©Copyright 2026
Cade Ballew

Computational approximation theory on higher-genus domains

Cade Ballew

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:

Thomas Trogdon, Chair

Bernard Deconinck

Heather Wilber

Program Authorized to Offer Degree:

Applied Mathematics

University of Washington

Abstract

Computational approximation theory on higher-genus domains

Cade Ballew

Chair of the Supervisory Committee:
Thomas Trogdon
Applied Mathematics

We develop computational approximation theory on higher-genus domains, finite unions of disjoint subintervals of the real axis that arise naturally in orthogonal polynomial theory, iterative numerical linear algebra, and the study of integrable nonlinear wave equations. In each case, exploiting this underlying structure to choose appropriate basis functions (orthogonal polynomials) is essential to obtain a fast and efficient numerical method. A unifying theme is the numerical solution of Riemann–Hilbert problems on these domains.

First, we develop a numerical framework and apply it to compute orthogonal polynomials. For Chebyshev-like weight functions supported on a union of disjoint intervals, we use the Fokas–Its–Kitaev Riemann–Hilbert problem [70] to produce an algorithm that computes recurrence coefficients and allows for the pointwise evaluation of Cauchy transforms in optimal (O(1)) complexity.

Using these orthogonal polynomials, we develop the Akhiezer iteration, a new iterative method for solving indefinite linear systems and computing matrix functions. The iteration uses orthogonal polynomial recurrence coefficients to efficiently compute the action of a matrix polynomial on a vector. It is inner-product-free, features an a priori computable convergence rate, and is often faster in practice than standard Krylov subspace methods. A particular extension yields a fast inverse-free iteration for solving large-scale Sylvester matrix equations.

Finally, we present a class of numerical methods for computing and evolving soliton gasses, ensembles of a very large or infinite number of interacting solitons, in the context of the Korteweg–de Vries and modified Korteweg–de Vries equations. While traditionally difficult to simulate due to ill-conditioning, the methods are based on the Riemann–Hilbert formulation of [61] and the analysis of [78, 79]. By using basis functions that encode the correct endpoint asymptotics, this approach enables the first accurate simulation of such solutions in standard (double) precision arithmetic. This allows for fast, pointwise evaluation before asymptotic estimates are valid.

TABLE OF CONTENTS

Page

List of Figures iv

List of Tables vii

Glossary viii

Chapter 1: Introduction 1
1.1 Orthogonal polynomials on higher-genus domains 1
1.2 Iterative methods for cut spectra 2
1.3 Soliton gasses and the KdV equation 3
1.4 Open problems 4

Part I: Orthogonal polynomials 6

Chapter 2: Approximation theory on higher-genus domains 7
2.1 Introduction 8
2.2 Orthogonal polynomials and Cauchy integrals 9
2.3 Potential theory and orthogonal polynomial series 14
2.4 Classical algorithms for recurrence coefficients 18
2.5 The Single-Interval Case: Chebyshev Polynomials 20
2.6 Jacobi Elliptic and Theta Functions 26
2.7 The Two-Interval Case: Akhiezer Polynomials 28
2.8 The Multi-Interval Case: Genus $g \geq 2$ 31

Chapter 3: Riemann–Hilbert problems on higher-genus domains 33
3.1 Riemann–Hilbert problems 34
3.2 The FIK and soliton gas RHPs 36
3.3 Numerical methods 38

Chapter 4: Computing orthogonal polynomials on higher-genus domains 45
4.1 Introduction 46
4.2 Transforming the Riemann–Hilbert problem 47
4.3 Recurrence coefficients 55
4.4 Special simplifying weight functions 57
4.5 Numerical implementation 59
4.6 Solving the deformed FIK problem 64
4.7 Examples and applications 66

Part II: Numerical linear algebra 79

Chapter 5: The Akhiezer iteration 80
5.1 Introduction 81
5.2 The Akhiezer iteration and matrix function evaluation 83
5.3 The Akhiezer iteration for matrix function approximation 87
5.4 Adaptivity 93
5.5 Examples and Applications 94
5.6 Discussion and open questions 99

Chapter 6: Sylvester matrix equations 101
6.1 Introduction 102
6.2 Method 1: the matrix sign function 109
6.3 Method 2: inverting the Sylvester operator 121
6.4 Examples and applications 130
6.5 Other applications 135
6.6 Discussion and future work 138
6.7 Bivariate matrix functions 139

Part III: The KdV equation 147

Chapter 7: Soliton gasses 148
7.1 Introduction 149
7.2 Computing soliton gasses 157
7.3 Nonlinear superposition of a pure soliton gas with solitons 162

7.4 Examples, demonstration of convergence, and performance 169
7.5 Conclusion 171

Chapter 8: Long-time computations of KdV and mKdV soliton gasses 173
8.1 Introduction 174
8.2 Steepest descent deformations 177
8.3 Numerical implementation 181
8.4 Modifications for KdV 188
8.5 Examples and demonstration of convergence 192

Chapter 9: Large-genus KdV solutions 195
9.1 Introduction 196
9.2 The Riemann–Hilbert problem 196
9.3 Diagonalization and preconditioning 198
9.4 Dispersive quantization 199

Bibliography 201

Appendix A: Deferred proofs 215
A.1 Akhiezer polynomial recurrence coefficients 215
A.2 Proof of Theorem 2.5.1 218
A.3 Proof of Lemma 6.2.3 219
A.4 Proof of Theorem 6.7.2 220

Appendix B: Computing the modulation parameter and regions 221

LIST OF FIGURES

Figure Number		Page
3.1	An oriented contour through an interior point	34
4.1	Example weight function of Chebyshev type on four disjoint intervals	46
4.2	Sample contours and regions for a three-interval problem	48
4.3	Jump conditions in the three-interval RHP	55
4.4	Effect of a smooth modification $h(x)$ on recurrence coefficients, one interval	67
4.5	Convergence of recurrence coefficients vs. collocation points, one interval	68
4.6	High-degree recurrence coefficient accuracy, one interval	69
4.7	Convergence of recurrence coefficients vs. collocation points, two intervals	70
4.8	Timing comparison: RHP method vs. RKPW for recurrence coefficients	71
4.9	Convergence of recurrence coefficients vs. collocation points, three intervals	72
4.10	Convergence of recurrence coefficients vs. collocation points, four intervals	73
4.11	Toda lattice evolution with Chebyshev-U initial data	75
4.12	Recurrence coefficients for a Chebyshev-$\underline{T}$-like weight on two intervals	75
4.13	Toda lattice evolution with Chebyshev-$\underline{T}$-like initial data on two intervals	76
4.14	Evolution of individual Toda lattice matrix elements	76
4.15	Recurrence coefficients for a Chebyshev-W-like weight on three intervals	77
4.16	Toda lattice evolution with Chebyshev-W-like initial data on three intervals	77
5.1	Eigenvalue histogram and Akhiezer iteration convergence	83
5.2	Akhiezer iteration convergence, uniform eigenvalues	95
5.3	Eigenvalue contour plot and Akhiezer iteration convergence	96
5.4	Matrix function convergence for exponential function	96
5.5	Matrix function convergence, randomly generated eigenvalues	97
5.6	Matrix function convergence, fixed eigenvalues	97
5.7	Convergence comparison, 2 vs. 3 spectral bands	98
5.8	Akhiezer iteration convergence, preconditioned BVP	99
6.1	Convergence rate bounds for polynomial Krylov, Method 1, and Method 2	104

6.2 Akhiezer polynomial vs. Zolotarev approximation errors for the sign function 107
6.3 Generalized Bernstein ellipses for two cut domains 108
6.4 Rank and error of low-rank solver iterates, unweighted compression 115
6.5 Rank and error of low-rank solver iterates, weighted compression 116
6.6 Akhiezer series coefficient decay and convergence rate heuristic 117
6.7 Convergence of Method 1 for a rank-2 Sylvester equation 119
6.8 Convergence of Method 2 with single- and two-interval Akhiezer polynomials 124
6.9 Rank and error of inverse-method iterates, unweighted and weighted compression 127
6.10 Convergence of Method 2 against the heuristic bound 129
6.11 Timing comparisons of Methods 1 and 2 against Bartels–Stewart and Krylov methods 131
6.12 Timing comparisons of Method 2 vs. block polynomial Krylov 133
6.13 Method 2 convergence and solution for a coupled integral equation 134
6.14 Solution of a coupled integral equation with preconditioned GMRES 135
6.15 Convergence for Fréchet derivative approximation of sign and exponential . . 137
6.16 Convergence of the bivariate Akhiezer iteration on three problems 146

7.1 KdV soliton gas on five band pairs with five superimposed solitons 150
7.2 Pure KdV soliton gas on five band pairs 152
7.3 KdV soliton gas near the wedge region with two solitons 153
7.4 Pure KdV soliton gas on two band pairs with ray plot 154
7.5 Sign of Reφ in the complex plane for the unmodulated region 158
7.6 Jump contours and matrices for $\mathbf{S}$ 160
7.7 Jump contours and matrices for $\mathbf{R}$ 163
7.8 KdV soliton gas on two band pairs with three superimposed solitons 165
7.9 KdV soliton gas on one band pair with one soliton, $\kappa_1 = 3$ 166
7.10 KdV soliton gas on one band pair with one soliton, $\kappa_1 = 1$ 167
7.11 KdV soliton gas on five band pairs with variable reflection coefficients 168
7.12 Pointwise errors vs. collocation points per interval 171

8.1 Long-time KdV soliton gas on five band pairs with five solitons 175
8.2 KdV soliton gas through the wedge region with two solitons 176
8.3 Deformation regions of the (x, t)-plane 177
8.4 Jump contours and matrices for $\mathbf{Z}$ 178

8.5 Local definitions of $\widetilde{\mathbf{Z}}$ . . . 179
8.6 Jump contours and matrices for $\widetilde{\mathbf{Z}}$ . . . 179
8.7 Jump contours and matrices for $\mathbf{S}$ . . . 180
8.8 Jump contours and matrices for $\mathbf{\Delta}$ . . . 181
8.9 Jump contours and matrices for $\widetilde{\mathbf{S}}$ . . . 182
8.10 Jump contours and matrices for $\widetilde{\mathbf{T}}$ . . . 186
8.11 mKdV soliton and anti-soliton gas . . . 189
8.12 KdV soliton gas on two band pairs with one soliton . . . 192
8.13 Maximum pointwise error in computing $\mathbf{\Delta}(z)$ . . . 193

9.1 Jump contour for the finite-genus KdV problem . . . 197
9.2 Large-genus approximation of dispersive quantization . . . 200

LIST OF TABLES

Table Number Page

6.1 Runtimes for inverse-free methods vs. polynomial Krylov (memory comparison) 130

7.1 Pointwise evaluation runtimes for computed solutions 172

GLOSSARY

$\diamond$	An independent variable, e.g., $1/\diamond$ denotes the function $f(x) = 1/x$.
$\mathbb{1}_S(\diamond)$	The indicator function of a set S: equals 1 if $\diamond \in S$ and 0 otherwise.
δ_{jk}	The Kronecker delta: equals 1 if $j = k$ and 0 otherwise.
$\mathrm{cl}(\diamond)$	The closure of a set.
$\|\diamond\|_2$	The vector 2-norm and the corresponding induced matrix norm.
$\|\diamond\|_\infty$	The vector ∞-norm (largest absolute entry) and the corresponding induced matrix norm (largest absolute row sum).
$\|\diamond\|_F$	The Frobenius norm: $\|\mathbf{M}\|_F = \sqrt{\sum_{j,k} \|\mathbf{M}_{j,k}\|^2}$.
$\mathbf{M}_{j,k}, \mathbf{M}_{:,k}, \mathbf{M}_{j,:}$	The (j,k) entry, kth column, and jth row of a matrix $\mathbf{M}$, respectively.
$\sigma(\mathbf{M})$	The spectrum (set of eigenvalues) of a matrix or operator $\mathbf{M}$.
$\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_3$	The Pauli matrices $\boldsymbol{\sigma}_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\boldsymbol{\sigma}_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$.
$\mathcal{C}_\Gamma$	The Cauchy integral over a curve Γ, $\mathcal{C}_\Gamma[f](z) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{f(x)}{x-z} \mathrm{d}x$.
$\mathcal{C}_\Gamma^\pm$	The boundary values of $\mathcal{C}_\Gamma$.
$\mathcal{S}_\Gamma$	The Stieltjes transform over a curve Γ, $\mathcal{S}_\Gamma[f](z) = \int_\Gamma \frac{f(x)}{x-z} \mathrm{d}x$.
g	The genus of the higher-genus domain.
Σ	A disjoint union of real intervals, $\Sigma = \bigcup_{j=1}^{g+1} [a_j, b_j] \subset \mathbb{R}$.
Σ_+, Σ_-	Collections of positive and negative real intervals: $\Sigma_+ = \bigcup_{j=1}^{g} [a_j, b_j]$, $\Sigma_- = \bigcup_{j=1}^{g} [-b_j, -a_j]$.
w	A weight function with support Σ.
$L^2_w(\Sigma)$	The weighted L^2 space on Σ with weight w.
$\mathbf{J}(w)$	The Jacobi operator corresponding to a weight function w.
$\breve{a}_k(w), \breve{b}_k(w)$	Three-term recurrence coefficients: $\breve{a}_k \in \mathbb{R}$ (diagonal) and $\breve{b}_k > 0$ (off-diagonal) entries of the Jacobi operator $\mathbf{J}(w)$.

$p_k(\diamond; w), \pi_k(\diamond; w)$	The orthonormal and monic orthogonal polynomials corresponding to w on Σ.
T_k, U_k, V_k, W_k	Chebyshev polynomials of the first, second, third, and fourth kinds. We use $\underline{T}_k$ to denote orthonormal Chebyshev polynomials of the first kind.
$P_n^{(\alpha,\beta)}$	Orthonormal Jacobi polynomial of degree n (different from the classical normalization); weight $w^{(\alpha,\beta)}(x) \propto (1-x)^\alpha(1+x)^\beta$ on $[-1,1]$.
$\mathfrak{g}_\Sigma(z)$	The g-function for Σ (2.10): its real part $G_\Sigma(z) = \mathrm{Re}\mathfrak{g}_\Sigma(z)$ is the exterior Green's function of Σ with pole at infinity, and $\Phi_\Sigma(z) = \mathrm{e}^{\mathfrak{g}_\Sigma(z)}$ is a complex exterior Green's function.
$\Phi_\Sigma(z)$	A conformal map $\mathrm{e}^{\mathfrak{g}_\Sigma(z)}$, mapping $\mathbb{C} \setminus \Sigma$ to the exterior of the unit disk.
$\mathfrak{c}$	The logarithmic capacity of Σ: $\mathfrak{c} = \lim_{z\to\infty} \mathrm{e}^{\mathfrak{g}_\Sigma(z)}/z$.
$\Gamma_\varrho(\Sigma)$	The level curve $\{z \in \mathbb{C} : \lvert\Phi_\Sigma(z)\rvert = \varrho\}$ for $\varrho > 1$ (2.8); a finite union of smooth closed loops enclosing Σ.
$\nu(z; \mathbf{A})$	The convergence parameter $\max_{\lambda\in\sigma(\mathbf{A})} \mathrm{Re}\mathfrak{g}_\Sigma(\lambda) - \mathrm{Re}\mathfrak{g}_\Sigma(z)$ for the Akhiezer iteration.
$J_+^{-1}(z)$	The right inverse of the Joukowsky map: $J_+^{-1}(z) = z - \sqrt{z-1}\sqrt{z+1}$, mapping $\mathbb{C} \setminus [-1,1]$ to the interior of the unit disk.
$K(k)$	The complete elliptic integral of the first kind with modulus k; $k' = \sqrt{1-k^2}$ is the complementary modulus.
$q(k)$	The nome $q = \mathrm{e}^{-\pi K(k')/K(k)} \in (0,1)$ (2.19).
$\mathrm{sn}, \mathrm{cn}, \mathrm{dn}$	Jacobi elliptic functions $\mathrm{sn}(u,k)$, $\mathrm{cn}(u,k)$, $\mathrm{dn}(u,k)$ with modulus k.
$\theta_1, \theta_2, \theta_3, \theta_4$	Jacobi theta functions.

ACKNOWLEDGMENTS

First and foremost, I thank my advisor, Tom Trogdon, for his support and willingness to spend hours discussing math. My other collaborators, Deniz Bilman and Heather Wilber, gave their time and effort generously, and without them, much of this work would not have been possible. Bernard Deconinck provided valuable feedback on this thesis and welcomed me into his research group meetings. Others who served on my committee at various stages, Bamdad Hosseini, Jingwei Hu, and Rekha Thomas, have my gratitude as well.

The impact of my friends and family throughout this journey has been immeasurable, and fully describing it would require a dissertation in and of itself. A few groups that I want to single out are my high school gaming friends (David, Kevin, Reece, and Young), The MOB's low brass section (Christen, Ricardo, Roberto, Ryan, and Ryan), and my current officemates (Alex, Alex, Charbel, James, and Juan).

Finally, Eric Shea-Brown deserves special thanks for enabling a hospitable working environment during hot summer months.

DEDICATION

To Jase, Evan, and Kinsey.

Chapter 1

INTRODUCTION

For our purposes, a *higher-genus domain* will consist of a finite union of disjoint closed real intervals,

$$\Sigma = \bigcup_{j=1}^{g+1} [a_j, b_j], \qquad a_1 < b_1 < a_2 < \cdots < b_{g+1}. \tag{1.1}$$

Such domains arise naturally in orthogonal polynomial theory, iterative numerical linear algebra, and the study of integrable nonlinear wave equations. In each of these settings a common set of tools, namely Riemann–Hilbert problems (RHPs) and their numerical solution, can be brought to bear.

The phrase "higher-genus" comes from the associated algebraic curve. Defining $R(z)^2 = \prod_{j=1}^{g+1}(z-a_j)(z-b_j)$, the curve $\{(z,w) \in \mathbb{C}^2 : w^2 = R(z)^2\}$ is, after smooth compactification, a Riemann surface of genus g, and the analytic quantities naturally attached to Σ live on this surface. For $g = 0$, the surface has genus zero and these quantities reduce to classical objects, e.g., the Chebyshev polynomials. For $g = 1$, the quantities are still classical but involve elliptic function theory. For $g \geq 2$, the surface is hyperelliptic and these quantities involve Riemann theta functions and the Abel map, which are much more expensive to evaluate. The central computational challenge of this thesis is to work in the regime $g \geq 2$ without paying this cost.

The strategy throughout is to represent the quantities of interest as solutions of RHPs and solve those RHPs numerically. After appropriate contour deformations, the RHP reduces to a well-conditioned singular integral equation on a system of intervals and circles, which can be discretized at high order by encoding the expected endpoint singularities of the solution directly into the approximation basis. This approach, developed systematically in the first part of this thesis, is then applied in all three settings considered here. We note that while this numerical method is observed to converge spectrally in all cases considered here, the only rigorous convergence proof available applies to the simplest case of a single circle [156]; convergence for the multi-component problems that arise throughout this thesis remains open.

1.1 Orthogonal polynomials on higher-genus domains

Every sequence of real-line orthogonal polynomials satisfies a three-term recurrence relation, and the recurrence coefficients encode the support of the orthogonality measure. For a measure supported on a single interval, the recurrence coefficients (sequences indexed by the

polynomial degree n) converge as $n \to \infty$ to constants determined by the endpoints of the support. For a measure supported on a higher-genus domain Σ, the coefficients are instead asymptotically quasiperiodic, with behavior determined by, among other things, the genus-g Riemann surface corresponding to Σ. This quasiperiodicity makes efficient computation difficult: existing methods require at least $\mathrm{O}(N^2)$ operations to produce N coefficients, and the known asymptotic formulae involve Riemann theta functions that are expensive to evaluate for large g.

Chapter 2 collects the background material used throughout the thesis. In addition to the basics of orthogonal polynomials and their recurrence relations, it covers the potential theory of Σ and the elliptic and theta function machinery that appears in the genus-one case. The chapter also introduces the four varieties of Chebyshev polynomials and their exact Cauchy transforms, which play a concrete role in the numerical methods of subsequent chapters.

Chapter 3 develops the numerical framework for solving RHPs used throughout this thesis. The key observation is that any RHP whose solution has Chebyshev-type endpoint singularities can be recast as a linear system for a smooth density, provided the Chebyshev weight is factored into the basis. In later chapters, we apply this idea to two RHPs: one that characterizes orthogonal polynomial data on Σ, and one that defines a KdV soliton gas. In both cases, the contour consists of a disjoint union of intervals and circles, and the linear system is assembled as a block system of smaller problems.

Chapter 4 uses this framework to compute orthogonal polynomial recurrence coefficients in optimal complexity. The starting point is the Fokas–Its–Kitaev (FIK) RHP whose solution encodes the orthogonal polynomials [70]. Direct numerical solution of this problem is complicated by singularities at infinity and at the endpoints of Σ, so a partial nonlinear steepest-descent transformation is applied first. The result is an $\mathrm{O}(N)$ algorithm for the first N recurrence coefficients (and $\mathrm{O}(1)$ for any individual coefficient) that avoids the computation of theta functions.

1.2 Iterative methods for cut spectra

One approach to polynomial iterative methods for linear systems is to approximate the function $f(x) = 1/x$ by a polynomial on the spectrum of the coefficient matrix. The quality of the approximation, and hence the convergence rate of the method, depends on how well a polynomial of given degree can approximate f on the spectrum while remaining small at zero. When the spectrum is a single interval $[a, b]$ not containing zero, this produces a modification of the classical Chebyshev iteration and converges geometrically at a rate determined by the ratio of the distance from zero to $[a, b]$ versus the length of $[a, b]$.

When the spectrum has multiple components, one can do better by using polynomials that are small on the entire spectrum rather than on a single enclosing interval. The *Akhiezer polynomials* [6] are orthogonal polynomials on Σ with respect to a specific weight function. While their expressions are more complicated than the Chebyshev polynomials, they satisfy the same three-term recurrence structure and can be computed using the methods of Chap-

ter 2 for $g = 1$ and Chapter 4 for $g \geq 2$. Their convergence rate is governed by the exterior Green's function with pole at infinity G_Σ for the complement of Σ, which is also accessible from the RHP framework.

Chapter 5 introduces the *Akhiezer iteration*, the analog of the Chebyshev iteration built on Akhiezer polynomials. The method is inner-product-free, requires only matrix-vector products with $\mathbf{A}$, and achieves geometric convergence at a rate determined a priori by G_Σ. The chapter also extends the method to the evaluation of matrix functions $f(\mathbf{A})\mathbf{b}$ for analytic f, using a contour integral representation and the same polynomial recurrences.

Chapter 6 extends the Akhiezer iteration to Sylvester matrix equations

$$\mathbf{XA} - \mathbf{BX} = \mathbf{C}, \quad \mathbf{X}, \mathbf{C} \in \mathbb{C}^{m\times n}, \quad \mathbf{A} \in \mathbb{C}^{n\times n}, \quad \mathbf{B} \in \mathbb{C}^{m\times m}, \tag{1.2}$$

where $\mathbf{X}$ is the unknown and the spectra of $\mathbf{A}$ and $\mathbf{B}$ lie in separated unions of intervals. Two methods are introduced. The first uses the Akhiezer iteration to approximate the matrix sign function applied to a block embedding of $\mathbf{A}$ and $\mathbf{B}$, then extracts the solution $\mathbf{X}$ from the result. The second directly approximates the inverse of the Sylvester operator as a polynomial series. Both achieve sub-cubic arithmetic complexity and sub-quadratic storage when $\mathbf{C}$ is low rank, with convergence rates determined a priori and without growing orthogonalization costs.

1.3 Soliton gasses and the KdV equation

The Korteweg–de Vries (KdV) equation,

$$u_t + 6uu_x + u_{xxx} = 0, \tag{1.3}$$

describes the behavior of long waves in shallow water and is the prototypical example of an "integrable" (exactly solvable) nonlinear PDE. The equation features solitons, a special class of spatially localized solitary waves. In 1971, Vladimir Zakharov introduced the concept of soliton gasses, ensembles of a very large or infinite number of interacting solitons [166]. They have emerged as critical models in mathematical physics, with applications spanning oceanography [128], fiber optics [145], and plasma dynamics [142]. A certain class of soliton gasses can be defined rigorously by an RHP whose jump contour consists of intervals on the imaginary axis, with a jump matrix determined by an accumulation density r_1 [78].

The connection to the methods developed in Chapters 3–4 is direct. The accumulation density r_1 appearing in the soliton gas RHP is assumed to have the same Chebyshev-like endpoint behavior as the weight functions treated in Chapter 4. After a nonlinear steepest-descent transformation analogous to the one in Chapter 4, the soliton gas RHP reduces to the same type of well-conditioned singular integral equation on intervals and circles, and can be solved using the collocation framework of Chapter 3. The (focusing) modified KdV (mKdV) equation,

$$u_t + 6u^2u_x + u_{xxx} = 0, \tag{1.4}$$

admits an analogous soliton gas theory, and the same methods apply.

Chapter 7 develops this approach for small to moderate times [62,78]. In the region of the (x,t)-plane where the steepest-descent deformation is well-conditioned, we produce the first effective implementation of soliton gasses in standard (double) precision arithmetic [145]. The chapter also treats nonlinear superpositions of a soliton gas with a finite number of discrete solitons.

Chapter 8 extends the method to the entire (x,t)-plane. At large times, the deformation used in Chapter 7 breaks down in wedge-shaped sectors of the (x,t)-plane where the solution undergoes a transition between behaviors of different genus. Inside such a wedge, the effective contour of the RHP must be modified by computing a scalar modulation parameter before the steepest-descent analysis can proceed. The result is a method with uniform accuracy across the entire (x,t)-plane, without high-precision arithmetic.

1.4 Open problems

Each part of this thesis raises natural questions that are not fully resolved.

The most fundamental concerns the convergence of the numerical method for RHPs. Trogdon [156] has proved convergence of the spectral collocation method for operator equations involving noncompact operators, with the primary application being RHPs on a single circle. For the multi-component problems that appear throughout this thesis (block systems on multiple intervals and circles), no convergence proof exists. The numerical evidence is uniformly convincing, but rigorous analysis of the multi-component case remains open.

In the orthogonal polynomial setting, the method of Chapter 4 handles Chebyshev-like weight functions. General Jacobi-type weight functions with arbitrary exponents introduce self-intersecting jump contours in the RHP and change the character of the error decay from exponential to polynomial. Extending the method to this setting will require similar techniques to those developed in Chapter 8.

Future work will further generalize the methods of Chapters 5 and 6. Perhaps the largest roadblock at present is that the Akhiezer iteration is most effective when applied to matrices with nearly-real spectra. A key advantage over existing methods is the reduced storage and computational overhead provided by the orthogonal polynomial three-term recurrence, but in general, this property is lost once orthogonal polynomials are defined in the complex plane. Future work will aim to develop analogous iterative methods that preserve advantageous storage properties when applied to matrices with general complex spectra by constructing appropriate families of complex plane orthogonal polynomials that retain sparse recurrence relations. How to extend the adaptivity techniques of Chapter 5 to nonsymmetric matrices is also an open problem that will require the analysis of the level curves of G_Σ to produce an analog of the "optimal ellipse" for the Chebyshev iteration [105].

In the soliton gas setting, the methods developed here are specific to KdV and mKdV. Extending them to other integrable equations that admit soliton gas solutions (e.g., the focusing nonlinear Schrödinger equation) will require new steepest-descent analyses adapted

to the different structure of those RHPs.

Part I

ORTHOGONAL POLYNOMIALS

Chapter 2

APPROXIMATION THEORY ON HIGHER-GENUS DOMAINS

2.1 Introduction

This chapter develops the approximation theory background needed for the rest of the thesis. The motivating problem is to iteratively approximate the function $f(x) = 1/(x - z)$, where $z \in \mathbb{C} \setminus \Sigma$ is fixed, for all $x \in \Sigma$ by polynomials of increasing degree. Furthermore, it should be possible to construct the degree k approximation from the degree $k - 1$ approximation in O(1) arithmetic operations. One way to accomplish this is with orthogonal polynomials. As we will see in Section 2.3, the classical approximation theory guarantees that doing so is asymptotically optimal with a convergence rate governed by a Green's function from potential theory. Given a weight function w supported on Σ, let $(p_j(x))_{j=0}^{\infty}$ denote the orthonormal polynomials corresponding to w. The $L^2_w(\Sigma)$-expansion of $1/(\diamond - z)$ is

$$\frac{1}{x - z} = \sum_{j=0}^{\infty} \mathcal{S}_\Sigma[p_j w](z)\, p_j(x), \qquad x \in \Sigma, \tag{2.1}$$

where the coefficients are given by the weighted *Stieltjes transforms* of the orthogonal polynomials

$$\mathcal{S}_\Sigma[p_k w](z) = \int_\Sigma \frac{p_k(x)\, w(x)}{x - z}\, \mathrm{d}x.$$

This framework is also extendable to general analytic f:

$$f(x) = \sum_{j=0}^{\infty} \alpha_j\, p_j(x), \quad \alpha_j = \langle f, p_j \rangle_{L^2_w(\Sigma)} = \int_\Sigma f(x)\, p_j(x)\, w(x)\, \mathrm{d}x. \tag{2.2}$$

Truncating (2.1) at degree k gives the degree-k polynomial approximation

$$\frac{1}{x - z} \approx \sum_{j=0}^{k} \mathcal{S}_\Sigma[p_j w](z)\, p_j(x). \tag{2.3}$$

The orthogonal polynomials can be generated through a three-term recurrence relation, so constructing the approximation (2.3) reduces to computing recurrence coefficients and evaluating Stieltjes transforms.

The difficulty of this task depends heavily on g. For $g = 0$, the classical Chebyshev polynomials on a single interval make everything explicit; the recurrence coefficients are constant and the Stieltjes transforms have closed form. For $g = 1$, classical results of Akhiezer [6] give formulae in terms of Jacobi theta and elliptic functions. For $g \geq 2$, analogous formulae involve Riemann theta functions and are both difficult and expensive to evaluate. Existing numerical methods do not achieve optimal complexity, leading us to consider a Riemann–Hilbert-based approach starting in Chapter 3.

In Section 2.2, we review the necessary orthogonal polynomial theory background. In Section 2.3, we review classical results on the convergence rate of orthogonal polynomial

series. Section 2.4 surveys general-purpose algorithms for computing recurrence coefficients, all of which require at least $\mathrm{O}(n^2)$ operations to compute n pairs of recurrence coefficients. Section 2.5 reviews classical results for the $g = 0$ case. Sections 2.6 and 2.7 review Jacobi elliptic and theta functions and Akhiezer's explicit formulae for the $g = 1$ case. In Section 2.8 we discuss computation difficulties in the $g \geq 2$ case, motivating the computational investigation of Riemann–Hilbert problems in Chapter 3.

2.2 Orthogonal polynomials and Cauchy integrals

2.2.1 Definitions and basic properties

Orthogonal polynomials on the real line can be generally defined for positive Borel measures with finite moments that are supported on an infinite number of points [74]. For the purposes of this thesis, we will restrict our definition to positive weight functions supported on a collection of disjoint intervals. See [147] for a general reference.

Let $\Sigma \subset \mathbb{R}$ be a finite union of closed intervals as in (1.1) and let $w : \Sigma \to [0, \infty)$ be a weight function that is continuous and strictly positive on the interior of Σ. For convenience, we also assume that w is normalized such that $\int_\Sigma w(x)\,\mathrm{d}x = 1$ unless stated otherwise. The weighted L^2 inner product on Σ is defined by

$$\langle f, g\rangle_{L^2_w(\Sigma)} = \int_\Sigma f(x)\overline{g(x)}\,w(x)\,\mathrm{d}x, \qquad \|f\|_{L^2_w(\Sigma)} = \sqrt{\langle f, f\rangle_{L^2_w(\Sigma)}}. \tag{2.4}$$

Consider a sequence of univariate monic polynomials $\pi_0(x; w), \pi_1(x; w), \pi_2(x; w), \ldots$ such that π_j has degree j for all $j \in \mathbb{N}$. These polynomials are said to be orthogonal with respect to w if

$$\langle \pi_j, \pi_k\rangle_{L^2_w(\Sigma)} = \int_\Sigma \pi_j(x; w)\pi_k(x; w)\,w(x)\,\mathrm{d}x = h_j\delta_{jk}, \qquad h_j > 0,$$

where δ_{jk} denotes the Kronecker delta. We refer to $(\pi_j(x; w))_{j=0}^\infty$ as the *monic orthogonal polynomials* corresponding to w.

The *orthonormal polynomials* $(p_j(x; w))_{j=0}^\infty$ are then defined by setting $p_j(x) = \gamma_j\pi_j(x)$ for all $j \in \mathbb{N}$ where $\gamma_j = 1/\sqrt{h_j}$. Clearly, the orthogonal polynomials satisfy

$$\langle p_j, p_k\rangle_{L^2_w(\Sigma)} = \int_\Sigma p_j(x; w)p_k(x; w)\,w(x)\,\mathrm{d}x = \delta_{jk}.$$

Such sequences exist and are unique for any weight function satisfying our assumptions [147]; their existence is established constructively via the Gram–Schmidt procedure applied to the monomials $1, \diamond, \diamond^2, \ldots$ under (2.4). We will often suppress the dependence on the weight function and simply write $p_j(x), \pi_j(x)$ when it is clear from context.

2.2.2 The three-term recurrence relation

The importance of the three-term recurrence is perhaps best stated by Walter Gautschi in his seminal treatise on computational orthogonal polynomials [74, Section 1.3].

> The three-term recurrence relation satisfied by orthogonal polynomials is arguably the single most important piece of information for the constructive and computational use of orthogonal polynomials. Apart from its obvious use in generating values of orthogonal polynomials and their derivatives, both within and without the spectrum of the measure, knowledge of the recursion coefficients (i) allows the zeros of orthogonal polynomials to be readily computed as eigenvalues of a symmetric tridiagonal matrix, and with them the all-important Gaussian quadrature rule, (ii) yields immediately the normalization coefficients needed to pass from monic orthogonal to orthonormal polynomials, (iii) opens access to polynomials of the second kind and related continued fractions, and (iv) allows an efficient evaluation of expansions in orthogonal polynomials.

The orthonormal polynomials satisfy a symmetric three-term recurrence

$$\begin{aligned} xp_0(x;w) &= \breve{a}_0(w)p_0(x;w) + \breve{b}_0(w)p_1(x;w), \\ xp_j(x;w) &= \breve{b}_{j-1}(w)p_{j-1}(x;w) + \breve{a}_j(w)p_j(x;w) + \breve{b}_j(w)p_{j+1}(x;w), \quad j \geq 1, \end{aligned} \tag{2.5}$$

for constants satisfying $\breve{a}_j(w) \in \mathbb{R}$ and $\breve{b}_j(w) > 0$ for all $j \in \mathbb{N}$. Note that $p_0(x) = 1$ when w is normalized. The constants $\{\breve{a}_j(w), \breve{b}_j(w)\}_{j=0}^{\infty}$ are known as the *recurrence coefficients*. As before, we will typically suppress the dependence on the weight function when clear from context. The existence of the three-term recurrence follows from the shift property

$$\langle \diamond p, q \rangle_{L^2_w(\Sigma)} = \langle p, \diamond q \rangle_{L^2_w(\Sigma)},$$

for real polynomials p, q and the fact that $(p_j(x;w))_{j=0}^{\infty}$ are orthogonal to all lower-degree polynomials.

The monic orthogonal polynomials also satisfy a three-term recurrence

$$\begin{aligned} x\,\pi_0(x;w) &= \breve{a}_0(w)\,\pi_0(x;w) + \pi_1(x;w), \\ x\,\pi_j(x;w) &= \breve{b}^2_{j-1}(w)\,\pi_{j-1}(x;w) + \breve{a}_j(w)\,\pi_j(x;w) + \pi_{j+1}(x;w), \quad j \geq 1, \end{aligned}$$

with the same initialization $\pi_0(x) = 1$. Given $\{\breve{a}_j, \breve{b}_j\}_{j=0}^{N-1}$, one can evaluate $p_0(x), \ldots, p_N(x)$ at any x in $\mathrm{O}(N)$ operations by simply running the recurrence. Thus, the orthonormal and/or monic orthogonal polynomials can be generated in optimal complexity once the recurrence coefficients are computed.

2.2.3 Jacobi matrices and spectral theory

The *Jacobi operator* corresponding to a sequence of orthonormal polynomials is defined as the infinite tridiagonal matrix

$$\mathbf{J}(w) = \begin{pmatrix} \check{a}_0(w) & \check{b}_0(w) & & \\ \check{b}_0(w) & \check{a}_1(w) & \check{b}_1(w) & \\ & \check{b}_1(w) & \check{a}_2(w) & \ddots \\ & & \ddots & \ddots \end{pmatrix}.$$

If we let

$$\mathbf{p}(x; w) = \begin{pmatrix} p_0(x; w) \\ p_1(x; w) \\ p_2(x; w) \\ \vdots \end{pmatrix},$$

the three-term recurrence can be expressed as

$$x\mathbf{p}(x; w) = \mathbf{J}(w)\mathbf{p}(x; w).$$

The spectral theorem for Jacobi operators gives that [53, Theorem 2.31]

$$\int_\Sigma \frac{w(x)\mathrm{d}x}{x - z} = \mathbf{e}_0^T \left(\mathbf{J}(w) - z\mathbf{I}\right)^{-1} \mathbf{e}_0, \quad z \in \mathbb{C} \setminus \mathbb{R}.$$

Letting $\mathbf{J}_N(w) \in \mathbb{R}^{N\times N}$ denote the leading $N \times N$ principal minor of $\mathbf{J}(w)$, the eigenvalues of $\mathbf{J}_N$ correspond to the zeroes of $p_N(\diamond; w)$. Thus, the Gauss quadrature rule corresponding to the weight function w has nodes $\sigma(\mathbf{J}_N(w))$ and weights corresponding to the first component of each (normalized) eigenvector [74, Theorem 3.1].

The correspondence between recurrence coefficients and measures is bijective. Given bounded sequences $\{\check{a}_k\} \subset \mathbb{R}$ and $\{\check{b}_k\} \subset (0, \infty)$, the polynomials defined by (2.5) are orthonormal with respect to some positive compactly supported[1] Borel measure μ on $\mathbb{R}$ [107]. This result is typically referred to as *Favard's theorem* [66], although it dates back to the work of Stieltjes [144]. In essence, this tells us that orthogonal polynomials are fully specified by their recurrence coefficients, without explicit reference to the weight function.

In this framework, the *spectral map* sends a Jacobi operator to its corresponding weight function [53, Chapter 2]. Thus, the problem of computing the recurrence coefficients for a given weight function, which we will consider in Chapter 4, is known as computing the *inverse spectral map*.

---

[1]The case of unbounded sequences, corresponding to measures with unbounded support, is more complicated [53, Section 2.4].

2.2.4 Cauchy integrals and the Stieltjes transform

Given an oriented curve $\Gamma \subset \mathbb{C}$, the *Cauchy transform* $\mathcal{C}_\Gamma$ is an operator that maps a function $f : \Gamma \to \mathbb{C}$ to its Cauchy integral, i.e.,

$$\mathcal{C}_\Gamma f(z) = \frac{1}{2\pi \mathrm{i}} \int_\Gamma \frac{f(\zeta)}{\zeta - z} \, \mathrm{d}\zeta, \qquad z \in \mathbb{C} \setminus \Gamma.$$

For convenience, we also define the related *Stieltjes transform* $\mathcal{S}_\Gamma$.

$$\mathcal{S}_\Gamma f(z) = \int_\Gamma \frac{f(\zeta)}{\zeta - z} \, \mathrm{d}\zeta = 2\pi \mathrm{i} \mathcal{C}_\Gamma f(z), \qquad z \in \mathbb{C} \setminus \Gamma.$$

The weighted Cauchy transforms of orthogonal polynomials

$$\mathcal{C}_\Sigma[p_k w](z) = \frac{1}{2\pi \mathrm{i}} \int_\Sigma \frac{p_k(x)\, w(x)}{x - z} \, \mathrm{d}x, \qquad z \in \mathbb{C} \setminus \Sigma.$$

will be of particular computational importance throughout this thesis. Importantly, they also satisfy the same three-term recurrence (2.5) as the orthogonal polynomials themselves, albeit with different initial conditions[2]. Denoting $C_k(z) = \mathcal{C}_\Sigma[p_k w](z)$ [74, 119],

$$\begin{aligned} z\, C_0(z) &= \check{a}_0\, C_0(z) + \check{b}_0\, C_1(z) - \frac{1}{2\pi \mathrm{i}}, \\ z\, C_k(z) &= \check{b}_{k-1}\, C_{k-1}(z) + \check{a}_k\, C_k(z) + \check{b}_k\, C_{k+1}(z), \qquad k \geq 1. \end{aligned} \tag{2.6}$$

However, the numerical use of the three-term recurrence for weighted Cauchy integrals is more complicated than with the orthogonal polynomials themselves. See [119, Chapter 7] for a full treatment. Initializing the recurrence requires a method to compute

$$C_0(z) = \frac{1}{2\pi \mathrm{i}} \int_\Sigma \frac{w(x)}{x - z} \, \mathrm{d}x.$$

Furthermore, simply running the forward recurrence (2.6) is inherently unstable because the Cauchy integrals $C_k(z)$ decay exponentially in the complex plane, $z \notin \Sigma$. Any error, including those on the order of machine precision, will be amplified exponentially as the recurrence is run.

Alternatively, one may consider directly solving the equivalent (infinite-dimensional) lin-

[2]The orthogonal polynomials and Cauchy transforms are the maximal and minimal solutions of the three-term recurrence, respectively [74, Section 2.3].

ear system

$$(\mathbf{J} - z\mathbf{I}) \begin{pmatrix} C_0(z) \\ C_1(z) \\ C_2(z) \\ \vdots \end{pmatrix} = \begin{pmatrix} \frac{1}{2\pi \mathrm{i}} \\ 0 \\ 0 \\ \vdots \end{pmatrix}.$$

See [119, Chapter 5] for a survey of available methods for solving linear systems of this form. Perhaps the simplest of these is the so-called finite section method: solve the truncated linear system

$$(\mathbf{J}_N - z\mathbf{I}_N)\mathbf{c} = \begin{pmatrix} \frac{1}{2\pi \mathrm{i}} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{c} \approx \begin{pmatrix} C_0(z) \\ C_1(z) \\ \vdots \\ C_N(z) \end{pmatrix}.$$

This can be done in $\mathrm{O}(N)$ operations via the Thomas algorithm [148]. However, this truncates the last row of the linear system, introducing errors when $C_{N+1}(z)$ is not small. This is particularly problematic when z approaches Σ, as the decay rate of the Cauchy integrals decreases. Fortunately, the forward recurrence (2.6) becomes increasingly stable as this occurs, so one needs only to determine when to use each method. A heuristic based on Bernstein ellipses is found in [119, Section 7.2] and can be extended to the higher-genus level curves (2.8) below. Alternatively, one can monitor the growth of the forward recurrence (2.5) run on the orthogonal polynomials (and hence the decay of the Cauchy integrals) to determine when to switch methods. When possible, we will prefer to avoid these complications and compute Cauchy integrals directly.

2.2.5 Orthogonal polynomial series coefficients

The weighted Cauchy/Stieltjes transforms enable the computation of orthogonal polynomial series coefficients. The series (2.1) is the $L^2_w(\Sigma)$-orthogonal expansion of $f(x) = 1/(x - z)$, for $z \notin \Sigma$. Writing out the coefficients explicitly,

$$\begin{aligned} \frac{1}{x - z} &= \sum_{j=0}^{\infty} \left\langle p_j, \frac{1}{\diamond - z} \right\rangle_{L^2_w(\Sigma)} p_j(x) \\ &= \sum_{j=0}^{\infty} \left(\int_{\Sigma} \frac{p_j(s)\, w(s)}{s - z} \, \mathrm{d}s \right) p_j(x) \\ &= \sum_{j=0}^{\infty} \mathcal{S}_{\Sigma}[p_j w]\,(z)\, p_j(x). \end{aligned}$$

A degree-k polynomial approximation to f is then given by truncating the series

$$\frac{1}{x-z} \approx \sum_{j=0}^{k} \mathcal{S}_\Sigma[p_j w]\,(z)\,p_j(x) =: P_k(x),$$

and the approximations may be generated iteratively

$$P_{k+1}(x) = P_k(x) + \mathcal{S}_\Sigma[p_{k+1}w]\,(z)p_{k+1}(x).$$

More generally, for any function f analytic in a neighborhood of Σ, the expansion (2.2) has coefficients $\alpha_k = \langle f, p_k\rangle_{L^2_w(\Sigma)}$. These are rarely available in closed form, but can be approximated by Cauchy integrals. Let Γ be a counterclockwise-oriented contour enclosing, but not intersecting, Σ such that f is analytic in a region containing Γ and its interior. Applying Cauchy's integral formula,

$$\begin{aligned}\alpha_k = \langle f, p_k\rangle_{L^2_w(\Sigma)} &= \int_\Sigma f(x)\,p_k(x)\,w(x)\,\mathrm{d}x = \int_\Sigma \left(\frac{1}{2\pi\mathrm{i}}\int_\Gamma \frac{f(z)}{z-x}\,\mathrm{d}z\right) p_k(x)\,w(x)\,\mathrm{d}x \\ &= \frac{1}{2\pi\mathrm{i}}\int_\Gamma f(z)\left(\int_\Sigma \frac{p_k(s)\,w(s)}{z-s}\,\mathrm{d}s\right)\mathrm{d}z = -\int_\Gamma f(z)\,\mathcal{C}_\Sigma[p_k w]\,(z)\,\mathrm{d}z.\end{aligned}$$

After parameterizing Γ and applying a quadrature rule, such as a trapezoid rule [153], resulting in nodes $\{z_j\}_{j=1}^m$ and weights $\{w_j\}_{j=1}^m$,

$$\alpha_j \approx \int_\Sigma \frac{1}{2\pi\mathrm{i}}\sum_{k=1}^{m}\frac{f(z_k)w_k}{z_k - x}p_j(x)w(x)\mathrm{d}x = -\sum_{k=1}^{m} f(z_k)w_k\mathcal{C}_\Sigma[p_j w](z_k).$$

We see that in order to approximate the coefficients accurately, it suffices to be able to evaluate the weighted Cauchy integrals pointwise. Therefore, to apply the approximation

$$f(x) = \sum_{j=0}^{\infty}\alpha_j p_j(x) \approx \sum_{j=0}^{k}\alpha_j p_j(x),$$

the recurrence coefficients and the pointwise evaluation of the Cauchy integrals of the desired orthonormal polynomials are all that is required. For reasons that will become clear in Chapter 5, we refer to these input coefficients as the *Akhiezer data*.

2.3 Potential theory and orthogonal polynomial series

In this section, we survey results from the literature regarding the convergence rate of orthogonal polynomial series on cut domains Σ. Our primary source for these results is the treatise of Walsh [162]. In Chapter 5, we will derive more explicit results for the specific

class of (Chebyshev-like) orthogonal polynomials that we consider.

The convergence rate of both orthogonal polynomial series and the optimal polynomial approximation to an analytic function is governed by potential theory on $\mathbb{C} \setminus \Sigma$ through a Green's function [127, 134, 162].

Definition 2.3.1 (Green's function [162, Section 4.1] cf. [134, Appendix A]). *Let $K \subset \mathbb{C}$ be compact and sufficiently regular*[3]*. The exterior Green's function of K with pole at infinity is the unique function $G_K : \mathbb{C} \setminus K \to [0, \infty)$ satisfying:*

- *G_K is harmonic in $\mathbb{C} \setminus K$ except at ∞.*
- *The function $G_K(z) - \log|z| = \mathrm{O}(1)$ as $z \to \infty$ and is harmonic in a neighborhood of $z = \infty$.*
- *G_K is continuous in $\mathrm{cl}(\mathbb{C} \setminus K)$ except at infinity and vanishes on*[4] *∂K.*

Such Green's functions relate to conformal mappings.

Theorem 2.3.2 (Walsh [162, Chapter 4, Theorem 1]). *Let $K \subset \mathbb{C}$ be compact and sufficiently regular and let H_K be a harmonic conjugate of G_K in $\mathbb{C} \setminus K$. Then, the function*

$$\Phi_K(z) = \mathrm{e}^{G_K(z) + \mathrm{i} H_K(z)}, \tag{2.7}$$

maps $\mathbb{C} \setminus K$ conformally onto the exterior of the unit disk so that the points at infinity in the two planes correspond to one another. Φ_K is uniquely determined up to a multiplicative constant of modulus one.

For each $\varrho > 1$, the level curve

$$\Gamma_\varrho(K) := \left\{ z \in \mathbb{C} : G_K(z) = \log \varrho \right\} = \left\{ z \in \mathbb{C} : |\Phi_K(z)| = \varrho \right\} \tag{2.8}$$

consists of finitely many smooth, non-nested closed loops. These loops are either completely disjoint, or they intersect at only a finite number of points.

Akhiezer used this conformal mapping relation to construct his namesake orthogonal polynomials supported on two intervals (see Section 2.7 below). It has also found use in computing Green's functions numerically [12, 63]. The level curves $\Gamma_\varrho(K)$ govern the convergence rate of orthogonal polynomial series supported on K. For instance, when K is the unit interval $[-1, 1]$, the level curves coincide with the classical Bernstein ellipses [32]

$$\Gamma_\varrho\left([-1, 1]\right) = \left\{ \frac{z + z^{-1}}{2} \ : \ |z| = \varrho \right\}, \tag{2.9}$$

---

[3] If K has no isolated points and $\mathbb{C} \setminus K$ is connected and of finite connectivity, then K is regular. In particular, $\Sigma = \bigcup_{j=1}^{g+1} [a_j, b_j]$ is regular.

[4] Here, the set boundary is taken in the complex plane, so $\partial\Sigma = \Sigma$.

that govern the convergence rate of Chebyshev projections and interpolants [151, Chapter 8].

To compute orthogonal polynomials supported on Σ, we will construct a so-called g-function [56] $\mathfrak{g}_\Sigma$ in Chapter 4. This construction first appeared in [164, Section 14], and it is straightforward to verify that

$$\operatorname{Re} \mathfrak{g}_\Sigma(z) = G_\Sigma(z), \quad \mathrm{e}^{\mathfrak{g}_\Sigma(z)} = \Phi_\Sigma(z), \tag{2.10}$$

provided the harmonic conjugate H_Σ in (2.7) is chosen appropriately.

The following theorem relates the level curves (2.8) to the convergence rate of orthogonal polynomial series:

Theorem 2.3.3 (Walsh [162, Chapter 6, Theorem 7])**.** *Let $(p_k(x))_{k\geq 0}$ be orthonormal polynomials corresponding to a weight function w on Σ, and let the function f be analytic interior to $\Gamma_\varrho(\Sigma)$ ($\varrho > 1$) but have a singularity on $\Gamma_\varrho(\Sigma)$. Then, the series coefficients $\alpha_k = \langle f, p_k\rangle_{L^2_w(\Sigma)}$ satisfy*

$$\limsup_{k\to\infty} |\alpha_k|^{1/k} \leq \varrho^{-1}.$$

Under the same assumptions, the series expansion (2.2) converges uniformly on any compact set interior to $\Gamma_\varrho(\Sigma)$ [162, Chapter 6 Theorem 9].

For a fixed $z \notin \Sigma$, taking $f(x) = 1/(x - z)$ yields the following corollary:

Corollary 2.3.4. *For any $z \notin \Sigma$, the Stieltjes transforms satisfy*

$$\limsup_{k\to\infty} \big|\mathcal{S}_\Sigma[p_k w](z)\big|^{1/k} = \mathrm{e}^{-G_\Sigma(z)}.$$

This corollary establishes the asymptotic decay rate of the Stieltjes/Cauchy transforms. Cf. Lemma 5.2.2 below where a limit is not necessary. The convergence rate $\mathrm{e}^{-G_\Sigma(z)}$ of the Stieltjes transforms is the convergence rate of the iterative $(\mathbf{A} - z\mathbf{I})x = \mathbf{b}$ solver developed in Chapter 5. For $\Sigma = [a, b]$ and $z = 0$ this coincides with the convergence rate of the classical Chebyshev iteration [69].

The asymptotic growth rate of orthogonal polynomials is also characterized by the exterior Green's function.

Theorem 2.3.5 (Stahl and Totik [143, Theorem 3.1.1])**.** *Let $(p_k(x))_{k\geq 0}$ be orthonormal polynomials corresponding to a weight function w on Σ. Then, for any fixed $z \in \mathbb{C}$,*

$$\limsup_{n\to\infty} |p_n(z)|^{1/n} \leq \mathrm{e}^{G_\Sigma(z)}.$$

Combining Theorems 2.3.3 and 2.3.5, we can bound the asymptotic convergence rate of orthogonal polynomial series.

Corollary 2.3.6. *Under the assumptions of Theorem 2.3.3:*

1. *For $x \in \Sigma$,*

$$\limsup_{N\to\infty} \left| f(x) - \sum_{k=0}^{N} \alpha_k p_k(x) \right|^{1/N} \leq \varrho^{-1}.$$

2. *For z strictly inside $\Gamma_\varrho(\Sigma)$,*

$$\limsup_{N\to\infty} \left| f(z) - \sum_{k=0}^{N} \alpha_k p_k(z) \right|^{1/N} \leq \mathrm{e}^{G_\Sigma(z)} \varrho^{-1}.$$

Proof. The combined theorems give that for z strictly inside $\Gamma_\varrho(\Sigma)$,

$$\limsup_{k\to\infty} |\alpha_k p_k(z)|^{1/k} \leq \mathrm{e}^{G_\Sigma(z)} \varrho^{-1} < 1. \tag{2.11}$$

Fix r such that $\mathrm{e}^{G_\Sigma(z)}/\varrho < r < 1$. By (2.11), there exists $K \in \mathbb{N}$ such that $|\alpha_k p_k(z)| \leq r^k$ for all $k \geq K$. For $N \geq K$,

$$\left| f(z) - \sum_{k=0}^{N} \alpha_k p_k(z) \right| \leq \sum_{k=N+1}^{\infty} |\alpha_k p_k(z)| \leq \sum_{k=N+1}^{\infty} r^k = \frac{r^{N+1}}{1-r},$$

so $\limsup_{N\to\infty} |f(z) - \sum_{k=0}^{N} \alpha_k p_k(z)|^{1/N} \leq r$. Since $r > \mathrm{e}^{G_\Sigma(z)}/\varrho$ was arbitrary, part 2 follows. Part 1 follows since $G_\Sigma(x) = 0$ for $x \in \Sigma$. □

It turns out that these bounds are tight, even if one considers the best possible polynomial approximation to f.

Theorem 2.3.7 (Walsh [162, Chapter 4, Theorems 7,8])**.** *Let f be a function that is analytic interior to $\Gamma_\varrho(\Sigma)$ ($\varrho > 1$) but has a singularity on $\Gamma_\varrho(\Sigma)$. Then, there exists a sequence of polynomials $(q_j(x))_{j=0}^{\infty}$ of respective degrees $0, 1, 2, \ldots$ such that for z strictly inside $\Gamma_\varrho(\Sigma)$ the inequality*

$$|f(z) - q_k(z)| \leq M R^k, \tag{2.12}$$

holds for $R = \mathrm{e}^{G_\Sigma(z)} \varrho^{-1}$, where $M > 0$ is some constant independent of k and z. However, there exist no polynomials such that (2.12) holds for $R < \mathrm{e}^{G_\Sigma(z)} \varrho^{-1}$.

In summary, the classical approximation theory demonstrates that orthogonal polynomial series achieve the optimal asymptotic convergence rate when approximating analytically extendable functions. This convergence rate is governed by the exterior Green's function with pole at infinity from potential theory which also governs the validity of the approximation away from the support of the orthogonal polynomials.

In subsequent chapters, we will make use of these ideas, developing more explicit results as needed. In Chapter 4, we develop a method to compute G_Σ numerically throughout the

complex plane. Approximations in Chebyshev-like orthogonal polynomials that we construct numerically in Chapter 4 attain the optimal convergence rate for all iteration counts, not just asymptotically. In Chapters 5 and 6, we will use these orthogonal polynomial approximations to approximate functions of matrices, and their convergence rate will govern that of the resulting iterative methods.

2.4 Classical algorithms for recurrence coefficients

Knowing that orthogonal polynomial expansions yield asymptotically optimal approximations, we seek a method to compute the Akhiezer data. Since Cauchy integrals can be computed in optimal complexity as described in Section 2.2.4, we restrict our attention to the recurrence coefficients $\{\check{a}_k, \check{b}_k\}_{k=0}^{\infty}$. In this section, we survey existing approaches, noting that none achieve optimal complexity. The primary reference for this section is [74, Section 2.2].

2.4.1 Discretizing the weight

Barring the use of expensive symbolic arithmetic, a numerical method for generating orthogonal polynomials and recurrence coefficients requires a way to approximate $L^2_w(\Sigma)$ inner products. A natural approach is to discretize the continuous measure $w(x)\,\mathrm{d}x$, effectively computing the inner products via quadrature. Consider a discrete N-point approximation to the weight w with nodes $x_1, \ldots, x_N$ and weights $w_1, \ldots, w_N$, i.e.,

$$\int_\Sigma f(x)w(x)\,\mathrm{d}x \approx \sum_{j=1}^{N} f(x_j)w_j.$$

That is, we approximate $w(x)\,\mathrm{d}x$ by a discrete N-point measure

$$\mu_N = \sum_{j=1}^{N} w_j\,\delta(x_j), \tag{2.13}$$

where $\delta(\diamond)$ denotes a Dirac delta mass. Under mild assumptions, the recursion coefficients of the discrete measure converge to the continuous recurrence coefficients $\check{a}_k(w)$, $\check{b}_k(w)$ as $N \to \infty$ [74, Theorem 2.32]. A full treatment of how to choose a discretization is provided in [74, Section 2.2.4]. For $\Sigma = \bigcup_{k=1}^{g+1}[a_k, b_k]$, it essentially suffices to use Gauss quadrature on each interval to capture the singularities of w.

2.4.2 The Stieltjes procedure

Given the discrete measure (2.13), the *Stieltjes procedure* is essentially the Gram–Schmidt algorithm applied to compute the corresponding orthogonal polynomials and recurrence coefficients simultaneously. As presented in [74, Section 2.2.3.1], the Stieltjes procedure constructs

the monic orthogonal polynomials $\pi_0, \pi_1, \ldots$ corresponding to (2.13). Initializing $\pi_{-1} \equiv 0$ and $\pi_0 \equiv 1$, the recursion coefficients and next polynomial at each step are computed as

$$\check{a}_j = \frac{\langle t\,\pi_j,\, \pi_j\rangle_{\mathrm{d}\mu_N}}{\langle \pi_j,\, \pi_j\rangle_{\mathrm{d}\mu_N}}, \qquad \check{b}_j^2 = \frac{\langle \pi_j,\, \pi_j\rangle_{\mathrm{d}\mu_N}}{\langle \pi_{j-1},\, \pi_{j-1}\rangle_{\mathrm{d}\mu_N}}, \qquad \pi_{j+1}(x) = (x - \check{a}_j)\pi_j(x) - \check{b}_j^2\,\pi_{j-1}(x).$$

Since the discrete inner products are finite sums, each step evaluates π_j at all N nodes via the three-term recurrence at cost $\mathrm{O}(N)$, so n pairs $(\check{a}_j, \check{b}_j)$ cost $\mathrm{O}(nN)$ in total. Unfortunately, the Stieltjes procedure often succumbs to numerical errors. In particular, it is susceptible to so-called *pseudostability* [73]: as j approaches N, rounding errors in the three-term recurrence can cause the computed coefficients to diverge from the true values. To avoid this, one must set $N \gg n$, so the cost of the Stieltjes procedure to compute $(\check{a}_j, \check{b}_j)_{j=0}^n$ is effectively at least $\mathrm{O}(n^2)$.

2.4.3 Householder, Lanczos, and RKPW

The same computation can be organized as a matrix factorization. Let $\mathbf{d} = (\sqrt{w_1}, \ldots, \sqrt{w_N})^T$, $\mathbf{\Lambda} = \mathrm{diag}(x_1, \ldots, x_N)$, and $b_{-1} = \|\mathbf{d}\|_2 = \sqrt{\sum_{j=1}^N w_j}$. By [74, Equation 2.2.24], there exists an orthogonal $\mathbf{Q} \in \mathbb{R}^{N\times N}$ such that

$$\underbrace{\begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{Q} \end{pmatrix}}_{\hat{\mathbf{Q}}} \underbrace{\begin{pmatrix} 1 & \mathbf{d}^T \\ \mathbf{d} & \mathbf{\Lambda} \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{Q}^T \end{pmatrix}}_{\hat{\mathbf{Q}}^T} = \underbrace{\begin{pmatrix} 1 & b_{-1}\mathbf{e}_1^T \\ b_{-1}\mathbf{e}_1 & \mathbf{J}_N(\mu_N) \end{pmatrix}}_{\mathbf{T}}, \tag{2.14}$$

where $\mathbf{J}_N(\mu_N)$ is the Jacobi matrix corresponding to the discrete measure. If one can compute $\mathbf{Q}$, then $\mathbf{J}_N(\mu_N) = \mathbf{Q}\mathbf{D}\mathbf{Q}^T$ can be immediately recovered. Building the factorization (2.14) reduces to a tridiagonalization problem: Given a real symmetric matrix $\mathbf{A}$, find an orthogonal matrix $\hat{\mathbf{Q}}$ and a Jacobi matrix[5] such that

$$\hat{\mathbf{Q}}\mathbf{A}\hat{\mathbf{Q}}^T = \mathbf{T}.$$

The unknowns $\hat{\mathbf{Q}}$ and $\mathbf{T}$ are uniquely determined by the first column of $\hat{\mathbf{Q}}$ [123, Theorem 7.2.1]. (Cf. [80, Theorem 8.3.2].) Since $\mathbf{d}$ and $\mathbf{\Lambda}$ are known, the factorization (2.14) matches this setting.

One way to solve the tridiagonalization problem is the Householder algorithm [80, Algorithm 8.3.1], which constructs $\mathbf{Q}$ as a product of Householder reflectors [89]. This approach is numerically stable, but requires $\mathrm{O}(n^3)$ arithmetic operations to tridiagonalize an $n \times n$ matrix [152, Theorem 26.1]. An alternative is the Lanczos algorithm [102] which requires only $\mathrm{O}(n^2)$ arithmetic operations to tridiagonalize an $n \times n$ matrix; however, in finite precision, the

[5] A Jacobi matrix is a real symmetric tridiagonal matrix with positive entries on the sub/superdiagonal. That is, it is a finite upper left subblock of a Jacobi operator.

Lanczos algorithm is known to be unstable. This can be circumvented by reorthogonalizing the computed vectors, but this increases the complexity to $O(n^3)$. The Rutishauser–Kahan–Pal–Walker (RKPW) algorithm of Gragg and Harrod [81] offers a solution. It provides an $O(n^2)$ stable implementation of Lanczos for matrices $\mathbf{A}$ with the arrowhead structure of (2.14). It interfaces with built-in eigenvalue computations, benefiting from optimizations of the QR algorithm. See `lanczos.m` in [74, Section 2.2.3.2] for an efficient implementation. It outperforms the other approaches and is very effective in general. In search of an algorithm with better complexity, in Chapter 4, we will develop an efficient optimal complexity ($O(n)$) numerical method applicable in a specialized (Chebyshev-like) case.

2.5 The Single-Interval Case: Chebyshev Polynomials

When $\Sigma = [a, b]$ consists of a single interval (genus $g = 0$), shifted and scaled Chebyshev polynomials, which satisfy a particular simple three-term recurrence, make a numerical method for computing recurrence coefficients unnecessary. This case is particularly instructive, as most quantities are made explicit.

2.5.1 Chebyshev polynomials

The classical Chebyshev polynomials defined on $\Sigma = \mathbb{I} = [-1, 1]$ are ubiquitous in numerical analysis and will be particularly useful throughout this thesis. A general reference is [109].

First kind (Chebyshev-T)

The Chebyshev polynomials of the first kind $(T_j(x))_{j=0}^{\infty}$ are typically defined by

$$T_j(\cos\theta) = \cos(j\theta).$$

They satisfy the recurrence relation

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_{j+1}(x) &= 2x\,T_j(x) - T_{j-1}(x), \quad j \geq 1, \end{aligned}$$

and are orthogonal with respect to the weight function

$$w_T(x) = \frac{1}{\pi}\mathbb{1}_{[-1,1]}(x)\frac{1}{\sqrt{1-x}\sqrt{x+1}}. \tag{2.15}$$

As defined, $(T_j(x))_{j=0}^{\infty}$ are not orthonormal, so for convenience, we define the normalized Chebyshev polynomials of the first kind by $\underline{T}_0(x) = 1$ and $\underline{T}_j(x) = \sqrt{2}T_j(x)$ for $j \geq 1$. Their

Jacobi operator is

$$\mathbf{J}\left(w_T\right) = \begin{pmatrix} 0 & 1/\sqrt{2} & & \\ 1/\sqrt{2} & 0 & 1/2 & \\ & 1/2 & 0 & \ddots \\ & & \ddots & \ddots \end{pmatrix}.$$

We will always use the square root symbol to denote the principal branch defined on $\mathbb{C} \setminus (-\infty, 0)$.

Second kind (Chebyshev-U)

The Chebyshev polynomials of the second kind $(U_j(x))_{j=0}^{\infty}$ are defined by

$$U_j(\cos\theta) = \frac{\sin((j+1)\theta)}{\sin\theta}.$$

They are orthonormal with respect to the weight function

$$w_U(x) = \frac{2}{\pi}\mathbb{1}_{[-1,1]}(x)\sqrt{1-x}\sqrt{x+1},$$

and correspond to the Jacobi operator

$$\mathbf{J}\left(w_U\right) = \begin{pmatrix} 0 & 1/2 & & \\ 1/2 & 0 & 1/2 & \\ & 1/2 & 0 & \ddots \\ & & \ddots & \ddots \end{pmatrix}.$$

Third kind (Chebyshev-V)

The Chebyshev polynomials of the third kind $(V_j(x))_{j=0}^{\infty}$ are defined by

$$V_j(\cos\theta) = \frac{\cos((j+1/2)\theta)}{\cos(\theta/2)}.$$

They are orthonormal with respect to the weight function

$$w_V(x) = \frac{1}{\pi}\mathbb{1}_{[-1,1]}(x)\frac{\sqrt{1+x}}{\sqrt{1-x}},$$

and correspond to the Jacobi operator

$$\mathbf{J}\left(w_V\right) = \begin{pmatrix} 1/2 & 1/2 & & \\ 1/2 & 0 & 1/2 & \\ & 1/2 & 0 & \ddots \\ & & \ddots & \ddots \end{pmatrix}.$$

Fourth kind (Chebyshev-W)

The Chebyshev polynomials of the fourth kind $(W_j(x))_{j=0}^{\infty}$ are defined by

$$W_j(\cos\theta) = \frac{\sin((j+1/2)\theta)}{\sin(\theta/2)}.$$

They are orthonormal with respect to the weight function

$$w_W(x) = \frac{1}{\pi}\mathbb{1}_{[-1,1]}(x)\sqrt{\frac{1-x}{1+x}},$$

and correspond to the Jacobi operator

$$\mathbf{J}\left(w_W\right) = \begin{pmatrix} -1/2 & 1/2 & & \\ 1/2 & 0 & 1/2 & \\ & 1/2 & 0 & \ddots \\ & & \ddots & \ddots \end{pmatrix}.$$

2.5.2 The Joukowsky map and explicit Cauchy integrals

The *Joukowsky map*

$$J(z) = \frac{z + z^{-1}}{2},$$

maps the unit circle to $[-1,1]$ and its interior and the exterior to the rest of the complex plane. It appears in the definition (2.9) of a Bernstein ellipse. We define a right inverse

$$J_+^{-1}(z) = z - \sqrt{z-1}\sqrt{z+1},$$

that maps $\mathbb{C}\backslash[-1,1]$ to the interior of the unit circle. This function represents the Chebyshev-T polynomials throughout the complex plane:

$$T_j(z) = \frac{J_+^{-1}(z)^j + J_+^{-1}(z)^{-j}}{2}.$$

The Cauchy integrals of the four kinds of Chebyshev polynomials on $\mathbb{I} = [-1, 1]$ have explicit closed forms in terms of J_+^{-1}. Recalling that the weight function of Chebyshev-$\underline{T}$ on $\mathbb{I} = [-1, 1]$ is given by $w_T(x) = \frac{1}{\pi\sqrt{x+1}\sqrt{1-x}}$ and that $\underline{T}_0 = 1$,

$$\mathcal{C}_{\mathbb{I}}[\underline{T}_0 w_T](z) = \frac{\mathrm{i}}{2\pi\sqrt{z-1}\sqrt{z+1}}.$$

Similarly,

$$\mathcal{C}_{\mathbb{I}}[U_0 w_U](z) = \frac{\mathrm{i}}{\pi}\left(z - \sqrt{z-1}\sqrt{z+1}\right),$$
$$\mathcal{C}_{\mathbb{I}}[V_0 w_V](z) = \frac{\mathrm{i}}{2\pi}\left(-1 + \frac{\sqrt{z+1}}{\sqrt{z-1}}\right), \quad \mathcal{C}_{\mathbb{I}}[W_0 w_W](z) = \frac{\mathrm{i}}{2\pi}\left(1 - \frac{\sqrt{z-1}}{\sqrt{z+1}}\right).$$

Theorem 2.5.1. *Let $C_k^{\square}(z) = \mathcal{C}_{\mathbb{I}}[\square_k w_{\square}](z)$ where $\square$ denotes the kind of Chebyshev polynomial $\underline{T}, U, V, W$. Then, for $j \geq 1$,*

$$C_j^{\underline{T}}(z) = \sqrt{2} C_0^{\underline{T}}(z) J_+^{-1}(z)^j, \qquad C_j^U(z) = C_0^U(z) J_+^{-1}(z)^j,$$
$$C_j^V(z) = C_0^V(z) J_+^{-1}(z)^j, \qquad C_j^W(z) = C_0^W(z) J_+^{-1}(z)^j.$$

Proof. See Appendix A.2. □

Importantly, because J_+^{-1} maps $\mathbb{C}\setminus[-1, 1]$ to the interior of the unit circle, these formulae yield numerically stable recurrences for evaluating the Cauchy integrals. They can also be scaled and shifted appropriately to compute $\mathcal{C}_{[a,b]}[fw]$ for an arbitrary interval $[a, b]$. Define the shifted polynomials and weights by

$$\tilde{\underline{T}}_k(x; [a, b]) = \underline{T}_k(M^{-1}(x; [a, b])), \qquad \tilde{w}_{\underline{T}}(x; [a, b]) = \tfrac{1}{\pi\sqrt{x-a}\sqrt{b-x}},$$
$$\tilde{U}_k(x; [a, b]) = U_k(M^{-1}(x; [a, b])), \qquad \tilde{w}_U(x; [a, b]) = \tfrac{2}{\pi}\left(\tfrac{2}{b-a}\right)^2 \sqrt{x-a}\sqrt{b-x},$$
$$\tilde{V}_k(x; [a, b]) = V_k(M^{-1}(x; [a, b])), \qquad \tilde{w}_V(x; [a, b]) = \tfrac{2}{\pi(b-a)}\tfrac{\sqrt{x-a}}{\sqrt{b-x}},$$
$$\tilde{W}_k(x; [a, b]) = W_k(M^{-1}(x; [a, b])), \qquad \tilde{w}_W(x; [a, b]) = \tfrac{2}{\pi(b-a)}\tfrac{\sqrt{b-x}}{\sqrt{x-a}},$$

where

$$M^{-1}(x; [a, b]) = \frac{2}{b-a}\left(x - \frac{b+a}{2}\right),$$

then

$$\begin{aligned}
\mathcal{C}_{[a,b]}[\tilde{\underline{T}}_0\tilde{w}_T](z) &= \frac{\mathrm{i}}{2\pi\sqrt{z-a}\sqrt{z-b}}, \\
\mathcal{C}_{[a,b]}[\tilde{\underline{T}}_k\tilde{w}_T](z) &= \frac{\mathrm{i}}{\sqrt{2}\pi\sqrt{z-a}\sqrt{z-b}} J_+^{-1}\left(M^{-1}(z;[a,b])\right)^k, && k \geq 1, \\
\mathcal{C}_{[a,b]}[\tilde{U}_k\tilde{w}_U](z) &= \frac{\mathrm{i}}{\pi}\frac{2}{b-a} J_+^{-1}\left(M^{-1}(z;[a,b])\right)^{k+1}, && k \geq 0, \\
\mathcal{C}_{[a,b]}[\tilde{V}_k\tilde{w}_V](z) &= \frac{\mathrm{i}}{2\pi}\frac{2}{b-a}\left(-1+\frac{\sqrt{z-a}}{\sqrt{z-b}}\right) J_+^{-1}\left(M^{-1}(z;[a,b])\right)^k, && k \geq 0, \\
\mathcal{C}_{[a,b]}[\tilde{W}_k\tilde{w}_W](z) &= \frac{\mathrm{i}}{2\pi}\frac{2}{b-a}\left(1-\frac{\sqrt{z-b}}{\sqrt{z-a}}\right) J_+^{-1}\left(M^{-1}(z;[a,b])\right)^k, && k \geq 0.
\end{aligned}$$

It is important to determine the leading-order asymptotic behavior of these integrals for our later calculations. We have

$$\begin{aligned}
\mathcal{C}_{[a,b]}[\tilde{\square}_0\tilde{w}_\square](z) &= \frac{\mathrm{i}}{2\pi}z^{-1} + \mathrm{O}(z^{-2}), \\
\mathcal{C}_{[a,b]}[\tilde{\square}_j\tilde{w}_\square](z) &= \mathrm{O}(z^{-j-1}), \quad j > 0,
\end{aligned} \tag{2.16}$$

where $\square$ denotes any of the four kinds of Chebyshev polynomial as $z \to \infty$. This follows from the above normalizations.

Returning to the $\Sigma = [-1,1]$ case for simplicity and denoting $\epsilon_0 = 1$ and $\epsilon_j = 2$ for $j \geq 1$, these formulae enable the explicit expression of the function $f(x) = 1/(x-z)$ in a Chebyshev-T series for a fixed $z \notin [-1,1]$ [109, Equation 5.14]:

$$\frac{1}{x-z} = \frac{-1}{\sqrt{z-1}\sqrt{z+1}} \sum_{j=0}^{\infty} \epsilon_j\, J_+^{-1}(z)^j\, T_j(x), \quad x \in [-1,1]. \tag{2.17}$$

The error in the degree-k partial sum

$$P_k(x) = \frac{-1}{\sqrt{z-1}\sqrt{z+1}} \sum_{j=0}^{k} \epsilon_j\, J_+^{-1}(z)^j\, T_j(x),$$

satisfies

$$\begin{aligned}
\left|\frac{1}{x-z} - P_k(x)\right| &= \left|\frac{-2}{\sqrt{z-1}\sqrt{z+1}} \sum_{j=k+1}^{\infty} J_+^{-1}(z)^j\, T_j(x)\right| \\
&\leq \frac{2}{\left|\sqrt{z-1}\sqrt{z+1}\right|} \cdot \frac{1}{1-\left|\frac{J_+^{-1}(z)}{J_+^{-1}(x)}\right|} \cdot \left|\frac{J_+^{-1}(z)}{J_+^{-1}(x)}\right|^k,
\end{aligned}$$

where the bound is valid for any x satisfying $\left|J_+^{-1}(z)\right| < \left|J_+^{-1}(x)\right|$, and the error decays geometrically at rate $\left|J_+^{-1}(z)/J_+^{-1}(x)\right|$. The exterior Green's function with pole at infinity for $\Sigma = [-1,1]$ is

$$G_{[-1,1]}(z) = -\log\left|J_+^{-1}(z)\right|,$$

so we see that this error bound coincides with the general asymptotic convergence rate of Corollary 2.3.6.

2.5.3 Jacobi Polynomials

For the numerical solution of Riemann–Hilbert problems, more general orthogonal polynomials on an interval will sometimes be necessary. The *Jacobi polynomials* $\left(P_j^{(\alpha,\beta)}(x)\right)_{j=0}^{\infty}$, are the classical orthogonal polynomials on $\Sigma = [-1,1]$ corresponding to the weight function [119]

$$w^{(\alpha,\beta)}(x) = \frac{1}{2^{\alpha+\beta+1}B(\alpha+1,\beta+1)}\mathbb{1}_{[-1,1]}(x)(1-x)^{\alpha}(1+x)^{\beta}, \qquad \alpha,\beta > -1,$$

where B denotes the Beta function [59, Section 5.12] and appears only as a normalization. Breaking with the classical convention, we will use $P_j^{(\alpha,\beta)}$ to denote the *orthonormal* polynomials[6]:

$$\int_{-1}^{1} P_m^{(\alpha,\beta)}(x)\, P_n^{(\alpha,\beta)}(x)\, w^{(\alpha,\beta)}(x)\,\mathrm{d}x = \delta_{mn}.$$

These polynomials satisfy the three-term recurrence (2.5) with coefficients derived from [59, Section 18.9]:

$$\begin{aligned}
\check{a}_j(w^{(\alpha,\beta)}) &= \frac{\beta^2-\alpha^2}{(2j+\alpha+\beta)(2j+\alpha+\beta+2)},\\
\check{b}_j(w^{(\alpha,\beta)}) &= \frac{2}{2j+\alpha+\beta+2}\sqrt{\frac{(j+1)(j+\alpha+1)(j+\beta+1)(j+\alpha+\beta+1)}{(2j+\alpha+\beta+1)(2j+\alpha+\beta+3)}},
\end{aligned}$$

where the degenerate cases are understood through continuity in α and β. The Cauchy integrals are represented explicitly in terms of hypergeometric functions [147, Theorem 4.61.2], but it is often easier to compute them as described in Section 2.2.4. When the forward recurrence is used, it is initialized with

$$\mathcal{C}_{[-1,1]}\left[w^{(\alpha,\beta)}\right](z) = \frac{-1}{2\pi\mathrm{i}(z+1)}\,{}_2F_1\!\left(1,\,\beta+1;\,\alpha+\beta+2;\,\frac{2}{z+1}\right), \qquad z \notin [-1,1],$$

[6] Classically, the normalization requires $P_j^{(\alpha,\beta)}(1) = \binom{j+\alpha}{j}$ [147, Equation 4.1.1].

where ${}_2F_1$ denotes a hypergeometric function [59, Section 15.6]. We will also use the notation

$$\widetilde{P}_n^{(\alpha,\beta)}(\,\diamond\,;\,[a,b]) = P_n^{(\alpha,\beta)}\big(M^{-1}(\,\diamond\,;\,[a,b])\big), \quad \widetilde{w}^{(\alpha,\beta)}(x;\,[a,b]) \propto (b-x)^\alpha (x-a)^\beta, \tag{2.18}$$

for shifted and scaled Jacobi polynomials.

Special cases

The Chebyshev polynomials are all special cases of the Jacobi family:

$$\begin{aligned} T_n(x) &\propto P_n^{(-1/2,\,-1/2)}(x), & U_n(x) &= P_n^{(1/2,\,1/2)}(x), \\ V_n(x) &= P_n^{(-1/2,\,1/2)}(x), & W_n(x) &= P_n^{(1/2,\,-1/2)}(x). \end{aligned}$$

Additionally, the case $\alpha = \beta = 0$ corresponds to the classical *Legendre polynomials*.

2.6 Jacobi Elliptic and Theta Functions

When Σ consists of two intervals (genus $g = 1$), explicit formulae for the orthogonal polynomials are given in terms of *elliptic functions*. This section collects the necessary special functions and their properties, all of which can be found in [59, Chapters 19–22]. General references are [6, 163].

2.6.1 Elliptic integrals and modulus

The *complete elliptic integral of the first kind* is defined by [59, Section 19.2]

$$K(k) = \int_0^1 \frac{\mathrm{d}t}{\sqrt{(1-t^2)(1-k^2t^2)}} = \int_0^{\pi/2} \frac{\mathrm{d}\theta}{\sqrt{1-k^2\sin^2\theta}},$$

where $k \in (0,1)$ denotes the *elliptic modulus*. The *incomplete elliptic integral of the first kind* is defined by

$$F(\phi,k) = \int_0^{\sin\phi} \frac{\mathrm{d}t}{\sqrt{(1-t^2)(1-k^2t^2)}} = \int_0^{\phi} \frac{\mathrm{d}\theta}{\sqrt{1-k^2\sin^2\theta}}.$$

We also denote by $k' = \sqrt{1-k^2}$ the *complementary modulus* and define the *nome* $q = q(k)$ by

$$q(k) = \exp\left(-\pi \frac{K(k')}{K(k)}\right) \in (0,1). \tag{2.19}$$

Elliptic integrals can be efficiently computed via standard algorithms [59, Section 19.36] implemented in most scientific programming languages[7].

2.6.2 Jacobi theta functions

The four *Jacobi theta functions* are defined by [59, Section 20.2]

$$\begin{aligned}
\theta_1(z,q) &= 2\sum_{n=0}^{\infty}(-1)^n q^{(n+\frac{1}{2})^2}\sin\big((2n+1)z\big),\\
\theta_2(z,q) &= 2\sum_{n=0}^{\infty} q^{(n+\frac{1}{2})^2}\cos\big((2n+1)z\big),\\
\theta_3(z,q) &= 1+2\sum_{n=1}^{\infty} q^{n^2}\cos(2nz),\\
\theta_4(z,q) &= 1+2\sum_{n=1}^{\infty}(-1)^n q^{n^2}\cos(2nz),
\end{aligned} \tag{2.20}$$

where $q = q(k) \in (0,1)$ is the nome (2.19). Each series converges absolutely for $|q| < 1$. For small values of q, the above series converges rapidly, and the Jacobi theta functions and their derivatives (computed by differentiating termwise) can be rapidly evaluated to machine precision by summing the first few series terms [59, Section 20.14]. See [101] for error bounds. In the Akhiezer formulae below, q is typically very small.

2.6.3 Jacobi elliptic functions

Setting $\zeta = \pi z/(2K(k))$, the three basic Jacobi elliptic functions are defined by [59, Section 22.2]

$$\begin{aligned}
\operatorname{sn}(z,k) &= \frac{\theta_3(0,q)}{\theta_2(0,q)}\cdot\frac{\theta_1(\zeta,q)}{\theta_4(\zeta,q)},\\
\operatorname{cn}(z,k) &= \frac{\theta_4(0,q)}{\theta_2(0,q)}\cdot\frac{\theta_2(\zeta,q)}{\theta_4(\zeta,q)},\\
\operatorname{dn}(z,k) &= \frac{\theta_4(0,q)}{\theta_3(0,q)}\cdot\frac{\theta_3(\zeta,q)}{\theta_4(\zeta,q)}.
\end{aligned}$$

For real $z \in [-K(k), K(k)]$, these functions are inverses of the elliptic integral $F(\phi,k)$. That is, if $z = F(\phi,k)$, then $\operatorname{sn}(z,k) = \sin\phi$, $\operatorname{cn}(z,k) = \cos\phi$, and $\operatorname{dn}(z,k) = \sqrt{1-k^2\sin^2\phi}$ [59,

[7] Bizarrely, the widely-used `Julia` package `Elliptic.jl` only permits real-valued inputs to $F(\diamond,k)$ despite implementing a method [40] that permits general complex values. We include an alternative implementation in [13] but advocate for the package `EllipticFunctions.jl` in general usage.

Section 22.16]. They satisfy the Pythagorean identities $\mathrm{sn}^2 + \mathrm{cn}^2 = 1$ and $k^2\,\mathrm{sn}^2 + \mathrm{dn}^2 = 1$, and are doubly periodic meromorphic functions. Their derivatives are [59, Section 22.13]:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}z}\,\mathrm{sn}(z,k) &= \mathrm{cn}(z,k)\,\mathrm{dn}(z,k),\\
\frac{\mathrm{d}}{\mathrm{d}z}\,\mathrm{cn}(z,k) &= -\,\mathrm{sn}(z,k)\,\mathrm{dn}(z,k),\\
\frac{\mathrm{d}}{\mathrm{d}z}\,\mathrm{dn}(z,k) &= -k^2\,\mathrm{sn}(z,k)\,\mathrm{cn}(z,k).
\end{aligned}$$

In the limit $k \to 0$, the functions reduce to classical trigonometric functions: $\mathrm{sn}(z,0) = \sin z$, $\mathrm{cn}(z,0) = \cos z$, $\mathrm{dn}(z,0) = 1$ [59, Section 22.5].

2.7 The Two-Interval Case: Akhiezer Polynomials

In 1936, Akhiezer used elliptic functions and conformal mappings to construct orthogonal polynomials supported on two disjoint intervals [2]. This was later translated into English as part of Akhiezer's treatise on elliptic functions [6, Section 53]. In this section, we summarize Akhiezer's construction and derive formulae for the $g = 1$ Akhiezer polynomial recurrence coefficients.

2.7.1 Akhiezer's formulae and notation

Following Akhiezer's convention, we consider[8] $\Sigma = [-1, \alpha] \cup [\beta, 1]$ with $-1 < \alpha < \beta < 1$ and define the weight function

$$w(x) = \frac{1}{\pi}\mathbb{1}_\Sigma(x)\frac{\sqrt{x-\alpha}}{\sqrt{1-x}\sqrt{x+1}\sqrt{x-\beta}}. \tag{2.21}$$

In some sense, (2.21) provides a natural extension of the Chebyshev polynomials to two intervals[9], as it limits to the Chebyshev-T weight function (2.15) as $\alpha \to \beta$.

Consider the elliptic modulus [6, Equation 6]

$$k = \sqrt{\frac{2(\beta-\alpha)}{(1-\alpha)(1+\beta)}},$$

and set $K = K(k)$. This will be the modulus of all elliptic functions that follow. The

[8] This can be generalized to any two disjoint intervals by shifting and scaling.

[9] A more common way to define Chebyshev polynomials on domains other than $[-1, 1]$ is through the minimax property of the Chebyshev-T polynomials [67]. The Akhiezer polynomials only retain this under certain conditions on the domain endpoints and the degree of the polynomial [68, Chapter 3].

parameter ρ is defined so that

$$1 - 2\,\mathrm{sn}^2\,\rho = \alpha,$$

and $0 < \rho < K$ with $\mathrm{sn}(\diamond) = \mathrm{sn}(\diamond, k)$ [6, Equation 10]. Solving for ρ,

$$\rho = F\left(\arcsin\sqrt{\frac{1-\alpha}{2}}, k\right).$$

Now, given $x \in \mathbb{C}$, define $u = u(x)$ such that [6, Equation 9]

$$x - \alpha = \frac{1-\alpha^2}{2\,\mathrm{sn}^2(u) + \alpha - 1}.$$

An explicit formula for u is

$$u = F\left(\arcsin\sqrt{\frac{(\alpha-1)(1+x)}{2(\alpha-x)}}, k\right). \tag{2.22}$$

We also define the auxiliary functions

$$H(z) = \theta_1\Big(\frac{\pi z}{2K}, q\Big), \quad \Theta(z) = \theta_4\Big(\frac{\pi z}{2K}, q\Big),$$

with the nome $q = q(k)$ from (2.19). The nth degree orthonormal polynomial p_n is [6]

$$p_n(x) = \frac{C_n}{2}\left(\left(\frac{H(u-\rho)}{H(u+\rho)}\right)^n \frac{\Theta(u+2n\rho)}{\Theta(u)} + \left(\frac{H(u+\rho)}{H(u-\rho)}\right)^n \frac{\Theta(u-2n\rho)}{\Theta(u)}\right), \tag{2.23}$$

where

$$C_n = \frac{\sqrt{2}\Theta(\rho)}{\sqrt{\Theta\left((2n-1)\rho\right)\Theta\left((2n+1)\rho\right)}},$$

for $n \geq 1$, $C_0 = 1$, and u is given by (2.22).

Akhiezer's derivation also includes a formula for the Cauchy integrals of these polynomials:

$$\mathcal{C}_\Sigma\left[p_n w\right](x) = -\frac{C_n}{2\pi\mathrm{i}}\left(\frac{H(u-\rho)}{H(u+\rho)}\right)^n \frac{\Theta(u+2n\rho)}{\Theta(u)} \frac{\sqrt{x-\alpha}}{\sqrt{x-1}\sqrt{x+1}\sqrt{x-\beta}}. \tag{2.24}$$

A complex exterior Green's function with pole at infinity is also derived:

$$\Phi_\Sigma(z) = -\frac{H(u+\rho)}{H(u-\rho)}.$$

In Appendix A.1 we derive formulae for the recurrence coefficients of the Akhiezer polyno-

mials (cf. [68, Theorem 3.3.7]).

$$\begin{aligned}
\check{a}_0 =&(1-\alpha^2)\left(\frac{1}{8\,\mathrm{sn}^2\,\rho}+\frac{\mathrm{sn}''\,\rho}{8\,\mathrm{sn}\,\rho(\mathrm{sn}'\,\rho)^2}+\frac{H'(2\rho)}{H(2\rho)}\frac{1}{4\,\mathrm{sn}\,\rho\,\mathrm{sn}'\,\rho}\right)+\frac{1-\alpha^2}{2\,\mathrm{sn}\,\rho\,\mathrm{sn}'\,\rho}\frac{\Theta'(\rho)}{\Theta(\rho)}+\alpha,\\
\check{a}_n =&(1-\alpha^2)\left(\frac{1}{8\,\mathrm{sn}^2\,\rho}+\frac{\mathrm{sn}''\,\rho}{8\,\mathrm{sn}\,\rho(\mathrm{sn}'\,\rho)^2}\right)\\
&+\frac{1-\alpha^2}{4\,\mathrm{sn}\,\rho\,\mathrm{sn}'\,\rho}\left(\frac{H'(2\rho)}{H(2\rho)}+\frac{\Theta'((2n+1)\rho)}{\Theta((2n+1)\rho)}-\frac{\Theta'((2n-1)\rho)}{\Theta((2n-1)\rho)}\right)+\alpha,\quad n\geq 1,\\
\check{b}_0 =&\frac{1-\alpha^2}{4\,\mathrm{sn}\,\rho\,\mathrm{sn}'\,\rho}\frac{H'(0)}{H(2\rho)}\sqrt{\frac{2\Theta(3\rho)}{\Theta(\rho)}},\\
\check{b}_n =&\frac{1-\alpha^2}{4\,\mathrm{sn}\,\rho\,\mathrm{sn}'\,\rho}\frac{H'(0)}{H(2\rho)}\frac{\sqrt{\Theta((2n+3)\rho)\Theta((2n-1)\rho)}}{\Theta((2n+1)\rho)},\quad n\geq 1.
\end{aligned}\tag{2.25}$$

The polynomials themselves, their Cauchy integrals, and their recurrence coefficients can be efficiently computed using the standard algorithms described in Section 2.6 above. We find that these formulae are extremely efficient for evaluating the Akhiezer polynomials, their Cauchy integrals, and their recurrence coefficients in this special $g = 1$ case, outperforming by multiple orders of magnitude both the $\mathrm{O}(n)$ Riemann–Hilbert approach of Chapter 4 and the optimized $\mathrm{O}(n^2)$ algorithm given as RKPW in [81] and `lanczos.m` in [74]. Thus, these formulae should be used for all two interval problems for which one can choose the weight function to be of the form (2.21).

