times k$ matrix at the intersection of rows I and columns J.

Definition 2.3. *If $A(I, J)$ is invertible then the rank-k matrix*

$$A_{IJ} := A(:, J)A(I, J)^{-1}A(I, :) \tag{2.3}$$

is called a cross approximation.

In the literature, cross approximations are also called *skeleton decompositions* [89] and *CUR decompositions*[1] [133, Section 13.1]. Low-rank approximations of the form (2.3) also feature prominently in the Nyström method for kernel-based learning [9, 83] and spectral clustering [77]; in these cases, the involved matrices are SPSD.

2.1.1 Existence results for cross approximation

When A has rank exactly k, a cross approximation of the form (2.3) is equal to A; otherwise, A_{IJ} interpolates A exactly in the rows and columns corresponding to I and J, but the error

$$E_{IJ} := A - A(:, J)A(I, J)^{-1}A(I, :)$$

crucially depends on the choice of the index sets. In this section we review some existing results on the existence of good index sets for cross approximation and we summarize the existing polynomial-time algorithms that ensure a quasi-optimal output in Section 2.1.2.

[1] Note that our definition of CUR approximation in (2.11) is different from the CUR decomposition in [133, Section 13.1]. We use the definition of CUR approximation from [166, 172].

It is useful to introduce the notion of *volume* of a matrix.

Definition 2.4. *The volume of an $m \times n$ matrix A is the product of its singular values.*

Equivalently, $\mathrm{Vol}\,(A) = \sqrt{\det(A^T A)}$ (when $m \leq n$). Note that the volume of a square matrix is the absolute value of its determinant. We denote by $\mathrm{maxVol}_k(A)$ the volume of a $k \times k$ maximum volume submatrix of A.

Definition 2.5. *Given $\gamma \geq 1$, a $k \times k$ submatrix $A(I, J)$ of A has* local γ-maximum volume *in A if, for every pair of index sets $(\widetilde{I}, \widetilde{J})$ of cardinality k such that $\widetilde{I}$ differs from I by at most one index and $\widetilde{J}$ differs from J by at most one index, we have $\mathrm{Vol}\,(A(I, J)) \geq \frac{1}{\gamma}\,\mathrm{Vol}\,(A(\widetilde{I}, \widetilde{J}))$. $A(I, J)$ has* local maximum volume *if $\gamma = 1$. We define local γ-maximum volume rectangular submatrices $A(:, J)$ analogously (see also [153, Definition 3.2]).*

We will say *local quasi-maximum volume* submatrices when γ is "sufficiently small" but not specified (e.g. $\gamma = 1.2$). A local maximum volume submatrix of A can be very far from being a maximum volume submatrix. For example, in the $m \times 2m$ matrix

$$A = \begin{bmatrix} 1 & & & & -1 & -1 & \cdots & -1 \\ & 1 & & & 1 & -1 & \cdots & -1 \\ & & \ddots & & & \ddots & \ddots & \vdots \\ & & & 1 & & & 1 & -1 \end{bmatrix}$$

the first m columns form a local maximum volume submatrix which has volume 1, while the last m columns form a submatrix with volume 2^{m-1}.

The following theorem shows that local quasi-maximum volume submatrices provide a good choice for index sets (I, J) for cross approximation.

Theorem 2.6 ([88, Theorem 2.2] and [91, Theorem 1])**.** *Let I and J be index sets of cardinality k such that $A(I, J)$ is a local γ-maximum volume submatrix of A. Then*

$$\|E_{IJ}\|_{\max} \leq \gamma(k+1)\sigma_{k+1}(A) \text{ and } \|E_{IJ}\|_{\max} \leq \gamma(k+1)^2\delta_{k+1}(A). \tag{2.4}$$

Note that the assumption of Theorem 2.6 is weaker than the assumptions of [88, 91]; however, the proofs from these papers still work with the weaker assumption. For the sake

of completeness, we present the proof of Theorem 2.6, following [88, 91] and emphasizing the fact that the *locality* of the quasi-maximum volume constraints is the only assumption we need.

Proof. We prove that the absolute value of each entry of the error matrix E_{IJ} is bounded by $\gamma(k+1)\sigma_{k+1}(A)$ and by $\gamma(k+1)^2\delta_{k+1}(A)$. The only entries of E_{IJ} that can be nonzero are of the form $E_{IJ}(i,j)$ with $i \notin I$ and $j \notin J$; these are the entries of the Schur complement of $A(I,J)$ in A. For one of these, let us consider the $(k+1)\times(k+1)$ submatrix of A given by

$$\hat{A} := \begin{bmatrix} A(I,J) & A(I,j) \\ A(i,J) & A(i,j) \end{bmatrix}.$$

The cross approximation error in $\hat{A}$ associated to the leading $k \times k$ principal submatrix $A(I,J)$ is the Schur complement $|E_{IJ}(i,j)| = |A(i,j) - A(i,J)A(I,J)^{-1}A(I,j)|$. By [112, Equation (0.8.5.1)], we have that $|\det A(I,J)| \cdot |E_{IJ}(i,j)| = |\det \hat{A}|$. If $\hat{A}$ is singular, then $|E_{IJ}(i,j)| = 0$; therefore, in the following we consider the case in which $\hat{A}$ is invertible. Note that by the adjugate formula for the inverse of a matrix we also have $\|\hat{A}^{-1}\|_{\max} = \operatorname{maxVol}_k(\hat{A})/|\det \hat{A}|$. Therefore,

$$|E_{IJ}(i,j)| = \frac{|\det \hat{A}|}{|\det A(I,J)|} \leq \frac{\gamma|\det \hat{A}|}{\operatorname{maxVol}_k(\hat{A})} = \frac{\gamma}{\|\hat{A}^{-1}\|_{\max}} \leq \frac{\gamma(k+1)}{\|\hat{A}^{-1}\|_2} = \gamma(k+1)\sigma_{k+1}(\hat{A}), \tag{2.5}$$

where we used that $\|\hat{A}^{-1}\|_2 \leq \|\hat{A}^{-1}\|_F \leq (k+1)\|\hat{A}^{-1}\|_{\max}$ and that $|\det A(I,J)| \geq \frac{1}{\gamma}\operatorname{maxVol}_k(\hat{A})$ thanks to the local γ-maximum volume assumption, because $\hat{A}$ is of size $(k+1)\times(k+1)$. By the interlacing properties of singular values (see, e.g., [112, Corollary 7.3.6]) we have $\sigma_{k+1}(\hat{A}) \leq \sigma_{k+1}(A)$. Therefore, for every choice of $i \notin I$ and $j \notin J$ we have

$$|E_{IJ}(i,j)| \leq \gamma(k+1)\sigma_{k+1}(\hat{A}) \leq \gamma(k+1)\sigma_{k+1}(A),$$

which implies the first part of (2.4). Moreover, $\sigma_{k+1}(\hat{A}) \leq (k+1)\delta_{k+1}(\hat{A}) \leq (k+1)\delta_{k+1}(A)$, with the last inequality being a direct consequence of the definition of δ_k. This implies, together with (2.5), the second part of (2.4). □

Results for cross approximation in the spectral norm exist in the literature of the closely related topic of rank-revealing factorizations. More specifically, for an $n \times n$ matrix

A, for $\gamma \geq 1$, there exist index sets I and J of cardinality k such that

$$\|E_{IJ}\|_2 \leq \left(1 + \gamma^2 k\sqrt{(n-k)(m-k)}\right)\sigma_{k+1}. \tag{2.6}$$

In [153, Theorem 3.8] this is achieved by choosing $A(:, J)$ to be a rectangular γ-maximum volume submatrix of A and by taking $A(I, J)$ to be a local γ-maximum volume submatrix of $A(:, J)$. For an SPSD matrix with a Cholesky decomposition $A = LL^T$, the relation (2.6) holds when $I = J$ and $L(:, I)$ is a local γ-maximum volume submatrix of L [97].

In the case of the Frobenius norm, for which the best rank-k approximation error is $\sqrt{\sigma_{k+1}^2 + \ldots + \sigma_m^2}$, Zamarashkin and Osinsky [196] proved the existence of index sets I and J of cardinality k such that

$$\|E_{IJ}\|_F \leq (k+1)\sqrt{\sigma_{k+1}^2 + \ldots + \sigma_m^2}. \tag{2.7}$$

Their proof follows a probabilistic argument; we will discuss this result and the techniques used in the proof in detail in Chapter 4. Using a similar argument, a result for SPSD matrices for the nuclear norm – which we denote by $\|\cdot\|_*$ – was obtained in [134] and states that there exists an index set I of cardinality k such that $\|E_{II}\|_* \leq (k+1)(\sigma_{k+1} + \ldots + \sigma_n)$.

The results mentioned in this section are summarized in Table 2.1. The fourth result in the table actually implies a better existence result for the spectral norm, that is, there exist index sets I and J such that

$$\|E_{IJ}\|_2 \leq \|E_{IJ}\|_F \leq (k+1)\sqrt{\sigma_{k+1}^2 + \ldots + \sigma_m^2} \leq (k+1)\sqrt{m-k} \cdot \sigma_{k+1}.$$

We included the second and third rows of the table because of their connection to quasi-maximum volume submatrices.

There are also results for cross approximation when more than k rows and columns are chosen (see, e.g., [151]), but we do not consider them in this thesis.

2.1.2 Algorithms for cross approximation

Finding a submatrix of maximum volume of A (which would give a good choice of index sets by Theorem 2.6) is an NP-hard problem [154]. Fortunately, finding index sets for a good cross approximation is easier. A common algorithm for cross approximation is

Existence result	Assumptions	Reference
$\|E_{IJ}\|_{\max} \leq \gamma(k+1)\sigma_{k+1}$ $\|E_{IJ}\|_{\max} \leq \gamma(k+1)^2\delta_{k+1}$	$A \in \mathbb{R}^{m\times n}$, $A(I,J)$ local γ-maximum volume in A	[88, Theorem 2.2] [91, Theorem 1]
$\|E_{IJ}\|_2 \leq (\sqrt{1+\gamma^2 k(n-k)}$ $\cdot\sqrt{1+\gamma^2 k(m-k)})\sigma_{k+1}$	$A \in \mathbb{R}^{m\times n}$, $A(:,J)$ local γ-maximum volume in A and $A(I,J)$ local γ-maximum volume in $A(:,J)$	[153, Theorem 3.8]
$\|E_{II}\|_2 \leq (1+\gamma^2 k(n-k))\sigma_{k+1}$	$A = LL^T \in \mathbb{R}^{n\times n}$ SPD, $L(:,I)$ local γ-maximum volume in L	[97, page 82]
$\|E_{IJ}\|_F \leq (k+1)\sqrt{\sigma_{k+1}^2+\ldots+\sigma_m^2}$	$A \in \mathbb{R}^{m\times n}$, no explicit characterization, related to volume sampling	[196, Theorem 1]
$\|E_{II}\|_* \leq (k+1)(\sigma_{k+1}+\ldots+\sigma_n)$	$A \in \mathbb{R}^{n\times n}$ SPSD, no explicit characterization, related to vol. sampling	[134, Theorem 1]

Table 2.1 – Existence results for cross approximation.

ACA [13], which is a greedy algorithm for volume maximization. We will recall ACA in Section 3.2. For general matrices its cost is $\mathcal{O}(nmk)$; moreover, it has the advantage that for $n \times n$ SPSD or diagonally dominant (DD) matrices the cost is reduced to $\mathcal{O}(nk^2)$. In general ACA does not ensure that a local quasi-maximum volume submatrix is returned. A suitable modification, which will be proposed in Section 3.3, ensures that the relation (2.4) holds (but makes the algorithm slightly more expensive).

An algorithm for finding k columns of A which form a submatrix of local quasi-maximum volume is presented in [97] in the context of strong rank-revealing QR factorizations. This result can be used to derive algorithms that match the second and third rows in Table 2.1. More specifically, for an SPSD matrix A with Cholesky decomposition $A = LL^T$, this algorithm can be used on the factor L to get an index set I satisfying the third line of Table 2.1. An algorithm that matches the result of the second row of Table 2.1 is discussed in [153], but there are no complexity bounds in that paper. However,

if one uses the strong rank-revealing QR algorithm from [97] twice – once on A to find J and once on $A(:,J)^T$ to find I – one gets an $\mathcal{O}\left(\frac{nmk\log n}{\log\gamma}\right)$ algorithm.

A result that matches the fourth row in Table 2.1 will be presented in Chapter 4. Returning again to SPSD matrices, an $\mathcal{O}(n^3)$ algorithm that matches the existence result in the last row of Table 2.1 is proposed in [134, Algorithm 4].

2.2 Column subset selection

The *column subset selection problem* is a classical problem in numerical linear algebra which has broad applications in a variety of disciplines, such as scientific computing, model reduction, and statistical data analysis and is closely connected to rank-revealing QR factorizations [40, 42, 96]. The aim is to determine an index set $I \in \{1,\ldots,n\}^k$ of cardinality k such that the corresponding k columns $A(:,I)$ represent a good approximation of the range of A.

In this section, we give an overview of some existing results and algorithms for column subset selection. Let us focus, for now, on the Frobenius norm. The best approximation error attained by an *arbitrary* $m \times k$ matrix Q is

$$\min_{Q\in\mathbb{R}^{m\times k}} \|A - QQ^\dagger A\|_F^2 = \sigma_{k+1}^2 + \ldots + \sigma_m^2 \tag{2.8}$$

and a minimizer is $Q = U_k$ from the truncated SVD of A. Here, $\dagger$ denotes the Moore–Penrose inverse of a matrix and $QQ^\dagger$ is the orthogonal projector onto the range of Q. Also for the column subset selection problem the volume of submatrices plays an important role. In [58] it is proven that if index sets (tuples) X of cardinality k are sampled in such a way that $\mathbb{P}(X = I)$ is proportional to the squared volume of $A(:,I)$, then

$$\mathbb{E}[\|A - A(:,X)A(:,X)^\dagger A\|_F^2] \leq (k+1)(\sigma_{k+1}^2 + \ldots + \sigma_m^2). \tag{2.9}$$

In turn, this implies that there *exists* a set I of cardinality k such that

$$\|A - A(:,I)A(:,I)^\dagger A\|_F \leq \sqrt{k+1}\sqrt{\sigma_{k+1}^2 + \ldots + \sigma_m^2}. \tag{2.10}$$

We remark that taking I corresponding to the maximum volume submatrix is not always the best choice, and not always satisfies (2.10) (see Section 4.1). The bound (2.10)

measures how well all the columns of A are approximated by the subset of columns contained in I. This is remarkable because the bound is larger than the best possible result (2.8) by a factor that is only linear in k. We will call any quasi-optimal bound with a factor that is at most polynomial in k (and independent of m, n, and A) a *polynomial bound*. We remark that the factor $\sqrt{k+1}$ in (2.10) cannot be improved [58, Proposition 3.3]. In [57], a *deterministic* polynomial-time algorithm has been developed by derandomizing this approach using the method of conditional expectations. We will recall such an algorithm in Section 4.1.1. Another line of work that exploits (2.9) investigates methods to do (approximate) volume sampling, which leads to randomized algorithms; see, e.g., [20, 56, 57].

Let us briefly consider the column subset selection problem in the spectral norm. The discrete empirical interpolation method (DEIM) [64] allows us to obtain in time $\mathcal{O}(nm^2)$ an index set I such that

$$\|A - A(:,I)A(:,I)^\dagger A\|_2 \leq 2^k \sqrt{\frac{nk}{3}} \sigma_{k+1}.$$

The already mentioned strong rank-revealing QR algorithm from [95] implies the existence of an index set I of cardinality k such that

$$\|A - A(:,I)A(:,I)^\dagger A\|_2 \leq \sqrt{1+\gamma^2 k(n-k)}\sigma_{k+1}$$

for $\gamma \geq 1$, which can be found in time $\mathcal{O}\left(m^2\left(n + \frac{m \log m}{\log \gamma}\right)\right)$ for $\gamma > 1$. It is not possible, in the case of the spectral norm, to get bounds that do not depend on the size of the matrix.

We have restricted our discussion to the selection of *exactly* k columns; let us mention that several other greedy and randomized algorithms have been proposed and analyzed, some of which require to select more than k columns; see, e.g., [35, 63, 64, 78, 147, 192] for a few references representing this research direction.

2.3 CUR approximation

In Section 2.1 we introduced cross approximation as a type of low-rank approximation built from distinct rows and columns of A, and the $k \times k$ matrix $A(I,J)^{-1}$. CUR approximations are a more general format in which the middle matrix can be replaced by

any matrix $U \in \mathbb{R}^{k\times k}$: they are of the form

$$A \approx CUR, \qquad C = A(:, J), \qquad R = A(I, :), \tag{2.11}$$

for some row and column index sets I and J of cardinality k. Such decompositions have been considered initially in [89] (with the name of *pseudoskeleton decompositions*). Later, Stewart [175] proved that, given C and R, the choice that minimizes the approximation error in the Frobenius norm is $U = C^{\dagger}AR^{\dagger}$. This is, in general, different from the matrix $U = A(I, J)^{-1}$ chosen for cross approximation; in turn, the decomposition (2.11) does not, in general, interpolate the matrix A in the selected rows and columns.

There is a simple and well established strategy to derive a CUR approximation; see, e.g., [63, 172]. One first applies column subset selection to A and A^T in order to determine C and R, respectively. Using the results on DEIM, in [172] a CUR approximation is constructed such that $\|A - CUR\|_2 \leq 2^k \left(\sqrt{\frac{mk}{3}} + \sqrt{\frac{nk}{3}}\right)\sigma_{k+1}$. Analogously, using the results from [95], in [166] it is proven that one can construct I and J such that $\|A - CUR\|_F \leq \sqrt{2 + \gamma^2 k(m + n - 2k)}\sqrt{\sigma_{k+1}^2 + \ldots + \sigma_m^2}$. Note that the first bound features an exponential dependence on k and both bounds depend on the dimension of the matrix.

2.4 Low-rank approximation of tensors

The low-rank approximations discussed before can be extended to tensors. For example, a strategy similar to the CUR approximation has been used in [62, 90, 166] for low-order tensors, and cross approximation of tensors has been discussed in [148, 149]. In this thesis (in Section 4.3) we only focus on the generalization of CUR approximation to tensors, which is summarized below. We will need some basic definitions regarding tensors; we refer to [120] for more details.

Definition 2.7. *Let $\mathcal{A} \in \mathbb{R}^{n_1\times\ldots\times n_d}$ be a d-th order tensor. Generalizing the notion of rows and columns of a matrix, the vectors obtained from $\mathcal{A}$ by fixing all indices but the μth one are called* μ-mode fibers*, where $\mu \in \{1, \ldots, d\}$. The matrix $A^{(\mu)} \in \mathbb{R}^{n_\mu\times(n_1\cdots n_d)/n_\mu}$ containing all μ-mode fibers as columns is called the* μ-mode matricization *of $\mathcal{A}$. The μ-mode product of a matrix $B \in \mathbb{R}^{m\times n_\mu}$ with $\mathcal{A}$ is denoted by $\mathcal{A} \times_\mu B$ and it is the $n_1 \times \ldots \times n_{\mu-1} \times m \times n_{\mu+1} \times \ldots \times n_d$ tensor such that its μ-mode matricization is $B \cdot A^{(\mu)}$.*

Definition 2.8. *The Frobenius norm of a tensor is defined as*

$$\|\mathcal{A}\|_F^2 := \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \mathcal{A}(i_1, \ldots, i_d)^2.$$

Note that $\|\mathcal{A}\|_F = \|A^{(\mu)}\|_F$ for all $\mu = 1, \ldots, d$.

Definition 2.9. *The tuple $(k_1, \ldots, k_d)$ defined by $k_\mu = \operatorname{rank}(A^{(\mu)})$ is called the multilinear rank of $\mathcal{A}$ and we can decompose $\mathcal{A}$ as*

$$\mathcal{A} = \mathcal{C} \times_1 B_1 \times_2 \ldots \times_d B_d, \tag{2.12}$$

for coefficient matrices $B_\mu \in \mathbb{R}^{n_\mu \times k_\mu}$ for $\mu = 1, \ldots, d$ and a so called core tensor *$\mathcal{C} \in \mathbb{R}^{k_1 \times \ldots \times k_d}$. The relation* (2.12) *is called a* Tucker decomposition *of $\mathcal{A}$.*

The Tucker decomposition is particularly beneficial when the multilinear rank is much smaller than the size of a tensor. Tucker approximations have been considered that use as $B_1, \ldots, B_d$ some subsets of $k_1, \ldots, k_d$ fibers of $\mathcal{A}$; see, e.g., [62, 90, 166]. For a d-th order tensor of size $n \times \ldots \times n$ and $k_1 = \ldots = k_d = k$, the bounds from [90, 166] state the existence of fiber subsets $B_1, \ldots, B_d$ of cardinality k and a core tensor $\mathcal{C} := \mathcal{A} \times_1 B_1^\dagger \times_2 \ldots \times_d B_d^\dagger$ such that

$$\|\mathcal{A} - \mathcal{C} \times_1 B_1 \times_2 \ldots \times_d B_d\|_F \leq \sqrt{d}\left(1 + \sqrt{1 + k(n^{d-1} - k)}\right) \cdot \|\mathcal{A} - \mathcal{A}_{\text{best}}\|_F, \tag{2.13}$$

where $\mathcal{A}_{\text{best}}$ is the best Tucker approximation of $\mathcal{A}$ of multilinear rank at most $(k_1, \ldots, k_d)$. Note that the suboptimality factor in (2.13) depends on the size of the tensor and grows exponentially with the order d.

2.5 Contributions

In Chapter 3 we focus on cross approximation and the ACA algorithm. First of all, motivated by Theorem 2.6 and the fact that ACA is much faster for SPSD or DD matrices, we consider the problem of finding a maximum volume submatrix in these special cases. We prove that, in both cases, the maximum volume submatrix can be chosen to be a principal submatrix. Then, we study a priori error bounds for ACA. We provide a convergence result that holds for any matrix $A \in \mathbb{R}^{m \times n}$. Our bound for general matrices

extends an existing result for SPSD matrices and yields new error estimates for DD matrices. In particular, for doubly DD matrices the error is shown to remain within a modest factor of the best approximation error. We also illustrate how the application of our results to cross approximation for functions leads to new and better convergence results. In the last part of the chapter, we show that, starting from the approximation given by ACA, an iterative strategy allows us to get a polynomial error bound of the form (2.4) in polynomial time for general matrices.

Output	Complexity	Reference
Cross approximation such that $\|E_{IJ}\|_{\max} \leq \gamma(k+1)\sigma_{k+1}$ and $\|E_{IJ}\|_{\max} \leq \gamma(k+1)^2\delta_{k+1}$	$\mathcal{O}\left(\frac{nmk^2\log^2 m}{\log\gamma}\right)$	Section 3.3
Symmetric cross approximation of SPD $n \times n$ matrix such that $\|E_{II}\|_2 \leq (1+\gamma^2 k(n-k))\sigma_{k+1}$ and $\|E_{II}\|_{\max} \leq \gamma^2(k+1)\sigma_{k+1}$	$\mathcal{O}\left(\frac{nk^2\log n}{\log\gamma}\right)$	[88, 97]
Cross approximation such that $\|E_{IJ}\|_2 \leq (\sqrt{1+\gamma^2 k(n-k)} \cdot\sqrt{1+\gamma^2 k(m-k)})\sigma_{k+1}$	$\mathcal{O}\left(\frac{nmk\log n}{\log\gamma}\right)$	[97, 153]
Cross approximation such that $\|E_{IJ}\|_F \leq (k+1)\sqrt{\sigma_{k+1}^2+\ldots+\sigma_m^2}$	$\mathcal{O}(nm^2k)$	Theorem 4.5
Symmetric cross approximation of SPD $n \times n$ matrix such that $\|E_{II}\|_* \leq (k+1)(\sigma_{k+1}+\ldots+\sigma_n)$	$\mathcal{O}\left(n^3\right)$	[134]

Table 2.2 – Algorithms for cross approximation with polynomial approximation errors.

In Chapter 4 we present a faster and more efficient variant of the deterministic column subset selection algorithm proposed in [57], which achieves the quasi-optimal bound (2.10). By applying this result to CUR and tensor approximation we improve the results mentioned in Section 2.3. In particular, our bounds do not depend on the matrix/tensor size and do not have exponentially large constants. Then, using the technique of derandomization by conditional expectations as in [57] we construct a polynomial-time deterministic algorithm for cross approximation which achieves the quasi-optimal error in the Frobenius norm from [196] (the fourth row in Table 2.1).

Table 2.2 summarizes the polynomial-time algorithms which ensure a cross approximation with a polynomial error bound that we reviewed in Section 2.1.2 and that we present in Chapters 3 and 4.

Chapter 3 is based on the paper [47] and Chapter 4 is based on the paper [45].

3 Maximum volume submatrices and cross approximation

This chapter contains an analysis of the ACA algorithm for cross approximation (recall Definition 2.3) and a discussion of maximum volume submatrices of SPSD and DD matrices, inspired by Theorem 2.6. More specifically, Chapter 3 is organized as follows. In Section 3.1 we prove that if A is SPSD or DD then there exists a maximum volume submatrix which is a principal submatrix. In Section 3.2 we recall the greedy ACA algorithm for volume maximization and we derive a priori error bounds for the approximation returned by ACA. Although the literature on rank-revealing LU factorizations contains related results, see in particular [75, Corollary 5.3], the non-asymptotic bound of Theorem 3.6 appears to be new. Our result includes existing work [74, 106] for SPSD matrices as a special case. It also allows us to obtain refined bounds for the convergence of cross approximation applied to functions [13, 180]. Finally, in Section 3.3 we discuss a strategy that allows us to obtain a cross approximation satisfying a polynomial error bound in the Chebyshev norm, by suitably modifying the ACA algorithm.

3.1 Maximum volume submatrices

In this section we consider the problem of finding a submatrix of *maximum volume* of A. Our primary motivation is Theorem 2.6, but the problem is connected to a range of other applications in discrete mathematics, engineering, and scientific computing; see, e.g., [5, 94, 189].

Finding a submatrix of maximum volume of A is an NP-hard problem [38, 154].

In [59] it is shown that there exists a universal constant $c > 1$ such that it is NP-hard to approximate the maximum volume of a $k \times k$ submatrix of a matrix $A \in \mathbb{R}^{n\times k}$ within a factor c^k. By a trivial embedding, this implies that it is also NP-hard to approximate the maximum volume of a $k \times k$ submatrix of an $n \times n$ matrix.

In many applications, the matrix A carries additional structure. For example, if A is the discretization of an integral operator with a positive semidefinite kernel then A is SPSD. In this section, we recall that the submatrix of maximum volume is always attained by a principal submatrix, that is, a submatrix of the form $A(I, I)$, if A is SPSD. This has a number of important consequences. For example, it allows us to draw a one-to-one correspondence to the problem of finding a column subset of maximum volume considered in [38]. In turn, the maximum volume problem remains NP-hard when restricted to SPSD matrices. We provide a new extension of this result to DD matrices.

3.1.1 Symmetric positive semidefinite matrices

For an SPSD matrix, an element of maximum absolute value can always be found on the diagonal. Using compound matrix theory (see, e.g., [112, Section 0.8.1]), this result extends to volumes of submatrices.

Theorem 3.1. *Let $A \in \mathbb{R}^{n\times n}$ be SPSD and let $1 \leq k \leq n$. Then the maximum volume $k \times k$ submatrix of A can be chosen to be a principal submatrix.*

Proof. The k-th compound matrix $C_k(A)$ is an $\binom{n}{k} \times \binom{n}{k}$ matrix containing the determinants of all $k \times k$ submatrices of A, such that the determinants of the principal submatrices are on the diagonal. The compound matrix of an SPSD matrix is again SPSD [112, problem 4.1.P25] and hence its diagonal contains an element of maximum absolute value. □

Trivially, the result of Theorem 3.1 extends to symmetric negative semidefinite matrices. On the other hand, it does not extend to the indefinite case; consider for example the $2k \times 2k$ matrix $A = \left[\begin{smallmatrix} 0 & I \\ I & 0 \end{smallmatrix}\right]$.

We include two additional proofs of Theorem 3.1.

Proof 2. Let $A(I, J)$ be any $k \times k$ submatrix of A. As A is SPSD, it admits a factorization

of the form $A = LL^T$ for a lower triangular matrix L and, in turn, $A(I,J) = L(I,:)L(J,:)^T$. The singular values of a principal submatrix satisfy

$$\sigma_i(A(I,I)) = \sigma_i\left(L(I,:)L(I,:)^T\right) = \sigma_i(L(I,:))^2.$$

Recalling that the absolute value of the determinant equals the product of the singular values, we obtain

$$\begin{aligned}
\det(A(I,J))^2 &= \left(\Pi_{i=1}^k \sigma_i(A(I,J))\right)^2 = \left(\Pi_{i=1}^k \sigma_i(L(I,:)L(J,:)^T)\right)^2 \\
&\leq \left(\Pi_{i=1}^k \sigma_i(L(I,:))\Pi_{j=1}^k \sigma_j(L(J,:))\right)^2 \\
&= \left(\Pi_{i=1}^k \sigma_i(L(I,:)L(I,:)^T)\right)\left(\Pi_{j=1}^k \sigma_j(L(J,:)L(J,:)^T)\right) \\
&= \left(\Pi_{i=1}^k \sigma_i(A(I,I))\right)\left(\Pi_{j=1}^k \sigma_j(A(J,J))\right) \\
&= \det(A(I,I)) \cdot \det(A(J,J)),
\end{aligned}$$

where we used [111, Theorem 3.3.4] for the inequality. This implies that the volume of $A(I,J)$ is not larger than the maximum of the volumes of $A(I,I)$ and $A(J,J)$. In turn, $A(I,J)$ can be replaced by a principal submatrix without decreasing the volume. □

Proof 3. Let $A(I,J)$ be any $k \times k$ submatrix of A. We will prove that

$$\det(A(I,J))^2 \leq \det(A(I,I)) \cdot \det(A(J,J)). \tag{3.1}$$

If $I \cap J = \varnothing$ then the inequality (3.1) is proved in Theorem 1 in [71].

In the general case, we will construct a matrix for which we can apply again this theorem. Let $d = |I \cap J|$. By choosing a suitable permutation and applying it to the rows and columns of A, we may assume without loss of generality that

$$I = (1, \ldots, k), \qquad J = (k-d+1, \ldots, 2k-d).$$

When partitioning

$$A = \begin{array}{c} \begin{array}{cccc} \overbrace{\phantom{A_{11}}}^{k-d} & \overbrace{\phantom{A_{12}}}^{d} & \overbrace{\phantom{A_{13}}}^{k-d} & \end{array} \\ \begin{bmatrix} A_{11} & A_{12} & A_{13} & \star \\ A_{21} & A_{22} & A_{23} & \star \\ A_{31} & A_{32} & A_{33} & \star \\ \star & \star & \star & \star \end{bmatrix} \end{array} \begin{array}{l} \}\, k-d \\ \}\, d \\ \}\, k-d \\ \end{array}$$

we now have $\left[\begin{smallmatrix} A_{12} & A_{13} \\ A_{22} & A_{23} \end{smallmatrix}\right] = A(I,J) =: B_{12}$. We also set $B_{11} := A(I,I) = \left[\begin{smallmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{smallmatrix}\right]$ and $B_{22} := A(J,J) = \left[\begin{smallmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{smallmatrix}\right]$, which both inherit the symmetric positive definiteness from A. We now build a $2k \times 2k$ matrix B by "repeating" the second block row and column of the leading $(2k-d) \times (2k-d)$ submatrix of A:

$$B = \left[\begin{array}{cc|cc} A_{11} & A_{12} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{22} & A_{23} \\ \hline A_{21} & A_{22} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{32} & A_{33} \end{array}\right] = \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix}.$$

By construction, B is SPSD. To see the latter, it is sufficient to notice that all principal minors are nonnegative. Indeed, any principal submatrix of B either has (at least) two equal rows or is a principal submatrix of A. Theorem 1 in [71] now gives

$$\det(B_{21})^2 \leq \det(B_{11}) \det(B_{22}),$$

which is (3.1). □

Connection to finding a subset of columns of maximum volume

In [38] the following problem was proven to be NP-hard:

> Given $B \in \mathbb{R}^{m \times n}$ and $1 \leq k < n$, select an $m \times k$ submatrix of maximum volume.

Theorem 3.1 allows us to relate such a problem to the classical maximum volume submatrix problem. Given $B \in \mathbb{R}^{m \times n}$, we consider the SPSD matrix $A = B^T B \in \mathbb{R}^{n \times n}$. As $A(I,I) = B(:,I)^T B(:,I)$ for any ordered index set (tuple) I, there is a one-to-one correspondence between the principal submatrices of A and the ordered subsets of k columns of B. Moreover, we have that

$$\mathrm{Vol}\,(A(I,I)) = \prod_{i=1}^{k} \sigma_i(A(I,I)) = \prod_{i=1}^{k} \sigma_i(B(:,I))^2 = \mathrm{Vol}\,(B(:,I))^2.$$

This shows that $B(:,I)$ has maximum volume if and only if $A(I,I)$ has maximum volume.

In turn, this proves that the maximum volume submatrix problem remains NP-hard when restricted to the subclass of SPSD matrices.

3.1.2 Diagonally dominant matrices

Definition 3.2. *A matrix $A \in \mathbb{R}^{n\times n}$ is called (row)* diagonally dominant *(DD) if*

$$\sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}| \leq |a_{ii}|, \qquad \text{for } i = 1, \ldots, n. \tag{3.2}$$

If (3.2) *holds with strict inequality for $i = 1, \ldots, n$, we call A* strictly DD*. A matrix A is called* doubly DD *if both A and A^T are DD.*

Lemma 3.3. *Let $T \in \mathbb{R}^{n\times n}$ be a strictly DD, upper triangular matrix. Then $|\det(T(I,J))| < |\det(T(I,I))|$ holds for all index sets $I \neq J$ with the same cardinality.*

Proof. Let D be the diagonal matrix with $d_{ii} = t_{ii}$ and set $\tilde{T} = D^{-1}T$. Because of

$$\begin{aligned} \det(T(I,J)) &= \det(D(I,I)) \cdot \det(\tilde{T}(I,J)), \\ \det(T(I,I)) &= \det(D(I,I)) \cdot \det(\tilde{T}(I,I)), \end{aligned}$$

the statement of the lemma holds for T if and only if it holds for $\tilde{T}$. In turn, this allows us to assume without loss of generality that T has ones on the diagonal. In particular, $\det(T(I,I)) = 1$.

The statement of the lemma will be proven by induction on $k := |I| = |J|$. The case $k = 1$ follows immediately from the diagonal dominance of T. Suppose now that the statement of the lemma is true for fixed k. To prove the statement for $k+1$, we consider an arbitrary $(k+1)\times(k+1)$ submatrix $B := T(I,J)$. If $I \neq J$ then there exists a row $B(i,:)$ that does not contain a diagonal element of T. By diagonal dominance of T,

$$|b_{i,1}| + |b_{i,2}| + \ldots + |b_{i,k+1}| < 1. \tag{3.3}$$

Denote by B_{ij} the $k \times k$ submatrix of T obtained from eliminating the ith row and jth column of B. By induction assumption, $|\det(B_{ij})| \leq 1$. Thus, combining (3.3) with the

Laplace expansion gives

$$|\det(B)| = \left|\sum_{j=1}^{k+1}(-1)^{i+j}b_{ij}\det(B_{ij})\right| \leq \sum_{j=1}^{k+1}|b_{ij}|\,|\det(B_{ij})| \leq \sum_{j=1}^{k+1}|b_{ij}| < 1.$$

In other words, $|\det(T(I,J))| < |\det(T(I,I))|$. □

Theorem 3.4. *Let $A \in \mathbb{R}^{n\times n}$ be a DD matrix and $1 \leq k \leq n$. Then the maximum volume $k \times k$ submatrix of A can be chosen to be a principal submatrix.*

Proof. We prove the theorem in the case when A is strictly DD; the DD case follows by a continuity argument, noting that volumes of submatrices are continuous in A. Let $A(I,J)$ be a $k \times k$ submatrix of A. Also, by applying a suitable permutation to the rows and columns of A, we may assume that $I = (1,\ldots,k)$ and $J = (k-d+1,\ldots,2k-d)$ with $d = |I \cap J|$. The result of the theorem follows if we can prove

$$|\det(A(I,J))| \leq |\det(A(I,I))|. \tag{3.4}$$

For this purpose, we note that the LU factorization $A = LU$ always exists with U strictly DD; see Theorem 9.9 in [109]. We have that

$$A(I,I) = L(I,I)U(I,I), \quad A(I,J) = L(I,I)U(I,J).$$

As $L(I,I)$ is lower triangular with ones on the diagonal, we obtain

$$|\det(A(I,I))| = |\det(U(I,I))|, \quad |\det(A(I,J)| = |\det(U(I,J))|.$$

Thus, the inequality (3.4) follows from Lemma 3.3. □

For $k = n-1$, the result of Theorem 3.4 is covered in the proof of Theorem 2.5.12 in [111], while the result of Lemma 3.3 for $k = n-1$ follows from Proposition 2.1 in [155].

3.2 Adaptive Cross Approximation

In the following, we summarize the idea behind Bebendorf's cross approximation algorithm [13]. For this purpose, we first recall that a cross approximation of $A \in \mathbb{R}^{m\times n}$ (see

Definition 2.3) is closely connected to an incomplete LU decomposition of A. To see this, suppose that A has been permuted such that $I = J = (1, \ldots, k)$ and partition

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad A_{11} \in \mathbb{R}^{k \times k}.$$

Assume that A_{11} is invertible and admits an LU decomposition $A_{11} = L_{11}U_{11}$, where L_{11} is lower triangular and U_{11} is upper triangular with ones on the diagonal. By setting $L_{21} = A_{21}U_{11}^{-1}$ and $U_{12} = L_{11}^{-1}A_{12}$, we obtain

$$\begin{aligned} A &= A(:, J)A(I, J)^{-1}A(I, :) + \begin{bmatrix} 0 & 0 \\ 0 & A^{(k)} \end{bmatrix}, \\ &= \begin{bmatrix} L_{11} \\ L_{21} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & A^{(k)} \end{bmatrix} \qquad (3.5) \\ &= \begin{bmatrix} L_{11} & 0 \\ L_{21} & I_{m-k} \end{bmatrix} \begin{bmatrix} I_k & 0 \\ 0 & A^{(k)} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & I_{n-k} \end{bmatrix}, \qquad (3.6) \end{aligned}$$

where I_h denotes the identity matrix of size $h \times h$ and $A^{(k)} \in \mathbb{R}^{(m-k)\times(n-k)}$ is the Schur complement

$$A^{(k)} := A_{22} - A_{21}A_{11}^{-1}A_{12}.$$

This shows that the approximation error is governed by $A^{(k)}$ and that the factorized form (3.5) corresponds exactly to what is obtained after applying k steps of the LU factorization to A, see, e.g., [86, Chapter 3.2].

The first step of the greedy method for volume maximization consists in choosing indices (i_1, j_1) that maximize $|A(i_1, j_1)|$. Given index sets I and J of cardinality h for some $h \geq 1$, the next step of the greedy method for volume maximization consists of choosing indices such that

$$(i_{h+1}, j_{h+1}) = \arg\max \Big\{ \Big| \det\Big(A((I, i), (J, j)) \Big) \Big| : i \notin I, j \notin J \Big\}. \qquad (3.7)$$

Let us assume that $I = J = (1, \ldots, h)$ and set $\tilde{I} = (I, h + \tilde{\imath})$, $\tilde{J} = (J, h + \tilde{\jmath})$ for some $\tilde{\imath}, \tilde{\jmath} \in \{1, \ldots, n - h\}$. Then (3.6) implies

$$\det\Big(A(\tilde{I}, \tilde{J})\Big) = \det\Big(A(I, J)\Big) \cdot A^{(h)}(\tilde{\imath}, \tilde{\jmath}).$$

In other words, the local optimization problem (3.7) is solved by searching the entry of $A^{(h)}$ that has maximum modulus. This choice leads to Algorithm 3.1, which is equivalent to applying LU factorization with complete pivoting to A.

Algorithm 3.1 Cross approximation with complete pivoting [13]

Input: Matrix $A \in \mathbb{R}^{m \times n}$, desired rank k
Output: Index sets (tuples) I, J of cardinality k

1: Initialize $R_0 \leftarrow A$, $I \leftarrow ()$, $J \leftarrow ()$.
2: **for** $h = 0, \ldots, k - 1$ **do**
3: $\quad (i_{h+1}, j_{h+1}) \leftarrow \arg\max_{i,j} |R_h(i, j)|$
4: $\quad I \leftarrow (I, i_{h+1})$, $J \leftarrow (J, j_{h+1})$
5: $\quad p_{h+1} \leftarrow R_h(i_{h+1}, j_{h+1})$
6: $\quad R_{h+1} \leftarrow R_h - \frac{1}{p_{h+1}} R_h(:, j_{h+1}) R_h(i_{h+1}, :)$
7: **end for**

Remark 3.5. *Because of* (3.5)*, the remainder term of Algorithm 3.1 at each step satisfies* $R_h = \left[\begin{smallmatrix} 0 & 0 \\ 0 & A^{(h)} \end{smallmatrix}\right]$ *after a suitable permutation of the indices. Both for SPSD and DD matrices, the element of maximum modulus is on the diagonal. Positive definiteness and diagonal dominance are preserved by taking Schur complements; see, e.g., [112, Section 7.7 and Problem 6.1.P16]. In turn, the search for the pivot element in Step 3 can be restricted to the diagonal for such matrices. This significantly reduces the number of entries of* A *that need to be evaluated when running Algorithm 3.1. It also implies that Algorithm 3.1 returns* $I = J$*, which aligns nicely with the results from Section 3.1. Notice that if* A *is an SPSD matrix then the cross approximation*

$$A(:, I) A(I, I)^{-1} A(I, :)$$

obtained by Algorithm 3.1 is SPSD. In contrast, when Algorithm 3.1 is applied to a DD matrix, in general the low-rank approximation obtained by the index sets returned by the algorithm is not DD.

3.2.1 Error analysis for general matrices

Although not desirable, it may happen that the pivots p_h in Algorithm 3.1 grow. Upper bounds on the *growth factor* $\|A^{(h)}\|_{\max}/\|A\|_{\max}$ play an important role in the error analysis of Gaussian elimination (see e.g. [190]). In the setting of complete pivoting, we can define

$$g_h := \sup_A \left\{ \|A^{(h)}\|_{\max}/\|A\|_{\max} \right\}, \tag{3.8}$$

where the supremum is taken over all matrices of rank at least h. This condition ensures that there is no breakdown in the first h steps of Algorithm 3.1. By definition, $1 \leq g_1 \leq g_2 \leq \ldots \leq g_h$. Wilkinson [190] proved that

$$g_h \leq \sqrt{h+1} \cdot \sqrt{2 \cdot 3^{1/2} \cdot 4^{1/3} \cdot \ldots \cdot (h+1)^{1/h}} \leq 2\sqrt{h+1}(h+1)^{\log(h+1)/4}.$$

but it is known that this bound cannot be attained for $h \geq 3$. We use the notation "log" to indicate the natural logarithm. For matrices occurring in practice, it is rare to see any significant growth and it is not unreasonable to consider $g_h = \mathcal{O}(1)$; we refer to [109, Section 9.4] for more details. Extending the proof of [106, Theorem 3.2], we obtain the following result.

Theorem 3.6. *Let $A \in \mathbb{R}^{m \times n}$ have rank at least $k < \min\{m, n\}$. Then the index sets I, J returned by Algorithm 3.1 satisfy*

$$\|E_{IJ}\|_{\max} = \|A - A(:, J)A(I, J)^{-1}A(I, :)\|_{\max} \leq 4^k \cdot g_k \cdot \sigma_{k+1}(A). \tag{3.9}$$

Proof. Without loss of generality, we may assume $I = J = (1, \ldots, k)$. We perform one more step of Algorithm 3.1 and consider the relation $A_{11} = L_{11}U_{11}$ from (3.5) for $h = k+1$. Because of complete pivoting, the element of largest modulus in the jth column of L_{11} is on the diagonal and equals p_j. For such triangular matrices, Theorem 6.1 in [107] gives

$$\|L_{11}^{-1}\|_2 \leq 2^k \cdot \min\{|p_1|, \ldots, |p_{k+1}|\}^{-1}.$$

Analogously, using that the element of largest modulus in every row of U_{11} is on the diagonal and equals 1, we obtain $\|U_{11}^{-1}\|_2 \leq 2^k$. Hence,

$$\|A_{11}^{-1}\|_2 = \|U_{11}^{-1}L_{11}^{-1}\|_2 \leq 4^k \cdot \min\{|p_1|, \ldots, |p_{k+1}|\}^{-1}.$$

This implies

$$\min\{|p_1|, \dots, |p_{k+1}|\} \leq 4^k \|A_{11}^{-1}\|_2^{-1} = 4^k \sigma_{k+1}(A_{11}) \leq 4^k \sigma_{k+1}(A), \tag{3.10}$$

where we used interlacing properties of singular values, see [112, Corollary 7.3.6].

On the other hand, as $A^{(k)}$ is the matrix obtained after j steps of Algorithm 3.1 applied to the matrix $A^{(k-j)}$, the definition (3.8) gives the inequalities

$$p_{k+1} = \|A^{(k)}\|_{\max} \leq g_j \cdot \|A^{(k-j)}\|_{\max} = g_j \cdot |p_{k-j+1}| \leq g_k \cdot |p_{k-j+1}|$$

for $j = 1, 2, \dots, k$. We therefore obtain

$$\begin{aligned} &\|A - A(:, J)A(I, J)^{-1}A(I, :)\|_{\max} \\ &= \|A^{(k)}\|_{\max} = |p_{k+1}| \leq g_k \min\{|p_1|, \dots, |p_{k+1}|\}. \end{aligned} \tag{3.11}$$

Combined with (3.10), this shows the result of the theorem. □

Because of the factor 4^k, Theorem 3.6 only guarantees good low-rank approximations when the singular values are strongly decaying. This limitation does not correspond to the typical behavior observed in practice; the quantities $\|L_{11}^{-1}\|_2$ and $\|U_{11}^{-1}\|_2$ rarely assume the exponential growth estimates used in the proof of Theorem 3.6. In turn, the factor 4^k usually severely overestimates the error. Nevertheless, there are examples for which the error estimate of Theorem 3.6 is asymptotically tight; see Section 3.2.2 below.

Remark 3.7. *Note that Equation* (3.10) *ensures that among* $|p_1|, \dots, |p_{k+1}|$*, there exists at least one which is bounded by* $4^k \sigma_{k+1}(A)$*. In turn, a stronger statement is possible: The minimum of all cross approximation errors within the first* k *steps of Algorithm 3.1 is bounded by* $4^k \sigma_{k+1}(A)$ *and hence the growth factor* g_k *is avoidable.*

A result closely related to Theorem 3.6 has been shown in [108] in the context of perturbation analysis of LU factorizations. This result, however, requires the following additional assumptions on A. Letting A_k denote the best rank-k approximation of A in the spectral norm, it is assumed in [108] that Algorithm 3.1 applied to A and A_k returns the same index sets I, J and, moreover, $\left\|A_k(I, J)^{-1}\left(A(I, J) - A_k(I, J)\right)\right\|_2 < 1$. Then,

$$\|E_{IJ}\|_2 \leq \frac{1}{3}(4^k - 1)\sqrt{(n-k)(m-k)} \cdot \sigma_{k+1}(A) + O(\sigma_{k+1}(A)^2);$$

see Lemma 2.1 and Lemma 2.3, and the discussion in Section 5.3 in [108].

On mixed norms

In the following we develop a variant of Theorem 3.6 in which the best approximation error is also measured in terms of $\|\cdot\|_{\max}$. This will be useful later on, in Section 3.2.4, when considering approximation of functions. Recall that $\delta_k(A)$ denotes the error of the best rank-k approximation in the Chebyshev norm (see Eqn. (2.2)). If A is square and invertible then $\sigma_n(A) = \|A^{-1}\|_2^{-1}$. This result, relating the distance to singularity to the norm of the inverse, extends to general subordinate matrix norms; see, e.g., [109, Theorem 6.5]. In particular, we have

$$\delta_n(A) = \|A^{-1}\|_{\infty\to 1}^{-1}, \tag{3.12}$$

with $\|\cdot\|_{\infty\to 1}$ denoting the matrix norm induced by the 1- and ∞-norms. More generally, we set

$$\|B\|_{\alpha\to\beta} := \sup_{x\neq 0} \|Bx\|_\beta / \|x\|_\alpha$$

for vector norms $\|\cdot\|_\alpha$, $\|\cdot\|_\beta$. Note that $\|B\|_{1\to\infty} = \|B\|_{\max}$.

Theorem 3.8. *Under the assumptions of Theorem 3.6 and with the notation introduced above, we have*

$$\|E_{IJ}\|_{\max} \leq 2^{2k+1} \cdot g_k \cdot \delta_{k+1}(A).$$

Proof. Along the lines of the proof of Theorem 3.6, we first note that

$$\|L_{11}^{-1}\|_{\infty\to 1} \leq (2^{k+1} - 1) \min\{|p_1|, \ldots, |p_{k+1}|\}^{-1},$$

which can be shown by induction over k. Combined with $\|U_{11}^{-1}\|_{1\to 1} \leq 2^k$, see [107, Theorem 6.1], we obtain

$$\begin{aligned}\|A_{11}^{-1}\|_{\max} = \|A_{11}^{-1}\|_{\infty\to 1} &\leq \|U_{11}^{-1}\|_{1\to 1} \|L_{11}^{-1}\|_{\infty\to 1} \\ &\leq 2^{2k+1} \min\{|p_1|, \ldots, |p_{k+1}|\}^{-1},\end{aligned}$$

where we used submultiplicativity [109, Eqn (6.7)]. Using (3.12), this implies

$$\min\{|p_1|, \ldots, |p_{k+1}|\} \leq 2^{2k+1} \|A_{11}^{-1}\|_{\infty\to 1}^{-1} = 2^{2k+1} \delta_{k+1}(A_{11}) \leq 2^{2k+1} \delta_{k+1}(A).$$

The rest of the proof is identical with the proof of Theorem 3.6. □

3.2.2 Error analysis for SPSD matrices

In the SPSD case, the pivot elements of Algorithm 3.1 are always non-increasing. Thus, when restricting the supremum in (3.8) to SPSD matrices of rank at least k, one obtains $g_k = 1$. In turn, the following result from [74, 106] is a corollary of Theorem 3.6.

Corollary 3.9. *For an SPSD matrix A of rank at least k, the bound of Theorem 3.6 improves to*

$$\|E_{IJ}\|_{\max} \leq 4^k \cdot \sigma_{k+1}(A).$$

The bound of Corollary 3.9 is asymptotically tight, see [106, Remark 3.3] and [117, p. 791]. As the growth factor g_k which comes into play in Theorem 3.6 is small compared to the 4^k factor, this also proves that the bound of Theorem 3.6 is almost tight.

3.2.3 Error analysis for DD matrices

When restricting the supremum in (3.8) to DD matrices of rank at least k, one obtains $g_k \leq 2$; see Theorem 13.8 in [109].

Corollary 3.10. *For a DD matrix $A \in \mathbb{R}^{n\times n}$ of rank at least k, the bound of Theorem 3.6 improves to*

$$\|E_{IJ}\|_{\max} \leq (k+1) \cdot 2^{k+1} \cdot \sigma_{k+1}(A).$$

Proof. It is well known that the factor U in the LU decomposition of a DD matrix is again DD; see [112, Problem 6.1.P16]. In particular, this implies that the $(k+1)\times(k+1)$ unit upper triangular matrix U_{11} in the proof of Theorem 3.6 is DD. Then, for every entry of U_{11}^{-1} we have $|(U_{11}^{-1})_{ij}| \leq 1$ by [155, Prop. 2.1]. Therefore,

$$\|U_{11}^{-1}\|_2 \leq \|U_{11}^{-1}\|_F \leq \sqrt{(k+1)(k+2)/2} \leq k+1. \tag{3.13}$$

This shows that the factor 4^k can be reduced to $(k+1)2^k$ in the bound of Theorem 3.6. Combined with $g_k \leq 2$, this establishes the desired result. □

Corollary 3.11. *For a doubly DD matrix A of rank at least k, the bound of Corollary 3.10*

improves to

$$\|E_{IJ}\|_{\max} \leq 2 \cdot (k+1)^2 \cdot \sigma_{k+1}(A).$$

Proof. Trivially, A^T is DD. By the same arguments as in the proof of Corollary 3.10 this implies that not only U_{11} but also L_{11}^T is DD. Proceeding as in the derivation of (3.13), we get

$$\|L_{11}^{-1}\|_2 \leq \|L_{11}^{-1}\|_F \leq (k+1) \cdot \min\{|p_1|, \ldots, |p_{k+1}|\}^{-1}.$$

This shows that the factor $(k+1) \cdot 2^{k+1}$ of Corollary 3.10 can be improved to $2(k+1)^2$. □

Remark 3.12. *The fact that Algorithm 3.1 gives a polynomially good low-rank approximation of a doubly DD matrix does not imply that it also gives a polynomially good approximation of the maximum volume submatrix. For instance, let $n = 2k$ and consider $A = \begin{bmatrix} I_k & 0 \\ 0 & B_k \end{bmatrix}$, where $B_k = \operatorname{tridiag}\left(\frac{1}{2}, 1, -\frac{1}{2}\right)$. Then Algorithm 3.1 does not perform any pivoting during its k steps and thus the submatrix I_k is selected. Its volume is 1, while the volume of B_k is exponentially larger, it grows like $\left(\frac{1+\sqrt{2}}{2}\right)^k$.*

For the particular case of doubly DD matrices, we have shown that the approximation error returned by cross approximation is at most $2(k+1)$ times larger than the right-hand side of (2.4) (when $\gamma = 1$). This class of matrices includes symmetric DD matrices, which play a prominent role in [121, 173].

