COMPUTING NUMERICALLY WITH RATIONAL FUNCTIONS

A Dissertation
Presented to the Faculty of the Graduate School
of Cornell University
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by
Heather Denise Wilber
May 2021

COMPUTING NUMERICALLY WITH RATIONAL FUNCTIONS

Heather Denise Wilber, Ph.D.

Cornell University 2021

New numerical methods using rational functions are presented for applications in linear algebra and signal processing. Classical results from Zolotarev are applied to develop a collection of low rank methods and theoretical results for computing with matrices that have special displacement structures using the alternating direction implicit (ADI) method. This includes a new low rank method for solving Sylvester and Lyapunov matrix equations with right hand sides that have decaying singular values, spectrally accurate low rank solvers for certain elliptic partial differential equations with smooth right-hand sides, and explicit bounds on the singular values of special families of structured matrices.

Methods from conformal mapping and adaptive rational approximation are applied to build approximate solutions to Zolotarev's problem on sets where solutions are not known. This leads to new bounds on the numerical ranks of matrices, and it generalizes the regime in which ADI-based methods can be applied. The approximate solutions supply quasi-optimal ADI shift parameters for solving Sylvester matrix equations.

A superfast rank-structured solver for Toeplitz linear systems is designed with ADI-based compression methods, and theoretical arguments are supplied that justify the effectiveness of rank-structured solvers for Toeplitz and related linear systems. The solvers are competitive with the state of the art, and rational approximation arguments are used to derive explicit error bounds on the numerical ranks of important submatrices for various weakly admissible hierarchical formats.

A data-driven rational approximation framework is developed for reconstructing signals from samples with poorly separated spectral content. This approach combines a variant of Prony's method with a modified version of the AAA algorithm to construct representations of signals in both frequency and time space. The approximation methods are automatic and adaptive, requiring no tuning or manual parameter selection, and they are robust to various forms of corruption, including additive Gaussian noise, perturbed sampling grids, and missing data. A collection of algorithms and an accompanying software package for adaptively computing with these representations is introduced that includes procedures for differentiation/integration, rootfinding/polefinding, convolution, filtering, extrapolation, and more.

BIOGRAPHICAL SKETCH

Heather Denise Wilber was introduced to and fell in love with computational mathematics and approximation theory under the mentorship of Grady Wright at Boise State University. She began her PhD in Applied Mathematics at Cornell University in August of 2016 under the supervision of Alex Townsend, and will go on to complete an NSF postdoctoral fellowship at the Oden Institute at the University of Texas, Austin.

Dedicated to Daniel

ACKNOWLEDGEMENTS

An underrated aspect of academia is that it is fundamentally designed around mentoring relationships. Once a person in earnest decides to become an academic, a small circle of advisors and mentors instantly become hers, and they seem to arrive with built-in, unmerited levels of enthusiasm and faith. I have been especially fortunate in my mentors. This includes my MS advisor, Grady Wright, who patiently taught me the basics about numerical methods, coding, and how to think like a researcher when I had never done any of these things before. Like many students of numerical analysis, I look to Nick Trefethen as a great inspiration, and yet I am in the remarkable circumstance of also knowing him as a beloved personal mentor. I admire and have substantially benefited from the lucidity of his work on rational approximation, which spans decades. I am grateful to Daniel Kressner, who supported me in a visit to EPFL in Switzerland, and has greatly encouraged my interest in and knowledge of numerical linear algebra. I am grateful to Gunnar Martinsson for his encouragement and help in securing an NSF postdoctoral fellowship. I have benefited from the teaching, guidance and friendship of many others, including Bernhard Beckermann, James Sethna, Katherine Quinn, Alexander Vladimirsky, David Bindel, Yuji Nakatsukasa, Sheehan Olver, John Guckenheimer, Timothy Healey, Bob Strichartz, Steven Strogatz, and Erika Fowler–Decatur.

The moment I met Anil Damle, I knew he would be a fantastic numerical analysis teacher. I decided immediately to find a way to work with him and learn as much as I could. I didn't know that he would become one of my most treasured teachers in regards to academic life more generally. His sharp sense of mathematical inquiry has been a continual source of inspiration and joy for me, and it keeps my list of open questions long and interesting.

As for my advisor, Alex Townsend, I cannot begin to unravel how his mentorship has impacted me. Alex sets high standards for his students from the very beginning. Before the students catch on to the fact that the standards are higher than average, it's too late; habits and beliefs have already begun to crystalize, and it may as well be accepted as reality. This strategy works because Alex doesn't merely model adherence to such standards in his own life, he makes it look natural. He is a captivating, captainish kind of person who seems to bend the world to his will. To listen to him teach, think, or share advice is to have a lesson about happily broadening the imagination. Right now, I find his influence imprinted upon everything I do. He has shaped my daily habits, my expectations, my vision of myself, and the way I think and dream about mathematics. I couldn't have been given a better start.

I thank my colleagues and friends, John Chavis, Kathryn Drake, James Ford, Marc Aurèle Gilles, Andrew Horning, Tianyi Shi, Mateo Díaz, Nicolas Boullé, and Dan Fortunato, who have not only made this journey incredibly fun, but have been inspiring and creative mathematical companions. I thank my patient and loving family, including my parents, siblings, and also Ty, Kelsey, Gracie, Jeremiah, and Isaac. Some of my favorite mathematical excursions these past five years have been with my son, James, who has an extraordinary sense for the delight of mathematical discovery. My younger son, Owen, has made sure to remind me that my busy and serious life is actually just levity. Finally, words are inadequate, but let me try to express the gratitude, admiration, and enduring love for my husband Daniel that this time has produced in me. Before we got married, he told me he would make my dreams come true. Building a life with him has filled this time with heights from my dreams, and unimaginable joy.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This thesis develops numerical methods for a range of applications in computational mathematics where approximations by rational functions are useful. This includes theoretical and algorithmic advancements for low rank and rank-structured methods in numerical linear algebra, low rank spectral methods for solving certain partial differential equations (PDEs), as well as data-driven univariate methods for computing with functions, signals and nonlinear models. A central goal of this work is to translate ideas from approximation theory into computational tools for the wider scientific community. With this in mind, the major ideas in this thesis have been co-developed with open source software that is publicly available [40, 175].

Chapters 2-4 expand upon classical ideas that link rational approximation theory to the notion of displacement structure in numerical linear algebra. A collection of low rank methods and theoretical results is developed for computing with matrices with special displacement structures. This includes low rank approximation methods, explicit bounds on the singular values of these matrices, and methods for solving matrix equations and linear systems that involve these matrices. The main workhorse of this approach is an iterative method known as the alternating direction implicit (ADI) method, which has convergence properties that are explained by a rational approximation problem (see Section 1.5). A collection of key reference papers have been central to the development of these chapters and provide a good overview of the topics involved. This includes the survey paper on Sylvester and related matrix equations from Simoncini in [151], a collection of fundamental papers introducing the ADI and factored ADI methods [21, 105, 107, 109, 127], Beckermann and Townsend's paper on the singular values of matrices with low displacement rank [20], Achieser's texts on approximation theory and Zolotarev rational functions [1, 3], Saff's overview of applications of logarithmic potential theory to approximation theory [144], and Ganelius' papers on the Faber rationals [58, 59, 60]. Finally, though there are no English translations available (Achieser provides an English description of the work), the central question from rational approximation theory that motivates and illuminates our work is from a paper by Y.I. Zolotarev [182].

In Chapter 2, we develop a new ADI-based method for solving Sylvester and Lyapunov matrix equations with right hand sides that are numerically of low rank. We apply this method to develop spectrally accurate low rank solvers for certain elliptic PDEs with smooth right-hand sides, and we show how it leads to new bounds for certain families of matrices with numerically low displacement ranks (e.g., multidimensional Vandermonde matrices).

In Chapter 3, we use methods from conformal mapping and adaptive rational approximation to build Faber rational functions, which are approximate solutions to the rational approximation problem that lies at the heart of the ADI method. These solutions generalize the regime in which ADI-based approaches for bounding the singular values of matrices can be applied. We describe new bounds for the singular values of special Vandermonde and Cauchy matrices, and we show how these solutions supply shift parameters for ADI-based algorithms in more general settings.

In Chapter 4, we apply ADI-based methods to design a collection of superfast direct solvers for linear systems $Ax = b$, where A has a special displacement structure. Examples include Toeplitz, Toeplitz+Hankel, and special Vandermonde systems. Our methods use displacement structure to characterize and exploit rank-structured compression properties for matrices related to A by fast transforms. We combine ADI-based compression with modern numerical linear algebra techniques, such as inversion methods for hierarchical matrices. Explicit bounds on the singular values of submatrices arising in these systems are derived and used to justify weakly-admissible partitioning schemes.

In Chapter 5 we introduce a framework for computing with rational functions within the context of univariate signal processing. Whereas Chapters 2-4 are focused on a particular rational approximation problem in numerical linear algebra, Chapter 5 develops data-driven rational approximation methods that require little a priori knowledge about the underlying process being modeled. We use construction algorithms that are robust against various types of noise and corruption, and we combine ideas from rational approximation theory and harmonic analysis to develop a collection of automated algorithms for computing with our representations. Central texts relevant to the developments in Chapter 5 include several papers on barycentric rational interpolation and trigonometric rational functions [24, 25, 26, 86, 95], a paper that introduces the AAA algorithm [119], and several papers concerning the development, analysis, and practical use of Prony's method [27, 28, 132, 135, 136]. The spirit of our work is largely inspired by the Chebfun project [45, 177], and in particular, Trefethen's compelling paper about computing with functions instead of numbers [164].

1.1 Rational functions in computational mathematics

Rational and polynomial functions are a central part of both computational mathematics [5, 24, 88, 52, 142] and approximation studies more generally [1, 66, 145] (see [163, Ch. 23] for an overview). Trefethen, quoting Kirchberger, observes that when we limit our toolbox of mathematical capabilities to the machine arithmetic operations, the functions we can produce are polynomials and rationals [163, p.197]. It is no wonder then that polynomial and rational functions arise constantly in the development and analysis of numerical methods. This observation is nowhere more potent than in numerical linear algebra, where we have precious few tricks up our sleeves for computing with matrices and tensors, and performing even the basic arithmetic operations (matrix-vector products and solving linear systems) can be prohibitively expensive. Understanding the convergence behaviors of rational and polynomial approximations to functions is critical for efficiently carrying out fundamental tasks, such as computing eigenvalues (e.g., Rayleigh quotients [65], rational filters [130, 178]), evaluating functions of matrices (e.g., computing the square roots and exponentials of matrices [61, 88]), and solving linear systems (e.g., via (rational) Krylov methods [23, 142]). When fast shifted matrix-vector products are available for a matrix A, the construction of polynomials and their evaluation at A is a natural way to develop algorithms. Rational functions become useful when one also has the ability to efficiently solve shifted linear systems involving A.

In this work, we focus on settings where global approximation methods are desirable for approximating functions with singularities. Rationals often outshine their polynomial counterparts in exceptionally powerful ways in this regime. For example, rational approximations to the function $f(x) = |x|$

on $[-1,1]$ can achieve convergence rates in the infinity norm that are root-exponential in the degrees of freedom, whereas the rate attainable by polynomials is only $\mathcal{O}(1/m)$, where m is the degree of the polynomial [163, Ch. 25]. This convergence rate is also achievable by rationals when approximating functions with more complicated singularities, such as $f(x) = \sqrt{x}$ on $[0,1]$, which has a branch cut just off the interval of approximation [166]. In Figure 1.1 we show rational (black) and polynomial (purple) approximations to a characteristic function with jump discontinuities. Using the best polynomial approximations as measured in the infinity norm on $[0,1)$, the ringing error around the singularity decays at only an algebraic rate with respect to distance from the singularity [163, Ch. 9]. In contrast, there are rational approximations to f with errors that decay exponentially fast as one moves away from the singularity.

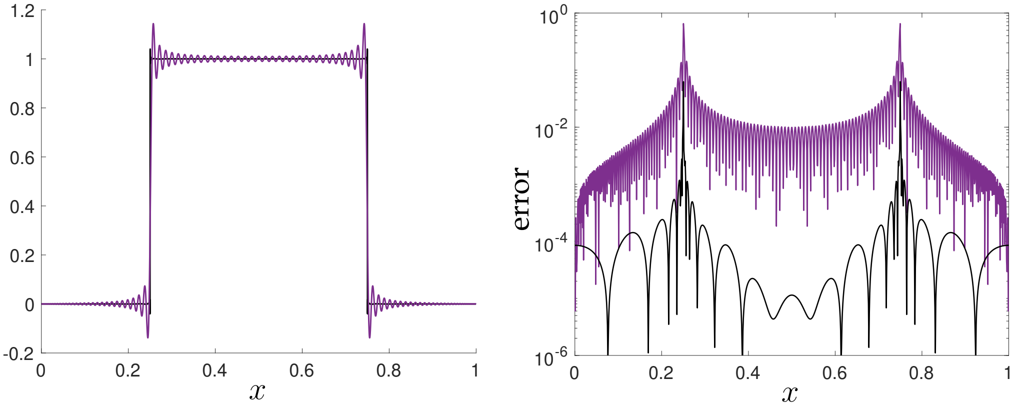


Figure 1.1: Left: Rational (black) and polynomial (purple) approximations to the characteristic function for the interval $I = [1/4, 3/4]$ are plotted on $[0,1]$. Right: The error in the approximation is plotted on a logarithmic scale against evaluation points $x \in [0,1)$.

In the context of linear algebra and the evaluation of functions of matrices, the dichotomy between rationals and polynomials is a natural one to consider, and this dissertation is largely focused on applications in numerical linear algebra. However, in Chapter 5 we develop data-driven rational approximation methods for the reconstruction of univariate signals. In this context, it makes sense to consider the trade-offs between rational models and other nonlinear or quasi-nonlinear models, such as those involving splines [42] or piecewise polynomials [125], radial basis functions (RBFs) [173], or wavelets [43]. The development and analyses of these methods are often closely connected [43, 169], and for particular tasks, the various benefits of each should be considered carefully. For example, a simple spline can be used to recover Figure 1.1 exactly, and in applications where all of the functions of interest are of this type, it may be that splines are more beneficial than rationals. As a general–purpose method for the automatic reconstruction of signals with wide-ranging behaviors, we find that rationals perform exceptionally well. We make extensive use of several of their less-heralded properties, including their connection to exponential sums, finite difference equations, and Hankel operators [27, 132], their representation in a form that is numerically stable to evaluate [8, 25, 51, 87], their use as a means for filtering noise [170], and their global properties (e.g., the locations of their poles), which can be used to detect and identify singularities [27, 166]. A major advantage of the rational approximation schemes that we apply in Chapter 5 is that unlike many schemes involving wavelets, RBFs, rationals, and windowing functions, our methods are data–driven. They do not involve tuning parameters (e.g, mother/father wavelets [43], shape parameters [173], etc), and they require no a priori knowledge about the expected locations or types of singularities.

Before turning to the development of general purpose univariate rational approximation methods in Chapter 5, we focus in Chapters 2-4 on a particularly important rational approximation problem that arises in numerical linear algebra in connection to the low rank properties of matrices with special displacement structures. The remainder of this chapter reviews the relevant material.

1.2 Low rank matrix approximation

Let $X \in \mathbb{C}^{m\times n}$. We are often interested in finding a low rank matrix Y such that the distance between X and Y is small in a norm of interest. For the operator and Frobenius norms, $\|\cdot\|_2$ and $\|\cdot\|_F$, respectively, the singular values of X completely characterize the extent to which this is possible. The singular value decomposition (SVD) of X is given by $U\Sigma V^*$, where $U \in \mathbb{C}^{m\times m}$, $V \in \mathbb{C}^{n\times n}$ are unitary, $[\cdot]^*$ denotes conjugate transposition, $\Sigma \in \mathbb{C}^{m\times n}$ is diagonal with entries $\Sigma_{jj} = \sigma_j(X)$, and $\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq \sigma_{\min\{m,n\}}(X) \geq 0$ are the singular values of X. As the following theorem shows, the best rank $\leq k$ approximation to X is given by $X_k^{SVD} = U(\,:\,,1:k)\Sigma(1:k,1:k)\,[V(:\;1\!:\!k)]^*$, where the column indexing follows MATLAB's convention (e.g., $U(\,:\,,J)$, $J \subset \{1,\ldots,n\}$, is the submatrix of U consisting of the columns indexed by the set J).

Theorem 1. *(Eckart-Young-Mirsky Theorem) Let $X, Y_k \in \mathbb{C}^{m\times n}$, $\operatorname{rank}(Y_k) = k$. Then,*

$$\sigma_{k+1}(X) = \|X - X_k^{SVD}\|_2 \leq \|X - Y_k\|_2,$$

$$\sqrt{\sum_{j=k+1}^{\min\{m,n\}} \sigma_j^2(X)} = \|X - X_k^{SVD}\|_F \leq \|X - Y_k\|_F.$$

Proof. See [65, sec. 2.5.3]. □

In many problems, a tolerance parameter $0 < \epsilon < 1$ is supplied, and one must determine the smallest k such that there is Y_k satisfying $\|X - Y_k\|_2 \leq \epsilon\|X\|_2$. We refer to this as the ϵ-rank of X. Applying Theorem 1, it can be formally defined using the singular values of X in the following way:

Definition 1. *Let $X \in \mathbb{C}^{m\times n}$, $m \geq n$, with $0 < \epsilon < 1$ given. The ϵ-rank of X, denoted by $\operatorname{rank}_\epsilon(X)$, is the smallest integer $k \geq 0$ such that $\sigma_{k+1}(X) \leq \epsilon\,\|X\|_2$.*

In Figure 1.2, we illustrate how the ϵ-rank of X can easily be interpreted from a plot of its singular values. When $(m+n) \leq mn$, we say that X is of low numerical rank with respect to ϵ. While the ϵ-rank of a matrix is most reliably determined computationally using the SVD, this is usually too expensive ($\mathcal{O}(n^3)$ operations when $m = n$) to be practically useful. Alternative methods include the randomized SVD [79], iterative methods (e.g., Lanczos bidiagonalization [65, Ch. 10]) as well as methods based on various rank-revealing factorizations [72]. In Chapters 2 and 4, we develop low rank methods for important families of matrices that have special displacement structures (e.g., Toeplitz, Vandermonde, and Cauchy matrices). Using rational approximations, the singular values of these matrices can be bounded explicitly, so that good estimates for their ϵ-ranks are known outright. Moreover, the bounds can be derived in a constructive fashion that leads to efficient low rank approximation methods.

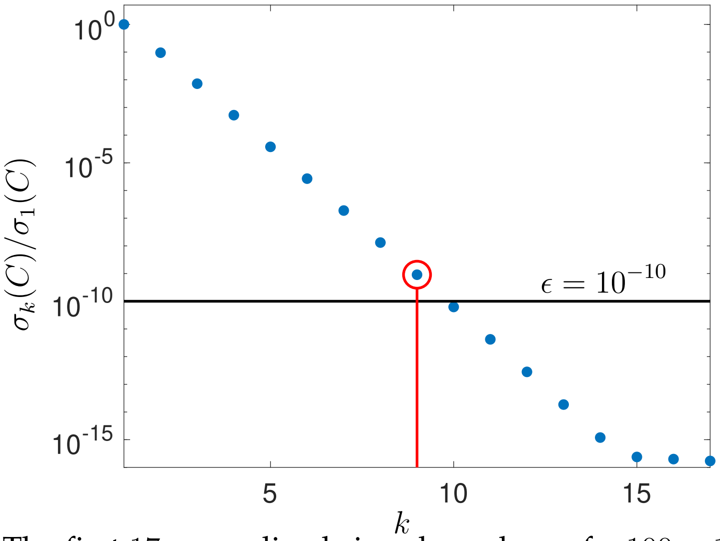


Figure 1.2: The first 17 normalized singular values of a 100×100 matrix C are plotted on a logarithmic scale against the indices of the singular values. C is a full rank matrix, but it is well-approximated by low rank matrices. The singular values show that for $\epsilon = 10^{-10}$, $\text{rank}_\epsilon(C) = 9$.

1.3 Matrices with displacement structure

A matrix X that satisfies the Sylvester matrix equation

$$AX - XB = F, \quad X, F \in \mathbb{C}^{m\times n}, \quad A \in \mathbb{C}^{m\times m}, \quad B \in \mathbb{C}^{n\times n}, \tag{1.1}$$

is said to have displacement structure, with an (A, B)–displacement rank of $\rho = \mathrm{rank}(F)$. In many applications, $\rho = 1$ or 2, and X is said to be of low displacement rank. Many important matrix families in computational mathematics, including Toeplitz ($\rho = 2$), Hankel ($\rho = 2$), Vandermonde ($\rho = 1$), and Cauchy ($\rho = 1$) matrices, have low displacement rank. In [19], this property is used to explain why certain matrices (including Löwner, Pick, Cauchy, real Vandermonde, and real positive definite Hankel), are of low numerical rank. In addition to low rank properties related the displacement structure of X, a wealth of literature has been developed around exploiting algebraic structures inherited from (1.1). These have been used, for example, to formulate efficient solvers for linear systems involving X [63, 83, 96].

We denote by $\lambda(A)$ and $\lambda(B)$ the spectra of A and B, respectively. Existence and uniqueness of X is guaranteed whenever $\lambda(A) \cap \lambda(B)$ is empty [91, Thm. 4.4.6]. There are several ways to express X in closed form, including via resolvents, integrals of exponentials, power sums, and as a Neumann series (see the survey [151] for an overview). However, X usually cannot be computed directly from these forms due to stability and/or cost issues. A basic way to compute X is to reshape (1.1) and solve the equivalent $mn \times mn$ linear system $\mathcal{A}\mathbf{x} = \mathbf{f}$, where $\mathbf{x}$ and $\mathbf{f}$ are the column-major vectorizations of X and F, respectively. Here, $\mathcal{A} = I_n \otimes A - B^T \otimes I_m$, where $\otimes$ is the Kronecker product operator and I_m is the $m \times m$ identity matrix. Another direct solver, especially useful when A and B are dense, is the $\mathcal{O}(m^3 + n^3)$ Bartels–Stewart method [64]. In this approach, the Schur factorizations of A and B are computed, and then a forward/backward-substitution scheme is used to compute the entries of X.

In most practical applications where (1.1) appears, such as in the discretization of PDEs [53, 159, 176] or in stability analysis for dynamical systems [5, 73, 128], A and B are sparse and/or structured (e.g. banded, banded+low rank), and F is low rank. Large problem dimensions can make it infeasible to directly store X, which is typically dense, let alone apply expensive direct solvers to (1.1). Instead, practitioners turn to iterative low rank approximation methods that can take advantage of the structures in A, B and F. These primarily include methods based on (rational) Krylov subspace projection [47, 150, 151] and methods based on the factored alternating direction implicit (fADI) [21, 128, 143, 107], though there are others [22, 126]. Of course, low rank approximation methods are only effective when X has low numerical rank. In the next section, we use the fADI algorithm to illustrate how the properties of A, B and F relate to the numerical rank of X. We remark that there are many ways to see this connection: In the important case where (1.1) is a Lyapunov matrix equation (i.e., $B = -A^*$, $F = F^*$), various arguments with qualitative bounds on the singular values of X [6, 17, 69, 143, 104, 151] have been used to justify low rank methods.

1.4 The ADI method

The ADI algorithm is an iterative method that numerically solves (1.1) by alternately updating the column and row spaces of an approximate solution [109, 127]. It is the main workhorse for most of the methods described in Ch. 2 and 4 of this thesis, and is fundamentally connected to the central rational approximation problem of interest in these chapters via its error equation.

One ADI iteration consists of the following two steps:

1. Solve for $X^{(j+1/2)}$, where

$$\left(A - \beta_{j+1} I\right) X^{(j+1/2)} = X^{(j)} \left(B - \beta_{j+1} I\right) + F. \tag{1.2}$$

2. Solve for $X^{(j+1)}$, where

$$X^{(j+1)} \left(B - \alpha_{j+1} I\right) = \left(A - \alpha_{j+1} I\right) X^{(j+1/2)} - F. \tag{1.3}$$

An initial guess, usually $X^{(0)} = 0$, is required to begin the iterations. The construction of $X^{(k)}$ requires selecting a set of k 2-tuples, $\{(\alpha_j, \beta_j)\}_{j=1}^{k}$, referred to as *shift parameters*.

1.4.1 Deriving the ADI iteration

To derive the ADI iteration from first principles, we first observe that for any pair of real numbers (α, β), it is true that

$$(A - \beta I_m) X (B - \alpha I_n) - (A - \alpha I_m) X (B - \beta I_n) = (\beta - \alpha) F. \tag{1.4}$$

This leads to a natural iteration:

$$X^{(j+1)}(B - \alpha I_n) = (\beta - \alpha)(A - \beta I_m)^{-1} F + (A - \beta I_m)^{-1}(A - \alpha I_m) X^{(j)} (B - \beta I_n). \tag{1.5}$$

Now we observe that

$$(\beta - \alpha)(A - \beta I_m)^{-1} = -I + (A - \alpha I_m)(A - \beta I_m)^{-1}, \tag{1.6}$$

and substitute this into (1.5). As a result, we have that

$$X^{(j+1)}(B-\alpha I_n) = -F+(A-\alpha I_m)(A-\beta I_m)^{-1}F+(A-\beta I_m)^{-1}(A-\alpha I_m)X^{(j)}(B-\beta I_n). \tag{1.7}$$

Now, we choose $X^{(j+1/2)}$ so that

$$(A-\beta I_m)X^{(j+1/2)} - F = X^{(j)}(B-\beta I_n).$$

By substituting this expression into (1.7) and observing that $(A-\alpha I_m)$ and $(A-\beta I_m)^{-1}$ commute, we recover the two-step ADI iteration shown in (1.2) and (1.3).

The ADI method was originally derived as (and remains widely known as) an implicit-explicit scheme for numerically solving the heat equation, though its potential as a solver for the Lyapunov (and then Sylvester) matrix equation was quickly recognized [127, 171]. ADI can also be viewed as a generalization of Smith's method [153].

1.4.2 The fADI method

The fADI method [21], first introduced as Cholesky-factored ADI for the Lyapunov matrix equation [107], is equivalent to ADI, but computes $X^{(k)}$ in factored form. The fADI iteration is derived by expressing $X^{(j)}$ in terms of $X^{(j-1)}$ using (1.2) and (1.3), and then substituting the factorizations $X^{(j)} = W^{(j)}D^{(j)}Y^{(j)^*}$ and $F = MN^*$, where $M \in \mathbb{C}^{m\times\rho}$ and $N \in \mathbb{C}^{n\times\rho}$, into the resulting equation. After k iterations, the following block matrices are constructured:

$$W^{(k)} = \left[\hat{W}^{(1)} \,\middle|\, \hat{W}^{(2)} \,\middle|\, \cdots \,\middle|\, \hat{W}^{(k)} \right], \quad \begin{cases} \hat{W}^{(1)} = (A - \beta_1 I)^{-1} M, \\ \hat{W}^{(j+1)} = (A - \alpha_j I)(A - \beta_{j+1} I)^{-1} \hat{W}^{(j)}, \end{cases} \tag{1.8}$$

$$Y^{(k)} = \left[\hat{Y}^{(1)} \,\middle|\, \hat{Y}^{(2)} \,\middle|\, \cdots \,\middle|\, \hat{Y}^{(k)} \right], \quad \begin{cases} \hat{Y}^{(1)} = (B^* - \overline{\alpha}_1 I)^{-1} N, \\ \hat{Y}^{(j+1)} = (B^* - \overline{\beta}_j I)(B^* - \overline{\alpha}_{j+1} I)^{-1} \hat{Y}^{(j)}, \end{cases} \tag{1.9}$$

$$D^{(k)} = \text{diag}\left((\beta_1 - \alpha_1) I_\rho, \ldots, (\beta_k - \alpha_k) I_\rho \right). \tag{1.10}$$

Using fADI, one clearly sees that after k iterations, the rank of the approximant $X^{(k)}$ is at most $k\rho$. By Theorem 1, we conclude that

$$\sigma_{k\rho+1}(X) \leq \|X - X^{(k)}\|_2, \qquad 0 \leq k\rho < n - 1. \tag{1.11}$$

Bounds on the numerical rank of X can be attained by bounding the infimum of the set $\{\|X - X^{(k)}\|_2, \quad X^{(k)} \in \mathcal{M}^{(k)}(A, B, F)\}$, where $\mathcal{M}^{(k)}(A, B, F)$ is the collection of all possible matrices that can be constructed by applying k steps of ADI (or, equivalently, fADI) to (1.1).

1.4.3 The ADI error equation

Using (1.2) and (1.3), one finds that the ADI error equation can be expressed as

$$X - X^{(k)} = r_k(A)(X - X^{(0)}) r_k(B)^{-1}, \qquad r_k(z) = \prod_{j=1}^{k} \frac{z - \alpha_j}{z - \beta_j}, \qquad k \geq 1. \tag{1.12}$$

Assuming that $X^{(0)}$ is chosen as the zero matrix, it follows that

$$\|X - X^{(k)}\|_2 \leq \|r_k(A)\|_2 \, \|r_k(B)^{-1}\|_2 \|X\|_2. \tag{1.13}$$

To minimize the bound on $\|X - X^{(k)}\|_2$, ADI shift parameters $\{(\alpha_j, \beta_j)\}_{j=1}^{k}$ are sought that minimize $\|r_k(A)\|_2 \|r_k(B)^{-1}\|_2$. When A and B are normal matrices, we have that

$$\|r_k(A)\|_2 \; \|r_k(B)^{-1}\|_2 \leq \sup_{z \in E} |r_k(z)| \sup_{z \in G} \frac{1}{|r_k(z)|} = \frac{\sup_{z \in E} |r_k(z)|}{\inf_{z \in G} |r_k(z)|}, \tag{1.14}$$

where $\lambda(A) \subset E$, $\lambda(B) \subset G$. In the general case, E and G can be taken as sets containing the fields of values of A and B, respectively. Then, we have by [39] that

$$\|r_k(A)\|_2 \; \|r_k(B)^{-1}\|_2 \leq (1 + \sqrt{2})^2 \frac{\sup_{z \in E} |r_k(z)|}{\inf_{z \in G} |r_k(z)|}. \tag{1.15}$$

Bounds involving other spectral sets and measures of nonnormality for A and B can also be applied [12, 19]. Unless otherwise specified, we assume for convenience throughout the text that A and B are normal matrices.

1.5 Zolotarev's third problem

The bounds in (1.14) and (1.15) let us approach the problem of finding optimal ADI shift parameters with tools from rational approximation theory. Specifically, we seek the rational function that attains the following infimum:

$$Z_k(E, G) := \inf_{r \in \mathcal{R}^k} \frac{\sup_{z \in E} |r(z)|}{\inf_{z \in G} |r(z)|}, \tag{1.16}$$

where $\mathcal{R}^k$ is the space of all rational functions with numerators and denominators both of degree $\leq k$. The number Z_k is referred to as the kth Zolotarev number associated with sets E and G, and a rational function in $\mathcal{R}^k$ that attains the infimum in (1.16) is called a Zolotarev rational function. The names are in honor of Y.I. Zolotarev, a student of Chebyshev who first posed (and subsequently solved) a version of the extremal approximation problem shown in (1.16) [182].

Among his other achievements, Zolotarev stated and solved 4 famous approximation problems [1, 157]; the one in (1.16) is referred to as Zolotarev's third problem.[1] The connection between ADI and Zolotarov's third problem were first pointed out by Lebedev in [105]. Zolotarev rationals arise in connection to many other applications, including in digital filter design [106, 75], the computation of matrix functions and polar decompositions [61, 120], approximation by sums of exponentials [29], and more.[2]

The following theorem is a restatement of the main observation in [19, Thm. 2.1]. It summarizes the links between the ADI error equation, the singular values of matrices with low displacement rank, and the Zolotarev numbers:

Theorem 2. *Let $X \in \mathbb{C}^{m \times n}$ satisfy the Sylvester matrix equation $AX - XB = F$, where A and B are normal matrices and $\mathrm{rank}(F) \leq \rho$. Suppose E and G are sets such that $\lambda(A) \subset E$ and $\lambda(B) \subset G$. Then, for k such that $1 \leq k\rho + 1 \leq \min(m, n)$,*

$$\sigma_{k\rho+1}(X) \leq \|X - X_Z^{(k)}\|_2 \leq Z_k(E, G)\|X\|_2, \tag{1.17}$$

where $X_Z^{(k)}$ is constructed by applying k steps of ADI (or fADI) to $AX - XB = F$, with the zeros and poles of the Zolotarev rational function associated with $Z_k(E, G)$ used as the ADI shift parameters.

Proof. The first inequality is given by (1.11). The last inequality follows directly from (1.14) and the definition of $Z_k(E, G)$. □

The bound in (1.17) is one instance of the more general result from [19,

[1] In some literature, the terms 'Zolotarev number' or 'Zolotarev function' are used in association with Zolotarev's fourth problem.

[2] Many of these applications concern Zolotarev's fourth problem, but the third and fourth problems are mathematically equivalent [94].

Thm. 2.1], where it is shown that for $1 \leq k\rho + j \leq \min(m, n)$,

$$\sigma_{j+k\rho}(X) \leq Z_k(E, G)\sigma_j(X). \tag{1.18}$$

A nice corollary to this result is that $Z_k(E, G)$ also bounds the relative error for the best rank k approximation to X in the Frobenius norm.

Corollary 1. *Let A, B, X, and F be as in Theorem 2. If $X_{\rho k}^{SVD}$ is the best rank ρk approximation to X in the Frobenius norm, then*

$$\|X - X_{\rho k}^{SVD}\|_F \leq Z_k(E, G)\|X\|_F.$$

Proof. See [149, Lemma 4.1]. □

Corollary 1 is especially valuable in the context of low rank tensor decompositions [149]. We remark that theorems similar to Theorem 2 can be stated for non-normal matrices using (1.15) or bounds based on other K-spectral sets (see [19, Cor. 2.2]).

Properties of Zolotarev numbers

As a general rule, $Z_k(E, G)$ decays rapidly with k whenever E and G are well-separated from one another, and the rate of decays slows as E and G are brought closer together. To see why, consider a simple example. Let $E = [10, 20]$ and let G be an interval disjoint from E. When E and G are closer together, it requires more degrees of freedom to construct a rational that remains small on E and also becomes large on G. In Figure 1.3, we illustrate this by plotting the $(4, 4)$ Zolotarev rationals r_4^E for $(E, -E)$ and $r_4^{G'}$ for (E, G'), where $G' = [-10, 0]$. Notice that $\max_{x \in E} |r_4^E(x)| < \max_{x \in E} |r_4^{G'}(x)|$ on E and

$\min_{x\in -E}|r_4^E(x)| > \min_{x\in G'}|r_4^{G'}(x)|$. The magnitudes of $Z_k(E,-E)$ and $Z_k(E,G')$ both decay exponentially fast as a function of k, but the rate of decay is slower for $Z_k(E,G')$ (see Theorem 3).

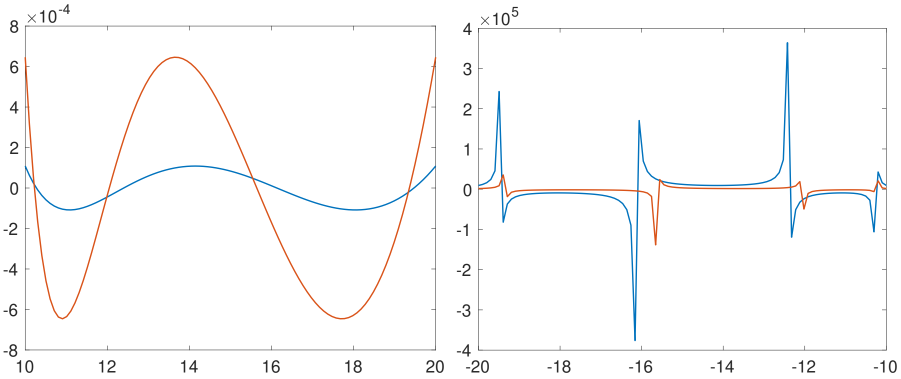


Figure 1.3: Left: The Zolotarev rational functions $r_4^E(z)$ associated with $(E,-E)$ (blue) and $r_4^{G'}(z)$ associated with (E,G') (red) are plotted for $z\in E$. Right: The rationals $r_4^E(z)$ and $r_4^{G'}(z)$ plotted for $z\in -E$ and $z\in G'$, respectively. The values of $r_4^{G'}$ have been translated to the left by 10 units for comparison's sake.

In the next lemma, we state several useful properties of Zolotarev numbers.

Lemma 1 (Zolotarev number properties). *For sets E and G, the following properties hold:*

(P1) $Z_0(E,G)=1$,

(P2) $Z_k(E,G)=Z_k(G,E)$,

(P3) $Z_{k+1}(E,G)\leq Z_k(E,G)$,

(P4) $Z_{k_1+k_2}(E,G)\leq Z_{k_1}(E,G)Z_{k_2}(E,G)$,

(P5) $Z_k(E_1,G_1)\leq Z_k(E,G)$ *whenever* $E_1\subseteq E$, $G_1\subseteq G$,

(P6) $Z_k(\mathcal{T}(E),\mathcal{T}(G))=Z_k(E,G)$, *where* $\mathcal{T}$ *is a Möbius transformation.*

Proof. See [19, sec. 3]. □

1.5.1 Bounding Zolotarev numbers

For general choices of E and G that are disjoint in $\mathbb{C}$, the solution to (1.16) is unknown. However, the asymptotic behavior of $Z_k(E, G)$ is known, along with a lower bound [66]:

$$Z_k(E,G) \geq h^{-k}, \quad \lim_{k\to\infty} \left(Z_k(E,G)\right)^{1/k} = h^{-1}, \quad h = \exp\left(\frac{1}{\mathrm{cap}(E,G)}\right). \tag{1.19}$$

Here, $\mathrm{cap}(E, G)$ is the capacity of a condenser consisting of plates E and G.

In view of (1.19), upper bounds on $Z_k(E, G)$ are sought that take the form

$$Z_k(E,G) \leq K_{E,G}\, h^{-k}, \tag{1.20}$$

where ideally $K_{E,G}$ is a constant that depends only on the geometry of E and G. For example, when E and G are intervals on the real line, $K_{E,G} = 4$ [19]. When they are disks, $K_{E,G} = 1$ [154]. In both cases, h is known (see Theorems 3 and 4), as are closed form expressions of the Zolotarev rationals r_k that attain $Z_k(E, G)$ in (1.16). Moreover, the zeros and poles of r_k are known and serve as optimal ADI shift parameters. A summary of known bounds for Zolotarev numbers is given in Table 1.1. The first three rows of the table are derived from special cases where Zolotarev's third problem has been solved exactly. We review the the first two of these results below. The bounds in the third row are discussed in Chapter 4, and bounds described in last row are discussed in Chapter 3 along with bounds for much more general choices of E and G.

Zolotarev rationals on intervals of the real line

Zolotarev's original solution to (1.16) considers only the case where E and $G = -E$ are two real intervals symmetric about the origin. In [19], analytic

Disjoint sets E and G	Bound	Reference
finite intervals of $\mathbb{R}$	$Z_k(E,G) \leq 4h^{-k}$	Theorem 3, [19]
disks in $\mathbb{C}$	$Z_k(E,G) \leq h^{-k}$	Theorem 4, [154]
arcs on a circle $\mathbb{C}$	$Z_k(E,G) \leq 4h^{-k}$	Theorem 8
convex closed polygons† in $\mathbb{C}$	$Z_k(E,G) \leq 9h^{-k} + \mathcal{O}(h^{-2k})$	Theorem 7, [141]

Table 1.1: A collection of known bounds for Zolotarev numbers associated with various sets E and G. Additional bounds for more general choices of E and G are discussed Chapter 3 and [141]. † This bound also holds for more general open convex sets in $\mathbb{C}$ (see Chapter 3).

descriptions of $Z_k([-\tau,-1],[1,\tau])$, $\tau > 1$, are bounded and then used to derive explicit bounds on $Z_k([a,b],[c,d])$, where $[a,b]$ and $[c,d]$ are taken to be disjoint, finite intervals of the real line. We make extensive use of the following theorem[3]:

Theorem 3. *Let $k > 0$ be an integer and let $[a,b]$ and $[c,d]$ be disjoint intervals on the real line. Then,*

$$Z_k([a,b],[c,d]) \leq 4h^{-k} \leq 4\mu_0^{-2k}, \quad \mu_0 = \exp\left(\frac{\pi^2}{2\log(16\gamma)}\right), \tag{1.21}$$

where $\gamma = (|c-a|\,|d-b|)/(|c-b|\,|d-a|)$ is the modulus of the cross-ratio of the points (a,b,c,d)

Proof. See Cor. 3.2 and Cor. 4.2 in [19]. □

We remark that a closed form expression for $h = \exp(1/\mathrm{cap}(E,G))$ involving the Grötzsch ring function can be found in [19, Thm. 3.1]. Zolotarev completely described the rational associated with $Z_k([-\tau,-1],[1,\tau])$ by expressing the zeros and poles of r_k using elliptic integrals [182]. A slight generalization of this result follows directly from (P6) in Lemma 1, and we have the following corollary to Theorem 3.

[3] Similar bounds are derived in [143, eq. 2.13] and [29], and related arguments are also found in [49].

Corollary 2. *Let $r_k(z)$ be the Zolotarev rational associated with $Z_k([a,b],[c,d])$, where $[a,b]$, $[c,d]$ are as in Theorem 3 and γ is the modulus of the cross-ratio of (a,b,c,d). Then, each zero α_j and pole β_j of r_k is as follows:*

$$\alpha_j = \mathcal{T}\left(-\tau \,\mathrm{dn}\left[\frac{2j+1}{2k}K(\Xi),\Xi\right]\right), \quad \beta_j = \mathcal{T}\left(\tau \,\mathrm{dn}\left[\frac{2j+1}{2k}K(\Xi),\Xi\right]\right), \quad (1.22)$$

where $\tau = -1+2\gamma+2\sqrt{\gamma^2-\gamma}$, $\Xi = \sqrt{1-1/\tau^2}$, K is the complete elliptic integral of the first kind [121, 19.2.8], $\mathrm{dn}(z,\Xi)$ is the Jacobi elliptic function of the third kind [121, 22.2.6], and $\mathcal{T}$ is a Möbius transformation such that $\mathcal{T}(a)=-\tau$, $\mathcal{T}(b)=-1$, $\mathcal{T}(c)=1$, $\mathcal{T}(d)=\tau$.

Proof. This result follows immediately from the solution of Zolotarev's third problem in [182] (see [1, Sec. 51] and the invariance of $Z_k(E,G)$ under Möbius transformations (P(6) in Lemma 1). □

We remark that our software package freeLyap [175] includes a subroutine that automatically computes ADI shift parameters $\{(\alpha_j,\beta_j)\}_{j=1}^k$ given any two disjoint intervals on the line. The routine uses MATLAB's `ellipk` and `ellipj` functions and has a trivial cost.

Zolotarev rationals on disks in the complex plane

The exact solution to Zolotarev's third problem is known when E and G are two disks in the complex plane. Explicit expressions for a special case are supplied in the next theorem.