Remark 2.7.1. *The mapped variable $u(x)$ is undefined at $x = \alpha$. While $u(\alpha) = \mathrm{i}K(k')$ by continuity, non-Lipschitz behavior of the involved functions near this point leads to numerical instabilities when this value is used in the formulae. If one needs to compute the polynomials at this endpoint, it is better to generate them via the recurrence coefficients rather than using* (2.23) *directly.*

Remark 2.7.2. *In the symmetric case $\Sigma = [-1, -a] \cup [a, 1]$ ($a \in (0, 1)$), the Akhiezer polynomial recurrence coefficients can be shown to take a particularly simple form:*

$$\begin{aligned}
&\check{a}_n = (-1)^n a,\quad n \geq 0,\\
&\check{b}_0 = \sqrt{\frac{1-a^2}{2}},\quad \check{b}_n = \frac{\sqrt{1-a^2}}{2},\quad n \geq 1.
\end{aligned}$$

This follows from a result of Peherstorfer [124], but can be found directly, for example, by using continued fractions to show that (2.21) *is the density of the spectral measure for the Jacobi matrix of recurrence coefficients.*

If $b_1+a_2 = a_1+b_2$, the Akhiezer polynomial recurrence coefficients for $\Sigma = [a_1, b_1]\cup[a_2, b_2]$, $b_1 < a_2$, can be obtained by shifting and scaling these formulae.

2.8 The Multi-Interval Case: Genus $g \geq 2$

For more than two intervals, exact formulae are still available, but these formulae are complicated to the point that it is faster and easier to employ a numerical method rather than evaluating them directly. Essentially, the Jacobi theta functions from the $g = 1$ case are replaced with higher-genus Riemann theta functions, which are considerably more difficult and expensive to evaluate [50].

On a general collection of disjoint intervals Σ, the Akhiezer polynomials are defined as the orthonormal polynomials with respect to the weight function[10]

$$w(x) = \frac{1}{\pi} \mathbb{1}_{\Sigma}(x) \frac{\prod_{j=1}^{g} \sqrt{x - b_j}}{\sqrt{b_{g+1} - x} \prod_{j=1}^{g+1} \sqrt{x - a_j}}. \tag{2.26}$$

This is clearly a generalization of both the classical Chebyshev polynomials and $g = 1$ Akhiezer polynomials; however, Akhiezer's conformal mapping construction breaks down for $g \geq 2$, making the analysis considerably more difficult. The general weight function (2.26) was first considered by Akhiezer himself in 1960 [5]. A year later, in joint work with Tomcuk, Akhiezer derived Bernstein–Szegő asymptotics and produced an incomplete description of the polynomials in terms of hyperelliptic functions [7]. It was not until 2002 that the description of the Akhiezer polynomials was completed. In [47], Chen and Lawrence express both the Akhiezer polynomials and their recurrence coefficients in terms of ratios of Riemann theta functions composed with the Abel map.

To understand why the higher-genus Akhiezer polynomial formulae are difficult to compute with, we investigate the Riemann theta function [59, Section 21.2]

$$\theta(\mathbf{z}; \mathbf{\Omega}) = \sum_{\mathbf{n} \in \mathbb{Z}^g} \exp\bigl(\pi \mathrm{i}\, \mathbf{n}^T \mathbf{\Omega}\, \mathbf{n} + 2\pi \mathrm{i}\, \mathbf{n}^T \mathbf{z}\bigr), \qquad \mathbf{z} \in \mathbb{C}^g, \tag{2.27}$$

where $\mathbf{\Omega}$ is a Riemann matrix. We immediately see a scaling issue from summing over a g-dimensional lattice. Like with the Jacobi theta functions, (2.27) can be computed to a desired accuracy by summing over finitely-many terms, but the number of terms required grows exponentially in g [50]. Another indication comes from the composition with the Abel map, which embeds a point on a 1-dimensional Riemann surface into a g-dimensional space. The Riemann theta function (2.27) then maps this back to 1 dimension. For $g > 1$, successively applying these functions is clearly inefficient. Furthermore, hyperelliptic integrals [159, Section 5.5], Riemann theta functions [50], the Abel map [51], and the Riemann constant vector [52, Section 2.12] each appear in these formulae and require a specialized numerical method to compute.

Later work points towards an alternative numerical approach. In [46], Chen and Its sim-

[10] Under this convention, the Chebyshev-T polynomials are the genus zero Akhiezer polynomials, and Akhiezer's original work [2] considered the genus one Akhiezer polynomials.

plify the construction of the Akhiezer polynomials and their recurrence coefficients by studying the corresponding Fokas–Its–Kitaev (FIK) Riemann–Hilbert problem (RHP). While the final formulae are unchanged, the FIK RHP has the orthogonal polynomials as its solution, so the complicated quasi-periodic behavior resulting from ratios of Riemann theta functions is encoded into the problem itself. By directly approaching this problem numerically, we circumvent the need to compute Riemann theta functions, Abel maps, and Riemann constant vectors. It is with this perspective that we turn our attention to numerical methods for Riemann–Hilbert problems on higher-genus domains.

Chapter 3

RIEMANN–HILBERT PROBLEMS ON HIGHER-GENUS DOMAINS

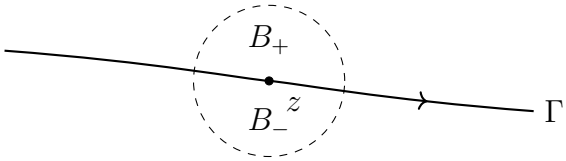


Figure 3.1: An oriented contour Γ passing through an interior point z. The ball $B(z)$ is divided by Γ into B_+ (to the left of the orientation) and B_- (to the right). Boundary values are defined by non-tangential limits taken from within $B_\pm$.

This chapter develops the necessary framework to numerically solve RHPs on higher-genus domains. Section 3.1 collects the general definitions and the Plemelj lemma. Section 3.2 states the FIK problem and its exact solution; the FIK-specific steepest-descent deformations are carried out in Chapter 4. Section 3.2.2 states the soliton gas RHP. Section 3.3 develops the unified numerical collocation framework that Chapters 4–8 use after their respective deformations.

3.1 Riemann–Hilbert problems

To define a Riemann–Hilbert problem, let $\Gamma \subset \mathbb{C}$ be a smooth, non-self-intersecting, oriented contour. To each interior point $z \in \Gamma$, we associate a ball $B = B(z)$ such that $B \setminus \Gamma$ has two connected components $B_\pm$, where B_+ (B_-) is to the left (resp. right) with respect to the orientation of Γ. For $F : \mathbb{C} \setminus \Gamma \to \mathbb{C}$, define the boundary values by

$$F^\pm(z) = \lim_{\substack{z' \to z, \\ z' \in B_\pm}} F(z'), \tag{3.1}$$

where the limit is assumed to be taken non-tangentially. See Figure 3.1 for an illustration. Under the same assumptions, we define notation for the Cauchy operators

$$\mathcal{C}_\Gamma^\pm f(z) = \lim_{\substack{z' \to z, \\ z' \in B_\pm}} \mathcal{C}_\Gamma f(z').$$

The Plemelj lemma is integral to the theory of RHPs.

Lemma 3.1.1 (Plemelj [113, Section 17])**.** *Let $\Gamma \subset \mathbb{C}$ and let f be Hölder continuous on Γ. Then, for $z \in \Gamma$ not an endpoint,*

$$\mathcal{C}_\Gamma^+ f(z) - \mathcal{C}_\Gamma^- f(z) = f(z).$$

3.1.1 Scalar Riemann–Hilbert problems

A homogeneous scalar Riemann–Hilbert problem can be defined as follows:

Riemann–Hilbert Problem 3.1.2. *Given $\Gamma \subset \mathbb{C}$ and a function*[1] *$v : \Gamma \to \mathbb{C}$, find $\Phi : \mathbb{C} \setminus \Gamma \to \mathbb{C}$ with the following properties:*

Analyticity: *Φ is analytic in $\mathbb{C} \setminus \Gamma$.*

Jump condition: *$\Phi^+(z) = \Phi^-(z)v(z)$, $z \in \Gamma$.*

Normalization: *$\lim_{z\to\infty} \Phi(z) = 1$.*

In short, a Riemann–Hilbert problem asks if one can recover a function given the behavior of its singularities. Indeed, since Φ is analytic everywhere except on Γ, its singularities are completely described by v. The normalization at infinity is typically included to ensure that the Riemann–Hilbert problem has a unique solution. Additional asymptotic conditions at the open endpoints of Γ (if they exist) may also be required to ensure uniqueness. For instance, if $\Gamma = [a, b]$, then one may wish to prescribe

$$\begin{aligned} \Phi(z) &= \mathrm{O}(|z-a|^{\alpha}), \quad z \to a, \\ \Phi(z) &= \mathrm{O}(|z-b|^{\beta}), \quad z \to b, \end{aligned} \tag{3.2}$$

for some $\alpha, \beta \in \mathbb{R}$.

3.1.2 Matrix Riemann–Hilbert problems

Riemann–Hilbert problems can be generalized to matrix-valued functions defined in the complex plane.

Riemann–Hilbert Problem 3.1.3. *Given $\Gamma \subset \mathbb{C}$ as in Riemann–Hilbert Problem 3.1.2 and a jump matrix $\mathbf{V} : \Gamma \to \mathbb{C}^{2\times 2}$, find $\mathbf{Y} : \mathbb{C} \setminus \Gamma \to \mathbb{C}^{2\times 2}$ with the following properties:*

Analyticity: *$\mathbf{Y}$ is analytic in $\mathbb{C} \setminus \Gamma$.*

Jump condition: *$\mathbf{Y}^+(z) = \mathbf{Y}^-(z)\mathbf{V}(z)$, $z \in \Gamma$.*

Normalization: *$\lim_{z\to\infty} \mathbf{Y}(z) = \mathbf{I}$.*

Asymptotic conditions in the vein of (3.2) should also be defined element-wise to ensure a unique solution when Γ has open endpoints.

---

[1] Generally, we want v to be Hölder continuous.

3.2 The FIK and soliton gas RHPs

3.2.1 Orthogonal polynomials as an RHP

The original formulation of orthogonal polynomials as a matrix Riemann–Hilbert problem is due to Fokas, Its, and Kitaev and is for orthogonal polynomials with $\Sigma = \mathbb{R}$ [70]. Consider the following RHP.

Riemann–Hilbert Problem 3.2.1 (Fokas–Its–Kitaev)**.** *Find a* 2×2 *matrix-valued function* $\mathbf{Y}_n(z)$ *with the following properties:*

> ***Analyticity:*** $\mathbf{Y}_n(z)$ *is analytic for* $z \in \mathbb{C} \setminus \mathbb{R}$*, admitting continuous boundary values on* $\mathbb{R}$*.*
>
> ***Jump condition:*** $\mathbf{Y}_n^+(z) = \mathbf{Y}_n^-(z) \begin{pmatrix} 1 & w(z) \\ 0 & 1 \end{pmatrix}$, $z \in \mathbb{R}$.
>
> ***Normalization:*** $\mathbf{Y}_n(z) = \big(\mathbf{I} + \mathrm{O}(z^{-1})\big) \begin{pmatrix} z^n & 0 \\ 0 & z^{-n} \end{pmatrix}$ *as* $z \to \infty$.

This problem has a unique solution that is given by [53]

$$\mathbf{Y}_n(z) = \begin{pmatrix} \pi_n(z;w) & \mathcal{C}_\Sigma[\pi_n w](z) \\ -2\pi\mathrm{i}\gamma_{n-1}^2\,\pi_{n-1}(z;w) & -2\pi\mathrm{i}\gamma_{n-1}^2\,\mathcal{C}_\Sigma[\pi_{n-1}w](z) \end{pmatrix}, \tag{3.3}$$

where γ_n is a normalization constant defined in Section 2.2.1. The recurrence coefficients are recovered from the first-order expansion $\mathbf{Y}_n(z) = (\mathbf{I} + z^{-1}\mathbf{Y}_n^{(1)} + \mathrm{O}(z^{-2}))\,\mathrm{diag}(z^n, z^{-n})$ by [53]

$$\check{a}_n(w) = \big(\mathbf{Y}_n^{(1)}\big)_{11} - \big(\mathbf{Y}_{n+1}^{(1)}\big)_{11}, \qquad \check{b}_n(w) = \sqrt{\big(\mathbf{Y}_{n+1}^{(1)}\big)_{12}\big(\mathbf{Y}_{n+1}^{(1)}\big)_{21}}. \tag{3.4}$$

Then, Riemann–Hilbert Problem 3.2.1 can be modified for orthogonal polynomials defined on a bounded set $\Sigma \subset \mathbb{R}$ by including asymptotic conditions at the boundary of Σ [100]. We specifically consider orthogonal polynomials corresponding to a Chebyshev-like weight function

$$w(x) = \sum_{j=1}^{g+1} \mathbb{1}_{[a_j,b_j]}(x)\,h_j(x)\big(\sqrt{x-a_j}\big)^{\alpha_j}\big(\sqrt{b_j-x}\big)^{\beta_j}, \qquad \alpha_j,\beta_j \in \{-1,1\}, \tag{3.5}$$

where each h_j is positive on $[a_j,b_j]$ and analytic in a neighborhood Ω_j of $[a_j,b_j]$. Here, α_j,β_j determine which Chebyshev weight function variant w behaves like on $[a_j,b_j]$. Note that the Akhiezer weight function (2.26) is a special case of (3.5). The Riemann–Hilbert problem for weight functions of the form (3.5) is given by

Riemann–Hilbert Problem 3.2.2 (Fokas–Its–Kitaev)**.** *Find a* 2×2 *matrix-valued function* $\mathbf{Y}_n(z)$ *with the following properties:*

Analyticity: $\mathbf{Y}_n(z)$ *is analytic for* $z \in \mathbb{C} \setminus \Sigma$*, admitting continuous boundary values on the interior of* Σ*.*

Jump condition: $\mathbf{Y}_n^+(z) = \mathbf{Y}_n^-(z) \begin{pmatrix} 1 & w(z) \\ 0 & 1 \end{pmatrix}$, $z \in \Sigma$.

Normalization: $\mathbf{Y}_n(z) = \big(\mathbf{I} + \mathrm{O}(z^{-1})\big) \begin{pmatrix} z^n & 0 \\ 0 & z^{-n} \end{pmatrix}$ *as* $z \to \infty$.

Endpoint behavior: *For each* $j = 1, \ldots, g+1$*, entry-wise,*

$$\mathbf{Y}_n(z) = \mathrm{O}\begin{pmatrix} 1 & 1 + |z - a_j|^{\alpha_j/2} \\ 1 & 1 + |z - a_j|^{\alpha_j/2} \end{pmatrix}, \quad z \to a_j,$$
$$\mathbf{Y}_n(z) = \mathrm{O}\begin{pmatrix} 1 & 1 + |z - b_j|^{\beta_j/2} \\ 1 & 1 + |z - b_j|^{\beta_j/2} \end{pmatrix}, \quad z \to b_j.$$

The unique solution is still given by (3.3), and the recurrence coefficients can still be recovered via (3.4). An important feature of this formulation is that the degree of the polynomial enters the RHP only as a parameter. In Chapter 4, we will develop a numerical method to solve Riemann–Hilbert Problem 3.2.2. Since the computational cost of this numerical method does not grow as n increases, this will enable us to compute n pairs of recurrence coefficients in $\mathrm{O}(n)$ arithmetic operations.

3.2.2 A KdV soliton gas RHP

In [78], the authors derive a Riemann–Hilbert problem describing certain soliton gasses in the context of the KdV equation (1.3). Here, we present only the RHP to demonstrate its strikingly similar structure to Riemann–Hilbert Problem 3.2.2; we will present the full details in Chapter 7.

Let $\Sigma_+ = \bigcup_{j=1}^g [a_j, b_j]$ and $\Sigma_- = \bigcup_{j=1}^g [-b_j, -a_j]$ be symmetric unions of real disjoint intervals, both oriented in the increasing direction. These intervals correspond to the spectral parameter of the solitons that accumulate into the gas. We consider a reflection coefficient of the form

$$r(z) = \sum_{j=1}^{g} \mathbb{1}_{[a_j, b_j]}(z) f_j(z) \big(\sqrt{z - a_j}\big)^{\alpha_j} \big(\sqrt{b_j - z}\big)^{\beta_j}, \quad z \in \Sigma_+, \quad \alpha_j, \beta_j \in \{-1, 1\},$$

where each f_j is positive on $[a_j, b_j]$ and analytic in a neighborhood of $[a_j, b_j]$, extended to Σ_- by $r(-z) = r(z)$. This function describes how the solitons accumulate. Essentially, r has the same structure and role as the Chebyshev-like class of weight functions (3.5). The behavior of the soliton gas is captured by the following (rotated) RHP:

Riemann–Hilbert Problem 3.2.3 (Pure KdV soliton gas)**.** *Find a* 1×2 *row-vector-valued function* $\mathbf{Y}:\mathbb{C}\setminus(\Sigma_+\cup\Sigma_-)\to\mathbb{C}^{1\times 2}$ *with the following properties:*

Analyticity: $\mathbf{Y}(z)$ *is analytic for* $z\in\mathbb{C}\setminus(\Sigma_+\cup\Sigma_-)$*, admitting continuous boundary values on the interior of* $\Sigma_+\cup\Sigma_-$*.*

Jump conditions:

$$\mathbf{Y}^+(z)=\mathbf{Y}^-(z)\begin{pmatrix}1 & 0\\ -\mathrm{i}r(z)\mathrm{e}^{-2xz+8tz^3} & 1\end{pmatrix},\quad z\in\Sigma_+, \tag{3.6}$$

$$\mathbf{Y}^+(z)=\mathbf{Y}^-(z)\begin{pmatrix}1 & \mathrm{i}r(z)\mathrm{e}^{2xz-8tz^3}\\ 0 & 1\end{pmatrix},\quad z\in\Sigma_-. \tag{3.7}$$

Normalization: $\mathbf{Y}(z)=\begin{pmatrix}1 & 1\end{pmatrix}+\mathrm{O}(z^{-1})$ *as* $z\to\infty$*.*

Endpoint behavior: *For each* $j=1,\dots,g$*, entry-wise,*

$$\mathbf{Y}(z)=\mathrm{O}\begin{pmatrix}|z-a_j|^{\alpha_j/2} & 1\end{pmatrix},\quad z\to a_j,$$
$$\mathbf{Y}(z)=\mathrm{O}\begin{pmatrix}|z-b_j|^{\beta_j/2} & 1\end{pmatrix},\quad z\to b_j,$$

with the columns reversed at the endpoints of Σ_-*.*

Symmetry: $\mathbf{Y}(-z)=\mathbf{Y}(z)\boldsymbol{\sigma}_1$*.*

This problem has a unique solution [78]. The KdV solution can be recovered by

$$u(x,t)=\lim_{z\to\infty}2z^2\big(y_1(z)y_2(z)-1\big),$$

where y_1,y_2 are the components of $\mathbf{Y}$.

3.3 Numerical methods

Now, we turn our attention to solving Riemann–Hilbert Problem 3.2.2 and Riemann–Hilbert Problem 3.2.3 numerically. Currently, these problems are not directly amenable to numerical methods in some or all regimes. The FIK RHP features singularities at infinity, and the jump conditions of the soliton gas RHP grow exponentially outside of a quiescent region in the (x,t)-plane.

Following the Deift–Zhou method of nonlinear steepest descent [55], we transform these problems into one that can be solved numerically through the following procedure:

- Step 1: Lens the Riemann–Hilbert problem, factoring and moving jump conditions to regions where we will induce exponential decay.

- Step 2: Determine and apply a g-function to either remove singularities at infinity or exponential growth from the bands.

- Step 3: Determine and apply an auxiliary function to remove the jump conditions on the gaps induced by the g-function, and "solve" the resulting problem.

This procedure is performed in full for orthogonal polynomials with support on a single interval in [99,100] and for multiple intervals in [58,165]. It is applied for soliton gasses with reflection coefficients supported on one pair of intervals in [78, 79], but has not previously been extended to the higher-genus case. The novelty of our approach lies in how Step 3 is applied and the fact that "solve" means constructing a reliable numerical approximation, not an asymptotic approximation.

Once this procedure has been applied, the deformed RHPs have jump conditions on the unions of circles and intervals. To solve these numerically, we develop methods to solve smaller and simpler problems that we then combine into a block linear system. In general, we follow the numerical prescription of [118], using Cauchy transforms to cast RHPs as singular integral equations.

3.3.1 Numerically solving scalar Riemann–Hilbert problems on a circle

As a first step, we numerically solve a Riemann–Hilbert problem with a jump condition occurring only on the unit circle. Letting $\mathbb{U}$ denote the unit circle in the complex plane oriented counterclockwise, consider the problem of finding a function $\Phi : \mathbb{C} \setminus \mathbb{U} \to \mathbb{C}$ that satisfies

$$\begin{aligned} \Phi \text{ is analytic in } \mathbb{C} \setminus \mathbb{U}, \\ \Phi^+(z) = \Phi^-(z) f(z), \quad z \in \mathbb{U}, \\ \lim_{z\to\infty} \Phi(z) = 1. \end{aligned} \tag{3.8}$$

To find a numerical solution, we make the ansatz

$$\Phi(z) = 1 + \mathcal{C}_{\mathbb{U}} u(z),$$

for some unknown function u; see [159, Section 2.5] for a rigorous discussion of this choice. Plugging this into the jump condition in (3.8),

$$\mathcal{C}^+_{\mathbb{U}} u(z) - f(z)\mathcal{C}^-_{\mathbb{U}} u(z) = f(z) - 1. \tag{3.9}$$

Now, we let u be given by a Laurent series

$$u(z) = \sum_{j=-\infty}^{\infty} c_j z^j.$$

A simple residue calculation yields that

$$\mathcal{C}_{\mathbb{U}}u(z) = \begin{cases} \sum_{j=0}^{\infty} c_j z^j, & |z| < 1, \\ -\sum_{j=-\infty}^{-1} c_j z^j, & |z| > 1. \end{cases} \tag{3.10}$$

To form a numerical solution, we truncate our Laurent series so that its terms range from $-N$ to N and consider $2N+1$ collocation points $z_0, \ldots, z_{2N}$ evenly distributed on $\mathbb{U}$ [2]. We then determine coefficients $c_{-N}, \ldots, c_N$ such that (3.9) is satisfied at all collocation points:

$$\sum_{j=0}^{N} c_j z_k^j + f(z_k) \sum_{j=-N}^{-1} c_j z_k^j = f(z_k) - 1, \tag{3.11}$$

for $k = 0, \ldots, 2N$. To solve for the coefficients, we write (3.11) as a linear system and solve for $c_{-N}, \ldots, c_N$. We then use (3.10) to evaluate our numerical solution $\Phi(z)$ at any $z \in \mathbb{C} \setminus \mathbb{U}$. One can of course generalize this to numerically solve a scalar Riemann–Hilbert problem on any circle by scaling and shifting appropriately [118].

3.3.2 Numerically solving scalar Riemann–Hilbert problems on an interval

Now, we numerically solve the Riemann–Hilbert problem of finding a function $\Phi : \mathbb{C} \setminus \mathbb{I} \to \mathbb{C}$ that satisfies

$$\begin{aligned} &\Phi \text{ is analytic in } \mathbb{C} \setminus \mathbb{I}, \\ &\Phi^+(z) = \Phi^-(z) f(z), \quad z \in \mathbb{I}, \\ &\lim_{z \to \infty} \Phi(z) = 1, \end{aligned}$$

where $\mathbb{I} = [-1, 1]$. We again make the ansatz

$$\Phi(z) = 1 + \mathcal{C}_{\mathbb{I}} u(z),$$

but need a different approach to determine u. As with most numerical analysis problems on the unit interval, we cannot safely use a monomial basis and equally-spaced points as in the problem on the unit circle. Instead, we need to use an orthogonal polynomial basis, meaning that we will need to use the Cauchy integrals of orthogonal polynomials when enforcing our jump condition of the form (3.9) but on $\mathbb{I}$. We choose $N+1$ collocation points to be more concentrated near the endpoints as the roots of Chebyshev-$\underline{T}$, i.e., we set

$$z_k = \cos\left(\frac{(2k+1)\pi}{2(N+1)}\right), \quad k = 0, \ldots, N.$$

---

[2] For instance, given by $z_k = \mathrm{e}^{2\pi \mathrm{i} k/(2N+1)}$.

If we use an orthogonal polynomial basis $p_0, \ldots, p_N$ with associated weight function w, then we find coefficients $d_0, \ldots, d_N$ such that the jump condition holds at all collocation points:

$$\sum_{j=0}^{N} d_j \mathcal{C}_{\mathbb{I}}^{+}[p_j w](z_k) - f(z_k) \sum_{j=0}^{N} d_j \mathcal{C}_{\mathbb{I}}^{-}[p_j w](z_k) = f(z_k) - 1,$$

for $k = 0, \ldots, N$. This is a linear system and can (potentially) be solved for $d_0, \ldots, d_N$ by computing $\mathcal{C}_{\mathbb{I}}^{+}[p_j w](z_k), \mathcal{C}_{\mathbb{I}}^{-}[p_j w](z_k)$ as described in accordance with Theorem 2.5.1 if we choose to use a Chebyshev basis for p_j, $j = 0, \ldots, N$. Of course, this problem can be shifted and scaled to an arbitrary interval $[a, b]$ [118].

While solving this linear system appears straightforward, a potential difficulty arises in choosing the orthogonal polynomial basis used to build the numerical solution. If the correct basis is not chosen, the method will not converge. The Cauchy integrals of different orthogonal polynomial bases have different asymptotic behavior near the endpoints $z \to \pm 1$, and therefore different choices of bases will produce functions that behave very differently. If one wishes to pose a Riemann–Hilbert problem with a unique solution, additional asymptotic conditions must be prescribed at the endpoints. Then, the correct orthogonal polynomial basis associated with that asymptotic condition must be selected.

3.3.3 Combining methods for multiple contours and matrix problems

Consider solving a Riemann–Hilbert problem on the contour $\Gamma = \Gamma_1 \cup \Gamma_2$ where Γ_1, Γ_2 are disjoint contours: Find a function $\Phi : \mathbb{C} \setminus \Gamma \to \mathbb{C}$ that satisfies

$$\begin{aligned}
&\Phi \text{ is analytic in } \mathbb{C} \setminus \Gamma, \\
&\Phi^{+}(z) = \Phi^{-}(z) f_1(z), \quad z \in \Gamma_1, \\
&\Phi^{+}(z) = \Phi^{-}(z) f_2(z), \quad z \in \Gamma_2, \\
&\lim_{z \to \infty} \Phi(z) = 1.
\end{aligned}$$

Define the projection operator P_j by

$$P_j u(z) = \begin{cases} u(z) & z \in \Gamma_j, \\ 0 & \text{otherwise.} \end{cases}$$

The ansatz becomes

$$\Phi(z) = 1 + \mathcal{C}_\Gamma u_1(z) + \mathcal{C}_\Gamma u_2(z),$$

where $u_j = P_j u$ for some unknown u. Noting that $\mathcal{C}_\Gamma = \mathcal{C}_{\Gamma_1} + \mathcal{C}_{\Gamma_2}$, the jump condition becomes

$$P_j \mathcal{C}_{\Gamma_j}^{+} u_j + P_j \mathcal{C}_{\Gamma_k}^{+} u_k - f_j (P_j \mathcal{C}_{\Gamma_j}^{-} u_j + P_j \mathcal{C}_{\Gamma_k}^{-} u_k) = f_j - 1,$$

for $j,k \in \{1,2\}$, $j \neq k$. This gives the block system

$$\begin{pmatrix} P_1\mathcal{C}^+_{\Gamma_1} - f_1P_1\mathcal{C}^-_{\Gamma_1} & P_1\mathcal{C}_{\Gamma_2} - f_1P_1\mathcal{C}_{\Gamma_2} \\ P_2\mathcal{C}_{\Gamma_1} - f_2P_2\mathcal{C}_{\Gamma_1} & P_2\mathcal{C}^+_{\Gamma_2} - f_2P_2\mathcal{C}^-_{\Gamma_2} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} f_1 - 1 \\ f_2 - 1 \end{pmatrix}, \tag{3.12}$$

where $P_j\mathcal{C}^+_{\Gamma_k} = P_j\mathcal{C}^-_{\Gamma_k} = P_j\mathcal{C}_{\Gamma_k}$ when $j \neq k$. We expand u_1, u_2 in the usual way[3] and choose collocation points for each on Γ_1 and Γ_2, respectively. Then, we enforce the jump conditions at the collocation points as a block linear system via the matrix structure (3.12) and solve for the basis coefficients for both u_1 and u_2. A block system can be formulated analogously when Γ is composed of an arbitrary number of disjoint contours. In this case, the (j,k)th entry in the block matrix is given by $P_j\mathcal{C}^+_{\Gamma_k} - f_jP_j\mathcal{C}^-_{\Gamma_k}$.

We can extend these methods to a matrix Riemann–Hilbert problem for $\mathbf{\Phi} : \mathbb{C}\backslash\Gamma \to \mathbb{C}^{2\times 2}$

$$\begin{aligned} &\mathbf{\Phi} \text{ is analytic in } \mathbb{C} \setminus \Gamma, \\ &\mathbf{\Phi}^+(z) = \mathbf{\Phi}^-(z)\mathbf{F}(z), \quad z \in \Gamma, \\ &\lim_{z\to\infty} \mathbf{\Phi}(z) = \mathbf{I}. \end{aligned}$$

If one considers the ansatz

$$\mathbf{\Phi}(z) = \mathbf{I} + \mathcal{C}_\Gamma \mathbf{U}(z), \tag{3.13}$$

where $\mathbf{U}$ is now matrix-valued and the Cauchy operator is taken element-wise, we find a system of the form

$$\begin{pmatrix} \mathcal{C}^+_\Gamma - \mathbf{F}_{11}\mathcal{C}^-_\Gamma & -\mathbf{F}_{21}\mathcal{C}^-_\Gamma \\ -\mathbf{F}_{12}\mathcal{C}^-_\Gamma & \mathcal{C}^+_\Gamma - \mathbf{F}_{22}\mathcal{C}^-_\Gamma \end{pmatrix} \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{21} \\ \mathbf{U}_{12} & \mathbf{U}_{22} \end{pmatrix} = \mathbf{F}^T - \mathbf{I}.$$

We solve this using our methods for scalar problems by forming a block system. Solving a matrix problem on multiple domains will require forming a block system of block systems [118]. The equations for $\mathbf{U}_{11}, \mathbf{U}_{12}$ and $\mathbf{U}_{21}, \mathbf{U}_{22}$ decouple, so $\mathbf{U}$ can be solved for row-wise.

3.3.4 Determining bases

As mentioned in Section 3.3.2, it is very important to choose a basis with the appropriate endpoint behavior. To illustrate this, consider a singular integral equation of the form

$$\mathcal{C}^+_\Gamma \mathbf{U}(z) - \mathcal{C}^-_\Gamma \mathbf{U}(z)\mathbf{J}(z) = \mathbf{J}(z) - \mathbf{I}, \quad z \in \Gamma,$$

[3] They are expanded using either a Laurent series or an appropriate orthogonal polynomial basis depending on if Γ_1, Γ_2 are intervals or circles.

where $\Gamma = \Gamma_1 \cup \Gamma_2$ are disjoint, $\Gamma_1 = [-1, 1]$, oriented from $z = -1$ to $z = 1$, and

$$\mathbf{J}|_{\Gamma_1}(z) = \begin{pmatrix} 0 & \mathrm{i}h(z) \\ \mathrm{i}/h(z) & 0 \end{pmatrix}, \quad h(z) = \sqrt{1 - z^2}.$$

This has the same off-diagonal $\begin{pmatrix} 0 & * \\ * & 0 \end{pmatrix}$ structure as the jump matrix $F_j^2(z)$ in the deformed Riemann–Hilbert Problem 4.2.5 below on each interval $[a_j, b_j]$; the weight function $w_j(z)$ plays the role of $h(z)$ and carries the same square-root endpoint singularities. We build a local solution near $z = 1$ to determine the singularity structure that $\mathbf{U}$ must inherit. The same local analysis applies to the soliton gas RHP in Chapter 7 in the unmodulated region. In the quiescent region, the RHP has simpler triangular jump matrices and local solutions can be found directly by Cauchy integrals.

Consider $\ell(z) = (z-1)^{1/4}(z+1)^{1/4}$, where we use the principal branch cut along $(-\infty, 0]$ for the quarter root function. It follows that $\ell^+(z)\ell^-(z) = h(z)$ for $z \in [-1, 1]$. Define

$$\mathbf{L}(z) = \ell(z)^{-\boldsymbol{\sigma}_3}.$$

We compute, in a neighborhood of $z = 1$,

$$\mathbf{L}^-(z)\mathbf{J}|_{\Gamma_1}(z)\mathbf{L}^+(z)^{-1} = \begin{pmatrix} 0 & \mathrm{i}\frac{h(z)}{\ell^+(z)\ell^-(z)} \\ \mathrm{i}\frac{\ell^+(z)\ell^-(z)}{h(z)} & 0 \end{pmatrix} = \begin{pmatrix} 0 & \mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}.$$

Then, we factorize

$$\begin{pmatrix} 0 & \mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix} = \mathbf{Q}\, a^-(z)^{-\boldsymbol{\sigma}_3} a^+(z)^{\boldsymbol{\sigma}_3} \mathbf{Q}^{-1}, \quad \mathbf{Q} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \quad a(z) = \left(\frac{z-1}{z+1} \right)^{1/4},$$

giving

$$a^-(z)^{\boldsymbol{\sigma}_3} \mathbf{Q}^{-1} \mathbf{L}^-(z) \mathbf{J}|_{\Gamma_1}(z) \mathbf{L}^+(z)^{-1} \mathbf{Q}\, a^+(z)^{-\boldsymbol{\sigma}_3} = \mathbf{I}.$$

All of this is to say that

$$\boldsymbol{\Psi}(z) = a(z)^{\boldsymbol{\sigma}_3} \mathbf{Q}^{-1} \mathbf{L}(z) = \frac{1}{\sqrt{2}} \begin{pmatrix} \frac{1}{\sqrt{z+1}} & \sqrt{z-1} \\ -\frac{1}{\sqrt{z-1}} & \sqrt{z+1} \end{pmatrix}$$

is a local solution to the RHP near $z = 1$, where the square roots denote the principal branch cut along $(-\infty, 0]$. Then define, for ϵ small,

$$\widehat{\mathbf{Y}}(z) = \begin{cases} \mathbf{Y}(z) & |z-1| > \epsilon, \\ \mathbf{Y}(z)\boldsymbol{\Psi}(z)^{-1} & |z-1| < \epsilon, \end{cases}$$

which is an analytic matrix-valued function near $z = 1$ (using the requisite endpoint conditions). We find the representation $\mathbf{Y}(z) = \widehat{\mathbf{Y}}(z)\mathbf{\Psi}(z)$ for z near $z = 1$. To back out the behavior of $\mathbf{U}$, we use Lemma 3.1.1,

$$\begin{aligned}\mathbf{U}(z) = \mathbf{Y}^+(z) - \mathbf{Y}^-(z) = \widehat{\mathbf{Y}}(z)\left(\mathbf{\Psi}^+(z) - \mathbf{\Psi}^-(z)\right)\\ = \widehat{\mathbf{Y}}(z)\begin{pmatrix} 0 & \sqrt{2(z-1)} \\ -\sqrt{2/(z-1)} & 0 \end{pmatrix}, \quad |z-1| < \epsilon.\end{aligned}$$

A similar analysis near $z = -1$, and patching the two together, gives

$$\mathbf{U}|_{\Gamma_1}(z) = \mathbf{A}(z)\begin{pmatrix} 0 & \sqrt{1-z^2} \\ \frac{1}{\sqrt{1-z^2}} & 0 \end{pmatrix},$$

where $\mathbf{A}(z)$ is an analytic matrix-valued function that can therefore be approximated well with polynomials of low degree. This local analysis motivates the basis choices in the subsection below: the first column of $\mathbf{U}$ behaves like $1/\sqrt{1-z^2}$ and should be expanded using the Chebyshev-$\underline{T}$ weight, while the second column behaves like $\sqrt{1-z^2}$ and should use the Chebyshev-U weight. The same logic applies column-by-column when $h(z)$ has mixed endpoint types, leading to the V- and W-type bases.

Chapter 4

COMPUTING ORTHOGONAL POLYNOMIALS ON HIGHER-GENUS DOMAINS

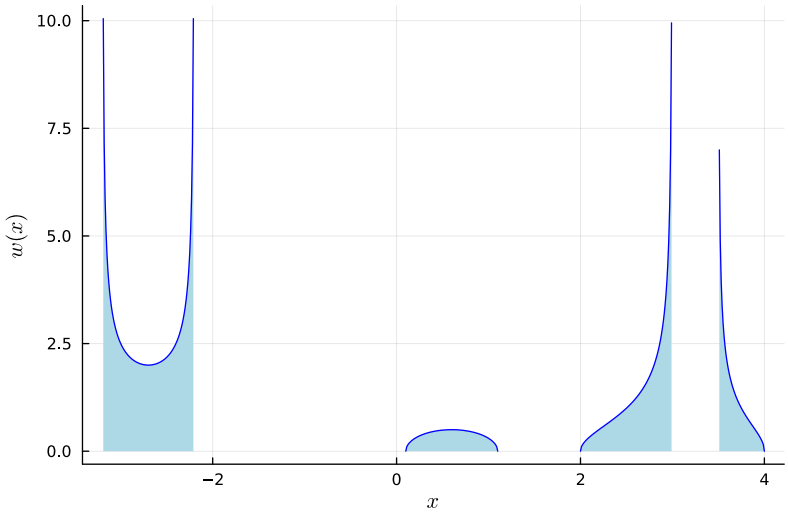


Figure 4.1: A weight function of the form (3.5) with support on $[-3.2, -2.2] \cup [0.1, 1.1] \cup [2, 3] \cup [3.5, 4]$ with $\alpha_1 = \beta_1 = -1$ (Chebyshev-$\underline{T}$), $\alpha_2 = \beta_2 = 1$ (Chebyshev-U), $\alpha_3 = 1, \beta_3 = -1$ (Chebyshev-V), $\alpha_4 = -1, \beta_4 = 1$ (Chebyshev-W).

4.1 Introduction

This chapter is based on [18]. An associated software package and code used to generate the plots in this chapter can be found at [13].

In this chapter, we develop and apply a numerical Riemann–Hilbert approach to solve Riemann–Hilbert Problem 3.2.2 and compute the recurrence coefficients for orthogonal polynomials corresponding to weight functions of the form (3.5). Such orthogonal polynomials have both analytical and computational applications [11, 49, 77, 132]. For example, such orthogonal polynomials produce a new class of iterative solvers [68, 132], improve matrix spectrum approximation when bounds on the eigenvalues are known a priori [45], and describe the numerical evolution of a Toda lattice [54]. Analytical formulae are not known for most of the cases we consider. The method remains accurate as the degree of the polynomial under consideration grows, and it bypasses the need to compute theta functions. We include a plot of a weight function of this form in Figure 4.1. In most cases, it will suffice to consider $h_j(x) = 1$ for all j, and recent work [84] has demonstrated that recurrence coefficients corresponding to a different choice of h_j can be obtained from those corresponding to $h_j(x) = 1$. In principle, our weight function (3.5) could be generalized to a Jacobi-like weight by allowing for any $\alpha_j, \beta_j > -2$; however this would complicate our numerics significantly, as it would result in a Riemann–Hilbert problem defined on self-intersecting contours. Furthermore, the asymptotics are different if α_j, β_j are not integers, with error terms behaving polynomially rather than exponentially [100]. Thus, while our method could, in principle, be extended

to an $\mathrm{O}(N)$ method for Jacobi-like weights, this method would likely have a much larger leading constant than the method that we consider.

We follow the notation of Chapter 2 for orthogonal polynomials and Cauchy integrals throughout this chapter. Section 4.2 carries out the Deift–Zhou steepest-descent deformations of the FIK problem, Section 4.3 recovers recurrence coefficients from the deformed problem, and Section 4.4 discusses simplifications for special weight functions. Section 4.5 develops the numerical implementation, with the shared Cauchy-integral collocation framework described in Chapter 3, Section 3.3. Our method computes the first 51 recurrence coefficients for a Chebyshev-$\underline{T}$-like weight ($\alpha_j = \beta_j = -1$) on the domain $[-1.8, -1] \cup [2, 3]$ to machine precision in 0.47 seconds[1]. A further timing comparison with an optimized $\mathrm{O}(N^2)$ method [81] can be found in Section 4.7.2. In Section 4.7, we demonstrate the numerical accuracy of our method and apply our method to evolving a Toda lattice and approximating $1/x$ on a disconnected domain. Our method could also potentially be extended to compute the associated N-point Gauss quadrature nodes and weights for (3.5) in $\mathrm{O}(N)$ arithmetic operations using techniques similar to [36, 85, 90, 150]. Many extra complexities are introduced for (3.5), including the possibility of orthogonal polynomials having roots outside the support of the weight. For this approach to have any advantage over using the individual rules for each subinterval $[a_j, b_j]$, it is likely that g will need to be large.

4.2 Transforming the Riemann–Hilbert problem

Step 1: Lensing

Consider

$$w_j(x) = h_j(x) \left(\sqrt{x - a_j}\right)^{\alpha_j} \left(\sqrt{b_j - x}\right)^{\beta_j}.$$

This function is analytic on $\Omega_j \setminus (\mathbb{R} \setminus (a_j, b_j))$. Let C_j be a counterclockwise-oriented curve lying in Ω_j that encircles, but does not intersect $[a_j, b_j]$, and denote its interior by D_j. In the end, our numerical method will take C_j to be a circle, imposing additional restrictions on Ω_j. We include examples of these regions in Figure 4.2. Define the transformation

$$\mathbf{Z}_n(z) = \begin{cases} \mathbf{Y}_n(z) \begin{pmatrix} 1 & 0 \\ -1/w_j(z) & 1 \end{pmatrix} & z \in D_j \cap \mathbb{C}^+, \\ \mathbf{Y}_n(z) \begin{pmatrix} 1 & 0 \\ 1/w_j(z) & 1 \end{pmatrix} & z \in D_j \cap \mathbb{C}^-, \\ \mathbf{Y}_n(z) & \text{otherwise}, \end{cases}$$

[1]All computations in this chapter are performed on a Lenovo laptop running Ubuntu version 20.04 with 8 cores and 16 GB of RAM with an Intel® Core™ i7-11800H processor running at 2.30 GHz.

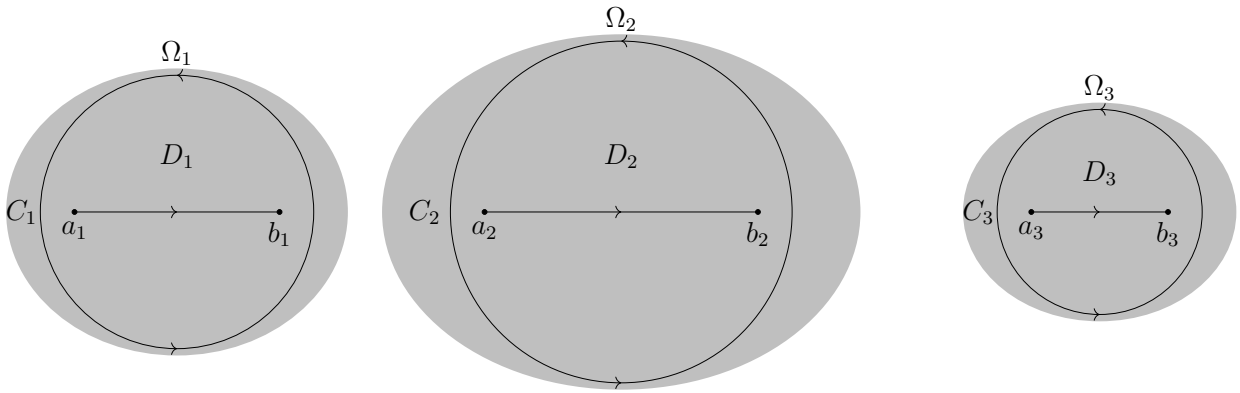


Figure 4.2: Sample regions Ω_j and D_j and curves C_j for a three interval case.

for $j = 1, \ldots, g+1$. It follows that for $z \in C_j \cap \mathbb{C}^+$,

$$\mathbf{Z}_n^+(z) = \mathbf{Z}_n^-(z) \begin{pmatrix} 1 & 0 \\ -1/w_j(z) & 1 \end{pmatrix},$$

and for $z \in C_j \cap \mathbb{C}^-$,

$$\mathbf{Z}_n^+(z) = \mathbf{Z}_n^-(z) \begin{pmatrix} 1 & 0 \\ 1/w_j(z) & 1 \end{pmatrix}.$$

For $z \in (a_j, b_j)$,

$$\begin{aligned} \mathbf{Z}_n^+(z) &= \mathbf{Y}_n^+(z) \begin{pmatrix} 1 & 0 \\ -1/w_j(z) & 1 \end{pmatrix} = \mathbf{Y}_n^-(z) \begin{pmatrix} 1 & w_j(z) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1/w_j(z) & 1 \end{pmatrix} \\ &= \mathbf{Z}_n^-(z) \begin{pmatrix} 1 & 0 \\ -1/w_j(z) & 1 \end{pmatrix} \begin{pmatrix} 1 & w_j(z) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1/w_j(z) & 1 \end{pmatrix} = \mathbf{Z}_n^-(z) \begin{pmatrix} 0 & w_j(z) \\ -1/w_j(z) & 0 \end{pmatrix}. \end{aligned}$$

Now, we compute the jump condition on $D_j \cap (\mathbb{R} \setminus (a_j, b_j))$. For $x > b_j$ or $x < a_j$, $w_j^+(x) = -w_j^-(x)$ where now the $+$ denotes a limit from the upper half plane, and $-$ denotes a limit from the lower. Then,

$$\begin{aligned} \mathbf{Z}_n^+(z) &= \mathbf{Y}_n^+(z) \begin{pmatrix} 1 & 0 \\ -1/w_j^+(z) & 1 \end{pmatrix} = \mathbf{Y}_n^-(z) \begin{pmatrix} 1 & 0 \\ -1/w_j^+(z) & 1 \end{pmatrix} \\ &= \mathbf{Z}_n^-(z) \begin{pmatrix} 1 & 0 \\ -1/w_j^+(z) & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -1/w_j^-(z) & 1 \end{pmatrix} \\ &= \mathbf{Z}_n^-(z) \begin{pmatrix} 1 & 0 \\ -\left(\frac{1}{w_j^+(z)} + \frac{1}{w_j^-(z)}\right) & 1 \end{pmatrix} = \mathbf{Z}_n^-(z). \end{aligned}$$

We find that there is no additional jump condition induced between $[a_j, b_j]$ and C_j. This is a special situation in which no local parametrices are required to compute the asymptotics of the polynomials. The asymptotic behavior at the endpoints is multiplied by $\pm 1/w_j(z)$ in this transformation, and the lensed Riemann–Hilbert problem is as follows.

Riemann–Hilbert Problem 4.2.1 (Lensed FIK)**.** *Find a* 2×2 *matrix-valued function* $\mathbf{Z}_n(z)$ *with the following properties:*

> ***Analyticity:*** $\mathbf{Z}_n(z)$ *is analytic for* $z \in \mathbb{C} \setminus \left(\bigcup_{j=1}^{g+1} C_j \cup [a_j, b_j] \right)$*, admitting continuous boundary values on each contour.*
>
> ***Jump conditions:*** *The boundary values* $\mathbf{Z}_n^+(z)$ *(resp.* $\mathbf{Z}_n^-(z)$*) from the interior (resp. exterior) of* C_j *on the circular arcs and from above (resp. below) on the real intervals are related via*
>
> $$\begin{aligned}
> \mathbf{Z}_n^+(z) &= \mathbf{Z}_n^-(z) \begin{pmatrix} 1 & 0 \\ -1/w_j(z) & 1 \end{pmatrix}, && z \in C_j \cap \mathbb{C}^+, \\
> \mathbf{Z}_n^+(z) &= \mathbf{Z}_n^-(z) \begin{pmatrix} 1 & 0 \\ 1/w_j(z) & 1 \end{pmatrix}, && z \in C_j \cap \mathbb{C}^-, \\
> \mathbf{Z}_n^+(z) &= \mathbf{Z}_n^-(z) \begin{pmatrix} 0 & w_j(z) \\ -1/w_j(z) & 0 \end{pmatrix}, && z \in (a_j, b_j), \quad j = 1, \ldots, g+1.
> \end{aligned}$$
>
> ***Normalization:*** $\mathbf{Z}_n(z) = (\mathbf{I} + \mathrm{O}(z^{-1})) \begin{pmatrix} z^n & 0 \\ 0 & z^{-n} \end{pmatrix}$ *as* $z \to \infty$*.*
>
> ***Asymptotic condition (endpoints):*** *For each* $j = 1, \ldots, g+1$*, taken entry-wise,*
>
> $$\begin{aligned}
> \mathbf{Z}_n(z) &= \mathrm{O} \begin{pmatrix} 1 + |z - a_j|^{-\alpha_j/2} & 1 + |z - a_j|^{\alpha_j/2} \\ 1 + |z - a_j|^{-\alpha_j/2} & 1 + |z - a_j|^{\alpha_j/2} \end{pmatrix}, \quad z \to a_j, \\
> \mathbf{Z}_n(z) &= \mathrm{O} \begin{pmatrix} 1 + |z - b_j|^{-\beta_j/2} & 1 + |z - b_j|^{\beta_j/2} \\ 1 + |z - b_j|^{-\beta_j/2} & 1 + |z - b_j|^{\beta_j/2} \end{pmatrix}, \quad z \to b_j.
> \end{aligned}$$

Step 2: The differential $\mathfrak{g}'_\Sigma$

To remove the growth/decay at infinity, we seek a function $\mathfrak{g}_\Sigma$ that satisfies the following:

- $\mathfrak{g}'_\Sigma(z) = z^{-1} + \mathrm{O}(z^{-2})$ as $z \to \infty$.
- $\mathfrak{g}'^+_\Sigma(z), \mathfrak{g}'^-_\Sigma(z)$ are purely imaginary on $[a_j, b_j]$ for $j = 1, \ldots, g+1$.
- $\int_{b_j}^{a_{j+1}} \mathfrak{g}'_\Sigma(z) \mathrm{d}z = 0$ for $j = 1, \ldots, g$.

It is worth noting that this construction first appeared in [164], predating the modern notion of g-functions [56]. We define

$$\mathfrak{g}'_\Sigma(z) = \frac{Q_g(z)}{R(z)}, \quad \text{where } R(z)^2 = \prod_{j=1}^{g+1}(z-a_j)(z-b_j),$$

such that $R(z) \sim z^{g+1}$ as $z \to \infty$. A genus-g hyperelliptic Riemann surface is given by $\{(z,w) \in \mathbb{C}^2 : w^2 = R(z)^2\}$. Here, Q_g is a monic polynomial of degree g

$$Q_g(z) = z^g + \sum_{k=0}^{g-1} h_k z^k,$$

whose coefficients h_k are determined by the linear system

$$\int_{b_j}^{a_{j+1}} \sum_{k=0}^{g-1} h_k \frac{z^k}{R(z)} \mathrm{d}z = -\int_{b_j}^{a_{j+1}} \frac{z^g}{R(z)} \mathrm{d}z, \tag{4.1}$$

for $j = 1, \ldots, g$.

Lemma 4.2.2. *The $g \times g$ matrix $\mathbf{G}$ defined by*

$$\mathbf{G} = \left(\int_{b_j}^{a_{j+1}} \frac{s^{k-1}}{R(s)} \mathrm{d}s \right)_{1 \le j,k \le g},$$

is invertible.

The proof is adapted from that of [135, Lemma 2].

Proof. Any vector in the nullspace of $\mathbf{G}$ corresponds to a polynomial q_{g-1} of degree at most $g-1$ such that

$$\int_{b_j}^{a_{j+1}} \frac{q_{g-1}(s)}{R(s)} \mathrm{d}s = 0,$$

for $j = 1, \ldots, g$. For $s \in (b_j, a_{j+1})$, the function $R(s)$ is real valued and sign definite. Thus, q_{g-1} has at least one zero in each gap (b_j, a_{j+1}) for $j = 1, \ldots, g$. Because q_{g-1} has g zeroes, it is identically zero, so the nullspace of $\mathbf{G}$ consists of only the zero vector. □

Then, we define $\mathfrak{g}_\Sigma$ by normalizing at the leftmost endpoint. If some function satisfies $\psi' = \mathfrak{g}'_\Sigma$, then we let

$$\mathfrak{g}_\Sigma(z) = \psi(z) - \psi(a_1).$$

We also define

$$\Delta_j = \mathfrak{g}_\Sigma^+(z) - \mathfrak{g}_\Sigma^-(z) = 2\sum_{k=1}^{j} \int_{a_k}^{b_k} (\mathfrak{g}'_\Sigma)^+(z) \mathrm{d}z \in \mathrm{i}\mathbb{R}, \tag{4.2}$$

for $z \in (b_j, a_{j+1})$, $j = 1, \ldots, g$, where this value is independent of z. We have that

$$\mathfrak{g}_\Sigma^+(z) + \mathfrak{g}_\Sigma^-(z) = 0,$$

for $z \in (a_j, b_j)$. Now, we consider the transformation

$$\mathbf{S}_n(z) = \mathfrak{c}^{n\boldsymbol{\sigma}_3} \mathbf{Z}_n(z) \mathrm{e}^{-n\mathfrak{g}_\Sigma(z)\boldsymbol{\sigma}_3}, \quad \mathfrak{c} = \lim_{z\to\infty} \frac{\mathrm{e}^{\mathfrak{g}_\Sigma(z)}}{z},$$

i.e., $\mathfrak{g}_\Sigma$ has asymptotic expansion

$$\mathfrak{g}_\Sigma(z) = \log \mathfrak{c} z + \frac{(\mathfrak{g}_\Sigma)_1}{z} + \ldots, \tag{4.3}$$

as $z \to \infty$. This asymptotic form follows from the asymptotic expansion of $\mathfrak{g}_\Sigma'$ at infinity. By construction, $\mathbf{S}_n(z) = \mathbf{I} + \mathrm{O}(z^{-1})$ as $z \to \infty$. Now, the jump condition for $z \in C_j \cap \mathbb{C}^+$ is given by

$$\mathbf{S}_n^+(z) = \mathbf{S}_n^-(z) \mathrm{e}^{n\mathfrak{g}_\Sigma(z)\boldsymbol{\sigma}_3} \begin{pmatrix} 1 & 0 \\ -1/w_j(z) & 1 \end{pmatrix} \mathrm{e}^{-n\mathfrak{g}_\Sigma(z)\boldsymbol{\sigma}_3} = \mathbf{S}_n^-(z) \begin{pmatrix} 1 & 0 \\ -\mathrm{e}^{-2n\mathfrak{g}_\Sigma(z)}/w_j(z) & 1 \end{pmatrix},$$

and similarly, the jump condition for $z \in C_j \cap \mathbb{C}^-$ is given by

$$\mathbf{S}_n^+(z) = \mathbf{S}_n^-(z) \begin{pmatrix} 1 & 0 \\ \mathrm{e}^{-2n\mathfrak{g}_\Sigma(z)}/w_j(z) & 1 \end{pmatrix}.$$

For $z \in (a_j, b_j)$, the jump condition is unchanged:

$$\begin{aligned} \mathbf{S}_n^+(z) &= \mathbf{S}_n^-(z) \mathrm{e}^{n\mathfrak{g}_\Sigma^-(z)\boldsymbol{\sigma}_3} \begin{pmatrix} 0 & w_j(z) \\ -1/w_j(z) & 0 \end{pmatrix} \mathrm{e}^{-n\mathfrak{g}_\Sigma^+(z)\boldsymbol{\sigma}_3} \\ &= \mathbf{S}_n^-(z) \begin{pmatrix} 0 & \mathrm{e}^{n(\mathfrak{g}_\Sigma^+(z)+\mathfrak{g}_\Sigma^-(z))} w_j(z) \\ -\mathrm{e}^{-n(\mathfrak{g}_\Sigma^+(z)+\mathfrak{g}_\Sigma^-(z))}/w_j(z) & 0 \end{pmatrix} = \mathbf{S}_n^-(z) \begin{pmatrix} 0 & w_j(z) \\ -1/w_j(z) & 0 \end{pmatrix}. \end{aligned}$$

For $z \in (b_j, a_{j+1})$, the jump condition is given by

$$\begin{aligned} \mathbf{S}_n^+(z) &= \mathbf{S}_n^-(z) \mathrm{e}^{n\mathfrak{g}_\Sigma^-(z)\boldsymbol{\sigma}_3} \mathrm{e}^{-n\mathfrak{g}_\Sigma^+(z)\boldsymbol{\sigma}_3} \\ &= \mathbf{S}_n^-(z) \begin{pmatrix} \mathrm{e}^{-n(\mathfrak{g}_\Sigma^+(z)-\mathfrak{g}_\Sigma^-(z))} & 0 \\ 0 & \mathrm{e}^{n(\mathfrak{g}_\Sigma^+(z)-\mathfrak{g}_\Sigma^-(z))} \end{pmatrix} = \mathbf{S}_n^-(z) \begin{pmatrix} \mathrm{e}^{-n\Delta_j} & 0 \\ 0 & \mathrm{e}^{n\Delta_j} \end{pmatrix}. \end{aligned}$$

The asymptotic conditions at the endpoints remain unchanged since $|\mathfrak{g}_\Sigma|$ is bounded in the neighborhood of the contours. We have arrived at the following normalized Riemann–Hilbert problem.

Riemann–Hilbert Problem 4.2.3 (Normalized FIK)**.** *Find a* 2×2 *matrix-valued function* $\mathbf{S}_n(z)$ *with the following properties:*

Analyticity: $\mathbf{S}_n(z)$ *is analytic for* $z \in \mathbb{C}\setminus\left([a_1, b_{g+1}] \cup \bigcup_{j=1}^{g+1} C_j\right)$*, admitting continuous boundary values on each contour.*

Jump conditions: *The boundary values* $\mathbf{S}_n^+(z)$ *(resp.* $\mathbf{S}_n^-(z)$*) from the interior (resp. exterior) of* C_j *on the circular arcs and from above (resp. below) on the real intervals are related via*

$$\begin{aligned}
\mathbf{S}_n^+(z) &= \mathbf{S}_n^-(z)\begin{pmatrix} 1 & 0 \\ -\frac{\mathrm{e}^{-2n\mathfrak{g}_\Sigma(z)}}{w_j(z)} & 1 \end{pmatrix}, && z \in C_j \cap \mathbb{C}^+,\\
\mathbf{S}_n^+(z) &= \mathbf{S}_n^-(z)\begin{pmatrix} 1 & 0 \\ \frac{\mathrm{e}^{-2n\mathfrak{g}_\Sigma(z)}}{w_j(z)} & 1 \end{pmatrix}, && z \in C_j \cap \mathbb{C}^-,\\
\mathbf{S}_n^+(z) &= \mathbf{S}_n^-(z)\begin{pmatrix} 0 & w_j(z) \\ -\frac{1}{w_j(z)} & 0 \end{pmatrix}, && z \in (a_j, b_j),\\
\mathbf{S}_n^+(z) &= \mathbf{S}_n^-(z)\begin{pmatrix} \mathrm{e}^{-n\Delta_j} & 0 \\ 0 & \mathrm{e}^{n\Delta_j} \end{pmatrix}, && z \in (b_j, a_{j+1}), \quad j = 1, \dots, g.
\end{aligned}$$

Normalization: $\mathbf{S}_n(z) = \mathbf{I} + \mathrm{O}(z^{-1})$ *as* $z \to \infty$*.*

Asymptotic condition (endpoints): *For each* $j = 1, \dots, g+1$*, taken entry-wise,*

$$\begin{aligned}
\mathbf{S}_n(z) &= \mathrm{O}\begin{pmatrix} 1+|z-a_j|^{-\alpha_j/2} & 1+|z-a_j|^{\alpha_j/2} \\ 1+|z-a_j|^{-\alpha_j/2} & 1+|z-a_j|^{\alpha_j/2} \end{pmatrix}, \quad z \to a_j,\\
\mathbf{S}_n(z) &= \mathrm{O}\begin{pmatrix} 1+|z-b_j|^{-\beta_j/2} & 1+|z-b_j|^{\beta_j/2} \\ 1+|z-b_j|^{-\beta_j/2} & 1+|z-b_j|^{\beta_j/2} \end{pmatrix}, \quad z \to b_j.
\end{aligned}$$

Note that $\operatorname{Re}\mathfrak{g}_\Sigma(z) > 0$ for $z \in C_j$ (see [58, Appendix B], for example) so the jump condition on C_j tends exponentially to the identity as $n \to \infty$.

Step 3: Removing the jump conditions on the gaps

Since the normalized Riemann–Hilbert problem 4.2.3 contains jump conditions on the gaps between intervals, the jump matrix is discontinuous. In order to appropriately solve this problem numerically, we must further transform it. We do this by devising an auxiliary function to remove the jump conditions on the gaps between intervals. Consider a function

$\mathfrak{h}_n$ defined as

$$\begin{aligned}\mathfrak{h}_n(z) &= R(z)\left(\sum_{j=1}^{g+1}\frac{1}{2\pi\mathrm{i}}\int_{a_j}^{b_j}\frac{A_j(n)}{R_+(s)(s-z)}\mathrm{d}s+\sum_{\ell=1}^{g}\frac{1}{2\pi\mathrm{i}}\int_{b_\ell}^{a_{\ell+1}}\frac{\log(\mathrm{e}^{n\Delta_\ell})}{R(s)(s-z)}\mathrm{d}s\right)\\ &= R(z)\left(\sum_{j=1}^{g+1}A_j(n)\mathcal{C}_{[a_j,b_j]}\left[\frac{1}{R_+}\right](z)+\sum_{\ell=1}^{g}\log(\mathrm{e}^{n\Delta_\ell})\mathcal{C}_{[b_\ell,a_{\ell+1}]}\left[\frac{1}{R}\right](z)\right),\end{aligned}\tag{4.4}$$

where $A_1(n),\dots,A_{g+1}(n)$ are undetermined constants. Recall that on the gaps (b_j,a_{j+1}), R is analytic. Since $R(z)\sim z^{g+1}$ as $z\to\infty$, we can use the asymptotics of Cauchy integrals to obtain conditions on $A_1(n),\dots,A_{g+1}(n)$ such that $\mathfrak{h}_n(z)\sim z^{-1}$ as $z\to\infty$. We need to solve

$$\sum_{j=1}^{g+1}A_j(n)\int_{a_j}^{b_j}\frac{s^{k-1}}{R_+(s)}\mathrm{d}s=-\sum_{\ell=1}^{g}\log(\mathrm{e}^{n\Delta_\ell})\int_{b_\ell}^{a_{\ell+1}}\frac{s^{k-1}}{R(s)}\mathrm{d}s,\tag{4.5}$$

for $k=1,\dots,g+1$ for constants $A_1(n),\dots,A_{g+1}(n)$. This forms a $(g+1)\times(g+1)$ linear system. Note that $A_1(n),\dots,A_{g+1}(n)\in\mathbb{R}$ are bounded above and below due to the fact that Δ_j is purely imaginary, and the coefficient matrix does not depend on n. Furthermore, this coefficient matrix is always invertible.

Lemma 4.2.4. *The $(g+1)\times(g+1)$ matrix $\mathbf{H}$ defined by*

$$\mathbf{H}=\left(\int_{a_j}^{b_j}\frac{s^{k-1}}{R_+(s)}\mathrm{d}s\right)_{1\le k,j\le g+1},$$

is invertible.

Proof. Noting that for $s\in(a_j,b_j)$, the function $R^+(s)$ is purely imaginary with sign-definite imaginary part, the result follows from the argument of Lemma 4.2.2 applied to $\mathbf{H}^*$. □

By construction, we have that

$$\begin{cases}\mathfrak{h}_n^+(z)+\mathfrak{h}_n^-(z)=A_j(n) & z\in(a_j,b_j),\\ \mathfrak{h}_n^+(z)-\mathfrak{h}_n^-(z)=\log(\mathrm{e}^{n\Delta_j}) & z\in(b_j,a_{j+1}).\end{cases}$$

Now, consider the transformation

$$\tilde{\mathbf{S}}_n(z)=\mathbf{S}_n(z)\mathrm{e}^{\boldsymbol{\sigma}_3\mathfrak{h}_n(z)}.$$

Noting that if $\mathbf{S}_n^+(z)=\mathbf{S}_n^-(z)\mathbf{K}(z)$ on a contour, then

$$\tilde{\mathbf{S}}_n^+(z)=\tilde{\mathbf{S}}_n^-(z)\mathrm{e}^{-\boldsymbol{\sigma}_3\mathfrak{h}_n^-(z)}\mathbf{K}(z)\mathrm{e}^{\boldsymbol{\sigma}_3\mathfrak{h}_n^+(z)},$$

we find that for $z \in (b_j, a_{j+1})$,

$$\tilde{\mathbf{S}}_n^+(z) = \tilde{\mathbf{S}}_n^-(z) \begin{pmatrix} \mathrm{e}^{-\mathfrak{h}_n^-(z)} & 0 \\ 0 & \mathrm{e}^{\mathfrak{h}_n^-(z)} \end{pmatrix} \begin{pmatrix} \mathrm{e}^{-n\Delta_j} & 0 \\ 0 & \mathrm{e}^{n\Delta_j} \end{pmatrix} \begin{pmatrix} \mathrm{e}^{\mathfrak{h}_n^+(z)} & 0 \\ 0 & \mathrm{e}^{-\mathfrak{h}_n^+(z)} \end{pmatrix} = \tilde{\mathbf{S}}_n^-(z),$$

meaning that the jump condition on the gaps has been eliminated. Applying our transformation to the remaining jump conditions, we use the fact that the asymptotic conditions at the endpoints are unaffected since $|\mathfrak{h}_n|$ is bounded by [1, Lemma 7.2.2]. We arrive at the following new Riemann–Hilbert problem, which we will solve numerically.

Riemann–Hilbert Problem 4.2.5 (Deformed FIK)**.** *Find a* 2×2 *matrix-valued function* $\tilde{\mathbf{S}}_n(z)$ *with the following properties:*

Analyticity: $\tilde{\mathbf{S}}_n(z)$ *is analytic for* $z \in \mathbb{C} \setminus \left(\bigcup_{j=1}^{g+1} [a_j, b_j] \cup \bigcup_{j=1}^{g+1} C_j \right)$*, admitting continuous boundary values on each contour.*

Jump conditions: *The boundary values* $\tilde{\mathbf{S}}_n^+(z)$ *(resp.* $\tilde{\mathbf{S}}_n^-(z)$*) from the interior (resp. exterior) of* C_j *on the circular arcs and from above (resp. below) on the real intervals are related via*

$$\begin{aligned}
\tilde{\mathbf{S}}_n^+(z) &= \tilde{\mathbf{S}}_n^-(z) \begin{pmatrix} 1 & 0 \\ -\frac{1}{w_j(z)} \mathrm{e}^{2\mathfrak{h}_n(z) - 2n\mathfrak{g}_\Sigma(z)} & 1 \end{pmatrix}, && z \in C_j \cap \mathbb{C}^+, \\
\tilde{\mathbf{S}}_n^+(z) &= \tilde{\mathbf{S}}_n^-(z) \begin{pmatrix} 1 & 0 \\ \frac{1}{w_j(z)} \mathrm{e}^{2\mathfrak{h}_n(z) - 2n\mathfrak{g}_\Sigma(z)} & 1 \end{pmatrix}, && z \in C_j \cap \mathbb{C}^-, \\
\tilde{\mathbf{S}}_n^+(z) &= \tilde{\mathbf{S}}_n^-(z) \begin{pmatrix} 0 & w_j(z) \mathrm{e}^{-A_j(n)} \\ -\frac{1}{w_j(z)} \mathrm{e}^{A_j(n)} & 0 \end{pmatrix}, && z \in (a_j, b_j), \quad j = 1, \ldots, g+1.
\end{aligned}$$

Normalization: $\tilde{\mathbf{S}}_n(z) = \mathbf{I} + \mathrm{O}(z^{-1})$ *as* $z \to \infty$*.*

Asymptotic condition (endpoints): *For each* $j = 1, \ldots, g+1$*, taken entry-wise,*

$$\begin{aligned}
\tilde{\mathbf{S}}_n(z) &= \mathrm{O} \begin{pmatrix} 1 + |z - a_j|^{-\alpha_j/2} & 1 + |z - a_j|^{\alpha_j/2} \\ 1 + |z - a_j|^{-\alpha_j/2} & 1 + |z - a_j|^{\alpha_j/2} \end{pmatrix}, \quad z \to a_j, \\
\tilde{\mathbf{S}}_n(z) &= \mathrm{O} \begin{pmatrix} 1 + |z - b_j|^{-\beta_j/2} & 1 + |z - b_j|^{\beta_j/2} \\ 1 + |z - b_j|^{-\beta_j/2} & 1 + |z - b_j|^{\beta_j/2} \end{pmatrix}, \quad z \to b_j.
\end{aligned}$$

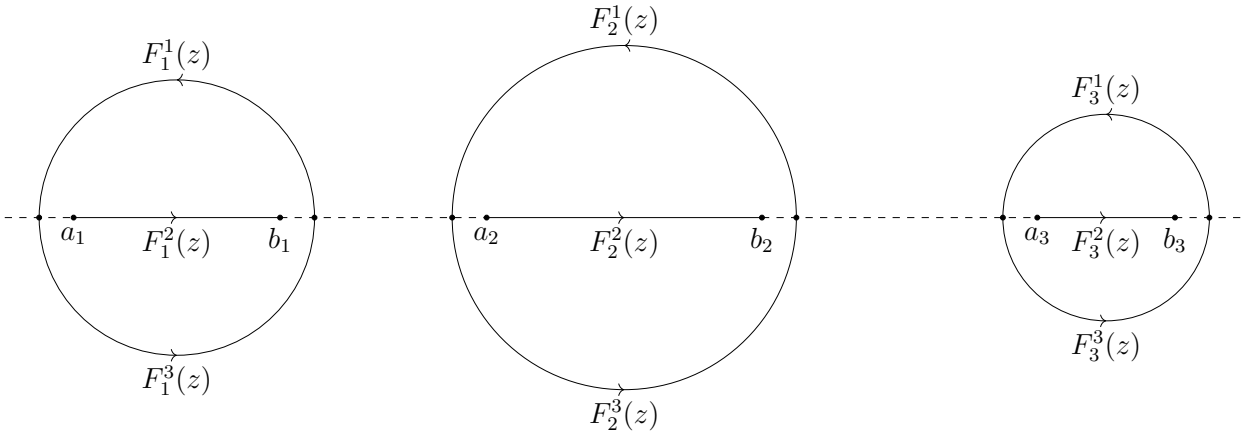


Figure 4.3: Jump conditions in a three interval case. $F_j^k(z)$ are as defined in (4.6).

As shorthand, we define

$$\begin{aligned} F_j^1(z) &= \begin{pmatrix} 1 & 0 \\ -\frac{1}{w_j(z)}\mathrm{e}^{2\mathfrak{h}_n(z)-2n\mathfrak{g}_\Sigma(z)} & 1 \end{pmatrix}, \quad F_j^2(z) = \begin{pmatrix} 0 & w_j(z)\mathrm{e}^{-A_j(n)} \\ -\frac{1}{w_j(z)}\mathrm{e}^{A_j(n)} & 0 \end{pmatrix}, \\ F_j^3(z) &= \begin{pmatrix} 1 & 0 \\ \frac{1}{w_j(z)}\mathrm{e}^{2\mathfrak{h}_n(z)-2n\mathfrak{g}_\Sigma(z)} & 1 \end{pmatrix}. \end{aligned} \tag{4.6}$$

We include example plots of the curves and jump conditions in this problem in Figure 4.3.

4.3 Recurrence coefficients

Recurrence coefficients for the orthogonal polynomials corresponding to the weight function (3.5) are directly related to the first-order asymptotic behavior (i.e., the residue at infinity) of the unique solution (3.3) to the original problem 3.2.2. If

$$\mathbf{Y}_n(z) = \left(\mathbf{I} + z^{-1}\mathbf{Y}_n^{(1)} + \mathrm{O}(z^{-2})\right) \begin{pmatrix} z^n & 0 \\ 0 & z^{-n} \end{pmatrix},$$

as $z \to \infty$, then the recurrence coefficients are given by

$$\begin{aligned} \check{a}_n &= \left(\mathbf{Y}_n^{(1)}\right)_{11} - \left(\mathbf{Y}_{n+1}^{(1)}\right)_{11}, \\ \check{b}_n &= \sqrt{\left(\mathbf{Y}_{n+1}^{(1)}\right)_{12}\left(\mathbf{Y}_{n+1}^{(1)}\right)_{21}}, \end{aligned}$$

for $n = 0, 1, 2, \ldots$ [53]. We now use this to derive equivalent formulae based on the first-order asymptotic behavior of 4.2.5 by working backwards through the transformations. Define $\mathfrak{h}_n^{(1)}$ by

$$\mathfrak{h}_n(z) = \frac{\mathfrak{h}_n^{(1)}}{z} + \mathrm{O}(z^{-2}), \quad z \to \infty. \tag{4.7}$$

Then,

$$\mathrm{e}^{\mathfrak{h}_n(z)} = 1 + \frac{\mathfrak{h}_n^{(1)}}{z} + \mathrm{O}(z^{-2}), \quad z \to \infty.$$

From this, if

$$\begin{aligned}\mathbf{S}_n(z) &= \mathbf{I} + z^{-1}\mathbf{S}_n^{(1)} + \mathrm{O}(z^{-2}),\\ \tilde{\mathbf{S}}_n(z) &= \mathbf{I} + z^{-1}\tilde{\mathbf{S}}_n^{(1)} + \mathrm{O}(z^{-2}),\end{aligned}$$

as $z \to \infty$, then

$$\begin{aligned}\mathbf{S}_n(z) = \tilde{\mathbf{S}}_n(z)\mathrm{e}^{-\mathfrak{h}_n(z)\boldsymbol{\sigma}_3} &= \left(\mathbf{I} + z^{-1}\tilde{\mathbf{S}}_n^{(1)} + \mathrm{O}(z^{-2})\right)\left(\mathbf{I} - z^{-1}\mathfrak{h}_n^{(1)}\boldsymbol{\sigma}_3 + \mathrm{O}(z^{-2})\right)\\ &= \left(\mathbf{I} + z^{-1}\left(\tilde{\mathbf{S}}_n^{(1)} - \mathfrak{h}_n^{(1)}\boldsymbol{\sigma}_3\right) + \mathrm{O}(z^{-2})\right),\end{aligned}$$

implying $\mathbf{S}_n^{(1)} = \tilde{\mathbf{S}}_n^{(1)} - \mathfrak{h}_n^{(1)}\boldsymbol{\sigma}_3$. Now, we obtain the first-order asymptotics of $\mathbf{Z}_n$ (4.2.1) from $\mathbf{S}_n$ (4.2.3). Recall that

$$\mathbf{Z}_n(z) = \mathfrak{c}^{-n\boldsymbol{\sigma}_3}\mathbf{S}_n(z)\mathrm{e}^{n\mathfrak{g}_\Sigma(z)\boldsymbol{\sigma}_3}.$$

Then, the asymptotic expansion of $\mathfrak{g}_\Sigma$ (4.3) gives

$$\mathbf{Z}_n(z) = \mathfrak{c}^{-n\boldsymbol{\sigma}_3}\left(\mathbf{I} + \mathbf{S}_n^{(1)}z^{-1} + \mathrm{O}(z^{-2})\right)\mathfrak{c}^{n\boldsymbol{\sigma}_3}z^{n\boldsymbol{\sigma}_3}\mathrm{e}^{n\boldsymbol{\sigma}_3\left((\mathfrak{g}_\Sigma)_1 z^{-1} + \mathrm{O}(z^{-2})\right)}.$$

Expanding

$$\mathrm{e}^{n\boldsymbol{\sigma}_3\left((\mathfrak{g}_\Sigma)_1 z^{-1} + \mathrm{O}(z^{-2})\right)} = \mathbf{I} + n(\mathfrak{g}_\Sigma)_1\boldsymbol{\sigma}_3 z^{-1} + \mathrm{O}(z^{-2}),$$

gives

$$\mathbf{Z}_n(z) = \left(\mathbf{I} + \left(\mathfrak{c}^{-n\boldsymbol{\sigma}_3}\mathbf{S}_n^{(1)}\mathfrak{c}^{n\boldsymbol{\sigma}_3} + n(\mathfrak{g}_\Sigma)_1\boldsymbol{\sigma}_3\right)z^{-1} + \mathrm{O}(z^{-2})\right)\begin{pmatrix} z^n & 0\\ 0 & z^{-n}\end{pmatrix},$$

and therefore

$$\mathbf{Z}_n^{(1)} = \mathfrak{c}^{-n\boldsymbol{\sigma}_3}\mathbf{S}_n^{(1)}\mathfrak{c}^{n\boldsymbol{\sigma}_3} + n(\mathfrak{g}_\Sigma)_1\boldsymbol{\sigma}_3 = \begin{pmatrix} \left(\mathbf{S}_n^{(1)}\right)_{11} + n(\mathfrak{g}_\Sigma)_1 & \mathfrak{c}^{2n}\left(\mathbf{S}_n^{(1)}\right)_{12}\\ \mathfrak{c}^{-2n}\left(\mathbf{S}_n^{(1)}\right)_{21} & \left(\mathbf{S}_n^{(1)}\right)_{22} - n(\mathfrak{g}_\Sigma)_1\end{pmatrix}.$$

Finally, the lensing of the problem in Step 1 does not affect the asymptotic behavior at infinity, so

$$\mathbf{Y}_n^{(1)} = \begin{pmatrix} \left(\tilde{\mathbf{S}}_n^{(1)}\right)_{11} - \mathfrak{h}_n^{(1)} + n(\mathfrak{g}_\Sigma)_1 & \mathfrak{c}^{2n}\left(\tilde{\mathbf{S}}_n^{(1)}\right)_{12} \\ \mathfrak{c}^{-2n}\left(\tilde{\mathbf{S}}_n^{(1)}\right)_{21} & \left(\tilde{\mathbf{S}}_n^{(1)}\right)_{22} + \mathfrak{h}_n^{(1)} - n(\mathfrak{g}_\Sigma)_1 \end{pmatrix},$$

yielding new recurrence formulae

$$\begin{aligned} \check{a}_n &= \left(\tilde{\mathbf{S}}_n^{(1)}\right)_{11} - \left(\tilde{\mathbf{S}}_{n+1}^{(1)}\right)_{11} - \mathfrak{h}_n^{(1)} + \mathfrak{h}_{n+1}^{(1)} - (\mathfrak{g}_\Sigma)_1, \\ \check{b}_n &= \sqrt{\left(\tilde{\mathbf{S}}_{n+1}^{(1)}\right)_{12}\left(\tilde{\mathbf{S}}_{n+1}^{(1)}\right)_{21}}, \end{aligned} \tag{4.8}$$

that we will use to recover the recurrence coefficients when numerically solving the problem 4.2.5. As $n \to \infty$, the jump conditions on C_j tend exponentially to the identity matrix, so the FIK Riemann–Hilbert problem can be considered with just the jump conditions on the intervals, asymptotically. While formulae for the recurrence coefficients in this asymptotic case are known [58, Appendix B], their evaluation is made difficult due to the presence of theta functions composed with an Abel map. Thus, a numerical method for solving this form of the FIK Riemann–Hilbert problem not only allows for efficient evaluation of these rather complicated formulae, but can potentially be utilized to compute theta functions. Related to this, one could choose the functions h_j so that $w_j(z) = w_k(z)$ for all j, k. In this case, individual circles around each interval $[a_j, b_j]$ could be replaced with a large circle around $[a_1, b_{g+1}]$. We will now demonstrate that in a specialized case, the jump condition on that circle can be removed entirely.

4.4 Special simplifying weight functions

If the weight function (3.5) satisfies $w_j = w_0|_{(a_j,b_j)}$ for some function w_0 analytic on $\mathbb{C} \setminus \overline{(\mathbb{R} \setminus \Sigma)}$—for instance the weight

$$w(x) \propto \mathbb{1}_\Sigma(x) \frac{\sqrt{b_{g+1} - x}\prod_{j=1}^{g+1}\sqrt{x - a_j}}{\prod_{j=1}^{g}\sqrt{x - b_j}}, \tag{4.9}$$

which arises naturally as the Akhiezer polynomial weight (Chapter 5)—the individual contours $C_1, \ldots, C_{g+1}$ in 4.2.5 can be replaced by a single counterclockwise-oriented circle C enclosing Σ. Moreover, the jump on C can be removed entirely, leaving the following simplified Riemann–Hilbert problem for $\mathbf{T}_n : \mathbb{C} \setminus \Sigma \to \mathbb{C}^{2\times 2}$.

Riemann–Hilbert Problem 4.4.1 (Simplified deformed FIK)**.** *Find a* 2×2 *matrix-valued function* $\mathbf{T}_n(z)$ *with the following properties:*

Analyticity: $\mathbf{T}_n(z)$ *is analytic for* $z \in \mathbb{C} \setminus \Sigma$*, admitting continuous boundary values on* Σ*.*

Jump conditions: *The boundary values* $\mathbf{T}_n^+(z)$ *(resp.* $\mathbf{T}_n^-(z)$*) from above (resp. below)* Σ *are related via*

$$\mathbf{T}_n^+(z) = \mathbf{T}_n^-(z) \begin{pmatrix} 0 & w_0(z)\mathrm{e}^{-A_j(n)} \\ -\frac{1}{w_0(z)}\mathrm{e}^{A_j(n)} & 0 \end{pmatrix}, \quad z \in (a_j, b_j), \quad j = 1, \ldots, g+1.$$

Normalization: $\mathbf{T}_n(z) = \mathbf{I} + \mathrm{O}(z^{-1})$ *as* $z \to \infty$*.*

Asymptotic condition (endpoints): *For each* $j = 1, \ldots, g+1$*, taken entry-wise,*

$$\mathbf{T}_n(z) = \mathrm{O}\begin{pmatrix} 1+|z-a_j|^{-\alpha_j/2} & 1+|z-a_j|^{\alpha_j/2} \\ 1+|z-a_j|^{-\alpha_j/2} & 1+|z-a_j|^{\alpha_j/2} \end{pmatrix}, \quad z \to a_j,$$
$$\mathbf{T}_n(z) = \mathrm{O}\begin{pmatrix} 1+|z-b_j|^{-\beta_j/2} & 1+|z-b_j|^{\beta_j/2} \\ 1+|z-b_j|^{-\beta_j/2} & 1+|z-b_j|^{\beta_j/2} \end{pmatrix}, \quad z \to b_j.$$

This problem is solved numerically by the same collocation method described in Section 4.5. Given its solution, the recurrence coefficients are recovered by

$$\check{a}_n = \left(\mathbf{T}_n^{(1)}\right)_{11} - \left(\mathbf{T}_{n+1}^{(1)}\right)_{11} - \mathfrak{h}_n^{(1)} + \mathfrak{h}_{n+1}^{(1)} - (\mathfrak{g}_\Sigma)_1,$$
$$\check{b}_n = \sqrt{\left(\mathbf{T}_{n+1}^{(1)}\right)_{12}\left(\mathbf{T}_{n+1}^{(1)}\right)_{21}},$$

and the Cauchy integrals by

$$\mathcal{C}_\Sigma\left[p_n w\right](z) = \left(\prod_{j=0}^{n-1} \frac{1}{\check{b}_j \mathfrak{c}}\right) \left(\mathbf{T}_n(z)\right)_{12} \mathrm{e}^{\mathfrak{h}_n(z) - n\mathfrak{g}_\Sigma(z)}.$$

To see why the jump on C can be discarded, consider a function $\mathbf{\Phi}$ solving the Riemann–Hilbert problem

$$\mathbf{\Phi}^+(z) = \mathbf{\Phi}^-(z)\mathbf{J}(z), \quad z \in C,$$
$$\mathbf{\Phi}(z) = \mathbf{I} + \mathbf{\Phi}_1 z^{-1} + \mathrm{O}(z^{-2}), \quad z \to \infty,$$

where $\mathbf{J}(z)$ is analytic exterior to C with

$$\mathbf{J}(z) = \mathbf{I} + \begin{pmatrix} 0 & 0 \\ j_1 & 0 \end{pmatrix} z^{-1} + \begin{pmatrix} 0 & 0 \\ \mathrm{O}(z^{-2}) & 0 \end{pmatrix}, \quad z \to \infty,$$

and which may satisfy additional jump conditions inside C (with interior D). Define

$$\mathbf{\Psi}(z) = \begin{cases} \mathbf{\Phi}(z), & z \in D, \\ \mathbf{\Phi}(z)\mathbf{J}(z), & z \in \mathbb{C} \setminus \overline{D}. \end{cases}$$

Then $\mathbf{\Psi}$ satisfies the same interior jumps as $\mathbf{\Phi}$, the jump on C becomes trivial, and

$$\mathbf{\Psi}(z) = \mathbf{I} + \left(\mathbf{\Phi}_1 + \begin{pmatrix} 0 & 0 \\ j_1 & 0 \end{pmatrix} \right) z^{-1} + \mathrm{O}(z^{-2}), \quad z \to \infty.$$

Applying this to $\tilde{\mathbf{S}}_n$ with $\mathbf{J}$ the consolidated jump on C, the jump is removable whenever $j_1 = 0$. For weights of the form (4.9), $w_0(z) = \mathrm{O}(z)$ as $z \to \infty$, so

$$\frac{1}{w_0(z)} \mathrm{e}^{2\mathfrak{h}_n(z) - 2n\mathfrak{g}_\Sigma(z)} = \mathrm{O}(z^{-1-2n}), \quad z \to \infty,$$

giving $j_1 = 0$ for all $n > 0$. For the Akhiezer weight (Chapter 5), $w_0(z) = \mathrm{O}(z^{-1})$, so the full problem 4.2.5 is needed only for $n = 0$; the simplified problem 4.4.1 applies for all $n \geq 1$.

4.5 Numerical implementation

4.5.1 Numerically computing $\mathfrak{g}_\Sigma$

To determine $\mathfrak{g}_\Sigma$, we first obtain the coefficients of $Q_g(z)$ via solving the linear system (4.1) by numerically computing the relevant integrals via a Chebyshev-$\underline{T}$-based quadrature rule[2]. Motivated by [159, Section 5.5], we use Lemma 3.1.1 to rewrite $\mathfrak{g}'_\Sigma$ as a Cauchy integral. We have that $R^+(z) = -R^-(z)$ for $z \in (a_j, b_j)$, and Q_g is of lower degree than R, so $\mathfrak{g}'_\Sigma$ solves the scalar Riemann–Hilbert problem

$$\begin{aligned} &(\mathfrak{g}'_\Sigma)^+(z) = (\mathfrak{g}'_\Sigma)^-(z) + 2\frac{Q_g(z)}{R^+(z)}, \quad z \in \bigcup_{j=1}^{g+1} (a_j, b_j), \\ &\mathfrak{g}'_\Sigma(z) = 0, \quad z \to \infty. \end{aligned}$$

Thus, by Lemma 3.1.1,

$$\mathfrak{g}'_\Sigma(z) = \sum_{j=1}^{g+1} 2\mathcal{C}_{[a_j, b_j]} \left[\frac{Q_g}{R^+} \right](z).$$

---

[2]We can easily do this by getting the first Chebyshev coefficient of $z^j \sqrt{z - a_k}\sqrt{b_k - z}/R(z)$ in a mapped Chebyshev-$\underline{T}$ series via the discrete cosine transform.

