Lanczos-based methods for matrix functions

Tyler Chen

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

Doctor of Philosophy

University of Washington
2022

Reading Committee:
Anne Greenbaum, Chair
Thomas Trogdon, Chair
Aleksandr Aravkin

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

Lanczos-based methods for matrix functions

Tyler Chen

Chairs of the Supervisory Committee:
Anne Greenbaum
Thomas Trogdon

Department of Applied Mathematics

We study Lanczos-based methods for tasks involving matrix functions. We begin by resurfacing a range of ideas regarding matrix-free quadrature which, to the best of our knowledge, have not been treated simultaneously. This enables the development of a unified perspective from which a number of commonly used randomized methods for spectrum and spectral sum approximation can be understood. We proceed to develop optimal Krylov subspace methods for approximating the product of a rational matrix function with a fixed vector. Finally, we show how the optimality of such methods can be used to obtain fine-grained spectrum dependent bounds for standard Lanczos-based methods for approximating a wide class of matrix functions applied to a vector.

Lanczos-based methods for matrix functions

TYLER CHEN

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

Doctor of Philosophy

2022
University of Washington
Department of Applied Mathematics

本文献给我的奶奶爷爷

Preface

At night, my apartment looks out to a thousand illuminated windows. I'm drawn such views because they make me feel an isolating sense of closeness; behind each window is a person—a family enjoying dinner, a student working on their homework, a cleaning person ending their shift. I do not know them and they do not know me, yet we are all connected in this moment of existence. This is *sonder*, a concept for which we find a definition in the Dictionary of Obscure Sorrows:

> **sonder**
>
> *n.* the realization that each random passerby is living a life as vivid and complex as your own

Sonder, even with the accompanying melancholy, has been the single most consistent force driving my success throughout my tertiary education. It only fitting, then, that it receives mention in my dissertation, the symbolic culmination of my formal education.

Chinatown/International District
Seattle, Washington

Acknowledgements

First and foremost, I would like to thank my thesis advisors, Anne Greenbaum and Thomas Trogdon, whose support and guidance throughout my PhD have shaped me into the researcher I am today. Many PhD students struggle to find a good advisor, but I had two excellent ones. Anne, thank you for sharing with me your vast wisdom on numerical linear algebra and for always making time for me when I stopped by your office unannounced. Tom, thank you for pushing me to produce work of the highest quality and for answering my many, perhaps repetitive, career advice questions. I feel truly fortunate to have crossed path's with each of you, and I look forward to ongoing collaboration in the future.

Many others have contributed directly to my academic development. I'm particularly grateful to my undergraduate advisor, Roger Tobin, for his mentorship throughout my time at Tufts. This was a formative time in my life during which much of my current perspective on academia was developed. I'm also thankful for the collaborators I've had over the past several years, Raghu Bollapragada, Erin Carson, Yu-Chen Cheng, Eric Hallman, Hexuan Liu, Cameron Musco, Christopher Musco, Shashanka Ubaru, Rachel Ward, Natalie Wellen, and Qichen Xu, each of who contributed, in one way or another, to the content of this thesis. I especially want to thank Yu-Chen for teaching me much of what I know about probability during the first year of my PhD. Finally, thanks to my thesis committee, Sasha Aravkin and Maryam Fazel, for contributing their time and energy to my thesis and defense.

I'm indebted to many folks at UW who supported me throughout my time here. This includes everyone at Hall Health, upstairs and downstairs, the staff and faculty in the Applied Math department, and the many students who I regularly interacted with. When I first vested the department as a prospective student, it was immediately clear that there was a wonderful community of students. This was the deciding factor in my decision to come to UW, and I have been continually grateful for this community throughout my time in the program. I was also extremely blessed to have a wonderful cohort without whom the last five years would have been far less enjoyable.

Finally, I want to acknowledge my family and friends. This thesis is dedicated to my grandparents, 陈祖浩 and 鲁月娥, the greatest supporters of my education.

To my parents, Ming-Guang (陈明光) and Peggy, thank you for providing for me since before I was even born. This achievement is as much yours as it is mine. To my partner, 黃羿綺, I'm so happy our lives intersected when they did. Thank you for the constant love and support, and for quarantining with me during the pandemic. I cannot imagine what the last few years would have looked like without you. Last, a shout out to Jesse, Matthew, Max, Tailong, and Vaibhav, my "internet friends", who have each been with me through many stages of my life, despite large geographic distances.

This material is based on work supported by the National Science Foundation under grant number DGE-1762114. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Contents

1 Introduction **1**

1.1 Lanczos-based methods . . . 3

1.1.1 The insufficiency of interval-based bounds . . . 5

1.1.2 The effect of finite precision arithmetic . . . 6

1.1.3 A motivating example . . . 7

1.2 Context and contributions . . . 9

2 Scalar polynomials **12**

2.1 Basic definitions . . . 12

2.2 Orthogonal polynomials . . . 15

2.2.1 Chebyshev polynomials . . . 17

2.3 Polynomial approximations and bounds . . . 18

2.A Proof of Jackson's theorem . . . 21

3 Matrix-free quadrature **28**

3.1 Extracting moments from a Krylov subspace . . . 29

3.1.1 Computing modified moments directly . . . 30

3.1.2 Connection coefficients to compute more modified moments 31

3.2 Quadrature approximations for weighted spectral measures . . . 34

3.2.1 Quadrature by interpolation . . . 35

3.2.2 Gaussian quadrature . . . 36

3.2.3 Quadrature by approximation . . . 38

3.2.4 Positivity by damping and the kernel polynomial method . 40

3.3 A priori error bounds on an interval . . . 41

3.4 Qualitative comparison of algorithms . . . 43

3.5 Numerical experiments 44
3.5.1 Comparison with classical quadrature 44
3.5.2 Finite precision convergence 47

4 Spectrum and spectral sum approximation 49

4.1 Related work and context 51
4.1.1 Note on history of stochastic quadratic trace estimators and their analysis 53
4.1.2 Other randomized trace estimation algorithms 54
4.2 Analysis 55
4.2.1 Uniform unit test vectors 55
4.3 Numerical experiments 59
4.3.1 Approximating sparse spectra 59
4.3.2 Approximating "smooth" densities 60
4.3.3 Energy spectra of small spin systems 67

5 Optimal rational matrix function approximation 70

5.1 A bit of notation 71
5.2 Existing algorithms 72
5.3 Optimal rational function approximation 73
5.3.1 Relation of Lanczos-OR to QMR on inverse quadratics . . . 77
5.4 Error estimates for Lanczos-OR 77
5.4.1 Numerical experiment 79
5.5 Implementing Lanczos-OR using low memory 79
5.5.1 Computing LDL factorization 81
5.5.2 Inverting the LDL factorization 82
5.5.3 Computing polynomials in $\mathbf{T}$ 86
5.5.4 Putting it all together 88
5.5.5 Some comments on implementation 88

6 General matrix function approximation 91

6.1 Explicit polynomial methods 91
6.2 Lanczos-FA 92
6.2.1 A priori error bounds on an interval 93
6.2.2 Two-pass Lanczos-FA 93
6.3 Lanczos-OR based methods 94
6.3.1 The matrix sign function 94
6.3.2 Rational function approximation 96
6.4 Numerical experiments 98
6.4.1 The matrix sign function 98
6.4.2 Rational matrix functions 100
6.4.3 Lanczos-FA vs Lanczos-OR vs CG 103

7 Spectrum dependent bounds and a posteriori error estimates 106

7.1 An integral representation of the Lanczos-FA error 106
7.1.1 A reduction to linear system error 109
7.1.2 Comparison with previous work 111
7.2 Applying our framework 113
7.2.1 A priori bounds 114
7.2.2 A posteriori error bounds 117
7.2.3 Numerical computation of integrals 117
7.3 Examples and numerical verification 119
7.3.1 Choice of contour 119
7.3.2 Piecewise analytic functions 122
7.3.3 Quadratic forms 123
7.4 Error bounds for Lanczos-FA on indefinite systems 125

8 Finite precision arithmetic 130

8.1 Preliminaries 131
8.2 Three term recurrences 132
8.2.1 The Lanczos algorithm 133
8.3 Lanczos-FA 135

8.4 Gaussian quadrature 136
8.5 Backwards stability of the Lanczos algorithm 138
8.5.1 A new approach 138
8.6 CIF bounds Finite precision 140
8.6.1 Numerical experiment 142

9 Outlook **143**
9.1 Randomization 143
9.2 Typicality 144
9.3 Accessibility to non-experts 145

10 Notation and other reference sheets **146**
10.1 Basic notation 146
10.2 Indexing for matrices 148
10.3 The model problem 149
10.4 Some basic properties 150
10.5 List of algorithms 152
10.6 List of figures 153

11 Bibliography **155**

Chapter 1
Introduction

Computational approaches to today's most pressing and world-changing questions are reliant on subroutines for fundamental linear algebraic tasks. The focus of this thesis is on the design and analysis of algorithms for an increasingly prevalent subset of such tasks: those involving matrix functions of Hermitian (or real symmetric) matrices. For the duration of this thesis, $\mathbf{A}$ will be a $n \times n$ Hermitian matrix with eigenvalues $\Lambda := \{\lambda_i\}_{i=0}^{n-1}$ and (orthonormal) eigenvectors $\{\mathbf{u}_i\}_{i=0}^{n-1}$; i.e.,

$$\mathbf{A} = \sum_{i=0}^{n-1} \lambda_i \mathbf{u}_i \mathbf{u}_i^{\mathsf{H}}. \tag{1.1}$$

A matrix function transforms the eigenvalues of a Hermitian (or symmetric) matrix according to some scalar function, while leaving the eigenvectors untouched.

Definition 1.1. *The matrix function $f(\mathbf{A})$, induced by $f : \mathbb{R} \to \mathbb{R}$ and $\mathbf{A}$, is defined as*

$$f(\mathbf{A}) := \sum_{i=0}^{n-1} f(\lambda_i) \mathbf{u}_i \mathbf{u}_i^{\mathsf{H}}.$$

Perhaps the most well known example of a matrix function is the matrix inverse $\mathbf{A}^{-1}$, which corresponds to the inverse function $f(x) = x^{-1}$. Other common matrix functions including the matrix sign, logarithm, exponential, square root, and inverse square root, each of which has many applications throughout the mathematical sciences.

A reference sheet containing common notation and useful factscan be found in Chapter 10.

A common task involving matrix functions is computing the product $f(\mathbf{A})\mathbf{v}$ of a matrix function $f(\mathbf{A})$ with a fixed vector $\mathbf{v}$; for instance, the matrix inverse applied to a vector corresponds to the solution of a linear system of equations. Beyond the multitude of applications of linear systems, matrix functions applied to vectors are used for computing the overlap operator in quantum chromodynamics [Esh+02], solving differential equations in applied math [Saa92; HL97], Gaussian process sampling in statistics [Ple+20], principle component projection and regression in data science [JS19], and a range of other applications [Hig08].

Another related and especially interesting task involving matrix functions is estimating the *spectral sum*,

$$\operatorname{tr}(f(\mathbf{A})) = \sum_{i=0}^{n-1} f(\lambda_i). \tag{1.2}$$

Applications of spectral sums include characterizing the degree of protein folding in biology [Est00], studying the thermodynamics of spin systems in quantum physics and chemistry [Wei+06; SS10; SRS20; Jin+21], benchmarking quantum devices in quantum information theory [Joz94], maximum likelihood estimation in statistics [BP99; PL04], designing better public transit in urban planning [BS22; Wan+21], and finding triangle counts and other structure in network science [Avr10; DBB19; BB20].