Theorem 4. *Let $E = \{z \in \mathbb{C} : |z-z_0| \leq \eta\}$, $0 < \eta < z_0$, $z_0, \eta \in \mathbb{R}$. Then, the infimum in* (1.16) *is attained by the rational function*

$$r_k(z) = \left(\frac{z-\phi}{z+\phi}\right)^k, \qquad \phi = \sqrt{z_0^2-\eta^2}, \qquad (1.23)$$

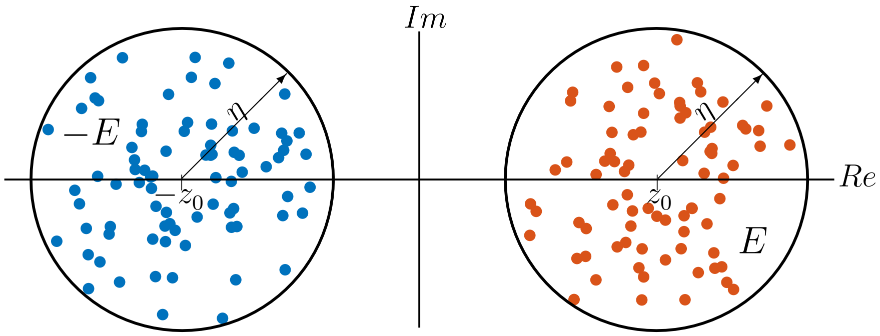


Figure 1.4: The value of $Z_k(E,-E)$ is known [154] when $E = \{z \in \mathbb{C} : |z - z_0| \leq \eta\}$, with $0 < \eta < z_0$. It can be used to bound the singular values of X satisfying $AX - XB = F$, where $\lambda(A) \subset E$ and $\lambda(B) \subset -E$.

and $Z_k(E,-E)$ *is given by*

$$Z_k(E,-E) = \mu_1^{-k}, \qquad \mu_1 = \exp\left(\frac{1}{\text{cap}(E,-E)}\right) = \frac{z_0+\phi}{z_0-\phi}. \tag{1.24}$$

Proof. The proof in Theorem 3.1 in [154] and the related discussion uses the near-circularity criterion. One can also apply an argument using conformal mapping (see Section 3.2). □

A similar result holds when E and G are general disjoint disks in $\mathbb{C}$. The expression for $Z_k(E,G)$ and the associated r_k is more technical. However, as in Theorem 4, the Zolotarev rational associated with $Z_k(E,G)$ has only one pole and one zero, each of multiplicity k. The optimal strategy for applying ADI in this case is to repeatedly apply the same shift parameter pair (e.g., $(\phi,-\phi)$ in Theorem 4) at every iteration. This is equivalent to Smith's method [153]. Our software package freeLyap [175] includes a routine that automatically computes the optimal shift parameter pair and the value of $Z_k(E,G)$ from the radii and centers of E and G.

1.5.2 A comparison with exponential sums

Completely monotonic functions such as $1/x$ and $\sqrt{x}$ on the interval $[a, b]$, $0 < a < b < \infty$, are well-approximated by exponential sums of the form

$$S_k(x) = \sum_{j=1}^{k} \omega_j e^{-t_j x}, \qquad \omega_j, t_j \in \mathbb{R}.$$

In [30], explicit bounds on the error $\|1/x - U_k(x)\|_{L_\infty([a,b])}$ are given, where U_k is the best approximation to $1/x$ by an exponential sum of at most k terms. The derivation of these bounds involves the construction of type $(k, k-1)$ rational approximations to $\sqrt{x}$ on $[a^2, b^2]$, and this in turn is related to rational approximations of the sign function on $[-b, -a] \cup [a, b]$,[4] i.e., Zolotarev's fourth problem [19].

Estimates from exponential sums can be used to bound the singular values of $X \in \mathbb{R}^{n \times n}$, where X satisfies (1.1). Here, we describe how this can be done and compare it to estimates derived from ADI. To simplify notation, we consider the Lyapunov equation $AX + XA^T = BB^T$, $\lambda(A) \subset [a, b]$, where $A \in \mathbb{C}^{m \times m}$ is a normal matrix [5]. Let $\mathcal{A} = I_n \otimes A + A \otimes I_n$. Then, $\mathbf{x} = \mathcal{A}^{-1}\mathbf{b}$, where $\mathbf{x}, \mathbf{b} \in \mathbb{C}^{n^2 \times 1}$ are the column-major vectorizations of X and BB^T, respectively. We can use the fact that $U_k(\mathcal{A}) \approx \mathcal{A}^{-1}$ to approximate X. Applying results from [19] and [30], we have that

$$\left\|\mathcal{A}^{-1} - U_k(\mathcal{A})\right\|_2 \leq \frac{K_\gamma k}{a} \mu_0^{-2k}, \tag{1.25}$$

where K_γ is a bounded constant dependent on $\gamma = a/b$, and μ_0 is as in (1.21).[6]

[4] The authors in [30] do not explicitly discuss Zolotarev numbers, but they present a collection of several fascinating methods for deriving bounds on $Z_k(E, G)$.

[5] This idea and the larger connection between the literature on best approximations by exponential sums and Zolotarev's results was first brought to my attention by Daniel Kressner.

[6] In [30], the left-hand side of (1.25) is bounded by an expression involving the Grötszch ring function [121, (19.2.8)], which is associated with elliptic integrals. The bound we give here uses instead Theorem 3.

Let $\tilde{X}$ be the $m \times m$ matrix associated with the vectorization $\tilde{\mathbf{x}} = U_k(\mathcal{A})\mathbf{b}$. The property $\exp(I_m \otimes A + A \otimes I_m) = \exp(A) \otimes \exp(A)$ can be used to show that $\mathrm{rank}(\tilde{X}) \leq k\rho$, where $\mathrm{rank}(BB^T) = \rho$. Since $\|X\|_F = \|\mathbf{x}\|_2$, where $\|\cdot\|_F$ is the Frobenius norm, it follows that

$$\sigma_{k\rho+1}(X) \leq \left\| \mathcal{A}^{-1}\mathrm{vec}(BB^T) - U_k(\mathcal{A})\mathrm{vec}(BB^T) \right\|_2 \leq \frac{K_\gamma k}{a} \mu_0^{-2k} \; \|BB^T\|_F.$$

To state a relative bound, we use the estimate $1/\|X\|_2 \leq 2b/\|BB^T\|_2$, so that

$$\sigma_{k\rho+1}(X) \leq \tilde{K}_\gamma k \mu_0^{-2k} \; \|X\|_2, \quad \tilde{K}_\gamma = \frac{2K_\gamma b}{a} \frac{\|BB^T\|_F}{\|BB^T\|_2}. \tag{1.26}$$

From Theorem 2 and Theorem 3, a bound attained via fADI is given by

$$\sigma_{k\rho+1}(X) \leq 4\mu_0^{-2k} \|X\|_2. \tag{1.27}$$

We observe that the bounds in (1.26) and (1.27) both achieve the same geometric decay rate, with the ADI-based bound resulting in a cleaner constant that does not include a factor of k.

1.5.3 Asymptotically optimal rational functions

For general sets of E and G in the complex plane, exact solutions to (1.16) remain unknown. One way to study $Z_k(E, G)$ in such a setting is to consider rationals s_k that behave similarly to Zolotarev rationals in an asymptotic sense. We say that $\{s_k\}_{k=1}^{\infty}$ is a set of asymptotically optimal rational functions if

$$\lim_{k \to \infty} \left(\frac{\sup_{z \in E} |s_k(z)|}{\inf_{z \in G} |s_k(z)|} \right)^{1/k} = h^{-1}, \qquad h = \exp\left(\frac{1}{\mathrm{cap}(E, G)} \right). \tag{1.28}$$

Logarithmic potential theory supplies a useful framework for studying and deriving sets of asymptotically optimal rational functions [154, 172]. It has been used to derive bounds on $Z_k(E, G)$ [66, 174], as well as develop heuristics for generating ADI shift parameters [154] and pole parameters in the rational Krylov subspace method (RKSM) [47] for solving (1.1) when A, B have complex-valued spectral sets. The fundamental connection between approximation theory and electrostatics is that the function $\log(1/|p(z)|)$, where $p(z) = \prod_{j=1}^{k}(z - z_j)$, can be expressed as a logarithmic potential:

$$\log \frac{1}{|p(z)|} = \int \log \frac{1}{|z-t|} \mathrm{d}\nu(t),$$

where ν is the discrete measure with mass 1 at each of the zeros z_j [145] . For rational functions, $-\log|p(z)/q(z)| = \log|1/p(z)| - \log|1/q(z)|$. The poles and zeros of a rational function s_k can be viewed as charged particles interacting on the plates E and G of the condenser (E, G). One ideally seeks an arrangement of poles and zeros that minimizes $|s_k(z)|$ for z on E and $|1/s_k(z)|$ for z on G. Instead of seeking optimal solutions for a fixed k, one can consider the problem in a distributional sense. With (1.28) in mind, we seek nonnegative and normalized measures ν_1 and ν_2, where $\mathrm{supp}(\nu_1) = E$ and $\mathrm{supp}(\nu_2) = G$, such that the following energy integral is minimized:

$$\mathcal{I}(\nu) = \int\int \log \frac{1}{|z-t|} \mathrm{d}\nu(z)\mathrm{d}\nu(t), \quad \nu = \nu_1 - \nu_2. \tag{1.29}$$

The condenser capacity is defined as $\mathrm{cap}(E, G) := 1/\inf_{\nu} \mathcal{I}(\nu)$ [145]. For some choices of E and G, this problem can be easily solved. For example, consider the sets E_1 and G_1, where E_1 is a disk of radius 1, and G_1 is an external disk of radius $h > 1$. One can show that [145] that (1.29) is minimized by choosing ν_1 and ν_2 such that $\mathrm{d}\nu_1 = 1/(2\pi)\mathrm{d}s$ and $\mathrm{d}\nu_2 = 1/(2\pi h)\mathrm{d}s$. This suggests that a reasonable approximation to the Zolotarev rational r_k might be found by choosing s_k to have as its k zeros the kth roots of unity, and as its k poles, the kth roots of unity scaled by h. We know that in the limit, these points become distributed in a way that minimizes (1.29), and this can be used to show that these rationals are in fact asymptotically optimal [172]. One notes via Theorem 4 that for any fixed k, this solution is never optimal.

To illustrate how this is useful, consider the case where $\mathbb{C} \setminus E \cup G$ is a doubly-connected region. Since $\mathrm{cap(E, G)}$ is conformally invariant, we construct the conformal map $\Psi : \mathbb{C} \setminus E \cup G \to \mathcal{A}$, where $\mathcal{A}$ is the annulus $\{1 < |z| < h,\ z \in \mathbb{C}\}$. The collection of k generalized Fejér points associated with E and G [172] is defined as $\left\{ \left(\Psi(e^{2\pi ijt/k}), \Psi(he^{2\pi ijt/k} \right) \right\}_{j=1}^{k}$. It was shown by Walsh that if these are taken to be the zeros and poles of a rational function $s_k(z)$, then the sequence $\{s_k\}_{k=1}^{\infty}$ is asymptotically optimal [172] . A stable method for numerically constructing and evaluating the conformal map Ψ is described in [165] (see also Section 3.3.2).

Another set of asymptotically optimal rationals can be defined using the generalized Leja points, which are derived recursively from a greedy process [11]. The kth set of generalized Leja points $L_k = \{(\phi_j, \psi_j)\}_{j=1}^{k}$ associated with E and G is constructed so that it has a useful nesting property, $L_k \subset L_{k+1}$. This makes generalized Leja points suitable for on-the-fly shift parameter generation schemes [47]. Defining s_{k-1} as the type $(k-1, k-1)$ rational with zero-pole pairs given by L_{k-1}, one defines $L_k = L_{k-1} \cup \{(\phi_k, \psi_k)\}$, where (ϕ_k, ψ_k) is chosen so that $\max_{z \in E} |s_{k-1}(z)| = |s_{k-1}(\phi_k)|$ and $\min_{z \in G} |s_{k-1}(z)| = |s_{k-1}(\psi_k)|$. Useful properties of the generalized Leja points and practical methods for computing them can be found in [154].

In Chapter 3, we construct a special set of rational functions called the Faber rational functions. These functions also behave similarly to the Zolotarev rational functions, and in fact coincide with them in special cases (see Section 3.1).

Our results on the behavior of the Faber rational functions are stronger than asymptotic optimality, since we bound $\sup_{z\in E}|s_k(z)|/\inf_{z\in G}|s_k(z)|$ explicitly when s_k is a Faber rational and $k > N_0$, where N_0 is known. This leads to explicit bounds on $Z_k(E,G)$ for rather general assumptions on E, G in $\mathbb{C}$. For finite and relatively small choices of k, the Faber rationals can sometimes greatly outperform the Fejér and Leja rationals as an approximation to the Zolotarev rationals r_k. For example, in the case where E and G are taken as disjoint disks in $\mathbb{C}$, the Fejér and Leja rationals are slow to converge in (1.28) [154, sec. 2], whereas the Faber rationals are equivalent to r_k for all k (see Section 3.1). On the other hand, these differences appear to be less extreme for other choices of E and G. We make more comparisons in Section 3.5.2.

1.6 ADI in practice

ADI can be an extremely efficient method for solving $AX - XB = F$ in settings where solutions to Zolotarev's problem are well-understood. With some adaptations, ADI can also be applied more broadly. The "ADI-friendliness" of a given problem can be evaluated using the criteria listed below:

1. A and B are normal matrices,
2. $\operatorname{rank}(F)$ is small,
3. The spectra of A and B are contained in two disjoint and well-separated sets E and G,
4. A solution to Zolotarev's problem is known for the sets E and G,
5. Shifted linear solves and matrix-vector products involving A and B cost $\mathcal{O}(n)$ operations.

For ADI to be effective, criterion 3 is essential, since without it, ADI iterations may not converge rapidly. When several of the other criteria are also met, we say that the problem at hand has high ADI-friendliness. It can be beneficial to design methods for solving problems with ADI-friendliness in mind. For example, in [53], a spectral method is developed for solving Poisson's equation in various geometries. The discretizations involved are intentionally designed so that they lead to ADI-friendly Sylvester equations, and this in turn allows the authors to prove that the method has quasi-optimal computational complexity.

When not all of the above criteria are met, the application of ADI requires some modifications. In some settings, measures of non-normality for A and B can be bounded so that an explicit bound on $\|X - X^{(k)}\|_2$ is still possible even when criterion 1 is not met. In other settings, it may be possible to define E and G and find optimal ADI shift parameters using K-spectral sets, such as the fields of values for A and B [19].

When criteria 1 and 3 are met, then explicit bounds on the approximation error $\|X - X^{(k)}\|_2$ are given via Theorem 2 and optimal ADI shift parameters are available. The number of ADI iterations required can be determined a priori, and ADI hardly resembles an iterative algorithm, since there is no need to monitor convergence. Truly iterative variants of ADI are sometimes used when criterion 4 is not met, especially when estimates on the boundaries of E and G are not available. In this setting, heuristic strategies are applied to choose shift parameters. This might entail choosing a small set of shift parameters a priori and then applying them cyclically (the cyclic Smith's method [153, 128]), or it may involve adaptively computing shift parameters on the fly at each iteration [47, 151]. Without an explicit error bound, a stopping condition is required, which is typically based on monitoring a relative residual (see [151, sec. 4.1]).

When suboptimal shifts are used, ADI can converge poorly, though combining ADI with Galerkin projections can help alleviate this issue [21]. Krylov-based methods often perform better than ADI in these settings [21, 151]. However, they are not a silver bullet. The per-iteration costs of these methods increases quickly with the iteration number, and the development of acceleration strategies is its own challenge [47, 151]. Moreover, these methods can also require the selection of parameters. For example, the RKSM requires pole parameters, and analyses of the error behavior of the method involves bounds that depend on the ADI error equation [47, 151]. In other words, improvements in our understanding of how to choose ADI parameters can lead to improvements for the implementation of projection-based methods such as the RKSM.

In practice, the effectiveness of ADI-based low rank approximation methods is primarily limited by the lack of known solutions to (1.16) (criterion 4), which limits our ability to find good ADI shift parameters, as well as the requirement in Theorem 2 that $\mathrm{rank}(F)$ is small (criterion 2). Criterion 2 is violated, for example, in applications where (1.1) arises from spectral discretization of PDEs with smooth right-hand sides [53, 159, 149, 161, 176]. In Chapters 2 and 3 of this thesis, we tackle these problems and develop methods that extend the applicability of ADI to new regimes. Then in Chapter 4, we develop ADI-based compression methods for matrices possessing more complicated, rank-structured compression properties.

CHAPTER 2

LOW RANK APPROXIMATION FOR MATRICES WITH HIGH DISPLACEMENT RANK

In this chapter,[1] bounds are derived for the singular values of X in (1.1) in cases where the bound from Theorem 2, $\sigma_{k\rho+1}(X) \leq Z_k(E,G)\|X\|_2$, fails to be informative. Such bounds depend on the brittle assumption that $\rho = \mathrm{rank}(F)$ is small. Here, we tackle the more stable variant of this problem, where F only needs decay in its singular values, so that $\mathrm{rank}_\epsilon(F)$ is small. Our method is constructive and leads to an efficient low rank approximation scheme that we call the factored-independent alternating direction implicit (FI-ADI) method. It can be used, among other things, to develop fast and spectrally-accurate low rank solvers for Poisson's equation in various domains (see Section 2.6).

Our results are rooted in two fundamental observations:

- **Splitting property:** Equation (1.1) can be split into ρ matrix equations, each with a rank 1 right-hand side. Specifically, $X = \sum_{i=1}^{\rho} X_i$, and each X_i satisfies

$$AX_i - X_iB = \sigma_i(F)u_iv_i^*, \tag{2.1}$$

  where $\sum_{i=1}^{\rho} \sigma_i(F)u_iv_i^*$ is the singular value decomposition (SVD) of F.
- **Bounding property:** Bounds on the singular values of X_i exist that depend on the size of $\sigma_i(F)$.

Since (2.1) is a Sylvester equation with a rank–1 right hand side, Theorem 2 can be applied to bound the singular values of each X_i. Then, the bounding

[1] This chapter is based on a paper with Alex Townsend [160]. I developed the algorithms, theorems, and software described in the paper, and was the lead author in writing it.

property can be used to exploit the decay of the singular values of F (see Theorem 5). Our underlying proof technique applies a modification of the ADI method to (1.1) using the above two observations.

This chapter is organized as follows: In Section 2.1, we briefly review how Theorem 2 can be applied to bound the singular values of X when $\operatorname{rank}(F)$ is small, and we then illustrate what goes wrong when F has full rank. In Section 2.2 and Section 2.3, we develop a new method for bounding singular values when F has rapidly decaying singular values. We discuss three examples in Section 5.4. Section 2.5 describes a practical method for solving (1.1) in low rank form, and we apply this method to develop fast low rank Poisson solvers in Section 2.6.

2.1 Bounding the singular values of matrices with displacement structure

In [19], explicit bounds on the Zolotarev numbers are used to bound the singular values of various matrices with low displacement rank. To illustrate and briefly review this approach, we consider a Cauchy matrix $C \in \mathbb{C}^{m \times n}$, where we assume without loss of generality that $m \geq n$. The entries of C are given by $C_{ij} = 1/(z_i - w_j)$, where $\{z_i\}_{i=1}^{m}$ and $\{w_j\}_{j=1}^{n}$ are distinct collections of complex numbers. The matrix C satisfies the Sylvester matrix equation

$$D_z C - C D_w = \mathbf{1}, \tag{2.2}$$

where $D_z = \operatorname{diag}(z_1, \ldots, z_m)$, $D_w = \operatorname{diag}(w_1, \ldots, w_n)$, and $\mathbf{1}$ is the rank 1 $m \times n$ matrix of all ones. Note that D_z and D_w are normal matrices with

$\lambda(D_z) = \{z_i\}_{i=1}^m$ and $\lambda(D_w) = \{w_j\}_{j=1}^n$, respectively. The displacement rank of C with respect to D_z and D_w is $\text{rank}(D_z C - C D_w) = 1$. We have from Theorem 2 that

$$\sigma_{k+1}(C) \leq Z_k(E, G) \|C\|_2, \qquad 0 \leq k < n, \tag{2.3}$$

where $\{z_i\}_{i=1}^m \subset E$ and $\{w_j\}_{j=1}^n \subset G$. If we assume, for example, that $E = \{z \in \mathbb{C} : |z - z_0| \leq \eta\}$, $0 < \eta < z_0$, and $G = -E$, then by Theorem 4, it follows that

$$\sigma_{k+1}(C) \leq \mu_1^{-k} \|C\|_2, \qquad 0 \leq k < n, \tag{2.4}$$

where μ_1 is as in (1.24). This shows that the singular values of C decay at least geometrically, and that for $0 < \epsilon < 1$, $\text{rank}_\epsilon(C) \leq \lceil \log(1/\epsilon) / \log(\mu_1) \rceil$. Similar arguments can be made for various choices of E and G using the bounds in Table 1.1.

This approach becomes uninformative if the displacement rank of a matrix satisfying (1.1) is large. To see this, consider $\tilde{C} \in \mathbb{C}^{m \times n}$, $m \geq n$, with entries $\tilde{C}_{ij} = 1/|z_i - w_j|^2$. This matrix satisfies

$$\overline{D}_z \tilde{C} - \tilde{C} \overline{D}_w = C, \tag{2.5}$$

where $\overline{M}$ denotes entrywise complex conjugation on M, and C, D_z and D_w are as in (2.2). The singular values of C have rapid decay. However, since the displacement rank of $\tilde{C}$ is $\text{rank}(C) = n$, the bound in (1.17) can only be used to bound $\sigma_1(\tilde{C})$. This reveals nothing about whether $\tilde{C}$ has low numerical rank. Figure 2.2 (left) shows that the singular values of $\tilde{C}$ decay rapidly. A new approach is required to bound them explicitly.

2.2 Bounds via a modification of Smith's method

Because we have assumed E and G are disjoint disks as in Theorem 4, the optimal ADI shift selection strategy for solving (2.5) uses the same shift parameters, $\alpha_j = \phi$ and $\beta_j = -\phi$, where ϕ is given in (1.23), at every iteration. When this happens, the fADI method is equivalent to Smith's method [153]. We first consider bounding singular values in this setting.

Eq. (2.5) is a special case of (1.1), with $A = \overline{D}_z$ and $B = \overline{D}_w$ satisfying the assumptions in Theorem 4, and $F = C$. Applying k iterations of fADI to (2.5) constructs an approximant $X^{(k)} = W^{(k)}D^{(k)}{Y^{(k)}}^*$, where the factors are given by (1.8), (1.9), and (1.10). The dimensions of $W^{(k)}$ and $Y^{(k)}$ are $m \times k\rho$ and $n \times k\rho$, respectively. When $\rho = n$, it is often the case that these matrices have linearly dependent columns, and this leads to an overestimation of $\operatorname{rank}_\epsilon(X)$. However, in applying k iterations of fADI to (1.1), several potential low rank approximants to X have been generated in addition to $X^{(k)}$. To see this, write $X^{(k)}$ as a sum of $k\rho$ rank 1 terms,

$$X^{(k)} = \sum_{i=1}^{\rho}\sum_{j=1}^{k} \underbrace{d_{ij}\mathbf{w}_{ij}\mathbf{y}_{ij}^*}_{=T_{ij}}, \tag{2.6}$$

where $\mathbf{w}_{ij}$ and $\mathbf{y}_{ij}$ are the ith columns of the blocks $\hat{W}^{(j)}$ and $\hat{Y}^{(j)}$, respectively, in (1.8) and (1.9), and d_{ij} is the (i,i) entry of $\hat{D}^{(j)}$ in (1.10). The sum in (2.6) exactly recovers the solution X in the limit as $k \to \infty$.

We now represent X by arranging the rank 1 terms in (2.6) in a $\rho \times \infty$ rectangle $\mathcal{R}$, so that each T_{ij} is represented by the box in the ith row and jth column of $\mathcal{R}$ (see Figure 2.1). An approximant can be constructed by choosing any finite collection of boxes and summing together the terms they represent. For example, the fADI algorithm constructs $X^{(k)}$ by summing together the terms represented in the first k columns of $\mathcal{R}$, as shown in Figure 2.1 (left). A natural question to ask is whether this is the best choice.

To answer this question, we examine the error associated with these approximants. If $\tilde{X}_t$ is constructed from a collection $\mathcal{K}_t$ of t boxes in $\mathcal{R}$, then $\|X - \tilde{X}_t\|_2$ is bounded above by $\sum_{\{(i,j)\in\mathcal{R}\setminus\mathcal{K}_t\}} \|T_{ij}\|_2$. To approximately minimize the error, we choose $\mathcal{K}_t$ by selecting terms in decreasing order of their norms. Careful examination of the fADI method reveals that $\|T_{ij}\|_2$ is influenced by $Z_{j-1}(E, -E)$ and $\sigma_i(F)$: In (1.8) and (1.9), F is written as MN^*. Assign $M = U\Sigma$ and $N = V$, where $U\Sigma V^*$ is the SVD of F. It follows that

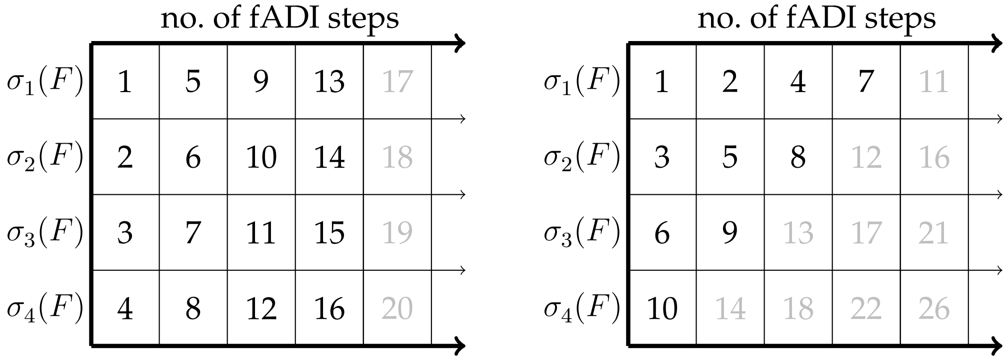
$$\|T_{ij}\|_2 \leq \frac{\phi}{2(z_0 - \eta)^2}\sigma_i(F) Z_{j-1}(E, -E), \tag{2.7}$$

where ϕ and $\tilde{r}_{j-1}(z)$ are given in (1.23).

Consider $\tilde{C}$ in (2.5). In this case, $Z_{j-1}(E, -E) = \mu_1^{-(j-1)}$ by Theorem 4. The right-hand side of (2.5) is the matrix C in (2.2), so it follows from (2.4) that $\|T_{ij}\|_2 \leq \phi\mu_1^{-(i+j-2)}\|C\|_2/(2(z_0-\eta)^2)$. This suggests that we construct $\tilde{X}_t$ by selecting rank 1 terms along the antidiagonals of $\mathcal{R}$ (see Figure 2.1 (right)). This strategy leads to bounds on the singular values of $\tilde{C}$ with indices that do not depend on $\rho = n$, since $\operatorname{rank}(\tilde{X}_t)$ is at most $k(k+1)/2$, as opposed to $k\rho$, and $\sigma_{k(k+1)/2+1}(\tilde{C}) \leq \|\tilde{C} - \tilde{X}_t\|_2$. The same reasoning applies for any matrix $X \in \mathbb{C}^{m\times n}$ satisfying (1.1), where A and B are as in Theorem 4 and $\sigma_i(F) \leq \mu_1^{-(i-1)}\|F\|_2$.

Explicit bounds on singular values

We now require explicit bounds on expressions of the form $\|X - \tilde{X}_t\|_2$. We find them using the splitting and bounding properties.



$\sigma_1(F)$	1	5	9	13	17
$\sigma_2(F)$	2	6	10	14	18
$\sigma_3(F)$	3	7	11	15	19
$\sigma_4(F)$	4	8	12	16	20

$\sigma_1(F)$	1	2	4	7	11
$\sigma_2(F)$	3	5	8	12	16
$\sigma_3(F)$	6	9	13	17	21
$\sigma_4(F)$	10	14	18	22	26

Figure 2.1: The box in the ith row and jth column represents the rank 1 term T_{ij} from (2.6). The terms reduce in norm as one applies successive ADI iterations (moving to the right), but they also reduce in norm as the indices of the singular values of F increase (moving down). In this illustration, we suppose that $\|T_{ij}\|_2 = \mathcal{O}(\mu_1^{-(i+j-2)})$ and $\mathrm{rank}(F) = 4$. Left: With $k = 4$, the fADI algorithm constructs $X^{(k)}$, where $\mathrm{rank}(X^{(k)}) \leq k^2 = 16$, by summing terms represented by the first k columns of the rectangle. The numbering of the boxes designates the order in which the rank 1 terms are constructed via fADI; decay in the singular values of F is not exploited. Right: The boxes are numbered in decreasing order with respect to their norms. Only the first $t = k(k+1)/2$ terms (numbered in black) are required to construct an approximant $\tilde{X}_t$ so that $\|X - \tilde{X}_t\|_2 \approx \|X - X^{(k)}\|_2$.

- **Applying the splitting property.** The strategy depicted in Figure 2.1 (right) is equivalent to splitting (1.1) into ρ equations and applying a different number of fADI iterations to each one. The ith row of $\mathcal{R}$ corresponds to the ith equation in (2.1). Applying s_i iterations of fADI to (2.1) results in $X_i^{(s_i)}$, where $\|\sum_{j=s_i+1}^{\infty} T_{ij}\|_2 = \|X_i - X_i^{(s_i)}\|_2$. The sum of these errors bounds the total error $\|X - \tilde{X}_t\|_2$, where $\tilde{X}_t = \sum_{i=1}^{\rho} X_i^{(s_i)}$ and $t = \sum_{i=1}^{\rho} s_i$.
- **Applying the bounding property.** For each X_i, we have a bound of the form $\|X_i - X_i^{(s_i)}\|_2 \leq Z_{s_i}(E, -E)\|X_i\|_2$. To find a bound that explicitly involves the singular value $\sigma_i(F)$, we use the following result:

Lemma 2. *Let $X \in \mathbb{C}^{m\times n}$ satisfy $AX - XB = F$ for normal matrices A and B. Further, suppose that $\lambda(A) \subset E$ and $\lambda(B) \subset -E$, where E is the disk*

$E := \{z \in \mathbb{C} : |z - z_0| \leq \eta\}$, *with* $z_0, \eta \in \mathbb{R}$ *and* $0 < \eta < z_0$. *Then,*

$$\|X\|_2 \leq \frac{\|F\|_2}{2(z_0 - \eta)}.$$

Proof. The lemma follows as a special case of [92, Thm. 2.1]. □

Applying Lemma 2 to (2.1), we find that $\|X_i\|_2 \leq \sigma_i(F)/(2(z_0 - \eta))$. Using this result, we can now derive explicit bounds on the singular values of X. We begin with the case where $\sigma_k(F)$ decays at the same rate as $Z_k(E, -E)$.

Theorem 5. *Let* $X \in \mathbb{C}^{m \times n}$, $m \geq n$, *satisfy* $AX - XB = F$, *with* $\lambda(A) \subset E$ *and* $\lambda(B) \subset -E$, *where* $E = \{z \in \mathbb{C} : |z - z_0| \leq \eta\}$, *with* $z_0, \eta \in \mathbb{R}$ *and* $0 < \eta < z_0$. *Suppose that for* $0 \leq j < n$, $\sigma_{j+1}(F) \leq K\mu_1^{-j}\|F\|_2$, *where* μ_1 *is given in* (1.24) *and* $K \geq 1$ *is a constant. For the triangular numbers* $1 \leq t = k(k+1)/2 < n$, *the singular values of* X *are bounded in the following way:*

$$\sigma_{t+1}(X) \leq K \frac{z_0 + \eta}{z_0 - \eta}\,(\tfrac{3}{2}\sqrt{t} + 1)\mu_1^{-(\sqrt{8t+1}-1)/2}\,\|X\|_2. \tag{2.8}$$

Proof. Let $\mathrm{rank}(F) = \rho$. Consider the approximant $\tilde{X}_t = \sum_{i=1}^{k}\sum_{j=1}^{k+1-i} T_{ij}$, where T_{ij} are given in (2.6). We allow the choice $k > \rho$ with the convention that for $s > \rho$, $\|T_{sj}\|_2 = 0$.[2] This corresponds to selecting terms along the antidiagonals of $\mathcal{R}$ in Figure 2.1. Since $\mathrm{rank}(\tilde{X}_t) \leq t = k(k+1)/2$, we have that $\sigma_{t+1}(X) \leq \|X - \tilde{X}_t\|_2$. The proof proceeds by bounding the approximation error $\|X - \tilde{X}_t\|_2$. The error equation is given by

$$X - \tilde{X}_t = \underbrace{\sum_{i=k+1}^{\rho}\sum_{j=1}^{\infty} T_{ij}}_{=S_1} + \underbrace{\sum_{i=1}^{k}\sum_{j=k+1-i}^{\infty} T_{ij}}_{=S_2}.$$

[2] For expository reasons, when $k > \rho$, we do not account for the non-contribution of the terms $\|T_{sj}\|_2 = 0$ in our bounds. This simple but notationally tedious task would improve the bounds associated with $k > \rho$.

Using the fact that $\sum_{j=1}^{\infty} T_{ij} = X_i$, where X_i is given in (2.1), we find that S_1 satisfies $AS_1 - S_1 B = \sum_{i=k+1}^{\rho} \sigma_i(F) u_i v_i^*$. It follows from Lemma 2 that

$$\|S_1\|_2 \leq \frac{\sigma_{k+1}(F)}{2(z_0 - \eta)} \leq \frac{K\|F\|_2 \mu_1^{-k}}{2(z_0 - \eta)}. \tag{2.9}$$

To bound $\|S_2\|_2$, observe that $S_2 = \sum_{i=1}^{k}(X_i - X_i^{(s_i)})$, where $X_i^{(s_i)}$ is constructed by applying $s_i = k+1-i$ steps of fADI to (2.1). For each i, we have

$$\|X_i - X_i^{(s_i)}\|_2 \leq Z_{s_i}(E, -E)\ \|X_i\|_2 \leq \frac{\sigma_i(F)}{2(z_0 - \eta)} \mu_1^{-s_i},$$

where Lemma 2 has been used to bound $\|X_i\|_2$. This implies that

$$\|S_2\|_2 \leq \frac{K\|F\|_2}{2(z_0 - \eta)} \sum_{i=1}^{k} \mu_1^{-(i-1)-s_i} \leq \frac{K\|F\|_2}{2(z_0 - \eta)} k \mu_1^{-k}, \tag{2.10}$$

and (2.9) and (2.10) together give the bound

$$\sigma_{t+1}(X) \leq \|X - \tilde{X}_t\|_2 \leq \frac{K\|F\|_2}{2(z_0 - \eta)} (k+1) \mu_1^{-k}. \tag{2.11}$$

To get a relative bound, we must divide the expressions in (2.11) by $\|X\|_2$. Trivially, the relation $AX - XB = F$ implies that $1/\|X\|_2 \leq (\|A\|_2 + \|B\|_2)/\|F\|_2$. Due to the assumptions on E we have $\|A\|_2 + \|B\|_2 \leq 2(z_0 + \eta)$. The theorem follows from the fact that $k = (\sqrt{8t+1} - 1)/2$, and for $t \geq 1$, $k \leq 3\sqrt{t}/2$. □

In Theorem 5, it is assumed for convenience that t is a triangular number. However, for any $1 \leq t < n$, a bound on $\sigma_{t+1}(X)$ is found by bounding the sum of the first t terms selected along the antidiagonals of $\mathcal{R}$ (see Figure 2.1 (right)). The constants in (2.8) are due to estimates on $\|X\|_2$, and are therefore not necessarily tight. The polynomial term in the bound is also suboptimal. However, as shown in the appendix of our related paper [160], there are A, B and F satisfying Theorem 5 so that for $1 \leq t \leq \rho(\rho+1)/2$, $\|X - \tilde{X}_t\|_2 \approx \sigma_{t+1}(X)$. This implies that $\tilde{X}_t$ is a near-best low rank approximation to X, and that the decay

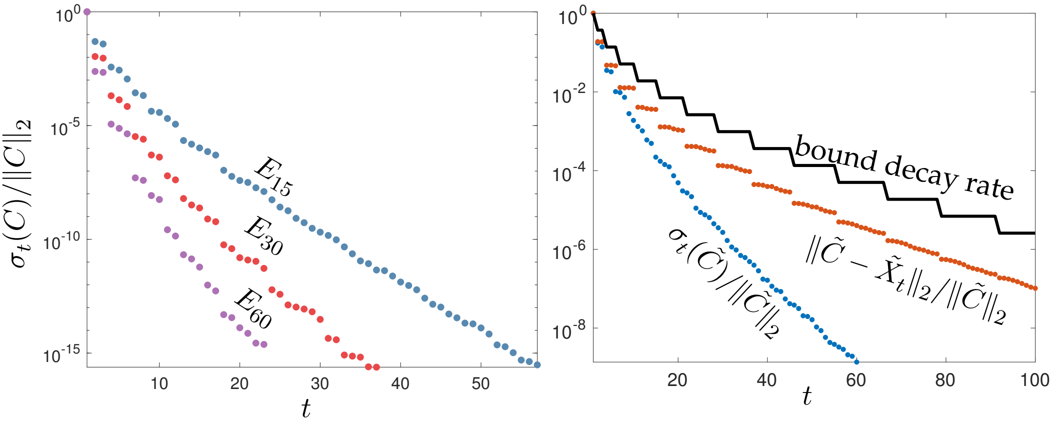


Figure 2.2: Left: The normalized singular values for matrices of the form $\tilde{C}_{ij} = 1/|z_i - w_j|^2$ are plotted for three different selections of sets $\{z_i\}_{i=1}^{100} \subset E_\gamma$ and $\{w_j\}_{j=1}^{100} \subset -E_\gamma$ for $\gamma = 15$ (blue), 30 (red), and 60 (purple), where E_γ is a disk of radius 10 with center $(\gamma, 0)$. Rapid decay of the singular values is observed. Right: The error from constructing a rank t approximant as described in Section 2.2 (red) bounds the normalized singular values for $\tilde{C}$ of size 1000×1000 (blue), and is bounded above by the decay rate (excluding the polynomial factor) from the bound in Theorem 5 (black).

rate $\mu_1^{-(\sqrt{8t+1}-1)/2}$ in (2.8) cannot be improved without additional assumptions on A, B, and F.

Applying Theorem 5 to $\tilde{C}$ shows that for triangular numbers t, $1 \leq t < n$,

$$\sigma_{t+1}(\tilde{C}) \leq \frac{z_0+\eta}{z_0-\eta}\,(\tfrac{3}{2}\sqrt{t}+1)\mu_1^{-(\sqrt{8t+1}-1)/2}\,\|\tilde{C}\|_2. \tag{2.12}$$

Figure 2.2 displays the decay rate from (2.12), as well as the error $\|\tilde{C}-\tilde{X}_t\|_2/\|\tilde{C}\|_2$, where $\tilde{X}_t$ is constructed as in Theorem 5. These results also give a bound on $\mathrm{rank}_\epsilon(\tilde{C})$. For $0 < \epsilon < 1$, we have that

$$\mathrm{rank}_\epsilon(\tilde{C}) \leq \frac{k^*(k^*+1)}{2}, \qquad k^* = \left\lceil \log\left(\frac{(z_0+\eta)(\frac{3}{2}\sqrt{n}+1)}{(z_0-\eta)\epsilon}\right) \Big/ \log\mu_1 \right\rceil, \tag{2.13}$$

where we have used the fact that $\sqrt{t} \leq \sqrt{n}$. For fixed $0 < \epsilon < 1$, the bound in (2.13) only grows polylogarithmically with n, so that for very large n, standard operations, such as matrix-vector multiplication, can be performed to an ϵ-accuracy in quasi-optimal computational complexity by using $\tilde{X}_t$.

We need not require that the decay rate of $\sigma_i(F)$ matches the decay rate of $Z_k(E,-E)$. As an example, suppose that $\sigma_i(F)$ decays with i at a geometric rate twice that of $Z_k(E,-E)$, i.e., $\sigma_{i+1}(F) \leq K\mu_1^{-2i}\|F\|_2$. It is no longer optimal to construct an approximant $\tilde{X}_t$ by selecting terms along antidiagonals of $\mathcal{R}$ (see Figure 2.1 (right)). Instead, the number of fADI iterations applied to each X_i in (2.1) (each row of $\mathcal{R}$) must be modified. Specifically, for each X_i, construct $X_i^{(s_i)}$ with $s_i = 2k - 2(i-1)$ to form $\tilde{X}_t = \sum_{i=1}^{k} X^{(s_i)}$. If, on the other hand, $\sigma_{i+1}(F) \leq K\mu_1^{-i/2}\|F\|_2$, then $\tilde{X}_t$ is constructed by performing $s_i = k+1-i$ iterations of fADI on X_i and X_{i+1} simultaneously. Generalizing from these examples, we have the following corollary:

Corollary 3. *Suppose that the assumptions of Theorem 5 hold, except that $\sigma_{i+1}(F) \leq K\mu_F^{-i}\|F\|_2$, with $\mu_F > 1$. Let $\mu = \min(\mu_F, \mu_1)$, and define the integer ℓ as $\ell = \lfloor \log\left(\max(\mu_F,\mu_1)\right) / \log\mu \rfloor$. Then, for the numbers $1 \leq t = \ell k(k+1)/2 < n$, the singular values of X are bounded as*

$$\sigma_{t+1}(X) \leq K\frac{z_0+\eta}{z_0-\eta}\,(\tfrac{3}{2}\sqrt{t}+1)\mu^{-\ell(\sqrt{8t+1}-1)/2}\,\|X\|_2.$$

Further generalizations of Theorem 5 hold whenever explicit bounds are known for the singular values of F, even if the rate of decay is not geometric. For example, with the same assumptions as Theorem 5 except that the singular values of F decay algebraically, the singular values of X can be shown to decay at the same algebraic rate.

A generalization of Theorem 5 and Corollary 3 can be stated when E and G are any two closed disks in the complex plane that are disjoint from each other. This follows from Theorem 3.1 in [154], where $Z_k(E,G)$ and r_k are given for disks E and G that are each symmetric about the real axis, as well as the observation that $Z_k(E,G)$ is invariant under rotation.