Of course, this implies that

$$\mathfrak{g}_\Sigma(z) = \int_{a_1}^{z} \mathfrak{g}'_\Sigma(z')\mathrm{d}z' = \sum_{j=1}^{g+1} 2\int_{a_1}^{z} \mathcal{C}_{[a_j,b_j]}\left[\frac{Q_g}{R^+}\right](z')\mathrm{d}z',$$

where the integration path is taken to be a straight line between a_1 and z. We approximate $\frac{Q_g(z)}{R^+(z)}\sqrt{z-a_j}\sqrt{z-b_j}$ to high-precision as a Chebyshev-$\underline{T}$ series on $[a_j, b_j]$ using the discrete cosine transform. Letting $w_{\underline{T},j}(x) = \tilde{w}_{\underline{T}}(x;[a_j,b_j])$, $\underline{T}_{j,k}(x) = \tilde{\underline{T}}_k(x;[a_j,b_j])$ and dropping the lower bound of integration that will be accounted for later, it thus suffices to consider computing the anti-derivatives

$$p_{j,k}(z) = -2\pi\mathrm{i}\alpha_{j,k}\int^{z}\frac{1}{2\pi\mathrm{i}}\int_{a_j}^{b_j}\frac{\underline{T}_{j,k}(s)w_{\underline{T},j}(s)}{s-z'}\mathrm{d}s\mathrm{d}z',$$

where $\alpha_{j,k}$ is the kth Chebyshev-$\tilde{\underline{T}}$ coefficient of $\frac{Q_g(z)}{R^+(z)}\sqrt{z-a_j}\sqrt{z-b_j}$ on $[a_j,b_j]$. For $k>0$, we swap the order of integration

$$p_{j,k}(z) = \alpha_{j,k}\int_{a_j}^{b_j}\underline{T}_{j,k}(s)w_{\underline{T},j}(s)\log(s-z)\mathrm{d}s.$$

Let $f(s) = \underline{T}_{j,k}(s)w_{\underline{T},j}(s)$. Using integration by parts,

$$\frac{1}{2\pi\mathrm{i}}\int_{a_j}^{b_j} f(s)\log(s-z)\mathrm{d}s = \frac{\log(b_j-z)}{2\pi\mathrm{i}}\int_{a_j}^{b_j} f(s)\mathrm{d}s - \mathcal{C}_{[a_j,b_j]}[F](z),$$

where

$$F(s) = \int_{a_j}^{s} f(s')\mathrm{d}s'.$$

By orthogonality,

$$\int_{a_j}^{b_j} f(s)\mathrm{d}s = \int_{a_j}^{b_j}\underline{T}_{j,k}(s)w_{\underline{T},j}(s)\mathrm{d}s = 0,$$

for $k>0$, and

$$p_{j,k}(z) = -2\pi\mathrm{i}\alpha_{j,k}\mathcal{C}_{[a_j,b_j]}\left[\int_{a_j}^{\diamond}\underline{T}_{j,k}(s)w_{\underline{T},j}(s)\mathrm{d}s\right](z).$$

Now, [59, Eq. 18.9.22] gives that

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(\sqrt{1-x}\sqrt{x+1}U_n(x)\right) = -(n+1)\frac{1}{\sqrt{1-x}\sqrt{x+1}}T_{n+1}(x),$$

where $\underline{T}_j(x) = \sqrt{2}T_j(x)$ in our convention. Substituting this in and integrating both sides gives

$$\int_{-1}^{x} \frac{\underline{T}_k(s)}{\sqrt{1-s}\sqrt{s+1}}\mathrm{d}s = -\frac{\sqrt{2}}{k}U_{k-1}(x)\sqrt{1-x}\sqrt{x+1},$$

and

$$\int_{-1}^{x} \underline{T}_k(s)w_T(s)\mathrm{d}s = -\frac{1}{\sqrt{2}k}U_{k-1}(x)w_U(x).$$

Using the change of variables $t = M_j^{-1}(s)$ where $M_j^{-1}(s) = M^{-1}(s;[a_j,b_j])$,

$$\int_{a_j}^{x} \underline{T}_{j,k}(s)w_{\underline{T},j}(s)\mathrm{d}s = \int_{-1}^{M_j^{-1}(x)} \underline{T}_{j,k}(M_j(t))w_{\underline{T},j}(M_j(t))\frac{b_j-a_j}{2}\mathrm{d}t.$$

Recall that by definition

$$\underline{T}_{j,k}(M_j(t)) = \underline{T}_k(t),$$

and

$$w_{\underline{T},j}(M_j(t)) = \frac{2}{b_j-a_j}w_T(t),$$

so that

$$\int_{a_j}^{x} \underline{T}_{j,k}(s)w_{\underline{T},j}(s)\mathrm{d}s = \int_{-1}^{M_j^{-1}(x)} \underline{T}_k(t)w_T(t)\mathrm{d}t = -\frac{1}{\sqrt{2}k}\frac{b_j-a_j}{2}U_{j,k-1}(x)w_{U,j}(x),$$

where we use the same notation $w_{U,j}(x) = w_U(x;[a_j,b_j])$, $U_{j,k}(x) = \tilde{U}_k(x;[a_j,b_j])$. Thus,

$$p_{j,k}(z) = \alpha_{j,k}\frac{\pi\mathrm{i}}{k}\frac{b_j-a_j}{\sqrt{2}}\mathcal{C}_{[a_j,b_j]}\left[U_{j,k-1}w_{U,j}\right](z),$$

for $k > 0$. For $k = 0$,

$$\begin{aligned} p_{j,0}(z) &= -2\pi\mathrm{i}\alpha_{j,0}\int^{z}\frac{1}{2\pi\mathrm{i}}\int_{a_j}^{b_j}\frac{w_{\underline{T},j}(s)}{s-z'}\mathrm{d}s\mathrm{d}z' = -2\pi\mathrm{i}\alpha_{j,0}\int^{z}\mathcal{C}_{[a_j,b_j]}\left[\underline{T}_{j,0}w_{\underline{T},j}\right]\mathrm{d}z' \\ &= \alpha_{j,0}\int^{z}\frac{1}{\sqrt{z'-a_j}\sqrt{z'-b_j}}\mathrm{d}z', \end{aligned}$$

which we have computed using the exact formulae for the Cauchy integrals of Chebyshev-$\underline{T}$ polynomials with their respective weight functions. With a change of variables $t = M(z')$ as before, we find [119]

$$p_{j,0}(z) = -\alpha_{j,0}\log J_+^{-1}\left(M_j^{-1}(z)\right). \tag{4.10}$$

Set

$$\Phi(z) = \sum_{j=1}^{g+1} \sum_{k=0}^{m_j} p_{j,k}(z), \tag{4.11}$$

where m_j is the number of terms we use in each Chebyshev series. Then, $\mathfrak{g}_\Sigma$ is approximated by

$$\mathfrak{g}_\Sigma(z) \approx \Phi(z) - \Phi(a_1).$$

A numerical instability can arise when computing $\Phi(a_1)$ with this approach due to the non-Lipschitz nature of $J_+^{-1}(z)$ at $z = -1$. This can be circumvented by replacing $J_+^{-1}(-1) = -1$ at each instance in the evaluation of $\Phi(a_1)$ that it occurs.

4.5.2 Asymptotic behavior

We compute the logarithmic[3] and first order terms $\mathfrak{c}$ and $(\mathfrak{g}_\Sigma)_1$ in the asymptotic expansion of $\mathfrak{g}_\Sigma$ (4.3). Recall that $\mathfrak{g}_\Sigma$ can be numerically computed as the Cauchy integrals of Chebyshev-U polynomials and the logarithm of the inverse Joukowsky map, so it suffices to compute the asymptotics of these functions.

The only term in our expansion (4.11) that contains a logarithmic term is (4.10). We have

$$\log J_+^{-1}\left(M_j^{-1}(z)\right) = \log\left(\frac{b_j - a_j}{4z}\right) + \frac{a_j + b_j}{2z} + \mathrm{O}(z^{-2}),$$

as $z \to \infty$, so that

$$p_{j,0}(z) = \alpha_{j,0} \log\left(\frac{4}{b_j - a_j}\right) + \alpha_{j,0} \log z - \alpha_{j,0} \frac{a_j + b_j}{2z} + \mathrm{O}(z^{-2}).$$

The only other contribution to the $\mathrm{O}(z^{-1})$ term comes from

$$\begin{aligned} p_{j,1}(z) &= \alpha_{j,1} \pi \mathrm{i} \frac{b_j - a_j}{\sqrt{2}} \mathcal{C}_{[a_j, b_j]}\left[U_{j,0} w_{U,j}\right](z) \\ &= \alpha_{j,1} \pi \mathrm{i} \frac{b_j - a_j}{\sqrt{2}} \frac{\mathrm{i}}{2\pi} z^{-1} + \mathrm{O}(z^{-2}) = -\alpha_{j,1} \frac{b_j - a_j}{2\sqrt{2}} z^{-1} + \mathrm{O}(z^{-2}), \end{aligned}$$

by (2.16). This gives

$$\mathfrak{c} \approx \exp\left(\sum_{j=1}^{g+1} \alpha_{j,0} \log\left(\frac{4}{b_j - a_j}\right) - \Phi(a_1)\right).$$

---

[3]This does not appear in our recurrence formulae but is necessary in (4.12) below. Additionally, $|\mathfrak{c}|$ is commonly referred to as the capacity of Σ [125].

The first order term can be approximated as

$$(\mathfrak{g}_\Sigma)_1 \approx -\sum_{j=1}^{g+1}\left(\alpha_{j,0}\frac{a_j+b_j}{2}+\alpha_{j,1}\frac{b_j-a_j}{2\sqrt{2}}\right).$$

4.5.3 Numerically computing $\mathfrak{h}_n$

To numerically compute $\mathfrak{h}_n$ we first compute $\Delta_1,\ldots,\Delta_g$ according to (4.2) by numerically evaluating $\mathfrak{g}_\Sigma^+(z)+\mathfrak{g}_\Sigma^-(z)$ on the gaps (b_j,a_{j+1}). We then obtain $A_1(n),\ldots,A_{g+1}(n)$ in the same manner as we obtained the coefficients for $\mathfrak{g}_\Sigma$. We do this by numerically computing the integrals via a Chebyshev-$\underline{T}$-based quadrature rule and solving the system (4.5).

Once the coefficients are known, we can evaluate $\mathfrak{h}_n$ in the complex plane in accordance with (4.4) by developing a method to compute the relevant Cauchy integrals. We do so by utilizing the formulae of Theorem 2.5.1. Since we know $R(z)$ and our coefficients, we need only numerically evaluate

$$\mathcal{C}_{[a_j,b_j]}\left[\frac{1}{R_+}\right](z),\quad \mathcal{C}_{[b_\ell,a_{\ell+1}]}\left[\frac{1}{R}\right](z),$$

for $j=1,\ldots,g+1$, $\ell=1,\ldots,g$. To do this, let $a=b_j$, $b=a_{j+1}$ or $a=a_j$, $b=b_j$ for some j. If we consider a truncated Chebyshev series

$$\frac{\sqrt{s-a}\sqrt{b-s}}{R_+(s)}\approx\sum_{j=0}^{p}\beta_j\tilde{\underline{T}}_j(s;[a,b]),$$

then

$$\begin{aligned}\mathcal{C}_{[a,b]}\left[\frac{1}{R_+}\right]&\approx\mathcal{C}_{[a,b]}\left[\sum_{j=0}^{p}\beta_j(-\pi\mathrm{i})\tilde{\underline{T}}_j(\diamond;[a,b])\tilde{w}_T(\diamond;[a,b])\right]\\&=-\pi\mathrm{i}\sum_{j=0}^{p}\beta_j\mathcal{C}_{[a,b]}\left[\tilde{\underline{T}}_j(\diamond;[a,b])\tilde{w}_T(\diamond;[a,b])\right].\end{aligned}$$

Of course, these Cauchy integrals can be computed via our exact formulae in Theorem 2.5.1, so we can approximate $\mathfrak{h}_n$ by computing a truncated series and taking linear combinations of Cauchy integrals as appropriate.

4.5.4 First-order asymptotic behavior

Now, we compute the first order term $\mathfrak{h}_n^{(1)}$ in (4.7). This is the $\mathrm{O}(z^{-g-2})$ term of the Cauchy integrals in (4.4) since $R(z)\sim z^{g+1}$ as $z\to\infty$. To compute this, we use the asymptotics of

Cauchy integrals to find

$$\mathfrak{h}_n^{(1)} = -\sum_{j=1}^{g+1} A_j(n) \int_{a_j}^{b_j} \frac{s^{g+1}}{R_+(s)} \mathrm{d}s - \sum_{\ell=1}^{g} \log(\mathrm{e}^{n\Delta_\ell}) \int_{b_\ell}^{a_{\ell+1}} \frac{s^{g+1}}{R(s)} \mathrm{d}s.$$

To compute this term numerically, we use a Chebyshev-$\underline{T}$-based quadrature rule on each interval to evaluate the integrals.

4.6 Solving the deformed FIK problem

The pieces are now in place to numerically solve the Riemann–Hilbert problem 4.2.5. As a simplifying assumption, we take each C_j in Figure 4.2 to be a circle centered at $(a_j + b_j)/2$ with a diameter greater than $b_j - a_j$[4] but small enough so that C_j and C_k do not intersect when[5] $j \neq k$. Then, 4.2.5 can be solved via a block linear system according to our methods for circles, intervals, multiple domains, and matrix problems by accounting for shifting and scaling things to the unit interval and circle as appropriate. The only remaining difficulty lies in choosing orthogonal polynomial bases so that the asymptotic conditions at the endpoints are satisfied.

On each interval $[a_j, b_j]$, we choose orthogonal polynomial bases based on the endpoint asymptotic conditions from 4.2.5:

$$\tilde{\mathbf{S}}_n(z) = \mathrm{O}\begin{pmatrix} 1 + |z - a_j|^{-\alpha_j/2} & 1 + |z - a_j|^{\alpha_j/2} \\ 1 + |z - a_j|^{-\alpha_j/2} & 1 + |z - a_j|^{\alpha_j/2} \end{pmatrix}, \quad z \to a_j,$$
$$\tilde{\mathbf{S}}_n(z) = \mathrm{O}\begin{pmatrix} 1 + |z - b_j|^{-\beta_j/2} & 1 + |z - b_j|^{\beta_j/2} \\ 1 + |z - b_j|^{-\beta_j/2} & 1 + |z - b_j|^{\beta_j/2} \end{pmatrix}, \quad z \to b_j.$$

Because the weight functions have asymptotic behavior

$$\begin{aligned}
w_{\underline{T},j}(z) &= \mathrm{O}(|z - a_j|^{-1/2}), & z \to a_j, \quad w_{\underline{T},j}(z) &= \mathrm{O}(|z - b_j|^{-1/2}), & z \to b_j, \\
w_{U,j}(z) &= \mathrm{O}(|z - a_j|^{1/2}), & z \to a_j, \quad w_{U,j}(z) &= \mathrm{O}(|z - b_j|^{1/2}), & z \to b_j, \\
w_{V,j}(z) &= \mathrm{O}(|z - a_j|^{1/2}), & z \to a_j, \quad w_{V,j}(z) &= \mathrm{O}(|z - b_j|^{-1/2}), & z \to b_j, \\
w_{W,j}(z) &= \mathrm{O}(|z - a_j|^{-1/2}), & z \to a_j, \quad w_{W,j}(z) &= \mathrm{O}(|z - b_j|^{1/2}), & z \to b_j,
\end{aligned}$$

we choose bases as follows:

- If $\alpha_j, \beta_j = -1$, then we use $\mathcal{C}_{[a_j,b_j]}[U_{j,k} w_{U,j}]$ in the first column of the matrix ansatz $\mathbf{U}$ (3.13) and $\mathcal{C}_{[a_j,b_j]}[\underline{T}_{j,k} w_{\underline{T},j}]$ in the second column of the ansatz.

---

[4]Heuristically, we find that the diameter should be at least $5(b_j - a_j)/4$ to avoid stability issues.

[5]To be fully robust, one should take C_j to be a Bernstein ellipse to which $1/h_j(z)$ is analytically continuable, but this requires a method to numerically solve Riemann–Hilbert problems on such an ellipse.

- If $\alpha_j, \beta_j = 1$, then we use $\mathcal{C}_{[a_j,b_j]}[\underline{T}_{j,k} w_{\underline{T},j}]$ in the first column and $\mathcal{C}_{[a_j,b_j]}[U_{j,k} w_{U,j}]$ in the second column.

- If $\alpha_j = 1, \beta_j = -1$, then we use $\mathcal{C}_{[a_j,b_j]}[W_{j,k} w_{W,j}]$ in the first column and $\mathcal{C}_{[a_j,b_j]}[V_{j,k} w_{V,j}]$ in the second column.

- If $\alpha_j = -1, \beta_j = 1$, then we use $\mathcal{C}_{[a_j,b_j]}[V_{j,k} w_{V,j}]$ in the first column and $\mathcal{C}_{[a_j,b_j]}[W_{j,k} w_{W,j}]$ in the second column.

4.6.1 Recovering coefficients

Once a numerical solution to 4.2.5 is obtained, we compute $\tilde{\mathbf{S}}_n^{(1)}$ by summing the first order terms of each approximate function in the block system. More explicitly, if the (ℓ, m) entry of $\tilde{\mathbf{S}}_n$ has a Laurent series representation

$$\sum_{k=-N_j}^{N_j} c_{j,k} \left(\frac{z - q_j}{r_j} \right)^k,$$

on each C_j where $q_j = (a_j + b_j)/2$ and r_j denotes the radius of C_j and an orthogonal polynomial series representation

$$\sum_{k=0}^{2L_j} d_{j,k} \mathcal{C}_{[a_j,b_j]} \left[p_k w \right],$$

on each $[a_j, b_j]$, then

$$\left(\tilde{\mathbf{S}}_n^{(1)} \right)_{\ell m} = \sum_{j=1}^{g+1} \left(r_j c_{j,-1} + \frac{\mathrm{i}}{2\pi} d_{j,0} \right),$$

where the scaling follows from (2.16) if p_k and w are a shifted and scaled Chebyshev polynomial and its weight, respectively. This combined with our computed first order contributions of $\mathfrak{g}_\Sigma$ and $\mathfrak{h}_n$ from earlier allows us to recover our recurrence coefficients from (4.8) as

$$\begin{aligned} \check{a}_n &= \left(\tilde{\mathbf{S}}_n^{(1)} \right)_{11} - \left(\tilde{\mathbf{S}}_{n+1}^{(1)} \right)_{11} - \mathfrak{h}_n^{(1)} + \mathfrak{h}_{n+1}^{(1)} - (\mathfrak{g}_\Sigma)_1, \\ \check{b}_n &= \sqrt{\left(\tilde{\mathbf{S}}_{n+1}^{(1)} \right)_{12} \left(\tilde{\mathbf{S}}_{n+1}^{(1)} \right)_{21}}. \end{aligned}$$

4.6.2 Cauchy integrals

To recover Cauchy integrals from our numerical method, we recall that the (1,2) entry of the true solution (3.3) to the original Riemann–Hilbert problem 3.2.2 is the Cauchy integral of

the monic orthogonal polynomial, meaning that

$$\mathcal{C}_\Sigma\left[p_n w\right](z) = \gamma_n \left(\mathbf{Y}_n(z)\right)_{12},$$

where γ_n is the coefficient on the x^n term in $p_n(x)$. From [53],

$$\gamma_n = \prod_{j=0}^{n-1} \check{b}_j^{-1},$$

with $\gamma_0 = 1$, so this term can be easily computed if we are already computing the recurrence coefficients. To obtain this from a computed solution $\tilde{\mathbf{S}}_n(z)$, we undo our series of transformations. If we assume that $z \notin \Sigma$ and furthermore that $z \notin \bar{D}_j$ where D_j is the interior of C_j for any j, then we need not worry about undoing the lensing step. In fact, the second column of $\mathbf{Y}_n$ is unaffected by the lensing step, so this need not be accounted for when computing $\mathcal{C}_\Sigma\left[p_n w\right](z)$, regardless of whether $z \in \bar{D}_j$. Since the remaining steps are diagonal, we can compute

$$\mathcal{C}_\Sigma\left[p_n w\right](z) = \left(\prod_{j=0}^{n-1} \frac{1}{\check{b}_j \mathfrak{c}}\right) \left(\tilde{\mathbf{S}}_n(z)\right)_{12} \mathrm{e}^{\mathfrak{h}_n(z) - n\mathfrak{g}_\Sigma(z)}. \tag{4.12}$$

For a given z, this is easily done in parallel with computing the recurrence coefficients by computing $\mathfrak{g}_\Sigma(z)$ once at the beginning and computing $\mathfrak{h}_n(z)$ each time we solve Riemann–Hilbert Problem 4.2.5.

4.7 Examples and applications

4.7.1 The single interval case

As a first example, we consider our methods applied to a single interval for which recurrence coefficients are known asymptotically. For simplicity, we consider this interval to be $[-1, 1]$. It turns out that the functions $\mathfrak{g}_\Sigma$ and $\mathfrak{h}_n$ can be computed exactly in this case, so solving the Riemann–Hilbert problem 4.2.5 numerically becomes simpler. First, we observe that there are no gaps between intervals on which jump conditions can be induced, so the function $\mathfrak{h}_n$ is identically zero for all n. Furthermore, determining $\mathfrak{g}'_\Sigma$ becomes trivial, and the exponential of $\mathfrak{g}_\Sigma$ can be computed explicitly as [99]

$$\mathrm{e}^{\mathfrak{g}_\Sigma(z)} = \varphi(z) = z + \sqrt{z+1}\sqrt{z-1},$$

and the weight function is given by

$$w(x) = h_1(x) \left(\sqrt{x+1}\right)^{\alpha_1} \left(\sqrt{1-x}\right)^{\beta_1},$$

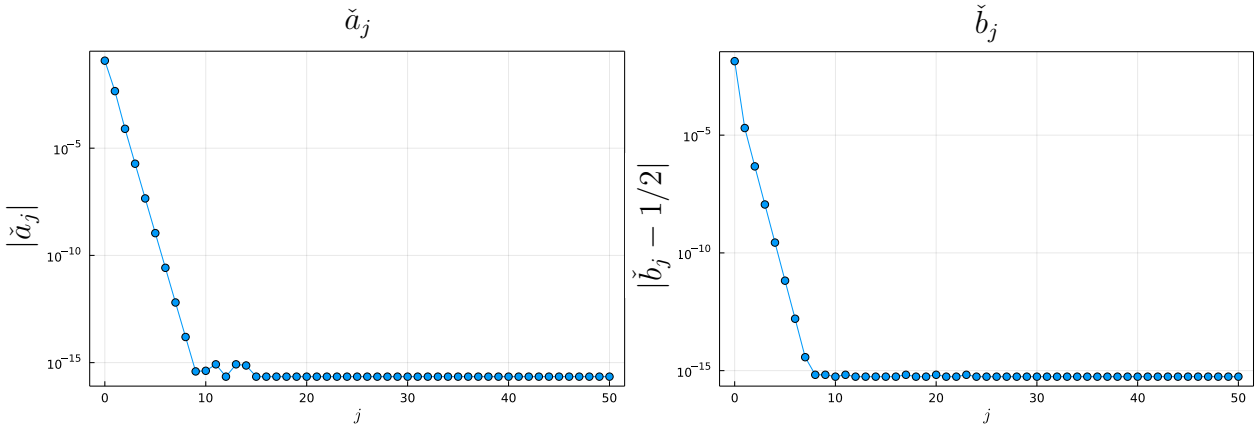


Figure 4.4: Plots of the difference between the first 51 orthogonal polynomial coefficients of a Chebyshev-U weight modified with (4.13) and an unmodified Chebyshev-U weight.

and h_1 satisfies the usual assumptions of being positive on $[-1, 1]$ and analytic in a neighborhood of it [100]. We additionally find that the first-order correction $(\mathfrak{g}_\Sigma)_1 = 0$ in this case, meaning that all correction terms in the recurrence coefficient formulae (4.8) drop out.

As with the multiple interval problem, we choose $C_1 = \{z : |z| = 5/4\}$, restricting our choice of h to functions that are positive on $[-1, 1]$ and whose reciprocals are analytic in an open set containing $\bar{D}_1$. For our example, we choose

$$h_1(x) = \frac{\mathrm{e}^x + 1}{4 + x^2}, \tag{4.13}$$

and $\alpha_1 = \beta_1 = 1$ to obtain a Chebyshev-U-like weight. We first compute the first 51 coefficients of these orthogonal polynomials using 160 collocation points on C_1 and 16 collocation points on $[-1, 1]$. We should expect these coefficients $\check{a}_j, \check{b}_j$ to tend to the Chebyshev-U coefficients $(\check{a}_j = 0, \check{b}_j = 1/2)$ exponentially as $j \to \infty$ [76]. In Figure 4.4, we plot these differences $|\check{a}_j|$ and $|\check{b}_j - 1/2|$ for the first 51 computed coefficients that exhibit the expected behavior.

In Figure 4.5, we plot the error at each of the first 51 coefficients for various numbers of collocation points on $[-1, 1]$, always choosing the number of collocation points on C_1 to be 10 times the number on $[-1, 1]$. The method is able to compute the recurrence coefficients $\check{a}_j, \check{b}_j$ starting at any choice of j without first computing $\check{a}_k, \check{b}_k$ for $k < j$. To demonstrate this, we include a similar plot showing the errors in computing $\check{a}_j$ and $\check{b}_j$ for $j = 1000, \ldots, 1020$ for various numbers of collocation points in Figure 4.6.

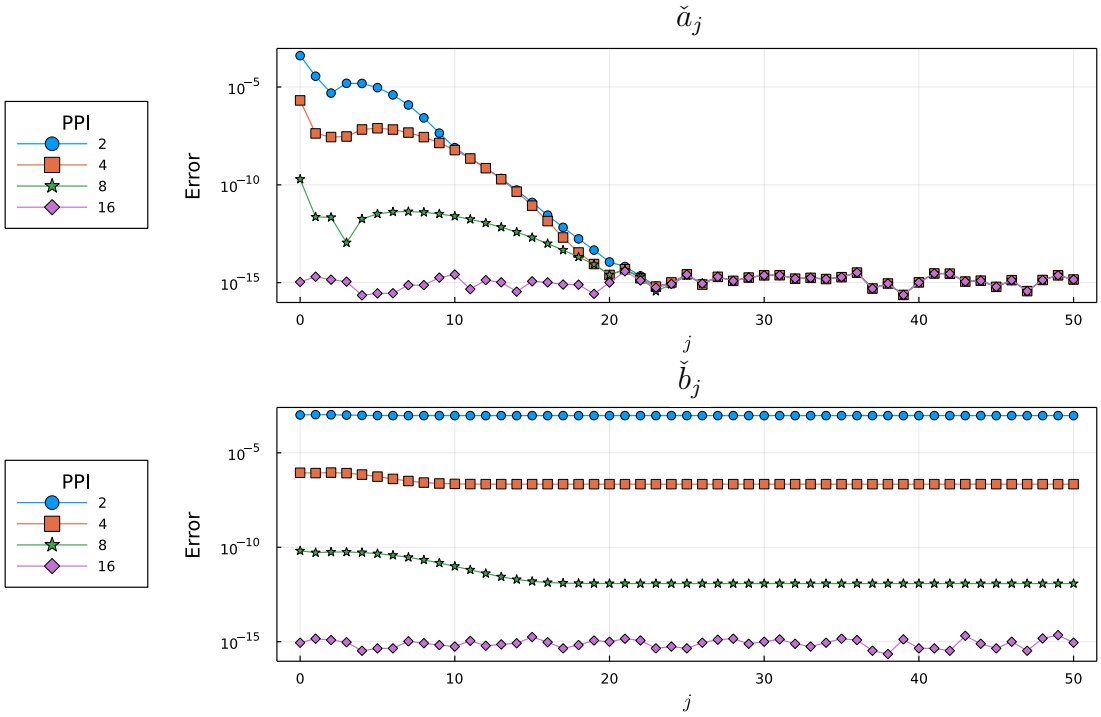


Figure 4.5: Absolute error in the computed first 51 orthogonal polynomial coefficients for a Chebyshev-U weight modified with (4.13) for 2, 4, 8, and 16 collocation points per interval (PPI) on $[-1, 1]$.

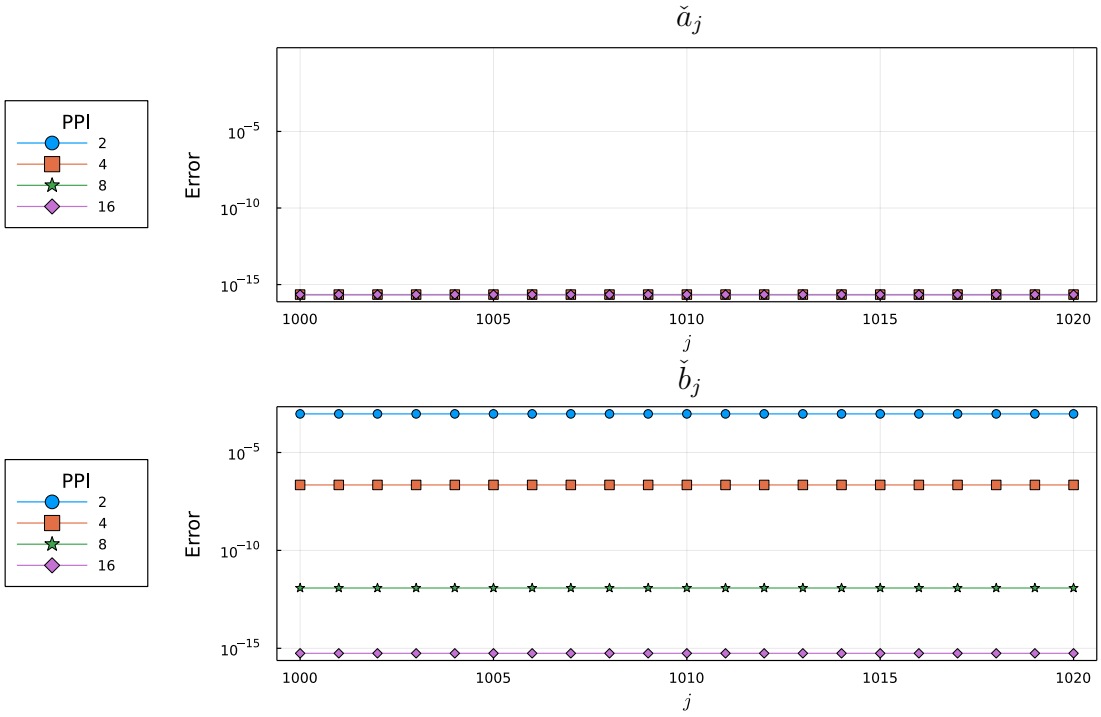


Figure 4.6: Error in the computed $\check{a}_j, \check{b}_j$ for $j = 1000, \ldots, 1020$ for a Chebyshev-U weight modified with (4.13) for 2, 4, 8, and 16 collocation points per interval (PPI) on $[-1, 1]$. In this case, $\check{a}_j = 0$, and the method approximates it well.

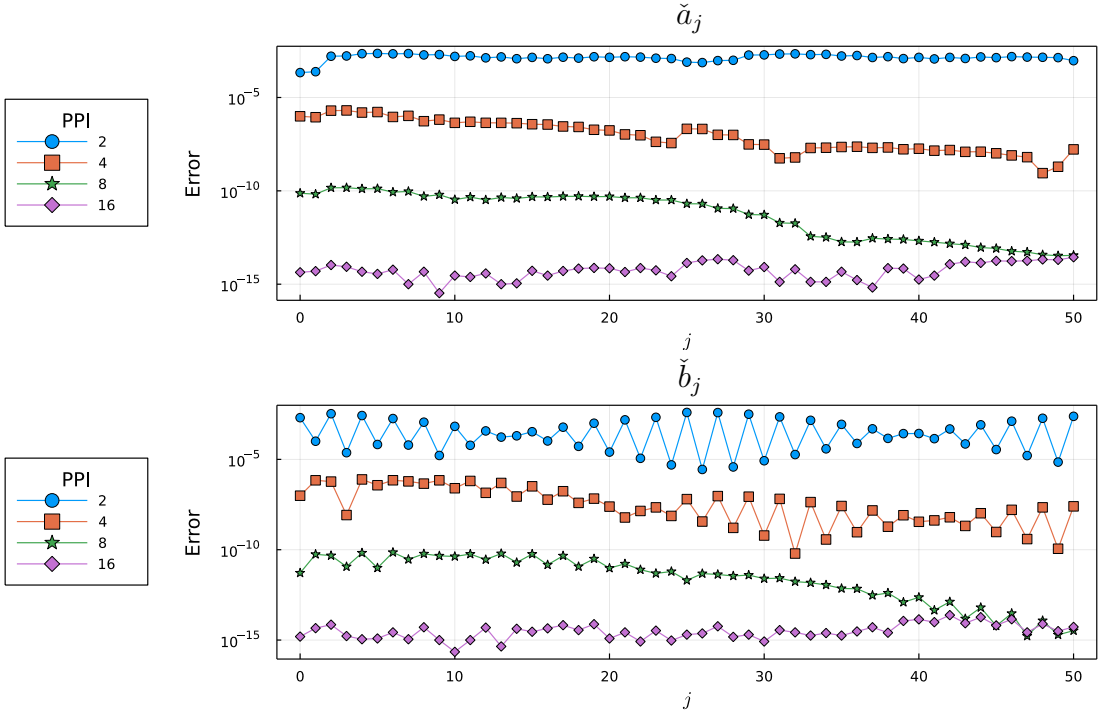


Figure 4.7: Error in the computed first 51 orthogonal polynomial coefficients for a Chebyshev-T-like weight on $[-1.8,-1]\cup[2,3]$ for 2, 4, 8, and 16 collocation points per interval (PPI). The convergence for large j in the 8 collocation point case suggests that ratio of collocation points on circles to points on intervals is not optimized. Computing these coefficients requires 0.13, 0.12, 0.30, and 0.97 seconds, respectively, via our method.

4.7.2 Multiple intervals

Next, the accuracy of our method is demonstrated in computing with orthogonal polynomials on multiple intervals. In all examples, we will take $h_j(x) = 1$ for all j, for simplicity, and compare our computed coefficients against those computed by the Householder method described in Section 4.7.1.

In Figure 4.7, we consider orthogonal polynomials defined on $[-1.8,-1]\cup[2,3]$ with a Chebyshev-$\underline{T}$-like weight on each ($\alpha_j = \beta_j = -1$ for $j = 1, 2$). Furthermore, in Figure 4.8, we include a timing comparison of our method to an optimized $\mathrm{O}(N^2)$ algorithm given as RKPW in [81] and `lanczos.m` in [74]. Since we are concerned with asymptotic timings for this comparison, our method is applied with only the jump conditions on the intervals as mentioned in Section 4.3. As is also mentioned in this section, there exist classes of weight functions for which the circles C_j are not needed for any polynomial degree, and we will explore this further in future work. We observe that our method begins to outperform the RKPW method if around 3200 pairs of recurrence coefficients need to be computed; however,

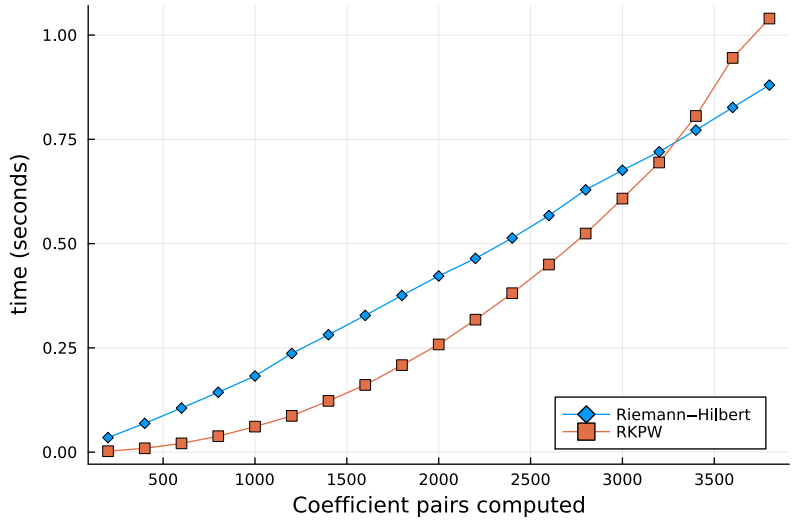


Figure 4.8: Timing comparison of the RKPW method [81] and our approach with only the jump conditions on the intervals considered. We observe that our method begins to outperform the RKPW method once around 3200 pairs of recurrence coefficients are computed.

since the RKPW method is heavily optimized while our method is coded suboptimally, we expect the threshold to be much lower with a better implementation. The RKPW method also has the aforementioned weakness in that it requires a priori knowledge of the desired recurrence length N, and altering this a posteriori requires potentially restarting the entire algorithm. Thus, in practice, one should consider using the RKPW method to compute the first few coefficients and our method asymptotically.

In Figure 4.9, we consider orthogonal polynomials defined on $[0.1, 1.1] \cup [2, 3] \cup [3.5, 4]$ with a Chebyshev-V-like weight on each ($\alpha_j = 1, \beta_j = -1$ for $j = 1, 2, 3$). Finally, in Figure 4.10, we consider orthogonal polynomials defined on $[-3.2, -2.2] \cup [0.1, 1.1] \cup [2, 3] \cup [3.5, 4]$ with a Chebyshev-$\underline{T}$-like weight on $[-3.2, -2.2]$, a Chebyshev-U-like weight on $[0.1, 1.1]$, a Chebyshev-V-like weight on $[2, 3]$, and a Chebyshev-W-like weight on $[3.5, 4]$ ($\alpha_1 = \beta_1 = -1$, $\alpha_2 = \beta_2 = 1$, $\alpha_3 = 1, \beta_3 = -1$, $\alpha_4 = -1, \beta_4 = 1$). In each example, we use the same number (2, 4, 8, and 16) of collocation points on each interval and 10 times that number on each circle C_j and compute the first 51 recurrence coefficients. In practice, one should consider using a different number of collocation points on each interval and varying the ratio of collocation points on circles to points on intervals to optimize runtime.

Heuristically, our method as described encounters stability issues once g reaches 4 or 5; however, 6 digits of accuracy is maintained until about $g = 11$. This is because the linear systems (4.1) and (4.5) become ill-conditioned as g increases. In Chapter 7, we will circumvent these instabilities by using a different basis.

In all figures, we observe rapid convergence to the "ground truth" coefficients $\check{a}_j, \check{b}_j$ for

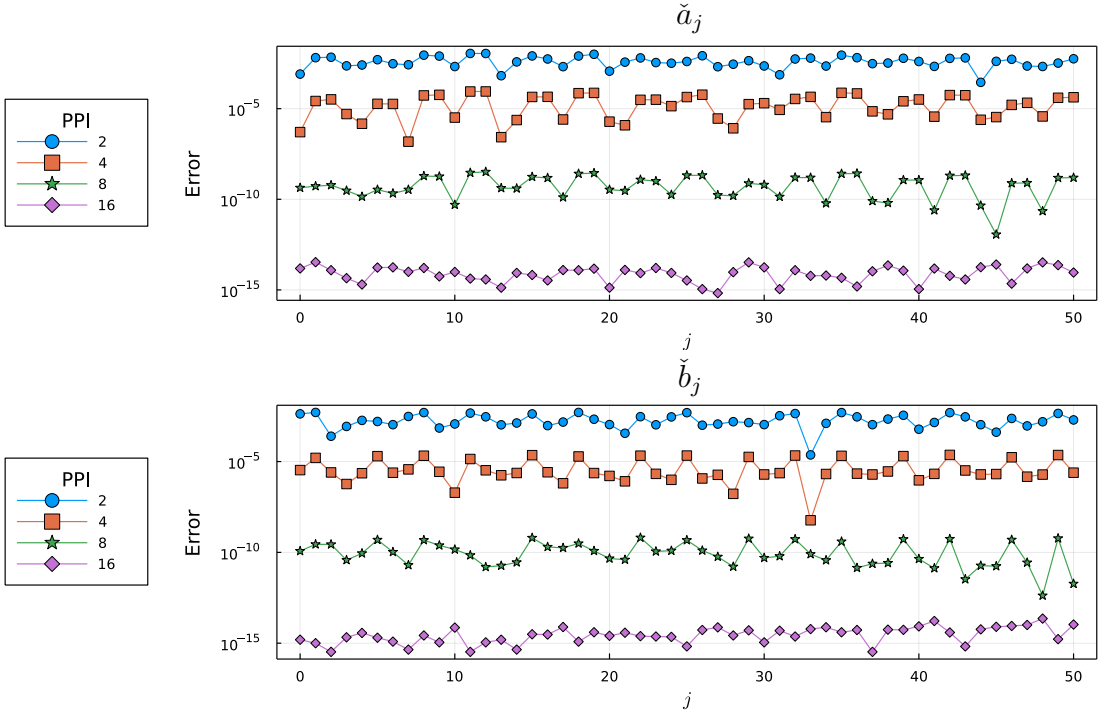


Figure 4.9: Error in the computed first 51 orthogonal polynomial coefficients for a Chebyshev-V-like weight on $[0.1, 1.1] \cup [2, 3] \cup [3.5, 4]$ for 2, 4, 8, and 16 collocation points per interval (PPI). Computing these coefficients requires 0.13, 0.25, 0.57, and 2.78 seconds, respectively, via our method.

any value of j. Errors on the order of machine precision are always achieved when we use 16 collocation points on each interval and 160 points on each circle. Oftentimes, fewer collocation points are needed to achieve this accuracy.

4.7.3 The Toda lattice

The semi-infinite Toda lattice as considered by Deift, Li, and Tomei [54] (see also [112]) with boundary condition $\check{b}_{-1}(t) = 0$ is defined by semi-infinite operators

$$\mathbf{X}(t) = \begin{pmatrix} \check{a}_0(t) & \check{b}_0(t) & & & \\ \check{b}_0(t) & \check{a}_1(t) & \check{b}_1(t) & & \\ & \check{b}_1(t) & \check{a}_2(t) & \check{b}_2(t) & \\ & & \ddots & \ddots & \ddots \end{pmatrix}, \quad \mathbf{B}(t) = \begin{pmatrix} 0 & \check{b}_0(t) & & & \\ -\check{b}_0(t) & 0 & \check{b}_1(t) & & \\ & -\check{b}_1(t) & 0 & \check{b}_2(t) & \\ & & \ddots & \ddots & \ddots \end{pmatrix},$$

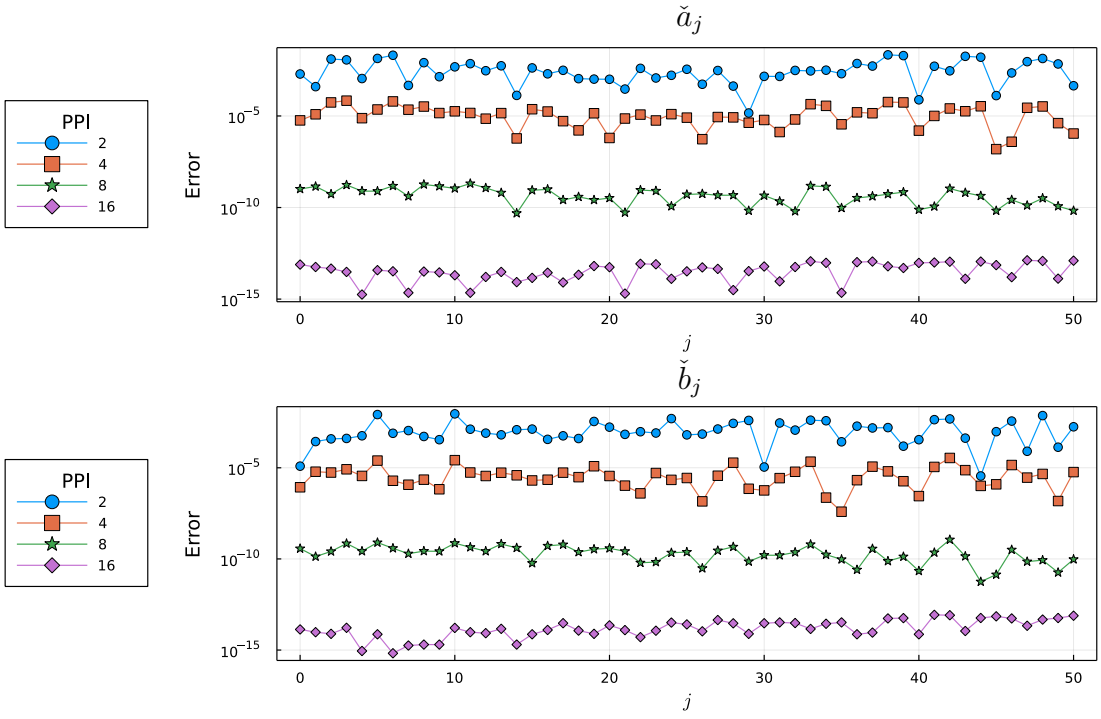


Figure 4.10: Error in the computed first 51 orthogonal polynomial coefficients for a Chebyshev-$\underline{T}$-like weight on $[-3.2, -2.2]$, a Chebyshev-U-like weight on $[0.1, 1.1]$, a Chebyshev-V-like weight on $[2, 3]$, and a Chebyshev-W-like weight on $[3.5, 4]$ for 2, 4, 8, and 16 collocation points per interval (PPI). This weight function is plotted in Figure 4.1. Computing these coefficients requires 0.15, 0.19, 0.99, and 5.96 seconds, respectively, via our method.

using the relation

$$\dot{\mathbf{X}}(t) = [\mathbf{B}(t), \mathbf{X}(t)],$$

where $[\cdot, \cdot]$ denotes the standard matrix commutator. Given an initial condition $\mathbf{X}(0) = \mathbf{J}(w)$ corresponding to the Jacobi operator to some orthogonal polynomial weight w, the time-dependent $\breve{a}_j(t), \breve{b}_j(t)$ are given by the respective recurrence coefficients $\breve{a}_j, \breve{b}_j$ of the orthogonal polynomials with weight $\mathrm{e}^{tx}w(x)$ [54] (see also [141]), i.e., if $\mathbf{X}(0) = \mathbf{J}(w)$, then $\mathbf{X}(t) = \mathbf{J}(w\mathrm{e}^{t\diamond})$.

Since our method allows for weights to be scaled by positive analytic functions, we are able to numerically evolve the Toda lattice for initial conditions corresponding to the various orthogonal polynomial weights in our arsenal; however, the scaling of e^{tx} grows exponentially in the complex plane, so we are only able to evolve the Toda lattice to a finite time horizon before encountering precision issues.

As a first example, we consider an initial condition of

$$\mathbf{X}(0) = \begin{pmatrix} 0 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ & 1/2 & 0 & 1/2 & \\ & & \ddots & \ddots & \ddots \end{pmatrix},$$

corresponding to a Chebyshev-U weight on $[-1, 1]$. Choosing to use 120 collocation points on C_1 and 20 on $[-1, 1]$, we compute the first 11 recurrence coefficients with $h_1(x) = \mathrm{e}^{tx}$ for times ranging from $t = 0$ to $t = 11.5$ and plot the results in Figure 4.11. Numerical blowup is observed shortly after time $t = 11.5$.

Now, we compute the time-evolution of the Toda lattice with initial condition $\mathbf{X}(0)$ corresponding to the recurrence coefficients of orthogonal polynomials with a Chebyshev-$\underline{T}$-like weight on $[-3, -2] \cup [2, 3]$. Numerically, we observe that $\breve{a}_j(0) = 0$ for all j and as $j \to \infty$, $\breve{b}_j(0)$ alternates between 0.5 and 2.5 as displayed in Figure 4.12.

We evolve the Toda lattice numerically for times ranging from $t = 0$ to $t = 4.5$ in the same manner as before[6] and plot the results in Figure 4.13.

Additionally, we emphasize that our method allows us to compute any recurrence coefficient without first computing the others, meaning that the evolution of $\breve{a}_j(t), \breve{b}_j(t)$ can be computed for any j. We demonstrate this by plotting $\breve{a}_{116}(t), \breve{b}_{116}(t)$ in Figure 4.14.

Finally, we consider the time-evolution of the Toda lattice with initial condition $\mathbf{X}(0)$ corresponding to the recurrence coefficients of orthogonal polynomials with a Chebyshev-W-like weight on $[0.1, 1.1] \cup [2, 3] \cup [3.5, 4]$. Now, the initial condition as displayed in Figure 4.15 appears almost noisy. We evolve the Toda lattice numerically for times ranging from $t = 0$ to $t = 5.7$ in the same manner as before and plot the results in Figure 4.16.

Now, we will turn our attention to the primary applications of this numerical method:

[6] By taking $h_j(x) = \mathrm{e}^{tx}$ for each j and computing the first 11 recurrence coefficients at each time with 120 collocation points on each C_j and 20 on each interval, we approximate the evolution of the Toda lattice.

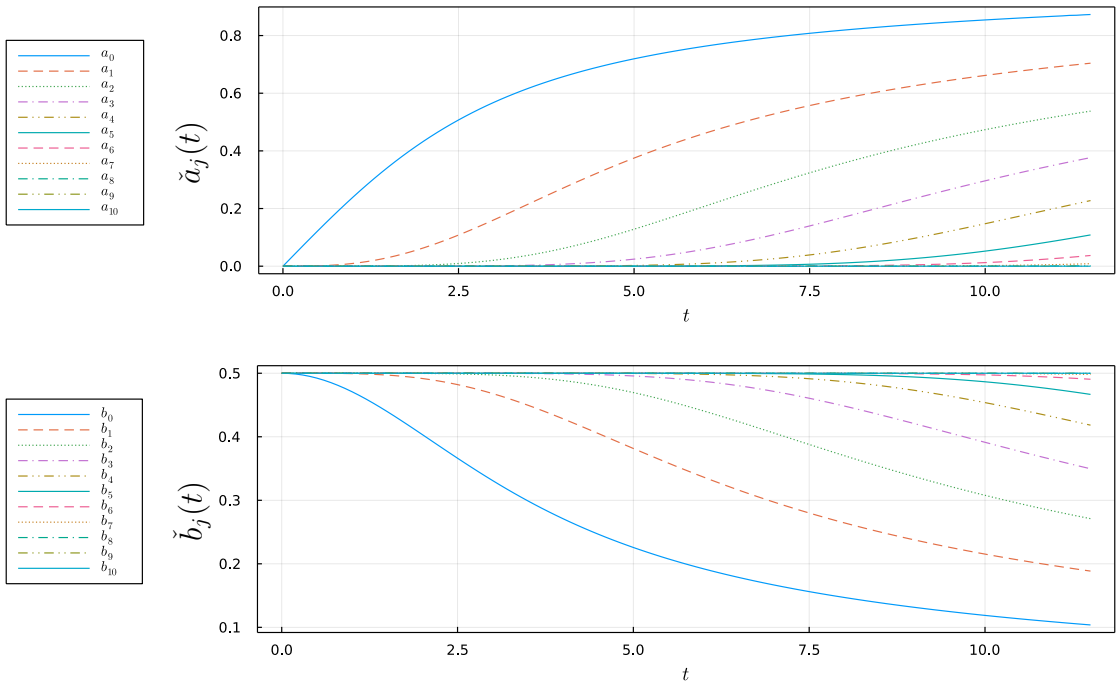


Figure 4.11: The evolution of the first 11 tridiagonal elements of the solution of the Toda lattice $\mathbf{X}(t)$ with initial condition corresponding to a Chebyshev-U weight on $[-1, 1]$ from time $t = 0$ to time $t = 11.5$.

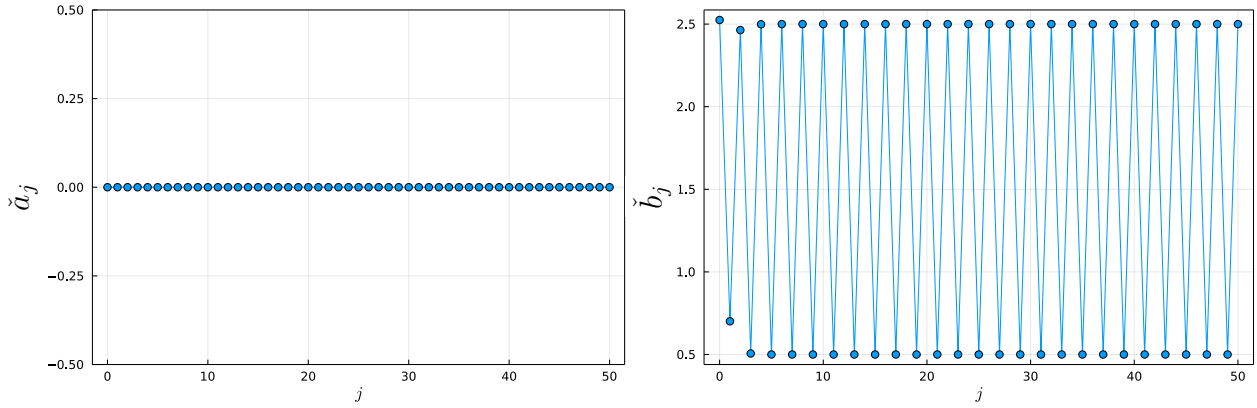


Figure 4.12: The first 51 recurrence coefficients for a Chebyshev-$\underline{T}$-like weight on $[-3, -2] \cup [2, 3]$ computed using 120 collocation points on each C_j and 20 on each interval.

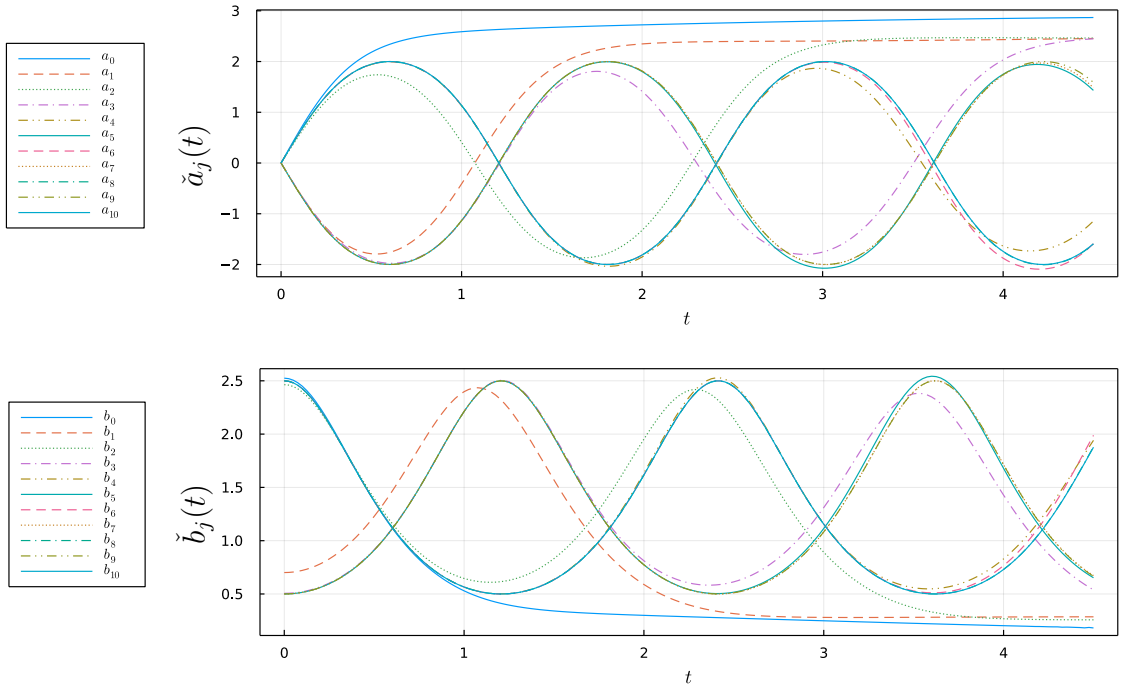


Figure 4.13: The evolution of the first 11 tridiagonal elements of the solution of the Toda lattice $\mathbf{X}(t)$ with initial condition corresponding to a Chebyshev-$\underline{T}$-like weight on $[-3,-2]\cup[2,3]$ from time $t=0$ to time $t=4.5$.

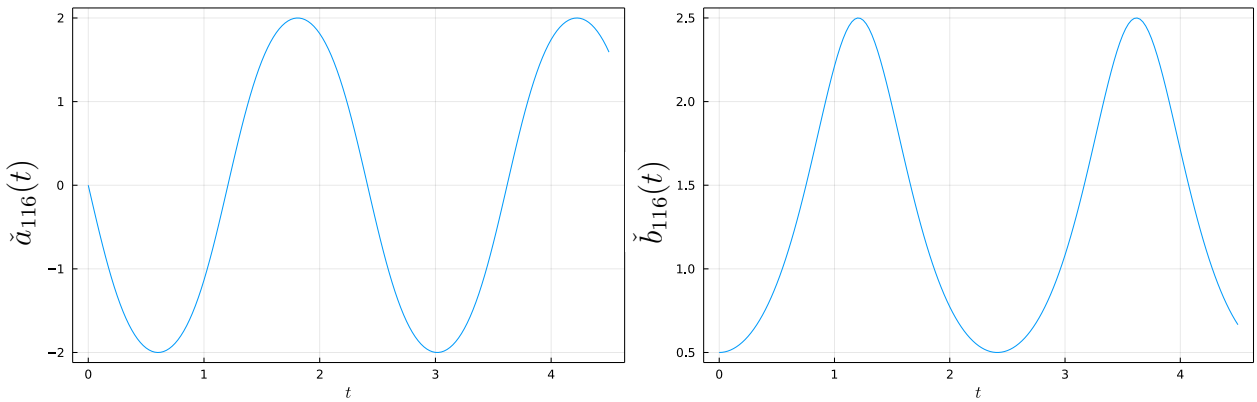


Figure 4.14: The evolution of $\check{a}_{116}(t), \check{b}_{116}(t)$ in the solution of the Toda lattice $\mathbf{X}(t)$ with initial condition corresponding to a Chebyshev-$\underline{T}$-like weight on $[-3,-2]\cup[2,3]$ from time $t=0$ to time $t=4.5$.

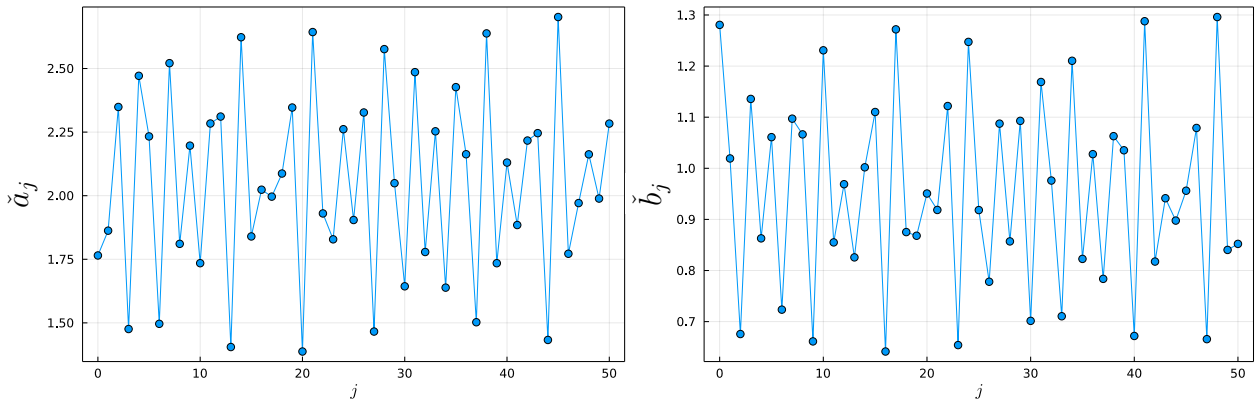


Figure 4.15: The first 51 recurrence coefficients for a Chebyshev-W-like weight on $[0.1, 1.1] \cup [2, 3] \cup [3.5, 4]$ computed using 120 collocation points on each C_j and 20 on each interval.

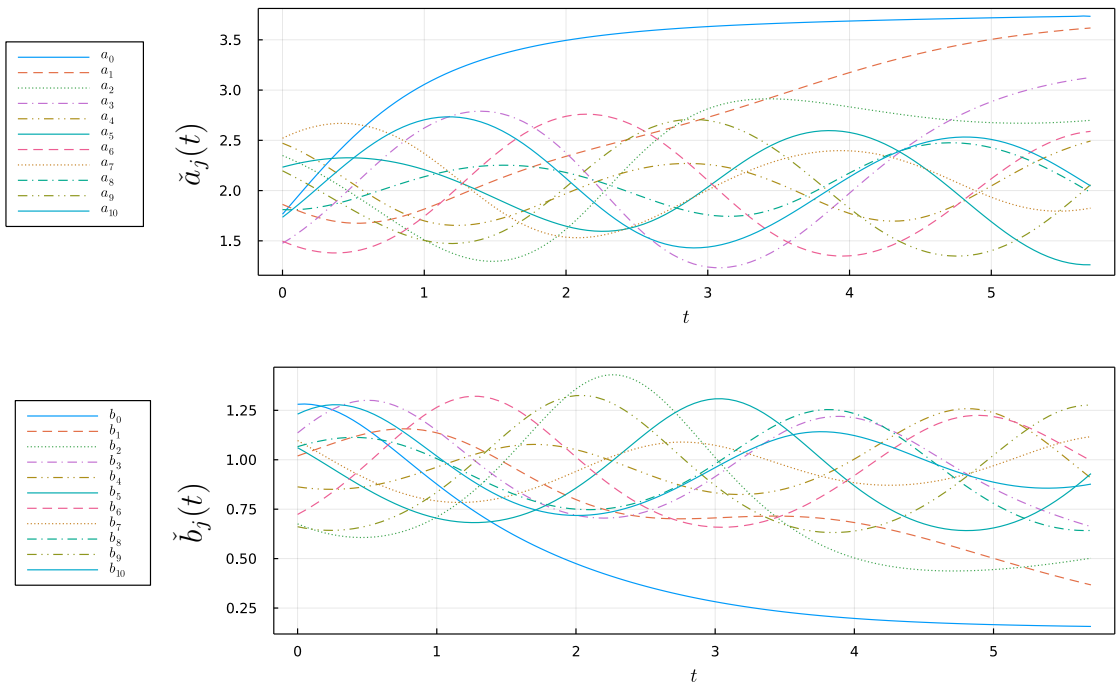


Figure 4.16: The evolution of the first 11 tridiagonal elements of the solution of the Toda lattice $\mathbf{X}(t)$ with initial condition corresponding to a Chebyshev-W-like weight on $[0.1, 1.1] \cup [2, 3] \cup [3.5, 4]$ from time $t = 0$ to time $t = 5.7$.

solving large-scale linear systems and computing functions of matrices.

Part II

NUMERICAL LINEAR ALGEBRA

Chapter 5

THE AKHIEZER ITERATION

5.1 Introduction

This chapter is based on [19]. Code used to generate the plots in this chapter can be found at [14].

The Chebyshev iteration [69,83,105] is an inner product-free iterative method often used to solve symmetric, definite linear systems $\mathbf{Ax} = \mathbf{b}$ where an estimate on the eigenvalues of $\mathbf{A}$ is known. This iteration is based on the minimax property of the Chebyshev polynomials. More specifically, let $\tilde{\underline{T}}_k(x) = \tilde{\underline{T}}_k(x; [a, b])$ where $[a, b]$ contains $\sigma(\mathbf{A})$ but not zero. Then, the polynomial $r_k(x) = \frac{\tilde{\underline{T}}_k(x)}{\tilde{\underline{T}}_k(0)}$ will be small on the spectrum of $\mathbf{A}$, and $q_k(x) = \frac{1-r_k(x)}{x}$ will satisfy $\mathbf{b} - \mathbf{A}q_k(\mathbf{A})\mathbf{b} \approx 0$ as well as a convenient recurrence relation.

A somewhat unconventional variant of this method arises from building a Chebyshev approximation to the function $f(x) = \frac{1}{x}$ on an interval containing the spectrum of $\mathbf{A}$. Specifically, f can be written in a Chebyshev-$\tilde{\underline{T}}$ series as

$$\frac{1}{x} = \sum_{j=0}^{\infty} \left\langle \tilde{\underline{T}}_j, \frac{1}{\diamond} \right\rangle_{L^2_{\tilde{w}_{\underline{T}}}} \tilde{\underline{T}}_j(x) = \sum_{j=0}^{\infty} \left(\int_a^b \frac{1}{z} \frac{\tilde{\underline{T}}_j(z)\mathrm{d}z}{\pi\sqrt{z-a}\sqrt{b-z}} \right) \tilde{\underline{T}}_j(x),$$

$$\int_a^b \frac{\tilde{\underline{T}}_j(z)\tilde{\underline{T}}_k(z)}{\pi\sqrt{z-a}\sqrt{b-z}}\mathrm{d}z = \delta_{jk},$$

where δ_{jk} is the Kronecker delta. This yields a series for the matrix inverse by replacing $\tilde{\underline{T}}_j(x)$ with the matrix polynomials $\tilde{\underline{T}}_j(\mathbf{A})$. Thus, the solution to the linear system can be approximated as

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \approx \sum_{j=0}^{k} \left(\int_a^b \frac{\tilde{\underline{T}}_j(z)}{z} \frac{\mathrm{d}z}{\pi\sqrt{z-a}\sqrt{b-z}} \right) \tilde{\underline{T}}_j(\mathbf{A})\mathbf{b} =: \mathbf{x}_{k+1}. \tag{5.1}$$

This formula can be implemented as an explicit iteration, see Algorithm 1. We advocate for this modification because, as we describe in this chapter, it allows one to directly generalize to general classes of orthogonal polynomials. Furthermore, the asymptotics of such polynomials give error bounds for the iteration.

The assumption that $\mathbf{A}$ is symmetric and definite can be relaxed provided that the eigenvalues of $\mathbf{A}$ lie sufficiently close to $[a, b]$ [105, 106]. In fact, given an ellipse containing the spectrum of $\mathbf{A}$, but not the origin, one can find an optimal interval $[a, b]$ for which the Chebyshev iteration converges (after possible rotation $\mathbf{A} \to \mathrm{e}^{\mathrm{i}\phi}\mathbf{A}$); however, the iteration fails if there is no such ellipse [105].

For instance, assume that the eigenvalues of $\mathbf{A}$ lie in $[a_1, b_1] \cup [a_2, b_2]$ where $a_1 < b_1 < 0 < a_2 < b_2$. In this case, to use the iteration (5.1), the shifted and scaled Chebyshev polynomials need to be replaced with polynomials with similar properties on multiple intervals to those of the Chebyshev polynomials on a single interval. In the classical view, one needs to replace r_k with polynomials that are small on $[a_1, b_1] \cup [a_2, b_2]$ and large at the origin. de

Algorithm 1: Modified Chebyshev iteration

Input: $\mathbf{A}$, $\mathbf{b}$, initial guess $\mathbf{x}_0$, parameters α, c such that $\alpha - c = a$ and $\alpha + c = b$.
Set $\mathbf{b} \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$.
Set $\mathbf{p}_0 = \mathbf{b}$.
Set $S_0 = \frac{1}{\sqrt{\alpha-c}\sqrt{\alpha+c}}$.
Set $\mathbf{x}_1 = S_0\mathbf{p}_0$.
for *k=1,2,...* **do**
  if *k=1* **then**
    Set $\mathbf{p}_1 = \frac{1}{c}(\mathbf{A}\mathbf{p}_0 - \alpha\mathbf{p}_0)$.
  else
    Set $\mathbf{p}_k = \frac{2}{c}\mathbf{A}\mathbf{p}_{k-1} - \frac{2\alpha}{c}\mathbf{p}_{k-1} - \mathbf{p}_{k-2}$.
  end
  Set $S_k = S_{k-1}\left(-\frac{\alpha}{c} + \sqrt{\frac{\alpha}{c} - 1}\sqrt{\frac{\alpha}{c} + 1}\right)$.
  Set $\mathbf{x}_{k+1} = \mathbf{x}_k + 2S_k\mathbf{p}_k$.
  if converged **then**
    Set $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_{k+1} + \mathbf{x}_0$.
    return $\mathbf{x}_{k+1}$
  end
end

Boor and Rice [49] explore using the minimax polynomials on $[a_1, b_1] \cup [a_2, b_2]$ for this, and they present a method to compute such polynomials. Unfortunately, such polynomials do not, in general, satisfy a three-term recurrence relation which makes generating the matrix polynomials difficult. Saad [132] circumvents this issue by instead using polynomials that are orthogonal with respect to a weight function w supported on $[a_1, b_1] \cup [a_2, b_2]$. Since such polynomials satisfy a three-term recurrence relation, the difficulty lies in computing the recurrence coefficients and the generalization of the integral terms in (5.1). Using a clever construction, for this two-interval case, Saad iteratively builds the first k of these quantities in $\mathrm{O}(k^2)$ arithmetic operations using Chebyshev polynomials defined separately on each interval. Specifically, he computes the degree k polynomial q_k that minimizes a weighted L^2 norm $\|1 - \diamond q_k(\diamond)\|_{L^2_w}$ for a chosen weight function w and uses their three-term recurrence to compute $q_k(\mathbf{A})\mathbf{b}$ iteratively. In this work we, in effect, improve upon that in [132] by taking $w(x) = x^2 v(x)$ where $v(x)$ is the weight function that Saad employs[1].

In Chapter 4, we developed a numerical Riemann–Hilbert-based method for computing recurrence coefficients and Cauchy integrals for orthogonal polynomials with weight functions supported on multiple, disjoint intervals (see [120, 158] for a method for measures supported on $\mathbb{R}$). This method computes the first k recurrence coefficients and Cauchy integrals in $\mathrm{O}(k)$ arithmetic operations. These quantities can be employed analogously to (5.1), yielding

[1] In [132], $v(x)$ is found by using h_j constant, $\alpha_j = \beta_j = -1$ in (3.5).

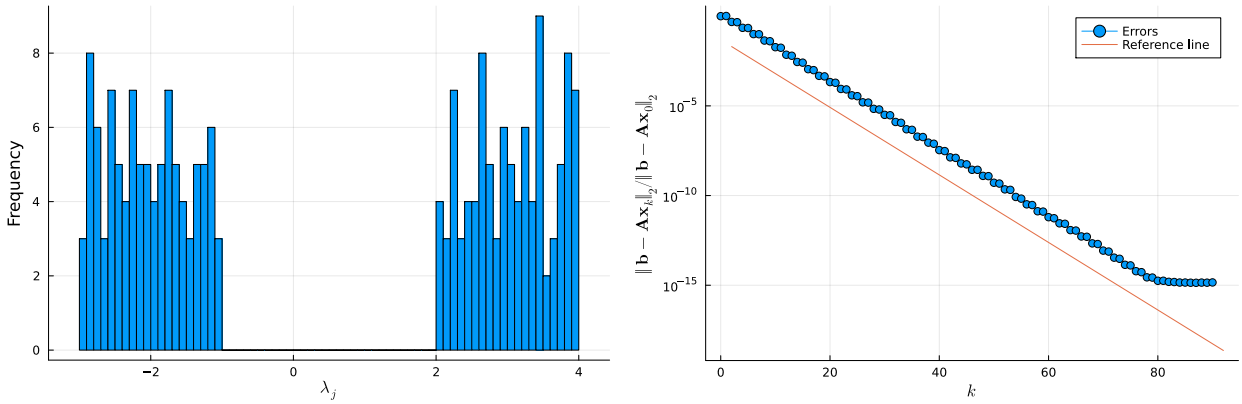


Figure 5.1: Histogram of the eigenvalues of $\mathbf{A} \in \mathbb{R}^{200\times 200}$ (left) and the convergence of the Akhiezer iteration applied to $\mathbf{A}\mathbf{x} = \mathbf{b}$ where $\mathbf{b} \in \mathbb{R}^{200}$ is a random Gaussian vector. The reference line that approximates the convergence rate is derived in Section 5.2.1.

a generalization of the Chebyshev iteration that improves the complexity of [132].

In this chapter, we demonstrate how the Akhiezer polynomials, along with the numerical method of Chapter 4, can be used to extend (5.1) to an iterative method for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ where $\mathbf{A}$ has eigenvalues on, or near, multiple, disjoint intervals. We will refer to this method as *the Akhiezer iteration*. Furthermore, we discuss how the Akhiezer iteration can be extended to compute matrix functions $f(\mathbf{A})\mathbf{b}$ for suitably analytic functions f, similar to the approach in [43], and implemented in optimal complexity with a provably exponential convergence rate. In Figure 5.1, we demonstrate the convergence of the Akhiezer iteration applied to a symmetric matrix with spectrum contained in $[-3, -1] \cup [2, 4]$.

The rest of this chapter is organized as follows. In Section 5.2, we present the Akhiezer iteration and its extension to matrix functions and discuss convergence rates. In Section 5.4, we discuss how the Akhiezer iteration can be implemented adaptively when $\mathbf{A}$ is symmetric. In Section 5.5, we present examples and applications, and in Section 5.6, we present our conclusions and discuss potential extensions. Akhiezer's orthogonal polynomial construction, including explicit formulae for recurrence coefficients and Cauchy integrals, is presented in Section 2.7 of Chapter 2.

5.2 The Akhiezer iteration and matrix function evaluation

5.2.1 The Akhiezer iteration for solving linear systems

Following [132], orthogonal polynomials with respect to a weight function w of the form (3.5) (in particular (2.21)) can be used to extend the classical Chebyshev iteration to solve indefinite linear systems. Assume that a square matrix $\mathbf{A}$ has eigenvalues on, or near, $\Sigma = \bigcup_{j=1}^{g+1}[a_j, b_j]$ and that $0 \notin \Sigma$. If $p_0, p_1, \ldots$ denote the orthonormal polynomials with

respect to w, then a p_j-series expansion for $1/(x-z)$, for $x \in \Sigma$, $z \notin \Sigma$, is given by

$$\begin{aligned}\frac{1}{x-z} &= \sum_{j=0}^{\infty} \left\langle p_j, \frac{1}{\diamond - z} \right\rangle_{L^2_w(\Sigma)} p_j(x) \\ &= \sum_{j=0}^{\infty} \left(\int_{\Sigma} \frac{p_j(s)}{s-z} w(s) \mathrm{d}s \right) p_j(x) \\ &= \sum_{j=0}^{\infty} \mathcal{S}_{\Sigma}\left[p_j w\right](z) p_j(x).\end{aligned} \tag{5.2}$$

This can be extended to an iterative method for solving the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ by truncating the series with $z = 0$:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} = \sum_{j=0}^{\infty} \mathcal{S}_{\Sigma}\left[p_j w\right](0) p_j(\mathbf{A})\mathbf{b} \approx \sum_{j=0}^{k} \mathcal{S}_{\Sigma}\left[p_j w\right](0) p_j(\mathbf{A})\mathbf{b} =: \mathbf{x}_{k+1}. \tag{5.3}$$

We call the iterative implementation of this approximation *the Akhiezer iteration*. The recurrence coefficients and Cauchy integrals can be computed either by the Riemann–Hilbert approach of Chapter 4 or by (2.24) and (2.25) in the case of the $g = 1$ Akhiezer weight. Then, the Akhiezer iteration for shifted linear solves can be implemented as in Algorithm 2.

Algorithm 2: Akhiezer iteration to approximate $(\mathbf{A} - z\mathbf{I})^{-1}\mathbf{b}$, $z \notin \Sigma$

Input: $\mathbf{A}$, $\mathbf{b}$, initial guess $\mathbf{x}_0$, and a function to compute recurrence coefficients $\check{a}_k, \check{b}_k$ and Stieltjes transforms $S_k = \mathcal{S}_{\Sigma}\left[p_k w\right](z)$.

Set $\mathbf{b} \leftarrow \mathbf{b} - \mathbf{A}\mathbf{x}_0$.
Set $\mathbf{p}_0 = \mathbf{b}$.
Set $\mathbf{x}_1 = S_0\mathbf{p}_0$.
for *k=1,2,...* **do**
  if *k=1* **then**
    Set $\mathbf{p}_1 = \frac{1}{\check{b}_0}(\mathbf{A}\mathbf{p}_0 - \check{a}_0\mathbf{p}_0)$.
  else
    Set $\mathbf{p}_k = \frac{1}{\check{b}_{k-1}}(\mathbf{A}\mathbf{p}_{k-1} - \check{a}_{k-1}\mathbf{p}_{k-1} - \check{b}_{k-2}\mathbf{p}_{k-2})$.
  end
  Set $\mathbf{x}_{k+1} = \mathbf{x}_k + S_k\mathbf{p}_k$.
  if converged **then**
    Set $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_{k+1} + \mathbf{x}_0$.
    return $\mathbf{x}_{k+1}$
  end
end

Algorithm 2 can instead be utilized to build an approximation to the matrix inverse $\mathbf{A}^{-1}$ by using $\mathbf{b} = \mathbf{I}$ and removing the initial guess $\mathbf{x}_0$. Then, at each step, the algorithm requires matrix-matrix multiplication rather than matrix-vector products but is otherwise identical.

Remark 5.2.1. *In* (5.2)*, one can differentiate with respect to z a number of times to find expansions for x^{-p} for $p \in \mathbb{N}$, provided one can compute derivatives of the Stieltjes transforms $\mathcal{S}_\Sigma$. The Riemann–Hilbert approach of Chapter 4 allows for this computation in a fairly straightforward manner, but we do not implement this here.*

Convergence

From [58], we have the following behavior of the orthogonal polynomials:

Lemma 5.2.2. *Let $\Sigma = \bigcup_{j=1}^{g+1}[a_j, b_j] \subset \mathbb{R}$ be a union of disjoint intervals and let w be a weight function of the form* (3.5)*. Denote the corresponding orthonormal polynomials by $(p_j(z))_{j=0}^{\infty}$, fix $\epsilon > 0$, and set $V = \{z \in \mathbb{C} \ : \ |z - a_j| \geq \epsilon, \ |z - b_j| \geq \epsilon, \ j = 1, \dots, g+1\}$. Then,*

$$p_n(z) = \hat{\delta}_n(z)\mathrm{e}^{n\mathfrak{g}_\Sigma(z)}, \qquad 2\pi\mathrm{i}\mathcal{C}_\Sigma\left[p_n w\right](z) = \delta_n(z)\mathrm{e}^{-n\mathfrak{g}_\Sigma(z)}, \quad z \in V,$$

where $\delta_n(z)$ and $\hat{\delta}_n(z)$ are uniformly bounded in both n and $z \in V$.

Proof. By [58, Equations B.2, B.3] and undoing the lensing step of the Riemann–Hilbert problem deformations, the above formulae hold for $\delta_n(z)$ and $\hat{\delta}_n(z)$ depending on bounded quantities and $\prod_{j=0}^{n-1} \frac{1}{b_j \mathfrak{c}}$ where $\log \mathfrak{c}$ is the $O(1)$ term in the expansion of $\mathfrak{g}_\Sigma$ at infinity. By [58, Equations B.3, B.12],

$$\mathfrak{c}^n \prod_{j=0}^{n-1} \check{b}_j = \mathrm{O}(1),$$

as $n \to \infty$, so $\delta_n(z)$ and $\hat{\delta}_n(z)$ are bounded from above. The rest of the claims follow immediately from [58]. □

To ease presentation, we introduce the following definitions.

Definition 5.2.3. *We say that a matrix is generic if it is diagonalizable and has no eigenvalues at the endpoints of Σ.*

This definition is here to simplify the analysis, and our theorems require the matrix in question to be generic. With a more technical approach, one can obtain similar bounds for non-generic matrices and the algorithms in question still converge at roughly the same rate. We remark on this below.

Definition 5.2.4. *Given Σ and $\mathbf{A} \in \mathbb{C}^{n\times n}$ define*

$$\nu(z; \mathbf{A}) = \max_j \mathrm{Re}\mathfrak{g}_\Sigma(\lambda_j) - \mathrm{Re}\mathfrak{g}_\Sigma(z), \quad \nu(\mathbf{A}) = \max_j \mathrm{Re}\mathfrak{g}_\Sigma(\lambda_j).$$

where λ_j are the eigenvalues of $\mathbf{A}$.

Theorem 5.2.5. *Consider a density w of the form (3.5). If $\mathbf{A} \in \mathbb{C}^{n\times n}$ is generic and its eigenvalues λ_j all satisfy $\mathrm{Re}\mathfrak{g}_\Sigma(\lambda_j) < \mathrm{Re}\mathfrak{g}_\Sigma(z)$, then there exists $c > 0$, depending only on the eigenvalues of $\mathbf{A}$, such that*

$$E_k(z) = \left\| \sum_{j=0}^{k-1} \mathcal{S}_\Sigma \left[p_j w \right](z) p_j(\mathbf{A})\mathbf{b} - (\mathbf{A} - z\mathbf{I})^{-1}\mathbf{b} \right\|_2,$$

satisfies

$$E_k(z) \le c\|\mathbf{V}\|_2 \left\|\mathbf{V}^{-1}\right\|_2 \|\mathbf{b}\|_2 \frac{\mathrm{e}^{k\nu(z;\mathbf{A})}}{1-\mathrm{e}^{\nu(z;\mathbf{A})}},$$

where $\mathbf{A}$ is diagonalized as $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. In particular, the Akhiezer iteration converges, at worst, at a geometric rate given by $\mathrm{e}^{\nu(z;\mathbf{A})}$ if $z \notin \Sigma$ since

$$\|\mathbf{x}_k - (\mathbf{A} - z\mathbf{I})^{-1}\mathbf{b}\|_2 = E_k(z).$$

Proof. We have

$$\begin{aligned} E_k(z) &= \left\| \sum_{j=0}^{k-1} \mathcal{S}_\Sigma \left[p_j w \right](z) p_j(\mathbf{A})\mathbf{b} - (\mathbf{A} - z\mathbf{I})^{-1}\mathbf{b} \right\|_2 = \left\| \sum_{j=k}^{\infty} \mathcal{S}_\Sigma \left[p_j w \right](z) p_j(\mathbf{A})\mathbf{b} \right\|_2 \\ &= \left\| \sum_{j=k}^{\infty} 2\pi\mathrm{i}\mathcal{C}_\Sigma \left[p_j w \right](z) \mathbf{V} p_j(\mathbf{\Lambda})\mathbf{V}^{-1}\mathbf{b} \right\|_2 \\ &\le \|\mathbf{V}\|_2 \left\|\mathbf{V}^{-1}\right\|_2 \|\mathbf{b}\|_2 \sum_{j=k}^{\infty} \max_{\ell\in\{1,\ldots,n\}} \left| 2\pi\mathrm{i}\mathcal{C}_\Sigma \left[p_j w \right](z) p_j(\lambda_\ell) \right|. \end{aligned}$$

Since λ_ℓ is not at an endpoint of Σ, by Lemma 5.2.2,

$$|2\pi\mathrm{i}\mathcal{C}_\Sigma \left[p_j w \right](z) p_j(\lambda_\ell)| = |\delta_j(z)||\hat{\delta}_j(\lambda_\ell)| \mathrm{e}^{j\mathrm{Re}(\mathfrak{g}_\Sigma(\lambda_\ell) - \mathfrak{g}_\Sigma(z))} \le c\mathrm{e}^{j\nu(z;\mathbf{A})},$$

for some constant c. The theorem follows from

$$\sum_{j=k}^{\infty} \mathrm{e}^{j\nu(z;\mathbf{A})} = \frac{\mathrm{e}^{k\nu(z;\mathbf{A})}}{1-\mathrm{e}^{\nu(z;\mathbf{A})}}.$$

□

Remark 5.2.6. *Theorem 5.2.5 could, in principle, be extended to a broader class of matrices. For instance, if $\mathbf{A}$ does have eigenvalues at the endpoints of Σ, the orthogonal polynomials at these endpoints are either bounded or grow at a polynomial rate. Since the convergence rate is geometric, this growth could be accounted for in the error bound without inhibiting convergence. However, one could instead perturb Σ so that these eigenvalues lie in the interior,*

and the iteration will suffer a small perturbation to its geometric rate of convergence.

Remark 5.2.7. *Jordan blocks, i.e., degenerate eigenspaces for* $\mathbf{A}$*, could be accounted for by investigating the derivatives of the orthogonal polynomials. Due to the Markov brothers' inequality, [129, Theorem 1.10], it is clear that this will not change the exponential rate of convergence. Yet, more refined estimates are possible.*

5.3 The Akhiezer iteration for matrix function approximation

The Akhiezer iteration can be extended to compute matrix functions for square matrices $\mathbf{A}$ with eigenvalues on or near $\Sigma = \bigcup_{j=1}^{g+1}[a_j, b_j]$. The function of a matrix $f(\mathbf{M})$ can be defined in several equivalent ways. For our purposes, the following two definitions will suffice:

Definition 5.3.1. *Suppose that* f *is a scalar-valued function defined on the spectrum of a diagonalizable matrix* $\mathbf{M} \in \mathbb{C}^{n\times n}$*, where* $\mathbf{M}$ *is diagonalized as* $\mathbf{M} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$*,* $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$*. Then,*

$$f(\mathbf{M}) := \mathbf{V}\operatorname{diag}\left(f(\lambda_1), \ldots, f(\lambda_n)\right)\mathbf{V}^{-1}.$$

With suitable assumptions on f, this definition can be extended to nondiagonalizable matrices via the Jordan normal form, but we will restrict most of our analysis to diagonalizable matrices. When f is appropriately analytic, we have the following equivalent definition [86, Theorem 1.12]:

Definition 5.3.2. *Suppose that* Γ *is a counterclockwise-oriented simple curve that encloses the spectrum of* $\mathbf{M} \in \mathbb{C}^{n\times n}$ *and that* f *is analytic in a region containing* Γ *and its interior. Then,*

$$f(\mathbf{M}) := \frac{1}{2\pi\mathrm{i}}\int_\Gamma f(z)(z\mathbf{I} - \mathbf{M})^{-1}\mathrm{d}z.$$

Applying a quadrature rule to Definition 5.3.2, such as a trapezoid rule [153], with nodes z_j and weights w_j, this can be approximated as

$$f(\mathbf{A})\mathbf{b} \approx \frac{1}{2\pi\mathrm{i}}\sum_{j=1}^{m} f(z_j)w_j(z_j\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}.$$

In practice, we find that it typically suffices to take $\Gamma = \bigcup_{j=1}^{g+1}\Gamma_j$ to be a collection of circles around each $[a_j, b_j]$. In this case the trapezoid rule on each Γ_k has weights and nodes

$$w_j = \frac{2\pi\mathrm{i}r_k}{m_k}\mathrm{e}^{\mathrm{i}\theta_j^{(k)}}, \quad z_j = r_k\mathrm{e}^{\mathrm{i}\theta_j^{(k)}} + c_k, \quad j = 1, \ldots, m_k,$$

where $\{\theta_j^{(k)}\}_{j=1}^{m_k}$ are evenly spaced points in $[0, 2\pi)$, r_k and c_k are the radius and center of Γ_k, respectively, and m_k is the number of quadrature nodes on Γ_k, $\sum_{j=1}^{g+1} m_j = m$. For

convenience, we reorder the weights and nodes for all contours Γ_k as

$$z_j, w_j, \quad j = 1, 2, \ldots, m,$$

with the first m_1 corresponding to Γ_1, the next m_2 corresponding to Γ_2, and so on.

To approximate $f(\mathbf{A})\mathbf{b}$, the resolvent is approximated by

$$(z_j\mathbf{I} - \mathbf{A})^{-1} = -\sum_{\ell=0}^{\infty} \mathcal{S}_\Sigma\left[p_\ell w\right](z_j) p_\ell(\mathbf{A}) \approx -\sum_{\ell=0}^{k-1} \mathcal{S}_\Sigma\left[p_\ell w\right](z_j) p_\ell(\mathbf{A}),$$

and the matrix function is approximated by

$$f(\mathbf{A})\mathbf{b} \approx \mathbf{f}_k := f_{k,m}(\mathbf{A})\mathbf{b} := -\sum_{\ell=0}^{k-1} \left(\sum_{j=1}^{m} f(z_j) w_j \mathcal{C}_\Sigma\left[p_\ell w\right](z_j) \right) p_\ell(\mathbf{A})\mathbf{b}. \tag{5.4}$$

As before, the recurrence coefficients and Cauchy integrals can be precomputed either by the Riemann–Hilbert approach of Chapter 4 or by (2.24) and (2.25) in the case of the $g = 1$ Akhiezer weight. Then, the Akhiezer iteration for matrix function approximation can be implemented as in Algorithm 3.

Algorithm 3: Akhiezer iteration for matrix function approximation

Input: f, $\mathbf{A}$, $\mathbf{b}$, quadrature nodes $z_1, \ldots, z_m$, quadrature weights $w_1, \ldots, w_m$, and a function to compute recurrence coefficients $\check{a}_k, \check{b}_k$ and Cauchy integrals $C_{k,j} = \mathcal{C}_\Sigma\left[p_k w\right](z_j)$.

Set $\mathbf{f}_0 = 0$.

for *k=0,1,...* **do**

  if *k=0* **then**

    Set $\mathbf{p}_0 = \mathbf{b}$.

  else if *k=1* **then**

    Set $\mathbf{p}_1 = \frac{1}{\check{b}_0}(\mathbf{A}\mathbf{p}_0 - \check{a}_0\mathbf{p}_0)$.

  else

    Set $\mathbf{p}_k = \frac{1}{\check{b}_{k-1}}(\mathbf{A}\mathbf{p}_{k-1} - \check{a}_{k-1}\mathbf{p}_{k-1} - \check{b}_{k-2}\mathbf{p}_{k-2})$.

  end

  Set $\alpha_k = -\sum_{j=1}^{m} C_{k,j} f(z_j) w_j$.