The trace of matrix functions is intimately related to the spectral measure of $\mathbf{A}$ which encodes the eigenvalues of $\mathbf{A}$.

Definition 1.2. *The cumulative empirical spectral measure (CESM)* $\Phi : \mathbb{R} \to [0, 1]$, *induced by* $\mathbf{A}$, *is defined by*

$$\Phi(x) = \Phi_{\mathbf{A}} := \sum_{i=0}^{n-1} n^{-1} \mathbb{1}[\lambda_i \leq x].$$

Here $\mathbb{1}[\texttt{true}] = 1$ and $\mathbb{1}[\texttt{false}] = 0$.

Not only is $\Phi(x)$ itself a spectral sum for each $x \in \mathbb{R}$, but

$$\operatorname{tr}(f(\mathbf{A})) = n \int f \, \mathrm{d}\Phi.$$

In this sense, approximating the CESM Φ is equivalent to approximating spectral sums. However, approximations to Φ are also useful in that they provide a

global picture of the spectrum of $\mathbf{A}$. Such coarse grained approximations are used in electronic structure computations and other tasks in physics[1] [Wei+06; Jin+21], probing the behavior of neural networks in machine learning [GKX19; Pap19; GWG19; Yao+20], load balancing modern parallel eigensolvers in numerical linear algebra [Pol09; Li+19], and computing the product of matrix functions with vectors [Fan+19].

The simplest, and arguably most elegant, approach to spectrum and spectral sum approximation involves computing quadratic forms $\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ for suitably chosen random vectors $\mathbf{v}$. For any fixed $\mathbf{v}$, the task of computing $\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ is intimately related to quadrature [GM94; GM09] and, besides the many applications of spectrum and spectral sum approximation, is used for estimating the error of Krylov subspace methods [DEG72; GS94; GM09].

1.1 Lanczos-based methods

The algorithms we study in this thesis fall into a general class of algorithms called Krylov subspace methods (KSMs). KSMs produce approximations using information from the set of low-degree polynomials in $\mathbf{A}$ applied to a vector $\mathbf{v}$; i.e. from the so-called Krylov subspace generated by $\mathbf{A}$ and $\mathbf{v}$.

Definition 1.3. *The dimension k Krylov subspace $\mathcal{K}_k$ generated by $\mathbf{A}$ and $\mathbf{v}$ is defined as*

$$\mathcal{K}_k = \mathcal{K}_k(\mathbf{A}, \mathbf{v}) := \operatorname{span}\{\mathbf{v}, \mathbf{A}\mathbf{v}, \dots, \mathbf{A}^{k-1}\mathbf{v}\} = \{p(\mathbf{A})\mathbf{v} : \deg(p) < k\}.$$

The information from a given Krylov subspace can be used to approximate $f(\mathbf{A})\mathbf{v}$ and $\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$. In particular, a natural approach is to use the approximations

$$f(\mathbf{A})\mathbf{v} \approx [f]_s^{\circ\mathrm{p}}(\mathbf{A})\mathbf{v}, \qquad \mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v} \approx \mathbf{v}^{\mathsf{H}} [f]_s^{\circ\mathrm{p}}(\mathbf{A})\mathbf{v},$$

where $[f]_s^{\circ\mathrm{p}} : \mathbb{R} \to \mathbb{R}$ is a degree s polynomial chosen to approximate f.

Throughout this thesis, the symbol "$\circ$" should be interpreted as a parameter encompassing any other parameters which impact how $[f]_s^{\circ\mathrm{p}}(\mathbf{A})$ is determined for f. For instance, once choice of $\circ$ may correspond to the interpolating polynomial to f at some set of nodes while another choice of $\circ$ may correspond

[1] In physics, the "density" $\mathrm{d}\Phi/\mathrm{d}x$ is often called the density of states (DOS).

to the Chebyshev approximation to f. Specific choices of $\circ$ corresponding to widely used algorithms will be defined as they come up. Here the use of "p" stands for polynomial, and will be used to differentiate between polynomial approximations of a function and quadrature approximations of a distribution function, which will be defined later.

Remark 1.4. When $\mathbf{A}$ is Hermitian, Krylov subspace methods are, in one way or another, related to the Lanczos algorithm [Lan50] described in Algorithm 1.1. Even so, we use the term *Lanczos-based methods* to refer to algorithms which make use of the information generated by the Lanczos algorithm in some non-trivial way. This is in contrast to methods, such as those based on explicit polynomial approximation, which can easily be constructed directly. △

Assumption 1.5. *From this point onwards, we will assume* $\|\mathbf{v}\|_2 = 1$.

The Lanczos algorithm (Algorithm 1.1) [Lan50] produces an orthonormal basis $\{\mathbf{q}_i\}_{i=0}^{k}$ for the Krylov subspace K_{k+1} such that, for all $i = 0, 1, \dots, k$,

$$\operatorname{span}\{\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_i\} = K_{i+1}.$$

These basis vectors satisfy a three term recurrence, for all $i = 0, 1, \dots, k-1$,

$$\mathbf{A}\mathbf{q}_i = \beta_{i-1}\mathbf{q}_{i-1} + \alpha_i \mathbf{q}_i + \beta_i \mathbf{q}_{i+1}$$

with initial conditions $\mathbf{q}_{-1} = \mathbf{0}$ and $\beta_{-1} = 0$. The coefficients $\{\alpha_i\}_{i=0}^{k-1}$ and $\{\beta_i\}_{i=0}^{k-1}$ defining the three term recurrence are also generated by the algorithm. This recurrence can be written in matrix form as

$$\mathbf{A}\mathbf{Q} = \mathbf{Q}\mathbf{T} + \beta_{k-1}\mathbf{q}_k \mathbf{e}_{k-1}^{\mathsf{T}} \tag{1.3}$$

where

$$\mathbf{Q} := \begin{bmatrix} | & | & & | \\ \mathbf{q}_0 & \mathbf{q}_1 & \cdots & \mathbf{q}_{k-1} \\ | & | & & | \end{bmatrix}, \quad \mathbf{T} := \begin{bmatrix} \alpha_0 & \beta_0 & & \\ \beta_0 & \alpha_1 & \ddots & \\ & \ddots & \ddots & \beta_{k-2} \\ & & \beta_{k-2} & \alpha_{k-1} \end{bmatrix}.$$

Algorithm 1.1 Lanczos algorithm

procedure LANCZOS($\mathbf{A}, \mathbf{v}, k$)
  $\mathbf{q}_0 = \mathbf{v}, \beta_{-1} = 0, \mathbf{q}_{-1} = \mathbf{0}$
  for $i = 0, 1, \dots, k-1$ **do**
    $\tilde{\mathbf{q}}_{i+1} = \mathbf{A}\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1}$
    $\alpha_i = \mathbf{q}_i^{\mathsf{H}}\tilde{\mathbf{q}}_{i+1}$
    $\hat{\mathbf{q}}_{i+1} = \tilde{\mathbf{q}}_{i+1} - \alpha_i\mathbf{q}_i$
    optionally, reorthogonalize, $\hat{\mathbf{q}}_{i+1}$ against $\{\mathbf{q}_j\}_{j=0}^{i}$
    $\beta_i = \|\hat{\mathbf{q}}_{i+1}\|$
    $\mathbf{q}_{i+1} = \hat{\mathbf{q}}_{i+1}/\beta_i$
  return $\{\mathbf{q}_i\}_{i=0}^{k}, \{\alpha_i\}_{i=0}^{k-1}, \{\beta_i\}_{i=0}^{k-1}$.

Remark 1.6. It is not uncommon for the matrices which we call $\mathbf{Q}$ and $\mathbf{T}$ to be denoted by $\mathbf{Q}_k$ and $\mathbf{T}_k$. We omit these subscripts for legibility, as the number of iterations k can be treated as fixed throughout this thesis. Note also that we begin indexing at zero so that indices match the degree of the corresponding polynomial. △

1.1.1 The insufficiency of interval-based bounds

As we previously noted, this thesis is concerned with polynomial approximations to $f(\mathbf{A})\mathbf{v}$ and $\mathbf{v}^{\mathsf{H}}f(\mathbf{A})\mathbf{v}$. The error of such methods is often closely related to problems in scalar polynomial approximation theory. In particular, note that

$$\|g(\mathbf{A})\|_2 = \max_{x\in\Lambda} |g(x)| =: \|g\|_\Lambda,$$

where Λ is the set of eigenvalues of $\mathbf{A}$. Here we have introduced the notation $\|g\|_S = \sup_{x\in S}|g(x)|$ for $g : \mathbb{C} \to \mathbb{C}$ and $S \subset \mathbb{C}$. Let $\|\cdot\|$ be any norm induced by a positive definite matrix with the same eigenvectors as $\mathbf{A}$. Then, a simple application of the sub-multiplicative property of matrix norms (see Lemma 10.1) implies

$$\frac{\|f(\mathbf{A})\mathbf{v} - [f]_s^{\circ\mathrm{p}}(\mathbf{A})\mathbf{v}\|}{\|\mathbf{v}\|} \le \|f(\mathbf{A}) - [f]_s^{\circ\mathrm{p}}(\mathbf{A})\|_2 = \|f - [f]_s^{\circ\mathrm{p}}\|_\Lambda. \tag{1.4}$$

Recalling our assumption $\|\mathbf{v}\|_2 = 1$, we also have

$$|\mathbf{v}^{\mathsf{H}}f(\mathbf{A})\mathbf{v} - \mathbf{v}^{\mathsf{H}}[f]_s^{\circ\mathrm{p}}(\mathbf{A})\mathbf{v}| \le \|f(\mathbf{A}) - [f]_s^{\circ\mathrm{p}}(\mathbf{A})\|_2\|\mathbf{v}\|_2^2 = \|f - [f]_s^{\circ\mathrm{p}}\|_\Lambda. \tag{1.5}$$

Thus, we see that the quality of our approximations can be studied in terms of the quality of the polynomial approximation $[f]_s^{\mathrm{op}}$ to f on the eigenvalues of $\mathbf{A}$.

Since polynomial approximation on an interval is well understood, it is common to bound the quality of Krylov subspace methods in terms of the best polynomial approximations on an interval. It is always true that

$$\|f - p\|_\Lambda \leq \|f - p\|_I,$$

where $I := [\lambda_{\min}, \lambda_{\max}]$ is the smallest interval containing all of the eigenvalues. Thus, it is common to bound the left hand sides of (1.4) and (1.5) by an expression like

$$2 \min_{\deg(p)<s} \|f - p\|_I. \tag{1.6}$$

However, while such bounds are useful in some situations, they often provide a large overestimate of the true behavior of Lanczos-based methods and are therefore unsuitable for use as practical stopping criteria.

1.1.2 The effect of finite precision arithmetic

While reorthogonalization in the Lanczos algorithm is unnecessary in exact arithmetic, omitting it often results in drastically different behavior when using finite precision arithmetic. Specifically, the Lanczos basis $\mathbf{Q}$ may be far from orthogonal, and the tridiagonal matrix $\mathbf{T}$ may be far from what would have been obtained in exact arithmetic. Because the Lanzcos algorithm is ostensibly unstable, there has been a widespread hesitance towards Lanczos-based approaches for problems involving matrix functions, at least without reorthogonalization [JP94; Sil+96; Aic+03; Wei+06; UCS17; GWG19].