2.3 Bounds via a modification of fADI

We now consider $AX - XB = F$, with $\lambda(A) \subset [-b,-a]$, $\lambda(B) \subset [a,b]$ for $a, b \in \mathbb{R}$ and $0 < a < b$. This scenario arises, for example, in the discretization of Poisson's equation (see Section 2.6). The Zolotarev rational function r_k that attains the infimum $Z_k([-b,-a],[a,b])$ in (1.16) is described in Corollary 2. Moreover, we have from Theorem 3 that

$$Z_k([-b,-a],[a,b]) \leq 4\mu_2^{-k}, \qquad \mu_2 = \exp\left(\frac{\pi^2}{\log(4b/a)}\right). \tag{2.14}$$

The zeros and poles of $r_k(z)$ can be computed using elliptic integrals [109, 182], and they form a set of k ADI shift parameters $\{(\alpha_\ell, \beta_\ell)\}_{\ell=1}^{k}$. In contrast to Section 2.2, one cannot expect that the extremal function $r_j(z)$ that attains the infimum $Z_j([-b,-a],[a,b])$ has any zeros or poles in common with $r_k(z)$ when $j \neq k$. To use explicit bounds associated with $Z_j([-b,-a],[a,b])$ for $1 \leq j \leq k$, we must allow for the use of several sets of shift parameters when constructing our ADI-based approximant $\tilde{X}_t$. This is a natural generalization of the approach used in Theorem 5, and it leads to the following theorem:

Theorem 6. *Let $X \in \mathbb{C}^{m\times n}$, $m \geq n$, satisfy $AX - XB = F$, and suppose that the assumptions in Theorem 5 hold, except that $\lambda(A) \subset [-b,-a]$ and $\lambda(B) \subset [a,b]$, with $a, b \in \mathbb{R}$ and $0 < a < b$. Then, for the triangular numbers $1 \leq t = k(k+1)/2 < n$, we have*

$$\sigma_{t+1}(X) \leq \frac{Kb}{a}(6\sqrt{t}+1)\mu_2^{-(\sqrt{8t+1}-1)/2}\|X\|_2.$$

Proof. Let $X = \sum_{i=1}^{\rho} X_i$, where $\operatorname{rank}(F) = \rho$ and each X_i satisfies (2.1). For each $i \leq k$, construct the approximant $\tilde{X}_t = \sum_{i=1}^{k} X_i^{(s_i)}$, $s_i = k+1-i$, where $X_i^{(s_i)}$ is constructed by applying $s_i = k+1-i$ iterations of fADI to (2.1) using optimal ADI shift parameters (these parameters are different for each i). It follows that

$\sigma_{t+1}(X) \leq \|X - \tilde{X}_t\|_2$, and the proof consists of bounding the error $\|X - \tilde{X}_t\|_2$. This can be done just as in Theorem 5 if one uses the fact that $[a, b]$ can be contained in a disk with radius $\eta = (b-a)/2$ and center $z_0 = (b+a)/2$, so that Lemma 2 is applicable. □

As before, we have the following corollary when the singular values of F and $Z_k([-b,-a],[a,b])$ decay at potentially different geometric rates:

Corollary 4. *Suppose that the assumptions of Theorem 6 hold, except that* $\sigma_{i+1}(F) \leq K\mu_F^{-i}\|F\|_2$*, with* $\mu_F > 1$*. Let* $\mu = \min(\mu_F, \mu_2)$*, and define* ℓ *as* $\ell = \lfloor \log(\max(\mu_F, \mu_2)) / \log \mu \rfloor$*. Then, for* $1 \leq t = \ell k(k+1)/2 < n$*, we have*

$$\sigma_{t+1}(X) \leq \frac{Kb}{a}(6\sqrt{t}+1)\mu^{-\ell(\sqrt{8t+1}-1)/2}\|X\|_2. \tag{2.15}$$

Proof. For a sketch of the proof, see the discussion preceding Corollary 3. □

Related bounds can be stated when $\lambda(A) \subset [a,b]$ and $\lambda(B) \subset [c,d]$, with $a < b < c < d$. In this case, with F as in Corollary 4, we find that

$$\sigma_{t+1}(X) \leq K\frac{\max(|a|,|b|)+\max(|c|,|d|)}{|c-b|}(6\sqrt{t}+1)\mu^{-\ell(\sqrt{8t+1}-1)/2}\|X\|_2,$$

where t is as in Corollary 4, $\mu = \min(\mu_F, \exp(\pi^2/\log 16\gamma))$, and γ is the cross-ratio $|c-a|\ |d-b|/(|c-b|\ |d-a|)$. This result is found by using a Möbius transformation that preserves $Z_k([a,b],[c,d])$ to map $[a,b]\cup[c,d]$ to symmetric intervals $[-\alpha,-1]\cup[1,\alpha]$ (see [19]), and then applying Corollary 4.

2.4 Examples

The ideas and results in Section 2.1 are connected to and inform a variety of other results. We give two examples that show how the splitting and bounding properties can be used. Then in Section 2.6 we apply these ideas to describe a low rank spectral method for solving Poisson's equation when the right-hand side is a smooth function.

2.4.1 The Hadamard product with a Cauchy matrix

Let A, B and F satisfy the assumptions of Theorem 5.[3] Since A and B are normal matrices, they have eigendecompositions $A = Y\Lambda_A Y^*$ and $B = W\Lambda_B W^*$. Therefore, X in (1.1) can be written in closed form as

$$X = Y\left(C \circ (Y^* F W)\right) W^*, \tag{2.16}$$

where C is a Cauchy matrix with entries $C_{jk} = 1/((\Lambda_A)_{jj} - (\Lambda_B)_{kk})$, and '$\circ$' is the Hadamard matrix product. Bounds on the singular values of X can be determined using (2.16), rather than the method in Section 2.1. First, we split $AX - XB = F$ into the $\rho = \mathrm{rank}(F)$ equations in (2.1). By (2.16), each X_i in (2.1) can be expressed as

$$X_i = \sigma_i(F) Y (C \circ (Y^* u_i v_i^* W)) W^*. \tag{2.17}$$

For each $i \leq k$, we use fADI on (2.2) to construct a rank $\leq s_i = k+1-i$ approximant $C^{(s_i)}$ to C. Substituting $C^{(s_i)}$ for C in (2.17) results in an approximant $X_i^{(s_i)}$, and the sum of the matrices $X_i^{(s_i)}$ is an approximant to X. This approach results in bounds of the form

$$\sigma_{t+1}(X) \leq 2K\|C\|_2(z_0 + \eta)(\tfrac{3}{2}\sqrt{t} + 1)\mu_1^{-(\sqrt{8t+1}-1)/2}\|X\|_2,$$

where $1 \leq t = k(k+1)/2 < n$. This relates the singular values of X to properties of the Cauchy matrix C. Generically, we have $\|C\|_2 \leq \sqrt{mn}/(2(z_0 - \eta))$ due

[3] Analogous results hold under the assumptions of Theorem 6, as well as the various generalizations of these theorems.

to (2.2) and Lemma 2, but unfortunately, this does not result in bounds with an improved polynomial term when compared to (2.8). However, a more useful bound on $\|C\|_2$ may be available in specific cases.

This approach leads to an efficient algorithm for approximating X in low rank form when fast matrix-vector products for Y and W are available (see Section 2.6 and also [65, Ch. 4.8]).[4]

2.4.2 Families of structured matrices

Let C be a Cauchy matrix as in (2.2) and define the family $\mathcal{F}_{m,n} = \{C^{\circ p}\}_{p=1}^{\infty}$, where $(C^{\circ p})_{ij} = 1/(z_i - w_j)^p$ and $m \geq n$. For $p \geq 2$, $C^{\circ p}$ satisfies the Sylvester equation $D_z C^{\circ p} - C^{\circ p} D_w = C^{\circ(p-1)}$, and a recursive argument can be used to bound the singular values of each matrix in $\mathcal{F}_{m,n}$. As an example, consider the matrix $C^{\circ 3}$. For $C^{\circ 2}$, Theorem 5 can be applied directly, revealing that the singular values are bounded exactly as in (2.12). To bound the singular values of $C^{\circ 3}$, define each X_i so that it satisfies

$$D_z X_i - X_i D_w = \sum_{j=i(i+1)/2}^{(i+1)(i+2)/2-1} \sigma_j(C^{\circ 2}) u_j v_j^*, \tag{2.18}$$

where u_j and v_j^* are the jth singular vectors of $C^{\circ 2}$. The approximant $\tilde{X}_t$ to $C^{\circ 3}$ is constructed by applying $k+1-i$ fADI iterations to (2.18) for each $i \leq k$, and then summing the resulting matrices. This is a variation on Theorem 5 and results in a bound of the form

$$\sigma_{t+1}(C^{\circ 3}) \leq K_1 \mu_1^{-k} \|C^{\circ 3}\|_2, \qquad 1 \leq t = \frac{1}{24} k(k+1)(k+2)(k+3) < n, \tag{2.19}$$

[4] To compute this approximant in low rank form, one uses the fact that for vectors $u_i = (u_{1i}, \ldots, u_{mi})$ and v_i, $C \circ u_i v_i^* = \text{diag}(u_{1i}, \ldots, u_{mi}) C \text{diag}(v_{1i}, \ldots, v_{ni})$.

where $K_1 = \mathcal{O}(\sqrt{n})$. It follows that as $n \to \infty$, $\text{rank}_\epsilon(C^{\circ 3}) = \mathcal{O}((\log(\sqrt{n}/\epsilon))^4)$. As p is increased, the bounds on the singular values of $C^{\circ p}$ become increasingly weak, but the bound on $\text{rank}_\epsilon(C^{\circ p})$ always grows polylogarithmically with n. This implies that for large enough n, the matrices in $\mathcal{F}_{m,n}$ are well-approximated by low rank matrices. The set of d-dimensional, real-valued Vandermonde matrices satisfies a more complicated recursive relation that leads to similar bounds, and related results hold for various matrix families defined using the structured matrices in [19].

2.5 The FI-ADI method

The low rank approximations employed to bound singular values in Section 2.1 can be automatically computed, resulting in an efficient method for approximately solving $AX - XB = F$ in low rank form whenever A and B satisfy the assumptions in Theorem 5 or Theorem 6 (or their corollaries and generalizations), and linear solves involving A and B can be performed cheaply (see Section 2.6 for an application). We refer to this method as factored-independent ADI (FI-ADI). An outline of the FI-ADI method is given in the pseudocode below,[5] where we assume the above conditions on A and B are met (see Sec Section 2.5.1 for a generalization). Key details for efficient implementation are described below.

[5] An implementation of FI-ADI in MATLAB is available at `https://github.com/ajt60gaibb/freeLYAP`.

```
The FI-ADI method

Input: ∘ $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$ , $F = \sum_{i=1}^{\rho} \sigma_i(F) u_i v_i^*$
       ∘ A tolerance $0 < \epsilon < 1$
       ∘ Disjoint sets $E, G \subset \mathbb{C}$ such that $\lambda(A) \subset E$ and $\lambda(B) \subset G$
       ∘ A batch number $d$ and batching parameters $\{\ell_i\}_{i=1}^{d+1}$, $\ell_{d+1} = \rho + 1$.
Output: Factors $W$, $D$ and $Y$, where $\|X - WDY^*\|_2 \approx \epsilon \|X\|_2$ , $AX - XB = F$.

1. Split $\mathcal{S} := AX - XB = \sum_{i=1}^{\rho} \sigma_i(F) u_i v_i^*$ into $d$ equations:
   $$\mathcal{S}_i := AX_i - X_i B = \sum_{j=\ell_i}^{\ell_{i+1}-1} \sigma_j(F) u_j v_j^*, \qquad 1 \leq i \leq d. \tag{2.20}$$
2. Find $\tau \approx \|X\|_2$
3. Set $W = [\ ]$, $D = [\ ]$, $Y = [\ ]$

for $i = 1, \ldots, d$
  (i) Determine $s_i$ so that for $Z_{s_i}(E, G)$,
      $$Z_{s_i}(E,G) \leq (\epsilon\, \tau\, \mathrm{dist}(E,G)) \,/\, \big(d\, \sigma_{\ell_i}(F)\big), \quad \mathrm{dist}(E,G) = \min_{z\in E, w\in G} |z - w| \tag{2.21}$$
  (ii) Compute the set $\{\alpha_{i,j}, \beta_{i,j}\}_{j=1}^{s_i}$ of optimal ADI shift parameters associated with $Z_{s_i}(E,G)$.
  (iii) Apply $s_i$ steps of fADI to $\mathcal{S}_i$ to find $Z_i$, $D_i$ and $Y_i$
  (iv) $W = \left[\begin{array}{c|c} W & W_i \end{array}\right]$, $Y = \left[\begin{array}{c|c} Y & Y_i \end{array}\right]$, $D = \mathrm{diag}(D, D_i)$
  (v) Compress $W$, $D$ and $Y$
```

Error estimates. As described in the pseudocode, the FI-ADI method constructs $\tilde{X} = WDY^*$. If $\tau \leq \|X\|_2$, then $\|X - \tilde{X}\|_2 \leq \epsilon\|X\|_2$: by Theorem 2, Lemma 2, and the bound on $Z_{s_i}(E,G)$ in (2.21), $\|X_i - X_i^{(s_i)}\|_2 \leq (\epsilon/d)\|X\|_2$. A simple but often overly pessimistic choice for τ is found using $\|F\|_2 \leq (\|A\|_2 + \|B\|_2)\|X\|_2$,. Settling for $\|X - \tilde{X}\|_2 \approx \epsilon\|X\|_2$, it is often more efficient to perform a few steps of FI-ADI and then estimate τ using this approximant. We also find it effective in practice to choose the number of fADI steps for each i as $s_i^* = \max(K_{max}, s_i)$, where s_i is computed as in Step (i) in the pseudocode and K_{max} satisfies $Z_{K_{max}}(E, G) \leq \epsilon$.

Factorizations of F. In Section 2.1, we used the SVD factorization $F = USV^*$ to derive bounds. This is also depicted in the pseudocode. However, the FI-ADI method works with any "approximate SVD" of the form $F = \tilde{U}\tilde{\Sigma}\tilde{V}^*$, where $\tilde{\Sigma}$ is diagonal with $\tilde{\Sigma}_{1,1} \geq \cdots \geq \tilde{\Sigma}_{n,n}$, and $\|\tilde{U}_{(:,i)}\tilde{V}^*_{(:,i)}\|_2 = 1$.

Computation of ADI shift parameters. If E and G are disks in the complex plane, the required single shift parameter (α, β) is given by Theorem 5, a rotation mapping, and the formula in [154]. When E and G are closed real intervals, we refer the reader to the formulas in [109], as well as the MATLAB code in [53, Appendix A]. For most other choices of E and G, heuristic shift selection strategies must be employed (see Section 2.5.1).

Compression. The approximant $\tilde{X}$ is potentally a near-best low rank approximant to X, but in practice, $\tilde{X}$ can often be further compressed. For large problems where memory is restrictive, an interim compression strategy (Step (v) in the pseudocode) is essential, and various schemes can be used (e.g., [73]). We apply the method from [16, Ch. 1.1.4], where the skinny QR factorizations $ZD = Q_zR_z$ and $Y = Q_yR_y$ are used to find the truncated SVD of the small matrix $R_zR_y^*$. The computational cost (in flops) of the compression step grows with the number of columns of W as $\mathcal{O}((m+n)t^2 + t^3)$, where W has t columns. It is beneficial to apply compression after each iteration i to keep the solution factors small.

Batching linear solves. Computational savings can be gained by grouping right-hand sides together when performing linear solves. For example, when the same ADI shift parameter is used in every fADI iteration for all $\mathcal{S}_i$ in (2.20), the uncompressed factors WDY^* are efficiently constructed by applying s_i fADI iterations to the equation $\sum_{j=1}^{i} \mathcal{S}_j$ at each iteration i, with $s_{i-1} \geq s_i$. Even when the shift parameters differ, efficiency is potentially improved by grouping right-hand sides together in Step 1. However, the cost of the compression step is also influenced by the batch sizes, and there is no simple choice of d and $\{\ell_i\}_{i=1}^{d+1}$ that generically optimizes performance.

FI-ADI versus fADI. The FI-ADI method can be seen as a generalization of fADI, where more freedom has been permitted in the order that the rank 1 terms used to approximate X are constructed. Using an FI-ADI-based method over fADI in theoretical settings results in improved bounds on singular values of X (see Section 2.1). However, the practical performance of either method depends greatly on implementation details, as well as the properties of A, B and F. Vectorization and batched solves are efficient, and fADI takes full advantage of this, whereas FI-ADI may not. The main practical benefit of FI-ADI is that re-ordering how rank 1 terms are constructed leads to an effective interim compression strategy. A related idea is discussed in [73] in connection to the low rank cyclic Smith method. There, the reordering of the terms is not designed to exploit the singular value decay of F, but rather to improve memory costs.

2.5.1 Generalized FI-ADI

We briefly review how an FI-ADI-based method can be used when the theorems and corollaries in Section 2.1 are not applicable.

Nonnormality. Let A and B be diagonalizable but non-normal matrices, with eigendecompositions $A = V_A \Lambda_A V_A^{-1}$ and $B = V_B \Lambda_B V_B^{-1}$. The ADI error is bounded as $\|X - X^{(k)}\|_2 \leq \kappa_2(V_A)\kappa_2(V_B)\|r_k(\Lambda_A)\|_2 \; \|r_k(\Lambda_B)\|_2 \|X\|_2$, where $\kappa_2(M) = \|M\|_2 \|M^{-1}\|_2$. If bounds on $\kappa_2(V_A)$ and $\kappa_2(V_B)$ are known or can be numerically estimated, then the influence of these terms on the number of ADI steps can be estimated [161, Sec. 5]. Alternatively, any spectral set [10] can be used to bound $\|r_k(A)\|_2 \; \|r_k(B)^{-1}\|_2$ [19, Cor. 2.2].

Non-optimal shift selection. If the sets E and G do not allow for optimal shift parameter selection, then one of many heuristic shift strategies may be applied [143, Ch. 4.4]. The use of suboptimal shifts affects convergence [151], and additional computational costs are incurred since either the ADI error equation or the residual equation must be monitored to determine convergence. We remark that only a few alternative schemes for solving $AX - XB = F$ when $\mathrm{rank}(F)$ is large have been proposed in the literature [104, 148]. When using FI-ADI, the residual error is given by $\Delta_k(\mathcal{S}) := \|r_k(A)Fr_k(B)^{-1}\|_F$ [17], where $\| \cdot \|_F$ is the Frobenius norm. Using the submultiplicative property for $\Delta_k(\mathcal{S}_i)$, the influence of the singular values of F can be exploited.

2.6 A collection of low rank Poisson solvers

In [53], spectral discretizations are developed so that the ADI method can be used to solve Poisson's equation on a variety of domains in optimal computational complexity (up to polylogarithmic factors). Combining these ideas with FI-ADI leads to highly efficient Poisson solvers that construct low rank approximations to solutions.

An FI-ADI–based Poisson solver on a square

Let u be the solution to Poisson's equation on the square, i.e.,

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f, \qquad x, y \in [-1, 1]^2, \qquad u(\pm 1, \cdot) = u(\cdot, \pm 1) = 0, \tag{2.22}$$

where f is a smooth function on $[-1, 1]^2$. A standard numerical approach for finding u is to discretize (2.22) using second-order finite differences. This leads to a Lyapunov equation $D_2 X + X D_2^T = F$ that can be solved in only $\mathcal{O}(n^2 \log n)$ operations using the closed form solution in (2.16) and the FFT [65, Ch. 4.8]. Applying the FI-ADI method or using (2.16) results in a fast low rank solver, but the accuracy of such an approach is limited.

To achieve spectral accuracy, we instead apply the method in [53], where (2.22) is discretized in a way that leads to the matrix equation $A\hat{X} - \hat{X}B = \tilde{D}\hat{F}\tilde{D}^T$. Here, $\hat{X}$ and $\hat{F}$ contain scaled expansion coefficients for expressing u and f, respectively, in a particular ultraspherical polynomial basis, and $\tilde{D}$ is diagonal. We refer the reader to [53, Sec. 3] for further details. This discretization is specifically designed for ADI-based approaches: A and B satisfy the assumptions in Theorem 6 and they are banded, so that linear solves involving them are cheap. For these reasons, FI-ADI is a highly efficient method for approximating $\hat{X}$ in low rank form. Recurrence relations among ultraspherical polynomials ensure that a rank k approximation to $\hat{X}$ can be transformed to a convenient Chebyshev basis in $\mathcal{O}(kn \log n)$ operations [53, 122]. The inverse transform is also fast, so that if f is a smooth function, a low rank factorization of the matrix $\hat{F}$ can be found efficiently using methods in [158].

The left panel of Figure 2.3 illustrates the computational savings gained from using the FI-ADI method to exploit the numerical rank of $\hat{X}$.[6] In this example, a matrix of bivariate Chebyshev coefficients for f is given in low rank form for several choices of f. We use this to find $\hat{F}$ in low rank form. A low rank approximation to $\hat{X}$ is then computed and transformed to the Chebyshev basis. We compare this approach to the optimal complexity solver in [53] that forms $\hat{F}$ explicitly, and then finds $\hat{X}$ in explicit form.

The right panel displays a solution $\tilde{u}$ to (2.22) computed in Chebfun [45] using this approach, where f is smooth and its 512×512 Chebyshev coefficient matrix is approximated by a rank 206 matrix. The exact solution is given by

$$u = (1-x^2)(1-y^2)\sin(3\pi(1+\cos(\pi x^2 - \pi y^2))(x-2y)(2x+y)\cos(\pi x^2 + \pi y^2)).$$

With the tolerance parameter set at $\epsilon = 10^{-10}$, our approach results in an error of $\|u - \tilde{u}\|_2 / \|u\|_2 \approx 7.01 \times 10^{-11}$.

[6] A faster implementation of both the FI-ADI and ADI-based solvers is achieved by performing the required linear solves with a subroutine written in C (see `https://github.com/danfortunato/fast-poisson-solvers`), which is not used here. The degrees of freedom in this experiment are increased artificially to demonstrate asymptotic complexity.

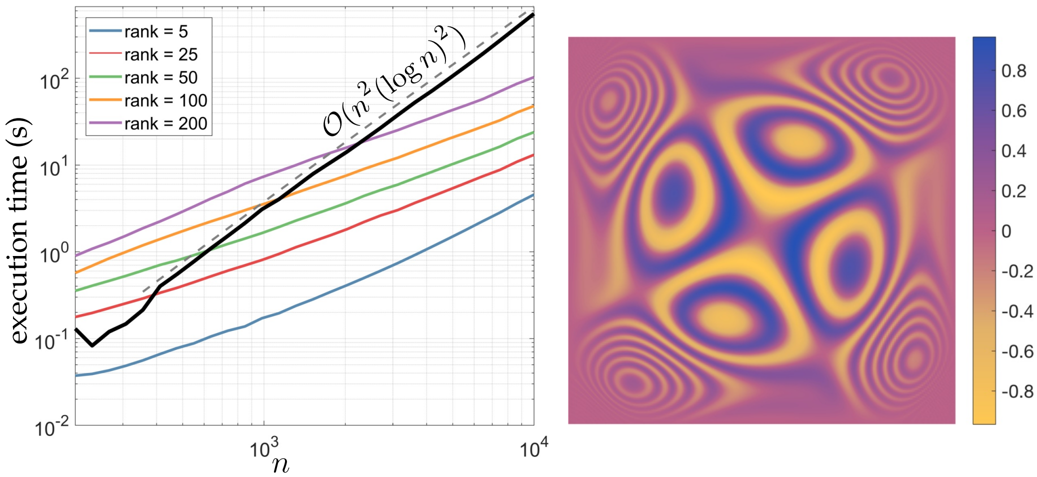


Figure 2.3: Left: The wall clock time in seconds is plotted against the problem size n for computing the Chebyshev coefficients of an approximate solution to (2.22), with a relative tolerance of 1×10^{-10}. A low rank, FI-ADI–based solver with right-hand sides of varying rank (colored lines), all with rapidly decaying singular values, is compared against a spectrally accurate, quasi-optimal complexity solver [53] that is unaffected by the rank of F and returns X in explicit form (black). The FI-ADI–based solver returns a low rank approximation $WDY \approx X$. Timings include compression of the factors W, D, and Y. In both cases, F is provided in low rank form (with approximate singular values). Right: The solution to (2.22), where f is a smooth function with a 512×512 Chebyshev coefficient matrix F of rank 206, computed with Chebfun using the spectral method in [53] and FI-ADI.

FI-ADI–based Poisson solvers for other domains

This approach is not limited to a square domain: Combining FI-ADI with the discretizations described in [53] and [176] leads to efficient low rank Poisson solvers for 2D functions on rectangles, disks, and on the surface of a sphere, and for 3D functions on solid spheres, cylinders, and cubes.

CHAPTER 3

FABER RATIONAL FUNCTIONS

Chapter 2 expands the applicability of ADI to settings where $\mathrm{rank}(F)$ may be large. However, a major limitation to the usefulness of ADI remains because we lack explicit solutions to the Zolotarev problem in (1.16) for general sets E and G in $\mathbb{C}$. In this chapter[1], we use a special class of functions called Faber rationals to make strides toward solving this problem. The Faber rational functions behave similarly to the Zolotarev rationals and approximately solve (1.16). They are analogous to the Faber polynomials [103, 147] introduced by Faber in his thesis [50]. The Faber rationals are described by Ganelius in [59, 58, 60], with a formal definition and a list of relevant properties supplied in [59, Sec. 3]. Our results in [141] use Faber rationals to bound $Z_k(E, G)$ under rather general assumptions on E and G.

After a brief description of the Faber rationals and the bounds they supply on $Z_k(E, G)$ (Sections 3.1 and 3.1.1), we discuss how Faber rationals and related quantities can be computed numerically (Section 3.3). Then, we show how they can be applied in various contexts: they lead to new bounds on the singular values of certain matrix families (Section 3.5.1) , and their zeros and poles (and proxies to these values) are effective as ADI shift parameters (Section 3.5.2). The derivation of the Faber rationals and their use in understanding $Z_k(E, G)$ is related to the development and analysis of rational approximation methods via logarithmic potential theory [60, 145] (see Section 1.5.3). We discuss the performance of the Faber rationals in comparison with other asymptotically optimal

[1] This chapter is based on sections 6 and 7 of a paper [141] by Daniel Rubin, Alex Townsend, and me. The derivation of the bounds on $Z_k(E, G)$ (sections 1-5) in terms of $\mathrm{cap}(E, G)$ is primarily the work of Daniel Rubin. This chapter focuses on my contributions, which include the development of numerical methods and applications.

rational functions in Section 3.5.2.

We remark that while the Faber rationals have valuable and interesting approximation properties, they are not computationally efficient to construct. The substantial challenge of making practical algorithms for solving Zolotarev's rational approximation problems remains, but we hope that the ability to explore such challenges with the Faber rationals will lead to new discoveries.

3.1 Faber rational functions

Let E and G be such that $\mathbb{C}\setminus G$ is open and simply connected and E is a compact, simply-connected subset of $\mathbb{C} \setminus G$. Throughout this chapter, we assume that the boundaries of E and G are rectifiable Jordan curves. To be concrete, the main situation we focus on is when

(A1) E and G are disjoint, simply-connected, compact sets (see Figure 3.1).

The Faber rationals offer a general approach for obtaining explicit bounds on $Z_k(E, G)$. Our construction and the resulting bounds on $Z_k(E, G)$ can be applied more generally. In Section 3.4, we discuss two other types of sets E and G:

(A2) $\mathbb{C} \setminus G$ is a bounded domain containing E (see Figure 3.5),

(A3) G is an unbounded domain and E is a compact domain contained in $\mathbb{C} \setminus G$ (see Figure 3.6).

The construction of Faber rationals requires conformal maps of doubly connected sets. A domain $\Omega \subset \mathbb{C}$ is said to be doubly connected if between any

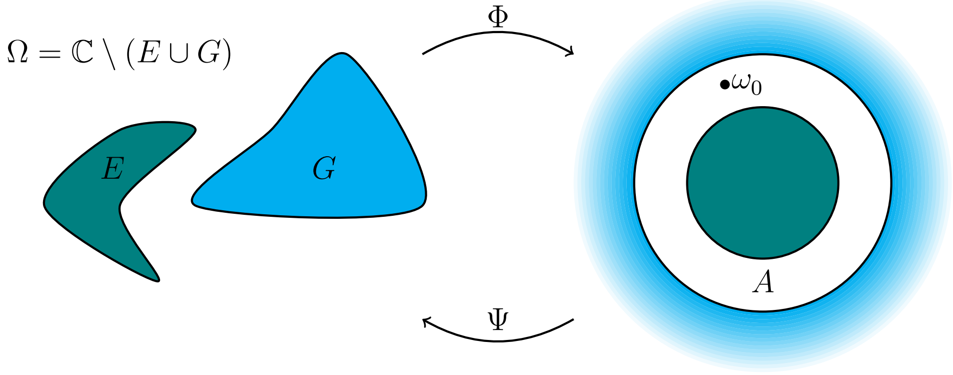


Figure 3.1: We mainly focus on the situation when E and G are disjoint and compact sets in the complex plane. Here, $\Phi : \Omega \to A$ is the conformal map that transplants Ω onto an annulus $A = \{z \in \mathbb{C} : 1 < |z| < h\}$ with $h = \exp(1/\mathrm{cap}(E, G))$. The location $\omega_0 \in \mathbb{C}$ is the pole of the inverse map $\Psi = \Phi^{-1}$.

two points in Ω there are two distinct paths, i.e., two paths that cannot be smoothly deformed into each other. Any doubly connected domain, except for those conformally equivalent to a punctured disk, are conformally equivalent to $A = \{z \in \mathbb{C} : 1 < |z| < h\}$ for some $h > 1$ [38, Ch.1, sec.7]. When $E, G \subset \mathbb{C}$ are as in Theorem 7, $\Omega = \mathbb{C} \setminus (E \cup G)$ is doubly connected and can be conformally mapped to an annulus, i.e.,

$$\Phi : \Omega \to A, \qquad A = \{z \in \mathbb{C} : 1 < |z| < h\}. \tag{3.1}$$

Since conformal maps preserve the condenser capacity of a pair of plate condensers and the condenser capacity of A is $1/\log(h)$ [85], the outer radius in (3.1) is $h = \exp(1/\mathrm{cap}(E, G))$. If E and G are disjoint polygons, then Φ can be constructed via a doubly-connected Schwarz–Christoffel mapping [93], though several numerical issues arise from the practical application of this strategy. The inverse conformal map is denoted by $\Psi = \Phi^{-1} : A \to \Omega$. We discuss effective numerical methods for constructing Φ and Ψ in Section 3.3.2.

A simple observation about the map Φ is that for $z \in E$, $|\Phi(z)| \leq 1$, and for $z \in G$, $|\Phi(z)| \geq h$. This suggests that $\Phi^k(z)$ may serve as a good proxy to the

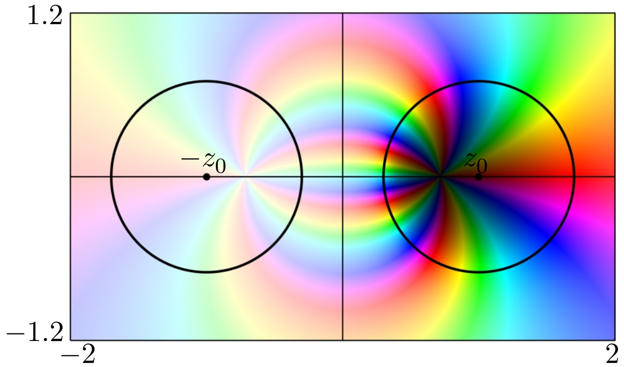


Figure 3.2: A phase plot of the type $(5,5)$ Faber rational function on two disjoint disks, $E = \{z \in \mathbb{C} : |z-1| \leq .7\}$ and $G = -E$. Since there is a Möbius transformation from $\mathbb{C} \setminus (E \cup G)$ to an annulus, $Z_k(E,G)$ is known explicitly.

Zolotarev rational $r_k(z)$ associated with E and G. In fact, if $\Phi(z)$ is a Möbius transformation, and therefore a rational function, we observe that due to the lower bound from Gončar in (1.19), the rational function $\Phi^k(z)$ satisfies

$$h^{-k} \leq \frac{\sup_{z\in E}\left|\Phi^k(z)\right|}{\inf_{z\in G}\left|\Phi^k(z)\right|} \leq h^{-k},$$

From this, we conclude that $Z_k(E,G) = h^{-k}$ and moreover, $\Phi^k(z)$ is a rational function that attains $Z_k(E,G)$ in (1.16). In other words, $\Phi^k(z)$ is a Zolotarev rational function associated with E and G.

As an example, consider the case where E and G are disjoint disks. An alternative way to find $Z_k(E,-E)$ in Theorem 4, where $E = \{z \in \mathbb{C} : |z - z_0| \leq \eta\}$, is with the Möbius transformation

$$\Phi(z) = \frac{z_0+\eta+\phi}{z_0+\eta-\phi}\frac{z-\phi}{z+\phi}, \qquad \phi = \sqrt{z_0^2-\eta^2},$$

which maps Ω to the annulus A with the outer radius $h = (z_0+\phi)/(z_0-\phi)$. Note that h agrees with μ_1 in (1.24). A phase plot of Φ is shown in Figure 3.2.

In the case where Φ^k is a type (k,k) rational function, we say it is the *Faber rational function associated with E and G*. When Φ in (3.1) is not a Möbius transformation, we find that $\Phi^k \notin \mathcal{R}_{k,k}$. Therefore, Φ^k is not immediately useful for bounding $Z_k(E,G)$. However, we still expect Φ^k to be $\mathcal{O}(h^k)$ near the boundary of G and $\mathcal{O}(1)$ near the boundary of E. Thus, the idea is to construct a rational function from Φ^k by "filtering" Φ^k using the so-called Faber operator associated with $\Psi = \Phi^{-1}$ [4], which was first introduced as an operator for constructing polynomial approximations known as Faber polynomials [50, 103] The rational function obtained from Φ^k after the "filtering" process is called the Faber rational associated with E and G. We describe this process in more detail in Section 3.2, but first describe the bounds attained in [141] on $Z_k(E,G)$ that come from this construction.

3.1.1 Bounding Zolotarev numbers with Faber rationals

The Faber rationals approximately solve Zolotarev's third problem. Their explicit construction leads to bounds on $Z_k(E,G)$ that involve a geometric quantity called the total rotation of the domains E and G [57, 137], abbreviated as $\mathrm{Rot}(E)$, and $\mathrm{Rot}(G)$, respectively. For any simply-connected domain, we note that $\mathrm{Rot}(E) \geq 1$. When E is a polygon, $2\pi\mathrm{Rot}(E)$ equals the sum of the absolute values of E's exterior angles. Moreover, when E is a convex domain, $\mathrm{Rot}(E) = 1$ [4, p. 6]. A full derivation of the bounds is found in [141], but we restate the main result here.

Theorem 7. *Let $E, G \subset \mathbb{C}$ be disjoint, simply-connected, compact sets with rectifiable Jordan boundaries. Then, for $h = \exp(1/\mathrm{cap}(E,G))$, we have*

$$Z_k(E,G) \leq (2\mathrm{Rot}(E)+2)(2\mathrm{Rot}(G)+2)h^{-k} + \mathcal{O}(h^{-2k}), \qquad \text{as } k \to \infty,$$

where $\mathrm{Rot}(E)$ and $\mathrm{Rot}(G)$ are the total rotation of the boundaries of the domains E and G, respectively. If, in addition, E and G are convex sets, then we simply have

$Z_k(E,G) \leq 16h^{-k} + \mathcal{O}(h^{-2k})$ *as* $n \to \infty$.

Theorem 7 shows that Gončar's lower bound on $Z_k(E,G)$ in (1.19) is sharp up to a constant. In particular, for disjoint, simply-connected, compact sets $E, G \subset \mathbb{C}$ with rectifiable Jordan boundaries, we have that

$$1 \leq \limsup_{k\to\infty} \frac{Z_k(E,G)}{h^{-k}} \leq (2\mathrm{Rot}(E)+2)(2\mathrm{Rot}(G)+2).$$

An explicit expression of the upper bound in Theorem 7 is inelegant [141, eq. 1.4], but simple to compute. Section 3.1.1 illustrates how the bounds behave for convex sets with varying values of h. The best previous explicit upper bound for sets satisfying (A1)-(A3) is $Z_k(E,G) \leq 4000k^2h^{-k}$ [60]. Our work is closely related to an argument in [58], where a bound of the form $Z_k(E,G) \leq Ch^{-k}$ is described with C independent from k, though the value of the constant is never worked out.

3.2 Constructing Faber rationals analytically

We now give an analytic description for the Faber rationals, which is based on the procedure in [60]. A detailed derivation is found in [141]. There are two main steps: (1) Constructing a function, $R_k(z)$, defined on $\mathbb{C} \setminus G$ with precisely k zeros, and (2) using R_k to construct the Faber rational function, $\tilde{r}_k(z)$, of type (k,k). Both steps are accomplished by taking Cauchy integrals along the boundaries of E and G.

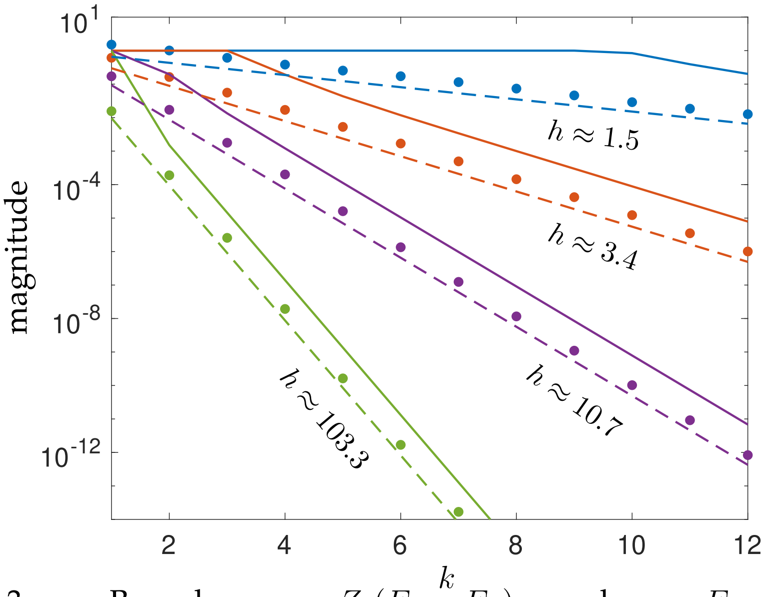


Figure 3.3: Bounds on $Z_k(E_\alpha, -E_\alpha)$, where E is the set $E_\alpha = \{z \in \mathbb{C} : Re(z) \in [-.4 - \alpha, .4 - \alpha], Im(z) \in [-.6, .6]\}$, with $\alpha = .45$ (blue), $.6$ (orange), 1 (purple), 3 (green). As α grows, $h = \exp(1/\text{cap}(E_\alpha, -E_\alpha))$ grows, and $Z_k(E_\alpha, -E_\alpha)$ decays more rapidly. The solid lines are the bounds from Theorem 7, combined with the trivial bound $Z_k(E, F) \leq 1$. The dotted lines are the lower bounds of $Z_k(E_\alpha, -E_\alpha) \geq h^{-k}$ in (1.19). The dots are computed by first constructing the Faber rational function $\tilde{r}_k(z)$ associated with $(E_\alpha, -E_\alpha)$, and then computing the $\max_{z \in E_\alpha} |\tilde{r}_k(z)| / \min_{z \in -E_\alpha} |\tilde{r}_k(z)|$.

3.2.1 Step 1: Constructing a function $R_k(z)$ with k zeros near E

Let $\gamma : [0, 1] \to \Omega$ be a positively oriented parameterization of the boundary E. We can define the following "filtered" function inside E:

$$R_k(z) := \frac{1}{2\pi i} \int_\gamma \frac{\Phi^k(\zeta) d\zeta}{\zeta - z}, \qquad z \in E. \tag{3.2}$$

The holomorphic function $R_k(z)$ is initially defined inside E. If it is possible to extend Φ homomorphically to the whole interior of E, then $R_k(z) = \Phi^k(z)$ there, but this occurs only in exceptional cases. Intuitively one might expect $R_k(z)$ to behave like a degree k polynomial whose zeros are all in E, since it has boundary values on ∂E close to those of Φ^k, which are the same as the function z^k on the boundary of the unit circle. In particular, since $|\Phi^k(z)| \leq 1$ for $z \in E$, the function $|R_k(z)|$ should be relatively small on E. This is shown to be true in [141, Lemma 3.1]. For example, in the useful case where E is assumed to be convex, it is shown that

$$\sup_{z\in E}|R_k(z)| \leq \frac{4(1+h^{-k})}{1-h^{-2k}}, \quad k \geq 0. \tag{3.3}$$

By analytic continuation, the definition of R_k can now be extended to $\Omega = \mathbb{C}\setminus(E\cup G)$. Fix $z\in\Omega$. First, we continuously deform the contour γ to a contour γ' that is contained in Ω and encloses z. By continuously deforming the contour γ' back to γ plus a path traversed in both directions extending to an arbitrarily small circle around z, we find that

$$R_k(z) = \frac{1}{2\pi i}\int_{\gamma'}\frac{\Phi^k(\zeta)d\zeta}{\zeta - z} = \Phi^k(z) + \frac{1}{2\pi i}\int_{\gamma}\frac{\Phi^k(\zeta)d\zeta}{\zeta - z}, \qquad z\in\Omega.$$

Here, the term $\Phi^k(z)$ appears because it is the average value of the Cauchy integral over an arbitrarily small circle around z. Since $|\Phi^k(z)| < h^k$ for $z\in\Omega$, we find that R_k is a bounded function in Ω.

Since the Cauchy transform of a continuous function on a closed contour can be used to define two distinct holomorphic functions — one in the interior of the region bounded by the contour and the other on the exterior — we can write

$$\begin{aligned}
\mathcal{C}^+_{\partial E}(\Phi^k)(z) &= \frac{1}{2\pi i}\int_{\gamma}\frac{\Phi^k(\zeta)d\zeta}{\zeta - z}, \qquad z \text{ inside of } \gamma,\\
\mathcal{C}^-_{\partial E}(\Phi^k)(z) &= \frac{1}{2\pi i}\int_{\gamma}\frac{\Phi^k(\zeta)d\zeta}{\zeta - z}, \qquad z \text{ outside of } \gamma,
\end{aligned}$$

where the subscript indicates that the integral is taken over the boundary of E. Therefore, the function $R_k(z)$ can be expressed as

$$R_n(z) = \begin{cases} \mathcal{C}^+_{\partial E}(\Phi^k)(z), & z\in E,\\[2ex] \Phi^k(z) + \mathcal{C}^-_{\partial E}(\Phi^k)(z), & z\in\mathbb{C}\setminus(E\cup G). \end{cases} \tag{3.4}$$

To further emphasize the interpretation that $R_k(z)$ is a filtered version of $\Phi^k(z)$, we highlight that $R_k(z)$ is relatively close to $\Phi^k(z)$ for $z \in \Omega$.

Lemma 3. *Let $E, G \in \mathbb{C}$ be sets satisfying the assumptions in Theorem 7. Then, $R_k(z)$ in* (3.4) *satisfies*

$$\sup_{z\in\Omega}\left|R_k(z) - \Phi^k(z)\right| \leq 1 + \sup_{z\in E}\left|R_k(z)\right|.$$

Proof. See [141, Lemma 3.2] □

Lemma 3 allows us to show that all the zeros of R_k lie in E or within a small neighborhood of E. Rouché's Theorem says that the winding numbers of Φ^k and R_k around a closed curve Γ will be equal provided that $|\Phi^k(z) - R_k(z)| < |\Phi^k(z)|$ for z on Γ [2]. By Lemma 3, the theorem applies on any closed curve Γ in Ω winding once around E such that $1 + \sup_{z\in E}|R_k(z)| < |\Phi^k(z)|$ for z on Γ. Such a curve Γ can always be found when the bound $1 + \sup_{z\in E}|R_k(z)| < h^k$, say, by taking the image of Γ to be an appropriate level set of $|\Phi^k|$. The map Φ^k has winding number of precisely k around Γ by definition (though it is not defined in E) and hence so does R_k. Since R_k is analytic inside Γ, it has k zeros (counting multiplicities) inside Γ. Moreover, the same reasoning shows that R_k has no additional zeros outside of Γ in Ω.

We denote the distinct zeros of R_k as $z_1, \ldots, z_K$ with corresponding multiplicities $m_1, \ldots, m_K$ such that $m_1 + \cdots + m_K = k$. In [60], a more precise statement is proved about the location of the zeros of $R_k(z)$. For example, it is shown that for sufficiently large k, the zeros of $R_k(z)$ lie inside E or in a neighborhood of E.