  Set $\mathbf{f}_{k+1} = \mathbf{f}_k + \alpha_k \mathbf{p}_k$.

  if converged **then**

    return $\mathbf{f}_{k+1}$.

  end

end

Remark 5.3.3. *To compute the full matrix function $f(\mathbf{A})$ rather than its action on a vector, one may take $\mathbf{b} = \mathbf{I}$. The algorithm then performs matrix–matrix multiplications rather than matrix–vector products at each step, but is otherwise identical.*

Remark 5.3.4. *The approximation to f need not be from the discretization of a Cauchy integral. It could be from a Padé approximation, a AAA approximation [116], or some other form of rational approximation. As long as the approximation can be written in pole-residue form, the Akhiezer iteration can be applied. We refer the reader back to Remark 5.2.1 for the case when the approximation contains more than just simple poles.*

5.3.1 Convergence

Theorem 5.3.5. *Suppose $\mathbf{A} \in \mathbb{C}^{n\times n}$ is generic with eigenvalues satisfying $\operatorname{Re}_{\mathfrak{g}_\Sigma}(\lambda_\ell) < \min_j \operatorname{Re}_{\mathfrak{g}_\Sigma}(z_j)$, and suppose $|f(z)| \le M$ for $z \in \Gamma$ and $\sum_{j=1}^m |w_j| \le 2\pi L$. Then,*

$$\|f(\mathbf{A})\mathbf{b} - f_{k,m}(\mathbf{A})\mathbf{b}\|_2 \le \frac{1}{2\pi}\|\mathbf{V}\|_2 \left\|\mathbf{V}^{-1}\right\|_2 \|\mathbf{b}\|_2 \max_{\ell\in\{1,\dots,n\}} |\rho_m(\lambda_\ell)| + LM \max_{j\in\{1,\dots,m\}} E_k(z_j),$$

where $\mathbf{A}$ is diagonalized as $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$, and

$$\rho_m(x) = \int_\Gamma \frac{f(z)}{z-x}\mathrm{d}z - \sum_{j=1}^m \frac{f(z_j)}{z_j - x} w_j.$$

Proof. By the triangle inequality,

$$\begin{aligned}\|f(\mathbf{A})\mathbf{b} - f_{k,m}(\mathbf{A})\mathbf{b}\|_2 &\le \left\| f(\mathbf{A})\mathbf{b} - \frac{1}{2\pi\mathrm{i}}\sum_{j=1}^m f(z_j)w_j(z_j\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}\right\|_2 \\ &+ \left\|\frac{1}{2\pi\mathrm{i}}\sum_{j=1}^m f(z_j)w_j(z_j\mathbf{I}-\mathbf{A})^{-1}\mathbf{b} - f_{k,m}(\mathbf{A})\mathbf{b}\right\|_2.\end{aligned}$$

Then,

$$\begin{aligned}&\left\| f(\mathbf{A})\mathbf{b} - \frac{1}{2\pi\mathrm{i}}\sum_{j=1}^m f(z_j)w_j(z_j\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}\right\|_2 \\ &= \left\|\frac{1}{2\pi\mathrm{i}}\mathbf{V}^{-1}\int_\Gamma f(z)(z\mathbf{I}-\mathbf{\Lambda})^{-1}\mathbf{V}\mathbf{b}\mathrm{d}z - \frac{1}{2\pi\mathrm{i}}\mathbf{V}^{-1}\sum_{j=1}^m f(z_j)w_j(z_j\mathbf{I}-\mathbf{\Lambda})^{-1}\mathbf{V}\mathbf{b}\right\|_2 \\ &\le \frac{1}{2\pi}\|\mathbf{V}\|_2\left\|\mathbf{V}^{-1}\right\|_2\|\mathbf{b}\|_2 \max_{\ell\in\{1,\dots,n\}}|\rho_m(\lambda_\ell)|.\end{aligned}$$

Then, in the notation of Theorem 5.2.5,

$$\begin{aligned}
&\left\| \frac{1}{2\pi \mathrm{i}} \sum_{j=1}^{m} f(z_j) w_j (z_j \mathbf{I} - \mathbf{A})^{-1} \mathbf{b} - f_{k,m}(\mathbf{A}) \mathbf{b} \right\|_2 \\
&\leq \frac{1}{2\pi} \sum_{j=1}^{m} \left\| f(z_j) w_j (z_j \mathbf{I} - \mathbf{A})^{-1} \mathbf{b} + f(z_j) w_j \sum_{\ell=0}^{k-1} \mathcal{S}_\Sigma \left[p_\ell w \right](z_j) p_\ell(\mathbf{A}) \mathbf{b} \right\|_2 \\
&= \frac{1}{2\pi} \sum_{j=1}^{m} |f(z_j)||w_j| E_k(z_j) \leq LM \max_{j \in \{1,\ldots,m\}} E_k(z_j),
\end{aligned}$$

and the result follows. □

For a reasonable quadrature rule, $\lim_{m\to\infty} \rho_m(x) = 0$ for all x, likely exponentially, so provided that m is taken to be sufficiently large, the Akhiezer iteration converges, at worst, at an asymptotic geometric rate given by $\mathrm{e}^{\max_j \nu(z_j;\mathbf{A})}$ provided that $\mathbf{A}$ is generic and its eigenvalues λ_ℓ all satisfy $\mathrm{Re}\mathfrak{g}_\Sigma(\lambda_\ell) < \min_{j=1,\ldots,m} \mathrm{Re}\mathfrak{g}_\Sigma(z_j)$. In practice, we find that this bound is quite pessimistic, so we offer an alternative view. We first state an analog of Theorem 2.3.3.

Lemma 5.3.6. *Let $\Gamma_\varrho(\Sigma)$ denote the level curve $\mathrm{e}^{\mathrm{Re}\mathfrak{g}_\Sigma(z)} = \varrho$ for $\varrho > 1$, and assume that f is analytic in the interior of $\Gamma_\varrho(\Sigma)$ and satisfies $|f(z)| \leq M$ for z interior to $\Gamma_\varrho(\Sigma)$. Then, there exists a constant $C > 0$, independent of f, such that for all $\ell \in \mathbb{N}$,*

$$\left| \langle f, p_\ell \rangle_{L^2_w(\Sigma)} \right| \leq CM \varrho^{-\ell}.$$

Proof. Define $\Omega_\eta = \{ z \in \mathbb{C} \ : \ \varrho^\eta \leq \mathrm{e}^{\mathrm{Re}\mathfrak{g}_\Sigma(z)} \leq \varrho \}$ for some $0 < \eta < 1$. Choose ϵ in Lemma 5.2.2 to be sufficiently small such that $\Omega_\eta \subset V$, and denote the uniform bound on $|\delta_\ell(z)|$ by D. Let ϱ' satisfy $1 < \varrho^\eta < \varrho' < \varrho$, and consider the level curve $\Gamma_{\varrho'}(\Sigma)$. Then, by Cauchy's integral formula and Lemma 5.2.2,

$$\begin{aligned}
\left| \langle f, p_\ell \rangle_{L^2_w(\Sigma)} \right| &= \left| \int_\Sigma \frac{1}{2\pi \mathrm{i}} \int_{\Gamma_{\varrho'}(\Sigma)} \frac{f(z)}{z - x} \mathrm{d}z p_\ell(x) w(x) \mathrm{d}x \right| \\
&= \left| - \int_{\Gamma_{\varrho'}(\Sigma)} f(z) \mathcal{C}_\Sigma \left[p_\ell w \right](z) \mathrm{d}z \right| \leq |\Gamma_\varrho(\Sigma)| \, M \frac{D}{2\pi} \varrho'^{-\ell},
\end{aligned}$$

using $C = \frac{1}{2\pi} |\Gamma_\varrho(\Sigma)| \, D$. Since this inequality holds for all $\varrho' < \varrho$, it must hold for ϱ. □

Theorem 5.3.7. *Under the assumptions of Lemma 5.3.6, suppose that*

$$\sum_{\ell=0}^{\infty} \varrho^{-\ell} \| p_\ell(\mathbf{A}) \mathbf{b} \|_2 < \infty.$$

Then, there exist $(\varepsilon_{m,\ell})$ *satisfying* $|\varepsilon_{m,\ell}| \leq \|\rho_m\|_\infty := \max_{x\in\Sigma} |\rho_m(x)|$ *and a constant* $C > 0$ *such that in any norm*

$$\left\| f(\mathbf{A})\mathbf{b} - f_{k,m}(\mathbf{A})\mathbf{b} + \sum_{\ell=0}^{k-1} \varepsilon_{m,\ell} p_\ell(\mathbf{A})\mathbf{b} \right\| \leq CM \sum_{\ell=k}^{\infty} \varrho^{-\ell} \|p_\ell(\mathbf{A})\mathbf{b}\|.$$

Proof. For a given ℓ,

$$-\sum_{j=1}^{m} f(z_j) w_j \mathcal{C}_\Sigma \left[p_\ell w\right](z_j) = \int_\Sigma p_\ell(x) \frac{1}{2\pi \mathrm{i}} \sum_{j=1}^{m} \frac{f(z_j) w_j}{z_j - x} w(x) \mathrm{d}x = \langle f_m, p_\ell \rangle_{L^2_w(\Sigma)},$$

where

$$f_m(x) = \frac{1}{2\pi \mathrm{i}} \sum_{j=1}^{m} \frac{f(z_j) w_j}{z_j - x} \approx f(x).$$

Thus,

$$f_{k,m} = \sum_{\ell=0}^{k-1} \langle f_m, p_\ell \rangle_{L^2_w(\Sigma)} p_\ell.$$

Let $\varepsilon_{m,\ell} = \langle f_m - f, p_\ell \rangle_{L^2_w(\Sigma)}$. By the Cauchy–Schwarz inequality,

$$|\varepsilon_{m,\ell}| \leq \|f_m - f\|_{L^2_w(\Sigma)} \leq \|\rho_m\|_\infty.$$

Then,

$$f(\mathbf{A})\mathbf{b} - f_{k,m}(\mathbf{A})\mathbf{b} + \sum_{\ell=0}^{k-1} \varepsilon_{m,\ell} p_\ell(\mathbf{A})\mathbf{b} = \sum_{\ell=k}^{\infty} \langle f, p_\ell \rangle_{L^2_w(\Sigma)} p_\ell(\mathbf{A})\mathbf{b}.$$

By the triangle inequality and Lemma 5.3.6, there exists some $C > 0$ such that

$$\left\| f(\mathbf{A})\mathbf{b} - f_{k,m}(\mathbf{A})\mathbf{b} + \sum_{\ell=0}^{k-1} \varepsilon_{m,\ell} p_\ell(\mathbf{A})\mathbf{b} \right\| \leq CM \sum_{\ell=k}^{\infty} \varrho^{-\ell} \|p_\ell(\mathbf{A})\mathbf{b}\|.$$

□

The bound from Theorem 5.3.7 implies that when $\mathbf{A}$ is diagonalized as $\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$,

$$\|f(\mathbf{A})\mathbf{b} - f_{k,m}(\mathbf{A})\mathbf{b}\|_2 \leq \|\mathbf{V}\|_2 \|\mathbf{V}^{-1}\|_2 \|\mathbf{b}\|_2 \left[\|\rho_m\|_\infty \sum_{\ell=0}^{k-1} \|p_\ell(\mathbf{\Lambda})\|_2 + CM \sum_{\ell=k}^{\infty} \varrho^{-\ell} \|p_\ell(\mathbf{\Lambda})\|_2 \right].$$

This demonstrates that if the quadrature rule is taken such that $\|\rho_m\|_\infty$ is sufficiently small, then the convergence rate will be largely determined by the distance from Σ to the nearest singularity of f. More precisely, $\|f(\mathbf{A})\mathbf{b} - f_{k,m}(\mathbf{A})\mathbf{b}\| = \mathrm{O}(\varrho^{-k} + \|\rho_m\|_\infty)$ when the spectrum

of $\mathbf{A}$ is contained in Σ, and the rate of convergence will be governed by the furthest level curve $\Gamma_\varrho(\Sigma)$ to which f can be analytically extended. If f is entire, the initial convergence is superexponential. The level curves of Lemma 5.3.6 can be thought of as higher genus analogs to Bernstein ellipses as they dictate the convergence rate in the same manner. A contour plot of $\mathrm{e}^{\mathrm{Reg}_\Sigma(z)}$ is included as part of Figure 5.3.

Because Theorem 5.3.5 gives a different bound, we have, supposing that Γ lies inside $\Gamma_\varrho(\Sigma)$,

$$\begin{aligned}\frac{\|f(\mathbf{A})\mathbf{b}-f_{k,m}(\mathbf{A})\mathbf{b}\|_2}{\|\mathbf{V}\|_2\|\mathbf{V}^{-1}\|_2\|\mathbf{b}\|_2} &\leq \min\left\{\|\rho_m\|_\infty\sum_{\ell=0}^{k-1}\|p_\ell(\mathbf{\Lambda})\|_2+CM\sum_{\ell=k}^{\infty}\varrho^{-\ell}\|p_\ell(\mathbf{\Lambda})\|_2,\right.\\ &\qquad\left.\frac{1}{2\pi}\|\rho_m(\mathbf{\Lambda})\|_2+cLM\max_{j\in\{1,\ldots,m\}}\frac{\mathrm{e}^{k\nu(z_j;\mathbf{A})}}{1-\mathrm{e}^{\nu(z_j;\mathbf{A})}}\right\}\\ &\leq \min\left\{c'\|\rho_m\|_\infty\sum_{\ell=0}^{k-1}\mathrm{e}^{\ell\nu(\mathbf{A})}+c'CM\frac{\mathrm{e}^{k\nu(z_*;\mathbf{A})}}{1-\mathrm{e}^{\nu(z_*;\mathbf{A})}},\right.\\ &\qquad\left.\frac{1}{2\pi}\|\rho_m(\mathbf{\Lambda})\|_2+cLM\max_{j\in\{1,\ldots,m\}}\frac{\mathrm{e}^{k\nu(z_j;\mathbf{A})}}{1-\mathrm{e}^{\nu(z_j;\mathbf{A})}}\right\},\end{aligned}$$

for another constant $c'>0$ by Lemma 5.2.2. Here, z_* is any point on $\Gamma_\varrho(\Sigma)$.

If $\mathbf{A}$ is generic and the spectrum of $\mathbf{A}$ is contained in Σ, then $\nu(\mathbf{A})=0$ ($\|p_\ell(\mathbf{A})\|_2=\mathrm{O}(1)$) and the first term inside the minimum is likely to be smaller than the second since the nodes z_j are inside $\Gamma_\varrho(\Sigma)$, but, eventually, the first term will dominate the second when $\sum_{\ell=0}^{k-1}\|p_\ell(\mathbf{\Lambda})\|_2$ becomes large. The second term then implies that the errors remain small, even if this sum is large. Also, if $\mathbf{A}$ is generic but the eigenvalues of $\mathbf{A}$ are not contained in Σ, the second term here will still be small, giving useful bounds. The first term will also likely be small, but as $\nu(\mathbf{A})>0$, the sum grows exponentially with respect to k. In this case, the balance, i.e., which term gives a better bound in a given situation, will change.

5.3.2 Complexity

To run Algorithm 3 for k steps, using that computing $\breve{a}_j,\breve{b}_j,S_j$ requires $\mathrm{O}(1)$ arithmetic operations and supposing that $T_{\mathbf{A}}$ is the complexity of matrix multiplication by $\mathbf{A}$, we need less than $kT_{\mathbf{A}}+7kn+km+\mathrm{O}((1+m)k)$ arithmetic operations if $\mathbf{A}\in\mathbb{C}^{n\times n}$. The $\mathrm{O}((1+m)k)$ term represents the computation of $\breve{a}_j,\breve{b}_j,S_j$ for $j=0,1,\ldots k$. The complexity of the first Akhiezer iteration, Algorithm 2, is found by setting $m=0$. Furthermore, if a tolerance of ϵ is desired for function approximation, we expect to set $m=\mathrm{O}(\log\epsilon^{-1})$. This is in comparison with the method in [43] where the complexity is $kT_{\mathbf{A}}+\mathrm{O}(k^2m+n)$, and $m=\mathrm{O}(\epsilon^{2/7})$ is required, in general, to achieve a tolerance of ϵ.

5.4 Adaptivity

In the case where the matrix $\mathbf{A}$ is symmetric, i.e., it has all real eigenvalues, the asymptotic behavior of the polynomials yields an adaptive algorithm that estimates a good choice for the bands $\Sigma = \bigcup_{j=1}^{g+1}[a_j, b_j]$ then applies the iteration (5.3) to solve the system $\mathbf{Ax} = \mathbf{b}$. For simplicity, we will consider only the $g = 1$ case $\Sigma = [a_1, b_1] \cup [a_2, b_2]$, but such an approach can be applied to the $g > 1$ cases as well. Furthermore, such an approach could be used as a first step in computing a matrix function by producing an estimate of the spectrum of $\mathbf{A}$.

By Lemma 5.2.2, if not all eigenvalues of $\mathbf{A}$ are contained in Σ, the matrix polynomials $p_j(\mathbf{A})$, and therefore $\|p_j(\mathbf{A})\mathbf{b}\|$ as well, grow at an asymptotic rate given by $\mathrm{e}^{\max_k \mathrm{Reg}_\Sigma(\lambda_k)}$, where $\lambda_1, \ldots, \lambda_n$ denote the eigenvalues of $\mathbf{A}$. Thus, if the approximate growth rate is given by r, then performing root finding on $\mathrm{e}^{\mathrm{Reg}(z)} - r$ around each endpoint of Σ will yield a new set of bands that contain the eigenvalues of $\mathbf{A}$. A naive implementation of this is given as Algorithm 4. This algorithm ensures that the Akhiezer iteration on the outputted

Algorithm 4: Adaptive band selection

Input: Initial guess for bands $\Sigma = [a_1, b_1] \cup [a_2, b_2]$ where $b_1 < 0 < a_2$, $\mathbf{A} \in \mathbb{C}^{n\times n}$, $\mathbf{A} = \mathbf{A}^*$, $\mathbf{b} \in \mathbb{C}^n$, parameters $\gamma_o > 1$, $0 < \gamma_i < 1$.

while Norms of matrix polynomials $\|p_j(\mathbf{A})\mathbf{b}\|$ grow as $j \to \infty$ **do**

- Compute the growth rate r of $\|p_j(\mathbf{A})\mathbf{b}\|$ as $j \to \infty$.
- Set $f(x) = \mathrm{e}^{\mathrm{Reg}_\Sigma(x)} - r$ for $\mathfrak{g}_\Sigma$ corresponding to $[a_1, b_1] \cup [a_2, b_2]$.
- Perform bisection on f on $[\gamma_o a_1, a_1]$. If bisection fails, set $a_1 \leftarrow \gamma_o a_1$; otherwise set $a_1 \leftarrow c$ where c is the computed root.
- Perform bisection on f on $[b_1, \gamma_i b_1]$. If bisection fails, set $b_1 \leftarrow \gamma_i b_1$; otherwise set $b_1 \leftarrow c$ where c is the computed root.
- Perform bisection on f on $[\gamma_i a_2, a_2]$. If bisection fails, set $a_2 \leftarrow \gamma_i a_2$; otherwise set $a_2 \leftarrow c$ where c is the computed root.
- Perform bisection on f on $[b_2, \gamma_o b_2]$. If bisection fails, set $b_2 \leftarrow \gamma_o b_2$; otherwise set $b_2 \leftarrow c$ where c is the computed root.
- Compute norms of matrix polynomials $\|p_j(\mathbf{A})\mathbf{b}\|$ for $p_0, p_1, \ldots$ corresponding to the updated bands $\Sigma = [a_1, b_1] \cup [a_2, b_2]$.

end

return $\Sigma = [a_1, b_1] \cup [a_2, b_2]$

bands $\Sigma = [a_1, b_1] \cup [a_2, b_2]$ applied to $\mathbf{Ax} = \mathbf{b}$ will converge at the rate $\mathrm{e}^{-\mathrm{Reg}_\Sigma(0)}$. In practice, computing the growth rate of the matrix polynomials can be challenging due to theta function oscillations implicit in the orthogonal polynomials. One way to do this is to consider the ratio

$$\frac{|p_{n+k}(z)|}{|p_n(z)|} = \frac{|\hat{\delta}_{n+k}(z)|}{|\hat{\delta}_n(z)|}\mathrm{e}^{k\mathrm{Reg}_\Sigma(z)},$$

which follows from Lemma 5.2.2. Since $\hat{\delta}_n(z) = \mathrm{O}(1)$, for sufficiently large n and k,

$$r \approx \mathrm{e}^{\mathrm{Reg}_\Sigma(z)} \approx \sqrt[k]{\frac{|p_{n+k}(z)|}{|p_n(z)|}}.$$

Of course, what "sufficiently large" means in practice is not always clear, and one may need to experiment with different values of n and k for different matrices.

The factor $\mathrm{e}^{-\mathrm{Reg}_\Sigma(0)}$ typically increases as the sizes of the intervals increase, so the convergence rate will be suboptimal if the eigenvalues of $\mathbf{A}$ are contained in some $\Sigma' \subsetneq \Sigma$. The convergence rate can be improved by modifying Algorithm 4 so that only one endpoint is moved at a time and the new growth rate is computed after each movement. If the growth rate decreases, the algorithm proceeds to the next endpoint. Otherwise, the endpoint is reverted to its previous value and the algorithm proceeds.

An improved band estimation technique can be employed if one is willing to compute up to 4 (or more generally, $2g + 2$) inner products. This is done by approximating the eigenvector corresponding to the eigenvalue with the most growth. In particular, one computes $\tilde{\mathbf{y}} = p_n(\mathbf{A})\mathbf{b}$ for a sufficiently large n and normalizes $\mathbf{y} = \frac{\tilde{\mathbf{y}}}{\|\tilde{\mathbf{y}}\|}$ to avoid round-off errors. This is iterated as $\tilde{\mathbf{y}} = p_n(\mathbf{A})\mathbf{y}$, $\mathbf{y} = \frac{\tilde{\mathbf{y}}}{\|\tilde{\mathbf{y}}\|}$ until convergence. Then, the eigenvalue corresponding to the largest growth rate is approximated by the Rayleigh quotient

$$\frac{\mathbf{y}^T\mathbf{A}\mathbf{y}}{\mathbf{y}^T\mathbf{y}}.$$

The relevant endpoint is then moved to strictly contain this eigenvalue, and the algorithm iterates until all approximated eigenvalues fall within the bands. We find this approach to be the most accurate, and the eigenvalues are computed to machine precision, so the algorithm needs only 4 Rayleigh quotient computations to accurately approximate the true bands.

On the other hand, if one only cares that the iteration converges and is not concerned with the rate, a simpler approach can be applied. If one takes the initial guess for the bands to be the symmetric intervals $[-b, -a] \cup [a, b]$, convergence can be obtained without moving the interior endpoints. This follows from the fact that the iteration converges if $\mathrm{Reg}_\Sigma(\lambda_k) < \mathrm{Reg}_\Sigma(0)$ for all eigenvalues λ_k of $\mathbf{A}$ and that $\mathrm{Reg}_\Sigma(z)$ is maximized on $[-a, a]$ at $z = 0$. Thus, one only needs to adjust the parameter b so that $\mathrm{Reg}_\Sigma(\lambda_k) < \mathrm{Reg}_\Sigma(0)$ for all eigenvalues such that $|\lambda_k| > b$.

5.5 Examples and Applications

Since this work is inspired by that of Saad, we first consider an example presented in [132, Section 8.1]. Let $\mathbf{A} \in \mathbb{R}^{200\times 200}$ be a diagonal matrix with eigenvalues spaced uniformly in $[-2, -0.5] \cup [0.5, 6]$, and let $\mathbf{b} = \mathbf{A}\mathbf{e}$ where $\mathbf{e} \in \mathbb{R}^{200}$ is the vector of all ones. We run the Akhiezer iteration using $g = 1$ Akhiezer polynomials and plot the relative residuals in Figure

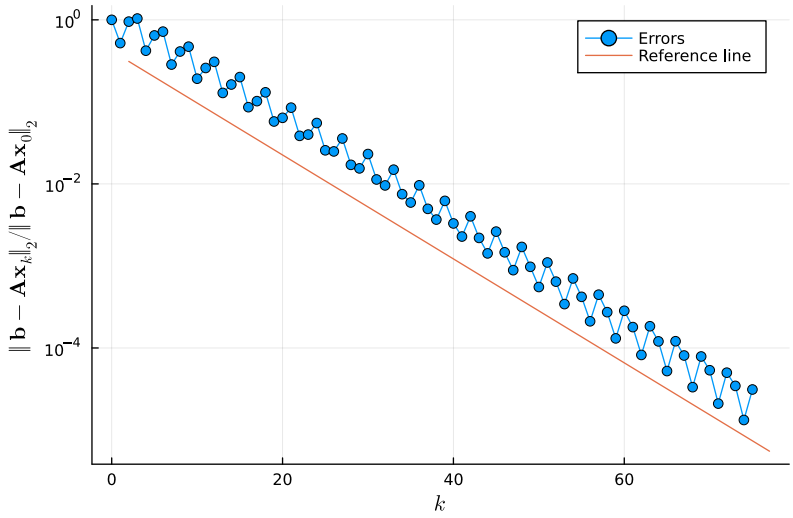


Figure 5.2: Relative residuals at each iteration of the Akhiezer iteration applied to a diagonal matrix $\mathbf{A} \in \mathbb{R}^{200\times 200}$ with eigenvalues spaced uniformly in $[-2,-0.5] \cup [0.5,6]$ and $\mathbf{b} = \mathbf{A}\mathbf{e}$. The rate of convergence is well-approximated by the reference line $Ce^{-k\mathrm{Reg}_\Sigma(0)}$.

5.2. We observe that this looks similar to the result in [132].

To construct a more complicated example, we consider a 200×200 real matrix with randomly generated eigenvalues in $[-2,-0.5] \cup [0.5,6]$. We construct $\mathbf{A}$ by moving the eigenvalues off the intervals by adding a Gaussian perturbation and removing the near-diagonal structure by conjugating by an orthogonal matrix. Again, we take $\mathbf{b} = \mathbf{A}\mathbf{e}$. In Figure 5.3, we plot the eigenvalues of $\mathbf{A}$ on a contour map of Reg_Σ and the relative residuals for the first 370 iterations of the Akhiezer iteration using $g = 1$ Akhiezer polynomials.

To demonstrate the convergence of the Akhiezer iteration for matrix function approximation, we consider the same matrix $\mathbf{A}$ as the previous example but let $\mathbf{b} \in \mathbb{R}^{200}$ be a random Gaussian vector. We let the contour be two circles: one centered around each interval with diameter 1.15 times the length of the interval. We take 200 quadrature nodes divided between the two circles proportionally to their width. First considering the entire function $f(x) = \exp(x)$, we use (5.4) with $g = 1$ Akhiezer polynomials to approximate $f(\mathbf{A})\mathbf{b}$ and plot the 2-norm of the difference between this approximation and $f(\mathbf{A})\mathbf{b}$ in Figure 5.4. We observe superexponential convergence until the iteration saturates due to rounding errors. Instead, letting $f(x) = \tanh(x)$ and then $f(x) = \exp(x)/x$, we plot the same errors at each iteration in Figures 5.5 and 5.6 for two different matrices $\mathbf{A}$. We observe that this approach allows one to compute $f(\mathbf{A})\mathbf{b}$ for functions f that have singularities at zero, albeit at a slower rate than for functions with singularities farther away from Σ. Furthermore, the bound of Theorem 5.3.7 estimates the convergence rate accurately when the spectrum of $\mathbf{A}$ is contained in Σ and is pessimistic but close when this is not true.

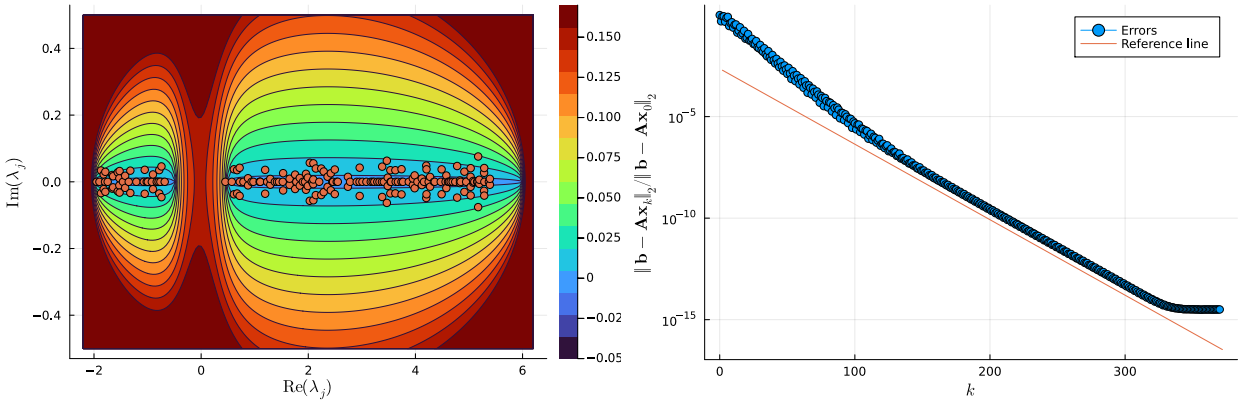


Figure 5.3: Eigenvalues of a matrix $\mathbf{A}$ superimposed on a contour plot of $\mathrm{Re}\mathfrak{g}_\Sigma(z)$ (left) and relative residuals at each iteration for the Akhiezer iteration applied to $\mathbf{A}\mathbf{x} = \mathbf{b} := \mathbf{A}\mathbf{e}$. The asymptotic rate of convergence is well-approximated by the reference line $Ce^{k\nu(0;\mathbf{A})}$.

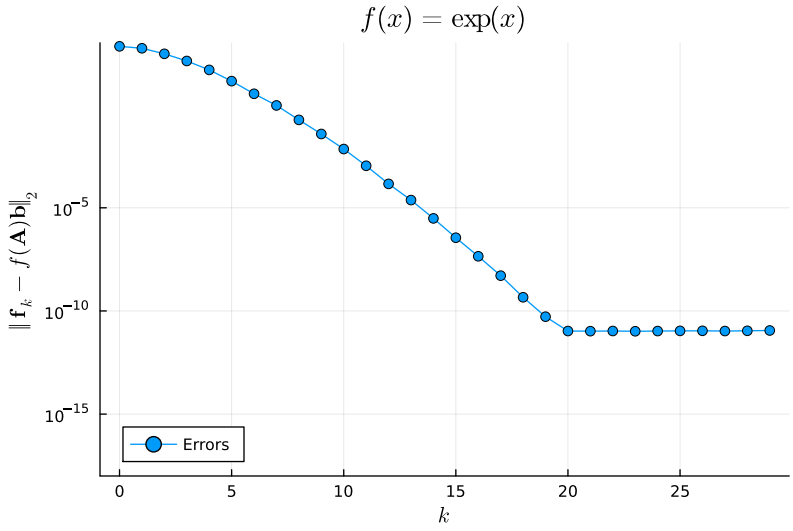


Figure 5.4: 2-norm error at each iteration for the Akhiezer iteration approximation to $f(\mathbf{A})\mathbf{b}$ where $f(x) = \exp(x)$, $\mathbf{A} \in \mathbb{R}^{200\times 200}$ is as in Figure 5.3, and $\mathbf{b} \in \mathbb{R}^{200}$ is a random Gaussian vector. The iteration converges superexponentially, saturating within 20 iterations.

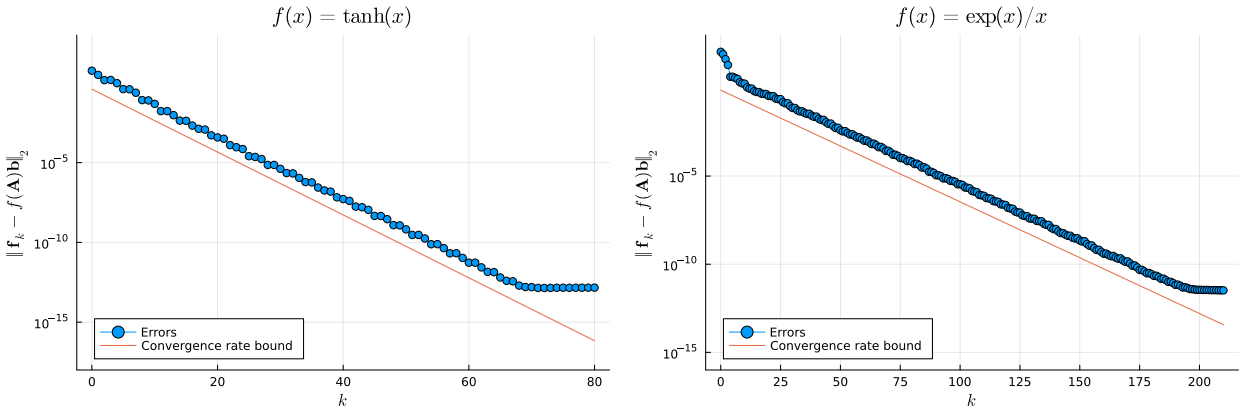


Figure 5.5: 2-norm error at each iteration for the Akhiezer iteration approximation to $f(\mathbf{A})\mathbf{b}$ where $f(x) = \tanh(x)$ (left) and $f(x) = \exp(x)/x$ (right), $\mathbf{A} \in \mathbb{R}^{200\times 200}$ has randomly generated eigenvalues in $[-2, -0.5] \cup [0.5, 6]$, and $\mathbf{b} \in \mathbb{R}^{200}$ is a random Gaussian vector. The convergence rate bounds $C_1 \mathrm{e}^{k\nu(\mathrm{i}\pi/2;\mathbf{A})}$ (left) and $C_2 \mathrm{e}^{k\nu(0;\mathbf{A})}$ (right) approximate the true convergence rates.

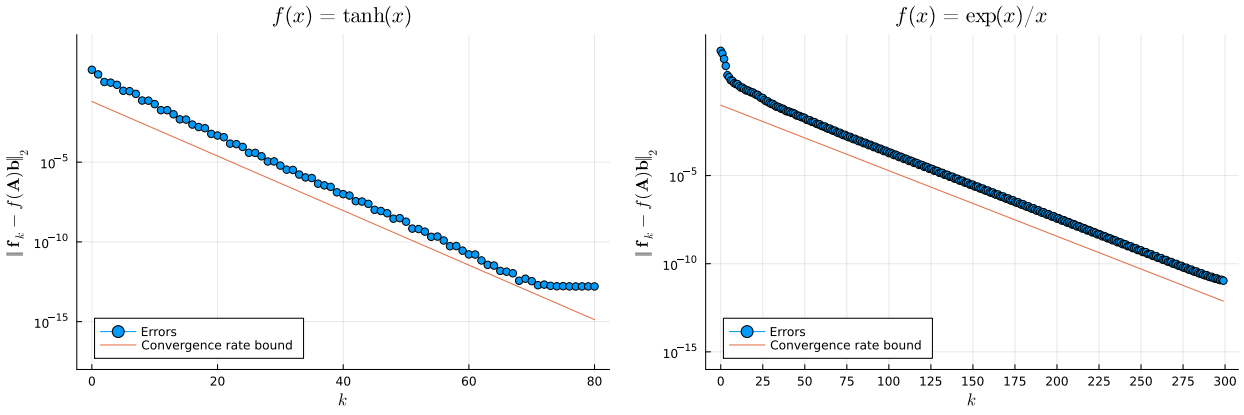


Figure 5.6: 2-norm error at each iteration for the Akhiezer iteration approximation to $f(\mathbf{A})\mathbf{b}$ where $f(x) = \tanh(x)$ (left) and $f(x) = \exp(x)/x$ (right), $\mathbf{A} \in \mathbb{R}^{200\times 200}$ is as in Figure 5.3, and $\mathbf{b} \in \mathbb{R}^{200}$ is a random Gaussian vector. The convergence rate bounds $C_1 \mathrm{e}^{k\nu(\mathrm{i}\pi/2;\mathbf{A})}$ (left) and $C_2 \mathrm{e}^{k\nu(0;\mathbf{A})}$ (right) approximate the true convergence rates.

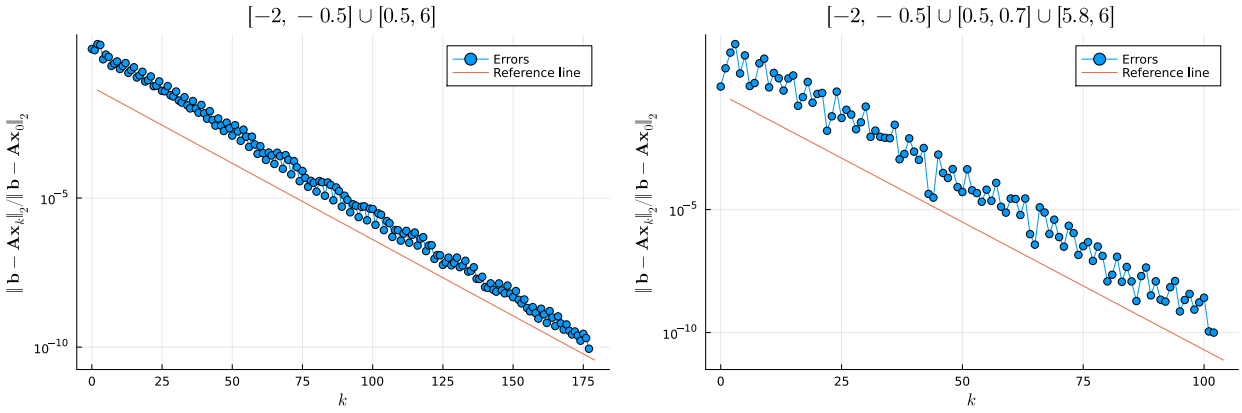


Figure 5.7: Relative residuals at each iteration for the Akhiezer iteration applied to $\mathbf{A}\mathbf{x} = \mathbf{b}$ where $\mathbf{A} \in \mathbb{R}^{200\times 200}$ has eigenvalues near $[-2, -0.5] \cup [0.5, 0.7] \cup [5.8, 6]$ and $\mathbf{b} \in \mathbb{R}^{200}$ is a random Gaussian vector with Akhiezer polynomials on $[-2, -0.5] \cup [0.5, 6]$ (left) and orthogonal polynomials with the weight (3.5) on $[-2, -0.5] \cup [0.5, 0.7] \cup [5.8, 6]$ (right). The convergence rate $\mathrm{e}^{\nu(0;\mathbf{A})}$ plotted in the reference line is approximately 0.888 (left) and 0.787 (right). The iteration converges to a relative residual of 10^{-10} in 177 iterations (left) and 102 iterations (right).

To illustrate the advantage of having more knowledge of the spectrum, we consider $\mathbf{A}$ and $\mathbf{b}$ constructed in the same manner as the previous example but with the original eigenvalues taken to be in $\Sigma = [-2, -0.5] \cup [0.5, 0.7] \cup [5.8, 6]$. In Figure 5.7, we plot the convergence of the Akhiezer iteration using Akhiezer polynomials on $[-2, -0.5] \cup [0.5, 6]$ and orthogonal polynomials with the weight (3.5) on Σ. We find that the iteration converges to a relative residual of 10^{-10} in 177 iterations in the former case and 102 in the latter in spite of the unbounded behavior at the endpoints of orthogonal polynomials with the weight (3.5).

As a final example, we consider the boundary-value problem (BVP) on $[0, 1]$:

$$-\frac{\mathrm{d}^2}{\mathrm{d}x^2}u(x) - 30\mathrm{e}^x u(x) = x,$$
$$u(0) = u(1) = 0.$$

Applying a centered finite difference approximation on an equally spaced grid of 100 points yields a symmetric indefinite linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with eigenvalues on the positive real axis that are unbounded as the grid spacing is refined. To control the eigenvalues, we apply a preconditioner $\mathbf{L}^{-1}$ where $\mathbf{L}$ corresponds to a centered finite difference approximation on the same grid applied to the BVP

$$-\frac{\mathrm{d}^2}{\mathrm{d}x^2}u(x) = 0,$$
$$u(0) = u(1) = 0.$$

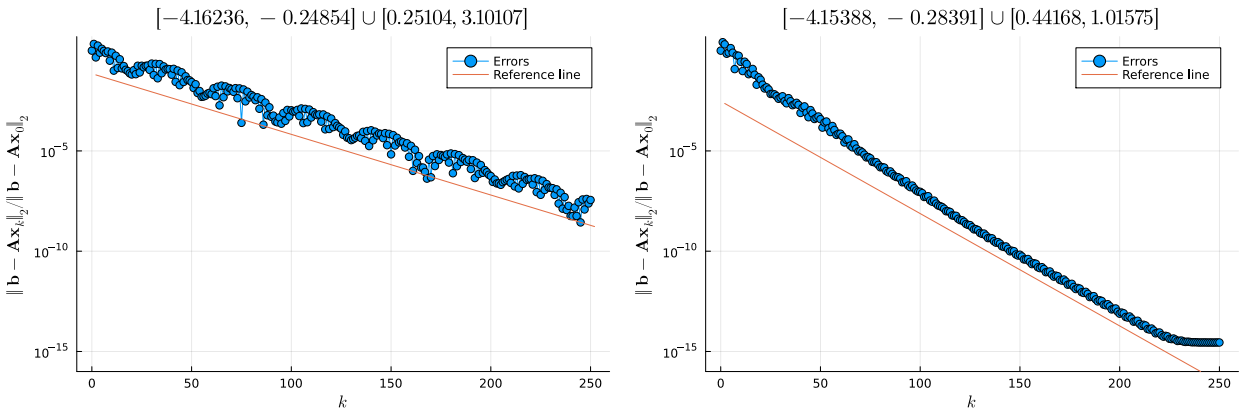


Figure 5.8: Relative residuals at each iteration for 250 iterations of the Akhiezer iteration applied to a preconditioned BVP for two different approaches for adaptive band selection (left: Algorithm 4, right: Algorithm 4, moving one endpoint at a time). The convergence rate $e^{\nu(0;\mathbf{A})}$ plotted in the reference line is approximately 0.933 (left) and 0.879 (right).

We then expect $\mathbf{L}^{-1}\mathbf{A}$ to have the majority of its eigenvalues bounded by 1 on the positive real axis with the remaining eigenvalues on the negative real axis. Thus, we apply Algorithm 4 to the system $\mathbf{L}^{-1}\mathbf{A}\mathbf{x} = \mathbf{L}^{-1}\mathbf{b}$ with an initial guess of $\Sigma = [-2, -0.5] \cup [0.5, 1]$ and parameters $\gamma_o = 5, \gamma_i = 0.7$ and obtain the bands $\Sigma = [-4.16236, -0.24854] \cup [0.25104, 3.10107]$. The true eigenvalues of $\mathbf{L}^{-1}\mathbf{A}$ fall in $[-4.14928, -0.28169] \cup [0.43062, 0.99921]$ with eigenvalues at each endpoint, so this algorithm does a good job of capturing the "worst offender" but far overshoots the right endpoint. However, applying the modification of Algorithm 4 to optimize the convergence rate described in Section 5.4 yields the bands $\Sigma = [-4.15388, -0.28391] \cup [0.44168, 1.01575]$; these are very close to the true bands, but note that the interval endpoint $-0.28391 < -0.28169$, and an eigenvalue lies outside the computed bands. This occurs due to difficulties in approximating the exact matrix polynomial growth rate (r in Algorithm 4) but gives just a slight perturbation to the geometric rate of convergence. Furthermore, applying the variation with inner products finds the true bands to full precision.

In Figure 5.8, we plot the relative residuals when applying the Akhiezer iteration to $\mathbf{L}^{-1}\mathbf{A}\mathbf{x} = \mathbf{L}^{-1}\mathbf{b}$ with the Akhiezer polynomials on $\Sigma = [-4.16236, -0.24854] \cup [0.25104, 3.10107]$ and the Akhiezer polynomials on $\Sigma = [-4.15388, -0.28391] \cup [0.44168, 1.01575]$ for 250 iterations.

5.6 Discussion and open questions

The Akhiezer iteration introduced in this chapter is extended to Sylvester matrix equations in Chapter 6, where the same polynomial approximation framework yields inverse-free iterative methods with sub-cubic arithmetic complexity for low-rank right-hand sides.

First, in this chapter, we have not addressed the question of adaptivity for matrices with eigenvalues off the real axis. In [106], this is handled for the classical Chebyshev iteration by estimating the convex hull of the spectrum via the power method and choosing the iteration parameters based on the known "optimal ellipse" containing this convex hull [105]. For the Akhiezer iteration, similar eigenvalue estimations could be employed, but one would instead need to look for at least two distinct regions of eigenvalues rather than the convex hull. Extending the notion of the optimal ellipse to a higher genus seems to be the primary difficulty. That is, given ellipses containing the spectrum, finding the intervals that optimize the convergence rate of the Akhiezer iteration requires a generalization of the results of [105] to more complex geometries based on the level curves of $\mathrm{Re}\mathfrak{g}_\Sigma(z)$. How to best approach this is an important and interesting question that we leave for future work.

On the other hand, the adaptivity techniques for symmetric matrices of Section 5.4 can be used for eigenvalue approximation, even when decoupled from the Akhiezer iteration. When the eigenvalues are contained in two disjoint intervals, the variant based on inner products computes both interior and exterior eigenvalues to full precision. In particular, given any symmetric matrix, one could compute interior eigenvalues in the interval $[b, c]$ by choosing an initial guess of $[a, b] \cup [c, d]$. Then, the algorithm would compute two consecutive eigenvalues in $[b, c]$ or leave b and/or c unchanged if there are zero/one eigenvalues in this interval. This requires at most 4 Rayleigh quotient computations, so this provides an efficient method to compute interior eigenvalues to full precision.

Chapter 6

SYLVESTER MATRIX EQUATIONS

6.1 Introduction

This chapter is based on [20]. Code used to generate the plots in this chapter can be found at [21].

In numerical linear algebra, a fundamental problem is computing the solution of the Sylvester matrix equation (1.2), with $\mathbf{X}$ as the unknown. It appears in applications related to model reduction and stability analysis for large-scale dynamical systems [10, 82, 126], eigenvalue assignment for vibrating structures [38], noise reduction in image processing [39], various tasks in control systems and signal processing [28, 72, 103, 161], and the solving of partial differential equations (PDEs) [37, 71, 110, 139]. We consider the case where the *coefficient matrices* $\mathbf{A}$ and $\mathbf{B}$ are diagonalizable with spectra each contained in disjoint regions on or near the real axis.

Iterative methods for solving these equations have been a central focus of research for several decades (see [137] and the references therein), and this has co-evolved alongside the broader development of rational Krylov methods [26, 31, 131] and related iterative solvers that fundamentally involve approximations via rational functions. In settings where $\mathbf{C}$ is low rank and $\mathbf{A}$ and $\mathbf{B}$ are banded, sparse, or otherwise structured so that shifted inverts are inexpensive, rational-based iterative methods can be extraordinarily effective. However, when $\mathbf{A}$ and $\mathbf{B}$ are dense or otherwise costly to invert, such methods will become prohibitively expensive, offering no advantage over the $\mathrm{O}(m^3+n^3)$ direct solver of Bartels and Stewart [23], which is itself too costly in the large-scale setting. A natural alternative one might look for is an iterative method based on polynomial approximation. This avoids the cost of inversion, instead requiring low-rank matrix-matrix products when $\mathbf{C}$ is low rank. Existing methods based on such ideas (e.g., (block) polynomial Krylov methods) incur a cost per iteration that grows cubically with respect to the iteration number due to the orthogonalization step [136], often suffer from storage constraints, as they require storing all previous iterations, and their convergence rates are often too slow to capture the singular value decay of the true solution [94, 126]. Such methods also require residual-monitoring, while the methods that we introduce here do not.

With the limitations of competing methods in mind, we consider a successful method for solving (1.2) when $\mathbf{C}$ is low rank and $\mathbf{A}$ and $\mathbf{B}$ are dense to be one that requires $\mathrm{o}(n^3+m^3)$ arithmetic operations and $\mathrm{o}(nm)$ stored entries. Direct and rational-based solvers typically fail at the former, while standard polynomial-based methods typically fail at the latter.

Here, we introduce two new polynomial-based iterative methods for solving Sylvester matrix equations that achieve both criteria using the Akhiezer polynomials. Because these polynomials have good convergence properties on cut domains, we are able to obtain rates of convergence in our algorithms that are geometric with the degree of the polynomial (and number of iterations). The result is a collection of inverse-free iterative methods with provable, and often explicit, geometric rates of convergence (see Figure 6.2). When $\mathbf{C}$ is low rank, as is often the case in practice, the methods developed here only require low-rank matrix-matrix products and some low-dimensional QR and singular value decompositions. They do

not require storage and orthogonalization against a growing collection of basis vectors. Since we have error bounds, monitoring of a residual or other convergence indicators is also not required. When $\mathbf{X}$ is numerically of low rank, these methods can be used to construct an approximate solution in low-rank form.

A particularly attractive feature of the methods that we introduce here is that their convergence rates are easily determined a priori. That is, denoting the true solution to (1.2) by $\mathbf{X}_*$, the kth iterate $\mathbf{X}_k$ satisfies

$$\|\mathbf{X}_k - \mathbf{X}_*\| \leq C\varrho^{-k}, \tag{6.1}$$

in an appropriate norm for some constants $C > 0$ and $0 < \varrho^{-1} < 1$. Let $\sigma(\mathbf{M})$ denote the set of eigenvalues of a linear operator $\mathbf{M}$, i.e., the spectrum. If $\sigma(\mathbf{A}) \subset [a, 1]$ and $\sigma(\mathbf{B}) \subset [-1, -a]$ for $a \in (0, 1)$, Method 1 (developed in Section 6.2) has convergence rate

$$\varrho^{-1}(a) = \sqrt{\frac{1-a}{1+a}}, \tag{6.2}$$

while Method 2 (developed in Section 6.3) has convergence rate

$$\varrho^{-1}(a) = \frac{1-\sqrt{a}}{1+\sqrt{a}}. \tag{6.3}$$

In [138, Equation 3.1], Simoncini and Druskin give an error bound of the form (6.1) for a polynomial Krylov method (e.g., [137, Algorithm 5]) in the Lyapunov case[1] $\mathbf{B} = -\mathbf{A}^T$, $\mathbf{C} = \mathbf{U}\mathbf{U}^T$ with

$$\varrho^{-1}(a) = \frac{\sqrt{a+1} - \sqrt{2a}}{\sqrt{a+1} + \sqrt{2a}}. \tag{6.4}$$

In Figure 6.1, we plot these convergence rate bounds, as a function of a, as well as the storage required for a test problem. We observe that both of our methods exhibit slower convergence than the polynomial Krylov method, although Method 2 is quite competitive. As we demonstrate below, Method 2 can be comparable to a polynomial Krylov method in runtime, as the iterations of such a method require orthogonalization and are therefore more expensive. On the other hand, run for enough iterations, both of our methods outperform a generic polynomial Krylov method [137, Algorithm 5] in terms of the maximum required storage. Despite having the slowest convergence rate, Method 1 tends to perform the best in this sense.

---

[1] See [98] for more general error bounds. Our methods should also be able to provide error bounds for polynomial Krylov methods, and this will be explored in future work.

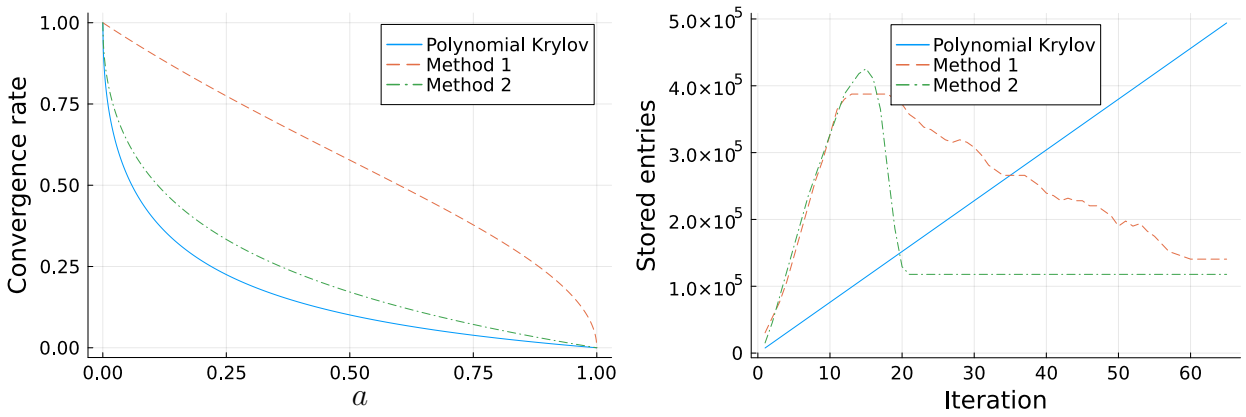


Figure 6.1: Left: Convergence rate bounds (6.4),(6.2),(6.3) for solving (1.2) with a polynomial Krylov method, Method 1, and Method 2, respectively, when $\mathbf{B} = -\mathbf{A}^T$, $\mathbf{C} = \mathbf{U}\mathbf{U}^T$, and $\sigma(\mathbf{A}) \subset [a, 1]$, plotted as a function of a. Right: Number of stored matrix entries required at each iteration of each method applied to the problem of Figure 6.4 described in Section 6.2.2, with $\mathbf{A}, \mathbf{B}$ dense.

6.1.1 An approach based on matrix functions

The two primary classes of iterative methods developed for solving (1.2) are those based on Krylov subspace projections, such as the block polynomial Krylov method, extended Krylov method [136] and the rational Krylov subspace method [25, 60], and those based around directly using rational approximations to construct low-rank solutions, including the cyclic Smith method [126, 133, 140], the low-rank alternating direction implicit (LR-ADI) method [29, 104], and their variants. A distinct idea, first introduced by Roberts in [130] in the context of solving matrix Riccati equations, is based instead on approximating the matrix sign function [30, 57, 86]. This is also the core idea animating Method 1 below. We define the sign function somewhat unconventionally as follows:

Definition 6.1.1. *Let $U_{\mathbf{A}}$ and $U_{\mathbf{B}}$ denote two disjoint sets in $\mathbb{C}$. Then, the function* $\operatorname{sign}(z) = \operatorname{sign}(z; U_{\mathbf{A}}, U_{\mathbf{B}})$ *is defined on $U_{\mathbf{A}} \cup U_{\mathbf{B}}$ as*

$$\operatorname{sign}(z) = \begin{cases} 1, & z \in U_{\mathbf{A}}, \\ -1, & z \in U_{\mathbf{B}}. \end{cases}$$

When the context is clear, we simplify the notation and refer to $\operatorname{sign}(z)$ without explicitly listing the sets it is defined on. If $\sigma(\mathbf{A}) \subset U_{\mathbf{A}}$ and $\sigma(\mathbf{B}) \subset U_{\mathbf{B}}$, then using the definitions for

matrix functions we develop below,

$$\operatorname{sign}\underbrace{\begin{pmatrix}\mathbf{A} & \mathbf{0}\\ \mathbf{C} & \mathbf{B}\end{pmatrix}}_{\mathbf{H}} = \begin{pmatrix}\mathbf{I}_n & \mathbf{0}\\ \mathbf{X} & \mathbf{I}_m\end{pmatrix}\begin{pmatrix}\mathbf{I}_n & \mathbf{0}\\ \mathbf{0} & -\mathbf{I}_m\end{pmatrix}\begin{pmatrix}\mathbf{I}_n & \mathbf{0}\\ -\mathbf{X} & \mathbf{I}_m\end{pmatrix} = \begin{pmatrix}\mathbf{I}_n & \mathbf{0}\\ 2\mathbf{X} & -\mathbf{I}_m\end{pmatrix}. \tag{6.5}$$

Using (6.5), the solution $\mathbf{X}$ of (1.2) can be obtained by computing the matrix $\frac{1}{2}\operatorname{sign}(\mathbf{H})$ and isolating the lower left block [86, Section 2.4]. To get an inverse-free iterative method for computing $\mathbf{X}$ from this, we construct an approximation

$$\operatorname{sign}(z) \approx \sum_{j=0}^{m} \alpha_j p_j(z), \tag{6.6}$$

where $(p_j(z))_{j=0}^{\infty}$ are the Akhiezer polynomials, and are therefore orthogonal with respect to a special inner product. An efficient numerical method for evaluating these polynomials (and computing the coefficients α_j in the expansion) was recently introduced in [18, 19]. In principle, one can simply evaluate (6.6) at $z = \mathbf{H}$ and extract the appropriate block to create an approximate solution to (1.2). Evaluating each $p_j(\mathbf{H})$ directly is expensive since the dimensions of $\mathbf{H}$ are double those of the original problem, and we seek to avoid this. Using the block triangular structure and the three-term recurrence satisfied by the Akhiezer polynomials, we develop a recursion that describes the evolution of the lower left block as j grows, so that in most cases[2], the resulting method only incurs a per-iteration computational cost on the same order as the cost of matrix-vector products with $\mathbf{A}$ and $\mathbf{B}$. Under our convergence heuristics, $\mathrm{O}\left(\log(m+n)\right)$ iterations are required to satisfy an error tolerance, and therefore the overall cost of the low-rank variation of the method when $\mathbf{A}$ and $\mathbf{B}$ are dense is $\mathrm{O}(m^2 \log m + n^2 \log n)$.

The second method that we introduce functions largely in the same manner. Rather than passing to a sign function, we directly approximate the inverse of the Sylvester operator $S_{\mathbf{A},\mathbf{B}}\mathbf{Y} = \mathbf{YA} - \mathbf{BY}$ via

$$\frac{1}{z} \approx \sum_{j=0}^{m} \hat{\alpha}_j p_j(z),$$

where $(p_j(z))_{j=0}^{\infty}$ are again the Akhiezer polynomials. Again, the resulting method incurs a per-iteration computational cost on the same order as the cost of matrix-vector products with $\mathbf{A}$ and $\mathbf{B}$ in most cases, requiring $\mathrm{O}\left(\log(m+n)\right)$ iterations to satisfy an error tolerance with an overall cost of $\mathrm{O}(m^2 \log m + n^2 \log n)$ when $\mathbf{A}$ and $\mathbf{B}$ are dense. Both methods that we introduce allow for compression at each step. Empirically, this, in combination with the three-term Akhiezer polynomial recurrence, ensures that $\mathrm{o}(nm)$ entries need to be stored when $\mathbf{C}$ is low rank. These methods can also be applied when the spectra of $\mathbf{A}$ and $\mathbf{B}$ lie in or near an arbitrary number of real intervals, as in Chapter 5.

---

[2]The cost of the compression step dominates if the cost of matrix-vector products is $\mathrm{o}(m+n)$.

Inverse-free iterative solvers can be advantageous in settings where the coefficient matrices are dense, where fast matrix-vector product routines for these matrices are available, and in settings where coefficient matrices are not directly accessible and only their action can be observed. We view the Akhiezer-based method presented here as an analog to the Chebyshev iteration for solving linear systems. The Chebyshev iteration can be advantageous over the conjugate gradient method or GMRES since it only requires short recurrences and avoids orthogonalization steps, and the Akhiezer-based approach is advantageous over polynomial Krylov methods in the same way. Our hope is that the availability of a new inverse-free method will prompt further developments, including investigations into other use-cases and the analysis and development of analogous polynomial families for more general domains. As we show in Section 6.5, Method 1 also supplies a new approach for related tasks, including the evaluation of the Fréchet derivative of matrix functions and the solving of other matrix equations. There is also a natural connection to the general evaluation of functions of matrices, which is thoroughly discussed in Chapter 5.

Figure 6.2 compares the errors in approximating the sign function via Akhiezer polynomials with the approximation error achieved via optimal rational approximation [6, Chapter 9] (see also [115, 154]) on various subsets of $[-1, 1]$. While we cannot expect to achieve the optimal rational convergence rate, the convergence rate is fast enough to yield a competitive algorithm in regimes where inversion is costly. Comparisons with other inverse-free methods are shown in Section 6.4. Furthermore, this convergence rate can be determined a priori using only the intervals constituting the cut domain. It is governed by the level curves of the function $e^{\mathrm{Re}\, \mathfrak{g}_\Sigma(z)}$, where $\mathfrak{g}_\Sigma(z)$ is the classical Green's function with pole at infinity defined in Section 6.2.3 [68, 162]. The convergence rate of an approximation to a function f is governed by the largest value of c such that f can be analytically extended to the level set $\mathrm{Re}\mathfrak{g}_\Sigma(z) < c$ [162].

For the sign function, this maximal level set will correspond to a value c^* such that the level curves $\mathrm{Re}\, \mathfrak{g}_\Sigma(z) = c^*$ around each interval intersect. Thus, as c crosses c^*, we see a bifurcation of a connected level curve to a disconnected level curve, or vice versa. In Figure 6.3, we demonstrate this. We describe an inexpensive method to compute this intersection point, and thus the rate of convergence of Method 1, in Section 6.2.3. Plots of these level curves for more general domains can be found in [63].

6.1.2 Related work

As stated above, many polynomial-based methods suffer from growing per-iteration cost, heavy storage requirements, and necessary residual computation which, in conjunction with the slower convergence rate of such methods relative to rational-based methods, make large-scale problems intractable. Considerable effort has focused on improving such methods on all three fronts [41, 94, 96, 121, 122]. In particular, the method of [121] both greatly reduces the per-iteration cost and storage requirements of polynomial-based Krylov subspace methods while accelerating convergence. Despite the fact that the convergence of polynomial-based Krylov methods is slow compared to that of rational Krylov methods, the savings induced by

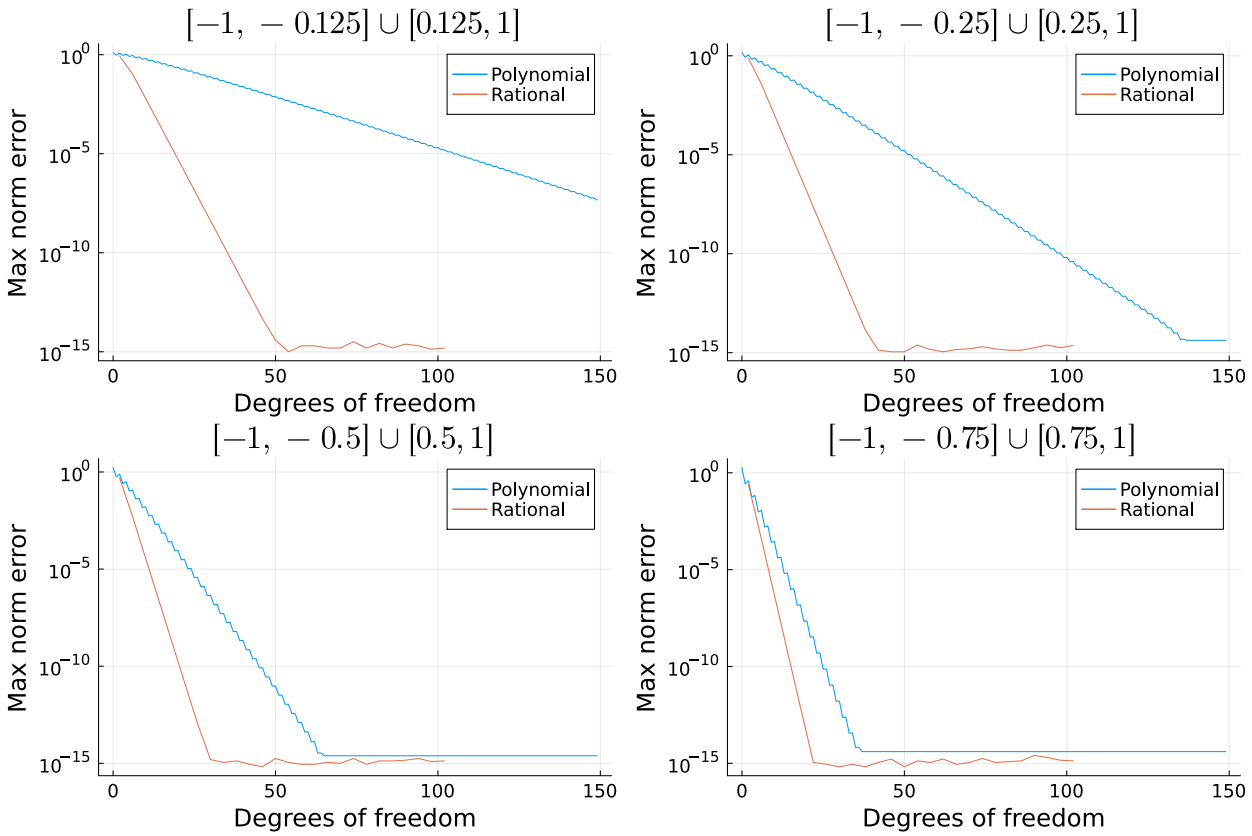


Figure 6.2: Error plots of Akhiezer polynomial series and optimal Zolotarev rational approximations to the sign function on different cut domains.

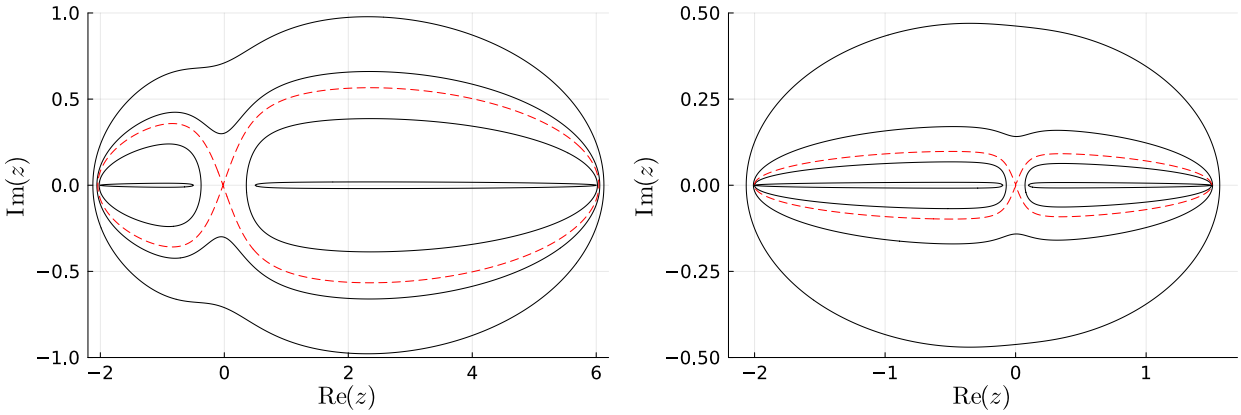


Figure 6.3: Generalized Bernstein ellipses for the cut domains $[-2, -0.1] \cup [0.1, 1.5]$ (left) and $[-2, -0.5] \cup [0.5, 6]$ (right). These are level curves of $e^{\operatorname{Re} \mathfrak{g}_\Sigma(z)}$ from Lemma 5.3.6. The red dashed line is the level curve where the curves around each interval first intersect, $\operatorname{Re} \mathfrak{g}_\Sigma(z) = c^*$, and this governs the convergence rate of the Akhiezer polynomial approximation to the sign function on these domains.

solving the smaller sketched problems can make up the difference because of the per-iteration savings. We suspect that the use of the Akhiezer polynomials in concert with a sketched Krylov subspace method might lead to faster runtimes for such schemes, but we leave this for future work.

Owing to the fact that $\operatorname{sign}(\mathbf{H}) = \mathbf{H}(\mathbf{H}^2)^{-1/2}$, iterative methods for computing matrix square roots can be used to evaluate the matrix sign function. While many of these algorithms make use of inversion, the Newton–Schulz iteration is an exception. More precisely, the Newton–Schulz iteration to compute the sign function is simple:

$$\mathbf{H}_{k+1} = \frac{1}{2}\mathbf{H}_k \left(3\mathbf{I} - \mathbf{H}_k^2\right).$$

This iteration converges quadratically, but it requires that $\|\mathbf{I} - \mathbf{H}^2\| < 1$ in a subordinate matrix norm [86, Section 5.3]. By contrast[3], the methods presented here can be applied whenever the eigenvalues of $\mathbf{H}$ are close to known intervals on the real axis. We discuss other connections with algorithms for the matrix square root in Section 6.5.3 and refer to [86] for a thorough survey.

[3]Newton–Schulz can be modified to converge globally, but this requires $\mathbf{H}$ to be Hermitian [44].

6.1.3 Organization

The rest of this chapter is organized as follows. In Section 6.2, we present Method 1, extending the Akhiezer iteration to an inverse-free iterative method for solving Sylvester matrix equations via the matrix sign function and analyze its complexity and convergence properties. In Section 6.3, we present Method 2, using the Akhiezer iteration to directly approximate the inverse of the Sylvester operator and again analyze its complexity and convergence properties. In Section 6.4, we apply the methods presented here to solve integral equations and two-dimensional PDEs. In Section 6.5, we discuss other related applications.

6.2 Method 1: the matrix sign function

We now turn to solving the Sylvester matrix equation (1.2). We assume that the spectrum of $\mathbf{A}$ is contained in $U_{\mathbf{A}}$ and that the spectrum of $\mathbf{B}$ is contained in $U_{\mathbf{B}}$. Suppose $\Xi = \alpha + \mathrm{e}^{\mathrm{i}\theta}\mathbb{R}$ is chosen such that the halfplanes $H_{\mathbf{A}}$ and $H_{\mathbf{B}}$ separated by Ξ contain $U_{\mathbf{A}}$ and $U_{\mathbf{B}}$, respectively. Then, define

$$\operatorname{sign} : \mathbb{C} \setminus \Xi \to \mathbb{C}, \quad \operatorname{sign}(z) = \begin{cases} 1, & z \in H_{\mathbf{A}}, \\ -1, & z \in H_{\mathbf{B}}. \end{cases} \tag{6.7}$$

The choice of Ξ will affect the convergence rate estimates that appear later in this section.

The solution of (1.2) can be obtained as a subblock of the matrix sign function of the block matrix $\mathbf{H}$ in (6.5). This follows from the factorizations

$$\begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C}+\mathbf{B}\mathbf{X} & \mathbf{B} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{B} \end{pmatrix} \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{X} & \mathbf{I}_m \end{pmatrix}, \quad \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{X}\mathbf{A} & \mathbf{B} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{X} & \mathbf{I}_m \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix},$$

so (1.2) is equivalent to the equation

$$\mathbf{H} := \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{B} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{X} & \mathbf{I}_m \end{pmatrix} \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{pmatrix} \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ -\mathbf{X} & \mathbf{I}_m \end{pmatrix}. \tag{6.8}$$

When $\sigma(\mathbf{A}) \subset [a_2, b_2] \subset \mathbb{R}$, $\sigma(\mathbf{B}) \subset [a_1, b_1] \subset \mathbb{R}$, and $b_1 < a_2$, the Akhiezer iteration for $\Sigma = [a_1, b_1] \cup [a_2, b_2]$, as presented in Algorithm 3, can be directly applied to compute $\operatorname{sign}(\mathbf{H})$ (with $\mathbf{b} = \mathbf{I}$), and thus solve the Sylvester equation (1.2). Note that despite its discontinuous nature, the sign function can always be defined so that it is analytic in a (disconnected) region containing Σ. We discuss this application and efficiency improvements in the following subsections.

This construction directly yields the following solution formula for (1.2), where the convergence of the infinite sum follows from Lemma 5.2.2 below:

Theorem 6.2.1. *Define the sign function as in* (6.7) *and let* $\sigma(\mathbf{A}), \sigma(\mathbf{B})$ *be contained in the interior of* $\Sigma = \bigcup_{j=1}^{g+1}[a_j, b_j] \subset \mathbb{R}$ *where* Ξ *does not intersect* Σ*. Let* w *be a weight function of the form* (3.5) *and denote the corresponding orthonormal polynomials by* $(p_j(x))_{j=0}^{\infty}$*. Then,*

the solution of the Sylvester equation (1.2) *is given by*

$$\mathbf{X} = \sum_{j=0}^{\infty} \alpha_j p_j(\mathbf{H})_{n+1:n+m,1:n}, \quad \mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{B} \end{pmatrix}, \quad \alpha_j = \int_{\Sigma} \operatorname{sign}(x) p_j(x) w(x) \mathrm{d}x.$$

Remark 6.2.2. *The assumption that* $\sigma(\mathbf{A}), \sigma(\mathbf{B}) \subset \Sigma$ *in Theorem 6.2.1 is unnecessary. In the notation of Chapter 5, this assumption may be replaced with the requirement that* $\nu(z_*; \mathbf{H}) < 0$*, where* z_* *is any point on the level curve* $\Gamma_\varrho(\Sigma)$*, interior to which the sign function is analytic. See Lemma 5.3.6.*

6.2.1 Isolating the lower left block

Since only the lower left block of $\operatorname{sign}(\mathbf{H})$ is required to solve the Sylvester equation (1.2), we can reduce the necessary arithmetic operations using the following lemma.

Lemma 6.2.3. *Given a sequence of orthonormal polynomials* $(p_j(z))_{j=0}^{\infty}$ *with three-term recurrence* (2.5)*, the lower left block of* $p_j(\mathbf{H})$ *where*

$$\mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \mathbf{B} \end{pmatrix},$$

is given by

$$\mathbf{C} p_j(\mathbf{A}) + \mathbf{G}_j,$$

where $\mathbf{G}_j$ *satisfies*

$$\begin{aligned} \mathbf{G}_0 &= -\mathbf{C}, \\ \mathbf{G}_1 &= \frac{1}{\check{b}_0}(\mathbf{G}_0 \mathbf{A} + (\check{a}_0 + 1)\mathbf{C}), \\ \mathbf{G}_j &= \frac{1}{\check{b}_{j-1}} \left(\mathbf{G}_{j-1}\mathbf{A} + p_{j-1}(\mathbf{B})\mathbf{C} - \check{a}_{j-1}\mathbf{G}_{j-1} - \check{b}_{j-2}\mathbf{G}_{j-2} \right), \quad j \geq 2. \end{aligned} \tag{6.9}$$

Proof. See Appendix A.3. □

Lemma 6.2.3 allows the Akhiezer iteration applied to Sylvester equations to be implemented such that full block $(n+m) \times (n+m)$ matrix-matrix multiplications are no longer needed; however, the matrix polynomials $p_j(\mathbf{A})$ and $p_j(\mathbf{B})$ are both still required to compute the desired lower left block. Thus, recurrences for these matrix polynomials and the approximate solution can be run in parallel, effectively decoupling the large matrix-matrix product into five smaller ones. A Sylvester equation solver based on this decoupling can be implemented as in Algorithm 5.

Algorithm 5: Decoupled $\mathbf{XA} - \mathbf{BX} = \mathbf{C}$ solver via the Akhiezer iteration

Input: Matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and the Akhiezer data (functions to compute recurrence coefficients $\check{a}_k, \check{b}_k$ and p_k-series coefficients α_k for the sign function).

Set $\mathbf{X}_0 = \mathbf{0}$.

for *k=0,1,...* **do**

  if *k=0* **then**

    Set $p_0(\mathbf{A}) = \mathbf{I}$.

    Set $p_0(\mathbf{B}) = \mathbf{I}$.

    Set $\mathbf{G}_0 = -\mathbf{C}$.

  else if *k=1* **then**

    Set $p_1(\mathbf{A}) = \frac{1}{\check{b}_0}(p_0(\mathbf{A})\mathbf{A} - \check{a}_0 p_0(\mathbf{A}))$.

    Set $p_1(\mathbf{B}) = \frac{1}{\check{b}_0}(\mathbf{B}p_0(\mathbf{B}) - \check{a}_0 p_0(\mathbf{B}))$.

    Set $\mathbf{G}_1 = \frac{1}{\check{b}_0}(\mathbf{G}_0\mathbf{A} + (\check{a}_0 + 1)\mathbf{C})$.

  else

    Set $p_k(\mathbf{A}) = \frac{1}{\check{b}_{k-1}}(p_{k-1}(\mathbf{A})\mathbf{A} - \check{a}_{k-1}p_{k-1}(\mathbf{A}) - \check{b}_{k-2}p_{k-2}(\mathbf{A}))$.

    Set $p_k(\mathbf{B}) = \frac{1}{\check{b}_{k-1}}(\mathbf{B}p_{k-1}(\mathbf{B}) - \check{a}_{k-1}p_{k-1}(\mathbf{B}) - \check{b}_{k-2}p_{k-2}(\mathbf{B}))$.

    Set $\mathbf{G}_k = \frac{1}{\check{b}_{k-1}}\big(\mathbf{G}_{k-1}\mathbf{A} + p_{k-1}(\mathbf{B})\mathbf{C} - \check{a}_{k-1}\mathbf{G}_{k-1} - \check{b}_{k-2}\mathbf{G}_{k-2}\big)$.

  end

  Set $\mathbf{X}_{k+1} = \mathbf{X}_k + \frac{\alpha_k}{2}(\mathbf{C}p_k(\mathbf{A}) + \mathbf{G}_k)$.

  if converged **then**

    return $\mathbf{X}_{k+1}$.

  end

end

6.2.2 Compression and low-rank structure

In the case where the data matrix $\mathbf{C}$ in (1.2) is low rank, Algorithm 5 is suboptimal in that it requires full matrix-matrix products. Moreover, it is known that when the rank of $\mathbf{C}$ is bounded above by a small number r and $\mathbf{A}$ and $\mathbf{B}$ are normal[4] with well-separated spectral sets, then $\mathbf{X}$ is well-approximated by a low-rank matrix [27]. We say that such a matrix is *numerically* of low rank. Specifically, we say that $\mathbf{X}$ is numerically of rank k with respect to the tolerance $0 < \epsilon < 1$ if the $(k+1)$st singular value of $\mathbf{X}$ is bounded above by $\|\mathbf{X}\|_F \epsilon$. Throughout this chapter, we take $\epsilon = 10^{-14}$. Rather than constructing $\mathbf{X}_k$ outright, it is preferable to construct low-rank factors $\mathbf{W}_k \mathbf{Z}_k = \mathbf{X}_k$.

Here, we suppose we are given a low-rank factorization of $\mathbf{C}$ in (1.2), and so consider the equation

$$\mathbf{XA} - \mathbf{BX} = \mathbf{UV}, \tag{6.10}$$

where $\mathbf{U} \in \mathbb{C}^{m \times r}$ and $\mathbf{V} \in \mathbb{C}^{r \times n}$ are given with r much smaller than m and n. To incorporate the low-rank structure into our iteration, we note that the recurrence (2.5) can be left- and right-multiplied, respectively, giving

$$\begin{aligned}
\mathbf{V}p_0(\mathbf{A}) &= \mathbf{V}, \\
\mathbf{V}p_1(\mathbf{A}) &= \frac{1}{\check{b}_0}(\mathbf{V}p_0(\mathbf{A})\mathbf{A} - \check{a}_0 \mathbf{V}p_0(\mathbf{A})), \\
\mathbf{V}p_k(\mathbf{A}) &= \frac{1}{\check{b}_{k-1}}(\mathbf{V}p_{k-1}(\mathbf{A})\mathbf{A} - \check{a}_{k-1}\mathbf{V}p_{k-1}(\mathbf{A}) - \check{b}_{k-2}\mathbf{V}p_{k-2}(\mathbf{A})), \quad k \geq 2,
\end{aligned}$$

and

$$\begin{aligned}
p_0(\mathbf{B})\mathbf{U} &= \mathbf{U}, \\
p_1(\mathbf{B})\mathbf{U} &= \frac{1}{\check{b}_0}(\mathbf{B}p_0(\mathbf{B})\mathbf{U} - \check{a}_0 p_0(\mathbf{B})\mathbf{U}), \\
p_k(\mathbf{B})\mathbf{U} &= \frac{1}{\check{b}_{k-1}}(\mathbf{B}p_{k-1}(\mathbf{B})\mathbf{U} - \check{a}_{k-1}p_{k-1}(\mathbf{B})\mathbf{U} - \check{b}_{k-2}p_{k-2}(\mathbf{B})\mathbf{U}), \quad k \geq 2.
\end{aligned}$$

By computing with only these recurrence formulae, we circumvent the need to compute the full diagonal blocks of $p_j(\mathbf{H})$, greatly reducing the size of the needed matrix-matrix products

[4]Generalizations for the nonnormal case are also known [27].

when r is small. Letting $\mathbf{G}_j = \mathbf{J}_j\mathbf{K}_j$, we write the recurrence (6.9) in block low-rank form as

$$\begin{aligned}
\mathbf{J}_0\mathbf{K}_0 &= \mathbf{G}_0 = -\mathbf{U}\mathbf{V}, \\
\mathbf{J}_1\mathbf{K}_1 &= \mathbf{G}_1 = \frac{1}{\check{b}_0}\begin{pmatrix}\mathbf{J}_0 & (\check{a}_0+1)\mathbf{U}\end{pmatrix}\begin{pmatrix}\mathbf{K}_0\mathbf{A} \\ \mathbf{V}\end{pmatrix}, \\
\mathbf{J}_j\mathbf{K}_j &= \mathbf{G}_j = \frac{1}{\check{b}_{j-1}}\begin{pmatrix}\mathbf{J}_{j-1} & p_{j-1}(\mathbf{B})\mathbf{U} & -\check{a}_{j-1}\mathbf{J}_{j-1} & -\check{b}_{j-2}\mathbf{J}_{j-2}\end{pmatrix}\begin{pmatrix}\mathbf{K}_{j-1}\mathbf{A} \\ \mathbf{V} \\ \mathbf{K}_{j-1} \\ \mathbf{K}_{j-2}\end{pmatrix}, \quad j \geq 2.
\end{aligned}$$

Then, letting $\mathbf{X}_k = \mathbf{W}_k\mathbf{Z}_k$, the update step from Algorithm 5 can be written as

$$\mathbf{W}_{k+1}\mathbf{Z}_{k+1} = \mathbf{X}_{k+1} = \begin{pmatrix}\mathbf{W}_k & \frac{\alpha_k}{2}\mathbf{U} & \frac{\alpha_k}{2}\mathbf{J}_k\end{pmatrix}\begin{pmatrix}\mathbf{Z}_k \\ \mathbf{V}p_k(\mathbf{A}) \\ \mathbf{K}_k\end{pmatrix}.$$

These relations allow Algorithm 5 to be decoupled such that two matrices, $\mathbf{W}_k$ and $\mathbf{Z}_k$, are updated separately and the approximate solution is the product of the two. However, their direct use is hampered by the fact that the number of rows or columns grows rapidly with k. To address this, we compress these matrices to their numerical rank as needed. A simple compression method based on the QR/LQ and singular value decompositions (SVD) is given in Algorithm 6.

Algorithm 6: Simple compression of $(\mathbf{J}, \mathbf{K})$

`COMPRESS`$(\mathbf{J}, \mathbf{K}, \epsilon)$
Input: Matrices $\mathbf{J} \in \mathbb{C}^{m\times n}$, $\mathbf{K} \in \mathbb{C}^{n\times \ell}$ and tolerance ϵ.
Run QR on $\mathbf{J}$ to get $\mathbf{Q_J R} = \mathbf{J}$.
Run LQ on $\mathbf{K}$ to get $\mathbf{L Q_K} = \mathbf{K}$.
Run SVD on $\mathbf{RL}$ to get $\mathbf{U\Sigma V} = \mathbf{RL}$.
Truncate $\mathbf{\Sigma}$ so that all discarded singular values satisfy $\sigma_j^2 / \sum_j \sigma_j^2 < \epsilon$, leaving $\tilde{\mathbf{\Sigma}} \in \mathbb{R}^{k\times k}$.
Truncate rows or columns of $\mathbf{U}$ and $\mathbf{V}$ as appropriate, leaving $\tilde{\mathbf{U}} \in \mathbb{C}^{m\times k}$ and $\tilde{\mathbf{V}} \in \mathbb{C}^{k\times \ell}$.
Return $\tilde{\mathbf{J}} = \mathbf{Q_J}\tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}^{1/2}$, $\tilde{\mathbf{K}} = \tilde{\mathbf{\Sigma}}^{1/2}\tilde{\mathbf{V}}\mathbf{Q_K}$.

Given a method that compresses the matrices $\mathbf{J}_k\mathbf{K}_k$ and $\mathbf{W}_k\mathbf{Z}_k$ to match their numerical rank, the above recurrence relations allow Algorithm 5 to be rewritten so that only low-rank matrix-matrix products and compression steps are necessary to compute a solution. An implementation of this is given as Algorithm 7.

This algorithm chooses compression tolerances in a manner that depends on the iteration

Algorithm 7: Low-rank $\mathbf{XA} - \mathbf{BX} = \mathbf{UV}$ solver using the Akhiezer iteration for the sign function

Input: Matrices $\mathbf{A}, \mathbf{B}, \mathbf{U}, \mathbf{V}$, the Akhiezer data (functions to compute recurrence coefficients $\check{a}_k, \check{b}_k$ and p_k-series coefficients α_k for the sign function), computed decay rate ϱ and constant c as in Section 6.2.3, and compression tolerance ϵ.

Initialize $\mathbf{W}_0$ and $\mathbf{Z}_0$ to be empty.

for *k=0,1,...* **do**

  if *k=0* **then**

    Set $\mathbf{V}p_0(\mathbf{A}) = \mathbf{V}$.

    Set $p_0(\mathbf{B})\mathbf{U} = \mathbf{U}$.

    Set $\mathbf{J}_0 = -\mathbf{U}$.

    Set $\mathbf{K}_0 = \mathbf{V}$.

  else if *k=1* **then**

    Set $\mathbf{V}p_1(\mathbf{A}) = \frac{1}{\check{b}_0}(\mathbf{V}p_0(\mathbf{A})\mathbf{A} - \check{a}_0\mathbf{V}p_0(\mathbf{A}))$.

    Set $p_1(\mathbf{B})\mathbf{U} = \frac{1}{\check{b}_0}(\mathbf{B}p_0(\mathbf{B})\mathbf{U} - \check{a}_0 p_0(\mathbf{B})\mathbf{U})$.

    Set $\mathbf{J}_1 = \frac{1}{\check{b}_0}\begin{pmatrix}\mathbf{J}_0 & (\check{a}_0 + 1)\mathbf{U}\end{pmatrix}$.

    Set $\mathbf{K}_1 = \begin{pmatrix}\mathbf{K}_0\mathbf{A} \\ \mathbf{V}\end{pmatrix}$.

  else

    Set $\mathbf{V}p_k(\mathbf{A}) = \frac{1}{\check{b}_{k-1}}(\mathbf{V}p_{k-1}(\mathbf{A})\mathbf{A} - \check{a}_{k-1}\mathbf{V}p_{k-1}(\mathbf{A}) - \check{b}_{k-2}\mathbf{V}p_{k-2}(\mathbf{A}))$.

    Set $p_k(\mathbf{B})\mathbf{U} = \frac{1}{\check{b}_{k-1}}(\mathbf{B}p_{k-1}(\mathbf{B})\mathbf{U} - \check{a}_{k-1}p_{k-1}(\mathbf{B})\mathbf{U} - \check{b}_{k-2}p_{k-2}(\mathbf{B})\mathbf{U})$.

    Set $\mathbf{J}_k = \frac{1}{\check{b}_{k-1}}\begin{pmatrix}\mathbf{J}_{k-1} & p_{k-1}(\mathbf{B})\mathbf{U} & -\check{a}_{k-1}\mathbf{J}_{k-1} & -\check{b}_{k-2}\mathbf{J}_{k-2}\end{pmatrix}$.

    Set $\mathbf{K}_k = \begin{pmatrix}\mathbf{K}_{k-1}\mathbf{A} \\ \mathbf{V} \\ \mathbf{K}_{k-1} \\ \mathbf{K}_{k-2}\end{pmatrix}$.

  end

  Set $\mathbf{J}_k, \mathbf{K}_k = \texttt{COMPRESS}(\mathbf{J}_k, \mathbf{K}_k, \frac{\varrho^k\epsilon}{c})$.

  Set $\mathbf{W}_{k+1} = \begin{pmatrix}\mathbf{W}_k & \frac{\alpha_k}{2}\mathbf{U} & \frac{\alpha_k}{2}\mathbf{J}_k\end{pmatrix}$.

  Set $\mathbf{Z}_{k+1} = \begin{pmatrix}\mathbf{Z}_k \\ \mathbf{V}p_k(\mathbf{A}) \\ \mathbf{K}_k\end{pmatrix}$.

  Set $\mathbf{W}_{k+1}, \mathbf{Z}_{k+1} = \texttt{COMPRESS}\,(\mathbf{W}_{k+1}, \mathbf{Z}_{k+1}, \epsilon)$.

  if converged **then**

    return $\mathbf{W}_{k+1}, \mathbf{Z}_{k+1}$.

  end

end

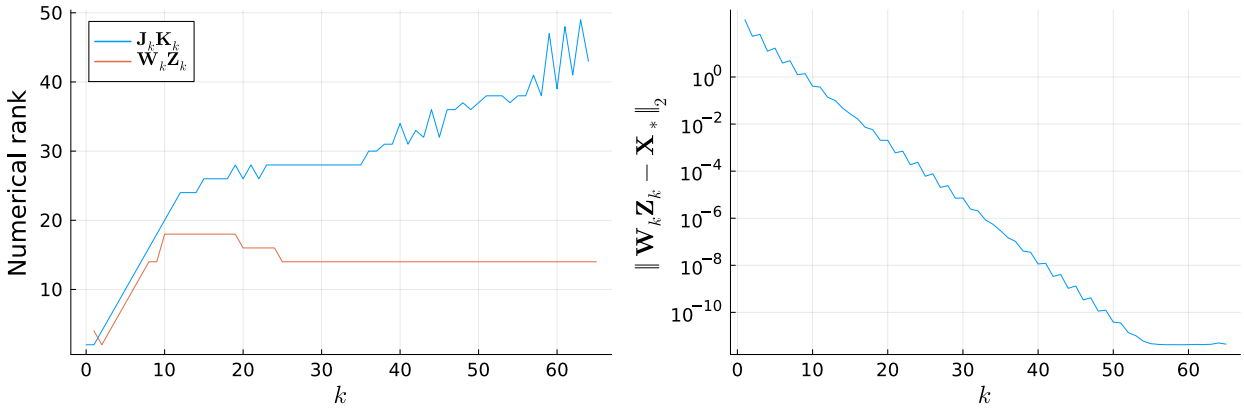


Figure 6.4: Numerical rank of the iterates of Algorithm 7 with unweighted compression (left) and error of the approximation at each iteration against the true solution $\mathbf{X}_*$ of (6.10), computed via `sylvester`, the built-in `Julia` implementation of the Bartels–Stewart algorithm [23] (right). Here, $\mathbf{A} \in \mathbb{R}^{1000\times1000}$, $\mathbf{B} \in \mathbb{R}^{900\times900}$, $\mathbf{UV}$ is rank 2, and the numerical solution appears to have numerical rank 14.

step which we now motivate. Since $\mathbf{X}$ will be numerically of low rank, we expect the approximate solution $\mathbf{W}_k\mathbf{Z}_k$ of Algorithm 7 to be numerically of low rank for all k. However, we have no reason to expect the lower left block of $p_j(\mathbf{H})$ to be numerically of low rank for large j, meaning that as the iteration count increases, a low-rank object is being constructed out of increasingly higher-rank objects. In Figure 6.4, we plot the numerical ranks of both the component pieces $\mathbf{J}_k\mathbf{K}_k$ and of the approximate solution $\mathbf{W}_k\mathbf{Z}_k$ for an example problem where $\mathbf{A} \in \mathbb{R}^{1000\times1000}$, $\sigma(\mathbf{A}) \subset [2,3]$, $\mathbf{B} \in \mathbb{R}^{900\times900}$, $\sigma(\mathbf{B}) \subset [-1.8,-0.5]$, $\mathbf{U} \in \mathbb{R}^{900\times2}$, and $\mathbf{V} \in \mathbb{R}^{2\times1000}$. The numerical rank of $\mathbf{J}_k\mathbf{K}_k$ appears to grow, while the numerical rank of $\mathbf{W}_k\mathbf{Z}_k$ converges to that of the true solution. The reason for this is that $\mathbf{J}_k\mathbf{K}_k$ is always scaled by a p_j-series coefficient α_k when $\mathbf{W}_k\mathbf{Z}_k$ is updated, but these coefficients decay exponentially as k increases [19, Lemma 4.10]. Due to this decay and the fact that $\mathbf{J}_k\mathbf{K}_k$ is used to generate $\mathbf{J}_{k+1}\mathbf{K}_{k+1}$, one may consider enlarging the tolerance (as in Algorithm 6) in the compression of $\mathbf{J}_k\mathbf{K}_k$ by the decay rate of $|\alpha_k|$. See Section 6.2.3 for a full discussion of this. In Figure 6.5, we plot the numerical rank of $\mathbf{J}_k\mathbf{K}_k$ and $\mathbf{W}_k\mathbf{Z}_k$ when the compression of $\mathbf{J}_k\mathbf{K}_k$ is weighted by the decay rate. We observe that the rank of $\mathbf{J}_k\mathbf{K}_k$ now decays as the rank of $\mathbf{W}_k\mathbf{Z}_k$ converges to that of the true solution, but the convergence rate of the iteration is not noticeably affected.