The two primary effects of finite precision arithmetic on Lanczos-based methods when run without reorthogonalization are (i) a delay of convergence (increase in the number of iterations to reach a given level of accuracy) and (ii) a reduction in the maximal attainable accuracy. However, while both effects are easily noticeable on most problems, they do not imply that reorthogonalization is needed. In fact, throughout this thesis, we argue that Lanczos-based methods are highly effective even without reorthogonalization.

1.1.3 A motivating example

We now provide a simple and familiar example chosen to illustrate the themes introduced in this section. The conjugate gradient algorithm (CG) [HS52] is used to solve positive definite linear systems of equations, and is perhaps the most well-known Lanczos-based KSM. When applied to a positive-definite linear system $\mathbf{A}\mathbf{x} = \mathbf{v}$, CG produces iterates $\mathsf{cg}_k \in \mathcal{K}_k$ optimal in the $\mathbf{A}$-norm. This optimality implies the error bounds[2]

$$\frac{\|\mathbf{A}^{-1}\mathbf{v} - \mathsf{cg}_k\|_{\mathbf{A}}}{\|\mathbf{v}\|_{\mathbf{A}}} \overset{\text{(a)}}{\leq} \min_{\deg(p)<k} \|x^{-1} - p\|_{\Lambda} \overset{\text{(b)}}{\leq} \min_{\deg(p)<k} \|x^{-1} - p\|_{\mathcal{I}}. \tag{1.7}$$

Given $\mathcal{I} = [\lambda_{\min}, \lambda_{\max}]$, (1.7b) can be computed analytically and, roughly speaking, it decreases linearly at a rate proportional to $(1 - 1/\sqrt{\kappa})$. In other words, to reach accuracy ϵ, CG requires at most $O(\sqrt{\kappa}\log(\epsilon^{-1}))$ iterations, where $\kappa = \lambda_{\max}/\lambda_{\min}$ is the condition number of $\mathbf{A}$. This is the well known *root condition number bound* for CG. On the other hand, (1.7a) may be significantly better than the latter bound involving $\mathcal{I}$ and provide a more realistic picture of the convergence of CG. However, since (1.7a) depends on the spectrum of $\mathbf{A}$, which is typically unknown, the bound's use is in that it provides intuition into the theoretical behavior of CG rather than as a practical stopping critera.

In Figure 1.1, we plot the bounds (1.7a) and (1.7b) as well as the actual errors in a Lanczos-based implementation of CG run with and without reorthogonalization for a spectrum with exponentially spaced eigenvalues (the precise details are not important at this point). Observe that the bound (1.7a) decreases significantly faster than (1.7b) for the given spectrum. Note also that, even without reorthogonalization, CG converges significantly faster than (1.7b).

An alternative to CG is the Chebyshev semi-iterative method [FS50; GR02]. This method can be implemented without using the Lanczos algorithm by directly constructing an explicit polynomial approximation to x^{-1} on $\mathcal{I}$. However, as a result, the algorithm is unable to adapt to the spectrum of $\mathbf{A}$ and usually converges very similarly to (1.7b). Moreover, if $\mathcal{I}$ is estimated inaccurately, then the algorithm may become unstable. Interestingly, even in finite precision

[2] Throughout, we will occasionally use symbol "x" for the identity function $x : \lambda \mapsto \lambda$ rather than an unspecified real value. Thus, expressions like xp and $x^{-1} - p$ should respectively be interpreted to mean the functions $\lambda \mapsto \lambda p(\lambda)$ and $\lambda \mapsto \lambda^{-1} - p(\lambda)$.

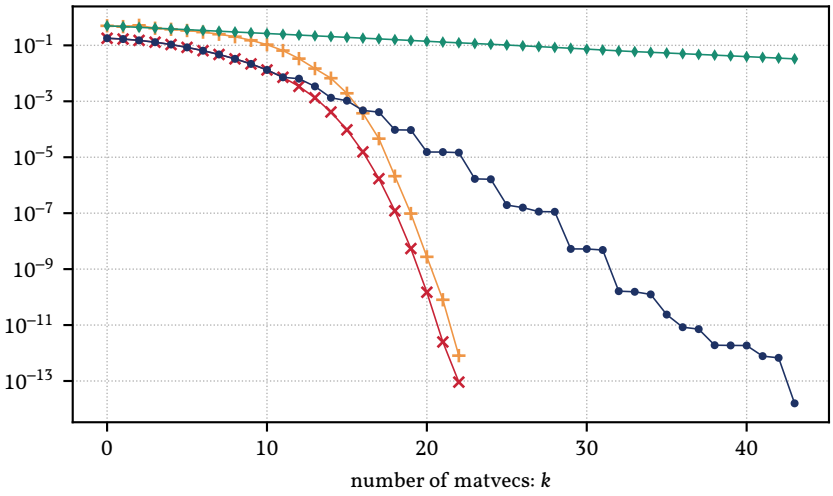


Figure 1.1: CG convergence (with and without reorthgonalization) and error bounds. *Legend*: CG error $\|\mathbf{A}^{-1}\mathbf{v} - \mathrm{cg}_k\|_{\mathbf{A}}/\|\mathbf{v}\|_{\mathbf{A}}$ with (—×—) and without (—•—) reorthgonalization, bound (1.7a) on Λ (—+—), and bound (1.7b) on $\mathcal{I}$ (—♦—).

arithmetic, an iterate very close to what would be produced by the Chebyshev method can be obtained from the Lanczos method [Gre89; DK91; MMS18]. Since the extreme eigenvalues of $\mathbf{T}$ typically provide a very good estimate for $\mathcal{I}$, this means that a Lanczos-based implementation of the Chebyshev method avoids the need for a priori parameter selection.

Remark 1.7. Optimization algorithms such as accelerated gradient descent also attain a root condition number iteration complexity on any smooth and strongly convex function (such as $\mathbf{x} \mapsto \frac{1}{2}\mathbf{x}^{\mathsf{H}}\mathbf{A}\mathbf{x} - \mathbf{x}^{\mathsf{H}}\mathbf{v}$ for positive definite linear systems). Since this rate is optimal among first-order methods for smooth and strongly convex functions, accelerated gradient descent is often referred to as "optimal". The fact that CG has a similar convergence guarantee often leads CG to be introduced as an alternative to accelerated gradient descent for linear systems. However, accelerated gradient descent is essentially equivalent to the Chebyshev semi-iterative method when applied to the above objective and therefore is not typically competitive with CG in terms of the number of matrix-vector products. △

1.2 Context and contributions

Essentially any numerical linear algebra textbook will have at least one chapter on KSMs. In fact, there are a number of texts which focus on the classical uses of KSMs: eigenvalue problems and solving linear systems of equations [Pai71; Gre97; Saa11; MS06; Meu06; LS13b]; see also [GO89] for a historical overview of early developments. The large number of such resources means that treatments of topics such as the behavior of algorithms in finite precision arithmetic can be found at a range of levels of detail.

Resources dealing with general matrix functions are less plentiful. While many textbooks might have a chapter on functions such as the exponential or square root, the only recent book I am aware of which is dedicated specifically to matrix functions is [Hig08]. However, this book is focused primarily on the case of computing $f(\mathbf{A})$, with only one chapter devoted to the task of computing $f(\mathbf{A})\mathbf{v}$ and no discussion on the task of $\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$. There are a number of more specialized texts on these topics. The topic of $\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ is covered thoroughly in [GM94; GM09] from a theoretical quadrature perspective. A widely used practical quadrature algorithm for estimating spectral sums and spectral densities called the kernel polynomial method is discussed in [Wei+06] but not analyzed theoretically. The thesis [Sch16] discusses practical error bounds for methods for computing $f(\mathbf{A})\mathbf{v}$ for symmetric and non-symmetric $\mathbf{A}$ as well as restarting techniques for non-symmetric problems, and the thesis [Cor22] discusses low-rank approximation of matrix functions as well as stochastic trace estimation of matrix functions. None of the above texts discuss thoroughly the impacts of finite precision arithmetic.

During my PhD studies, it became strikingly clear that the state of knowledge surrounding Lanczos-based methods for matrix functions is fragmented. For instance, there are several lines of work within the quantum physics literature which contain results not discovered in applied math until decades later. Conversely, practitioners in physics, data science, and machine learning, often lack knowledge regarding the practical behavior of Lanczos-based methods in finite precision arithmetic. While this can be partially attributed to a lack of due-diligence in studying background material, a larger problem is that the requisite background is not easily accessible to non-specialists. In fact, some of the most

relevant work on Lanczos-based methods in finite precision arithmetic is not even well known within the numerical analysis community.

This thesis aims to fill some of the gaps in the presentation of Lanczos-based methods for matrix functions by providing a comprehensive background on the topic. Indeed, several chapters are primarily expository, with the express goal of providing a more thorough context for the other chapters. While there are a number of technical contributions, arguably the most significant contributions of this thesis are the following two themes:

- Bounds based on polynomial approximation on a single interval are insufficient to describe the true behavior of Lanczos-based methods for matrix functions. Instead, one should seek bounds based on the spectrum which are able to take advantage of more fine-grained spectral structure such as gaps and outlying eigenvalues.
- While Lanczos-based methods may behave differently in finite precision arithmetic than exact arithmetic, they still outperform Krylov subspace methods based on explicit polynomial approximations. Moreover, the hyper-parameters in explicit polynomial methods can be determined effectively through the use of Lanczos-based implementations.

It is my hope that the contributions of this thesis are presented in a way which will promote understanding of and intuition for Lanczos-based methods for matrix functions outside of the numerical analysis community.

This thesis contains primarily work which appears in the following papers:

[Che+22a] T. Chen, A. Greenbaum, C. Musco, and C. Musco. "Error Bounds for Lanczos-Based Matrix Function Approximation". In: *SIAM Journal on Matrix Analysis and Applications* 43.2 (May 2022), pp. 787–811. DOI: 10.1137/21m1427784. arXiv: 2106.09806 [math.NA].

[Che+22b] T. Chen, A. Greenbaum, C. Musco, and C. Musco. *Low-memory Krylov subspace methods for optimal rational matrix function approximation*. 2022. arXiv: 2202.11251 [math.NA].

[CTU21] T. Chen, T. Trogdon, and S. Ubaru. "Analysis of stochastic Lanczos quadrature for spectrum approximation". In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 1728–1739. arXiv: 2105.06595 [cs.DS].

[CTU22] T. Chen, T. Trogdon, and S. Ubaru. *Randomized matrix-free quadrature for spectrum and spectral sum approximation*. 2022. arXiv: 2204.01941 [math.NA].

Rather than stapling these papers together, I have arranged the content in accordance with my broader goals for this thesis. In addition, a large amount of new exposition has been included in order to tie things together more cleanly and to provide additional context for non-experts. Towards this end, many of the numerical examples from the above papers have been modified for consistency with the rest of the thesis. The files needed to generate the figures in this thesis (as well as the thesis itself) are freely available online.

Chapter 2
Scalar polynomials

In this chapter, we introduce some basic theory regarding scalar polynomials[1] which will come in handy throughout the rest of this thesis. A scalar polynomial of degree k is a function of the form

$$x \mapsto c_0 + c_1 x + \cdots + c_k x^k,$$

where $c_0, c_1, \ldots, c_k$ are fixed scalars. Such a polynomial is naturally extended to matrices as

$$\mathbf{A} \mapsto c_0 \mathbf{I} + c_1 \mathbf{A} + \cdots + c_k \mathbf{A}^k$$

in a manner compatible with our definition of matrix functions. Thus, matrix polynomials are intimately related to Krylov subspace methods.