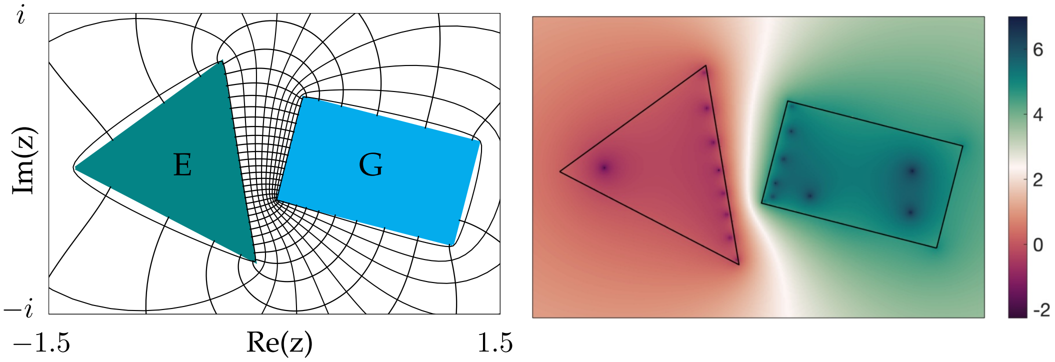


Figure 3.4: Left: A plot of the conformal map $\Psi = \Phi^{-1}$, where Φ maps $\mathcal{C} \setminus E \cup G$ to the annulus $A = \{z \in \mathbb{C} : 1 < |z| < h\}$. Right: The magnitude of the type $(9,9)$ Faber rational function is plotted on a logarithmic scale. As k increases, the Faber rational function grows increasingly larger on G and smaller on E, making it useful as an approximation to the Zolotarev rational that attains $Z_k(E,G)$ in (1.16).

3.2.2 Step 2: Constructing a Faber rational function

While R_k has precisely k zeros, it is typically not a rational function. We must "filter" R_k again to obtain a rational function. Let $\eta : [0,1] \to \Omega$ be a curve that winds around G once in the counterclockwise direction. We choose η close to the boundary of G: Letting $0 < \delta < 1$, η satisfies $|\Phi(\eta(t))| \geq h - \delta$ for $t \in [0,1]$.[2] By Lemma 3, R_k is close to Φ^k on η, and $|\Phi^k|$ is close to h^k on η. Making sure that δ is sufficiently small enough to avoid encircling any zeros of R_k, we can assume that $1/R_k$ is analytic on the curve η (see (3.4)). We can construct analytic functions inside and outside of η (the inside of η contains G) as

$$\begin{aligned} \mathcal{C}_\eta^+(1/R_k)(z) &= \frac{1}{2\pi i}\int_\eta \frac{d\zeta}{R_k(\zeta)(\zeta - z)}, \qquad z \text{ inside of } \eta, \\ \mathcal{C}_\eta^-(1/R_k)(z) &= \frac{1}{2\pi i}\int_\eta \frac{d\zeta}{R_k(\zeta)(\zeta - z)}, \qquad z \text{ outside of } \eta. \end{aligned} \tag{3.5}$$

[2] The requirement that $\delta > 0$ is a technical necessity as $R_k(z)$ is defined for $z \in \mathbb{C}\setminus G$. Later, we take $\delta \to 0$ so conceptually one may prefer to think of η as a parameterization of the boundary of G.

It is possible to give an exact expression for $\mathcal{C}_\eta^-(1/R_k)(z)$ in terms of $R_k(z)$ for z outside of η.

Lemma 4. *Let $E, G \in \mathbb{C}$ be sets satisfying the assumptions in Theorem 7 and $R_k(z)$ be defined as in* (3.4). *If $z_1, \ldots, z_K$ are the distinct zeros of $R_k(z)$ with multiplicities $m_1 + \cdots + m_K = k$, then for z outside of η we have*

$$\mathcal{C}_\eta^-(1/R_k)(z) = -\frac{1}{R_k(z)} + \sum_{k=1}^{K}\sum_{j=1}^{m_k} \frac{a_{-j}^k}{(z-z_k)^j} + \frac{1}{R_k(\infty)}, \tag{3.6}$$

where a_{-j}^k is the z^{-j} coefficient of the principal part of the Laurent series for $R_k(z)$ about z_k.

Proof. See [Lem. 3.4][141] □

Lemma 4 can be combined with the Sokhotski–Plemelj Theorem [84] to find an expression for $\mathcal{C}_\eta^+(1/R_k)(z)$ in terms of $R_k(z)$. We have

$$\mathcal{C}_\eta^+(1/R_k)(z) - \mathcal{C}_\eta^-(1/R_k)(z) = \frac{1}{R_k(z)}, \qquad \text{for } z \text{ on } \eta.$$

and, by analytic continuation, we have

$$\mathcal{C}_\eta^+(1/R_k)(z) = \sum_{k=1}^{K}\sum_{j=1}^{m_k} \frac{a_{-j}^k}{(z-z_k)^j} + \frac{1}{R_k(\infty)}, \qquad z \text{ inside of } \eta. \tag{3.7}$$

We conclude that $\mathcal{C}_\eta^+(1/R_k)$ is a rational function of type (k,k). Finally, we define the Faber rational $\tilde{r}_k$ associated with E and G as

$$\frac{1}{\tilde{r}_k(z)} = \sum_{j=1}^{K}\sum_{\ell=1}^{m_j} \frac{a_{-\ell}^j}{(z-z_j)^\ell} + \frac{1}{R_k(\infty)}. \tag{3.8}$$

The expression for $\tilde{r}_k(z)$ in (3.8) is ideal for identifying $\tilde{r}_k(z)$ as a rational function of type (k,k). The relationship $1/\tilde{r}_k(z) = \mathcal{C}_\eta^+(1/R_k)(z)$ in (3.5) is more convenient for practical computations as it does not involve computing z_k for $1 \le k \le K$ or the Laurent coefficients $\{a_{-\ell}^j\}$.

3.3 Constructing Faber rationals numerically

In this section, we describe algorithms evaluating Faber rational functions, as well computing $h = \exp(1/\mathrm{cap}(E, G))$. We also discuss a method for finding the poles and zeros of $\tilde{r}_k$.

3.3.1 Evaluating $\tilde{r}_k$

To evaluate $\tilde{r}_k(z)$, we use the integral formulations for $1/\tilde{r}_k(z)$ developed in Section 3.2.2. It is acceptable for numerical purposes to choose the contour η in Lemma 4 as ∂F. Taking this liberty, we have from the lemma that

$$1/\tilde{r}_k(z) = \begin{cases} \dfrac{1}{2\pi i}\displaystyle\int_{\partial G}\frac{d\zeta}{R_k(\zeta)(\zeta - z)}, & z \in G, \\ -\dfrac{1}{R_k(z)} + \dfrac{1}{2\pi i}\displaystyle\int_{\partial G}\frac{d\zeta}{R_k(\zeta)(\zeta - z)}, & z \in \mathbb{C} \setminus G, \end{cases} \tag{3.9}$$

where the first integral is understood in the principal value sense for $z \in \partial G$,[3] and R_k is defined in (3.4).

The integrals in (3.4) and (3.9) can be computed using a quadrature rule, but these computations can become numerically unstable when z is close to the contour of the integral being evaluated. To alleviate this issue, we apply a variant of the barycentric interpolation formula [25]. For $z \in G$, this takes the following form:

$$\frac{1}{\tilde{r}_K(z)} = \frac{\displaystyle\int_{\partial G}\frac{d\zeta}{R_k(\zeta)(\zeta - z)}}{\displaystyle\int_{\partial G}\frac{d\zeta}{\zeta - z}} \approx \frac{\displaystyle\sum_{j=1}^{N_Q}\frac{w_j}{R_k(x_j)(x_j - z)}}{\displaystyle\sum_{j=1}^{N_Q}\frac{w_j}{x_j - z}},$$

[3] We avoid sampling directly on ∂G in our applications, and so omit discussion on the numerical computation of principle value integrals.

where $\{(w_j, x_j)\}_{j=1}^{N_Q}$ are an appropriate set of quadrature weights and nodes. A similar procedure is used when evaluating $R_k(z)$ for $z \in E$ near ∂E. Once $f_z = 1/\tilde{r}_k(z)$ is computed, we set $\tilde{r}_k(z) = 1/f_z$. After one can evaluate $\tilde{r}_k$ on $E \cup G$, it can be represented as a rational function via the AAA algorithm [119], which makes further evaluations more efficient.

3.3.2 Computing the conformal map

Evaluating $R_k(z)$ requires the conformal map $\Phi : \Omega \to A$. We construct Φ using the method in [165]. In this approach, Φ is computed via the Green's function associated with the Laplacian operator on Ω. The problem reduces to solving $\Delta u = 0$ with boundary conditions as in [146, p. 253], [165, Sec. 4]. To solve for u, boundary data is used to find the least squares fit to the coefficients of an approximate rational expansion of u. This is especially effective for resolving singularities in corners of the domain because the poles of the expansion are chosen to be exponentially clustered near the singular points [67]. The value of h is treated as an additional unknown in the least squares system of equations, and it is recovered along with u.

This method is versatile and can be used when E and G are polygons, as well as when their boundaries are either analytic curves or piecewise continuous analytic curves. It can be adapted for use in the (A2) case from Section 3.4 where G is unbounded. An example displaying both the conformal map and the Faber rational is shown in Figure 3.4.

3.3.3 The poles and zeros of $\tilde{r}_k$

To compute the poles and zeros of $\tilde{r}_k$, we first construct a representation of $\tilde{r}_k$ in barycentric form via the AAA algorithm [119]. This construction is computationally expensive because $\tilde{r}_k$ must be sufficiently sampled on the sets E and G. The poles and zeros are then computed by solving an $(n+2) \times (n+2)$ generalized eigenvalue problem. To improve the accuracy of the computation, we apply AAA twice: first to $\tilde{r}_k$ on E to compute the zeros, and then again to $\tilde{r}_k$ on G to compute the poles. For an application involving poles and zeros, see Section 3.5.2.

3.4 Faber rationals on other sets in $\mathbb{C}$

In addition to the (A1) case where E and G are both compact sets (see Figure 3.1), there are two other types of sets E and G where our results in [141] using Faber rationals leads to explicit bounds on $Z_k(E, G)$:

(A2) $\mathbb{C} \setminus G$ is a bounded domain containing E (see Figure 3.5), and

(A3) G is an unbounded domain and E is a bounded subset of $\mathbb{C} \setminus G$.

For the (A3) case, the bound on $Z_k(E, G)$ is the same as in the (A1) case (see Theorem 7 and [141, eq. 1.4]). In the (A2) case, Ψ no longer has a pole in the annulus A, and this leads to a slightly different bounds on $Z_k(E, G)$, which are given explicitly in [141, Thm. 5.1].

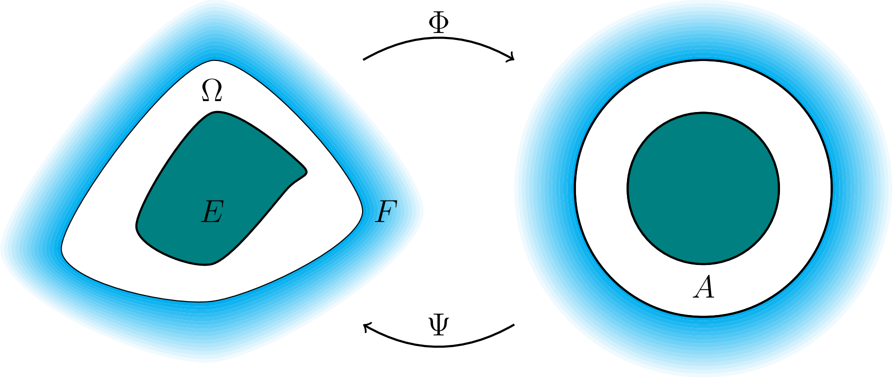


Figure 3.5: Illustration of the typical setup when $\mathbb{C} \setminus G$ is a bounded domain containing a compact set E.

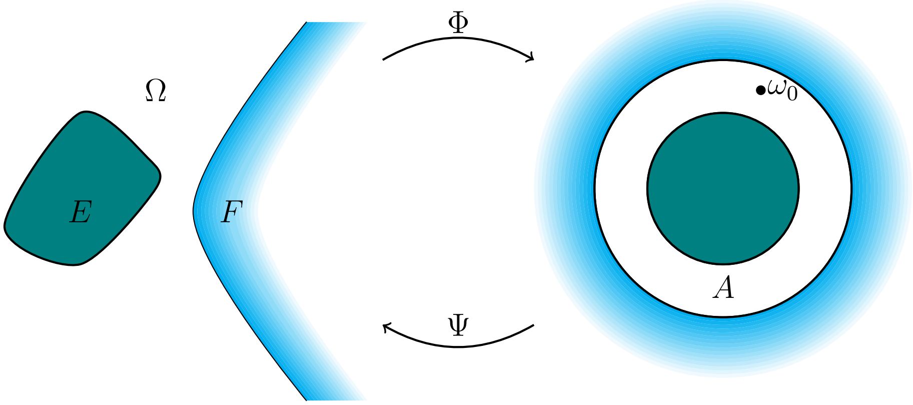


Figure 3.6: Illustration of the typical setup when G is an unbounded domain and E is a compact domain contained in $\mathbb{C} \setminus G$. The location $\omega_0 \in \mathbb{C}$ is the pole of the inverse map $\Psi = \Phi^{-1}$.

3.5 Applications in computational mathematics

We give two examples from numerical linear algebra where the Faber rationals can be applied. In the first, we bound the singular values of complex-valued Cauchy and Vandermonde matrices. In the second example, we treat $\tilde{r}_k$ as a proxy to the true Zolotarev rational function association $Z_k(E, G)$. We show that the poles and zeros of $\tilde{r}_k$ are near-optimal parameters in the alternating direction implicit (ADI) method, and we discuss related methods based on ideas from logarithmic potential theory.

3.5.1 Bounding the singular values of matrices.

We suppose that $X \in \mathbb{C}^{m\times n}$ satisfies the Sylvester matrix equation $AX - XB = F$, with $\rho = \mathrm{rank}(F)$. When A and B are normal matrices with spectra $\lambda(A) \subset E$, $\lambda(B) \subset G$, then we recall from Theorem 2 that the normalized singular values of X are bounded above in terms of Zolotarev numbers. Specifically,

$$\sigma_{k\rho+1}(X) \leq Z_k(E,G)\|X\|_2, \quad 0 \leq k\rho + 1 \leq \min(m,n). \tag{3.10}$$

Pairing this observation with the bounds on $Z_k(E,G)$ from Theorem 7 gives bounds on $\sigma_{k\rho+1}(X)$ whenever E and G are as in the theorem (or as in cases (A2-A3) from Section 3.4). One can of course also extend Theorem 5 and its generalizations from Chapter 2 to bound $\sigma_{k\rho+1}(X)$ in terms of the singular values of F when ρ is large. We illustrate the point with two examples.

Complex-valued Cauchy matrices

Let C be a Cauchy matrix in $\mathbb{C}^{m\times n}$, with entries given by

$$C_{j\ell} = 1/(x_j - y_\ell), \quad \underline{x} = \{x_j\}_{j=1}^{m} \subset E, \quad \underline{y} = \{y_\ell\}_{\ell=1}^{n} \subset G,$$

where E and G are as in Theorem 7 and the sets $\underline{x}, \underline{y}$ are each collections of distinct points. Since $\mathrm{rank}(D_{\underline{x}}C - CD_{\underline{y}}) \leq 1$, where $D_{\underline{x}} = \mathrm{diag}(x_1, \ldots, x_m)$, it immediately follows from (3.10) and Theorem 7 that for $0 \leq k \leq \min(m,n) - 1$,

$$\sigma_{k+1}(C) \leq K_{E,G}h^{-k}\|C\|_2, \qquad h = \exp(1/\mathrm{cap}(E,G)).$$

with $K_{E,G} = (2\mathrm{Rot}(E) + 2)(2\mathrm{Rot}(G) + 2) + \mathcal{O}(h^{-k})$. An explicit expression of $K_{E,G}$ is found in [141, eq. 1.4], and if we take E and G to be convex (say, two

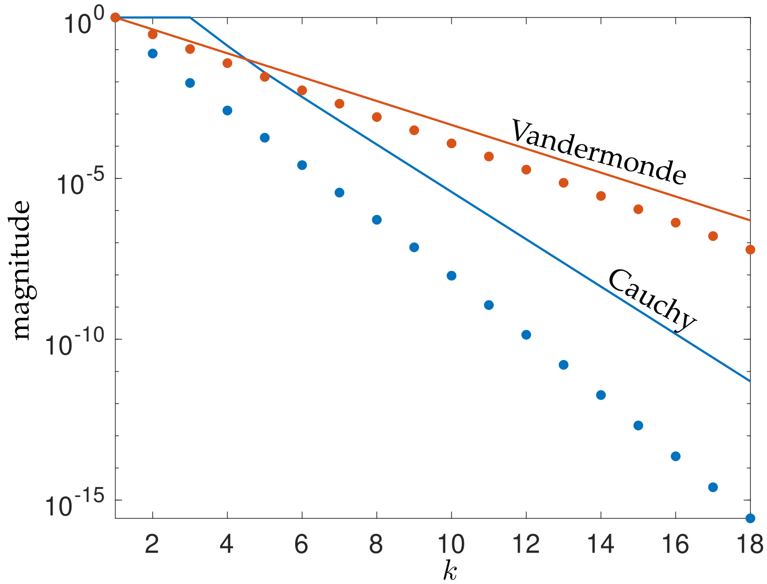


Figure 3.7: The first 18 normalized singular values of a Cauchy matrix (blue dots) and Vandermonde matrix (red dots) are plotted against the singular value index k on a logarithmic scale. The Cauchy matrix is given by $C_{j\ell} = 1/(x_j - y_\ell)$, $1 \leq j, \ell \leq 100$, where for all (j, ℓ), $x_j \in E_C := \{z \in \mathbb{C} : .3 \leq \mathrm{Re}(z) \leq 1.3, |\mathrm{Im}(z)| \leq .5\}$ and $y_\ell \in -E_C$. The nodes of the Vandermonde matrix $V \in \mathbb{C}^{100\times 80}$ all lie in $E_V = \{z \in \mathbb{C} : |z - (2+i)/10| < .4\}$. The solid lines show bounds on $\sigma_k(C)/\sigma_1(C)$ (blue) and $\sigma_k(V)/\sigma_1(V)$ (red) obtained via Theorem 7 and Lemma 5, respectively.

convex polygons), then $K_{E,G} = 16 + \mathcal{O}(h^{-k})$. To implement the bound, we compute h using the method in Section 3.3.2. A comparison of the bounds with computed singular values is shown in Figure 3.7. Prior to this, explicit bounds on the singular values of C were only known in settings where E and G were disjoint disks, intervals of the real line, or arcs on a circle (see Table 1.1).

Vandermonde matrices with nodes inside the unit circle

Let V_x be an $m \times n$ Vandermonde matrix with entries $(V_x)_{j\ell} = x_j^{(\ell-1)}$, where the nodes $x = \{x_j\}_{j=1}^m$ are distinct points in $\mathbb{C}$. The singular values of V_x are known to decay rapidly when each x_j is real [19], and there are multiple results on the (extremal) singular values of V_x when all $|x_j| = 1$ [14, 117]. Less is known about singular value decay when $|x_j| < 1$, despite the fact that this assumption is encountered in several applications [15, 27, 133]. We give the following lemma:

Lemma 5. *Let $V_x \in \mathbb{C}^{m\times n}$ have a set of distinct nodes contained in the disk $E := \{|z - z_0| < \eta_0\}$, $z_0 \neq 0$, where E is in the open unit disk. Then, the following bound holds for $0 \leq k - 1 \leq \min(m, n)$:*

$$\sigma_{k+1}(V_x) \leq h^{-k} \|V_x\|_2,$$

where

$$h = \left| \frac{z_0 - |z_0|\beta(z_0 + \eta_0)}{|z_0|(z_0 + \eta_0) - \beta z_0} \right|, \quad \beta = \frac{1}{2|z_0|}\left(1 + c - \sqrt{(1+c)^2 - 4|z_0|^2}\right), \quad c = |z_0|^2 - \eta_0^2.$$

Proof. We observe that $\operatorname{rank}(D_\alpha V - VQ) = 1$, where $Q = \left[\begin{smallmatrix} 0 & 1 \\ I_{n-1} & 0 \end{smallmatrix}\right]$ is the circulant shift matrix. The eigenvalues $\lambda(Q)$ are the nth roots of unity. We choose G as the set exterior to the open unit disk and note that $\lambda(Q) \subset G$. We map $\Omega := C \setminus E \cup G$ to the annulus $A := \{z \in \mathbb{C} : 1 < |z| < h\}$ with the following Möbius transformation:

$$\mathcal{T}(z) := \frac{h(|z_0|z - z_0\beta)}{z_0 - |z_0|\beta z},$$

where h, β, and c are as in the theorem. Since $\mathcal{T}$ maps $\Omega \to A$ conformally and $\mathcal{T}$ is rational, $r_k = \mathcal{T}^k$ is the rational function that attains $Z_k(E, G)$, and $Z_k(E, G) = h^{-k}$. Applying (3.10) completes the proof. □

The bounds from Lemma 5 are shown in Figure 3.7 along with computed singular values. We remark that if E is centered on the origin, then $h = 1/\eta_0$. For more general choices of E, an argument similar to the proof of Lemma 5 can be applied using the bounds on $Z_k(E, G)$ from Theorem 7 and its variations.

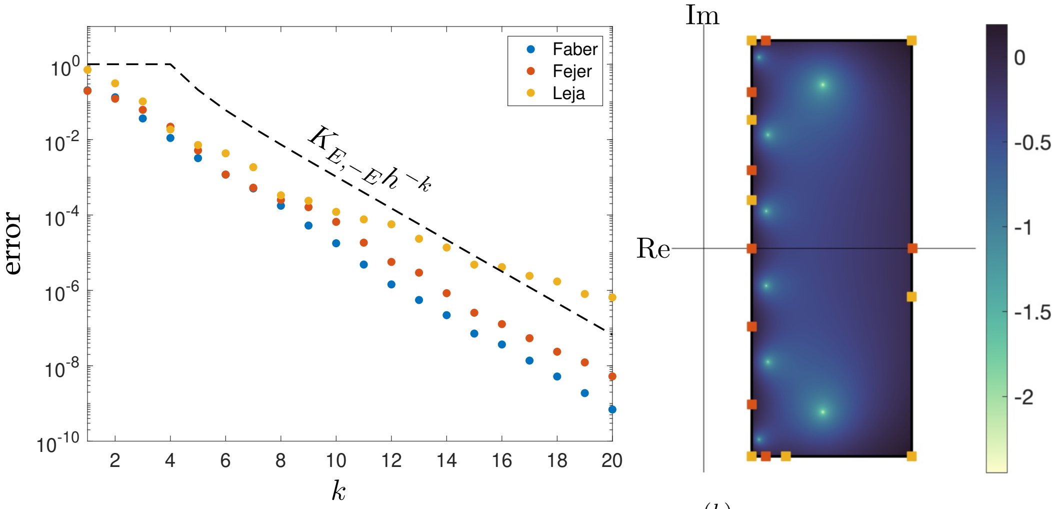


Figure 3.8: Left: The computed ADI error $\|X - X^{(k)}\|_2/\|X\|_2$ is plotted against the indices k on a logarithmic scale, where ADI is applied using Faber shifts (blue), generalized Fejér points (red), and generalized Leja points (yellow). The bound on the error for ADI with Faber shifts is shown as a dotted line. Here, X satisfies (1.1), with $m = n = 100$, $\lambda(A) \in E$, $\lambda(B) \in -E$, where $E = \{z \in \mathbb{C} : .3 \leq \text{Re}(z) \leq 1.3,\ -1.3 \leq \text{Im}(z) \leq 1.3\}$. Right: The magnitude of the Faber rational r_8 is plotted on a logarithmic scale over E. Generalized Fejér points (red squares) and generalized Leja points (yellow squares) associated with $(E, -E)$ are plotted. These are selected as the α_j parameters for ADI, and due to the symmetry of the domain, one sets $\beta_j = -\overline{\alpha}_j$. The Faber shifts are formed by using the zeros of r_8 as α_j parameters, and the poles of r_8 (not depicted) as β_j parameters.

3.5.2 ADI shift parameters from Faber rationals

We recall from Section 1.4 that the ADI error equation depends on the choice of $2k$ shift parameters, $\{(\alpha_j, \beta_j)\}_{j=1}^{2k}$. As shown in (1.13), the bound on the error is controlled as follows:

$$\|X - X^{(k)}\|_2 \leq \frac{\sup_{z\in E}|s_k(z)|}{\inf_{z\in G}|s_k(z)|}, \quad s_k(z) = \prod_{j=1}^{k} \frac{z-\alpha_j}{z-\beta_j}. \tag{3.11}$$

The minimal value of the bound, $Z_k(E, G)$, is attained when r_k is selected as the Zolotarev rational function associated with E and G. When E and G are as in Theorem 7, we choose instead to use the zeros and poles of Faber rational function $\tilde{r}_k$. We refer to these poles and zeros as Faber shifts. The bounds on $Z_k(E, G)$ in Theorem 7 also bound the expression involving s_k in (3.11). Since the bounds decay with k at essentially the same rate as $Z_k(E, G)$, the Faber shifts are nearly optimal shift parameters.

We do not claim to have an efficient method for computing Faber shifts; the approach in Section 3.3.3 is impractical for applications. For convex E, G, we observe that ADI with shifts derived from other so-called asymptotically optimal rational functions [154], i.e., rationals s_k such that the limit in (1.28) holds, often perform comparably to ADI with Faber shifts (see Figure 3.8). This includes the generalized Fejér points [172], which can be computed with the inverse conformal map Ψ from Section 3.2, and the generalized Leja points, which are computed recursively by a greedy process [11, 154]. We describe these points in more detail in Section 1.5.3. The great advantage of the Faber shifts is that explicit upper bounds on $Z_k(E, G)$ are available when k is finite. These bounds capture the capabilities of the ADI method, and they are also useful for error analysis in other contexts. For example, the convergence behavior of the RKSM algorithm [46] and the skeleton decomposition method for low rank matrix approximations [46, 124, 168] can be understood via bounds on $Z_k(E, G)$. The Faber rationals may also be illuminating in the study of rational approximations to the function $f(z) = \text{sign}(z)$ on on $E \cup G$, where E is confined to the left half-plane and G is confined to the right half-plane. This is because there is a mathematical equivalence between the Zolotarev rational functions and the best type (k, k) rational approximations to f in the infinity norm taken over $E \cup G$ [94].

CHAPTER 4

ADI-BASED HIERARCHICAL LINEAR SOLVERS

In some instances where X satisfies (1.1), the ADI method can be used as an ancillary compression routine for computing with X in other tasks (e.g., matrix–vector products, solving linear systems). The bounds on singular values in [19] for real-valued Vandermonde, Cauchy and Pick matrices (among others), reveal that these matrices are well-approximated by low rank matrices. Since the bounds are derived using solutions to Zolotarev's problem, fADI can be applied using optimal ADI shift parameters to cheaply construct low rank approximations (see Sections 1.4.2 and 1.5). For example, a rank k approximation $V^{(k)}$ to the real-valued $n \times n$ Vandermonde matrix V, where $\|V - V^{(k)}\| \leq Ch^{-k}\|V\|_2$, can be constructed in low rank form in only $\mathcal{O}(nk)$ operations.

In this chapter[1], we consider matrices that require more complicated compression schemes. Our methods can be used to explain compression schemes that are exploited in existing superfast solvers for linear systems $Ty = b$, where T is a Toeplitz matrix (Section 4.1). This leads to more general ADI-based rank-structured compression strategies, and lays the groundwork for developing superfast solvers for linear systems involving other matrices, including general Cauchy matrices, non-uniform discrete Fourier transform (NUDFT) matrices, Toeplitz+Hankel matrices, and various evaluation matrices for orthogonal polynomial expansions [100].

The key structure we exploit is that these matrices satisfy Sylvester equations

[1] This chapter is largely based on a manuscript [18] by Bernhard Beckermann, Daniel Kressner, and myself. I am the lead author, and I developed the algorithms, theorems, and accompanying software discussed in the chapter. A variation on the main theorems was independently derived by Beckermann; these can be found in [18].

of the form $AX - XB = LH^*$ where

(1) $\mathrm{rank}(LH^*)$ is small,

(2) A and B are normal and can be diagonalized using fast transforms,

(3) subsets of $\lambda(A)$ and $\lambda(B)$ are well-separated.

Unlike our assumptions in the last two chapters, here we consider the case where $\lambda(A) \subset E$ and $\lambda(B) \subset G$, with E and G intersecting or even coinciding. In these cases, we expect that X is not compressible. However, the fast diagonalization property means that we can efficiently solve $Xy = b$ by working instead with a transformed linear system $\widetilde{X}y = \widetilde{b}$, where $\widetilde{X}$ has the Cauchy-like displacement structure,

$$D_A\widetilde{X} - \widetilde{X}D_B = \widetilde{L}\widetilde{H}^*, \tag{4.1}$$

with D_A and D_B diagonal. The matrix $\widetilde{X}$ may not be of low numerical rank, but because of assumption (3), it has submatrices that are, and these submatrices inherit Cauchy-like displacement structures from $\widetilde{X}$. This observation motivates our development of ADI-based analogues to compression methods from the multipole expansion and randomized linear algebra communities [35, 77, 111, 112, 116, 180]. We develop our ideas within the context of explaining compression properties used in superfast Toeplitz linear systems, though we keep in mind the larger goal of developing a broader collection of ADI-based solvers.

4.1 Rank–structured superfast Toeplitz solvers

A matrix that is constant along each diagonal is called a Toeplitz matrix:

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix} \in \mathbb{C}^{n\times n}.$$

Toeplitz matrices appear frequently in computational mathematics, arising in applications such as time series analysis and dynamical systems [5, 96], image and signal processing [70, 96], and in numerical methods for solving PDEs [71, 83]. Although the development of fast (and superfast) solvers for linear systems with Toeplitz matrices is a classical topic that is more than 60 years old [83, 96], there have been significant advances in recent years. In particular, an efficient class of solvers is based on the observation that any Toeplitz matrix can be transformed into a matrix C with compressible off-diagonal blocks [33, 112, 180]. This allows one to leverage existing algorithms for matrices with hierarchical low rank structures. Here, we perform an analysis of the compressibility of C. This completes and improves upon existing analyses, and allows us to design a new, ADI-based algorithm for compressing C that can be extended to other matrices with displacement structures similar to that of T.

Most fast solvers for $Tx = b$ are based on the observation that the displacement rank of T is small. In this work, we use the fact that T satisfies

$$ST - TS = LH^*, \quad S = \begin{pmatrix} 0 & 1 \\ I_{n-1} & 0 \end{pmatrix}, \quad \mathrm{rank}(LH^*) = \rho \ll n. \tag{4.2}$$

For a Toeplitz matrix T, one has that $\rho \leq 2$.

Algorithm 1 Superfast solvers for a Toeplitz-like linear system (4.3).

```
1: Compute $\widetilde{b} = \mathcal{F}b$, $\widetilde{L} = \mathcal{F}L$, $\widetilde{H} = \mathcal{F}H$.
2: Determine $\widetilde{C}$, a hierarchical low rank approximation of $C = \mathcal{F}T\mathcal{F}^*$ from its
   generators $\widetilde{L}$, $\widetilde{H}$.
3: Solve $\widetilde{C}\widetilde{x} = \widetilde{b}$.
4: Compute $x = \mathcal{F}^*\widetilde{x}$.
```

This choice of the displacement equation for T is by no means unique. Our choice (4.2) is simple but not suitable for all purposes because the linear operator $T \mapsto ST - TS$ is singular. For this reason, T cannot be fully reconstructed from the so-called *generator matrices* L and H. Other choices have been made in the literature. Classical fast solvers of complexity $\mathcal{O}(n^2)$, such as the generalized Schur algorithm, exploit the observation that Schur complements and inverses preserve the displacement rank, which in turn allows one to perform certain operations on T, such as the LU factorization, entirely in terms of the generator matrices; see [96] for a survey.

The matrix S in (4.2) is circulant and thus diagonalized by a discrete Fourier transform $\mathcal{F}$, where $\mathcal{F}^*\mathcal{F} = I$. The transformed matrix $C = \mathcal{F}T\mathcal{F}^*$ satisfies a Sylvester equation with diagonal coefficients. This observation has been used in [63] to derive fast solvers with enhanced numerical stability. More recently, several works [33, 112, 180] have derived new superfast solvers by observing that submatrices of C not containing entries on the main diagonal are well-approximated by low rank matrices. Algorithm 1 describes the general framework of these solvers applied to a linear system

$$Tx = b, \quad T \in \mathbb{C}^{n \times n}, \tag{4.3}$$

with T satisfying (4.2).

Steps 1 and 4 of Algorithm 1 can be easily accomplished in $\mathcal{O}(n \log n)$ operations using fast Fourier transforms (FFTs).[2] For Step 2, several hierarchical approximations of C are possible. This includes the relatively simple HODLR format, as well as the more involved SSS (sequentially semi-separable) and HSS (hierarchical semi-separable) formats used in [33] and [111, 180], respectively.[3] In all these formats, $\widetilde{C}$ is constructed by recursively partitioning C into successively smaller submatrices and replacing each submatrix by a low rank approximation. The SSS and HSS formats have additional structure; the low rank factors at finer partition levels are nested within the low rank factors used at coarser levels of the recursion tree; see Section 4.4. Their use leads to particularly efficient $\mathcal{O}(nr^2)$ direct solvers for Step 3 [34, 180], where the rank of each low rank approximant in $\widetilde{C}$ is bounded by r . The efficiency of Steps 2 and 3 depends on (i) the hierarchical low rank format used, (ii) the compressibility of C, (iii) the method of approximation used to find $\widetilde{C}$, and (iv) the type of solver used in Step 3. In this work, we focus on analysis and improvements related to (ii) and (iii).

Main results. We introduce new, explicit bounds on the singular values of submatrices of C that thoroughly explain its low rank properties. While some theoretical descriptions of the rank structure of C are given in [33, 111], these arguments are qualitative. Upon careful inspection (see Section 4.2), they do not fully justify the use of HODLR, HSS, or other weakly-admissible [77] hierarchical formats. We note, however, that the larger focus of these papers is the development of efficient numerical techniques for constructing and inverting HSS approximations to C, and that the implemented methods work well in practice,

[2] For the singular Sylvester equation $ST - TS$, the main diagonal of C cannot be computed from the generator matrices. A simple formula for recovering the diagonal of C from T in only $\mathcal{O}(n \log n)$ flops follows from projecting T onto the subspace of circulant matrices.

[3] Note that the more general $\mathcal{H}$ and $\mathcal{H}^2$-type formats [16, 76] are also applicable but such a higher level of generality is not needed given the relatively simple structure of C.

even if they have incomplete justifications. Our arguments fully explain why these methods work well in practice, and they also supply a priori estimates on the numerical ranks of off–diagonal submatrices of C, including those that arise from HODLR and HSS partitioning. This leads to fully adaptive hierarchical factorization schemes that automatically select appropriate approximation parameters based on a relative tolerance parameter $0 < \epsilon < 1$. Our bounds show that generally, an off-diagonal HSS or HODLR submatrix of C has ϵ-rank of size $\mathcal{O}(\log n \log(1/\epsilon))$ (see Theorems 9 and 10).

To complement our bounds, we apply the alternating implicit direction (ADI) method to construct low rank approximations with controllable errors, and we introduce an ADI-based interpolative decomposition for constructing HSS factors with special, nested structures. The current state of the art for solving (4.3) given an arbitrary Toeplitz matrix is an implementation of Algorithm 1 that uses extremely efficient compression routines based on randomized linear algebra [180]. Our HSS-based solver in Section 4.4 is a competitive analogue that does not rely on randomized linear algebra or any particular properties of C (e.g., fast matrix-vector products inherited from T) other than its displacement structure. Our approach can therefore be used with linear systems involving other types of matrices. Complexity analyses and numerical tests reveal that the ADI-based method is competitive with the randomized method, and is in fact cheaper by a modest factor.

The rest of this discussion is organized as follows: In Section 4.2, we derive new bounds that characterize the compressibility of C. In Sections 4.3 and 4.4, we show how our bounds can be used, in combination with ADI, to construct HODLR and HSS approximations to C. We describe a practical implementation of an HSS-based solver, with numerical results, in Section 4.4.5. Then in Section 4.5, we briefly discuss the extension of these ideas to other linear systems.

4.1.1 The displacement structure of Toeplitz matrices

Starting with the Sylvester equation satisfied by T in (4.2), we first diagonalize S with a discrete Fourier transform $\mathcal{F}$, where $\mathcal{F}^*\mathcal{F} = I$. Throughout, we assume n is a power of 2.[4] Letting $\omega = \exp(i\pi/n)$, we have that

$$\mathcal{F}S\mathcal{F}^* = D, \quad \mathcal{F} = \left(\frac{\omega^{j(2k-1)}}{\sqrt{n}}\right)_{j,k=1,\ldots,n}, \tag{4.4}$$

where $D = \mathrm{diag}(\omega^2, \omega^4, \ldots, \omega^{2n})$. The transformed matrix $C = \mathcal{F}T\mathcal{F}^*$ satisfies the equation

$$DC - CD = \widetilde{L}\widetilde{H}^*, \quad \widetilde{L} = \mathcal{F}L, \quad \widetilde{H} = \mathcal{F}H, \tag{4.5}$$

with the same displacement rank $\rho \le 2$.

Each off-diagonal entry C_{jk}, $j \neq k$, can be recovered by multiplying the corresponding off-diagonal entry of $\widetilde{L}\widetilde{H}^*$ by $1/(\omega^{2j} - \omega^{2k})$. The latter can be viewed as the discretization of the function $f(x, y) = 1/(x - y)$ with x, y both on the unit circle. When $|x - y|$ is large enough, a truncated Taylor expansion of f can be used to construct good low rank approximations to those submatrices of C only containing indices (j, k) for which $|\omega^{2j} - \omega^{2k}|$ is not small. Intuitively, this suggests that submatrices of C far from the main diagonal as well as the top-right and bottom-left corners (due to periodicity) should be compressible. Such submatrices are depicted in Figure 4.1 (right). To get sharper bounds, especially

[4] This assumption simplifies the notation and discussion related to HODLR and HSS formats, since it relates these partitioning schemes to perfectly balanced binary trees. HODLR and HSS matrices can be constructed when n is not a power of 2 with balanced binary trees instead [113].

for regions close to the singularity of f so that we can describe the compressibility of all of the off-diagonal submatrices, we appeal to arguments based on C's displacement structure.

4.2 The submatrices of the transformed Toeplitz matrix

Bounds derived from the properties of the Zolotarev numbers and Theorem 2 cannot be applied directly to the Sylvester equation (4.5) because its coefficients are identical and thus have spectra that are anything but disjoint. On the other hand, any submatrix of C also satisfies a Sylvester equation. For subsets J and K of $I_0 = \{1, \ldots, n\}$, we let C_{JK} denote the $|J| \times |K|$ submatrix of C containing all entries C_{jk} with $j \in J$, $k \in K$. By (4.5), we have that

$$D_J C_{JK} - C_{JK} D_K = \widetilde{L}_J \widetilde{H}_K^*, \tag{4.6}$$

where the diagonal matrix D_J contains the diagonal elements ω^{2j} for $j \in J$, D_K is defined analogously, and $\widetilde{L}_J$, $\widetilde{H}_K$ contain the corresponding rows of $\widetilde{L}$, $\widetilde{H}$. As long as $J \cap K = \emptyset$, (4.6) is nonsingular and the spectral sets $\lambda(D_J)$ and $\lambda(D_K)$ are contained in disjoint arcs on the unit circle, which we call $\mathcal{A}_J$ and $\mathcal{A}_K$, respectively (see Figure 4.2). From Theorem 2, we have that for $0 \leq 2k \leq n-1$,

$$\sigma_{2k+1}(C_{JK}) \leq Z_k(\mathcal{A}_J, \mathcal{A}_K)\|C_{JK}\|_2. \tag{4.7}$$

An explicit bound on $Z_k(\mathcal{A}_J, \mathcal{A}_K)$ will depend on the distance between the endpoints of the arcs $\mathcal{A}_J$ and $\mathcal{A}_K$. We have the following result:

Theorem 8. *Let $\mathcal{A}_\tau = \{e^{it} : t \in [\tau_1, \tau_2]\}$, and let $\mathcal{A}_\kappa = \{e^{it} : t \in [\kappa_1, \kappa_2]\}$, where $0 < \tau_1 < \tau_2 < \kappa_1 < \kappa_2 \leq 2\pi$. Then,*

$$Z_k(\mathcal{A}_\tau, \mathcal{A}_\kappa) \leq 4\mu_0^{-2k}, \quad \mu_0 = \exp\left(\frac{\pi^2}{2\log(16\gamma)}\right), \tag{4.8}$$

where

$$\gamma = \frac{|\sin((\kappa_1 - \tau_1)/2)\sin((\kappa_2 - \tau_2)/2)|}{|\sin((\kappa_2 - \tau_1)/2)\sin((\kappa_1 - \tau_2)/2)|}.$$

Proof. Let $\mathcal{T}$ be a Möbius transformation that maps $\mathcal{A}_\tau \cup \mathcal{A}_\kappa$ to two disjoint intervals $[a, b] \cup [c, d]$ of $\mathbb{R}$, where

$$[a, b] := [\mathcal{T}(e^{i\tau_1}), \mathcal{T}(e^{i\tau_2})], \quad [c, d] := [\mathcal{T}(e^{i\kappa_1}), \mathcal{T}(e^{i\kappa_2})].$$

By Property (6) in Lemma 1, we have that $Z_k(\mathcal{A}_\tau, \mathcal{A}_\kappa) = Z_k([a, b], [c, d])$, so from Theorem 3, we have that

$$Z_k(\mathcal{A}_\tau, \mathcal{A}_\kappa) \leq 4\mu_0^{-2k},$$

where $\gamma = (|c - a|\,|d - b|)/(|c - b|\,|d - a|)$ is the modulus of the cross ratio of (a, b, c, d). The cross-ratio is invariant under $\mathcal{T}$, i.e.,

$$\gamma = \frac{|e^{i\kappa_1} - e^{i\tau_1}|\,|e^{i\kappa_2} - e^{i\tau_2}|}{|e^{i\kappa_1} - e^{i\tau_2}|\,|e^{i\kappa_2} - e^{i\tau_1}|}.$$

The theorem then follows from the chord identity. □

We can now use Theorem 8 to explicitly bound the ϵ-ranks of submatrices of C. We are particularly interested in the types of submatrices that appear in factorization schemes for constructing HODLR and HSS approximations to C. We review the partitioning schemes with more formality in Sections 4.3 and 4.4, but state the main results here. We begin with HODLR blocks, which are defined as square super-diagonal and sub-diagonal blocks of C (see Figure 4.3).