6.2.3 Parameter tuning and heuristics

To determine the convergence rate ϱ^{-1} when f is the sign function, consider $\Sigma = [a_1, b_1] \cup [a_2, b_2]$, $b_1 < a_2$. The largest value of ϱ such that $f(z) = \operatorname{sign}(z)$ can be defined so that it

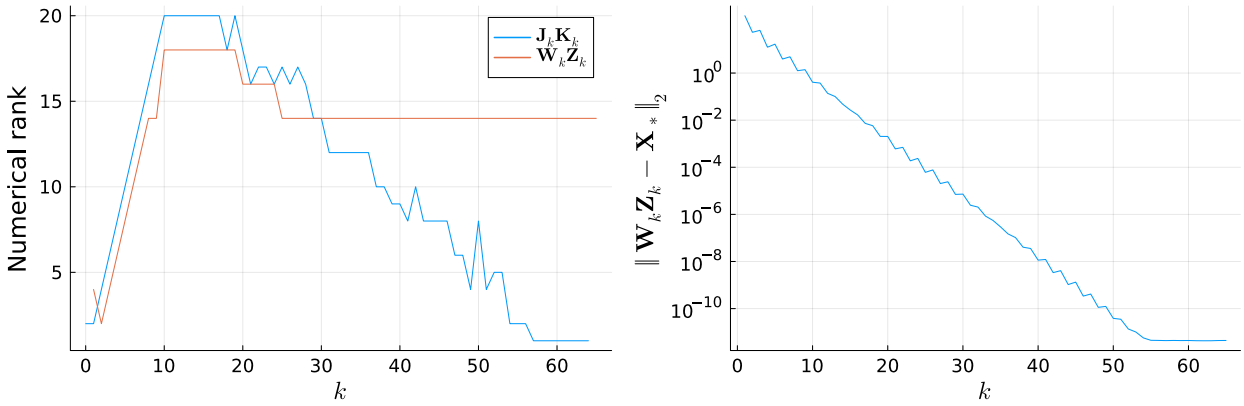


Figure 6.5: Numerical rank of the iterates of Algorithm 7 with weighted compression (left) and error of the approximation at each iteration against the true solution $\mathbf{X}_*$ (right). The problem is the same as that of Figure 6.4, and only the tolerance for the compression of $\mathbf{J}_k\mathbf{K}_k$ was changed.

is analytic in B_ϱ will correspond to the value of ϱ where the disjoint level curves around $[a_1, b_1]$ and $[a_2, b_2]$ first intersect, as ϱ increases. This intersection point is precisely the unique local maximum of $\mathrm{Reg}_\Sigma(z)$ for $z \in [b_1, a_2]$. The location of this intersection follows from [19, Appendix A] and is given by[5]

$$z^* = \frac{\int_{b_1}^{a_2} \frac{z\mathrm{d}z}{\sqrt{z-a_1}\sqrt{z-b_1}\sqrt{z-a_2}\sqrt{z-b_2}}}{\int_{b_1}^{a_2} \frac{\mathrm{d}z}{\sqrt{z-a_1}\sqrt{z-b_1}\sqrt{z-a_2}\sqrt{z-b_2}}}.$$

Given the intersection point z^* of the level curves, we have the following bound on the coefficients of a p_j-series expansion of the sign function:

$$|\alpha_j| = \left|\langle \mathrm{sign}, p_j\rangle_{L^2_w(\Sigma)}\right| \leq C\varrho^{-j}, \quad \varrho = \mathrm{e}^{\mathrm{Reg}_\Sigma(z^*)},$$

for some constant $C > 0$ that cannot in general be computed.

Remark 6.2.4. *In the symmetric case $\Sigma = [-b, -a] \cup [a, b]$, $0 < a < b$, we have that $z^* = 0$ and [65]*

$$\varrho = \sqrt{\frac{\frac{b}{a}+1}{\frac{b}{a}-1}}.$$

We use this bound to inform a heuristic for the weighted compression described in Sec-

[5] A formula for z^* may also be obtained by differentiating the formulae in [19, Section 3.1 and 4.1.1]; however, given the ease of computing $\mathfrak{g}_\Sigma$ via the formula in [19, Section 4.1.1], we find that it is fastest and easiest to find the intersection point z^* via a numerical method such as the golden-section search.

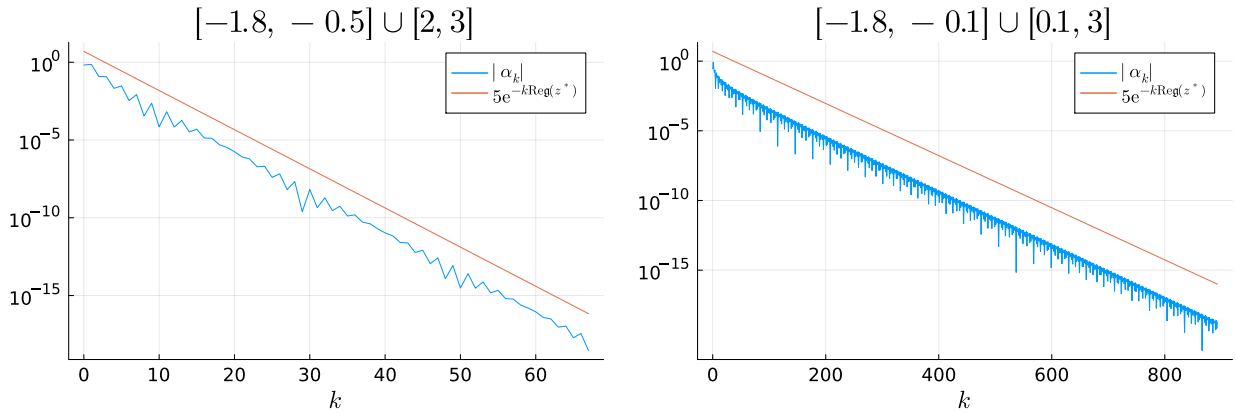


Figure 6.6: Modulus of the coefficients of an Akhiezer polynomial series approximation to the sign function on $[-1.8, -0.5] \cup [2, 3]$ (left) and $[-1.8, -0.1] \cup [0.1, 3]$ (right) plotted against the convergence rate heuristic with $c = 5$ until the heuristic is smaller than 10^{-16}.

tion 6.2.2. We find that the weighted compression step tends to behave well when the tolerance is simply multiplied by $\frac{\varrho^k}{c}$ at iteration k where ϱ is computed a priori and c is a chosen constant taken in place of C. Since the coefficients α_j are computed a priori, one may simply select c such that $|\alpha_j| \leq c\varrho^{-j}$ by plotting or other means; however, we find that it typically suffices to take $c = 5$. One can always plot the series coefficients if uncertain. See Figure 6.6 for plots of the coefficients against the heuristic with $c = 5$.

6.2.4 Convergence and stopping criteria

In this subsection, we present a convergence analysis that is largely a simplified version of that of [19, Section 4] under the additional assumption that the eigenvalues of the matrix lie in the interior of Σ. Some results are reproduced directly without proof. We begin with the following definition to simplify the analysis:

Definition 6.2.5. *We say that a matrix is generic if it is diagonalizable and none of its eigenvalues lie at endpoints of* Σ.

See [19, Section 4.1.1] for remarks on how the analysis can be extended to nongeneric matrices.

Theorem 6.2.6. *Under the assumptions of Lemma 5.3.6, suppose that* $\mathbf{M}$ *is generic and* $\sigma(\mathbf{M}) \subset \Sigma$. *Then, there exists a constant* $C' > 0$, *independent of* f, *such that the approximation* (5.4) *satisfies*

$$\|f(\mathbf{M}) - \mathbf{F}_k\|_2 \leq C' M \|\mathbf{V}\|_2 \left\|\mathbf{V}^{-1}\right\|_2 \frac{\varrho^{-k}}{1 - \varrho^{-1}},$$

where $\mathbf{M}$ *is diagonalized as* $\mathbf{M} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$.

Proof. By the triangle inequality,

$$\begin{aligned}\|f(\mathbf{M}) - \mathbf{F}_k\|_2 &= \left\|\sum_{j=k}^{\infty} \alpha_j p_j(\mathbf{M})\right\|_2 \leq \sum_{j=k}^{\infty} CM\varrho^{-j}\|\mathbf{V}\|_2 \|p_j(\mathbf{\Lambda})\|_2 \left\|\mathbf{V}^{-1}\right\|_2 \\ &= CM\|\mathbf{V}\|_2 \left\|\mathbf{V}^{-1}\right\|_2 \sum_{j=k}^{\infty} \varrho^{-j} \max_{\lambda\in\sigma(\mathbf{M})} |p_j(\lambda)|\,.\end{aligned}$$

Since λ lies in the interior of Σ for all $\lambda \in \sigma(\mathbf{M})$, by Lemma 5.2.2, there exists some constant c such that

$$|p_j(\lambda)| = \left|\hat{\delta}_j(\lambda)\right| \mathrm{e}^{j\mathrm{Reg}_\Sigma(\lambda)} \leq c,$$

for all j and $\lambda \in \sigma(\mathbf{M})$. The theorem follows from using $C' = cC$ and

$$\sum_{j=k}^{\infty} \varrho^{-j} = \frac{\varrho^{-k}}{1-\varrho^{-1}}. \tag{6.11}$$

□

Theorem 6.2.6 implies that the Akhiezer iteration converges, at worst, at a geometric rate given by ϱ^{-1}. If f is the sign function, then $M = 1$. This theorem then implies that there exists some constant $D_{\mathbf{M}} > 0$, dependent on the eigenvectors of $\mathbf{M}$, such that

$$\|\mathrm{sign}(\mathbf{M}) - \mathbf{F}_k\|_2 \leq D_{\mathbf{M}} \frac{\varrho^{-k}}{1-\varrho^{-1}}. \tag{6.12}$$

In particular, for a given tolerance $\epsilon > 0$, we require

$$k = \left\lceil -\log_\varrho \frac{\epsilon(1-\varrho^{-1})}{D_{\mathbf{M}}} \right\rceil, \tag{6.13}$$

iterations to ensure that

$$\|\mathrm{sign}(\mathbf{M}) - \mathbf{F}_k\|_2 < \epsilon.$$

Since it depends on eigenvectors and orthogonal polynomial asymptotics, $D_{\mathbf{M}}$ cannot be cheaply computed in general. Like with the orthogonal polynomial series coefficients, we hypothesize an upper bound $D_{\mathbf{M}} \leq 10\ell$ for generic $\mathbf{M} \in \mathbb{C}^{\ell\times\ell}$ that are not highly nonnormal. One can always run more iterations if this hypothesized bound appears insufficient.

We note that in finite-precision, the iteration will saturate when the coefficients α_j reach machine precision $\varepsilon_{\mathrm{mach}}$. We use

$$k = \left\lceil \min\left\{ -\log_\varrho\left(\frac{\epsilon(1-\varrho^{-1})}{10\ell}\right), -\log_\varrho\left(\frac{\varepsilon_{\mathrm{mach}}}{5}\right)\right\}\right\rceil,$$

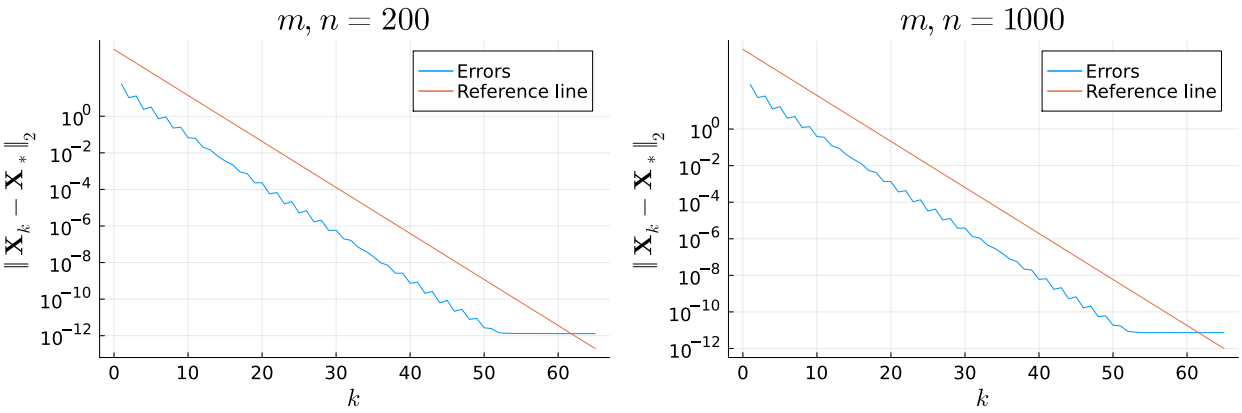


Figure 6.7: 2-norm error for $\mathbf{X}_k$ against the true solution $\mathbf{X}_*$ to (1.2) (computed via `sylvester` in `Julia`) at each iteration where $\mathbf{C}$ is rank 2, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{200\times 200}$ (left), $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{1000\times 1000}$ (right), $\sigma(\mathbf{A}) \subset [2, 3]$, and $\sigma(\mathbf{B}) \subset [-1.8, -0.5]$. Here, the reference line denotes the convergence rate heuristic (6.14) with $D_{\mathbf{H}}$ replaced by the hypothesized upper bound $10(n + m)$.

iterations in place of (6.13).

Now, consider $\mathbf{H}$ from (6.8) and let $\mathbf{X}_*$ denote the true solution of the Sylvester equation (1.2). Of course, the error of interest is $\mathbf{X}_k - \mathbf{X}_*$, not $\operatorname{sign}(\mathbf{H}) - \mathbf{F}_k$. Naively, we have

$$\|\mathbf{X}_k - \mathbf{X}_*\|_2 \leq \frac{1}{2} \|\operatorname{sign}(\mathbf{H}) - \mathbf{F}_k\|_2 \leq \frac{D_{\mathbf{H}}}{2} \frac{\varrho^{-k}}{1 - \varrho^{-1}}. \tag{6.14}$$

Empirically, we find that the error in this iteration does not decay faster than the geometric rate given by ϱ^{-1}, so we do not seek to improve it further — we use the same hypothesized upper bound $D_{\mathbf{H}} \leq 10(n + m)$ to determine the iteration count (6.13). In Figure 6.7, we plot the errors $\|\mathbf{X}_k - \mathbf{X}_*\|_2$ at each iteration against this heuristic and observe that it does indeed capture the convergence rate and provide an upper bound until errors saturate.

6.2.5 Quadrature error analysis

In this subsection, we discuss the inexactness of quadrature rules. These results can also be found in Chapter 5. One assumption that was made in the preceding convergence analysis was that the coefficients α_j in the p_j-series expansion of f were computed exactly. To relax this assumption, let

$$\rho_m(x) := \int_\Gamma \frac{f(z)}{z - x} \mathrm{d}z - \sum_{j=1}^{m} \frac{f(z_j)}{z_j - x} w_j, \tag{6.15}$$

be a measure of the quadrature rule error. Denote the kth iteration of Algorithm 3, with the approximate coefficients, by

$$f_{k,m}(\mathbf{M}) := -\sum_{\ell=0}^{k-1} \left(\sum_{j=1}^{m} f(z_j) w_j \mathcal{C}_\Sigma \left[p_\ell w \right] (z_j) \right) p_\ell(\mathbf{M}).$$

Theorem 6.2.7. *Under the assumptions of Lemma 5.3.6, let* $\mathbf{M}$ *be a square matrix and suppose that*

$$\sum_{\ell=0}^{\infty} \varrho^{-\ell} \|p_\ell(\mathbf{M})\|_2 < \infty.$$

Further suppose that Γ, f *are as in Definition 5.3.2 and* $\{z_j\}_{j=1}^m$, $\{w_j\}_{j=1}^m$ *are quadrature nodes and weights for* Γ*, respectively. Then, there exist* $\varepsilon_{m,\ell}$ *satisfying* $|\varepsilon_{m,\ell}| \leq \|\rho_m\|_\infty := \max_{z \in \Sigma} |\rho_m(z)|$ *and a constant* $C > 0$*, independent of* f*, such that*

$$\left\| f(\mathbf{M}) - f_{k,m}(\mathbf{M}) + \sum_{\ell=0}^{k-1} \varepsilon_{m,\ell} p_\ell(\mathbf{M}) \right\|_2 \leq CM \sum_{\ell=k}^{\infty} \varrho^{-\ell} \|p_\ell(\mathbf{M})\|_2.$$

In particular, following the proof of Theorem 6.2.6, Theorem 6.2.7 implies that when a generic $\mathbf{M}$ is diagonalized as $\mathbf{M} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ and satisfies $\sigma(\mathbf{M}) \subset \Sigma$, there exists some constant $C' > 0$ such that

$$\|f(\mathbf{M}) - f_{k,m}(\mathbf{M})\|_2 \leq \|\mathbf{V}\|_2 \|\mathbf{V}^{-1}\|_2 \left(\|\rho_m\|_\infty \sum_{\ell=0}^{k-1} \|p_\ell(\mathbf{\Lambda})\|_2 + C' M \frac{\varrho^{-k}}{1 - \varrho^{-1}} \right). \tag{6.16}$$

Since $\|p_\ell(\mathbf{M})\|_2 = \mathrm{O}(1)$ when $\mathbf{M}$ is generic and $\sigma(\mathbf{M}) \subset \Sigma$, the former term in the sum is likely to be small; however, as k increases, this term will eventually become large. See [20, Appendix A] for a discussion of why the errors still tend to remain small in this case. This appendix also contains convergence analysis for generic matrices whose eigenvalues may not lie in Σ.

6.2.6 Complexity and storage

To analyze the complexity of Algorithm 7, we assume that the numerical rank of $c\varrho^{-k}\mathbf{J}_k\mathbf{K}_k$ is given by $R(k)$ and that $R(k) = 0$ for all $k \geq k^*$ and some $k^* \in \mathbb{N}$. That is, we assume that the numerical rank of the weighted compression of $\mathbf{J}_k\mathbf{K}_k$ is bounded and eventually becomes zero. This implies that the numerical rank of $\mathbf{W}_k\mathbf{Z}_k$ is also bounded, and we denote this quantity by $\hat{R}(k)$. Suppose that $T_{\mathbf{A}}$ is the cost of a left matrix-vector multiplication by $\mathbf{A}$ and $T_{\mathbf{B}}$ is the cost of a right matrix-vector multiplication by $\mathbf{B}$.

At the kth iteration of Algorithm 7, constructing the matrices $\mathbf{V}p_k(\mathbf{A})$ and $p_k(\mathbf{B})\mathbf{U}$ requires at most $r(T_{\mathbf{A}} + T_{\mathbf{B}}) + 5r(m+n)$ arithmetic operations. The construction of $\mathbf{J}_k$

and $\mathbf{K}_k$ requires at most $(2R(k-1)+R(k-2)+r)m + R(k-1)T_{\mathbf{A}}$ arithmetic operations, and the construction of $\mathbf{W}_{k+1}$ and $\mathbf{Z}_{k+1}$ requires at most $(R(k)+r)m$ arithmetic operations. The compression of $\mathbf{J}_k\mathbf{K}_k$ $\texttt{COMPRESS}(\mathbf{J}_k, \mathbf{K}_k, \frac{\varrho^k\epsilon}{c})$ requires $\mathrm{O}((2R(k-1)+R(k-2)+r)^3 + (2R(k-1)+R(k-2)+r)^2(m+n))$ arithmetic operations, while the compression of $\mathbf{W}_k\mathbf{Z}_k$ $\texttt{COMPRESS}\,(\mathbf{W}_k, \mathbf{Z}_k, \epsilon)$ requires $\mathrm{O}((R(k)+\hat{R}(k)+r)^3 + (R(k)+\hat{R}(k)+r)^2(m+n))$ arithmetic operations. Finally, the cost of computing the recurrence coefficients and p_j-series coefficients is assumed to be $\mathrm{O}(1)$, since computing the Akhiezer data requires $\mathrm{O}(k)$ operations for k iterations.

To bound the complexity of Algorithm 7 run for k iterations, denote

$$\mathcal{R} := \max_{j\in\mathbb{N}} \left\{ R(j), \hat{R}(j) \right\}.$$

Then, to run Algorithm 7 for k iterations, we need $\mathrm{O}(kr(T_{\mathbf{A}}+T_{\mathbf{B}}) + k\mathcal{R}T_{\mathbf{A}} + k(\mathcal{R}+r)^3 + k(\mathcal{R}+r)^2(m+n))$ arithmetic operations. Note that when the stopping criterion (6.13) is applied, $k = \mathrm{O}(\log(m+n))$.

In the general case when $T_{\mathbf{A}} = \mathrm{O}(n^2)$, $T_{\mathbf{B}} = \mathrm{O}(m^2)$, $r, \mathcal{R} \ll m, n$, and the stopping criterion (6.13) is applied, Algorithm 7 requires[6] $\mathrm{O}\left(m^2\log m + n^2 \log n\right)$ arithmetic operations. In contrast, the Bartels–Stewart algorithm requires $\mathrm{O}\left(m^3+n^3\right)$ arithmetic operations [23]. Furthermore, methods such as Alternating-Directional-Implicit (ADI) iterations that require computing matrix inverses will also necessitate $\mathrm{O}\left(m^3+n^3\right)$ arithmetic operations in general.

Under these same assumptions, a maximum of $(10\mathcal{R}+6r)(m+n)$ stored entries are required at any given iteration. Therefore, in the general case $r, \mathcal{R} \ll m, n$, Algorithm 7 requires at most $\mathrm{O}(m+n)$ stored entries. By contrast, a standard polynomial Krylov method such as [137, Algorithm 5] can require $\mathrm{O}(mn)$ (or worse) stored entries when the convergence rate is slow.

6.3 Method 2: inverting the Sylvester operator

For a Sylvester equation (1.2), define the Sylvester operator $\mathcal{S}_{\mathbf{A},\mathbf{B}} : \mathbb{C}^{m\times n} \to \mathbb{C}^{m\times n}$ by

$$\mathcal{S}_{\mathbf{A},\mathbf{B}}\mathbf{Y} = \mathbf{YA} - \mathbf{BY}. \tag{6.17}$$

Our second method directly builds a polynomial approximation to $\mathcal{S}_{\mathbf{A},\mathbf{B}}^{-1}\mathbf{C}$, the solution to (1.2). Applying the Akhiezer iteration to solve (6.17) requires only the action of $\mathcal{S}_{\mathbf{A},\mathbf{B}}$ on a matrix, so it can be implemented by iteratively applying (6.17). In particular, we have the following theorem:

Theorem 6.3.1. *Assume that $\sigma(\mathcal{S}_{\mathbf{A},\mathbf{B}})$ is contained in the interior of $\Sigma = \bigcup_{j=1}^{g+1}[a_j, b_j] \subset \mathbb{R}$ where $0 \notin \Sigma$. Let w be a weight function of the form (3.5) and denote the corresponding*

[6]Note that $\mathrm{O}\left((m^2+n^2)\log(m+n)\right)$ is equivalent to $\mathrm{O}\left(m^2\log m + n^2\log n\right)$.

orthonormal polynomials by $(p_j(x))_{j=0}^{\infty}$ *with recurrence coefficients* $(\check{a}_j, \check{b}_j)_{j=0}^{\infty}$. *Then, the solution of the Sylvester equation* (1.2) *is given by*

$$\mathbf{X} = \sum_{j=0}^{\infty} \alpha_j \mathbf{P}_j, \quad \alpha_j = 2\pi \mathrm{i} \mathcal{C}\left[p_j w\right](0),$$

where the matrices $\mathbf{P}_j$ *are given by the recurrence*

$$\begin{aligned}
\mathbf{P}_0 &= \mathbf{C}, \\
\mathbf{P}_1 &= \frac{1}{\check{b}_0}(\mathbf{P}_0\mathbf{A} - \mathbf{B}\mathbf{P}_0 - \check{a}_0\mathbf{P}_0), \\
\mathbf{P}_j &= \frac{1}{\check{b}_{j-1}}(\mathbf{P}_{j-1}\mathbf{A} - \mathbf{B}\mathbf{P}_{j-1} - \check{a}_{j-1}\mathbf{P}_{j-1} - \check{b}_{j-2}\mathbf{P}_{j-2}), \quad j \geq 2.
\end{aligned}$$

Proof. By Lemma 5.2.2, the function $f(x) = 1/x$ is represented as an absolutely convergent p_j-series on Σ as

$$\frac{1}{x} = \sum_{j=0}^{\infty} \alpha_j p_j(x), \quad \alpha_j = \int_{\Sigma} \frac{p_j(x)w(x)}{x} \mathrm{d}x = 2\pi \mathrm{i} \mathcal{C}\left[p_j w\right](0).$$

By the assumptions on $\sigma(\mathcal{S}_{\mathbf{A},\mathbf{B}})$,

$$\mathbf{X} = \mathcal{S}_{\mathbf{A},\mathbf{B}}^{-1}\mathbf{C} = \sum_{j=0}^{\infty} \alpha_j p_j\left(\mathcal{S}_{\mathbf{A},\mathbf{B}}\right)\mathbf{C}.$$

□

As before, the assumption that $\sigma(\mathcal{S}_{\mathbf{A},\mathbf{B}}) \subset \Sigma$ is unnecessary, see Remark 6.2.2. As with Method 1, truncating the infinite series of Theorem 6.3.1 directly yields an iterative method for solving the Sylvester equation (1.2); however, the recurrence and series coefficients can be made more explicit in the typical use case. The eigenvalues of $\mathcal{S}_{\mathbf{A},\mathbf{B}}$ are precisely the set $\{\lambda - \mu\}_{\lambda \in \sigma(\mathbf{A}), \mu \in \sigma(\mathbf{B})} = \sigma(\mathbf{A}) - \sigma(\mathbf{B})$, so intervals containing $\sigma(\mathcal{S}_{\mathbf{A},\mathbf{B}})$ are easily determined from those containing $\sigma(\mathbf{A})$ and $\sigma(\mathbf{B})$. In particular, if $\sigma(\mathbf{B}) \subset [a_1, b_1]$ and $\sigma(\mathbf{A}) \subset [a_2, b_2]$, $b_1 < a_2$, then $\sigma(\mathcal{S}_{\mathbf{A},\mathbf{B}}) \subset \Sigma = [a_2 - b_1, b_2 - a_1]$. In this case, the iteration can be constructed with Chebyshev polynomials (shifted and scaled to Σ) and their particularly simple recurrence relation. An implementation of the iteration in this case is given as Algorithm 8 (cf. [19, Algorithm 1]). In the single-interval case, this method is essentially a modification of the classical Chebyshev iteration applied to solve the equivalent linear system $\left(\mathbf{A}^T \otimes \mathbf{I}_m - \mathbf{I}_n \otimes \mathbf{B}\right) \mathrm{vec}(\mathbf{X}) = \mathrm{vec}(\mathbf{C})$. See [19, Appendix C] for a discussion of this. When Σ consists of more than one interval, Algorithm 8 (and Algorithm 9 below) may be generalized by constructing $\mathbf{P}_k$ and S_k with the Akhiezer data as in Theorem 6.3.1 and

Algorithm 8: $\mathbf{XA} - \mathbf{BX} = \mathbf{C}$ solver via the Akhiezer iteration for $1/x$

Input: Matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and parameters α, c such that $\sigma(\mathbf{A}) - \sigma(\mathbf{B}) \subset [\alpha - c, \alpha + c]$.

Set $\varrho^{-1} = -\frac{\alpha}{c} + \sqrt{\frac{\alpha}{c} - 1}\sqrt{\frac{\alpha}{c} + 1}$.

for *k=0,1,...* **do**

  if *k=0* **then**

    Set $\mathbf{P}_0 = \mathbf{C}$.

    Set $S_0 = \frac{1}{\sqrt{\alpha - c}\sqrt{\alpha + c}}$.

    Set $\mathbf{X}_1 = S_0 \mathbf{P}_0$.

  else if *k=1* **then**

    Set $\mathbf{P}_1 = \frac{1}{c}(\mathbf{P}_0 \mathbf{A} - \mathbf{P}_0 \mathbf{B} - \alpha \mathbf{P}_0)$.

    Set $S_1 = S_0 \varrho^{-1}$.

    Set $\mathbf{X}_2 = \mathbf{X}_1 + 2 S_1 \mathbf{P}_1$.

  else

    Set $\mathbf{P}_k = \frac{2}{c}(\mathbf{P}_{k-1} \mathbf{A} - \mathbf{P}_{k-1} \mathbf{B} - \alpha \mathbf{P}_{k-1}) - \mathbf{P}_{k-2}$.

    Set $S_k = S_{k-1} \varrho^{-1}$.

    Set $\mathbf{X}_{k+1} = \mathbf{X}_k + 2 S_k \mathbf{P}_k$.

  end

  if converged **then**

    return $\mathbf{X}_{k+1}$.

  end

end

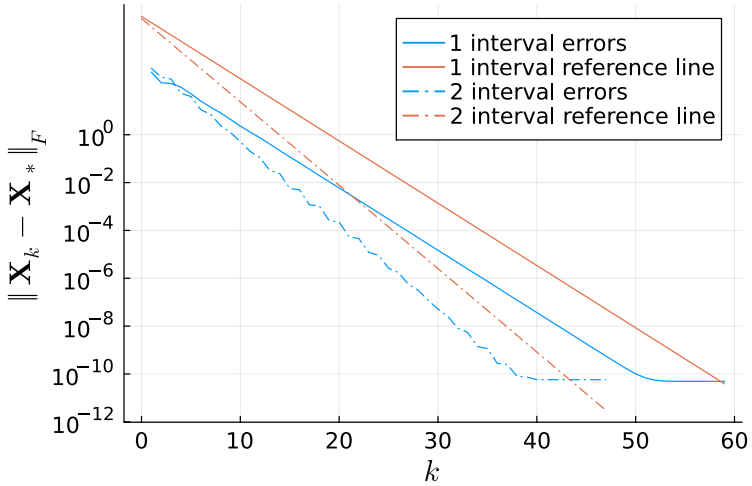


Figure 6.8: Frobenius norm error for $\mathbf{X}_k$ against the true solution $\mathbf{X}_*$ to (1.2) (computed via `sylvester` in `Julia`) at each iteration of Method 2 where $\mathbf{C}$ is rank 2, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{1000\times 1000}$, $\sigma(\mathbf{A}) \subset [0.5, 1] \cup \{10\}$, and $\sigma(\mathbf{B}) \subset [-1.8, -0.5]$. For the solid lines, Algorithm 9 is used with $\Sigma = [1, 11.8]$, and for the dashed lines, Algorithm 9 is modified to use Akhiezer polynomials with $\Sigma = [1, 2.8] \cup [10.5, 11.8]$. The reference lines denote the convergence rate of Corollary 6.3.3 with the leading constants replaced by the heuristic $D_{\mathbf{A},\mathbf{B},\mathbf{C}} = 20(m+n)$.

Algorithm 3. In Figure 6.8, we include an example where $\mathbf{A}$ has a single outlier eigenvalue, and we observe that convergence is accelerated when Σ is chosen to consist of two intervals instead of one.

6.3.1 Low-rank structure and compression

As with Method 1, we handle low-rank problems of the form (6.10) by writing the recurrence in block low-rank form and compressing matrices to their numerical rank as needed. The procedure for this is essentially the same as for the sign function iteration. In particular, letting $\mathbf{P}_j = \mathbf{J}_j\mathbf{K}_j$, the recurrence of Theorem 6.3.1 becomes

$$\mathbf{J}_0\mathbf{K}_0 = \mathbf{P}_0 = \mathbf{C},$$

$$\mathbf{J}_1\mathbf{K}_1 = \mathbf{P}_1 = \frac{1}{\check{b}_0}\begin{pmatrix}\mathbf{J}_0 & -\mathbf{B}\mathbf{J}_0 & -\check{a}_0\mathbf{J}_0\end{pmatrix}\begin{pmatrix}\mathbf{K}_0\mathbf{A}\\ \mathbf{K}_0\\ \mathbf{K}_0\end{pmatrix},$$

$$\mathbf{J}_j\mathbf{K}_j = \mathbf{P}_j = \frac{1}{\check{b}_{j-1}}\begin{pmatrix}\mathbf{J}_{j-1} & -\mathbf{B}\mathbf{J}_{j-1} & -\check{a}_{j-1}\mathbf{J}_{j-1} & -\check{b}_{j-2}\mathbf{J}_{j-2}\end{pmatrix}\begin{pmatrix}\mathbf{K}_{j-1}\mathbf{A}\\ \mathbf{K}_{j-1}\\ \mathbf{K}_{j-1}\\ \mathbf{K}_{j-2}\end{pmatrix}, \quad j \geq 2.$$

Letting $\mathbf{X}_k = \mathbf{W}_k\mathbf{Z}_k$, the update step can be written as

$$\mathbf{W}_{k+1}\mathbf{Z}_{k+1} = \mathbf{X}_{k+1} = \begin{pmatrix}\mathbf{W}_k & \alpha_k\mathbf{J}_k\end{pmatrix}\begin{pmatrix}\mathbf{Z}_k \\ \mathbf{K}_k\end{pmatrix}.$$

With this in hand, Method 2 can be implemented in low-rank form as in Algorithm 9.

We now motivate the choice of compression tolerance. As before, Lemma 5.2.2 guarantees that the coefficients α_j satisfy

$$|\alpha_j| \leq C\varrho^{-j}, \quad \varrho = \mathrm{e}^{\mathrm{Reg}_\Sigma(0)}, \tag{6.18}$$

for some constant $C > 0$ that cannot in general be computed. Heuristically, we again find that the weighted compression step behaves well when the tolerance is multiplied by ϱ^k/c with $c = 5$. In the single-interval case presented in Algorithm 9, the coefficients S_k are obtained by iteratively dividing by ϱ, meaning that (6.18) holds for $C = 2S_0$. Thus, one may instead weight the compression directly by S_k. Empirically, some precision is lost when this is done, so we continue to use the heuristic in this case.

To demonstrate our heuristics, we repeat the experiment of Figures 6.4 and 6.5. In Figure 6.9, we plot the numerical rank of $\mathbf{J}_k\mathbf{K}_k$ and $\mathbf{W}_k\mathbf{Z}_k$ with and without weighting the compression step. We observe that the weighting bounds the numerical rank of $\mathbf{J}_k\mathbf{K}_k$ while the error appears unchanged.

6.3.2 Convergence and stopping criteria

We now present a convergence analysis of Method 2 that essentially follows from that in Section 6.2.4 and is again largely a simplified version of that of [19, Section 4]. The following theorem results from [19, Theorem 4.5] and describes the convergence of the Akhiezer iteration applied to solve a linear system.

Theorem 6.3.2. *Let $\Sigma = \bigcup_{j=1}^{g+1}[a_j, b_j] \subset \mathbb{R}$ be a union of disjoint intervals such that $0 \notin \Sigma$, and let a generic matrix $\mathbf{M} \in \mathbb{C}^{\ell\times\ell}$ be diagonalized as $\mathbf{M} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ and satisfy $\sigma(\mathbf{M}) \subset \Sigma$. Let w be a weight function of the form (3.5) and denote the corresponding orthonormal polynomials by $(p_j(x))_{j=0}^{\infty}$. Then, there exists a constant $d > 0$, depending only on the eigenvalues of $\mathbf{M}$, such that*

$$\left\|\sum_{j=0}^{k-1} 2\pi\mathrm{i}\mathcal{C}\left[p_j w\right](0)p_j(\mathbf{M})\mathbf{b} - \mathbf{M}^{-1}\mathbf{b}\right\|_2 \leq d\|\mathbf{V}\|_2\left\|\mathbf{V}^{-1}\right\|_2\|\mathbf{b}\|_2\frac{\varrho^{-k}}{1-\varrho^{-1}}, \quad \varrho = \mathrm{e}^{\mathrm{Reg}_\Sigma(0)}.$$

The following corollary describes the convergence rate of Method 2.

Corollary 6.3.3. *Under the assumptions of Theorem 6.3.1 on $\mathbf{M} = \mathcal{S}_{\mathbf{A},\mathbf{B}}$, denote the solution of (1.2) by $\mathbf{X}$, and let $\mathbf{A}$ and $\mathbf{B}$ be diagonalizable and $\sigma(\mathbf{A}) - \sigma(\mathbf{B})$ be contained in the interior of Σ. Then, there exists $d > 0$, depending only on the eigenvalues of $\mathbf{M}$, such that*

Algorithm 9: Low-rank $\mathbf{XA} - \mathbf{BX} = \mathbf{UV}$ solver via the Akhiezer iteration for $1/x$

Input: Matrices $\mathbf{A}, \mathbf{B}, \mathbf{U}, \mathbf{V}$, parameters α, c such that $\sigma(\mathbf{A}) - \sigma(\mathbf{B}) \subset [\alpha - c, \alpha + c]$, and compression tolerance ϵ.

Initialize $\mathbf{W}_0$ and $\mathbf{Z}_0$ to be empty.

Set $\varrho^{-1} = -\frac{\alpha}{c} + \sqrt{\frac{\alpha}{c} - 1}\sqrt{\frac{\alpha}{c} + 1}$.

for *k=0,1,...* **do**

  if *k=0* **then**

    Set $\mathbf{J}_0 = \mathbf{U}$.

    Set $\mathbf{K}_0 = \mathbf{V}$.

    Set $S_0 = \frac{1}{\sqrt{\alpha - c}\sqrt{\alpha + c}}$.

  else if *k=1* **then**

    Set $\mathbf{J}_1 = \begin{pmatrix} \frac{1}{c}\mathbf{J}_0 & -\frac{1}{c}\mathbf{B}\mathbf{J}_0 & -\frac{\alpha}{c}\mathbf{J}_0 \end{pmatrix}$.

    Set $\mathbf{K}_1 = \begin{pmatrix} \mathbf{K}_0\mathbf{A} \\ \mathbf{K}_0 \\ \mathbf{K}_0 \end{pmatrix}$.

    Set $S_1 = S_0\varrho^{-1}$.

  else

    Set $\mathbf{J}_k = \begin{pmatrix} \frac{2}{c}\mathbf{J}_{k-1} & -\frac{2}{c}\mathbf{B}\mathbf{J}_{k-1} & -\frac{2\alpha}{c}\mathbf{J}_{k-1} & -\mathbf{J}_{k-2} \end{pmatrix}$.

    Set $\mathbf{K}_k = \begin{pmatrix} \mathbf{K}_{k-1}\mathbf{A} \\ \mathbf{K}_{k-1} \\ \mathbf{K}_{k-1} \\ \mathbf{K}_{k-2} \end{pmatrix}$.

    Set $S_k = S_{k-1}\varrho^{-1}$.

  end

  Set $\mathbf{J}_k, \mathbf{K}_k = \texttt{COMPRESS}(\mathbf{J}_k, \mathbf{K}_k, \frac{\varrho^k\epsilon}{c})$.

  if *k=0* **then**

    Set $\mathbf{W}_{k+1} = \begin{pmatrix} \mathbf{W}_k & S_k\mathbf{J}_k \end{pmatrix}$.

  else

    Set $\mathbf{W}_{k+1} = \begin{pmatrix} \mathbf{W}_k & 2S_k\mathbf{J}_k \end{pmatrix}$.

  end

  Set $\mathbf{Z}_{k+1} = \begin{pmatrix} \mathbf{Z}_k \\ \mathbf{K}_k \end{pmatrix}$.

  Set $\mathbf{W}_{k+1}, \mathbf{Z}_{k+1} = \texttt{COMPRESS}\left(\mathbf{W}_{k+1}, \mathbf{Z}_{k+1}, \epsilon\right)$.

  if converged **then**

    return $\mathbf{W}_{k+1}, \mathbf{Z}_{k+1}$.

  end

end

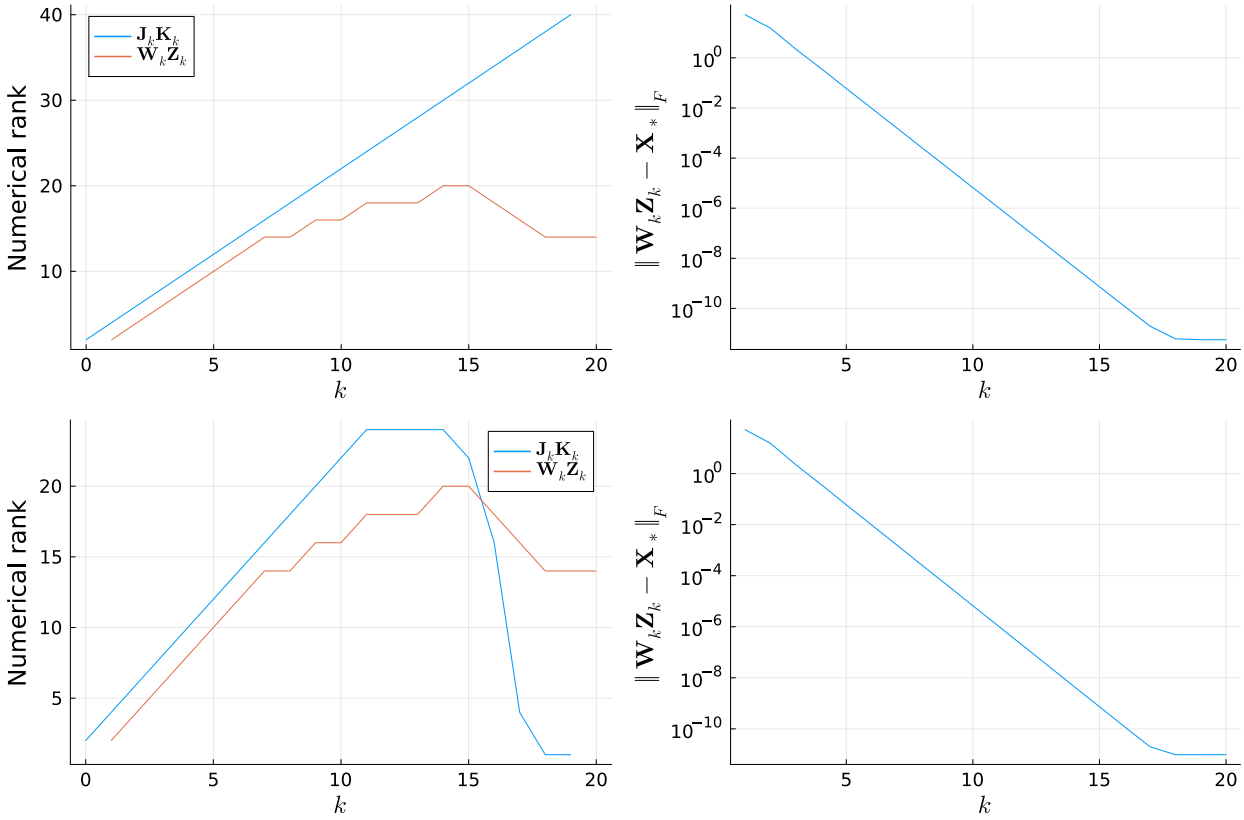


Figure 6.9: Numerical rank and error of the iterates of Algorithm 9 with unweighted compression (top) and weighted compression (bottom). The matrices $\mathbf{A}, \mathbf{B}, \mathbf{U}, \mathbf{V}$ are the same as in Figure 6.4.

the iterate $\mathbf{X}_k = \sum_{j=0}^{k-1} \alpha_j \mathbf{P}_j$ *satisfies*

$$\|\mathbf{X}_k - \mathbf{X}\|_F \leq d\|\mathbf{V_A}\|_2 \left\|\mathbf{V_A}^{-1}\right\|_2 \|\mathbf{V_B}\|_2 \left\|\mathbf{V_B}^{-1}\right\|_2 \|\mathbf{C}\|_F \frac{\varrho^{-k}}{1-\varrho^{-1}}, \quad \varrho = \mathrm{e}^{\mathrm{Reg}_\Sigma(0)},$$

where $\mathbf{A}$ *and* $\mathbf{B}$ *are diagonalized as* $\mathbf{A} = \mathbf{V_A}\mathbf{\Lambda_A}\mathbf{V_A}^{-1}$ *and* $\mathbf{B} = \mathbf{V_B}\mathbf{\Lambda_B}\mathbf{V_B}^{-1}$*, respectively.*

Proof. The eigenvector matrix of $\mathbf{A}^T \otimes \mathbf{I}_m - \mathbf{I}_n \otimes \mathbf{B}$ is given by $\mathbf{V_A}^T \otimes \mathbf{V_B}$. The result follows from the identity $\|\mathbf{M} \otimes \mathbf{N}\|_2 = \|\mathbf{M}\|_2 \|\mathbf{N}\|_2$. □

Thus, Method 2 converges at a geometric rate $\varrho^{-1} = \mathrm{e}^{-\mathrm{Reg}_\Sigma(0)}$. In the single-interval case $\Sigma = [a, b]$,

$$\varrho^{-1} = -\frac{b+a}{b-a} + \sqrt{\frac{b+a}{b-a} - 1}\sqrt{\frac{b+a}{b-a} + 1}.$$

Denote $D_{\mathbf{A},\mathbf{B},\mathbf{C}} = d\|\mathbf{V_A}\|_2 \left\|\mathbf{V_A}^{-1}\right\|_2 \|\mathbf{V_B}\|_2 \left\|\mathbf{V_B}^{-1}\right\|_2 \|\mathbf{C}\|_F$. Then, for a given tolerance $\epsilon > 0$, we require

$$k = \left\lceil -\log_\varrho \frac{\epsilon(1-\varrho^{-1})}{D_{\mathbf{A},\mathbf{B},\mathbf{C}}} \right\rceil,$$

iterations to ensure that $\|\mathbf{X}_k - \mathbf{X}\|_F < \epsilon$. For generic $\mathbf{A} \in \mathbb{C}^{n\times n}$ and $\mathbf{B} \in \mathbb{C}^{m\times m}$ that are not highly nonnormal, we hypothesize an upper bound $D_{\mathbf{A},\mathbf{B},\mathbf{C}} \leq 20(m+n)$. In finite precision, for a given tolerance ϵ, we use

$$k = \left\lceil \min\left\{ -\log_\varrho \left(\frac{\epsilon(1-\varrho^{-1})}{20(m+n)} \right), -\log_\varrho \left(\frac{\varepsilon_{\mathrm{mach}}}{5} \right) \right\} \right\rceil,$$

iterations as our stopping criterion.

We demonstrate these heuristics by repeating the experiment of Figure 6.7. In Figure 6.10, we plot the errors $\|\mathbf{X}_k - \mathbf{X}_*\|_F$ against the bound of Corollary 6.3.3 with $D_{\mathbf{A},\mathbf{B},\mathbf{C}}$ replaced by $20(m+n)$ and observe that it does indeed bound the error.

6.3.3 Complexity and storage

We analyze the complexity of Algorithm 7 under the same assumptions as with Method 1. That is, we assume that the numerical rank of $c\varrho^{-k}\mathbf{J}_k\mathbf{K}_k$ is given by $R(k)$ and that $R(k) = 0$ for all $k \geq k^*$ and some $k^* \in \mathbb{N}$. We denote the numerical rank of $\mathbf{W}_k\mathbf{Z}_k$ by $\hat{R}(k)$, the cost of a left matrix-vector multiplication by $\mathbf{A}$ by $T_\mathbf{A}$, and the cost of a right matrix-vector multiplication by $\mathbf{B}$ by $T_\mathbf{B}$.

At the kth iteration of Algorithm 9, the construction of $\mathbf{J}_k$ and $\mathbf{K}_k$ requires at most $3mR(k-1) + (T_\mathbf{A} + T_\mathbf{B})R(k-1)$ arithmetic operations, and the construction of $\mathbf{W}_{k+1}$ and $\mathbf{Z}_{k+1}$ requires at most $mR(k)$ arithmetic operations. The compression of $\mathbf{J}_k\mathbf{K}_k$ `COMPRESS`$(\mathbf{J}_k, \mathbf{K}_k, \frac{\varrho^k\epsilon}{c})$ requires $\mathrm{O}((3R(k-1) + R(k-2))^3 + (3R(k-1) + R(k-2))^2(m+n))$ arithmetic operations, while the compression of $\mathbf{W}_k\mathbf{Z}_k$ `COMPRESS` $(\mathbf{W}_k, \mathbf{Z}_k, \epsilon)$ requires $\mathrm{O}((R(k)+\hat{R}(k))^3+$

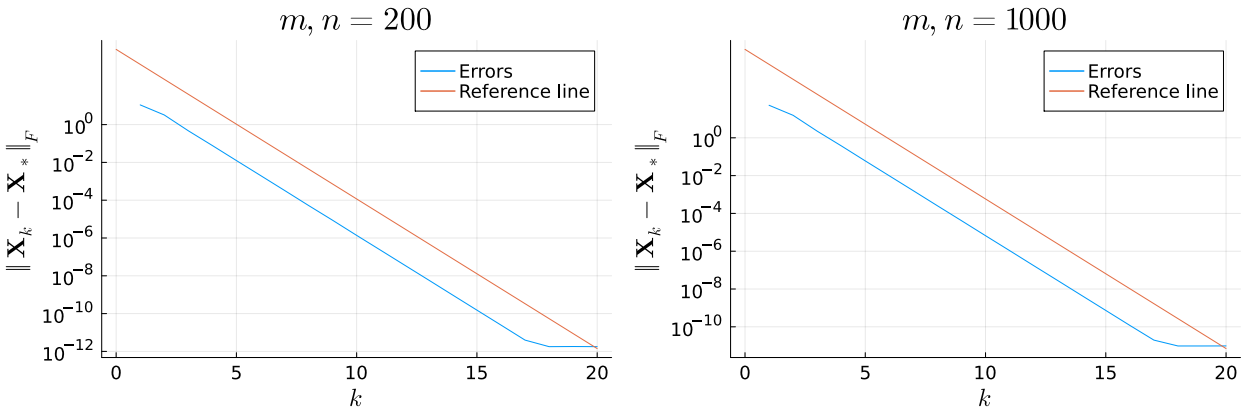


Figure 6.10: Frobenius norm error for $\mathbf{X}_k$ against the true solution $\mathbf{X}_*$ to (1.2) at each iteration where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are the same as in Figure 6.7. Here, the reference line denotes the convergence rate of Corollary 6.3.3 with the leading constants replaced by the heuristic $D_{\mathbf{A},\mathbf{B},\mathbf{C}} = 20(m+n)$.

$(R(k) + \hat{R}(k))^2(m+n))$ arithmetic operations. Finally, the cost of computing the coefficient S_k is $\mathrm{O}(1)$. For multiple intervals, the cost of computing the recurrence coefficients and p_j-series coefficients is again assumed to be $\mathrm{O}(1)$, as computing the Akhiezer data requires $\mathrm{O}(k)$ operations for k iterations.

As before, we bound the complexity of Algorithm 9 run for k iterations by denoting

$$\mathcal{R} := \max_{j \in \mathbb{N}} \left\{ R(j), \hat{R}(j) \right\}.$$

Then, to run Algorithm 9 for k iterations, we need $\mathrm{O}(k\mathcal{R}(T_{\mathbf{A}} + T_{\mathbf{B}}) + k(\mathcal{R}+r)^3 + k(\mathcal{R}+r)^2(m+n))$ arithmetic operations. Due to the extra $k\mathcal{R}T_{\mathbf{B}}$ term, this will typically be more expensive than running Method 1 (Algorithm 7) for the same number of iterations; however, as we demonstrate in Figure 6.1, fewer iterations are typically needed to reach a given tolerance. As with Method 1, when $T_{\mathbf{A}} = \mathrm{O}(n^2)$, $T_{\mathbf{B}} = \mathrm{O}(m^2)$, $r, \mathcal{R} \ll m, n$, and our stopping criterion is applied, Algorithm 9 requires $\mathrm{O}\left(m^2 \log m + n^2 \log n\right)$ arithmetic operations.

Under these same assumptions, a maximum of $10\mathcal{R}(m+n)$ stored entries are required at any given iteration. Therefore, in the general case $r, \mathcal{R} \ll m, n$, Algorithm 9 requires at most $\mathrm{O}(m+n)$ stored entries.

Matrix size ($n \times n$)	Method 1	Method 2	Polynomial Krylov
$n = 4000$	1.26813	0.349146	0.302879
6000	2.43285	0.74685	0.661923
8000	4.81385	1.35561	1.34982
10000	7.01396	2.32737	2.18676
12000	10.4664	3.27932	3.18396
14000	15.3646	4.3167	4.65676
16000	18.6218	5.96538	6.22975
18000	25.5412	7.20563	7.77245

Table 6.1: Runtimes (in seconds) for the inverse-free methods as shown in Figure 6.11 (left). Despite requiring more iterations, Method 2 eventually matches and slightly outperforms the polynomial Krylov method. Importantly, it requires far less memory to run. Unlike the polynomial Krylov method, Methods 1 and 2 have memory costs that remain bounded as the number of iterations grows.

6.4 Examples and applications

6.4.1 Comparison of methods

Here, we test the performance of Methods 1 and 2 against various competitors. In Figure 6.11, we include a timing comparison[7] of these methods (Algorithms 7 and 9) with a standard block polynomial Krylov method [137], factored ADI [29], and `sylvester`, the built-in `Julia` implementation of the Bartels–Stewart algorithm [23] for problems of various size[8]. In the case of the block polynomial Krylov method, we implement the method as described in [137, Algorithm 5] with the method in [96, Equation 2.4] applied to cheaply monitor the residual. We do not implement any restarting strategies.

It is unsurprising that even when $\sigma(\mathbf{A})$ and $\sigma(\mathbf{B})$ are close, all of the inverse-free methods eventually outperform factored ADI and Bartels–Stewart as the size of $\mathbf{X}$ increases due to the dense matrices involved. We note that while the required number of iterations of the inverse-free methods grows as $\sigma(\mathbf{A})$ and $\sigma(\mathbf{B})$ approach each other, if Σ remains constant as

[7]The timing tests in this subsection were done on a Lenovo laptop running Linux Mint 22.1 Cinnamon with 16 cores and 62.2 GB of RAM with an Intel® Core™ Ultra 9 185H processor. All other computations in this chapter were performed using a single thread on a Lenovo laptop running Ubuntu version 20.04 with 8 cores and 16 GB of RAM with an Intel® Core™ i7-11800H processor running at 2.30 GHz.

[8]For all problems in this subsection, $\mathbf{A}$ and $\mathbf{B}$ are generated by randomly generating eigenvalues in the designated interval and conjugating by a random orthogonal matrix, and $\mathbf{C} = \mathbf{UV}$ corresponds to random $\mathbf{U}, \mathbf{V}$.

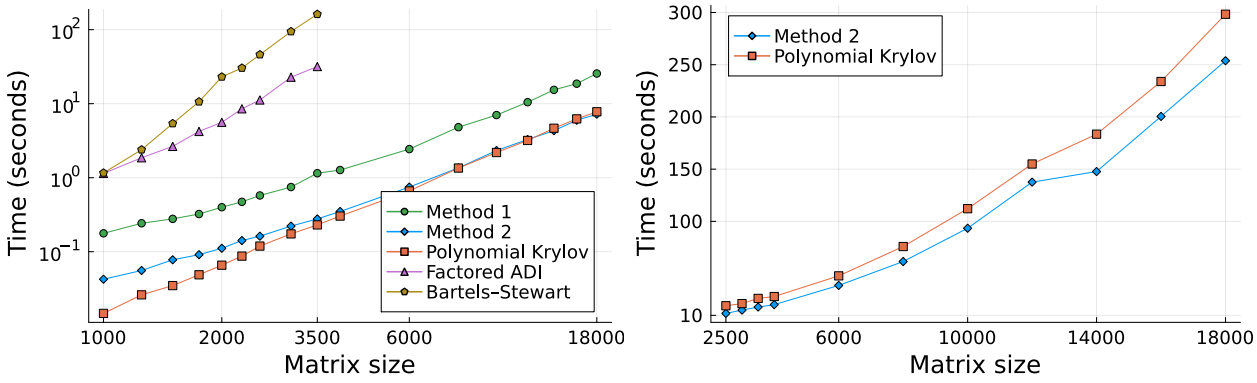


Figure 6.11: **Left:** Timing comparisons of Methods 1 and 2 against the factored ADI, Bartels–Stewart, and polynomial Krylov methods for solving (1.2). Both factored ADI and Methods 1 and 2 are run with a tolerance of $\epsilon = 2 \times 10^{-8}$ (left) while the polynomial Krylov method is run until the residual is less than 2×10^{-10} (left), which requires between 16 and 30 iterations. Here, $\mathbf{A}$ and $\mathbf{B}$ are both dense square matrices of varying size, with $\sigma(\mathbf{A}) \subset [0.5, 1.95]$, $\sigma(\mathbf{B}) \subset [-4, -2]$, and $\mathbf{C}$ is rank 1. Since the timing comparisons between Method 2 and the polynomial Krylov method are hard to parse in this figure, we include a table of timings for the three inverse-free methods with $n \geq 4000$ in Table 6.1. **Right:** Timings for Method 2 and the polynomial Krylov method, which are both substantially faster than the other considered methods, for a more challenging problem. Here, tolerance parameters are kept the same, but $\mathbf{C}$ is a rank 10 matrix, $\sigma(\mathbf{A}) \subset [.5, 1.95]$, and $\sigma(\mathbf{B}) \subset [-40, -2]$. Convergence is slower for both methods (between 62 and 80 iterations per run for both methods), and the cost per iteration is greater due to the higher-rank right-hand side.

the matrix size increases, there will always be a threshold past which inverse-free methods are advantageous.

As shown in Figure 6.11 and Table 6.1, for large problems where convergence is relatively rapid (e.g., < 50 iterations), Method 2 matches or slightly outperforms the polynomial Krylov method. For problems involving higher-rank right-hand sides and eigenvalue distributions that slow the convergence of iterative methods, Method 2 clearly performs better than all other methods. However, the major advantage of both Methods 1 and 2 in either setting is that they require less memory relative to the polynomial Krylov method, with no specialized restarting strategies needed to achieve these savings. Moreover, there is no need to monitor the residual since we can precompute the number of iterations required using our heuristics and bounds on the solution error norm. We also observe that while Method 1 appears to require the least memory (see Figure 6.1), it requires a substantially higher runtime than the other inverse-free methods.

In Figure 6.12, we compare timings for the two fastest inverse-free methods on a fixed problem size (1500×1500) with varying eigenvalue distributions. We observe that when one of the intervals containing eigenvalues of $\mathbf{A}$ and $\mathbf{B}$ is much larger than the other, Method 2 (Algorithm 9) outperforms the polynomial Krylov method. In this setting, the number of iterations required by the polynomial Krylov method is only slightly less than the iterations required for Method 2 (e.g., 161 vs. 169 when $a = 0.1$). Since the polynomial Krylov method has a cost that grows as the number of iterations grows, it can be slower than Method 2 when many iterations are required. If instead the two intervals are symmetric and small, the polynomial Krylov method converges more rapidly than Method 2 (e.g., 36 vs. 49 iterations when $a = 0.1$) and is also more efficient in terms of runtime.

While there are some regimes where Method 2 is faster than a basic polynomial Krylov method, these experiments can be summarized as follows: Method 2 can essentially match the performance of a basic polynomial Krylov method while requiring far less memory (see Figure 6.1) and no special restarting strategies.

6.4.2 Integral equations

Given Hölder continuous, symmetric kernels $K_1, K_2 : [-1, 1]^2 \to \mathbb{R}$ and continuous functions $f_1, \ldots, f_r, g_1, \ldots, g_r : [-1, 1] \to \mathbb{R}$, we consider the integral equation

$$2u(x, y) + \int_{-1}^{1} K_1(x, x')u(x', y)\mathrm{d}x' + \int_{-1}^{1} K_2(y, y')u(x, y')\mathrm{d}y' = \sum_{\ell=1}^{r} f_\ell(x)g_\ell(y), \tag{6.19}$$

for the unknown $u : [-1, 1]^2 \to \mathbb{R}$. We assume that the $L^2([-1, 1])$ eigenvalues of the operators defined by K_1, K_2 lie in the interval $(-1 + \delta, \infty)$ for some $\delta > 0$. Denote the Gauss–Legendre quadrature nodes and weights for n sample points by $\{x_j\}_{j=1}^n$ and $\{w_j\}_{j=1}^n$, respectively. Applying this quadrature rule to each integral in (6.19) and collocating at the

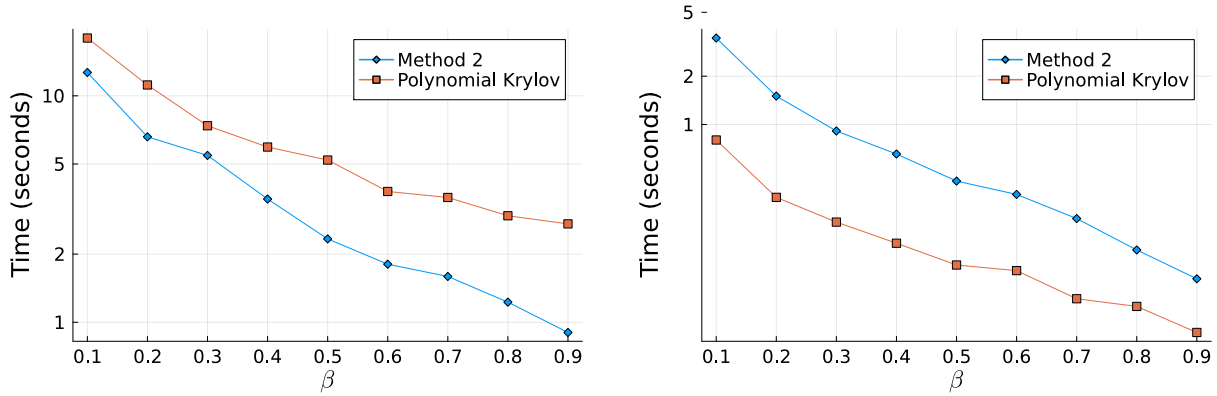


Figure 6.12: Timing comparisons of Method 2 against a block polynomial Krylov method [137, Algorithm 5] for solving (1.2) where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{1500\times 1500}$ and $\mathbf{C}$ is rank 5. Here, Method 2 (Algorithm 9) is run with a tolerance of $\epsilon = 2 \times 10^{-9}$, while the polynomial Krylov method is run until the residual is less than 2×10^{-10}. The parameter a corresponds to the distribution of the eigenvalues of $\mathbf{A}, \mathbf{B}$, with $\sigma(\mathbf{A}) \subset [a, 1]$, and $\sigma(\mathbf{B}) \subset [-20, -a]$ (left) and $\sigma(\mathbf{B}) \subset [-1, -a]$ (right). The latter case enables much faster convergence for both methods.

points $\{x_j\}_{j=1}^n$ in both x and y yields the Sylvester equation

$$(\mathbf{I} + \mathbf{K}_1)\mathbf{U} + \mathbf{U}(\mathbf{I} + \mathbf{K}_2) = \sum_{\ell=1}^{r} \mathbf{f}_\ell \mathbf{g}_\ell^T, \tag{6.20}$$

where $\mathbf{K}_1, \mathbf{K}_2, \mathbf{U} \in \mathbb{R}^{n\times n}$ and $\mathbf{f}_\ell, \mathbf{g}_\ell \in \mathbb{R}^n$ have entries

$$\begin{aligned} (\mathbf{K}_{1,2})_{j,k} &= \sqrt{w_j w_k} K_{1,2}(x_j, x_k), \quad \mathbf{U}_{j,k} \approx \sqrt{w_j w_k} u(x_j, x_k), \\ (\mathbf{f}_\ell)_j &= \sqrt{w_j} f_\ell(x_j), \quad (\mathbf{g}_\ell)_j = \sqrt{w_j} g_\ell(x_j), \quad \ell = 1, \ldots, r. \end{aligned} \tag{6.21}$$

The eigenvalues of $\mathbf{I} + \mathbf{K}_{1,2}$ are larger than δ and less than[9] $1 + \|\mathbf{K}_{1,2}\|_2$, which remains bounded as n increases. Because $\sum_{\ell=1}^r \mathbf{f}_\ell \mathbf{g}_\ell^T$ has rank at most r, Method 2 can be applied as in Algorithm 9. The efficiency of this method allows for such integral equations to be solved on fine grids. In Figure 6.13, we plot both an approximate solution to (6.19) with $K_1(x, y) = K_2(x, y) = \exp(-2|x - y|)$, $r = 1$, $f_1(x) = \frac{\cos(4x)}{1.04 - x^2}$, and $g_1(x) = \sin(20x)$ at $n = 2000$ gridpoints and the error in (6.20) at each iteration of Method 2.

As an extension, given additional rank 1 kernel functions[10] $K_3, K_4 : [-1, 1]^2 \to \mathbb{R}$, consider

[9] An upper bound can be more cheaply computed by instead using the Frobenius norm.

[10] This is a convenient but unnecessary assumption. To apply Method 2 in the manner described, at least one of the resulting matrices $\mathbf{K}_3$ and $\mathbf{K}_4$ must be representable in low-rank form. In this case, the number

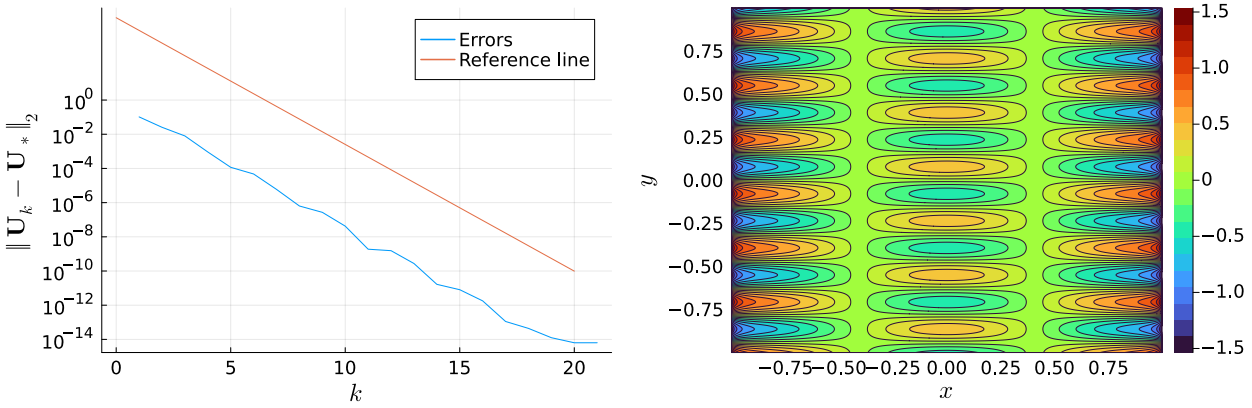


Figure 6.13: 2-norm error for $\mathbf{U}_k$ against the true solution $\mathbf{U}_*$ (computed via `sylvester` in `Julia`) at each iteration of Method 2 (Algorithm 9) in solving (6.20) (left) and corresponding approximate solution of (6.19) at $n = 2000$ gridpoints with $K_1(x, y) = K_2(x, y) = \exp(-2|x-y|)$, $f_1(x) = \frac{\cos(4x)}{1.04-x^2}$, and $g_1(x) = \sin(20x)$ (right). Here, the reference line denotes the convergence rate of Corollary 6.3.3 with $D_{\mathbf{A},\mathbf{B},\mathbf{C}}$ replaced by the hypothesized upper bound $20(n + m)$.

the integral equation

$$\begin{aligned} &2u(x,y) + \int_{-1}^{1} K_1(x,x')u(x',y)\mathrm{d}x' + \int_{-1}^{1} K_2(y,y')u(x,y')\mathrm{d}y' \\ &+ \int_{-1}^{1}\int_{-1}^{1} K_3(x,x')u(x',y')K_4(y,y')\mathrm{d}x'\mathrm{d}y' = \sum_{\ell=1}^{r} f_\ell(x)g_\ell(y). \end{aligned} \tag{6.22}$$

Applying Gauss–Legendre quadrature to (6.22) in the same manner yields the generalized Sylvester equation

$$(\mathbf{I} + \mathbf{K}_1)\mathbf{U} + \mathbf{U}(\mathbf{I} + \mathbf{K}_2) + \mathbf{K}_3\mathbf{U}\mathbf{K}_4^T = \sum_{\ell=1}^{r} \mathbf{f}_\ell\mathbf{g}_\ell^T, \tag{6.23}$$

where $\mathbf{K}_3, \mathbf{K}_4 \in \mathbb{R}^{n\times n}$ have entries

$$(\mathbf{K}_{3,4})_{j,k} = \sqrt{w_j w_k}K_{3,4}(x_j, x_k),$$

and the other matrices are defined as in (6.21). To see how Method 2 can be used to solve

of required GMRES iterations is one plus the rank of $\mathbf{K}_4 \otimes \mathbf{K}_3$.

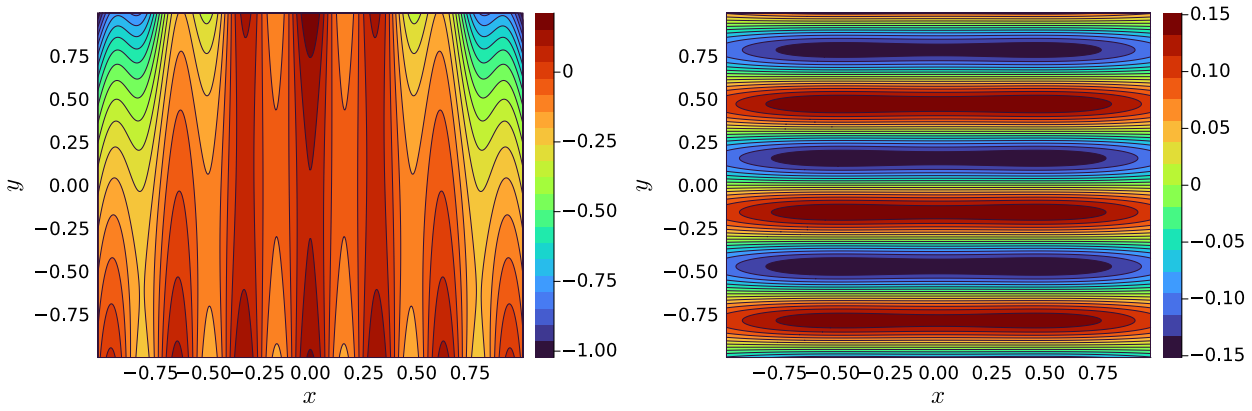


Figure 6.14: Approximate solution of (6.22) at $n = 2000$ gridpoints with $K_1(x,y) = K_2(x,y) = \exp(-2|x-y|)$, $K_3(x,y) = \cos(20x)\exp(y)$, $K_4(x,y) = \cosh(x)\sinh(y)$, $f_1(x) = x^2$, and $g_1(x) = -\exp(x)$ (left) and $K_1(x,y) = K_2(x,y) = \exp(-(x-y)^2)$, $K_3(x,y) = y\,\mathrm{sech}^2(x)$, $K_4(x,y) = \exp(x-y)$, $f_1(x) = \frac{1}{x^4+2}$, and $g_1(x) = -\sin(10x)$ (right). In both cases, preconditioned GMRES applied to (6.23) converges in at most two iterations.

(6.23), denote the solution to the Sylvester equation (1.2) by the linear map

$$\mathcal{S}^{-1}_{\mathbf{A},\mathbf{B}}\mathbf{C} = \mathcal{T}_{\mathbf{A},\mathbf{B}}\mathbf{C} := \mathbf{X}.$$

Then, (6.23) is equivalent to the linear system

$$\left(\mathbf{I} + \mathcal{T}_{(\mathbf{I}+\mathbf{K}_1),(\mathbf{I}+\mathbf{K}_2)}\right)\mathbf{K}_3\mathbf{U}\mathbf{K}_4^T = \mathcal{T}_{(\mathbf{I}+\mathbf{K}_1),(\mathbf{I}+\mathbf{K}_2)}\sum_{\ell=1}^{r}\mathbf{f}_\ell\mathbf{g}_\ell^T. \tag{6.24}$$

Using Method 2 to compute the action of $\mathcal{T}_{(\mathbf{I}+\mathbf{K}_1),(\mathbf{I}+\mathbf{K}_2)}$, matrix-free GMRES or another iterative method can be used to solve (6.24). Furthermore, since this system is identity plus rank 1, GMRES will converge in at most two iterations. We include two solutions computed in this manner in Figure 6.14.

6.5 Other applications

6.5.1 Algebraic Riccati equations

An algebraic Riccati equation is a generalization of the Sylvester equation of the form

$$\mathbf{A}\mathbf{X} - \mathbf{X}\mathbf{B} - \mathbf{X}\mathbf{D}\mathbf{X} = \mathbf{C}. \tag{6.25}$$

Under certain conditions on the data matrices, a solution to (6.25) can be obtained by computing the sign of the matrix [35]

$$\mathbf{H} = \begin{pmatrix} \mathbf{A} & \mathbf{D} \\ \mathbf{C} & \mathbf{B} \end{pmatrix}.$$

If the eigenvalues of $\mathbf{H}$ are known to lie on or near a collection of disjoint intervals Σ, solving the Riccati equation (6.25) is a direct application of the Akhiezer iteration, albeit without the decoupled and low-rank structure of the Sylvester equation algorithms. However, if only the eigenvalues of the individual data matrices are known, little can be said in general about the eigenvalues of $\mathbf{H}$. Particular cases when the eigenvalues of $\mathbf{H}$ are easily determined are when $\mathbf{C} = \mathbf{0}$ and when $\mathbf{D} = \mathbf{0}$. The latter case reduces precisely to the Sylvester equation (1.2). The former case has the trivial solution $\mathbf{X} = \mathbf{0}$. If an invertible solution $\mathbf{X}$ exists, (6.25) is equivalent to

$$\mathbf{X}^{-1}\mathbf{A} - \mathbf{B}\mathbf{X}^{-1} - \mathbf{D} = 0,$$

which is simply a Sylvester equation for $\mathbf{X}^{-1}$. Thus, while Method 1 can, in principle, be generalized to algebraic Riccati equations, knowledge of the eigenvalues of the full block matrix $\mathbf{H}$ is required to compute anything nontrivial. If the eigenvalues of $\mathbf{H}$ are suspected to lie on a collection of disjoint intervals, the adaptive procedure described in [19, Section 5] could be employed to approximate these intervals, then the Akhiezer iteration could be applied directly to compute sign($\mathbf{H}$) and solve (6.25).

6.5.2 Fréchet derivatives

A natural application of this work is to the computation of Fréchet derivatives. Given a matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$ and matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$, the Fréchet derivative of f at $\mathbf{A}$ is a linear map $L_f(\mathbf{A}, \diamond)$ satisfying

$$f(\mathbf{A} + \mathbf{E}) - f(\mathbf{A}) - L_f(\mathbf{A}, \mathbf{E}) = \mathrm{o}\left(\|\mathbf{E}\|\right),$$

for all $\mathbf{E} \in \mathbb{C}^{n\times n}$. Fréchet derivatives need not exist, but must be unique if they do [86, Section 3.1]. By modifying [111, Theorem 2.1] and its proof to use lower triangular matrices, if p denotes the size of the largest Jordan block of $\mathbf{A}$ and f is $2p-1$ times continuously differentiable on an open set containing the spectrum of $\mathbf{A}$, then for any $\mathbf{E} \in \mathbb{C}^{n\times n}$,

$$f\begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{E} & \mathbf{A} \end{pmatrix} = \begin{pmatrix} f(\mathbf{A}) & \mathbf{0} \\ L_f(\mathbf{A}, \mathbf{E}) & f(\mathbf{A}) \end{pmatrix}.$$

Since computing the Fréchet derivative requires only the lower left block of this matrix function, the recurrences utilized in Method 1 can be applied nearly directly with f in place of the sign function; only the series coefficients α_j need be modified appropriately. In Figure 6.15, we plot the error at each iteration of computing the Fréchet derivative for the

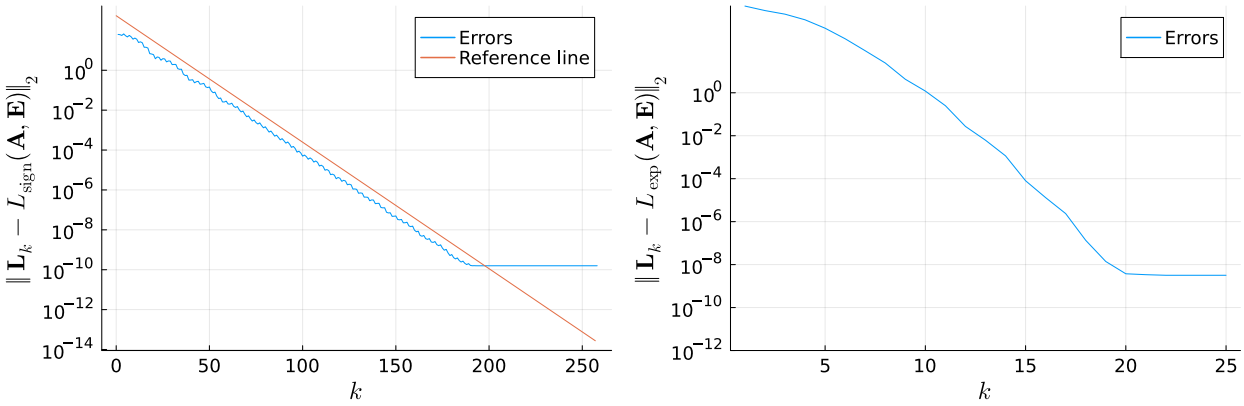


Figure 6.15: 2-norm error for approximation $\mathbf{L}_k$ to $L_f(\mathbf{A}, \mathbf{E})$ at each iteration of Algorithm 7 modified to compute Fréchet derivatives for $f(x) = \text{sign}(x)$ (left) and $f(x) = \exp(x)$ (right). Here, $\mathbf{A} \in \mathbb{R}^{200\times 200}$ has eigenvalues in $[-2, -0.5] \cup [0.5, 6]$ and $\mathbf{E}$ is rank 4. Because the exponential function is entire, our algorithm achieves superexponential convergence before errors saturate. The true solution is computed via the Newton iteration [8, Section 5] (left) and the built-in `Julia` function `exp` applied to $\begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{E} & \mathbf{A} \end{pmatrix}$ (right). Here, the reference line denotes the convergence rate bound (6.12) with $D_{\mathbf{M}}$ replaced by the hypothesized upper bound 10ℓ for generic $\mathbf{M} \in \mathbb{C}^{\ell\times\ell}$.

matrix sign function and matrix exponential. We observe that the method converges quickly with the same convergence guarantees as when it is applied to Sylvester equations. The Fréchet derivative of the matrix sign function is useful for analyzing the stability of algorithms to compute the matrix sign, matrix square root, and other related matrix functions [92].

6.5.3 Matrix square roots and the polar decomposition

The matrix sign function can also be used to compute the matrix square root and inverse square root. In particular, if $\mathbf{A} \in \mathbb{C}^{n\times n}$ has no eigenvalues on $(-\infty, 0]$, then [86, Equation 6.32]

$$\text{sign}\begin{pmatrix} 0 & \mathbf{A} \\ \mathbf{I} & 0 \end{pmatrix} = \begin{pmatrix} 0 & \mathbf{A}^{1/2} \\ \mathbf{A}^{-1/2} & 0 \end{pmatrix}.$$

If $\mathbf{A}$ has eigenvalues contained in or near some interval $[a, b]$ on the positive real axis, the eigenvalues of $\begin{pmatrix} 0 & \mathbf{A} \\ \mathbf{I} & 0 \end{pmatrix}$ are in or near $[-\sqrt{b}, -\sqrt{a}] \cup [\sqrt{a}, \sqrt{b}]$, and the Akhiezer iteration can be applied to compute $\mathbf{A}^{1/2}$ and $\mathbf{A}^{-1/2}$ in parallel. The off-diagonal structure of the relevant block matrix prevents the direct application of Lemma 6.2.3, and an analogous result for off-diagonal block 2×2 matrices would need to be worked out for this approach to be made

efficient. It is currently unclear whether this would be more efficient than directly applying the Akhiezer iteration to $\mathbf{A}$ to compute $\mathbf{A}^{1/2}$ and $\mathbf{A}^{-1/2}$ separately. Regardless, the Akhiezer iteration should produce an inverse-free method for computing the matrix square root and inverse square root; we leave this for future work.

Similarly, the matrix sign function is connected to the polar decomposition, which can be used to solve the orthogonal Procrustes problem [86, Theorem 8.6a]. If the polar decomposition of a matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ $(m \geq n)$ is given by $\mathbf{A} = \mathbf{U}\mathbf{H}$ where $\mathbf{U} \in \mathbb{C}^{m\times n}$ has orthonormal columns and $\mathbf{H} \in \mathbb{C}^{n\times n}$ is Hermitian and positive semidefinite, then [86, Equation 8.6]

$$\operatorname{sign}\begin{pmatrix} 0 & \mathbf{A} \\ \mathbf{A}^* & 0 \end{pmatrix} = \begin{pmatrix} 0 & \mathbf{U} \\ \mathbf{U}^* & 0 \end{pmatrix}.$$

Thus, the Akhiezer iteration can be applied to compute the polar factor $\mathbf{U}$ of the polar decomposition, in particular when the eigenvalues of $\mathbf{A}^*\mathbf{A}$ are located in or near a known interval on the real axis. In the special case where $\mathbf{A}$ is symmetric, an algorithm is immediate as $\operatorname{sign}(\mathbf{A}) = \mathbf{U}$. Otherwise, an off-diagonal analog to Lemma 6.2.3 would likely be necessary for this approach to be made efficient. Another option is to compute the polar factor by applying the Akhiezer iteration to the inverse square root with the formula $\mathbf{U} = \mathbf{A}(\mathbf{A}^*\mathbf{A})^{-1/2}$ [87, Chapter 8].

The Fréchet derivative of the polar decomposition can be obtained via the identity [75]

$$\operatorname{sign}\begin{pmatrix} 0 & \mathbf{A} & 0 & \mathbf{E} \\ \mathbf{A}^* & 0 & \mathbf{E}^* & 0 \\ 0 & 0 & 0 & \mathbf{A} \\ 0 & 0 & \mathbf{A}^* & 0 \end{pmatrix} = \begin{pmatrix} 0 & \mathcal{P}(\mathbf{A}) & 0 & L_{\mathcal{P}}(\mathbf{A},\mathbf{E}) \\ \mathcal{P}(\mathbf{A})^* & 0 & L_{\mathcal{P}}(\mathbf{A},\mathbf{E})^* & 0 \\ 0 & 0 & 0 & \mathcal{P}(\mathbf{A}) \\ 0 & 0 & \mathcal{P}(\mathbf{A})^* & 0 \end{pmatrix}.$$

Here, $\mathcal{P}$ denotes a function that maps a matrix to its polar factor and $\mathbf{A}, \mathbf{E} \in \mathbb{C}^{n\times n}$. The Akhiezer iteration can again be applied to compute $L_{\mathcal{P}}(\mathbf{A}, \mathbf{E})$, in particular when the eigenvalues $\mathbf{A}^*\mathbf{A}$ are located in or near a known interval on the real axis since the eigenvalues of the relevant block matrix are given by both square roots of these eigenvalues. Once again, an analog of Lemma 6.2.3 for 4×4 block matrices with the relevant sparsity pattern is likely needed for efficiency, but this would yield an inverse-free method that, in contrast with that of [75], would not require that all singular values of $\mathbf{A}$ be small.

6.6 Discussion and future work

Perhaps the largest question that we have not addressed is that of adaptivity. Throughout this chapter, we have assumed access to intervals on the real axis that (approximately) contain the spectra of the coefficient matrices $\mathbf{A}$ and $\mathbf{B}$. If these intervals are instead unknown, the adaptive approach outlined in Chapter 5, Section 5.4, can be applied to the block matrix $\mathbf{H}$ from (6.8) to approximate the necessary intervals. Alternatively, since the problem of finding the eigenvalues of $\mathbf{H}$ decouples, the same method with a single interval and shifted

and scaled Chebyshev polynomials could be applied to each coefficient matrix individually. In the low-rank case of Method 1 (Algorithm 7), one could also consider performing this analysis with the norms of the iterates $\mathbf{J}_k$, $\mathbf{K}_k$ as empirically, these appear to grow at the same rate as the norm of $\mathbf{H}$.

A more difficult generalization would be allowing the coefficient matrices to have nonreal eigenvalues. For instance, $\mathbf{A}$ and $\mathbf{B}$ could have eigenvalues contained in disjoint open sets in the complex plane. If these sets are well-separated, there potentially exists a pair of intervals on the real axis, or a pair of lines in the complex plane, that enable Method 1 to converge in a reasonable number of iterations. Such a result exists for the related Chebyshev iteration and relies on the notion of an "optimal ellipse" [105, 106]. Extending this notion to higher genus cut domains is a challenging open problem. Solving it likely requires a complete understanding of the generalized Bernstein ellipses of Figure 6.3. Such a generalization could, for instance, enable an inverse-free method for the matrix-sign approach to eigenvalue computation when used in place of the Newton iteration in [22, Section 4].

For symmetric matrices, $\operatorname{sign}(\mathbf{M})$ is equal to the polar factor for $\mathbf{M}$. The Akhiezer polynomials could be applied in the context of divide-and-conquer eigensolvers based on spectral projections attained via the polar factor [115]. An inverse-free polar decomposition method based on Halley's iteration was developed in [114], where inverse factors are rewritten using QR decompositions of a block matrix related to the iterate. However, the Akhiezer polynomials should lead to an eigensolver with the polar factorization step only requiring matrix products. Akhiezer polynomials could also potentially provide a cheaper alternative to the minimax polynomials in the recent Polar Express algorithm [9].

We have not attempted to implement methods involving hierarchical numerical linear algebra, but a natural connection to consider is the setting where $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are rank-structured. As shown in [97], it is often the case that $\mathbf{X}$ is well approximated in this setting by a rank-structured matrix. Combining the Akhiezer iteration with the divide-and-conquer scheme in [97] would take advantage of fast multipole-based matrix-vector products available for $\mathbf{A}$ and $\mathbf{B}$ and potentially supply an inverse-free version of the algorithms.

6.7 Bivariate matrix functions

This section is based on in-progress work with Thomas Trogdon and Heather Wilber.

As demonstrated in Figure 6.1, both methods introduced in this chapter achieve slower convergence rates than the "worst case" error bound of a polynomial Krylov method. This is because a polynomial Krylov method constructs a bivariate approximation, while the framework of Chapter 5 only allows for univariate approximation. In this section, we develop an extension of the Akhiezer iteration to bivariate matrix functions.

A result of Kreĭn [93] is that when the Sylvester equation (1.2) is uniquely solvable, the

solution is given by [48, Equation 3.10]

$$\mathbf{X} = -\frac{1}{4\pi^2} \oint_{\Gamma_\mathbf{A}} \oint_{\Gamma_\mathbf{B}} \frac{(x\mathbf{I} - \mathbf{B})^{-1}\mathbf{C}\,(y\mathbf{I} - \mathbf{A})^{-1}}{y - x}\,\mathrm{d}x\,\mathrm{d}y,$$

where $\Gamma_\mathbf{A}$ and $\Gamma_\mathbf{B}$ are counterclockwise-oriented simple curves enclosing the spectra of $\mathbf{A}$ and $\mathbf{B}$, respectively, and $\Gamma_\mathbf{A} \cap \Gamma_\mathbf{B} = \emptyset$. More generally, given a uniquely solvable[11] linear matrix equation of the form

$$\sum_{j,k=0}^{n} c_{jk}\mathbf{B}^j\mathbf{X}\mathbf{A}^k = \mathbf{C}, \tag{6.26}$$

where each c_{jk} is a complex number, then [48, Theorem 3.2]

$$\mathbf{X} = -\frac{1}{4\pi^2} \oint_{\Gamma_\mathbf{A}} \oint_{\Gamma_\mathbf{B}} \frac{(x\mathbf{I} - \mathbf{B})^{-1}\mathbf{C}\,(y\mathbf{I} - \mathbf{A})^{-1}}{P(x,y)}\,\mathrm{d}x\,\mathrm{d}y,$$

where

$$P(x,y) = \sum_{j,k=0}^{n} c_{jk}x^j y^k.$$

This motivates Kressner's definition[12] of a *bivariate matrix function*.

Definition 6.7.1 (Kressner [95, Theorem 2.8])**.** *Let $\mathbf{A} \in \mathbb{C}^{n\times n}$ and $\mathbf{B} \in \mathbb{C}^{m\times m}$, let $\Omega_\mathbf{A}, \Omega_\mathbf{B} \subset \mathbb{C}$ be open sets containing $\sigma(\mathbf{A})$ and $\sigma(\mathbf{B})$, respectively, and let $f(x,y)$ be analytic in $\Omega_\mathbf{B} \times \Omega_\mathbf{A}$ and continuous on $\mathrm{cl}\,(\Omega_\mathbf{B} \times \Omega_\mathbf{A})$. The associated bivariate matrix function $f\{\mathbf{B},\mathbf{A}\} : \mathbb{C}^{m\times n} \to \mathbb{C}^{m\times n}$ is defined by*

$$f\{\mathbf{B},\mathbf{A}\}(\mathbf{C}) = -\frac{1}{4\pi^2} \oint_{\Gamma_\mathbf{A}} \oint_{\Gamma_\mathbf{B}} f(x,y)(x\mathbf{I} - \mathbf{B})^{-1}\mathbf{C}(y\mathbf{I} - \mathbf{A})^{-1}\,\mathrm{d}x\,\mathrm{d}y,$$

where $\Gamma_\mathbf{A}$ and $\Gamma_\mathbf{B}$ are counterclockwise-oriented simple curves enclosing $\sigma(\mathbf{A})$ and $\sigma(\mathbf{B})$ in $\Omega_\mathbf{A}$ and $\Omega_\mathbf{B}$, respectively.