Tightness of estimates for DD matrices

To study the tightness of the estimates from Section 3.2.3, it is useful to connect Algorithm 3.1 to LDU decompositions. From now on, let $A \in \mathbb{R}^{n \times n}$ be a DD matrix and let $k = n - 1$. Suppose that the application of k steps of Algorithm 3.1 yields $I = J = (1, \ldots, n-1)$. As in the proof of Theorem 3.6, we exploit the relation (3.5) for $k + 1 = n$ to obtain the factorization

$$A = L_{11}U_{11} = LDU, \quad D := \operatorname{diag}(p_1, \ldots, p_n)$$

where $L := L_{11}D^{-1}$ and $U := U_{11}$ are lower and upper unit triangular matrices, respectively. We recall from (3.11) that the error of the approximation returned by Algorithm 3.1 is governed by $|p_n|$.

As A is DD, the pivot growth factor does not exceed 2 and we have that $|p_n| \leq 2\|D^{-1}\|_2^{-1} \leq 2|p_n|$. In turn, the ratio between $|p_n|$ and the best rank-$(n-1)$ approximation error satisfies

$$r_k := \frac{|p_n|}{\sigma_n(A)} = |p_n|\,\|A^{-1}\|_2 \leq |p_n|\,\|U^{-1}\|_2\|D^{-1}\|_2\|L^{-1}\|_2 \leq 2\|L^{-1}\|_2\|U^{-1}\|_2. \tag{3.14}$$

Inheriting the diagonal dominance from A, the matrix U is well conditioned; see (3.13). Therefore, large r_k requires $\|L^{-1}\|_2$ to become large.

The quantity $\|L^{-1}\|_2$ also plays a prominent role in the stability analysis of LDU decompositions, see [61] and the references therein. In particular, the *potential* rapid growth of $\|L^{-1}\|_2$ under complete pivoting has motivated the search for alternative pivoting strategies [155]. However, the existing literature is scarce on examples actually exhibiting such rapid growth. The worst example we could find is by Barreras and Peña [11, Sec. 3], which exhibits linear growth. A more rapid growth is attained by the $n \times n$ matrix

$$A = \left[\begin{array}{cccc|cccc}
1 & -1 & & & & & & \\
 & 1 & \ddots & & & & & \\
 & & \ddots & -1 & & & & \\
 & & & 1 & -\frac{1}{n/2+1} & -\frac{1}{n/2+1} & \cdots & -\frac{1}{n/2+1} \\
\hline
-1 & & & & 1 & & & \\
-1 & & & & & 1 & & \\
\vdots & & & & & & \ddots & \\
-1 & & & & & & & 1
\end{array}\right],$$

where n is even and each block has size $n/2 \times n/2$. When applying complete pivoting to this matrix, no interchanges are performed and the LDU factorization satisfies

$$\|L^{-1}\|_2 = \Theta(k\sqrt{k}), \quad \|D^{-1}\|_2 = 1/|p_n| = 2, \quad \|U^{-1}\|_2 = \Theta(k).$$

Note that, for this example, the right-hand side of (3.14) overestimates the error. This example attains quadratic growth: $r_k = \|A^{-1}\|_2 = \Theta(k^2)$. This is still far away from the exponential growth estimated in Corollary 3.10, but closer than the example from [11,

Sec. 3], which yields $r_k = \Theta(k\sqrt{k})$.

For a doubly DD matrix, one obtains linear growth in (3.14) by considering the $n \times n$ lower bidiagonal matrix B having 1 on the diagonal and -1 on the first subdiagonal. In this case, $L = B$, $D = U = I_n$ and hence $\|B_n^{-1}\|_2 = \Theta(k)$, showing that r_k can grow at least linearly with k. We have not found an example exhibiting the quadratic growth estimated by Corollary 3.11.

3.2.4 Cross approximation for functions

Let us consider the approximation of a function $f : [-1, 1]^2 \to \mathbb{R}$ by a linear combination of separable functions:

$$f(x, y) \approx \sum_{i=1}^{i_{\max}} c_i \cdot f_i^{(1)}(x) \cdot f_i^{(2)}(y), \qquad c_i \in \mathbb{R}.$$

In the context of *cross approximation*, the factors are restricted to functions $f_i^{(1)}$ of the form $f_i^{(1)} = f(x, \bar{y}_i)$ and $f_i^{(2)} = f(\bar{x}_i, y)$, where $\bar{x}_i$ and $\bar{y}_i$ are fixed elements of $[-1, 1]$. In particular, Micchelli and Pinkus [141] considered interpolating approximations of the following form:

$$f(x, y) \approx \begin{bmatrix} f(x, y_1) \\ \vdots \\ f(x, y_k) \end{bmatrix}^T \cdot \begin{bmatrix} f(x_1, y_1) & \cdots & f(x_1, y_k) \\ \vdots & & \vdots \\ f(x_k, y_1) & \cdots & f(x_k, y_k) \end{bmatrix}^{-1} \cdot \begin{bmatrix} f(x_1, y) \\ \vdots \\ f(x_k, y) \end{bmatrix},$$

for some $x_1, \ldots, x_k, y_1, \ldots, y_k \in [-1, 1]$. Townsend and Trefethen [180] use a strategy for choosing the interpolation points which is basically equivalent to Algorithm 3.1 and they prove a convergence result under some analyticity hypotheses on the function f. There also exist error analyses for cross approximation of functions when using different pivoting strategies, see, e.g., [15, 168].

Let us explain how the greedy strategy of Algorithm 3.1 can be translated to the function setting. Choosing the first pivot corresponds to finding a point $(x_1, y_1) \in [-1, 1]^2$ which maximizes the absolute value of f. The rank-1 separable function approximating f

is given by

$$f_1(x,y) := \frac{1}{f(x_1,y_1)} f(x_1,y) f(x,y_1)$$

and the first "residual" function is defined as

$$e_1(x,y) := f(x,y) - f_1(x,y).$$

The next step greedily chooses a point (x_2, y_2) that maximizes the absolute value of e_1 and defines functions e_2, f_2 accordingly. Algorithm 3.2 summarizes cross approximation of functions with complete pivoting.

Algorithm 3.2 Cross approximation of functions [179, Figure 2.1]

Input: $f : [-1,1]^2 \to \mathbb{R}$ and $k > 0$

1: $e_0(x,y) \leftarrow f(x,y)$
2: $f_0(x,y) \leftarrow 0$
3: **for** $h = 0, \dots, k-1$ **do**
4: $\quad (x_{h+1}, y_{h+1}) \leftarrow \arg\max_{(x,y)\in[-1,1]^2} \{|e_h(x,y)|\}$
5: $\quad e_{h+1} \leftarrow e_h - \frac{e_h(x_{h+1},\cdot)\cdot e_h(\cdot,y_{h+1})}{e_h(x_{h+1},y_{h+1})}$
6: $\quad f_{h+1} \leftarrow f_h + \frac{e_h(x_{h+1},\cdot)\cdot e_h(\cdot,y_{h+1})}{e_h(x_{h+1},y_{h+1})}$
7: **end for**

We now explain in more detail the connection of Algorithm 3.2 to Algorithm 3.1, which allows us to prove a bound on the error e_k of the separable approximation obtained after k steps of Algorithm 3.2. Fix $(x,y) \in [-1,1]^2$ and consider the points $x_1, \dots, x_k$ and $y_1, \dots, y_k$ obtained by Algorithm 3.2. Consider what happens when applying Algorithm 3.1 to the $(k+1) \times (k+1)$ matrix obtained by interpolating f in the points mentioned above:

$$A_{(x,y)} := \begin{bmatrix} f(x_1,y_1) & \cdots & f(x_1,y_k) & f(x_1,y) \\ \vdots & & \vdots & \vdots \\ f(x_k,y_1) & \cdots & f(x_k,y_k) & f(x_k,y) \\ f(x,y_1) & \cdots & f(x,y_k) & f(x,y) \end{bmatrix}.$$

The first chosen pivot will be $p_1 = f(x_1,y_1)$ because it is the largest entry of the matrix. Now observe that the Schur complement $A^{(1)}$ obtained after the first step, is the matrix that interpolates the function e_1 in the points $x_2, \dots, x_k, x$ and $y_2, \dots, y_k, y$. At this point, the second pivot chosen by Algorithm 3.1 will be $e_1(x_2, y_2)$ because of how Algorithm 3.2

choses (x_2, y_2) in line 5. After k steps of Algorithm 3.1 we will be left with only one nonzero entry in position $(k+1, k+1)$ and this will be $e_k(x, y)$. This allows us to estimate $|e_k(x, y)|$ via Theorem 3.8:

$$|e_k(x, y)| \leq 2^{2k+1} \cdot g_k \cdot \delta_{k+1}(A_{(x,y)}). \tag{3.15}$$

The last thing we need is an estimate on $\delta_{k+1}(A_{(x,y)})$ that is uniform in $(x, y) \in [-1, 1]^2$. This will follow from analyticity assumptions on the functions $f(\cdot, y)$ for $y \in [-1, 1]$.

Definition 3.13. *The Bernstein ellipse $\mathcal{E}_r$ of radius $r > 1$ is the ellipse in $\mathbb{C}$ with foci in -1 and 1 and with sum of the semi-axes equal to r.*

Corollary 3.14. *Let $f : [-1, 1]^2 \to \mathbb{R}$ be such that $f(\cdot, y)$ admits an analytic extension – which we will denote by $\tilde{f}$ – in the Bernstein ellipse $\mathcal{E}_{r_0}$ of radius r_0 for each $y \in [-1, 1]$. Let $1 < r < r_0$ and*

$$M := \sup_{\eta \in \partial\mathcal{E}_r,\, \xi \in [-1,1]} |\tilde{f}(\eta, \xi)|,$$

where $\partial\mathcal{E}_r$ denotes the boundary of the ellipse. After k steps of Algorithm 3.2 the error function satisfies

$$\|e_k\|_{\max} \leq \frac{2Mg_k}{1 - 1/r} \cdot \left(\frac{r}{4}\right)^{-k}.$$

Proof. Fix $(x, y) \in [-1, 1]^2$ and let $b : [-1, 1] \to \mathbb{R}^{k+1}$ be the vector-valued function defined by

$$b(\eta) := \begin{bmatrix} f(\eta, y_1) & \cdots & f(\eta, y_k) & f(\eta, y) \end{bmatrix}^T.$$

The analyticity hypothesis allows us to apply standard polynomial approximation results (see e.g. Corollary 2.2 in [125]) and conclude that there exists an approximation $\hat{b} : [-1, 1] \to \mathbb{R}^{k+1}$ given by

$$\hat{b}(\eta) = \sum_{h=1}^{k} p_h(\eta) v_h, \tag{3.16}$$

where $v_h \in \mathbb{R}^{k+1}$ are constant vectors and $p_h : [-1, 1] \to \mathbb{R}$ are polynomials, such that

$$\max \|b(\eta) - \hat{b}(\eta)\|_{\max} \leq \frac{2}{1 - r^{-1}} \cdot \max_{\alpha \in \mathcal{E}_r} \|b(\alpha)\|_{\max} \cdot r^{-k}$$

for any $1 < r < r_0$. We can clearly bound $\max_{\alpha \in \mathcal{E}_r} \|b(\alpha)\|_{\max} \leq M$.

The matrix $A_{(x,y)}$ is obtained by sampling b in the points $x_1, \ldots, x_k, x$, i.e.

$$A_{(x,y)} = \begin{bmatrix} b(x_1) & \cdots & b(x_k) & b(x) \end{bmatrix}.$$

Let us define, analogously,

$$\hat{A}_{(x,y)} = \begin{bmatrix} \hat{b}(x_1) & \cdots & \hat{b}(x_k) & \hat{b}(x) \end{bmatrix}.$$

Notice that $\hat{A}_{(x,y)}$ has rank as most k because by (3.16) each of the $k+1$ columns of $\hat{A}_{(x,y)}$ is a linear combination of the k vectors $v_1, \ldots, v_k$, so

$$\delta_{k+1}(A_{(x,y)}) \leq \|A_{(x,y)} - \hat{A}_{(x,y)}\|_{1\to\infty} = \max_{\alpha \in \{x_1,\ldots,x_k,x\}} \|b(\alpha) - \hat{b}(\alpha)\|_{\max} \leq \frac{2M}{1-r^{-1}} \cdot r^{-k}.$$

The result then follows from Equation (3.15). □

To get convergence of the error function to zero as $k \to \infty$, in Corollary 3.14 it is sufficient that the function $f(\cdot, y)$ admits an analytic extension to the Bernstein ellipse $\mathcal{E}_{r_0}$ with $r_0 > 4$ for each y, because the factor g_k has subexponential growth. Our result compares favorably to Theorem 8.1 in [180], which requires an analytic extension to the region K consisting of all points at a distance ≤ 4 from $[-1, 1]$. Figure 3.1 compares the two domains and it is evident that the requirement from [180] is significantly more restrictive.

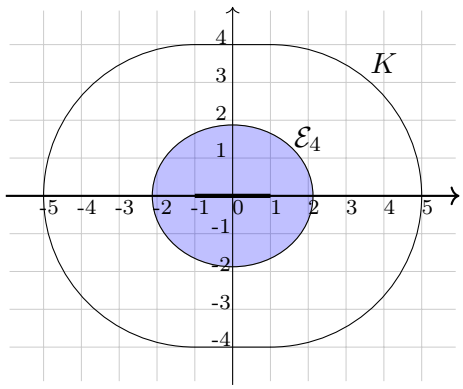


Figure 3.1 – Analyticity regions ensuring convergence of Algorithm 3.2 according to Corollary 3.14 and [180, Theorem 8.1].

For a positive semidefinite kernel function f, the matrix $A_{(x,y)}$ in (3.15) is positive semidefinite and hence the bound of Corollary 3.14 holds with $g_k \equiv 1$. This matches an asymptotic result given in [180, Theorem 9.1].

3.3 Improving ACA to achieve a polynomial error bound

The results for ACA in Section 3.2 are far from the guarantees of Theorem 2.6. However, successive row and column "swaps" can be performed after obtaining initial index sets I and J from ACA. The idea is that if the error E_{IJ} has an entry (i, j) which is larger, in absolute value, than $\gamma(k+1)\sigma_{k+1}(A)$, then the $(k+1)\times(k+1)$ matrix $\hat{A} := A((I, i), (J, j))$ contains a $k \times k$ submatrix whose volume is larger than $\operatorname{Vol}(A(I, J))$ by a factor larger than γ. In such case, one can update the index sets I and J taking those corresponding to this submatrix of larger volume. The resulting procedure is summarized in Algorithm 3.3.

Algorithm 3.3 Cross approximation with swaps.

Input: $A \in \mathbb{R}^{m\times n}$, rank k, parameter $\gamma > 1$
Output: Index sets I and J of cardinality k such that (2.4) holds.

```
Initialize (I_new, J_new) ← ACA(A, k) using Algorithm 3.1
repeat
    I_old ← I_new, J_old ← J_new
    (i, j) ← indices of largest element of E_{I_old J_old}
    (I_new, J_new) ← maximum volume k × k submatrix of Â := A((I_old, i), (J_old, j))
until Vol A(I_new, J_new) / Vol A(I_old, J_old) ≤ γ
I ← I_old, J ← J_old.
```

First of all, let us show that Algorithm 3.3 achieves (2.4).

Lemma 3.15. *The index sets I and J returned by Algorithm 3.3 satisfy*

$$\|E_{IJ}\|_{\max} \le \gamma(k+1)\sigma_{k+1}(A) \text{ and } \|E_{IJ}\|_{\max} \le \gamma(k+1)^2\delta_{k+1}(A).$$

Proof. Let (i, j) be the indices selected in Algorithm 3.3 the last time that the cycle has been executed. As the algorithm stopped after that, in the matrix $\hat{A} := A((I, i), (J, j))$ we have $\operatorname{Vol} A(I, J) = \operatorname{Vol} \hat{A}(I, J) \ge \frac{1}{\gamma} \operatorname{maxVol}(\hat{A})$. Therefore, by Theorem 2.6 applied to $\hat{A}$, we have

$$\|\hat{A} - \hat{A}(:, J)A(I, J)^{-1}\hat{A}(I, :)\|_{\max} \le \gamma(k+1)\sigma_{k+1}(\hat{A}).$$

Noting that $\|\hat{A} - \hat{A}(:,J)A(I,J)^{-1}\hat{A}(I,:)\|_{\max} = |E_{IJ}(i,j)| = \|E_{IJ}\|_{\max}$, $\sigma_{k+1}(\hat{A}) \leq \sigma_{k+1}(A)$ (by interlacing properties of singular values), and $\sigma_{k+1}(\hat{A}) \leq (k+1)\delta_{k+1}(\hat{A}) \leq (k+1)\delta_{k+1}(A)$ concludes the proof. □

The idea of performing successive swaps is not new. This has been done, for example, in the context of finding local quasi-maximum volume submatrices for (strong) rank-revealing factorizations [96, 97, 153, 169], and for cross approximation in [87, 150]. The algorithm proposed in [87, 150] differs from Algorithm 3.3 in that it performs swaps until $A(I,J)$ is a local quasi-maximum volume submatrix in $A(I,:)$ and $A(:,J)$. Their strategy is faster but it does not guarantee that the final index sets satisfy (2.4).

3.3.1 Time complexity of Algorithm 3.3

In this section we show that Algorithm 3.3 takes polynomial time. An ingredient of our analysis is a bound on the number of swaps S, which will follow from a bound on the suboptimality of the volume of the submatrix given by ACA.

Theorem 3.16. *Let $A \in \mathbb{R}^{m\times n}$ $(m \leq n)$ and $k < m$. Assume (without loss of generality) that ACA with full pivoting applied to A (Algorithm 3.1) selects the leading submatrix A_{11}. Then, for any index sets I and J of cardinality k (in particular, for those corresponding to the maximum volume submatrix of A), we have*

$$|\det(A(I,J))| \leq m^k k^k 2^k (m+1)^{k\left(\frac{1}{2}+\frac{1}{4}\log(m+1)\right)} |\det(A_{11})|.$$

Proof. Without loss of generality, we assume that no permutation matrices are needed in ACA with full pivoting. If we do all n steps, we obtain a decomposition $A = LDU$ where $L \in \mathbb{R}^{m\times m}$ and $U \in \mathbb{R}^{m\times n}$ are triangular matrices, with ones on the diagonal and all other entries of absolute value ≤ 1; the diagonal matrix $D \in \mathbb{R}^{m\times m}$ contains the pivots $p_1, \ldots, p_m$.

We have that $A(I,J) = L(I,:) \cdot D \cdot U(:,J)$. Using the Cauchy-Binet formula for the determinant of a product, and the fact that the non-principal submatrices of D are singular, we obtain

$$\det(A(I,J)) = \sum_{\substack{|K|=k \\ \text{non-ordered sets } K}} \det(L(I,K)) \cdot \det(D(K,K)) \cdot \det(U(K,J)).$$

We are going to obtain an upper bound on $|\det A(I,J)|$ by bounding the terms in the previous sum separately. First, the number of unordered sets $K \subset \{1,\dots,m\}$ of cardinality k is $\binom{m}{k} \le m^k$. As the rows of $L(I,K)$ have norm bounded by $k^{1/2}$, we have $|\det(L(I,K))| \le k^{k/2}$ by Hadamard inequality. Similarly, $|\det(U(K,J))| \le k^{k/2}$. Finally, to bound $|\det(D(K,K))|$, note that if $K = \{i_1,\dots,i_k\}$ with $i_1 < \dots < i_k$, we have $|p_{i_h}| \le g_m |p_h|$ where $g_m \le 2\sqrt{m+1}(m+1)^{\log(m+1)/4}$ is the growth factor of Gaussian elimination with complete pivoting (3.8). Therefore,

$$|\det(D(K,K))| = |p_{i_1}\cdots p_{i_k}| \le g_m^k \cdot |p_1\cdots p_k| \le 2^k (m+1)^{k\left(\frac{1}{2}+\frac{1}{4}\log(m+1)\right)} \cdot |p_1\cdots p_k|.$$

We obtain the upper bound

$$|\det(A(I,J))| \le k^k m^k |\det(D(K,K))| \le k^k m^k 2^k (m+1)^{k\left(\frac{1}{2}+\frac{1}{4}\log(m+1)\right)} \cdot |p_1\cdots p_k|.$$

To conclude, note that $|p_1\cdots p_k| = |\det(A_{11})|$. □

Corollary 3.17. *Denote by (I_0, J_0) the starting index pair obtained in line 1 of Algorithm 3.3 and by (I_S, J_S) the pair obtained after S swaps. Then*

$$S \le \mathcal{O}\left(\frac{k\log^2 m}{\log\gamma}\right).$$

Proof. Leveraging Theorem 3.16 and recalling that the volume of the selected submatrix $A(I_{\text{new}}, J_{\text{new}})$ grows at least by γ at every step, we have

$$\begin{aligned}\operatorname{maxVol}(A) &\ge |\det(A(I_S,J_S))| \ge \gamma^S |\det(A(I,J))| \\ &\ge \frac{\gamma^S}{m^k k^k 2^k (m+1)^{k\left(\frac{1}{2}+\frac{1}{4}\log(m+1)\right)}} \operatorname{maxVol}(A).\end{aligned}$$

Taking the logarithm gives the result. □

Computing $E_{I_{\text{old}}J_{\text{old}}}$ in line 4 costs $\mathcal{O}(nmk)$. Finding the maximum volume $k \times k$ submatrix in $\hat{A}$ in line 5 costs $\mathcal{O}(k^3)$ because we can use [142, Proposition 1] to compute the volume change with respect to $\operatorname{Vol} A(I_{\text{old}}, J_{\text{old}})$. Therefore, each cycle of Algorithm 3.3 can be done in $\mathcal{O}(nmk)$ time. Using Corollary 3.17 we conclude that the time complexity of Algorithm 3.3 is $\mathcal{O}\left(\frac{nmk^2\log^2 m}{\log\gamma}\right)$.

Note that one $\log m$ factor in the time complexity of Algorithm 3.3 is coming from the growth factor of Gaussian elimination with full pivoting, therefore it is usually negligible in practice.

Remark 3.18. *If A is SPSD, a local γ-maximum volume submatrix can be found in time $\mathcal{O}\left(\frac{nk^2 \log k}{\log \gamma}\right)$ (see [134, Algorithm 4]) and this ensures that* (2.4) *holds. In particular, a result analogous to Theorem 3.16 holds with the constant $(k!)^2$ instead of $m^k k^k 2^k (m+1)^{k\left(\frac{1}{2}+\frac{1}{4}\log(m+1)\right)}$; see [39] and the discussion in [134, Section 2.2.3].*

4 Low-rank approximation in the Frobenius norm by column and row subset selection

In this chapter we consider algorithms for the low-rank approximation of matrices and tensors which are guaranteed to have a polynomial error bound. We start in Section 4.1 by recalling the deterministic column subset selection algorithm from [57], which is obtained from the existence result (2.9) by derandomization with the technique of conditional expectations. The conditional expectations are given in terms of coefficients of certain characteristic polynomials and the algorithm from [57] attains efficiency by cheaply updating these coefficients. However, it is well known that working with characteristic polynomials in finite precision arithmetic is prone to massive numerical cancellation [161] and, as we will see, the algorithm from [57] is also affected by numerical instability. Our first contribution, presented in Section 4.1, consists of deriving a formulation of the algorithm that updates singular values instead of coefficients of characteristic polynomials. While our new variant enjoys the same favorable complexity, numerical experiments with matrices of different singular value decay indicate that it is numerically robust, achieving (2.10) even when the right-hand side $\sqrt{k+1}\sqrt{\sigma_{k+1}^2 + \ldots + \sigma_m^2}$ is at the level of unit roundoff. Based on a minor extension of the theory from [57, 58], we will also present a modification of the column selection strategy that results in significant speed ups of the algorithm.

In Section 4.2 we derive a result for CUR approximation using the column subset

selection algorithm from Section 4.1. This strategy results in an error that is at most a factor $\sqrt{2k+2}$ larger than the best rank-k approximation error in the Frobenius norm. One major contribution of this work is to derive a polynomial-time deterministic algorithm that guarantees a polynomial error bound for cross approximation in the Frobenius norm, via an extension of [57]; our algorithm matches the existence result (2.7).

Section 4.3 contains an extension to the Tucker decomposition of tensors, which is suitable for tensors of low order. In particular, we derive a deterministic algorithm that obtains a multilinear low-rank approximation that is constructed from the fibers of the tensor and satisfies a polynomial bound. Although our approach is a relatively straightforward extension of (2.11) and related approaches have been proposed in the literature [62, 90, 166], we are not aware that such an algorithm has been explicitly formulated and analyzed.

4.1 Column subset selection

We start by providing more details on the approach from [57, 58] for the column subset selection problem. In the following we assume that the matrix $A \in \mathbb{R}^{m\times n}$(with $m \leq n$) has rank at least k. We let a_i denote the ith column of A and $\pi_{i_1,\ldots,i_k}A$ the orthogonal projection of A on the subspace spanned by the columns $a_{i_1},\ldots,a_{i_k}$, that is,

$$\pi_{i_1,\ldots,i_k}A := A(:,I)\cdot A(:,I)^{\dagger}\cdot A = QQ^TA,$$

where $I = (i_1,\ldots,i_k) \in \{1,\ldots,n\}^k$ and Q denotes an orthonormal basis of $A(:,I)$. We emphasize that I is a tuple. Although order is not important and we are ultimately interested in an index *set*, working with tuples simplifies the subsequent definition and manipulation of probability distributions.

We now define a discrete probability distribution on integer tuples of the form $I \in \{1,\ldots,n\}^k$ corresponding to a selection of k columns from A. For this purpose, let $X = (X_1,\ldots,X_k)$ be a k-tuple of random variables with values in $\{1,\ldots,n\}$ such that

$$\mathbb{P}(X = I) := \frac{\mathrm{Vol}^2(A(:,I))}{\sum_{J\in\{1,\ldots,n\}^k}\mathrm{Vol}^2(A(:,J))}. \tag{4.1}$$

It follows from the definition that $\mathrm{Vol}\,(A(:,I)) = 0$ whenever $i_1,\ldots,i_k$ contain repeated

indices. Then [58, Theorem 1.3] shows that

$$\mathbb{E}[\|A - \pi_{X_1,\ldots,X_k} A\|_F^2] \leq (k+1)\left(\sigma_{k+1}^2 + \ldots + \sigma_m^2\right). \tag{4.2}$$

In particular, this implies the existence of I satisfying the bound (2.10).

In view of (4.1) and the prominent role played by maximum volume submatrices in low-rank approximation [88], it is tempting to expect that the k columns of maximum volume satisfy (2.10). However, choosing such columns is not only an NP-hard problem – as mentioned in Section 3.1.1 – but these might also fail to satisfy (2.10). For instance, for $k = 1$ consider the $2 \times n$ matrix

$$A = \begin{bmatrix} a(1+\varepsilon) & b & b & \cdots & b \\ -b(1+\varepsilon) & a & a & \cdots & a \end{bmatrix}$$

with $a^2 + b^2 = 1$ and $\varepsilon > 0$. The column of maximum volume (that is, of maximum Euclidean norm) is the first one. The approximation error obtained by this choice is given by $\|A - \pi_1 A\|_F^2 = n - 1$, which is much larger than $2\sigma_2^2 = 2(1+\varepsilon)^2$ for ε sufficiently small. Note that choosing any of the other columns yields the best approximation error $(1+\varepsilon)^2 = \sigma_2^2$.

4.1.1 Algorithm by Deshpande and Rademacher

Deshpande and Rademacher [57] derived a deterministic algorithm for column subset selection by derandomizing (4.2) using the method of conditional expectations.

More specifically, the first step of the algorithm chooses an index i_1 such that

$$\mathbb{E}\left[\|A - \pi_{X_1,\ldots,X_k} A\|_F^2 \mid X_1 = i_1\right]$$

is minimized. By construction, this quantity still satisfies the bound (4.2). More generally, having $t-1$ indices $i_1, \ldots, i_{t-1}$ selected, step t chooses an index i_t such that

$$\mathbb{E}\left[\|A - \pi_{X_1,\ldots,X_k} A\|_F^2 \mid X_1 = i_1, \ldots, X_{t-1} = i_{t-1}, X_t = i_t\right] \tag{4.3}$$

is minimized. After k steps we arrive at an index set I of cardinality k such that the

desired bound (2.10) holds.

For the algorithm to be practical, it is crucial to compute the conditional expectations (4.3) efficiently. Lemma 21 in [57] shows that

$$\mathbb{E}\left[\|A - \pi_{X_1,\ldots,X_k} A\|_F^2 \mid X_1 = i_1, \ldots, X_t = i_t\right] = (k-t+1)\frac{c_{m-k+t-1}(BB^T)}{c_{m-k+t}(BB^T)},$$

where the right-hand side involves the matrix $B = A - \pi_{i_1,\ldots,i_t} A$ and coefficients $c_j \equiv c_j(BB^T)$ of the characteristic polynomial

$$(-\lambda)^m + c_{m-1}(-\lambda)^{m-1} + \ldots + c_1(-\lambda) + c_0 := \det(BB^T - \lambda I). \tag{4.4}$$

It is therefore required to compute in every step for all values of i, the ratios

$$\frac{c_{m-k+t-1}(B_i B_i^T)}{c_{m-k+t}(B_i B_i^T)} \tag{4.5}$$

where $B_i = A - \pi_{i_1,\ldots,i_{t-1},i} A$.

In the following, we discuss the computation of (4.5) and show how the minimization problem (4.3) can be relaxed in order to accelerate the search for suitable indices.

4.1.2 Computation of characteristic polynomial coefficients

Assuming that the first $t-1$ indices have been selected, we set $B := A - \pi_{i_1,\ldots,i_{t-1}} A$. Then

$$B_i = B - \pi_i B = \left(I - \frac{b_i b_i^T}{\|b_i\|_2^2}\right) B$$

is a rank-1 modification of B. Deshpande and Rademacher [57] propose two methods to compute (4.5) for $i = 1, \ldots, n$. In the following, we summarize them briefly.

1. Algorithm 2 in [57] computes BB^T explicitly and then computes $B_i B_i^T$ as a rank-2 update of BB^T for every $i = 1, \ldots, n$. The characteristic polynomial of $B_i B_i^T$ is computed by establishing a similarity transformation to a matrix in Frobenius normal form [36, Section 16.6]. Fast matrix-matrix multiplication and inversion can be exploited so that the cost of this approach is $\mathcal{O}(nm^\omega \log m)$, where $\omega \leq 2.373$ is the best exponent of matrix-matrix multiplication complexity.

2. Algorithm 3 in [57] first computes the thin SVD

$$B = U\Sigma V^T, \qquad U \in \mathbb{R}^{m\times m}, \quad \Sigma \in \mathbb{R}^{m\times m}, \quad V \in \mathbb{R}^{n\times m}, \tag{4.6}$$

where U and V have orthonormal columns and Σ is a diagonal matrix. Then it computes the characteristic polynomial of BB^T from the squared singular values of B, and the auxiliary polynomials $g_j(x) = \prod_{\ell\neq j}\left(x - \sigma_\ell^2(B)\right)$ for $j = 1, \ldots, m$. For $h = m - k + t$ and $h = m - k + t - 1$, the coefficient $c_h(B_iB_i^T)$ can then be computed as the coefficient of x^h in

$$\det(xI - BB^T) + \frac{1}{\|b_i\|_2^2}\sum_{j=1}^{n}\sigma_j^2(B)v_{ij}^2 g_j(x). \tag{4.7}$$

The cost of this second approach is $\mathcal{O}(m^2n)$.

The problem of computing the Frobenius normal form of a matrix is "numerically not viable" [160]. Also, updating directly the characteristic polynomial as in (4.7) is prone to numerical cancellation, leading to inaccurate results. For instance, consider the 2×2 matrix

$$A = \begin{bmatrix} 6.583644 \cdot 10^{-7} & 8.113362 \cdot 10^{-3} \\ 8.113362 \cdot 10^{-3} & 100 \end{bmatrix},$$

and the column selection problem for $k = 1$. Algorithm 4 in [57] using (4.7) selects the first column, giving an error $\|A - A(:,1)A(:,1)^\dagger A\|_F \approx 1.2 \cdot 10^{-6} \gg \sqrt{2}\sigma_2(A) = 1.4 \cdot 10^{-10}$.

Therefore, from now on we will avoid updating coefficients of characteristic polynomials and work with singular values instead. More specifically, we will compute the singular values of B_i by updating the SVD of B and then apply the Summation Algorithm [161, Algorithm 1] to compute the coefficients of the characteristic polynomial of $B_iB_i^T$ from its eigenvalues (that is, the squared singular values of B_i) with $\mathcal{O}(m^2)$ operations in a numerically forward stable manner. To describe the updating procedure, consider the (thin) SVD $B = U\Sigma V^T$ as in (4.6). The (nonzero) singular values of B_i and

$$U^TB_iV = (I - U^T\pi_iU)U^TBV = \left(I - \frac{U^Tb_ib_i^TU}{\|b_i\|_2^2}\right)\Sigma = \left(I - qq^T\right)\Sigma,$$

with $q = U^Tb_i/\|b_i\|_2$, are identical. Using standard bulge chasing algorithms (see, e.g., [195,

Algorithm 3.4] and [6]) it is possible to find orthogonal matrices $Q, W \in \mathbb{R}^{m\times m}$ such that $Q^T q = e_1$, where e_1 denotes the first unit vector, and $Q^T \Sigma W$ is upper bidiagonal. In turn, the singular values can be computed from the bidiagonal matrix

$$Q^T(I - qq^T)\Sigma W = (I - e_1 e_1^T)(Q^T \Sigma W).$$

The matrices Q and W are composed of $\mathcal{O}(m^2)$ Givens rotations [86, Section 5.1] and the computation of $Q^T \Sigma W$ requires to apply each of these rotations to at most 3 vectors. In turn, the cost of computing this bidiagonal matrix is $\mathcal{O}(m^2)$, which is identical to the cost of computing its singular values [86, Section 8.6].

4.1.3 Overall algorithm

The described variation of the column subset selection algorithm by Deshpande and Rademacher is summarized in Algorithm 4.1. One execution of line 3 is $\mathcal{O}(nm^2)$, lines 6–9 are $\mathcal{O}(m^2)$, and lines 14–15 are $\mathcal{O}(knm)$. In summary, the overall complexity of Algorithm 4.1 is $\mathcal{O}(knm^2)$. This is identical to the complexity of [57, Algorithm 4] combined with [57, Algorithm 3], and it is better than [57, Algorithm 4] combined with [57, Algorithm 2].

Note that instead of lines 14–15 we could have updated $B \leftarrow B - \pi_{i_t} B$. However, we noticed that recomputing B in lines 14–15 tends to improve accuracy and it does not change the overall asymptotic complexity.

4.1.4 Early stopping of column search

For each column index, Algorithm 4.1 needs to traverse $\mathcal{O}(n)$ columns in order to find the one that minimizes the coefficient ratio or, equivalently, the conditional expectation. This column search can be shortened. To describe the idea, we revisit the argument from Section 4.1.1 that has led to Algorithm 4.1. Recall that (4.2) states $\mathbb{E}[\|A - \pi_{X_1,\ldots,X_k} A\|_F^2] \leq (k+1)\left(\sigma_{k+1}^2 + \ldots + \sigma_m^2\right)$. This implies that there exists i_1 such that

$$\mathbb{E}\left[\|A - \pi_{X_1,\ldots,X_k} A\|_F^2 \mid X_1 = i_1\right] \leq (k+1)\left(\sigma_{k+1}^2 + \ldots + \sigma_m^2\right).$$

In particular, an index i_1 that minimizes the left-hand side will satisfy the bound, which is the choice made in Algorithm 4.1. However, there may be other choices of i_1 that satisfy

Algorithm 4.1 Column Subset Selection

Input: $A \in \mathbb{R}^{m \times n}$, rank $1 \leq k < m$
Output: Column indices $S \in \{1, \ldots, n\}^k$

1: Initialize $S \leftarrow ()$ and $B \leftarrow A$
2: **for** $t = 1, \ldots, k$ **do**
3: Compute U and Σ from the thin SVD of $B = U\Sigma V^T$
4: minRatio$\leftarrow +\infty$
5: **for** $i = 1, \ldots, n$ **do**
6: $q \leftarrow U^T b_i / \|b_i\|_2$
7: $D \leftarrow Q^T \Sigma W$ bidiagonal matrix obtained by bulge chasing [195, Algorithm 3.4]
8: Compute singular values $\sigma_1, \ldots, \sigma_m$ of $(I - e_1 e_1^T)D$
9: Apply Summation Algorithm [161, Algorithm 1] to compute $c_{m-k+t-1}(B_i B_i^T)$ and $c_{m-k+t}(B_i B_i^T)$ from eigenvalues $\sigma_1^2, \ldots, \sigma_m^2$
10: Set ratio$\leftarrow c_{m-k+t-1}(B_i B_i^T)/c_{m-k+t}(B_i B_i^T)$
11: **if** ratio $<$ minRatio **then** Set minRatio$\leftarrow$ratio and $i_t \leftarrow i$ **end if**
12: **end for**
13: Append index $S \leftarrow (S, i_t)$
14: Compute orthonormal basis Q of $A(:, S)$
15: $B \leftarrow A - QQ^T A$
16: **end for**

the bound. *Any* such i_1 is a suitable choice. More generally, suppose that $i_1, \ldots, i_{t-1}$ have already been selected such that

$$\mathbb{E}\left[\|A - \pi_{X_1,\ldots,X_k} A\|_F^2 \mid X_1 = i_1, \ldots, X_{t-1} = i_{t-1}\right] \leq (k+1)(\sigma_{k+1}^2 + \ldots + \sigma_m^2)$$

holds. This implies the existence of i_t such that

$$\mathbb{E}\left[\|A - \pi_{X_1,\ldots,X_k} A\|_F^2 \mid X_1 = i_1, \ldots, X_t = i_t\right] \leq (k+1)(\sigma_{k+1}^2 + \ldots + \sigma_m^2). \tag{4.8}$$

Again, there is no need to choose an index i_t that minimizes the left-hand side; *any* i_t such that (4.8) holds is a suitable choice. By induction, choosing in every step an index such that (4.8) is verified implies that the error bound (2.10) holds.

The discussion above suggests to modify Algorithm 4.1 such that it computes

$$\text{bound} \leftarrow (k+1) \cdot \left(\sigma_{k+1}^2 + \ldots + \sigma_m^2\right)$$

in the beginning and substitute line 11 with

11: **if** $(k-t+1)\cdot \text{ratio} \leq$ bound **then** Set $i_t \leftarrow i$ and break **end if**

To be able to stop the search early, it is important to test the columns in a suitable order. We found it beneficial to test the columns of B in descending Euclidean norm. For each step t, computing the norms of all columns of B and sorting them has complexity $\mathcal{O}(mn + n\log n)$.

Although this choice is clearly heuristic, the following lemma provides some justification for it by showing that the column of largest norm is the right choice for $k = 1$ provided that all other columns are sufficiently small.

Lemma 4.1. *Let* $A = \begin{bmatrix} a_1 & A_2 \end{bmatrix}$. *If* $\|A_2\|_F \leq \|a_1\|_2$ *then choosing the first column solves the column selection problem for* $k = 1$*, that is,*

$$\|A - a_1 a_1^\dagger A\|_F^2 \leq 2(\sigma_2^2 + \ldots + \sigma_m^2).$$

Note that the condition of the lemma is satisfied if the column norms of A decay sufficiently fast, for instance if $\|a_i\|_2 \leq \frac{\|a_1\|_2}{i}$ for $i = 2, \ldots, n$.

Proof. Without loss of generality we may assume that $\|a_1\|_2 = 1$. By setting $B = A_2 - a_1 a_1^\dagger A_2 = A_2 - a_1 a_1^T A_2$ and $b = A_2^T a_1$, we have

$$A^T A = \begin{bmatrix} 1 & b^T \\ b & A_2^T A_2 \end{bmatrix} = \begin{bmatrix} 1 & b^T \\ b & B^T B + bb^T \end{bmatrix}$$

and obtain

$$\|A^T A\|_2 \leq \left\| \begin{bmatrix} 1 & \|b\|_2 \\ \|b\|_2 & \|B^T B + bb^T\|_2 \end{bmatrix} \right\|_2 \leq \left\| \begin{bmatrix} 1 & \|b\|_2 \\ \|b\|_2 & \|B\|_F^2 + \|b\|_2^2 \end{bmatrix} \right\|_2 . \tag{4.9}$$

Here, the first inequality is a norm-compression inequality [27, Section 9.10] and the second inequality follows from the fact that the involved matrices are positive.

We aim at proving

$$\|A - a_1 a_1^\dagger A\|_F^2 = \|B\|_F^2 \leq 2(\|A\|_F^2 - \|A\|_2^2),$$

which is equivalent to

$$\|A\|_2^2 \leq 1 + \|b\|_2^2 + \frac{\|B\|_F^2}{2} =: \gamma.$$

Thus, it remains to show that the larger eigenvalue of the symmetric positive definite 2×2 matrix on the right-hand side of (4.9) is bounded by γ. For this purpose, we note that its characteristic polynomial is given by

$$p(\lambda) = (\lambda - 1)\left(\lambda - \|b\|_2^2 - \|B\|_F^2\right) - \|b\|_2^2.$$

Setting $\gamma = 1 + \|b\|_2^2 + \|B\|_F^2/2$, we obtain

$$p(\gamma) = \|B\|_F^2/2 \cdot \left(1 - \|b\|_2^2 - \|B\|_F^2/2\right) \geq 0,$$

where we used that $\|b\|_2^2 + \|B\|_F^2 = \|A_2\|_F^2 \leq \|a_1\|_2^2 = 1$. Because p is a parabola with vertex $(1 + \|b\|_2^2 + \|B\|_F^2)/2 \leq \gamma$, it follows that the larger root of p is bounded by γ, which completes the proof. $\square$

It is important to not draw too many conclusions from Lemma 4.1. Consider, for example, the matrix

$$A = \begin{bmatrix} 1 & 0 & 10^{-b} \\ 0 & 1 & 10^{-b} \\ 0 & 0 & 10^{-2b} \end{bmatrix}$$

for some $b > 1$, say $b = 16$. For $k = 1$, the optimal choice is the third column, which is the one of smallest norm. This matrix also nicely illustrates that the obvious greedy approach (in order to get k columns of A, one first chooses the column i_1 that minimizes $\|A - \pi_{i_1} A\|_F$, then the column i_2 that minimizes $\|A - \pi_{i_1,i_2} A\|_F$, and so on) comes with no guarantees and may, in fact, utterly fail. For $k = 2$ the optimal choice consists of the first two columns. On the other hand, the greedy approach for $k = 2$ first selects the third column and then the first column, resulting in the arbitrarily bad error ratio (error greedy)/(error best) $\approx 10^b$.

This example also shows that, given a column subset of cardinality $k - 1$ selected by Algorithm 4.1 one cannot obtain a suitable selection of k columns by simply performing another step of Algorithm 4.1. In order to ensure that (2.10) holds, Algorithm 4.1 needs to be re-run from scratch with k instead of $k - 1$.

4.1.5 Numerical experiments

Both variants of Algorithm 4.1, without and with early stopping, have been implemented in Matlab version R2019a. As the bulge chasing algorithm in line 7 would perform poorly in Matlab, this part has been implemented in C++ and is called via a MEX interface. Our implementation is available at https://github.com/Alice94/CSS-Code together with the codes to reproduce the figures in this chapter. All numerical experiments in this chapter have been run on an eight-core Intel Core i7-8650U 1.90 GHz CPU, with 256 KB of level 2 Cache and 16 GB of RAM. Multi-threading has been turned off in order to not distort the findings.

We have applied the algorithm to the following three matrices:

1. the Hilbert matrix $A_{\text{hilb}} \in \mathbb{R}^{200\times 200}$ given by $A_{\text{hilb}}(i,j) = \frac{1}{i+j-1}$;
2. $A_{\text{exp}} \in \mathbb{R}^{100\times 200}$ given by $A_{\text{exp}}(i,j) = \exp(-0.3 \cdot |i-j|/200)$;
3. $A_{\text{poly}} \in \mathbb{R}^{100\times 200}$ given by $A_{\text{poly}}(i,j) = \left(\left(\frac{i}{200}\right)^{20} + \left(\frac{j}{200}\right)^{20}\right)^{1/20}$.

The obtained results are shown in Figures 4.1, 4.2, and 4.3 respectively. Each left plot contains, for different values of k, the approximation error $\|A - A(:,S)A(:,S)^{\dagger}A\|_F$ returned by Algorithm 4.1, without and with early stopping. We compare with the best rank-k approximation error $\sqrt{\sigma_{k+1}^2 + \ldots + \sigma_m^2}$ and the upper bound (2.10), that is, $\sqrt{(k+1)(\sigma_{k+1}^2 + \ldots + \sigma_m^2)}$. It can be seen that both variants of our algorithm stay below the upper bound, until it reaches the level of roundoff error. Interestingly, for the matrix A_{exp}, which features the slowest singular value decay, the observed approximation error is much closer to the best approximation error than to the upper bound. The right plots of the figures show, for different values of k, the ratio between the total execution times of Algorithm 4.1 without early stopping and with early stopping. For example, for the matrix A_{hilb}, using early stopping in Algorithm 4.1 reduces the time for constructing an approximation of rank $k = 15$ by a factor 22. For the variant with early stopping, we also plot the number of columns that were examined. In the most optimistic scenario, only k columns need to be examined, which means that in every step of the algorithm already the first satisfies the desired criterion. The plots reveal that our algorithm actually stays pretty close to this ideal situation, at least for the matrices considered. Note that for values of k larger than the numerical rank of the matrix, Algorithm 4.1 starts computing ratios (4.5)

from singular values of the order of machine precision. In turn, the computations are severely affected by roundoff error and it may, in fact, happen that the early stopping criterion is never satisfied. This leads to meaningless results and we therefore truncate the plots before this happens. A proper implementation of Algorithm 4.1 needs to detect such a situation and reduce k accordingly.

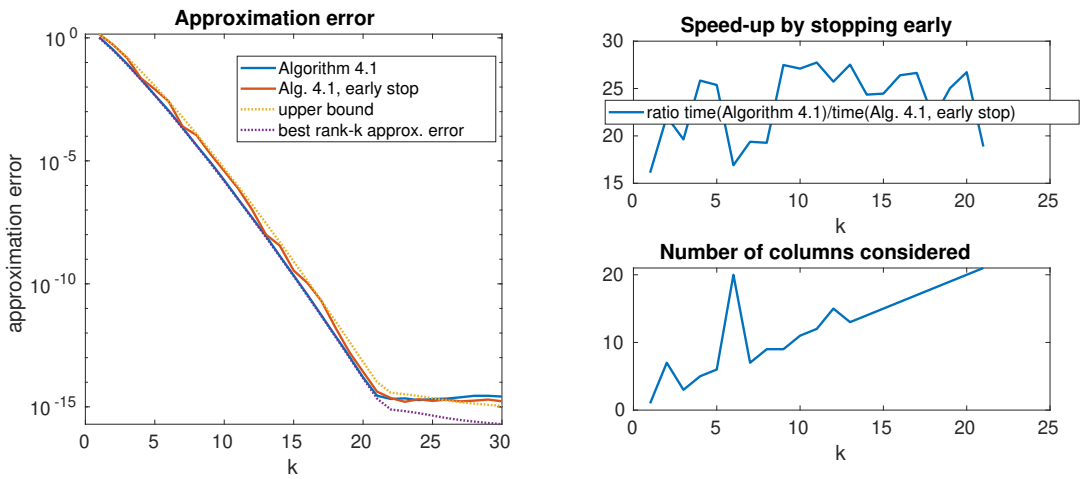


Figure 4.1 – Results for matrix A_{hilb}.

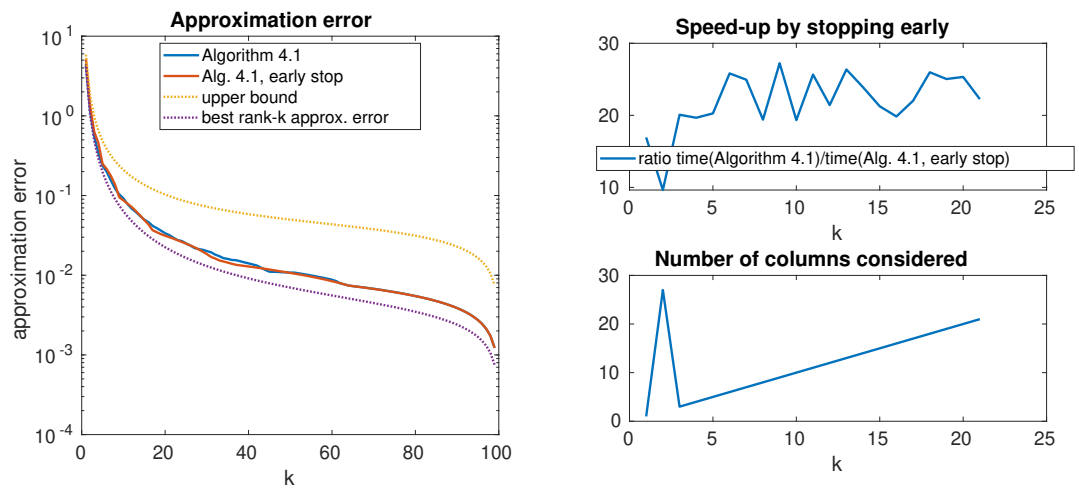


Figure 4.2 – Results for matrix A_{exp}.

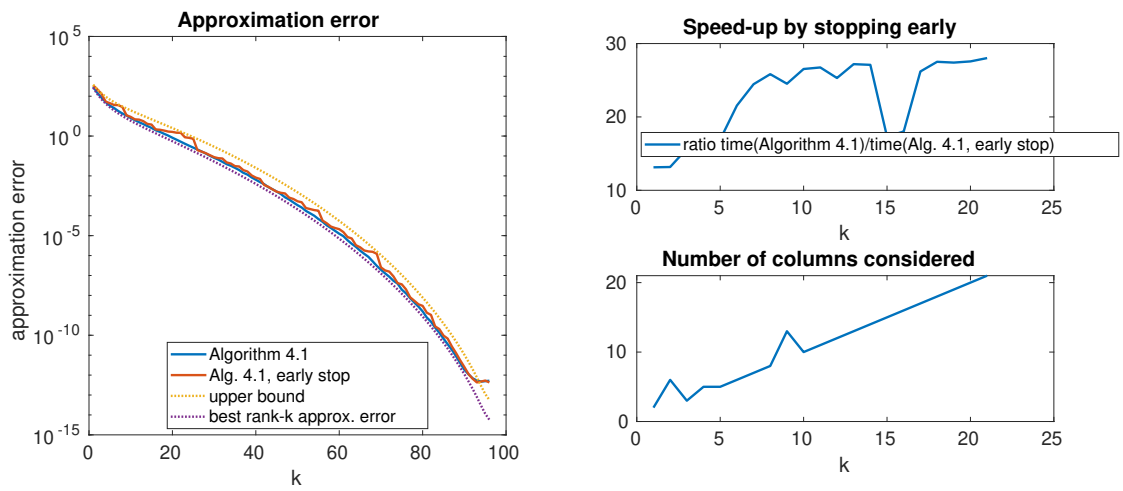


Figure 4.3 – Results for matrix A_{poly}.

4.2 Matrix approximation

In this section, we extend the developments from Section 4.1 on column subset selection to compute certain low-rank matrix approximations of a matrix $A \in \mathbb{R}^{m\times n}$ with $m \leq n$. We will pursue two ways. First, in Section 4.2.1, we discuss a general CUR approximation (see Section 2.3) obtained from applying column subset selection to the columns and rows of the matrix. Second, in Section 4.2.2, we present a novel approach to cross approximation, with guaranteed error bounds.

4.2.1 CUR approximation induced by column subset selection

As mentioned in Section 2.3, when a subset of columns $C \in \mathbb{R}^{m\times k}$ and a subset of rows $R \in \mathbb{R}^{k\times n}$ have been chosen, the matrix $U \in \mathbb{R}^{k\times k}$ that minimizes $\|A - CUR\|_F$ is given by the projection $U = C^\dagger A R^\dagger$, see [175, p. 320]. The following corollary provides an error bound for the case when C and R are determined by the techniques from Section 4.1, leading to Algorithm 4.2. The results in [63, Theorem 4], [166, Corollary 3.5], and [172, Theorem 4.1] are closely related.

Corollary 4.2. *Let $A \in \mathbb{R}^{m\times n}$, with $1 \leq k \leq m \leq n$. Then the CUR approximation returned by Algorithm 4.2 satisfies*

$$\|A - CUR\|_F \leq \sqrt{2k+2}\sqrt{\sigma_{k+1}^2(A) + \ldots + \sigma_m^2(A)}.$$

Algorithm 4.2 Matrix approximation by column subset selection

Input: $A \in \mathbb{R}^{m\times n}$, rank k
Output: Rank-k CUR approximation, with C, R containing columns and rows of A
1: Compute C by applying Algorithm 4.1 to select k columns of A
2: Compute R by applying Algorithm 4.1 to select k columns of A^T
3: Compute $U \leftarrow C^\dagger A R^\dagger$

Proof. Using the inequality (2.10) twice and the fact that $CC^\dagger$ is an orthogonal projection, we obtain

$$\begin{aligned}
\|A - CUR\|_F^2 &= \|A - CC^\dagger A R^\dagger R\|_F^2 \\
&= \|A - CC^\dagger A\|_F^2 + \|CC^\dagger (A - AR^\dagger R)\|_F^2 \\
&\leq \|(I - CC^\dagger)A\|_F^2 + \|A(I - R^\dagger R)\|_F^2 \\
&\leq 2(k+1)\left(\sigma_{k+1}^2(A) + \ldots + \sigma_m^2(A)\right).
\end{aligned}$$

□

Numerical experiments

We have tested a Matlab implementation of Algorithm 4.2 in the setting and for the matrices $A_{\text{hilb}}, A_{\text{exp}}, A_{\text{poly}}$ described in Section 4.1.5. Figure 4.4 displays the obtained approximation errors $\|A - CUR\|_F$ for different values of k. Again, we have tested both variants of Algorithm 4.1, without and with early stopping, within Algorithm 4.2. The speedups obtained from early stopping are very similar to the ones reported Section 4.1.5 and, therefore, we refrain from providing details.