Theorem 9 (HODLR submatrix bounds)**.** *Let C_{JK} be an $m \times m$ off–diagonal HODLR block of the $n \times n$ matrix C, where $1 \leq m \leq n/2$ and C is as in* (4.5)*. Then, for $0 < \epsilon < 1$,*

$$\mathrm{rank}_\epsilon(C_{JK}) \leq 2\left\lceil \frac{2}{\pi^2}\log\left(\frac{2n}{\sqrt{\min(2m-1, n-2m+1)}}\right)\log\left(\frac{4}{\epsilon}\right)\right\rceil. \tag{4.9}$$

Proof. We make the assumption that C_{JK} is a superdiagonal block, since for subdiagonal blocks, one can apply the same argument using C^T. By definition (see Section 4.3), the index sets J and K are of the form

$$J = \{p, p+1, \dots p+m-1\} \subset I_0, \quad K = \{p+m, p+m+1, \dots, p+2m-1\} \subset I_0,$$

where $I_0 = \{1, \dots, n\}$. Since C_{JK} satisfies (4.6), the singular values of C_{JK} are bounded as in (4.7), where

$$\mathcal{A}_J = \{e^{it} : t \in [2\pi p/n, 2\pi(p+m-1)/n]\} \tag{4.10}$$

$$\mathcal{A}_K = \{e^{it} : t \in [2\pi(p+m)/n, 2\pi(p+2m-1)/n]\}. \tag{4.11}$$

To complete the proof, we need to bound $Z_k(\mathcal{A}_J, \mathcal{A}_K)$ in terms of m and n. We note from Theorem 8 that

$$Z_k(\mathcal{A}_J, \mathcal{A}_K) \leq 4 \exp\left(\frac{\pi^2}{2\log(16\gamma)}\right)^{-2k}.$$

Using basic estimates of $\sin(\pi x)$, we find that

$$\gamma = \frac{|\sin\theta_1|^2}{|\sin\theta_2|, \; |\sin\theta_3|} \leq \frac{1}{\left(\dfrac{2\min(2m-1, n-2m+1)}{n}\right)\dfrac{2}{n}} = \frac{n^2}{4\min(2m-1, n-2m+1)},$$

where $\theta_1 = \pi m/n$, $\theta_2 = \pi\left((2m-1)/n \mod (1/2)\right)$, $\theta_3 = \pi/n$. It follows that

$$Z_k(\mathcal{A}_J, \mathcal{A}_K) \leq 4 \exp\left(\pi^2 / \log\left(\frac{2n}{\sqrt{\min(2m-1, n-2m+1)}}\right)/4\right)^{-2k}, \tag{4.12}$$

and the theorem then follows from the definition of ϵ-rank. □

Many HSS factorization schemes require a special type of submatrix, which we call a maximal submatrix. They are also referred to as HSS block columns in the literature [111].

Definition 2. *The submatrix C_{JK}, where $J = \{p, p+1, \dots, m+p-1\} \subset I_0$ and $K = I_0 \setminus J$, is called a maximal submatrix of size m.*

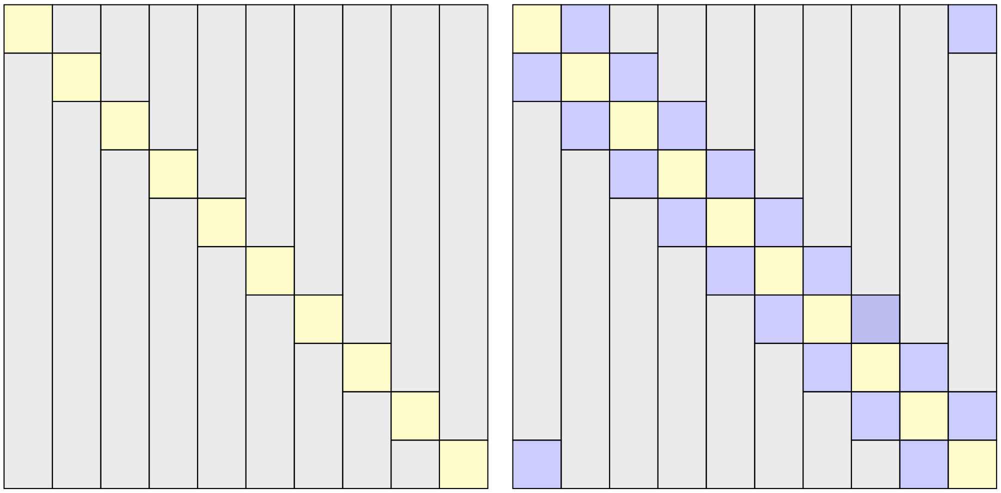


Figure 4.1: (Left) Definition 2 applied to the block columns of an $n \times n$ matrix partitioned into slabs of m columns. The size m maximal submatrices are shown in gray. Only the diagonal blocks (yellow) are not admissible. (Right) Strongly admissible submatrices are shown in gray. Now the super/subdiagonal blocks, as well as the bottom left/top right blocks (blue), are not admissible.

These submatrices are displayed in Figure 4.1 (left). The notion of HSS block rows is obtained by applying Definition 2 to C^T. In the literature on hierarchical and $\mathcal{H}^2$ matrices [77], these blocks satisfy the weak admissibility criterion. In contrast, the blocks depicted on the right loosely correspond to the strong admissibility criterion. More details on the HSS format are given in Section 4.4. Here, we prove that they have ϵ-ranks of size $\mathcal{O}(\log n \log(1/\epsilon))$. We remark that Theorem 8 implies that the ϵ-ranks of the strongly admissible blocks can be bounded independently of n.[5]

Theorem 10 (HSS submatrix bounds). *Let C_{JK} be a size m maximal submatrix of C or C^T, where $1 \leq m \leq n/2$ and C is as in* (4.5). *Then, for $0 < \epsilon < 1$,*

$$\operatorname{rank}_\epsilon(C_{JK}) \leq 2\left\lceil \frac{2}{\pi^2}\log(2n)\log\left(\frac{4}{\epsilon}\right)\right\rceil. \tag{4.13}$$

Proof. The proof is similar to the argument given in the proof for Theorem 9,

[5] This observation was made by Beckermann; details can be found in our manuscript [18].

with the only difference being that the arcs $\mathcal{A}_J$ and $\mathcal{A}_K$ are defined using J and K from Definition 2. This leads to a slight modification in the bound on γ. □

In many HSS compression schemes, one must also work with submatrices of maximal submatrices. We make the simple but important observation here that Zolotarev numbers also control the ϵ–ranks of these submatrices.

Corollary 5. *Let X be any submatrix of the maximal size m submatrix C_{JK}, where $1 \leq m \leq n/2$ and the $n \times n$ matrix C is as in* (4.5). *Then, for $0 < \epsilon < 1$,*

$$\operatorname{rank}_\epsilon(X) \leq \operatorname{rank}_\epsilon(C_{JK}). \tag{4.14}$$

Proof. Observe that $\operatorname{rank}(D_{\widehat{J}}X - XD_{\widehat{K}}) \leq 2$, where $D_{\widehat{J}}$ is a diagonal submatrix of D_J, and $D_{\widehat{K}}$ is a diagonal submatrix of D_K. It follows from Theorem 2 that the nonzero singular values of X are bounded such that

$$\sigma_{2k+1}(X) \leq Z_k(\lambda(D_{\widehat{J}}), \lambda(D_{\widehat{K}}))\|X\|_2.$$

Since $\lambda(D_{\widehat{J}}) \subset \mathcal{A}_J$ and $\lambda(D_{\widehat{K}}) \subset \mathcal{A}_K$, we have from (P5) in Lemma 1 that $Z(\lambda(D_{\widehat{J}}), \lambda(D_{\widehat{K}}) \leq Z(\mathcal{A}_J, \mathcal{A}_K)$. The theorem then follows. □

Related results. The authors of [33] also present an argument for the low numerical rank of the off-diagonal submatrices of C. However, an observation made by Beckermann (see [18]) shows that this argument is incomplete. We repeat the details here. Consider the submatrix $C_{\widehat{J}\widehat{K}}$, where $\widehat{J} \subset J = \{p, p+1, \ldots, p+m-1\} \subset I_0$ and $\widehat{K} \subset K = I_0 \setminus J$. The authors in [33] construct two concentric disks centered at $c \in \mathbb{C}$, with radii selected so that $\lambda(D_{\widehat{J}}) \subset \{|z-c| \leq r_E\}, \lambda(D_{\widehat{K}}) \subset \{|z-c| \geq r_F\}$, and $r_E/r_F < 1$ (see [33, Sec.2.2, eq. 2.6]). They do not explicitly mention Zolotarev numbers, but their

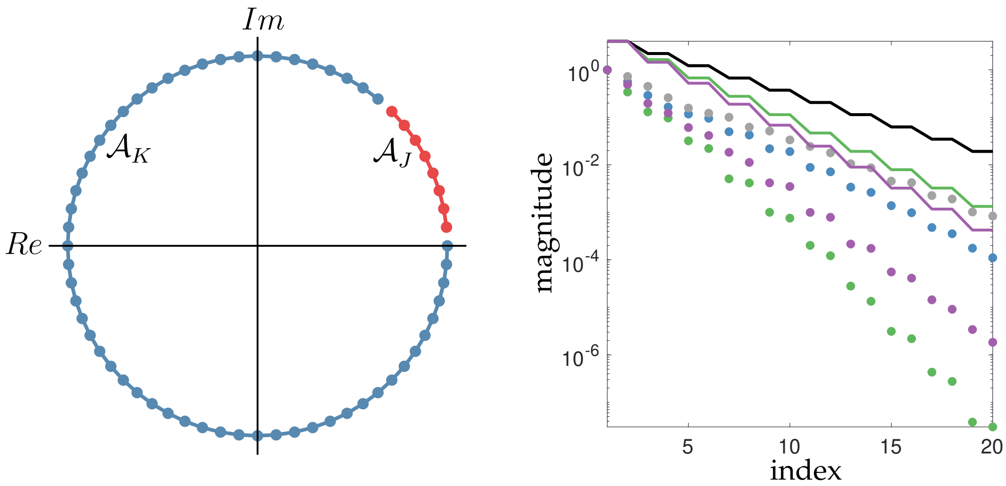


Figure 4.2: Left: The spectra $\lambda(D_J)$ (blue dots) and $\lambda(D_K)$ (red dots) in (4.6) for $J = \{1, 2, \ldots, m\}$, $K = I_0 \setminus J$ are contained in arcs $\mathcal{A}_J$ and $\mathcal{A}_K$. In this case, C_{JK} is a maximal $m = 8$ submatrix of the 64×64 matrix C. Our bounds depend on the cross ratio of the end points of $\mathcal{A}_J$ and $\mathcal{A}_K$. Right: The normalized singular values (dots) for maximal submatrices and HODLR submatrices of C are plotted on a log scale along with explicit bounds on their magnitudes from our theorems (solid lines). Here, C is of size $n = 2048$. We plot singular values for maximal submatrices (HSS columns) of size $m = 128$ (blue), $m = 512$ (grey), as well as for superdiagonal HODLR submatrices where $m = 128$ (green) and $m = 512$ (purple). The bound for the singular values of the maximal submatrices only depends on n and is shown in black.

argument is equivalent to finding k, where

$$Z_k\left(\lambda(D_{\widehat{J}}), \lambda(D_{\widehat{K}})\right) \leq (r_E/r_F)^k \leq \epsilon,$$

where the second inequality can be derived from potential theory [145] (see Section 1.5.3). The argument in [33] is qualitative and gives no explicit formula for c. However, a near-optimal choice seems to be given by $c = \exp(i\frac{\pi}{n}(2p+m-1))$, i.e., the midpoint of the arc $\mathcal{A}_J$. With this choice,

$$r_E = 2\sin\left(\pi\frac{m-1}{n}\right), \quad r_F = 2\sin\left(\pi\frac{m}{n}\right),$$

and if $m \sim \sqrt{n}$, then the quantity $(r_E/r_F)^k$ is only small when $k \geq m$. In other words, this approach does not show that in general, the off-diagonal blocks of C are of small numerical rank. Similar problems occur in [112, Lemma 3], which relies on results from [48, Sec. 2.2]. It follows that these approaches do not fully justify the use of HSS and HODLR-type submatrices. Even so, HSS partitioning schemes are used and perform well in practice [33, 180]. Our theorems now explain why this is the case.

4.3 An ADI-based HODLR approximation for the transformed Toeplitz matrix

With bounds established that describe the rank structure of C, we now turn to practical considerations. Our bounds can be used to implement compression strategies and provide error estimates for superfast Toeplitz solvers that follow the general outline of Algorithm 1. We provide descriptions for two such solvers. The first is based on a HODLR approximation of C, and has the advantage of being relatively simple to implement. Then in Section 4.4, we discuss in more detail an HSS-based solver.

As we show in Theorem 9, the off-diagonal blocks of C are numerically of low rank. One of the simplest ways to take advantage of the compressible off-diagonal blocks of C is to use the HODLR hierarchical structure [76]. We provide a brief review of the format here. A matrix $\tilde{C} \in \mathbb{C}^{n\times n}$ is called a HODLR matrix [76, 111] if it can be partitioned into equal-sized blocks

$$\tilde{C} = \begin{bmatrix} \tilde{C}_{11} & \tilde{C}_{12} \\ \tilde{C}_{21} & \tilde{C}_{22} \end{bmatrix}, \tag{4.15}$$

where $\tilde{C}_{12}$, $\tilde{C}_{21}$ have low rank representations, and $\tilde{C}_{11}$, $\tilde{C}_{22}$ can again be partitioned with the same form as (4.15). This partitioning is continued until a minimum block size is reached. It is convenient to associate HODLR matrices with

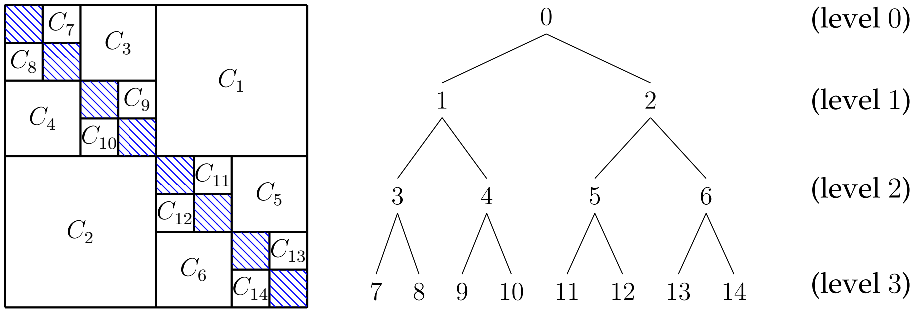


Figure 4.3: A HODLR approximation to C is found by partitioning C according to the binary tree $\mathcal{T}$ (right) of depth 3. Every labeled block corresponds to a vertex on the tree, with the root vertex (level $\ell = 0$) corresponding to C as a whole. Note that the row indices of a parent block are split among its children blocks (e.g., the rows of C_1 are shared by its children C_3 and C_4). Each labeled block can be approximately represented in low rank form. The filled diagonal blocks (blue) at the finest partition level require an explicit representation.

perfectly balanced binary trees. Let $\mathcal{T}$ denote a tree of depth $\widehat{\ell}$. We number the vertices of the tree consecutively, level by level, as in Figure 4.3, so that at level ℓ, the vertices are numbered $(2^{\ell}-1), \dots, (2^{\ell+1}-2)$. Each vertex v then has children $2v+1$ and $2v+2$. As shown in Figure 4.3, each v is associated with an off-diagonal block C_v from the HODLR partition of C. Each v then naturally has an index set associated with it, namely, the row indices defining C_v (see Figure 4.3). In notational terms, the indices are as follows: Set $I_0 = \{1, \dots, n\}$. For each parent node v, define an index set I_v of recursively so that so that $I_v = \{I_{2v+1}, I_{2v+2}\}$ is length m, with $I_{2v+1} = \{(I_v)_1, \dots (I_v)_{m/2}\}$ and $I_{2v+2} = I_v \setminus I_{2v+1}$. Then, $C_v = C(I_v, I_{\widetilde{v}})$, where $\widetilde{v}$ is the sibling node of v in $\mathcal{T}$. We call the matrices C_v the HODLR blocks of C.

To find a HODLR matrix that approximates C, we first partition C according to the tree in Figure 4.3. Then, a low rank approximation to each off-diagonal HODLR block C_v must be computed. Taken together, the collection of low rank approximations to C_v for all $v > 0$ in $\mathcal{T}$ is referred to as the HODLR factorization of $\widetilde{C}$, where $\widetilde{C} \approx C$. This completes Step 2 in Algorithm 1. Implemented naively with a rank-revealing method such as the randomized SVD [79], the cost for compressing the largest blocks C_1 and C_2 is $\mathcal{O}(n^2 r + nr^2)$[6], where r is the rank of the constructed approximant. By Theorem 9, we have that $r = \mathcal{O}(\log n \log(1/\epsilon))$. However, we can use the fact that approximate solutions to (4.6) constructed with fADI (see Section 1.4.2) are low rank approximations to C_v. The optimal ADI shift parameters for applying fADI are the zeros and poles of the Zolotarev rational function associated with $Z_k(\mathcal{A}_J, \mathcal{A}_K)$, and these have an exact formula in terms of elliptic integrals (see Corollary 2). Given a tolerance parameter ϵ_v, we choose

$$k_v = \left\lceil \frac{2}{\pi^2} \log\left(\frac{2n}{\sqrt{\min(2m-1, n-2m+1)}} \right) \log\left(\frac{4}{\epsilon}\right) \right\rceil ,$$

as in Theorem 9, and then apply k_v steps of fADI to (4.6) to construct $\widetilde{C}_v = C_v^{(k_v)}$ in low rank form. Here, $\operatorname{rank}(C_v^{(k_v)}) \leq 2k_v$ and we are guaranteed that $\|C_v - \tilde{C}_v\|_2 \leq \epsilon_v \|C_v\|_2$. The cost for compressing the largest HODLR blocks C_1 and C_2 of C with fADI is only $\mathcal{O}(n \log n \log(1/\epsilon))$, where we have used that k_v is on the order $\mathcal{O}(\log n \log(1/\epsilon))$. Once the HODLR approximation to C is constructed, fast recursive procedures can then be used to find, for example, an LU decomposition of $\widetilde{C}$ and approximately solve $\widetilde{C}\widetilde{x} = \widetilde{b}$ in only $\mathcal{O}(r^2 n \log^2 n)$ flops [113], completing Step 3 of Algorithm 1.

In addition to its Cauchy-like displacement structure, C has special self-similarity structures that can be exploited to further reduce the overall cost of Step 2 [113, 180]. If these structures are used, fADI only needs to be applied to one block at each level ℓ in the tree $\mathcal{T}$. The rest of the low rank factorizations can be generated without any additional computations using the self-similarity

[6] If a fast matrix-vector product for C is used, the cost is $\mathcal{O}(nr \log n + nr^2)$.

structure. If there are $\mathcal{O}(\log n \log(1/\epsilon))$ levels in $\mathcal{T}$, then total cost for constructing a HODLR approximation to C is less than $\mathcal{O}(n \log^2 n \log^2(1/\epsilon))$. This is not the main observation of this work; we refer to our manuscript [18] for details.

4.4 An ADI-based HSS approximation to the transformed Toeplitz matrix

Significant savings in Steps 2 and 3 of Algorithm 1 can be gained by using more complicated hierarchical matrix formats. Here, we consider the HSS format [16, 76, 111]. An HSS matrix $\widetilde{C} \approx C$ is constructed by finding low rank approximations to the HODLR blocks of C that satisfy special recursive relations across partition levels. Specifically, let every low rank block in the HODLR partition of $\widetilde{C}$ be expressed as $\widetilde{C}_v = U_v B_v V_v^*$, where for simplicity, we assume each block has a rank of r (so B_v is $r \times r$).

We require that every parent block at level $\ell \geq 1$ has low rank factors of the form

$$U_v = \begin{bmatrix} U_{2v+1} & 0 \\ 0 & U_{2v+2} \end{bmatrix} R_v, \quad V_v = \begin{bmatrix} V_{2v+1} & 0 \\ 0 & V_{2v+2} \end{bmatrix} W_v, \quad R_v, W_v \in \mathbb{C}^{2r \times r}. \quad (4.16)$$

The so-called translation matrices, R_v and W_v, serve to recompress the factorization and link U_v to the low rank factors of its children blocks [111]. As a result, factors from finer partition levels are always nested within factors constructed at coarser levels. For example, following the partitioning and submatrix labeling scheme shown in Figure 4.3, an approximate HSS factorization for the submatrix $C_1 \approx U_1 B_1 V_1^*$ must be of the form

$$\underbrace{\begin{bmatrix} U_7 & & & \\ & U_8 & & \\ & & U_9 & \\ & & & U_{10} \end{bmatrix}}_{4m\times 4p} \underbrace{\begin{bmatrix} R_3 & \\ & R_4 \end{bmatrix}}_{4p\times 2p} \underbrace{R_1}_{2p\times p} \underbrace{B_1}_{p\times p} W_1^* \begin{bmatrix} W_3^* & \\ & W_4^* \end{bmatrix} \begin{bmatrix} V_7^* & & & \\ & V_8^* & & \\ & & V_9^* & \\ & & & V_{10}^* \end{bmatrix}, \tag{4.17}$$

where the blocks at the finest partition (level $\widehat{\ell}$) are of size $m \times m$, with $m = n/2^{\widehat{\ell}}$. Notice that for each v on level $\ell = 1, \ldots, \widehat{\ell}-1$, we only require the construction and storage of the small matrices R_v, W_v, and B_v, since U_v and V_v can be recursively constructed from factors at deeper levels. At the finest level $\widehat{\ell}$, we must store the diagonal $m \times m$ blocks that cannot be compressed, as well as $U_v, B_v,$ and V_v for each of the $2^{\widehat{\ell}}$ vertices. We refer to the collection of these factors as the *HSS factorization* of $\widetilde{C}$.

4.4.1 Superfast HSS-based solvers

Once $\widetilde{C} \approx C$ is known via its HSS factorization, storage requires only $\mathcal{O}(nr)$ memory and the matrix-vector product $\widetilde{C}y$ can be found in only $\mathcal{O}(nr)$ flops [111, Alg. 1]. Crucially, the HSS structure makes solving $\widetilde{C}\widetilde{x} = \widetilde{b}$ especially efficient: A ULV factorization for HSS matrices is introduced in [34]. The 'U' and 'V' refer to sequences of unitary factors that are applied to partially triangularize certain blocks $\widetilde{C}$ at each level of $\mathcal{T}$. This is done in a way that takes advantage of the nested hierarchical structure of the HSS factors. Once the ULV factorization is known, a fast merge and back–substitution routine, followed by a forward-substitution scheme, can be used to solve the linear system. If blocks at the finest level are size $m \approx 2r$, and $\operatorname{rank}(\widetilde{C}_v) \leq r$ for all $v > 0$ in $\mathcal{T}$, then the ULV factorization and solve only require $\mathcal{O}(nr^2)$ and $\mathcal{O}(nr)$ flops, respectively [180].

In [180], a related ULV-like solver is introduced that relaxes the requirement that the factors used to triangularize blocks of the HSS representation C are unitary. If this method is implemented, it saves about a factor of 2 over the standard ULV solver [180, Sec. 3.6]. For this approach to work, every U_v, V_v, R_v, and W_v in the HSS factorization must have the special 'interpolative structure' shown in (4.19). Our displacement-based interpolative decomposition ensures that this is the case. For details (and explanatory pictures) about the ULV and ULV-like solvers, we refer the reader to [34, 180]. Our focus here is on computing the HSS factorization so that either of these solvers can be applied.

4.4.2 HSS rows and columns and the HSS rank

We are interested in an efficient strategy for constructing an HSS factorization. It is standard to work with the special HSS rows and HSS columns for this purpose [111, 179]. Illustrations of these submatrices are given in Figure 4.4. Let v be a vertex at level ℓ in the tree $\mathcal{T}$. We associate v with the $\tau \times \tau$ HODLR block C_v (see Figure 4.3). We denote by C_v^{row} the submatrix $C(I_v, I_v^c)$, where I_v are the row indices of C_v, and $I_v^c = I_0 \setminus I_v$. This selects the block row containing C_v, but excludes the $\tau \times \tau$ diagonal block in the row, as shown in Figure 4.4.[7] In the same way, we define the HSS column C_v^{col} as $C(I_v^c, I_v)$. Note that the HSS columns are maximal size τ submatrices of C (the rows are maximal size τ submatrices of C^T) as defined in Definition 2. Theorem 10 supplies explicit bounds

[7] In some texts, the HSS row C_v^{row} is instead defined instead as $C(I_v, :)$, with the order τ diagonal block in $C(I_v, :)$ set identically to zero [111].

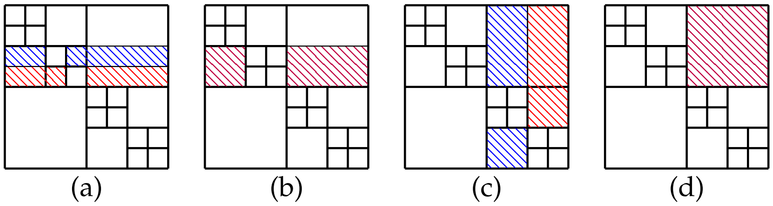


Figure 4.4: Various HSS rows and columns for a tree of depth 3. In (a), the two HSS rows C_9^{row} (blue) and C_{10}^{row} (red) are shown. These are the children of the row C_4^{row}, shown in (b). The two HSS columns C_5^{col} (red) and C_6^{col} (blue) in (c) are children of the parent C_1^{col} in (d).

on $\mathrm{rank}_\epsilon(C_v^{row})$, $\mathrm{rank}_\epsilon(C_v^{col})$ for every vertex v. Crucially, these bounds are also applicable for submatrices of C_v^{row} and C_v^{col} (see Corollary 5).

As discussed in [179], the construction of the HSS matrix $\widetilde{C}$ with each block $\widetilde{C}_v$ of rank at most r is possible if and only if every HSS row and column of $\widetilde{C}$ is of rank at most r. We define the $(\epsilon, \mathcal{T})$-HSS rank of C as r such that $\mathrm{rank}_\epsilon(C_v^{row}), \mathrm{rank}_\epsilon(C_v^{col}) \leq r$ for all $v \neq 0$ in $\mathcal{T}$. An immediate consequence of Theorem 10 is that an upper bound on the $(\epsilon, \mathcal{T})$-HSS rank of C is given by the bound in (4.13).

4.4.3 Interpolative decompositions

We now discuss how to construct an approximate HSS factorization of C. A standard approach involves the use of interpolative decompositions [111, 180]. The structure of the interpolative decompositions is particularly important if one wants to apply the ULV-like solver from [180, Sec. 3.4] in Step 3 of Algorithm 1. Interpolative decompositions can be constructed with extraordinary efficiency using fADI. We briefly review the general use of interpolative decompositions in forming HSS factorizations. Then, in Section 4.4.4, we describe fADI-based interpolative decompositions that lead to a displacement-based method for finding HSS approximations to C.

Interpolative decompositions on leaf nodes

Suppose that v is a leaf node on $\mathcal{T}$ that corresponds to the block C_v, of size $m \times m$. We seek $\widetilde{C}_v$, a low rank approximation to C_v. An interpolative decomposition is a type of low rank factorization that takes the form [35]

$$\widetilde{C}_v = U_v B_v V_v^*, \quad B_v = C(J_v, K_v), \tag{4.18}$$

where J_v and K_v are each a small subset of indices, and U_v, V_v are as in (4.19). To compute U_v and J_v, one requires an approximate one-sided row interpolative decomposition [35] of the HSS row C_v^{row}, which is of size $m \times n-m$. Supposing $\mathrm{rank}(C_v^{row}) \leq r$, this process chooses a subset of r rows as vectors that approximately span the rowspace of C_v^{row} [35, 111]. This results in a factorization of the form

$$C_v^{row} \approx \widetilde{C}_v^{row} = U_v C_v^{row}(\widetilde{J}_v, :) = P \begin{bmatrix} \mathcal{I}_r \\ \mathcal{R} \end{bmatrix} C_v^{row}(\widetilde{J}_v, :), \tag{4.19}$$

where P is a permutation matrix, $\mathcal{I}_r$ is an $r \times r$ identity matrix, and $\widetilde{J}_v$ is a subset of the row indices of C_v^{row}. Recalling that I_v are the row indices of C_v with respect to C, we abuse notation and say that $J_v = I_v(\widetilde{J}_v)$, meaning that J_v is the subset of I_v that is indexed by $\widetilde{J}_v$. Similarly, a one-sided column interpolative decomposition [35] can be applied to C_v^{col} to find V_v and K_v in (4.18). We refer to the structure of U_v as an ‘interpolative structure’.

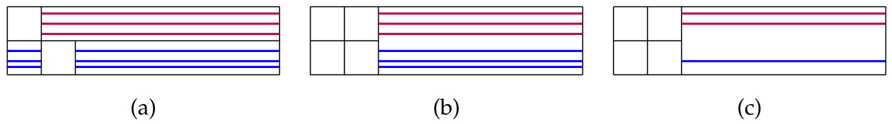
Merging HSS rows and columns

Once the decomposition (4.18) is known for each block at the finest level $\widehat{\ell}$ on $\mathcal{T}$, factors associated with the non-leaf nodes of $\mathcal{T}$ can be determined from them [111]. For illustration's sake, let v be a vertex on $\mathcal{T}$ at level $\ell = \widehat{\ell} - 1$. Then, as before, $\widetilde{C}_v = U_v B_v V_v^*$. In this case, U_v and V_v are as in (4.16), and it only remains to determine R_v, W_v, and the indices defining $B_v = C(J_v, K_v)$. Consider the associated HSS row C_v^{row}, which has children rows C_{2v+1}^{row}, C_{2v+2}^{row}. Since $C_\tau^{row} \approx U_\tau C_\tau^{row}(\widetilde{J}_\tau, :)$, $\tau = 2v+1, 2v+2$, we can write

$$C_v^{row} \approx \begin{bmatrix} U_{2v+1} & 0 \\ 0 & U_{2v+2} \end{bmatrix} C_v^{row}(\widehat{J}, :), \tag{4.20}$$

where $\widehat{J} = \{\widetilde{J}_{2v+1} \cup \widetilde{J}_{2v+2}\}$ is the index set with respect to the block C_v^{row} that selects the $2r$ rows indexed by $\widetilde{J}_{2v+1}$ and $\widetilde{J}_{2v+2}$ (see Figure 4.5, (b)).

A row interpolative decomposition is applied to find the approximation $C_v^{row}(\widehat{J}, :) \approx R_v C_v^{row}(\widetilde{J}_v, :)$, where $\widetilde{J}_v \subset \widehat{J}$, $|\widetilde{J}_v| = r$. This chooses a size r subset of the $2r$ rows comprising $C_v^{row}(\widehat{J}, :)$, and also ensures that R_v has the interpolative structure. We set $J_v = I_v(\widetilde{J}_v)$. As shown in Figure 4.5, this process chooses a subset of row vectors that approximately span the rowspace of C_v^{row}. We may substitute $R_v C_v^{row}(\widetilde{J}_v, :)$ as an approximation for $C_v^{row}(\widehat{J}, :)$ in (4.20). An analogous method is applied to C_v^{col} to find W_v and K_v. The resulting factorization satisfies the nested structure shown in (4.16). This procedure can be repeated at each level as we traverse $\mathcal{T}$ from the bottom up in order to find an approximate HSS factorization of C [111].



(a) (b) (c)

Figure 4.5: (a) Selected row vectors indexed by $\widetilde{J}_\tau$ for the two HSS rows $C_\tau^{row} \approx U_\tau C_\tau^{row}(\widetilde{J}_\tau, :)$, $\tau = 2v+1$ (red) and $\tau = 2v+2$ (blue). (b) The coarser parent HSS row C_v^{row} can be represented using a low rank factorization involving the combined row vectors (colored lines) of its children rows. (c) We apply another interpolative decomposition to represent the parent HSS row with a low rank factorization involving a subset of the available row vectors.

4.4.4 ADI-based interpolative decompositions

Interpolative decompositions can be constructed in many ways. A basic approach involves computing a (strong) rank-revealing QR decomposition of each block being compressed [35, 72]. However, this is expensive. For example, if v is a leaf node, then C_v^{row} is of size $m \times (n - m)$, so the cost is $\mathcal{O}(nm^2)$, and there are $\mathcal{O}(n)$ leaf nodes. Using randomized sampling [79] with a carefully devised sample updating strategy improves the cost significantly [111, 180], but even this approach requires an initial precomputation that multiplies C by an $n \times \mathcal{O}(r)$ Gaussian random matrix, where r is the $(\epsilon, \mathcal{T})$-HSS rank of the HSS approximation to C.[8] However, due to its displacement structure, a row interpolative decomposition for C_v^{row} can be found using fADI with a cost that depends only on m and r, not n. This is comparable in cost to proxy surface methods [35, 89, 116, 181]. A key reason this is possible is that when fADI constructs a rank r approximation of the form $C_v^{row} \approx ZW^*$, the construction of Z is decoupled from the construction of W (see Section 1.4.2). Here, Z is of size $m \times r$, and to find a one-sided interpolative decomposition of C_v^{row}, we only actually need to compute Z explicitly.

[8] In [180], this step is improved with a fast matrix-vector multiply routine for C based on its relationship to the Toeplitz matrix T.

To illustrate this, set $X = C_v^{row}$, where v is a leaf node of $\mathcal{T}$ and X is of size $m \times (n - m)$. We know from (4.5) and Theorem 10 that X satisfies $D_J X - X D_K = \widetilde{L}_J \widetilde{H}_K^*$ for an index set $(J = I_v, K = I_v^c)$. After k iterations of fADI, the approximation $X \approx X^{(k)} = ZW^*$ is constructed, where $Z \in \mathbb{C}^{m \times r}$, with $r = 2k$. As we shall see, the explicit computation of the large matrix W is not needed. The construction of Z only involves the $m \times 2$ matrix $\widetilde{L}_J$ and the $m \times m$ diagonal matrix D_J (see (1.8)), and it therefore requires only $\mathcal{O}(mr)$ flops.

An approximate row interpolative decomposition is computed using a pivoted QR decomposition of Z^*. Specifically, we find unitary $Q \in \mathbb{C}^{r \times r}$, lower triangular $R \in \mathbb{C}^{m \times r}$, and a permutation matrix P, so that $Z = PRQ$. Ideally, the permutation matrix is selected to ensure that R_a in the partition $R = [R_a \, R_b]^T$ is as well-conditioned as possible, where R_a is of size $r \times r$. It follows that

$$X \approx ZW^* = P \begin{bmatrix} \mathcal{I}_r \\ R_b R_a^{-1} \end{bmatrix} R_a Q W^* \approx \underbrace{P \begin{bmatrix} \mathcal{I}_r \\ R_b R_a^{-1} \end{bmatrix}}_{U_v} X(\widetilde{J}_v, :), \tag{4.21}$$

where $\widetilde{J}_v$ are the row indices selected by P^T. We see that (4.21) has the same interpolative structure as (4.19).

For rows at the non-leaf level, the process is similar. We work with submatrices of the form $X = C_v^{row}(\widehat{J}, :)$, as in (4.20), where $\widehat{J} = \{\widetilde{J}_{2v+1} \cup \widetilde{J}_{2v+2}\}$. Now X satisfies

$$D_{I_v(\widehat{J})} X - X D_K = \widetilde{L}_{I_v(\widehat{J})} \widetilde{H}_K^*. \tag{4.22}$$

We can again use fADI to compress X, and then use (4.21) rewrite this compression as an interpolative decomposition.

Error bounds for ADI-based interpolative decompositions. The next lemma concerns the stability of an approximate row interpolative decomposition. The error bound depends on the method used to compute the QR factorization of Z in (4.21). In particular, we must control the magnitude of the entries in $R_bR_a^{-1}$. As a matter of practicality, it is safe to use column-pivoted QR (CPQR) [65, Sec. 5.4].[9] However, bounds on $|R_bR_a^{-1}|_{jk}$ produced via CPQR account for an unlikely worst-case scenario, and are exponential in r [72]. If a strong rank-revealing QR (SRRQR) algorithm is used instead, then the magnitude of the entries in $R_bR_a^{-1}$ can be bounded by a small factor algebraic in r. The SRRQR subroutine described in [72, Alg. 4, 5] and used in [35] gives the following bound:

$$|R_bR_a^{-1}|_{jk} \leq \sqrt{r}, \quad 0 \leq j \leq m-r-1, \quad 0 \leq k \leq r-1, \tag{4.23}$$

This leads to the following lemma for ADI-based row interpolative decompositions. An analogous result holds for column interpolative decompositions.

Lemma 6. *Let X be a submatrix of a maximal size m submatrix of C, where C is as in* (4.5) *and $X \in \mathbb{C}^{m\times\widetilde{n}}$, $m \leq \widetilde{n}$. If $\operatorname{rank}_\epsilon(X) \leq r = 2k \leq m$, then there is a fADI-based approximate interpolative decomposition of X with rank $\leq r$ that satisfies*

$$\|X - U_vX(J_v,:)\|_2 \leq 4\mu_0^{-2k}\left(1+\sqrt{r+r^2(m-r)}\right)\|X\|_2, \tag{4.24}$$

where U_v is as in (4.21)*, and μ_0 is given in Theorem 10.*

Proof. Since X is submatrix of a maximal submatrix of C, it satisfies (4.22) for some index sets J, K. We apply k iterations of fADI to (4.22) to find $X \approx ZW^*$, with Z of size $m \times r$. By Corollary 5 and Theorem 10, we have that

$$\|X - ZW^*\| \leq 4\mu_0^{-2k}\|X\|_2. \tag{4.25}$$

[9] Despite famous examples where CPQR fails to produce a rank-revealing factorization, it is considered highly reliable in practice [72].

We then find $ZW^* = U_vX(J_v, :)$, $J_v \subset J$, as in (4.21), using SRRQR. It follows that

$$\|X - U_vX(J_v, :)\|_2 \leq \|X - ZW^*\|_2 + \|ZW^* - U_vX(J_v, :)\|_2.$$

According to (4.21), $ZW^* = U_vR_aQ^*W^*$, so we have that

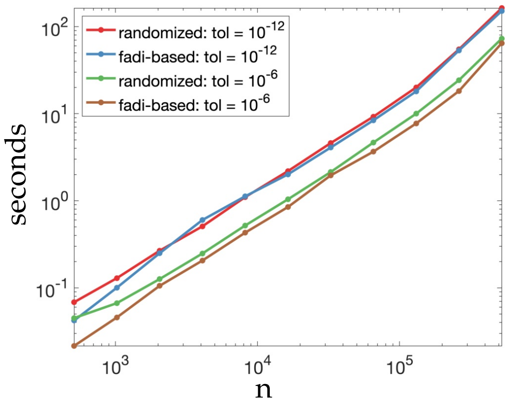
$$\|ZW^* - U_vX(J_v, :)\|_2 \leq \|U_v\|_2\|R_aQ^*W^* - X(J_v, :)\|_2 \leq \|U_v\|_2\|X - ZW^*\|_2,$$

where the last inequality follows from the fact that $R_aQ^*W^* - X(J_v, :)$ is a submatrix of $ZW^* - X$. Since SRRQR was used to construct $R_bR_a^{-1}$, we have by (4.23) that $\|U_v\|_F \leq (r + r^2(m - r))^{1/2}$. The lemma follows from (4.25). □

As with the HODLR matrix construction, one can take further advantage of special structures unique to C. For example, one can compute the $U_vB_vV_v$ factors at the leaf level for only one leaf, and then use self-similarity properties to generate all of the additional leaf factors with no additional computation (see [18]). If it is not important to retain the interpolative structure of the HSS factors, then there are even more ways to take advantage of the self-similarity structure of C [180].

4.4.5 A practical ADI-based HSS solver

In this section, we discuss our practical implementation of Algorithm 1 that uses the ADI-based interpolative decomposition method from Section 4.4.4 to construct the HSS factors comprising $\widetilde{C}$, where $\widetilde{C} \approx C$. Our Toeplitz solver is implemented in MATLAB [18], and can be executed with the single line of code `x = Toeplitz_solve(trow, tcol, b, tol)`, where `trow`, `tcol` are the first row and column of T. This finds x in $Tx = b$ with a relative error in the 2-norm that is approximately given by `tol` $= \epsilon$.



ϵ	$\|C - \widetilde{C}\|_2/\|C\|_2$	$\|x - \widehat{x}\|_2/\|x\|_2$
10^{-3}	1.887×10^{-3}	5.648×10^{-3}
10^{-6}	4.567×10^{-7}	9.110×10^{-7}
10^{-9}	3.623×10^{-12}	4.611×10^{-11}
10^{-12}	6.445×10^{-14}	3.431×10^{-13}

Figure 4.6: Left: Time in seconds required for constructing the HSS factorization of C is plotted against n, where C is size $n \times n$. We compare the ADI-based factorization described in Section 4.4.4 with the randomized sampling method described in [180]. Tests were performed for two different tolerance parameter settings. Right: The ADI-based fast solver for $Tx = b$ comes with explicit low rank approximation error bounds and can be tuned to choose the (ϵ, τ)-HSS rank adaptively to a given tolerance. The table displays the relative accuracy of $\widetilde{C}$, an approximate HSS factorization of C in (4.5), and the relative accuracy of the computed solution $\widehat{x}$, for various choices of ϵ. For these tests, Toeplitz matrices T and right-hand sides b were chosen randomly.

Since we have explicit bounds on the low rank approximation errors in our construction, our implementation is adaptive to a given tolerance ϵ. To control the error, we use the bound in Theorem 10 and determine r so that $\text{rank}_\epsilon(C_{JK}) \leq r$ for each compressed submatrix C_{JK}. Along with these bounds, we automatically compute optimal ADI shift parameters that construct approximations that achieve the bounds. This simplified strategy is designed so that $\|C - \widetilde{C}\|_2 \approx \epsilon\|C\|_2$. It works well in practice, as demonstrated in Figure 4.6 (right). One could instead devise a more elaborate scheme that guarantees (in infinite precision) that $\|C - \widetilde{C}\|_2 \leq \epsilon\|C\|_2$.

To construct $\widetilde{C}$ in Step 2 of Algorithm 1, we use standard CPQR, though one could implement the procedure with SRRQR if desired. Once the factors of $\widetilde{C}$ are known, we apply the ULV solver from [34] to solve $\widetilde{C}\widetilde{x} = \widetilde{b}$ with only $\mathcal{O}(nr)$ flops, where r is the (ϵ, τ)-HSS rank of C, computed a priori via (4.13).

Since we preserve the special interpolative structure of the factors in our HSS construction, one could use instead the ULV-like solver described in [180].

A detailed complexity analysis in Appendix A reveals that the cost for Step 2 of Algorithm 1, i.e., the construction of $\widetilde{C}$, is $\mathcal{O}(nr^2) = \mathcal{O}(n\log^2 n \log^2(1/\epsilon))$. The overall complexity of Algorithm 1 is identical to this. We note that our implementation takes advantage of the similarity structure of C at the leaf level, though this isn't necessary for achieving the $\mathcal{O}(n\log^2 n\log^2(1/\epsilon))$ complexity and is not included in our complexity analysis. Asymptotically, our complexity matches the state of the art Toeplitz solver in [180]. Numerical results in Figure 4.6 confirm that the different compression schemes also perform similarly in practice. Since our HSS factorization allows for the use of the same inversion methods to solve $\widetilde{C}\widetilde{x} = \widetilde{b}$ as in [180], the overall performances of these methods are similar.

The solver also performs comparably with [180] in terms of stability. For well–conditioned matrices, the ADI-based method is accurate (see Figure 4.6, right). In Figure 4.7, we compare the relative error $\|\widehat{x} - x\|_2/\|x\|_2$ achieved by the two methods for solving $Tx = b$ with an ill-conditioned choice of T. In the experiment, we choose T as the KMS Toeplitz matrix generated from a vector $t = (t_{-n+1}, \cdots, t_{n-1})$ with entries $t_k = \phi^{|k|}$ [167], where $\phi = 1 - 10^{-j}$. T is well–conditioned when $j = 1$, but as j increases, T becomes extremely ill-conditioned. When $j = 10$ and $n = 4096$, $\|T\|_2\|T^{-1}\|_2 \approx 10^{-13}$. We set the tolerance parameter to $\epsilon = 10^{-11}$ in all cases.