2.1 Basic definitions

Our discussion on quadrature centers around approximating distributions and integrals against distribution functions. Several examples of such functions are illustrated in Figure 2.1.

Definition 2.1. *A (signed) unit mass distribution function Υ is a right continuous function $\Upsilon : \mathbb{R} \to \mathbb{R}$ such that $\lim_{x\to\infty} \Upsilon(x) = 1$ and $\lim_{x\to-\infty} \Upsilon(x) = 0$. If Υ is also weakly increasing, we say it is a probability distribution function.*

[1] Polynomials were perhaps the first abstract mathematical object I encountered — I distinctly remember grappling with the meaning of a variable x, which is "somehow a number but also not", in the later years of elementary school. It's amazing, then, that the theory of polynomials of a single variable underlies so much of my PhD thesis.

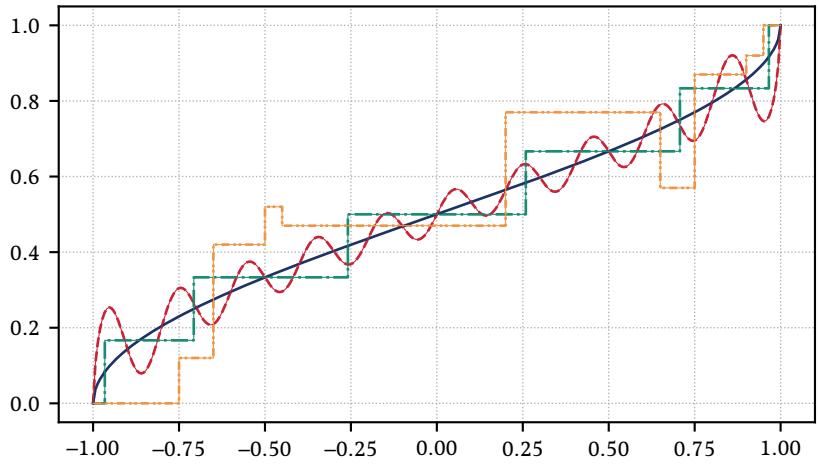


Figure 2.1: Sample unit mass distribution functions. *Legend*: continuous increasing distribution function (———), continuous distribution function which is not weakly-increasing (- - -), discrete weakly-increasing distribution function (—·—), discrete distribution function which is not weakly-increasing (·····).

Remark 2.2. If Υ is differentiable, then the derivative $\mathrm{d}\Upsilon/\mathrm{d}x = \Upsilon'$ is the usual probability density function. Likewise, in the sense of distributions, $\mathrm{d}\mathbb{1}[a \leq x]/\mathrm{d}x = \delta(x-a)$ where $\delta(x-a)$ is a unit mass Dirac delta function centered at a. Thus, if Υ is piecewise constant, then $\mathrm{d}\Upsilon/\mathrm{d}x$ can be expressed in terms of the sum of weighted Dirac delta functions, where a delta function is located at each discontinuity and weighted by the size of the corresponding jump. △

We now introduce several definitions which we will use throughout the next several chapters.

Definition 2.3. *Given a function f and distribution function Υ we denote by $\int_a^b f \,\mathrm{d}\Upsilon$ the standard Riemann–Stieltjes integral*

$$\int_a^b f \,\mathrm{d}\Upsilon := \lim_{\|P\|\to 0} \sum_{i=0}^{p-1} f(c_i)\big(\Upsilon(x^{(i+1)}) - \Upsilon(x^{(i)})\big),$$

where $P = \{a = x^{(0)} < \cdots < x^{(p)} = b\}$ is a partition of $[a,b]$, $\|P\| = \max_i |x^{(i+1)} - x^{(i)}|$, and $c_i \in [x_i, x_{i+1}]$.

For notational clarity, we will often write $\int f \, \mathrm{d}\Upsilon$ in which case a, b can be taken as $\pm\infty$.

Definition 2.4. *In the setting of the previous definition,*

$$\int_a^b f \, |\mathrm{d}\Upsilon| := \lim_{\|P\| \to 0} \sum_{i=0}^{p-1} f(c_i) \left| \Upsilon(x^{(i+1)}) - \Upsilon(x^{(i)}) \right|.$$

Definition 2.5. *Let Υ be a (distribution) function. The total variation (TV) of Υ, denoted $d_{\mathrm{TV}}(\Upsilon)$, is defined by*

$$d_{\mathrm{TV}}(\Upsilon) := \int |\mathrm{d}\Upsilon|.$$

Remark 2.6. If Ψ is a weakly-increasing unit-mass distribution function, then $d_{\mathrm{TV}}(\Psi) = 1$. △

To measure the similarity of two distribution functions, we will typically use the Wasserstein (earth mover) distance.

Definition 2.7. *Let Υ_1 and Υ_2 be two probability distribution functions. The Wasserstein distance between Υ_1 and Υ_2, denoted $d_{\mathrm{W}}(\Upsilon_1, \Upsilon_2)$, is defined by*

$$d_{\mathrm{W}}(\Upsilon_1, \Upsilon_2) := \int |\Upsilon_1 - \Upsilon_2| \, \mathrm{d}x.$$

It is well known that the Wasserstein distance between two distribution functions has a dual form involving 1-Lipshitz functions.

Definition 2.8. *We say that $f \in \mathsf{Lip}(L, S)$ if $|f(x) - f(y)| \leq L|x - y|$ for all $x, y \in S \subseteq \mathbb{R}$.*

Lemma 2.9. *Suppose Υ_1 and Υ_2 are two probability distribution functions of bounded total variation each constant on $(-\infty, a)$ and (b, ∞). Then,*

$$d_{\mathrm{W}}(\Upsilon_1, \Upsilon_2) = \sup \left\{ \int f \mathrm{d}\big(\Upsilon_1 - \Upsilon_2\big) : f \in \mathsf{Lip}(1, [a, b]) \right\}.$$

Remark 2.10. In some situations, other metrics may be more meaningful. For instance, if it is important for two distribution functions to agree to very high precision in a certain region, but only to moderate accuracy in others, then the Wasserstein distance may be unsuitable. △

2.2 Orthogonal polynomials

Throughout this thesis, μ will be a non-negative unit mass distribution function. Associated with μ is the inner product $\langle \cdot, \cdot \rangle_\mu$ defined by

$$\langle f, g \rangle_\mu := \int f g \, \mathrm{d}\mu. \tag{2.1}$$

The set of μ-square-integrable functions forms a Hilbert space with respect to this inner product, so we may hope to find an orthonormal basis $\{p_i\}_{i=0}^{\infty}$ of polynomials with $\deg(p_i) = i$. Such a basis is easily produced by a simple modification of the Gram-Schmidt algorithm which results in a naive implementation of the so-called Stieltjes algorithm. An implementation is described in Algorithm 2.1.

Algorithm 2.1 Stieltjes algorithm (naive)

1: **procedure** STIELTJES(μ, k)
2: $p_0 = 1$
3: **for** $i = 0, 1, \ldots, k-1$ **do**
4: $\tilde{p}_{i+1} = x p_i$
5: $\hat{p}_{i+1} = \tilde{p}_{i+1} - \left(\langle p_0, \tilde{p}_{i+1} \rangle_\mu \, p_0 + \cdots + \langle p_i, \tilde{p}_{i+1} \rangle_\mu \, p_i \right)$
6: $p_{i+1} = \hat{p}_{i+1} / \|\hat{p}_{i+1}\|_\mu$
7: **return** $\{p_i\}_{i=0}^{k}$

Note that the polynomials satisfy, for all $i \geq 0$,

$$x p_i = \|\hat{p}_{i+1}\|_\mu \, p_{i+1} + \langle p_0, \tilde{p}_{i+1} \rangle_\mu \, p_0 + \cdots + \langle p_i, \tilde{p}_{i+1} \rangle_\mu \, p_i. \tag{2.2}$$

This can be written in matrix form as

$$x \, [p_0, p_1, \ldots] = [p_0, p_1, \ldots] \mathbf{H},$$

where $[p_0, p_1, \ldots]$ is a *quasi-matrix* whose columns are the polynomials $\{p_i\}_{i=0}^{\infty}$ and $\mathbf{H}$ is a semi-infinite upper-Hessenberg matrix. Moreover, for all $i, j \geq 0$,

$$[\mathbf{H}]_{i,j} = \langle p_i, x p_j \rangle_\mu = \langle p_j, x p_i \rangle_\mu = [\mathbf{H}]_{j,i};$$

that is, $\mathbf{H}$ is symmetric. Since, by construction, $\mathbf{H}$ is upper-Hessenberg, this implies that $\mathbf{H}$ is symmetric tridiagonal!

Therefore, for all $i \geq 0$, (2.2) becomes the symmetric three-term recurrence

$$xp_i = \beta_{i-1}p_{i-1} + \alpha_i p_i + \beta_i p_{i+1} \tag{2.3}$$

with initial conditions $p_0 = 1$, $p_{-1} = 0$, and $\beta_{-1} = 0$, where $\{\alpha_i\}_{i\geq 0}$ and $\{\beta_i\}_{i\geq 0}$ are chosen to enforce orthogonality. In particular, Algorithm 2.1 can be modified to take advantage of this short-recurrence, resulting in the standard implementation of the Stieltjes algorithm, Algorithm 2.2. In the case that $\beta_i = 0$, the algorithm should be terminated as the dimension of Krylov subspaces does not continue to grow.

Algorithm 2.2 Stieltjes algorithm

procedure STIELTJES(μ, k)
  $p_0 = 1$
  for $i = 0, 1, \ldots, k-1$ **do**
    $\tilde{p}_{i+1} = xp_i$
    $\alpha_i = \langle p_i, \tilde{p}_{i+1}\rangle_\mu$
    $\hat{p}_{i+1} = \tilde{p}_{i+1} - \alpha_i p_i$
    $\beta_i = \|\hat{p}_{i+1}\|_\mu$
    $p_{i+1} = \hat{p}_{i+1}/\beta_i$
  return $\{p_i\}_{i=0}^{k}, \{\alpha_i\}_{i=0}^{k-1}, \{\beta_i\}_{i=0}^{k-1}$

Definition 2.11. *The, possibly semi-infinite, tridiagonal matrix* $\mathbf{M} = \mathbf{M}(\mu)$ *giving the three-term recurrence coefficients for the orthogonal polynomials of* μ *is called the Jacobi matrix corresponding to* μ*. Unless specified otherwise, the coefficients are*

$$\mathbf{M} = \begin{bmatrix} \alpha_0 & \beta_0 & & \\ \beta_0 & \alpha_1 & \beta_1 & \\ & \beta_1 & \alpha_2 & \ddots \\ & & \ddots & \ddots \end{bmatrix}.$$

An important property of a distribution function Υ are it's polynomial moments. We are particularly interested in those induced by μ.

Definition 2.12. *For each* $i \geq 0$*, the modified moments of* Υ *(with respect to* μ*) are*

$$m_i = m_i(\Upsilon, \mu) := \int p_i \, \mathrm{d}\Upsilon.$$

If $m_0, \ldots, m_s < \infty$*, we say* Υ *has finite moments through degree* s*.*

Jacobi matrices have many interesting properties, several of which we review here.