We also consider, for $0 < \alpha < 1$, the $n \times n$ matrix

$$A = Q \cdot \operatorname{diag}(1, \alpha, \alpha^2, \ldots, \alpha^{n-1}) \cdot Q^T,$$

where $Q \in \mathbb{R}^{n\times n}$ is determined as the orthogonal factor from the QR decomposition of

$$\begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ -1 & -1 & 1 & & \\ \vdots & \vdots & & \ddots & \\ -1 & -1 & -1 & \cdots & 1 \end{bmatrix}.$$

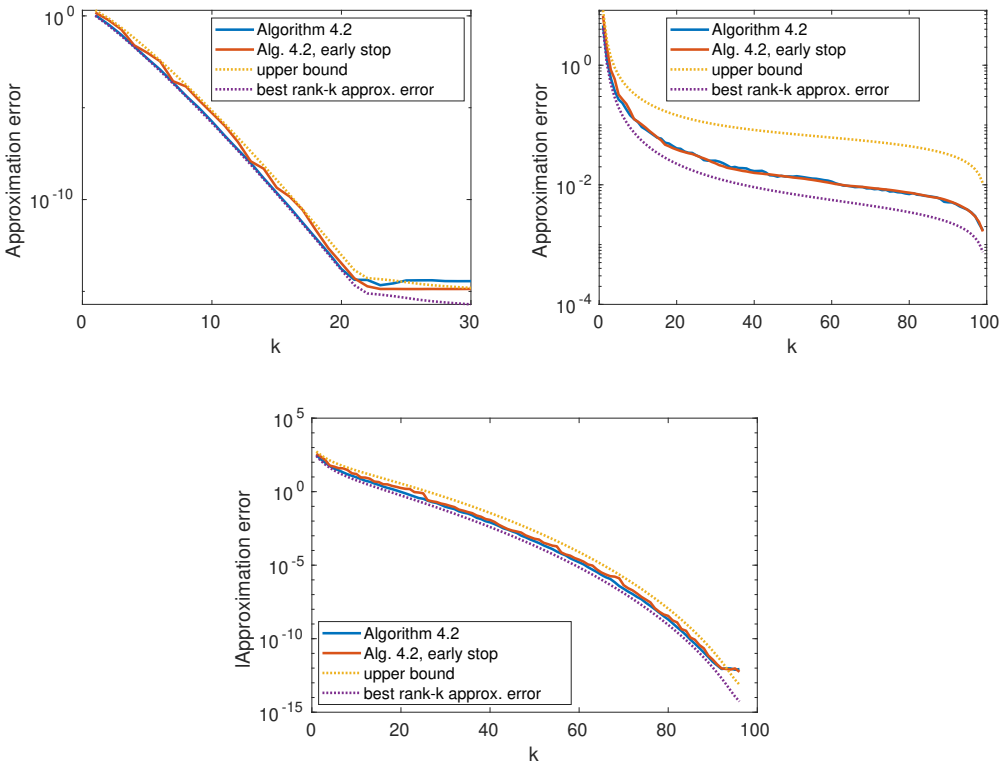


Figure 4.4 – Approximation errors for matrices A_{hilb} (top left), A_{exp} (top right), and A_{poly} (bottom).

This is known to be a challenging example for the CUR approximation induced by DEIM (discrete interpolation method); see [172, Section 4.2], which determines the row and column indices by greedily choosing a maximum volume submatrix of U_k and V_k containing the first k left and right singular vectors of A, respectively. For the example above, the DEIM induced CUR approximation always chooses $1, \ldots, k$ for the column and row indices. For $\alpha = 0.1$, $n = 6$, $k = 5$, the error resulting from this choice is given by

$$\|A - A(:, 1:5)A(:, 1:5)^{\dagger} AA(1:5, :)^{\dagger} A(1:5, :)\|_F \approx 2.6 \cdot 10^{-9},$$

which is a magnitude larger than the upper bound $\sqrt{2(k+1)}\sigma_6 \approx 3.5 \cdot 10^{-10}$ guaranteed by Algorithm 4.2. Note that the latter algorithm selects the last 5 rows and columns for this example, leading to an error of $\approx 1.3 \cdot 10^{-10}$.

4.2.2 Cross approximation

We now consider cross approximations (Definition 2.3). As discussed in Section 2.1.1 and Chapter 3, it is crucial to choose the row/column index tuples $(I, J) \in \Omega := \{1, \ldots, m\}^k \times \{1, \ldots, n\}^k$ wisely. In particular, as the following example shows, choosing the indices I, J as in Algorithm 4.2 may lead to poor approximation error.

Example 4.3. *Consider* $A = \left[\begin{smallmatrix} 2\varepsilon & 1 \\ 1 & \varepsilon \end{smallmatrix}\right]$ *for* $\varepsilon > 0$ *and* $k = 1$. *Clearly, the first column and row satisfy the bound* (2.10) *for* $k = 1$ *with respect to* A *and* A^T, *respectively. However, the error of the corresponding cross approximation,* $\|A - A(:,1)A(1,1)^{-1}A(1,:)\|_F = \frac{1}{2\varepsilon} - \varepsilon$, *becomes arbitrarily large as* $\varepsilon \to 0$.

Zamarashkin and Osinsky [196] have shown the existence of a cross approximation that satisfies a polynomial error bound in the Frobenius norm. To summarize their result, let

$$(X, Y) = (X_1, \ldots, X_k, Y_1, \ldots, Y_k)$$

be a $(2k)$-tuple of random variables with values in Ω such that

$$\mathbb{P}\left(X = I, Y = J\right) := \frac{\mathrm{Vol}^2\left(A(I, J)\right)}{\sum_{(I',J')\in\Omega} \mathrm{Vol}^2(A(I', J'))}. \tag{4.10}$$

Note that $\mathrm{Vol}\left(A(I, J)\right) = 0$ whenever $i_1, \ldots, i_k$ or $j_1, \ldots, j_k$ contain repeated indices. Then [196, Theorem 1] shows that

$$\mathbb{E}[\|A - A(:, Y)A(X, Y)^{-1}A(X, :)\|_F^2] \leq (k + 1)^2 \left(\sigma_{k+1}^2 + \ldots + \sigma_m^2\right). \tag{4.11}$$

In particular, this implies that there exists $(I, J) \in \Omega$ such that (2.7) holds.

In analogy to Section 4.1.1 and [57], we will now derandomize this result producing a polynomial-time deterministic algorithm that returns a cross approximation satisfying (2.7). The key for doing so is to find an expression for the conditional expectations that is easy to work with.

Conditional expectations

Lemma 4.4. *Let $1 \leq t \leq k$ and $(i_1, \ldots, i_t, j_1, \ldots, j_t)$ be such that*

$$\mathbb{P}\left(\substack{X_1=i_1,\ldots,X_t=i_t \\ Y_1=j_1,\ldots,Y_t=j_t}\right) > 0$$

for a random $(2k)$-tuple (X, Y) with the probability distribution defined by (4.10). Consider

$$B = A - A\left(:, (j_1, \ldots, j_t)\right) A\left((i_1, \ldots, i_t), (j_1, \ldots, j_t)\right)^{-1} A\left((i_1, \ldots, i_t), :\right),$$

the remainder of cross approximation after choosing row indices $i_1, \ldots, i_t$ and column indices $j_1, \ldots, j_t$. Then

$$\mathbb{E}\left[\|A - A(:, Y)A(X, Y)^{-1}A(X, :)\|_F^2 \,\middle|\, \substack{X_1=i_1,\ldots,X_t=i_t \\ Y_1=j_1,\ldots,Y_t=j_t}\right] = (k-t+1)^2 \cdot \frac{c_{m-k+t-1}(BB^T)}{c_{m-k+t}(BB^T)},$$

where the coefficients $c_{m-k+t}, c_{m-k+t-1}$ are defined as in (4.4) and the expectation is taken with respect to the distribution (4.10) defined on the $(2k)$-tuples in Ω.

Proof. To simplify notation, we let $I_1 = (i_1, \ldots, i_t)$, $I_2 = (i_{t+1}, \ldots, i_k)$, $I = (I_1, I_2) = (i_1, \ldots, i_k)$ and define J_1, J_2, J analogously. In the following, we always use the convention that row and column summation indices range from 1 to m and from 1 to n, respectively. We have that

$$\begin{aligned}
&\mathbb{E}\left[\|A - A(:, Y)A(X, Y)^{-1}A(X, :)\|_F^2 \,\middle|\, \substack{X_1=i_1,\ldots,X_t=i_t \\ Y_1=j_1,\ldots,Y_t=j_t}\right] \\
&= \sum_{\substack{i_{t+1},\ldots,i_k \\ j_{t+1},\ldots,j_k}} \|A - A(:, J)A(I, J)^{-1}A(I, :)\|_F^2 \cdot \mathbb{P}\left(X = I, Y = J \,\middle|\, \substack{X_1=i_1,\ldots,X_t=i_t \\ Y_1=j_1,\ldots,Y_t=j_t}\right) \\
&= \frac{1}{\gamma} \cdot \sum_{\substack{i_{t+1},\ldots,i_k,i_{k+1} \\ j_{t+1},\ldots,j_k,j_{k+1}}} \mathrm{Vol}^2\Big(A((I, i_{k+1}), (J, j_{k+1}))\Big),
\end{aligned} \tag{4.12}$$

with

$$\gamma = \sum_{\substack{i_{t+1},\ldots,i_k \\ j_{t+1},\ldots,j_k}} \mathrm{Vol}^2(A(I, J)).$$

For establishing the equality in (4.12) we used from [196, Lemma 1] that

$$\|A - A(:,J)A(I,J)^{-1}A(I,:)\|_F^2 = \frac{\sum_{i_{k+1},j_{k+1}} \mathrm{Vol}^2\Big(A((I,i_{k+1}),(J,j_{k+1}))\Big)}{\mathrm{Vol}^2(A(I,J))},$$

and, from (4.10), that

$$\mathbb{P}\left(X = I, Y = J \middle| \substack{X_1=i_1,\dots,X_t=i_t \\ Y_1=j_1,\dots,Y_t=j_t}\right) = \frac{\mathbb{P}(X=I,Y=J)}{\mathbb{P}\left(\substack{X_1=i_1,\dots,X_t=i_t \\ Y_1=j_1,\dots,Y_t=j_t}\right)} = \frac{1}{\gamma} \cdot \mathrm{Vol}^2(A(I,J)).$$

We now aim at simplifying the expression (4.12). For this purpose, we assume without loss of generality that $i_1 = 1, \dots, i_t = t$ and $j_1 = 1, \dots, j_t = t$. This allows us to partition

$$A(I,J) = \begin{bmatrix} A(I_1,J_1) & A(I_1,J_2) \\ A(I_2,J_1) & A(I_2,J_2) \end{bmatrix}, \quad B(I,J) = \begin{bmatrix} 0 & 0 \\ 0 & B(I_2,J_2) \end{bmatrix},$$

where $B(I_2,J_2) = A(I_2,J_2) - A(I_2,J_1)A(I_1,J_1)^{-1}A(I_1,J_2)$ by the definition of B. By the relation between determinants and Schur complements [112, Equation (0.8.5.1)], $\mathrm{Vol}\,(A(I,J)) = \mathrm{Vol}\,(A(I_1,J_1)) \cdot \mathrm{Vol}\,(B(I_2,J_2))$. Therefore,

$$\gamma = \sum_{\substack{i_{t+1},\dots,i_k \\ j_{t+1},\dots,j_k}} \mathrm{Vol}^2(A(I,J)) = \sum_{\substack{i_{t+1},\dots,i_k \\ j_{t+1},\dots,j_k}} \mathrm{Vol}^2(B(I_2,J_2)) \cdot \mathrm{Vol}^2(A(I_1,J_1)).$$

Analogously, one shows

$$\sum_{\substack{i_{t+1},\dots,i_{k+1} \\ j_{t+1},\dots,j_{k+1}}} \mathrm{Vol}^2\Big(A((I,i_{k+1}),(J,j_{k+1}))\Big)$$
$$= \sum_{\substack{i_{t+1},\dots,i_{k+1} \\ j_{t+1},\dots,j_{k+1}}} \mathrm{Vol}^2\Big(B((I_2,i_{k+1}),(J_2,j_{k+1}))\Big)\, \mathrm{Vol}^2(A(I_1,J_1)).$$

Inserting these expressions into (4.12) yields

$$\frac{\sum_{\substack{i_{t+1},\dots,i_{k+1} \\ j_{t+1},\dots,j_{k+1}}} \mathrm{Vol}^2\Big(B((I_2,i_{k+1}),(J_2,j_{k+1}))\Big)}{\sum_{\substack{i_{t+1},\dots,i_k \\ j_{t+1},\dots,j_k}} \mathrm{Vol}^2(B(I_2,J_2))}.$$

By [119, Theorem 7] this ratio is equal to

$$\frac{c_{m-k+t-1}(BB^T)\cdot((k-t+1)!)^2}{c_{m-k+t}(BB^T)\cdot((k-t)!)^2} = (k-t+1)^2\cdot\frac{c_{m-k+t-1}(BB^T)}{c_{m-k+t}(BB^T)}. \qquad \square$$

Derandomized cross approximation algorithm

With Lemma 4.4 at hand, we can proceed analogously to Section 4.1.1 and sequentially find k pairs of row/column indices such that (2.7) is satisfied. Suppose that $t-1$ index pairs $(i_1, j_1), \ldots, (i_{t-1}, j_{t-1})$ have been determined. Then the tth step of the algorithm proceeds by choosing (i_t, j_t) such that

$$\mathbb{E}\left[\|A - A(:,Y)A(X,Y)^{-1}A(X,:)\|_F^2 \,\middle|\, \begin{smallmatrix} X_1=i_1,\ldots,X_t=i_t \\ Y_1=j_1,\ldots,Y_t=j_t \end{smallmatrix}\right] \tag{4.13}$$

is minimized. We will show in Theorem 4.5 below that this choice of index pairs leads to a cross approximation satisfying the desired error bound (2.7). In view of Lemma 4.4, the minimization of (4.13) means that in each step of the algorithm we need to compute the ratios

$$\frac{c_{m-k+t-1}(C_{ij}C_{ij}^T)}{c_{m-k+t}(C_{ij}C_{ij}^T)}, \quad i = 1,\ldots,m, \quad j = 1,\ldots,n, \tag{4.14}$$

where

$$C_{ij} = A - A(:,(j_1,\ldots,j_{t-1},j))A((i_1,\ldots,i_{t-1},i),(j_1,\ldots,j_{t-1},j))^{-1}A((i_1,\ldots,i_{t-1},i),:).$$

Analogous to the developments in Section 4.1.2, we now show how the coefficients in (4.14) can be computed via updating the singular values of C_{ij}. Let us denote the remainder from the previous step by

$$B = A - A(:,(j_1,\ldots,j_{t-1}))A((i_1,\ldots,i_{t-1}),(j_1,\ldots,j_{t-1}))^{-1}A((i_1,\ldots,i_{t-1}),:).$$

Then it follows that

$$C_{ij} = B - \frac{1}{B(i,j)}B(:,j)B(i,:), \tag{4.15}$$

see, e.g., [13]. We compute a thin SVD $B = U\Sigma V^T$ such that $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{m\times n}$ have orthonormal columns and $\Sigma \in \mathbb{R}^{m\times m}$ is diagonal. Note that

$$B(:,j) = U\Sigma V(j,:)^T, \qquad B(i,:) = U(i,:)\Sigma V^T.$$

Inserted into (4.15), this shows that the nonzero singular values of C_{ij} match the singular values of

$$U^T C_{ij} V = \Sigma - \Sigma V(j,:)^T \cdot \frac{U(i,:)\Sigma}{B(i,j)} = \Sigma - xy^T,$$

where $x = \Sigma V(j,:)^T$ and $y = \frac{1}{B(i,j)} \Sigma U(i,:)^T$ are vectors of length m and can be computed with $\mathcal{O}(m^2)$ operations.

Similarly as in Section 4.1.2, we transform $\Sigma - xy^T$ into bidiagonal form, after which its singular values can be computed with $O(m^2)$ operations. This transformation proceeds in three steps:

1. We compute orthogonal matrices Q and W such that $Q^T \Sigma W$ is upper bidiagonal and $Q^T x = \pm \|x\|_2 \cdot e_1$ using, for example, [195, Algorithm 3.4]. In turn, the matrix

$$D_1 := Q^T (\Sigma - xy^T) W \tag{4.16}$$

   is bidiagonal with an additional nonzero first row; see the first plot in Figure 4.5 for an illustration.

2. By a bulge chasing algorithm, we transform D_1 to an upper banded matrix D_2 with two superdiagonals using $\mathcal{O}(m^2)$ Givens rotations. We refrain from giving a detailed description of the algorithm and refer to Figure 4.5 for an illustration.

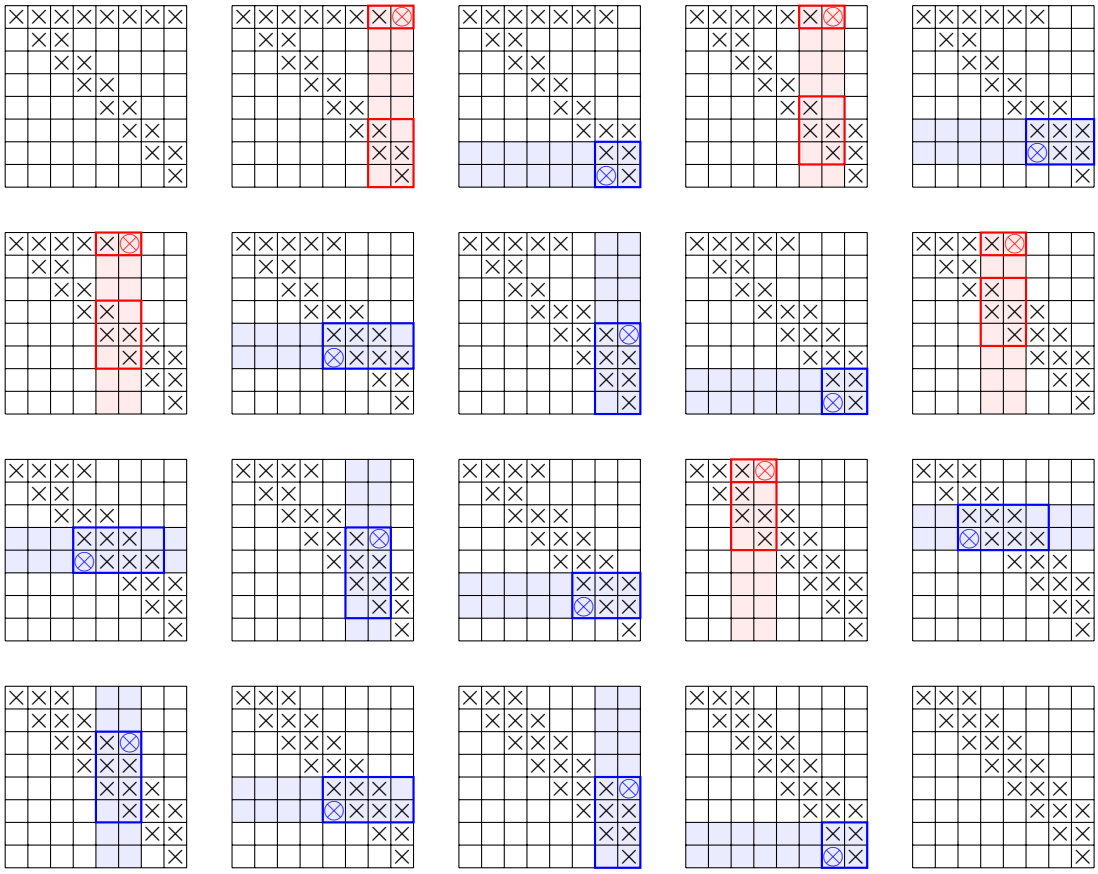
3. The banded matrix D_2 is reduced to a bidiagonal matrix D_3 using the LAPACK [3] routine `dgbbrd`.

The overall procedure described above can be implemented by means of $\mathcal{O}(m^2)$ Givens rotations, each of which is applied to a small matrix of size independent of m, n. Hence, it has complexity $\mathcal{O}(m^2)$.

Algorithm 4.3 summarizes our newly proposed method for cross approximation. The SVD needed at the beginning of each outer loop is of complexity $\mathcal{O}(m^2 n)$ and each of the mn inner loops costs $\mathcal{O}(m^2)$ operations; the total complexity of Algorithm 4.3 is therefore $\mathcal{O}(knm^3)$.

Theorem 4.5. *For a matrix A of rank at least k, Algorithm 4.3 returns index sets I and J such that* (2.7) *is satisfied.*

Figure 4.5 – Illustration of bulge chasing algorithm to transform a bidiagonal matrix with an additional nonzero first row to an upper banded matrix. In each plot, except for the first and last ones, a Givens rotation is applied to a pair of row or columns to zero out the entry denoted by $\otimes$.



Proof. Let $B_{\{X,Y\}} = A - A(:,Y)A(X,Y)^{-1}A(X,:)$. For $t = 1,\ldots,k$ we have that

$$\begin{aligned}
&\mathbb{E}\left[\|B_{\{X,Y\}}\|_F^2 \,\middle|\, \substack{X_1=i_1,\ldots,X_{t-1}=i_{t-1}\\ Y_1=j_1,\ldots,Y_{t-1}=j_{t-1}}\right]\\
&=\sum_{i,j}\mathbb{E}\left[\|B_{\{X,Y\}}\|_F^2 \,\middle|\, \substack{X_1=i_1,\ldots,X_{t-1}=i_{t-1},X_t=i\\ Y_1=j_1,\ldots,Y_{t-1}=j_{t-1},Y_t=j}\right]\mathbb{P}\Big(X_t=i,Y_t=j_t \,\Big|\, \substack{X_1=i_1,\ldots,X_{t-1}=i_{t-1}\\ Y_1=j_1,\ldots,Y_{t-1}=j_{t-1}}\Big).
\end{aligned}$$

Therefore, as (4.11) holds, the choice (4.13) inductively ensures that

$$\begin{aligned}
\mathbb{E}\left[\|B_{\{X,Y\}}\|_F^2 \,\middle|\, \substack{X_1=i_1,\ldots,X_t=i_t\\ Y_1=j_1,\ldots,Y_t=j_t}\right] &\le \mathbb{E}\left[\|B_{\{X,Y\}}\|_F^2 \,\middle|\, \substack{X_1=i_1,\ldots,X_{t-1}=i_{t-1}\\ Y_1=j_1,\ldots,Y_{t-1}=j_{t-1}}\right]\\
&\le (k+1)^2(\sigma_{k+1}^2+\ldots+\sigma_m^2).
\end{aligned}$$

Algorithm 4.3 Derandomized cross approximation

Input: $A \in \mathbb{R}^{m \times n}$ with $m \leq n$, integer $k \leq m$
Output: Index sets I, J of cardinality k defining the cross approximation (2.3)

```
Initialize I ← (), J ← (), and B ← A
for t = 1, ..., k do
  [U, Σ, V] ← thin SVD of B
  minRatio ← +∞
  for i = 1, ..., m do
    for j = 1, ..., n do
      x ← ΣV(j,:)^T, y ← (1/B(i,j)) ΣU(i,:)^T
      Compute matrix D_1 defined in (4.16) using [195, Algorithm 3.4]
      Transform D_1 into upper banded form D_2 using bulge chasing algorithm
      Transform D_2 into bidiagonal matrix D_3 using LAPACK's dgbbrd
      Compute singular values σ_1, ..., σ_m of D_3
      Apply Summation Algorithm [161, Algorithm 1] to obtain c_{m-k+t-1}(C_ij C_ij^T)
      and c_{m-k+t}(C_ij C_ij^T) from eigenvalues σ_1^2, ..., σ_m^2
      Set r ← c_{m-k+t-1}(C_ij C_ij^T) / c_{m-k+t}(C_ij C_ij^T)
      if r < minRatio then Row ← i, Col ← j, minRatio ← r end if
    end for
  end for
  I ← (I, Row), J ← (J, Col)
  B ← B − B(:,Col)·B(Row,:) / B(Row,Col)
end for
```

Therefore, the index sets I and J computed by Algorithm 4.3 satisfy the bound (2.7). □

In analogy to the discussion in Section 4.1.4, let us emphasize that it is not necessary to select the pair (i_t, j_t) that minimizes the ratio r. Any pair (i, j) for which the inequality

$$(k-t+1)^2 \frac{c_{m-k+t-1}(C_{ij}C_{ij}^T)}{c_{m-k+t}(C_{ij}C_{ij}^T)} \leq (k+1)^2(\sigma_{k+1}^2(A) + \ldots + \sigma_m^2(A)) \tag{4.17}$$

holds will lead to index sets I and J such that (2.7) is satisfied. Inspired by ACA with full pivoting (Algorithm 3.1), we traverse the entries of B from the largest to the smallest (in magnitude) and stop the search once we have found an index pair (i_t, j_t) satisfying (4.17).

A (theoretically) faster algorithm

It is possible to improve the worst-case time complexity of the derandomized cross approximation algorithm from $\mathcal{O}(knm^3)$ to $\mathcal{O}(knm^2)$. This does not improve the practical performance of the algorithm because in practice (see the numerical experiments' section) the heuristic criterion for choosing the new entries as the largest entries of B usually works well; however, it is satisfying from a theoretical point of view, as it gives an algorithm of the same time complexity as column subset selection[1].

We start with the following lemma.

Lemma 4.6. *Let $A \in \mathbb{R}^{m\times k}$ with $k \le m$. If to each column of A a linear combination of the other columns is added, the volume does not change.*

Proof. Adding to a column a linear combination of other columns means computing AB for a matrix $B \in \mathbb{R}^{k\times k}$ which has ones on the diagonal and has only one column with nonzero entries elsewhere, therefore $\det B = 1$. Therefore, $(\operatorname{Vol} A)^2 = \det(A^T A) = \det((AB)^T(AB)) = (\operatorname{Vol}(AB))^2$. □

Lemma 4.7. *Let $0 \le t < k$ and $(i_1, \dots, i_t, j_1, \dots, j_t, j_{t+1})$ be such that*

$$\mathbb{P}\left(\substack{X_1=i_1,\dots,X_t=i_t \\ Y_1=j_1,\dots,Y_t=j_t}, Y_{t+1} = j_{t+1}\right) > 0$$

for a random $(2k)$-tuple (X, Y) with the probability distribution defined by volume sampling. Consider

$$B = A - A(:, (j_1, \dots, j_t))A((i_1, \dots, i_t), (j_1, \dots, j_t))^{-1}A((i_1, \dots, i_t), :),$$

the remainder of cross approximation after choosing row indices $i_1, \dots, i_t$ and column indices $j_1, \dots, j_t$, and the matrix $C = B - \pi_{j_{t+1}} B$.

Then

$$\mathbb{E}\left[\|A - A(:, Y)A(X, Y)^{-1}A(X, :)\|_F^2 \mid \substack{X_1=i_1,\dots,X_t=i_t \\ Y_1=j_1,\dots,Y_t=j_t}, Y_{t+1} = j_{t+1}\right]$$

[1]The discussion in this subsection follows from a discussion with Alexander Osinsky and Nikolai Zamarashkin at the 5th International Conference on Matrix Methods in Mathematics and Applications, in August 2019 in Moscow, and is not contained in the paper [45].

$$= (k-t)(k-t+1)\frac{c_{m-k+t}(CC^T)}{c_{m-k+t+1}(CC^T)},$$

with the coefficients $c_{m-k+t}, c_{m-k+t-1}$ *defined as the coefficients of the characteristic polynomial.*

Proof. Using the same notation as in Lemma 4.4, we have that

$$\begin{aligned}
&\mathbb{E}\left[\|A - A(:,Y)A(X,Y)^{-1}A(X,:)\|_F^2 \mid {}^{X_1=i_1,\dots,X_t=i_t}_{Y_1=j_1,\dots,Y_t=j_t}, Y_{t+1}=j_{t+1}\right] \\
&= \sum_{\substack{i_{t+1},\dots,i_k \\ j_{t+2}\dots,j_k}} \|A - A(:,J)A(I,J)^{-1}A(I,:)\|_F^2 \cdot \frac{\mathbb{P}(X=I, Y=J)}{\mathbb{P}\left({}^{X_1=i_1,\dots,X_t=i_t}_{Y_1=j_1,\dots,Y_t=j_t}, Y_{t+1}=j_{t+1}\right)} \\
&= \sum_{\substack{i_{t+1},\dots,i_{k+1} \\ j_{t+2}\dots,j_{k+1}}} \mathrm{Vol}\,(A((I,i_{k+1}),(J,j_{k+1})))^2 \Big/ \sum_{\substack{i_{t+1},\dots,i_k \\ j_{t+2}\dots,j_k}} \mathrm{Vol}\,(A(I,J))^2 \\
&= \sum_{\substack{i_{t+1},\dots,i_{k+1} \\ j_{t+2}\dots,j_{k+1}}} \mathrm{Vol}\,(B((I_2,i_{k+1}),(J_2,j_{k+1})))^2 \Big/ \sum_{\substack{i_{t+1},\dots,i_k \\ j_{t+2}\dots,j_k}} \mathrm{Vol}\,(B(I_2,J_2))^2 \\
&= \frac{(k-t+1)!}{(k-t)!} \sum_{j_{t+2},\dots,j_{k+1}} \mathrm{Vol}\,(B(:,(J_2,j_{k+1})))^2 \Big/ \sum_{j_{t+2},\dots,j_k} \mathrm{Vol}\,(B(:,J_2))^2 \\
&= (k-t+1) \sum_{j_{t+2},\dots,j_{k+1}} \mathrm{Vol}\,(C(:,(j_{t+2},\dots,j_{k+1})))^2 \Big/ \sum_{j_{t+2},\dots,j_k} \mathrm{Vol}\,(C(:,(j_{t+2},\dots,j_k)))^2 \\
&= (k-t+1)\frac{(k-t)!c_{m-k+t}(CC^T)}{(k-t-1)!c_{m-k+t+1}(CC^T)} = (k-t+1)(k-t)\frac{c_{m-k+t}(CC^T)}{c_{m-k+t+1}(CC^T)}.
\end{aligned}$$

All equalities but the fifth follow similarly to the proof of Lemma 4.4. For the fifth equality, note that C is obtained from B by adding a multiple of the j_{k+1}th column of B. Therefore,

$$\begin{aligned}
\mathrm{Vol}\,B(:,J_2)^2 &= \mathrm{Vol}\left(\begin{bmatrix} C(:,(j_{t+2},\dots,j_k)) & B(:,j_{t+1}) \end{bmatrix}\right)^2 \\
&= \det\left(\begin{bmatrix} C(:,(j_{t+2},\dots,j_k)) & B(:,j_{t+1}) \end{bmatrix}^T \begin{bmatrix} C(:,(j_{t+2},\dots,j_k)) & B(:,j_{t+1}) \end{bmatrix}\right) \\
&= \det\begin{bmatrix} C(:,(j_{t+2},\dots,j_k))^T C(:,(j_{t+2},\dots,j_k)) & 0 \\ 0 & \|B(:,j_{t+1})\|_2^2 \end{bmatrix} \\
&= \mathrm{Vol}\,(C(:,(j_{t+2},\dots,j_k)))^2 \cdot \|B(:,j_{t+1})\|_2^2.
\end{aligned}$$

The third equality follows from the fact that $C^T B(:, j_{t+1}) = 0$. Analogously,

$$\operatorname{Vol} B(:, (J_2, j_{k+1}))^2 = \operatorname{Vol}(C(:, (j_{t+2}, \ldots, j_{k+1})))^2 \cdot \|B(:, j_{t+1})\|_2^2,$$

so the factor $\|B(:, j_{t+1})\|_2^2$ is present in both the numerator and the denominator and it simplifies. □

Combined with Lemma 4.4, Lemma 4.7 shows that, instead of choosing an index pair at each step as in Algorithm 4.3, we can subsequently select j_1, i_1, j_1, i_2, …, j_k, i_k. Selecting each index costs $\mathcal{O}(nm^2)$, leading to an $\mathcal{O}(knm^2)$ cross approximation algorithm.

Numerical experiments

We have implemented both variants of Algorithm 4.3, without and with early stopping, in Matlab. Again, the two inner loops have been implemented in a C++ function that is called via a MEX interface. The computational environment is the one described in Section 4.1.5 but the test matrices are smaller because Algorithm 4.3 without early stopping is significantly slower. We choose A_{hilb} to be 100×100, A_{exp} to be 50×100, and the matrix $A_{\text{poly}} \in \mathbb{R}^{50\times 100}$ is given by

$$A_{\text{poly}}(i, j) = \left(\left(\frac{i}{100} \right)^{10} + \left(\frac{j}{100} \right)^{10} \right)^{1/10}.$$

The approximation error $\|E_{IJ}\|_F = \|A - A(:, J)A(I, J)^{-1}A(I, :)\|_F$ for the index sets returned by both variants of Algorithm 4.3 is displayed in the left plots of Figures 4.6, 4.7, 4.8. The right plots display the ratios between the execution time of Algorithm 4.3 without and with early stopping, as well as the total number of index pairs that needed to be tested in Algorithm 4.3 with early stopping. It can be observed that early stopping dramatically accelerates the computation and is thus the preferred variant.

It can be seen that the approximation errors often stay close to the best rank-k approximation error $\sqrt{\sigma_{k+1}^2 + \ldots + \sigma_m^2}$ and do not exceed the upper bound (2.7), modulo roundoff error. However, for larger values of k, Algorithm 4.3 without early stopping appears to encounter stability issues; the approximation error is distorted well above

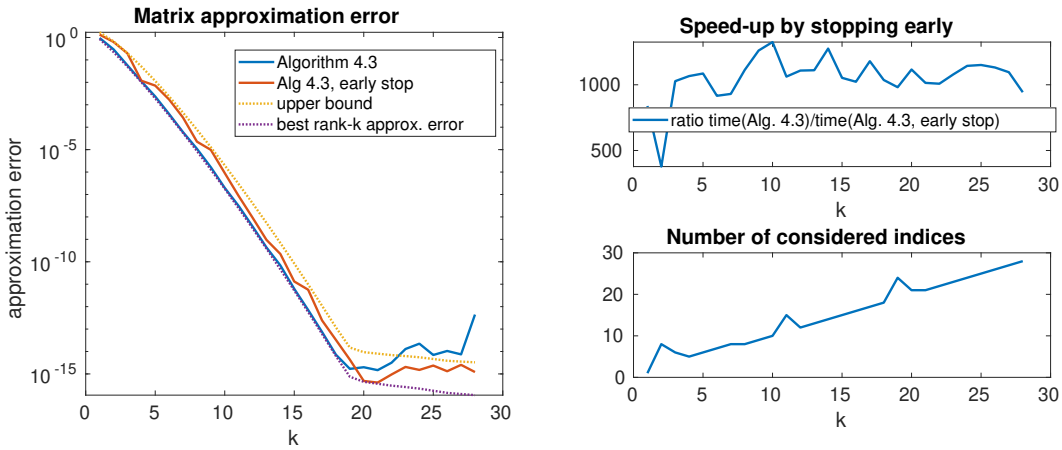


Figure 4.6 – Results for matrix A_{hilb}.

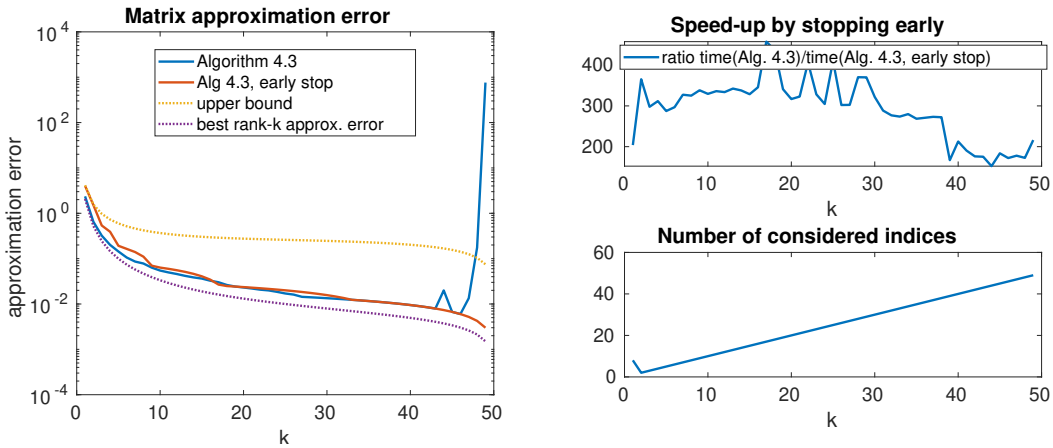


Figure 4.7 – Results for matrix A_{exp}.

the level of roundoff error. To better understand the numerical instability of our cross approximation algorithm, we analyzed what happens for the matrix A_{poly} for rank $k = 43$, for which Algorithm 4.3 without early stopping gives a cross approximation error out of the bounds predicted by (2.7). We computed the Frobenius norm of the intermediate residuals

$$A - A(:, (j_1 \dots j_t)) \cdot A((i_1 \dots i_t), (j_1 \dots j_t))^{-1} \cdot A((i_1 \dots i_t), :) \tag{4.18}$$

for $t = 1, \dots, 43$, for the index sets given by Algorithm 4.3 with and without early stopping. We used Matlab's `vpa` with 200 digits of accuracy to compute the Frobenius norm of the residual. The results are shown in Figure 4.9. The intermediate residuals

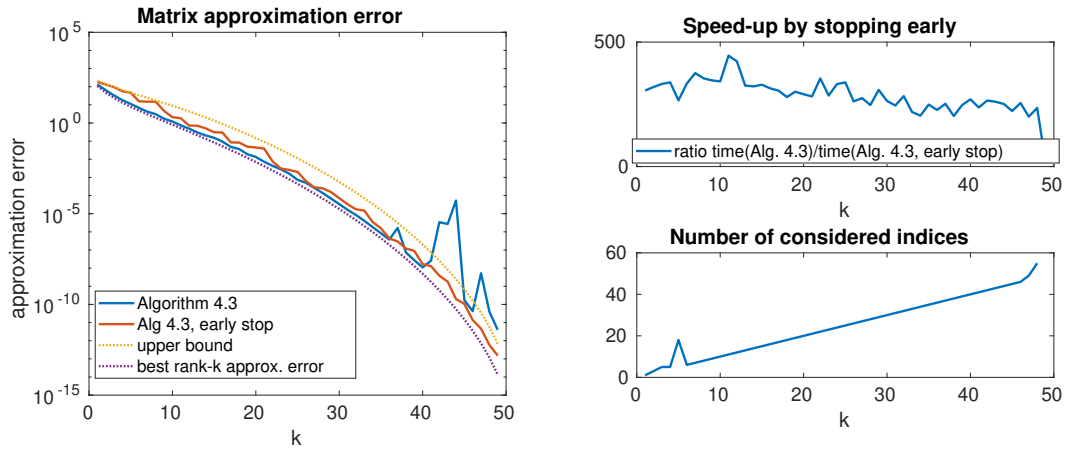


Figure 4.8 – Results for matrix A_{poly}.

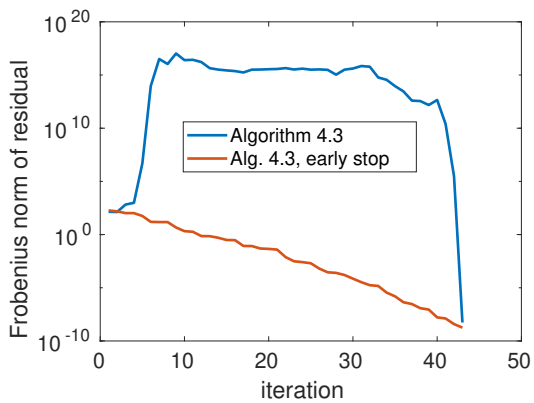


Figure 4.9 – Frobenius norm of the residual (4.18) *for $t = 1, \ldots, k$ for the matrix A_{poly} with target rank $k = 43$, when using Algorithm 4.3 with and without early stopping.*

grow significantly in Algorithm 4.3 without early stopping, which may completely spoil the accuracy of the singular values computed in lines 7–11. As these are needed to determine the indices to select at each step, this selection is not guaranteed to satisfy the bound (4.17). For A_{poly} and $k = 43$, this happens for the first time at iteration $t = 34$.

Such an intermediate growth of the residual can already happen for small matrices. For example, consider

$$A = \begin{bmatrix} -10^{-4} & 3 & -4 \\ 4 & 1 & 2 \\ 8 & -1 & 1 \end{bmatrix}.$$

When aiming at a rank-2 approximation, the choice that minimizes the expectation at the first step is the pivot in position $(1,1)$, which results in a residual more than 10^4 larger than the norm of the original matrix.

The intermediate growth of the residuals may explain why Algorithm 4.3 with early stopping shows more stability in the examples we considered: If possible, it chooses one of the largest entries of the residual, which, in turn, should prevent the residuals from becoming too large. However, there are no results that ensure that we can take a "large entry" at each step of the algorithm; further investigation would be needed to understand the stability of Algorithm 4.3 with early stopping.

We also consider the $n \times n$ matrix $A = LDL^T$, where

$$L = \begin{bmatrix} 1 & & & & \\ -c & 1 & & & \\ -c & -c & 1 & & \\ \vdots & \vdots & & \ddots & \\ -c & -c & -c & \cdots & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 1 & & & & \\ & s^2 & & & \\ & & s^4 & & \\ & & & \ddots & \\ & & & & s^{2(n-1)} \end{bmatrix}$$

with $s = \sin(\theta), c = \cos(\theta)$ for some $0 < \theta < \pi$. This is known to be a challenging example for greedy cross approximation [106]: When $k = n-1$ the greedy algorithm selects the leading $k \times k$ submatrix and returns an approximation error that is exponentially larger than the best approximation error. In contrast, Algorithm 4.3, with and without early stopping, makes the correct choice by selecting the last $n-1$ rows and columns. For instance, for $n = 6$ and $\theta = 0.1$, we obtain the error

$$\|A - A(:, 2:6)A(2:6, 2:6)^{-1}A(2:6, :)\|_F \approx 3.9 \cdot 10^{-13} < 1.8 \cdot 10^{-12} \approx \sqrt{6}\sigma_n.$$

Selecting the first 5 rows and columns results in an error of $9.8 \cdot 10^{-11}$.

Finally, we would like to point out an interesting observation concerning the preservation of structure. In Section 3.1, we have shown that for a symmetric positive definite matrix A there is always a symmetric choice of indices, $J = I$, leading to a symmetric cross approximation such that the favorable error bound (2.4) is attained. For cross approximation in the Frobenius norm, the situation appears to be more complicated; it is

generally not true that a symmetric choice of indices achieves the error bound (2.7) even when A is symmetric positive definite. For instance, for $n = 3$ and $k = 1$ consider

$$A = \begin{bmatrix} 1.87 & -1.82 & -2.11 \\ -1.82 & 1.87 & 2.11 \\ -2.11 & 2.11 & 2.54 \end{bmatrix}.$$

The best symmetric choice is $I = J = (3)$ but this leads to an error $\approx 0.1911 > 2\sqrt{\sigma_2^2 + \sigma_3^2} \approx 0.1821$.

4.3 Tensor approximation

As mentioned in Section 2.4, column subset selection can be used to approximate tensors as well. In the following, we demonstrate the use of the algorithm from Section 4.1 to obtain approximations of low multilinear rank constructed from the fibers of a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.

Algorithm 4.4 produces an approximate Tucker decomposition for a given tensor such that each coefficient matrix B_μ is composed of μ-mode fibers. The following result shows that the obtained approximation error remains close to the best approximation error.

Algorithm 4.4 Approximation of tensors by column selection

Input: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, integers k_1, k_2, k_3

Output: Approximate Tucker decomposition of multilinear rank (k_1, k_2, k_3) in terms of coefficient matrices B_1, B_2, B_3 and core tensor $\mathcal{C}$

1: **for** $\mu = 1, 2, 3$ **do**

2: Compute $B_\mu \leftarrow A^{(\mu)}(:, S_\mu)$ by applying Algorithm 4.1 to select k_μ columns from $A^{(\mu)}$

3: **end for**

4: Compute $\mathcal{C} \leftarrow \mathcal{A} \times_1 B_1^\dagger \times_2 B_2^\dagger \times_3 B_3^\dagger$

Corollary 4.8. *Consider $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and integers k_1, k_2, k_3 such that $1 \leq k_\mu \leq n_\mu$ for $\mu = 1, 2, 3$. Then the output of Algorithm 4.4 satisfies*

$$\|\mathcal{A} - \mathcal{C} \times_1 B_1 \times_2 B_2 \times_3 B_3\|_F \leq \sqrt{k_1 + k_2 + k_3 + 3} \cdot \|\mathcal{A} - \mathcal{A}_{\text{best}}\|_F,$$

where $\mathcal{A}_{\text{best}}$ is the best Tucker approximation of $\mathcal{A}$ of multilinear rank at most (k_1, k_2, k_3).

Proof. The proof is similar to existing proofs on the quasi-optimality of the Higher-Order SVD [54] and related results in [62, 90, 166].

Using (2.10) and setting $\pi_\mu = B_\mu B_\mu^+$, the result of Algorithm 4.1 applied to $A^{(\mu)}$ satisfies

$$\begin{aligned}\|A^{(\mu)} - \pi_\mu(A^{(\mu)})\|_F^2 &\leq (k_\mu + 1)\Big(\sigma_{k_\mu+1}^2(A^{(\mu)}) + \ldots + \sigma_{n_\mu}^2(A^{(\mu)})\Big) \\ &\leq (k_\mu + 1)\|A^{(\mu)} - A_{\text{best}}^{(\mu)}\|_F^2 = (k_\mu + 1)\|\mathcal{A} - \mathcal{A}_{\text{best}}\|_F^2,\end{aligned}$$

where the second inequality follows from the fact that $A_{\text{best}}^{(\mu)}$, the μ-mode matricization of $\mathcal{A}_{\text{best}}$, has rank at most k_μ. Using the orthogonality of the projections π_μ, we obtain

$$\begin{aligned}&\|\mathcal{A} - \mathcal{C} \times_1 B_1 \times_2 B_2 \times_3 B_3\|_F^2 = \|\mathcal{A} - \mathcal{A} \times_1 \pi_1 \times_2 \pi_2 \times_3 \pi_3\|_F^2 \\ &\quad = \|\mathcal{A} - \mathcal{A} \times_1 \pi_1\|_F^2 + \|(\mathcal{A} - \mathcal{A} \times_2 \pi_2) \times_1 \pi_1\|_F^2 + \|(\mathcal{A} - \mathcal{A} \times_3 \pi_3) \times_1 \pi_1 \times_2 \pi_2\|_F^2 \\ &\quad \leq \sum_{\mu=1}^{3} \|\mathcal{A} - \mathcal{A} \times_\mu \pi_\mu\|_F^2 = \sum_{\mu=1}^{3} \|A^{(\mu)} - \pi_\mu(A^{(\mu)})\|_F^2 \leq \sum_{\mu=1}^{3} (k_\mu + 1)\|\mathcal{A} - \mathcal{A}_{\text{best}}\|_F^2 \\ &\quad = (k_1 + k_2 + k_3 + 3)\|\mathcal{A} - \mathcal{A}_{\text{best}}\|_F^2,\end{aligned}$$

where the second equality follows from [186, Theorem 5.1]. □

Remark 4.9. *Algorithm 4.4 easily generalizes to tensors of arbitrary order. Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ and integers $k_1, \ldots, k_d$, this generalization constructs subsets of fibers $B_1, \ldots, B_d$ and a core tensor $\mathcal{C}$ such that*

$$\|\mathcal{A} - \mathcal{C} \times_1 B_1 \times_2 \ldots \times_d B_d\|_F \leq \sqrt{k_1 + \ldots + k_d + d} \cdot \|\mathcal{A} - \mathcal{A}_{\text{best}}\|_F.$$

This compares favorably with other existence results in the literature, which feature much larger constants that grow exponentially with the order; see [90], [149, Theorem 3.1], and [166, Theorem 3.1].

4.3.1 Numerical experiments

We have implemented Algorithm 4.4 in Matlab and tested it on two $50 \times 50 \times 50$ tensors, given by $\mathcal{A}_{\text{hilb}}(i, j, h) = \frac{1}{i+j+h-1}$ and $\mathcal{A}_{\text{poly}}(i, j, h) = \left(i^{10} + j^{10} + h^{10}\right)^{1/10}/50$. We choose $k_1 = k_2 = k_3 = k$ and report in Figure 4.10 the obtained approximation errors $\|\mathcal{A} - \mathcal{C} \times_1 B_1 \times_2 B_2 \times_3 B_3\|_F$ for different values of k, where $B_1, B_2, B_3, \mathcal{C}$ are returned

by Algorithm 4.4, with and without early stopping in the column selection part. We compare with the quantity

$$\left(\sum_{\mu=1}^{3} \sigma_{k_\mu+1}^2(A^{(\mu)}) + \ldots + \sigma_{n_\mu}^2(A^{(\mu)}) \right)^{1/2},$$

which provides a (tight) upper bound on the best approximation error. It can be seen that the errors obtained from Algorithm 4.4 remain close to this quasi-best approximation error.

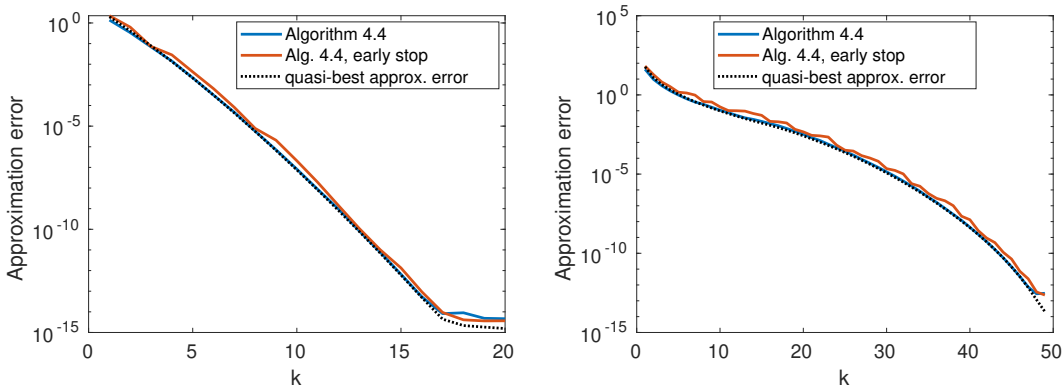


Figure 4.10 – Results for tensors $\mathcal{A}_{\text{hilb}}$ (left) and $\mathcal{A}_{\text{poly}}$ (right).

Part II

Low-rank updates and divide-and-conquer for matrix functions

5 Introduction to matrix functions

This second part of the thesis is concerned with matrix functions. The formal definition is the following.

Definition 5.1. *For $A \in \mathbb{R}^{n\times n}$ and a complex-valued function f which is analytic on and inside a contour Γ which encloses the eigenvalues of A, the matrix function $f(A)$ is defined as*

$$f(A) := \frac{1}{2\pi i}\int_\Gamma f(z)(zI_n - A)^{-1}\mathrm{d}z,$$

where $I_n \in \mathbb{R}^{n\times n}$ denotes the identity matrix.

In fact, it is not necessary for f to be analytic inside Γ in order to define the matrix function $f(A)$. For example, when A is diagonalizable, that is, $A = V \cdot \mathrm{diag}(\lambda_1, \ldots, \lambda_n) \cdot V^{-1}$, and the function f is defined on the eigenvalues $\lambda_1, \ldots, \lambda_n$ of A, the definition above is equivalent to

$$f(A) := V \cdot \mathrm{diag}(f(\lambda_1), \ldots, f(\lambda_n)) \cdot V^{-1};$$

this expression does not depend on the choice of the eigenvector matrix V. When A is not diagonalizable, $f(A)$ can be defined using the Jordan canonical form. In this case, letting $\lambda_1, \ldots, \lambda_s$ be the distinct eigenvalues of A and letting μ_i be the size of the largest Jordan block corresponding to the eigenvalue λ_i, for $i = 1, \ldots, s$, it is sufficient that f and its derivatives up to the $(\mu_i - 1)$th order are defined in λ_i, for all $i = 1, \ldots, s$. We refer the reader to [110, Section 1.2] for the precise definition of $f(A)$ using the Jordan canonical form and some additional equivalent definitions.