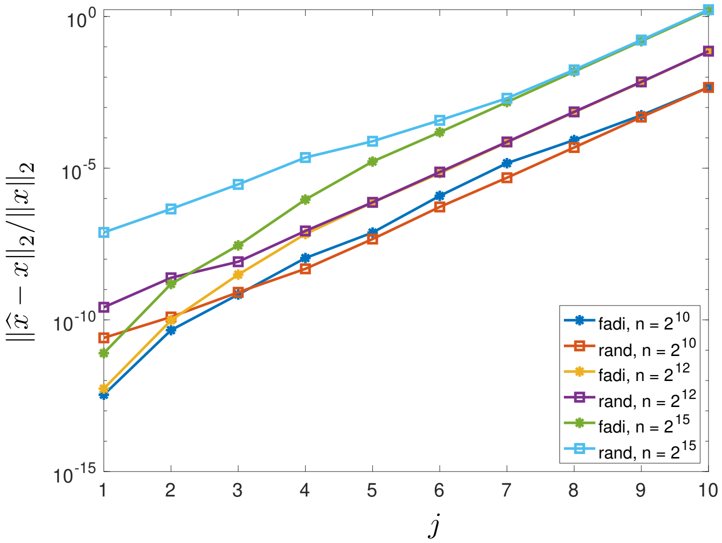


Figure 4.7: The relative error of the computed solution $\widehat{x}$ to $Tx = b$, where T is the KMS matrix [167] of size $n \times n$, is shown for various choices of n. The error is plotted against j, where $\phi = 1 - 10^{-j}$ and the entries of T are of the form $t_k = \phi^{|k|}$. As j increases, T becomes increasingly ill–conditioned. The accuracy of the fADI-based solver is compared with the accuracy of the solver from [180].

4.5 Related linear systems

Our approach can be extended and applied to related linear systems with a cost that remains the same with respect to r and depends linearly on the displacement rank ρ. As an example, consider the Vandermonde matrix with entries $V_{jk} = \gamma_j^{k-1}$, $1 \leq j, k \leq n$. The numbers $(\gamma_1, \ldots, \gamma_n)$ are referred to as the “nodes” of V. If $N = \mathrm{diag}(\gamma_1, \ldots, \gamma_n)$, then

$$NV - VS = ac^T, \tag{4.26}$$

where S is as in (4.2) and $a, c \in \mathbb{C}^{n\times 1}$. One can use (4.26) to show that when the nodes are real–valued, V is a low rank matrix with rapidly decaying singular values [19]. Bounds on the ϵ-rank of V can also be derived when the nodes lie in a set contained either inside or outside the unit disk (see Section 3.5.1). When the nodes coincide with a shifted set of n roots of unity, then V is a scaled discrete Fourier transform matrix [37], and when they lie elsewhere on the unit circle,

V is called a nonuniform discrete Fourier transform matrix. In configurations where the nodes are not close to an equally spaced grid, V can be ill-conditioned and it may be desirable to have a direct solver for $Vx = b$ with a cost independent from the condition number of V.

Since N is already diagonal, we diagonalize the Sylvester equation in (4.26) by right-multiplying the equation by $\mathcal{F}$ from (4.4). Setting $\mathcal{C} = V\mathcal{F}^*$, we have that

$$N\mathcal{C} - \mathcal{C}D = a\widetilde{c}^*, \tag{4.27}$$

where $\widetilde{c} = \mathcal{F}c$ and $D = \mathcal{F}S\mathcal{F}^* = \operatorname{diag}(\omega^2, \ldots, \omega^{2n})$. The matrix $\mathcal{C}$ is Cauchy-like, and if one assumes that the points $\{\lambda_j\}_{j=1}^n$, where $\gamma_j = e^{2\pi i\lambda_j/n}$, are restricted so that for each j, $|\lambda_j - j| \leq \beta < 1$, then arguments similar to those given for Theorem 10 show that $\mathcal{C}$ is well-approximated by an HSS matrix. In particular, the same argument shows that if $\mathcal{C}_{JK}$ is a maximal submatrix of $\mathcal{C}$, then

$$\operatorname{rank}_\epsilon(\mathcal{C}_{JK}) \leq \left\lceil \frac{2}{\pi^2} \log\left(\frac{2n}{1-\beta}\right) \log\left(\frac{4}{\epsilon}\right) \right\rceil.$$

If we drop the assumption that $|\lambda_j - j| \leq \beta < 1$ and allow the nodes γ_j to occur anywhere (remaining distinct from one another), it is no longer guaranteed that V has low rank off-diagonal submatrices. With the nodes irregularly clustered, it is possible, for example, that the arcs associated with the indices of a HODLR submatrix of $\mathcal{C}$ are not disjoint from one another. However, V still contains compressible submatrices. We must instead use a more general partitioning scheme based on a different tree structure (e.g., a quadtree) and an admissibility criterion [16, 76]. Fortunately, admissibility is easy to check. Given a submatrix $\mathcal{C}_{JK}$, we associate J and K with arcs $\mathcal{A}_J$, $\mathcal{A}_K$ on the unit circle, where $\{\gamma_j\}_{j\in J} \subset \mathcal{A}_J$, and $\{\omega^{2k}\}_{k\in K} \subset \mathcal{A}_K$. In the simplest setup, we choose a "separation parameter", $0 < \alpha < 1/2n$, and say that $\mathcal{C}_{JK}$ passes the admissibility test if $\operatorname{dist}(\mathcal{A}_J, \mathcal{A}_K) > \alpha$, where

$$\text{dist}(\mathcal{A}_J, \mathcal{A}_K) = \min\{|s_j - s_k|,\ s_j \in \mathcal{A}_J, s_k \in \mathcal{A}_K\}.$$

We may then derive a theorem similar to Theorem 10 where the numerical rank of $\mathcal{C}_{JK}$ is bounded in terms of α and n. Future work will consider the development of such methods for the rectangular and 2D cases.

Another example of a linear system $Xy = b$ where these methods can be applied is when $X = T + R$, where T is Toeplitz and R is Hankel. As shown in [100], one has that

$$\text{rank}(\mathcal{Y}_{0,0}X - X\mathcal{Y}_{1,1}) \leq 4, \quad \mathcal{Y}_{v,\delta} = \begin{bmatrix} v & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & 1 \\ 0 & \cdots & 0 & 1 & \delta \end{bmatrix}.$$

The matrices $\mathcal{Y}_{0,0}$ and $\mathcal{Y}_{1,1}$ can be diagonalized with discrete sine and discrete cosine transforms, respectively, in only $\mathcal{O}(n \log n)$ operations. Using these transformations, one may relate X to a Cauchy-like matrix $\widetilde{X}$ satisfying

$$D_S\widetilde{X} - \widetilde{X}D_C = \tilde{L}\tilde{H}^*$$

for some generators $\tilde{L}, \tilde{H}$, each of rank ≤ 4. Here,

$$D_C = 2\text{diag}\left(1, \cos\left(\frac{\pi}{n}\right), \cdots, \cos\left(\frac{(n-1)\pi}{n}\right)\right), \tag{4.28}$$

$$D_S = 2\text{diag}\left(1, \cos\left(\frac{\pi}{n+1}\right), \cdots, \cos\left(\frac{n\pi}{n+1}\right)\right), \tag{4.29}$$

so the spectra $\lambda(D_S)$ and $\lambda(D_C)$ are interlaced on the real line. An argument using Theorem 2 can be applied to bound the ϵ–ranks for various off-diagonal submatrices of $\widetilde{X}$, and fADI-based methods can be applied to construct rank-structured approximations to $\widetilde{X}$. More examples of matrices with fast diagonalizable displacement structures can be found in [100].

CHAPTER 5

DATA-DRIVEN RATIONAL FUNCTION APPROXIMATION

The previous three chapters in this dissertation focused on one particularly important rational approximation problem that arises in numerical linear algebra. The original solution to this problem was developed in the 1800s using classical techniques from analysis [3, 182], long before the computational questions that it pertains to had ever been posed. Advancements on these computational questions have often depended on knowledge of the classical solution. In this chapter,[1] we consider a very different type of approximation problem. We seek to reliably and automatically generate rational approximations to functions, even though we have no a priori knowledge about the types, locations, or numbers of singularities they possess. In particular, we develop methods that apply rational approximation in order to reconstruct signals from samples that may be corrupted by noise. Our construction algorithms are complemented by the development of a new software system for computing adaptively with 1D trigonometric rational functions and their Fourier transforms.

5.1 Introduction

Recovering functions from noisy, incomplete, or corrupted samples is a ubiquitous task in signal and data processing [99]. Here, we recover underlying signals that contain impulses, shocks, or other algebraic singularities that can cause traditional Fourier-based methods to underperform or fail. Examples of these types of signals include sensor monitoring and event detection tasks in

[1] This chapter is related to a manuscript [40] authored by Anil Damle, Alex Townsend, and me. I am the lead author of the manuscript and developed the ideas and examples therein. I also developed and wrote the software package based on these ideas.

Barycentric form	Exponential sums
Differentiation (closed-form formula) [24]	Filtering and recompression [81]
Imputing missing data [119]	Pole symmetry preservation [28]
Stable evaluation [8, 87]	Robustness to noise [129, 135]
Rootfinding [119], identifying extrema	convolution [135], cross-correlations

Table 5.1: Operations that are efficient and robust in the two representations. By having both representations and toggling between them, we can efficiently compute a range of operations in signal processing.

seismology and oceanography [36, 97, 80, 110, 152], biomedical signal processing [54, 55, 62, 102], and time evolution along rays in nonsmooth media [32, 138]. We present a novel computing framework based on data-driven approximation with two complementary representations: (1) barycentric trigonometric rational approximations and (2) their Fourier transforms, which take the form of short sums of complex exponentials. Toggling between these two representations lets us overcome computational and data-related challenges.

Several families of functions, including rationals, wavelets, and radial basis functions, are well-suited for resolving sharp features in data and modeling phenomena with slow-decaying spectral content [43, 163, 169]. However, methods that employ these functions often require the a priori selection of shape parameters [139], mother/father wavelets [43], initial pole configurations [74], or special rational basis functions [55]. One must carefully select parameters to avoid numerical instability and computational inefficiency. In contrast to this, we introduce flexible, data-driven, general-purpose software tools that can be applied without special knowledge about the locations or types of singularities in the signal. Our methods construct trigonometric rational representations of signals, and we develop a collection of algorithms for computing adaptively and efficiently with them. Our approach combines adaptations of two primary approximation methods, the AAA algorithm for rational approximation [119] and the Fourier inversion method [28]. Both of these methods can efficiently and automatically construct near-optimal rational approximations to functions without tuning parameters, and we develop a framework that unifies the representations they construct. The result is an automatic rational approximation method that enjoys two main advantages: (1) it is more robust to various forms of corruption than either method alone, and (2) taken collectively, the two representations are efficient for performing a range of fundamental post-processing operations (see Table 5.1).

5.1.1 The approximation problem

Let $f : [0, 1) \to \mathbb{R}$ be an unknown continuous periodic function of bounded variation, and suppose that for some integer N, we observe $2N + 1$ noisy samples of f at $T := \{x_j\}_{j=0}^{2N}$, i.e., $y_j = f(x_j) + s_j$ for $0 \leq j \leq 2N$. Here, s_j can be: (i) additive white Gaussian noise (i.i.d. normally distributed), (ii) popcorn noise (sparsely corrupted or arbitrarily large errors), or (iii) bounded deterministic errors. Throughout, we assume $\|f\|_\infty = 1$, where $\|\cdot\|_\infty$ is the infinity norm on $[0, 1)$, and that the mean value of f over $[0, 1)$ is 0. Our central approximation problem is to fit a special class of trigonometric rational functions (see Section 5.2) to the $2N + 1$ samples to construct r_m, a type $(m-1, m)$ trigonometric rational where m is selected adaptively so that the sampling error satisfies $\max_{x_j \in T} |f(x_j) - r_m(x_j)| \leq \epsilon$, where $0 < \epsilon < 1$ is a tolerance parameter.

In practice, samples of f are often available under far from ideal circumstances. For example, T may consist of poorly distributed (i.e., not equally-spaced) points due to missing or corrupted data, N may be too small to adequately resolve features of interest, or frequencies of interest may be cut off or otherwise distorted during observations. The approximation problem may also be ill-conditioned, so that regularization is needed to define and numerically construct a meaningful solution.

5.1.2 Software

Once a rational approximant is constructed, we want to reliably compute with it. Answering the following questions has shaped the overall development of our software:

1. *Which applications are of importance?* Our method is a general-purpose approach for working with periodic univariate signals. It works with noisy samples, and requires no a priori knowledge about features of the signal, such as the locations or types of singularities. The methods are designed to be flexible enough for use in a range of applications that involve the detection and identification of events (e.g., ECG and geophone monitoring tasks, engineering and financial applications where change-point detection is important, and in the analysis of some dynamical systems, such as periodic contagion modeling).
2. *What properties should our approximants have?* Like f, we want approximants to be periodic, real-valued, and continuous on $[0, 1)$. It is for this reason that we employ trigonometric rational functions, which are the periodic analogue of rational functions.
3. *What form of approximant should we use?* Trigonometric rationals can be expressed in many forms, but not all of these forms are numerically stable.

For stable evaluation and rootfinding, we use the barycentric formula [26]. For efficient recompression and operations carried out in Fourier space, we represent the Fourier transforms of our trigonometric rational approximants using short sums of weighted complex exponentials (see Lemma 7).

4. *Which tools should we provide?* We provide a basic set of computational tools. This includes simple algebraic operations (addition, products), calculus-based operations (integration, differentiation), and tools for filtering, (de)convolving, rootfinding/polefinding. Whenever possible, we automatically recompress representations as trigonometric rationals/exponential sums to maintain efficiency. These tools can be combined to perform more complicated tasks.

Accompanying this work is the open-source code REfit [40], which is written in MATLAB and uses two classes called `rfun` and `efun`. Our software is largely inspired by the Chebfun software package [45, 177]. An `rfun` object stores a representation of f as a barycentric trigonometric rational function. An `efun` object stores a representation of the Fourier transform of f as a weighted sum of complex exponentials. After an `rfun` or `efun` object is constructed, a function can be manipulated and analyzed through the operations implemented in the package (see Table 5.2). The commands are overloaded so that they can be applied to either type of object, and binary operators can be used between objects of different type.

Table 5.2: A selection of REfit commands.

command	Operation
`+,−,.*,./`	basic arithmetic
`diff(·),cumsum(·)`	differentiation, indefinite integration
`conv(·)`	convolution
`corr(·,·)`	cross-correlation

The rest of this chapter is organized as follows: In Section 5.2 we begin by briefly reviewing trigonometric rational functions and the barycentric form. We then introduce the trigonometric variant of the AAA algorithm that we use to construct barycentric trigonometric interpolants (Section 5.2.3). In Section 5.2.4, we apply the regularized Prony's method (RPM) to construct approximations in Fourier space. In Section 5.3, we introduce stable Fourier and inverse Fourier transform methods for moving between representations in the time and frequency domains. Examples and descriptions of the algorithms for computing with these representations can be found in Sections 5.4 and 5.5, respectively.

5.2 Trigonometric rational functions and their Fourier transforms

We begin with a review of trigonometric rationals and the two representations that we use: (1) the barycentric form in the time domain, and (2) sums of exponentials in Fourier space.

The trigonometric rationals are the periodic analogue of rational functions [86, 95]. A trigonometric rational of period 1 is the quotient of two trigonometric polynomials of period 1, i.e., a function of the form

$$r(x) = \frac{p_\ell(x)}{q_m(x)} = \frac{\sum_{j=-\ell}^{\ell} a_j e^{2\pi i j x}}{\sum_{j=-m}^{m} b_j e^{2\pi i j x}}, \quad x \in [0, 1). \tag{5.1}$$

We call r a type (ℓ, m) trigonometric rational function. We follow the convention that unless stated explicitly, a type (ℓ, m) function is in reduced form, meaning that p_ℓ and q_m have no zeros in common. We restrict our interest to the family of period 1 trigonometric rationals r_m of type $(m-1, m)$ that are real-valued and

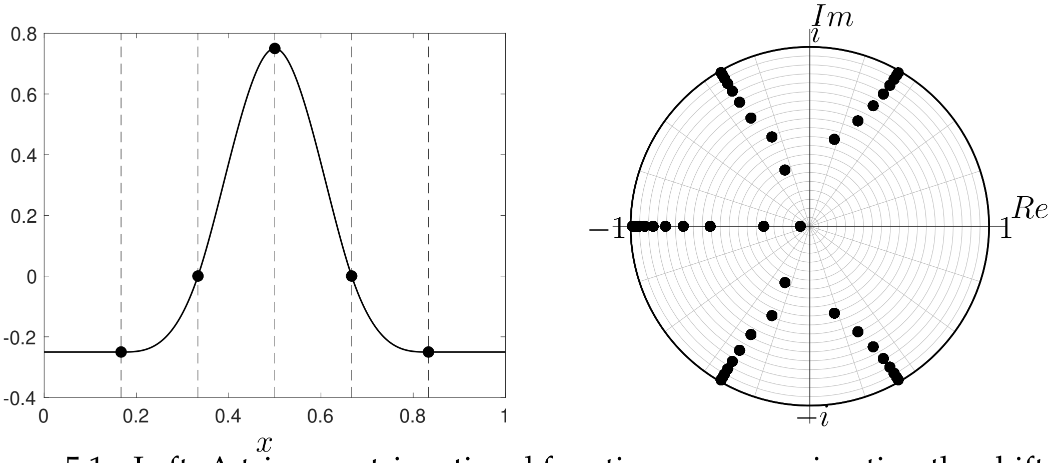


Figure 5.1: Left: A trignometric rational function r_m approximating the shifted and scaled cubic B-spline on the interval $[0, 1)$ is constructed and plotted. The knots of the spline occur at the dotted vertical lines. Here, $m = 44$. Away from the singularities, the absolute approximation error is on the order of 10^{-11}. Approximate knot locations (black dots) are automatically computed using the poles of the rational $\tilde{r}_m(z) = r_m(x)$, where $z = e^{2\pi i x}$(see (5.3)). Right: The poles of $\tilde{r}_m$ of magnitude ≤ 1 are plotted in the unit disk in the complex plane. They cluster toward the points $e^{2\pi i x_k}$ on the unit circle, where each x_k is a knot in the spline.

continuous on $[0, 1)$. Furthermore, we assume that the roots of the denominator q_m are simple. Under these assumptions, if η_j is a root of q_m, then so is its complex conjugate $\overline{\eta}_j$, as well as $\eta_j \pm K$, where K is any integer. We say that a pole of r_m is any root, η_j, of q_m such that $0 \leq Re(\eta_j) < 1$. In the same way, any root of p_{m-1} with a real part in the interval $[0, 1)$ is called a zero of r_m.

5.2.1 Why trigonometric rationals?

Three key properties of this family of functions make them ideal for our setting. First, like standard rational functions, they are especially effective at resolving singularities. For example, for particular choices of f, such as $f(x) = |x - 1/2| - 1/4$, it is known that trigonometric rationals of type (m, m) can converge to f at a root-exponential rate with respect to m [163]. Second, these functions can be represented efficiently in Fourier space. We make extensive use of the following fact:

Lemma 7. *Let $r_m(x)$ be a type $(m-1, m)$ nonzero trigonometric rational function that is real-valued and continuous on $[0, 1)$ with exactly $2m$ simple poles. Let $\{\eta_j\}_{j=1}^m$ denote the collection of poles with $Im(\eta_j) > 0$. If the Fourier coefficients of r_m are given by $\{(\hat{r}_m)_k\}_{k=-\infty}^{\infty}$, then there exist ω_j such that*

$$(\hat{r}_m)_k = R_m(k) := \begin{cases} \sum_{j=1}^m \omega_j e^{\alpha_j k}, & k \geq 0, \\ \sum_{j=1}^m \overline{\omega}_j e^{-\overline{\alpha}_j k}, & k < 0, \end{cases} \tag{5.2}$$

where $\alpha_j = 2\pi i \eta_j$. Here, $\overline{\omega}_j$ is the complex conjugate of ω_j.

Proof. See [155, Ch. 4]. □

Lemma 7 shows that the Fourier series of r_m can be represented as sum of m weighted decaying exponentials. We say that $\mathcal{F}(r_m) = R_m$ is the Fourier transform of r_m, and similarly, $\mathcal{F}^{-1}(R_m) = r_m$ is the inverse Fourier transform of R_m.

Third, the properties of trigonometric rationals can be used for feature detection. For example, in Figure 5.1, we plot the poles $z_j = e^{2\pi i \eta_j}$, $Im(\eta_j) > 0$, of the rational function

$$\tilde{r}_m(z) = \frac{z^m p_{m-1}(z)}{z^m q_m(z)}, \quad z = e^{2\pi i x}, \tag{5.3}$$

where $r_m(x) = p_{m-1}(x)/q_m(x)$ is an approximation to a shifted and scaled uniform cubic B-spline [41] f with five equally-spaced knots at $x = \{1/6, \ldots, 5/6\}$. The knots are not easily identifiable in a plot of f, but the poles of $\tilde{r}_m$, which are chosen adaptively via Algorithm 3, cluster toward the singularities, revealing their locations. Using the five poles in $\{z_j\}_{j=1}^m$ with the largest magnitudes, estimates of the knot locations are given by $\tilde{x}_k = \arg(z_k)/2\pi$ for $1 \leq k \leq 5$. The radial coordinates of the poles in Figure 5.1 also encode information about f in the frequency domain. The Fourier coefficient $\hat{f}_k$ is well-approximated by $(\hat{r}_m)_k = R_m(k)$, where R_m is as in (5.2). The terms in R_m with $|z_j| \leq \tilde{\epsilon} \ll 1$ have negligible influence when k is small, so they capture aspects of the signal that are only observable at low frequencies.

5.2.2 Barycentric trigonometric rational functions

Large numerical errors can be incurred when evaluating trigonometric rationals that are numerically constructed using (5.1) directly [51, 163]. Another natural way to represent r_m is in a pole-residue format with respect to the pairs of poles $(z_j = e^{2\pi i \eta_j}, 1/\overline{z}_j)$ of the rational function $\tilde{r}_m$ in (5.3). Evaluation using this format is often reliable in practice, but it can potentially result in catastrophic cancellation if the evaluation point is too near to the poles. In the aperiodic setting, the AAA algorithm [119] safeguards against such instabilities by using barycentric rational interpolants with backward-stable evaluation on the interval of approximation [8, 87].

The trigonometric analogue of the barycentric rational interpolants is described in [86], and more recently, in [9, 95]. For type $(m-1, m)$ rationals on $[0, 1)$, they take the following form:

$$r_m^{\gamma,t}(x) = \frac{n_{m-1}(x)}{d_m(x)} = \frac{\sum_{j=1}^{2m} \gamma_j f_j \cot\big(\pi(x - t_j)\big)}{\sum_{j=1}^{2m} \gamma_j \cot\big(\pi(x - t_j)\big)}, \qquad \sum_{j=1}^{2m} \gamma_j f_j = 0, \tag{5.4}$$

where the interpolating points $t = \{t_1, \ldots, t_{2m}\}$, which are always assumed to be distinct, are called barycentric nodes, $\gamma = \{\gamma_1, \ldots, \gamma_{2m}\}$ are called barycentric weights, and $f_j = f(t_j)$. It is easily shown that $r_m^{\gamma,t}(t_j) = f_j$ for any choice γ

with all nonzero entries [26], but it is not obvious from (5.4) that $r_m^{\gamma,t}$ is in fact a type $(m-1,m)$ trigonometric rational. This can be seen by using the trigonometric polynomial $\ell_t(x) = \prod_{j=1}^{2m} \sin(\pi(x-t_j))$, and rewriting $r_m^{\gamma,t}$ as $\ell_t n_{m-1}/\ell_t d_m$ (see [86]). The condition $\sum_{j=1}^{2m} \gamma_j f_j = 0$ enforces that the numerator is a trigonometric polynomial of degree $m-1$, rather than m. It is not clear from (5.4) where the poles of $r_m^{\gamma,t}$ lie, and in particular, whether they lie off $[0,1)$ (see Section 5.2.3). Stability properties (and other advantages) of the barycentric form were popularized in the context of polynomial interpolation [25], where the weights γ are always chosen so that $d_m(x) = 1$. Thorough analysis can be found in [8, 87], with additional relevant discussions for the rational case in [51, 119].

5.2.3 Approximations in time

In the noiseless setting, the trigonometric barycentric rational $r_m^{\gamma,t}$ can be directly constructed from samples of f using an analogue of the AAA algorithm (pronyAAA) that we introduce here.[2] This supplies a fast, automated way to construct near-optimal trigonometric rational representations of signals. Other highly effective rational approximation algorithms include the RKFIT algorithm [23] and vector fitting [74], though these have not been explicitly adapted to the periodic setting. These methods are not suited to our needs as they require and can be sensitive to a set of initialization parameters (e.g., guesses of the number and location of the poles).

A major advantage of AAA-type methods over other approximation

[2] Independently, AAA for type (m,m) trigonometric rationals has been developed with applications related to conformal mapping [9]. A closely related idea where rational approximation is used in Fourier space (and so aperiodic rationals are used) and related to exponential sums in value space was developed in [44] concurrently with and independently from our work.

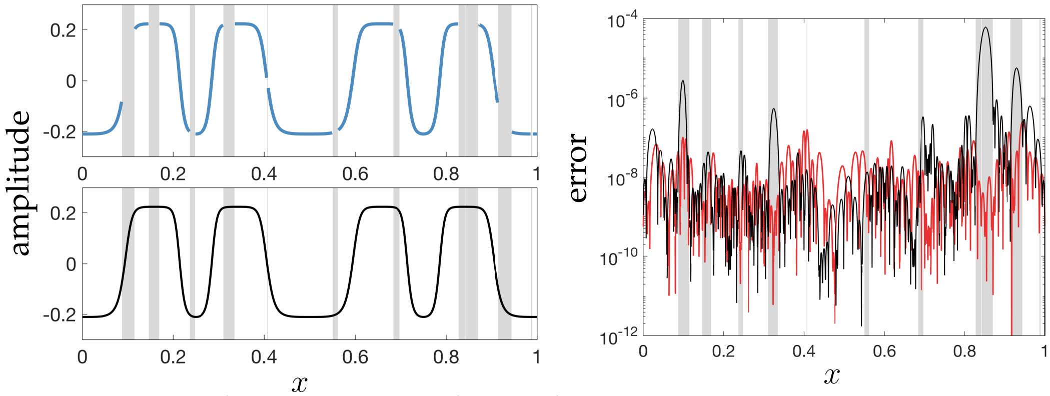


Figure 5.2: Left: 3000 samples from a function f are taken on an equally spaced grid from $[0, 1)$. However, the data inside the grey regions are corrupted and thus deleted from the sample (a total of 609 observations are missing). The available data are plotted (blue, upper panel) and used to construct a type $(66, 68)$ trigonometric barycentric rational interpolant r_{68} via pronyAAA. The resulting approximant, evaluated on an equally spaced grid and plotted (black, lower panel), imputes the missing data in the grey regions. Right: The absolute error $|f(x) - r_{68}(x)|$ is plotted in black on a logarithmic scale against the original grid of sampled points. For comparison, the absolute error when pronyAAA is applied to an uncorrupted sample of f with no missing data is also displayed (red).

schemes is that they can be blithely applied to samples from non-uniform grids, and can even be used for recovering functions defined on disjoint sets of support [119]. In particular, these methods are robust to missing samples. In Figure 5.2, we use pronyAAA to recover a function from data that has been deleted in several corrupted regions.

Constructing barycentric trigonometric interpolants

Our construction algorithm is a straightforward extension of the standard AAA algorithm, with differences being the use of trigonometric basis functions, the restriction that the numerator is always of degree m–1, and the requirement that the constructed interpolant has an even number of interpolating points.[3] These

[3] This is so that $r_m^{\gamma,t}$ has an even number of simple poles that occur in approximate conjugate pairs.

restrictions are so that the constructed approximant corresponds to a length m exponential sum in Fourier space. As with standard AAA, pronyAAA employs a greedy residual minimization method to adaptively select interpolating points. In this way, it builds up a trigonometric rational interpolant one pair of support points at a time, updating the weights γ at each step by solving a least squares problem. We briefly discuss the process here, and refer to [119] for more details.

Let $\{f_0, \ldots, f_{2N}\}$ be the samples of a function f we seek to approximate, where $T = \{x_j\}_{j=0}^{2N} \subset [0,1)$ are the sample locations and $f(x_j) = f_j$. We first describe how the barycentric weights are updated at each iteration. Suppose that at the mth iteration, the barycentric support points for $r_m^{\gamma,t}(x)$ in (5.4) are $t = \{t_1, \ldots, t_{2m}\} \subset T$. Let $\widetilde{T} = T \setminus \{t_1, \ldots, t_{2m}\}$. We must now choose $\{\gamma_j\}_{j=1}^{2m}$. Defining the vector $\gamma = (\gamma_1, \ldots, \gamma_{2m})^T$ and recalling that n_{m-1} and d_m are such that $r_m^{\gamma,t} = n_{m-1}/d_m$ as in (5.4), we select the weight vector γ as the solution to the following constrained optimization problem:

$$\min_{\gamma \in \mathbb{C}} \sum_{x_j \in \widetilde{T}} \big(f(x_j)d_m(x_j) - n_{m-1}(x_j)\big)^2, \quad \text{s.t.} \sum_{j=1}^{2m} f(t_j)\gamma_j = 0, \quad \|\gamma\|_2 = 1. \tag{5.5}$$

The constraint is satisfied if we select a vector of the form $\gamma = Q\tilde{\gamma}$, $\|\tilde{\gamma}\|_2 = 1$, where Q is a $2m \times (2m-1)$ matrix with orthonormal columns that span the null space of the vector $(f(t_1), \ldots, f(t_{2m}))$. Since for each $x_j \in \widetilde{T}$,

$$f(x_j)d_m(x_j) - n_{m-1}(x_j) = \sum_{\ell=1}^{2m} \big(f(x_j) - f(t_\ell)\big)\gamma_\ell \cot(\pi x_j - \pi t_\ell), \tag{5.6}$$

we see that $\tilde{\gamma}$ is given by the last right singular vector of the matrix CQ, where $C \in \mathbb{R}^{|\widetilde{T}| \times 2m}$ has entries $C_{j\ell} = (f(x_j) - f(t_\ell))\cot(\pi x_j - \pi t_\ell)$, where each x_j is a unique member of $\widetilde{T}$.

Once γ is computed, we have constructed $r_m^{\gamma,t}$ and the iteration is complete.

If the error $\max_{x_j \in \widetilde{T}} |f(x_j) - r_k^{\gamma,t}(x_j)|$ is not sufficiently small, we begin the next iteration by choosing two additional interpolating points from $\widetilde{T}$. It is not clear how to efficiently choose two points in a single step, so we introduce an interim step. The first point is chosen as $t_{2m+1} = \max_{x_j \in \widetilde{T}} |f(x_j) - r_{2m}^{\gamma,t}(x_j)|$, and then t and $\widetilde{T}$ are updated appropriately. To pick the second point, we construct an interim trigonometric interpolant $r_{m+1/2}^{\gamma,t}$ with an odd number of interpolating points, using the basis functions $\csc(\pi(x - t_j))$ instead of $\cot(\pi(x - t_j))$. The different basis functions ensure that $r_{k+1/2}^{\gamma,t}$ is a trigonometric rational: see [86]. The weights in $r_{m+1/2}^{\gamma,t}$ are updated by solving a problem similar to (5.5), though the constraint differs due to the different basis functions [86]. The second interpolating point is then chosen as $t_{2m+2} = \max_{x_j \in \widetilde{T}} |f(x_j) - r_{m+1/2}^{\gamma,t}(x_j)|$.

Spurious poles

As with AAA, nothing in the pronyAAA algorithm directly controls where the poles of $r_m^{\gamma,t}$ occur. The advantage of this approach is that unlike fixed–pole approximation methods, the pole locations are adaptively determined, and automatically cluster in patterns that are highly effective for resolving singularities [166]. One drawback to this approach is that the conjugate-pair symmetry of the poles is not explicitly enforced. A more problematic issue is that so-called "spurious poles" may appear on the interval of approximation.

Spurious poles are poles that are undesirable or unnecessary [26, 119]. They can arise as artifacts related to numerical error, but they can also appear within perfectly mathematically valid solutions to an interpolation problem constrained by (5.5). For example, a spurious pole may co-occur with a nearby zero that effectively cancels out its influence except within a small in-

Algorithm 2 The pronyAAA algorithm.

```
Input: tolerance parameter $\epsilon$, sample locations $T = \{x_0, \dots, x_{2N}\}$, samples $\{f(x_0), \dots, f(x_{2N})\}$.
Output: barycentric support points $t$ and weights $\gamma$ defining $r_m^{\gamma,t}$ in (5.4).
1. Set $m = 1$, $t = \{t_1 = \max_{x_j \in T} |f(x_j)|, t_2 = \max_{x_j \in T \setminus t_1} |f(x_j)|\}$, $\widetilde{T} = T \setminus t$.
2. Solve for $\gamma$ as in (5.6) to define $r_1^{\gamma,t}$. Set $err = \max_{x_j \in \widetilde{T}} |f(x_j) - r_1(x_j)|$.
3. While $err > \epsilon$
      for $\ell = 1, 2$
            (i) Set $t_{2m+\ell} = \max_{x_j \in \tilde{T}} |f(x_j) - r^{\gamma,t}_{m+(\ell-1)/2}|$.
            (ii) $t \leftarrow \{t_1, \dots, t_{2m+\ell}\}$,  $\widetilde{T} \leftarrow \widetilde{T} \setminus t_{2m+\ell}$.
            (iii) Update $\gamma = (\gamma_1, \dots, \gamma_{2m+\ell})^T$ by solving a constrained
            optimization problem on $\widetilde{T}$ (see (5.6) and discussion).
      end for
      $err \leftarrow \max_{x_j \in \widetilde{T}} |f(x_j) - r^{\gamma,t}_{m+1}(x_j)|$.
      $m \leftarrow m + 1$.
   end while
4. Compute poles and residues (see Section 5.5.7).
5. If there are spurious poles, apply cleanup routine (see Section 5.2.3).
```

terval I, where $I \subset [x_j, x_{j+1}]$ for some neighboring sample locations $x_j < x_{j+1}$, so that for $x \notin I$ it holds that $|r_m^{\gamma,t}(x) - f(x)| < \epsilon$. This configuration is a perfectly acceptable way to solve (5.5), but it leads to a solution where $\|r_m^{\gamma,t} - f\|_\infty$ is unbounded.

Spurious poles are eliminated in the standard AAA algorithm with an added cleanup routine [119], which we adapt to our setting. Pairs of spurious poles are detected via their small residues. The barycentric nodes nearest to the spurious poles are eliminated. This forces the degree of the interpolant to drop and requires the barycentric weights to be recomputed, which changes the number and location of the poles. When poles cannot be eliminated without destroying the approximation, it is often because the function being interpolated is not well-approximated by low to moderate degree type $(m-1, m)$ trigonometric rationals. We observe this, for example, when pronyAAA is applied to samples perturbed by additive Gaussian noise. An example is described in Section 5.4.2 that involves the reconstruction of ECG data from 645 samples. The direct application of pronyAAA to the noisy data results in a type $(199, 200)$ trigonometric rational interpolant, with 62 spurious poles appearing on the interval of approximation. However, by combining pronyAAA with the construction method we describe in the next section, we are able to instead construct a type $(69, 70)$ trigonometric barycentric rational representation of the signal with no spurious poles (see Figure 5.6).

5.2.4 Approximations in Fourier space

The pronyAAA algorithm allows us to directly construct trigonometric rational representations to signals, but the exclusive use of pronyAAA is inadequate. For example, it cannot be applied in the presence of Gaussian noise, and the barycentric form is not conducive to efficient recompression techniques. These issues can be remedied by instead representing f in Fourier space using the exponential sums in (5.2).

To construct the sums, we require the Fourier coefficients of f. We use the fast Fourier transform (FFT) to compute the coefficients $v = (\hat{f}_0, \dots, \hat{f}_N)^T$ associated with samples $\{f(x_j)\}_{j=0}^{2N}$, where $x_j = j/(2N+1)$. When f is not a bandlimited function (or has a bandlimit $> N$), this process introduces error into the Fourier coefficients and motivates the use of the following notion:

Definition 3 (ϵ-resolution)**.** *For $0 < \epsilon < 1$, the ϵ-resolution of f is the smallest non-negative integer N_ϵ such that*

$$\|f - f_{trunc}\|_\infty \leq \epsilon,$$

where f_{trunc} is the best $\mathcal{L}_{\infty[0,1)}$ projection of f onto the functions of bandlimit N_ϵ.

For bandlimited functions, N_0 is the bandlimit of f. The ϵ-resolution for non-bandlimited functions can be understood in relation to the smoothness of f and its region of analyticity in the complex plane [163]. In our setting, the assumption is that f contains algebraic singularities and N_ϵ is large. However, when f is well-approximated by a type $(m-1, m)$ trigonometric rational, Lemma 7 indicates that $\mathcal{F}(f)$ can be represented with far fewer degrees of freedom via the exponential sum R_m. One way to find R_m is by fitting the nonlinear model $R_m(k) = \sum_{j=1}^{m} \omega_j e^{\alpha_j k}$ to the Fourier coefficients $\hat{f}_k$, $0 \leq k \leq N$. We emphasize that in the general setting, m is unknown and must be determined adaptively.

Regularized Prony's method

To construct R_m, we follow an idea in [28] and use the regularized version of Prony's method (RPM) from [27]. Variants of this method go by many names across various disciplines, and we refer to [135] for an overview. The problem of finding R_m can be recast as a structured low rank approximation problem involving Hankel matrices. This connection can be understood via the following lemma, a version of which was first proven by Prony in 1795 [136].

Lemma 8. *Let N be an even integer.*[4] *Let R_m be as in Lemma 7 and let H_{R_m} be an $(N/2+1) \times (N/2+1)$ Hankel matrix with entries $(H_{R_m})_{k\ell} = R_m(k+\ell)$, $0 \leq k, \ell \leq N/2$, where $N \geq 2m$. Let $\mathcal{P}_m(z) = \sum_{j=0}^{m} c_j z^j$ be a polynomial with roots $z_j = e^{\alpha_j}$, $1 \leq j \leq m$. Then,* $\mathrm{rank}(H_{R_m}) = m$, *and the kernel of H_{R_m} is spanned by $\{c, Sc, \ldots, S^{N/2-m}c\}$, where $c = (c_0, \ldots, c_m, 0, \ldots, 0)^T$ and S is the forward shift matrix.*

[4] The statement and proof can be adjusted to account for odd N [27].

Proof. See [Lem. 2.1][134]. □

In our case, we seek $R_m \approx \mathcal{F}^{-1}(f)$. Lemma 8 indicates that this is equivalent to finding a Hankel matrix H_{R_m} of rank m so that $H_{R_m} \approx H_v$, where $(H_v)_{jk} = \hat{f}_{j+k}$, $0 \leq j, k \leq N/2$. Moreover, it shows that the complex exponentials in (5.2) are the roots of a special polynomial, often referred to as Prony's polynomial [135], whose coefficients form a vector in the null space of H_{R_m}. The regularized Prony's method (RPM), described in pseudocode in Algorithm 3, finds a vector c of polynomial coefficients in the numerical null space of H_v, i.e., c such that $\|H_v c\|_2 \leq \epsilon$. Note that unlike in Lemma 8, H_v is not exactly of rank m, so the polynomial $\mathcal{P}(z)$ with monomial coefficients given by the entries of c generally has $N/2$ roots. Moreover, it is not the case that all the roots of $\mathcal{P}(z)$ always lie inside the unit disk. Since we only want decaying exponentials in R_m, we keep only the m roots with modulus < 1. The exponents $\{\alpha_j\}_{j=1}^{m}$ for R_m are determined from these roots. Then, a least squares fit to the Fourier coefficients of f supplies the weights $\{\omega_j\}_{j=1}^{m}$. Pseudocode for the RPM is given in Algorithm 3.

Thorough details on the RPM, including a qualitative error bound in terms of the singular values of H_v, can be found in [27]. In some settings, such as in the example in Figure 5.3, a good choice for the tolerance parameter ϵ may be unclear. In this case, we modify Algorithm 3 so that ϵ is chosen automatically by detecting gaps in the small singular values of H_v that indicate the presence of a numerical null space. Algorithm 3 naively implemented has an $\mathcal{O}(N^3)$ cost because it requires finding the singular value decomposition (SVD) of H_v. This is improved if one finds the SVD with an algorithm that takes advantage of fast matrix-vector products for Hankel matrices (e.g., the randomized SVD [79],

Algorithm 3 The regularized Prony method.

```
Input: tolerance parameter $\epsilon$ and Fourier coefficients $v = (\hat{f}_0, \dots, \hat{f}_N)^T$.
Output: $\{(\omega_j, \alpha_j)\}_{j=1}^{m}$ defining $R_m$ in (5.2), so that $|R_m(j) - \hat{f}_j| \approx \epsilon$.
1. Construct the Hankel matrix $H_v$, where $(H_v)_{j+k} = \hat{f}_{j+k}$ for $0 \leq j, k, \leq N/2$.
2. Compute the SVD of $H_v$ to find $c$, where $\|H_v c\|_2 \leq \epsilon$, $\|c\|_2 = 1$.
3. Set $\mathcal{P}(z) = \sum_{\ell=0}^{N/2} c_\ell z^\ell$.
4. Find the $m \leq N/2$ roots $\{z_j\}_{j=1}^{m}$ of $\mathcal{P}(z)$ with $|z_j| < 1$. Set $\alpha_j = \log z_j$.
5. Compute the least squares solution to the system $V\omega = v$, where $V_{jk} = z_{k+1}^{j}$, $0 \leq j \leq N, 0 \leq k \leq m-1$.
```

Lanczos-based methods [65], etc.).

The regularized Prony method as a filter

In practice, one expects that samples of f are corrupted by noise. The RPM has a natural interpretation as a type of filter. Rather than, for example, filtering out the high frequency components of a signal, it separates a signal into the sum of two parts by splitting the Hankel matrix H_v into the sum $H_v = H_{R_m} + H_{\mathcal{N}}$. The first term encodes a sequence of coefficients that are well–approximated in Fourier space by a length m sum of exponentials (and thus correspond to a trigonometric rational). The second term encodes a sequence of coefficients that are not well approximated by such an expression. This is referred to as an annihilating filter in the literature on signals with so-called finite rates of innovation [170]. The example in Figure 5.3 displays noisy data collected by a hydrophone. The noise is not well represented by low to moderate degree trigonometric rationals, so it is filtered out by this structured low rank approximation process.

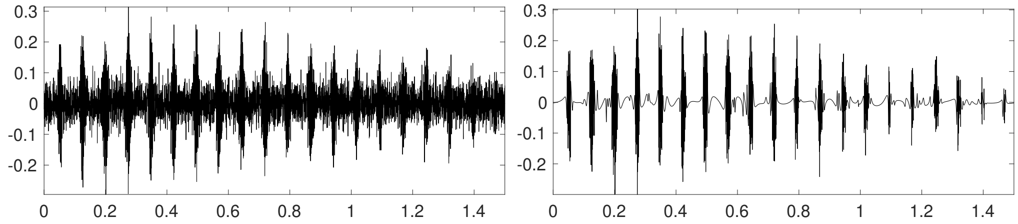


Figure 5.3: Left: A noisy recording of the trill portion of a Pacific blue whale's song. The sample consists of 6001 equally spaced observations recorded over 1.5 seconds [114]. Right: A type $(274, 275)$ trigonometric rational approximation to the signal. The approximant is constructed by applying Algorithm 3 directly to the data with the tolerance parameter $\epsilon = 2 \times 10^{-4}$. The RPM filters out highly oscillatory noise that is not well-captured by trigonometric rationals, making it easier to identify the time-localized pulses in the trill. Once the approximant is constructed, one can toggle between a barycentric representation and the RPM-constructed representation as a sum of complex exponentials. Various postprocessing tasks can also be performed (see Table 5.2).