Lemma 2.13. *The upper-leftmost $k\times k$ submatrix of a Jacobi matrix is entirely determined by the moments through degree $2k-1$ of the associated distribution function.*

Proof. This is a direct consequence of the fact that the k-point (degree $2k-1$) Gaussian quadrature rule for a distribution function can be determined from the upper-leftmost $k \times k$ submatrix of the associated Jacobi matrix. This argument will be made whole in Section 3.2.2. □

Lemma 2.14. *Denote the zeros of p_k by $\{\theta_j^{(k)}\}_{j=0}^{k-1}$. Then, for any $j = 0, 1, \dots, k-1$,*

$$\left[p_0(\theta_j^{(k)}), p_1(\theta_j^{(k)}), \dots, p_{k-1}(\theta_j^{(k)})\right]^{\mathsf{H}}$$

is an eigenvector of $[\mathbf{M}]_{:k,:k}$ with eigenvalue $\theta_j^{(k)}$. Moreover, all eigenvectors are obtained in this way.

Proof. In matrix form, (2.3) becomes

$$x\,[p_0, p_1, \dots, p_{k-1}] = [p_0, p_1, \dots, p_{k-1}]\mathbf{M} + \beta_{k-1} p_k \mathbf{e}_{k-1}^{\mathsf{T}}.$$

Evaluating each side of the above equality at $\theta_j^{(k)}$ gives the first part of the result.

To show all eigenvectors are obtained in this way, it suffices to show that $\{\theta_j^{(k)}\}_{0\le j<k}$ are distinct. Let $\{t_j\}_{j=0}^{k'-1}$ be the points at which p_k changes signs. Then,

$$\int p_k \prod_{j=0}^{k'-1} (x - t_j)\, \mathrm{d}\mu \neq 0$$

since the integrand does not change signs. This implies $k' = k$ since p_k is orthogonal to all polynomials of lower degree. □

2.2.1 Chebyshev polynomials

Owing to the deep connection between Chebyshev polynomials and approximation theory [Tre19], one particularly important choice of μ is the distribution function corresponding to the Chebyshev polynomials of the first kind. We will often treat this case with special care.

Definition 2.15. *The Chebyshev distribution function of the first kind,* $\mu^T_{a,b} : [a,b] \to [0,1]$*, is defined as*

$$\mu^T_{a,b} := \frac{1}{2} + \frac{1}{\pi} \arcsin\left(\frac{2}{b-a}x - \frac{b+a}{b-a}\right).$$

Thus, for $x \in [a,b]$*,*

$$\frac{\mathrm{d}\mu^T_{a,b}}{\mathrm{d}x} = \frac{2}{\pi(b-a)}\left(1 - \left(\frac{2}{b-a}x - \frac{b+a}{b-a}\right)^2\right)^{-1/2}.$$

Definition 2.16. *The Chebyshev polynomials of the first kind, denoted* $\{T_i\}_{i=0}^{\infty}$*, are defined by the recurrence* $T_0 = 1$*,* $T_1 = x$*, and, for all* $i \geq 1$*,*

$$T_{i+1} := 2xT_i - T_{i-1}.$$

It can be verified that the orthogonal polynomials $\{p_i\}_{i=0}^{\infty}$ with respect to $\mu^T_{a,b}$ are given by $p_0 = T_0 = 1$ and, for all $i \geq 1$,

$$p_i = \sqrt{2}T_i\left(\frac{2}{b-a}x + \frac{b+a}{b-a}\right).$$

Therefore, the Jacobi matrix $\mathbf{M}(\mu^T_{a,b})$ has diagonal and off diagonals entries given by

$$\left[\frac{a+b}{2}, \frac{a+b}{2}, \ldots\right] \quad \text{and} \quad \left[\frac{b-a}{2\sqrt{2}}, \frac{b-a}{4}, \frac{b-a}{4}, \ldots\right],$$

respectively.

2.3 Polynomial approximations and bounds

As noted in the introduction, we use the notation $[f]_s^{\circ \mathrm{p}}$ to denote a degree s polynomial obtained from f by some algorithm parameterized by $\circ$. In particular, we will make the following definitions.

Definition 2.17. *Given a non-negative unit mass distribution function* μ *with degree* $s+1$ *orthogonal polynomial* p_{s+1} *we define,*

$$[f]_s^{\circ \mathrm{p}} := \begin{cases} \circ = i & \text{degree } s \text{ interpolant to } f \text{ at roots of } p_{s+1} \\ \circ = a & \text{degree } s \text{ truncated series for } f \text{ in } \langle \cdot, \cdot \rangle_\mu \end{cases}$$

.

The damped projection $[f]_s^{\text{d-ap}}$ and damped interpolant $[f]_s^{\text{d-ip}}$ are respectively obtained by scaling each of the coefficients of $[f]_s^{\text{ap}}$ and $[f]_s^{\text{ip}}$, when represented as a linear combination of the orthogonal polynomials $\{p_i\}_{i=0}^s$, by constants ρ_i for each $i = 0, 1, \dots, s$.

Definition 2.18. *Write $[f]_s^{\circ\text{p}}$ in a polynomial series with respect to $\langle \cdot, \cdot \rangle_\mu$; i.e. as*

$$[f]_s^{\circ\text{p}} = \sum_{i=0}^{s} c_i p_i.$$

Then, given damping coefficients $\{\rho_i\}_{i=0}^s$ with $0 \le \rho_i \le 1$ for all i,

$$[f]_s^{\text{d-}\circ\text{p}} := \sum_{i=0}^{s} \rho_i c_i p_i.$$

We now review several classical results from approximation theory which we will use throughout this thesis. These are constructive bounds for the case $\mu = \mu_{-1,1}^T$ which provide upper bounds for the quality of the *best* polynomial approximation to f. In fact, both $[f]_s^{\text{ap}}$ and $[f]_s^{\text{ip}}$ provide nearly optimal approximations in many settings [Tre19].

A full treatment requires at least a textbook, and we refer readers to [Tre19] for an excellent such book. The following theorems are summarized from Theorems 7.2 and 8.2 in [Tre19].

Definition 2.19. *We say that $f \in \mathsf{BV}(d, V, S)$ if, on $S \subseteq \mathbb{R}$, f is d times differentiable, its derivatives through $f^{(d-1)}$ are absolutely continuous, and the d-th derivative $f^{(d)}$ has total variation bounded above by some constant V on S.*

Definition 2.20. *For $\rho \ge 1$ the Bernstein ellipse $E_\rho(a, b)$ is the ellipse centered at $\frac{a+b}{2}$ with semi-axis lengths $\frac{b-a}{2}\frac{1}{2}(\rho + \rho^{-1})$ and $\frac{b-a}{2}\frac{1}{2}(\rho + \rho^{-1})$ along the real and imaginary directions; i.e*

$$E_\rho(a, b) = \left\{ z \in \mathbb{C} : z = \frac{b-a}{2}\frac{1}{2}(u + u^{-1}) + \frac{a+b}{2},\ u = \rho \exp(i\theta),\ \theta \in [0, 2\pi) \right\}.$$

Definition 2.21. *We say that $f \in \mathsf{Anl}(\rho, M, [a, b])$ if f is analytic on the region enclosed by $E_\rho(a, b)$ where it satisfies $|f(x)| \le M$.*

Theorem 2.22. *For an integer $d \ge 0$, suppose f is d times differentiable, its derivatives through $f^{(d-1)}$ are absolutely continuous, and the d-th derivative $f^{(d)}$ has total variation*

bounded above by some constant V on $[-1,-1]$; i.e., suppose $f \in \mathsf{BV}(d, V, [-1,1])$. Then, with $\mu = \mu^T_{-1,1}$, for any $s > d$,

$$\|f - [f]_s^{\mathrm{ap}}\|_{[-1,1]} \le \frac{2V}{\pi d(s-d)^d}, \qquad \|f - [f]_s^{\mathrm{ip}}\|_{[-1,1]} \le \frac{4V}{\pi d(s-d)^d}.$$

Theorem 2.23. *Suppose f is analytic on the region enclosed by the Bernstein ellipse $E_\rho(-1,1)$ where it satisfies $\|f\|_{E_\rho(-1,1)} \le M$; i.e., suppose $f \in \mathsf{Anl}(\rho, M, [-1,1])$. Then, with $\mu = \mu^T_{-1,1}$, for any $s \ge 0$,*

$$\|f - [f]_s^{\mathrm{ap}}\|_{[-1,1]} \le \frac{2M\rho^{-k}}{\rho - 1}, \qquad \|f - [f]_s^{\mathrm{ip}}\|_{[-1.1]} \le \frac{4M\rho^{-k}}{\rho - 1}.$$

Lemma 2.24. *Set $c_{\mathrm{i}} = 2$ and $c_{\mathrm{a}} = 1$. Then, for $\circ \in \{\mathrm{i}, \mathrm{a}\}$, $\|f - [f]_s^{\circ\mathrm{p}}\|_\infty < \epsilon/2$ provided*

$$s \ge \begin{cases} \frac{1}{\ln(\rho)} \ln\left(\frac{4c_\circ M}{\rho - 1}\right) + \frac{1}{\ln(\rho)} \ln\left(\varepsilon^{-1}\right) & f \in \mathsf{Anl}(\rho, M, [a,b]), \\ d + \left(\frac{4c_\circ V}{\pi d}\right)^{1/d} \varepsilon^{-1/d} & f \in \mathsf{BV}(d, V, [a,b]). \end{cases}$$

Proof. Define $\tilde{f} : \mathbb{R} \to \mathbb{R}$ by $\tilde{f}(x) = f(\frac{b-a}{2}x + \frac{a+b}{2})$. Then we have that,

$$\min_{\deg(p) \le s} \|f - p\|_{[a,b]} = \min_{\deg(p) \le s} \|\tilde{f} - p\|_{[-1,1]}.$$

Note that if $f \in \mathsf{Anl}(\rho, M, [a,b])$ then $\tilde{f} \in \mathsf{Anl}(\rho, M, [-1,1])$ and if $f \in \mathsf{BV}(d, V, [a,b])$ then $\tilde{f} \in \mathsf{BV}(d, V, [-1,1])$.

The result then follows by setting the upper bounds in Theorems 2.22 and 2.23 to $\epsilon/2$ and solving for s. □

Next, we consider polynomial approximations to 1-Lipshitz functions. Note that there exist 1-Lipshitz functions whose derivatives are not of bounded variation. Therefore we cannot simply use Theorem 2.22. Fortunately, the best approximation of differentiable functions is well studied. In particular, we have the following theorem due to Jackson; see [Ach92, Section 87] and [Che00, Section 6] for details.

Theorem 2.25. *Suppose f is 1-Lipshitz on $[-1,1]$; i.e., suppose that $f \in \mathsf{Lip}(1, [-1,1])$. Then,*

$$\min_{\deg(p) \le s} \|f - p\|_{[-1,1]} \le \frac{\pi}{2}(s+1)^{-1}.$$

In fact, the constant $\pi/2$ is the best possible under the stated conditions.

While the "vanilla" Chebyshev projection and interpolation do not attain this rate for all 1-Lipshitz functions, we can constructively obtain polynomial approximations which do attain this rate by damping.