With this definition, (1.2) is solved by $\mathbf{X} = f\{\mathbf{B},\mathbf{A}\}(\mathbf{C})$ for $f(x,y) = \frac{1}{y-x}$, and (6.26) is solved by $\mathbf{X} = f\{\mathbf{B},\mathbf{A}\}(\mathbf{C})$ for $f(x,y) = \frac{1}{P(x,y)}$. The Fréchet derivative of an analytic function g is likewise given by $f\{\mathbf{A},\mathbf{A}\}(\mathbf{E}) = L_g(\mathbf{A},\mathbf{E})$ where f is the divided difference [95, Theorem 5.1]

$$f(x,y) = \begin{cases} \dfrac{g(x) - g(y)}{x - y}, & x \neq y, \\ g'(x), & x = y. \end{cases} \tag{6.27}$$

[11] A matrix equation of this form is uniquely solvable if $P(x,y) \neq 0$ for all $(x,y) \in \sigma(\mathbf{B}) \times \sigma(\mathbf{A})$.

[12] The definition here differs from Kressner's by a transpose. As noted in [95], this is purely a matter of convention.

The key idea behind a bivariate Akhiezer iteration is to construct a low-rank approximation at the scalar level. That is, if we wish to compute $f\{\mathbf{B}, \mathbf{A}\}(\mathbf{C})$, we construct an approximation

$$f(x, y) \approx \sum_{\ell=1}^{r} \phi_\ell(x)\psi_\ell(y), \tag{6.28}$$

where each ϕ_ℓ and ψ_ℓ is an appropriately analytic[13] univariate function. If such an approximation is accurate to machine precision for a relatively small r, we say that the function is *numerically of low rank*. See [149, Chapter 3] for a discussion of when functions possess this property.

Given a low-rank approximation to f of the form (6.28), Definition 6.7.1 decouples into univariate matrix functions

$$f\{\mathbf{B}, \mathbf{A}\}(\mathbf{C}) \approx \sum_{\ell=1}^{r} \phi_\ell(\mathbf{B})\mathbf{C}\psi_\ell(\mathbf{A}). \tag{6.29}$$

When $\mathbf{C} = \mathbf{UV}$ is low rank, (6.29) immediately yields an approximation in low-rank form:

$$f\{\mathbf{B}, \mathbf{A}\}(\mathbf{C}) \approx \sum_{\ell=1}^{r} \phi_\ell(\mathbf{B})\mathbf{UV}\psi_\ell(\mathbf{A}) = \begin{pmatrix}\phi_1(\mathbf{B})\mathbf{U} & \cdots & \phi_r(\mathbf{B})\mathbf{U}\end{pmatrix} \begin{pmatrix}\mathbf{V}\psi_1(\mathbf{A}) \\ \vdots \\ \mathbf{V}\psi_r(\mathbf{A})\end{pmatrix}. \tag{6.30}$$

Once such an approximation is obtained, the Akhiezer iteration applies directly to each block entry in (6.30). As before, assume that both $\mathbf{A}$ and $\mathbf{B}$ have real spectra contained in disjoint subintervals. That is, $\Sigma_{\mathbf{B}} \supseteq \sigma(\mathbf{B})$ and $\Sigma_{\mathbf{A}} \supseteq \sigma(\mathbf{A})$ are unions of intervals of the form (1.1). Let $w_{\mathbf{B}}, w_{\mathbf{A}}$ denote weight functions of the form (3.5) and denote their corresponding orthonormal polynomials by $\left(p_k^{\mathbf{B}}\right)_{k=0}^{\infty}$ and $\left(p_k^{\mathbf{A}}\right)_{k=0}^{\infty}$, respectively. Let $\Gamma'_{\mathbf{B}}$ and $\Gamma'_{\mathbf{A}}$ be contours enclosing $\Sigma_{\mathbf{B}}$ and $\Sigma_{\mathbf{A}}$ that lie within the regions of analyticity of all factors ϕ_ℓ and ψ_ℓ, respectively. We denote these contours with a prime to distinguish them from the contours in Definition 6.7.1, as the domains of analyticity for the factors need not align exactly with those of f. We equip them with quadrature nodes $\{x_i\} \subset \Gamma'_{\mathbf{B}}$ and $\{y_j\} \subset \Gamma'_{\mathbf{A}}$ and quadrature weights $\left\{\omega_i^{\mathbf{B}}\right\}$ and $\left\{\omega_j^{\mathbf{A}}\right\}$. As before, it typically suffices to take each contour to be a union of circles around the constituent intervals and apply the trapezoidal rule.

Given the univariate functions ϕ_ℓ and ψ_ℓ, their series coefficients are computed in precisely the same manner as in the univariate Akhiezer iteration of Chapter 5, and this computation only requires sampling ϕ_ℓ and ψ_ℓ at quadrature points. Writing

$$\phi_\ell(x) = \sum_{k=0}^{\infty} \beta_{\ell,k}\, p_k^{\mathbf{B}}(x), \qquad \psi_\ell(y) = \sum_{k=0}^{\infty} \gamma_{\ell,k}\, p_k^{\mathbf{A}}(y), \tag{6.31}$$

[13]By this, we mean that each ϕ_ℓ is analytic in a region containing $\sigma(\mathbf{B})$, and each ψ_ℓ is analytic in a region containing $\sigma(\mathbf{A})$.

the coefficients are the weighted inner products

$$\beta_{\ell,k} = \int_{\Sigma_{\mathbf{B}}} \phi_\ell(x)\, p_k^{\mathbf{B}}(x)\, w_{\mathbf{B}}(x)\, \mathrm{d}x, \qquad \gamma_{\ell,k} = \int_{\Sigma_{\mathbf{A}}} \psi_\ell(y)\, p_k^{\mathbf{A}}(y)\, w_{\mathbf{A}}(y)\, \mathrm{d}y.$$

Applying the Cauchy integral formula to ϕ_ℓ and ψ_ℓ,

$$\beta_{\ell,k} = -\oint_{\Gamma'_{\mathbf{B}}} \phi_\ell(z)\, \mathcal{C}_{\Sigma_{\mathbf{B}}}\big[p_k^{\mathbf{B}} w_{\mathbf{B}}\big](z)\, \mathrm{d}z, \qquad \gamma_{\ell,k} = -\oint_{\Gamma'_{\mathbf{A}}} \psi_\ell(z)\, \mathcal{C}_{\Sigma_{\mathbf{A}}}\big[p_k^{\mathbf{A}} w_{\mathbf{A}}\big](z)\, \mathrm{d}z.$$

where $\Gamma'_{\mathbf{B}}$ (resp. $\Gamma'_{\mathbf{A}}$) encloses $\sigma(\mathbf{B})$ (resp. $\sigma(\mathbf{A})$), and each ϕ_ℓ (resp. ψ_ℓ) is analytic in a region containing its interior. Applying a quadrature rule to each gives the approximations

$$\beta_{\ell,k} \approx -\sum_i \phi_\ell(x_i)\, \omega_i^{\mathbf{B}}\, \mathcal{C}_{\Sigma_{\mathbf{B}}}\big[p_k^{\mathbf{B}} w_{\mathbf{B}}\big](x_i), \qquad \gamma_{\ell,k} \approx -\sum_j \psi_\ell(y_j)\, \omega_j^{\mathbf{A}}\, \mathcal{C}_{\Sigma_{\mathbf{A}}}\big[p_k^{\mathbf{A}} w_{\mathbf{A}}\big](y_j). \quad (6.32)$$

Given these (approximate) series coefficients, we assemble the factors of (6.30) via the series

$$\phi_\ell(\mathbf{B})\mathbf{U} = \sum_{k=0}^{\infty} \beta_{\ell,k}\, p_k^{\mathbf{B}}(\mathbf{B})\mathbf{U}, \qquad \mathbf{V}\psi_\ell(\mathbf{A}) = \sum_{k=0}^{\infty} \gamma_{\ell,k}\, \mathbf{V} p_k^{\mathbf{A}}(\mathbf{A}).$$

The matrix polynomials $\mathbf{L}_k := p_k^{\mathbf{B}}(\mathbf{B})\mathbf{U}$ and $\mathbf{R}_k := \mathbf{V}p_k^{\mathbf{A}}(\mathbf{A})$ satisfy the three-term recurrences (2.5) of their respective families and are generated once, shared across all r factors. Accumulating the truncated sums produces the blocks $\mathbf{W}^{(\ell)} \approx \phi_\ell(\mathbf{B})\mathbf{U}$ and $\mathbf{Z}^{(\ell)} \approx \mathbf{V}\psi_\ell(\mathbf{A})$, whose concatenations $\mathbf{W} = \begin{pmatrix}\mathbf{W}^{(1)} & \cdots & \mathbf{W}^{(r)}\end{pmatrix}$ and $\mathbf{Z} = \begin{pmatrix}\mathbf{Z}^{(1)} \\ \vdots \\ \mathbf{Z}^{(r)}\end{pmatrix}$ form the low-rank solution $\mathbf{WZ} \approx \mathbf{X}$. If $\mathbf{U} \in \mathbb{C}^{m\times s}$ and $\mathbf{V} \in \mathbb{C}^{s\times n}$, each block $\mathbf{W}^{(\ell)}$ is $m \times s$, so $\mathbf{W} \in \mathbb{C}^{m\times rs}$ and $\mathbf{Z} \in \mathbb{C}^{rs\times n}$. In contrast to Methods 1 and 2, the factor dimensions are fixed at rs throughout the iteration, so no compression is needed to control storage, though the final output may be compressed to its numerical rank if desired. An implementation of this procedure is given as Algorithm 10.

It remains to construct the low-rank approximation (6.28) numerically. Methods to construct two-dimensional low-rank function approximations, including the SVD and continuous Gaussian elimination, are surveyed in [149]. As noted above, to compute the Akhiezer data needed for Algorithm 10, one only needs to evaluate the functions ϕ_ℓ and ψ_ℓ at quadrature points. Thus, it suffices to approximate f in low-rank form on the grid $\{x_i\}\times\{y_j\}$. Sampling f there yields the matrix $\mathbf{F}$ with entries $\mathbf{F}_{ij} = f(x_i, y_j)$. Adaptive cross approximation [24] can be applied to $\mathbf{F}$ to generate the desired low-rank approximation. An implementation of the complete-pivoting LU variant is given as Algorithm 11. This generates a low-rank approximation $\mathbf{F} \approx \hat{\mathbf{C}}\hat{\mathbf{D}}\hat{\mathbf{R}}$ where the factors $\hat{\mathbf{C}}\hat{\mathbf{D}}^{1/2}$ and $\hat{\mathbf{D}}^{1/2}\hat{\mathbf{R}}$ contain the desired information. The columns of $\hat{\mathbf{C}}\hat{\mathbf{D}}^{1/2}$ are the samples $(\phi_\ell(x_i))_i$, and the rows of $\hat{\mathbf{D}}^{1/2}\hat{\mathbf{R}}$ are the

Algorithm 10: Bivariate Akhiezer iteration for $\mathbf{X} = f\{\mathbf{B}, \mathbf{A}\}(\mathbf{UV})$

Input: Matrices $\mathbf{A}, \mathbf{B}, \mathbf{U}, \mathbf{V}$ and the Akhiezer data for $\Sigma_{\mathbf{A}}$ and $\Sigma_{\mathbf{B}}$ (functions to compute recurrence coefficients $\check{a}_k^{\mathbf{A}}, \check{b}_k^{\mathbf{A}}, \check{a}_k^{\mathbf{B}}, \check{b}_k^{\mathbf{B}}$ and series coefficients $\beta_{\ell,k}, \gamma_{\ell,k}$ as in (6.31)).

Initialize $\mathbf{W}^{(\ell)} = \mathbf{0}$ and $\mathbf{Z}^{(\ell)} = \mathbf{0}$ for $\ell = 1, \ldots, r$.

for *k=0,1,...* **do**

  if *k=0* **then**

    Set $\mathbf{L}_0 = \mathbf{U}$ and $\mathbf{R}_0 = \mathbf{V}$.

  else if *k=1* **then**

    Set $\mathbf{L}_1 = \frac{1}{\check{b}_0^{\mathbf{B}}}(\mathbf{B}\mathbf{L}_0 - \check{a}_0^{\mathbf{B}}\mathbf{L}_0)$ and $\mathbf{R}_1 = \frac{1}{\check{b}_0^{\mathbf{A}}}(\mathbf{R}_0\mathbf{A} - \check{a}_0^{\mathbf{A}}\mathbf{R}_0)$.

  else

    Set $\mathbf{L}_k = \frac{1}{\check{b}_{k-1}^{\mathbf{B}}}(\mathbf{B}\mathbf{L}_{k-1} - \check{a}_{k-1}^{\mathbf{B}}\mathbf{L}_{k-1} - \check{b}_{k-2}^{\mathbf{B}}\mathbf{L}_{k-2})$.

    Set $\mathbf{R}_k = \frac{1}{\check{b}_{k-1}^{\mathbf{A}}}(\mathbf{R}_{k-1}\mathbf{A} - \check{a}_{k-1}^{\mathbf{A}}\mathbf{R}_{k-1} - \check{b}_{k-2}^{\mathbf{A}}\mathbf{R}_{k-2})$.

  end

  Set $\mathbf{W}^{(\ell)} \leftarrow \mathbf{W}^{(\ell)} + \beta_{\ell,k}\mathbf{L}_k$ and $\mathbf{Z}^{(\ell)} \leftarrow \mathbf{Z}^{(\ell)} + \gamma_{\ell,k}\mathbf{R}_k$ for $\ell = 1, \ldots, r$.

  if converged **then**

    return $\mathbf{W} = \begin{pmatrix}\mathbf{W}^{(1)} & \cdots & \mathbf{W}^{(r)}\end{pmatrix}$, $\mathbf{Z} = \begin{pmatrix}\mathbf{Z}^{(1)} \\ \vdots \\ \mathbf{Z}^{(r)}\end{pmatrix}$.

  end

end

samples $(\psi_\ell(y_j))_j$. With these matrices in hand, the Akhiezer data are generated through the quadrature approximations (6.32).

Algorithm 11: Complete-pivoting LU cross approximation ($\mathbf{A} \approx \mathbf{CDR}$)

```
CPLU(A, k, ϵ)
Input: Matrix A ∈ ℂ^{m×n}, maximum rank k, and tolerance ϵ.
Set E ← A and let C, R, d be empty.
for j=1,2,...,k do
    Set (p, q) ← arg max_{p,q} |E_{p,q}|.
    Set C ← (C  E_{:,q}), R ← (R ; E_{p,:}), and d ← (d ; E_{p,q}^{-1}).
    Set E ← E − E_{:,q} E_{p,q}^{-1} E_{p,:}.
    if ‖E‖_∞ < ϵ then
        Set D ← diag(d).
        return C, D, R
    end
end
Set D ← diag(d).
return C, D, R
```

6.7.1 Convergence

In this subsection, we provide a brief convergence analysis of Algorithm 10 under the idealization that both the low-rank approximation (6.28) and the series coefficients $\beta_{\ell,k}, \gamma_{\ell,k}$ are exact. The convergence is best understood through the bivariate orthogonal polynomial series of f. Expanding f in such a series,

$$f(x,y) = \sum_{i,j=0}^{\infty} c_{ij}\, p_i^{\mathbf{B}}(x) p_j^{\mathbf{A}}(y), \qquad c_{ij} = \int_{\Sigma_{\mathbf{B}}} \int_{\Sigma_{\mathbf{A}}} f(x,y) p_i^{\mathbf{B}}(x) p_j^{\mathbf{A}}(y) w_{\mathbf{B}}(x) w_{\mathbf{A}}(y)\, \mathrm{d}y\, \mathrm{d}x, \tag{6.33}$$

the bivariate matrix function decouples termwise into

$$\mathbf{X} = \sum_{i,j=0}^{\infty} c_{ij}\, p_i^{\mathbf{B}}(\mathbf{B}) \mathbf{U} \mathbf{V} p_j^{\mathbf{A}}(\mathbf{A}).$$

Applying the approximation (6.28) to (6.33), we find that

$$c_{ij} \approx \sum_{\ell=1}^{r} \beta_{\ell,i} \gamma_{\ell,j},$$

so the kth iterate of Algorithm 10 approximates the kth square partial sum:

$$\mathbf{X}_k \approx \sum_{i,j=0}^{k-1} c_{ij}\, p_i^{\mathbf{B}}(\mathbf{B})\mathbf{U}\mathbf{V}p_j^{\mathbf{A}}(\mathbf{A}).$$

Thus, the convergence rate of Algorithm 10 is governed by the decay rate of the bivariate orthogonal polynomial series coefficients c_{ij}. This decay is governed by the joint analyticity of f, measured in each variable through the generalized Bernstein ellipses of Lemma 5.3.6. Denote by $\varrho_{\mathbf{B}}$ the largest ϱ such that $f(\diamond, y)$ extends analytically to the interior of $\Gamma_\varrho(\Sigma_{\mathbf{B}})$ for every $y \in \Sigma_{\mathbf{A}}$ and by $\varrho_{\mathbf{A}}$ the largest ϱ such that $f(x, \diamond)$ extends analytically to the interior of $\Gamma_\varrho(\Sigma_{\mathbf{A}})$ for every $x \in \Sigma_{\mathbf{B}}$.

Theorem 6.7.2. *Assume that both the low-rank approximation* (6.28) *and the series coefficients $\beta_{\ell,k}, \gamma_{\ell,k}$ are computed exactly. Let $\mathbf{A} = \mathbf{V}_{\mathbf{A}}\mathbf{\Lambda}_{\mathbf{A}}\mathbf{V}_{\mathbf{A}}^{-1}$ and $\mathbf{B} = \mathbf{V}_{\mathbf{B}}\mathbf{\Lambda}_{\mathbf{B}}\mathbf{V}_{\mathbf{B}}^{-1}$ be diagonalizable with $\sigma(\mathbf{A})$ and $\sigma(\mathbf{B})$ in the interiors of $\Sigma_{\mathbf{A}}$ and $\Sigma_{\mathbf{B}}$, respectively. Then there is a constant $D > 0$, independent of k, such that the iterates of Algorithm 10 satisfy*

$$\|\mathbf{X}_k - \mathbf{X}\|_2 \le D\, \varrho^{-k}, \qquad \varrho = \min\{\varrho_{\mathbf{A}}, \varrho_{\mathbf{B}}\}.$$

Proof. See Appendix A.4. □

The asymptotic convergence rate is $\varrho^{-1} = \max\{\varrho_{\mathbf{A}}^{-1}, \varrho_{\mathbf{B}}^{-1}\}$, the slower of the two univariate rates. For Sylvester equations (1.2), the corresponding function $f(x, y) = 1/(y - x)$ has a singularity only at $x = y$, so the convergence rate is governed by $\mathfrak{g}_{\Sigma_{\mathbf{A}}}$ (resp. $\mathfrak{g}_{\Sigma_{\mathbf{B}}}$) evaluated at the nearest endpoint of $\Sigma_{\mathbf{B}}$ (resp. $\Sigma_{\mathbf{A}}$). Writing $\Sigma_{\mathbf{B}} = [a_1, b_1]$ and $\Sigma_{\mathbf{A}} = [a_2, b_2]$ with $b_1 < a_2$,

$$\varrho^{-1} = \max\left\{ \mathrm{e}^{-\mathrm{Re}\,\mathfrak{g}_{\Sigma_{\mathbf{B}}}(a_2)},\, \mathrm{e}^{-\mathrm{Re}\,\mathfrak{g}_{\Sigma_{\mathbf{A}}}(b_1)} \right\}. \tag{6.34}$$

One can verify that in the symmetric single-interval case, this recovers the error bound of a polynomial Krylov method (6.4) displayed in Figure 6.1.

6.7.2 Examples

We demonstrate Algorithm 10 on three problems, collected in Figure 6.16: a Sylvester equation, a Stein equation, and a Fréchet derivative. Throughout, $\mathbf{A}$ and $\mathbf{B}$ have real spectra and $\sigma(\mathbf{A}) \subset [2, 3]$, each kernel function f is approximated in low-rank form by Algorithm 11 sampled on circular contours, and all errors and residuals are measured in the matrix 2-norm.

For the Sylvester equation (1.2) $\mathbf{X}\mathbf{A} - \mathbf{B}\mathbf{X} = \mathbf{U}\mathbf{V}$, we take $\sigma(\mathbf{B}) \subset [-2, 1.2]$ and $f(x, y) = 1/(y - x)$. The error against the Bartels–Stewart solution [23] (`sylvester` in `Julia`) decays at the predicted rate (6.34) (the reference line), saturating near 35 iterations.

For the Stein equation $\mathbf{X} - \mathbf{B}\mathbf{X}\mathbf{A} = \mathbf{U}\mathbf{V}$, a case of (6.26) with $P(x, y) = 1 - xy$, we take $\sigma(\mathbf{B}) \subset [-0.2, 0.2]$ and $f(x, y) = 1/(1 - xy)$. This kernel is singular on $xy = 1$, so its factors carry singularities at the reciprocals of the opposite spectrum, and the rate (6.34) is set by Lemma 5.3.6 evaluated there; the residual saturates near 35 iterations.

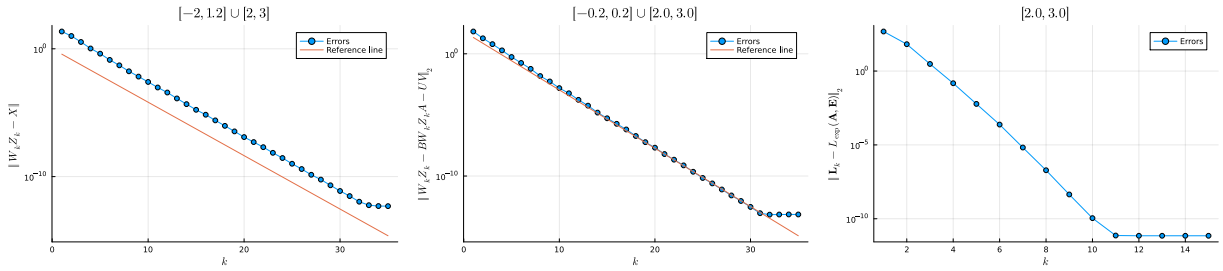


Figure 6.16: Convergence of Algorithm 10, with $\sigma(\mathbf{A}) \subset [2, 3]$ in all three panels. **Left:** the error in solving the Sylvester equation (1.2) $\mathbf{XA} - \mathbf{BX} = \mathbf{UV}$ with $\sigma(\mathbf{B}) \subset [-2, 1.2]$; the reference line is the rate (6.34). **Center:** the residual in solving the Stein equation $\mathbf{X} - \mathbf{BXA} = \mathbf{UV}$ with $\sigma(\mathbf{B}) \subset [-0.2, 0.2]$. **Right:** the error in computing the Fréchet derivative $L_{\exp}(\mathbf{A}, \mathbf{E})$ with $\mathbf{B} = \mathbf{A}$ and a low-rank direction $\mathbf{E}$.

For the Fréchet derivative $L_{\exp}(\mathbf{A}, \mathbf{E})$, we take $g(x) = \mathrm{e}^x$ and compute $f\{\mathbf{A}, \mathbf{A}\}(\mathbf{E})$ for f defined as in (6.27), comparing against the lower-left block of $\exp(\begin{smallmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{E} & \mathbf{A} \end{smallmatrix})$. This kernel is entire, so convergence is faster than any geometric rate, saturating in about 11 iterations.

Part III

THE KDV EQUATION

Chapter 7

SOLITON GASSES

7.1 Introduction

This chapter is based on [16]. Solution animations and `Julia` code used to generate the plots in this chapter can be found at [15].

A soliton gas can be thought of as a very large number of solitons, with possibly random amplitudes and phases, interacting weakly (rarefied gas) or strongly (dense gas) in a medium modeled by an integrable nonlinear wave model. This concept goes back to the work of Zakharov [166]. There has been a growing interest in the study of such solutions at the analytical, numerical, and experimental fronts in the past decade or so, and the literature on soliton gasses is vast. We refer the reader to the comprehensive review article [146] on the subject and the references therein. The notion behind the construction of such solutions from an integrable systems point of view is the so-called "dressing" method [167]. This idea was employed in [61] to construct the so-called primitive potentials, which can be taken as initial data for a solution modeling a soliton gas.

The current state of the art for computing soliton gas solutions of integrable nonlinear wave models is largely limited to either computing a suitable nonlinear superposition of N solitons for N large or numerically implementing an available asymptotic formula in a suitable asymptotic region of the space-time domain. The former approach is equivalent to computing an iterated (or N-fold) Darboux transformation, and it requires high-precision arithmetic because the size of the underlying linear algebra system is proportional to N (the number of solitons), and that linear system becomes numerically ill-conditioned as N grows. The latter approach has its difficulties as the accurate evaluation of asymptotic formulæ involving Riemann theta functions is a challenging task, if at all possible, especially when the genus of the underlying Riemann surface is not small. Moreover, the availability of spacetime asymptotic formulæ is limited to soliton gasses obtained by the accumulation of eigenvalues on one pair of bands in the spectral plane. Regardless, such formulæ cease to be accurate for intermediate values of (x, t) outside of asymptotic regimes. An approach that is more closely related to our work was taken in [61], where soliton gas (primitive) potentials were computed via the solution of a system of singular integral equations arising from a dressing construction. The numerical solution of this system necessitated the use of high-precision (i.e., quadruple or higher) arithmetic. In the current work, we avoid the use of high-precision arithmetic by using Riemann–Hilbert (RH) steepest-descent deformations [55] to, in effect, precondition the singular integral equations. Despite the growing body of work on both theoretical and experimental fronts, an efficient framework for computing soliton gas solutions of integrable nonlinear wave models has been elusive.

Specifically, in this chapter, we introduce a fast and accurate method to compute a class of soliton gas primitive potentials in the context of the Korteweg–de Vries (KdV) equation

$$u_t + 6uu_x + u_{xxx} = 0, \quad -\infty < x < +\infty, \tag{7.1}$$

based on the numerical solution of their Riemann–Hilbert problem (RHP) representations. An advantage of this approach is that there is no time-stepping involved, as (x, t) enter the

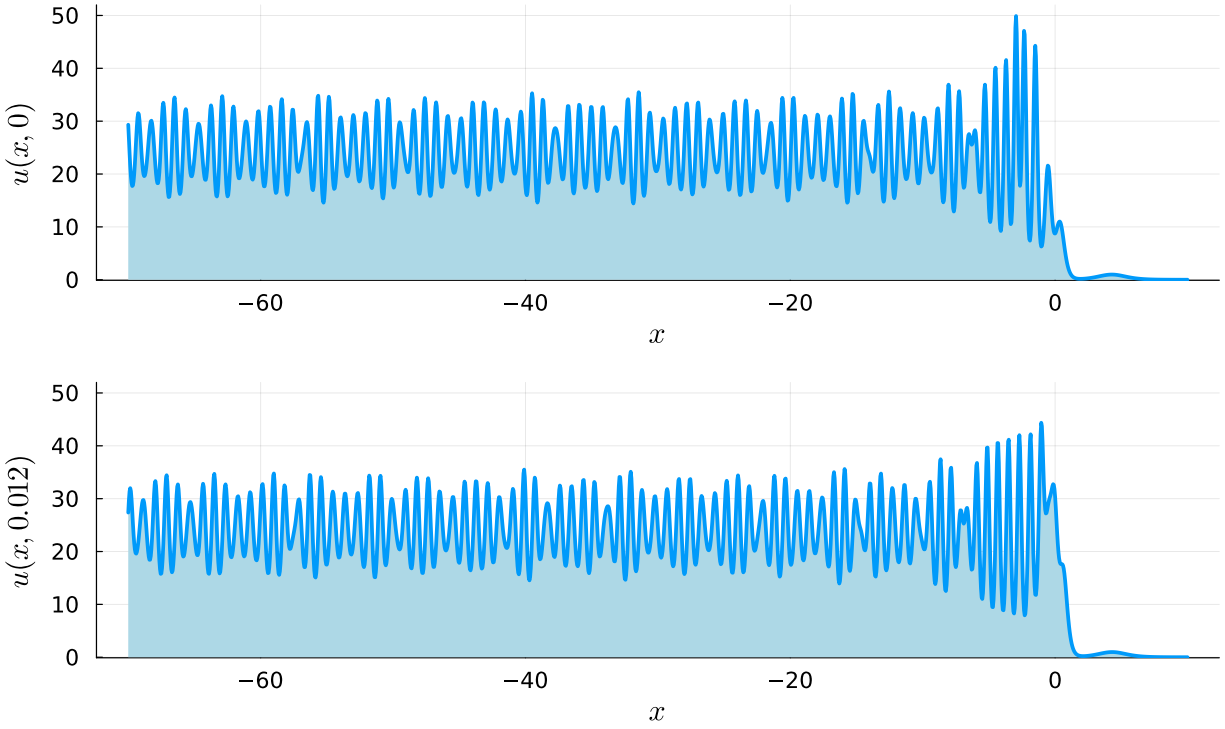


Figure 7.1: KdV soliton gas with $r_1(\lambda)$ supported on five pairs of bands, nonlinearly superimposed with five solitons. In the notation of (7.4), $I_1 = [0.25, 0.5]$, $I_2 = [0.8, 1.2]$, $I_3 = [1.5, 2]$, $I_4 = [2.5, 3]$, $I_5 = [4, 5]$, with $f_1(z) = 1$, $f_2(z) = 1/2$, $f_3(z) = 1/4$, $f_4(z) = 1/8$, $f_5(z) = 1/16$ and $\mathfrak{a}_j = \mathfrak{b}_j = \frac{1}{2}$ for $j = 1, \dots, 5$. The solitons are associated with (see Riemann–Hilbert Problem 7.3.1) $\kappa_1 = 0.1$, $\kappa_2 = 0.7$, $\kappa_3 = 2.25$, $\kappa_4 = 3.5$, $\kappa_5 = 5.5$ with the norming constants $\chi_1 = 10^5$, $\chi_2 = 1000$, $\chi_3 = 100$, $\chi_4 = 10$, and $\chi_5 = 10^{-6}$.

problem as (explicit) parameters. The method is easily parallelizable over values of (x, t).

The method presented here can compute primitive potentials of soliton gasses (at $t = 0$, for all x) and their time evolution under the KdV flow for small values of t, still on the entire x-axis. In this regime, the method is asymptotically accurate as $x \to \pm\infty$ and does not require high-precision arithmetic. Importantly, it allows us to compute soliton gas solutions of the KdV equation with associated Riemann–Hilbert jump conditions supported on many pairs of bands. We also compute such soliton gas solutions nonlinearly superposed with a number of solitons. See Figure 7.1 for a soliton gas solution of (7.1) with associated Riemann–Hilbert jump conditions supported on five (pairs of) bands and nonlinearly superimposed with five solitons, computed with our method. Pointwise evaluation of the solution in Figure 7.1 does not require higher precision arithmetic and takes about 0.7 seconds in the unmodulated region (see Section 7.2.2) and 0.08 seconds in the quiescent region (see Section 7.2.1) on a standard laptop[1].

The method presented here can also compute soliton gas solutions of the KdV equation in the entire (x, t)-plane outside of an unbounded wedge-shaped region emanating from $(x, t) = (0, 0)$. See Figure 7.2 for a pure soliton gas (no solitons) supported on five pairs of bands. The bottom panel presents the computed large-time evolution in the tail $x < -Kt$ for some suitable choice of constant $K > 0$. Figure 7.3 provides the density plot of a soliton gas with associated Riemann–Hilbert jump conditions supported on a single pair of bands, nonlinearly superimposed with two solitons, in the complement of the aforementioned wedge-shaped region. As is apparent from this figure, the validity of our method extends a little bit into the wedge (see the boundary curves in black). The extension stops once the exponential factors supported on the suitable deformed jump contours become too large as (x, t) penetrates into the wedge.

In Figure 7.4, we present a plot of the computed large-time evolution of a soliton gas with associated Riemann–Hilbert jump conditions supported on two pairs of bands (instead of five as in Figures 7.1 and 7.2), and in the notation of (7.4), we choose $f_1(z)$ to be much larger than $f_2(z)$.

This work is a culmination of recent advances made in Chapter 4 in computing orthogonal polynomials that are orthogonal on multiple disjoint intervals and work in computing large-genus solutions of the KdV equation [33]. The key ingredient behind these developments is the choice of a basis that encodes the behavior (e.g., the singularity structure) of the solution of an RHP at the endpoints of the jump contour. We also make use of the machinery available in the `OperatorApproximation.jl` software package [155].

This work focuses mainly on the computation of soliton gas potentials supported on many bands and completing the groundwork for nonlinear superpositions of such potentials with many solitons. Extensions of our method to the entire (x, t)-plane, along with routines to compute KdV and mKdV soliton gas solutions, are developed in Chapter 8.

---

[1]All computations in this chapter can be performed on a Lenovo laptop running Ubuntu version 20.04 with 8 cores and 16 GB of RAM with an Intel® Core™ i7-11800H processor running at 2.30 GHz. However, due to the ease of parallelization, a cluster can greatly reduce the time it takes to generate the figures.

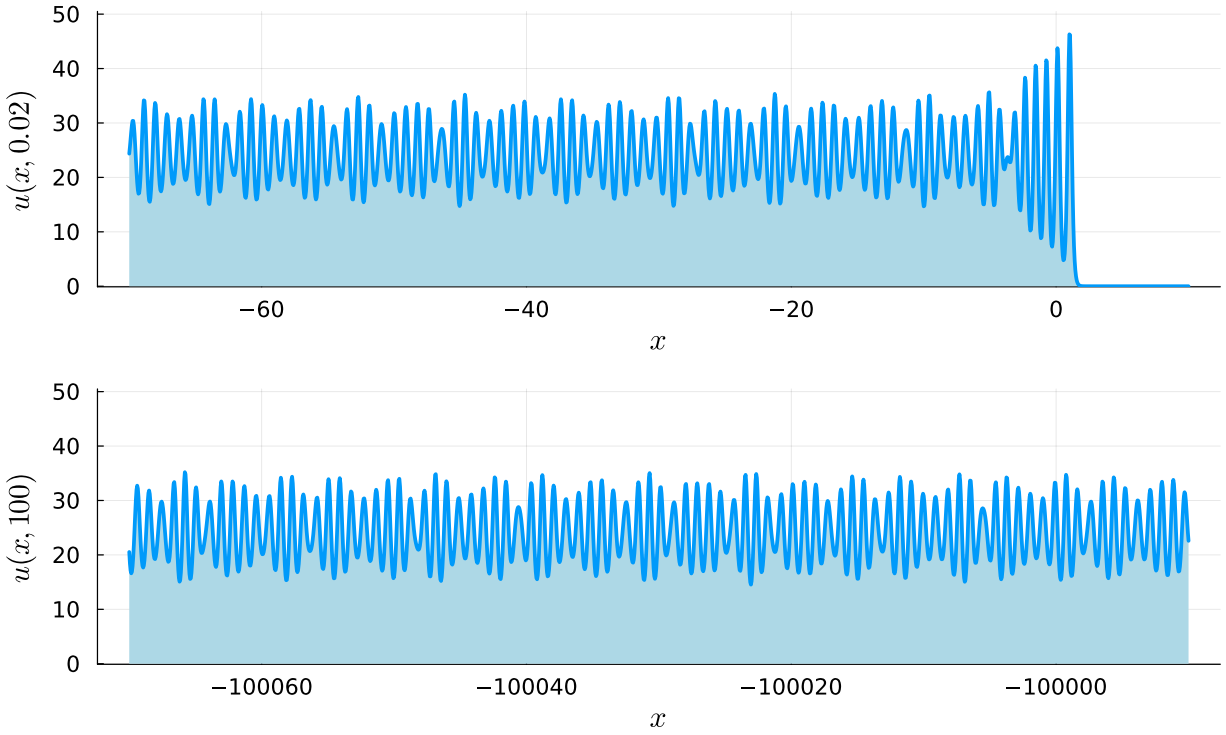


Figure 7.2: A pure KdV soliton gas with $r_1(\lambda)$ supported on five pairs of bands. In the notation of (7.4), $I_1 = [0.25, 0.5]$, $I_2 = [0.8, 1.2]$, $I_3 = [1.5, 2]$, $I_4 = [2.5, 3]$, $I_5 = [4, 5]$ with $f_1(z) = 1$, $f_2(z) = 1/2$, $f_3(z) = 1/4$, $f_4(z) = 1/8$, $f_5(z) = 1/16$ and $\mathfrak{a}_j = \mathfrak{b}_j = \frac{1}{2}$ for $j = 1, \ldots, 5$.

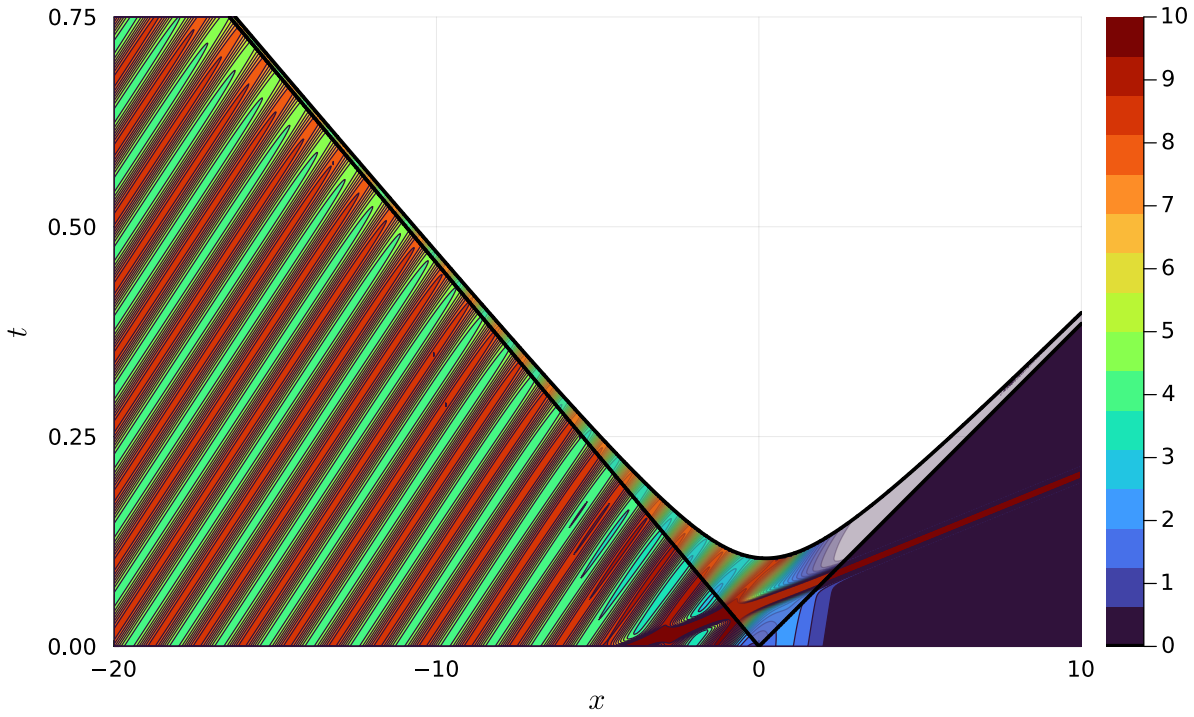


Figure 7.3: Density plot of the computed soliton gas with $r_1(\lambda)$ supported on a single pair of bands with two solitons. In the notation of (7.4), $I_1 = [1.5, 2.5]$, $f_1(z) = 1$, and $\mathfrak{a}_1 = \mathfrak{b}_1 = \frac{1}{2}$. The solitons are associated with the eigenvalue parameters $\kappa_1 = 1$, $\kappa_2 = 4$ and the norming constants $\chi_1 = 10$, $\chi_2 = 10^{-10}$. Outside of the wedge region, the numerical method presented here is seen to be uniformly accurate with a computational cost that is independent of (x, t). Inside the wedge, the numerical method begins to break down, and additional RH deformations need to be incorporated. See Chapter 8.

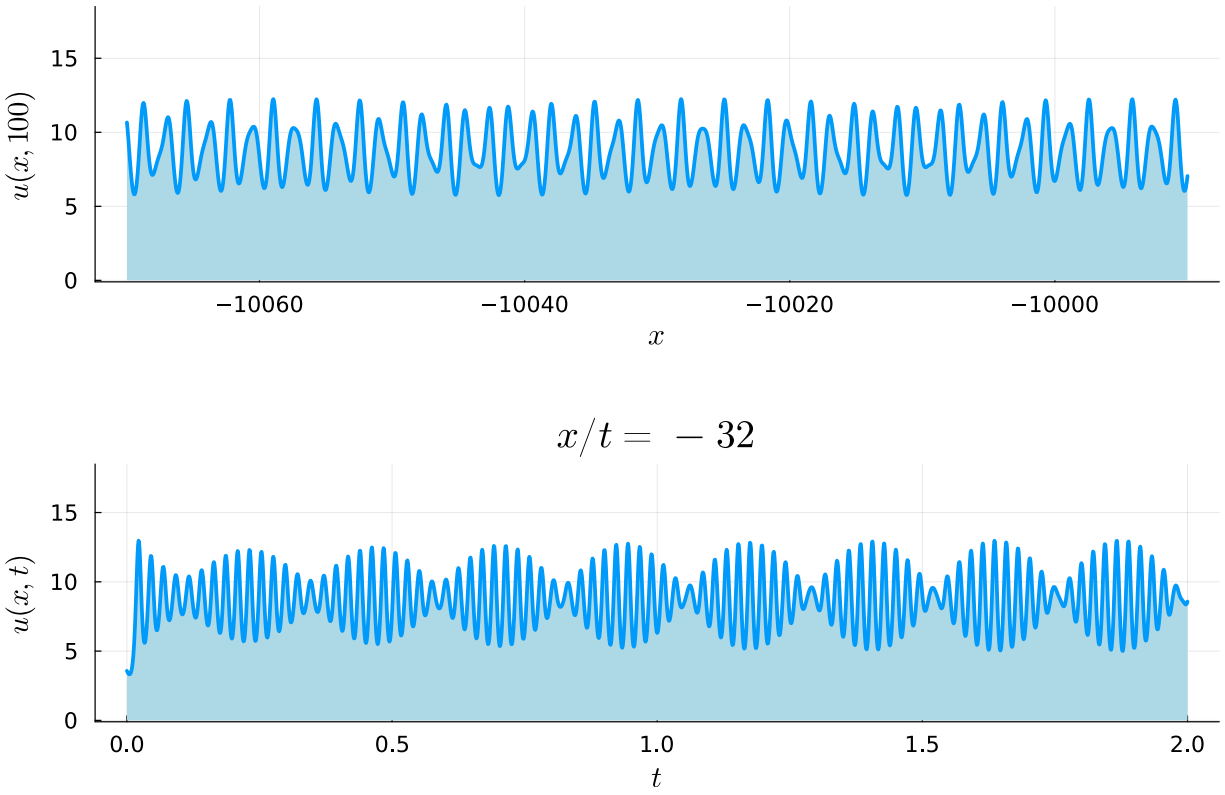


Figure 7.4: A pure KdV soliton gas with $r_1(\lambda)$ supported on two pairs of bands. In the notation of (7.4), $I_1 = [1, 2]$, $I_2 = [2.5, 3]$ with $f_1(z) = 100$, $f_2(z) = 1$ and $\mathfrak{a}_j = \mathfrak{b}_j = \frac{1}{2}$, $j = 1, 2$. Bottom panel: The same solution plotted along the ray $x/t = -32$.

7.1.1 Riemann–Hilbert definition of a KdV soliton gas

Let $g \in \mathbb{Z}_+$ be fixed and consider a collection of disjoint closed intervals $\mathrm{i}[a_j, b_j]$, $1 \le j \le g$, on the imaginary axis with $0 < a_j < b_j < a_{j+1}$ for each $1 \le j \le g-1$. We denote the union of these intervals and of their reflections by

$$\Sigma_+ := \bigcup_{j=1}^{g} [a_j, b_j] \qquad \text{and} \qquad \Sigma_- := \bigcup_{j=1}^{g} [-b_j, -a_j],$$

and take all of these intervals to be oriented *upwards*. We consider a function $r_1 \colon (\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-) \to \mathbb{C}$ such that $r_1(\lambda)$ is positive for λ in the interior of each interval $\mathrm{i}[a_j, b_j]$, extending analytically to a neighborhood of each $\mathrm{i}[a_j, b_j]$ with the local behavior

$$r_1(\mathrm{i}z) = f_{a_j}(z)(z - a_j)^{\mathfrak{a}_j}, \quad \mathfrak{a}_j \in \left\{-\tfrac{1}{2}, \tfrac{1}{2}\right\}, \tag{7.2}$$

for z in a neighborhood of a_j, where $f_{a_j}(z)$ is analytic and satisfies $f_{a_j}(z) > 0$ for $z \ge a_j$. Similarly,

$$r_1(\mathrm{i}z) = f_{b_j}(z)(b_j - z)^{\mathfrak{b}_j}, \quad \mathfrak{b}_j \in \left\{-\tfrac{1}{2}, \tfrac{1}{2}\right\}, \tag{7.3}$$

for z in a neighborhood of b_j, where $f_{b_j}(z)$ is analytic and satisfies $f_{b_j}(z) > 0$ for $z \le b_j$. Here, all of the roots are taken to be the principal branch. Therefore, in the computations in this chapter, we take

$$r_1(\mathrm{i}z) = f_j(z)(z - a_j)^{\mathfrak{a}_j}(b_j - z)^{\mathfrak{b}_j}, \quad \text{for } z \in I_j := [a_j, b_j], \quad \Sigma_+ = \bigcup_{j=1}^{g} I_j, \tag{7.4}$$

for analytic functions f_j and constants $\mathfrak{a}_j, \mathfrak{b}_j \in \left\{-\frac{1}{2}, \frac{1}{2}\right\}$. For $\lambda \in \mathrm{i}\Sigma_-$, r_1 is defined by symmetry: $r_1(\lambda) = r_1(-\lambda)$. With these ingredients, we consider the following problem, which we take as the definition of a soliton gas:

Riemann–Hilbert Problem 7.1.1 (Pure KdV soliton gas)**.** *Let $(x,t) \in \mathbb{R} \times \mathbb{R}_+$ be fixed. Find a 1×2 row vector-valued function $\mathbf{M}(\lambda) \equiv \mathbf{M}(\lambda; x, t)$ with the following properties:*

> ***Analyticity:*** *$\mathbf{M}(\lambda)$ is analytic for $\lambda \in \mathbb{C} \setminus (\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-)$, admitting continuous boundary values on the interior of $\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-$.*
>
> ***Jump conditions:*** *The boundary values $\mathbf{M}^+(\lambda)$ (resp. $\mathbf{M}^-(\lambda)$) from the left (resp. right) are related via*
>
> $$\mathbf{M}^+(\lambda) = \mathbf{M}^-(\lambda) \begin{pmatrix} 1 & 0 \\ -2\mathrm{i}r_1(\lambda)\mathrm{e}^{2\mathrm{i}(x\lambda + 4t\lambda^3)} & 1 \end{pmatrix}, \quad \lambda \in \mathrm{i}\Sigma_+,$$
>
> $$\mathbf{M}^+(\lambda) = \mathbf{M}^-(\lambda) \begin{pmatrix} 1 & 2\mathrm{i}r_1(\lambda)\mathrm{e}^{-2\mathrm{i}(x\lambda + 4t\lambda^3)} \\ 0 & 1 \end{pmatrix}, \quad \lambda \in \mathrm{i}\Sigma_-, \quad j = 1, 2, \ldots, g.$$

Normalization: $\mathbf{M}(\lambda) = \begin{pmatrix} 1 & 1 \end{pmatrix} + \mathrm{O}(\lambda^{-1})$ *as* $\lambda \to \infty$.

Symmetry condition: $\mathbf{M}(-\lambda) = \mathbf{M}(\lambda) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

This problem has appeared in [61,78] with different assumptions on the behavior of $r_1(\lambda)$ at the endpoints[2]. The setting of [78] assumes that $r_1(\lambda)$ is nonvanishing and bounded at the endpoints, in contrast with the assumptions in our work. As the reader will see, the endpoint behavior that we assume for $r_1(\lambda)$ enables a particularly fast computational method. Riemann–Hilbert Problem 7.1.1 has a unique solution [78] (see Section 3.3.4 in Chapter 3 for the notion of a solution when $\mathfrak{a}_j$ or $\mathfrak{b}_j$ is negative), and we can recover the soliton gas solution of (7.1) by the second-order term at infinity [62]

$$u(x,t) = -\lim_{\lambda\to\infty} 2\lambda^2(\mathbf{M}_1(\lambda;x,t)\mathbf{M}_2(\lambda;x,t) - 1), \tag{7.5}$$

where $\mathbf{M}_j(\lambda;x,t)$ is the jth component of the row vector $\mathbf{M}(\lambda;x,t)$.

We proceed with rotating the λ-plane clockwise by introducing $r(z) := 2r_1(\mathrm{i}z)$ and $\mathbf{Y}(z) := \mathbf{M}(\mathrm{i}z)$, which satisfies the jump conditions

$$\mathbf{Y}^+(z) = \mathbf{Y}^-(z) \begin{pmatrix} 1 & 0 \\ -\mathrm{i}r(z)\mathrm{e}^{-2\theta(z;x,t)} & 1 \end{pmatrix}, \quad z \in \Sigma_+, \qquad \theta(z;x,t) := xz - 4tz^3, \tag{7.6}$$

$$\mathbf{Y}^+(z) = \mathbf{Y}^-(z) \begin{pmatrix} 1 & \mathrm{i}r(z)\mathrm{e}^{2\theta(z;x,t)} \\ 0 & 1 \end{pmatrix}, \qquad z \in \Sigma_-, \tag{7.7}$$

and is analytic elsewhere. Note that all (real) intervals are oriented in the increasing direction. We consider the numerical solution of the RHP satisfied by $\mathbf{Y}(z)$, which inherits the normalization and symmetry conditions in Riemann–Hilbert Problem 7.1.1.

Regarding the behavior of $\mathbf{Y}(z)$ at the endpoints, let c_j denote either of the endpoints a_j or b_j, and let γ_j denote the corresponding power $\mathfrak{a}_j$ or $\mathfrak{b}_j$ as in (7.2) and (7.3). As $z \to c_j$, we have

$$\mathbf{Y}(z) = \begin{cases} \begin{pmatrix} \mathrm{O}(1) & \mathrm{O}(1) \end{pmatrix}, & \text{if} \quad \gamma_j = \frac{1}{2}, \\ \begin{pmatrix} \mathrm{O}(|z-c_j|^{-\frac{1}{2}}) & \mathrm{O}(1) \end{pmatrix}, & \text{if} \quad \gamma_j = -\frac{1}{2}. \end{cases} \tag{7.8}$$

This situation is mirrored on the negative real axis. As $z \to -c_j$, we have

$$\mathbf{Y}(z) = \begin{cases} \begin{pmatrix} \mathrm{O}(1) & \mathrm{O}(1) \end{pmatrix}, & \text{if} \quad \gamma_j = \frac{1}{2}, \\ \begin{pmatrix} \mathrm{O}(1) & \mathrm{O}(|z+c_j|^{-\frac{1}{2}}) \end{pmatrix}, & \text{if} \quad \gamma_j = -\frac{1}{2}. \end{cases}$$

[2]A derivation of Riemann–Hilbert Problem 7.1.1 via a limiting procedure involving the accumulation of eigenvalues of the Schrödinger operator on a single pair of bands with respect to a suitable density can be found in [78].

The endpoint behavior stated in Riemann–Hilbert Problem 7.1.1 is thus consistent with and implied by the behavior of the jump matrix at the endpoints: it is not an independent condition but rather specifies the appropriate function space in which the solution is sought.

Writing $\lambda = \mathrm{i}z$ in the large-λ expansions that led to (7.5), the soliton gas solution $u(x,t)$ of (7.1) is recovered from $\mathbf{Y}(z) \equiv \mathbf{Y}(z;x,t)$ via the formula

$$u(x,t) = \lim_{z\to\infty} 2z^2(\mathbf{Y}_1(z;x,t)\mathbf{Y}_2(z;x,t) - 1). \tag{7.9}$$

The choice of basis for the Cauchy density below is motivated by a local singularity analysis carried out in Section 3.3.4 of Chapter 3; the structure encountered in the unmodulated region (Section 7.2.2) is analyzed there in detail, while the quiescent region (Section 7.2.1) is simpler since its triangular jump matrices admit local solutions directly by Cauchy integrals.

7.2 Computing soliton gasses

7.2.1 Quiescent region

When $t = 0$, the jump matrices in (7.6)–(7.7) are O(1) (away from the endpoints) for $x \geq 0$ and become exponentially close to identity as $x \to +\infty$ since $\mathrm{Re}(-2zx) < 0$ on Σ_+ and $\mathrm{Re}(2zx) < 0$ on Σ_- in this case. This configuration extends to a region of the (x,t)-plane for $t > 0$ as long as $\mathrm{Re}(-2\theta(z;x,t)) < 0$ is maintained on Σ_+ (the situation is mirrored on Σ_- automatically). The axis $\mathrm{Re}(z) = 0$ is always a part of the locus $\mathrm{Re}(-2\theta(z;x,t)) = 0$. The other branch of this locus is given by $\mathrm{Re}(z)^2 = 3\mathrm{Im}(z)^2 + \frac{x}{4t}$. Thus, the aforementioned boundedness or exponential decay of the jump matrices to the identity is preserved for all (x,t) with $t \geq 0$ satisfying

$$x \geq 4Kt, \tag{7.10}$$

for an arbitrary constant $K > b_g^2$. In this setting, the RHP at hand can be treated numerically as is, without implementing any contour deformations, by using an appropriate basis with weights encoding the endpoint behavior of $\mathbf{Y}(z)$ as described in Chapter 3. The region described by (7.10) is the analog of the "constant region" as $t \to +\infty$ in [78] for the case $g = 1$.

7.2.2 Unmodulated region

When $t = 0$ and $x < 0$, the situation is dramatically different: All of the exponentials in (7.6)–(7.7) grow exponentially and unboundedly as $x \to -\infty$. This behavior cannot be put under control solely by employing one of the four canonical matrix factorizations for the jump matrices since the exponents are real-valued. In this setting, the RHP is stabilized by modifying the exponent via introducing a g-function. This is a generalization of the approach taken in [78, 79] for the case $g = 1$. Let $\Gamma_{\pm j}$ denote the gaps: $\Gamma_0 = [-a_1, a_1]$, $\Gamma_j = [b_j, a_{j+1}]$, and $\Gamma_{-j} = [-a_{j+1}, -b_j]$ for $j = 1, 2, \ldots, g-1$. To have a uniform treatment that also captures a region when $t > 0$, we proceed with allowing for $t \geq 0$ with $x < 0$ and do

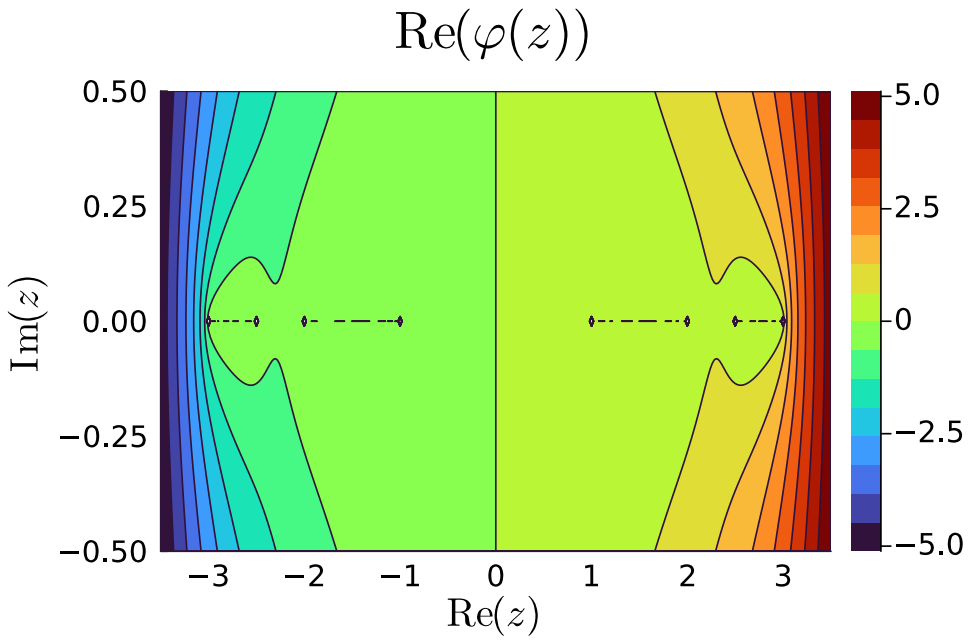


Figure 7.5: Contour plot showing the sign of $\mathrm{Re}(\varphi(z;x,t))$ in the complex plane for $t>0$ in the unmodulated region treated in Section 7.2.2. Here, $x=-2$, $t=0.01$, and in the notation of (7.4), $I_1=[1,2]$, $I_2=[2.5,3]$.

not designate a particular large parameter. Since $\theta(-z;x,t) = -\theta(z;x,t)$ and $R(-z) = R(z)$ (where $R(z)^2 = \prod_{j=1}^{g}(z^2 - a_j^2)(z^2 - b_j^2)$), the solution satisfies $\mathfrak{g}(-z;x,t) = -\mathfrak{g}(z;x,t)$. This odd symmetry halves the effective size of the problem: it suffices to construct $\mathfrak{g}$ in the *folded plane* $\zeta = z^2$, following the approach of Chapter 8. Denote

$$\hat{\Sigma} = \bigcup_{j=1}^{g}[a_j^2, b_j^2], \qquad \hat{\Gamma}_0 = (0, a_1^2), \qquad \hat{\Gamma}_k = (b_k^2, a_{k+1}^2), \quad k = 1, \dots, g-1,$$

and define the folded phase $\hat{\theta}(\zeta;x,t) := x\sqrt{\zeta} - 4t\zeta^{3/2}$, so that $\theta(z;x,t) = \hat{\theta}(z^2;x,t)$ for $z > 0$. Let $\hat{R}(\zeta)$ be the function analytic for $\zeta \notin \hat{\Sigma}$ satisfying

$$\hat{R}(\zeta)^2 = \prod_{j=1}^{g}(\zeta - a_j^2)(\zeta - b_j^2), \qquad \hat{R}(\zeta) = \zeta^g + \mathrm{O}(\zeta^{g-1}) \quad \text{as } \zeta \to \infty.$$

We seek $\hat{\mathfrak{g}}(\zeta) \equiv \hat{\mathfrak{g}}(\zeta;x,t)$ analytic for $\zeta \notin (-\infty, b_g^2]$, admitting continuous boundary values on $(-\infty, b_g^2)$, satisfying

$$\begin{aligned} &\hat{\mathfrak{g}}^+(\zeta) + \hat{\mathfrak{g}}^-(\zeta) = 2\hat{\theta}(\zeta;x,t), \quad \zeta \in \hat{\Sigma}, \\ &\hat{\mathfrak{g}}^+(\zeta) - \hat{\mathfrak{g}}^-(\zeta) = \mathrm{i}\hat{\Omega}_k(x,t), \quad \zeta \in \hat{\Gamma}_k, \quad k = 0, 1, \dots, g-1, \end{aligned}$$

where $\hat{\Omega}_k(x,t)$ are real-valued constants, along with $\hat{\mathfrak{g}}(\zeta) = \mathrm{O}(\zeta^{-1/2})$ as $\zeta \to \infty$ and $\hat{\mathfrak{g}}(\zeta) = \mathrm{O}(\zeta - c^2)^{1/2}$ as $\zeta \to c^2$ for each endpoint $c \in \{a_1, b_1, \dots, a_g, b_g\}$. Then,

$$\hat{\mathfrak{g}}(\zeta) = \frac{\hat{R}(\zeta)}{2\pi\mathrm{i}} \int_{\hat{\Sigma}} \frac{2\hat{\theta}(s;x,t)}{\hat{R}^+(s)(s-\zeta)} \mathrm{d}s + \sum_{k=0}^{g-1} \frac{\hat{\Omega}_k(x,t)\hat{R}(\zeta)}{2\pi} \int_{\hat{\Gamma}_k} \frac{\mathrm{d}s}{\hat{R}(s)(s-\zeta)},$$

where the g constants $\hat{\Omega}_0(x,t), \dots, \hat{\Omega}_{g-1}(x,t)$ are determined by the decay condition at infinity. These moment conditions result in the linear system

$$\sum_{k=0}^{g-1} \mathrm{i}\hat{\Omega}_k(x,t) \int_{\hat{\Gamma}_k} \frac{s^m \mathrm{d}s}{\hat{R}(s)} = -\int_{\hat{\Sigma}} \frac{2s^m \hat{\theta}(s;x,t)}{\hat{R}^+(s)} \mathrm{d}s, \quad m = 0, 1, \dots, g-1. \tag{7.11}$$

The unique solvability of this linear system is established by the following lemma.

Lemma 7.2.1. *The $g \times g$ matrix $\mathbf{G}$ defined by*

$$\mathbf{G} = \left(\int_{\hat{\Gamma}_k} \frac{s^m \, \mathrm{d}s}{\hat{R}(s)} \right)_{0 \le m,k \le g-1}$$

is invertible.

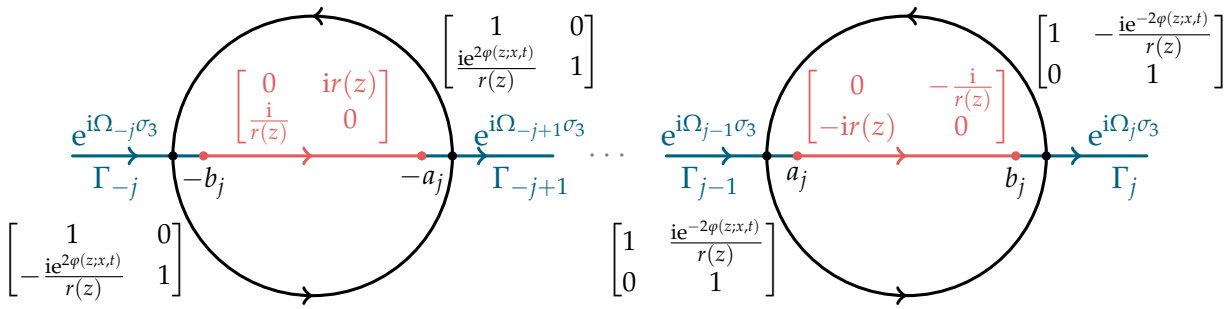


Figure 7.6: Jump contours and jump matrices associated with $\mathbf{S}(z;x,t)$ near each interval. $\varphi(z;x,t) = \mathfrak{g}(z;x,t) - \theta(z;x,t)$.

Proof. The proof is identical to that of Lemma 4.2.2. □

This system is half the size of the corresponding system in the unfolded plane: the symmetry $\Omega_{-k} = \Omega_k$ reduces the $2g-1$ original gap constants to g independent unknowns $\hat{\Omega}_k = \Omega_k$. As with the unfolded system, conditioning is improved by replacing s^m with a degree-$(g-1)$ polynomial basis vanishing at the midpoints of $\hat{\Gamma}_0, \ldots, \hat{\Gamma}_{g-1}$. We compute the integrals in (7.11) via Gauss–Chebyshev quadrature. Finally, $\mathfrak{g}$ is recovered from $\hat{\mathfrak{g}}$ by

$$\mathfrak{g}(z;x,t) = \begin{cases} \hat{\mathfrak{g}}(z^2;x,t), & \operatorname{Re}(z) \geq 0, \\ -\hat{\mathfrak{g}}(z^2;x,t), & \operatorname{Re}(z) < 0. \end{cases}$$

Now, introduce open and disjoint disks D_j, $j = 1, 2, \ldots, g$, chosen so that D_j encloses $[a_j, b_j]$ in its interior, and let D_{-j} be the analog of D_j for $[-b_j, -a_j]$. We take the disk boundaries to be oriented counter-clockwise and make the following definitions (with correlated signs on each line):

$$\mathbf{S}(z) := \mathbf{Y}(z) \begin{pmatrix} 1 & \frac{i}{r(z)} e^{2\theta(z;x,t)} \\ 0 & 1 \end{pmatrix}^{\mp 1} e^{\mathfrak{g}(z;x,t)\boldsymbol{\sigma}_3}, \qquad z \in D_j \cap \mathbb{C}^{\pm}, \tag{7.12}$$

$$\mathbf{S}(z) := \mathbf{Y}(z) \begin{pmatrix} 1 & 0 \\ \frac{-i}{r(z)} e^{-2\theta(z;x,t)} & 1 \end{pmatrix}^{\mp 1} e^{\mathfrak{g}(z;x,t)\boldsymbol{\sigma}_3}, \qquad z \in D_{-j} \cap \mathbb{C}^{\pm}, \tag{7.13}$$

for $j = 1, 2, \ldots, g$. Recall that $\boldsymbol{\sigma}_3 = \operatorname{diag}(1, -1)$. We set $\mathbf{S}(z) := \mathbf{Y}(z) e^{\mathfrak{g}(z;x,t)\boldsymbol{\sigma}_3}$ elsewhere. Using $r^+(z) = -r^-(z)$ on $\Sigma_+ \cup \Sigma_-$, we find that the jump conditions satisfied by $\mathbf{S}(z)$ are as given in Figure 7.6. We denote the modified exponent by $\varphi(z;x,t) := \mathfrak{g}(z;x,t) - \theta(z;x,t)$. A contour plot of the sign of $\operatorname{Re}(\varphi(z;x,t))$ is given in Figure 7.5, and one can see that the jump matrices on the circles are exponentially close to the identity matrix if $t > 0$ becomes large for x/t in this region. This behavior together with the fact that the circular jump contours are detached from $\Sigma_+ \cup \Sigma_-$ (no self-intersection points with jump matrices involving $r(z)$)

is what enables an efficient numerical method. Therefore, the assumptions on the behavior of $r(z)$ at the endpoints are absolutely essential for our approach.

Again, let c_j denote either of the endpoints a_j or b_j, and let γ_j denote the corresponding power $\mathfrak{a}_j$ or $\mathfrak{b}_j$ as in (7.2) and (7.3). From (7.8), (7.12), and (7.13), we find that as $z \to c_j$,

$$\mathbf{S}(z) = \begin{cases} \begin{pmatrix} \mathrm{O}(1) & \mathrm{O}(|z-c_j|^{-\frac{1}{2}}) \end{pmatrix}, & \text{if} \quad \gamma_j = \frac{1}{2}, \\ \begin{pmatrix} \mathrm{O}(|z-c_j|^{-\frac{1}{2}}) & \mathrm{O}(1) \end{pmatrix}, & \text{if} \quad \gamma_j = -\frac{1}{2}. \end{cases} \tag{7.14}$$

The situation is again mirrored as z approaches an endpoint of $[-b_j, -a_j]$, but the behavior of the columns is flipped.

As in Chapter 4, we introduce a correction $\mathfrak{h}(z) \equiv \mathfrak{h}(z; x, t)$ to eliminate the constant jump conditions on the gaps at the expense of introducing real constants on the intervals. By the same symmetry, $\mathfrak{h}(-z; x, t) = -\mathfrak{h}(z; x, t)$, so we construct $\mathfrak{h}$ via a folded counterpart $\hat{\mathfrak{h}}(\zeta)$. We seek $\hat{\mathfrak{h}}(\zeta)$ analytic for $\zeta \notin (-\infty, b_g^2]$, admitting continuous boundary values on $(-\infty, b_g^2)$, satisfying

$$\begin{aligned} &\hat{\mathfrak{h}}^+(\zeta) - \hat{\mathfrak{h}}^-(\zeta) = \log\Big(\exp(-\mathrm{i}\hat{\Omega}_k(x,t))\Big), \quad \zeta \in \hat{\Gamma}_k, && k = 0, 1, \ldots, g-1, \\ &\hat{\mathfrak{h}}^+(\zeta) + \hat{\mathfrak{h}}^-(\zeta) = \hat{A}_j(x,t), && \zeta \in (a_j^2, b_j^2), \quad j = 1, 2, \ldots, g, \end{aligned}$$

along with $\hat{\mathfrak{h}}(\zeta) = \mathrm{O}(\zeta^{-1/2})$ as $\zeta \to \infty$ and $\hat{\mathfrak{h}}(\zeta) = \mathrm{O}(\zeta - c^2)^{1/2}$ as $\zeta \to c^2$ for each endpoint $c \in \{a_1, b_1, \ldots, a_g, b_g\}$. Then,

$$\hat{\mathfrak{h}}(\zeta) = \frac{\hat{R}(\zeta)}{2\pi\mathrm{i}} \left[\sum_{j=1}^{g} \int_{a_j^2}^{b_j^2} \frac{\hat{A}_j(x,t)\mathrm{d}s}{\hat{R}^+(s)(s-\zeta)} + \sum_{k=0}^{g-1} \int_{\hat{\Gamma}_k} \frac{\log(\exp(-\mathrm{i}\hat{\Omega}_k(x,t)))\mathrm{d}s}{\hat{R}(s)(s-\zeta)} \right],$$

where the g real constants $\hat{A}_1(x,t), \ldots, \hat{A}_g(x,t)$ are determined by the decay condition at infinity. The moment conditions result in the linear system

$$\sum_{j=1}^{g} \hat{A}_j(x,t) \int_{a_j^2}^{b_j^2} \frac{s^m \mathrm{d}s}{\hat{R}^+(s)} = -\sum_{k=0}^{g-1} \log(\exp(-\mathrm{i}\hat{\Omega}_k(x,t))) \int_{\hat{\Gamma}_k} \frac{s^m \mathrm{d}s}{\hat{R}(s)}, \quad m = 0, 1, \ldots, g-1. \tag{7.15}$$

The unique solvability of this system is established by the following lemma.

Lemma 7.2.2. *The $g \times g$ matrix $\mathbf{H}$ defined by*

$$\mathbf{H} = \left(\int_{a_j^2}^{b_j^2} \frac{s^m \, \mathrm{d}s}{\hat{R}^+(s)} \right)_{0 \le m \le g-1,\ 1 \le j \le g}$$

is invertible.

Proof. The proof is identical to that of Lemma 4.2.4, noting that $\hat{R}^+(s)$ is purely imaginary with sign-definite imaginary part for $s \in (a_j^2, b_j^2)$. □

The coefficients on the left-hand side are purely imaginary for each m (since $\hat{R}^+(s)$ is purely imaginary on $\hat{\Sigma}$), confirming that $\hat{A}_j(x,t) \in \mathbb{R}$. The coefficient matrix is independent of (x,t), and the right-hand side remains bounded in (x,t) even though $\hat{\Omega}_k(x,t)$ may grow, so $\hat{A}_j(x,t)$ are bounded. The integrals in (7.15) are computed via Gauss–Chebyshev quadrature using the same polynomial basis approach as in (7.11). Finally,

$$\mathfrak{h}(z;x,t) = \begin{cases} \hat{\mathfrak{h}}(z^2;x,t), & \mathrm{Re}(z) \geq 0, \\ -\hat{\mathfrak{h}}(z^2;x,t), & \mathrm{Re}(z) < 0. \end{cases} \tag{7.16}$$

We now make a global definition and introduce an exponent:

$$\mathbf{R}(z) := \mathbf{S}(z)\mathrm{e}^{\mathfrak{h}(z;x,t)\boldsymbol{\sigma}_3} \quad \text{and} \quad \phi(z;x,t) := \mathfrak{h}(z;x,t) + \mathfrak{g}(z;x,t) - \theta(z;x,t).$$

Note that $\mathbf{R}(z) = \begin{pmatrix} 1 & 1 \end{pmatrix} + \mathrm{O}(z^{-1})$ as $z \to \infty$. Since the transformations $\mathbf{Y}(z) \mapsto \mathbf{S}(z) \mapsto \mathbf{R}(z)$ involve right-multiplications by diagonal unimodular matrices near $z = \infty$, it follows from (7.9) that

$$u(x,t) = \lim_{z\to\infty} 2z^2(\mathbf{R}_1(z;x,t)\mathbf{R}_2(z;x,t) - 1).$$

For z lying over a gap $\hat{\Gamma}_k$ (i.e., $z^2 \in \hat{\Gamma}_k$, $\mathrm{Re}(z) > 0$), the unfolding (7.16) and condition (7.15) give $\mathfrak{h}^+(z;x,t) - \mathfrak{h}^-(z;x,t) = \log(\exp(-\mathrm{i}\hat{\Omega}_k(x,t)))$. Therefore

$$\mathfrak{h}^+(z) - \mathfrak{h}^-(z) + \mathrm{i}\hat{\Omega}_k = \log(\exp(-\mathrm{i}\hat{\Omega}_k)) + \mathrm{i}\hat{\Omega}_k = 2\pi\mathrm{i}\ell$$

for some $\ell \in \mathbb{Z}$, and similarly on $-\hat{\Gamma}_k$ (the reflected gaps). Thus $\mathbf{R}(z)$ has no jumps across any of the gaps Γ_ℓ. The jump conditions satisfied by $\mathbf{R}(z)$ are described in Figure 7.7. This final RHP satisfied by $\mathbf{R}(z)$ is again treated numerically via the method of Chapter 3.

Remark 7.2.3. *This method could, in principle, be sped up significantly by considering r with a common analytic extension to a region containing $[-b_g, b_g]$. In this case, the $2g$ circular jump contours can be replaced by a single large one. See Section 4.4.*

Note that the gap in the (x,t)-plane for $t > 0$ between the regions described in Section 7.2.1 and Section 7.2.2 disappears at $t = 0$, allowing us to compute the soliton gas primitive potentials quickly and accurately for all $x \in \mathbb{R}$.

7.3 Nonlinear superposition of a pure soliton gas with solitons

It is possible to insert a multi-soliton into a soliton gas by supplying suitable residue conditions for the RHP satisfied by $\mathbf{Y}(z)$ for a number of simple poles. In this case, we consider solving the following RHP:

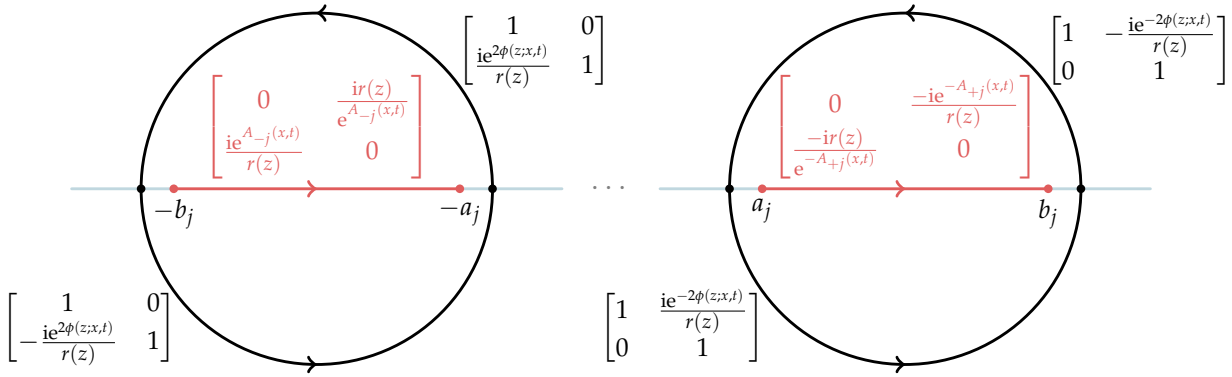


Figure 7.7: Jump contours and jump matrices associated with $\mathbf{R}(z;x,t)$ near each interval. $\phi(z;x,t) = \varphi(z;x,t) + \mathfrak{h}(z;x,t)$.

Riemann–Hilbert Problem 7.3.1 (KdV soliton gas with solitons)**.** *Let $(x,t) \in \mathbb{R} \times \mathbb{R}_+$ be fixed. Find a 1×2 (row vector-valued) function $\mathbf{N}(\lambda) \equiv \mathbf{N}(\lambda;x,t)$ with the following properties:*

Analyticity: *$\mathbf{N}(\lambda)$ is analytic for $\lambda \in \mathbb{C} \setminus (\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-)$, admitting continuous boundary values on the interior of $\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-$, with the exception of simple poles at $\lambda = \pm\mathrm{i}\kappa_j$, $\kappa_j > 0$, $j = 1,2,\dots,N$, in the complement of $\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-$.*

Endpoint behavior: *The same as in Riemann–Hilbert Problem 7.1.1.*

Jump conditions: *The boundary values $\mathbf{N}^+(\lambda)$ (resp. $\mathbf{N}^-(\lambda)$) from left (resp. from right) are related via*

$$\mathbf{N}^+(\lambda) = \mathbf{N}^-(\lambda) \begin{pmatrix} 1 & 0 \\ -2\mathrm{i}r_1(\lambda)\mathrm{e}^{2\mathrm{i}(x\lambda+4t\lambda^3)} & 1 \end{pmatrix}, \quad \lambda \in \mathrm{i}[a_j, b_j],$$
$$\mathbf{N}^+(\lambda) = \mathbf{N}^-(\lambda) \begin{pmatrix} 1 & 2\mathrm{i}r_1(\lambda)\mathrm{e}^{-2\mathrm{i}(\lambda x+4t\lambda^3)} \\ 0 & 1 \end{pmatrix}, \quad \lambda \in \mathrm{i}[-b_j, -a_j], \quad j = 1,2,\dots,g.$$

Residue conditions: *At each simple pole $\lambda = \pm\mathrm{i}\kappa_j$, $j = 1,2,\dots,N$, $\mathbf{N}(\lambda)$ satisfies*

$$\operatorname*{Res}_{\lambda=\mathrm{i}\kappa_j} \mathbf{N}(\lambda) = \lim_{\lambda\to\mathrm{i}\kappa_j} \mathbf{N}(\lambda) \begin{pmatrix} 0 & 0 \\ \mathrm{i}\chi_j \mathrm{e}^{2\mathrm{i}(\mathrm{i}\kappa_j x - 4\mathrm{i}\kappa_j^3 t)} & 0 \end{pmatrix},$$
$$\operatorname*{Res}_{\lambda=-\mathrm{i}\kappa_j} \mathbf{N}(\lambda) = \lim_{\lambda\to-\mathrm{i}\kappa_j} \mathbf{N}(\lambda) \begin{pmatrix} 0 & -\mathrm{i}\chi_j \mathrm{e}^{2\mathrm{i}(\mathrm{i}\kappa_j x - 4\mathrm{i}\kappa_j^3 t)} \\ 0 & 0 \end{pmatrix},$$

with norming constants $\chi_j \in \mathbb{R}_+ \setminus \{0\}$.

Normalization: $\mathbf{N}(\lambda) = \begin{pmatrix} 1 & 1 \end{pmatrix} + \mathrm{O}(\lambda^{-1})$ *as* $\lambda \to \infty$.

Symmetry condition: $\mathbf{N}(-\lambda) = \mathbf{N}(\lambda) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

As with Riemann–Hilbert Problem 7.1.1, we rotate the λ-plane and consider the related function $\mathbf{Y}(z) := \mathbf{N}(\mathrm{i}z)$. The jump contours and pole locations $z = \pm\kappa_j$ for $\mathbf{Y}(z)$ are then on the real axis. The solution $u(x,t)$ of the KdV equation (7.1) is obtained from (7.9). In Figure 7.8, we present the computed large-time evolution of a soliton gas supported on two pairs of bands, nonlinearly superimposed with three solitons. In the notation of (7.4), we choose $f_1(z)$ to be much larger than $f_2(z)$. We also note here that our method is by no means limited to constant functions $f_j(z)$ in (7.4) (see Figure 7.11).

7.3.1 The solution procedure

An overview of our procedure is as follows: We first ignore the poles in Riemann–Hilbert Problem 7.3.1 (in the rotated plane of $z = -\mathrm{i}\lambda$) and compute the matrix solution[3] $\mathbf{P}(z)$, normalized so that $\mathbf{P}(z) \to \mathbb{I}$ as $z \to \infty$, and use $\mathbf{P}(z)$ as a global parametrix for the RHP with poles, reducing it to a discrete finite-dimensional linear algebra problem. We solve the discrete problem, computing the solution $\mathbf{S}(z)$, and superpose its solution via $\mathbf{S}(z)\mathbf{P}(z)$ to compute the nonlinear superposition of solitons with soliton gasses. We now describe the steps in this procedure in more detail.

To handle potential exponential growth in the residue conditions, we flip their triangularities as needed via the transformation

$$\hat{\mathbf{Y}}(z) = \mathbf{Y}(z)v(z)^{\boldsymbol{\sigma}_3}, \quad v(z) = \prod_{j\in K} \frac{z-\kappa_j}{z+\kappa_j},$$

where K is an index set corresponding to "large" residue conditions. In particular, for some constant $c > 0$, we define

$$K = K(x,t) = \left\{ j : \left| \chi_j \mathrm{e}^{8\kappa_j^3 t - 2\kappa_j x} \right| > c \right\},$$

in the quiescent region and

$$K = \left\{ j : \left| \chi_j \mathrm{e}^{8\kappa_j^3 t - 2\kappa_j x + 2\mathfrak{g}(\kappa_j) + 2\mathfrak{h}(\kappa_j)} \right| > c \right\},$$

in the unmodulated region. Our implementation takes $c = 10$ as the default value.

[3] A matrix solution ceases to exist at a set of isolated points (x,t), but we ignore this behavior for our numerical purposes just as in [34] and [160].

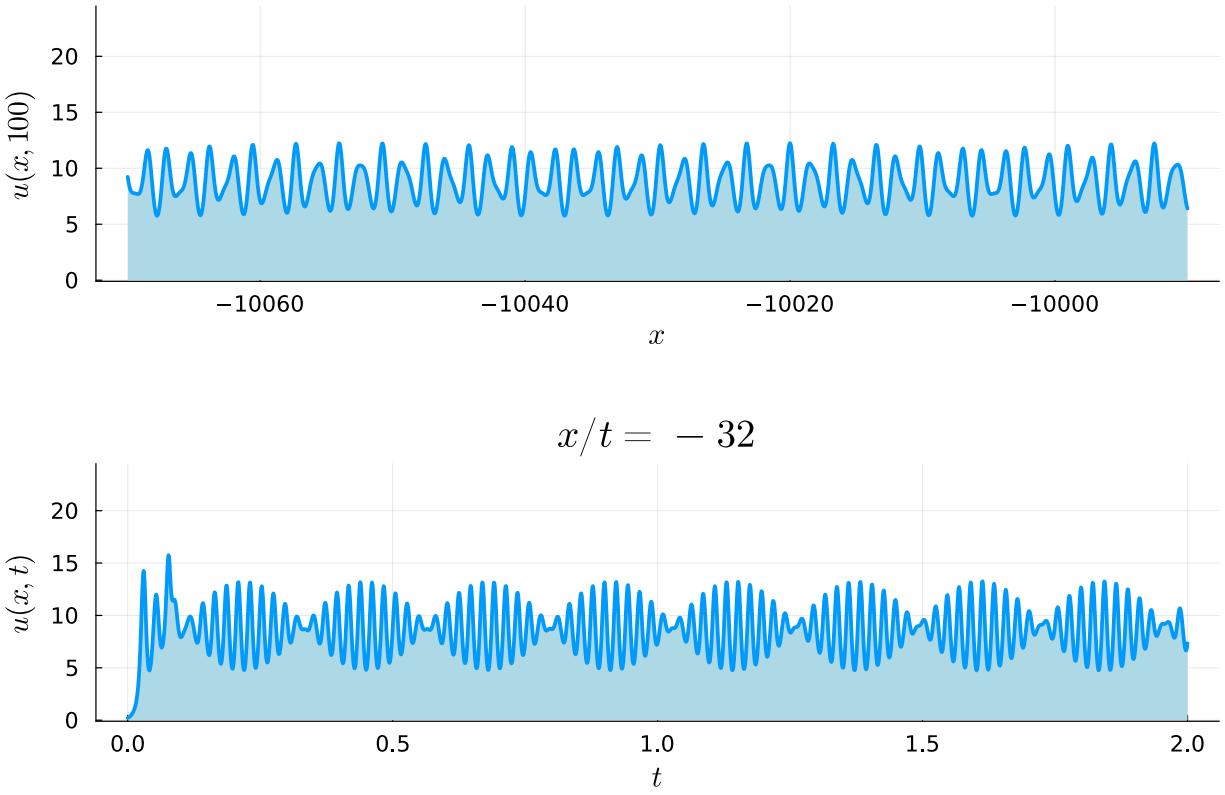


Figure 7.8: A KdV soliton gas with $r_1(\lambda)$ supported on two pairs of bands. In the notation of (7.4), $I_1 = [1, 2]$, $I_2 = [2.5, 3]$ with $f_1(z) = 100$, $f_2(z) = 1$ and $\mathfrak{a}_j = \mathfrak{b}_j = \frac{1}{2}$, $j = 1, 2$. The three solitons superimposed are associated with $\kappa_1 = 0.8$, $\kappa_2 = 2.25$, and $\kappa_3 = 3.5$ and the norming constants $\chi_1 = 10^6$, $\chi_2 = 10^5$, $\chi_3 = 10^{-12}$. Bottom panel: The same solution plotted along the ray $x/t = -32$.

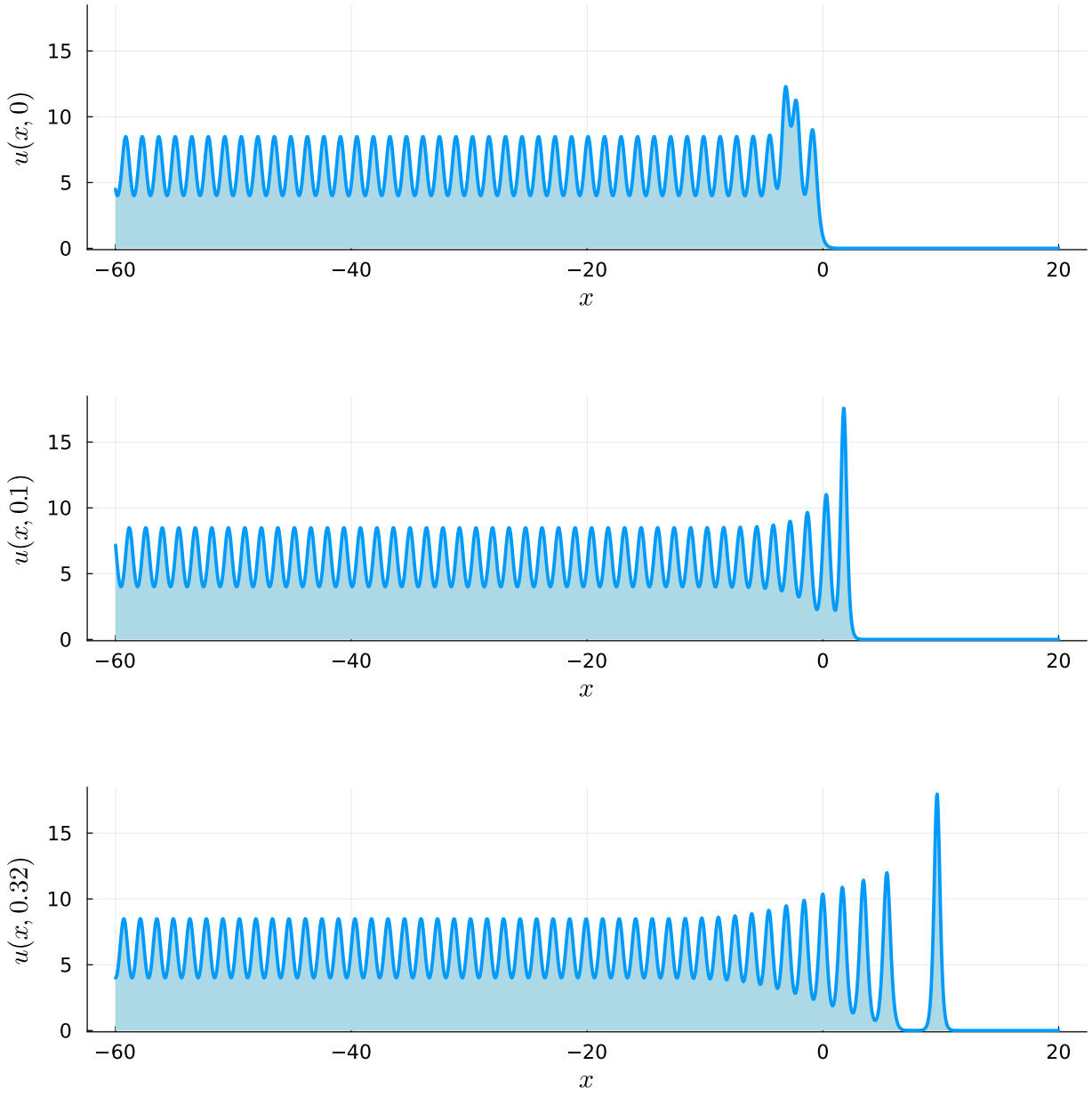


Figure 7.9: A KdV soliton gas with $r_1(\lambda)$ supported on one pair of bands. In the notation of (7.4), $I_1 = [1.5, 2.5]$ with $f_1(z) = 1$ and $\mathfrak{a}_1 = \mathfrak{b}_1 = \frac{1}{2}$. The superimposed soliton is associated with $\kappa_1 = 3$ and the norming constant $\chi_1 = 10^{-4}$.