5.3 Fourier and inverse Fourier transforms

The RPM and pronyAAA automatically construct compressed representations of f, but these representations are very different from one another. In this section, we describe Fourier/inverse Fourier transforms that allow us to move between these representations. If R_m is a length m sum of exponentials, the existence of a trigonometric rational $r_m = \mathcal{F}^{-1}(R_m)$ is guaranteed by Lemma 7. However, the lemma does not reveal if or how r_m can be expressed in barycentric form. In the same way, given a trigonometric barycentric interpolant $r_m^{\gamma,t}$, it is not clear from (5.4) how one can recover the sum of exponentials $R_m = \mathcal{F}(r_m^{\gamma,t})$. The exact recovery of one representation from the other is an ill-conditioned problem. With this in mind, we develop lossy but stable transform routines. In our software package REfit [40], these transforms are accessed with the commands `ft` and `ift`.

5.3.1 The forward transform

Given a barycentric rational $r_m^{\gamma,t}$ as in (5.4), we seek

$$R_m(k) = \mathcal{F}(r_m^{\gamma,t})(k) = \sum_{j=1}^{m} \omega_j e^{\alpha_j k}.$$

The parameters of interest can be expressed explicitly in terms of the poles and residues of $r_m^{\gamma,t}$. For each j, $\alpha_j = 2\pi i \eta_j$ and $\omega_j = e^{-\eta_j}\mathrm{Res}(r_m^{\gamma,t}(z), e^{\eta_j})$, where $z = e^{2\pi i x}$ and $\{\eta_j\}_{j=1}^{m}$ are those poles of $r_m^{\gamma,t}$ with $Im(\eta_k) > 0$ [28]. However, using these formulas requires the accurate computation of the poles η_j, $1 \leq j \leq m$, and their residues. In general, this is an ill-conditioned problem involving extrapolation off the interval of approximation. Trigonometric rationals with different pole-residue forms can behave almost indistinguishably on $[0, 1)$. Known stability results depend on the poles of $r_m^{\gamma,t}$ being sufficiently well-separated from one another and the residues $\mathrm{Res}(r_m^{\gamma,t}(z), e^{\eta_j})$ being bounded well away from zero [Sec. 2][115]. However, in our setting, we assume that $r_m^{\gamma,t}$ is an approximation to a function f that has algebraic singularities. Good resolution of these features is possible precisely because $r_m^{\gamma,t}$ has poles that cluster up near the singularities (see Figure 5.1). For this reason, we do not expect that the pole locations or the exact values of their residues can be computed with high accuracy. Similarly, the exact recovery of the parameters of R_m from its samples is known to be an ill-conditioned problem [135, 162].[5]

Instead of trying to recover R_m exactly, we apply a regularization that finds $\tilde{R}_{\tilde{m}} \approx R_m$, where $\tilde{m} \leq m$.[6] The poles of $r_m^{\gamma,t}$ can be approximately computed by solving a $(2m+1) \times (2m+1)$ generalized eigenvalue problem (see Section 5.5.7).

[5] In a related discussion, it is shown in [117] that in the case where $\{\alpha_1, \ldots \alpha_m\}$ are purely imaginary, the problem only becomes well-conditioned when the exponents $\{\alpha_1, \ldots \alpha_m\}$ are sufficiently well-separated from one another.

[6] If one allows for some of the weights in R_m to be zero, then one can construct sums of exponentials where it is always true that $\tilde{m} = m$.

Suppose $\{\tilde{\eta}_k\}_{k=1}^m$ are the computed poles with $Im(\tilde{\eta}_k) > 0$. We set $\tilde{\alpha}_k = 2\pi i\tilde{\eta}_k$. Then, instead of computing $\{\omega_1, \ldots, \omega_m\}$ using the explicit formula, we find a vector of weights $\tilde{\omega} = (\tilde{\omega}_1, \ldots, \tilde{\omega}_m)^T$ by solving the overdetermined linear system $V_{\tilde{\alpha}}\tilde{\omega} = \hat{r}$, where $\hat{r} = [(\hat{r}_m^{\gamma,t})_0 \cdots (\hat{r}_m^{\gamma,t})_{M-1}]^T$ is a vector of Fourier coefficients of $r_m^{\gamma,t}$, and $(V_{\tilde{\alpha}})_{j,k} = e^{\tilde{\alpha}_k(j-1)}$, $1 \leq j \leq M, 1 \leq k \leq m$. Note that all of the Fourier coefficients of $\tilde{r}_{\tilde{m}} = \mathcal{F}(\tilde{R}_{\tilde{m}})$ are exactly produced by a length $\tilde{m}$ sum of exponentials with exponents $\tilde{\alpha}_j = 2\pi i\tilde{\eta}_k$. In infinite precision, we would only need $\tilde{m}$ Fourier coefficients of $\tilde{r}_{\tilde{m}}$ to solve for the weights $\tilde{\omega}$ (see Lemma 8). Instead, we must fit to the coefficients of the nearby rational $r_m^{\gamma,t}$, and so apply a modest level of oversampling. We then test the accuracy of $\tilde{R}_{\tilde{m}}$ against a randomized sample of the Fourier coefficients of $r_m^{\gamma,t}$, and systematically increase M as needed. It is typically sufficient to choose $M = 2m$.

Since finding $\{e^{\tilde{\alpha}_j}\}_{j=1}^m$ and solving $V_{\tilde{\alpha}}\tilde{\omega} = \hat{r}$ are each $\mathcal{O}(m^3)$ operations, the cost for computing $\tilde{R}_{\tilde{m}}$ is dominated by procuring an accurate sample $\hat{r}$. This is done first by evaluating $2N_\epsilon + 1$ samples of $r_m^{\gamma,t}$, where N_ϵ is the ϵ-resolution of $r_m^{\gamma,t}$, on an equally spaced grid, and then applying an FFT. By default, ϵ is taken to be near machine precision, and N_ϵ can be approximately found automatically using, for example, an adaptation of Chebfun's `chop` algorithm [7]. In total, computing $\tilde{R}_m$ from $r_m^{\gamma,t}$ requires $\mathcal{O}(N_\epsilon \log N_\epsilon + N_\epsilon m + m^3)$ operations.

5.3.2 The inverse transform

We now assume $R_m(k) = \sum_{j=1}^m \omega_j e^{\alpha_j k}$ is given, where each α_j is distinct, $Re(\alpha_j) < 0$, and $\omega_j \neq 0$. We seek an efficient representation for $r_m = \mathcal{F}^{-1}(R_m)$, which is defined to be

$$r_m(x) = \sum_{k=-\infty}^{-1} \overline{R_m(-k)} e^{2\pi i x k} + \sum_{k=0}^{\infty} R_m(k) e^{2\pi i x k}.$$

It is not always true that r_m is a type $(m-1, m)$ trignometric rational function. However, this result does hold under the additional assumption that $R_m(0) = 0$, or, equivalently, $\int_0^1 r_m(x)\mathrm{d}(x) = 0$. We take this to be the case, and our objective is to construct a barycentric interpolant $r_m^{\gamma,t}$ to r_m. We show in the next lemma that for any set of distinct points $t = \{t_1, \ldots, t_{2m}\} \subset [0, 1)$, there is γ such that $r_m^{\gamma,t} = r_m$. However, except in special cases, it is numerically unstable to compute γ directly. The stable computation of γ and subsequently, the error $\|r_m - r_m^{\gamma,t}\|_\infty$, depends strongly on the choice of t.

Lemma 9. *Let r_m be a type $(m-1, m)$ trigonometric rational function with simple poles that is real-valued, continuous, and periodic on $[0, 1)$. Let $t \subset [0, 1)$ be a set of $2m$ distinct interpolating points. Then, there is a set of weights γ such that the trigonometric barycentric interpolant $r_m^{\gamma,t}$ recovers r_m exactly.*

Proof. Consider the denominator q_m in $r_m = p_{m-1}/q_m$, and assume that q_m has no shared zeros with p_{m-1}. Since q_m is a trigonometric polynomial, we can write it in barycentric form with respect to the interpolating points in t:

$$q_m(x) = \ell_t(x) \sum_{j=1}^{2m} w_j q_m(t_j) \cot(\pi(x - t_j)), \qquad \ell_t(x) = \prod_{j=1}^{2m} \sin(\pi(x - t_j)), \quad (5.7)$$

where $w_j = 1/\prod_{k=1, j\neq k}^{2m} \sin(\pi(t_k - t_j))$ are the polynomial barycentric weights [25] associated with t. By setting $\gamma_j = q_m(t_j)w_j$ and $f_j = r_m(t_j)$, we have via (5.4) that there is $r_m^{\gamma,t}(x) = n(x)\ell_t(x)/q_m(x)$ for some function n, where for each j, $r_m^{\gamma,t}(t_j) = r_m(t_j)$. We must now show that $n(x)\ell_t(x) = p_{m-1}(x)$.

The barycentric trigonometric polynomial interpolant to p_{m-1} on t exists and is given by $p_{m-1}(x) = \ell_t(x)\sum_{j=1}^{2m} w_j p_{m-1}(t_j)\cot(\pi(x-t_j))$. Expanding this in exponential form, we have that $p_{m-1}(x) = c_m e^{2\pi i m x} + \ldots + c_{-m}e^{-2\pi i m x}$, where

$$c_m = \frac{1}{4i}\exp\left(-\pi i\sum_{j=1}^{2m} t_j\right)\sum_{j=1}^{2m} w_j p_{m-1}(t_j), \quad c_{-m} = -\exp\left(2\pi i\sum_{j=1}^{2m} t_j\right)c_m.$$

Since p_{m-1} is of degree $m-1$, $c_m = c_{-m} = 0$, so $\sum_{j=1}^{2m} w_j p_{m-1}(t_j) = 0$. This implies that $\sum_{j=1}^{2m}\gamma_j r_m(t_j) = 0$, as $r_m(t_j) = p_{m-1}(t_j)/q_m(t_j)$. Now it is clear that $n(x)\ell_t(x)$ is also a trigonometric polynomial of degree $m-1$. Since $n\ell_t$ interpolates p_{m-1} at $2m$ points, they must agree everywhere. □

The computation of γ as in Lemma 9 via polynomial barycentric weights and the evaluation of q_m is numerically unstable except in very special cases [163]. Constructing a stable interpolant requires a rather careful selection of barycentric nodes. As we describe in the next section, some of the more obvious methods for selecting the nodes perform poorly and lead to instabilities in the form of spurious poles. The following discussion is somewhat technical, but it introduces an effective heuristic for choosing a "good" set of barycentric nodes and then stably constructing $r_m^{\gamma,t} \approx r_m$.

A modified pronyAAA for rational recovery

A simple strategy for choosing nodes is to evaluate r_m on a fine enough grid, and then apply m steps of pronyAAA to construct the interpolant $r_m^{\gamma,t}$. This method does not usually exactly recover r_m (see the discussion of exact recovery in Section 5.3.1), and it can be the case that the error $\|r_m^{\gamma,t} - r_m\|_\infty$ is unacceptably large. A few additional steps of pronyAAA may drive the error down, though this results in a trigonometric rational interpolant with more poles than

r_m. However, a more pernicious problem with this approach is that demanding accuracy close to machine precision from AAA-based methods can result in spurious poles on the interval of approximation that cannot be eliminated without adversely impacting accuracy [119].

To avoid introducing spurious poles, we make use of the poles of r_m, which are known explicitly from R_m via Lemma 7. There is no hope of exactly preserving the poles. However, if m is fixed and $r_m^{\gamma,t}$ is constructed such that it approximately preserves the given poles, then it cannot also admit arbitrary spurious poles. This motivates a three-step procedure for constructing $r_m^{\gamma,t}$ that mixes a pole-preservation strategy involving a type $(m+K-1, m+K)$ trigonometric rational with a data-driven strategy from pronyAAA:

(1) A candidate set $\tilde{t}$ of $2m+2K$ barycentric nodes is chosen, where $K \geq 0$ is an oversampling parameter. Subsets of $\tilde{t}$ admit type $(m-1, m)$ barycentric trigonometric interpolants with poles close to those of r_m.

(2) The interpolant $r_{m+2K}^{\tilde{\gamma},\tilde{t}}$ is constructed via a pole-preserving linearized least-squares fit to samples of r_m, so that it has $2m$ poles close to the poles of r_m.

(3) The pronyAAA cleanup procedure (see Section 5.2.3) is applied to remove the $2K$ poles of $r_{m+2K}^{\tilde{\gamma},\tilde{t}}$ with the smallest residues. This selects t, a set of $2m$ barycentric nodes, from $\tilde{t}$. The barycentric weights $\{\gamma_1, \ldots, \gamma_{2m}\}$ are then computed via (5.5). Note that the poles of $r_m^{\gamma,t}$ must also be recomputed.

A version of this method without oversampling (i.e., with $K = 0$, $\tilde{t} = t$, and $\tilde{\gamma} = \gamma$) is useful for motivating how the barycentric nodes in Step (1) are selected. In such a setting, Step (2) simplifies substantially and Step (3) is not needed. However, it is more stable to choose $K > 0$ and this is usually required in practice. We first describe the $K = 0$ case, and then use it to explain the method for $K > 0$.

Case 1: $K = 0$**.** Suppose that T, the discretization of $[0, 1)$ from which t is chosen, consists of points $x_0 < x_1 < \ldots < x_{2N}$. Let $P = \{\eta_1, \ldots, \eta_{2m}\}$ be the poles of r_m. Ideally, $r_m^{\gamma,t}$ can be constructed so that its poles are given by P. Noting that the poles of $r_m^{\gamma,t}$ are the zeros of the denominator polynomial $d_m(x) = \sum_{j=1}^{2m} \cot(\pi(x - t_j))$ in (5.4), we introduce the matrix $D_T \in \mathbb{C}^{(2m+1)\times(2N+1)}$:

$$D_T = \left[\begin{array}{cccc} \ell_{1,0} & \cdots & \cdots & \ell_{1,2N} \\ \vdots & & & \vdots \\ \ell_{2m,0} & \cdots & \cdots & \ell_{2m,2N} \\ \hline r_m(x_0) & \cdots & \cdots & r_m(x_{2N}) \end{array}\right], \quad \ell_{j,k} = \cot(\pi(\eta_j - x_k)). \tag{5.8}$$

Using D_T, we relate the selection of barycentric nodes to a column subset selection problem. Indexing from 0, denote by $(D_T)_k$ the kth column of D_T. The kth column is associated with the point x_k in T. The set of nodes $t = \{x_{k_1}, x_{k_2}, \ldots x_{k_{2m}}\}$ then corresponds to a collection of columns that form the submatrix $D_t = \left[(D_T)_{k_1}, \ldots, (D_T)_{k_{2m}}\right]$. From Lemma 9, there is $\gamma = (\gamma_1, \ldots, \gamma_{2m})^T$ such that $r_m^{\gamma,t} = r_m$. We note that γ is in the null space of D_t: the first $2m$ entries of $D_t\gamma$ are evaluations of d_m at its zeros. The last entry of $D_t\gamma$ is also zero, since numerator of $r_m^{\gamma,t}$ is of degree $m-1$ (see Section 5.2.2). If γ can be computed from D_t accurately, then clearly t is an excellent set of interpolating points for constructing an interpolant to r_m. However, the accuracy of this computation depends on properties of D_t. In particular, there are stable ways to compute γ if $2m-1$ of the columns of D_t form a well-conditioned matrix [72, 156].

This suggests that we choose the points t by choosing a subset of columns from D_T that are close to orthogonal. Several kinds of rank-revealing algorithms can be applied to D_T to approximately solve this problem, including the column-pivoted QR (CPQR) algorithm. This constructs the factorization $D_T P = Q_T R$, where P is a permutation matrix, and the leading $\ell \leq \operatorname{rank}(D_T)$ columns of $D_T P$ have been greedily selected to minimize their linear dependence on one another [65, Sec. 6.4]. As a consequence of Lemma 9, any submatrix of D_T consisting of $2m$ or more columns is rank-deficient, so $\operatorname{rank}(D_T) \leq 2m-1$.[7] We choose $2m$–1 points in t by performing CPQR on D_T. In principle, the final point in the set t should be chosen so that the accuracy of the computed right singular vector in the nullspace of D_t in (5.8) is maximized[8]. Instead, we simply choose the point associated with the column in the trailing $(|T|$–$2m$+$1)$ columns of $D_T P$ that has the smallest 2-norm. Though they are selected quite differently, these CPQR-selected barycentric nodes concentrate around singularities, just like the nodes selected by pronyAAA (see Section 5.3.2 and Figure 5.4).

If vectors in the null space of D_t can be accurately computed, then γ can be taken as the last right singular vector of D_t. However, this is rarely the case. The accurate recovery of γ from D_t can be problematic even with the best possible choice $t \subset [0, 1)$. For this reason, we require a strategy that additionally incorporates a fit to samples. The simplest idea is to use the $2m$ points as selected above, and then find the barycentric weights via (5.5). However, this strategy does not seem to eliminate spurious poles or reduce error as effectively as the procedure we describe below.

Case 2: $K \neq 0$**.** In practice, we take $K = 1$, though one can also choose a larger K. Construct an oversized candidate set $\tilde{t} = \{x_{k_1}, \ldots, x_{k_{2m+2K}}\}$ using

[7] If we assume that r_m has a denominator of exactly degree m, then $\operatorname{rank}(D_T) = 2m-1$.

[8] As a proxy, one could maximize the gap between the last two singular values of D_t [156, Ch. 4]

the CPQR-based method from Case 1. The matrix $D_{\tilde{t}}$ is then of dimensions $(2m+1) \times (2m+2K)$ and has a numerically detectable null space. We compute the barycentric weights of the interpolant $r_{2m+2K}^{\tilde{t},\tilde{\gamma}}$ by requiring that $\tilde{\gamma} = \tilde{Q}\eta$, where the columns of $\tilde{Q}$ are orthogonal and approximately span the null space of $D_{\tilde{t}}$. We select η to minimize the value $\|C\tilde{Q}\eta\|_2$, with C constructed as in (5.6). Approximate poles and residues of $r_{2m+2K}^{\tilde{t},\tilde{\gamma}}$ can then be computed in $\mathcal{O}(m^3)$ operations (see Section 5.5.7). It is almost always the case in practice that $2K$ poles of $r_{2m+2K}^{\tilde{t},\tilde{\gamma}}$ are negligible in that they have residues with tiny magnitudes. With this in mind, we sort the poles by the magnitude of their residues. As in the pronyAAA cleanup routine, for each of the $2K$ poles with the smallest residues, we eliminate the point in $\tilde{t}$ that is nearest to the pole. The remaining points in $\tilde{t}$ are taken as t, and the set of barycentric weights are found as in a standard step of pronyAAA, i.e., as the minimizer of (5.5).

This strategy first selects a set of interpolating points for which an interpolant with good properties (e.g, poles off $[0, 1)$) is known to exist, and then fits the interpolant to samples of r_m. We remark that this is a heuristic: there is no guarantee in this algorithm that spurious poles are avoided, nor is there a guarantee on the accuracy to which the original poles of r_m are preserved. It remains unclear why the solution in Step (3) often seems to inherit the good pole properties associated with the initial solution in Step (2), and under what circumstances this inheritance can be assured. Nonetheless, we find that the method works extremely well in many cases where simply applying pronyAAA fails.

Implementational details. In practice, we start with $K = 0$. When γ can be recovered with high accuracy directly from D_t , we recover it and end the procedure. This can be checked by computing the singular values of D_t or by using

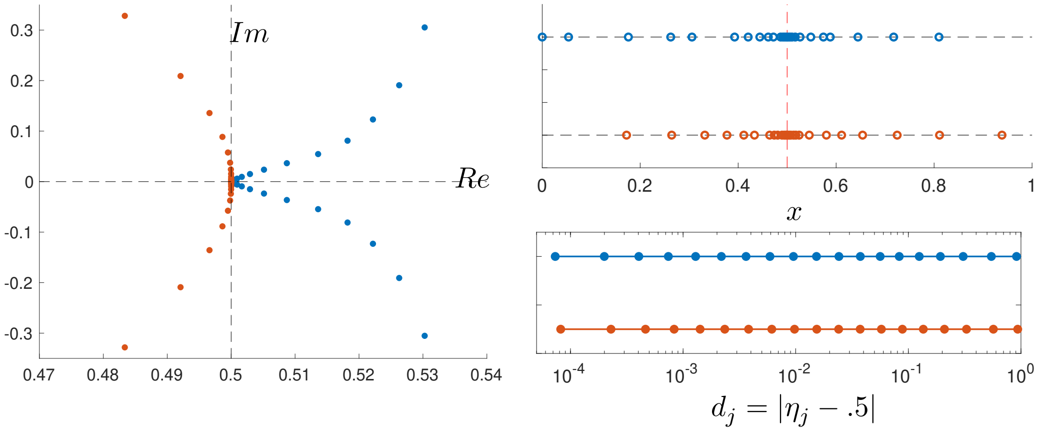


Figure 5.4: Left: poles of r_{bary} (blue) and r_{ift} (orange) are plotted in the complex plane. Here, r_{bary} is a type $(37, 38)$ barycentric interpolant to f constructed via pronyAAA, and $r_{ift} = \mathcal{F}^{-1}(R_{19}) \approx f$, where R_{19} is an exponential sum as in (5.2), and r_{ift} is then constructed by applying the inverse Fourier transform algorithm from Section 5.3.2 to R_{19}. The function f is given by $f(x) = |\sin(\pi(x - 1/2))| - \pi/2$, and has a singularity at $x = 1/2$. Right upper: The locations of the barycentric nodes for r_{bary} (blue) and r_{ift} (orange). Right lower: The distances $d_j = |\eta_j - .5|$ from the singularity, where each η_j is a pole with $\mathrm{Im}(\eta_j) > 0$, are sorted by size and plotted on a logarithmic scale (shown in blue for r_{bary}, orange for r_{ift}).

estimates related to the CPQR routine [72]. When this isn't possible, we set $K = 1$ and enlarge our selection of candidate barycentric nodes, which requires no additional computation. Then, we move on to Steps (2) and (3). If the method fails and spurious poles are detected, we first try enlarging $\tilde{t}$ by setting $K = 2$ and trying again. If this fails, it can often be remedied by resampling r_m on a denser grid and starting over at Step (1). When resampling does not solve the issue, we instead construct a stable barycentric interpolant using pronyAAA by accepting a lower level of accuracy.

Example: Two types of barycentric interpolants

In Figure 5.4, we compare the properties of two types of rational approximations to $f(x) = |\sin(\pi(x-1/2))|-\pi/2$. First, we apply pronyAAA to a set of 6000 samples of f taken on an equally-spaced grid T. This constructs r_{bary}, a type $(37, 38)$ trigonometric rational, where away from the singularity, $|f(x) - r_{bary}(x)| \approx 10^{-8}$. The locations of the barycentric nodes selected by pronyAAA are plotted (blue) in the upper right panel of Figure 5.4. In the left panel, a subset of the poles of r_{bary} are plotted (blue) in the complex plane. Both the nodes and the poles cluster up near the singularity $x = 1/2$. Shown in red in the same plots are the CPQR-selected barycentric nodes from Section 5.3.2, and the poles of the barycentric trigonometric rational $r_{ift} = \mathcal{F}^{-1}(R_m)$, where $m = 19$. Here, R_m is an exponential sum constructed via the RPM using samples of f on T, and r_{ift} is constructed using the procedure in Section 5.3.2. The nodes and poles of r_{ift} also cluster near $x = 1/2$, but in spatial patterns that are quite different from those of r_{bary}. A closer investigation of the pole clustering patterns (Figure 5.4, lower right) reveals that in both cases, the sets of distances $d_1 \leq d_2 \leq \ldots \leq d_{19}$ from the singularity, where $d_j = |\eta_j - 1/2|$ and each η_j is a pole with $\mathrm{Im}(\eta_j) > 0$, have the tapered-type spacing on a logarithmic scale that is associated with best (and near-best) convergence rates [166].

5.4 Signal reconstruction in time and frequency space

With the Fourier and inverse Fourier transforms available, we can combine the advantages of pronyAAA and the RPM to overcome various issues, such as undersampling or noise. In this section, we illustrate this idea with two examples. Then in Section 5.5, we describe a collection of algorithms for computing with trigonometric rational functions and exponential sums that exploits our ability to move stably between the representations.

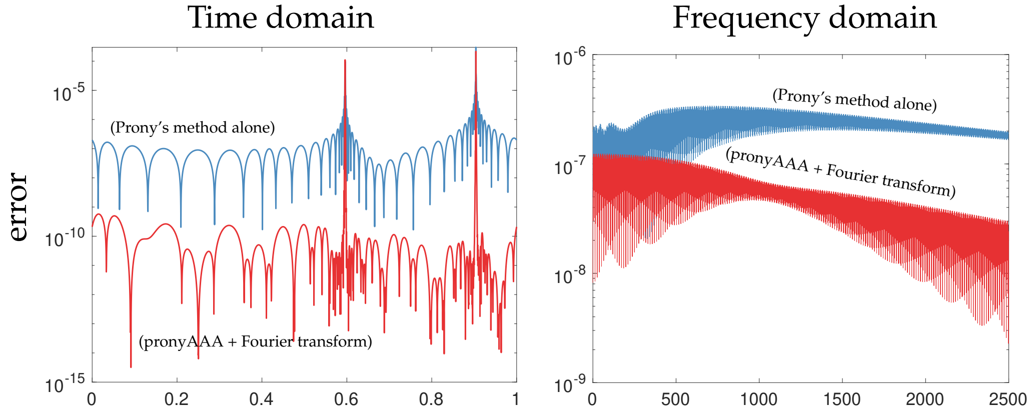


Figure 5.5: Left: The absolute error in approximating $f(x) = |\sin(2\pi x)/4 + \exp(\sin(2\pi x))|/4 + c$ with two different rational approximants, $\mathcal{F}(R_b)$ (blue) and $\mathcal{F}(R_r)$ (red), is plotted on a logarithmic scale at 3000 equally-spaced points: R_b is an exponential sum of length 14 that was adaptively constructed via the RPM (Algorithm 3) from a sample consisting of only 1401 equally-spaced points. The tolerance parameter is set to $\epsilon = 10^{-10}$, but the coarseness of the sample limits the achievable accuracy of the representation. R_r (red) is constructed by first applying pronyAAA in signal space to construct a barycentric interpolant $r_r^{\gamma,t}$, and then using the Fourier transform function to compute $R_r = \mathcal{F}(r_r^{\gamma,t})$. Right: The absolute errors in Fourier space between accurately computed Fourier coefficients of f and the exponential sums R_b (blue) and R_r (red) are plotted on a logarithmic scale against the modes $0, 1, \ldots 2500$.

5.4.1 An undersampled function

In this example, we consider a function $f(x) = |\sin(2\pi x)/4 + \exp(\sin(2\pi x))|/4 + c$, which has Fourier coefficients that decay asymptotically like $\mathcal{O}(|k|^{-2})$, where k denotes the kth Fourier mode. Here, c is a normalization parameter ensuring that the mean value of f over $[0, 1)$ is zero. We suppose that f is sampled at 1401 equally spaced points, and that an exponential sum representing $\mathcal{F}(f)$ is desirable for downstream tasks. The direct application of Prony's method performs poorly because f is undersampled. We denote the constructed exponential sum as R_b. The error in the computation of the Fourier coefficients via the FFT is on the order of 10^{-6}, so we cannot expect accuracy much better than that. However, an alternative approach is to apply pronyAAA to construct the barycentric interpolant $r_r^{\gamma,t}$, and then apply the Fourier transform function from Section 5.3.1

to compute $R_r = \mathcal{F}(r_r^{\gamma,t})$.

In Figure 5.5 (left), we use the pole-residue format to directly evaluate the values of the rationals associated with the two types of constructed exponential sums. There is a tiny band around the two singularities in time space where the errors incurred by the two methods are approximately the same. Elsewhere, the accuracy achieved by first applying pronyAAA is nearly double that attained by Prony's method alone. The error in recovering the Fourier coefficients of f is diffuse but also more accurate, especially in the extrapolation of the tail (Figure 5.5, right). The exponential sum R_r, with only 13 terms, is a representation of f with highly localized error behavior, and it is in a form efficient for storage, convolution, and other tasks (see Section 5.5).

5.4.2 Reconstruction of an ECG signal

Rational approximation methods are effective in many biomedical monitoring tasks, including the processing of electrocardiogram (ECG) signals [55, 62]. In [55], rational functions constructed in the orthogonal rational Malmquist–Takenaka basis are used to reconstruct ECG signals and then classify them. The rationals perform with better overall compression properties and a number of other advantages when compared to wavelets, splines, and other families of functions [101]. We do not expect to outperform such a highly specialized scheme with our approach. However, we use this example to illustrate that our more general-purpose method is extremely effective at constructing a denoised representation of the signal directly from samples.

In this example, we apply the RPM and fit a rational function directly to noisy ECG data taken from the PhysioNet MIT BIH arrhythmia database [118]. As in [55], the location of its poles can be used for classification and feature recognition tasks. Using the inverse Fourier transform function described in Section 5.3.2, we can construct a trigonometric barycentric trigonometric rational representation of the function, which is a convenient format for identifying local extrema (see Section 5.5). This can all be done with three lines of code in REfit:

```
R = efun(data, 'tol', 1e-3);
r = ift(R);
extrema = [ min(r); max(r)];
```

If one tries to use pronyAAA directly, the result is a trigonometric rational with 200 poles, and the data set only contains 645 samples. Of these poles, 62 are spurious and lie on the interval of approximation. This happens because the pronyAAA algorithm does not distinguish between the signal and the noise, and it tries to induce a fit to noise by adding poles. A better approach is to first apply the RPM. Within the first two lines of the above code, several tasks are being executed: First, the exponential sum R_m (here, $m = 35$) stored in `R` is constructed via the RPM. The RPM automatically filters out additive noise on the sample with magnitudes approximately at or below the tolerance level $\epsilon = 10^{-3}$. Then, `R` is used to extrapolate high frequency information that lies beyond the noise limitation (see Figure 5.6, left). This provides an enriched sample for selecting interpolating points and constructing the barycentric interpolant `r`. The construction of `r` in this way can be viewed as a form of super-resolution [31]. If one tries to construct a barycentric interpolant without enriching the given sample, spurious poles appear that cannot be eliminated without destroying the accuracy of the approximation. This is because the signal is not well-resolved in the time domain by the original sample. Once `r` is available, one can then auto-

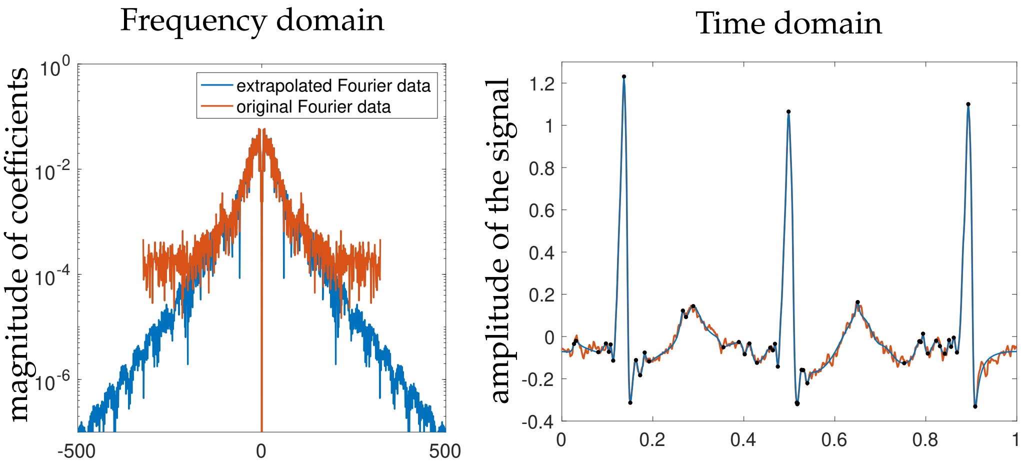


Figure 5.6: The superresolution of an ECG signal. Left: The magnitude of the Fourier coefficients of the original signal (orange). The decay of the coefficients stagnates due to noise in the signal, and this pollutes the higher frequencies. Once an exponential sum representation R_m is constructed, we can extrapolate to higher frequencies by evaluating R_m, and thereby super–resolve the signal (blue). Right: A barycentric rational approximant (blue) in the time domain is computed using the extrapolated Fourier data. It serves as a denoised version of the original ECG signal (orange). The local extrema are identified (black dots) using the differentiation and rootfinding algorithms from Section 5.5.

matically and efficiently perform a variety of processing tasks, such as rootfinding and the detection of maxima and minima.

5.5 Algorithms for computing with rationals and exponential sums

In this section, we give an overview of several of the algorithms used in our software [40] to compute with trigonometric rational functions. Throughout, the Fourier and inverse Fourier transform functions can be used to move between representations as needed. For operations on trigonometric rational functions that return trigonometric rationals, we recompress and represent the function using exponential sums and/or barycentric forms whenever possible.

5.5.1 Compression for suboptimal sums of exponentials

Exponential sums are closed under addition and multiplication, but a sum R_m resulting from the naive application of these and other operations is often suboptimal in the sense that a shorter sum $\tilde{R}_{\tilde{m}}$ exists, where $|R_m(j) - \tilde{R}_{\tilde{m}}(j)| < \epsilon$ for $0 \leq j$. One of the major advantages of working with exponential sums is that $\tilde{R}_{\tilde{m}}$ can be constructed at a computational cost that usually depends on m, rather than the ϵ-resolution parameter N_ϵ associated with R_m.

Using AAK theory for finite rank Hankel operators, one can show (see [132, Thm. 3.2]) that there is a length $\tilde{m} \leq m$ approximation that satisfies the inequality

$$\|\mathcal{F}^{-1}(R_m) - \mathcal{F}^{-1}(\tilde{R}_{\tilde{m}})\|_{L_2} \leq 2\,\sigma_{\tilde{m}}(\Gamma_R),$$

where Γ_R is the infinite matrix with entries $(\Gamma_R)_{j+k} = R_m(j+k)$, $j, k \geq 0$. In [132], an $\mathcal{O}(m^3)$ algorithm for recovering $\tilde{R}_{\tilde{m}}$ directly from the parameters of R_m is developed; a closely-related approach using only properties of finite-dimensional Hankel matrices is described in [27]. This method is successfully employed in [82] within a scheme that uses rational approximations to solve Burger's equation. However, the implementation requires the judicious use of high-precision arithmetic, which we wish to avoid.

Instead, we note that when a length $\tilde{m} < m$ recurrence is approximately satisfied by the sequence $\{R_m(0), R_m(1), \ldots\}$, this fact is often captured surprisingly well with a Pisarenko-like method [56, 131] closely related to Prony's method that involves only a small sample of $M > 2m$ observations of R_m. Specifically, we construct a small $(M/2+1) \times (m+1)$ rectangular Hankel matrix H with entries $H_{jk} = R_m(j+k)$. Then, we apply the RPM from Algorithm 3 on H to construct $\tilde{R}_{\tilde{m}}$. We check the error $|R_m(j) - \tilde{R}_{\tilde{m}}(j)|$ on a random sample of integers $0 \leq j \leq N$, where N is the ϵ-resolution parameter used in the original construction of R_m. When the error is too large, we increase M and try again. The cost to compute $\tilde{R}_{\tilde{m}}$ is $\mathcal{O}(Mm^2)$. In a worst-case scenario, M can grow as large as N. We observe experimentally that this approach is often very effective, but more work is needed to understand the conditions under which it is guaranteed that $M \ll N$.

5.5.2 Sums of trigonometric rationals

If S_ℓ and G_n are exponential sums, then $R_m = S_\ell + G_n$ can be constructed straightforwardly. However, R_m may be of suboptimal length. We apply the compression algorithm with $\epsilon \approx \epsilon_{mach}$ to R_m, where $m = \ell + n$, to find $\tilde{R}_{\tilde{m}}$. This "compression–plus" method is especially useful for tasks that involve repeated summations and require many recompressions. The compression–plus algorithm is used automatically in REfit when the '+' operator is used between efun objects. For summing trigonometric rationals s_ℓ and g_n represented as rfuns, we simply evaluate the sum and then apply pronyAAA to find $r_m = s_\ell + g_n$.[9] The rfun and efun objects can also be combined in various ways. The syntax `r = s + g` adds two rfuns and returns an rfun by default. The expression `[r, R] = s + g` automatically retrieves the efun `R = ft(r)` in addition to `r`. When an rfun and efun are summed together, both rfun and efun outputs are returned by default.

[9] If this proves difficult due to spurious poles, we use the Fourier and inverse Fourier transforms to convert to efuns, perform the addition, and then convert back to an rfun.

5.5.3 Convolutions of trigonometric rationals

The convolution of two trigonometric rationals s_ℓ and g_n can be constructed in Fourier space by finding the exponential sum $R_{m_0} = (\mathcal{F}s_\ell)(\mathcal{F}g_n) = S_\ell G_n$. The product can be computed directly in a closed form, but this results in a large sum with $m_0 = mn$. Sums of this type often have small weights and/or exponential terms that do not contribute substantially to the sum. To find a shorter sum $\tilde{R}_{\tilde{m}}$, we find an upper bound m on $\tilde{m}$ by determining how many terms in R_{m_0} have a negligibly small influence. Then, we apply the compression algorithm using rectangular Hankel matrices of the form $H_{jk} = S_\ell(j+k)G_n(j+k)$, $0 \le j \le M$, $0 \le k \le m$. For efuns, this operation is accessed by typing `S .* G`. For rfuns, the command `[r, R] = conv(s, g)` uses the Fourier and inverse Fourier transform functions to apply the above scheme, returning `r` as an rfun and `R` as an efun.

5.5.4 Products of trigonometric rationals

The product of two trigonometric rationals r_ℓ and s_n in the time domain is equivalent to their convolution in Fourier space. If $R_\ell = \mathcal{F}(r_\ell)$ and $S_n = \mathcal{F}(s_n)$ are each sums of complex exponentials, then

$$\mathcal{F}(r_\ell s_n)(k) = (R_\ell * S_n)(k) = \sum_{j=-\infty}^{\infty} R_\ell(k-j)S_n(j) \approx \sum_{j=-N_\epsilon}^{N_\epsilon} R_\ell(k-j)S_n(j), \quad (5.9)$$

where N_ϵ is the ϵ-resolution parameter for S_n. The fast evaluation of (5.9) at M consecutive points is equivalent to a matrix-vector multiply with a Toeplitz matrix. Since $r_\ell s_m$ is a trigonometric rational with at most $\ell + n$ poles, we apply the compression algorithm to find $G_m \approx \mathcal{F}(r_\ell s_n)(k)$, with $m = \ell + n$. If `R` and

`S` are efuns, this command is accessed by typing `conv(R, S)`. If r_ℓ and s_n are represented with rfuns `r` and `s`, respectively, then the syntax `r.*s` returns a new rfun representing the product. The new rfun is constructed by simply applying pronyAAA to the function $r_\ell(x)s_n(x)$.[10]

5.5.5 Differentiation

The kth derivative of r_m, denoted $r_m^{(k)}$, is a type $(km-1, km)$ trigonometric rational. However, $r_m^{(k)}$ is fundamentally of a different form than the trigonometric rationals constructed via pronyAAA and the RPM. It has m conjugate-pairs of poles, and each pole is of multiplicity k. While it is often possible to represent derivatives with trigonometric rationals having simple poles, it isn't always a sensible choice. By default, REfit returns a function handle for evaluating derivatives (or their Fourier transforms) whenever `diff(·,k)` is applied to an rfun (or an efun). However, one can also use `diff(·,k, 'type')`, to specify that an efun or rfun should be returned.

Differentiation Fourier space. When r_m is represented by the complex exponential sum R_m in Fourier space, the Fourier coefficients of $r_m^{(k)}$ are given by $\mathcal{F}(r_m^{(k)})(j) = (2\pi ij)^k R_m(j)$. The command `h=diff(R,k)` by default returns a handle for evaluating this function in Fourier space. If instead, for example, one types `diff(R,k, 'efun')`, the RPM is applied to construct a representation of $\mathcal{F}(r_m^{(k)})$ as a sum of weighted complex exponentials (without polynomial coefficients).

[10] If this proves difficult due to spurious poles, we use the Fourier and inverse Fourier transforms to convert to efuns, perform the convolution in Fourier space, and then convert back to an rfun.

Differentiation in the time domain. Derivatives of barycentric trignometric rational interpolants satisfy a recurrence relation and can be expressed in a simple closed form. To see this, consider the linearization of $r_m^{\gamma,t} = n_{m-1}/d_m$, which can be differentiated as $(r_m^{\gamma,t} d_m)' = (n_{m-1})'$. Plugging in the definitions from (5.4) results in the following formula, which holds everywhere on $[0, 1)$ except at the interpolating points:

$$(r_m^{\gamma,t})'(x) = -\pi \frac{\sum_{j=1}^{2m} \gamma_j \csc^2(\pi x - \pi t_j) \left(f_j - r_m^{\gamma,t}(x)\right)}{\sum_{j=1}^{2m} \gamma_j \cot(\pi x - \pi t_j)}. \tag{5.10}$$

To evaluate $(r_m^{\gamma,t})'$ at the interpolating points $t = (t_1, \ldots, t_{2m})^T$, we use the special differentiation matrices introduced in [13]. Explicit descriptions of recursive formulas for computing higher derivatives are also found in [13]. All of this is encoded within a function handle that is accessed in REfit by applying the command `diff` to an rfun.

5.5.6 Integration

The indefinite integral of a trigonometric rational r_m is not itself a trigonometric rational. In Fourier space, if $R_m = \mathcal{F}(r_m)$ is a sum of complex exponentials as in (5.2), then except at $k = 0$, the Fourier coefficients of $\mathcal{F}(g)$, where

$$g(y) = \int_0^y r_m(x) dx,$$

are given by $\hat{g}_k = R_m(k)/2\pi ik$ for $k \neq 0$. A function handle for evaluating $\mathcal{F}(g)$ is returned when `cumsum` is applied to an efun. One can also try to fit a new complex exponential to $\mathcal{F}(g)$ by typing `cumsum(·, 'efun')`, though it may not be an efficient representation.

If `cumsum` is applied to an rfun, we supply a handle for g that applies Gauss-

Legendre quadrature [78]. The stable evaluation property of the barycentric form is advantageous here, which is why we do not instead make use of the pole-residue form of the rational $r_m(z)$ in (5.3). To integrate r_m over a finite interval $[a, b] \subset [0, 1)$, the command `sum(·,a,b)` can be applied to an rfun or an efun.

5.5.7 Rootfinding and polefinding

The roots of the barycentric trigonometric rational $r_m^{\gamma,t}$ coincide with the eigenvalues of a linear pencil. Specifically, if $r_m^{\gamma,t}(\zeta_j) = 0$ and $\mu = e^{2\pi i \zeta_j}$, then there is nonzero vector y such that $Ey = \mu By$, where

$$E = \left[\begin{array}{ccc|c} e^{2\pi i x_1} & & & i\omega_1 e^{2\pi i x_1} \\ & \ddots & & \vdots \\ & & e^{2\pi i x_{2m}} & i\omega_{2m} e^{2\pi i x_{2m}} \\ \hline f_1 & \cdots & f_{2m} & 0 \end{array}\right], B = \left[\begin{array}{ccc|c} 1 & & & i\omega_1 \\ & \ddots & & \vdots \\ & & 1 & i\omega_{2m} \\ \hline 0 & \cdots & 0 & 0 \end{array}\right]. \tag{5.11}$$

The pencil (E, B) has at least two infinite eigenvalues and one eigenvalue at $\mu_0 = 0$ corresponding to $\zeta_0 = -\infty$ (this captures the asymptotic behavior of $r_m^{\gamma,t}$). Once the remaining $2m - 2$ eigenvalues are found, the zeros of $r_m^{\gamma,t}$ are immediate. The command `roots` applied to rfuns or efuns applies this algorithm and returns real-valued roots. For efuns, this requires first converting to an rfun via the Fourier transform. The command `roots(·, 'all')` additionally returns complex-valued roots.