Functions of matrices arise, for instance, in the solution of ordinary or partial differ-

ential equations; see, e.g., [65, 128]. A well-known example is that the solution of the system of linear ordinary differential equations

$$\begin{cases} \dot{x}(t) = Ax(t) & x(t) \in \mathbb{R}^n, \qquad A \in \mathbb{R}^{n\times n} \\ x(0) = x_0 & x_0 \in \mathbb{R}^n \end{cases}$$

is given in closed form by the matrix exponential

$$x(t) = \exp(At)x_0 = \sum_{k=0}^{\infty} \frac{t^k A^k}{k!} x_0.$$

Other applications of matrix functions include electronic structure calculations [19, 84] and social network analysis [70]; see Section 7.2 for some more examples. In some cases, only some quantities related to matrix functions are of interest. The diagonal of a matrix function is needed, for instance, in Density Functional Theory [19], electronic structure calculations [130], and uncertainty quantification [177]. The trace of matrix functions is used to compute determinants [80] (see also Part III), spectral densities [131], Schatten p-norms [66], the Estrada index of a graph [70] (which will be defined in Section 7.2.6), and it also arises in lattice quantum chromodynamics [193].

A general-purpose algorithm for computing matrix functions is the Schur-Parlett algorithm [51]. For specific choices of f, such as the exponential, the sign function, the square root, or the logarithm, ad-hoc methods have been developed. We refer the reader to the book [110] for a detailed discussion of methods for general (dense) matrices.

In this thesis, we focus on the interplay between low-rank structures and matrix functions, analyzing low-rank updates and functions of rank-structured matrices, for which fast algorithms can be developed.

5.1 Low-rank updates

The first problem that we consider is the computation of a low-rank update of a matrix function. More precisely, when a matrix function $f(A)$ has been computed and the matrix A undergoes an additive low-rank modification R for some $R \in \mathbb{R}^{n\times n}$ with $\operatorname{rank}(R) \ll n$, we are interested in computing $f(A + R)$ without starting from scratch. This means that

we look for an efficient way to compute/approximate the update

$$f(A+R) - f(A).$$

For example, this is useful when computing matrix functions of adjacency matrices of graphs in which edges are added or removed: these are rank-2 modifications of the adjacency matrix. Moreover, having an algorithm for low-rank updates allows us to devise the D&C algorithms for computing matrix functions that we develop in Chapter 7.

A classical result, the Sherman-Morrison formula, states that if $f(z) = z^{-1}$ and $\operatorname{rank}(R) = 1$ then the update has rank 1.

Theorem 5.2 (Sherman-Morrison formula)**.** *For a matrix $A \in \mathbb{R}^{n\times n}$ and vectors $b, c \in \mathbb{R}^n$, it holds*

$$(A + bc^T)^{-1} - A^{-1} = -\frac{A^{-1}bc^TA^{-1}}{1 + c^TA^{-1}b},$$

provided that all involved quantities exist.

The Sherman-Morrison formula can be generalized to rational functions [28], but for other matrix functions the update is usually full-rank. However, it was observed in [18] that in many contexts the update $f(A + bc^T) - f(A)$ is *numerically* low-rank; see Figure 5.1 for an example.

In [18, 17] the updates $f(A + R) - f(A)$ are approximated by low-rank matrices obtained by a Krylov subspace projection method, which we review in Chapter 6.

5.1.1 Contributions

In Chapter 6 we present two extensions of the convergence analysis of the low-rank updates algorithm presented in [18]. First, we show that the error of the low-rank update of matrix functions can be linked to polynomial approximation of the derivative of the function, which simplifies the analysis from [18] for non-symmetric matrices. Second, we show that the update of the trace of matrix functions has a quicker convergence when the matrix and the update are symmetric. These results have been published in [17] and [48], respectively.

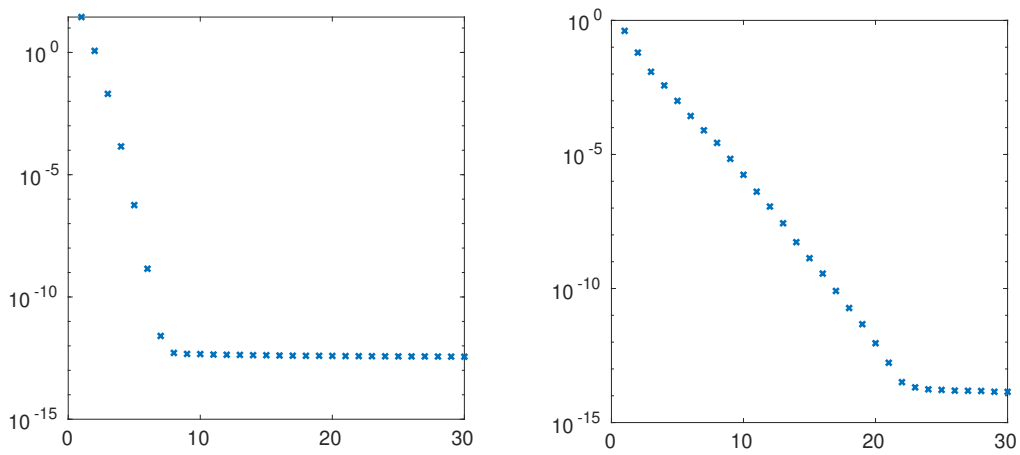


Figure 5.1 – We consider the 1024×1024 *tridiagonal matrix* A *which has* 2 *on the diagonal and* -1 *on the super-diagonal and sub-diagonal. We denote* $A = \text{tridiag}(-1, 2, -1)$*. We consider a rank-1 update of the form* bb^T *where* b *is a unit vector chosen at random, and we consider the update* $f(A + bb^T) - f(A)$ *for* $f(z) = \exp(z)$ *(left) and* $f(z) = \sqrt{z}$ *(right). The two plots show the 30 largest singular values of the update. While the update is* mathematically *full rank, the numerical rank is much lower than* 1024 *in both cases.*

5.2 Functions of rank-structured matrices

The second problem that we address is the computation of functions of rank-structured matrices. For now, let us consider a banded matrix A. If f is well approximated by a low-degree polynomial p on the spectrum of A, the matrix function $f(A)$ can usually be well approximated by the banded matrix $p(A)$. Many a priori results confirm this property. For example, the entries of the inverse of a tridiagonal matrix A decay quickly with the distance to the diagonal, provided that A is well conditioned [55]. Such decay properties extend to inverses of symmetric banded matrices [55], to more general matrix functions of symmetric banded matrices [23], and to symmetric sparse matrices with more general sparsity patterns [25]. When A is not symmetric, one can prove similar results via diagonalization assuming a well conditioned eigenvector matrix [24, 158] or by considering polynomial approximations of $f(z)$ on a larger set, the numerical range of A (see Definition 6.4) [21, 50].

When f cannot be well approximated on the spectrum of A by a low-degree polynomial, often other low-rank structures come to the rescue. Let us illustrate this with an example.

Example 5.3. *Let* $A = \text{tridiag}(-1, 2, -1)$ *and let us consider the inverse of such matrix. The matrix* A^{-1} *is not tridiagonal nor banded, but it has the property that all off-diagonal*

blocks have rank at most 1*. Indeed, consider an off-diagonal block of size* $k \times (n-k)$*, without loss of generality in the upper-right part of* A*, and note that we have the decomposition*

$$A = \begin{array}{c} \overbrace{\qquad\qquad}^{k} \quad \overbrace{\qquad\qquad\qquad}^{n-k} \\ \left[\begin{array}{cccccc|cccccc} 2 & -1 & & & & & & & & & & \\ -1 & \ddots & \ddots & & & & & & & & & \\ & \ddots & 2 & -1 & & & & & & & & \\ & & -1 & 3 & & & & & & & & \\ \hline & & & & & & 3 & -1 & & & & \\ & & & & & & -1 & 2 & \ddots & & & \\ & & & & & & & \ddots & \ddots & \ddots & & \\ & & & & & & & & \ddots & \ddots & \ddots & -1 \\ & & & & & & & & & & -1 & 2 \end{array}\right] \end{array} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \hline 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot \left[\begin{array}{cccc|cccc} 0 & \cdots & 0 & 1 & 1 & 0 & \cdots\cdots & 0 \end{array}\right] = \mathrm{blkdiag}(A_1, A_2) - bb^T,$$

where A_1 *and* A_2 *are* $k \times k$ *and* $(n-k) \times (n-k)$ *matrices and* b *is a vector which contains* 1 *in the* k*-th and* $(k+1)$*th entries; thanks to the Sherman-Morrison formula,* $A^{-1} - \mathrm{blkdiag}(A_1, A_2)^{-1} = A^{-1} - \mathrm{blkdiag}(A_1^{-1}, A_2^{-1})$ *has rank* 1*, and the considered off-diagonal block of* A^{-1} *has rank* 1*.*

The reasoning above can be applied to any invertible tridiagonal matrix A. Similarly, if we consider a rational function r of degree m and apply it to a tridiagonal matrix A, one can prove that the off-diagonal blocks of $r(A)$ have rank at most m.

5.2.1 Existing algorithms

For computing a matrix function $f(A)$ when A is banded, one can take $f(A) \approx p(A)$ if a good polynomial approximation p of the function f is known a priori. Compared to A, the width of the band gets multiplied by the degree of the polynomial p. This technique is used, for instance, in electronic structure methods [84] combined with Chebyshev interpolation [22]. In [24], polynomial approximation is combined with a dropping strategy in order to maintain a low bandwidth in the approximation of $f(A)$. A possible alternative to a priori polynomial approximations is to adapt an existing method for dense matrices to banded matrices, possibly combining it with thresholding in order to maintain sparsity; for example, Newton-Schultz iterations have been used for the sign function in the context of electronic structure calculations [53, 144]. For functions of banded Toeplitz matrices, structured thresholding techniques have been designed in order to maintain a Toeplitz plus low-rank structure [30].

When polynomial approximation of f is difficult, one can instead use an a priori

rational approximation $f(A) \approx r(A)$ for a rational function r of small degree. Matrices with off-diagonal blocks with low rank can be stored conveniently in the HSS format [100], which will be formally introduced in Section 7.1.3, and fast arithmetic can be performed with them. An advantage of rational approximation is that it also works for matrices that have off-diagonal low-rank structure but are not necessarily banded. For the exponential, there exists an excellent rational approximation on the negative real axis [4], which implies a good approximation of $\exp(-A)$ for an SPD matrix A even when the norm of A is large; see [98] for further examples. Another favorable class of functions is the one of Markov functions, that has been recently discussed in [16] in the context of a Toeplitz matrix argument. Rational functions approximating f can also be obtained by discretizing the Cauchy integral representation of the function; this approach is used, for instance, for the exponential operator [81], for step functions arising in the computation of spectral projectors [126], and for matrix functions that may have singularities inside the contour of integration [136]. An alternative to a priori rational approximation is the use of iterative methods, such as for the matrix sign function [92] or the matrix square root of an SPD matrix [100, Section 15.3]; the iterations can be done in HSS arithmetic, and possibly some truncation strategies are needed in order to maintain a low-rank structure.

5.2.2 Contributions

In Chapter 7 we design new algorithms for approximating functions of matrices with off-diagonal low-rank structure – for example banded matrices – which can be decomposed as the sum of a block-diagonal matrix D and a low-rank correction. We approximate the update $f(A) - f(D)$ with the Krylov subspace projection method described in Chapter 6 and compute $f(D)$ recursively, leading to a D&C algorithm. Our convergence analysis of the D&C algorithm is linked to polynomial or rational approximation of the functions, and we show several numerical examples. We also derive a simpler (and faster) algorithm for banded matrices. The results in Chapter 7 are contained in [48].

6 Low-rank updates of matrix functions

Let $A, R \in \mathbb{R}^{n\times n}$ with $\text{rank}(R) \ll n$ and assume that $f(A)$ has already been computed. In this chapter we consider algorithms for computing the update

$$f(A+R) - f(A), \tag{6.1}$$

without computing $f(A+R)$ from scratch. In Section 6.1 we review the low-rank updates algorithm proposed in [18, 17]. At the beginning of Section 6.2 we review some convergence results from [18, 17], then in Section 6.2.1 we prove a new result that relates the convergence to polynomial approximation of the derivative of f. In Section 6.2.2 we prove that under suitable assumptions the trace of the update (6.1) converges faster than the full matrix function update.

6.1 Approximation via Krylov subspace projections

The fact that the update (6.1) is often numerically low-rank – as mentioned in Chapter 5 – motivates the search for approximations of the form

$$f(A+R) - f(A) \approx U_m X_m(f) V_m^T, \tag{6.2}$$

where U_m, V_m are orthonormal bases of suitable (low-dimensional) subspaces $\mathcal{U}_m, \mathcal{V}_m$ of $\mathbb{R}^n$ and $X_m(f)$ is a suitably chosen (small) matrix.

A natural choice for $\mathcal{U}_m$ and $\mathcal{V}_m$ are Krylov subspaces.

Definition 6.1. *The* polynomial Krylov subspace *associated to A and a (block) vector $B \in \mathbb{R}^{n\times r}$ is*

$$\mathcal{K}_m(A,B) := \operatorname{span}\left\{B, AB, A^2B, \dots, A^{m-1}B\right\}.$$

The rational Krylov subspace *[164] associated with $q_m(z) = (z-\xi_1)\cdots(z-\xi_m)$ for prescribed poles $\xi_1,\dots,\xi_m \in \mathbb{C}$ is*

$$q_m(A)^{-1}\mathcal{K}_m(A,B) := \operatorname{span}\left\{q_m(A)^{-1}B, q_m(A)^{-1}AB, q_m(A)^{-1}A^2B, \dots, q_m(A)^{-1}A^{m-1}B\right\}.$$

Given a factorization of the low-rank matrix R,

$$R = BJC^T, \quad B, C \in \mathbb{R}^{n\times \operatorname{rank}(R)}, \quad J \in \mathbb{R}^{\operatorname{rank}(R)\times\operatorname{rank}(R)},$$

we let $\mathcal{U}_m$ and $\mathcal{V}_m$ be Krylov subspaces generated with the matrices A and A^T and starting (block) vectors B and C, respectively. To make sure that $\mathcal{U}_m$ and $\mathcal{V}_m$ are real when using rational Krylov subspaces, the set of poles is assumed to be closed under complex conjugation. Also, we allow for infinite poles and consider the polynomial Krylov subspace as the particular case where $\xi_j = \infty$, $j = 1,\dots,m$.

Let us now consider the choice of $X_m(f)$ in (6.2). Lemma 2.2 in [18] states that

$$f\left(\begin{bmatrix} A & R \\ 0 & A+R \end{bmatrix}\right) = \begin{bmatrix} f(A) & f(A+R)-f(A) \\ 0 & f(A+R) \end{bmatrix}. \tag{6.3}$$

For this reason, the coefficient matrix $X_m(f)$ is set to be the $(1,2)$-block of the (small) matrix

$$f\left(\begin{bmatrix} U_m^TAU_m & U_m^TRV_m \\ 0 & V_m^T(A+R)V_m \end{bmatrix}\right).$$

The whole procedure is summarized in Algorithm 6.1. The orthonormal bases U_m, V_m of $q_m(A)^{-1}\mathcal{K}_m(A,B)$, $q_m(A^T)^{-1}\mathcal{K}_m(A^T,C)$ are computed with the block rational Arnoldi method described in [68, 26]. This computation is performed incrementally with respect to m and yields the compressed matrices $U_m^TAU_m$ and $V_m^T(A+R)V_m$ nearly for free. For

choosing m, we use the following heuristic:

$$\begin{aligned}\|f(A+R)-f(A)-U_{m-d}X_{m-d}(f)V_{m-d}^T\|_2 &\approx \|U_mX_m(f)V_m^T-U_{m-d}X_{m-d}(f)V_{m-d}^T\|_2\\ &= \left\| X_m(f) - \begin{bmatrix} X_{m-d}(f) & 0 \\ 0 & 0 \end{bmatrix} \right\|_2,\end{aligned}$$

for a small integer d, the so called *lag parameter*. Each step of the block rational Arnoldi method in lines 4-5 requires either matrix-vector products with A, A^T (for an infinite pole) or solving shifted linear systems with A, A^T (for a finite pole). When only a few different finite poles are present, it can be beneficial to precompute the LU factorization of the shifted matrix A and reuse it across several steps. We refer to [17, Section 3.1] and the references therein concerning further implementation details.

Algorithm 6.1 Krylov subspace projection for approximating $f(A+R)-f(A)$

Input: Matrix A, update $R = BJC^T$, poles $\xi = (\xi_1, \ldots, \xi_{m_{\max}})^T$, function $f(z)$, lag parameter d, desired accuracy ε

1: **function** KRYLOV_PROJ$(A, B, J, C, \xi, f(z), d, \varepsilon)$
2: **for** $m = 1, \ldots, m_{\max}$ **do**
3: $\quad q_m(z) \leftarrow (z-\xi_1)\cdots(z-\xi_m)$
4: $\quad$ Compute orthonormal basis U_m of $q_m(A)^{-1}\mathcal{K}_m(A,B)$
5: $\quad$ Compute orthonormal basis V_m of $q_m(A^T)^{-1}\mathcal{K}_m(A^T,C)$
6: $\quad$ Compute $X_m(f)$ as the $(1,2)$ block of $f\left(\begin{bmatrix} U_m^TAU_m & U_m^TRV_m \\ 0 & V_m^T(A+R)V_m \end{bmatrix}\right)$
7: $\quad$ **if** $m > d$ and $\left\| X_m(f) - \begin{bmatrix} X_{m-d}(f) & 0 \\ 0 & 0 \end{bmatrix} \right\|_2 < \varepsilon$ **then**
8: $\quad\quad$ **break**
9: $\quad$ **end if**
10: **end for**
11: **return** $U_m, X_m(f), V_m$

When A and R are symmetric, we can choose $C = B$. It follows that $U_m = V_m$ and hence only one basis needs to be generated; line 5 of Algorithm 6.1 is skipped. Moreover,

the computation of $X_m(f)$ in line 6 simplifies to

$$X_m(f) = f\left(U_m^T(A+R)U_m\right) - f(U_m^T A U_m).$$

6.2 Exactness and convergence results for Algorithm 6.1

In the following, we let

$$E_m(f) := f(A+R) - f(A) - U_m X_m(f) V_m^T \tag{6.4}$$

denote the error of the approximation returned by Algorithm 6.1. Moreover, we let Π_m denote the space of polynomials with degree bounded by m and Π_m/q_m denote the set of all rational functions of the form $p(z)/q_m(z)$ with $p \in \Pi_m$.

The properties of Krylov subspaces and the choice of $X_m(f)$ in line 6 of Algorithm 6.1 allow us to prove the following exactness result.

Theorem 6.2 ([18, Theorem 3.2] and [17, Theorem 3.3])**.** *When using rational Krylov subspaces associated to poles $\xi_1, \ldots, \xi_m$ (possibly infinite), it holds that*

$$E_m(f) = 0$$

for all $f \in \Pi_m/q_m$.

Remark 6.3. *For future reference, we note here that the exactness result in Theorem 6.2 also holds when U_m and V_m are orthonormal bases of subspaces of $\mathbb{R}^n$ which* contain *the Krylov subspaces $\mathcal{U}_m$ and $\mathcal{V}_m$, respectively.*

Such exactness results can be turned into convergence results via polynomial/rational approximation. For a function f and a set $\mathbb{D} \subseteq \mathbb{C}$ we denote by $\|f\|_{\mathbb{D}} := \sup_{z\in\mathbb{E}} |f(z)|$ the supremum norm on $\mathbb{D}$.

Definition 6.4. *The* numerical range *(or* field of values*) of A is*

$$W(A) := \{z^* A z \mid z \in \mathbb{C}^n, \|z\|_2 = 1\}.$$

Theorem 6.5 ([18, Theorem 4.1] and [17, Theorem 4.5])**.** *Let A and $R = BJB^T$ be symmetric, let the set of poles be closed under complex conjugation, and let $U_m = V_m$ be*

an orthonormal basis of $q_m(A)^{-1}\mathcal{K}_m(A,B)$*. Furthermore, let* f *be analytic in a compact domain* $\mathbb{D}$ *containing the union of* $W(A)$ *and* $W(A+R)$*. Then the error* (6.4) *returned by Algorithm 6.1 satisfies*

$$\|E_m(f)\|_2 \leq 4 \min_{r \in \Pi_m/q_m} \|f - r\|_{\mathbb{D}}.$$

When A and R are symmetric, the sets $W(A)$ and $W(A+R)$ are closed bounded intervals on the real line. For non-symmetric matrices, one needs to consider an approximation problem on a larger set. We state the result in the case of polynomial Krylov subspaces.

Theorem 6.6 ([18, Theorem 4.2])**.** *Let* $\mathcal{A} := \begin{bmatrix} A & R \\ 0 & A+R \end{bmatrix}$ *and assume that* f *is analytic in a neighborhood of a compact set* $\mathbb{D} \supseteq W(\mathcal{A})$*. When polynomial Krylov subspaces are used in Algorithm 6.1 we have that*

$$\|E_m(f)\|_2 \leq (2 + 2\sqrt{2}) \min_{p \in \Pi_m} \|f - p\|_{\mathbb{D}}.$$

The numerical range of $\mathcal{A}$ can be much larger than the union of $W(A)$ and $W(A+R)$. Indeed, there are situations [18, Figure 6.2] in which $W(\mathcal{A})$ contains a singularity of f and hence the bound becomes void. In order to deal with these situations, alternative convergence results based on integral representations have been developed in [18, 17]. In Section 6.2.1 we provide an alternative way to deal with this problem in the case of polynomial Krylov subspaces: We provide a convergence result based on polynomial approximation of the derivative of f on a convex set containing $W(A)$ and $W(A+R)$.

6.2.1 Convergence analysis for polynomial Krylov subspaces

In the rest of this chapter, we consider the case in which $\mathcal{U}_m$ and $\mathcal{V}_m$ are polynomial Krylov subspaces. The following lemma is key to our analysis; its proof uses a recent bound on the Fréchet derivative from [49].

Lemma 6.7. *Let* $\mathcal{B} = \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{bmatrix}$ *for some matrices* $B_{11}, B_{12}, B_{22} \in \mathbb{R}^{n \times n}$*, let* $\mathbb{D}$ *be a*

compact convex set containing $W(B_{11})$ and $W(B_{22})$, and let f be analytic in $\mathbb{D}$. Then

$$\|[f(\mathcal{B})]_{1,2}\|_F \leq (1+\sqrt{2})^2 \|f'\|_{\mathbb{D}} \|B_{12}\|_F.$$

Proof. For $n \times n$ matrices A and B, let $L_f(A,B)$ denote the Fréchet derivative of f at A applied to the matrix B and let $L_f(A,\cdot)$ denote the corresponding linear operator represented as an $n^2 \times n^2$ matrix. By [110, Theorem 4.12],

$$f(\mathcal{B}) = f(\mathcal{D}) + L_f(\mathcal{D},\mathcal{N}), \text{ where } \mathcal{D} := \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix} \text{ and } \mathcal{N} := \begin{bmatrix} 0 & B_{12} \\ 0 & 0 \end{bmatrix}.$$

Because $f(\mathcal{D})$ is block-diagonal, we have that

$$\|[f(\mathcal{B})]_{1,2}\|_F = \|L_f(\mathcal{D},\mathcal{N})\|_F \leq \|L_f(\mathcal{D},\cdot)\|_2 \cdot \|B_{12}\|_F.$$

Corollary 5.1 in [49] states that $\|L_f(\mathcal{D},\cdot)\|_2 \leq (1+\sqrt{2})^2 \|f'\|_{W(\mathcal{D})}$, which concludes the proof because $W(\mathcal{D})$, as the convex hull of $W(B_{11})$ and $W(B_{22})$, is contained in $\mathbb{D}$. □

Lemma 6.7 applied to the matrix $\begin{bmatrix} A & R \\ 0 & A+R \end{bmatrix}$ combined with (6.3) immediately implies the following result, which might be of independent interest.

Corollary 6.8. *Let $A, R \in \mathbb{R}^{n\times n}$, let $\mathbb{D}$ be a compact convex set containing the union of $W(A)$ and $W(A+R)$, and let f be analytic in $\mathbb{D}$. Then*

$$\|f(A+R) - f(A)\|_F \leq (1+\sqrt{2})^2 \|f'\|_{\mathbb{D}} \|R\|_F. \tag{6.5}$$

When A and R are Hermitian, it is well known that the inequality (6.5) holds without the constant $(1+\sqrt{2})^2$; see, e.g., [171, Proposition 3.1.5]. For general diagonalizable matrices A and $A+R$, Corollary 2.4 in [82] states that

$$\|f(A+R) - f(A)\|_F \leq \kappa_{\text{ev}}(A)\kappa_{\text{ev}}(A+R) \max|f'| \cdot \|R\|_F,$$

where $\kappa_{\text{ev}}(A)$, $\kappa_{\text{ev}}(A+R)$ are the condition numbers of any eigenvector matrices of A and $A+R$, respectively. The maximum of $|f'|$ is taken over the convex hull of the

spectra of $A + R$ and A. Corollary 6.8 instead holds for any matrix and does not feature the potentially large constant $\kappa_{\text{ev}}(A)\kappa_{\text{ev}}(A + R)$, at the cost of bounding f' on a larger domain.

We are now prepared to state a convergence result for Algorithm 6.1.

Theorem 6.9. *Let $A \in \mathbb{R}^{n\times n}$, $R = BJC^T \in \mathbb{R}^{n\times n}$, and let f be analytic in a compact convex set $\mathbb{D}$ containing $W(A)$ and $W(A + R)$. Let U_m and V_m be orthonormal bases of $\mathcal{U}_m = \mathcal{K}_m(A, B)$ and $\mathcal{V}_m = \mathcal{K}_m(A^T, C)$, respectively. Then the error of Algorithm 6.1 satisfies*

$$\|E_m(f)\|_F \leq 2(1 + \sqrt{2})^2\|R\|_F \inf_{p\in\Pi_{m-1}} \|f' - p\|_{\mathbb{D}}.$$

Proof. The first part of the proof is the same as in Theorem 4.2 in [18]: Theorem 6.2 implies that for all $q \in \Pi_m$ we have $E_m(f) = E_m(f - q)$, therefore

$$\begin{aligned}\|E_m(f)\|_F &= \|(f - q)(A + R) - (f - q)(A) - U_m X_m(f - q)V_m^T\|_F \\ &\leq \|(f - q)(A + R) - (f - q)(A)\|_F + \|U_m X_m(f - q)V_m^T\|_F \\ &\leq \|(f - q)(A + R) - (f - q)(A)\|_F + \|X_m(f - q)\|_F. \end{aligned} \tag{6.6}$$

Moreover, by definition (line 6 in Algorithm 6.1), we have $X_m(f - q) = [(f - q)(\tilde{\mathcal{A}})]_{1,2}$, where $\tilde{\mathcal{A}} := \begin{bmatrix} U_m^T A U_m & U_m^T R V_m \\ 0 & V_m^T(A + R)V_m \end{bmatrix}$. We can now leverage Corollary 6.8 to get

$$\|(f - q)(A + R) - (f - q)(A)\|_F \leq (1 + \sqrt{2})^2\|(f - q)'\|_{\mathbb{D}}\|R\|_F.$$

and Lemma 6.7 to get

$$\|X_m(f - q)\|_F \leq (1 + \sqrt{2})^2\|(f - q)'\|_{\mathbb{D}}\|U_m^T R V_m\|_F \leq (1 + \sqrt{2})^2\|(f - q)'\|_{\mathbb{D}}\|R\|_F,$$

because of the inclusions $W(U_m^T A U_m) \subseteq W(A)$ and $W(V_m^T(A + R)V_m) \subseteq W(A + R)$. Combining these with (6.6) gives the result of the theorem, because $q' \in \Pi_{m-1}$ can be chosen arbitrarily. □

Remark 6.10. *Let us compare the result of Theorem 6.6 with Theorem 6.9. Note that although the first bound features a somewhat smaller constant and the approximation of f instead of f', the latter has the major advantage that the convex hull of $W(A)$ and*

$W(A+R)$ can be much smaller than the numerical range of $\mathcal{A}$.

6.2.2 Convergence analysis for the trace of the update

Let us consider a symmetric matrix A and a symmetric update $R = BJB^T$. When we are interested in the trace of $f(A+R)$ given that the trace of $f(A)$ has already been computed, Algorithm 6.1 can be used to obtain the approximation

$$\operatorname{tr}\left(f(A+BJB^T) - f(A)\right) \approx \operatorname{tr}\left(U_m X_m(f) U_m^T\right) = \operatorname{tr}\left(X_m(f)\right) \tag{6.7}$$

When polynomial Krylov subspaces are used, we can prove an exactness result for (6.7) which is stronger than Theorem 6.2.

Theorem 6.11. *Let $A \in \mathbb{R}^{n\times n}$ and $R = BJB^T \in \mathbb{R}^{n\times n}$ be symmetric, and let U_m be an orthonormal basis of $\mathcal{K}_m(A,B)$. Then*

$$\operatorname{tr}\left(X_m(p)\right) = \operatorname{tr}\left(p(A+R) - p(A)\right) \text{ for all } p \in \Pi_{2m}.$$

Proof. By linearity it is sufficient to show that the theorem holds for monomials, that is, we need to prove that

$$\operatorname{tr}\left((U_m^T(A+R)U_m)^j\right) - \operatorname{tr}\left((U_m^T A U_m)^j\right) = \operatorname{tr}\left((A+R)^j\right) - \operatorname{tr}\left(A^j\right)$$

for $j = 0, 1, 2, \ldots, 2m$. The left hand side is a sum of terms of the following form:

$$\operatorname{tr}\left((U_m^T A U_m)^{a_0}(U_m^T BJB^T U_m)^{b_1}(U_m^T A U_m)^{a_1}\cdots(U_m^T BJB^T U_m)^{b_h}(U_m^T A U_m)^{a_h}\right), \tag{6.8}$$

for some $h \geq 1$, $a_0, a_h \geq 0$, $a_1, \ldots, a_{h-1} \geq 1$, $b_1, \ldots, b_h \geq 1$, and $a_0 + b_1 + \ldots + a_{h-1} + b_h + a_h = j$. By [165, Lemma 3.1] we have that

$$U_m(U_m^T A U_m)^k U_m^T B = A^k B \tag{6.9}$$

for all $k = 0, \ldots, m-1$. Moreover, it is easy to see that for $k \geq 1$ we have

$$(U_m^T BJB^T U_m)^k = U_m^T (BJB^T)^k U_m = U_m^T B (JB^T B)^{k-1} JB^T U_m.$$

Then, using (6.9) and the cyclic property of the trace we rewrite (6.8) as

$$\begin{aligned}
&\operatorname{tr}\left((U_m^TAU_m)^{a_0}(U_m^TBJB^TU_m)^{b_1}(U_m^TAU_m)^{a_1}\cdots(U_m^TBJB^TU_m)^{b_h}(U_m^TAU_m)^{a_h}\right)\\
&= \operatorname{tr}\left((U_m^TAU_m)^{a_0}U_m^TB\left(\prod_{i=1}^{h-1}C_{a_i,b_i}\right)(JB^TB)^{b_h-1}JB^TU_m(U_m^TAU_m)^{a_h}\right)\\
&= \operatorname{tr}\left(C_{a_0+a_h,b_h}\prod_{i=1}^{h-1}C_{a_i,b_i}\right)
\end{aligned} \tag{6.10}$$

with $C_{a,b} := (JB^TB)^{b-1}JB^TU_m(U_m^TAU_m)^aU_m^TB$ for $b \geq 1$ and $0 \leq a \leq 2m-1$.

We claim that $C_{a,b} = (JB^TB)^{b-1}JB^TA^aB$: if $a \leq m-1$, this follows directly from the exactness property (6.9); if $a \geq m$, we write $C_{a,b}$ as

$$(JB^TB)^{b-1}J\underbrace{B^TU_m(U_m^TAU_m)^{m-1}U_m^T}_{B^TA^{m-1}}A\underbrace{U_m(U_m^TAU_m)^{a-m}U_m^TB}_{A^{a-m}B}$$

and use the exactness property (6.9) on the two selected parts to arrive at the same conclusion. Finally, incorporating the rightmost factor B of C_{a_i,b_i} into $C_{a_{i+1},b_{i+1}}$ we obtain that (6.10) is equal to

$$\operatorname{tr}\left((JB^TB)^{b_h-1}JB^TA^{a_0+a_h}(BJB^T)^{b_1}A^{a_1}\cdots(BJB^T)^{b_{h-1}}A^{a_{h-1}}B\right).$$

By means of the cyclic property of the trace we finally get

$$\operatorname{tr}\left(A^{a_0}(BJB^T)^{b_1}A^{a_1}\cdots(BJB^T)^{b_h}A^{a_h}\right)$$

which matches the terms in the expansion of $\operatorname{tr}\left((A+BJB^T)^j\right) - \operatorname{tr}(A^j)$. □

The following theorem provides an a priori estimate on the error of the approximation of the trace of a matrix function update obtained by Algorithm 6.1.

Theorem 6.12. *Let A and $R = BJB^T$ be symmetric and let f be defined on an interval $\mathbb{D} \subset \mathbb{R}$ containing the eigenvalues of A and $A+R$. Then*

$$\left|\operatorname{tr}\left(f(A+BJB^T) - f(A)\right) - \operatorname{tr}\left(X_m(f)\right)\right| \leq 4n \min_{p\in\Pi_{2m}} \|f-p\|_{\mathbb{D}}.$$

Proof. By Theorem 6.11, for all polynomials $p \in \Pi_{2m}$ we have that

$$\begin{aligned}
&|\operatorname{tr}(f(A+R) - f(A)) - \operatorname{tr}(X_m(f))| \\
&\quad = |\operatorname{tr}((f-p)(A+R)) - \operatorname{tr}((f-p)(A)) + \\
&\qquad \operatorname{tr}((f-p)(U_m^T(A+R)U_m)) - \operatorname{tr}((f-p)(U_m^T A U_m))| \\
&\quad \leq |\operatorname{tr}((f-p)(A+R))| + |\operatorname{tr}((f-p)(A))| \\
&\qquad + |\operatorname{tr}((f-p)(U_m^T(A+R)U_m))| + |\operatorname{tr}((f-p)(U_m^T A U_m))| \\
&\quad \leq n\|(f-p)(A+R)\|_2 + n\|(f-p)(A)\|_2 \\
&\qquad + n\|(f-p)(U_m^T(A+R)U_m)\|_2 + n\|(f-p)(U_m^T A U_m)\|_2.
\end{aligned}$$

For a normal matrix X and a function g, it holds that $\|g(X)\|_2 \leq \|g\|_{\Lambda(X)}$, where $\Lambda(X)$ denotes the convex hull of the eigenvalues of X. As $\Lambda(A+R)$, $\Lambda(A)$, $\Lambda(U_m^T(A+R)U_m)$, and $\Lambda(U_m^T A U_m)$ are all contained in $\mathbb{D}$, it follows that the right hand side of the above equation is upper bounded by $4n\|f-p\|_{\mathbb{D}}$. Taking the minimum over all polynomials $p \in \Pi_{2m}$ concludes the proof. □

Example 6.13. *Consider SPD matrices $A, R \in \mathbb{R}^{n\times n}$, and denote by $[\alpha, \beta]$ an interval containing the eigenvalues of A and $A+R$. The best polynomial approximation error on such interval when $f(z) = \sqrt{z}$ decreases as γ^m, where*

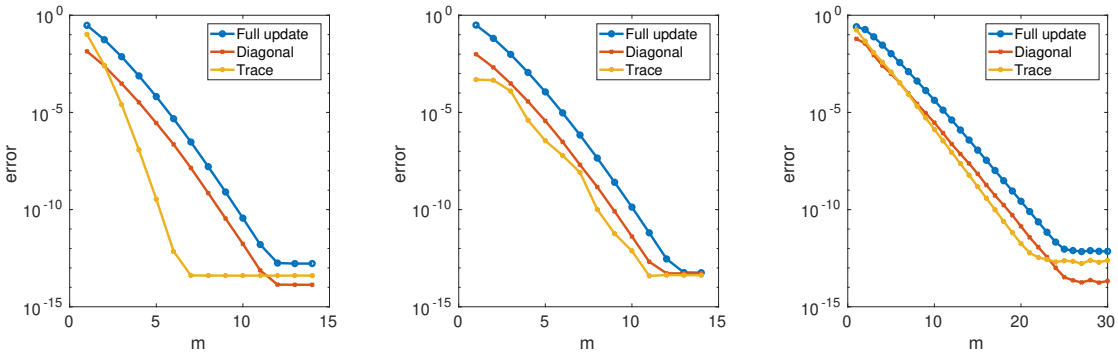
$$\gamma := \frac{\sqrt{\beta/\alpha} - 1}{\sqrt{\beta/\alpha} + 1};$$

see, e.g., [181, Theorem 8.2]. Therefore, the error in the approximation of $f(A+R) - f(A)$ via Algorithm 6.1 decreases geometrically with rate γ, while the error in the approximation of $\operatorname{tr}(f(A+R) - f(A))$ decreases with rate γ^2 thanks to Theorem 6.12, that is, twice as fast.

Numerical examples

Figure 6.1 reports numerical experiments to explore the scope of the result of Theorem 6.12. In Figure 6.1(a) we have applied Algorithm 6.1 with polynomial Krylov subspaces to a random symmetric matrix A and random symmetric update R. As expected from Theorem 6.12, the trace of the update converges at double speed with respect to the error of the full update.

Figure 6.1(b) features polynomial Krylov subspaces with a random non-symmetric matrix A and Figure 6.1(c) features a rational Krylov subspace method applied to symmetric A and R. In these two situations, there is no significant difference in the convergence of the trace of the update.



(a) Polynomial Krylov applied to symmetric A and R

(b) Polynomial Krylov applied to non-symmetric A

(c) Rational Krylov applied to symmetric A and R

Figure 6.1 – Convergence of the errors $\|f(A+R) - f(A) - U_m X_m(f) V_m^T\|_F$, $\|\mathrm{diag}(f(A+R) - f(A) - U_m X_m(f) V_m^T)\|_2$, and $|\,\mathrm{tr}(f(A+R) - f(A) - U_m X_m(f) V_m^T)|$ for $f = \exp$.

7 Divide-and-conquer algorithms for matrix functions

In this chapter, we design new algorithms for approximating functions of matrices that can be recursively decomposed as the sum of a block-diagonal matrix D and a low-rank correction. This is the case for banded matrices, HSS matrices (see Section 7.1.3), and sparse matrices corresponding to graphs with community structure [146].

The update $f(A) - f(D)$ is approximated using Algorithm 6.1 with suitable Krylov subspaces. We perform the evaluation of $f(D)$ recursively, leading to a D&C algorithm. Similarly to the a priori bounds on $f(A)$ mentioned in Chapter 5, we prove that the effectiveness of the D&C algorithm is related to best polynomial or rational approximation. The advantage of our approach is that the use of Krylov subspaces bypasses the need of choosing an a priori polynomial or rational approximation of f and this can be beneficial if there are some outliers in the spectrum of A.

For banded matrices A, polynomial Krylov subspaces associated to low-rank updates inherit sparsity. Thanks to this fact, we can develop an algorithm that allows for a more compact description of the low-rank updates and a more efficient implementation, which we call *block diagonal splitting method*. Our algorithm is based on covering A with overlapping blocks and only needs the evaluation of f on these blocks. A related, although significantly different, technique has been proposed in [170] for approximating the exponential of infinite banded matrices. The equivalence of our method with low-rank updates allows us to prove a convergence result that connects the error of the algorithm with polynomial approximations of f. When only the diagonal of $f(A)$ is needed, we observe accelerated convergence and we confirm this by theoretical results.

The remainder of the chapter is organized as follows. Section 7.1 is dedicated to the D&C algorithm for matrix functions and its convergence analysis. Numerical experiments for banded and HSS matrix arguments are presented in Section 7.2. In Section 7.3 we present and analyze the block diagonal splitting algorithm for banded matrices. The performance of the splitting algorithm is validated in Section 7.4.

7.1 Divide-and-conquer for matrix functions

7.1.1 Divide-and-conquer for matrices with low-rank off-diagonal blocks

In this section we use low-rank updates to devise a D&C method for functions of matrices that have low-rank off-diagonal blocks. More specifically, let us assume that $A \in \mathbb{R}^{n\times n}$ can be block-partitioned as

$$A = \underbrace{\begin{bmatrix} A_{11} & \\ & A_{22} \end{bmatrix}}_{A_D} + \underbrace{\begin{bmatrix} & A_{12} \\ A_{21} & \end{bmatrix}}_{A_O}, \quad A_{11} \in \mathbb{R}^{n_1\times n_1}, \quad A_{22} \in \mathbb{R}^{n_2\times n_2}, \tag{7.1}$$

where the off-diagonal part A_O has low rank and the diagonal blocks can be recursively block-partitioned in the same fashion. Examples of matrix structures that have this property are banded matrices and HSS matrices [41]; see also Section 7.1.3 below.

The computation of $f(A)$ is split in two tasks: computing $f(A_D)$ and computing $f(A) - f(A_D)$. The latter quantity is approximated via Algorithm 6.1 exploiting that $A_O = A - A_D$ has low rank; the former decouples into the computation of $f(A_{11})$ and $f(A_{22})$. Since we assume that the blocks A_{ii} can again be decomposed into the form (7.1), the described procedure is applied recursively for computing $f(A_{ii})$, $i = 1, 2$. Finally, when the size of a block A_{ii} is below a minimal block size $n_i \leq n_{\min}$, we evaluate $f(A_{ii})$ with a standard dense method, like the scaling and squaring method [110] for $f = \exp$.

Algorithm 7.1 summarizes the described D&C method for matrix functions. The D&C method simplifies when certain selected quantities of $f(A)$, like the diagonal or the trace, are of interest. Because of linearity, it suffices to evaluate the diagonal or the trace of the low-rank update $UXV^T \approx f(A) - f(A_D)$; see lines 19 and 21 of Algorithm 7.1.

Theorem 6.2 directly implies the following result.

Algorithm 7.1 Template of D&C algorithm for matrix functions

Input: Matrix $A \in \mathbb{R}^{n\times n}$, poles ξ, function $f(z)$, lag parameter d, desired accuracy ε, minimum block size $n_{\min}$, parameter `flag` that indicates whether the full matrix function, its diagonal, or its trace is needed

1: **function** D&C_FUNM$(A, \xi, f(z), d, \varepsilon, n_{\min}$, `flag`$)$
2: **if** $n \le n_{\min}$ **then**
3: **if** `flag` = "full" **then**
4: **return** $f(A)$
5: **else if** `flag` = "diagonal" **then**
6: **return** $\texttt{diag}(f(A))$
7: **else if** `flag` = "trace" **then**
8: Compute the eigenvalues λ_j, $j = 1, \dots, n$, of A
9: **return** $\sum_{j=1}^{n} f(\lambda_j)$
10: **end if**
11: **end if**
12: Given a decomposition (7.1), retrieve a low-rank factorization $A_O = BJC^T$
13: $[U, X, V] \leftarrow$KRYLOV_PROJ$(A_D, B, J, C, \xi, f(z), d, \varepsilon)$ *(Algorithm 6.1)*
14: $F_{11} \leftarrow$ D&C_FUNM$(A_{11}, \xi, f(z), d, \varepsilon, n_{\min}$, `flag`$)$ *(Recursion)*
15: $F_{22} \leftarrow$ D&C_FUNM$(A_{22}, \xi, f(z), d, \varepsilon, n_{\min}$, `flag`$)$ *(Recursion)*
16: **if** `flag` = "full" **then**
17: **return** $\begin{bmatrix} F_{11} & \\ & F_{22} \end{bmatrix} + UXV^T$
18: **else if** `flag` = "diagonal" **then**
19: **return** $\begin{bmatrix} F_{11} \\ F_{22} \end{bmatrix} + \texttt{diag}(UXV^T)$
20: **else if** `flag` = "trace" **then**
21: **return** $F_{11} + F_{22} + \texttt{trace}(V^T U X)$
22: **end if**

Proposition 7.1. *Let $A \in \mathbb{R}^{n\times n}$ and consider $q_m(z) := \prod_{i=1}^{m}(z - \xi_i)$ for a set of m poles $\xi_1, \dots, \xi_m \in \mathbb{C} \cup \{\infty\}$ closed under complex conjugation. Then Algorithm 7.1 applied to A and a function $f \in \Pi_m/q_m$ is exact, provided that Algorithm 6.1 called in Line 13 utilizes all m poles.*

7.1.2 Algorithm 7.1 for banded matrices

Let us first consider the application of Algorithm 7.1 to a banded matrix A with bandwidth b, that is, such that $a_{ij} = 0$ whenever $|i - j| > b$. Then the off-diagonal part A_O in the decomposition (7.1) has rank at most $2b$. Under the idealistic assumption that Algorithm 6.1 converges in a constant number of iterations (independent of n), computing the low-rank update on the top level of recursion requires $\mathcal{O}(b^2 n)$ operations when

using either polynomial or rational Krylov subspaces. Thus, the total complexity of Algorithm 7.1 is $\mathcal{O}(b^2 n \log n)$, provided that $n_{\min} = \mathcal{O}(1)$ (that is, the minimum block size is a constant which is independent of n).

Remark 7.2. *By an appropriate correction of the diagonal blocks in the decomposition* (7.1)*, it is possible to reduce the rank of the off-diagonal part to b (similarly to Example 5.3). Although this clearly has the potential to result in lower-dimensional Krylov subspaces in the low-rank update, it also bears the danger of leading to diagonal blocks for which f is not defined or difficult to approximate. When A is SPD then the rank-b update can be chosen such that the diagonal blocks remain SPD [124, Section 4.4.2].*

Remark 7.3. *When Algorithm 7.1 is used with polynomial Krylov subspaces for banded A then it can be shown that the output is again banded (but with larger bandwidth). However, in such a situation a much simpler approach is possible, which will be described in Section 7.3.*

7.1.3 Storing the output of Algorithm 7.1 using HSS matrices

Except for the situation described in Remark 7.3, the approximation of $f(A)$ constructed in line 17 of Algorithm 7.1 is not banded. To still efficiently represent this approximation, we use HSS matrices.

In the following we give a brief introduction to HSS matrices; see [132, 137, 194] for more details. The HSS format is associated to a recursive partitioning of the matrix, which we now formalize.

Definition 7.4. *Given $n \in \mathbb{N}$, let $\mathcal{T}_L$ be a perfect binary tree of depth L whose nodes are subsets of $\{1, \dots, n\}$. We say that $\mathcal{T}_L$ is a* cluster tree *if it satisfies:*

- *The root is $I_1^0 := I = \{1, \dots, n\}$.*
- *The nodes at level ℓ, denoted by $I_1^\ell, \dots, I_{2^\ell}^\ell$, form a partitioning of $\{1, \dots, n\}$ into consecutive indices:*
$$I_i^\ell = \{n_{i-1}^{(\ell)} + 1 \dots, n_i^{(\ell)} - 1, n_i^{(\ell)}\}$$
for some integers $0 = n_0^{(\ell)} \leq n_1^{(\ell)} \leq \dots \leq n_{2^\ell}^{(\ell)} = n$, $\ell = 0, \dots, L$. In particular, if $n_{i-1}^{(\ell)} = n_i^{(\ell)}$ then $I_i^\ell = \emptyset$.
- *The children form a partitioning of their parent.*

Usually, the cluster tree $\mathcal{T}_L$ is defined such that the index sets on the same level ℓ have nearly equal cardinalities and the depth of the tree is determined by a minimal diagonal block size $n_{\min}$ for stopping the recursion. In particular, if $n = 2^L n_{\min}$, such a construction yields a perfect tree of depth L in which all the leaves correspond to $n_{\min}$ indices.

The block structure of an HSS matrix is determined by $\mathcal{T}_L$. The diagonal blocks $D_i := A(I_i^L, I_i^L)$ are treated as (small) dense matrices. All other blocks are of the form $A(I_i^\ell, I_j^\ell)$ for some siblings I_i^ℓ, I_j^ℓ in $\mathcal{T}_L$ (that is, for any pairs of nodes with the same parent); for an HSS matrix of rank k, each block has rank (at most) k. Therefore, each block admits a factorization

$$A(I_i^\ell, I_j^\ell) = U_i^{(\ell)} S_{i,j}^{(\ell)} (V_j^{(\ell)})^T, \quad S_{i,j}^{(\ell)} \in \mathbb{R}^{k\times k}, \quad U_i^{(\ell)} \in \mathbb{R}^{n_i^{(\ell)}\times k}, \quad V_j^{(\ell)} \in \mathbb{R}^{n_j^{(\ell)}\times k}.$$

Moreover, the factors $U_i^{(\ell)}, V_j^{(\ell)}$ are nested across different levels of $\mathcal{T}_L$ [194]. More specifically, there exist so called *translation operators*, $R_{U,i}^{(\ell)}, R_{V,j}^{(\ell)} \in \mathbb{R}^{2k\times k}$ such that

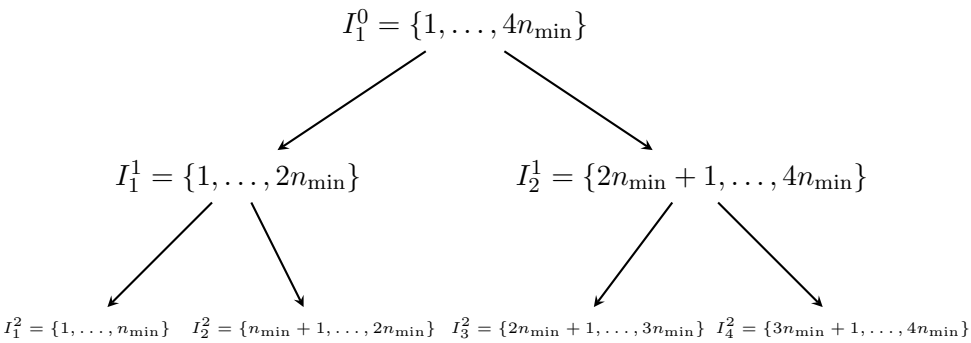
$$U_i^{(\ell)} = \begin{bmatrix} U_{2i-1}^{(\ell+1)} & 0 \\ 0 & U_{2i}^{(\ell+1)} \end{bmatrix} R_{U,i}^{(\ell)}, \qquad V_j^{(\ell)} = \begin{bmatrix} V_{2j-1}^{(\ell+1)} & 0 \\ 0 & V_{2j}^{(\ell+1)} \end{bmatrix} R_{V,j}^{(\ell)},$$

where $I_{2i-1}^{\ell+1}, I_{2i}^{\ell+1}$ and $I_{2j-1}^{\ell+1}, I_{2j}^{\ell+1}$ denote the children of I_i^ℓ and I_j^ℓ, respectively. Given the bases $U_i^{(L)}$ and $V_i^{(L)}$ at the deepest level L, the low-rank factors $U_i^{(\ell)}$ and $V_i^{(\ell)}$ for the higher levels $\ell = 1, \ldots, L-1$, can be retrieved by means of the translation operators. Figure 7.1 illustrates the HSS format graphically.

Summarizing, for storing an HSS matrix A we need:

- the diagonal blocks D_i,
- the bases $U_i^{(L)}$, $V_i^{(L)}$,
- the core factors $S_{i,j}^{(\ell)}$ and $S_{j,i}^{(\ell)}$,
- the translation operators $R_{U,i}^{(\ell)}$, $R_{V,i}^{(\ell)}$.

The storage cost is $\mathcal{O}(kn)$. Note that we have used a uniform rank k for the off-diagonal blocks to simplify the description; in practice these ranks are chosen adaptively. The HSS



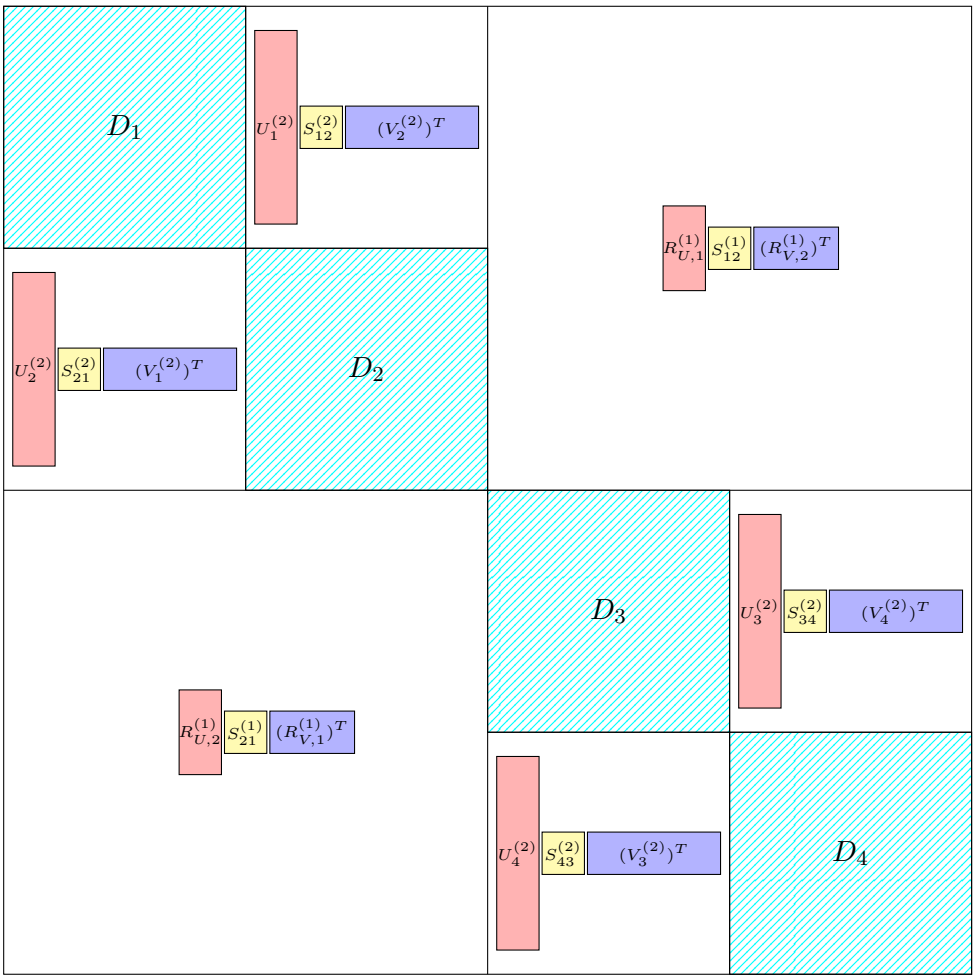


Figure 7.1 – Illustration of an HSS representation of a matrix of size $n \times n$ with cluster tree of depth $L = 2$, where for simplicity we assume that $n = 4n_{\min}$ is a multiple of 4.

format also allows for fast algorithms for matrix operations. For example, multiplying a matrix with HSS rank k with a vector costs $\mathcal{O}(kn)$; solving a linear system and computing the corresponding decomposition costs $\mathcal{O}(k^2n)$ (see, e.g., [137, Figure 6]).