Definition 2.26. *For $i = 0, 1, \dots, s$, the degree s Jackson's damping coefficients are*

$$\rho_i^J = \frac{(s-i+2)\cos\left(\frac{i\pi}{s+2}\right) + \sin\left(\frac{i\pi}{s+2}\right)\cot\left(\frac{\pi}{s+2}\right)}{s+2}.$$

The damped projection and interpolant then satisfy a similar bound to Theorem 2.25.

Theorem 2.27. *Suppose $f \in \mathrm{Lip}(1, [-1,1])$, $\mu = \mu^T_{-1,1}$, and we use Jackson's damping coefficients as in Theorem 2.26. Then for $\circ \in \{\text{d-i}, \text{d-a}\}$,*

$$\|f - [f]_s^{\circ\text{p}}\|_{[-1,1]} \leq \frac{\pi^2}{2}(s+2)^{-1}.$$

We provide a proof of this statement in Section 2.A. While our proof is based closely on [Riv81], the exact constant we obtain is sharper than other bounds we know of for the quality of the damped projection $[f]_s^{\text{d-ap}}$. Moreover, our version of the proof works for the damped interpolant $[f]_s^{\text{d-ip}}$. We were unable to find a similar result in the literature, although we suspect such a result is known.

2.A Proof of Jackson's theorem

In this section we prove Theorem 2.27. We follow Chapter 1 of [Riv81] closely, starting with trigonometric polynomials on $[-\pi, \pi]$ and then mapping to algebraic polynomials on $[-1, 1]$. Throughout this section we maintain the notation of [Riv81], so the constants in this section do not necessarily have the same meaning as the rest of the paper. In particular, n is the degree of the trigonometric polynomials used.

Given $g : \mathbb{R} \to \mathbb{R}$, 1-Lipshitz and 2π-periodic, for $\circ \in \{\text{i}, \text{a}\}$, define

$$s_n^\circ(\theta) := \frac{a_0^\circ}{2} + \sum_{k=1}^{n} \left(a_k^\circ \cos(k\theta) + b_k^\circ \sin(k\theta)\right)$$

where, for $k = 0, 1, \dots, n$

$$a_k^\circ := \frac{1}{\pi}\int_{-\pi}^{\pi^-} g(\phi)\cos(k\phi)M_n^\circ(\phi)\,\mathrm{d}\phi, \qquad b_k^\circ := \frac{1}{\pi}\int_{-\pi}^{\pi^-} g(\phi)\sin(k\phi)M_n^\circ(\phi)\,\mathrm{d}\phi.$$

Here $M_n^{\mathrm{a}}(\phi) := 1$ and

$$M_n^{\mathrm{i}}(\phi) := \frac{\pi}{n}\sum_{i\in\mathbb{Z}} \delta(\phi - \phi_i), \qquad \phi_i := \frac{2\pi(i-1/2)}{2n} - \pi$$

where $\delta(\phi)$ is a Dirac delta distribution centered at zero. Thus, s_n^{a} is the truncation of the Fourier series of g while s_n^{i} is the interpolant to g at the equally spaced nodes $\{\phi_i\}_{i-0}^{2n}$.

Remark 2.28. Note that $\int_{-\pi}^{\pi^-}$ means an integral over $[-\pi, \pi)$; i.e. the upper endpoint of integration is excluded. This is important for integrals involving M_n^{i} which can have nonzero integral at a single point. △

Finally, define the damped interpolant/approximant

$$q_n^\circ(\theta) := \frac{a_0^\circ}{2} + \sum_{k=1}^{n} \rho_k \left(a_k^\circ \cos(k\theta) + b_k^\circ \sin(k\theta)\right)$$

where the damping coefficients $\{\rho_k\}_{k=1}^n$ are arbitrary real numbers. Our aim is to bound $\|q_n^\circ - g\|_{[-\pi,\pi]}$.

Lemma 2.29. *For all $k = 0, 1, \dots, n$,*

$$\frac{1}{\pi}\int_{-\pi}^{\pi^-} \cos(k\phi)M_n^\circ(\phi)\,\mathrm{d}\phi = \begin{cases} 2 & k = 0 \\ 0 & k \in 1, 2, \dots, n \end{cases}$$

and

$$\frac{1}{\pi}\int_{-\pi}^{\pi^-} \sin(kx)M_n^\circ(\phi)\,\mathrm{d}\phi = 0.$$

Proof. Clearly $\frac{1}{\pi}\int_{-\pi}^{\pi^-} \cos(0\phi)M_n^{\mathrm{a}}\mathrm{d}\phi = 2$, and for $k > 0$, we have,

$$\frac{1}{\pi}\int_{-\pi}^{\pi^-} \cos(k\phi)M_n^{\mathrm{a}}(\phi)\,\mathrm{d}\phi = 0.$$

By definition,

$$\frac{1}{\pi}\int_{-\pi}^{\pi^-} \cos(k\phi)M_n^{\mathrm{i}}(\phi)\,\mathrm{d}\phi = \frac{1}{n}\sum_{j=1}^{2n} \cos(k\phi_j) = \frac{1}{n}\,\mathrm{Re}\sum_{j=1}^{2n} \exp(\boldsymbol{i}k\phi_j).$$

The case for $k = 0$ is clear. Assume $k > 0$. Then, using that $\phi_j = \frac{\pi}{n}(j - \frac{1}{2}) - \pi$ we have

$$\frac{1}{n}\operatorname{Re}\sum_{j=1}^{2n}\exp(\boldsymbol{i}k\phi_j) = \frac{1}{n}\operatorname{Re}\left[\exp\left(\boldsymbol{i}k\left(\frac{\pi}{2n} - \pi\right)\right)\sum_{j=0}^{2n-1}\exp\left(\boldsymbol{i}\frac{k\pi}{n}j\right)\right].$$

The result follows by observing that

$$\sum_{j=0}^{2n-1}\exp\left(\boldsymbol{i}\frac{k\pi}{n}j\right) = \frac{\exp(2\boldsymbol{i}k\pi) - 1}{\exp(\boldsymbol{i}k\pi/n) - 1} = 0.$$

Finally, since $\sin(k\phi)$ is odd and M_n° is symmetric about zero, the corresponding integrals are zero. □

We now introduce a generalized version of [Riv81, Lemma 1.4].

Lemma 2.30. *Define*

$$u_n(\phi) := \frac{1}{2} + \sum_{k=1}^{n}\rho_k\cos(k\phi).$$

Then,

$$q_n^\circ(\theta) = \frac{1}{\pi}\int_{-\pi}^{\pi} g(\phi + \theta)u_n(\phi)M_n^\circ(\phi + \theta)\,\mathrm{d}\phi.$$

Proof. First, note that

$$\begin{aligned}\pi q_n^\circ(\theta) &= \frac{1}{2}\left(\int_{-\pi}^{\pi} g(\phi)M_n^\circ(\phi)\,\mathrm{d}\phi\right) + \sum_{k=1}^{n}\rho_k\left(\left(\int_{-\pi}^{\pi} g(\phi)\cos(k\phi)M_n^\circ(\phi)\,\mathrm{d}\phi\right)\cos(k\theta)\right.\\ &\qquad\left. + \left(\int_{-\pi}^{\pi} g(\phi)\sin(k\phi)M_n^\circ(\phi)\,\mathrm{d}\phi\right)\sin(k\theta)\right)\\ &= \int_{-\pi}^{\pi} g(\phi)\left(\frac{1}{2} + \sum_{k=1}^{n}\rho_k(\cos(k\phi)\cos(k\theta) + \sin(k\phi)\sin(k\theta))\right)M_n^\circ(\phi)\,\mathrm{d}\phi.\end{aligned}$$

Thus, using the identity $\cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta) = \cos(\alpha - \beta)$ and the definition of u_n

$$\begin{aligned}q_n^\circ(\theta) &= \frac{1}{\pi}\int_{-\pi}^{\pi} g(\phi)\left(\frac{1}{2} + \sum_{k=1}^{n}\rho_k\cos(k(\phi - \theta))\right)M_n^\circ(\phi)\,\mathrm{d}\phi\\ &= \frac{1}{\pi}\int_{-\pi}^{\pi} g(\phi)u_n(\phi - \theta)M_n^\circ(\phi)\,\mathrm{d}\phi\end{aligned}$$

Now, note that g, u_n, and M_n° are 2π-periodic so by a change of variables,

$$\begin{aligned}\frac{1}{\pi}\int_{-\pi}^{\pi^-} g(\phi)u_n(\phi-\theta)M_n^\circ(\phi)\,\mathrm{d}\phi &= \frac{1}{\pi}\int_{-\pi-\theta}^{\pi^--\theta} g(\phi+\theta)u_n(\phi)M_n^\circ(\phi+\theta)\,\mathrm{d}\phi \\ &= \frac{1}{\pi}\int_{-\pi}^{\pi^-} g(\phi+\theta)u_n(\phi)M_n^\circ(\phi+\theta)\,\mathrm{d}\phi. \qquad \square\end{aligned}$$

Next, we prove a result similar to [Riv81, Lemma 1.7], but by assuming that g is 1-Lipshitz we obtain a slightly better constant.

Lemma 2.31. *Suppose $u_n(\phi) \geq 0$ for all ϕ. Then, if g is 1-Lipshitz,*

$$\|g - q_n^\circ\|_{[-\pi,\pi]} \leq \frac{\pi}{\sqrt{2}}(1-\rho_1)^{1/2}.$$

Proof. Fix any $\theta \in [-\pi,\pi]$. Recall that g is 1-Lipshitz so that $|g(\theta)-g(\phi+\theta)| \leq |\phi|$. Using this and the fact that u_n is non-negative,

$$\begin{aligned}|g(\theta) - q_n^\circ(\theta)| &= \left|\frac{1}{\pi}\int_{-\pi}^{\pi^-} (g(\theta)-g(\phi+\theta))\,u_n(\phi)M_n^\circ(\phi+\theta)\,\mathrm{d}\phi\right| \\ &\leq \frac{1}{\pi}\int_{-\pi}^{\pi^-} |\phi|u_n(\phi)M_n^\circ(\phi+\theta)\,\mathrm{d}\phi.\end{aligned}$$

Next, note M_n° and u_n are 2π-periodic. Using this followed by the fact that $\cos(k(\phi-\theta)) = \cos(k\phi)\cos(k\theta) - \sin(k\phi)\sin(k\theta)$, the definition of u_n, and Lemma 2.29, we have

$$\frac{1}{\pi}\int_{-\pi}^{\pi^-} M_n^\circ(\phi+\theta)u_n(\phi)\,\mathrm{d}\phi = \frac{1}{\pi}\int_{-\pi}^{\pi^-} M_n^\circ(\phi)u_n(\phi-\theta)\,\mathrm{d}\phi = 1.$$

Therefore, by the Cauchy–Schwarz inequality,

$$\begin{aligned}&\left(\frac{1}{\pi}\int_{-\pi}^{\pi^-} |\phi|u_n(\phi)M_n^\circ(\phi+\theta)\,\mathrm{d}\phi\right)^2 \\ &\qquad= \left(\frac{1}{\pi}\int_{-\pi}^{\pi^-} |\phi|u_n(\phi)\cdot u_n(\phi)M_n^\circ(\phi+\theta)\mathrm{d}\phi\right)^2 \\ &\qquad\leq \left(\frac{1}{\pi}\int_{-\pi}^{\pi^-} \phi^2 u_n(\phi)M_n^\circ(\phi+\theta)\,\mathrm{d}\phi\right)\left(\frac{1}{\pi}\int_{-\pi}^{\pi^-} u_n(\phi)M_n^\circ(\phi+\theta)\,\mathrm{d}\phi\right) \\ &\qquad= \frac{1}{\pi}\int_{-\pi}^{\pi^-} \phi^2 u_n(\phi)M_n^\circ(\phi+\theta)\,\mathrm{d}\phi.\end{aligned}$$