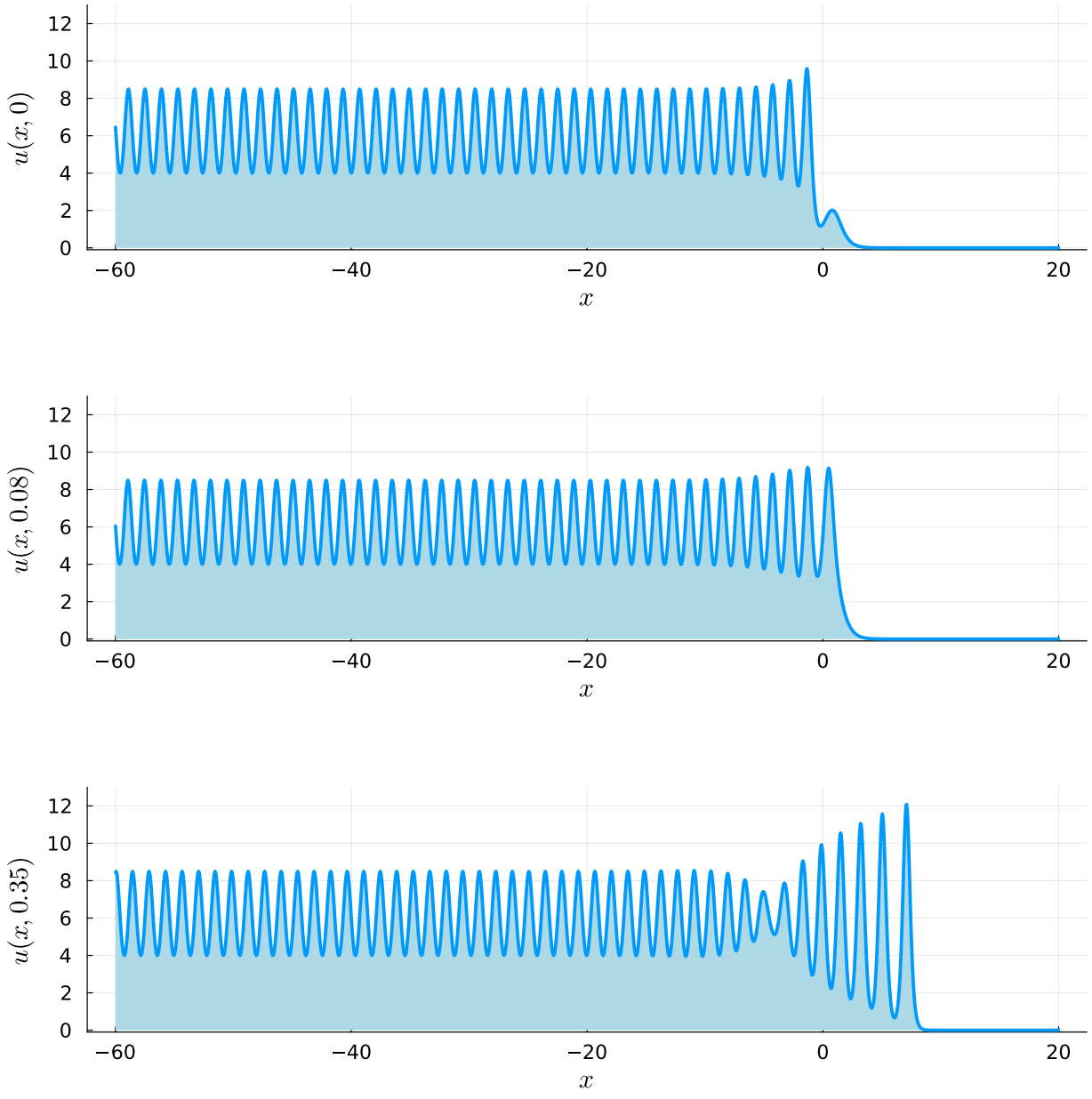


Figure 7.10: A KdV soliton gas with $r_1(\lambda)$ supported on one pair of bands. In the notation of (7.4), $I_1 = [1.5, 2.5]$ with $f_1(z) = 1$ and $\mathfrak{a}_1 = \mathfrak{b}_1 = \frac{1}{2}$. The superimposed soliton is associated with $\kappa_1 = 1$ and the norming constant $\chi_1 = 10$.

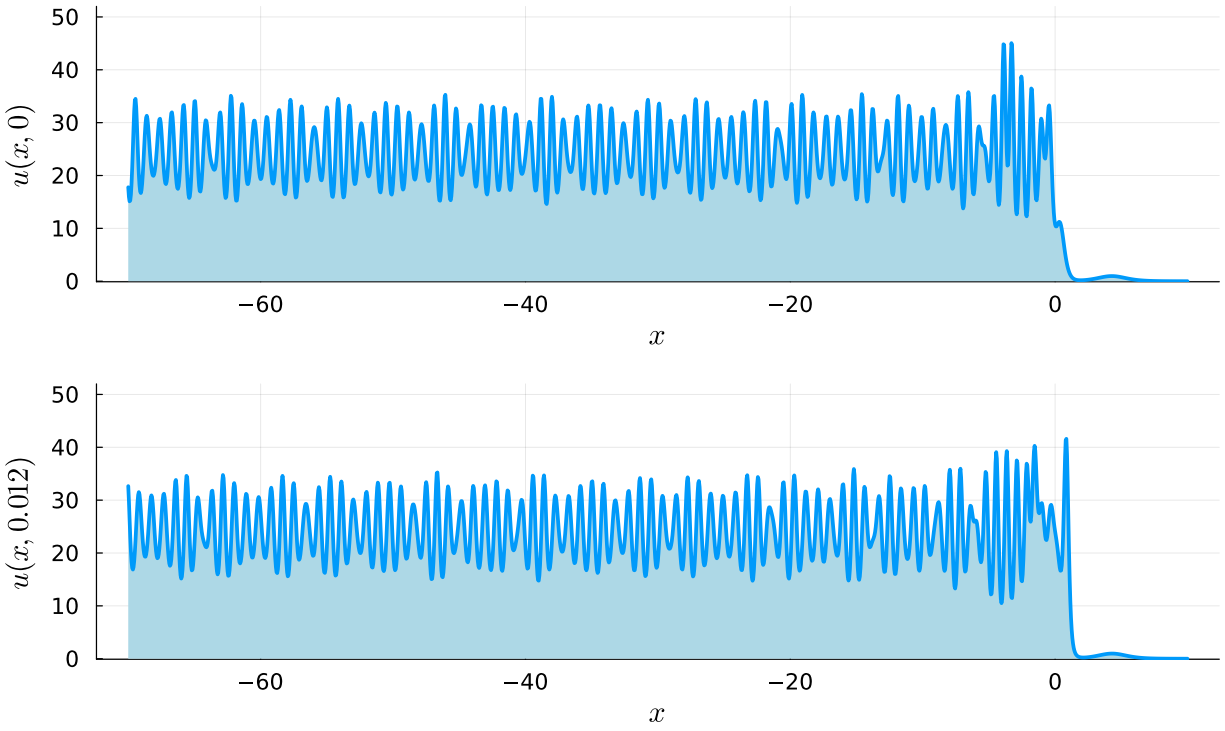


Figure 7.11: A KdV soliton gas with $r_1(\lambda)$ supported on five pairs of bands, nonlinearly superimposed with five solitons. In the notation of (7.4), $I_1 = [0.25, 0.5]$, $I_2 = [0.8, 1.2]$, $I_3 = [1.5, 2]$, $I_4 = [2.5, 3]$, and $I_5 = [4, 5]$ with $f_1(z) = (z - 0.375)^2 + 1$, $f_2(z) = (z - 1)^4 + 1$, $f_3(z) = (z - 1.75)^6 + 1$, $f_4(z) = \exp(z - 2.75) + 1$, $f_5(z) = \exp(-(z - 4.5)^2) + 1$ and $\mathfrak{a}_1 = \mathfrak{b}_1 = -\frac{1}{2}$, $\mathfrak{a}_2 = \mathfrak{b}_2 = \frac{1}{2}$, $\mathfrak{a}_3 = \frac{1}{2}$, $\mathfrak{b}_3 = -\frac{1}{2}$, $\mathfrak{a}_4 = -\frac{1}{2}$, $\mathfrak{b}_4 = \frac{1}{2}$, $\mathfrak{a}_5 = \mathfrak{b}_5 = -\frac{1}{2}$. The five solitons are associated with the eigenvalue parameters $\kappa_1 = 0.1$, $\kappa_2 = 0.7$, $\kappa_3 = 2.25$, $\kappa_4 = 3.5$, $\kappa_5 = 5.5$ and the norming constants $\chi_1 = 10^5$, $\chi_2 = 1000$, $\chi_3 = 100$, $\chi_4 = 10$, $\chi_5 = 10^{-6}$.

This transformation has the effect of modifying the residue conditions to

$$\operatorname*{Res}_{z=\kappa_j} \hat{\mathbf{Y}}(z) = \lim_{z\to\kappa_j} \hat{\mathbf{Y}}(z) \begin{pmatrix} 0 & 0 \\ \chi_j \mathrm{e}^{8\kappa_j^3 t - 2\kappa_j x} v(\kappa_j)^2 & 0 \end{pmatrix}, \quad j \notin K,$$
$$\operatorname*{Res}_{z=\kappa_j} \hat{\mathbf{Y}}(z) = \lim_{z\to\kappa_j} \hat{\mathbf{Y}}(z) \begin{pmatrix} 0 & \frac{\mathrm{e}^{-8\kappa_j^3 t + 2\kappa_j x}}{\chi_j v'(\kappa_j)^2} \\ 0 & 0 \end{pmatrix}, \quad j \in K.$$

The residue conditions on the negative real axis follow from the symmetry condition.

In the unmodulated region, the disjoint disks D_j are assumed to be sufficiently small so as not to contain the poles κ_j. The steepest descent deformations then result in the residue conditions

$$\operatorname*{Res}_{z=\kappa_j} \mathbf{R}(z) = \lim_{z\to\kappa_j} \mathbf{R}(z) \begin{pmatrix} 0 & 0 \\ \chi_j \mathrm{e}^{8\kappa_j^3 t - 2\kappa_j x + 2\mathfrak{g}(\kappa_j) + 2\mathfrak{h}(\kappa_j)} v(\kappa_j)^2 & 0 \end{pmatrix}, \quad j \notin K,$$
$$\operatorname*{Res}_{z=\kappa_j} \mathbf{R}(z) = \lim_{z\to\kappa_j} \mathbf{R}(z) \begin{pmatrix} 0 & \frac{\mathrm{e}^{-8\kappa_j^3 t + 2\kappa_j x - 2\mathfrak{g}(\kappa_j) - 2\mathfrak{h}(\kappa_j)}}{\chi_j v'(\kappa_j)^2} \\ 0 & 0 \end{pmatrix}, \quad j \in K.$$

Once the matrix solution $\mathbf{P}$ is obtained, the residue conditions that $\mathbf{S}(z) = \mathbf{R}(z)\mathbf{P}(z)^{-1}$ satisfies are computed through the following formula: Given a residue condition

$$\operatorname*{Res}_{z=\pm\kappa_j} \mathbf{R}(z) = \lim_{z\to\pm\kappa_j} \mathbf{R}(z)\boldsymbol{\Xi}_{\pm j},$$

of $\mathbf{R}$, the corresponding residue condition of $\mathbf{S}$ is given by

$$\operatorname*{Res}_{z=\pm\kappa_j} \mathbf{S}(z) = \lim_{z\to\pm\kappa_j} \mathbf{S}(z)\mathbf{P}(\pm\kappa_j)\boldsymbol{\Xi}_{\pm j}\mathbf{P}(\pm\kappa_j)^{-1} \left(\mathbb{I} - \mathbf{P}'(\pm\kappa_j)\boldsymbol{\Xi}_{\pm j}\mathbf{P}(\pm\kappa_j)^{-1}\right)^{-1},$$

where $\mathbf{P}'(z)$ denotes the componentwise derivative of $\mathbf{P}(z)$. We compute $\mathbf{P}'(\pm\kappa_j)$ by numerically expanding $\mathbf{P}(z)$ in a Laurent series in a small circle centered at $\pm\kappa_j$.

7.4 Examples, demonstration of convergence, and performance

7.4.1 Examples

In this subsection, we enumerate the examples found throughout this chapter.

5 pairs of bands with 5 solitons

In Figures 7.1 and 7.11, we plot a soliton gas corresponding to $r_1(\lambda)$ supported on five pairs of bands, nonlinearly superimposed with five solitons. In the notation of (7.4), Figure 7.11 considers variable f_j and various endpoint behaviors $\mathfrak{a}_j, \mathfrak{b}_j$.

5 pairs of bands

In Figure 7.2, we plot a pure soliton gas corresponding to $r_1(\lambda)$ supported on five pairs of bands, including for large negative x at $t = 100$.

1 pair of bands with 2 solitons

In Figure 7.3, we consider a soliton gas corresponding to $r_1(\lambda)$ supported on one pair of bands, nonlinearly superimposed with two solitons. We include a density plot of this solution in the (x, t)-plane outside of the wedge-shaped region where our method begins to break down.

2 pairs of bands

In Figure 7.4, we consider a pure soliton gas corresponding to $r_1(\lambda)$ supported on two pairs of bands. We plot its tail at $t = 100$ and its behavior along the ray $x/t = -32$.

2 pairs of bands with 3 solitons

In Figure 7.8, we include the same plots as Figure 7.4, but with the solution nonlinearly superimposed with three solitons.

1 pair of bands with a taller soliton

In Figure 7.9, we plot the time evolution of a soliton gas corresponding to $r_1(\lambda)$ supported on one pair of bands, nonlinearly superimposed with one taller soliton, showing the soliton escape the soliton gas.

1 pair of bands with a shorter soliton

In Figure 7.10, we plot the time evolution of a soliton gas corresponding to $r_1(\lambda)$ supported on one pair of bands, nonlinearly superimposed with one shorter soliton, showing the soliton becoming trapped within the soliton gas.

7.4.2 Convergence

In Figure 7.12, we demonstrate the convergence of our computational method as we increase the number of collocation points used on each component of the jump contour associated with the RHP that is treated numerically. We plot the absolute pointwise error made in computing a soliton gas potential

$$E_{\mathrm{p}}(x) := |u_{\mathrm{p}}(x, 0) - u_{\mathrm{true}}(x, 0)|,$$

where p refers to the number of collocation points used on each interval, denoted by "PPI" in Figure 7.12. Here, $u_{\mathrm{true}}(x, 0)$ denotes the solution computed with a large value of p. The

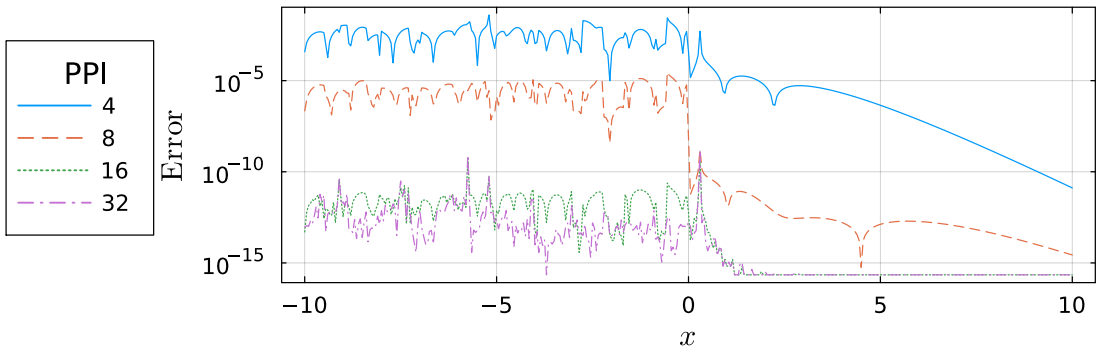


Figure 7.12: Pointwise errors for various numbers of collocation points. PPI (Points Per Interval) denotes the number of collocation points used per interval I_j, and 10 times that number of points are used on each circle. 50 PPI is used as the exact solution in comparisons.

soliton gas potential computed for Figure 7.12 is associated with $r_1(\lambda)$ supported on two pairs of bands (in the notation of (7.4)) $I_1 = [1.2, 2]$, $I_2 = [2.5, 3]$ with $f_1(z) = 100$ and $f_2(z) = 1$ and $\mathfrak{a}_j = \mathfrak{b}_j = \frac{1}{2}$, $j = 1, 2$. This soliton gas potential is also nonlinearly superimposed with two solitons associated with the eigenvalue parameters $\kappa_1 = 1$, $\kappa_2 = 4$ and the norming constants $\chi_1 = 10$, $\chi_2 = 10^{-10}$. While we observe that our method is quite accurate, some precision is lost due to the large condition number of the collocation matrix (empirically on the order of 10^7). Future work will aim to improve the conditioning of this linear system.

Finally, in all of the plots presented in this chapter, 20 collocation points are used on each interval and 120 points are used on each circle in the jump contour for the RHP treated numerically.

7.4.3 Notes on performance

Perhaps the most attractive feature of our method is that it is trivially parallelizable. Since x and t are only parameters in the RHP, all computations in this chapter can, in principle, be done in the time it takes to solve a single RHP, i.e., a pointwise evaluation of $u(x, t)$. With this perspective in mind, we give rough pointwise evaluation runtimes on a standard laptop for the solution plots above in Table 7.1.

7.5 Conclusion

This chapter introduced a fast and accurate method for computing KdV soliton gas solutions via the numerical solution of the associated Riemann–Hilbert problem. The key contribution is the use of Deift–Zhou steepest-descent deformations to precondition the singular integral equations, avoiding the high-precision arithmetic required by earlier approaches [61]. The

	Unmodulated region	**Quiescent region**
Figures 7.1 and 7.11	0.7 s	0.08 s
Figure 7.2	0.4 s	2 ms
Figure 7.3	0.03 s	5 ms
Figure 7.4	0.02 s	0.4 ms
Figure 7.8	0.1 s	0.02 s
Figures 7.9 and 7.10	0.02 s	3 ms

Table 7.1: Pointwise evaluation runtimes in seconds (s) or milliseconds (ms) to compute the solutions $u(x,t)$ presented in each figure listed in Section 7.4.1.

$\mathfrak{g}$-function and its correction $\mathfrak{h}$, constructed in the folded plane $\zeta = z^2$ as described in Section 7.2.2, exploit the odd symmetry of the problem to halve the effective linear system sizes and reduce the domain of integration from $\Sigma_+ \cup \Sigma_-$ to $\hat{\Sigma}$. The method treats the quiescent region (Section 7.2.1), where no deformations are needed, and the unmodulated region (Section 7.2.2), covering $x < 0$ and small to moderate t, and handles nonlinear superpositions with finitely many solitons via the residue conditions of Section 7.3.

The primary limitation of the method is its restriction to the complement of a wedge-shaped region in the (x,t)-plane: the steepest-descent deformations break down when $\xi = x/(4t)$ enters the modulated regime where the effective genus of the g-function changes. This limitation is overcome in Chapter 8, where the modulation parameter is computed and the g-function is adapted accordingly, enabling uniform accuracy throughout the entire (x,t)-plane for both KdV and mKdV soliton gasses.

Chapter 8

LONG-TIME COMPUTATIONS OF KDV AND MKDV SOLITON GASSES

8.1 Introduction

This chapter is based on a manuscript in preparation [17].

In this chapter, we develop a numerical method that extends that of Chapter 7 to be valid in the entire (x,t)-plane. Due to technicalities, we present the method in the context of the mKdV equation; the necessary modifications for the KdV case are described in Section 8.4. The corresponding definition of an mKdV soliton gas, in the same setting as before, is given by the following RHP.

Riemann–Hilbert Problem 8.1.1 (Pure mKdV soliton gas)**.** *Let $(x,t) \in \mathbb{R}\times\mathbb{R}_+$ be fixed. Find a 2×2 matrix-valued function $\mathbf{M}(\lambda) \equiv \mathbf{M}(\lambda;x,t)$ with the following properties:*

> ***Analyticity:*** *$\mathbf{M}(\lambda)$ is analytic for $\lambda \in \mathbb{C}\setminus(\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-)$, admitting continuous boundary values on the interior of $\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-$.*
>
> ***Jump conditions:*** *The boundary values $\mathbf{M}^+(\lambda)$ (resp. $\mathbf{M}^-(\lambda)$) from the left (resp. right) are related via*
>
> $$\mathbf{M}^+(\lambda) = \mathbf{M}^-(\lambda)\begin{pmatrix} 1 & 0 \\ 2\mathrm{i}r_1(\lambda)\mathrm{e}^{2\mathrm{i}(x\lambda+4t\lambda^3)} & 1\end{pmatrix}, \quad \lambda\in\mathrm{i}\Sigma_+,$$
> $$\mathbf{M}^+(\lambda) = \mathbf{M}^-(\lambda)\begin{pmatrix} 1 & 2\mathrm{i}r_1(\lambda)\mathrm{e}^{-2\mathrm{i}(x\lambda+4t\lambda^3)} \\ 0 & 1\end{pmatrix}, \quad \lambda\in\mathrm{i}\Sigma_-, \quad j=1,2,\dots,g.$$
>
> ***Normalization:*** *$\mathbf{M}(\lambda) = \mathbf{I} + \mathrm{O}(\lambda^{-1})$ as $\lambda\to\infty$.*

As in the KdV case, Riemann–Hilbert Problem 8.1.1 has a unique solution [79]. Given a solution $\mathbf{M}$ to Riemann–Hilbert Problem 8.1.1, the soliton gas solution of (1.4) is recovered from the residue at infinity:

$$u(x,t) = \lim_{\lambda\to\infty} 2\mathrm{i}\lambda\mathbf{M}_{12}(\lambda;x,t).$$

As in Chapter 7, we rotate the λ-plane clockwise by defining $z=-\mathrm{i}\lambda$ and $r(z)=2r_1(\mathrm{i}z)$. The numerical method we describe below solves for $\mathbf{Y}(z)=\mathbf{M}(\mathrm{i}z)$ with the convention that all (real) intervals are oriented in the increasing direction.

See Figure 8.1 for a soliton gas solution of (1.3) with associated Riemann–Hilbert jump conditions supported on five (pairs of) bands and nonlinearly superimposed with five solitons, computed with our method. This is a longer-time evolution of Figure 7.1. Figure 8.2 provides the density plot of a soliton gas with associated Riemann–Hilbert jump conditions supported on a single pair of bands, nonlinearly superimposed with two solitons. Our numerical method is demonstrated to be uniformly accurate in the wedge-shaped region where that of Chapter 7 breaks down.

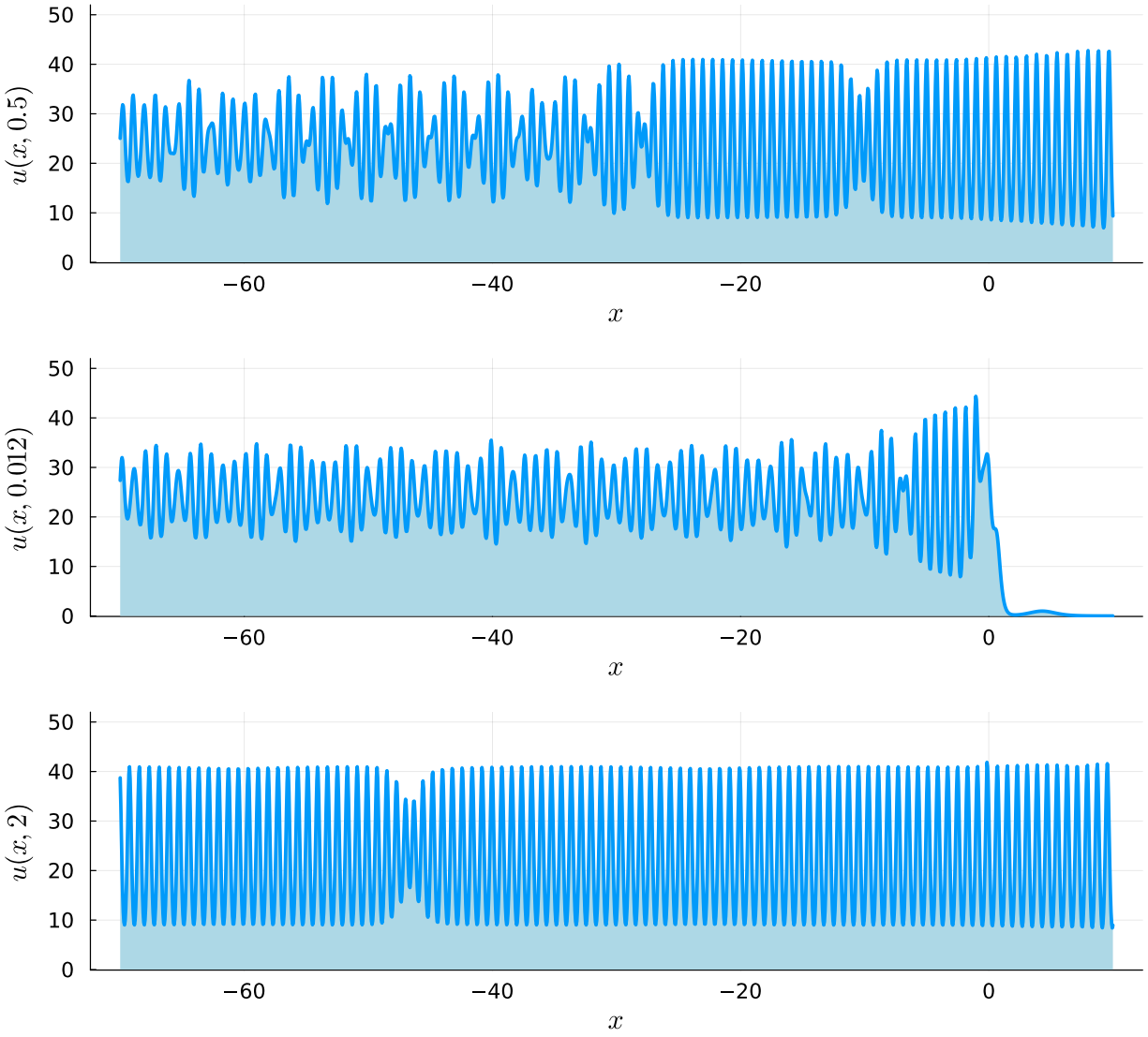


Figure 8.1: KdV soliton gas with $r_1(\lambda)$ supported on five pairs of bands, nonlinearly superimposed with five solitons (cf. Figure 7.1).

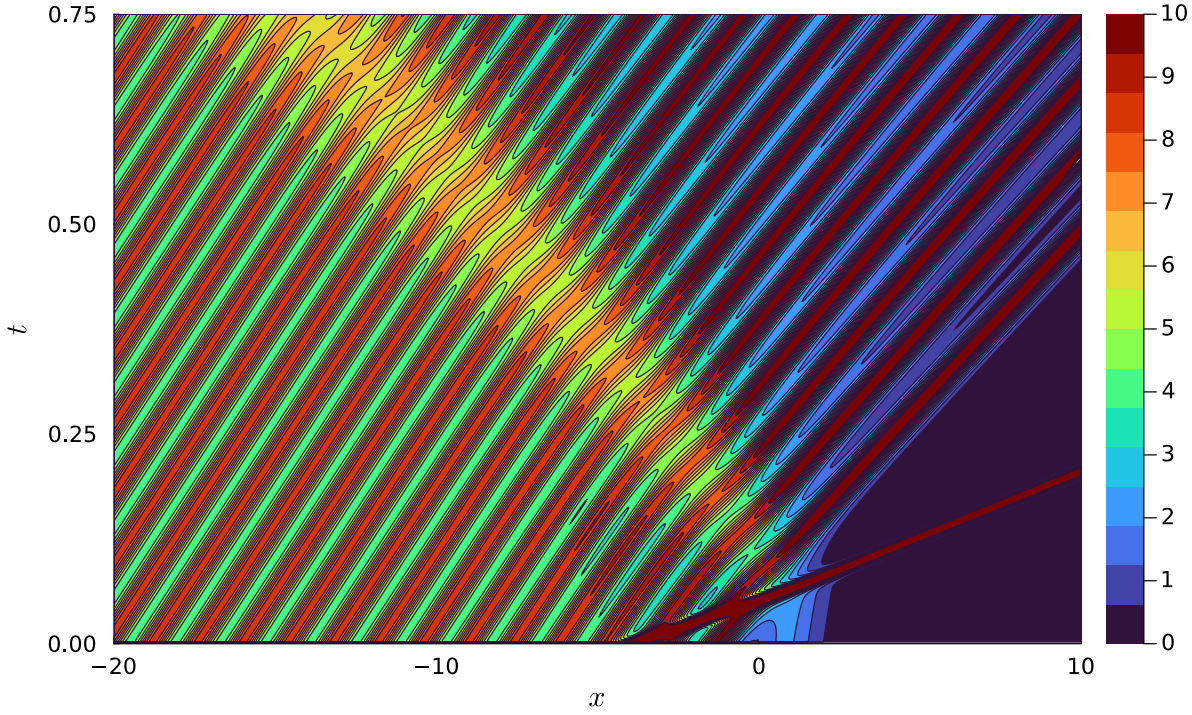


Figure 8.2: Density plot of a KdV soliton gas with $r_1(\lambda)$ supported on a single pair of bands with two additional solitons (cf. Figure 7.3).

Our method uses Deift–Zhou steepest descent deformations to, in effect, precondition the singular integral formulations of Riemann–Hilbert Problem 7.1.1 and Riemann–Hilbert Problem 8.1.1. The deformations we employ vary in the (x,t)-plane and depend on the parameter $\xi := \frac{x}{4t}$. In particular, we partition the (x,t)-plane into wedges as in Figure 8.3, where the critical values $\xi_{1,1}, \xi_{1,2}, \xi_{2,1}, \ldots, \xi_{g,2}$ are computed as described in Appendix B, and apply different deformations in each region. In the unmodulated regions, the support of the g-function is fixed, while in the modulated regions, one must first compute the modulation parameter $\alpha_j(\xi)$ as described in Appendix B to determine the support of the g-function. The method of Chapter 7 is uniformly accurate in the genus 0 and genus g unmodulated regions and can be easily extended to the remaining unmodulated regions. In this chapter, we describe a numerical method that is valid in the modulated regions, enabling long-time computations in the entire (x,t)-plane.

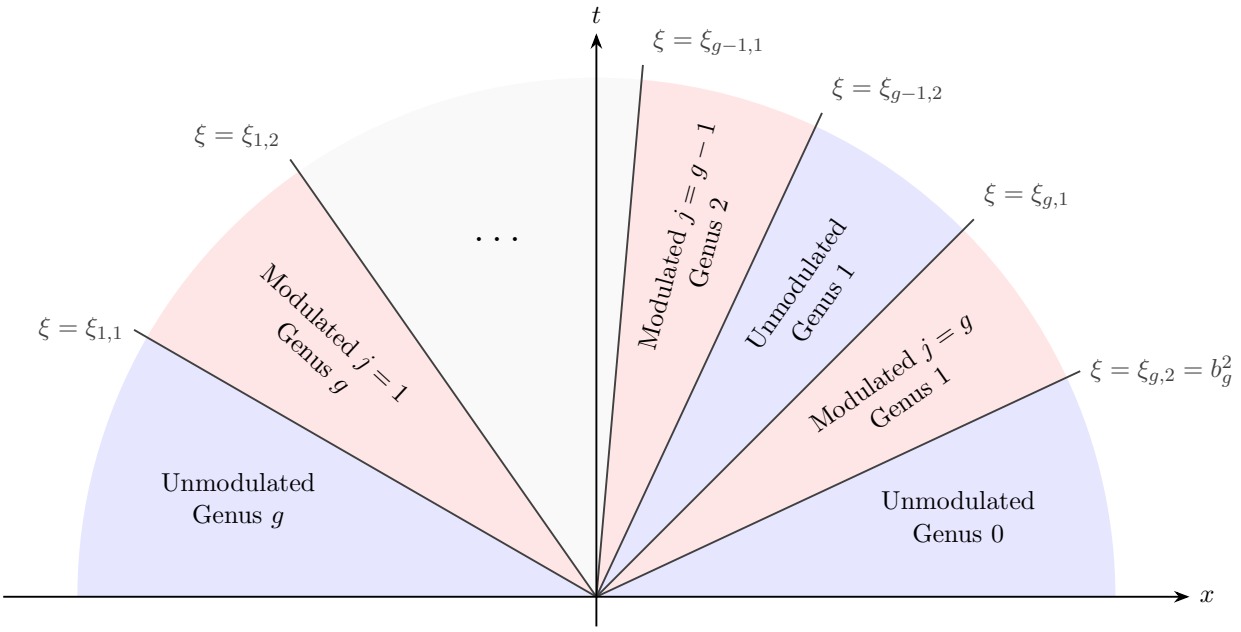


Figure 8.3: Deformation regions of the (x,t)-plane for a genus g soliton gas. The wedge-shaped regions are bounded by rays of $\xi = \frac{x}{4t}$, with critical values $\xi_{j,k}$ computed as described in Appendix B. With an appropriate g-function, soliton gas solutions in the unmodulated regions can be evaluated via the method of Chapter 7, while the current work develops a numerical method valid in the modulated regions.

8.2 Steepest descent deformations

In this section, we describe the steepest descent deformations that we perform on Riemann–Hilbert Problem 8.1.1 to solve the mKdV equation (1.4). These deformations depend on the modulation parameter $\alpha_j(\xi)$ described in Appendix B. The necessary modifications for the KdV equation (1.3) and Riemann–Hilbert Problem 7.1.1 are discussed in Section 8.4.

Define

$$\mathbf{J}_L(z;\xi,t) = \begin{pmatrix} 1 & 0 \\ \mathrm{i}r(z)\mathrm{e}^{-8t\theta(z;\xi)} & 1 \end{pmatrix}, \quad \mathbf{J}_U(z;\xi,t) = \begin{pmatrix} 1 & \mathrm{i}r(z)\mathrm{e}^{8t\theta(z;\xi)} \\ 0 & 1 \end{pmatrix}.$$

To solve Riemann–Hilbert Problem 8.1.1 in the rotated $z = -\mathrm{i}\lambda$ plane, we seek a 2×2 matrix-valued function $\mathbf{Y}(z;\xi,t) = \mathbf{Y}(z)$ satisfying

$$\begin{aligned} \mathbf{Y}^+(z) &= \mathbf{Y}^-(z)\mathbf{J}_L(z;\xi,t), \quad z\in\Sigma_+, \qquad \theta(z;\xi) := \xi z - z^3, \\ \mathbf{Y}^+(z) &= \mathbf{Y}^-(z)\mathbf{J}_U(z;\xi,t), \quad z\in\Sigma_-, \\ \mathbf{Y}(z) &= \mathbf{I} + \mathrm{O}\left(z^{-1}\right), \qquad z\to\infty, \end{aligned}$$

that is analytic elsewhere.

To describe the steepest descent deformations, we assume that ξ is within the jth modulated region, i.e. $\xi \in (\xi_{j,1}, \xi_{j,2})$. For ξ in an unmodulated region, local deformations around α_j are no longer needed, and the deformations around the intervals not containing α_j remain unchanged.

We introduce open and disjoint disks D_k, $k = 1, 2, \ldots, g$, chosen so that D_k encloses $[a_k, b_k]$ in its interior and let D_{-k} be the analog of D_k for $[-b_k, -a_k]$. We also let $\tilde{D}_j = \tilde{D}_j(\alpha_j)$ be an open disk, disjoint from all D_k, chosen such that $(\alpha_j, b_j] \subset \tilde{D}_j$ and $\alpha_j \in \partial\tilde{D}_j$. We denote its counterpart by $\tilde{D}_{-j}$ and take all disk boundaries to be oriented counterclockwise.

Now, define

$$\mathbf{T}_L(z;\xi,t) = \begin{pmatrix} 0 & \frac{\mathrm{i}}{r(z)}\mathrm{e}^{8t\theta(z;\xi)} \\ \mathrm{i}r(z)\mathrm{e}^{-8t\theta(z;\xi)} & 0 \end{pmatrix}, \quad \mathbf{L}(z;\xi,t) = \begin{pmatrix} 1 & 0 \\ \frac{\mathrm{i}}{r(z)}\mathrm{e}^{-8t\theta(z;\xi)} & 1 \end{pmatrix},$$
$$\mathbf{T}_U(z;\xi,t) = \begin{pmatrix} 0 & \mathrm{i}r(z)\mathrm{e}^{8t\theta(z;\xi)} \\ \frac{\mathrm{i}}{r(z)}\mathrm{e}^{-8t\theta(z;\xi)} & 0 \end{pmatrix}, \quad \mathbf{U}(z;\xi,t) = \begin{pmatrix} 1 & \frac{\mathrm{i}}{r(z)}\mathrm{e}^{8t\theta(z;\xi)} \\ 0 & 1 \end{pmatrix},$$

corresponding to the canonical LTL and UTU factorizations, so

$$\mathbf{J}_L = \mathbf{U}\mathbf{T}_L\mathbf{U}, \quad \mathbf{J}_U = \mathbf{L}\mathbf{T}_U\mathbf{L}.$$

We define $\mathbf{Z}(z;\xi,t) = \mathbf{Z}(z)$ by

$$\mathbf{Z}(z) := \begin{cases} \mathbf{Y}(z)\mathbf{U}(z;\xi,t)^{\mp 1}, & z \in \left(\tilde{D}_j \cup \bigcup_{k=j+1}^{g} D_k\right) \cap \mathbb{C}^{\pm}, \\ \mathbf{Y}(z)\mathbf{L}(z;\xi,t)^{\mp 1}, & z \in \left(\tilde{D}_{-j} \cup \bigcup_{k=j+1}^{g} D_{-k}\right) \cap \mathbb{C}^{\pm}, \\ \mathbf{Y}(z), & \text{otherwise.} \end{cases}$$

Because $r^+ = -r^-$ along the branch cuts, $\mathbf{Z}$ satisfies the jump conditions given in Figure 8.4.

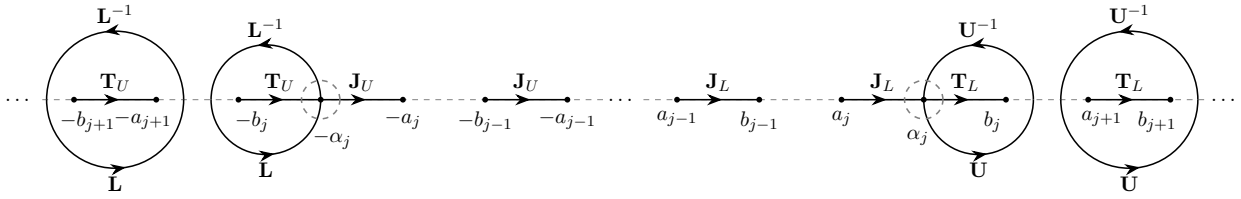


Figure 8.4: Jump contours and matrices associated with $\mathbf{Z}(z;\xi,t)$ near each interval.

To move the jump conditions away from the singularities at $\pm\alpha_j$, we solve the RHP for $\mathbf{Z}$ locally. That is, we define a new function $\widetilde{\mathbf{Z}}(z;\xi,t)$ in the circles around $\pm\alpha_j$ according

to Figure 8.5 and set $\widetilde{\mathbf{Z}} = \mathbf{Z}$ outside of these circles. Then, $\widetilde{\mathbf{Z}}$ satisfies the jump conditions

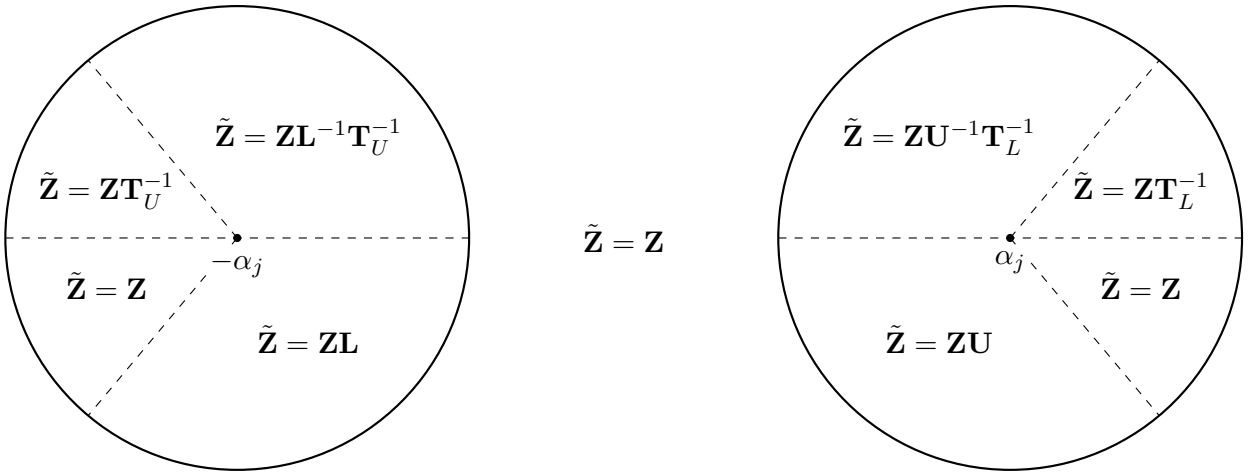


Figure 8.5: Local definitions of $\widetilde{\mathbf{Z}}(z;\xi,t)$.

given in Figure 8.6 near $\pm\alpha_j$ and the same jump conditions as $\mathbf{Z}$ elsewhere.

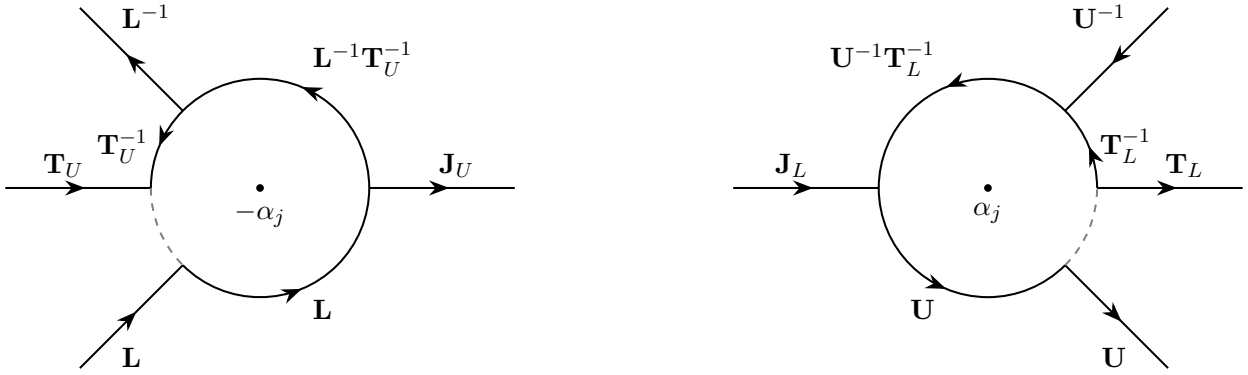


Figure 8.6: Jump contours and matrices associated with $\widetilde{\mathbf{Z}}(z;\xi,t)$ near $\pm\alpha_j$.

The functions $\mathfrak{g}(z;\xi)$ and $\mathfrak{h}(z;\xi,t)$ are constructed exactly as in Section 7.2.2 of Chapter 7, working in the folded plane $\zeta = z^2$. The only modification is that in the jth modulated region the left endpoint a_j is replaced throughout by the modulation parameter $\alpha_j(\xi)$ from Appendix B. The g-function support in the jth modulated region is

$$\begin{cases} [\alpha_j^2, b_j^2] \cup \bigcup_{k=j+1}^{g} [a_k^2, b_k^2], & j < g, \\ [\alpha_j^2, b_g^2], & j = g, \end{cases}$$

replacing the fixed support $\hat{\Sigma} = \bigcup_{k=1}^{g}[a_k^2, b_k^2]$ of Chapter 7. By convention, we set $\mathfrak{g}$ and $\mathfrak{h}$ to be identically zero in the $(g+1)$th modulated region.

To remove the exponential growth, denote

$$\mathbf{P}(z;\xi,t) = \mathrm{e}^{(4t\mathfrak{g}(z;\xi)+\mathfrak{h}(z;\xi,t))\sigma_3}, \quad \mathbf{Q}(z;\xi,t) = \mathrm{e}^{4t\theta(z;\xi)\sigma_3}, \quad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Now, define $\mathbf{S}(z;\xi,t) = \mathbf{S}(z)$ by

$$\mathbf{S} := \begin{cases} \widetilde{\mathbf{Z}}\mathbf{Q}, & \text{inside the circles centered at } \pm\alpha_j, \\ \widetilde{\mathbf{Z}}\mathbf{P}, & \text{otherwise.} \end{cases}$$

Then, $\mathbf{S}$ satisfies the jump conditions given in Figure 8.7. These jump conditions can be

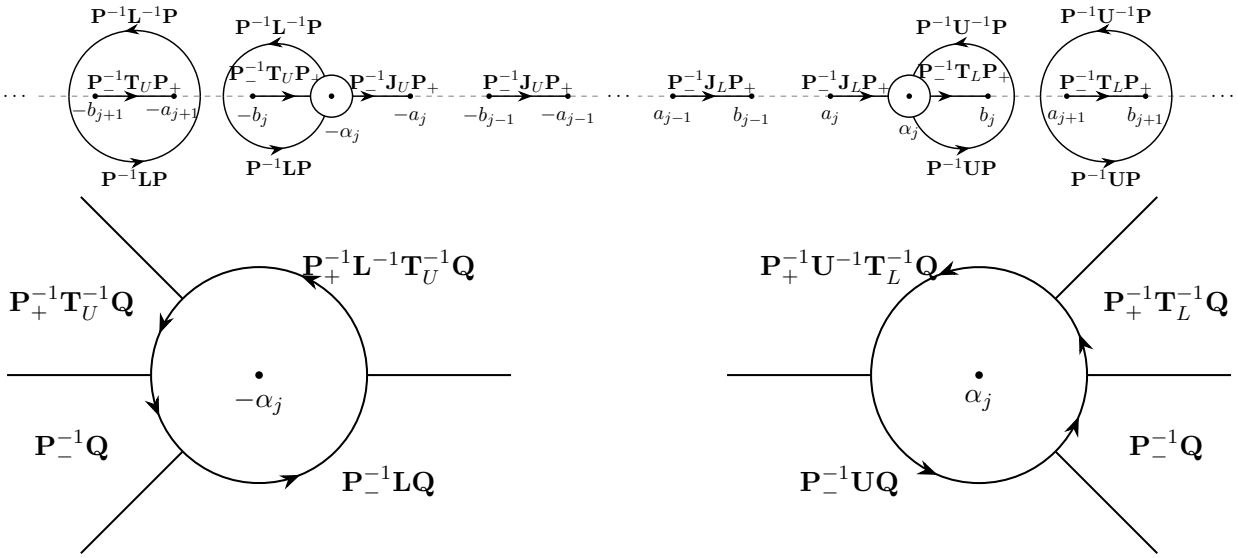


Figure 8.7: Jump contours and matrices associated with $\mathbf{S}(z;\xi,t)$.

written in terms of the function

$$\phi(z;\xi,t) = 4t\left(\mathfrak{g}(z;\xi) - \theta(z;\xi)\right) + \mathfrak{h}(z;\xi,t),$$

which is controlled at the singularities $\pm\alpha_j$. On the appropriate contours, we have

$$\mathbf{P}_-^{-1}\mathbf{T}_L\mathbf{P}_+ = \begin{pmatrix} 0 & \frac{\mathrm{i}}{\mathrm{e}^{A_k(\xi,t)}r} \\ \mathrm{i}\mathrm{e}^{A_k(\xi,t)}r & 0 \end{pmatrix}, \qquad \mathbf{P}^{-1}\mathbf{U}\mathbf{P} = \begin{pmatrix} 1 & -\frac{\mathrm{i}}{r}\mathrm{e}^{-2\phi} \\ 0 & 1 \end{pmatrix},$$
$$\mathbf{P}_-^{-1}\mathbf{J}_L\mathbf{P}_+ = \begin{pmatrix} 1 & 0 \\ \mathrm{i}r\mathrm{e}^{\phi_+ + \phi_-} & 1 \end{pmatrix}, \qquad \mathbf{P}_+^{-1}\mathbf{T}_L^{-1}\mathbf{Q} = \begin{pmatrix} 0 & -\frac{\mathrm{i}}{r}\mathrm{e}^{-\phi_+} \\ -\mathrm{i}r\mathrm{e}^{\phi_+} & 0 \end{pmatrix},$$
$$\mathbf{P}_+^{-1}\mathbf{U}^{-1}\mathbf{T}_L^{-1}\mathbf{Q} = \begin{pmatrix} \mathrm{e}^{-\phi_+} & -\frac{\mathrm{i}}{r}\mathrm{e}^{-\phi_+} \\ -\mathrm{i}r\mathrm{e}^{\phi_+} & 0 \end{pmatrix}, \qquad \mathbf{P}_-^{-1}\mathbf{U}\mathbf{Q} = \begin{pmatrix} \mathrm{e}^{-\phi_-} & -\frac{\mathrm{i}}{r}\mathrm{e}^{-\phi_-} \\ 0 & \mathrm{e}^{\phi_-} \end{pmatrix}.$$

Thus, all jump conditions of the function $\mathbf{S}$ are controlled, and we will solve the corresponding RHP numerically.

8.3 Numerical implementation

8.3.1 Solution procedure

While the jump conditions in the RHP describing $\mathbf{S}$ are controlled, the nonlocal jump conditions $\mathbf{P}_-^{-1}\mathbf{T}_L\mathbf{P}_+$ and $\mathbf{P}_-^{-1}\mathbf{T}_U\mathbf{P}_+$ cannot approach the identity in any regime. To obtain uniform accuracy as in [159, Chapter 7], we implement a two-stage solution procedure that first solves for these jump conditions via the method of Chapter 7 then solves for the remaining jump conditions via the method of [118]. That is, we first solve for a matrix-valued function $\mathbf{\Delta}(z;\xi,t) = \mathbf{\Delta}(z)$ with jump conditions pictured in Figure 8.8.

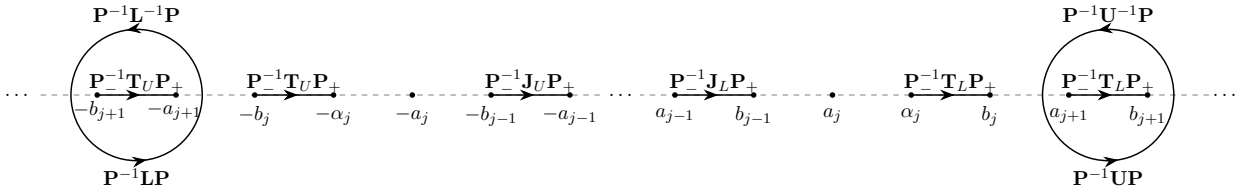


Figure 8.8: Jump contours and matrices associated with $\mathbf{\Delta}(z;\xi,t)$.

Once $\mathbf{\Delta}$ is obtained, we define $\widetilde{\mathbf{S}}(z;\xi,t) = \widetilde{\mathbf{S}}(z)$ by

$$\widetilde{\mathbf{S}} := \begin{cases} \mathbf{S}, & \text{inside the circles centered at } \pm\alpha_j, \\ \mathbf{S}\mathbf{\Delta}^{-1}, & \text{otherwise.} \end{cases} \tag{8.1}$$

Then, $\tilde{\mathbf{S}}$ satisfies the jump conditions given in Figure 8.9. All nonlocal jump conditions tend to the identity as $t \to \infty$ for fixed ξ, so we solve for $\tilde{\mathbf{S}}$ with uniform accuracy via the method of [118, 159].

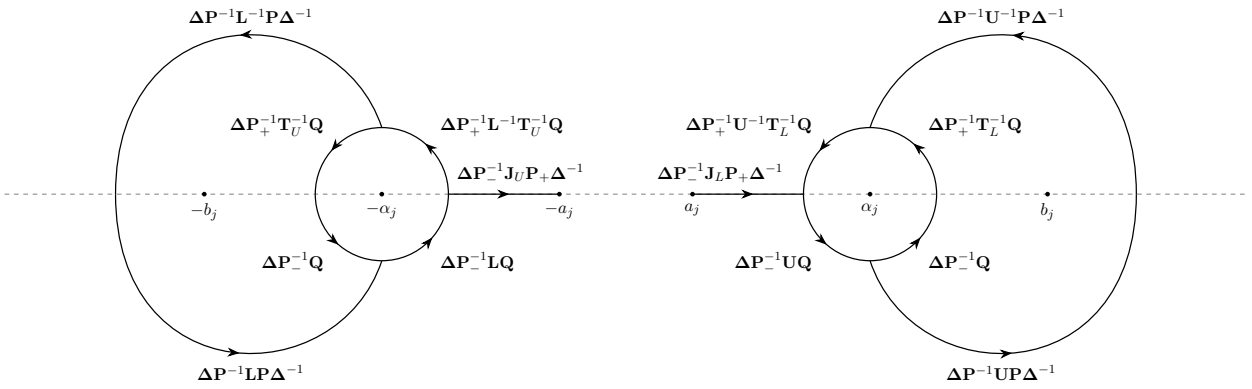


Figure 8.9: Jump contours and matrices associated with $\widetilde{\mathbf{S}}(z;\xi,t)$.

8.3.2 Endpoint asymptotics and series expansions

The limiting RHP for $\boldsymbol{\Delta}$

For $k \neq j$, the jump conditions on $\Sigma_{\pm k}$ and D_k are incorporated into the numerical method in the manner described in Chapter 7. At $\pm\alpha_j$ the solution $\boldsymbol{\Delta}$ exhibits quarter root and inverse quarter root behavior, while at $\pm b_j$ it has the same square root behavior as in Chapter 7.

This mixed endpoint behavior must be handled more delicately than the pure square root behavior found in Chapters 4 and 7. For instance, if the $(1,1)$ entry of the solution $\mathbf{U}$ of the singular integral equation on (α_j, b_j) is of the form

$$\mathbf{U}_{11}(z) = a_1(z)\frac{\sqrt[4]{1-z}}{\sqrt{z+1}} + a_2(z)\frac{1}{\sqrt[4]{1-z}\sqrt{z+1}},$$

for some smooth functions a_1, a_2, approximating these functions with polynomials is now a problem of frame approximation rather than basis approximation (see [3, Example 3]). That is, we consider the frame

$$\left\{ P_n^{\left(\frac{1}{4},-\frac{1}{2}\right)} w^{\left(\frac{1}{4},-\frac{1}{2}\right)} \right\}_{n\in\mathbb{N}} \cup \left\{ P_n^{\left(-\frac{1}{4},-\frac{1}{2}\right)} w^{\left(-\frac{1}{4},-\frac{1}{2}\right)} \right\}_{n\in\mathbb{N}}.$$

Because $w^{\left(\frac{1}{4},-\frac{1}{2}\right)}$ admits an expansion

$$w^{\left(\frac{1}{4},-\frac{1}{2}\right)} = \sum_{n=0}^{\infty} c_n P_n^{\left(-\frac{1}{4},-\frac{1}{2}\right)} w^{\left(-\frac{1}{4},-\frac{1}{2}\right)},$$

for some coefficients c_n, this frame is linearly dependent. Indeed, if one naively approximates

$$\mathbf{U}_{11} \approx \sum_{n=0}^{N/2} c_n P_n^{\left(-\frac{1}{4},-\frac{1}{2}\right)} w^{\left(-\frac{1}{4},-\frac{1}{2}\right)} + \sum_{n=0}^{N/2} \hat{c}_n P_n^{\left(\frac{1}{4},-\frac{1}{2}\right)} w^{\left(\frac{1}{4},-\frac{1}{2}\right)}, \tag{8.2}$$

for unknown coefficients c_n and $\hat{c}_n$, the collocation matrix resulting from discretizing the singular integral equation becomes increasingly ill-conditioned as $N \to \infty$.

To amend this conditioning issue, we follow the prescriptions of [3, 4]. That is, we make the following three modifications to the construction and solution of the collocation system:

- Oversample collocation points: The singular integral equation is enforced at more collocation points. That is, if we include N frame elements in each component of the approximate solution $\hat{\mathbf{U}} \approx \mathbf{U}$ and let $x_1, \ldots, x_M$ be the zeroes of T_M, the Mth Chebyshev polynomial of the first kind, we enforce

$$\mathcal{C}_\Gamma^+ \hat{\mathbf{U}}(x_j) - \mathcal{C}_\Gamma^- \hat{\mathbf{U}}(x_j)\mathbf{J}(x_j) \approx \mathbf{J}(x_j) - \mathbf{I}, \quad j = 1, \ldots, M, \tag{8.3}$$

  for some $M > N$. The approximation in (8.3) is made precise as an overdetermined least squares problem of the form

$$\mathbf{c} = \min_{\hat{\mathbf{c}}} \|\mathbf{A}\hat{\mathbf{c}} - \mathbf{b}\|_2. \tag{8.4}$$

  To obtain an accurate method, M/N must be sufficiently large. Our implementation defaults to $M = 3N$.

- Truncated SVD: The least squares problem (8.4) is solved via a truncated SVD which can be implemented as in Algorithm 12. Our implementation defaults to $\epsilon = 10\epsilon_{\text{mach}}$ where ϵ_{mach} denotes machine precision.

Algorithm 12: Truncated SVD to solve $\mathbf{c} = \min_{\hat{\mathbf{c}}} \|\mathbf{A}\hat{\mathbf{c}} - \mathbf{b}\|_2$.

Input: Matrices $\mathbf{A} \in \mathbb{C}^{m\times n}$, $\mathbf{b} \in \mathbb{C}^{m\times \ell}$ $(m \geq n)$ and tolerance ϵ.
Run SVD on $\mathbf{A}$ to get $\mathbf{U\Sigma V} = \mathbf{A}$.
Truncate $\mathbf{\Sigma}$ so that all discarded singular values $\varsigma_1 \geq \varsigma_2 \geq \ldots \geq \varsigma_n$ satisfy $\varsigma_j^2/\varsigma_1^2 < \epsilon$, leaving $\widetilde{\mathbf{\Sigma}} \in \mathbb{R}^{k\times k}$.
Truncate rows or columns of $\mathbf{U}$ and $\mathbf{V}$ as appropriate, leaving $\widetilde{\mathbf{U}} \in \mathbb{C}^{m\times k}$ and $\widetilde{\mathbf{V}} \in \mathbb{C}^{k\times n}$.
Return $\mathbf{c} = \widetilde{\mathbf{V}}^* \widetilde{\mathbf{\Sigma}}^{-1} \widetilde{\mathbf{U}}^* \mathbf{b}$.

- Basis enrichment: The frame approximation is constructed by using considerably more

terms from one basis than the other. For instance, (8.2) is modified to

$$\mathbf{U}_{11} \approx \sum_{n=0}^{N_1} c_n P_n^{\left(-\frac{1}{4},-\frac{1}{2}\right)} w^{\left(-\frac{1}{4},-\frac{1}{2}\right)} + \sum_{n=0}^{N_2} \hat{c}_n P_n^{\left(\frac{1}{4},-\frac{1}{2}\right)} w^{\left(\frac{1}{4},-\frac{1}{2}\right)}, \tag{8.5}$$

where $N_1 + N_2 = N$ and either $N_1 \gg N_2$ or $N_2 \gg N_1$. Our implementation defaults to $N_1 = 25$ and $N_2 = 5$.

Remark 8.3.1. *The collocation matrix resulting from these modifications remains ill-conditioned as $N_1 \to \infty$. Despite this, it is shown in [3, 4] that such frame approximations are highly accurate.*

The local RHP for $\widetilde{\mathbf{S}}$

Once a numerical representation of $\mathbf{\Delta}$ is obtained, $\widetilde{\mathbf{S}}$ can then be treated via the numerical method of [118, 159] using the machinery available in the `OperatorApproximation.jl` software package [155]. However, pursuant to [159, Remark 5.6.1], we construct the collocation system using Legendre polynomials $(P_n)_{n=0}^{\infty}$ rather than Chebyshev. Because Legendre polynomials have constant weight function $w(x) = 1$, their Cauchy integrals are easily computed via the method of Section 2.2.4, bypassing the need for special function evaluations. To evaluate near the contour of interest, this requires the initialization

$$\begin{aligned} C_0(z) = \mathcal{C}_{[-1,1]}[P_0 w](z) &= \frac{1}{2\pi \mathrm{i}} \int_{-1}^{1} \frac{\mathrm{d}x}{x - z} \\ &= \frac{\log(z-1) - \log(z+1)}{2\pi\mathrm{i}} = \frac{\log(1-z) - \log(-1-z)}{2\pi\mathrm{i}}. \end{aligned}$$

To obtain uniform accuracy at the intersection points of jump contours, one should construct the collocation system using the finite-part Cauchy integral [159, Definition 5.22]. This differs from the classical Cauchy integral only at the endpoints $z = \pm 1$. At these endpoints, the finite-part Cauchy integrals are generated through the forward substitution procedure using the initializations

$$\begin{aligned} C_0(-1 + 0\mathrm{e}^{\mathrm{i}\theta}) = \mathrm{FP}\mathcal{C}_{[-1,1]}[P_0 w](-1 + 0\mathrm{e}^{\mathrm{i}\theta}) &= \frac{\mathrm{i}\theta - \mathrm{i}\arg\left(-\mathrm{e}^{-\mathrm{i}\theta}\right)}{2\pi\mathrm{i}}, \\ C_0(1 + 0\mathrm{e}^{\mathrm{i}\theta}) = \mathrm{FP}\mathcal{C}_{[-1,1]}[P_0 w](1 + 0\mathrm{e}^{\mathrm{i}\theta}) &= \frac{\mathrm{i}\theta - \log(2)}{2\pi\mathrm{i}}, \end{aligned}$$

where θ denotes the angle of approach.

The use of Legendre polynomials encodes the correct asymptotic behavior of $\tilde{\mathbf{S}}(z)$ at all

contour endpoints except $z = \pm a_j$. As $z \to a_j$,

$$\mathbf{S}(z) = \begin{cases} \begin{pmatrix} \mathrm{O}(1) & \mathrm{O}(1) \end{pmatrix}, & \text{if} \quad \mathfrak{a}_j = \frac{1}{2}, \\ \begin{pmatrix} \mathrm{O}(|z-a_j|^{-\frac{1}{2}}) & \mathrm{O}(1) \end{pmatrix}, & \text{if} \quad \mathfrak{a}_j = -\frac{1}{2}, \end{cases}$$

and as $z \to -a_j$,

$$\mathbf{S}(z) = \begin{cases} \begin{pmatrix} \mathrm{O}(1) & \mathrm{O}(1) \end{pmatrix}, & \text{if} \quad \mathfrak{a}_j = \frac{1}{2}, \\ \begin{pmatrix} \mathrm{O}(1) & \mathrm{O}(|z+a_j|^{-\frac{1}{2}}) \end{pmatrix}, & \text{if} \quad \mathfrak{a}_j = -\frac{1}{2}, \end{cases}$$

from which the behavior of $\widetilde{\mathbf{S}}$ can be obtained since $\mathbf{\Delta}(z)$ is analytic in a neighborhood of $z = \pm a_j$. This behavior can be encoded with an appropriate basis or frame of shifted Jacobi polynomials and weights with parameter 0 at one endpoint and $\pm\frac{1}{2}$ at the other.

Alternatively, this asymptotic behavior can be removed by applying a local solution in a neighborhood of $z = \pm a_j$. Then, all asymptotic conditions are encoded with the use of shifted and scaled Legendre polynomials as basis functions. It is easiest to construct a local matrix solution to $\mathbf{S}$ near each endpoint before applying the transformation (8.1). In a neighborhood of $z = a_j$, we seek a function satisfying

$$\mathbf{\Psi}_R^+(z) = \mathbf{\Psi}_R^-(z) \begin{pmatrix} 1 & 0 \\ -\mathrm{i}r(z)\mathrm{e}^{\phi_+(z)+\phi_-(z)} & 1 \end{pmatrix}, \quad z \in (a_j, c),$$

where $a_j < c < \alpha_j$ and c is outside the neighborhood of a_j where the local solution will be applied. By the Plemelj lemma, one such function is

$$\mathbf{\Psi}_R(z) = \begin{pmatrix} 1 & 0 \\ -\frac{1}{2\pi}\int_{a_j}^{c} \frac{r(s)\mathrm{e}^{\phi_+(s)+\phi_-(s)}\mathrm{d}s}{s-z} & 1 \end{pmatrix}.$$

To evaluate $\mathbf{\Psi}_R(z)$ numerically, we must compute

$$\begin{aligned} \phi_{21}(z) = -\frac{1}{2\pi}\int_{a_j}^{c} \frac{r(s)\mathrm{e}^{\phi_+(s)+\phi_-(s)}\mathrm{d}s}{s-z} \\ = -\mathrm{i}\mathcal{C}_{[a_j,c]}\left[f_j \mathrm{e}^{\phi_+ + \phi_-}(\diamond - a_j)^{\mathfrak{a}_j}(b_j - \diamond)^{\mathfrak{b}_j}\right](z), \quad z \notin [a_j, c]. \end{aligned}$$

Using the shifted Jacobi notation of (2.18) with the interval $[a_j, c]$, we approximate

$$f_j(x)\mathrm{e}^{\phi_+(x)+\phi_-(x)}(b_j - x)^{\mathfrak{b}_j} = \sum_{n=0}^{\infty} c_n \tilde{P}_n^{(0,\mathfrak{a}_j)}(x) \approx \sum_{n=0}^{L} c_n \tilde{P}_n^{(0,\mathfrak{a}_j)}(x), \quad x \in [a_j, c],$$

to high precision. Then, ϕ_{21} is approximated by

$$\phi_{21}(z) \approx -\sum_{n=0}^{L} \mathrm{i}c_n \mathcal{C}_{[a_j,c]} \left[\tilde{P}_n^{(0,\mathfrak{a}_j)} \tilde{w}^{(0,\mathfrak{a}_j)} \right](z),$$

where the Cauchy integrals can be computed via the method of Section 2.2.4 of Chapter 2, enabling the evaluation of $\phi_{21}(z)$ for a given z. A local solution in a neighborhood of $z = -a_j$ is given by

$$\mathbf{\Psi}_L(z) = \begin{pmatrix} 1 & \frac{1}{2\pi} \int_{-c}^{-a_j} \frac{r(s)\mathrm{e}^{-\phi_+(s)-\phi_-(s)}\mathrm{d}s}{s-z} \\ 0 & 1 \end{pmatrix},$$

which is evaluated in the same manner, mutatis mutandis.

Given local solutions $\mathbf{\Psi}_L, \mathbf{\Psi}_R$, we define $\mathbf{T}(z;\xi,t) = \mathbf{T}(z)$ by

$$\mathbf{T} := \begin{cases} \mathbf{S}\mathbf{\Psi}_L^{-1}, & \text{inside a neighborhood of } -a_j, \\ \mathbf{S}\mathbf{\Psi}_R^{-1}, & \text{inside a neighborhood of } a_j, \\ \mathbf{S}, & \text{otherwise,} \end{cases}$$

and define $\widetilde{\mathbf{T}}(z;\xi,t) = \widetilde{\mathbf{T}}(z)$ by

$$\widetilde{\mathbf{T}} := \begin{cases} \mathbf{T}, & \text{inside the circles centered at } \pm\alpha_j, \\ \mathbf{T}\mathbf{\Delta}^{-1}, & \text{otherwise.} \end{cases} \tag{8.6}$$

Then, $\tilde{\mathbf{T}}$ satisfies the jump conditions given in Figure 8.10. Then, $\tilde{\mathbf{T}}$ can be solved for

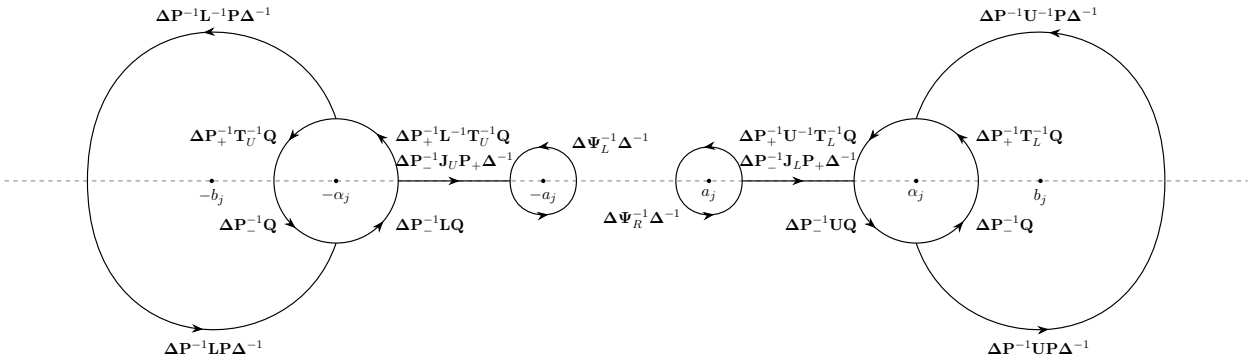


Figure 8.10: Jump contours and matrices associated with $\widetilde{\mathbf{T}}(z;\xi,t)$.

numerically using bases of Legendre polynomials as described above. The soliton gas solution

$u(x,t)$ of (1.4) is then recovered from $\mathbf{T}(z) = \mathbf{T}(z;x,t)$ via the formula

$$u(x,t) = \lim_{z\to\infty} -2z\mathbf{T}_{12}(z;x,t).$$

8.3.3 Soliton addition

Multi-solitons are inserted into soliton gasses via the method of Section 7.3 of Chapter 7. Additional solitons can be inserted into an mKdV soliton gas by supplementing Riemann–Hilbert Problem 8.1.1 with appropriate residue conditions. That is, to insert n solitons with spectral parameters $\kappa_1,\dots,\kappa_n \in \mathbb{R}_+ \setminus \{0\}$ and norming constants $\chi_1,\dots,\chi_n \in \mathbb{R}\setminus\{0\}$, we consider solving the following RHP:

Riemann–Hilbert Problem 8.3.2 (mKdV soliton gas with solitons)**.** *Let $(x,t) \in \mathbb{R}\times\mathbb{R}_+$ be fixed. Find a 2×2 matrix-valued function $\mathbf{N}(\lambda) \equiv \mathbf{N}(\lambda;x,t)$ with the following properties:*

Analyticity: *$\mathbf{N}(\lambda)$ is analytic for $\lambda \in \mathbb{C}\setminus(\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-)$, admitting continuous boundary values on the interior of $\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-$, with the exception of simple poles at $\lambda = \pm\mathrm{i}\kappa_j$, $\kappa_j > 0$, $j = 1,2,\dots,n$, in the complement of $\mathrm{i}\Sigma_+ \cup \mathrm{i}\Sigma_-$.*

Endpoint behavior: *The same as in Riemann–Hilbert Problem 8.1.1.*

Jump conditions: *The boundary values $\mathbf{N}^+(\lambda)$ (resp. $\mathbf{N}^-(\lambda)$) from left (resp. from right) are related via*

$$\mathbf{N}^+(\lambda) = \mathbf{N}^-(\lambda)\begin{pmatrix} 1 & 0 \\ 2\mathrm{i}r_1(\lambda)\mathrm{e}^{2\mathrm{i}(x\lambda+4t\lambda^3)} & 1\end{pmatrix}, \quad \lambda\in\mathrm{i}\Sigma_+,$$
$$\mathbf{N}^+(\lambda) = \mathbf{N}^-(\lambda)\begin{pmatrix} 1 & 2\mathrm{i}r_1(\lambda)\mathrm{e}^{-2\mathrm{i}(x\lambda+4t\lambda^3)} \\ 0 & 1\end{pmatrix}, \quad \lambda\in\mathrm{i}\Sigma_-, \quad j=1,2,\dots,g.$$

Residue conditions: *At each simple pole $\lambda = \pm\mathrm{i}\kappa_j$, $j = 1,2,\dots,n$, $\mathbf{N}(\lambda)$ satisfies*

$$\operatorname*{Res}_{\lambda=\mathrm{i}\kappa_j} \mathbf{N}(\lambda) = \lim_{\lambda\to\mathrm{i}\kappa_j} \mathbf{N}(\lambda)\begin{pmatrix} 0 & 0 \\ -\mathrm{i}\chi_j\mathrm{e}^{2\mathrm{i}(\mathrm{i}\kappa_j x - 4\mathrm{i}\kappa_j^3 t)} & 0\end{pmatrix},$$
$$\operatorname*{Res}_{\lambda=-\mathrm{i}\kappa_j} \mathbf{N}(\lambda) = \lim_{\lambda\to-\mathrm{i}\kappa_j} \mathbf{N}(\lambda)\begin{pmatrix} 0 & -\mathrm{i}\chi_j\mathrm{e}^{2\mathrm{i}(\mathrm{i}\kappa_j x - 4\mathrm{i}\kappa_j^3 t)} \\ 0 & 0\end{pmatrix},$$

with norming constants $\chi_j \in \mathbb{R}\setminus\{0\}$.

Normalization: *$\mathbf{N}(\lambda) = \mathbf{I} + \mathrm{O}(\lambda^{-1})$ as $\lambda\to\infty$.*

The residue conditions are included in the second stage of our numerical method (solving for $\widetilde{\mathbf{T}}$), with deformations applied as described in Section 7.3 of Chapter 7. In particular, we apply the pole-flipping transformation

$$\hat{\mathbf{Y}}(z) = \mathbf{Y}(z)v(z)^{\sigma_3}, \quad v(z) = \prod_{j\in K} \frac{z-\kappa_j}{z+\kappa_j}, \tag{8.7}$$

where for some constant $c > 0$, K is an index set corresponding to "large" residue conditions:

$$K = \left\{ j : \left| \chi_j \mathrm{e}^{\phi_+(\kappa_j)+\phi_-(\kappa_j)} \right| > c \right\}. \tag{8.8}$$

Our implementation takes $c = 10^5$ as the default value.

Remark 8.3.3. *Applying the pole-flipping transformation* (8.7) *may introduce conditioning issues into the collocation system. When applying the steepest descent deformations to* $\hat{\mathbf{Y}}$, $\frac{1}{v^2}$ *appears in the jump conditions, and this function can become large on the jump contours*[1]*. This is partially mitigated by choosing a large value for* c *in* (8.8) *but comes at the expense of conditioning issues introduced by large residue conditions.*

Empirically, we find that when the collocation system $\mathbf{Ac} = \mathbf{b}$ *is ill-conditioned, row- and column-scaled left and right diagonal preconditioners improve the accuracy of the computed solution. That is, we define a diagonal left preconditioner by*

$$(\mathbf{D}_L)_{jj} = \max_k |\mathbf{A}_{jk}|,$$

and a diagonal right preconditioner by

$$(\mathbf{D}_R)_{jj} = \max_k \left| \left(\mathbf{D}_L^{-1}\mathbf{A}\right)_{kj} \right|,$$

and solve the equivalent system

$$\mathbf{D}_L^{-1}\mathbf{A}\mathbf{D}_R^{-1}\mathbf{y} = \mathbf{D}_L^{-1}\mathbf{b}.$$

8.4 Modifications for KdV

Now, consider the Korteweg–de Vries (KdV) equation (1.3) Define

$$\mathbf{J}_L(z;\xi,t) = \begin{pmatrix} 1 & 0 \\ -\mathrm{i}r(z)\mathrm{e}^{-8t\theta(z;\xi)} & 1 \end{pmatrix}, \quad \mathbf{J}_U(z;\xi,t) = \begin{pmatrix} 1 & \mathrm{i}r(z)\mathrm{e}^{8t\theta(z;\xi)} \\ 0 & 1 \end{pmatrix}.$$

[1]For instance, this can be on the order of 10^7 in Example 8.5.2.

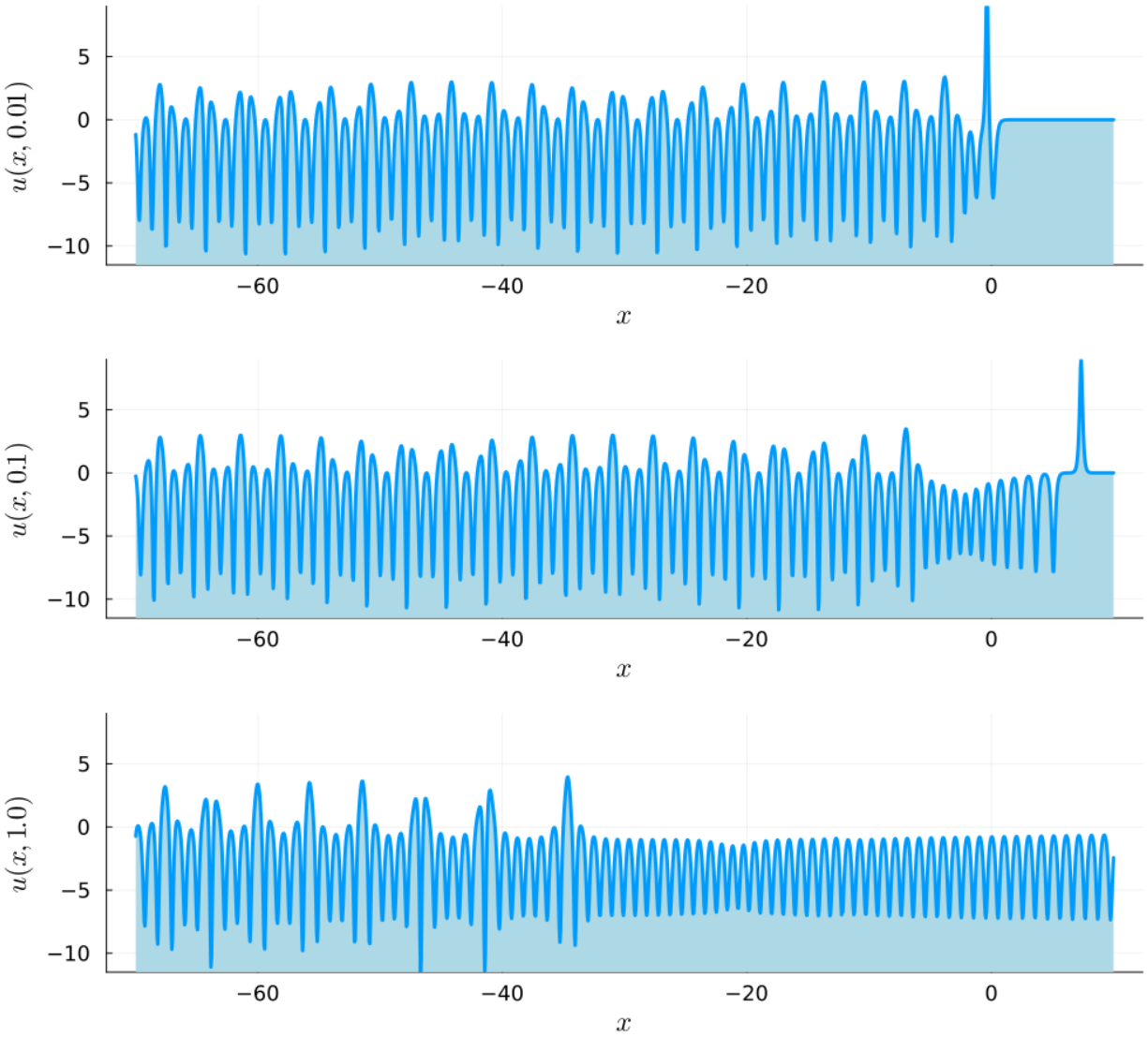


Figure 8.11: mKdV hybrid soliton and anti-soliton gas with $r_1(\lambda)$ supported on two pairs of bands with opposite signs on each, nonlinearly superimposed with a soliton and an anti-soliton.

To solve Riemann–Hilbert Problem 7.1.1 in the rotated $z = -\mathrm{i}\lambda$ plane, we seek a vector-valued function $\mathbf{Y}(z;\xi,t) = \mathbf{Y}(z)$ satisfying

$$\begin{aligned}
\mathbf{Y}^+(z) &= \mathbf{Y}^-(z)\mathbf{J}_L(z;\xi,t), \qquad z \in \Sigma_+, \qquad \theta(z;\xi) := \xi z - z^3,\\
\mathbf{Y}^+(z) &= \mathbf{Y}^-(z)\mathbf{J}_U(z;\xi,t), \qquad z \in \Sigma_-,\\
\mathbf{Y}(z) &= \begin{pmatrix} 1 & 1 \end{pmatrix} + \mathrm{O}\left(z^{-1}\right), \quad z \to \infty,
\end{aligned}$$

that is analytic elsewhere. We do not explicitly enforce the symmetry condition $\mathbf{Y}(-z) = \mathbf{Y}(z)\sigma_1$ in our numerical method. We similarly define

$$\mathbf{T}_L(z;\xi,t) = \begin{pmatrix} 0 & -\frac{\mathrm{i}}{r(z)}\mathrm{e}^{8t\theta(z;\xi)} \\ -\mathrm{i}r(z)\mathrm{e}^{-8t\theta(z;\xi)} & 0 \end{pmatrix}, \quad \mathbf{L}(z;\xi,t) = \begin{pmatrix} 1 & 0 \\ -\frac{\mathrm{i}}{r(z)}\mathrm{e}^{-8t\theta(z;\xi)} & 1 \end{pmatrix},$$
$$\mathbf{T}_U(z;\xi,t) = \begin{pmatrix} 0 & \mathrm{i}r(z)\mathrm{e}^{8t\theta(z;\xi)} \\ \frac{\mathrm{i}}{r(z)}\mathrm{e}^{-8t\theta(z;\xi)} & 0 \end{pmatrix}, \qquad \mathbf{U}(z;\xi,t) = \begin{pmatrix} 1 & \frac{\mathrm{i}}{r(z)}\mathrm{e}^{8t\theta(z;\xi)} \\ 0 & 1 \end{pmatrix}.$$

With these definitions, we perform the same sequence of steepest descent deformations as before. Then, $\mathbf{T}$ is 1×2 row vector-valued and satisfies the jump conditions given in Figure 8.7. On the appropriate contours,

$$\mathbf{P}_-^{-1}\mathbf{T}_L\mathbf{P}_+ = \begin{pmatrix} 0 & -\frac{\mathrm{i}}{\mathrm{e}^{A_k(\xi,t)}r} \\ -\mathrm{i}\mathrm{e}^{A_k(\xi,t)}r & 0 \end{pmatrix}, \qquad \mathbf{P}^{-1}\mathbf{U}\mathbf{P} = \begin{pmatrix} 1 & \frac{\mathrm{i}}{r}\mathrm{e}^{-2\phi} \\ 0 & 1 \end{pmatrix},$$
$$\mathbf{P}_-^{-1}\mathbf{J}_L\mathbf{P}_+ = \begin{pmatrix} 1 & 0 \\ -\mathrm{i}r\mathrm{e}^{\phi_+ + \phi_-} & 1 \end{pmatrix}, \qquad \mathbf{P}_+^{-1}\mathbf{T}_L^{-1}\mathbf{Q} = \begin{pmatrix} 0 & \frac{\mathrm{i}}{r}\mathrm{e}^{-\phi_+} \\ \mathrm{i}r\mathrm{e}^{\phi_+} & 0 \end{pmatrix},$$
$$\mathbf{P}_+^{-1}\mathbf{U}^{-1}\mathbf{T}_L^{-1}\mathbf{Q} = \begin{pmatrix} \mathrm{e}^{-\phi_+} & \frac{\mathrm{i}}{r}\mathrm{e}^{-\phi_+} \\ \mathrm{i}r\mathrm{e}^{\phi_+} & 0 \end{pmatrix}, \qquad \mathbf{P}_-^{-1}\mathbf{U}\mathbf{Q} = \begin{pmatrix} \mathrm{e}^{-\phi_-} & \frac{\mathrm{i}}{r}\mathrm{e}^{-\phi_-} \\ 0 & \mathrm{e}^{\phi_-} \end{pmatrix}.$$

To solve this problem numerically, we again solve for a 2×2 matrix-valued function $\mathbf{\Delta}(z;\xi,t) = \mathbf{\Delta}(z)$ with jump conditions pictured in Figure 8.8. However, in contrast with the mKdV problem, such a $\mathbf{\Delta}$ fails to exist at a set of isolated points (x,t). While prior numerical work has largely ignored this behavior [34, 160], we observe that numerical errors resulting from it compound within our two-stage procedure, leading to notable accuracy loss near these points. One remedy is to directly solve for $\mathbf{T}$ rather than using a two-stage procedure. While this approach avoids compounding errors, the nonexistence of a bounded matrix solution still impacts numerical accuracy near the relevant points. Empirically, a one-stage method is also less robust at handling long-time evolution. Instead, we implement a modified two-stage solution procedure.

Rather than attempt to construct a bounded solution that may not exist, we introduce a simple pole at the origin to the construction of $\mathbf{\Delta}$. In particular, we require $\mathbf{\Delta}$ to additionally

satisfy the residue condition

$$\operatorname*{Res}_{z=0} \mathbf{\Delta}(z) = \lim_{z\to 0} \mathbf{\Delta}(z) \begin{pmatrix} \mathrm{i} & \mathrm{i} \\ -\mathrm{i} & -\mathrm{i} \end{pmatrix}. \tag{8.9}$$

At generic points[2] (x,t), this construction ensures that $\mathbf{\Delta}$ has determinant 1, satisfies the appropriate symmetry $\mathbf{\Delta}(-z) = \sigma_1 \mathbf{\Delta}(z)\sigma_1$, and has a pole at the origin of the form

$$\mathbf{\Delta}(z) = \frac{a_{-1}}{z} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} + \mathrm{O}(1), \quad a_{-1} \in \mathbb{C},$$

as $z \to 0$.

Once a numerical representation of $\mathbf{\Delta}$ is obtained, $\widetilde{\mathbf{T}}$ as defined in (8.6) again satisfies the jump conditions given in Figure 8.10, with the exception of a possible pole at the origin. Denote as $z \to 0$,

$$\begin{aligned} \widetilde{\mathbf{T}}(z) &= \widetilde{\mathbf{T}}_{-1} z^{-1} + \widetilde{\mathbf{T}}_0 + \mathrm{O}(z), \\ \mathbf{\Delta}(z) &= \mathbf{\Delta}_{-1} z^{-1} + \mathbf{\Delta}_0 + \mathrm{O}(z). \end{aligned}$$

In order for $\mathbf{T}$ to be regular at the origin, it must hold that

$$\widetilde{\mathbf{T}}_{-1}\mathbf{\Delta}_0 + \widetilde{\mathbf{T}}_0 \mathbf{\Delta}_{-1} = 0,$$

which corresponds to the residue condition

$$\operatorname*{Res}_{z=0} \widetilde{\mathbf{T}}(z)\mathbf{\Delta}_0 = -\lim_{z\to 0} \widetilde{\mathbf{T}}(z)\mathbf{\Delta}_{-1}.$$

Because $\mathbf{T}(-z) = \mathbf{T}(z)\sigma_1$ and $\mathbf{\Delta}(-z) = \sigma_1\mathbf{\Delta}(z)\sigma_1$, $\widetilde{\mathbf{T}}$ satisfies the symmetry $\widetilde{\mathbf{T}}(-z) = \widetilde{\mathbf{T}}(z)\sigma_1$. This implies that $\widetilde{\mathbf{T}}_0$ is of the form $\tilde{t}_0 \begin{pmatrix} 1 & 1 \end{pmatrix}$ where $\tilde{t}_0 \in \mathbb{C}$, so

$$\operatorname*{Res}_{z=0} \widetilde{\mathbf{T}}(z)\mathbf{\Delta}_0 = \tilde{t}_0 \begin{pmatrix} 1 & 1 \end{pmatrix} a_{-1} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \end{pmatrix}.$$

Thus, no residue condition need be imposed, and it suffices to find some $\widetilde{\mathbf{T}}$ that is regular at the origin. This means that once a numerical representation of $\mathbf{\Delta}$ is obtained, $\widetilde{\mathbf{T}}$ is computed in the same manner as for the mKdV equation, modulo computing a vector, rather than matrix, solution and the necessary sign changes in the jump conditions. The soliton gas solution $u(x,t)$ of (1.3) is then recovered from $\mathbf{T}(z) = \mathbf{T}(z;x,t)$ via the formula

$$u(x,t) = \lim_{z\to\infty} 2z^2(\mathbf{T}_1(z;x,t)\mathbf{T}_2(z;x,t) - 1).$$

[2]By this, we mean points (x,t) where a bounded matrix solution does exist. At nongeneric points, these properties can be validated empirically.

8.4.1 Soliton addition

Multi-solitons are again inserted into KdV soliton gasses by supplementing Riemann–Hilbert Problem 7.1.1 with appropriate residue conditions, exactly as in Riemann–Hilbert Problem 7.3.1 of Chapter 7. The residue conditions are included in the second stage of our numerical method (solving for $\widetilde{\mathbf{T}}$), with pole-flipping transformations and deformations applied in the same manner as for mKdV.

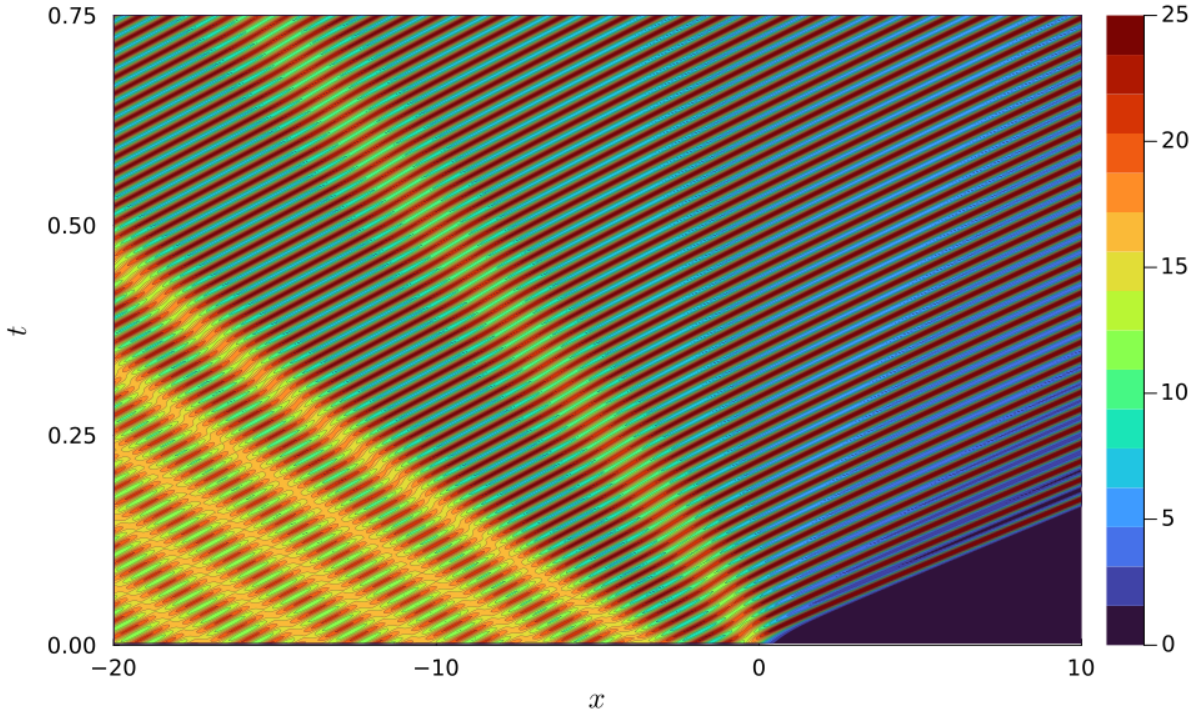


Figure 8.12: Density plot of a KdV soliton gas with $r_1(\lambda)$ supported on two pairs of bands with an additional soliton in between.