In the context of Algorithm 7.1, we choose a cluster tree that aligns with the (recursive) decompositions (7.1). In turn, the sum at line 17 is performed using HSS arithmetic and is combined with a re-compression step to mitigate the increase of the HSS rank. This costs $\mathcal{O}(k^2n)$ operations, assuming that the HSS ranks of F_{11}, F_{22} and the rank of UXV^T are $\mathcal{O}(k)$ [137].

7.1.4 Algorithm 7.1 for HSS matrices

We now discuss the situation when the HSS structure is not only used for storing the output of Algorithm 7.1 but when the input matrix A itself is also an HSS matrix. In this case the decomposition (7.1) is aligned with the cluster tree $\mathcal{T}_L$ associated with A as this choice guarantees that the rank of A_O is bounded by $2k$ and that the outcome inherits the same cluster tree of the input matrix. In Algorithm 7.1 we can exploit the fact that fast algorithms are available for HSS matrices as follows:

- We retrieve B and C in line 12 by means of the translation operators ($\mathcal{O}(k^2n)$).
- The Krylov subspaces in Algorithm 6.1 are generated by performing matrix-vector products and/or solving shifted linear systems with HSS algorithms.
- We use the HSS structures of A_{11}, A_{22} in the recursive calls in lines 14-15 and we return HSS matrices F_{11} and F_{22}.

Let us analyze the cost of Algorithm 7.1 for the input $(\mathcal{T}_L, k)$-HSS matrix A, with $L = \mathcal{O}(\log(n))$, and `flag` = "full". We again make the idealistic assumption that Algorithm 6.1 converges in a constant number of iterations, independent of k and n, and that the outcome of the (compressed) sum at line 17 has always HSS rank $\mathcal{O}(k)$. Then, we have that the low-rank updates at level $\ell \in \{0, 1, \ldots, L-1\}$ cost $\mathcal{O}(k^2(n_i^{(\ell)} - n_{i-1}^{(\ell)}))$, $i = 1, \ldots, 2^\ell$, when using either polynomial or rational Krylov subspaces. Since the sum at line 17 costs $\mathcal{O}(k^2(n_i^{(\ell)} - n_{i-1}^{(\ell)}))$ too, the asymptotic complexity of each non-base level

of the recursion is

$$\mathcal{O}\left(k^2 \sum_{i=1}^{2^\ell} (n_i^{(\ell)} - n_{i-1}^{(\ell)})\right) = \mathcal{O}(k^2 n).$$

The base of the recursion requires us to evaluate $\mathcal{O}(n/n_{min})$ functions of matrices of size at most $n_{\min} \times n_{\min}$; assuming a cubic cost for matrix function evaluations yields $\mathcal{O}(n_{\min}^2 n)$. Hence, the overall complexity of Algorithm 7.1 for HSS matrices is $\mathcal{O}(k^2 n \log(n))$.

7.1.5 Convergence results for D&C algorithm

Convergence results for Algorithm 7.1 can be obtained from the convergence results on low-rank updates of matrix functions discussed in Section 6.2.2. In the following, we let $\mathcal{T}_L$ denote the (perfectly balanced) binary tree of depth L associated with the recursive decompositions performed in line 12.

Theorem 7.5. *Let A be symmetric and let f be a function analytic on an interval $\mathbb{D}$ containing the eigenvalues of A. Suppose that Algorithm 7.1 uses rational Krylov subspaces with poles $\xi_1, \ldots, \xi_m$, closed under complex conjugation, for computing updates. Then the output F_A of Algorithm 7.1 satisfies*

$$\|f(A) - F_A\|_2 \leq 4L \cdot \min_{r \in \Pi_m/q_m} \|f - r\|_{\mathbb{D}},$$

where $q_m(z) = \prod_{i=1}^m (z - \xi_i)$.

Proof. Using the index sets contained in $\mathcal{T}_L$ (see Definition 7.4), the matrices to which Algorithm 7.1 is applied to in the ℓth level of recursion are denoted by

$$A_j^\ell := A(I_j^\ell, I_j^\ell)$$

for $\ell < L$. Analogously, we let G_j^ℓ denote the update of the form UXV^T computed in line 13. We aim at proving the following bound for the error of Algorithm 7.1:

$$\|f(A) - F_A\|_2 \leq \sum_{\ell=0}^{L-1} \max_{j=1,\ldots,2^\ell} \left\| f(A_j^\ell) - \begin{bmatrix} f(A_{2j-1}^{\ell+1}) & \\ & f(A_{2j}^{\ell+1}) \end{bmatrix} - G_j^\ell \right\|_2 . \tag{7.2}$$

This bound implies the statement of the theorem because by Theorem 6.5 each term

appearing in the sum can be bounded by

$$\left\| f(A_j^\ell) - \begin{bmatrix} f(A_{2j-1}^{\ell+1}) & \\ & f(A_{2j}^{\ell+1}) \end{bmatrix} - G_j^\ell \right\|_2 \leq 4 \min_{r \in \Pi_m/q_m} \|f - r\|_{\mathbb{D}},$$

where we used that the eigenvalues of principal submatrices of A are contained in $\mathbb{D}$.

The proof of (7.2) is by induction on L, the number of levels. When $L = 1$, the definition of F_A yields

$$\|f(A) - F_A\|_2 = \left\| f(A_1^0) - \begin{bmatrix} f(A_1^1) & \\ & f(A_2^1) \end{bmatrix} - G_1^0 \right\|_2 .$$

Now, suppose that (7.2) holds for $L-1$. Then the result for $L \geq 2$ is proven by observing

$$\begin{aligned}
\|f(A) - F_A\|_2 &= \left\| f(A_1^0) - \left(\begin{bmatrix} F_{A_1^1} & \\ & F_{A_2^1} \end{bmatrix} + G_1^0 \right) \right\|_2 \\
&= \left\| f(A_1^0) - \begin{bmatrix} f(A_1^1) & \\ & f(A_2^1) \end{bmatrix} - G_1^0 + \begin{bmatrix} f(A_1^1) & \\ & f(A_2^1) \end{bmatrix} - \begin{bmatrix} F_{A_1^1} & \\ & F_{A_2^1} \end{bmatrix} \right\|_2 \\
&\leq \left\| f(A_1^0) - \begin{bmatrix} f(A_1^1) & \\ & f(A_2^1) \end{bmatrix} - G_1^0 \right\|_2 + \left\| \begin{bmatrix} f(A_1^1) - F_{A_1^1} & \\ & f(A_2^1) - F_{A_2^1} \end{bmatrix} \right\|_2 \\
&= \left\| f(A_1^0) - \begin{bmatrix} f(A_1^1) & \\ & f(A_2^1) \end{bmatrix} - G_1^0 \right\|_2 + \max_{k \in \{1,2\}} \|f(A_k^1) - F_{A_k^1}\|_2 .
\end{aligned}$$

Each of the terms $\|f(A_k^1) - F_{A_k^1}\|_2$ corresponds to applying Algorithm 7.1 with a cluster tree of depth $L-1$, for which (7.2) holds by the induction assumption; therefore, (7.2) also holds for L. □

Corollary 7.6. *Under the assumptions of Theorem 7.5, when using polynomial Krylov subspaces in Algorithm 6.1, we have that*

$$|\mathrm{trace}(f(A)) - \mathrm{trace}(F_A)| \leq 4nL \min_{p \in \Pi_{2m}} \|f - p\|_{\mathbb{D}}.$$

Proof. Analogously to the proof of Theorem 7.5, we can bound

$$
\begin{aligned}
&|\mathrm{trace}(f(A)) - \mathrm{trace}(F_A)| \\
&\qquad \leq \sum_{\ell=0}^{L-1} \sum_{j=1}^{2^\ell} \left| \mathrm{trace}(f(A_j^\ell)) - \mathrm{trace} \begin{bmatrix} f(A_{2j-1}^\ell) & \\ & f(A_{2j}^\ell) \end{bmatrix} - \mathrm{trace}(G_j^\ell) \right|
\end{aligned}
$$

and use Theorem 6.12 to conclude. □

7.2 Numerical tests for Algorithm 7.1

In this section we test Algorithm 7.1 on a variety of matrices and functions coming from different applications. The minimum block size parameter $n_{\min}$ is set to 256 for all our experiments, and the tolerance is $\varepsilon = 10^{-8}$ for all experiments, unless otherwise noted. The lag parameter in Algorithm 6.1 is set to $d = 1$. The algorithm has been implemented in Matlab, version 9.9 (R2020b) and the code for reproducing the experiments in this chapter is available at https://github.com/Alice94/MatrixFunctions-Banded-HSS. The computations with HSS matrices have been performed using the hm-toolbox [137]. This requires choosing a minimum block size and a tolerance parameter, which we set to be equal to $n_{\min}$ and ε, respectively.

In all tables referring to the computation of matrix functions $f(A)$ the columns denoted by "Err" contain the relative error in the Frobenius norm computed with respect to the value of $f(A)$ obtained by dense arithmetic, whenever the size of the matrix allows for computations in dense arithmetic.

7.2.1 Space-fractional diffusion equation without source

Let us consider the fractional diffusion problem:

$$
\begin{cases}
\frac{\partial u(x,t)}{\partial t} = \frac{\partial^\alpha u(x,t)}{\partial_- x^\alpha} + \frac{\partial^\alpha u(x,t)}{\partial_+ x^\alpha} & (x,t) \in (0,1) \times (0,T] \\
u(x,t) = 0 & (x,t) \in (\mathbb{R} \setminus [0,1]) \times [0,T] \\
u(x,0) = u_0(x) & x \in [0,1]
\end{cases}
$$

where $\alpha \in (1,2)$ is a fractional order of derivation and $\frac{\partial^\alpha}{\partial_- x^\alpha}, \frac{\partial^\alpha}{\partial_+ x^\alpha}$ denote the left-looking and right-looking αth derivatives. Discretizing in space by means of the finite difference scheme based on Grünwald-Letnikov formulas, with step size $\Delta x = \frac{1}{n+1}$, yields

$$\begin{cases} \dot{\mathbf{u}}(t) = A\mathbf{u}(t) \\ \mathbf{u}(0) = \mathbf{u_0} \end{cases} \qquad A = T_n + T_n^T, \quad T_n = \frac{1}{\Delta x^\alpha} \begin{bmatrix} g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \dots & 0 & 0 \\ g_2^{(\alpha)} & g_1^{(\alpha)} & g_0^{(\alpha)} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ g_{n-1}^{(\alpha)} & \ddots & \ddots & \ddots & g_1^{(\alpha)} & g_0^{(\alpha)} \\ g_n^{(\alpha)} & g_{n-1}^{(\alpha)} & \dots & \dots & g_2^{(\alpha)} & g_1^{(\alpha)} \end{bmatrix},$$

where

$$g_0^{(\alpha)} = -1, \qquad g_k^{(\alpha)} = \frac{(-1)^{k+1}}{k!}\alpha(\alpha-1)\cdots(\alpha-k+1), \quad k = 1,\dots,n,$$

and $\mathbf{u}(t), \mathbf{u_0} \in \mathbb{R}^n$ contain the sampling of the solution and of the boundary condition, respectively, at the spatial points $j\Delta x$, for $j = 1,\dots,n$. In particular, evaluating the solution at time $t = 1$ as $\mathbf{u}(1) = e^A\mathbf{u_0}$ requires the computation of the matrix exponential of A which is well approximated in the HSS format [135].

Concerning the latter task, we compare the performances of our D&C method (Algorithm 7.1) with polynomial Krylov subspaces and of the function `expm` of the `hm-toolbox` that makes use of a Padé approximant combined with scaling and squaring.

The results are reported in Table 7.1. The column labeled as "Dense" corresponds to the evaluation of the matrix exponential with dense arithmetic via Matlab's `expm` function. This has been computed up to size $n = 8192$ and demonstrates that D&C is slightly more accurate; `expm` (HSS) and D&C are cheaper than the dense method from sizes 4096 and 2048, respectively.

7.2.2 Sampling from a Gaussian Markov random field

This case study, taken from [115], arises from computational statistics and it concerns a tool often used to model spatially structured uncertainty in the data. Given a cloud of

A		D&C		`expm` (HSS)		Dense	e^A
Size	HSS rank	Time	Err	Time	Err	Time	HSS rank
512	10	0.09	$1.89 \cdot 10^{-8}$	0.57	$3.78 \cdot 10^{-8}$	**0.03**	13
1,024	11	0.18	$2.34 \cdot 10^{-8}$	1.01	$6.75 \cdot 10^{-8}$	**0.15**	15
2,048	13	**0.38**	$4 \cdot 10^{-8}$	1.77	$9.88 \cdot 10^{-8}$	0.96	23
4,096	14	**1.09**	$4.85 \cdot 10^{-8}$	3.99	$1.44 \cdot 10^{-7}$	8.94	23
8,192	15	**3.11**	$6.09 \cdot 10^{-8}$	10.22	$1.16 \cdot 10^{-7}$	70.75	25
16,384	15	**8.61**		18.48			26
32,768	16	**22.18**		37.22			27

Table 7.1 – Computation of e^A in the HSS format for the coefficient matrix A of the fractional diffusion problem discussed in Section 7.2.1. We compare the performances of the `expm` *function of the* `hm-toolbox` *[137] and of the D&C approach proposed in Algorithm 7.1.*

points $\{s_i\}_{i=1}^n \subset \mathbb{R}^d$ we introduce Gaussian random variables x_i for $i = 1, \dots, n$ at each point. The vector $x = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^T$ is referred to as a *Gaussian Markov random field (GMRF)* when it is distributed according to the precision (inverse covariance) matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ depending on two positive parameters ϕ and δ as follows:

$$a_{ij} = \begin{cases} 1 + \phi \cdot \sum\limits_{k=1, k \neq i}^{n} \chi_{ki}^{\delta} & \text{if } i = j \\ -\phi \cdot \chi_{ij}^{\delta} & \text{if } i \neq j \end{cases} \qquad \text{where } \chi_{ij}^{\delta} = \begin{cases} 1 & \text{if } ||s_i - s_j||_2 < \delta, \\ 0 & \text{otherwise.} \end{cases}$$

A sample $v \in \mathbb{R}^n$ from a zero-mean GMRF with precision matrix A is obtained as $v = A^{-\frac{1}{2}} z$, where z is a vector of independently and identically distributed standard normal random variables.

When many samples are needed, it is convenient to store an HSS representation of $A^{-\frac{1}{2}}$ so that each sample requires only a matrix vector product with an HSS matrix. In this experiment we set $\phi = 3$, we generate $n = 2^j$ pseudo-random points s_i in the unit interval $(0, 1)$, and we choose $\delta = 0.02 \cdot 2^{9-j}$ for $j = 9, \dots, 18$. Sorting the points s_i yields precision matrices that are symmetric, diagonally dominant and with bandwidth in the range $[19, 26]$.

As suggested in Remark 7.2, as the matrix A is banded and SPD we use a decomposition which features rank-b updates; we observed a speed up with respect to doing rank-$2b$

updates in our experiments. In Algorithm 6.1 the projection method used for computing the updates in the D&C is the *Extended Krylov method*, which alternates poles 0 and ∞. More precisely, the mth extended Krylov subspace associated to a matrix A and a (block) vector B is

$$A^{-m}\mathcal{K}_{2m}(A,B) := \operatorname{span}\Big[B, A^{-1}B, AB, \ldots, A^{m-1}B, A^{-m}B\Big].$$

We compare the computation of $A^{-\frac{1}{2}}$ in the HSS format by means of our D&C scheme with the function `sqrtm` contained in the `hm-toolbox` [137] which combines the Denman and Beavers iteration (see, e.g., [110, Section 6.3]) with the HSS arithmetic.

The results reported in Table 7.2 show that the D&C approach yields a significant reduction of the computational time with respect to `sqrtm` (HSS). For the largest instance, $n = 32,768$, we have profiled the computing time spent at the different stages of the D&C method. The generation of the bases of the extended Krylov subspaces consumed about 25% of the total time while about 50% was spent to sum the (low-rank) updates to the block diagonal intermediate results. Around 20% was used for computing the projected matrices and evaluating the inverse square roots of the diagonal blocks at the lowest level of recursion and of the projected matrices.

7.2.3 Merton model for option pricing

We consider the evaluation of option prices in the Merton model for one single underlying asset, as in [123, Section 6.3]. More specifically, we compute the exponential of the non-symmetric Toeplitz matrix A arising from the discretization of the partial integro-differential equation

$$\omega_t = \frac{\nu^2}{2}\omega_{\xi\xi} + \left(r - \lambda\kappa - \frac{\nu^2}{2}\right)\omega_\xi - (r+\lambda)\omega + \lambda\int_{-\infty}^{+\infty}\omega(\xi+\eta,t)\phi(\eta)\mathrm{d}\eta,$$

where $\omega(\xi,t)$ on $(-\infty,+\infty)\times[0,T]$ is the option value, T is the time to maturity, $\nu \geq 0$ is the volatility, r is the risk-free interest rate, $\lambda \geq 0$ is the arrival intensity of a Poisson process, ϕ is the normal distribution with mean μ and standard deviation σ, and $\kappa = e^{\mu+\sigma^2/2} - 1$. We use the same discretization and parameters as [123, Section 6.3] and [128, Example 3].

We aim at approximating $\exp(A)$, for different values of the matrix size n. To do so,

A		D&C		`sqrtm` (HSS)		Dense	$A^{-\frac{1}{2}}$
Size	Band	Time	Err	Time	Err	Time	HSS rank
512	22	0.05	$2.02 \cdot 10^{-9}$	0.49	$3.44 \cdot 10^{-9}$	**0.02**	14
1,024	20	0.16	$3.45 \cdot 10^{-9}$	1.41	$5.22 \cdot 10^{-9}$	**0.13**	17
2,048	19	**0.37**	$3.76 \cdot 10^{-9}$	3.99	$6.38 \cdot 10^{-9}$	0.95	19
4,096	21	**0.8**	$3.23 \cdot 10^{-9}$	9.05	$5.61 \cdot 10^{-9}$	9.03	19
8,192	22	**2.46**	$3.46 \cdot 10^{-9}$	21.27	$6.61 \cdot 10^{-9}$	70.42	20
16,384	25	**5.7**		48.92			22
32,768	26	**15.12**		102.65			25
65,536	26	**26.25**		209.56			24
131,070	25	**60.44**		417.21			24
262,140	26	**146.97**		918.81			26

Table 7.2 – Computation of $A^{-\frac{1}{2}}$ in the HSS format for the precision matrix A of the Gaussian Markov random field discussed in Section 7.2.2. We compare the performances of the `sqrtm` *function of the* `hm-toolbox` *[137] and of the D&C approach proposed in Algorithm 7.1.*

we first compute an HSS approximation H of A using the hm-toolbox [137], then rescale it by dividing by $2^{\lceil \log_2 \|H\|_2 \rceil}$, then we apply Algorithm 7.1, and finally we square the result $\lceil \log_2 \|H\|_2 \rceil$ times in the HSS format. We use polynomial Krylov subspaces for the updates in Algorithm 7.1. For different values of n, we compare the output of the described method with the `expm` algorithm from the hm-toolbox [137] and the algorithm `sexpmt` proposed in [123]. In order to attain a similar accuracy to the `sexpmt` algorithm, we set the tolerance parameter $\varepsilon = 10^{-12}$ in Algorithm 7.1 and for HSS computations in the hm-toolbox [137]. The results are summarized in Table 7.3.

7.2.4 Neumann-to-Dirichlet operator

Consider

$$\frac{\partial^2}{\partial x^2} u = Au, \qquad \frac{\partial}{\partial x} u \mid_{x=0} = -b, \qquad u \mid_{x=+\infty} = 0 \tag{7.3}$$

for a non-singular matrix A which is the discretization of a differential operator on some spatial domain $\Omega \subseteq \mathbb{R}^\ell$. Then (7.3) is a semidiscretization of an $(\ell+1)$-dimensional PDE on $[0, +\infty) \times \Omega$; the solution is given by $u(x) = \exp\left(-xA^{-1/2}\right) A^{-1/2} b$. In particular, $u(0) = A^{-1/2} b$ and the operator $A^{-1/2}$ is called *Neumann-to-Dirichlet (NtD)* operator as

A	D&C		`expm` (HSS)		`sexpmt`		Dense	e^A
Size	Time	Err	Time	Err	Time	Err	Time	HSS rank
512	0.49	$2.66 \cdot 10^{-11}$	0.57	$2.34 \cdot 10^{-10}$	0.16	$2.95 \cdot 10^{-12}$	**0.1**	18
1,024	0.86	$7.13 \cdot 10^{-10}$	1.82	$5.99 \cdot 10^{-10}$	**0.54**	$2.04 \cdot 10^{-11}$	0.69	18
2,048	2.4	$1.18 \cdot 10^{-9}$	4.16	$8.03 \cdot 10^{-9}$	**1.31**	$3.37 \cdot 10^{-11}$	5.05	17
4,096	**5.53**	$6.19 \cdot 10^{-8}$	8.06	$2.96 \cdot 10^{-8}$	7.39	$1.86 \cdot 10^{-10}$	42.33	19
8,192	**9.6**	$5.65 \cdot 10^{-7}$	16.23	$3.1 \cdot 10^{-7}$	25.52	$1.35 \cdot 10^{-9}$	323.18	18
16,384	**20.58**		33.71		98.41			20
32,768	**46.13**		67.43		419.82			20

Table 7.3 – Computation of e^A in the HSS format for the coefficient matrix A in Section 7.2.3. We compare the performances of our Algorithm 7.1 with the `expm` function of the `hm-toolbox` [137] and the `sexpmt` algorithm of [123].

it allows for conversion of the Neumann data $-b$ at the boundary $x = 0$ into the Dirichlet data $u(0)$, without needing to solve (7.3) on its unbounded domain.

As in [65, Example 6.1], we consider the inhomogeneous Helmholtz equation

$$\Delta u(x, y) + k^2 u(x, y) = f(x, y), \quad f(x, y) = 10\delta(x - 511\pi/512)\delta(y - 50\pi/512)$$

for $k = 50$ on the domain $[0, \pi]^2$. The matrix A corresponds to the discretization of $-\frac{\partial^2}{\partial y^2} - k^2$ on $[0, \pi]$ by central finite differences. We consider step sizes $h \in \{\pi/2^9, \ldots, \pi/2^{15}\}$ and compute the NtD operator $A^{-1/2}$ in the HSS format using the D&C algorithm 7.1; Table 7.4 illustrates the comparison with the computation of $A^{-1/2}$ in dense arithmetic. For computing the inverse square root, we move the branch cut to the negative imaginary axis, that is, when expressing $z = \rho \exp(i\theta)$ for $\theta \in \left[-\frac{\pi}{2}, \frac{3\pi}{2}\right]$ we define $f(z) := \rho^{-1/2} \exp\left(-i\frac{\theta}{2}\right)$. This avoids a discontinuity on the negative real axis, where some of the eigenvalues of A lie. For the updates, we use the complex extension of Algorithm 6.1 with rational Krylov subspaces where we cyclically repeat 6 poles coming from the degree-6 approximation to $f(z) = z^{-1/2}$ on the set $S := [-b, -a] \cup [a, b]$ for $b = \|A\|_2$ (estimated with `normest(A)`) and $a = 1/\|A^{-1}\|_2$ (computed via `b / condest(A)`) described in [65, Section 2]. As the spectral interval of A contains zero, which is a singularity of the inverse square root function, the assumptions of Theorem 7.5 are not satisfied. When applying Algorithm 6.1 for the low-rank updates, the projected matrix in line 6 in practice could be almost singular, leading to instabilities in the computation of its inverse square root; however, this does not happen in our example.

A	D&C		Dense	$A^{-1/2}$
Size	Time	Err	Time	HSS rank
512	1.18	$2.53 \cdot 10^{-8}$	**0.13**	12
1,024	0.99	$3.59 \cdot 10^{-8}$	**0.21**	13
2,048	1.84	$4.26 \cdot 10^{-8}$	**0.92**	20
4,096	**3.23**	$6.39 \cdot 10^{-8}$	6.73	20
8,192	**7.89**	$8.2 \cdot 10^{-8}$	64.74	20
16,384	**16.75**			20
32,768	**37.7**			20

Table 7.4 – Computation of $A^{-1/2}$ in the HSS format for Neumann-to-Dirichlet problem discussed in Section 7.2.4.

7.2.5 Computing charge densities

The approximation of the diagonal of a matrix function applies to the calculation of the electronic structure of systems of atoms. In particular, the charge densities of a system are contained in the diagonal of $f(H)$, where f is the Heaviside function

$$f(x) = \begin{cases} 1 & x < 0 \\ 0 & x \geq 0 \end{cases}$$

and H is the Hamiltonian matrix that is given by the sum of the kinetic and potential energies. The entries of Hamiltonian matrices usually decay rapidly away from the main diagonal. Let us consider the parametrized model Hamiltonian given in [19, Section 4.3]:

$$H \in \mathbb{R}^{N_b \cdot N_s \times N_b \cdot N_s}, \qquad H_{N_b \cdot (i-1)+j, i' \cdot N_b (i'-1)+j'} = \begin{cases} (i-1)\Delta + (j-1)\delta & i = i',\ j = j' \\ C \cdot e^{-|j-j'|} & i = i',\ j \neq j' \\ \frac{C}{n_{od}(|i-i'|+1)} \cdot e^{-|j-j'|} & \text{otherwise} \end{cases},$$

where we have set the parameters' values: $N_b = 5, N_s = 1600, \Delta = 10^{-1}, \delta = 10^{-4}, C = 10^{-1}$, and $n_{od} = 5000$. The HSS structure of the matrix H is shown in the left part of Figure 7.2. We compute the diagonal of $f(H)$ by means of Algorithm 7.1 and exploiting the relation

$$f(x) = (1 - \text{sign}(x))/2.$$

More specifically, we use Algorithm 7.1 to compute the diagonal of $\mathrm{sign}(H)$; then we subtract the latter from the vector of all ones and we divide by 2. The procedure has terminated after 3.52 seconds. As benchmark method we evaluate $f(H)$ by diagonalization with dense arithmetic. This has required 78.62 seconds. The Euclidean distance of the vectors obtained with the two approaches is $2.68 \cdot 10^{-11}$. In Figure 7.2, the first 500 components of the two charge densities are shown.

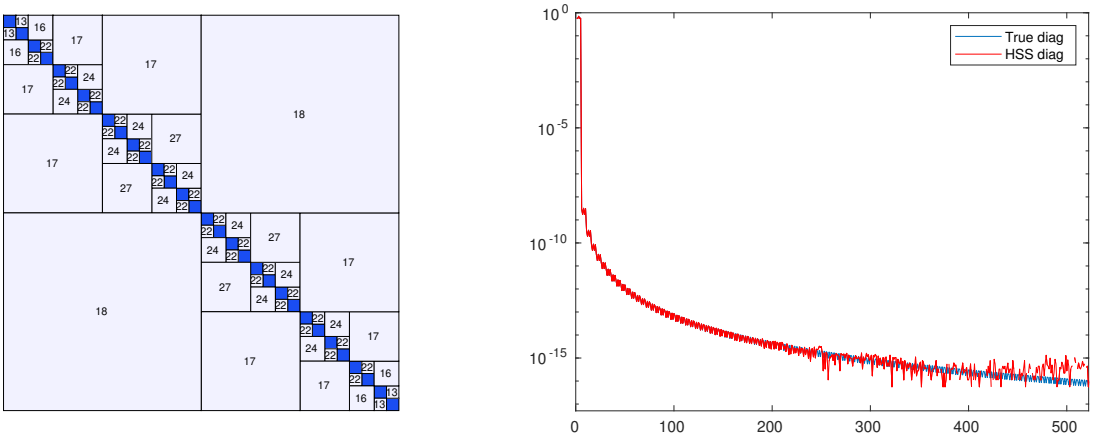


Figure 7.2 – Left: Ranks of the off-diagonal blocks of the Hamiltonian matrix H from Section 7.2.5; the blue blocks indicate matrices for which dense arithmetics is used. Right: Charge densities estimated with dense arithmetic (blue) and with the HSS D&C method (red).

7.2.6 Computing subgraph centralities and Estrada index

Given an undirected graph $\mathcal{G}$ with adjacency matrix A, the diagonal entries of $\exp(A)$ are called the *subgraph centralities* of the vertices. Their normalized sum

$$EE_n(\mathcal{G}) := \frac{1}{n}\,\mathrm{tr}(\exp(A))$$

is called the normalized Estrada index of the graph; it was introduced in [69] to characterize the folding of molecular structures and has found applications in network analysis [70].

When aiming at the diagonal of $\exp(A)$, as the baseline method we use the `mmq` algorithm [85], which approximates each diagonal entry of $\exp(A)$ by Gauss quadrature (see also Section 9.4 in Chapter 9). For our D&C method, at each step we run a clustering algorithm [118] on the matrix to divide it into two components that have few edges

between them; the `ufactor` parameter, which measures how "unbalanced" the clusters are allowed to be, is set to 100. If the rank of the off-diagonal part is less than 1/15 of the matrix size, we compute a low-rank update, otherwise we use `mmq`. We also use `mmq` on matrices of size less than $n_{\min} = 256$. We compare our D&C algorithm for the diagonal with `mmq` and `diag(expm(full(A)))`.

When aiming at trace($\exp(A)$), we use `sum(exp(eig(full(A))))` instead of `mmq` to address small blocks or blocks that cannot be divided into smaller blocks with a low-rank correction; we noticed that this is faster than letting Matlab work with the matrices in sparse format. As a competitor for the computation of the trace we consider `sum(exp(eig(full(A))))`.

In Table 7.5 we report the errors and the time needed by our algorithm. The matrices we used are `minnesota`, `power`, `as-735`, `nopoly`, `worms20_10NN`, and `fe_body` from the SuiteSparse Matrix Collection [52].

A	D&C diagonal		mmq diagonal		`expm`	D&C trace		`eig`
Size	Time	Err	Time	Err	Time	Time	Err	Time
2,642	1.01	$6.24 \cdot 10^{-10}$	**0.8**	$1.82 \cdot 10^{-11}$	1.98	**0.14**	$7.71 \cdot 10^{-13}$	0.44
4,941	**2.06**	$1.29 \cdot 10^{-8}$	5.15	$3.39 \cdot 10^{-11}$	16.11	**0.47**	$7.75 \cdot 10^{-11}$	3.61
7,716	**8.01**	$4.03 \cdot 10^{-9}$	24.19	$2.29 \cdot 10^{-10}$	56.59	**3.91**	$1.96 \cdot 10^{-12}$	8.73
10,774	**15.87**	$1.04 \cdot 10^{-8}$	39.42	$3.54 \cdot 10^{-10}$	151.52	**2.98**	$2.69 \cdot 10^{-10}$	21.04
20,055	**38.49**	$2.59 \cdot 10^{-9}$	97.53	$1.4 \cdot 10^{-11}$	929.25	**6.99**	$2.66 \cdot 10^{-13}$	124.34
45,087	**182.19**		603.57			**27.99**		

Table 7.5 – Computation of the diagonal and the trace of e^A for the graphs from Section 7.2.6.

The lag parameter

We compare the timings and the accuracy of our D&C algorithm on the matrices `nopoly` and `worms20_10NN` for values of the lag parameter in the range $\{1, 2, 3, 4\}$. The results are reported in Table 7.6. In general, it looks like we can safely put the lag parameter equal to 1.

	nopoly				worms20_10NN			
Lag	Diag	Trace	Err diag	Err trace	Diag	Trace	Err diag	Err trace
1	19.44	**2.81**	$1.04 \cdot 10^{-8}$	$2.69 \cdot 10^{-10}$	**47.82**	**6.37**	$2.59 \cdot 10^{-9}$	$2.6 \cdot 10^{-13}$
2	16.54	3.13	$1 \cdot 10^{-8}$	$3.78 \cdot 10^{-12}$	61.76	12.04	$2.46 \cdot 10^{-9}$	$5.19 \cdot 10^{-16}$
3	**14.63**	4	$9.29 \cdot 10^{-9}$	$2.41 \cdot 10^{-14}$	87.85	11.55	$2.34 \cdot 10^{-9}$	$5.19 \cdot 10^{-16}$
4	16.08	3.76	$8.51 \cdot 10^{-9}$	$1.42 \cdot 10^{-14}$	89.01	20.86	$2.16 \cdot 10^{-9}$	$1.73 \cdot 10^{-16}$

Table 7.6 – For two matrices from [52] we investigate the influence of the lag parameter on the timing of the D&C algorithm for computing the diagonal and the trace of $\exp(A)$.

7.3 Block diagonal splitting algorithm for banded matrices

As already mentioned in Remark 7.3 and shown in more detail below, Algorithm 7.1 applied to a banded matrix returns again a banded matrix when polynomial Krylov subspace bases are used. The purpose of this section is to go further and use this observation to bypass the need for building Krylov subspaces. We can also avoid recursion and arrive at a simpler algorithm.

7.3.1 Block diagonal splitting algorithm from low-rank updates

Let $A \in \mathbb{R}^{n\times n}$ be banded with bandwidth b. Our algorithm will be based on splitting A into a block-diagonal matrix with many small diagonal blocks and an off-diagonal part. To explain this construction, we will first discuss splitting off one small diagonal block. We consider the partitioning

$$A = D + R, \quad D = \begin{bmatrix} D_1 & \\ & \widetilde{D_1} \end{bmatrix}, \quad D_1 \in \mathbb{R}^{s\times s}, \quad R = A - D, \tag{7.4}$$

but we now suppose that the first diagonal block is small, that is, $s \ll n$; see also Figure 7.3.

The matrix R can be written as $R = U_1 J U_1^T$ where

$$U_1 := \begin{bmatrix} \underbrace{0}_{s-b} & \underbrace{I_{2b}}_{2b} & \underbrace{0}_{n-s-b} \end{bmatrix}^T$$

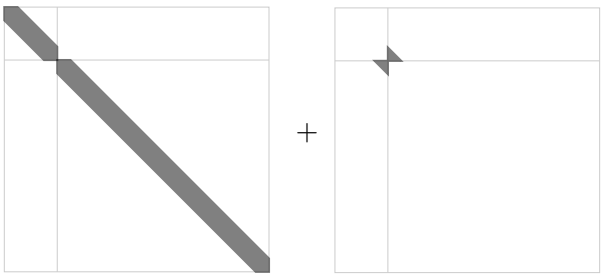
$A = D + R =$ 

Figure 7.3 – Illustration of decomposition (7.4).

and

$$J := \begin{bmatrix} & A(s-b+1:s, s+1:s+b) \\ A(s+1:s+b, s-b+1:s) & \end{bmatrix}.$$

When applying Algorithm 6.1 to approximate the low-rank update $f(A) - f(D)$ the polynomial Krylov subspaces remain sparse in the following sense.

Lemma 7.7. *Given the setting described above, assume that $2mb \leq s$. Then the Krylov subspaces $\mathcal{K}_m(D, U_1)$ and $\mathcal{K}_m(D^T, U_1)$ are each contained in the column span of the $n \times 2mb$ matrix*

$$U_m := \begin{bmatrix} \underbrace{0}_{s-mb} & \underbrace{I_{2mb}}_{2mb} & \underbrace{0}_{n-s-mb} \end{bmatrix}^T.$$

Proof. For every polynomial $p \in \Pi_{m-1}$, the matrix $p(D)$ is banded with bandwidth $(m-1)b$. In turn, $p(D)U_1$ only has nonzero rows at positions $s-mb+1, \ldots, s+mb$ or, in other words, every column of $p(D)U_1$ is contained in the column span of U_m. Combined with the definition $\mathcal{K}_m(D, U_1) = \operatorname{span}[U_1, DU_1, \ldots, D^{m-1}U_1]$, this proves the statement of the lemma. □

The compressions of D and A with respect to the orthonormal basis U_m from Lemma 7.7 takes the form

$$\begin{aligned} G_m &= U_m^T D U_m \\ &= \operatorname{blkdiag}(A(s-mb+1:s, s-mb+1:s), A(s+1:s+mb, s+1:s+mb))) \\ &=: \operatorname{blkdiag}(C_1^{(1)}, C_1^{(2)}), \\ H_m &= U_m^T A U_m = A(s-mb+1:s+mb, s-mb+1:s+mb) =: B_1. \end{aligned}$$

Following Algorithm 6.1, we define the approximate low-rank update as

$$\begin{aligned} f(A) - f(D) &= f(A) - \text{blkdiag}(f(D_1), f(\widetilde{D_1})) \\ &\approx U_m f(B_1) U_m^T - U_m f(\text{blkdiag}(C_1^{(1)}, C_1^{(2)})) U_m^T. \end{aligned} \tag{7.5}$$

By Lemma 7.7, this approximation becomes in fact identical to the one returned by Algorithm 6.1 if $\mathcal{K}_m(D, U_1)$ and $\mathcal{K}_m(D^T, U_1)$ each have dimension $2mb$. If the Krylov subspaces are of smaller dimension then the approximations may differ, but the exactness property of Theorem 6.2 still holds (see Remark 6.3).

7.3.2 The block diagonal splitting algorithm

From (7.5), it follows that the first part of Algorithm 7.1 (lines 12–14) reduces to the computation of $f(B_1)$, $f(C_1^{(1)})$, $f(C_1^{(2)})$, $f(D_1)$, that is, functions of small submatrices of A. For the second part (line 15) one can apply the same reasoning recursively to $\widetilde{D_1}$.

With the simplified assumptions that $n = ks$ for an integer k and $m := \frac{s}{2b}$ is an integer, the discussion above shows that Algorithm 7.1 reduces to the simpler Algorithm 7.2, where

- $D := \text{blkdiag}(D_1, \ldots, D_k)$ and $D_1, \ldots, D_k$ are the consecutive $s \times s$ diagonal blocks of A;
- $\widetilde{B} := \text{blkdiag}(B_1, \ldots, B_{k-1})$ and $B_1, \ldots, B_{k-1}$ are consecutive $s \times s$ diagonal blocks of A starting from index $\frac{s}{2} + 1$;
- $\widetilde{C} := \text{blkdiag}(C_1^{(1)}, C_1^{(2)}, \ldots, C_{k-1}^{(1)}, C_{k-1}^{(2)})$ where $C_1^{(1)}, \ldots, C_{k-1}^{(2)}$ are the consecutive $\frac{s}{2} \times \frac{s}{2}$ diagonal blocks of A starting from index $\frac{s}{2} + 1$;
- $B := \text{blkdiag}(Z, \widetilde{B}, Z)$, $C := \text{blkdiag}(Z, \widetilde{C}, Z)$, where $Z := \texttt{zeros}(\frac{s}{2})$.

The resulting splitting $A = D + B - C$ is illustrated in Figure 7.4. Note that Algorithm 7.2 is embarrassingly parallel and attains nearly perfect weak scalability on k processors.

Algorithm 7.2 Approximation of $f(A)$ for banded A

Input: Banded matrix $A \in \mathbb{R}^{n\times n}$ of bandwidth b, block size s, function f
Output: Approximation $\mathrm{approx}_f^{(s)}(A)$ of $f(A)$

1: Define $\widetilde{B}$, B, $\widetilde{C}$, C, and split $A = D + B - C$ as explained in Section 7.3.2
2: Compute $f(D)$, $f(\widetilde{B})$, and $f(\widetilde{C})$ by evaluating f on each block of D, $\widetilde{B}$, and $\widetilde{C}$
3: Set $f_B \leftarrow \mathrm{blkdiag}(Z, f(\widetilde{B}), Z)$ and $f_C \leftarrow \mathrm{blkdiag}(Z, f(\widetilde{C}), Z)$, where $Z := \texttt{zeros}(\frac{s}{2})$
4: Return $f(D) + f_B - f_C$

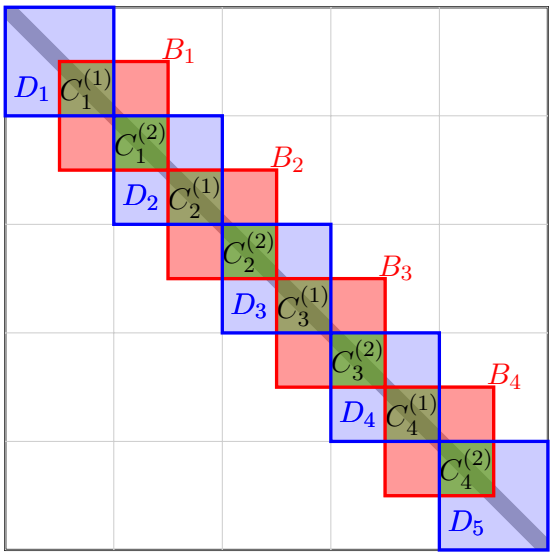


Figure 7.4 – The blocks that are involved in the computation of $f(A)$ for a banded matrix A.

7.3.3 Convergence analysis of block diagonal splitting method

Algorithm 7.2 corresponds to Algorithm 7.1 where the updates are performed using projection onto spaces that *include* polynomial Krylov subspaces of dimension $m := \lfloor \frac{s}{2b} \rfloor$; thanks to Remark 6.3 and Proposition 7.1 this implies that Algorithm 7.2 is exact for all $f \in \Pi_m$. This property allows us to prove convergence results for Algorithm 7.2.

Theorem 7.8. *Let $A \in \mathbb{R}^{n\times n}$ be a banded matrix with bandwidth b. For a given block size s, the output $\mathrm{approx}_f^{(s)}(A)$ of Algorithm 7.2 satisfies*

$$\|f(A) - \mathrm{approx}_f^{(s)}(A)\|_2 \leq 4C \min_{p\in\Pi_m} \|f - p\|_{W(A)},$$

where $C = 1$ if A is normal and $C = 1 + \sqrt{2}$ otherwise, and $m := \lfloor \frac{s}{2b} \rfloor$.

Proof. Algorithm 7.2 is exact for a polynomial in Π_m and is linear with respect to f, therefore for all $p \in \Pi_m$ we have

$$\begin{aligned}\|f(A) - \mathrm{approx}_f^{(s)}(A)\|_2 &= \|f(A) - p(A) + \mathrm{approx}_p^{(s)}(A) - \mathrm{approx}_f^{(s)}(A)\|_2 \\ &= \|f(A) - p(A) - \mathrm{approx}_{f-p}^{(s)}(A)\|_2 \\ &\leq \|(f-p)(A)\|_2 + \|\mathrm{approx}_{f-p}^{(s)}(A)\|_2.\end{aligned}$$

Using a result by Crouzeix and Palencia [50], we have $\|(f-p)(A)\|_2 \leq C\|f-p\|_{W(A)}$. Since the spectral norm of a block-diagonal matrix is the maximum spectral norm of its blocks, it holds that

$$\begin{aligned}\|\mathrm{approx}_{f-p}^{(s)}(A)\|_2 &\leq \max_i \|(f-p)(D_i)\|_2 + \max_i \|(f-p)(B_i)\|_2 + \max_{i,j} \|(f-p)(C_i^{(j)})\|_2 \\ &\leq 3C\|(f-p)\|_{W(A)}.\end{aligned}$$

In the latter inequality, we used again [50] combined with the fact that the numerical range of a principal submatrix of A is contained in $W(A)$. We conclude that

$$\|f(A) - \mathrm{approx}_f^{(s)}(A)\|_2 \leq 4C\|(f-p)\|_{W(A)},$$

and the claim follows from taking the minimum over all polynomials $p \in \Pi_m$. □

When considering the approximation of the *trace* of a matrix function by Algorithm 7.2, a stronger convergence result could be proved, because of the exactness of the low-rank updates (and therefore of the D&C algorithm) for polynomials in Π_{2m}. In the specific case of Algorithm 7.2, however, we can prove a stronger result even for the diagonal entries of $f(A)$.

Theorem 7.9. *Let $A \in \mathbb{R}^{n\times n}$ with bandwidth b, let us fix a block size s, let $m := \lfloor \frac{s}{2b} \rfloor$. Then the output* $\mathrm{approx}_p^{(s)}(A)$ *of Algorithm 7.2 satisfies*

$$\mathrm{diag}(p(A)) = \mathrm{diag}(\mathrm{approx}_p^{(s)}(A)) \tag{7.6}$$

for all polynomials $p \in \Pi_{2m+1}$.

Proof. The proof is in the spirit of [159, Lemma 5.1], but the aim is different. By linearity of Algorithm 7.2, it is sufficient to prove (7.6) when $p(x) = x^k$, with $0 \leq k \leq 2m+1$, that

is, to prove that the diagonal entries of A^k and $\text{approx}_p^{(s)}(A)$ coincide.

To study the entries of A^k, it is helpful to consider the associated directed graph $\mathcal{G}(A)$ with vertices $1, \ldots, n$ and adjacency matrix A. The jth diagonal entry of A^k is given by the sum of the weights of all the paths of length exactly k that start and end at vertex i; we recall that the weight of a path $v_1 \to v_2 \to \ldots \to v_k$ of length k is defined as the product of the weights of the edges $\prod_{h=1}^{k-1} A_{v_h v_{h+1}}$. We also consider the graphs $\mathcal{G}(B_i)$, $\mathcal{G}(D_i)$, $\mathcal{G}(C_i^{(1,2)})$. The diagonal entries of $\text{approx}_p^{(s)}(A)$ are obtained by summing the weights of the paths of length exactly k in the graphs $\mathcal{G}(D_i)$ and $\mathcal{G}(B_i)$ and subtracting the weights of the paths of length exactly k in the graphs $\mathcal{G}(C_i^{(1)})$ and $\mathcal{G}(C_i^{(2)})$ for all indices i. Therefore, it is sufficient to prove that this sum coincides with the sum of the weights of the paths of length exactly k in $\mathcal{G}(A)$.

Note that, for all indices i, $\mathcal{G}(C_i^{(1)})$ is a subgraph of $\mathcal{G}(D_i)$ and $\mathcal{G}(B_i)$; $\mathcal{G}(C_i^{(2)})$ is a subgraph of $\mathcal{G}(D_{i+1})$ and $\mathcal{G}(B_i)$; all these are subgraphs of $\mathcal{G}(A)$. The distance from a vertex in $\mathcal{G}(D_i)$ and one in $\mathcal{G}(B_{i+1})$ or $\mathcal{G}(B_{i-2})$ is at least $m+1$. Therefore, for each vertex $v \in \{1, \ldots, n\}$ each path in $\mathcal{G}(A)$ of length at most $2m+1$ from v to itself satisfies one (and only one) of the following conditions for some $i \in \{1, \ldots, \frac{n}{s} - 1\}$:

1. It is contained in $\mathcal{G}(C_i^{(1)})$, $\mathcal{G}(B_i)$, and $\mathcal{G}(D_i)$, but in no other subgraph.
2. It is contained in $\mathcal{G}(C_i^{(2)})$, $\mathcal{G}(B_i)$, and $\mathcal{G}(D_{i+1})$, but in no other subgraph.
3. It is contained in $\mathcal{G}(B_i)$ but in no other subgraph.
4. It is contained in $\mathcal{G}(D_i)$ but in no other subgraph.

In all these four cases, the weight of the path is counted exactly once in $\text{approx}_p^{(s)}(A)$; we conclude that the diagonal entries of $\text{approx}_p^{(s)}(A)$ coincide with the ones of $p(A)$ for $p(x) = x^k$ for $k \leq 2m+1$ and therefore for all polynomials in Π_{2m+1}. □

A convergence result for the diagonal elements of the output of Algorithm 7.2 follows from Theorem 7.9 similarly to Theorem 7.8.

Corollary 7.10. *With the same assumptions of Theorem 7.8 it holds that*

$$|f(A)_{ii} - \text{approx}_f^{(s)}(A)_{ii}| \leq 4C \min_{p \in \Pi_{2m+1}} \|f - p\|_{W(A)}$$

for all $i = 1, \dots, n$ *and therefore*

$$|\operatorname{tr}(f(A)) - \operatorname{tr}(\operatorname{approx}_f^{(s)}(A))| \leq 4Cn \min_{p \in \Pi_{2m+1}} \|f - p\|_{W(A)},$$

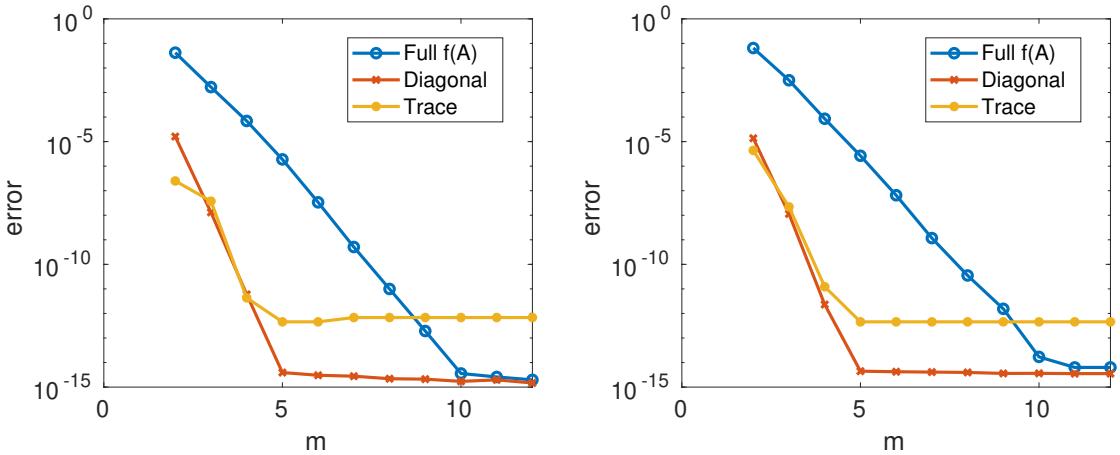
where $C = 1$ *for normal matrices* A*, and* $C = 1 + \sqrt{2}$ *otherwise.*

Proof. According to Theorem 7.9, for all polynomials $p \in \Pi_{2m+1}$ we have that

$$|f(A)_{ii} - \operatorname{approx}_f^{(s)}(A)_{ii}| = |(f-p)(A)_{ii} - \operatorname{approx}_{f-p}^{(s)}(A)_{ii}| \leq \|(f-p)(A) - \operatorname{approx}_{f-p}^{(s)}(A)\|_2.$$

From here one proceeds as in the proof of Theorem 7.8. □

In Figure 7.5 we illustrate the convergence of $\operatorname{approx}_f^{(m)}(A)$ for the exponential of two banded matrices and we observe that the diagonal – and therefore the trace – converges much faster than the full matrix function.



(a) Normalized random symmetric tridiagonal matrix.

(b) Normalized random non-symmetric pentadiagonal matrix.

Figure 7.5 – Convergence of the errors $\|f(A) - \operatorname{approx}_f^{(m)}(A)\|_F$, $\|diag(f(A) - \operatorname{approx}_f^{(m)}(A))\|_2$, *and* $|\operatorname{tr}(f(A) - \operatorname{approx}_f^{(m)}(A))|$ *for* $f = \exp$.

7.3.4 Adaptive algorithm

In Algorithm 7.2 the block size s, which determines the accuracy of the approximation of $f(A)$, needs to be chosen a priori and is uniform across the whole matrix. In the following,

we develop a strategy to choose the block size adaptively and possibly differently in different parts of the matrix.

When f is a polynomial of degree m and A has bandwidth b, $f(A)$ has bandwidth (at most) bm and the discussion in Section 7.3.3 implies that Algorithm 7.3 is exact for block size $s = 2bm$. This motivates the following strategy. For a target accuracy ε, we define the *ε-approximate bandwidth* of a matrix to be the bandwidth that the matrix has if we discard all the entries with absolute value smaller than ε. In the first phase, we choose the sizes of the blocks $D_1, D_2, \ldots, D_k$ in such a way that their sizes are at least twice the ε-approximate bandwidth of $f(D_1), f(D_2), \ldots, f(D_k)$, and we set $F := \mathrm{blkdiag}(f(D_1), \ldots, f(D_k))$. In the second phase we compute the "updates" between each pair of consecutive blocks D_j and D_{j+1} corresponding to indices $\{j_1, \ldots, h\}$ and $\{h+1, \ldots, j_2\}$ of A, respectively, similarly to (7.5). More precisely, we take

$$P := f(B) - \mathrm{blkdiag}(f(C^{(1)}), f(C^{(2)})), \tag{7.7}$$

with

$$B := A(J, J), \quad C^{(1)} := A(J_1, J_1), \quad C^{(2)} := A(J_2, J_2)$$

for $J_1 := \lfloor \frac{j_1+h}{2} \rfloor : h$, $J_2 := (h+1) : \lceil \frac{j_2+h}{2} \rceil$, and $J := J_1 \cup J_2$, and add the matrix P to the submatrix of F corresponding to the indices J. As a heuristic criterion to check convergence, we check if the absolute value of all the entries corresponding to the first and last column and row of P is smaller than ε; if this is not the case, the sets J_1, J_2, and J are enlarged. The procedure is summarized in Algorithm 7.3.

7.4 Numerical tests for Algorithms 7.2 and 7.3

In this section we test the block diagonal splitting algorithm on a variety of functions of banded matrices. Both the input and output matrices of Algorithms 7.2 and 7.3 are represented in the sparse format in Matlab.