Using the fact that $\phi^2 \le \frac{\pi^2}{2}(1-\cos(\phi))$ we have

$$\frac{1}{\pi}\int_{-\pi}^{\pi} \phi^2 u_n(\phi) D_n(\phi+\theta)\,\mathrm{d}\phi \le \frac{\pi^2}{2}\frac{1}{\pi}\int_{-\pi}^{\pi}(1-\cos(\phi))u_n(\phi)M_n^\circ(\phi+\theta)\,\mathrm{d}\phi.$$

Next, we use that $\cos(\phi)\cos(k\phi) = \frac{1}{2}(\cos((k-1)\phi)+\cos((k+1)\phi))$ and Lemma 2.29 to obtain

$$\frac{\pi^2}{2}\frac{1}{\pi}\int_{-\pi}^{\pi}(1-\cos(\phi))u_n(\phi)M_n^\circ(\phi+\theta)\,\mathrm{d}\phi = \frac{\pi^2}{2}(1-\rho_1).$$

Combining this sequence of inequalities we find that

$$|g(\theta)-q_n^\circ(\theta)| \le \frac{\pi}{\sqrt{2}}(1-\rho_1)^{1/2}. \qquad \square$$

Lemma 2.32. *If we use Jackson's damping coefficents from Theorem 2.26, then u_n is positive and*

$$\frac{\pi}{\sqrt{2}}(1-\rho_1)^{1/2} \le \frac{\pi^2}{2}(n+2)^{-1}.$$

Proof. Let $\{c_\ell\}_{\ell=0}^n$ be any real numbers. Then

$$\left(\sum_{\ell=0}^n c_\ell \exp(\boldsymbol{i}\ell\theta)\right)\left(\sum_{\ell=0}^n c_\ell \exp(-\boldsymbol{i}\ell\theta)\right) = \left|\sum_{\ell=0}^n c_\ell \exp(\boldsymbol{i}\ell\theta)\right|^2 \ge 0.$$

Expanding and using that $\exp(\boldsymbol{i}k\theta)+\exp(-\boldsymbol{i}k\theta) = 2\cos(k\theta)$ we find

$$\left(\sum_{\ell=0}^n c_\ell \exp(\boldsymbol{i}\ell\theta)\right)\left(\sum_{\ell=0}^n c_\ell \exp(-\boldsymbol{i}\ell\theta)\right) = \sum_{k=0}^n c_k^2 + 2\sum_{p=1}^n\sum_{k=0}^{n-p} c_k c_{k+p}\cos(p\theta).$$

Because u_n must have the constant term equal to $1/2$ we require $c_0^2+\ldots+c_n^2 = 1/2$.

For $\ell = 0, 1, \ldots, n$, let

$$c_\ell = c\sin\left(\frac{\ell+1}{n+2}\pi\right)$$

where

$$c^2 = \left(\sum_{\ell=0}^n 2\sin^2\left(\frac{\ell+1}{n+2}\pi\right)\right)^{-1} = \frac{1}{n+2}.$$

Then setting $\rho_0 = 1$ and

$$\rho_k = 2\sum_{\ell=0}^{n-k} c_\ell c_{c+k}$$

we obtain u_n.

Next, we show that these damping coefficients are equal to those described above. Following [Wei+06] we have that

$$\begin{aligned}
2\sum_{\ell=0}^{n-k} c_\ell c_{\ell+k} &= 2c^2 \sum_{\ell=0}^{n-k} \sin\left(\frac{\ell+1}{n+2}\pi\right)\sin\left(\frac{\ell+k+1}{n+2}\pi\right) \\
&= 2c^2 \sum_{\ell=1}^{n-k+1} \sin\left(\frac{\ell}{n+2}\pi\right)\sin\left(\frac{\ell+k}{n+2}\pi\right) \\
&= c^2 \sum_{\ell=1}^{n-k+1} \left(\cos\left(\frac{k}{n+2}\pi\right) - \cos\left(\frac{2\ell+k}{n+2}\pi\right)\right) \\
&= c^2 \left((n-k)\cos\left(\frac{k}{n+2}\pi\right) - \operatorname{Re}\sum_{\ell=1}^{n-k+1} \exp\left(\boldsymbol{i}\frac{2\ell+k}{n+2}\pi\right)\right) \\
&= c^2 \left((n-k+1)\cos\left(\frac{k}{n+2}\pi\right) - \sin\left(\frac{\ell}{n+2}\pi\right)\cot\left(\frac{\pi}{n+2}\right)\right).
\end{aligned}$$

These are exactly Jackson's damping coefficients.

Using this expression, it's easy to verify that $\rho_1 = \cos(\pi/(n+2))$. Thus

$$(1-\rho_1)^{1/2} = \left(1-\cos\left(\frac{\pi}{n+2}\right)\right)^{1/2} = \sqrt{2}\sin\left(\frac{\pi}{2n+4}\right) \le \sqrt{2}\frac{\pi}{2n+4}$$

so

$$\frac{\pi}{\sqrt{2}}(1-\rho_1)^{1/2} \le \frac{\pi^2}{2n+4}$$ □

Finally, we prove the desired theorem.

Proof of Theorem 2.27. Without loss of generality, we can consider the case that f is 1-Lipshitz. For $\theta \in [-\pi, \pi)$ define g by $g(\theta) = f(\cos(\theta))$. Then g is 1-Lipshitz, 2π-periodic, and even. Next define the inverse mapping of the damped trigonometric polynomial q_n° for g as

$$p_n^\circ(t) = q_n^\circ(\arccos(t)).$$

For any $t \in [-1, 1]$, setting $\theta = \arccos(t) \in [0, \pi]$ we use Lemmas 2.31 and 2.32 to obtain the bound

$$|p_n^\circ(t) - f(t)| = |p_n^\circ(\cos(\theta)) - f(\cos(\theta))| = |q_n^\circ(\theta) - g(\theta)| \le \frac{\pi^2}{2} n^{-1}.$$

We will now show that $p_n^\circ(t) = [f]_n^{\text{d-}\circ}$. The mapping $\theta = \arccos(t)$ gives the Chebyshev polynomials; indeed, it is well known that

$$T_k(t) = \cos(k \arccos(t)).$$

Since g is even we have $b_k^\circ = 0$ so

$$p_n^\circ(t) = q_n^\circ(\arccos(t)) = \frac{a_0^\circ}{2} + \sum_{k=1}^{n} \rho_k a_k^\circ T_k(t).$$

Thus, our goal is to show that a_k° are the coefficients for the Chebyshev approximation/interpolation series.

Towards this end, recall that

$$a_k^\circ = \frac{1}{\pi} \int_{-\pi}^{\pi^-} g(\phi) \cos(k\phi) M_n^\circ(\phi) \, \mathrm{d}\phi.$$

Since g is even we can replace the integral on $[-\pi, \pi)$ an integral on $(0, \pi)$ and an integral on $[0, \pi)$. We first consider the case $M_n^{\mathrm{a}}(\phi) = 1$. Noting that $-\pi^{-1} \arccos(t) = \mu_{-1,1}^T(t)$, we find

$$a_k^{\mathrm{a}} = 2 \int_{-1}^{1} f T_k \, \mathrm{d}\mu_{-1,1}^T$$

as desired. For $j = 1, 2, \dots, 2n$, we have the Chebyshev nodes

$$\cos(\phi_j) = \cos\left(\frac{2\pi(j - 1/2)}{2n} - \pi\right) = -\cos\left(\frac{\pi(j - 1/2)}{n}\right).$$

Thus,

$$M_n^{\mathrm{i}}(t) = \frac{\pi}{n} \sum_{i \in \mathbb{Z}} \delta(x - x_i), \qquad x_i = -\cos\left(\frac{\pi(j - 1/2)}{n}\right)$$

so

$$a_k^{\mathrm{i}} = 2 \sum_{i=1}^{2n} \frac{\pi}{n} f(x_i) T_k(x_i) = 2 \int_{-1}^{1} f T_k \, \mathrm{d}[\mu_{-1,1}^T]_n^{\mathrm{gq}}$$

as desired. The result follows by renaming n to s. □

Chapter 3
Matrix-free quadrature

This chapter focuses primarily on quadrature rules for the *weighted* CESM induced by $\mathbf{A}$ and a unit vector $\mathbf{v}$.

Definition 3.1. *The weighted CESM $\Psi : \mathbb{R} \to [0, 1]$, induced by $\mathbf{A}$ and a unit vector $\mathbf{v}$, is defined by*

$$\Psi(x) = \Psi_{\mathbf{A},\mathbf{v}}(x) := \mathbf{v}^{\mathsf{H}} \mathbb{1}[\mathbf{A} \leq x]\mathbf{v}.$$

This definition implies that

$$\int f \, \mathrm{d}\Psi = \mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v},$$

so it is clear that Ψ is closely related to the task of approximating $\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$. In fact, in this chapter, we take the perspective that Krylov subspace methods for $\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ are in correspondence with quadrature rules for Ψ. Such a perspective was popularized by [GM94; GM09]

Remark 3.2. It is now clear that the Lanczos algorithm Algorithm 1.1 is simply the Stieltjes procedure Algorithm 2.2 applied to the weighted CESM Ψ. Specifically, $\mathbf{q}_i \propto p_i(\mathbf{A})\mathbf{v}$ for $i = 0, 1, \ldots, k$ and the tridiagonal matrix $\mathbf{T}$ generated by Lanczos is equal to the Jacobi matrix $\mathbf{M}(\Psi)$. △

The chapter title, *matrix-free quadrature* refers to the fact that our approach to constructing quadrature approximations to Ψ involve matrix-free algorithms; i.e. algorithms which access $\mathbf{A}$ only through matrix-vector products. While the quadrature rules we study are standard in approximation theory, it is worth

noting several critical differences between classical quadrature methods and the algorithm studied in this chapter. First, the costs of the algorithms in this paper are determined primarily by the number of matrix-vector products. This is because we typically only want to approximate $\int f \, d\Psi$ for a single, or perhaps a few, functions. On the other hand, the weight functions which classical quadrature rules approximate never change, so nodes and weights can be precomputed and the dominant cost becomes the cost to evaluate f at the quadrature nodes. Second, while classical weight functions, such as the weight functions for Jacobi or Hermite polynomials, are typically relatively uniform in the interior of the interval of integration, Ψ may vary wildly from application to application. In some cases Ψ might resemble the distribution function of a classical weight function whereas in others it might have large gaps, jumps, and other oddities. These distinctions are hinted at throughout the chapter and illustrated explicitly in the numerical examples at the end of this chapter.

Moment based methods for estimating the weighted CESM[1] have been used in physics for at least half a century. Early approaches were based on monomial moments [Cyr67; Cyr69; DC70; DC71; CD71], but the use of modified Chebyshev moments [WB72] and Lanczos-based approaches [Hay+72; HHK72; HHK75] were soon introduced.