8.5 Examples and demonstration of convergence

8.5.1 Convergence

In Figure 8.13, we demonstrate the convergence of our numerical method for computing $\mathbf{\Delta}$ in the context of the KdV equation, near a point (x, t) where a bounded matrix solution appears to not exist. We plot the absolute pointwise error made in computing $\mathbf{\Delta}$ on a small

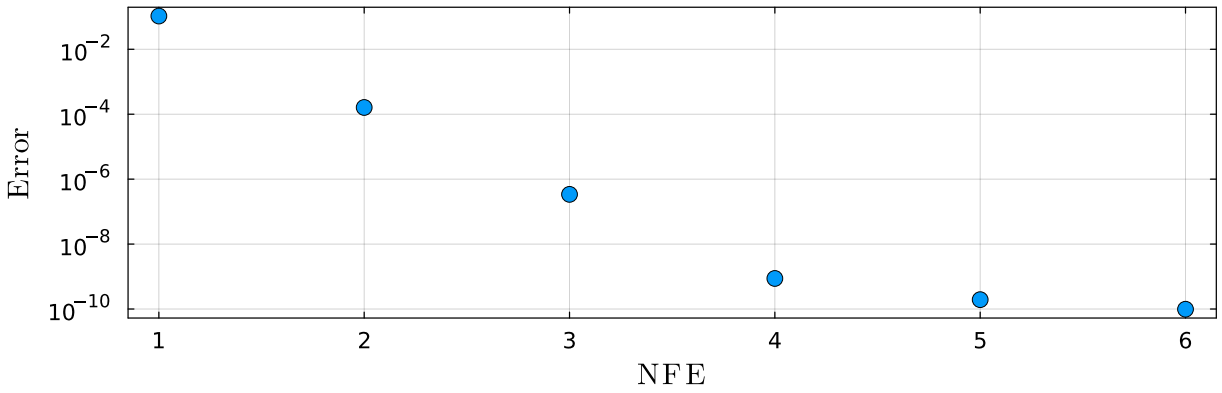


Figure 8.13: Maximum pointwise errors in the computation of $\mathbf{\Delta}(z)$ on the circle $|z| = 0.01$ for a varying number of frame elements (NFE). In the notation of (8.5), $N_2 = \text{NFE}$ and $N_1 = 4\text{NFE}$. 16 NFE is used as the exact solution in comparisons.

circle around the origin:

$$\max_{\theta\in[-\pi,\pi]} \left\| \mathbf{\Delta}_p(0.01\mathrm{e}^{\mathrm{i}\theta}) - \mathbf{\Delta}_{\text{true}}(0.01\mathrm{e}^{\mathrm{i}\theta}) \right\|_2 ,$$

where p denotes a measure of the number of frame elements used, denoted by "NFE" in Figure 8.13, and $\mathbf{\Delta}_{\text{true}}$ denotes the solution computed with a large value of p. Here, $r_1(\lambda)$ is supported on one pair of bands (in the notation of (7.4)) $I_1 = (1.5, 2.5)$ with $f_1(z) = \frac{1}{2}$ and $\mathfrak{a}_1 = \mathfrak{b}_1 = \frac{1}{2}$, and we consider the point $x = -0.4376211$, $t = 0.1$. We observe that the addition of the residue condition (8.9) appears to yield a convergent numerical method for computing $\mathbf{\Delta}$.

8.5.2 Examples

In this subsection, we enumerate the examples found throughout this chapter. All parameters are given in the notation of (7.4).

5 pairs of bands with 5 solitons

In Figure 8.1, we plot a KdV soliton gas corresponding to $r_1(\lambda)$ supported on five pairs of bands, nonlinearly superimposed with five solitons. Here, $I_1 = (0.25, 0.5)$, $I_2 = (0.8, 1.2)$, $I_3 = (1.5, 2)$, $I_4 = (2.5, 3)$, $I_5 = (4, 5)$, with $f_1(z) = 1$, $f_2(z) = 1/2$, $f_3(z) = 1/4$, $f_4(z) = 1/8$, $f_5(z) = 1/16$ and $\mathfrak{a}_k = \mathfrak{b}_k = \frac{1}{2}$ for $k = 1, \ldots, 5$. The solitons are associated with (see Riemann–Hilbert Problem 7.3.1) $\kappa_1 = 0.1$, $\kappa_2 = 0.7$, $\kappa_3 = 2.25$, $\kappa_4 = 3.5$, $\kappa_5 = 5.5$ with the norming constants $\chi_1 = 10^5$, $\chi_2 = 1000$, $\chi_3 = 100$, $\chi_4 = 10$, and $\chi_5 = 10^{-6}$.

1 pair of bands with 2 solitons

In Figure 8.2, we consider a KdV soliton gas corresponding to $r_1(\lambda)$ supported on one pair of bands, nonlinearly superimposed with two solitons. Here, $I_1 = (1.5, 2.5)$, $f_1(z) = 1$, and $\mathfrak{a}_1 = \mathfrak{b}_1 = \frac{1}{2}$. The solitons are associated with the eigenvalue parameters $\kappa_1 = 1$, $\kappa_2 = 4$ and the norming constants $\chi_1 = 10$, $\chi_2 = 10^{-10}$.

2 pairs of bands with 1 soliton

In Figure 8.12, we consider a KdV soliton gas corresponding to $r_1(\lambda)$ supported on two pairs of bands, nonlinearly superimposed with one soliton. Here, $I_1 = (1.5, 2.5)$, $I_2 = (3, 4)$, $f_1(z) = f_2(z) = 1$, and $\mathfrak{a}_1 = \mathfrak{b}_1 = \mathfrak{a}_2 = \mathfrak{b}_2 = \frac{1}{2}$. The soliton is associated with the eigenvalue parameter $\kappa_1 = 2.75$ and the norming constant $\chi_1 = 1$.

2 (opposite sign) pairs of bands with 1 soliton and 1 anti-soliton

In Figure 8.11, we consider an mKdV hybrid soliton and anti-soliton gas corresponding to $r_1(\lambda)$ supported on two pairs of bands, nonlinearly superimposed with a soliton and an anti-soliton. Here, $I_1 = (1.5, 2.5)$, $I_2 = (3, 4)$, $f_1(z) = 1$, $f_2(z) = -1$, and $\mathfrak{a}_1 = \mathfrak{a}_2 = \frac{1}{2}$, $\mathfrak{b}_1 = \mathfrak{b}_2 = -\frac{1}{2}$. The soliton is associated with the eigenvalue parameter $\kappa_1 = 4.5$ and the norming constant $\chi_1 = 10^{-2}$, while the anti-soliton is associated with the eigenvalue parameter $\kappa_2 = 2.75$ and the norming constant $\chi_2 = -1$.

Chapter 9

LARGE-GENUS KDV SOLUTIONS

9.1 Introduction

This chapter is based on in-progress work with Deniz Bilman and Thomas Trogdon.

The finite-genus (quasi-periodic) solutions of the KdV equation (1.3) are algebro-geometric solutions associated with a genus-g hyperelliptic Riemann surface. Classically, they are expressed through the Its–Matveev formula in terms of Riemann theta functions [91]. As discussed in Section 2.8, evaluating this representation directly is difficult for large genera. Much like with the Akhiezer polynomials, a Riemann–Hilbert representation of finite-genus solutions was introduced in [157], resulting in a numerical method for small-genus solutions (roughly $g < 10$). In [33], the authors reformulate the finite-genus RHP and develop a numerical method for computing large-genus solutions (roughly $g < 1000$).

We refer to [33] for a full treatment of the finite-genus solutions, the associated RHP, and applications. The purpose of this chapter is to observe that the finite-genus RHP is a clean instance of the framework developed in Chapter 3 and to record the resulting numerical method. Because the jump conditions are piecewise constant and unitary, we are able to improve the numerical method of [33] by explicitly constructing a preconditioner at the operator level and developing a matrix-free implementation of the collocation scheme. The resulting method is effective for even larger genus (roughly $g < 5000$).

9.2 The Riemann–Hilbert problem

As before, fix $g \in \mathbb{Z}_+$ and let $0 < a_1 < b_1 < a_2 < \cdots < a_g < b_g$. Denoting

$$\Sigma_+ := \bigcup_{j=1}^{g} [a_j, b_j] \qquad \text{and} \qquad \Sigma_- := \bigcup_{j=1}^{g} [-b_j, -a_j],$$

like the (rotated) soliton gas RHP of Chapters 7 and 8, the jump conditions for the (unfolded) finite-genus RHP of [33] are supported on $\Sigma_+ \cup \Sigma_-$. The RHP is as follows:

Riemann–Hilbert Problem 9.2.1 (Finite-genus KdV solution)**.** *Let $(x,t) \in \mathbb{R} \times \mathbb{R}$ be fixed. Find a 1×2 row-vector-valued function $\mathbf{S}(z) \equiv \mathbf{S}(z; x, t)$ with the following properties:*

> ***Analyticity:*** *$\mathbf{S}(z)$ is analytic for $z \in \mathbb{C} \setminus (\Sigma_+ \cup \Sigma_-)$, admitting continuous boundary values on the interior of $\Sigma_+ \cup \Sigma_-$.*
>
> ***Jump conditions:*** *The boundary values $\mathbf{S}^+(z)$ (resp. $\mathbf{S}^-(z)$) from the above (resp. below) are related via*
>
> $$\mathbf{S}^+(z) = \mathbf{S}^-(z) \begin{pmatrix} 0 & \mathrm{e}^{\mathrm{i}\Omega_j(x,t)} \\ \mathrm{e}^{-\mathrm{i}\Omega_j(x,t)} & 0 \end{pmatrix}, \quad z \in [a_j, b_j],$$
> $$\mathbf{S}^+(z) = \mathbf{S}^-(z) \begin{pmatrix} 0 & \mathrm{e}^{-\mathrm{i}\Omega_j(x,t)} \\ \mathrm{e}^{\mathrm{i}\Omega_j(x,t)} & 0 \end{pmatrix}, \quad z \in [-b_j, -a_j],$$

for $j = 1, \ldots, g$.

Normalization: $\mathbf{S}(z) = \begin{pmatrix} 1 & 1 \end{pmatrix} + \mathrm{O}(z^{-1})$ *as* $z \to \infty$.

Symmetry condition: $\mathbf{S}(-z) = \mathbf{S}(z) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Endpoint behavior: $\mathbf{S}(z) = \mathrm{O}\left(|z \mp a_j|^{-1/2}\right)$ *as* $z \to \pm a_j$ *and* $\mathbf{S}(z) = \mathrm{O}(1)$ *as* $z \to \pm b_j$.

Here, the phases $\Omega_j(x,t) = \Omega_j^{(0)} + \Omega_j^{(x)} x + \Omega_j^{(t)} t \in \mathbb{R}$ depend affinely on x and t [33]. Given $\mathbf{S}(z) = \begin{pmatrix} \mathbf{S}_1(z) & \mathbf{S}_2(z) \end{pmatrix}$, the KdV solution is recovered via the product formula [62]

$$u(x,t) = -\lim_{z \to \infty} 2z^2 \big(\mathbf{S}_1(z;x,t) \mathbf{S}_2(z;x,t) - 1 \big).$$

Note that this is different from the recovery formula used in [33], and has the effect of halving the number of required linear solves in the numerical method.

Denote the $2g$ bands as $I_j = [a_j, b_j]$ and $I_{-j} = [-b_j, -a_j]$. For each I_n, we denote the corresponding jump condition by

$$\mathbf{J}_n = \begin{pmatrix} 0 & \zeta_n \\ 1/\zeta_n & 0 \end{pmatrix}, \qquad |\zeta_n| = 1, \tag{9.1}$$

where $\zeta_{\pm j} = \mathrm{e}^{\pm \mathrm{i} \Omega_j(x,t)}$ for $j = 1, \ldots, g$. The jump conditions of Riemann–Hilbert Problem 9.2.1 are pictured in Figure 9.1.

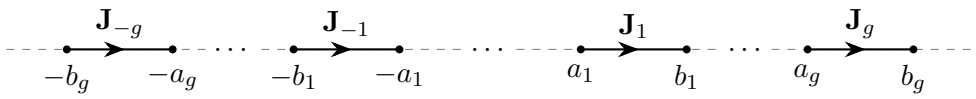


Figure 9.1: The jump contour for Riemann–Hilbert Problem 9.2.1. The jump matrix is the constant unitary matrix $\mathbf{J}_{\pm j}$ on each band, and $\mathbf{S}$ is normalized to $\begin{pmatrix} 1 & 1 \end{pmatrix}$ at infinity.

9.3 Diagonalization and preconditioning

Following Section 3.3, we make the ansatz

$$\mathbf{S}(z) = \begin{pmatrix} 1 & 1 \end{pmatrix} + \mathcal{C}_{\Sigma_+ \cup \Sigma_-}\mathbf{u}(z), \qquad \mathbf{u} = \sum_{1 \le |n| \le g} \mathbf{u}_n, \quad \mathbf{u}_n = P_n \mathbf{u}, \tag{9.2}$$

where $\mathbf{u}$ is a 1×2 row-vector density on $\Sigma_+ \cup \Sigma_-$ and P_n is the projection onto I_n. On each band I_n, this yields the singular integral equation

$$\mathcal{C}^+_{I_n}\mathbf{u}_n - \mathcal{C}^-_{I_n}\mathbf{u}_n\mathbf{J}_n + \sum_{k \ne n} \mathcal{C}_{I_k}\mathbf{u}_k(\mathbf{I} - \mathbf{J}_n) = \begin{pmatrix} 1 & 1 \end{pmatrix}(\mathbf{J}_n - \mathbf{I}). \tag{9.3}$$

Because each jump condition (9.1) is Hermitian and unitarily diagonalizable with constant eigenvectors,

$$\mathbf{J}_n = \mathbf{U}(\zeta_n)\,\mathbf{D}\,\mathbf{U}(\zeta_n)^*, \qquad \mathbf{D} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad \mathbf{U}(\zeta) = \frac{1}{\sqrt{2}}\begin{pmatrix} \zeta & -\zeta \\ 1 & 1 \end{pmatrix},$$

conjugating (9.3) by $\mathbf{U}(\zeta_n)$ and defining the unknown $\mathbf{v}_n := \mathbf{u}_n\mathbf{U}(\zeta_n)$ moves all coupling into the off-diagonal blocks. The system becomes

$$\mathcal{C}^+_{I_n}\mathbf{v}_n - \mathcal{C}^-_{I_n}\mathbf{v}_n\mathbf{D} + \sum_{k \ne n} \mathcal{C}_{I_k}\mathbf{v}_k\,\mathbf{U}(\zeta_k)^*\mathbf{U}(\zeta_n)\,(\mathbf{I} - \mathbf{D}) = \begin{pmatrix} 1 & 1 \end{pmatrix}\mathbf{U}(\zeta_n)(\mathbf{D} - \mathbf{I}), \quad z \in I_n. \tag{9.4}$$

To remove the noncompact diagonal operator, we precondition (9.4) on the left by the block-diagonal inverse

$$\left(\mathcal{C}^+_{I_n} - \mathcal{C}^-_{I_n}(\diamond)\,\mathbf{D}\right)^{-1} = \begin{pmatrix} \left(\mathcal{C}^+_{I_n} - \mathcal{C}^-_{I_n}\right)^{-1} & 0 \\ 0 & \left(\mathcal{C}^+_{I_n} + \mathcal{C}^-_{I_n}\right)^{-1} \end{pmatrix}, \tag{9.5}$$

which is available in closed form for our chosen bases. Following the endpoint behavior in Riemann–Hilbert Problem 9.2.1, each component density is expanded in weighted Chebyshev-V or Chebyshev-W series. On the unit interval $[-1, 1]$ [108],

$$\begin{aligned} (\mathcal{C}^+ - \mathcal{C}^-)[V_k w_V] &= V_k w_V, & (\mathcal{C}^+ - \mathcal{C}^-)[W_k w_W] &= W_k w_W, \\ (\mathcal{C}^+ + \mathcal{C}^-)[V_k w_V] &= \tfrac{1}{\pi \mathrm{i}} W_k, & (\mathcal{C}^+ + \mathcal{C}^-)[W_k w_W] &= -\tfrac{1}{\pi \mathrm{i}} V_k, \end{aligned}$$

and these formulae can be shifted and scaled appropriately. To apply (9.5) to the resulting collocation system, one only needs to map between Chebyshev polynomial coefficients and values which can be done in a matrix-free manner via appropriate discrete cosine and sine transforms [119, Table 4.1]. Once (9.5) is applied to (9.4), we are left with a system of the

form

$$(\mathbf{I} + \mathcal{K})\,\mathbf{v} = \mathbf{f}, \tag{9.6}$$

where $\mathcal{K}$ is compact. The application of $\mathcal{K}$ in the collocation system requires applying (9.5) and evaluating the weighted Cauchy integrals of Chebyshev polynomials away from their support which can be done in a matrix-free manner via the formulae of Theorem 2.5.1. The collocation system is solved by matrix-free GMRES in a small number of iterations [33, Figure 12].

9.4 Dispersive quantization

One instance of where large-genus solutions are useful is in modeling a phenomenon known as *dispersive quantization*. When the linearized KdV equation is evolved from periodic piecewise-constant initial data, at times corresponding to rational multiples of the domain length, the solution is quantized, forming a profile consisting of piecewise-constant steps. At other times, the solution exhibits fractal-like, continuous but nowhere-differentiable behavior [117]. Understanding this phenomenon in the context of the (nonlinear) KdV equation has been the subject of recent numerical experiments [42] and theoretical analysis [64, 88]. To illustrate dispersive quantization with our method, we compute the finite-genus KdV solution approximating a box initial condition from [33, Section 5.4]. Figure 9.2 displays the evolution at selected times, contrasting a low-genus ($g = 3$) and a large-genus ($g = 1600$) approximation. The large-genus solution accurately captures the formation of the dispersive quantization profile.

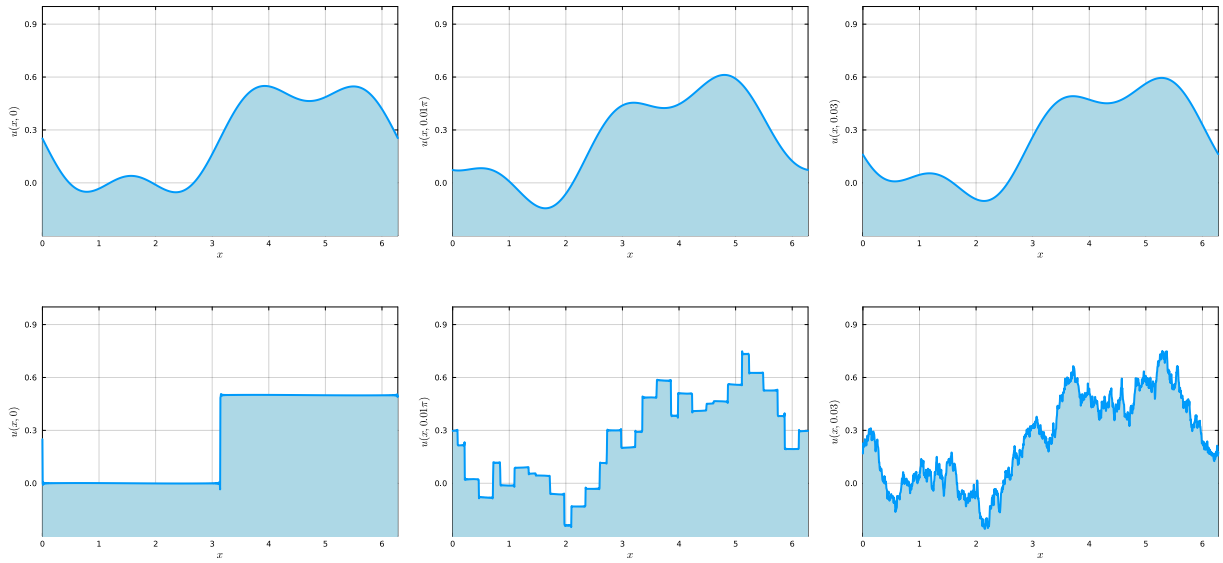


Figure 9.2: Snapshots of the finite-genus approximation of a box initial condition. Top row: genus $g = 3$. Bottom row: genus $g = 1600$, demonstrating the resolution of the dispersive quantization structure. Computations performed with the method of this chapter. Cf. [33, Figure 10].

BIBLIOGRAPHY

[1] M. J. Ablowitz and A. S. Fokas. *Complex Variables: Introduction and Applications*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2 edition, 2003.

[2] N. Achyeser. Verallgemeinerung einer Korkine-Zolotareffschen Minimum-Aufgabe. *Communications de l'Institut des Sciences Mathématiques et Mécaniques de l'Université de Kharkoff et de la Société Mathématique de Kharkoff*, 13(1):3–14, 1936.

[3] B. Adcock and D. Huybrechs. Frames and Numerical Approximation. *SIAM Review*, 61(3):443–473, 2019.

[4] B. Adcock and D. Huybrechs. Frames and Numerical Approximation II: Generalized Sampling. *Journal of Fourier Analysis and Applications*, 26, 2020.

[5] N. I. Akhiezer. Orthogonal polynomials on several intervals. *Soviet Math. Dokl.*, 1:1006–1009, 1960.

[6] N. I. Akhiezer. *Elements of the Theory of Elliptic Functions*, volume 79 of *Translations of Mathematical Monographs*. American Mathematical Society, 1970.

[7] N. I. Akhiezer and Y. Y. Tomcuk. On the theory of orthogonal polynomials over several intervals. *Soviet Math. Dokl.*, 2:687–690, 1961.

[8] A. H. Al-Mohy and N. J. Higham. The complex step approximation to the Fréchet derivative of a matrix function. *Numerical Algorithms*, 53(1):133–148, 2009.

[9] N Amsel, D. Persson, C. Musco, and R. Gower. The Polar Express: Optimal Matrix Sign Methods and Their Application to the Muon Algorithm. *arXiv preprint 2505.16932*, 2025.

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

[11] A. I. Aptekarev. Asymptotic properties of polynomials orthogonal on a system of contours, and periodic motions of Toda lattices. *Mathematics of the USSR-Sbornik*, 53(1):233–260, February 1986.

[12] P. J. Baddoo and L. N. Trefethen. Log-lightning computation of capacity and Green's function. *Maple Transactions*, 1(1):14124, 2021.

[13] C. Ballew. `https://github.com/cade-b/RecurrenceCoefficients.jl`, 2023.

[14] C. Ballew. `https://github.com/cade-b/AkhiezerIteration`, 2024.

[15] C. Ballew. `https://github.com/cade-b/KdVSolitonGas.jl`, 2025.

[16] C. Ballew, D. Bilman, and T. Trogdon. Efficient computation of soliton gas primitive potentials. *Journal of Nonlinear Waves*, 1:e12, 2025.

[17] C. Ballew, D. Bilman, and T. Trogdon. Long-time computations of KdV and mKdV soliton gasses. *In preparation*, 2026+.

[18] C. Ballew and T. Trogdon. A Riemann–Hilbert approach to computing the inverse spectral map for measures supported on disjoint intervals. *Studies in Applied Mathematics*, 152(1):31–72, 2024.

[19] C. Ballew and T. Trogdon. The Akhiezer iteration. In *Contemporary Mathematics*, volume 822, pages 1–34. American Mathematical Society, 2025.

[20] C. Ballew, T. Trogdon, and H. Wilber. The Akhiezer iteration and inverse-free solvers for Sylvester matrix equations. *arXiv preprint 2503.17496*, 2025.

[21] C. Ballew and H. Wilber. `https://github.com/cade-b/AkhiezerSylvester`, 2025.

[22] J. Banks, J. Garza-Vargas, A. Kulkarni, and N. Srivastava. Pseudospectral Shattering, the Sign Function, and Diagonalization in Nearly Matrix Multiplication Time. *Foundations of Computational Mathematics*, 23(6):1959–2047, 2022.

[23] R. H. Bartels and G. W. Stewart. Algorithm 432 [C2]: Solution of the matrix equation AX + XB = C [F4]. *Commun. ACM*, 15(9):820–826, 1972.

[24] M. Bebendorf. Approximation of boundary element matrices. *Numerische Mathematik*, 86(4):565–589, 2000.

[25] B. Beckermann. An error analysis for rational Galerkin projection applied to the Sylvester equation. *SIAM Journal on Numerical Analysis*, 49(6):2430–2450, 2011.

[26] B. Beckermann, A. Cortinovis, D. Kressner, and M. Schweitzer. Low-rank updates of matrix functions II: Rational Krylov methods. *SIAM Journal on Numerical Analysis*, 59(3):1325–1347, 2021.

[27] B. Beckermann and A. Townsend. Bounds on the singular values of matrices with displacement structure. *SIAM Review*, 61(2):319–344, 2019.

[28] P. Benner. Solving large-scale control problems. *IEEE Control Systems Magazine*, 24(1):44–59, 2004.

[29] P. Benner, R.-C. Li, and N. Truhar. On the ADI method for Sylvester equations. *Journal of Computational and Applied Mathematics*, 233(4):1035–1045, 2009.

[30] P. Benner and E. S. Quintana-Ortí. Solving stable generalized Lyapunov equations with the matrix sign function. *Numerical Algorithms*, 20(1):75–100, 1999.

[31] M. Berljafa and S. Guttel. The RKFIT algorithm for nonlinear rational approximation. *SIAM Journal on Scientific Computing*, 39(5):A2049–A2071, 2017.

[32] S. Bernstein. Sur l'ordre de la meilleure approximation des fonctions continues par des polynômes de degré donné. *Mémoires de l'Académie Royale de Belgique, Classe des Sciences*, 4(1):1–104, 1912.

[33] D. Bilman, P. Nabelek, and T. Trogdon. Computation of large-genus solutions of the Korteweg–de Vries equation. *Physica D: Nonlinear Phenomena*, 449:133715, 2023.

[34] D Bilman and T Trogdon. Numerical Inverse Scattering for the Toda Lattice. *Communications in Mathematical Physics*, 352(2):805–879, 2017.

[35] D. A. Bini, B. Iannazzo, and B. Meini. *Numerical Solution of Algebraic Riccati Equations*. Society for Industrial and Applied Mathematics, 2011.

[36] I. Bogaert. Iteration-Free Computation of Gauss–Legendre Quadrature Nodes and Weights. *SIAM Journal on Scientific Computing*, 36(3):A1008–A1026, 2014.

[37] N. Boullé and A. Townsend. Computing with Functions in the Ball. *SIAM Journal on Scientific Computing*, 42(4):C169–C191, 2020.

[38] S. Brahma and B. Datta. An optimization approach for minimum norm and robust partial quadratic eigenvalue assignment problems for vibrating structures. *Journal of Sound and Vibration*, 324(3-5):471–489, 2009.

[39] D. Calvetti and L. Reichel. Application of ADI iterative methods to the restoration of noisy images. *SIAM Journal on Matrix Analysis and Applications*, 17(1):165–186, 1996.

[40] B. C. Carlson. Numerical computation of real or complex elliptic integrals. *Numer. Algorithms*, 10(1-2):13–26, 1995.

[41] A. A. Casulli, F. Hrobat, and D. Kressner. Lanczos with compression for symmetric matrix Lyapunov equations. *arXiv preprint 2505.22498*, 2025.

[42] G. Chen and P. J. Olver. Numerical simulation of nonlinear dispersive quantization. *Discrete amp; Continuous Dynamical Systems - A*, 34(3):991–1008, 2014.

[43] J. Chen, M. Anitescu, and Y. Saad. Computing $f(A)b$ via Least Squares Polynomial Approximations. *SIAM Journal on Scientific Computing*, 33(1):195–222, 2011.

[44] J. Chen and E. Chow. A stable scaling of Newton-Schulz for improving the sign function computation of a Hermitian matrix. *Preprint ANL/MCS-P5059-0114*, 2014.

[45] T. Chen, T. Trogdon, and S. Ubaru. Randomized matrix-free quadrature for spectrum and spectral sum approximation. *arXiv preprint 2204.01941*, 2022.

[46] Y. Chen and A. R. Its. A Riemann–Hilbert approach to the Akhiezer polynomials. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1867):973–1003, 2008.

[47] Y. Chen and N. Lawrence. A generalization of the Chebyshev polynomials. *J. Phys. A: Math. Gen*, 35:4651–4699, 2002.

[48] Ju. L. Daleckiĭ and M. G. Kreĭn. *Stability of solutions of differential equations in Banach space*, volume 43 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, Rhode Island, 1974.

[49] C. de Boor and J. R. Rice. Extremal Polynomials with Application to Richardson Iteration for Indefinite Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 3(1):47–57, 1982.

[50] B. Deconinck, M. Heil, A. Bobenko, M. van Hoeij, and M. Schmies. Computing Riemann Theta Functions. *Mathematics of Computation*, 73(247):1417–1442, 2004.

[51] B. Deconinck and M. S. Patterson. Computing the Abel map. *Physica D: Nonlinear Phenomena*, 237(24):3214–3232, 2008.

[52] B. Deconinck and M. S. Patterson. Computing with Plane Algebraic Curves and Riemann Surfaces: The Algorithms of the Maple Package `algcurves`. In *Computational Approach to Riemann Surfaces*, volume 2013 of *Lecture Notes in Mathematics*, pages 67–123. Springer, Berlin, Heidelberg, 2011.

[53] P. Deift. *Orthogonal polynomials and random matrices: A Riemann-Hilbert approach*. American Math. Soc., 2000.

[54] P. Deift, L. C. Li, and C. Tomei. Toda flows with infinitely many variables. *Journal of Functional Analysis*, 64(3):358–402, 1985.

[55] P Deift and X Zhou. A steepest descent method for oscillatory Riemann–Hilbert problems. Asymptotics for the MKdV Equation. *Annals of Mathematics*, 137(2):295–368, 1993.

[56] P. A. Deift and X. Zhou. Asymptotics for the Painlevé II equation. *Communications on Pure and Applied Mathematics*, 48(3):277–337, 1995.

[57] E. D. Denman and A. N. Beavers, Jr. The matrix sign function and computations in systems. *Applied mathematics and Computation*, 2(1):63–94, 1976.

[58] X. Ding and T. Trogdon. A Riemann–Hilbert Approach to the Perturbation Theory for Orthogonal Polynomials: Applications to Numerical Linear Algebra and Random Matrix Theory. *International Mathematics Research Notices*, 2023.

[59] *NIST Digital Library of Mathematical Functions*. http://dlmf.nist.gov/, Release 1.1.8 of 2022-12-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

[60] V. Druskin, L. Knizhnerman, and V. Simoncini. Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation. *SIAM Journal on Matrix Analysis and Applications*, 49(5):1875–1898, 2011.

[61] S. Dyachenko, D. Zakharov, and V. Zakharov. Primitive potentials and bounded solutions of the KdV equation. *Physica D: Nonlinear Phenomena*, 333:148–156, 2016.

[62] I. Egorova, M. Piorkowski, and G. Teschl. Asymptotics of KdV shock waves via the Riemann-Hilbert approach. *Indiana Univ. Math. J.*, 73:645–690, 2024.

[63] M. Embree and L. N. Trefethen. Green's Functions for Multiply Connected Domains via Conformal Mapping. *SIAM Review*, 41(4):745–761, 1999.

[64] M. B. Erdoğan and N. Tzirakis. Global Smoothing for the Periodic KdV Evolution. *International Mathematics Research Notices*, 2013(20):4589–4614, 2013.

[65] A. Eremenko and P. Yuditskii. Polynomials of the best uniform approximation to sgn(x) on two intervals. *Journal d'Analyse Mathématique*, 114(1):285–315, 2011.

[66] J. Favard. Sur les polynômes de Tchebicheff. *Comptes Rendus de l'Académie des Sciences, Paris*, 200:2052–2053, 1935.

[67] B. Fischer. Chebyshev Polynomials for Disjoint Compact Sets. *Constructive Approximation*, 8:309–329, 1992.

[68] B. Fischer. *Polynomial Based Iteration Methods for Symmetric Linear Systems*. Society for Industrial and Applied Mathematics, 2011.

[69] D. A. Flanders and G. Shortley. Numerical Determination of Fundamental Modes. *Journal of Applied Physics*, 21(12):1326–1332, 1950.

[70] A. S. Fokas, A. R. Its, and A. V. Kitaev. The isomonodromy approach to matric models in 2D quantum gravity. *Communications in Mathematical Physics*, 147(2):395–430, 1992.

[71] D. Fortunato and A. Townsend. Fast Poisson solvers for spectral methods. *IMA Journal of Numerical Analysis*, 40(3):1994–2018, 2020.

[72] Z. Gajic and M. T. J. Qureshi. *Lyapunov matrix equation in system stability and control*. Courier Corporation, 2008.

[73] W. Gautschi. Is the recurrence relation for orthogonal polynomials always stable? *BIT*, 33(2):277–284, 1993.

[74] W. Gautschi. *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, 2004.

[75] E. S. Gawlik and M. Leok. Iterative Computation of the Fréchet Derivative of the Polar Decomposition. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1354–1379, 2017.

[76] J. S. Geronimo. Scattering Theory, Orthogonal Polynomials, and q-Series. *SIAM Journal on Mathematical Analysis*, 25(2):392–419, 1994.

[77] J. S. Geronimo and W. Van Assche. Orthogonal polynomials on several intervals via a polynomial mapping. *Transactions of the American Mathematical Society*, 308(2):559–581, 1988.

[78] M. Girotti, T. Grava, R. Jenkins, and K. D. T.-R. McLaughlin. Rigorous asymptotics of a KdV soliton gas. *Communications in Mathematical Physics*, 384(2):733–784, 2021.

[79] M. Girotti, T. Grava, R. Jenkins, K. T.-R. McLaughlin, and A. Minakov. Soliton versus the gas: Fredholm determinants, analysis, and the rapid oscillations behind the kinetic equation. *Communications on Pure and Applied Mathematics*, 76(11):3233–3299, 2023.

[80] G. H. Golub and C. F. Van Loan. *Matrix Computations*. JHU Press, Baltimore, MD, 4 edition, 2013.

[81] W. B. Gragg and W. J. Harrod. The numerically stable reconstruction of Jacobi matrices from spectral data. *Numerische Mathematik*, 44(3):317–335, 1984.

[82] S. Gugercin, D. C. Sorensen, and A. C. Antoulas. A modified low-rank Smith method for large-scale Lyapunov equations. *Numerical Algorithms*, 32(1):27–55, 2003.

[83] M. H. Gutknecht and S. Röllin. The Chebyshev iteration revisited. *Parallel Computing*, 28(2):263–283, 2002.

[84] T. S. Gutleb, S. Olver, and R. M. Slevinsky. Polynomial and Rational Measure Modifications of Orthogonal Polynomials via Infinite-Dimensional Banded Matrix Factorizations. *Foundations of Computational Mathematics*, 25(5), 2025.

[85] N. Hale and A. Townsend. Fast and Accurate Computation of Gauss-Legendre and Gauss-Jacobi Quadrature Nodes and Weights. *SIAM J. Sci. Comput.*, 35(2):A652–A674, 2013.

[86] N. J. Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

[87] N. J. Higham and S. D. Relton. Higher Order Fréchet Derivatives of Matrix Functions and the Level-2 Condition Number. *SIAM Journal on Matrix Analysis and Applications*, 35(3):1019–1037, 2014.

[88] M. Hofmanová and K. Schratz. An exponential-type integrator for the KdV equation. *Numerische Mathematik*, 136(4):1117–1137, December 2016.

[89] A. S. Householder. Unitary Triangularization of a Nonsymmetric Matrix. *J. ACM*, 5(4):339–342, 1958.

[90] D. Huybrechs and P. Opsomer. Construction and implementation of asymptotic expansions for Laguerre-type orthogonal polynomials. *IMA Journal of Numerical Analysis*, 38(3):1085–1118, 2018.

[91] A. R. Its and V. B. Matveev. Hill's operator with finitely many gaps. *Functional Analysis and Its Applications*, 9(1):69–70, 1975.

[92] C. Kenney and A. J. Laub. Polar Decomposition and Matrix Sign Function Condition Estimates. *SIAM Journal on Scientific and Statistical Computing*, 12(3):488–504, 1991.

[93] M. G. Kreĭn. *Lektsii po teorii ustoichivosti reshenii differentsialnykh uravnenii v Banakhovom prostranstve*. Izdat. Akad. Nauk Ukrain. SSR, Kiev, 1964.

[94] D. Kressner. Memory-efficient Krylov subspace techniques for solving large-scale Lyapunov equations. In *2008 IEEE International Conference on Computer-Aided Control Systems*, pages 613–618, 2008.

[95] D. Kressner. Bivariate matrix functions. *Operators and Matrices*, 8(2):449–466, 2014.

[96] D. Kressner, K. Lund, S. Massei, and D. Palitta. Compress-and-restart block Krylov subspace methods for Sylvester matrix equations. *Numerical Linear Algebra with Applications*, 28(1):e2339, 2021.

[97] D. Kressner, S. Massei, and L. Robol. Low-rank updates and a divide-and-conquer method for linear matrix equations. *SIAM Journal on Scientific Computing*, 41(2):A848–A876, 2019.

[98] D. Kressner and C. Tobler. Krylov Subspace Methods for Linear Systems with Tensor Product Structure. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1688–1714, 2010.

[99] A. B. J. Kuijlaars. *Riemann-Hilbert Analysis for Orthogonal Polynomials*, pages 167–210. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[100] A. B. J. Kuijlaars, K. T.-R. McLaughlin, W. Van Assche, and M. Vanlessen. The Riemann-Hilbert approach to strong asymptotics for orthogonal polynomials on [–1,1]. *Advances in Mathematics*, 188(2):337–398, 2004.

[101] H. Labrande. Computing Jacobi's θ in quasi-linear time. *Math. Comp.*, 87(311):1479–1508, 2018.

[102] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255, 1950.

[103] A. Laub. Numerical linear algebra aspects of control design computations. *IEEE transactions on automatic control*, 30(2):97–108, 1985.

[104] J.-R. Li and J. White. Low rank solution of Lyapunov equations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):260–280, 2002.

[105] T. A. Manteuffel. The Tchebychev iteration for nonsymmetric linear systems. *Numerische Mathematik*, 28(3):307–327, 1977.

[106] T. A. Manteuffel. Adaptive procedure for estimating parameters for the nonsymmetric Tchebychev iteration. *Numerische Mathematik*, 31(2):183–208, June 1978.

[107] F. Marcellán and R. Álvarez-Nodarse. On the "Favard theorem" and its extensions. *Journal of Computational and Applied Mathematics*, 127(1–2):231–254, 2001.

[108] J. C. Mason. Chebyshev polynomials of the second, third and fourth kinds in approximation, indefinite integration, and integral transforms. *Journal of Computational and Applied Mathematics*, 49(1–3):169–178, 1993.

[109] J. C. Mason and D. C. Handscomb. *Chebyshev Polynomials*. Chapman and Hall/CRC, 2002.

[110] S. Massei, M. Mazza, and L. Robol. Fast solvers for two-dimensional fractional diffusion equations using rank structured matrices. *SIAM Journal on Scientific Computing*, 41(4):A2627–A2656, 2019.

[111] R. Mathias. A Chain Rule for Matrix Functions and Applications. *SIAM Journal on Matrix Analysis and Applications*, 17(3):610–620, 1996.

[112] J. Moser. Three integrable Hamiltonian systems connected with isospectral deformations. *Advances in Mathematics*, 16(2):197–220, 1975.

[113] N. I. Muskhelishvili. *Singular Integral Equations*. Springer Netherlands, 1977.

[114] Y. Nakatsukasa, Z. Bai, and F. Gygi. Optimizing Halley's iteration for computing the matrix polar decomposition. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2700–2720, 2010.

[115] Y. Nakatsukasa and R. W Freund. Computing fundamental matrix decompositions accurately via the matrix sign function in two iterations: The power of Zolotarev's functions. *SIAM Review*, 58(3):461–493, 2016.

[116] Y. Nakatsukasa, O. Sète, and L. N. Trefethen. The AAA Algorithm for Rational Approximation. *SIAM Journal on Scientific Computing*, 40(3):A1494–A1522, January 2018.

[117] P. J. Olver. Dispersive Quantization. *The American Mathematical Monthly*, 117(7):599–610, 2010.

[118] S. Olver. A general framework for solving Riemann–Hilbert problems numerically. *Numerische Mathematik*, 122(2):305–340, 2012.

[119] S. Olver, R. M. Slevinsky, and A. Townsend. Fast algorithms using orthogonal polynomials. *Acta Numerica*, 29:573–699, 2020.

[120] S. Olver and T. Trogdon. Numerical solution of Riemann–Hilbert Problems: Random matrix theory and orthogonal polynomials. *Constructive Approximation*, 39(1):101–149, 2013.

[121] D. Palitta, M. Schweitzer, and V. Simoncini. Sketched and truncated polynomial Krylov subspace methods: Matrix Sylvester equations. *Mathematics of Computation*, 2024.

[122] D. Palitta and V. Simoncini. Computationally enhanced projection methods for symmetric Sylvester and Lyapunov matrix equations. *Journal of Computational and Applied Mathematics*, 330:648–659, 2018.

[123] B. N. Parlett. *The Symmetric Eigenvalue Problem*. SIAM, Philadelphia, PA, 1998.

[124] F. Peherstorfer. Orthogonal- and Chebyshev polynomials on two intervals. *Acta Mathematica Hungarica*, 55(3–4):245–278, 1990.

[125] F. Peherstorfer. Orthogonal polynomials on several intervals: Accumulation points of recurrence coefficients and of zeros. *Journal of Approximation Theory*, 163(7):814–837, 2011. 10th International Symposium on "Orthogonal Polynomials, Special Functions and Applications".

[126] T. Penzl. A cyclic low-rank Smith method for large sparse Lyapunov equations. *SIAM Journal on Scientific Computing*, 21(4):1401–1418, 1999.

[127] T. Ransford. *Potential Theory in the Plane*, volume 28 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1995.

[128] I. Redor, E. Barthélemy, H. Michallet, M. Onorato, and N. Mordant. Experimental Evidence of a Hydrodynamic Soliton Gas. *Phys. Rev. Lett.*, 122:214502, 2019.

[129] T. J. Rivlin. *An introduction to the approximation of functions*. Dover books on advanced mathematics. Dover, unabridged and corr. republication of the 1969 ed edition, 1981.

[130] J. D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *International Journal of Control*, 32(4):677–687, 1980.

[131] A. Ruhe. Rational Krylov sequence methods for eigenvalue computation. *Linear Algebra and its Applications*, 58:391–405, 1984.

[132] Y. Saad. Iterative Solution of Indefinite Symmetric Linear Systems by Methods Using Orthogonal Polynomials over Two Disjoint Intervals. *SIAM Journal on Numerical Analysis*, 20(4):784–811, 1983.

[133] J. Sabino. *Solution of large-scale Lyapunov equations via the block modified Smith method*. PhD thesis, Rice University, 2007.

[134] E. B. Saff and V. Totik. *Logarithmic Potentials with External Fields*, volume 316 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, Berlin, 1997.

[135] J. Shen, G. Strang, and A. J. Wathen. The potential theory of several intervals and its applications. *Applied Mathematics & Optimization*, 44(1):67–85, 2001.

[136] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM Journal on Scientific Computing*, 29(3):1268–1288, 2007.

[137] V. Simoncini. Computational methods for linear matrix equations. *SIAM Review*, 58(3):377–441, 2016.

[138] V. Simoncini and V. Druskin. Convergence Analysis of Projection Methods for the Numerical Solution of Large Lyapunov Equations. *SIAM Journal on Numerical Analysis*, 47(2):828–843, 2009.

[139] J. Słomka, A. Townsend, and J. Dunkel. Stokes' second problem and reduction of inertia in active fluids. *Physical Review, Fluids*, 3(10):103304, 2018.

[140] R. A. Smith. Matrix Equation XA+BX=C. *SIAM Journal on Applied Mathematics*, 16(1):198–201, 1968.

[141] K. Sogo. Time-Dependent Orthogonal Polynomials and Theory of Soliton Applications to Matrix Model, Vertex Model and Level Statistics. *Journal of The Physical Society of Japan - J PHYS SOC JPN*, 62:1887–1894, 1993.

[142] S. R. Spangler and J. P. Sheerin. A soliton gas model for astrophysical magnetized plasma turbulence. *The Astrophysical Journal*, 257:855, 1982.

[143] H. Stahl and V. Totik. *General Orthogonal Polynomials*, volume 43 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1992.

[144] T. J. Stieltjes. Recherches sur les fractions continues. *Annales de la Faculté des Sciences de Toulouse*, 8:J1–J122, 1894.

[145] P. Suret, M. Dufour, G. Roberti, G. El, F. Copie, and S. Randoux. Soliton refraction by an optical soliton gas. *Phys. Rev. Res.*, 5:L042002, 2023.

[146] P. Suret, S. Randoux, A. Gelash, D. Agafontsev, B. Doyon, and G. El. Soliton gas: Theory, numerics, and experiments. *Phys. Rev. E*, 109:061001, 2024.

[147] G. Szegő. *Orthogonal Polynomials*, volume 23 of *Colloquium Publications*. American Mathematical Society, Providence, Rhode Island, 1939.

[148] L. H. Thomas. Elliptic problems in linear difference equations over a network. Technical report, Watson Scientific Computing Laboratory, Columbia University, New York, 1949.

[149] A. Townsend. *Computing with functions in two dimensions*. DPhil thesis, University of Oxford, 2014.

[150] A. Townsend, T. Trogdon, and S. Olver. Fast computation of Gauss quadrature nodes and weights on the whole real line. *IMA Journal of Numerical Analysis*, 36(1):337–358, 2014.

[151] L. N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2019.

[152] L. N. Trefethen and D. Bau III. *Numerical Linear Algebra*. SIAM, Philadelphia, PA, 1997.

[153] L. N. Trefethen and J. A. C. Weideman. The Exponentially Convergent Trapezoidal Rule. *SIAM Review*, 56(3):385–458, 2014.

[154] L. N. Trefethen and H. D. Wilber. Computation of Zolotarev Rational Functions. *SIAM Journal on Scientific Computing*, 47(4):A2205–A2220, 2025.

[155] T. Trogdon. https://github.com/tomtrogdon/OperatorApproximation.jl, 2025.

[156] T. Trogdon. On the convergence of Fourier spectral methods involving noncompact operators. *IMA Journal of Numerical Analysis*, 45(3):1743–1785, 2025.

[157] T. Trogdon and B. Deconinck. A Riemann–Hilbert problem for the finite-genus solutions of the KdV equation and its numerical solution. *Physica D: Nonlinear Phenomena*, 251:1–18, 2013.

[158] T. Trogdon and S. Olver. A Riemann–Hilbert approach to Jacobi operators and Gaussian quadrature. *IMA Journal of Numerical Analysis*, 36(1):174–196, 2015.

[159] T. Trogdon and S. Olver. *Riemann–Hilbert Problems, Their Numerical Solution, and the Computation of Nonlinear Special Functions*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015.

[160] T. Trogdon, S. Olver, and B. Deconinck. Numerical inverse scattering for the Korteweg–de Vries and modified Korteweg–de Vries equations. *Physica D: Nonlinear Phenomena*, 241(11):1003–1025, 2012.

[161] P. M. Van Dooren. Structured linear algebra problems in digital signal processing. In *Numerical linear algebra, digital signal processing and parallel algorithms*, pages 361–384. Springer, 1991.

[162] J. L. Walsh. *Interpolation and Approximation by Rational Functions in the Complex Domain*, volume 20 of *Colloquium Publications*. American Mathematical Society, Providence, RI, 5th edition, 1969.

[163] E. T. Whittaker and G. N. Watson. *A Course of Modern Analysis*. Cambridge University Press, 4th edition, 1927.

[164] H. Widom. Extremal polynomials associated with a system of curves in the complex plane. *Advances in Mathematics*, 3(2):127–232, 1969.

[165] M. L. Yattselev. Nuttall's theorem with analytic weights on algebraic S-contours. *Journal of Approximation Theory*, 190:73–90, 2015. Special Issue Dedicated to the memory of Andrei Aleksandrovich Gonchar and Herbert Stahl.

[166] V. Zakharov. Kinetic equation for solitons. *Sov. Phys. JETP*, 33(3):538–541, 1971.

[167] V. Zakharov and S. Manakov. Construction of higher-dimensional nonlinear integrable systems and of their solutions. *Functional Analysis and Its Applications*, 19(2):89–101, 1985.

Appendix A

DEFERRED PROOFS

A.1 Akhiezer polynomial recurrence coefficients

Assume that the nth degree orthonormal polynomial can be represented as

$$p_n(x) = k_n x^n + \ell_n x^{n-1} + \dots.$$

Then, the recurrence coefficients can be recovered by [53]

$$\check{a}_n = \frac{\ell_n}{k_n} - \frac{\ell_{n+1}}{k_{n+1}}, \quad \check{b}_n = \frac{k_n}{k_{n+1}}, \tag{A.1}$$

with the convention that $\ell_0 = 0, k_0 = 1$. To obtain such a representation, we must expand (2.22) around $u = -\rho$, the point corresponding to $x = \infty$ on one sheet of the Riemann surface. Noting that H is an odd function, we expand

$$\begin{aligned}\frac{H(u-\rho)}{H(u+\rho)} &= \left(-H(2\rho) + H'(2\rho)(u+\rho) + \mathrm{O}((u+\rho)^2)\right) \frac{1}{H'(0)(u+\rho)} \left(1 + \mathrm{O}((u+\rho)^2)\right)^{-1} \\ &= -\frac{H(2\rho)}{H'(0)} \frac{1}{u+\rho} + \frac{H'(2\rho)}{H'(0)} + \mathrm{O}(u+\rho).\end{aligned}$$

Similarly, Θ is an even function, and we expand

$$\begin{aligned}\frac{\Theta(u+2n\rho)}{\Theta(u)} &= \left(\Theta\left((2n-1)\rho\right) + \Theta'\left((2n-1)\rho\right)(u+\rho) + \dots\right) \\ &\qquad \cdot \frac{1}{\Theta(\rho)} \left(1 - \frac{\Theta'(\rho)}{\Theta(\rho)}(u+\rho) + \dots\right)^{-1} \\ &= \frac{\Theta\left((2n-1)\rho\right)}{\Theta(\rho)} + \left(\frac{\Theta'\left((2n-1)\rho\right)}{\Theta(\rho)} + \frac{\Theta\left((2n-1)\rho\right)\Theta'(\rho)}{\Theta^2(\rho)}\right)(u+\rho) \\ &\qquad + \mathrm{O}\left((u+\rho)^2\right).\end{aligned}$$

The other product summed in (2.23) is $\mathrm{O}(u+\rho)$ and will not contribute to the recurrence coefficients. To convert these expansions from $u+\rho$ to x, we utilize the relation

$$x - \alpha = \frac{1-\alpha^2}{2(\mathrm{sn}^2\, u - \mathrm{sn}^2\, \rho)},$$

which we rewrite as

$$(\operatorname{sn} u + \operatorname{sn}\rho)(\operatorname{sn} u - \operatorname{sn}\rho) = \frac{1-\alpha^2}{2(x-\alpha)}. \tag{A.2}$$

Noting that sn is an odd function, we expand

$$\begin{aligned}(\operatorname{sn} u + \operatorname{sn}\rho)(\operatorname{sn} u - \operatorname{sn}\rho) &= \left(\operatorname{sn}'\rho(u+\rho) - \frac{\operatorname{sn}''\rho}{2}(u+\rho)^2\ldots\right)(-2\operatorname{sn}\rho + \operatorname{sn}'\rho(u+\rho)+\ldots)\\ &= -2\operatorname{sn}\rho\operatorname{sn}'\rho(u+\rho)\left(1-\left(\frac{\operatorname{sn}'\rho}{2\operatorname{sn}\rho}+\frac{\operatorname{sn}''\rho}{2\operatorname{sn}'\rho}\right)(u+\rho)\right.\\ &\quad \left.+\,\mathrm{O}\left((u+\rho)^2\right)\right).\end{aligned}$$

Substituting this into (A.2), solving for $u+\rho$, and iterating,

$$\begin{aligned}u+\rho &= -\frac{1-\alpha^2}{4\operatorname{sn}\rho\operatorname{sn}'\rho}\left(1-\left(\frac{\operatorname{sn}'\rho}{2\operatorname{sn}\rho}+\frac{\operatorname{sn}''\rho}{2\operatorname{sn}'\rho}\right)(u+\rho)+\mathrm{O}\left((u+\rho)^2\right)\right)^{-1}\frac{1}{x-\alpha}\\ &= -\frac{1-\alpha^2}{4\operatorname{sn}\rho\operatorname{sn}'\rho}\left(1-\frac{(1-\alpha^2)}{4\operatorname{sn}\rho\operatorname{sn}'\rho}\left(\frac{\operatorname{sn}'\rho}{2\operatorname{sn}\rho}+\frac{\operatorname{sn}''\rho}{2\operatorname{sn}'\rho}\right)\frac{1}{x-\alpha}+\mathrm{O}\left(\left(\frac{1}{x-\alpha}\right)^2\right)\right)\frac{1}{x-\alpha}.\end{aligned}$$

Inverting this,

$$\frac{1}{u+\rho} = -\frac{4\operatorname{sn}\rho\operatorname{sn}'\rho}{1-\alpha^2}(x-\alpha) - \left(\frac{\operatorname{sn}'\rho}{2\operatorname{sn}\rho}+\frac{\operatorname{sn}''\rho}{2\operatorname{sn}'\rho}\right)+\mathrm{O}\left(\frac{1}{x-\alpha}\right).$$

Substituting these into the series in $u+\rho$,

$$\begin{aligned}&\frac{H(u-\rho)}{H(u+\rho)} = \frac{H(2\rho)}{H'(0)}\frac{4\operatorname{sn}\rho\operatorname{sn}'\rho}{1-\alpha^2}(x-\alpha)+\frac{H(2\rho)}{H'(0)}\left(\frac{\operatorname{sn}'\rho}{2\operatorname{sn}\rho}+\frac{\operatorname{sn}''\rho}{2\operatorname{sn}'\rho}\right)+\frac{H'(2\rho)}{H'(0)}+\mathrm{O}\left(\frac{1}{x-\alpha}\right),\\ &\frac{\Theta(u+2n\rho)}{\Theta(u)} = \frac{\Theta\left((2n-1)\rho\right)}{\Theta(\rho)}-\frac{1-\alpha^2}{4\operatorname{sn}\rho\operatorname{sn}'\rho}\left(\frac{\Theta'\left((2n-1)\rho\right)}{\Theta(\rho)}+\frac{\Theta\left((2n-1)\rho\right)\Theta'(\rho)}{\Theta^2(\rho)}\right)\frac{1}{x-\alpha}\\ &\quad+\mathrm{O}\left(\left(\frac{1}{x-\alpha}\right)^2\right).\end{aligned}$$

To obtain the recurrence coefficients from these expansions, denote

$$\begin{aligned}c &= \frac{H(2\rho)}{H'(0)}\frac{4\operatorname{sn}\rho\operatorname{sn}'\rho}{1-\alpha^2}, & d &= \frac{H(2\rho)}{H'(0)}\left(\frac{\operatorname{sn}'\rho}{2\operatorname{sn}\rho}+\frac{\operatorname{sn}''\rho}{2\operatorname{sn}'\rho}\right)+\frac{H'(2\rho)}{H'(0)},\\ \gamma_n &= \frac{\Theta\left((2n-1)\rho\right)}{\Theta(\rho)}, & \delta_n &= -\frac{1-\alpha^2}{4\operatorname{sn}\rho\operatorname{sn}'\rho}\left(\frac{\Theta'\left((2n-1)\rho\right)}{\Theta(\rho)}+\frac{\Theta\left((2n-1)\rho\right)\Theta'(\rho)}{\Theta^2(\rho)}\right).\end{aligned}$$

Then,

$$\begin{aligned} p_n(x) &= C_n \left(c(x-\alpha)+d+\ldots\right)^n \left(\gamma_n + \frac{\delta_n}{x-\alpha}+\ldots\right) + \mathrm{O}\left(\frac{1}{x-\alpha}\right) \\ &= C_n\gamma_n c^n x^n + C_n(n\gamma_n c^{n-1}d + \delta_n c^n - n\alpha\gamma_n c^n)x^{n-1} + \ldots, \end{aligned}$$

so

$$k_n = C_n\gamma_n c^n, \quad \ell_n = C_n(n\gamma_n c^{n-1}d + \delta_n c^n - n\alpha\gamma_n c^n).$$

Applying (A.1), we get that

$$\begin{aligned} \check{a}_n &= \frac{\ell_n}{k_n} - \frac{\ell_{n+1}}{k_{n+1}} = \left(\frac{nd}{c} + \frac{\delta_n}{\gamma_n} - n\alpha\right) - \left(\frac{(n+1)d}{c} + \frac{\delta_{n+1}}{\gamma_{n+1}} - (n+1)\alpha\right) \\ &= -\frac{d}{c} + \frac{\delta_n}{\gamma_n} - \frac{\delta_{n+1}}{\gamma_{n+1}} + \alpha, \\ \check{b}_n &= \frac{k_n}{k_{n+1}} = \frac{C_n}{C_{n+1}}\frac{1}{c}, \end{aligned}$$

for $n \geq 1$ and

$$\begin{aligned} \check{a}_0 &= -\frac{\ell_1}{k_1} = -\frac{d}{c} - \frac{\delta_1}{\gamma_1} + \alpha, \\ \check{b}_0 &= \frac{2}{C_1}\frac{1}{c}. \end{aligned}$$

Finally, we substitute in c, d, γ_n, δ_n and C_n and get

$$\begin{aligned} \check{a}_0 =& (1-\alpha^2)\left(\frac{1}{8\,\mathrm{sn}^2\,\rho} + \frac{\mathrm{sn}''\,\rho}{8\,\mathrm{sn}\,\rho\,(\mathrm{sn}'\,\rho)^2} + \frac{H'(2\rho)}{H(2\rho)}\frac{1}{4\,\mathrm{sn}\,\rho\,\mathrm{sn}'\,\rho}\right) + \frac{1-\alpha^2}{2\,\mathrm{sn}\,\rho\,\mathrm{sn}'\,\rho}\frac{\Theta'(\rho)}{\Theta(\rho)} + \alpha, \\ \check{a}_n =& (1-\alpha^2)\left(\frac{1}{8\,\mathrm{sn}^2\,\rho} + \frac{\mathrm{sn}''\,\rho}{8\,\mathrm{sn}\,\rho\,(\mathrm{sn}'\,\rho)^2}\right) \\ &+ \frac{1-\alpha^2}{4\,\mathrm{sn}\,\rho\,\mathrm{sn}'\,\rho}\left(\frac{H'(2\rho)}{H(2\rho)} + \frac{\Theta'((2n+1)\rho)}{\Theta((2n+1)\rho)} - \frac{\Theta'((2n-1)\rho)}{\Theta((2n-1)\rho)}\right) + \alpha, \quad n \geq 1, \\ \check{b}_0 =& \frac{1-\alpha^2}{4\,\mathrm{sn}\,\rho\,\mathrm{sn}'\,\rho}\frac{H'(0)}{H(2\rho)}\sqrt{\frac{2\Theta(3\rho)}{\Theta(\rho)}}, \\ \check{b}_n =& \frac{1-\alpha^2}{4\,\mathrm{sn}\,\rho\,\mathrm{sn}'\,\rho}\frac{H'(0)}{H(2\rho)}\frac{\sqrt{\Theta((2n+3)\rho)\Theta((2n-1)\rho)}}{\Theta((2n+1)\rho)}, \quad n \geq 1. \end{aligned}$$

A.2 Proof of Theorem 2.5.1

Proof. We proceed by induction by first considering the base case $k = 1$ separately for each kind of Chebyshev polynomial. Using the recurrence relation (2.6) that $C_k^{\square}$ must satisfy, we can directly compute the following.

$$\begin{aligned} C_1^T(z) &= \sqrt{2}\left(zC_0^T(z) + \frac{1}{2\pi\mathrm{i}}\right) = \sqrt{2}\left(z\frac{\mathrm{i}}{2\pi\sqrt{z-1}\sqrt{z+1}} + \frac{1}{2\pi\mathrm{i}}\right) \\ &= \sqrt{2}\frac{\mathrm{i}}{2\pi}\frac{1}{\sqrt{z-1}\sqrt{z+1}}\left(z - \sqrt{z-1}\sqrt{z+1}\right) = \sqrt{2}C_0^T(z)J_+^{-1}(z). \end{aligned}$$

$$\begin{aligned} C_1^U(z) &= 2\left(zC_0^U(z) + \frac{1}{2\pi\mathrm{i}}\right) = \frac{\mathrm{i}}{\pi}\left(z^2 - 2z\sqrt{z-1}\sqrt{z+1} + (z^2-1)\right) \\ &= \frac{\mathrm{i}}{\pi}J_+^{-1}(z)^2 = \frac{\mathrm{i}}{\pi}\left(z - \sqrt{z-1}\sqrt{z+1}\right)J_+^{-1}(z) = C_0^U(z)J_+^{-1}(z). \end{aligned}$$

$$\begin{aligned} C_1^V(z) &= (2z-1)C_0^V(z) + 2\frac{1}{2\pi\mathrm{i}} = \frac{\mathrm{i}}{2\pi}\left(-(z+1) + z\frac{\sqrt{z+1}}{\sqrt{z-1}} + \sqrt{z+1}\sqrt{z-1} - z\right) \\ &= \frac{\mathrm{i}}{2\pi}\left(-1 + \frac{\sqrt{z+1}}{\sqrt{z-1}}\right)\left(z - \sqrt{z-1}\sqrt{z+1}\right) = C_0^V(z)J_+^{-1}(z). \end{aligned}$$

$$\begin{aligned} C_1^W(z) &= (2z+1)C_0^W(z) + 2\frac{1}{2\pi\mathrm{i}} = \frac{\mathrm{i}}{2\pi}\left((z-1) - z\frac{\sqrt{z-1}}{\sqrt{z+1}} - \sqrt{z+1}\sqrt{z-1} + z\right) \\ &= \frac{\mathrm{i}}{2\pi}\left(1 - \frac{\sqrt{z-1}}{\sqrt{z+1}}\right)\left(z - \sqrt{z-1}\sqrt{z+1}\right) = C_0^W(z)J_+^{-1}(z). \end{aligned}$$

Now,

$$2zJ_+^{-1}(z) - 1 = 2z\left(z - \sqrt{z-1}\sqrt{z+1}\right) - 1 = z^2 - 2z\sqrt{z-1}\sqrt{z+1} + (z^2-1) = J_+^{-1}(z)^2.$$

Using this for the second base case $k = 2$,

$$\begin{aligned} C_2^T(z) &= 2zC_1^T(z) - \sqrt{2}C_0^T(z) = 2\sqrt{2}zC_0^T(z)J_+^{-1}(z) - \sqrt{2}C_0^T(z) \\ &= \sqrt{2}C_0^T(z)\left(2zJ_+^{-1}(z) - 1\right) = \sqrt{2}C_0^T(z)J_+^{-1}(z)^2. \end{aligned}$$

As an inductive step, we let $k \geq 3$ and find that our recurrence gives

$$\begin{aligned} C_k^{\underline{T}}(z) &= 2zC_{k-1}^{\underline{T}}(z) - C_{k-2}^{\underline{T}}(z) = 2z\sqrt{2}C_0^{\underline{T}}(z)J_+^{-1}(z)^{k-1} - \sqrt{2}C_0^{\underline{T}}(z)J_+^{-1}(z)^{k-2} \\ &= \sqrt{2}C_0^{\underline{T}}(z)J_+^{-1}(z)^{k-2}\left(2zJ_+^{-1}(z) - 1\right) = \sqrt{2}C_0^{\underline{T}}(z)J_+^{-1}(z)^k, \end{aligned}$$

completing the proof for Chebyshev-$\underline{T}$. For U, V, and W, see that the theorem is trivially true for $k = 0$, so we can use $k = 0, 1$ as our base cases. Letting $\square$ denote U, V, or W, we let $k \geq 2$ and find from the recurrence that

$$\begin{aligned} C_k^{\square}(z) &= 2zC_{k-1}^{\square}(z) - C_{k-2}^{\square}(z) = 2zC_0^{\square}(z)J_+^{-1}(z)^{k-1} - C_0^{\square}(z)J_+^{-1}(z)^{k-2} \\ &= C_0^{\square}(z)J_+^{-1}(z)^{k-2}\left(2zJ_+^{-1}(z) - 1\right) = C_0^{\square}(z)J_+^{-1}(z)^k, \end{aligned}$$

completing the proof for all cases. $\square$

A.3 Proof of Lemma 6.2.3

Proof. For $j = 0$, the lower left block of $p_0(\mathbf{H}) = \mathbf{I}$ is zero. Since $p_0(\mathbf{A}) = \mathbf{I}$, $\mathbf{C}p_0(\mathbf{A}) + \mathbf{G}_0 = \mathbf{0}$ when $\mathbf{G}_0 = -\mathbf{C}$.

When $j = 1$, by (2.5), the lower left block of $p_1(\mathbf{H})$ is given by $\frac{1}{\check{b}_0}\mathbf{C}$. Applying (2.5) to $p_1(\mathbf{A})$,

$$\begin{aligned} \frac{1}{\check{b}_0}\mathbf{C} &= \frac{1}{\check{b}_0}\mathbf{C}(-\check{a}_0\mathbf{I} + (\check{a}_0 + 1)\mathbf{I}) = \frac{1}{\check{b}_0}\mathbf{C}(\check{b}_0 p_1(\mathbf{A}) - \mathbf{A} + (\check{a}_0 + 1)\mathbf{I}) \\ &= \mathbf{C}p_1(\mathbf{A}) + \frac{1}{\check{b}_0}(\mathbf{G}_0\mathbf{A} + (\check{a}_0 + 1)\mathbf{C}). \end{aligned}$$

Using induction, let $j \geq 2$ and assume that the lemma holds for $p_{j-1}(\mathbf{H})$ and $p_{j-2}(\mathbf{H})$. Then,

$$p_{j-1}(\mathbf{H}) = \begin{pmatrix} p_{j-1}(\mathbf{A}) & \mathbf{0} \\ \mathbf{C}p_{j-1}(\mathbf{A}) + \mathbf{G}_{j-1} & p_{j-1}(\mathbf{B}) \end{pmatrix}, \quad p_{j-2}(\mathbf{H}) = \begin{pmatrix} p_{j-2}(\mathbf{A}) & \mathbf{0} \\ \mathbf{C}p_{j-2}(\mathbf{A}) + \mathbf{G}_{j-2} & p_{j-2}(\mathbf{B}) \end{pmatrix}.$$

Applying (2.5) and noting that matrices commute with polynomials of themselves, the lower left block of $p_j(\mathbf{H})$ is given by

$$\begin{aligned} &\frac{1}{\check{b}_{j-1}}\left(\mathbf{C}p_{j-1}(\mathbf{A})\mathbf{A} + \mathbf{G}_{j-1}\mathbf{A} + p_{j-1}(\mathbf{B})\mathbf{C} - \check{a}_{j-1}\mathbf{C}p_{j-1}(\mathbf{A})\right. \\ &\qquad \left. -\check{a}_{j-1}\mathbf{G}_{j-1} - \check{b}_{j-2}\mathbf{C}p_{j-2}(\mathbf{A}) - \check{b}_{j-2}\mathbf{G}_{j-2}\right) \\ &= \mathbf{C}p_j(\mathbf{A}) + \frac{1}{\check{b}_{j-1}}\left(\mathbf{G}_{j-1}\mathbf{A} + p_{j-1}(\mathbf{B})\mathbf{C} - \check{a}_{j-1}\mathbf{G}_{j-1} - \check{b}_{j-2}\mathbf{G}_{j-2}\right), \end{aligned}$$

which completes the proof. $\square$

A.4 Proof of Theorem 6.7.2

Proof. The error corresponds to the tail of the bivariate series (6.33),

$$\mathbf{X} - \mathbf{X}_k = \sum_{\max\{i,j\}\geq k} c_{ij}\, p_i^{\mathbf{B}}(\mathbf{B})\mathbf{U}\mathbf{V}p_j^{\mathbf{A}}(\mathbf{A}).$$

We write

$$\mathbf{X} - \mathbf{X}_k = \sum_{i\geq k} p_i^{\mathbf{B}}(\mathbf{B})\,\mathbf{U}\mathbf{V}\,b_i(\mathbf{A}) + \sum_{j\geq k} a_j(\mathbf{B})\,\mathbf{U}\mathbf{V}\,p_j^{\mathbf{A}}(\mathbf{A}) - \sum_{i,j\geq k} c_{ij}\, p_i^{\mathbf{B}}(\mathbf{B})\mathbf{U}\mathbf{V}p_j^{\mathbf{A}}(\mathbf{A}), \quad \text{(A.3)}$$

where the one-variable coefficient functions

$$a_j(x) = \int_{\Sigma_{\mathbf{A}}} f(x,y)\, p_j^{\mathbf{A}}(y)\, w_{\mathbf{A}}(y)\,\mathrm{d}y, \qquad b_i(y) = \int_{\Sigma_{\mathbf{B}}} f(x,y)\, p_i^{\mathbf{B}}(x)\, w_{\mathbf{B}}(x)\,\mathrm{d}x$$

have full series $a_j(\mathbf{B}) = \sum_i c_{ij}\, p_i^{\mathbf{B}}(\mathbf{B})$ and $b_i(\mathbf{A}) = \sum_j c_{ij}\, p_j^{\mathbf{A}}(\mathbf{A})$.

By Lemma 5.3.6,

$$\max_{y\in\Sigma_{\mathbf{A}}} |b_i(y)| \leq C\varrho_{\mathbf{B}}^{-i}, \qquad \max_{x\in\Sigma_{\mathbf{B}}} |a_j(x)| \leq C\varrho_{\mathbf{A}}^{-j}.$$

Diagonalizing $\mathbf{A}$ and $\mathbf{B}$ and denoting $\kappa = \|\mathbf{V}_{\mathbf{A}}\|_2\|\mathbf{V}_{\mathbf{A}}^{-1}\|_2\|\mathbf{V}_{\mathbf{B}}\|_2\|\mathbf{V}_{\mathbf{B}}^{-1}\|_2$, by Lemma 5.2.2 the first term in (A.3) is bounded by

$$\left\|\sum_{i\geq k} p_i^{\mathbf{B}}(\mathbf{B})\,\mathbf{U}\mathbf{V}\,b_i(\mathbf{A})\right\|_2 \leq C\kappa\,\|\mathbf{U}\mathbf{V}\|_2 \sum_{i\geq k}\varrho_{\mathbf{B}}^{-i} \leq C_1\,\frac{\varrho_{\mathbf{B}}^{-k}}{1-\varrho_{\mathbf{B}}^{-1}},$$

for some constants $C, C_1 > 0$. The second term is bounded by $C_2\,\varrho_{\mathbf{A}}^{-k}/(1-\varrho_{\mathbf{A}}^{-1})$ for some $C_2 > 0$ in the same manner. For the final term, applying Lemma 5.3.6 to the two representations $c_{ij} = \langle b_i, p_j^{\mathbf{A}}\rangle_{L^2_{w_{\mathbf{A}}}} = \langle a_j, p_i^{\mathbf{B}}\rangle_{L^2_{w_{\mathbf{B}}}}$ gives the two estimates

$$|c_{ij}| \leq C\varrho_{\mathbf{B}}^{-i} \qquad \text{and} \qquad |c_{ij}| \leq C\varrho_{\mathbf{A}}^{-j}.$$

Multiplying these two estimates and taking the square root yields

$$|c_{ij}| \leq C\varrho_{\mathbf{B}}^{-i/2}\varrho_{\mathbf{A}}^{-j/2},$$

so that

$$\left\|\sum_{i,j\geq k} c_{ij}\, p_i^{\mathbf{B}}(\mathbf{B})\mathbf{U}\mathbf{V}p_j^{\mathbf{A}}(\mathbf{A})\right\|_2 \leq C\kappa\,\|\mathbf{U}\mathbf{V}\|_2 \sum_{i,j\geq k}\varrho_{\mathbf{B}}^{-i/2}\varrho_{\mathbf{A}}^{-j/2} = \mathrm{O}\big(\varrho^{-k}\big).$$

Summing the three contributions gives the desired result. □

Appendix B

COMPUTING THE MODULATION PARAMETER AND REGIONS

To determine the bounds $\xi_{j,1}, \xi_{j,2}$ of the jth modulation region and the value of the modulation parameter $\alpha_j(\xi)$ for $\xi \in [\xi_{j,1}, \xi_{j,2}]$, we construct functions $F_j(\xi, \alpha)$ with zeroes corresponding to critical values. Throughout this appendix, for a given $j \in \{1, \dots, g\}$ and $\alpha \in (a_j, b_j)$, denote

$$\check{\Sigma}_j(\alpha) = \begin{cases} [\alpha^2, b_j^2] \cup \bigcup_{k=j+1}^{g} [a_k^2, b_k^2], & j < g, \\ [\alpha^2, b_g^2], & j = g, \end{cases}$$

and

$$\check{\Gamma}_{j,k}(\alpha) = \begin{cases} [0, \alpha^2], & k = 0, \\ [b_{k+j-1}^2, a_{k+j}^2], & k = 1, \dots, g-j. \end{cases}$$

Consider the following scalar RHP:

Riemann–Hilbert Problem B.0.1. *Find a function $\check{\mathfrak{g}}$ with the following properties:*

Analyticity: *$\check{\mathfrak{g}}(\zeta)$ is analytic for $\zeta \in \mathbb{C} \setminus (-\infty, b_g^2]$, admitting continuous boundary values on $(-\infty, b_g^2)$.*

Jump conditions: *The boundary values $\check{\mathfrak{g}}_\pm(\zeta) = \lim_{\epsilon \to 0^+} \check{\mathfrak{g}}_\pm(\zeta \pm \mathrm{i}\epsilon)$ are related via*

$$\begin{aligned} &\check{\mathfrak{g}}_+(\zeta) + \check{\mathfrak{g}}_-(\zeta) = 2\left(\xi\zeta^{1/2} - \zeta^{3/2}\right), && \zeta \in \check{\Sigma}_j(\alpha), \\ &\check{\mathfrak{g}}_+(\zeta) - \check{\mathfrak{g}}_-(\zeta) = \mathrm{i}\Omega_k(\xi, \alpha), && \zeta \in \check{\Gamma}_{j,k}(\alpha), \quad k = 0, \dots, g-j, \\ &\check{\mathfrak{g}}_+(\zeta) + \check{\mathfrak{g}}_-(\zeta) = 0, && \zeta \in (-\infty, 0), \end{aligned}$$

where $\Omega_k(\xi, \alpha)$ are real-valued constants.

Asymptotic condition (infinity):

$$\check{\mathfrak{g}}(\zeta) = \mathrm{O}(\zeta^{-1/2}), \quad \zeta \to \infty.$$

Asymptotic condition (left endpoint):

$$\check{\mathfrak{g}}(\zeta) - \left(\xi\zeta^{1/2} - \zeta^{3/2}\right) = \mathrm{O}(\zeta - \alpha^2)^{3/2}, \quad \zeta \to \alpha^2. \tag{B.1}$$

Asymptotic condition (zero):

$$\check{\mathfrak{g}}(\zeta) = \mathrm{O}\left(\zeta^{1/2}\right), \quad \zeta \to 0.$$

Asymptotic condition (other endpoints): *For each fixed endpoint $c \in \{a_{j+1}, \dots, a_g, b_j, \dots, b_g\}$,*

$$\check{\mathfrak{g}}(\zeta) - \left(\xi\zeta^{1/2} - \zeta^{3/2}\right) = \mathrm{O}(\zeta - c^2)^{1/2}, \quad \zeta \to c^2.$$

We construct F_j so that its zeroes correspond to points where Riemann–Hilbert Problem B.0.1 is solvable. We do this by first constructing the derivative $\check{\mathfrak{g}}'$ of a candidate solution $\check{\mathfrak{g}}$, i.e., we solve the following RHP:

Riemann–Hilbert Problem B.0.2. *Find a function $\check{\mathfrak{g}}'$ with the following properties:*

Analyticity: *$\check{\mathfrak{g}}'(\zeta)$ is analytic for $\zeta \in \mathbb{C}\backslash\big((-\infty,0] \cup \operatorname{cl}\big(\check{\Sigma}_j(\alpha)\big)\big)$, admitting continuous boundary values on $(-\infty,0) \cup \check{\Sigma}_j(\alpha)$.*

Jump conditions: *The boundary values $\check{\mathfrak{g}}'_{\pm}(\zeta) = \lim_{\epsilon\to 0^+} \check{\mathfrak{g}}'_{\pm}(\zeta \pm \mathrm{i}\epsilon)$ are related via*

$$\begin{aligned} &\check{\mathfrak{g}}'_+(\zeta) + \check{\mathfrak{g}}'_-(\zeta) = \frac{\xi}{\sqrt{\zeta}} - 3\sqrt{\zeta}, && \zeta \in \check{\Sigma}_j(\alpha), \\ &\check{\mathfrak{g}}'_+(\zeta) + \check{\mathfrak{g}}'_-(\zeta) = 0, && \zeta \in (-\infty,0). \end{aligned}$$

Asymptotic condition (infinity):

$$\check{\mathfrak{g}}'(\zeta) = \mathrm{O}(\zeta^{-3/2}), \quad \zeta \to \infty.$$

Asymptotic condition (left endpoint):

$$\check{\mathfrak{g}}'(\zeta) - \left(\frac{\xi}{2\sqrt{\zeta}} - \frac{3}{2}\sqrt{\zeta}\right) = \mathrm{O}(\zeta - \alpha^2)^{1/2}, \quad \zeta \to \alpha^2.$$

Asymptotic condition (zero):

$$\check{\mathfrak{g}}'(\zeta) = \mathrm{O}\left(\zeta^{-1/2}\right), \quad \zeta \to 0.$$

Asymptotic condition (other endpoints): *For each fixed endpoint $c \in \{a_{j+1}, \dots, a_g, b_j, \dots, b_g\}$,*

$$\check{\mathfrak{g}}'(\zeta) - \left(\frac{\xi}{2\sqrt{\zeta}} - \frac{3}{2}\sqrt{\zeta}\right) = \mathrm{O}(\zeta - c^2)^{-1/2}, \quad \zeta \to c^2.$$

Let $\tilde{R}_j(\zeta;\alpha) = \tilde{R}_j(\zeta)$ be the function satisfying

$$\tilde{R}_j(\zeta)^2 = \begin{cases} \frac{(\zeta-\alpha^2)\prod_{k=j+1}^{g}(\zeta-a_k^2)}{\zeta\prod_{k=j}^{g}(\zeta-b_k^2)}, & j < g, \\ \frac{\zeta-\alpha^2}{\zeta(\zeta-b_g^2)}, & j = g, \end{cases}$$

along with the condition $\tilde{R}_j(\zeta) = \zeta^{-1/2} + \mathrm{O}\left(\zeta^{-3/2}\right)$ as $\zeta \to \infty$. Then, a solution to Riemann–Hilbert Problem B.0.2 is

$$\check{\mathfrak{g}}'(\zeta) = \frac{\tilde{R}_j(\zeta)}{2\pi \mathrm{i}} \int_{\tilde{\Sigma}_j(\alpha)} \frac{\frac{\xi}{\sqrt{s}} - 3\sqrt{s}}{\tilde{R}_j^+(s)} \frac{\mathrm{d}s}{s-\zeta}. \tag{B.2}$$

Lemma B.0.3. *If $j = g$,* (B.2) *is the unique solution of Riemann–Hilbert Problem B.0.2.*

Proof. By construction, (B.2) is a solution to Riemann–Hilbert Problem B.0.2. To show uniqueness, let $\check{\mathfrak{g}}_1', \check{\mathfrak{g}}_2'$ each be solutions of Riemann–Hilbert Problem B.0.2 and define

$$H(\zeta) = \frac{\check{\mathfrak{g}}_1'(\zeta) - \check{\mathfrak{g}}_2'(\zeta)}{\tilde{R}_j(\zeta)}.$$

Then, H is analytic and $H(\zeta) = \mathrm{O}(\zeta^{-1})$ as $\zeta \to \infty$. Thus, H is identically zero and $\check{\mathfrak{g}}_1' = \check{\mathfrak{g}}_2'$. □

If $j = g$, it follows from a residue calculation that

$$\check{\mathfrak{g}}_+'(\zeta) + \check{\mathfrak{g}}_-'(\zeta) - \left(\frac{\xi}{\sqrt{\zeta}} - 3\sqrt{\zeta}\right) = -\tilde{R}_j(\zeta)\left(\xi - 3\zeta + \frac{3}{2}\left(b_g^2 - \alpha^2\right)\right).$$

Now, consider the function

$$\check{\mathfrak{g}}(\zeta) = \int_0^\zeta \check{\mathfrak{g}}'(\zeta')\mathrm{d}\zeta', \tag{B.3}$$

where the integration path is taken to be a straight line between 0 and ζ. By construction, $\check{\mathfrak{g}}$ solves Riemann–Hilbert Problem B.0.1, if

$$\int_0^{\alpha^2} \left(\check{\mathfrak{g}}_+'(\zeta) + \check{\mathfrak{g}}_-'(\zeta) - \left(\frac{\xi}{\sqrt{\zeta}} - 3\sqrt{\zeta}\right)\right) \mathrm{d}\zeta = 0. \tag{B.4}$$

Thus, we define

$$F_g(\xi, \alpha) = -\int_0^{\alpha^2} \tilde{R}_j(\zeta)\left(\xi - 3\zeta + \frac{3}{2}\left(b_g^2 - \alpha^2\right)\right) \mathrm{d}\zeta.$$

If $j < g$, Riemann–Hilbert Problem B.0.2 no longer has a unique solution. To amend this, let $\check{R}_j(\zeta;\alpha) = \check{R}_j(\zeta)$ be the function satisfying

$$\check{R}_j(\zeta)^2 = \frac{\zeta - \alpha^2}{\zeta(\zeta - b_j^2)\prod_{k=j+1}^g (\zeta - a_k^2)(\zeta - b_k^2)},$$

along with the condition $\check{R}_j(\zeta) = \zeta^{-1/2+j-g} + \mathrm{O}\left(\zeta^{-3/2+j-g}\right)$ as $\zeta \to \infty$.

Lemma B.0.4. *If $j < g$, all solutions to Riemann–Hilbert Problem B.0.2 can be expressed*

in the form

$$\check{\mathfrak{g}}'(\zeta) = \frac{\tilde{R}_j(\zeta)}{2\pi \mathrm{i}} \int_{\check{\Sigma}_j(\alpha)} \frac{\frac{\xi}{\sqrt{s}} - 3\sqrt{s}}{\tilde{R}_j^+(s)} \frac{\mathrm{d}s}{s-\zeta} + P_{g-j-1}(\zeta)\check{R}_j(\zeta),$$

where P_{g-j-1} is a polynomial of degree at most $g-j-1$.

Proof. By construction,

$$\check{\mathfrak{g}}_1'(\zeta) = \frac{\tilde{R}_j(\zeta)}{2\pi \mathrm{i}} \int_{\check{\Sigma}_j(\alpha)} \frac{\frac{\xi}{\sqrt{s}} - 3\sqrt{s}}{\tilde{R}_j^+(s)} \frac{\mathrm{d}s}{s-\zeta}, \tag{B.5}$$

is a solution to Riemann–Hilbert Problem B.0.2. Let $\check{\mathfrak{g}}_2'$ be another solution to Riemann–Hilbert Problem B.0.2 and define

$$H(\zeta) = \frac{\check{\mathfrak{g}}_1'(\zeta) - \check{\mathfrak{g}}_2'(\zeta)}{\check{R}_j(\zeta)}.$$

Then, H is analytic and $H(\zeta) = \mathrm{O}(\zeta^{g-j-1})$ as $\zeta \to \infty$, so H is a polynomial of degree at most $g-j-1$. Thus, any solution of Riemann–Hilbert Problem B.0.2 must be of the form (B.5). □

If $j < g$, a residue calculation yields

$$\begin{aligned}\check{\mathfrak{g}}_+'(\zeta) + \check{\mathfrak{g}}_-'(\zeta) - \left(\frac{\xi}{\sqrt{\zeta}} - 3\sqrt{\zeta}\right) \\ = -\tilde{R}_j(\zeta)\left(\xi - 3\zeta + \frac{3}{2}\left(b_j^2 - \alpha^2\right) + \frac{3}{2}\sum_{k=j+1}^{g}\left(b_k^2 - a_k^2\right)\right) + P_{g-j-1}(\zeta)\check{R}_j(\zeta).\end{aligned}$$

Defined as in (B.3), $\check{\mathfrak{g}}$ solves Riemann–Hilbert Problem B.0.1 if $\check{\mathfrak{g}}'$ satisfies (B.4) and

$$\int_{\check{\Gamma}_{j,k}(\alpha)} \left(\check{\mathfrak{g}}_+'(\zeta) + \check{\mathfrak{g}}_-'(\zeta) - \left(\frac{\xi}{\sqrt{\zeta}} - 3\sqrt{\zeta}\right)\right) \mathrm{d}\zeta = 0, \tag{B.6}$$

for $k = 1, \ldots, g-j$. We determine $P_{g-j-1}(\zeta)$ by enforcing (B.6). That is, the coefficients of $P_{g-j-1}(\zeta) = \gamma_1 + \ldots + \gamma_{g-j}\zeta^{g-j-1}$ are determined by the linear system

$$\sum_{\ell=1}^{g-j} \gamma_\ell \int_{\check{\Gamma}_{j,k}(\alpha)} \zeta^{\ell-1}\check{R}_j(\zeta)\mathrm{d}\zeta = \int_{\check{\Gamma}_{j,k}(\alpha)} \tilde{R}_j(\zeta)\left(\xi - 3\zeta + \frac{3}{2}\left(b_j^2 - \alpha^2\right) + \frac{3}{2}\sum_{k=j+1}^{g}\left(b_k^2 - a_k^2\right)\right)\mathrm{d}\zeta, \tag{B.7}$$

for $k = 1, \ldots, g-j$.

Lemma B.0.5. *The $(g-j)\times(g-j)$ matrix defined by*

$$\check{\mathbf{G}} = \left(\int_{\check{\Gamma}_{j,k}(\alpha)} \zeta^{\ell-1} \check{R}_j(\zeta) \mathrm{d}\zeta \right)_{1\le k,\ell\le g-j}$$

is invertible.

Proof. The proof follows from the argument of Lemma 4.2.2 applied to $\check{\mathbf{G}}$. □

As before, this matrix becomes ill-conditioned when $g-j$ is large due to the presence of the monomial basis. We instead solve an equivalent linear system with basis polynomials chosen to vanish at the midpoints of the gaps.

Once P_{g-j-1} has been determined in this manner, $\check{\mathfrak{g}}$ solves Riemann–Hilbert Problem B.0.1 if

$$0 = F_j(\xi,\alpha) := \int_0^{\alpha^2} \Big(P_{g-j-1}(\zeta)\check{R}_j(\zeta) \\ - \tilde{R}_j(\zeta)\Big(\xi - 3\zeta + \frac{3}{2}\left(b_j^2-\alpha^2\right) + \frac{3}{2}\sum_{k=j+1}^{g}\left(b_k^2-a_k^2\right)\Big)\Big)\mathrm{d}\zeta.$$

Remark B.0.6. *For any given $\xi\in\mathbb{R}$, $\alpha>0$, $F_j(\xi,\alpha)$ can be computed via appropriate Gauss–Chebyshev quadrature rules. Furthermore, the limiting value*

$$F_j(\xi,b_j) := \lim_{\alpha\to b_j} F_j(\xi,\alpha),$$

evaluates to $2b_g^3 - 2\xi b_g$ if $j=g$ and can be computed via appropriate Gauss–Jacobi quadrature rules if $j<g$.

For any fixed $\alpha>0$, $F_j(\xi,\alpha)$ depends linearly on ξ, so the boundaries of the modulated regions are determined by

$$F_j\left(\xi_{j,1},a_j\right)=0, \quad F_j\left(\xi_{j,2},b_j\right)=0, \quad j=1,\dots,g,$$

where the roots $\xi_{j,1},\xi_{j,2}$ can be computed by the secant method in a single step. Given $\xi\in(\xi_{j,1},\xi_{j,2})$ in the jth modulated region, $\alpha_j(\xi)$ can then be determined by applying a scalar root finding algorithm to solve

$$F_j\left(\xi,\alpha_j\right)=0, \quad a_j<\alpha_j<b_j.$$

Remark B.0.7. *The function $\check{\mathfrak{g}}$ constructed above is used only to compute the modulation parameter α_j and the critical values $\xi_{j,1}$, $\xi_{j,2}$. It differs from the g-function $\hat{\mathfrak{g}}$ employed in the steepest descent deformations of Chapter 8. The latter is preferred for the deformations*

because they remain valid even when α_j is computed inexactly; in this case the final jump conditions exhibit exponential growth rather than exponential decay as $x, t \to \infty$ with fixed ξ.