7.4.1 Fermi-Dirac density matrix of one-dimensional Anderson model

As a first numerical experiment, we test Algorithm 7.3 on the function

$$f(z) = (\exp(\beta(z - \mu)) + 1)^{-1}$$

Algorithm 7.3 Block diagonal splitting algorithm: Adaptive version

Input: Banded matrix $A \in \mathbb{R}^{n\times n}$, tolerance ε, function f, minimum block size $n_{\min}$
Output: Approximation F of $f(A)$

```
Initialize F ← zeros(n), s ← n_min, i ← 1 (i denotes where the next diagonal block starts)
while i ≤ n do
    if f(A(i : i+s−1, i : i+s−1)) has ε-approximate bandwidth ≤ s/2 then
        Set F(i : i+s−1, i : i+s−1) ← f(A(i : i+s−1, i : i+s−1)), i ← i+s, s ← min{s/2, n_min}
    else
        Choose a larger block size s ← min{2s, n−i+1}
    end if
end while
for each pair of consecutive diagonal blocks do
    Compute P using matrices B, C^(1), C^(2) corr. to indices J, J_1, and J_2 as in (7.7)
    while the update has not converged do
        Enlarge matrices B, C^(1), C^(2) in (7.7) corresp. to indices J, J_1, and J_2, and recompute P
    end while
    Sum F(J, J) ← F(J, J) + P
end for
```

and a symmetric tridiagonal matrix with diagonal entries uniformly randomly distributed in $[0, 1]$ and all other nonzero elements equal to -1, as in [24, Section 5]; this is the Fermi–Dirac density matrix corresponding to a one-dimensional Anderson model. We use $\mu = 0.5$ and $\beta = 1.84$. We set $\varepsilon = 10^{-5}$, $n_{\min} = 32$, and we consider values of n ranging from 2^9 to 2^{19}. For each value of n, we compare the approximation F returned by Algorithm 7.3 to the approximation $p(A)$ where p is a Chebyshev polynomial interpolating f on $[-2, 3]$ of degree $d := \lceil \mathtt{nnz}(F)/(2n) \rceil$; choosing the degree in this way gives a banded approximation of $f(A)$ with roughly the same storage cost and a comparable accuracy. The results are reported in Table 7.7; the approximation errors (relative errors in the Frobenius norm) and the timings are comparable.

7.4.2 Spectral adaptivity: Comparison with interpolation by Chebyshev polynomials

An advantage of (polynomial) Krylov subspace over polynomial interpolation on the spectral interval of A is the fact that Krylov methods are less impacted by outliers in the

A	Splitting algorithm			Chebyshev interpolation		Dense
Size	Time	Err	nnz/n	Time	Err	Time
512	0.02	$4.42 \cdot 10^{-7}$	47.00	**0.01**	$1.59 \cdot 10^{-7}$	0.03
1,024	0.03	$4.56 \cdot 10^{-7}$	47.50	**0.02**	$1.59 \cdot 10^{-7}$	0.12
2,048	0.04	$4.56 \cdot 10^{-7}$	47.75	**0.02**	$1.61 \cdot 10^{-7}$	0.60
4,096	0.06	$4.58 \cdot 10^{-7}$	47.88	**0.05**	$1.61 \cdot 10^{-7}$	3.34
8,192	0.16	$4.59 \cdot 10^{-7}$	47.94	**0.12**	$1.61 \cdot 10^{-7}$	21.51
16,384	0.36	$4.60 \cdot 10^{-7}$	47.97	**0.27**	$1.61 \cdot 10^{-7}$	150.50
32,768	**0.51**		47.98	0.59		
65,536	**1.07**		47.99	1.11		
131,070	**2.23**		48.00	2.67		
262,140	**4.68**		48.00	5.38		
524,290	**9.24**		48.00	11.48		

Table 7.7 – Computation of $f(A)$ by Algorithm 7.3, where $f(z) = (\exp(\beta(z-\mu))+1)^{-1}$ and the matrices A are symmetric tridiagonal matrices with diagonal entries uniformly randomly distributed in $[0,1]$ and all other nonzero elements equal to -1, as discussed in Section 7.4.1.

spectrum of A. In the next experiment, we consider three 2048×2048 matrices:

- The exponential of $A_1 = \text{tridiag}(-1, 2, -1)$;
- The exponential of the matrix A_2 which is obtained from A_1 by changing the entry in position $(1,1)$ to 10;
- The square root of the matrix A_3 which is the tridiagonal matrix with `linspace(2, 3, n)` on the diagonal and -1 on the super-diagonal and subdiagonal.

We run Algorithm 7.2 with different block sizes and Chebyshev interpolation with different degrees of Chebyshev polynomial and we plot in Figure 7.6 the relative error in the Frobenius norm versus the number of nonzero entries in the approximation of the matrix functions described above. For the matrix A_1, Chebyshev outperforms Algorithm 7.2. However, for the matrix A_2 which has an outlier in the eigenvalues, and for the matrix A_3 for which it is difficult to find a good polynomial approximation on the whole spectral interval, Algorithm 7.2 achieves a smaller error with the same number of nonzero entries.

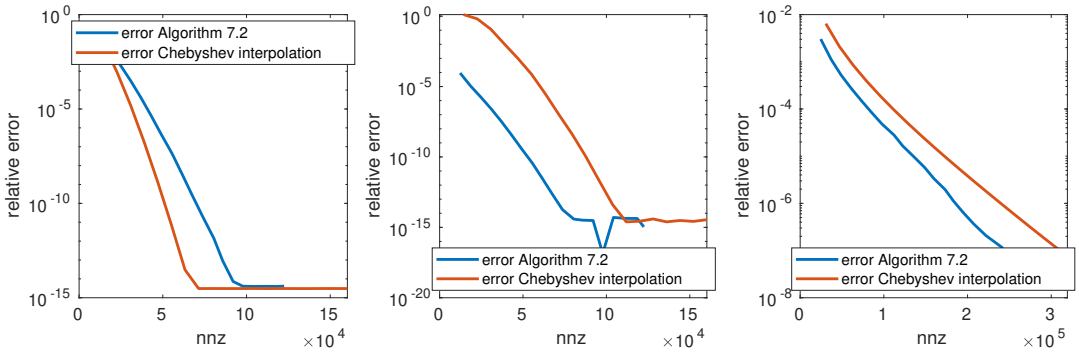


Figure 7.6 – Relative error in the Frobenius norm of the approximations of $\exp(A_1)$, $\exp(A_2)$, *and* $\sqrt{A_3}$ *from Section 7.4.2 obtained by Algorithm 7.2 and by Chebyshev interpolation.*

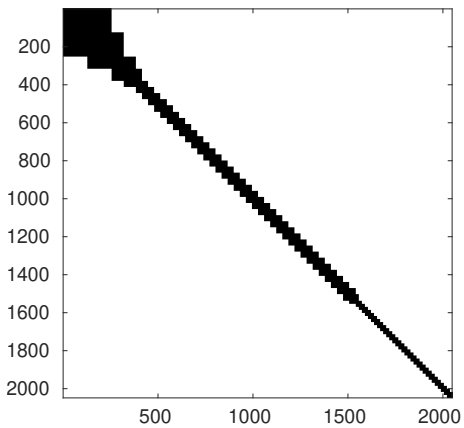
We do not report the timings: In general, Chebyshev interpolation is faster than our splitting algorithm; however, Chebyshev interpolation is only suitable for symmetric matrices (for non-symmetric matrices one needs more refined techniques such as using Faber polynomials as discussed, e.g., in [24]), while the splitting method works for any banded matrix, can automatically adapt to different spectral distributions, and could exploit the Toeplitz structure of A producing an approximation in constant time (as the matrices D, B, and C are made of equal blocks, we could compute only a constant number of matrix functions of the small blocks).

7.4.3 Adaptivity in the size of blocks

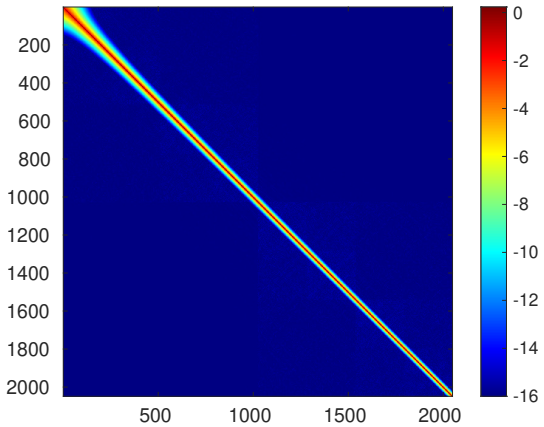
The matrix square root of A_3 has slower off-diagonal decay in the upper-left region, as shown in Figure 7.7(b). We run Algorithm 7.3 to compute ${A_3}^{1/2}$, setting $\varepsilon = 10^{-8}$. The sparsity pattern of the output is shown in Figure 7.7(a), where different block sizes are selected for different parts of the matrix; the relative error of the computed approximation is $2.6 \cdot 10^{-10}$ in the Frobenius norm.

7.4.4 Comparison with HSS algorithm

We expect Algorithm 7.3 to be faster than the general D&C algorithm (Algorithm 7.1) as the first one should scale as $\mathcal{O}(n)$ and the latter as $\mathcal{O}(n \log n)$, plus the fact that we have no overhead computations needed for HSS arithmetic. We compare the timings of the two algorithms for the computation of $\exp(-A)$ for $A = \text{tridiag}(-1, 2, -1)$. For Algorithm 7.3



(a) Sparsity structure of the output of Algorithm 7.3 applied to A_3 and $f(z) = z^{1/2}$.



(b) Logarithm of the absolute values of the entries of $A_3^{1/2}$.

Figure 7.7 – The matrix A_3 is the tridiagonal matrix with `linspace(2, 3, n)` *on the diagonal and -1 on the first super- and sub-diagonals.*

we use a minimum block size of 64, while for Algorithm 7.1 we set $n_{\min} = 128$ and we write each low-rank update as a rank-1 update as discussed in Remark 7.2; in both cases we set the tolerance parameter $\varepsilon = 10^{-8}$. We report the results in Figure 7.8, together with the timings of Matlab's `expm`, for matrix dimensions ranging from $n = 2^8$ to $n = 2^{18}$.

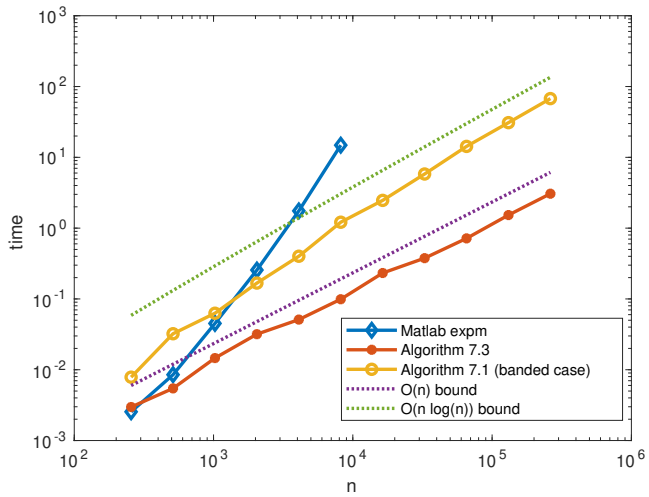


Figure 7.8 – Timings of Algorithms 7.3 and 7.1 for $\exp(-\mathrm{tridiag}(-1, 2, -1))$.

Stochastic trace estimation Part III

8 Introduction to stochastic trace estimation

Part III is concerned with the estimation of the trace of a symmetric matrix $B \in \mathbb{R}^{n\times n}$. As mentioned in Chapter 5, in many applications one is interested in computing the trace of a matrix function $B = f(A)$ for a symmetric matrix $A \in \mathbb{R}^{n\times n}$. The application we consider here is the computation of the log-determinant of an SPD matrix $A \in \mathbb{R}^{n\times n}$, which is related to the trace of $B = \log(A)$ via the equality

$$\log(\det(A)) = \operatorname{tr}(\log(A)); \tag{8.1}$$

see, e.g., [10]. The need for estimating determinants arises, for instance, in statistical learning [2, 73, 80], lattice quantum chromodynamics [178], and Markov random fields models [188]; moreover, certain quantities associated with graphs can be expressed as determinants, such as the number of spanning trees [67].

Computing the trace of a matrix function is, of course, a trivial task if we are willing to compute the full matrix $f(A)$, which usually has cubic cost. To compute the determinant, the standard way is to compute a Cholesky decomposition of A – which also has cubic cost – and to multiply the squares of its diagonal entries. However, if one is happy with an *estimate* of the trace of a matrix B, the Hutchinson trace estimator [114] allows us to get an approximation of $\operatorname{tr}(B)$ by computing some quadratic forms $X^T BX$ for suitable random vectors X of length n. The Hutchinson trace estimator is described in Section 8.1 and existing convergence results are presented in Section 8.2. The advantage of dealing with quadratic forms is that, if the matrix A has no low-rank structure, often the quantity $X^T BX = X^T f(A)X$ can be approximated via Krylov subspace projection methods much

faster than the computation of the whole $f(A)$; see Section 8.3 below.

Let us briefly mention that there are other methods to approximate traces and determinants, including randomized subspace iteration [167] and block Krylov methods [129], but they only work well in specific cases, e.g., when $A = \sigma I + C$ for a matrix C of low numerical rank. The Hutch++ trace estimator, recently proposed and analyzed for the SPD case in [140], overcomes this limitation via a combination of randomized low-rank approximation with the Hutchinson trace estimator. The Hutch++ algorithm will be briefly discussed in Chapter 10. Another direction of work on large-scale determinant estimation has explored the use of spectral sparsifiers for symmetric diagonally dominant matrices [67, 113].

8.1 The Hutchinson trace estimator

Let $B \in \mathbb{R}^{n\times n}$ be symmetric. The Hutchinson trace estimator is based on the following fact.

Proposition 8.1. *Let* $X = \begin{bmatrix} X_1 & \dots & X_n \end{bmatrix}^T$ *be a random vector of length* n *such that* $\mathbb{E}\left[XX^T\right] = I$. *Then*

$$\mathbb{E}\left[X^T B X\right] = \operatorname{tr}(B).$$

Proof. We have that

$$\mathbb{E}\left[X^T B X\right] = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}\mathbb{E}[X_i X_j] = \sum_{i=1}^{n} b_{ii} = \operatorname{tr}(B),$$

where the first equality follows from the linearity of the expectation and the second equality follows from the assumption that $\mathbb{E}[XX^T] = I$. □

The Hutchinson estimator is obtained by sampling an average of N quadratic forms:

$$\operatorname{tr}_N(B) := \frac{1}{N}\sum_{i=1}^{N}(X^{(i)})^T B X^{(i)}, \tag{8.2}$$

where the vectors $X^{(i)}$, $i = 1, \dots, N$, are independent copies of X, which we call *probe vectors*. The most common choices for X are standard Gaussian and Rademacher random

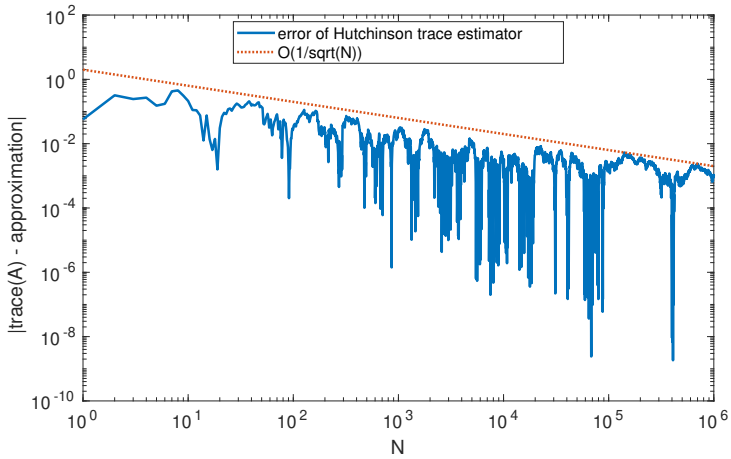


Figure 8.1 – For a randomly chosen symmetric matrix $A \in \mathbb{R}^{1000\times 1000}$ we plot the error $|\operatorname{tr}(A) - \operatorname{tr}_N^{\mathsf{R}}(A)|$ for increasing values of N. The error oscillates because this is a stochastic process, but overall converges to zero with rate $O(1/\sqrt{N})$, as it can be expected from a Monte Carlo method. The estimator $\operatorname{tr}_N^{\mathsf{G}}(B)$ behaves in an analogous way.

vectors; the latter are defined by having i.i.d. entries that take values ± 1 with equal probability. We will consider both choices and denote the resulting trace estimates by $\operatorname{tr}_N^{\mathsf{G}}(B)$ and $\operatorname{tr}_N^{\mathsf{R}}(B)$, respectively. Other possible strategies include spherical random vectors [8, 162] and probing vectors [79, 174]; we will briefly discuss the probing vectors in Chapter 10. A typical example of the behavior of the estimator (8.2) in the case of Rademacher random vectors is displayed in Figure 8.1.

8.2 Existing tail bounds for the Hutchinson estimator

By the central limit theorem, the estimate (8.2) can be expected to become more reliable as N increases; see, e.g., [44, Corollaries 3.3 and 4.3] for such an asymptotic result as $N \to \infty$. Most existing non-asymptotic results for trace estimation are specific to an SPD matrix B; see [8, 93, 162] for examples. They provide a bound on the estimated number N of probe vectors needed to ensure a small relative error with high probability:

$$\mathbb{P}\left(\left|\frac{\operatorname{tr}(B) - \operatorname{tr}_N(B)}{\operatorname{tr}(B)}\right| \geq \varepsilon\right) \leq \delta. \tag{8.3}$$

More specifically, in [8] it is shown that $N = 20\varepsilon^{-2}\log\frac{2}{\delta}$ and $N = 6\varepsilon^{-2}\log\frac{2n}{\delta}$ are sufficient for Gaussian and Rademacher vectors, respectively. In [162] these bounds are improved and it is shown that $N = 8\varepsilon^{-2}\frac{\|A\|_2}{\operatorname{tr}(A)}\log\frac{2}{\delta}$ and $N = 6\varepsilon^{-2}\log\frac{2}{\delta}$ are sufficient for Gaussian

and Rademacher vectors, respectively. In [93] it is proven that $N = 2\varepsilon^{-2} \log \frac{2}{\delta}$ is sufficient for Gaussian vectors when $\varepsilon \in \left(0, \frac{1}{2}\right)$.

The assumption that B is SPD is usually not met when computing the determinant of an SPD matrix A via (8.1) because $\log(A)$ is SPD only if all eigenvalues of A are larger than one. For general symmetric indefinite B, it is unrealistic to aim at a bound of the form (8.3) for the *relative* error, because the fact that $\operatorname{tr}(B) = 0$ does not imply that the Hutchinson trace estimator has zero error. The only result that holds for general symmetric indefinite B, as far as we know, is contained in [7]; it is shown that

$$\mathbb{P}\Big(|\operatorname{tr}_N(B) - \operatorname{tr}(B)| \geq \varepsilon\Big) \leq \delta \tag{8.4}$$

holds for Gaussian vectors if the number of samples is $N = 20\varepsilon^{-2}\|B\|_*^2 \log \frac{4}{\delta}$ and holds for Rademacher vectors if $N = 6\varepsilon^{-2}\|B\|_*^2 \log \frac{2 \cdot \operatorname{rank}(B)}{\delta}$; see Remarks 9.13 and 9.17 for a comparison with our results.

In the context of log-determinant approximation, there are several results obtained by suitably rescaling or modifying the results for SPD matrices contained in [8, 162]. Ubaru, Chen, and Saad [183] derive a bound for the absolute error via rescaling, that is, the results from [162] are applied to the matrix $C := -\log(\lambda A)$ for a value of $\lambda > 0$ that ensures that C is SPD. Specifically, for Rademacher vectors it is shown in [183, Corollary 4.5] that

$$\mathbb{P}\left(|\operatorname{tr}_N^{\mathsf{R}}(\log(A)) - \log\det(A)| \geq \varepsilon\right) \leq \delta \tag{8.5}$$

is satisfied with fixed failure probability δ if the number of samples N grows proportionally to $\varepsilon^{-2} n^2 \log(1 + \kappa(A))^2 \log \frac{2}{\delta}$ where $\kappa(A)$ denotes the condition number of A. A similar rescaling approach is used in [104], in which determinant estimation for indefinite matrices A is addressed by applying trace estimation to a suitable rescaled version of AA^T, in such a way that its logarithm is negative definite. Theorem 2 in [104] gives a number of samples that grows as $14\varepsilon^{-2} n^2 (\log(1 + \kappa(A)^2)^2 \log \frac{2}{\delta}$ to get an approximation Γ such that $\mathbb{P}(|\log\det(A) - \Gamma| \leq \varepsilon) > 1 - \delta$. Also in this case, they get a quadratic dependence on n.

Unfortunately, these *estimated* numbers of samples compare unfavorably with a much simpler approach; computing the trace from the diagonal elements of $\log(A)$ only requires the evaluation of n quadratic forms, using all n unit vectors of length n.

8.3 Approximating the quadratic forms

When using the Hutchinson trace estimator to compute $\operatorname{tr}(B)$ for the matrix function $B = f(A)$, the quadratic forms

$$(X^{(i)})^T B X^{(i)} = (X^{(i)})^T f(A) X^{(i)}$$

can in general not be computed exactly. For this, a polynomial approximation of f can be used. We restrict our discussion to $f(z) = \log(z)$, which corresponds to the log-determinant.

One possible strategy is to use a priori polynomial approximations. Chebyshev expansion/interpolation has been used in [103, 152] and approximation by Taylor series expansion has been investigated in [12, 33, 197]. Often, a better approximation can be obtained by the Lanczos method (see, e.g., [10]). Given the orthonormal basis U_m of the Krylov subspace $\mathcal{K}_m(A, X^{(i)})$ obtained by the Lanczos algorithm, one takes the approximations

$$f(A)X^{(i)} \approx U_m f(U_m^T A U_m) e_1, \qquad (X^{(i)})^T f(A) X^{(i)} \approx e_1^T f(U_m^T A U_m) e_1.$$

Note that the size of $U_m^T A U_m$ can be much smaller than the size of A and therefore computing $f(U_m^T A U_m)$ is cheaper than computing $f(A)$. The Lanczos method for computing quadratic forms is equivalent to applying Gaussian quadrature to the integral $\int \log(\lambda) d\mu(\lambda)$ on the spectral interval of A, for a suitably defined measure μ; see [85]. In the case of the logarithm, upper and lower bounds for the quantity $(X^{(i)})^T \log(A) X^{(i)}$ can be determined without much additional effort [10]. Moreover, the convergence of Gaussian quadrature for the quadratic form can be related to the best polynomial approximation of the logarithm on the spectral interval of A; see [183, Theorem 4.2] and Section 9.4.

By combining the polynomial approximation error with (8.5), one obtains a total error bound for log-determinant approximation that takes into account both sources of errors. Such a result is presented in [183, Corollary 4.5] for Rademacher vectors; the fact that all such vectors have bounded norm is essential in the analysis.

8.4 Contributions

In Chapter 9, we improve the results from [7, 183] by first showing that the number of samples required to achieve (8.4) with symmetric indefinite matrices is much lower for both Gaussian and Rademacher vectors. Our result for Rademacher vectors also implies an improved bound for SPD matrices. A summary of the bounds that we discussed in Section 8.2 and that we will derive in Chapter 9 on the number of probe vectors for the Hutchinson trace estimator in the SPD and indefinite case is presented in Tables 8.1 and 8.2.

Bound on the number of samples	Assumptions	Reference
$N = 20\varepsilon^{-2} \log \frac{2}{\delta}$	X Gaussian	[8]
$N = 2\varepsilon^{-2} \log \frac{2}{\delta}$	X Gaussian and $0 < \varepsilon < \frac{1}{2}$	[93]
$N = 8\varepsilon^{-2} \frac{\lVert B \rVert_2}{\mathrm{tr}(B)} \log \frac{2}{\delta}$	X Gaussian	[162]
$N = 6\varepsilon^{-2} \log \frac{2n}{\delta}$	X Rademacher	[8]
$N = 6\varepsilon^{-2} \log \frac{2}{\delta}$	X Rademacher	[162]
$N = 8\varepsilon^{-2}(1+\varepsilon) \frac{\lVert B \rVert_2}{\mathrm{tr}(B)} \log \frac{2}{\delta}$	X Rademacher	Corollary 9.24

Table 8.1 – Summary of the bounds on the number of probe vectors that ensure that (8.3) *holds, for an SPD matrix B.*

Bound on the number of samples	Assumptions	Reference
$N = 20\varepsilon^{-2} \lVert B \rVert_*^2 \log \frac{4}{\delta}$	X Gaussian	[7]
$N = 4\varepsilon^{-2}(\lVert B \rVert_F^2 + \varepsilon \lVert B \rVert_2) \log \frac{2}{\delta}$	X Gaussian	Theorem 9.12
$N = \varepsilon^{-2} n^2 \log(1+\kappa(A))^2 \log \frac{2}{\delta}$	X Rademacher, $B = \log(A)$	[183]
$N = 6\varepsilon^{-2} \lVert B \rVert_*^2 \log \frac{2 \cdot \mathrm{rank}(B)}{\delta}$	X Rademacher	[7]
$N = 8\varepsilon^{-2} \left(\lVert B \rVert_F^2 + 2\varepsilon \lVert B \rVert_2 \right) \log \frac{2}{\delta}$	X Rademacher	Corollary 9.16

Table 8.2 – Summary of the bounds on the number of probe vectors that ensure that (8.4) *holds, for a symmetric indefinite matrix B.*

Specialized to determinant computation, we combine our results with an improved analysis of the Lanczos method for estimating the quadratic forms $X^T \log(A)X$, to get a sharper total error bound for Rademacher vectors. Finally, we extend this combined error bound to Gaussian vectors, which requires some additional consideration because of the unboundedness of such vectors.

Chapter 9 is based on the paper [46].

9 Trace estimates for indefinite matrices with an application to determinants

In the first part of this chapter we prove new tail bounds for the Hutchinson trace estimator. More specifically, in Section 9.1 we consider a single-sample estimate ($N = 1$) and in Section 9.2 we extend the results to $\operatorname{tr}_N(B)$. We show for a general symmetric matrix B that

$$\mathbb{P}\Big(|\operatorname{tr}_N(B) - \operatorname{tr}(B)| \geq \varepsilon\Big) \leq \delta, \tag{9.1}$$

for both Gaussian and Rademacher vectors, is satisfied with fixed failure probability δ if the number of samples N grows proportionally with the stable rank

$$\rho(B) := \frac{\|B\|_F^2}{\|B\|_2^2}.$$

As $1 \leq \rho(B) \leq n$ (see, e.g., [182, Section 2.1.15]), our result improves the ones in [7] and in [183] by a factor which can be as large as n. We demonstrate that the dependence on n is asymptotically tight with an explicit example. For SPSD matrices B, our bound also improves the state-of-the-art result [162, Theorem 1] for Rademacher vectors by establishing that the number of probe vectors is inversely proportional to the stable rank of $B^{1/2}$. Section 9.3 contains numerical examples illustrating the behavior of our bounds and a comparison with previous results.

In the second part of the chapter we apply the analysis to the computation of the log-determinant of SPD matrices. More specifically, in Section 9.4 we provide an improved

analysis of Lanczos method for the approximation of quadratic forms $X^T \log(A)X$, in Section 9.5 we combine it with the results from Section 9.2 to get convergence bounds for log-determinant estimation. Finally, Section 9.6 contains some numerical examples.

9.1 Bounds for a single-sample estimate

In this section we consider single-sample estimates, that is, we look for tail bounds for the random variable $X^T BX$ when X is a Gaussian or Rademacher random vector. Such results will be generalized to tail bounds for $\mathrm{tr}_N^{\mathsf{G}}(B)$ and $\mathrm{tr}_N^{\mathsf{R}}(B)$ in Section 9.2.1. Here, B is a symmetric, possibly indefinite, $n \times n$ matrix.

A straightforward approach to get a tail bound for $X^T BX$ is via Chebyshev inequality, which gives

$$\mathbb{P}(|X^T BX - \mathrm{tr}(B)| \geq \varepsilon) \leq \frac{\mathrm{Var}(X^T BX)}{\varepsilon^2} \quad \text{for all } \varepsilon > 0. \tag{9.2}$$

The variance of the random variable $X^T BX$ is $2\|B\|_F^2$ for Gaussian vectors and is $2\|B - \mathrm{D}_B\|_F^2$ for Rademacher vectors, where D_B denotes the diagonal matrix containing the diagonal entries of B (see [8]). However, (9.2) is meaningless for small values of ε, and the failure probability in the right-hand-side decreases slowly when ε increases.

In fact, the random variable $X^T BX$ has been largely studied in the literature and goes under the name of *chaos of order 2*. In particular, the Hanson-Wright inequality [105, 163] is a tail bound for such quadratic form for vectors X whose entries are independent sub-Gaussian random variables.

Definition 9.1. *A random variable Y is sub-Gaussian if the quantity*

$$\|Y\|_{\psi_2} := \sup_{p \geq 1} p^{-1/2} \left(\mathbb{E}[|Y|^p]\right)^{1/p}$$

is finite.

We refer the reader to [76, Section 7.4] for equivalent definitions and properties of sub-Gaussian random variables.

Theorem 9.2 ([163, Theorem 1.1])**.** *There exists a universal constant $c > 0$ such that if X is a random vector of length n with $\mathbb{E}[X_i] = 0$, independent components which are*

sub-Gaussian with $\|X_i\|_{\psi_2} \leq K$*, and* $B \in \mathbb{R}^{n\times n}$ *is symmetric, then for all* $\varepsilon > 0$

$$\mathbb{P}\left(|X^T BX - \operatorname{tr}(B)| > \varepsilon\right) \leq 2\exp\left(-c\min\left\{\frac{\varepsilon^2}{K^4\|B\|_F^2}, \frac{\varepsilon}{2K^2\|B\|_2}\right\}\right).$$

Both Rademacher and Gaussian vectors satisfy the assumptions of Theorem 9.2, as both types of random variables are sub-Gaussian. However, to obtain practical bounds we are interested in finding the best constant c in these two particular cases.

For Gaussian vectors, results with explicit constants appear in [32, Example 2.12] and [127, Lemma 1], but they apply to symmetric matrices with zero diagonal and SPD matrices, respectively. Lemma 9.6 below is similar, but not identical, to these results.

For Rademacher vectors, the homogeneous case, corresponding to a matrix B with zero diagonal, has been studied extensively in the literature; see, e.g., [32, 76, 105, 122, 176]. In particular, the results stated in [1, Theorem 6] and [32, Exercise 6.9] give

$$\mathbb{P}\left(|X^T BX| \geq \varepsilon\right) \leq 2\exp\left(-\frac{\varepsilon^2}{16\|B\|_F^2 + 16\|B\|_2\varepsilon}\right)$$

and

$$\mathbb{P}\left(|X^T BX| \geq \varepsilon\right) \leq 2\exp\left(-\frac{\varepsilon^2}{32\|B\|_F^2 + 128\|B\|_2\varepsilon}\right),$$

respectively. Proposition 8.13 in [76] states

$$\mathbb{P}\left(|X^T BX| \geq \varepsilon\right) \leq 2\exp\left(-\min\left\{\frac{3\varepsilon^2}{128\|B\|_F^2}, \frac{\varepsilon}{32\|B\|_2}\right\}\right).$$

We will improve these constants in Theorem 9.10 below. The non-homogeneous case is easily obtained from the homogeneous case; see Corollary 9.16 below.

9.1.1 Sub-Gamma random variables

For both the Gaussian and the Rademacher case we will use a Chernoff bound for sub-Gamma random variables – a larger class than sub-Gaussian random variables; see, e.g., [32].

Definition 9.3. *A random variable* X *is called* sub-Gamma *with variance parameter*

$\nu > 0$ and scale parameter $c > 0$ if

$$\mathbb{E}[\exp(\lambda X)] \leq \exp\left(\frac{\nu\lambda^2}{2(1-c\lambda)}\right) \qquad \textit{for all } 0 < \lambda < \frac{1}{c}.$$

Lemma 9.4 ([32, Section 2.4])**.** *Let X be a sub-Gamma random variable with parameters (ν, c). Then, for all $\varepsilon \geq 0$, we have*

$$\mathbb{P}(X \geq \sqrt{2\varepsilon\nu} + c\varepsilon) \leq \exp(-\varepsilon).$$

Lemma 9.5 ([187, Proposition 2.10])**.** *Let X be a random variable such that $\mathbb{E}[X] = 0$, and such that both X and $-X$ are sub-Gamma with parameters (ν, c). Then, for all $\varepsilon \geq 0$, we have*

$$\mathbb{P}\left(|X| \geq \varepsilon\right) \leq 2\exp\left(-\frac{\varepsilon^2}{2(\nu + c\varepsilon)}\right).$$

9.1.2 Tail bounds for a single-sample estimate with Gaussian vectors

Let B be a real symmetric matrix and let $B = Q\Lambda Q^T$ be a spectral decomposition, where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ contains the eigenvalues of B and Q is an orthogonal matrix. Lemma 9.5 implies the following result for the tail of a single-sample trace estimate with Gaussian vectors.

Lemma 9.6. *For a Gaussian vector X of length n we have*

$$\mathbb{P}\left(|X^T BX - \operatorname{tr}(B)| \geq \varepsilon\right) \leq 2\exp\left(-\frac{\varepsilon^2}{4\|B\|_F^2 + 4\varepsilon\|B\|_2}\right)$$

for all $\varepsilon > 0$.

Proof. We let

$$Y := X^T BX - \operatorname{tr}(B) = X^T Q\Lambda Q^T X - \operatorname{tr}(B) = \sum_{i=1}^{n} \lambda_i (Z_i^2 - 1),$$

where $Z_i \sim \mathcal{N}(0,1)$ is the ith component of the Gaussian vector $Q^T X$. To show that Y is sub-Gamma, we define for $\lambda \in \mathbb{R}$ the function

$$\psi(\lambda) := \log \mathbb{E}[\exp(\lambda(Z^2 - 1))], \quad Z \sim \mathcal{N}(0,1).$$

We have that $\psi(\lambda) = \log \frac{\mathbb{E}[\exp(\lambda Z^2)]}{\exp(\lambda)} = -\lambda - \frac{1}{2}\log(1-2\lambda)$ for $\lambda < \frac{1}{2}$. In particular, this implies $\psi(\lambda) \leq \frac{\lambda^2}{1-2\lambda}$ for $0 \leq \lambda < \frac{1}{2}$, and $\psi(\lambda) \leq \lambda^2 \leq \frac{\lambda^2}{1+c\lambda}$ for $-\frac{1}{c} < \lambda < 0$ for all $c > 0$. Using the independence of Z_i for different i we obtain

$$\begin{aligned}\log \mathbb{E}[\exp(\lambda Y)] &= \sum_{i=1}^{n} \log \mathbb{E}[\exp(\lambda\lambda_i(Z_i^2 - 1))] = \sum_{i=1}^{n} \psi(\lambda\lambda_i) \\ &\leq \sum_{i=1}^{n} \frac{\lambda_i^2\lambda^2}{1-2|\lambda_i|\lambda} \leq \frac{\|B\|_F^2\lambda^2}{1-2\|B\|_2\lambda}\end{aligned}$$

for $0 < \lambda < \frac{1}{2\|B\|_2}$. This shows that Y is sub-Gamma with parameters $(\nu, c) = (2\|B\|_F^2, 2\|B\|_2)$. Moreover, $-Y = X^T(-B)X - \operatorname{tr}(-B)$ is also sub-Gamma with the same parameters. Because $\mathbb{E}[Y] = 0$, Lemma 9.5 implies the desired result. □

9.1.3 Tail bounds for a single-sample estimate with Rademacher vectors

We now assume that X is a Rademacher vector. The property that the multiplication with orthogonal matrices preserves Gaussian random vectors, which has been exploited in the proof of Lemma 9.6, does not extend to the Rademacher case, and the moment generating function of X^TBX cannot be written in a simple form. Therefore, other strategies need to be used. Note that, as Rademacher random variables are bounded, classical tools such as Hoeffding and Bernstein inequalities can be used to directly obtain tail estimates for $\operatorname{tr}_N^{\mathsf{R}}(B)$, but they do not give sharp results; see Remark 9.18. Instead, we make use of the the entropy method [32] to establish the tail bound in Theorem 9.10 for a single-sample trace estimate.

Definition 9.7. *The entropy of a random variable Z is defined as*

$$\mathbb{H}(Z) := \mathbb{E}[Z \log Z] - \mathbb{E}[Z] \log \mathbb{E}[Z],$$

provided that all expected values exist.

We will need the two following ingredients. The Herbst argument (see, e.g., [32, page 11], [76, pages 239–240], and [187, Section 3.1.2]) turns a bound on the entropy of a random variable into a bound on the moment generating function. By Chernoff's bound, the latter implies a bound on the tail of the random variable. Specifically, we use the

following (modified) Herbst argument.

Lemma 9.8. *Let Z be a random variable and $g : [0, a) \to \mathbb{R}$ such that*

$$\mathbb{H}(\exp(\lambda Z)) \leq \lambda^2 g(\lambda)\mathbb{E}[\exp(\lambda Z)]. \tag{9.3}$$

Then for all $\lambda \in [0, a)$ it holds

$$\log \mathbb{E}[\exp(\lambda Z)] \leq \lambda \mathbb{E}[Z] + \lambda \int_0^\lambda g(\xi)\mathrm{d}\xi.$$

Proof. For $\psi(\lambda) := \log \mathbb{E}[\exp(\lambda Z)]$, it holds that $\psi'(\lambda) = \mathbb{E}[Z \exp(\lambda Z)]/\mathbb{E}[\exp(\lambda Z)]$. Recalling the definition of entropy, this allows us to rewrite (9.3) as

$$\lambda\psi'(\lambda)\exp(\psi(\lambda)) - \psi(\lambda)\exp(\psi(\lambda)) \leq \lambda^2 g(\lambda)\exp(\psi(\lambda)),$$

which is equivalent to

$$\frac{\mathrm{d}}{\mathrm{d}\lambda}\left(\frac{\psi(\lambda)}{\lambda}\right) \leq g(\lambda).$$

Integration on the interval $[0, \lambda]$ gives

$$\frac{\psi(\lambda)}{\lambda} - \lim_{\lambda \to 0^+} \frac{\psi(\lambda)}{\lambda} \leq \int_0^\lambda g(\xi)\mathrm{d}\xi.$$

We conclude by noting that $\lim_{\lambda \to 0^+} \frac{\psi}{\lambda} = \mathbb{E}[Z]$. □

For deriving bounds on the entropy, we need the following two variations of Gross' logarithmic Sobolev inequality.

Theorem 9.9. *Let $f : \{-1, 1\}^n \to \mathbb{R}$ and let X be a Rademacher vector with components $X_1, \ldots, X_n$. Define $f(\bar{X}^{(i)}) := f(X_1, \ldots, X_{i-1}, -X_i, X_{i+1}, \ldots, X_n)$ for $i = 1, \ldots, n$. Then for all $\lambda > 0$ we have*

$$\mathbb{H}(\exp(\lambda f(X))) \leq \frac{\lambda^2}{4}\mathbb{E}\left[\exp(\lambda f(X)) \sum_{i=1}^n \left(f(X) - f(\bar{X}^{(i)})\right)_+^2\right] \tag{9.4}$$

and

$$\mathbb{H}(\exp(\lambda f(X))) \leq \frac{\lambda^2}{8}\mathbb{E}\left[\exp(\lambda f(X)) \sum_{i=1}^n \left(f(X) - f(\bar{X}^{(i)})\right)^2\right]. \tag{9.5}$$

Proof. The inequality (9.4) is a standard result and can be found, e.g., in [32, page 122]. The inequality (9.5) is a variation of the same argument; see also [32, Exercise 5.5] for a related (but not identical) result. The inequality (9.5) can, in fact, be found in a Master's thesis [1, Theorem 5]. For convenience of the reader, we provide a proof of (9.5) based on the textbook [32].

In [32, page 122] it is proven that

$$\mathbb{H}(\exp(\lambda f(X))) \leq \frac{1}{2}\mathbb{E}\left[\sum_{i=1}^{n}\left(\exp(\lambda f(X)/2) - \exp(\lambda f(\bar{X}^{(i)})/2)\right)^2\right]. \tag{9.6}$$

For $a \geq b$ we have

$$\begin{aligned}\exp\left(\frac{a}{2}\right) - \exp\left(\frac{b}{2}\right) &= \int_{b/2}^{a/2} \exp(t)\mathrm{d}t \leq \frac{a-b}{2} \cdot \frac{\exp\left(\frac{a}{2}\right) + \exp\left(\frac{b}{2}\right)}{2} \\ &\leq \frac{a-b}{2}\sqrt{\frac{\exp(a)+\exp(b)}{2}},\end{aligned}$$

where the first inequality follows from the concavity of the exponential and the Hermite-Hadamard inequality. Therefore, for all $a, b \in \mathbb{R}$ we have

$$(\exp(a/2) - \exp(b/2))^2 \leq \frac{1}{8}(a-b)^2(\exp(a) + \exp(b)). \tag{9.7}$$

Applying (9.7) to each summand in Equation (9.6) one obtains

$$\begin{aligned}\mathbb{H}(\exp(\lambda f(X))) &\leq \frac{\lambda^2}{16}\sum_{i=1}^{n}\mathbb{E}\left[(f(X) - f(\bar{X}^{(i)}))^2\left(\exp(\lambda f(X)) + \exp(\lambda f(\bar{X}^{(i)}))\right)\right] \\ &= \frac{\lambda^2}{16}\sum_{i=1}^{n}\mathbb{E}\left[(f(X) - f(\bar{X}^{(i)}))^2\exp(\lambda f(X))\right] \\ &+ \frac{\lambda^2}{16}\sum_{i=1}^{n}\mathbb{E}\left[(f(X) - f(\bar{X}^{(i)}))^2\exp(\lambda f(\bar{X}^{(i)}))\right] \\ &= \frac{\lambda^2}{8}\mathbb{E}\left[\exp(\lambda f(X))\sum_{i=1}^{n}\left(f(X) - f(\bar{X}^{(i)})\right)^2\right],\end{aligned}$$

where the last equality follows from the fact that $f(X)$ and $f(\bar{X}^{(i)})$ have the same distribution and changing the sign of the ith entry of $\bar{X}^{(i)}$ gives X again. □

We are now ready to state and prove our tail bound for Rademacher chaos of order 2, that is, for a single-sample estimate.

Theorem 9.10. *Let X be a Rademacher vector of length n and let B be a nonzero symmetric matrix such that $B_{ii} = 0$ for $i = 1, \ldots, n$. Then, for all $\varepsilon > 0$,*

$$\mathbb{P}\left(|X^T BX| \geq \varepsilon\right) \leq 2\exp\left(-\frac{\varepsilon^2}{8\|B\|_F^2 + 8\varepsilon\|B\|_2}\right). \tag{9.8}$$

Proof. The proof follows closely [1, Theorem 6] and [32, Theorem 17]; see Remark 9.11 for a comparison with these results. The main idea of the proof is as follows. Using the logarithmic Sobolev inequalities presented in Theorem 9.9, a bound on the entropy of the random variable $X^T BX$ is obtained. Using the modified Herbst argument of Lemma 9.8, we derive a bound on the moment generating function (MGF) of $X^T BX$, establishing that it is sub-Gamma with certain constants, which then allows us to apply Lemma 9.5.

Without loss of generality, we may assume $\|B\|_2 = 1$; the general case follows from applying the result to $\tilde{B} := B/\|B\|_2$. Let us consider the function $f : \{-1, 1\}^n \to \mathbb{R}$ defined as

$$f(x) = x^T Bx = \sum_{i \neq j} x_i x_j B_{ij}.$$

We want to apply the logarithmic Sobolev inequality (9.5) from Theorem 9.9 to $f(X)$. For this purpose, we let

$$\bar{X}^{(i)} = \left[X_1, \ldots, X_{i-1}, -X_i, X_{i+1}, \ldots, X_n\right]^T = X - 2X_i e_i, \quad i = 1, \ldots, n,$$

where e_i denotes the ith unit vector. Using that B has zero diagonal entries, we obtain

$$f(X) - f(\bar{X}^{(i)}) = \langle BX, X\rangle - \langle BX - 2X_i Be_i, X - 2X_i e_i\rangle = 4X_i\langle Be_i, X\rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $\mathbb{R}^n$. Therefore, denoting

$$Y := \|BX\|_2^2 = \sum_{i=1}^n \left(\sum_{j=1}^n B_{ij} X_j\right)^2,$$

Theorem 9.9 establishes, for all $\lambda > 0$,

$$\mathbb{H}(\exp(\lambda f(X))) \leq 2\lambda^2 \mathbb{E}\left[Y \exp(\lambda f(X))\right]. \tag{9.9}$$

The decoupling inequality in [76, Lemma 8.50], which follows from Jensen's inequality, gives

$$\lambda\mathbb{E}[Y\exp(\lambda f(X))] \leq \mathbb{H}(\exp(\lambda f(X))) + \mathbb{E}[\exp(\lambda f(X))]\log\mathbb{E}[\exp(\lambda Y)].$$

Combined with (9.9), this implies

$$\mathbb{H}(\exp(\lambda f(X))) \leq \frac{2\lambda}{1-2\lambda}\mathbb{E}[\exp(\lambda f(X))]\cdot\log\mathbb{E}[\exp(\lambda Y)] \text{ for } 0<\lambda<\frac{1}{2}. \tag{9.10}$$

To find an upper bound on the MGF of Y, we use again a logarithmic Sobolev inequality, then transform the obtained bound on the entropy into a bound on the MGF by Herbst argument. We do so by applying the inequality (9.4) from Theorem 9.9 to the function $h:\mathbb{R}^n\to\mathbb{R}$ defined by $h(x):=\|Bx\|_2^2$. For this purpose, note that

$$\begin{aligned} h(X)-h(\bar{X}^{(i)}) &= \langle BX,BX\rangle - \langle B\bar{X}^{(i)},B\bar{X}^{(i)}\rangle = \langle B(X-\bar{X}^{(i)}),B(X+\bar{X}^{(i)})\rangle \\ &= 4\langle X_iBe_i, BX-X_iBe_i\rangle \leq 4X_i\langle Be_i,BX\rangle \end{aligned}$$

and, hence,

$$\sum_{i=1}^n\left(h(X)-h(\bar{X}^{(i)})\right)_+^2 \leq 16\sum_{i=1}^n\langle Be_i,BX\rangle^2 = 16\|B^TBX\|_2^2 \leq 16\|BX\|_2^2,$$

where we used that $\|B\|_2=1$ for the last inequality. Therefore Theorem 9.9 gives

$$\mathbb{H}(\exp(\lambda Y)) \leq 4\lambda^2\mathbb{E}[Y\exp(\lambda Y)].$$

Letting $g(\lambda):=4\mathbb{E}[Y\exp(\lambda Y)]/\mathbb{E}[\exp(\lambda Y)]$, we have obtained a bound of the form (9.3), as required by Lemma 9.8. Note that $g(\lambda)=4\psi'(\lambda)$, where $\psi(\lambda):=\log\mathbb{E}[\exp(\lambda Y)]$. The result of Lemma 9.8 gives

$$\log\mathbb{E}[\exp(\lambda Y)] \leq \frac{\lambda}{1-4\lambda}\|B\|_F^2 \quad \text{for } \lambda\in\left(0,\frac{1}{4}\right).$$

Inserting this inequality into (9.10) gives

$$\mathbb{H}(\exp(\lambda f(X))) \leq \frac{2\lambda^2\|B\|_F^2}{(1-4\lambda)(1-2\lambda)}\mathbb{E}[\exp(\lambda f(X))] \quad \text{for } \lambda\in\left(0,\frac{1}{4}\right).$$

The random variable $f(X)$ satisfies (9.3) for the function $g(\lambda) := \frac{2\|B\|_F^2}{(1-4\lambda)(1-2\lambda)}$ in the interval $[0, 1/4)$. Recalling that $\mathbb{E}[f(X)] = 0$, the result of Lemma 9.8 gives

$$\log \mathbb{E}[\exp(\lambda f(X))] \leq \lambda \|B\|_F^2 \log \frac{1-2\lambda}{1-4\lambda} \leq \frac{2\|B\|_F^2 \lambda^2}{1-4\lambda}, \quad \lambda \in [0, 1/4),$$

where we used $\log(1+x) \leq x$ in the last inequality.

Replacing f by $-f$ and B by $-B$, we also obtain

$$\log \mathbb{E}[\exp(-\lambda f(X))] \leq \frac{2\|B\|_F^2 \lambda^2}{1-4\lambda}, \quad \lambda \in [0, 1/4).$$

Therefore the random variables $f(X)$ and $-f(X)$ are sub-Gamma with parameters $(4\|B\|_F^2, 4)$. Applying Lemma 9.5 concludes the proof. □

Remark 9.11. *The proof of Theorem 9.10 follows the proof of [1, Theorem 6], which in turn refines a result from [31, Theorem 17] (see also [32]) by substituting the more general logarithmic Sobolev inequality from [31, Proposition 10] with the ones from Theorem 9.9 specific for Rademacher random variables. However, let us stress that the results in [1, 31] feature larger constants partly because they deal with the more general Rademacher chaos*

$$f(X) = \sup_{B \in \mathcal{B}} \sum_{i \neq j} X_i X_j B_{ij},$$

where $\mathcal{B}$ is a set of symmetric matrices with zero diagonal.

9.2 Tail bounds for trace estimation

In this section we use a diagonal embedding trick to turn Lemma 9.6 and Theorem 9.10 into tail bounds of the form (9.1) for the Hutchinson trace estimator applied to a symmetric, possibly indefinite matrix $B \in \mathbb{R}^{n \times n}$.

9.2.1 Tail bounds for trace estimation with Gaussian vectors

Here we assume that X is a standard Gaussian vector.

Theorem 9.12. *Let $B \in \mathbb{R}^{n\times n}$ be symmetric. Then*

$$\mathbb{P}\left(|\operatorname{tr}_N^{\mathsf{G}}(B) - \operatorname{tr}(B)| \geq \varepsilon\right) \leq 2\exp\left(-\frac{N\varepsilon^2}{4\|B\|_F^2 + 4\varepsilon\|B\|_2}\right)$$

for all $\varepsilon > 0$. In particular, for $N \geq \frac{4}{\varepsilon^2}(\|B\|_F^2 + \varepsilon\|B\|_2)\log\frac{2}{\delta}$ it holds that $\mathbb{P}(|\operatorname{tr}_N^{\mathsf{G}}(B) - \operatorname{tr}(B)| \geq \varepsilon) \leq \delta$.

Proof. We apply Lemma 9.6 to the matrix

$$\mathcal{B} := \operatorname{diag}\Big(N^{-1}B, \ldots, N^{-1}B\Big) \in \mathbb{R}^{Nn\times Nn}, \tag{9.11}$$

that is, the block diagonal matrix with the N diagonal blocks containing rescaled copies of B. In turn, the trace estimate (8.2) equals $X^T\mathcal{B}X$ for a Gaussian vector X of length Nn. Noting that $\|\mathcal{B}\|_F = N^{-1/2}\|B\|_F$ and $\|\mathcal{B}\|_2 = N^{-1}\|B\|_2$, the first part of the corollary follows from Lemma 9.6. Setting

$$\delta := 2\exp\left(-\frac{\varepsilon^2}{4\|\mathcal{B}\|_F^2 + 4\varepsilon\|\mathcal{B}\|_2}\right) = 2\exp\left(-\frac{N\varepsilon^2}{4\|B\|_F^2 + 4\varepsilon\|B\|_2}\right)$$

we obtain $N = \frac{4}{\varepsilon^2}\left(\|B\|_F^2 + \varepsilon\|B\|_2\right)\log\frac{2}{\delta}$. □

Remark 9.13. *The result of Theorem 9.12 compares favorably with Lemma 4 in [7], which shows that $\mathbb{P}(|\operatorname{tr}_N^{\mathsf{G}}(B) - \operatorname{tr}(B)| \geq \varepsilon) \leq \delta$ for $N \geq \frac{20}{\varepsilon^2}\|B\|_*^2\log\frac{4}{\delta}$. Because of $\|B\|_F \leq \|B\|_* \leq \sqrt{n}\|B\|_F$, the bound of Theorem 9.12 is always better for reasonably small values of ε (e.g. $\varepsilon \leq 5\|B\|_*$), and it can improve the estimated number of samples N in [7] by a factor proportional to n.*

We recall that the *stable rank* of B is defined as $\rho = \|B\|_F^2/\|B\|_2^2$ and satisfies $\rho \in [1, n]$. In particular, $\rho(B) = 1$ when B has rank one and $\rho(B) = n$ when all singular values are equal. Intuitively, $\rho(B)$ tends to be large when B has many singular values not significantly smaller than the largest one. The minimum number of probe vectors required by Theorem 9.12 depends on the stable rank of B in the following way:

$$\frac{4}{\varepsilon^2}(\rho\|B\|_2^2 + \varepsilon\|B\|_2)\log\frac{2}{\delta} \leq \frac{4}{\varepsilon^2}(n\|B\|_2^2 + \varepsilon\|B\|_2)\log\frac{2}{\delta}.$$

The upper bound indicates that N may need to be chosen proportionally with n to reach a fixed (absolute) accuracy ε with constant success probability, provided that $\|B\|_2$ remains

constant as well. The following lemma shows for a simple matrix B that such a linear growth of N can actually not be avoided.