The poles of $r_m^{\gamma,t}$ can be found in a similar way: the pencil $(\tilde{E}, B)$, where $\tilde{E}$ is identical to E except that each f_j in the last row is replaced by 1, has at least one infinite eigenvalue. If $\tilde{\mu}_j$ is one of the remaining finite eigenvalues, then

$\eta_j = \log \tilde{\mu}_j / 2\pi i$ is a pole of $r_m^{\gamma,t}$. This approach does not ensure that the conjugate symmetry of the poles is exactly preserved. If it is important to exactly preserve the pole symmetry, it is better to represent r_m with an exponential sum R_m.

Residues. We compute the residues of the barycentric interpolant $r_m^{\gamma,t}$ using the fact that $r_m^{\gamma,t} = n_{m-1}/d_m$, where n_{m-1} and d_m are trigonometric polynomials as in (5.4). Since the poles of $r_m^{\gamma,t}$ are simple, the residue for a given pole η_j can be evaluated as

$$\mathrm{Res}(r_m^{\gamma,t}, \eta_j) = \frac{n_{m-1}(\eta_j)}{d_m'(\eta_j)}.$$

The residues of the poles of $\mathcal{F}^{-1}(R_m)$, where R_m is an exponential sum, have a closed form formula involving the parameters of R_m (see Section 5.3.1). The command `[res, pol] = Res(·)` returns the residues along with the associated poles.

Minima and Maxima. The commands `min` or `max` a return the local minima (or maxima) attained by the represented trigonometric rational on the interval $[0, 1)$. The global minimum, for example, can be found by typing `min(min(·))`. To compute the extrema, we use an rfun and apply the differentiation formula in (5.10) to evaluate its derivative. We use this to construct an rfun representing $(r_m^{\gamma,t})'$, find its roots, and then test for concavity. For an application, see Section 5.4.2.

5.5.8 Other commands

The REfit package includes several other commands, including commands for data visualization and common tasks in signal processing, such as the application of filters and the computation of cross-correlations. Commands related to the pole–residue format of the rational $\tilde{r}_m(z)$ from (5.3) are also available. This format is closely related to the notion of the z-transform [123], and is important for analysis and interpretation. It is also possible to construct efun representations of signals in the time domain, and to build barycentric rational approximations with the nonperiodic AAA algorithm in the frequency domain. We remark that an independently-developed method that uses these two types of representations has recently been introduced in [44].

5.6 Conclusion

We have introduced a framework for signal reconstruction and automated computing that employs efficient representations in both time and frequency space. Our work integrates ideas from the harmonic analysis community involving exponential sums and Hankel operator theory [27, 132, 135] with developments in adaptive barycentric rational interpolation [26, 86, 119]. An implementation of all of the described methods, as well as access to the examples, is publicly available in the REfit software package [40].

CHAPTER 6

CONCLUSIONS

The ADI method was originally introduced in [127] as a means for solving the heat equation and Poisson's equation on a square domain. When Poisson's equation is discretized on $[-1,1]^2$ using the second-order finite difference operator, it results in a large, banded linear system of equations. The main advantage of ADI at the time was that it reduced this system to two smaller tridiagonal systems, which each could be solved incredibly efficiently via Thomas' algorithm [65]. This is all elementary to us now, but at the time, it was a striking development in computational efficiency. With the advent of the FFT, ADI became no longer useful in this context. However, it has remained an important and active subject of research in other communities (e.g., as a matrix equation solver in reduced order modeling and dynamical systems, as a type of splitting method for solving parabolic PDEs, and as an ADMM scheme in the optimization community).

Early motivation for the work in Chapters 2-4 of this dissertation came from the desire to develop *spectrally* accurate low rank and optimal complexity solvers for elliptic PDEs, first for simple domains like disks and squares, but with the larger notion of efficient schemes for more complicated domains in view. We quickly realized that this was achievable for discretizations that led to model ADI matrix equations. With this observation, the problem should be tackled on two fronts: one can try to develop new spectral discretizations for PDEs that are "ADI-friendly", and one can also try to expand the regimes for which ADI is an effective tool. The former issue is taken up by my research colleague Dan Fortunato in [53]. The latter issue is addressed in Chapters 2-3 of this dissertation. We first tackled the problem of constructing low rank solutions to Sylvester equations with right hand sides that have singular value decay, such as right-hand sides derived from the discretization of smooth 2D functions. This justifies using low rank methods and gives meaningful a priori estimates on the numerical ranks of solutions. It also supplies a more natural method for solving PDEs when the right-hand side is stored in low rank form, such as in the Chebfun software system [45]. The success of the Poisson solvers in [53] hinges on the fact that their discretizations result in Sylvester equations $AX - XB = F$, where the spectral sets of A and B are enclosed on intervals of the real line and optimal ADI shift parameters are known. To expand on this idea and develop solvers for more general elliptic equations of splitting order 2, we require approximate solutions to Zolotarev's third problem for sets in the complex plane. Our work in Chapter 3 is a substantial step in the right direction, though there is much remaining work left to do to turn these largely theoretical results into practical computational tools. We believe that progress on this problem will require continued work in the direction of recent developments in conformal mapping [165]. The fusion of these insights with ideas applied in projection-based iterative methods, such as the rational Krylov subspace method [47, 151], will also be useful. A related and highly challenging set of questions involves developing efficient solvers for generalized Sylvester equations of the form $\sum_{j=1}^{N} A_j X B_j = F$, which arise in ultraspherical discretizations of PDEs of higher splitting ranks [159]. An overview of ideas, approaches, and open questions related to this topic can be found in [151].

A more general theme in our ADI-based work has been the development of compression methods for computing with matrices via their displacement structures. The inspiring work in [19] explains why many important matrices in computational math are of low numerical rank. This idea can be expanded upon and used to explain more complicated compression properties when one observes that solutions to diagonalizable Sylvester matrix equations $AX - XB = F$ can always be expressed in terms of Cauchy matrices of the form $C_{jk} = 1/(\lambda_j(A) - \lambda_k(B))$. The use of these matrices naturally invites analogies with the fast multipole method and related ideas involving the modeling of pairwise interactions between particles. Future work that builds on Chapter 4 looks to develop a more unified framework for understanding the low rank properties of the Cauchy-like matrices that are connected to Toeplitz, Hankel, Vandermonde and related matrices via fast transforms. A related direction of importance for ADI-based compression schemes includes the development of new compression methods and theoretical results for bounding the compressibility of d-dimensional tensors in various formats [149].

Two observations inspired the work in Chapter 5: The first comes from [81], where in devising a scheme for solving Burger's equation, the authors comment on the notion of a "numerical calculus" for working with rational representations of functions. Such a calculus would allow one to adaptively compute efficiently with functions containing algebraic singularities by combining, multiplying, convolving, differentiating, and integrating them using optimal rational representations. Connections between exponential sums, rational functions, and finite dimensional Hankel operator theory lie at the heart of this idea [27, 132]. The introduction of REfit is meant to be a stabilized version of this rational-based numerical calculus.

The second observation that motivated our work in Chapter 5 was the explosion of developments in computational mathematics (for solving PDEs [81, 90], solving nonlinear eigenvalue problems [108], creating model order reduction schemes [98, 140], evaluating functions of matrices [68], constructing conformal maps [9, 165], and more) that have been made possible with data–driven rational approximation methods such as the AAA algorithm and Prony's method. This is largely because these approaches make approximation by rationals accessible, flexible, and automatic. In the context of signal reconstruction, truly nonlinear rational approximation methods have been treated as somewhat exotic in comparison to methods where fixed collections of basis vectors are first selected. Our work makes the many benefits of data-driven rational approximation methods readily available, with schemes that are robust against noise and corruption. We hope this will open up new possibilities for tackling problems in regimes that are currently out of reach in areas such as biomedical monitoring, geophysics, and data-driven computation more generally.

APPENDIX A

COMPLEXITY ANALYSIS FOR ADI-BASED HSS FACTORIZATION

We provide a complexity analysis of the ADI-based construction of the HSS matrix $\widetilde{C}$ from Section 4.4.4 in this section. In Table A.1, we list the cost associated with each step of finding a fADI-based interpolative decomposition $X \approx UX(J,:)$, where X is a submatrix of C of size $m \times \widetilde{n}$, $|J| = r$, and $\widetilde{n} \geq m > r$. Here, we denote by ρ the displacement rank of C. For transformed Toeplitz matrices, $\rho = 2$. For the QR decomposition, we include the costs associated with CPQR (used in practice), rather than SRRQR (which gives stronger theoretical error bounds).

We count the cost for finding approximate HSS row factorizations and note that the cost for the HSS column factorizations is the same. Assume that ϵ is provided and suppose that for all relevant submatrices X, $\operatorname{rank}_\epsilon(X) \leq r$. Suppose that at the finest partition level, the blocks C_v are of size $m \times m$. For each v that is a leaf node of $\mathcal{T}$, about $3mr^2 + 4mr + 3m\rho - 5/3r^3$ flops are required according to Table A.1 to find the pair U_v, J_v in (4.16). In total, there are n/m leaf nodes, so constructing all of the leaf node factorizations costs about η_1 flops, where

$$\eta_1 = 2\frac{n}{m}\left[3mr^2 + 4mr + 3m\rho - 5/3r^3\right]. \tag{A.1}$$

For non-leaf nodes, we replace m in Table A.1 with $2r$ (see Section 4.4.3). The non-leaf node factorizations therefore require about η_2 flops, where

$$\eta_2 = 2\frac{n}{m}\left[\frac{13}{3}r^3 + 8r^2 + 6r\rho\right]. \tag{A.2}$$

It follows that in total, the displacement-based HSS factorization requires about η flops, with

$$\eta = \eta_1 + \eta_2 \approx n\left(6r^2 + 8r + \frac{16}{3}\frac{r^3}{m} + 16\frac{r^2}{m} + 12\rho\frac{r}{m}\right). \tag{A.3}$$

Step	Method	# flops	section
1. Compute Z in $X^{(k)} = ZW^*$	fADI [160]	$4mr + 3m\rho$	sec. 1.4.2
2. Compute R and P in $Z^*P = QR$	CPQR [65, Ch. 6.4]	$2mr^2 - 2r^3/3$	sec. 4.4.4
3. Form U by computing $R_2R_1^{-1}$	Back substitution	$mr^2 - r^3$	sec. 4.4.4

Table A.1: Computational cost for each subroutine used to find $X \approx UX(J, :)$, where X is of size $m \times \widetilde{n}$ and of numerical rank r.

A complexity count for the HSS factorization via randomized sampling is provided in [180]. To compare complexities with the ADI-based method, we treat the oversampling factor [79] required in the randomized method as negligible. Just as with the ADI-based approach, the randomized sampling approach in [180] allows for the use of either CPQR or SRRQR. We assume that both methods are applied using CPQR. We assume the rank of each low rank factorization used in both HSS approximations is at most r. Blocks at the finest partition level are set as size $m = 2r$, and we set $\rho = 2$. The final complexity counts for both methods are given in the first column of Table A.2. In Figure 4.6, we compare the practical performance of the two methods.[1]

To get an overall picture of the complexity of the Toeplitz solver (Algorithm 1), we include in Table A.2 the cost for solving $\widetilde{C}\widetilde{x} = \widetilde{b}$ with the ULV solver in [33], as well as the cost associated with assembling the HSS matrix $\widetilde{C}$. Recall that for each v in $\mathcal{T}$, $\widetilde{C}_v = U_vB_vV_v^*$, with $B_v = C(J_v, K_v)$. In the HSS factorization step, we find and store the indices J_v, K_v, but to use $\widetilde{C}$ in Step 3, we must assemble each B_v, as well as the incompressible diagonal blocks at the finest partition level. This is done efficiently with the identity $C = \mathcal{C} \circ LH^* + C_{diag}$, where $\mathcal{C}_{jk} = 1/(\omega^{2j} - \omega^{2k})$ for $j \neq k$ and $\mathcal{C}_{jj} = 0$. Using this, each B_v and $m \times m$ diagonal block can be populated in about $(1 + 2\rho)r^2$ and $(1 + 2\rho)m^2$ flops, respectively.

[1] Our implementation includes a few additional cost-saving measures not reflected in the above complexity analysis (see Section 4.4.5). These do not impact the asymptotic complexity of the solver.

	HSS factorization	HSS assembly	Solving $\widetilde{C}\widetilde{x} = \widetilde{b}$
Randomized	$(80r\log n + \frac{74}{3}r^2)n$	$15rn$	$42r^2n + 37rn$
ADI-based	$(\frac{26}{3}r^2 + 16r)n$		

Table A.2: The cost for computing $\widetilde{C}$, an HSS approximation to C, is given for two different compression strategies. Here we set $\rho = 2$, and let $m = 2r$ be the size of the blocks at the finest partition level in $\mathcal{T}$, where the $(\epsilon, \mathcal{T})$-rank of C is r. The cost for assembling $\widetilde{C}$ and solving $\widetilde{C}\widetilde{x} = \widetilde{b}$ with the ULV solver described in [33] (see also [180]).

Since our bounds indicate that $r = \mathcal{O}(\rho \log n \log(1/\epsilon))$, the asymptotic complexity of our superfast Toeplitz-like solver is $\mathcal{O}(\rho^2 n \log^2 n \log^2(1/\epsilon))$. Furthermore, as shown in Table A.2, the constants involved are reasonable enough that the solver is efficient, even for moderate-sized n.

BIBLIOGRAPHY

[1] Naum I. Achieser. *Theory of approximation*. Courier Corporation, 2013.

[2] Lars Valerian Ahlfors. *Complex analysis: an introduction to the theory of analytic functions of one complex variable*, volume 2. McGraw-Hill New York, 1966.

[3] Naum I. Akhiezer. *Elements of the theory of elliptic functions*, volume 79. Amer. Math. Soc., 1990.

[4] J.M. Anderson. The Faber operator. In *Rational approximation and interpolation*, pages 1–10. Springer, 1984.

[5] Athanasios C. Antoulas. *Approximation of large-scale dynamical systems*, volume 6. SIAM, 2005.

[6] Athanasios C. Antoulas, Danny C. Sorensen, and Yunkai Zhou. On the decay rate of Hankel singular values and related issues. *Systems & Control Letters*, 46(5):323–342, 2002.

[7] Jared L. Aurentz and Lloyd N. Trefethen. Chopping a Chebyshev series. *ACM Trans. on Math. Soft. (TOMS)*, 43(4):1–21, 2017.

[8] Anthony P. Austin and Kuan Xu. On the numerical stability of the second barycentric formula for trigonometric interpolation in shifted equispaced points. *IMA J. of Num. Anal.*, 37(3):1355–1374, 2017.

[9] Peter J. Baddoo. The AAAtrig algorithm for rational approximation of periodic functions. *arXiv preprint arXiv:2008.05446*, 2020.

[10] Catalin Badea and Bernhard Beckermann. *Spectral Sets*. Chapman and Hall/CRC, Chapter 37 of L. Hogben, Handbook of Linear Algebra, second edition, 2013.

[11] Thomas Bagby. On interpolation by rational functions. *Duke Math. J.*, 36(1):95–104, 1969.

[12] Jonathan Baker, Mark Embree, and John Sabino. Fast singular value decay for Lyapunov solutions with nonnormal coefficients. *SIAM J. Matrix Anal. Appl.*, 36(2):656–668, 2015.

[13] R. Baltensperger. Some results on linear rational trigonometric interpolation. *Comp. & Math. w Appl.*, 43(6-7):737–746, 2002.

[14] Dmitry Batenkov, Benedikt Diederichs, Gil Goldman, and Yosef Yomdin. The spectral properties of Vandermonde matrices with clustered nodes. *Linear Algebra Appl.*, 609:37–72, 2021.

[15] Fermín S.V. Bazán. Conditioning of rectangular Vandermonde matrices with nodes in the unit disk. *SIAM J. Matrix Anal. Appl.*, 21(2):679–693, 2000.

[16] Mario Bebendorf. *Hierarchical matrices*. Springer, 2008.

[17] Bernhard Beckermann. An error analysis for rational Galerkin projection applied to the Sylvester equation. *SIAM J. Numer. Anal.*, 49(6):2430–2450, 2011.

[18] Bernhard Beckermann, Daniel Kressner, and Heather Wilber. Compression properties in rank structured solvers for Toeplitz-like linear systems, 2021. in preparation.

[19] Bernhard Beckermann and Alex Townsend. On the singular values of matrices with displacement structure. *SIAM J. Matrix Anal. Appl.*, 38(4):1227–1248, 2017.

[20] Bernhard Beckermann and Alex Townsend. Bounds on the singular values of matrices with displacement structure. *SIAM Rev.*, 61(2):319–344, 2019.

[21] Peter Benner, Ren-Cang Li, and Ninoslav Truhar. On the ADI method for Sylvester equations. *J. Comput. Appl. Math.*, 233(4):1035–1045, 2009.

[22] Peter Benner and Enrique S. Quintana-Ortí. Solving stable generalized Lyapunov equations with the matrix sign function. *Num. Algorithms*, 20(1):75–100, 1999.

[23] Mario Berljafa and Stefan Güttel. The RKFIT algorithm for nonlinear rational approximation. *SIAM J. Sci. Comput.*, 39(5):A2049–A2071, 2017.

[24] Jean-Paul Berrut, Richard Baltensperger, and Hans D. Mittelmann. Recent developments in barycentric rational interpolation. In *Trends and Appl. in Const. Approx.*, pages 27–51. Springer, 2005.

[25] Jean-Paul Berrut and Lloyd N. Trefethen. barycentric Lagrange interpolation. *SIAM Rev.*, 46(3):501–517, 2004.

[26] J.P. Berrut. Rational functions for guaranteed and experimentally well-conditioned global interpolation. *Comp. Math. Appl.*, 15(1):1–16, 1988.

[27] Gregory Beylkin and Lucas Monzón. On approximation of functions by exponential sums. *Appl. and Comp. Harmonic Analysis*, 19(1):17–48, 2005.

[28] Gregory Beylkin and Lucas Monzón. Nonlinear inversion of a band-limited Fourier transform. *Appl. and Comp. Harmonic Analysis*, 27(3):351–366, 2009.

[29] Dietrich Braess and Wolfgang Hackbusch. Approximation of 1/x by exponential sums in [1,y). *IMA J. of Num. Anal.*, 25(4):685–697, 2005.

[30] Dietrich Braess and Wolfgang Hackbusch. On the efficient computation of high-dimensional integrals and the approximation by exponential sums. In *Multiscale, Nonlinear and Adaptive Approximation*, pages 39–74. Springer, 2009.

[31] Emmanuel J Candès and Carlos Fernandez-Granda. Towards a mathematical theory of super-resolution. *Comm. on Pure and Appl. Math.*, 67(6):906–956, 2014.

[32] Vlastislav Červenỳ, Mikhail M Popov, and Ivan Pšenčík. Computation of wave fields in inhomogeneous media—Gaussian beam approach. *Geophys. J. Int.*, 70(1):109–128, 1982.

[33] S Chandrasekaran, M Gu, X Sun, J Xia, and J Zhu. A superfast algorithm for Toeplitz systems of linear equations. *SIAM J. Matrix Anal. Appl.*, 29(4):1247–1266, 2007.

[34] Shiv Chandrasekaran, Ming Gu, and Timothy Pals. A fast ULV decomposition solver for hierarchically semiseparable representations. *SIAM J. Matrix Anal. Appl.*, 28(3):603–622, 2006.

[35] Hongwei Cheng, Zydrunas Gimbutas, Per-Gunnar Martinsson, and Vladimir Rokhlin. On the compression of low rank matrices. *SIAM J. Sci. Comput.*, 26(4):1389–1404, 2005.

[36] Artur Cichowicz. An automatic s-phase picker. *Bulletin of the Seismological Society of America*, 83(1):180–189, 1993.

[37] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, 19(90):297–301, 1965.

[38] Richard Courant. *Dirichlet's principle, conformal mapping, and minimal surfaces*. Springer, 1977.

[39] Michel Crouzeix and César Palencia. The numerical range is a $(1+\sqrt{2})$-spectral set. *SIAM J. Matrix Anal. Appl.*, 38(2):649–655, 2017.

[40] Anil Damle, Alex Townsend, and Heather Wilber. Data-driven methods for signal processing with rational functions, 2021. in preparation.

[41] Carl De Boor. On calculating with B-splines. *J. of Approx. theory*, 6(1):50–62, 1972.

[42] Carl De Boor. *A practical guide to splines*, volume 27. springer-verlag New York, 1978.

[43] Lokenath Debnath. *Wavelets and signal processing*. Springer Science & Business Media, 2003.

[44] Nadiia Derevianko and Gerlind Plonka. Exact reconstruction of extended exponential sums using rational approximation of their Fourier coefficients. *arXiv preprint arXiv:2103.07743*, 2021.

[45] T. A. Driscoll, N. Hale, and L. N. Trefethen, editors. *Chebfun Guide*. Pafnuty Publications, Oxford, 2014.

[46] Vladimir Druskin, Leonid Knizhnerman, and Valeria Simoncini. Analysis of the rational krylov subspace and ADI methods for solving the Lyapunov equation. *SIAM J. Matrix Anal. Appl.*, 49(5):1875–1898, 2011.

[47] Vladimir Druskin and Valeria Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Systems & Control Letters*, 60(8):546–560, 2011.

[48] Alok Dutt and Vladimir Rokhlin. Fast Fourier transforms for nonequispaced data, II. *Appl. Comput. Harm.Anal.*, 2(1):85–100, 1995.

[49] Nancy S. Ellner and Eugene L. Wachspress. Alternating direction implicit iteration for systems with complex spectra. *SIAM J. Numer. Anal.*, 28(3):859–870, 1991.

[50] Georg Faber. Über polynomische entwickelungen. *Mathematische Annalen*, 57(3):389–408, 1903.

[51] Silviu-Ioan Filip, Yuji Nakatsukasa, Lloyd N. Trefethen, and Bernhard Beckermann. Rational minimax approximation via adaptive barycentric representations. *SIAM J. Sci. Comput.*, 40(4):A2427–A2455, 2018.

[52] Bengt Fornberg and J.A.C. Weideman. A numerical methodology for the Painlevé equations. *J. Comput. Phys.*, 230(15):5957–5973, 2011.

[53] Daniel Fortunato and Alex Townsend. Fast Poisson solvers for spectral methods. *IMA J. Num. Anal.*, 40(3):1994–2018, 2020.

[54] Sándor Fridli, Péter Kovács, Levente Lócsi, and Ferenc Schipp. Rational modeling of multi-lead QRS complexes in ECG signals. In *Annales Univ. Sci. Budapest., Sect. Comp*, volume 37, pages 145–155, 2012.

[55] Sándor Fridli, Levente Lócsi, and Ferenc Schipp. Rational function systems in ECG processing. In *Int. Conf. on Comp. Aided Sys. Theory*, pages 88–95. Springer, 2011.

[56] J. Fuchs. The rectangular Pisarenko method. In *1996 IEEE Int. Conf. on Acoustics, Speech, and Sig. Proc. Conf. Proc.*, volume 5, pages 2495–2498. IEEE, 1996.

[57] Dieter Gaier. *Lectures on complex approximation*, volume 36.2. Springer, 1987.

[58] T Ganelius. Some extremal functions and approximation. In *Fourier analysis and approximation theory. Proceedings of a Colloquium (Budapest)*, pages 371–381, 1976.

[59] Tord Ganelius. *Rational approximation in the complex plane and on the line*. Chalmers Tekniska Högskola/Göteborgs Universitet. Dept. Math., 1975.

[60] Tord Ganelius, DA Brannan, and JG Clunie. Rational functions, capacities and approximation. *Aspects of contemp. complex anal. (Proc. NATO Adv. Study Inst., Univ. Durham)*, pages 409–414, 1979.

[61] Evan S. Gawlik and Yuji Nakatsukasa. Approximating the pth root by composite rational functions. *J. of Approx. Theory*, 266:105577, 2021.

[62] Zoltán Gilián. ECG-based heart beat detection using rational functions. In *Conf. on Dyadic Anal. and Rel. Fields with Appl.*, volume 13, 2014.

[63] Israel Gohberg, Thomas Kailath, and Vadim Olshevsky. Fast Gaussian elimination with partial pivoting for matrices with displacement structure. *Math. of Comp.*, 64(212):1557–1576, 1995.

[64] Gene Golub, Stephen Nash, and Charles Van Loan. A Hessenberg-Schur method for the problem AX+ XB= C. *IEEE Trans. on Automatic Control*, 24(6):909–913, 1979.

[65] Gene H. Golub and Charles F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[66] A.A. Gončar. Zolotarev problems connected with rational functions. *Mathematics of the USSR-Sbornik*, 7(4):623, 1969.

[67] Abinand Gopal and Lloyd N. Trefethen. Solving Laplace problems with corner singularities via rational functions. *SIAM J. Numer. Anal.*, 57(5):2074–2094, 2019.

[68] Ion Victor Gosea and Stefan Güttel. Algorithms for the rational approximation of matrix-valued functions. *arXiv preprint arXiv:2003.06410*, 2020.

[69] Lars Grasedyck. Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure. *Computing*, 72(3):247–265, 2004.

[70] Robert M. Gray et al. Toeplitz and circulant matrices: A review. *Foundations and Trends in Comm. and Inf. Theory*, 2(3):155–239, 2006.

[71] Ulf Grenander and Gabor Szegö. *Toeplitz forms and their applications*. Univ. of California Press, 1958.

[72] Ming Gu and Stanley C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J. Sci. Comput.*, 17(4):848–869, 1996.

[73] Serkan Gugercin, Danny C. Sorensen, and Athanasios C. Antoulas. A modified low-rank Smith method for large-scale Lyapunov equations. *Num. Algorithms*, 32(1):27–55, 2003.

[74] Bjorn Gustavsen and Adam Semlyen. Rational approximation of frequency domain responses by vector fitting. *IEEE Trans. on power deliv.*, 14(3):1052–1061, 1999.

[75] Stefan Guttel, Eric Polizzi, Ping Tak Peter Tang, and Gautier Viaud. Zolotarev quadrature rules and load balancing for the FEAST eigensolver. *SIAM J. Sci. Comput.*, 37(4):A2100–A2122, 2015.

[76] Wolfgang Hackbusch. *Hierarchical matrices: algorithms and analysis*, volume 49. Springer, 2015.

[77] Wolfgang Hackbusch, Boris N. Khoromskij, and Ronald Kriemann. Hierarchical matrices based on a weak admissibility criterion. *Computing*, 73(3):207–243, 2004.

[78] Nicholas Hale and Alex Townsend. Fast and accurate computation of Gauss–Legendre and Gauss–Jacobi quadrature nodes and weights. *SIAM J. Sci. Comput.*, 35(2):A652–A674, 2013.

[79] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.

[80] Fujia Han, Gareth Taylor, and Maozhen Li. Towards a data driven robust event detection technique for smart grids. In *2018 IEEE Power & Energy Soc. Gen. Meeting (PESGM)*, pages 1–5. IEEE, 2018.

[81] Terry Haut, Gregory Beylkin, and Lucas Monzón. Solving Burgers' equation using optimal rational approximations. *Appl. and Comp. Harm. Anal.*, 34(1):83–95, 2013.

[82] T.S. Haut and Gregory Beylkin. Fast and accurate con-eigenvalue algorithm for optimal rational approximations. *SIAM J. Matrix Anal. Appl.*, 33(4):1101–1125, 2012.

[83] Georg Heinig et al. *Algebraic methods for Toeplitz-like matrices and operators*, volume 13. Birkhäuser, 2013.

[84] P. Henrici. *Applied and Computational Complex Analysis*, volume 3. John Wiley and Sons, New York, 1986.

[85] Peter Henrici. Applied and computational complex analysis. *Bull. Amer. Math. Soc*, 84:943–950, 1978.

[86] Peter Henrici. Barycentric formulas for interpolating trigonometric polynomials and their conjugates. *Num. Math.*, 33(2):225–234, 1979.

[87] Nicholas J. Higham. The numerical stability of barycentric Lagrange interpolation. *IMA J. Num. Anal.*, 24(4):547–556, 2004.

[88] Nicholas J Higham. *Functions of matrices: theory and computation*, volume 104. SIAM, 2008.

[89] Kenneth L Ho and Leslie Greengard. A fast direct solver for structured linear systems by recursive skeletonization. *SIAM J. Sci. Comput.*, 34(5):A2507–A2532, 2012.

[90] Clemens Hofreither. A unified view of some numerical methods for fractional diffusion. *Comp. & Math. with Appl.*, 80(2):332–350, 2020.

[91] R. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, UK, 1991.

[92] Roger A. Horn and Fuad Kittaneh. Two applications of a bound on the Hadamard product with a Cauchy matrix. *Electron. J. Linear Algebra*, 3:4–12, 1998.

[93] C. Hu. Algorithm 785: a software package for computing Schwarz–Christoffel conformal transformation for doubly connected polygonal regions. *ACM Trans. Math. Soft.*, 24.3:317–333, 1998.

[94] M-P Istace and J-P Thiran. On the third and fourth Zolotarev problems in the complex plane. *SIAM J. Numer. Anal.*, 32(1):249–259, 1995.

[95] Mohsin Javed. *Algorithms for trigonometric polynomial and rational approximation*. PhD thesis, University of Oxford, 2016.

[96] Thomas Kailath and Ali H. Sayed. Displacement structure: theory and applications. *SIAM Rev.*, 37(3):297–386, 1995.

[97] Hiroo Kanamori, Egill Hauksson, and Thomas Heaton. Real-time seismology and earthquake hazard mitigation. *Nature*, 390(6659):461–464, 1997.

[98] DS Karachalios, Ion Victor Gosea, and Athanasios C Antoulas. The Loewner framework for system identification and reduction. *Handbook on Model Reduction*, 1, 2020.

[99] Steven M. Kay. *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.

[100] Plamen Koev. Matrices with displacement structure–a survey. 1999.

[101] Péter Kovács, Sándor Fridli, and Ferenc Schipp. Generalized rational variable projection with application in ECG compression. *IEEE Trans. on Sig. Proc.*, 68:478–492, 2019.

[102] Péter Kovács and Levente Lócsi. RAIT: the rational approximation and interpolation toolbox for Matlab, with experiments on ECG signals. *Intl. J. of Adv. in Telecommunications, Electrotechnics, Signals and Systems*, 1(2-3):67–75, 2012.

[103] T Kövari and C. Pommerenke. On Faber polynomials and Faber expansions. *Mathematische Zeitschrift*, 99(3):193–206, 1967.

[104] Daniel Kressner and André Uschmajew. On low-rank approximability of solutions to high-dimensional operator equations and eigenvalue problems. *Linear Algebra Appl.*, 493:556–572, 2016.

[105] V.I. Lebedev. On a Zolotarev problem in the method of alternating directions. *USSR Comput. Math. Math.Phys.*, 17(2):58–76, 1977.

[106] Ralph Levy. Generalized rational function approximation in finite intervals using Zolotarev functions. *IEEE Trans. on Microwave Th. and Tech.*, 18(12):1052–1064, 1970.

[107] Jing-Rebecca Li and Jacob White. Low rank solution of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 24(1):260–280, 2002.

[108] Pieter Lietaert, Javier Pérez, Bart Vandereycken, and Karl Meerbergen. Automatic rational approximation and linearization of nonlinear eigenvalue problems. *arXiv preprint arXiv:1801.08622*, 2018.

[109] An Lu and Eugene L. Wachspress. Solution of Lyapunov equations by

alternating direction implicit iteration. *Comp. & Math. with Appl.*, 21(9):43–58, 1991.

[110] Eileen R. Martin. A linear algorithm for ambient seismic noise double beamforming without explicit crosscorrelations. *Geophysics*, 86(1):F1–F8, 2021.

[111] Per-Gunnar Martinsson. A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix. *SIAM J. Matrix Anal. Appl.*, 32(4):1251–1274, 2011.

[112] Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. A fast algorithm for the inversion of general Toeplitz matrices. *Comput. Math. Appl.*, 50(5-6):741–752, 2005.

[113] Stefano Massei, Leonardo Robol, and Daniel Kressner. hm-toolbox: Matlab software for HODLR and HSS matrices. *SIAM J. Sci. Comput.*, 42(2):C43–C68, 2020.

[114] Mathworks. *MATLAB Signal Processing Toolbox: (v R2020b)*. MathWorks, 2020.

[115] Keith Miller. Stabilized numerical analytic prolongation with poles. *SIAM J. on Appl. Math.*, 18(2):346–363, 1970.

[116] Victor Minden, Kenneth L. Ho, Anil Damle, and Lexing Ying. A recursive skeletonization factorization based on strong admissibility. *Multiscale Modeling & Simulation*, 15(2):768–796, 2017.

[117] Ankur Moitra. Super-resolution, extremal functions and the condition number of Vandermonde matrices. In *Proc. of the 47th annual ACM symposium on Theory of Comput.*, pages 821–830. ACM, 2015.

[118] G.B. Moody, R.G. Mark, and A.L. Goldberger. Physionet: A research resource for studies of complex physiologic and biomedical signals. In *Computers in Cardiology 2000. Vol. 27 (Cat. 00CH37163)*, pages 179–182. IEEE, 2000.

[119] Y. Nakatsukasa, O. Sète, and L.N. Trefethen. The AAA algorithm for rational approximation. *SIAM J. Sci. Comput.*, 40(3):A1494–A1522, 2018.

[120] Yuji Nakatsukasa and Roland W Freund. Computing fundamental matrix decompositions accurately via the matrix sign function in two iterations: The power of Zolotarev's functions. *SIAM Rev.*, 58(3):461–493, 2016.

[121] Frank W.J. Olver, Daniel W. Lozier, and Ronald F. Boisvert. *NIST handbook of mathematical functions*. Cambridge University Press, 2010.

[122] Sheehan Olver and Alex Townsend. A fast and well-conditioned spectral method. *SIAM Review*, 55(3):462–489, 2013.

[123] Alan V Oppenheim, John R Buck, and Ronald W Schafer. *Discrete-time signal processing. Vol. 2*. Upper Saddle River, NJ: Prentice Hall, 2001.

[124] Ivan V. Oseledets. Lower bounds for separable approximations of the Hilbert kernel. *Sbornik: Math.*, 198(3):425, 2007.

[125] Ricardo Pachón, Rodrigo B Platte, and Lloyd N Trefethen. Piecewise-smooth chebfuns. *IMA journal of numerical analysis*, 30(4):898–916, 2010.

[126] Davide Palitta and Valeria Simoncini. Numerical methods for large-scale Lyapunov equations with symmetric banded data. *SIAM J. Sci. Comput.*, 40(5):A3581–A3608, 2018.

[127] Donald W. Peaceman and Henry H. Rachford, Jr. The numerical solution of parabolic and elliptic differential equations. *J. Soc. for ind. Appl. Math.*, 3(1):28–41, 1955.

[128] Thilo Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM J. Sci. Comput.*, 21(4):1401–1418, 1999.

[129] Thomas Peter and Gerlind Plonka. A generalized Prony method for reconstruction of sparse sums of eigenfunctions of linear operators. *Inverse Problems*, 29(2):025001, 2013.

[130] Ping Tak Peter Tang and Eric Polizzi. FEAST as a subspace iteration eigensolver accelerated by approximate spectral projection. *SIAM J. Matrix Anal. Appl.*, 35(2):354–390, 2014.

[131] Vladilen F. Pisarenko. The retrieval of harmonics from a covariance function. *Geophys. J. Int.*, 33(3):347–366, 1973.

[132] Vlada Pototskaia and Gerlind Plonka. Application of the AAK theory and Prony-like methods for sparse approximation of exponential sums. *PAMM*, 17(1):835–836, 2017.

[133] Daniel Potts and Manfred Tasche. Parameter estimation for exponential sums by approximate Prony method. *Sig. Proc.*, 90(5):1631–1642, 2010.

[134] Daniel Potts and Manfred Tasche. Nonlinear approximation by sums of nonincreasing exponentials. *Applicable Anal.*, 90(3-4):609–626, 2011.

[135] Daniel Potts and Manfred Tasche. Parameter estimation for nonincreasing exponential sums by Prony-like methods. *Linear Algebra Appl.*, 439(4):1024–1039, 2013.

[136] G.R. Prony. Essai experimental et analytique. *J. de l'Ecole Polytechnique*, 2, 1795.

[137] Johann Radon. Über die randwertaufgaben beim logarithmischen potential. *Sitzber. Akad. Wiss. Wien*, 128(7):1123–1167, 1919.

[138] Norman Ricker. *Transient waves in visco-elastic media*, volume 10. Elsevier, 2012.

[139] Shmuel Rippa. An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Adv. in Comp. Math.*, 11(2):193–210, 1999.

[140] Andrea Carracedo Rodriguez and Serkan Gugercin. The p-AAA algorithm for data driven modeling of parametric dynamical systems. *arXiv preprint arXiv:2003.06536*, 2020.

[141] Daniel Rubin, Alex Townsend, and Heather Wilber. Bounding Zolotarev numbers using Faber rational functions. *arXiv preprint arXiv:1911.11882*, 2020.

[142] Axel Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra and its Appl.*, 58:391–405, 1984.

[143] John Sabino. *Solution of large-scale Lyapunov equations via the block modified Smith method*. PhD thesis, Rice University, 2007.

[144] E.B. Saff. Logarithmic potential theory with applications to approximation theory. *Surveys in Approximation Theory*, 5:165–200, 2010.

[145] Edward B Saff and Vilmos Totik. *Logarithmic potentials with external fields*, volume 316. Springer Science & Business Media, 2013.

[146] M Schiffer. Some recent developments in the theory of conformal mapping. *Appendix to: R. Courant, Dirichlet's principle, conformal mapping and minimal surfaces.*, 1950.

[147] Issai Schur. On Faber polynomials. *Amer. J. Math.*, 67(1):33–41, 1945.

[148] Stephen David Shank. *Low-rank Solution Methods for Large-scale Linear Matrix Equations*. PhD thesis, Temple University, 2014.

[149] Tianyi Shi and Alex Townsend. On the compressibility of tensors. *SIAM J. Matrix Anal. Appl.*, 42(1):275–298, 2021.

[150] Valeria Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM J. Sci. Comput.*, 29(3):1268–1288, 2007.

[151] Valeria Simoncini. Computational methods for linear matrix equations. *SIAM Rev.*, 58(3):377–441, 2016.

[152] Laurence C. Smith, Donald L. Turcotte, and Bryan L. Isacks. Stream flow characterization and feature detection using a discrete wavelet transform. *Hydrological processes*, 12(2):233–249, 1998.

[153] R.A. Smith. Matrix equation XA+BX=C. *SIAM J. Appl. Math.*, 16(1):198–201, 1968.

[154] Gerhard Starke. Near-circularity for the rational Zolotarev problem in the complex plane. *Journal of approx. theory*, 70(1):115–130, 1992.

[155] Elias M. Stein and Rami Shakarchi. *Complex analysis*, volume 2. Princeton University Press, 2010.

[156] Gilbert W Stewart. *Matrix perturbation theory*. Citeseer, 1990.

[157] John Todd. Applications of transformation theory: A legacy from Zolotarev (1847–1878). In *Approximation theory and spline functions*, pages 207–245. Springer, 1984.

[158] Alex Townsend. *Computing with functions in two dimensions*. PhD thesis, University of Oxford, 2014.

[159] Alex Townsend and Sheehan Olver. The automatic solution of partial differential equations using a global spectral method. *J. Comput. Phys.*, 299:106–123, 2015.

[160] Alex Townsend and Heather Wilber. On the singular values of matrices with high displacement rank. *Linear Algebra Appl.*, 548:19–41, 2018.

[161] Alex Townsend, Heather Wilber, and Grady B. Wright. Computing with functions in spherical and polar geometries I. the sphere. *SIAM J. Sci. Comput.*, 38(4):C403–C425, 2016.

[162] M. K. Transtrum, B. B Machta, and J. P. Sethna. Why are nonlinear fits to data so challenging? *Physical rev. letters*, 104(6):060201, 2010.

[163] L. N. Trefethen. *Approximation Theory and Approximation Practice*. SIAM, 2013.

[164] Lloyd N Trefethen. Computing numerically with functions instead of numbers. *Math. in Comp. Sci.*, 1(1):9–19, 2007.

[165] Lloyd N. Trefethen. Numerical conformal mapping with rational functions. *Comp. Methods and Func.Theory*, 20(3):369–387, 2020.

[166] Lloyd N Trefethen, Yuji Nakatsukasa, and J.A.C. Weideman. Exponential

node clustering at singularities for rational approximation, quadrature, and PDEs. *Numer. Math.*, 147(1):227–254, 2021.

[167] F. William Trench. Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices. *SIAM J. Matrix Anal. Appl.*, 10(2):135–146, 1989.

[168] Eugene Tyrtyshnikov. Mosaic-skeleton approximations. *Calcolo*, 33(1-2):47–57, 1996.

[169] Michael A. Unser and Thierry Blu. Wavelets and radial basis functions: A unifying perspective. In *Wavelet Appl. in Sig. and Im. Proc. VIII*, volume 4119, pages 487–493. Int. Soc. for Optics and Photonics, 2000.

[170] Martin Vetterli, Pina Marziliano, and Thierry Blu. Sampling signals with finite rate of innovation. *IEEE trans. on Sig. Proc.*, 50(6):1417–1428, 2002.

[171] Eugene L. Wachspress. Optimum alternating-direction-implicit iteration parameters for a model problem. *J. Soc. for Ind. and Appl. Math.*, 10(2):339–350, 1962.

[172] J.L. Walsh. Hyperbolic capacity and interpolating rational functions. *Duke Math. J.*, 32(3):369–379, 1965.

[173] Holger Wendland. *Scattered data approximation*, volume 17. Cambridge university press, 2004.

[174] Harold Widom. Rational approximation and n-dimensional diameter. *J. Approx. Theory*, 5(4):343–361, 1972.

[175] Heather Wilber. freeLYAP for matrix equations in MATLAB: Iterative solvers package. `https://github.com/ajt60gaibb/freeLYAP`, 2018–2020.

[176] Heather Wilber, Alex Townsend, and Grady B. Wright. Computing with functions in spherical and polar geometries II. the disk. *SIAM J. Sci. Comput.*, 39(3):C238–C262, 2017.

[177] Grady B. Wright, Mohsin Javed, Hadrien Montanelli, and Lloyd N. Trefethen. Extension of Chebfun to periodic functions. *SIAM J. Sci. Comput.*, 37(5):C554–C573, 2015.

[178] Yuanzhe Xi and Yousef Saad. Computing partial spectra with least-squares rational filters. *SIAM J. Sci. Comput.*, 38(5):A3020–A3045, 2016.

[179] Jianlin Xia, Shivkumar Chandrasekaran, Ming Gu, and Xiaoye S. Li. Fast algorithms for hierarchically semiseparable matrices. *Linear Algebra Appl.*, 17(6):953–976, 2010.

[180] Jianlin Xia, Yuanzhe Xi, and Ming Gu. A superfast structured solver for Toeplitz linear systems via randomized sampling. *SIAM J. Matrix Anal. Appl.*, 33(3):837–858, 2012.

[181] Xin Ye, Jianlin Xia, and Lexing Ying. Analytical low-rank compression via proxy point selection. *SIAM J. Matrix Anal. Appl.*, 41(3):1059–1085, 2020.

[182] E.E. Zolotarev. Application of elliptic functions to questions of functions deviating least and most from zero. *Zap. Imp. Akad. Nauk. St. Petersburg*, 30(5):1–59, 1877.