Lemma 9.14. *Let n be even and consider the traceless matrix $B = \begin{bmatrix} I_{n/2} & 0 \\ 0 & -I_{n/2} \end{bmatrix}$. Then, for every $\varepsilon > 0$, it holds that*

$$\mathbb{P}(|\operatorname{tr}_N^{\mathsf{G}}(B)| \leq \varepsilon) \leq \varepsilon\sqrt{\frac{N}{\pi n}}.$$

Proof. By the definition of B, the trace estimate takes the form

$$\operatorname{tr}_N^{\mathsf{G}}(B) = \frac{1}{N}\left(\sum_{i=1}^{nN/2} X_i^2 - \sum_{j=1}^{nN/2} Y_j^2\right)$$

for independent $X_i, Y_j \sim N(0,1)$. In other words,

$$N \cdot \operatorname{tr}_N^{\mathsf{G}}(B) = X - Y,$$

where X, Y are independent Chi-squared random variables with $\frac{nN}{2}$ degrees of freedom. The probability density function f of $Z = X - Y$ can be expressed as

$$f(z) = \frac{1}{2^{nN/4}\sqrt{\pi}\Gamma(nN/4)}|z|^{\frac{nN}{4}-\frac{1}{2}}K_{\frac{nN}{4}-\frac{1}{2}}(|z|),$$

where $K_{\frac{nN}{4}-\frac{1}{2}}$ is a modified Bessel function of the second kind [60]. In particular,

$$f(0) = \frac{1}{4\sqrt{\pi}}\frac{\Gamma\left(\frac{nN}{4}-\frac{1}{2}\right)}{\Gamma\left(\frac{nN}{4}\right)} = \frac{1}{4\sqrt{\pi}}\frac{\sqrt{\pi}}{2^{\frac{nN}{2}-2}}\binom{\frac{nN}{2}-2}{\frac{nN}{4}-1} \leq \frac{1}{2\sqrt{\pi n N}},$$

where we used the duplication formula for Gamma functions and the inequality $\frac{1}{2^{2k}}\binom{2k}{k} \leq \frac{1}{\sqrt{\pi k}}$; see [185].

As f is an autocorrelation function (of the density function of a Chi-squared variable with $nN/2$ degrees of freedom), its maximum is at 0. We can therefore estimate the probability of $X - Y$ being in the interval $[-N\varepsilon, N\varepsilon]$ in the following way:

$$\mathbb{P}(|\operatorname{tr}_N^{\mathsf{G}}(B)| \leq \varepsilon) = \mathbb{P}(|X - Y| \leq N\varepsilon) \leq 2N\varepsilon f(0) \leq \varepsilon\sqrt{\frac{N}{\pi n}}. \qquad \square$$

We can reformulate Theorem 9.12 in such a way that, given a number N of probe

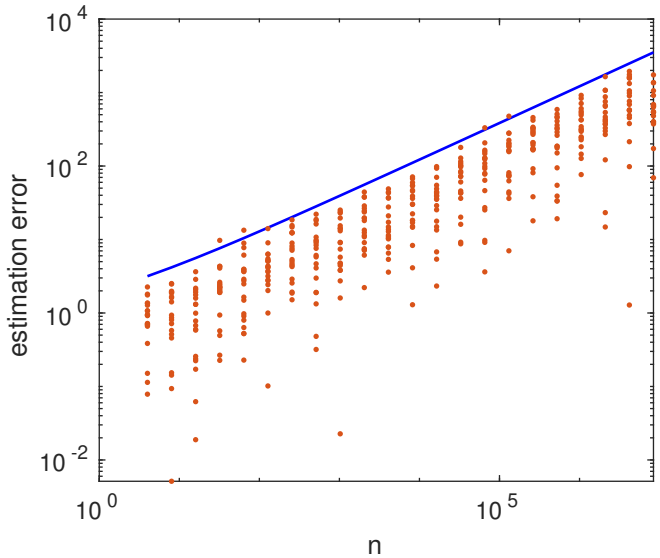


Figure 9.1 – Dots: Errors $|\operatorname{tr}_{10}^{\mathsf{G}}(B) - \operatorname{tr}(B)|$ *of* 20 *samples for each* $n = 2^k$ *with* $k = 2, \ldots, 23$, *where* B *is the matrix from Lemma 9.14. Blue line: Error bound* $\varepsilon(B, 0.05, 10)$ *from* (9.12).

vectors and a failure probability $\delta \in (0,1)$, we have $\varepsilon = \varepsilon(B, \delta, N)$ such that with probability at least $1 - \delta$ one has $\operatorname{tr}_N^{\mathsf{G}}(B) \in [\operatorname{tr}(B) - \varepsilon, \operatorname{tr}(B) + \varepsilon]$. The random variable $X^T \mathcal{B} X - \operatorname{tr}(\mathcal{B})$, where $\mathcal{B}$ is defined as in (9.11) and X is a Gaussian vector of length nN, is sub-Gamma with parameters $\left(2\frac{\|B\|_F^2}{N}, 2\frac{\|B\|_2}{N}\right)$, and the same holds for $-X^T \mathcal{B} X$. By Lemma 9.4 we have

$$\varepsilon \equiv \varepsilon(B, \delta, N) = \frac{2}{\sqrt{N}} \|B\|_F \sqrt{\log \frac{2}{\delta}} + \frac{2}{N} \|B\|_2 \log \frac{2}{\delta} \leq 2 \left(\sqrt{\frac{n}{N} \log \frac{2}{\delta}} + \frac{1}{N} \log \frac{2}{\delta} \right) \|B\|_2. \tag{9.12}$$

As the example in Lemma 9.14 shows, the potential growth of ε with $\sqrt{n}$ cannot be avoided in general. Figure 9.1 illustrates this growth. In the case of relative error estimates for symmetric positive semidefinite (SPSD) matrices, it is shown in [191] that the dependence on $\log \frac{2}{\delta}$ and $\frac{1}{\varepsilon^2}$ cannot be improved.

Remark 9.15. *For a nonzero SPSD matrix* B*, the result of Theorem 9.12 can be turned into a relative error estimate. Let* $\gamma := \|B\|_2 / \operatorname{tr}(B) = \rho(B^{1/2})^{-1}$ *be the inverse of the intrinsic dimension of* B *(see, e.g., [182, Section 7.1]). Replacing* ε *by* $\varepsilon \cdot \operatorname{tr}(B)$ *in Theorem 9.12 and noting that* $\|B\|_F^2 / \operatorname{tr}(B)^2 \leq \gamma$*, one obtains*

$$\mathbb{P}\left(\frac{|\operatorname{tr}_N^{\mathsf{G}}(B) - \operatorname{tr}(B)|}{\operatorname{tr}(B)} \geq \varepsilon \right) \leq \delta \quad \textit{for } N \geq \frac{4}{\varepsilon^2}(1+\varepsilon)\gamma \log \frac{2}{\delta}.$$

State-of-the-art results of a similar form are Theorem 3 in [162], which requires $N \geq \frac{8}{\varepsilon^2}\gamma \log \frac{2}{\delta}$*, and Corollary 3.3 in [93], which requires* $N \geq \frac{2}{\varepsilon^2}\gamma \log \frac{2}{\delta}$ and $\varepsilon \in \left(0, \frac{1}{2}\right)$*. Compared to [93], our result imposes no restriction on* ε *at the expense of a somewhat larger constant. On the other hand, as* $\varepsilon \leq 1$*, our result is always more favorable than the result from [162] for SPSD matrices.*

9.2.2 Tail bounds for trace estimation with Rademacher vectors

As for Gaussian vectors, the result of Theorem 9.10 can be turned into a tail bound for $\operatorname{tr}_N^{\mathsf{R}}(B)$ by block-diagonal embedding.

Corollary 9.16. *Let* B *be a nonzero symmetric matrix. Then*

$$\mathbb{P}\Big(|\operatorname{tr}_N^{\mathsf{R}}(B) - \operatorname{tr}(B)| \geq \varepsilon\Big) \leq 2\exp\left(-\frac{N\varepsilon^2}{8\|B - \mathrm{D}_B\|_F^2 + 8\varepsilon\|B - \mathrm{D}_B\|_2}\right)$$

for every $\varepsilon > 0$*. In particular, for*

$$N \geq \frac{8}{\varepsilon^2}\left(\|B - \mathrm{D}_B\|_F^2 + \varepsilon\|B - \mathrm{D}_B\|_2\right)\log\frac{2}{\delta}$$

it holds that $\mathbb{P}\Big(|\operatorname{tr}_N^{\mathsf{R}}(B) - \operatorname{tr}(B)| \geq \varepsilon\Big) \leq \delta$*.*

Proof. Let $C := B - \mathrm{D}_B$ and $\mathcal{C} := \operatorname{diag}\Big(N^{-1}C, \ldots, N^{-1}C\Big) \in \mathbb{R}^{Nn\times Nn}$. Then, $\operatorname{tr}_N^{\mathsf{R}}(B) - \operatorname{tr}(B) = X^T\mathcal{C}X$ for a Rademacher vector X of length Nn. The matrix $\mathcal{C}$ has zero diagonal, $\|\mathcal{C}\|_F = N^{-1/2}\|C\|_F$, and $\|\mathcal{C}\|_2 = N^{-1}\|C\|_2$. Now, the first part of the corollary directly follows from Theorem 9.10. Imposing a failure probability of δ in (9.8) gives

$$\delta := 2\exp\left(-\frac{\varepsilon^2}{8\|\mathcal{C}\|_F^2 + 8\varepsilon\|\mathcal{C}\|_2}\right) = 2\exp\left(-\frac{N\varepsilon^2}{8\|C\|_F^2 + 8\varepsilon\|C\|_2}\right),$$

and hence $N = \frac{8}{\varepsilon^2}\left(\|C\|_F^2 + \varepsilon\|C\|_2\right)\log\frac{2}{\delta}$. □

An alternative expression for the lower bound on N is obtained by noting that $\|B - \mathrm{D}_B\|_F \leq \|B\|_F$ and $\|B - \mathrm{D}_B\|_2 \leq 2\|B\|_2$ (the factor 2 in the latter inequality is asymptotically tight, see, e.g., [29]). The result of Corollary 9.16 thus states that N needs

to be at least as large as:

$$\frac{8}{\varepsilon^2}(\rho\|B\|_2^2 + 2\varepsilon\|B\|_2)\log\frac{2}{\delta} \leq \frac{8}{\varepsilon^2}(n\|B\|_2^2 + 2\varepsilon\|B\|_2)\log\frac{2}{\delta},$$

where ρ is the stable rank of B.

Remark 9.17. *In analogy to the Gaussian case (see Remark 9.13), the result of Corollary 9.16 compares favorably with Lemma 5 in [7], which shows that* $\mathbb{P}(|\operatorname{tr}_N^{\mathsf{R}}(B) - \operatorname{tr}(B)| \geq \varepsilon) \leq \delta$ *for* $N \geq \frac{6}{\varepsilon^2}\|B\|_*^2 \log\frac{2\cdot\operatorname{rank}(B)}{\delta}$.

Remark 9.18. *It is instructive to compare the result of Corollary 9.16 to the straightforward application of Hoeffding's and Bernstein's inequalities, two classical concentration inequalities for the sum of bounded random variables.*

Theorem 9.19 (Hoeffding's inequality)**.** *Consider a sum of independent random variables* $\sum_{i=1}^N Y_i$ *such that there exist constants* $a_i \leq Y_i \leq b_i$ *for all* $i = 1, \ldots, N$. *Denoting* $C := \max_{i=1,\ldots,N}(b_i - a_i)$ *we have*

$$\mathbb{P}\left(\left|\sum_{i=1}^N Y_i - \sum_{i=1}^N \mathbb{E}[Y_i]\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{2\varepsilon^2}{NC^2}\right).$$

Theorem 9.20 (Bernstein's inequality)**.** *Consider a sum of independent random variables* $\sum_{i=1}^N Y_i$ *such that there exist constants* $a_i \leq Y_i \leq b_i$ *for all* $i = 1, \ldots, N$. *Denoting* $C := \max_{i=1,\ldots,N}(b_i - a_i)$ *we have*

$$\mathbb{P}\left(\left|\sum_{i=1}^N Y_i - \sum_{i=1}^N \mathbb{E}[Y_i]\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{\varepsilon^2/2}{\sum_{i=1}^N \operatorname{Var}(Y_i) + C\varepsilon/3}\right).$$

We apply these inequalities to the random variables $Y_i := \frac{1}{N}(X^{(i)})^T(B - \mathrm{D}_B)X^{(i)}$ *for* $i = 1, \ldots, N$ *where the vectors* $X^{(1)}, \ldots, X^{(N)}$ *are Rademacher vectors. These random variables are bounded, in particular we have*

$$|Y_i| \leq \frac{1}{N}\|X^{(i)}\|_2^2\|B - \mathrm{D}_B\|_2 = \frac{n}{N}\|B - \mathrm{D}_B\|_2.$$

Moreover, the variance is

$$\operatorname{Var}(Y_i) = \frac{2}{N^2}\|B - \mathrm{D}_B\|_F^2.$$

Plugging these values into Hoeffding's and Bernstein's inequalities immediately gives the following tail bounds for trace estimates with Rademacher random variables.

Corollary 9.21. *For a symmetric matrix B we have*

$$\mathbb{P}(|\operatorname{tr}_N^{\mathsf{R}}(B) - \operatorname{tr}(B)| \geq \varepsilon) \leq 2\exp\left(-\frac{\varepsilon^2 N}{2n^2\|B - \mathrm{D}_B\|_2^2}\right).$$

Corollary 9.22. *For a symmetric matrix B we have*

$$\mathbb{P}(|\operatorname{tr}_N^{\mathsf{R}}(B) - \operatorname{tr}(B)| \geq \varepsilon) \leq 2\exp\left(-\frac{\varepsilon^2 N}{4\|B - \mathrm{D}_B\|_F^2 + \frac{4}{3}\varepsilon n\|B - \mathrm{D}_B\|_2}\right).$$

Clearly, a disadvantage of these bounds is the explicit dependence of the denominator on n or n^2, which does not appear in Corollary 9.16.

In analogy to the Gaussian case, the following lemma shows that a potential linear dependence of N on n cannot be avoided in general.

Lemma 9.23. *Let n be even and consider the traceless matrix $B = \begin{bmatrix} & & & 1 & 1 \\ & & 1 & & \\ & \iddots & & & \\ 1 & & & & \end{bmatrix}$. Then*

$$\mathbb{P}\left(|\operatorname{tr}_N^{\mathsf{R}}(B)| \leq \varepsilon\right) \leq \varepsilon\sqrt{\frac{N}{\pi n}}$$

for every $\varepsilon > 0$.

Proof. We first note that $\operatorname{tr}_N^{\mathsf{R}}(B) = \frac{2}{N}\sum_{i=1}^{nN/2} Z_i$ with independent Rademacher random variables Z_i. In turn, $\mathbb{P}\left(|\operatorname{tr}_N^{\mathsf{R}}(B)| \leq \varepsilon\right) = \mathbb{P}\left(\left|\sum_{i=1}^{nN/2} Z_i\right| \leq \frac{N\varepsilon}{2}\right)$ equals the probability that the number of variables satisfying $Z_i = 1$ is at least $\frac{n-\varepsilon}{4}N$ and at most $\frac{n+\varepsilon}{4}N$. Therefore,

$$\begin{aligned}\mathbb{P}\Big(|\operatorname{tr}_N^{\mathsf{R}}(B)| \leq \varepsilon\Big) &= \frac{1}{2^{nN/2}} \sum_{i=\lceil \frac{n-\varepsilon}{4}N\rceil}^{\lfloor \frac{n+\varepsilon}{4}N\rfloor} \binom{nN/2}{i} \leq \frac{N\varepsilon}{2} \cdot \frac{1}{2^{nN/2}} \cdot \binom{nN/2}{nN/4} \\ &\leq \frac{N\varepsilon}{2} \cdot \frac{2}{\sqrt{\pi n N}} = \varepsilon\sqrt{\frac{N}{\pi n}},\end{aligned}$$

where we used the inequality $\frac{1}{2^{2k}}\binom{2k}{k} \leq \frac{1}{\sqrt{\pi k}}$. □

We do not report a figure analogous to Figure 9.1 because the observed errors are very similar to the Gaussian case.

For SPSD matrices, a relative error estimate follows from Corollary 9.16 similarly to what has been discussed in Remark 9.15 for Gaussian vectors.

Corollary 9.24. *For a nonzero SPSD matrix B, we have*

$$\mathbb{P}\left(\frac{|\operatorname{tr}_N^{\mathsf{R}}(B) - \operatorname{tr}(B)|}{\operatorname{tr}(B)} \geq \varepsilon\right) \leq \delta \quad \textit{for} \quad N \geq \frac{8}{\varepsilon^2}(1+\varepsilon)\gamma \log\frac{2}{\delta}, \quad \textit{where} \quad \gamma := \frac{\|B\|_2}{\operatorname{tr}(B)}.$$

Proof. First of all, it is immediate that $\|B - \mathrm{D}_B\|_F \leq \|B\|_F$. As shown, e.g., in [29, Theorem 4.1], the same holds for the spectral norm when B is SPSD. For convenience, we provide a short proof: For every $y \in \mathbb{R}^n$ it holds that

$$|y^T(B - \mathrm{D}_B)y| \leq \max\{y^T By, y^T \mathrm{D}_B y\} \leq \max\{\|B\|_2, \|\mathrm{D}_B\|_2\} \leq \|B\|_2,$$

where the first inequality uses that both $y^T By$ and $y^T \mathrm{D}_B y$ are nonnegative. By taking the maximum with respect to all vectors of norm 1 one obtains $\|B - \mathrm{D}_B\|_2$ on the left-hand side, which shows that it is bounded by $\|B\|_2$. Now, the claimed result follows from Corollary 9.16 using the arguments of Remark 9.15. □

Corollary 9.24 improves the result from [162, Theorem 1], which requires $N \geq \frac{6}{\varepsilon^2}\log\frac{2}{\delta}$; a lower bound that does not improve as γ decreases.

9.3 Numerical examples

9.3.1 Triangle counting

To illustrate the estimates from Theorem 9.12 and Corollary 9.16, we compare them with the convergence of the Hutchinson trace estimation using Gaussian and Rademacher vectors for an example from [7, 140]. The number of triangles in an undirected graph is equal to $\frac{1}{6}\operatorname{tr}(A^3)$ where A is the (usually indefinite) adjacency matrix. Note that the quadratic forms $X^T A^3 X$ can be evaluated exactly using two matrix-vector multiplications. We consider an arXiv collaboration network with $n = 5\,242$ nodes and $48\,260$ triangles taken from https://snap.stanford.edu/data/ca-GrQc.html.

We estimate $\operatorname{tr}(A^3)$ using $N = 2, 2^2, 2^3, \ldots, 2^{11}$ samples. For each value of N we performed $1\,000$ experiments and discarded the 5% worst approximations in order to

estimate an error bound that holds with probability 95%. The obtained results are represented by the shaded regions in Figure 9.2 and match the obtained bounds fairly well, especially for Gaussian vectors.

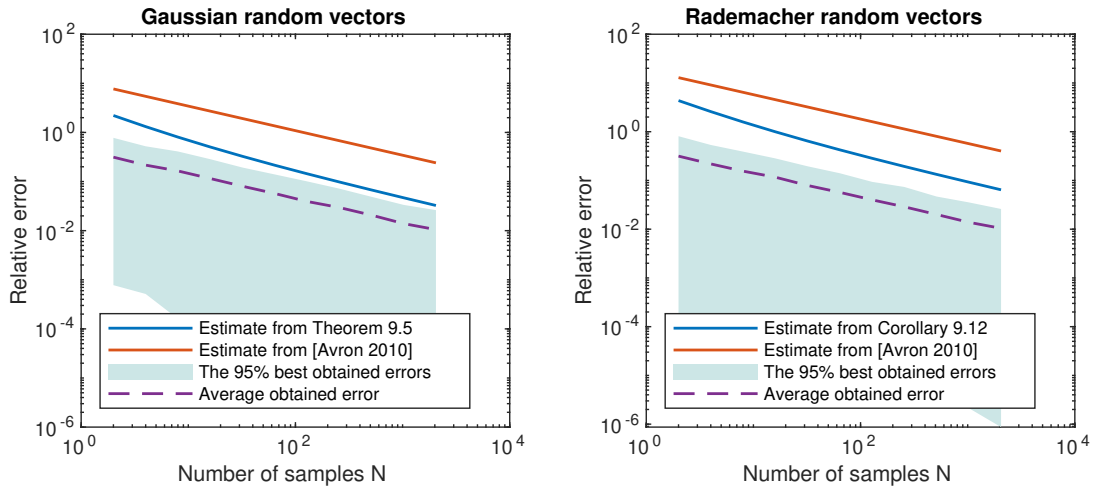


Figure 9.2 – Estimation of $\operatorname{tr}(A^3)$ *with Gaussian and Rademacher vectors for the matrix from Section 9.3.1. Error bounds from Theorem 9.12, Corollary 9.16, and [7] for failure probability* $\delta = 0.05$ *compared with the observed error.*

Figure 9.3 shows the empirical failure probability $\mathbb{P}(|\operatorname{tr}_N(A^3) - \operatorname{tr}(A^3)| \geq \varepsilon)$ with $\varepsilon = \frac{1}{10}\operatorname{tr}(A^3)$ using 1 000 experiments for $N = 2, 2^2, 2^3, \ldots, 2^{11}$ (blue and red lines). The vertical purple and yellow lines are the estimated number of samples needed to achieve failure probability $\delta = 0.05$ from Theorem 9.12 and Corollary 9.16, respectively.

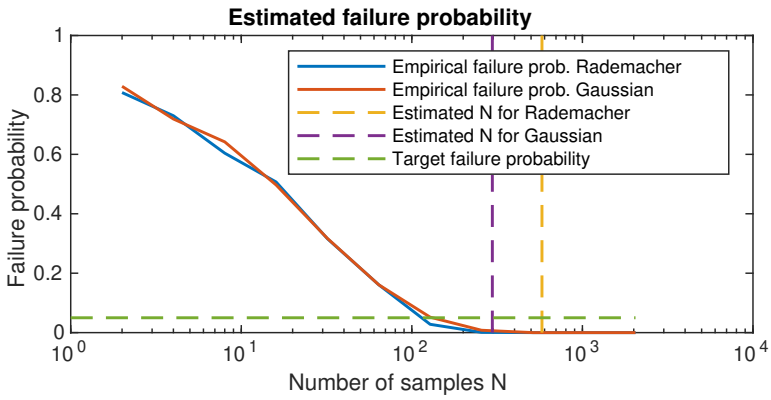


Figure 9.3 – Number of samples needed to attain error $\varepsilon = \frac{1}{10}\operatorname{tr}(A^3)$ *with failure probability* 5% *for Section 9.3.1. Empirical failure probability vs. bounds from Theorem 9.12 and Corollary 9.16.*

9.3.2 Comparison of estimates for indefinite matrices with Rademacher vectors

We consider the Hutchinson trace estimator with Rademacher vectors applied to two symmetric indefinite matrices: A is created by $A = \texttt{randn(2000)}; A = A + A^T$ and has stable rank $\rho(A) \approx 500$; B has eigenvalues $1, \frac{1}{2^2}, \dots, \frac{1}{1000^2}, -\frac{1}{1001^2}, \dots, -\frac{1}{2000^2}$ and has stable rank $\rho(B) \approx 1.08$. We compare the estimates coming from Hoeffding's and Bernstein's inequalities (Corollaries 9.21 and 9.22), the result in [7], and our result (Corollary 9.16), for $\delta = 0.05$. Moreover, we estimate empirically the minimum value of ε such that $\mathbb{P}(|\operatorname{tr}_N^{\mathsf{R}}(A) - \operatorname{tr}(A)| \geq \varepsilon) \leq 0.05$ in the following way: For each value of N from 1 to $10,000$ we run the Hutchinson's estimator 100 times, discard the worst 5% and plot the maximum error of the remaining estimates.

The results are shown in Figure 9.4. Corollary 9.16 gives the most accurate bound. Note that the estimate coming from Bernstein's inequality seems to have a different convergence rate with respect to all the other ones. This is due to the factor n in such estimate, and for *very large* values of the number of samples N the rate is actually $\mathcal{O}(1/\sqrt{N})$; however, it does not make sense to consider a value of N which is larger than the matrix dimension.

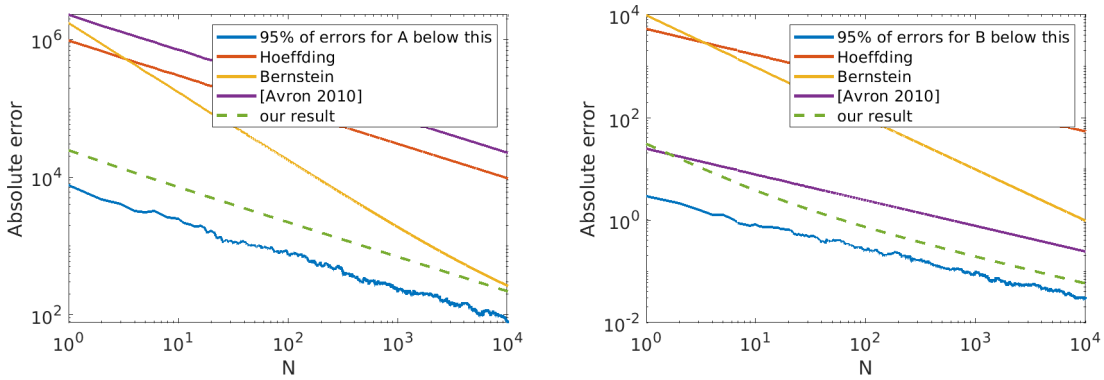


Figure 9.4 – Comparison of different estimates for the indefinite matrices A (left) and B (right) from Section 9.3.2.

9.3.3 An SPD example

We consider the Hutchinson trace estimator with Rademacher vectors applied to two SPD matrices: A has eigenvalues $d = \texttt{rand}(1000, 1)$ and $\gamma \approx 0.002$, B has eigenvalues

$1, \frac{1}{2^2}, \frac{1}{3^2}, \ldots, \frac{1}{1000^2}$ and $\gamma \approx 0.6$. For each value of N from 1 to 10,000 we run the Hutchinson trace estimator 100 times, discard the worst 5% and plot the maximum error of the remaining estimates in Figure 9.5 with a solid line; this corresponds to the accuracy that we get – empirically – with failure probability $\delta = 0.05$. We compare this with the result in [162] (black line) and the results from Corollary 9.24 (pink and blue dotted lines). The decay as $\frac{1}{\sqrt{N}}$ is correctly captured by both estimates, but the constant in front of it depends on γ.

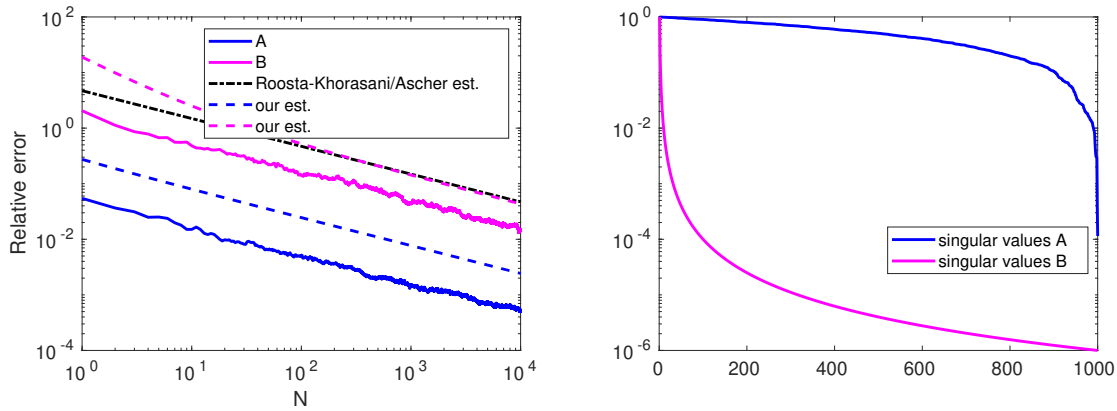


Figure 9.5 – Results for the SPD matrices A (left) and B (right) from Section 9.3.3.

9.4 Lanczos method to approximate quadratic forms

Let us now consider the problem of estimating the log-determinant through $\log(\det(A)) = \operatorname{tr}(\log(A))$, or more generally the problem of computing the trace of $f(A)$ for an analytic function f.

Applying the Hutchinson trace estimator to $\operatorname{tr}(f(A))$ requires the (approximate) computation of the quadratic forms $x^T f(A)x$ for fixed vectors $x \in \mathbb{R}^n$. As mentioned in Section 8.3, we use the Lanczos method, Algorithm 9.1, for this purpose.

For theoretical considerations, it is helpful to view the quadratic form as an integral. For this purpose, we consider the spectral decomposition $A = Q\Lambda Q^T$, $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, with $\lambda_{\min} = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n = \lambda_{\max}$. Then

$$x^T f(A)x = \mathbb{I} := \int_{\lambda_{\min}}^{\lambda_{\max}} f(\lambda)\, \mathrm{d}\mu(\lambda),$$

Algorithm 9.1 Lanczos method to approximate quadratic form $x^T f(A)x$

Input: Matrix $A \in \mathbb{R}^{n\times n}$, nonzero vector $x \in \mathbb{R}^n$, number of iterations m
Output: Approximation of $x^T f(A)x$

```
Initialize u_1 ← x/‖x‖_2 and β_0 ← 0
for i = 1, ..., m do
    α_i ← u_i^T A u_i
    r_i ← A u_i − α_i u_i − β_{i−1} u_{i−1}
    β_i ← ‖r_i‖_2
    u_{i+1} ← r_i/β_i
end for
```

$$T_m \leftarrow \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \beta_{m-1} & \alpha_m \end{bmatrix}$$

Return $\|x\|_2^2 \cdot e_1^T f(T_m) e_1$

with the piecewise constant measure

$$\mu(\lambda) := \sum_{i=1}^{n} z_i^2 \chi_{[\lambda_i,\infty)}(\lambda), \quad z := Q^T x, \tag{9.13}$$

where χ denotes the indicator function. It is well known [85, Theorem 6.2] that the approximation $\mathbb{I}_m$ returned by the m-points Gaussian quadrature rule applied to I is identical to the approximation returned by m steps of the Lanczos method:

$$\mathbb{I}_m := \|x\|_2^2 \cdot e_1^T f(T_m) e_1.$$

To bound the error $|\mathbb{I} - \mathbb{I}_m|$, the analysis in [183] proceeds by using existing results on the polynomial approximation error of analytic functions. Although our analysis is along the same lines, it differs in a key technical aspect; we derive and use an improved error bound for the approximation of the logarithm; see Corollary 9.29. We have also noted two minor issues in [183]; see the proof of Theorem 9.25 and the remark after Corollary 9.26 for details.

Theorem 9.25. *Let $f : [-1, 1] \to \mathbb{R}$ admit an analytic continuation to a Bernstein ellipse $\mathcal{E}_{r_0}$ with foci ± 1 and elliptical radius r_0. For $1 < r < r_0$, let M_r be the maximum of $|f(z)|$*

on $\mathcal{E}_r$. Then

$$|\mathbb{I} - \mathbb{I}_m| \leq \|x\|_2^2 \cdot \frac{4M_r}{1 - r^{-1}} r^{-2m}.$$

Proof. As in [183], this result follows directly from bounds on the polynomial approximation error of analytic functions via Chebyshev expansion, combined with the fact that m-points Gaussian quadrature is exact for polynomials up to degree $2m - 1$. However, the proof of [183, Theorem 4.2] uses an extra ingredient, which seems to be wrong. It claims that the integration error for odd-degree Chebyshev polynomials is zero thanks to symmetry. While this fact is indeed true for the standard Lebesgue measure, it does not hold for the measure (9.13). In turn, one obtains the slightly worse factor $1 - r^{-1}$ in the denominator, compared to the factor $1 - r^{-2}$ that would have been obtained from [183, Theorem 4.2] translated into our setting. □

The affine linear transformation

$$\varphi : [\lambda_{\min}, \lambda_{\max}] \to [-1, 1], \qquad x \mapsto \frac{2}{\lambda_{\max} - \lambda_{\min}} t - \frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}},$$

is used to map an interval $[\lambda_{\min}, \lambda_{\max}]$ containing the eigenvalues of A to the interval $[-1, 1]$ of Theorem 9.25. Defining $g := f \circ \varphi^{-1}$, one has

$$x^T g(\varphi(A))x = x^T f(A)x, \quad e_1^T g(\varphi(T_m))e_1 = e_1^T f(T_m)e_1. \tag{9.14}$$

By its shift and scaling invariance, the Lanczos method with g, $\varphi(A)$, and x returns the approximation $e_1^T g(\varphi(T_m))e_1$. This allows us to apply Theorem 9.25. Combined with the relations (9.14), the following result is obtained.

Corollary 9.26. *With the notation introduced above, it holds that*

$$\left| x^T f(A)x - \|x\|_2^2 \cdot e_1^T f(T_m)e_1 \right| \leq \|x\|_2^2 \cdot \frac{4M_r}{1 - r^{-1}} r^{-2m},$$

Note that M_r is the maximum of g on $\mathcal{E}_r$, which is equal to the maximum of f on the transformed ellipse with foci $\lambda_{\min}, \lambda_{\max}$, and elliptical radius $(\lambda_{\max} - \lambda_{\min})r/2$. The result of Corollary 9.26 differs from the corresponding result in [183, page 1087], which features an additional, erroneous factor $(\lambda_{\max}(A) - \lambda_{\min}(A))/2$.

Before addressing the special case of the logarithm, we present an elementary result,

which will be needed in the proof of Corollary 9.29.

Lemma 9.27. *Consider a circle in the complex plane with center $a \in \mathbb{R}^+$, $a > 1$ and radius b such that $b^2 = a^2 - 1$. Then the maximum absolute value of the logarithm on this circle is attained on the real axis.*

Proof. We consider the functions $\ell : [0, \pi] \to \mathbb{C}$ and $f : [0, \pi] \to \mathbb{R}$ given by

$$\ell(t) := \log(a + b\cos(t) + ib\sin(t)), \qquad f(t) := |\ell(t)|^2.$$

We will prove that f has two local maxima at $t = 0$ and $t = \pi$ and one local minimum. This is sufficient for the conclusion, because the problem is symmetric with respect to the real axis. Denoting by

$$r(t) := \sqrt{a^2 + b^2 + 2ab\cos(t)}, \qquad \theta(t) := \arctan \frac{b\sin(t)}{a + b\cos(t)},$$

we have $\ell(t) = \log\left(r(t)\exp(i\theta(t))\right) = \log(r(t)) + i\theta(t)$ and its derivative is

$$\begin{aligned}\ell'(t) &= \frac{-b\sin(t) + ib\cos(t)}{a + b\cos(t) + ib\sin(t)} = \frac{b}{r(t)^2}(-\sin(t) + \cos(t))(a + b\cos(t) - ib\sin(t)) \\ &= \frac{b}{r(t)^2}\left(-a\sin(t) + i(b + a\cos(t))\right).\end{aligned}$$

Therefore we have

$$f'(t) = 2\mathrm{Re}(\ell'(t) \cdot \overline{\ell(t)}) = \frac{2b}{r(t)^2}\left(-a\sin(t)\log(r(t)) + (b + a\cos(t))\theta(t)\right).$$

Note that $t = 0$, $t = \pi$ and $t = \arccos\left(-\frac{b}{a}\right)$ are zeros of f'. To prove that 0 and π are local maxima and $t = \arccos\left(-\frac{b}{a}\right)$ is a local minimum it is sufficient to prove that

$$\begin{cases} f'(t) < 0 & \text{for } t \in \mathcal{I}_1 := \left(0, \arccos\left(-\frac{b}{a}\right)\right); \\ f'(t) > 0 & \text{for } t \in \mathcal{I}_2 := \left(\arccos\left(-\frac{b}{a}\right), \pi\right). \end{cases}$$

Now consider the function $g : [-1, 1] \to \mathbb{R}$ given by

$$g(t) := -a\log\left(\sqrt{a^2 + b^2 + 2abt}\right) + b\frac{b + at}{a + bt} = -\frac{a}{2}\log(a^2 + b^2 + 2abt) + b\frac{b + at}{a + bt}.$$

As $\arctan(x) \leq x$ for all $x \geq 0$ with equality only for $x = 0$, $\sin(t) > 0$ on $\mathcal{I}_1 \cup \mathcal{I}_2$,

$b + a\cos(t) > 0$ on $\mathcal{I}_1$, $b + a\cos(t) < 0$ on $\mathcal{I}_2$, and $\frac{b\sin(t)}{a+b\cos(t)} > 0$ on $\mathcal{I}_1 \cup \mathcal{I}_2$, we have

$$\begin{cases} f'(t) < \frac{2b\sin(t)}{r(t)^2} \cdot g(\cos(t)) & \text{for } t \in \mathcal{I}_1; \\ f'(t) > \frac{2b\sin(t)}{r(t)^2} \cdot g(\cos(t)) & \text{for } t \in \mathcal{I}_2. \end{cases} \tag{9.15}$$

We show that the function g is decreasing: its derivative is

$$g'(t) = b\left(-\frac{a^2}{a^2+b^2+2abt} + \frac{1}{(a+bt)^2}\right)$$

and we have

$$g'(t) \leq 0 \Leftrightarrow a^2b^2t^2 + 2t(a^3b - ab) + a^4 - a^2 - b^2 \geq 0.$$

The latter expression is a convex parabola which has its minimum in $t = -\frac{a^3b-ab}{a^2b^2} = -\frac{b}{a}$, for which we have $g'\left(-\frac{b}{a}\right) = 0$. Therefore $g'(t) \leq 0$ for all $t \in [-1, 1]$ so $g(t)$ is decreasing. Moreover, $g\left(-\frac{b}{a}\right) = 0$, so $g(t) \geq 0$ in $\left[-1, -\frac{b}{a}\right]$ and $g(t) \leq 0$ in $\left[-\frac{b}{a}, 1\right]$, which implies (9.15). □

Corollary 9.28. *Consider an ellipse $\mathcal{E}$ in the open right-half complex plane, with foci on the real axis. Then the maximum absolute value of the logarithm on this ellipse is attained on the real axis.*

Proof. Let $0 < \alpha < \beta$ be the two intersections of the ellipse with the real axis. If $|\log\alpha| \geq |\log\beta|$ then $\mathcal{E}$ is contained in the circle $\mathcal{C}_1$ of center $a := \frac{1}{2}\left(\frac{1}{\alpha} + \alpha\right)$ and radius $b := \frac{1}{2}\left(\frac{1}{\alpha} - \alpha\right) = \sqrt{a^2-1}$, and $\mathcal{E}$ is tangent to $\mathcal{C}_1$ in α; otherwise $\mathcal{E}$ is contained in the circle $\mathcal{C}_2$ of center $a := \frac{1}{2}\left(\beta + \frac{1}{\beta}\right)$ and radius $b := \frac{1}{2}\left(\beta - \frac{1}{\beta}\right) = \sqrt{a^2-1}$, and $\mathcal{E}$ is tangent to $\mathcal{C}_2$ in β. In both cases, the result follows from Lemma 9.27. □

By specializing Corollary 9.26 to the logarithm we obtain the following result.

Corollary 9.29. *Let $A \in \mathbb{R}^{n\times n}$ be SPD with condition number $\kappa(A)$, $f \equiv \log$ and $x \in \mathbb{R}^n\backslash\{0\}$. Then the error of the Lanczos method after m steps satisfies*

$$|x^T\log(A)x - \|x\|_2^2 \cdot e_1^T\log(T_m)e_1| \leq c_A\|x\|_2^2\left(\frac{\sqrt{\kappa(A)+1}-1}{\sqrt{\kappa(A)+1}+1}\right)^{2m}.$$

where $c_A := 2(\sqrt{\kappa(A)+1}+1)\log(2\kappa(A))$.

Proof. The proof consists of applying Corollary 9.26 to a rescaled matrix. More specifically, we choose $B := \lambda A$ with $\lambda := 1/(2\lambda_{\min}) > 0$. The tridiagonal matrix returned by the Lanczos method with A replaced by B satisfies $T_m^B = \lambda T_m$. Together with the identity $\log(\lambda A) = \log \lambda I_n + \log(A)$, this implies

$$x^T \log(A)x - \|x\|_2^2 \cdot e_1^T \log(T_m)e_1 = x^T \log(B)x - \|x\|_2^2 \cdot e_1^T \log(T_m^B)e_1.$$

Note that the smallest/largest eigenvalues of B are given by $1/2$ and $\kappa(A)/2$, respectively. Applying Corollary 9.26 to B with[1] $r := \frac{\sqrt{\kappa(A)+1}+1}{\sqrt{\kappa(A)+1}-1}$ thus gives

$$|x^T \log(A)x - \|x\|_2^2 \cdot e_1^T \log(T_m)e_1| \leq \|x\|_2^2 \cdot \frac{4M_r}{1-r^{-1}} r^{-2m}.$$

The constant M_r is the maximum absolute value of the logarithm on the ellipse with foci $1/2$ and $\kappa(A)/2$ that intersects the real axis at $\alpha := \frac{1}{2\kappa(A)}$ and $\beta := \frac{\kappa(A)^2+\kappa(A)-1}{2\kappa(A)}$. By Corollary 9.28, $M_r = |\log(\alpha)| = \log(2\kappa(A))$, where we used $\alpha \leq 1/\beta \leq 1$. Noting that

$$\frac{4M_r}{1-r^{-1}} = 2(\sqrt{\kappa(A)+1}+1)\log(2\kappa(A)) = c_A$$

concludes the proof. □

9.5 Combined bounds for determinant estimation

Combining Hutchinson trace estimation with the Lanczos method, we obtain the following (stochastic) estimate for $\log(\det(A))$:

$$\mathrm{est}_{N,m}^{\mathsf{G},\mathsf{R}} := \sum_{i=1}^{N} \|X^{(i)}\|_2^2 \cdot e_1^T \log(T_m^{(i)})e_1,$$

where $X^{(1)}, \ldots, X^{(N)}$ are independent Gaussian or Rademacher random vectors and $T_m^{(i)}$ is the tridiagonal matrix obtained from the Lanczos method with starting vector $X^{(i)}/\|X^{(i)}\|_2$. By combining the results obtained so far, we now derive new bounds on the number of samples and number of Lanczos steps needed to ensure an approximation error of at most ε (with high probability).

[1] In fact, it is possible to choose $r = \frac{\sqrt{\kappa(A)+\varepsilon}+1}{\sqrt{\kappa(A)+\varepsilon}-1}$ for arbitrary $\varepsilon > 0$.

9.5.1 Standard Gaussian random vectors

Theorem 9.30. *Suppose that the following holds for N (number of Gaussian probe vectors) and m (number of Lanczos steps per probe vector):*

(i) $N \geq 16\varepsilon^{-2}(\rho_{\log}\|\log(A)\|_2^2 + \varepsilon\|\log(A)\|_2)\log\frac{4}{\delta}$, *where* $\rho_{\log}$ *denotes the stable rank of* $\log(A)$*;*

(ii) $m \geq \frac{\sqrt{\kappa(A)+1}}{4}\log\left(4\varepsilon^{-1}n^2(\sqrt{\kappa(A)+1}+1)\log(2\kappa(A))\right)$.

If, additionally, $n \geq 2$ *and* $N \leq \frac{\delta}{2}\exp\left(\frac{n^2}{16}\right)$ *then* $\mathbb{P}(|\operatorname{est}_{N,m}^{\mathsf{G}} - \log\det(A)| \geq \varepsilon) \leq \delta$.

Proof. For a Gaussian vector X, the squared norm $\|X\|_2^2$ is a Chi-squared random variable with n degrees of freedom. Therefore, by [127, Lemma 1] we have

$$\mathbb{P}(\|X\|_2^2 \geq n + 2\sqrt{nt} + 2t) \leq \exp(-t)$$

for every $t > 0$. For $t = \log\frac{2N}{\delta}$, the additional assumptions of the theorem imply

$$n + 2\sqrt{nt} + 2t \leq n + 2\sqrt{n}\cdot\frac{n}{4} + 2\cdot\frac{n^2}{16} < n^2,$$

and therefore $\mathbb{P}(\|X\|_2^2 \geq n^2) \leq \frac{\delta}{2N}$. By the union bound, it holds that

$$\mathbb{P}\left(\text{exists } i \in \{1,\dots,N\} \text{ s.t. } \|X^{(i)}\|_2^2 \geq n^2\right) \leq \frac{\delta}{2}. \tag{9.16}$$

Corollary 9.29, together with condition (ii) and (9.16) imply that $|\operatorname{est}_{N,m}^{\mathsf{G}} - \operatorname{tr}_N^{\mathsf{G}}(\log(A))| \leq \frac{\varepsilon}{2}$ holds with probability at least $1-\delta/2$, where we also used that

$$\log\left(\frac{\sqrt{\kappa(A)+1}+1}{\sqrt{\kappa(A)+1}-1}\right) \geq \frac{2}{\sqrt{\kappa(A)+1}}.$$

Applying Theorem 9.12 to the matrix $\log(A)$, for which $\|\log(A)\|_F^2 = \rho_{\log}\|\log(A)\|_2^2$, we find that $|\operatorname{tr}_N^{\mathsf{G}}(\log(A)) - \log\det(A)| \leq \frac{\varepsilon}{2}$ holds with probability at least $1-\delta/2$. The proof is concluded by applying the triangle inequality. □

9.5.2 Rademacher random vectors

Theorem 9.31. *Suppose that the following holds for N (number of Rademacher probe vectors) and m (number of Lanczos steps per probe vector):*

(i) $N \geq 32\varepsilon^{-2}\left(\rho_{\mathrm{logd}}\|\log(A) - \mathrm{D}_{\log(A)}\|_2^2 + \frac{\varepsilon}{2}\|\log(A) - \mathrm{D}_{\log(A)}\|_2\right)\log\frac{2}{\delta}$, *where* ρ_{logd} *denotes the stable rank of* $\log(A) - \mathrm{D}_{\log(A)}$ *and* $\mathrm{D}_{\log(A)}$ *is the diagonal matrix containing the diagonal entries of* $\log(A)$*;*

(ii) $m \geq \frac{\sqrt{\kappa(A)+1}}{4}\log\left(4\varepsilon^{-1}n(\sqrt{\kappa(A)+1}+1)\log(2\kappa(A))\right)$.

Then $\mathbb{P}(|\operatorname{est}_{N,m}^{\mathsf{R}} - \log\det(A)| \geq \varepsilon) \leq \delta$.

Proof. Using Corollary 9.26 and the fact that Rademacher random vectors have norm $\sqrt{n}$, the bound $\left|\operatorname{est}_{N,m}^{\mathsf{R}} - \operatorname{tr}_N^{\mathsf{R}}(\log(A))\right| \leq \frac{\varepsilon}{2}$ holds if

$$m \geq \frac{1}{2}\log\left(4\varepsilon^{-1}n(\sqrt{\kappa(A)+1}+1)\log(2\kappa(A))\right)\Big/\log\left(\frac{\sqrt{\kappa(A)+1}+1}{\sqrt{\kappa(A)+1}-1}\right).$$

Because of $\log\left(\frac{\sqrt{\kappa(A)+1}+1}{\sqrt{\kappa(A)+1}-1}\right) \geq \frac{2}{\sqrt{\kappa(A)+1}}$, condition (ii) ensures that this inequality is satisfied.

Applying Corollary 9.16 to $\log(A)$ and with ε replaced by $\varepsilon/2$, immediately shows

$$|\operatorname{tr}_N^{\mathsf{R}}(\log(A)) - \log\det(A)| \leq \frac{\varepsilon}{2} \tag{9.17}$$

with probability at least $1-\delta$ if condition (i) is satisfied. The proof is concluded by applying the triangle inequality. □

Comparison with an existing result. To compare Theorem 9.31 with an existing result from [183], it is helpful to first derive a simpler (but usually stronger) condition on N.

Lemma 9.32. *The statement of Theorem 9.31 holds with condition (i) replaced by* $N \geq 8\varepsilon^{-2}\left(n\log^2\kappa(A) + 2\varepsilon\log\kappa(A)\right)\log\frac{2}{\delta}$.

Proof. We set $B := \lambda A$ with $\lambda := 1/\sqrt{\lambda_{\min}(A)\lambda_{\max}(A)}$ and note that

$$\operatorname{tr}_N^{\mathsf{R}}(\log(A)) - \log\det(A) = \operatorname{tr}_N^{\mathsf{R}}(\log(\lambda A)) - \log\det(\lambda A).$$

Using $\lambda_{\max}(B) = \sqrt{\kappa(A)}$, $\lambda_{\min}(B) = 1/\sqrt{\kappa(A)}$, and $\kappa(B) = \kappa(A)$, we obtain

$$\begin{aligned}
&\|\log(B) - \mathrm{D}_{\log(B)}\|_2 \leq 2\|\log(B)\|_2 = \log\kappa(A);\\
&\|\log(B) - \mathrm{D}_{\log(B)}\|_F^2 \leq \|\log(B)\|_F^2 = \rho(\log(B))\frac{\log^2\kappa(A)}{4} \leq \frac{n}{4}\log^2\kappa(A).
\end{aligned}$$

An application of Corollary 9.16 to $\log(B)$ therefore yields (9.17) with probability at least $1-\delta$ for $N \geq 8\varepsilon^{-2}\Big(n\log^2\kappa(A) + 2\varepsilon\log\kappa(A)\Big)\log\frac{2}{\delta}$. □

Correcting for the two minor erratas explained above, the result from [183, Corollary 4.5] states that $\mathbb{P}(|\operatorname{est}_{N,m}^{\mathsf{R}} - \operatorname{tr}(\log(A))| \geq \varepsilon) \leq \delta$ holds if

$$N \geq 24\varepsilon^{-2}n^2\left(\log(1+\kappa(A))\right)^2\log\frac{2}{\delta} \tag{9.18}$$

and

$$m \geq \frac{\sqrt{3\kappa(A)}}{4}\log\left(20\varepsilon^{-1}n\Big(\sqrt{2\kappa(A)+1}+1\Big)\log(2\kappa(A)+2)\right). \tag{9.19}$$

Compared to (9.18), Lemma 9.32 reduces the explicit dependence on the matrix size from n^2 to n, while the dependence of the bounds on $\kappa(A)$ is comparable. Let us stress that even a dependence on n does not compare favorably to simply computing the diagonal elements, but the bound from condition (i) of Theorem 9.31 can often be expected to be significantly better than the simplified bound of Lemma 9.32. Below we describe a situation in which the former only depends logarithmically on n. Condition (ii) of Theorem 9.31 improves (9.19) clearly but less drastically, roughly by a factor $\sqrt{3}$.

Implications of low stable rank. Let us consider a family of matrices $\{A_\ell\}$ of increasing dimensions $n_1 < \ldots < n_\ell < \ldots$, a fixed failure probability δ, and a fixed accuracy ε; the number of probe vectors required to get $\mathbb{P}(|\operatorname{tr}_N(\log(A_\ell)) - \operatorname{tr}(\log(A_\ell))| \geq \varepsilon) \leq \delta$ is proportional to $O(\rho_\ell\|\log(A_\ell)\|_2^2)$, where ρ_ℓ is the stable rank of $\log(A_\ell)$. In certain applications, including regularized kernel matrices (see, e.g., [43, 80]), the stable rank grows slowly when the matrix size increases. For such situations, our bounds lead to favorable implications. To illustrate this, let us consider matrices $A_\ell := I_{n_\ell} + B_\ell$,

where the eigenvalues satisfy $\lambda_i(B_\ell) \leq n_\ell C \alpha^i$ for some constants $C > 0$ and $0 < \alpha < 1$, for all $i \leq n_\ell$, such as in the discretization of a radial basis function kernel on a fixed domain [80]. In this case, $\rho_\ell = O(\log n_\ell)$. As a second example, if B_ℓ comes from a discretization of a Matérn kernel on a regular grid in a fixed domain, its eigenvalues satisfy $\lambda_i(B_\ell) \leq n_\ell C i^{-\beta}$ for some constants $C > 0$ and $\beta > 1$, for all $i \leq n_\ell$ [43]; the stable rank of $\log(A_\ell) = \log(I_{n_\ell} + B_\ell)$ is bounded by $\rho_{n_\ell} = O(n_\ell^{1/\beta})$. To apply Theorems 9.30 and 9.31 one also needs to take into account that, for both our examples, $\|\log(A_\ell)\|_2$ and $\kappa(A_\ell)$ grow proportionally to $\log(n_\ell)$ and n_ℓ, respectively. Finally, note that in practice one would consider $A_\ell = \sigma I_{n_\ell} + B_\ell$ with the regularization parameter σ chosen adaptively; see, e.g., [37].

9.6 Numerical experiments for log-determinant

To compare the results of Theorems 9.30 and 9.31 with the number of sample vectors N and Lanczos steps m (per sample) required to reach a fixed accuracy, we consider the matrices listed in Table 9.1. The matrix labeled as `thermo` is the `thermomec_TC` matrix contained in the University of Florida sparse matrix collection [52] and has been considered, for instance, in [33, 73, 183]. The matrix `lowrank` is defined in [167, 129] as

$$A = \sum_{j=1}^{40} \frac{10}{j^2} x_j x_j^T + \sum_{j=41}^{300} \frac{1}{j^2} x_j x_j^T,$$

where each x_j is a sparse vector of length 20 000 with approximately 2.5% uniformly distributed nonzero entries, generated with the MATLAB command `sprand`. The matrix `precip` is a two-dimensional Gaussian kernel matrix with length parameter $\gamma = 64$ and regularization parameter $\lambda = 0.008$ taken from [140], involving precipitation data from Slovakia [145]. As the matrices `thermo` and `lowrank` are too large for $\log(A)$ to be computed explicitly, the quantities $\|\log(A)\|_F$ and $\|\log(A) - \mathrm{D}_{\log(A)}\|_F$ are approximated by randomized trace estimation combined with the Lanczos method to estimate the diagonal elements of $\log(A)$.

For quadratic forms involving the logarithm, there is a relatively inexpensive way to obtain an upper bound on the error of the Lanczos method. As discussed in [10], Gauss quadrature always yields an upper bound for $x^T \log(A) x$, while Gauss-Lobatto quadrature always yields a lower bound. We fix $\delta = 0.1$ and for several values of ε we

Name	Size	Ref.	$\log\det(A)$	$\kappa(A)$	$\|\log(A)\|_F$	$\|\log(A) - \mathrm{D}_{\log(A)}\|_F$
`thermo`	102158	[52]	$-5.47 \cdot 10^5$	67.2	$1.72 \cdot 10^3$	122.8
`lowrank`	20000	[129]	89.4	1560	17.04	16.99
`precip`	6400	[140]	$-2.56 \cdot 10^4$	6738	357	157

Table 9.1 – Summary of the matrices used for log-determinant experiments.

investigate how many samples and Lanczos iterations are needed in practice. When approximating quadratic forms while aiming at accuracy ε, we stop the Lanczos method when the difference between upper and lower bound is less than $\varepsilon/2$. Starting from $N = 1$, we compute the empirical failure probability $\mathbb{P}(|\mathrm{est}_{N,m} - \log\det(A)| \geq \varepsilon)$; if this probability is larger than δ, we double the number of samples N and repeat.

The results for the three matrices from Table 9.1 are reported in Figures 9.6, 9.7, and 9.8. The left plots show, for the considered values of ε (which have been normalized by dividing them by the true $|\log\det(A)|$), the number of samples required to attain 90% success probability over 30 runs of the algorithm, versus the number of samples given by Theorems 9.30 and 9.31. The plots on the right show, for the same (normalized) values of ε, the average number of Lanczos steps required to reach accuracy $\varepsilon/2$ versus the number of Lanczos steps predicted by Theorems 9.30 and 9.31.

For `thermo`, the diagonal of $\log(A)$ is large relative to the rest of the matrix: $\|\log(A) - \mathrm{D}_{\log(A)}\|_F / \|\log(A)\|_F \approx 0.07$. Therefore, our bounds predict that Rademacher vectors perform much better than Gaussian vectors; this is indeed confirmed by Figure 9.6. The matrix A is well conditioned and, hence, the bounds correctly predict that the Lanczos method only needs relatively few iterations to attain good accuracy. For `lowrank`, Figure 9.7 shows that Rademacher and Gaussian vectors perform similarly. Although the condition number of A is $\kappa(A) \approx 1560$, the eigenvalues have a strong decay, and hence its adaptivity lets the Lanczos method perform much better than predicted by our bounds, see, e.g., [99] for a discussion. For `precip`, the ratio $\|\log(A) - \mathrm{D}_{\log(A)}\|_F / \|\log(A)\|_F \approx 0.44$ is reflected in Figure 9.8, showing that Rademacher vectors attain somewhat better accuracy. The condition number of A is high and there is no strong decay or gaps in the singular values; a relatively large number of Lanczos steps is necessary to obtain the desired accuracy when approximating the quadratic forms.

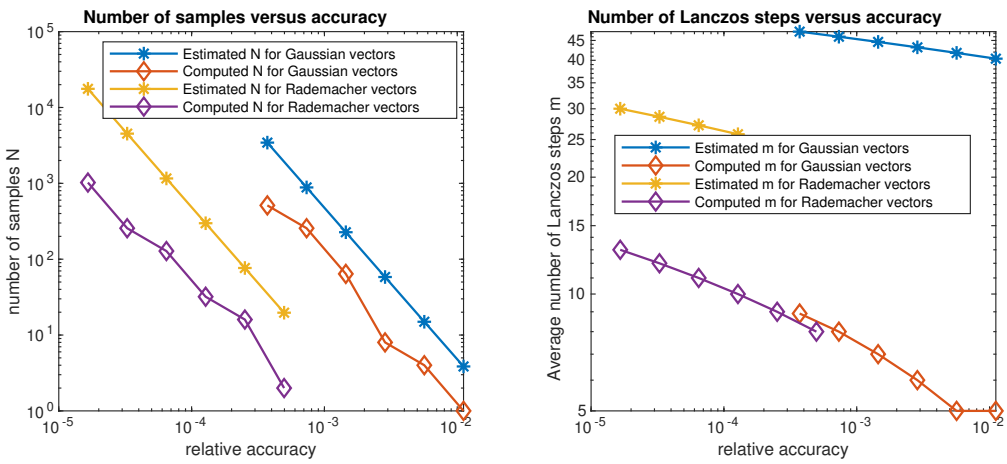


Figure 9.6 – Results for matrix `thermomec_TC` *from [52].*

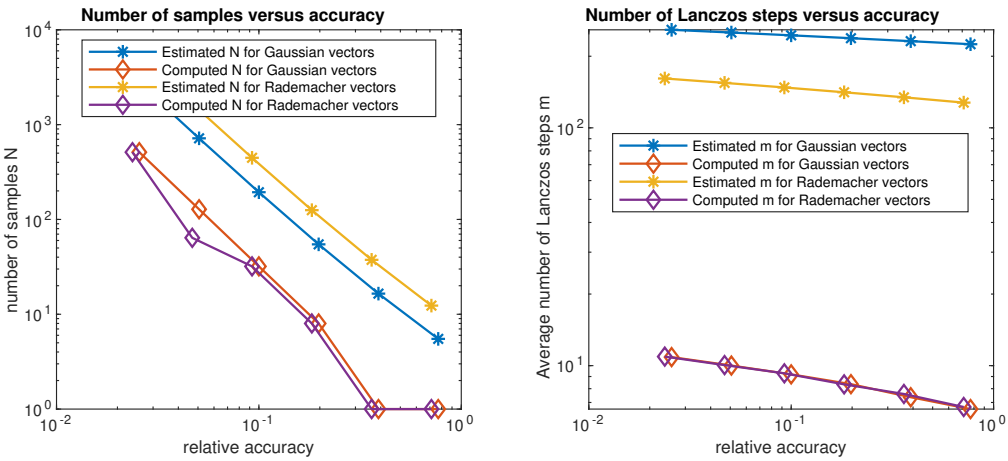


Figure 9.7 – Results for matrix `lowrank` *from [129].*

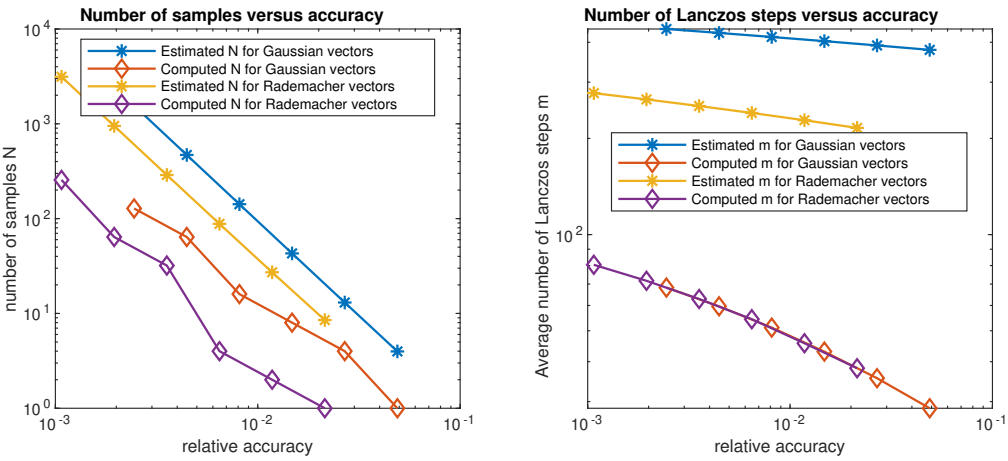


Figure 9.8 – Results for matrix `precip` *from [140].*

10 Conclusions and outlook

In Chapters 3 and 4 we have presented and analyzed deterministic algorithms for cross approximation, column subset selection, CUR approximation, and Tucker approximation of tensors. We started Chapter 3 by showing that the search for a $k \times k$ maximum volume submatrix can be restricted to principal submatrices for SPSD and DD matrices; a fact that appears intuitive but does not appear to be widely known. In fact, for DD matrices Theorem 3.4 is the first result providing a mathematical justification to this intuition. For cross approximation, Theorem 3.6 appears to be the first non-asymptotic error bound that holds for general matrices. Except for [106], previous results for cross approximation applied to matrices or functions [15, 180] are based on a step-by-step analysis of the error. In contrast, our technique takes a more global view and can, in turn, leverage existing results on the pivot growth in Gaussian elimination. As illustrated in Section 3.2.4, this can yield significant advantages.

A number of fundamental questions remain open. Most importantly, there is a mismatch between the derived error bounds and the known worst-case examples for ACA applied to DD and doubly DD matrices. Especially for DD matrices, the theoretical results feature an exponential growth in the L factor of an LU decomposition, but the worst known examples only show polynomial growth. This problem appears to be difficult to overcome and was encountered previously in the context of the error analysis of LDU factorizations [11, 61].

In Chapter 4, we have dealt with algorithms that guarantee a quasi-optimal low-rank approximation in the Frobenius norm. We have proposed several improvements

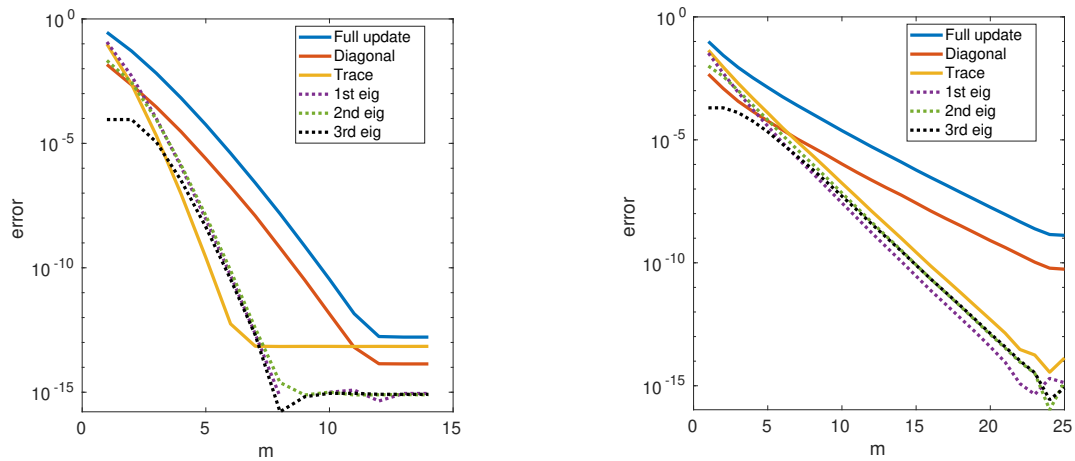
to the column selection algorithm by Deshpande and Rademacher [57]. The numerical experiments indicate that updating singular values (instead of characteristic polynomials) leads to numerical robustness, in the sense that the approximation error obtained in finite precision arithmetic is not affected unduly by roundoff error. We have also developed an extension of an existence result from [57] to produce a deterministic polynomial time algorithm that yields a cross approximation with a guaranteed polynomial error bound in the Frobenius norm. We have introduced a mechanism for stopping early the search for indices in column subset selection or cross approximation. Although relatively simple, this mechanism tremendously reduces the execution time for all examples tested.

A number of issues remain for future study, such as the numerical stability analysis of our algorithms. In particular, it would be desirable to study the numerical robustness of the cross approximation returned by Algorithm 4.3 with early stopping. Also, by combining early stopping with a more aggressive reuse of the SVD might lead to further complexity reduction, but a rigorous complexity analysis would require deeper understanding of early stopping, well beyond the limited scope of Lemma 4.1. Finally, we stress that the algorithms presented in Chapter 4 are intended for small to medium sized matrices and tensors. For large-scale data, the algorithms presented in this chapter need to be combined with other, possibly heuristic/randomized dimensionality reduction techniques.

In Chapter 7 we have proposed two new algorithms for computing matrix functions of structured matrices, based on a D&C paradigm. The algorithms have been tested on a wide range of examples of practical relevance that require to compute, for a medium- to large-scale matrix, the whole matrix function, its diagonal or its trace. The numerical results in Sections 7.2 and 7.4 demonstrate that, most of the time, the proposed methods outperform state-of-art techniques with respect to time consumption and offer a comparable accuracy. The convergence analysis of the splitting algorithm from Section 7.3 highlights stronger convergence properties for the entries located on the main diagonal, which applies also to non-Hermitian matrix arguments. The block diagonal splitting approach can, in principle, be applied to matrices arising from the discretization of two-dimensional partial differential equations. On the one hand, the bandwidth becomes much larger, on the other hand these matrices have additional sparsity structure that is not exploited by our algorithm. It would be interesting to explore whether there is a variant of Algorithm 7.2 that also covers this case efficiently.

We have also expanded the framework of low-rank updates of matrix functions [18, 17]

towards several directions. When polynomial Krylov subspaces are used, we have proved in Chapter 6 a convergence result related to polynomial approximation of the derivative of the function. In the Hermitian case, we have shown that the approximation of the trace of the update, computed by projection on the polynomial Krylov subspace, has a higher convergence rate with respect to the full update. We briefly mention an observation on low-rank updates that would require further study. When polynomial Krylov subspaces are used and the matrices A and R are symmetric, we noticed that the eigenvalues of the update seem to converge faster than the whole matrix function update $f(A+R)-f(A)$; see Figure 10.1 for an example. However, there does not seem to be a "clean" exactness result similar to Theorem 6.11, so other techniques should be used for proving results in this direction.



(a) Normalized random symmetric matrix A, normalized random update $R = bb^T$, $f = \exp$.

(b) Well conditioned SPD matrix A, normalized random update $R = bb^T$, $f = \sqrt{\cdot}$.

Figure 10.1 – We consider two examples of symmetric A and R with polynomial Krylov subspaces for approximating the update $f(A+R)-f(A)$. The solid lines denote the convergence of the errors $\|f(A+R)-f(A)-U_mX_m(f)V_m^T\|_F$, $\|\mathrm{diag}(f(A+R)-f(A)-U_mX_m(f)V_m^T)\|_2$, and $|\operatorname{tr}(f(A+R)-f(A)-U_mX_m(f)V_m^T)|$; the dotted lines denote the convergence of the three largest eigenvalues of the approximate update $U_mX_m(f)U_m^T$ to the three largest eigenvalues of $f(A+R)-f(A)$. Note that in both cases the eigenvalues are converging as fast as the trace, and faster than the full update and its diagonal.

In Chapter 9 we presented new tail bounds for the Hutchinson trace estimator applied to symmetric but indefinite matrices. These improve the results from [7] by lowering the number of samples required to get accuracy ε with failure probability δ by a factor which can be as large as n, the matrix size. We have then combined these bounds with an improved analysis of the Lanczos method for the computation of quadratic forms

$x^T \log(A)x$ to obtain results on the approximation of the log-determinant of SPD matrices, improving the results from [183].

The error of the Hutchinson trace estimator decreases as the inverse square root of the number of samples; this is a typical behavior of Monte Carlo algorithms. In the recent paper [102] it is shown that a multilevel Monte Carlo approach can give some advantages for trace estimation; the setting they consider, however, is not the same as ours, as they use Chebyshev approximations instead of Lanczos method. We conclude this thesis by touching on two further directions in which trace estimation algorithms can be improved; these also allow us to draw a connection between Part III and low-rank approximation (Part I) and rank-structured matrices (Part II).

A disadvantage of the Hutchinson trace estimator is that it does not take into account any structure of the matrix A. For example, if one needs to compute the trace of a matrix A which is approximately low-rank, we can obtain a good approximation by taking $\mathrm{tr}(A_k)$, where A_k is a low-rank approximation of A. The randomized SVD [101] allows us to compute a low-rank approximation of A via matrix-vector multiplications with random vectors. By combining this with the Hutchinson trace estimator, the recently introduced Hutch++ algorithm [140] ensures that $\mathcal{O}(\varepsilon^{-1})$ samples are sufficient to reach accuracy ε for an SPD matrix A. Our bounds for the Hutchinson trace estimator from Section 9.2 can also be useful in this context [157].

Another case in which trace estimation can be improved is when the matrix has some known (approximate) sparsity structure. As discussed in Part II, when A is banded or has some known sparsity structure in many cases $f(A)$ approximately preserves some structure. When estimating $\mathrm{tr}(f(A))$ this can (and should) be taken into consideration. Probing techniques have been developed for this aim; see, e.g., [19, 79, 174, 177]. The idea is that, if $f(A)$ has bandwidth b, computing b quadratic forms with carefully chosen vectors will allow us to compute the trace *exactly*. For a tridiagonal matrix, for instance, the vectors probing can be chosen to be $v_1 = (1, 0, 0, 1, 0, 0, 1, 0, 0, \ldots)^T$, $v_2 = (0, 1, 0, 0, 1, 0, 0, 1, 0, \ldots)^T$, $v_3 = (0, 0, 1, 0, 0, 1, 0, 0, 1, \ldots)^T$. In the more practical case in which $f(A)$ is only approximately banded such deterministic vectors will still give a good approximation of the trace. Probing can be combined, in principle, with the Hutchinson trace estimator by substituting the "ones" in the probing vectors with i.i.d. Rademacher random variables; our results from Chapter 9 can be applied to this case as well.

Bibliography

[1] R. Adamczak. The entropy method and concentration of measure in product spaces. Master's thesis, University of Warsaw and Vrije Universiteit van Amsterdam, 2003. Available at http://duch.mimuw.edu.pl/~radamcz/Old/Papers/master.pdf.

[2] R. H. Affandi, E. B. Fox, R. P. Adams, and B. Taskar. Learning the parameters of determinantal point process kernels. In *31st International Conference on Machine Learning, ICML 2014*, volume 4, pages 2967–2981, 2014.

[3] E. Anderson, Z. Bai, C. H. Bischof, S. Blackford, J. W. Demmel, J. J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. C. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, PA, third edition, 1999.

[4] J.-E. Andersson. Approximation of e^{-x} by rational functions with concentrated negative poles. *J. Approx. Theory*, 32(2):85–95, 1981.

[5] M. Arioli and I. S. Duff. Preconditioning linear least-squares problems by identifying a basis matrix. *SIAM J. Sci. Comput.*, 37(5):S544–S561, 2015.

[6] J. L. Aurentz, T. Mach, L. Robol, R. Vandebril, and D. S. Watkins. *Core-chasing Algorithms for the Eigenvalue Problem*, volume 13 of *Fundamentals of Algorithms*. SIAM, Philadelphia, PA, 2018.

[7] H. Avron. Counting triangles in large graphs using randomized matrix trace estimation. In *Workshop on Large-scale Data Mining: Theory and Applications*, 2010.

[8] H. Avron and S. Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *J. ACM*, 58(2):Art. 8, 17, 2011.

[9] F. R. Bach and M. I. Jordan. Predictive low-rank decomposition for kernel methods. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 33–40. ACM, 2005.

[10] Z. Bai, M. Fahey, and G. Golub. Some large-scale matrix computation problems. volume 74, pages 71–89. 1996. TICAM Symposium (Austin, TX, 1995).

[11] A. Barreras and J. M. Peña. Accurate and efficient LDU decompositions of diagonally dominant M-matrices. *Electron. J. Linear Algebra*, 24:152–167, 2012/13.

[12] R. P. Barry and R. K. Pace. Monte Carlo estimates of the log determinant of large sparse matrices. volume 289, pages 41–54. 1999. Linear algebra and statistics (Istanbul, 1997).

[13] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589, 2000.

[14] M. Bebendorf. *Hierarchical matrices. A means to efficiently solve elliptic boundary value problems*, volume 63 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2008.

[15] M. Bebendorf. Adaptive cross approximation of multivariate functions. *Constr. Approx.*, 34(2):149–179, 2011.

[16] B. Beckermann, J. Bisch, and R. Luce. Computing Markov functions of Toeplitz matrices. *arXiv preprint arXiv:2106.05098*, 2021.

[17] B. Beckermann, A. Cortinovis, D. Kressner, and M. Schweitzer. Low-rank updates of matrix functions II: rational Krylov methods. *SIAM J. Numer. Anal.*, 59(3):1325–1347, 2021.

[18] B. Beckermann, D. Kressner, and M. Schweitzer. Low-rank updates of matrix functions. *SIAM J. Matrix Anal. Appl.*, 39(1):539–565, 2018.

[19] C. Bekas, E. Kokiopoulou, and Y. Saad. An estimator for the diagonal of a matrix. *Appl. Numer. Math.*, 57(11-12):1214–1229, 2007.

[20] A. Belhadji, R. Bardenet, and P. Chainais. A determinantal point process for column subset selection. *J. Mach. Learn. Res.*, 21:1–62, 2020.

[21] M. Benzi and P. Boito. Decay properties for functions of matrices over C^*-algebras. *Linear Algebra Appl.*, 456:174–198, 2014.

[22] M. Benzi, P. Boito, and N. Razouk. Decay properties of spectral projectors with applications to electronic structure. *SIAM Rev.*, 55(1):3–64, 2013.

[23] M. Benzi and G. H. Golub. Bounds for the entries of matrix functions with applications to preconditioning. *BIT*, 39(3):417–438, 1999.

[24] M. Benzi and N. Razouk. Decay bounds and $O(n)$ algorithms for approximating functions of sparse matrices. *Electron. Trans. Numer. Anal.*, 28:16–39, 2007/08.

[25] M. Benzi and V. Simoncini. Decay bounds for functions of Hermitian matrices with banded or Kronecker structure. *SIAM J. Matrix Anal. Appl.*, 36(3):1263–1282, 2015.

[26] M. Berljafa, S. Elsworth, and S. Güttel. A rational Krylov toolbox for MATLAB. 2014.

[27] D. S. Bernstein. *Matrix Mathematics.* Princeton University Press, Princeton, NJ, second edition, 2009.

[28] D. S. Bernstein and C. F. Van Loan. Rational matrix functions and rank-1 updates. *SIAM J. Matrix Anal. Appl.*, 22(1):145–154, 2000.

[29] R. Bhatia, M. D. Choi, and C. Davis. Comparing a matrix to its off-diagonal part. In *The Gohberg anniversary collection, Vol. I (Calgary, AB, 1988)*, volume 40 of *Oper. Theory Adv. Appl.*, pages 151–164. Birkhäuser, Basel, 1989.

[30] D. A. Bini, S. Massei, and B. Meĭni. On functions of quasi-Toeplitz matrices. *Mat. Sb.*, 208(11):56–74, 2017.

[31] S. Boucheron, G. Lugosi, and P. Massart. Concentration inequalities using the entropy method. *Ann. Probab.*, 31(3):1583–1614, 2003.

[32] S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities. A nonasymptotic theory of independence. With a foreword by Michel Ledoux.* Oxford University Press, Oxford, 2013.

[33] C. Boutsidis, P. Drineas, P. Kambadur, E.-M. Kontopoulou, and A. Zouzias. A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. *Linear Algebra Appl.*, 533:95–117, 2017.

[34] C. Boutsidis and M. Magdon-Ismail. Deterministic feature selection for k-means clustering. *IEEE Trans. Inform. Theory*, 59(9):6099–6110, 2013.

[35] C. Boutsidis, M. W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 968–977. SIAM, Philadelphia, PA, 2009.

[36] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, Berlin, 1997.

[37] A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Found. Comput. Math.*, 7(3):331–368, 2007.

[38] A. Çivril and M. Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoret. Comput. Sci.*, 410(47-49):4801–4811, 2009.

[39] A. Çivril and M. Magdon-Ismail. Exponential inapproximability of selecting a maximum volume sub-matrix. *Algorithmica*, 65(1):159–176, 2013.

[40] T. F. Chan. Rank revealing QR factorizations. *Linear Algebra Appl.*, 88/89:67–82, 1987.

[41] S. Chandrasekaran, M. Gu, and T. Pals. A fast ULV decomposition solver for hierarchically semiseparable representations. *SIAM J. Matrix Anal. Appl.*, 28(3):603–622, 2006.

[42] S. Chandrasekaran and I. C. F. Ipsen. On rank-revealing factorisations. *SIAM J. Matrix Anal. Appl.*, 15(2):592–622, 1994.

[43] J. Chen. On the use of discrete Laplace operator for preconditioning kernel matrices. *SIAM J. Sci. Comput.*, 35(2):A577–A602, 2013.

[44] J. Chen. How accurately should I compute implicit matrix-vector products when applying the Hutchinson trace estimator? *SIAM J. Sci. Comput.*, 38(6):A3515–A3539, 2016.

[45] A. Cortinovis and D. Kressner. Low-Rank Approximation in the Frobenius Norm by Column and Row Subset Selection. *SIAM J. Matrix Anal. Appl.*, 41(4):1651–1673, 2020.

[46] A. Cortinovis and D. Kressner. On randomized trace estimates for indefinite matrices with an application to determinants. *Found. Comput. Math.*, 2021.

[47] A. Cortinovis, D. Kressner, and S. Massei. On maximum volume submatrices and cross approximation for symmetric semidefinite and diagonally dominant matrices. *Linear Algebra Appl.*, 593:251–268, 2020.

[48] A. Cortinovis, D. Kressner, and S. Massei. Divide-and-Conquer Methods for Functions of Matrices with Banded or Hierarchical Low-Rank Structure. *SIAM J. Matrix Anal. Appl.*, 43(1):151–177, 2022.

[49] M. Crouzeix and D. Kressner. A bivariate extension of the Crouzeix-Palencia result with an application to Fréchet derivatives of matrix functions. *arXiv preprint arXiv:2007.09784*, 2020.

[50] M. Crouzeix and C. Palencia. The numerical range is a $(1+\sqrt{2})$-spectral set. *SIAM J. Matrix Anal. Appl.*, 38(2):649–655, 2017.

[51] P. I. Davies and N. J. Higham. A Schur-Parlett algorithm for computing matrix functions. *SIAM J. Matrix Anal. Appl.*, 25(2):464–485, 2003.

[52] T. A. Davis and Y. Hu. The University of Florida sparse matrix collection. *ACM Trans. Math. Software*, 38(1):Art. 1, 25, 2011.

[53] W. Dawson and T. Nakajima. Massively parallel sparse matrix function calculations with ntpoly. *Computer Physics Communications*, 225:154–165, 2018.

[54] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

[55] S. Demko, W. F. Moss, and P. W. Smith. Decay rates for inverses of band matrices. *Math. Comp.*, 43(168):491–499, 1984.

[56] M. Dereziński and M. W. Mahoney. Determinantal point processes in randomized numerical linear algebra. *Notices Amer. Math. Soc.*, 68(1):34–45, 2021.

[57] A. Deshpande and L. Rademacher. Efficient volume sampling for row/column subset selection. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science—FOCS 2010*, pages 329–338. IEEE Computer Soc., Los Alamitos, CA, 2010.

[58] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. *Theory Comput.*, 2:225–247, 2006.

[59] M. Di Summa, F. Eisenbrand, Y. Faenza, and C. Moldenhauer. On largest volume simplices and sub-determinants. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 315–323. SIAM, Philadelphia, PA, 2015.

[60] Distribution of Difference of Chi-squared Variables. https://math.stackexchange.com/questions/85249/distribution-of-difference-of-chi-squared-variables. Accessed: 06/03/2020.

[61] F. M. Dopico and P. Koev. Perturbation theory for the LDU factorization and accurate computations for diagonally dominant matrices. *Numer. Math.*, 119(2):337–371, 2011.

[62] P. Drineas and M. W. Mahoney. A randomized algorithm for a tensor-based generalization of the singular value decomposition. *Linear Algebra Appl.*, 420(2-3):553–571, 2007.

[63] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM J. Matrix Anal. Appl.*, 30(2):844–881, 2008.

[64] Z. Drmač and S. Gugercin. A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions. *SIAM J. Sci. Comput.*, 38(2):A631–A648, 2016.

[65] V. Druskin, S. Güttel, and L. Knizhnerman. Near-optimal perfectly matched layers for indefinite Helmholtz problems. *SIAM Rev.*, 58(1):90–116, 2016.

[66] E. Dudley, A. K. Saibaba, and A. Alexanderian. Monte Carlo estimators for the Schatten p-norm of symmetric positive semidefinite matrices. *Electron. Trans. Numer. Anal.*, 55:213–241, 2022.

[67] D. Durfee, J. Peebles, R. Peng, and A. B. Rao. Determinant-preserving sparsification of SDDM matrices. *SIAM J. Comput.*, 49(4):350–408, 2020.

[68] S. Elsworth and S. Güttel. The block rational Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 41(2):365–388, 2020.

[69] E. Estrada. Characterization of 3D molecular structure. *Chemical Physics Letters*, 319(5):713–718, 2000.

[70] E. Estrada and D. J. Higham. Network properties revealed through matrix functions. *SIAM Rev.*, 52(4):696–714, 2010.

[71] W. N. Everitt. A note on positive definite matrices. *Proc. Glasgow Math. Assoc.*, 3:173–175, 1958.

[72] A. K. Farahat, A. Elgohary, A. Ghodsi, and M. S. Kamel. Distributed column subset selection on mapreduce. In *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 171–180, 2013.

[73] J. Fitzsimons, D. Granziol, K. Cutajar, M. Osborne, M. Filippone, and S. Roberts. Entropic trace estimates for log determinants. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 323–338. Springer, 2017.

[74] L. Foster, A. Waagen, N. Aijaz, M. Hurley, A. Luis, J. Rinsky, C. Satyavolu, M. J. Way, P. Gazis, and A. Srivastava. Stable and efficient Gaussian process calculations. *J. Mach. Learn. Res.*, 10:857–882, 2009.

[75] L. V. Foster and X. Liu. Comparison of rank revealing algorithms applied to matrices with well defined numerical ranks, 2006. Manuscript.

[76] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013.

[77] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.

[78] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, 2004.

[79] A. Frommer, C. Schimmel, and M. Schweitzer. Analysis of probing techniques for sparse approximation and trace estimation of decaying matrix functions. *SIAM J. Matrix Anal. Appl.*, 42(3):1290–1318, 2021.

[80] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, volume 2018-December, pages 7576–7586, 2018.

[81] I. P. Gavrilyuk, W. Hackbusch, and B. N. Khoromskij. H-matrix approximation for the operator exponential with applications. *Numer. Math.*, 92(1):83–111, 2002.

[82] M. I. Gil'. Perturbations of functions of diagonalizable matrices. *Electron. J. Linear Algebra*, 20:303–313, 2010.

[83] A. Gittens and M. W. Mahoney. Revisiting the Nyström method for improved large-scale machine learning. *J. Mach. Learn. Res.*, 17:Paper No. 117, 65, 2016.

[84] S. Goedecker. Linear scaling electronic structure methods. *Reviews of Modern Physics*, 71(4):1085, 1999.

[85] G. H. Golub and G. Meurant. *Matrices, moments and quadrature with applications.* Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2010.

[86] G. H. Golub and C. F. Van Loan. *Matrix computations.* Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.

[87] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. How to find a good submatrix. In *Matrix methods: theory, algorithms and applications*, pages 247–256. World Sci. Publ., Hackensack, NJ, 2010.

[88] S. A. Goreinov and 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)*, volume 280 of *Contemp. Math.*, pages 47–51. Amer. Math. Soc., Providence, RI, 2001.

[89] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra Appl.*, 261:1–21, 1997.

[90] S. A. Goreĭnov. Cross approximation of a multi-index array. *Dokl. Akad. Nauk*, 420(4):439–441, 2008.

[91] S. A. Goreĭnov and E. E. Tyrtyshnikov. Quasi-optimality of a skeleton approximation of a matrix in the Chebyshev norm. *Dokl. Akad. Nauk*, 438(5):593–594, 2011.

[92] L. Grasedyck, W. Hackbusch, and B. N. Khoromskij. Solution of large scale algebraic matrix Riccati equations by use of hierarchical matrices. *Computing*, 70(2):121–165, 2003.

[93] S. Gratton and D. Titley-Peloquin. Improved bounds for small-sample estimation. *SIAM J. Matrix Anal. Appl.*, 39(2):922–931, 2018.

[94] P. Gritzmann, V. Klee, and D. Larman. Largest j-simplices in n-polytopes. *Discrete Comput. Geom.*, 13(3-4):477–515, 1995.

[95] M. Gu and S. C. Eisenstat. A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem. *SIAM J. Matrix Anal. Appl.*, 16(1):172–191, 1995.

[96] M. Gu and S. C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J. Sci. Comput.*, 17(4):848–869, 1996.

[97] M. Gu and L. Miranian. Strong rank revealing Cholesky factorization. *Electron. Trans. Numer. Anal.*, 17:76–92, 2004.

[98] S. Güttel. Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. *GAMM-Mitt.*, 36(1):8–31, 2013.

[99] S. Güttel, D. Kressner, and K. Lund. Limited-memory polynomial methods for large-scale matrix functions. *GAMM-Mitt.*, 43(3):e202000019, 19, 2020.

[100] W. Hackbusch. *Hierarchical matrices: algorithms and analysis*, volume 49 of *Springer Series in Computational Mathematics*. Springer, Heidelberg, 2015.

[101] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.

[102] E. Hallman and D. Troester. A multilevel approach to stochastic trace estimation. *Linear Algebra Appl.*, 638:125–149, 2022.

[103] I. Han, D. Malioutov, H. Avron, and J. Shin. Approximating spectral sums of large-scale matrices using stochastic Chebyshev approximations. *SIAM J. Sci. Comput.*, 39(4):A1558–A1585, 2017.

[104] I. Han, D. Malioutov, and J. Shin. Large-scale log-determinant computation through stochastic Chebyshev expansions. In *International Conference on Machine Learning*, pages 908–917, 2015.

[105] D. L. Hanson and F. T. Wright. A bound on tail probabilities for quadratic forms in independent random variables. *Ann. Math. Statist.*, 42:1079–1083, 1971.

[106] H. Harbrecht, M. Peters, and R. Schneider. On the low-rank approximation by the pivoted Cholesky decomposition. *Appl. Numer. Math.*, 62(4):428–440, 2012.

[107] N. J. Higham. A survey of condition number estimation for triangular matrices. *SIAM Rev.*, 29(4):575–596, 1987.

[108] N. J. Higham. Analysis of the Cholesky decomposition of a semi-definite matrix. In *Reliable numerical computation*, Oxford Sci. Publ., pages 161–185. Oxford Univ. Press, New York, 1990.

[109] N. J. Higham. *Accuracy and stability of numerical algorithms.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2002.

[110] N. J. Higham. *Functions of matrices. Theory and computation.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.

[111] R. A. Horn and C. R. Johnson. *Topics in matrix analysis.* Cambridge University Press, Cambridge, 1994. Corrected reprint of the 1991 original.

[112] R. A. Horn and C. R. Johnson. *Matrix analysis.* Cambridge University Press, Cambridge, second edition, 2013.

[113] T. Hunter, A. E. Alaoui, and A. M. Bayen. Computing the log-determinant of symmetric, diagonally dominant matrices in near-linear time. *CoRR*, abs/1408.1693, 2014.

[114] M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Comm. Statist. Simulation Comput.*, 18(3):1059–1076, 1989.

[115] M. Ilić, I. W. Turner, and D. P. Simpson. A restarted Lanczos approximation to functions of a symmetric matrix. *IMA J. Numer. Anal.*, 30(4):1044–1061, 2010.

[116] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Trans. Signal Process.*, 57(2):451–462, 2009.

[117] W. M. Kahan. Numerical linear algebra. *Canadian Mathematical Bulletin*, 9:757–801, 01 1966.

[118] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1998.

[119] O. Knill. Cauchy-Binet for pseudo-determinants. *Linear Algebra Appl.*, 459:522–547, 2014.

[120] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009.

[121] I. Koutis, A. Levin, and R. Peng. Faster spectral sparsification and numerical algorithms for SDD matrices. *ACM Trans. Algorithms*, 12(2):Art. 17, 16, 2016.

[122] F. Krahmer and R. Ward. New and improved Johnson-Lindenstrauss embeddings via the restricted isometry property. *SIAM J. Math. Anal.*, 43(3):1269–1281, 2011.

[123] D. Kressner and R. Luce. Fast computation of the matrix exponential for a Toeplitz matrix. *SIAM J. Matrix Anal. Appl.*, 39(1):23–47, 2018.

[124] D. Kressner, S. Massei, and L. Robol. Low-rank updates and a divide-and-conquer method for linear matrix equations. *SIAM J. Sci. Comput.*, 41(2):A848–A876, 2019.

[125] D. Kressner and C. Tobler. Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM J. Matrix Anal. Appl.*, 32(4):1288–1316, 2011.

[126] D. Kressner and A. Šušnjara. Fast computation of spectral projectors of banded matrices. *SIAM J. Matrix Anal. Appl.*, 38(3):984–1009, 2017.

[127] B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Ann. Statist.*, 28(5):1302–1338, 2000.

[128] S. T. Lee, H.-K. Pang, and H.-W. Sun. Shift-invert Arnoldi approximation to the Toeplitz matrix exponential. *SIAM J. Sci. Comput.*, 32(2):774–792, 2010.

[129] H. Li and Y. Zhu. Randomized block Krylov space methods for trace and log-determinant estimators. *arXiv preprint arXiv:2003.00212*, 2020.

[130] L. Lin, J. Lu, L. Ying, R. Car, and W. E. Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems. *Commun. Math. Sci.*, 7(3):755–777, 2009.

[131] L. Lin, Y. Saad, and C. Yang. Approximating spectral densities of large matrices. *SIAM Rev.*, 58(1):34–65, 2016.

[132] P. G. Martinsson. A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix. *SIAM J. Matrix Anal. Appl.*, 32(4):1251–1274, 2011.

[133] P.-G. Martinsson and J. A. Tropp. Randomized numerical linear algebra: foundations and algorithms. *Acta Numer.*, 29:403–572, 2020.

[134] S. Massei. Some algorithms for maximum volume and cross approximation of symmetric semidefinite matrices. *BIT*, 62(1):195–220, 2022.

[135] S. Massei, M. Mazza, and L. Robol. Fast solvers for two-dimensional fractional diffusion equations using rank structured matrices. *SIAM J. Sci. Comput.*, 41(4):A2627–A2656, 2019.

[136] S. Massei and L. Robol. Decay bounds for the numerical quasiseparable preservation in matrix functions. *Linear Algebra Appl.*, 516:212–242, 2017.

[137] S. Massei, L. Robol, and D. Kressner. hm-toolbox: MATLAB software for HODLR and HSS matrices. *SIAM J. Sci. Comput.*, 42(2):C43–C68, 2020.

[138] C. Maung and H. Schweitzer. Pass-efficient unsupervised feature selection. In *Advances in Neural Information Processing Systems*, 2013.

[139] A. K. Menon and C. Elkan. Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2011.

[140] R. A. Meyer, C. Musco, C. Musco, and D. P. Woodruff. Hutch++: Optimal stochastic trace estimation. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 142–155. SIAM, 2021.

[141] C. A. Micchelli and A. Pinkus. Some problems in the approximation of functions of two variables and n-widths of integral operators. *J. Approx. Theory*, 24(1):51–77, 1978.

[142] L. Miranian and M. Gu. Strong rank revealing *LU* factorizations. *Linear Algebra Appl.*, 367:1–16, 2003.

[143] S. Morozov, N. Zamarashkin, and E. Tyrtyshnikov. On the algorithm of best approximation by low rank matrices in the Chebyshev norm. *arXiv preprint arXiv:2201.12301*, 2022.

[144] K. Németh and G. E. Scuseria. Linear scaling density matrix search based on sign matrices. *The Journal of Chemical Physics*, 113(15):6035–6041, 2000.

[145] M. Neteler and H. Mitasova. *Open source GIS: a GRASS GIS approach*, volume 689. Springer Science & Business Media, 2013.

[146] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E (3)*, 74(3):036104, 19, 2006.

[147] B. Ordozgoiti, A. Mozo, and J. G. López de Lacalle. Regularized greedy column subset selection. *Inform. Sci.*, 486:393–418, 2019.

[148] I. Oseledets and E. Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.*, 432(1):70–88, 2010.

[149] I. V. Oseledets, D. V. Savostianov, and E. E. Tyrtyshnikov. Tucker dimensionality reduction of three-dimensional arrays in linear time. *SIAM J. Matrix Anal. Appl.*, 30(3):939–956, 2008.

[150] A. Osinsky. Rectangular maximum volume and projective volume search algorithms. *arXiv preprint arXiv:1809.02334*, 2018.

[151] A. I. Osinsky and N. L. Zamarashkin. Pseudo-skeleton approximations with better accuracy estimates. *Linear Algebra Appl.*, 537:221–249, 2018.

[152] R. K. Pace and J. P. LeSage. Chebyshev approximation of log-determinants of spatial weight matrices. *Comput. Statist. Data Anal.*, 45(2):179–196, 2004.

[153] C.-T. Pan. On the existence and computation of rank-revealing *LU* factorizations. *Linear Algebra Appl.*, 316(1-3):199–222, 2000. Conference Celebrating the 60th Birthday of Robert J. Plemmons (Winston-Salem, NC, 1999).

[154] C. H. Papadimitriou. The largest subdeterminant of a matrix. *Bull. Soc. Math. Grèce (N.S.)*, 25:95–105, 1984.

[155] J. M. Peña. LDU decompositions with L and U well conditioned. *Electron. Trans. Numer. Anal.*, 18:198–208, 2004.

[156] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[157] D. Persson, A. Cortinovis, and D. Kressner. Improved variants of the Hutch++ algorithm for trace estimation. *arXiv preprint arXiv:2109.10659*, 2021.

[158] S. Pozza and V. Simoncini. Inexact Arnoldi residual estimates and decay properties for functions of non-Hermitian matrices. *BIT*, 59(4):969–986, 2019.

[159] S. Pozza and F. Tudisco. On the stability of network indices defined by means of matrix functions. *SIAM J. Matrix Anal. Appl.*, 39(4):1521–1546, 2018.

[160] R. Rehman and I. C. Ipsen. La Budde's method for computing characteristic polynomials. *arXiv preprint arXiv:1104.3769*, 2011.

[161] R. Rehman and I. C. F. Ipsen. Computing characteristic polynomials from eigenvalues. *SIAM J. Matrix Anal. Appl.*, 32(1):90–114, 2011.

[162] F. Roosta-Khorasani and U. Ascher. Improved bounds on sample size for implicit matrix trace estimators. *Found. Comput. Math.*, 15(5):1187–1212, 2015.

[163] M. Rudelson and R. Vershynin. Hanson-Wright inequality and sub-Gaussian concentration. *Electron. Commun. Probab.*, 18:no. 82, 9, 2013.

[164] A. Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra Appl.*, 58:391–405, 1984.

[165] Y. Saad. Analysis of some Krylov subspace approximations to the matrix exponential operator. *SIAM J. Numer. Anal.*, 29(1):209–228, 1992.

[166] A. K. Saibaba. HOID: higher order interpolatory decomposition for tensors based on Tucker representation. *SIAM J. Matrix Anal. Appl.*, 37(3):1223–1249, 2016.

[167] A. K. Saibaba, A. Alexanderian, and I. C. F. Ipsen. Randomized matrix-free trace and log-determinant estimators. *Numer. Math.*, 137(2):353–395, 2017.

[168] J. Schneider. Error estimates for two-dimensional cross approximation. *J. Approx. Theory*, 162(9):1685–1700, 2010.

[169] L. Schork and J. Gondzio. Rank revealing Gaussian elimination by the maximum volume concept. *Linear Algebra Appl.*, 592:1–19, 2020.

[170] M. Shao. On the finite section method for computing exponentials of doubly-infinite skew-Hermitian matrices. *Linear Algebra Appl.*, 451:65–96, 2014.

[171] A. Skripka and A. Tomskova. *Multilinear operator integrals*, volume 2250 of *Lecture Notes in Mathematics*. Springer, Cham, 2019.

[172] D. C. Sorensen and M. Embree. A DEIM induced CUR factorization. *SIAM J. Sci. Comput.*, 38(3):A1454–A1482, 2016.

[173] D. A. Spielman and S.-H. Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM J. Matrix Anal. Appl.*, 35(3):835–885, 2014.

[174] A. Stathopoulos, J. Laeuchli, and K. Orginos. Hierarchical probing for estimating the trace of the matrix inverse on toroidal lattices. *SIAM J. Sci. Comput.*, 35(5):S299–S322, 2013.

[175] G. W. Stewart. Four algorithms for the efficient computation of truncated pivoted QR approximations to a sparse matrix. *Numer. Math.*, 83(2):313–323, 1999.

[176] M. Talagrand. New concentration inequalities in product spaces. *Invent. Math.*, 126(3):505–563, 1996.

[177] J. M. Tang and Y. Saad. A probing method for computing the diagonal of a matrix inverse. *Numer. Linear Algebra Appl.*, 19(3):485–501, 2012.

[178] C. Thron, S. J. Dong, K. F. Liu, and H. P. Ying. Padé-Z_2 estimator of determinants. *Physical Review D - Particles, Fields, Gravitation and Cosmology*, 57(3):1642–1653, 1998.

[179] A. Townsend. *Computing with functions in two dimensions*. ProQuest LLC, Ann Arbor, MI, 2014. Thesis (D.Phil.)–University of Oxford, UK.

[180] A. Townsend and L. N. Trefethen. Continuous analogues of matrix factorizations. *Proc. A.*, 471(2173):20140585, 21, 2015.

[181] L. N. Trefethen. *Approximation theory and approximation practice*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2013.

[182] J. A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[183] S. Ubaru, J. Chen, and Y. Saad. Fast estimation of $\mathtt{tr}(f(A))$ via stochastic Lanczos quadrature. *SIAM J. Matrix Anal. Appl.*, 38(4):1075–1099, 2017.

[184] M. Udell and A. Townsend. Why are big data matrices approximately low rank? *SIAM J. Math. Data Sci.*, 1(1):144–160, 2019.

[185] Upper Limit on the Central Binomial Coefficient. https://mathoverflow.net/questions/133732/upper-limit-on-the-central-binomial-coefficient. Accessed: 23/03/2020.

[186] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen. A new truncation strategy for the higher-order singular value decomposition. *SIAM J. Sci. Comput.*, 34(2):A1027–A1052, 2012.

[187] M. J. Wainwright. *High-dimensional statistics. A non-asymptotic viewpoint*, volume 48 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2019.

[188] M. J. Wainwright and M. I. Jordan. Log-determinant relaxation for approximate inference in discrete Markov random fields. *IEEE transactions on signal processing*, 54(6):2099–2109, 2006.

[189] B. H. Wang, H. T. Hui, and M. S. Leong. Global and fast receiver antenna selection for mimo systems. *IEEE Transactions on Communications*, 58(9):2505–2510, 2010.

[190] J. H. Wilkinson. Error analysis of direct methods of matrix inversion. *J. Assoc. Comput. Mach.*, 8:281–330, 1961.

[191] K. Wimmer, Y. Wu, and P. Zhang. Optimal query complexity for estimating the trace of a matrix. In *International Colloquium on Automata, Languages, and Programming*, pages 1051–1062. Springer, 2014.

[192] D. P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):iv+157, 2014.

[193] L. Wu, J. Laeuchli, V. Kalantzis, A. Stathopoulos, and E. Gallopoulos. Estimating the trace of the matrix inverse by interpolating from the diagonal of an approximate inverse. *J. Comput. Phys.*, 326:828–844, 2016.

[194] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li. Fast algorithms for hierarchically semiseparable matrices. *Numer. Linear Algebra Appl.*, 17(6):953–976, 2010.

[195] P. A.-C. Yoon. *Modifying two-sided orthogonal decompositions: Algorithms, implementation, and applications.* ProQuest LLC, Ann Arbor, MI, 1996. Thesis (Ph.D.)–The Pennsylvania State University.

[196] N. L. Zamarashkin and A. I. Osinsky. On the existence of a nearly optimal skeleton approximation of a matrix in the Frobenius norm. *Dokl. Akad. Nauk*, 97(2):164–166, 2018.

[197] Y. Zhang and W. E. Leithead. Approximate implementation of the logarithm of the matrix determinant in Gaussian process regression. *J. Stat. Comput. Simul.*, 77(3-4):329–348, 2007.

Curriculum Vitae

Personal information

Name	Alice Cortinovis
Date of birth	20.07.1994.
Place of birth	Monza, Italy
Nationality	Italian

Education

2018 – 2022	**Doctoral studies in Applied Mathematics** École Polytechnique Fédérale de Lausanne Thesis: *Fast deterministic and randomized algorithms for low-rank approximation, matrix functions, and trace estimation* Phd Advisor: Prof. D. Kressner
2016 – 2018	**M.Sc. in Mathematics** University of Pisa Thesis: *Minimizing the optimality residual for algebraic Riccati equations* Master Thesis advisor: Prof. F. Poloni
2013 – 2016	**B.Sc. in Mathematics** University of Pisa Thesis: *Immersioni di spazi metrici finiti in* $\mathbb{R}^n$ *(Immersions of finite metric spaces into* $\mathbb{R}^n$ *)* Bachelor Thesis advisor: Prof. G. Alberti
2013 – 2018	**Scuola Normale Superiore di Pisa**, Allievo ordinario

Publications and preprints

- D. Persson, A. Cortinovis, and D. Kressner, *Improved variants of the Hutch++ algorithm for trace estimation* (2021). Accepted for publication in SIAM Journal on Matrix Analysis and Applications. https://arxiv.org/abs/2109.10659
- A. Cortinovis, D. Kressner, and S. Massei, *Divide and conquer methods for functions of matrices with banded or hierarchical low-rank structure*, SIAM Journal on Matrix Analysis and Applications (2022). https://epubs.siam.org/doi/abs/10.1137/21M1432594
- A. Cortinovis and D. Kressner, *On randomized trace estimates for indefinite matrices with an application to determinants*, Foundations of Computational Mathematics (2021). https://link.springer.com/article/10.1007/s10208-021-09525-9
- B. Beckermann, A. Cortinovis, D. Kressner, and M. Schweitzer, *Low-rank updates of matrix functions II: Rational Krylov methods*, SIAM Journal on Numerical Analysis (2021). https://epubs.siam.org/doi/pdf/10.1137/20M1362553
- A. Cortinovis and D. Kressner, *Low-rank approximation in the Frobenius norm by column and row subset selection*, SIAM Journal on Matrix Analysis and Applications (2020). https://epubs.siam.org/doi/pdf/10.1137/19M1281848
- A. Cortinovis, D. Kressner, and S. Massei, *On maximum volume submatrices and cross approximation for symmetric semidefinite and diagonally dominant matrices*, Linear Algebra and its Applications (2020). https://www.sciencedirect.com/science/article/pii/S0024379520300768
- A. Cortinovis, D. Kressner, S. Massei, and B. Peherstorfer, *Quasi-optimal sampling to learn basis updates for online adaptive model reduction with adaptive empirical interpolation.* In: American Control Conference (ACC) 2020. https://ieeexplore.ieee.org/iel7/9140048/9147203/09147832.pdf

Conferences and schools

- 47th Annual Spring Lecture Series. Numerical Linear Algebra: from Scientific Computing to Data Science Applications. University of Arkansas (Fayetteville, AR, USA), 2022.
  Invited talk: *Randomized trace estimation and determinants.*
- SCAN Seminar (Cornell University, USA), 2022 (online).
  Invited talk: *Randomized algorithms for trace estimation.*

- Due giorni di Algebra Lineare Numerica (Two days of Numerical Linear Algebra), Napoli (Italy), 2022.
  Contributed talk: *Randomized algorithms for trace estimation.*
- Numerical Analysis Group Internal Seminar (University of Oxford, UK), 2021.
  Invited talk: *Randomized algorithms for trace estimation.*
- Conference on Fast Direct Solvers (online), 2021.
  Contributed talk: *Divide and conquer methods for functions of matrices with banded or hierarchical low-rank structure.*
- Swiss Numerics Day, EPF Lausanne (Switzerland), 2021.
  Contributed talk: *Divide and conquer methods for functions of matrices with banded or hierarchical low-rank structure.*
- Matrix Equations and Tensor Techniques IX, Perugia (Italy), 2021.
  Contributed talk: *Divide and conquer methods for functions of matrices with banded or hierarchical low-rank structure.*
- SIAM Conference on Applied Linear Algebra (online), 2021.
  Talk in minisymposium: *Low-rank approximation by row and column subset selection.*
- Numpi seminar, University of Pisa (online), March 2021.
  Invited talk: *Randomized trace estimates for indefinite matrices with an application to determinants.*
- Communications in Numerical Linear Algebra, online seminar series, 2020.
  Talk: *Randomized trace estimates for indefinite matrices with an application to determinants.*
- Low-rank models 2020 (winter school), Villars-sur-Ollon (Switzerland).
  Poster: *Analysis of algorithms for cross approximation.*
- Low-rank models and applications, Mons (Belgium), 2019.
  Contributed talk: *On maximum volume submatrices and cross approximation.*
- 5th International conference on Matrix Methods in Mathematics and Applications, Moscow, 2019.
  Contributed talk: *On maximum volume submatrices and cross approximation.*
- Advances in Numerical Linear Algebra: Celebrating the Centenary of the Birth of James H. Wilkinson, Cambridge (UK), 2019.
- Swiss Numerics Day, Lugano (Switzerland), 2019.
  Poster: *On maximum volume submatrices and cross approximation.*

- Winterschool on Hierarchical Matrices 2019, Kiel (Germany).
- Due giorni di Algebra Lineare Numerica (Two days of Numerical Linear Algebra), Rome, 2019.
  Contributed talk: *On maximum volume submatrices and cross approximation.*
- Rome-Moscow school of Matrix Methods and Applied Linear Algebra, Rome and Moscow, 2018.
  Student session talk: *Minimizing the optimality residual for algebraic Riccati equations.*