3.1 Extracting moments from a Krylov subspace

Bases for the Krylov subspace $\mathcal{K}_{k+1} = \operatorname{span}\{\mathbf{v}, \mathbf{A}\mathbf{v}, \dots, \mathbf{A}^k\mathbf{v}\}$ can be computed using k matrix-vector products with $\mathbf{A}$ and contain a wealth of information about the interaction of $\mathbf{A}$ with $\mathbf{v}$; in particular, they contain the information necessary to compute the moments of Ψ through degree $2k$. Indeed, for all $i, j \geq 0$,

$$(\mathbf{A}^j\mathbf{v})^{\mathsf{H}}(\mathbf{A}^j\mathbf{v}) = \mathbf{v}^{\mathsf{H}}\mathbf{A}^{i+j}\mathbf{v} = \int x^{i+j} \, d\Psi.$$

Note, however, that it is sometimes more straightforward to obtain the moments through degree s, for some $s \leq 2k$. Thus, we will use s to denote the degree of the maximum moment we compute and k to denote the number of matrix-vector products used.

[1] In physics, the "density" $d\Psi/dx$ is often called the local density of states (local DOS).

3.1.1 Computing modified moments directly

Perhaps the most obvious approach to computing modified moments is to construct the basis $[p_0(\mathbf{A})\mathbf{v}, \dots, p_k(\mathbf{A})\mathbf{v}]$ for $\mathcal{K}_{k+1}$ and then compute

$$\mathbf{v}^{\mathsf{H}}[p_0(\mathbf{A})\mathbf{v}, \dots, p_k(\mathbf{A})\mathbf{v}].$$

This can be done using k matrix-vector products and $O(n)$ storage using the matrix recurrence version of (2.3). Indeed for all $i = 0, 1, \dots, k-1$ we have that

$$\mathbf{A}p_i(\mathbf{A})\mathbf{v} = \beta_{i-1}p_{i-1}(\mathbf{A})\mathbf{v} + \alpha_i p_i(\mathbf{A})\mathbf{v} + \beta_i p_{i+1}(\mathbf{A})\mathbf{v}$$

from which we can implement an efficient matrix-free algorithm to compute the modified moments $\{m_i\}_{i=0}^k$, as shown in Algorithm 3.1.

Algorithm 3.1 Get modified moments of Ψ wrt. μ

```
1: procedure GET-MOMENTS(A, v, k, μ)
2:     q_0 = v, m_0 = v^H v, q_{-1} = 0, β_{-1} = 0
3:     for i = 0, 1, ..., k − 1 do
4:         q_{i+1} = (1/β_i) (A q_i − α_i q_i − β_{i−1} q_{i−1})
5:         m_{i+1} = v^H q_{i+1}
6:     return {m_i}_{i=0}^k
```

If we instead compute

$$[p_0(\mathbf{A})\mathbf{v}, \dots, p_k(\mathbf{A})\mathbf{v}]^{\mathsf{H}}[p_0(\mathbf{A})\mathbf{v}, \dots, p_k(\mathbf{A})\mathbf{v}],$$

then we have the information required to compute the modified moments though degree $2k$. However, it is not immediately clear how to do this without the $O(kn)$ memory required to store a basis for $\mathcal{K}_{k+1}$. It turns out it is indeed generally possible to compute these moments without storing the entire basis, and we discuss a principled approach for doing so using connection coefficients in Section 3.1.2.

One case where extracting the moments to degree $2k$, without storing a basis for Krylov subspace, is straightforward is when $\mu = \mu_{a,b}^T$. This is because, for all $i \geq 0$, the Chebyshev polynomials satisfy the identities

$$T_{2i} = 2(T_i)^2 - 1, \qquad T_{2i+1} = 2T_iT_{i+1} - x.$$

Thus, using the recurrence for the Chebyshev polynomials and their relation to the orthogonal polynomials with respect to $\mu_{a,b}^T$, we obtain Algorithm 3.2. This algorithm is well-known in papers on the kernel polynomial method are variants; see for instance [Ski89; SR94; Wei+06; Hal21].

Remark 3.3. The Chebyshev polynomials grow rapidly outside of the interval $[-1, 1]$. Therefore, if the spectrum of $\mathbf{A}$ extends beyond this interval, then computing the Chebyshev polynomials in $\mathbf{A}$ may suffer from numerical instabilities. Instead, the distribution function $\mu_{a,b}^T$ and corresponding orthogonal polynomials should be used for some choice of a and b with $\mathcal{I} \subset [a, b]$.

△

Algorithm 3.2 Get modified moments of Ψ wrt. $\mu_{a,b}^T$

procedure GET-CHEBYSHEV-MOMENTS($\mathbf{A}, \mathbf{v}, k, a, b$)
  $\mathbf{q}_0 = \mathbf{v}, m_0 = \mathbf{q}_0^{\mathsf{H}}\mathbf{q}_0$
  $\mathbf{q}_1 = \frac{2}{b-a}(\mathbf{A}\mathbf{q}_0 - \frac{a+b}{2}\mathbf{q}_0), m_1 = \sqrt{2}\mathbf{q}_0^{\mathsf{H}}\mathbf{q}_1$
  for $i = 1, 2, \ldots, k-1$ **do**
    $m_{2i} = \sqrt{2}(2\mathbf{q}_i^{\mathsf{H}}\mathbf{q}_i - m_0)$
    $\mathbf{q}_{i+1} = 2\frac{2}{b-a}(\mathbf{A}\mathbf{q}_i - \frac{a+b}{2}\mathbf{q}_i) - \mathbf{q}_{i-1}$
    $m_{2i+1} = \sqrt{2}(2\mathbf{q}_i^{\mathsf{H}}\mathbf{q}_{i+1}) - m_1$
  $m_{2k} = \sqrt{2}(2\mathbf{q}_k^{\mathsf{H}}\mathbf{q}_k - m_0)$
  return $\{m_i\}_{i=0}^{2k}$

3.1.2 Connection coefficients to compute more modified moments

We now discuss how to use connection coefficients to compute the modified moments of Ψ with respect to μ given knowledge of either (i) the modified moments of Ψ with respect to some distribution ν or (ii) the tridiagonal matrix computed using Algorithm 1.1. Much of our discussion on connection coefficients is based on [WO21]; see also [FG91].

Definition 3.4. *The connection coefficient matrix $\mathbf{C} = \mathbf{C}_{\mu\to\nu}$ is the upper triangular matrix representing a change of basis between the orthogonal polynomials $\{p_i\}_{i=0}^{\infty}$ with respect to μ and the orthogonal polynomials $\{q_i\}_{i=0}^{\infty}$ with respect to ν, whose entries satisfy,*

$$p_s = [\mathbf{C}]_{0,s}q_0 + [\mathbf{C}]_{1,s}q_1 + \cdots + [\mathbf{C}]_{s,s}q_s.$$

Definition 3.4 implies that, for all $i = 0, 1, \dots, s$,

$$m_i = \int p_i \, \mathrm{d}\Psi = \sum_{j=0}^{i} [\mathbf{C}]_{j,i} \int q_j \, \mathrm{d}\Psi = \sum_{j=0}^{i} [\mathbf{C}]_{j,i} n_j$$

where $\{n_i\}_{i=0}^{s}$ are the modified moments of Ψ with respect to ν. Thus, we can easily obtain the modified moments $\{m_i\}_{i=0}^{s}$ of Ψ with respect to μ from the modified moments of Ψ with respect to ν. In particular, if $\mathbf{m}$ and $\mathbf{n}$ denote the vectors of modified moments, then $\mathbf{m} = \mathbf{C}^{\mathsf{T}}\mathbf{n}$.

Moreover, in the special case that ν has the same moments as Ψ through degree s, so, for any $j \geq 0$,

$$n_j = \int q_j \, \mathrm{d}\Psi = \int q_j \, \mathrm{d}\nu = \int q_0 q_j \, \mathrm{d}\nu = \mathbb{1}[j = 0].$$

Therefore, the modified moments of Ψ (with respect to μ) through degree s can be computed by

$$m_i = \int p_i \, \mathrm{d}\Psi = [\mathbf{C}]_{0,i}.$$

In order to use the above expressions, we must compute the connection coefficient matrix. Definition 3.4 implies that for all $i \leq j$, the entries of the connection coefficient matrix are given by

$$[\mathbf{C}]_{i,j} = \int q_i p_j \, \mathrm{d}\nu.$$

Unsurprisingly, then, the entries of the connection coefficient matrix $\mathbf{C} = \mathbf{C}_{\mu\to\nu}$ can be obtained by a recurrence relation.

Proposition 3.5 ([WO21, Corollary 3.3])**.** *Suppose the Jacobi matrices for μ and ν are respectively given by*

$$\mathbf{M}(\mu) = \begin{bmatrix} \alpha_0 & \beta_0 & & \\ \beta_0 & \alpha_1 & \beta_1 & \\ & \beta_1 & \alpha_2 & \ddots \\ & & \ddots & \ddots \end{bmatrix}, \qquad \mathbf{M}(\nu) = \begin{bmatrix} \gamma_0 & \delta_0 & & \\ \delta_0 & \gamma_1 & \delta_1 & \\ & \delta_1 & \gamma_2 & \ddots \\ & & \ddots & \ddots \end{bmatrix}.$$

Then the entries of $\mathbf{C} = \mathbf{C}_{\mu\to\nu}$ satisfy, for $i, j \geq 0$, the following recurrence:

$$[\mathbf{C}]_{0,0} = 1$$

$$
\begin{aligned}
[\mathbf{C}]_{0,1} &= (\gamma_0 - \alpha_0)/\beta_0 \\
[\mathbf{C}]_{1,1} &= \delta_0/\beta_0 \\
[\mathbf{C}]_{0,j} &= ((\gamma_0 - \alpha_{j-1})[\mathbf{C}]_{0,j-1} + \delta_0[\mathbf{C}]_{2,j-1} - \beta_{j-2}[\mathbf{C}]_{0,j-2})/\beta_{j-1} \\
[\mathbf{C}]_{i,j} &= (\delta_{i-1}[\mathbf{C}]_{i-1,j-1} + (\gamma_i - \alpha_{j-1})[\mathbf{C}]_{i,j-1} + \delta_i[\mathbf{C}]_{i+1,j-1} - \beta_{j-2}[\mathbf{C}]_{i,j-2})/\beta_{j-1}.
\end{aligned}
$$

Proposition 3.5 yields a natural algorithm for computing the connection coefficient matrix $\mathbf{C}_{\mu\to\nu}$. This algorithm is shown as Algorithm 3.3. Note that $\mathbf{C}$ is, by definition, upper triangular, so $[\mathbf{C}]_{i,j} = 0$ whenever $i > j$. We remark that for certain cases, particularly transforms between the Jacobi matrices of classical orthogonal polynomials, faster algorithms are known [TWO17]. We do not focus on such details in this paper as the cost of products with $\mathbf{A}$ is typically far larger than the cost of computing $\mathbf{C}_{\mu\to\nu}$.

Algorithm 3.3 Get connection coefficients

procedure GET-CONNECTION-COEFFS($\mu, k_\mu, k'_\mu, \nu, k_\nu, k'_\nu$)
  $[\mathbf{C}]_{0,0} = 1, [\mathbf{C}]_{i',j'} = 0$ if $i' > j'$ or $j' = -1$
  for $j = 1, 2, \dots, \min(k'_\mu, k_\nu + k'_\nu)$ **do**
    for $i = 0, 1, \dots, \min(j, k_\nu + k'_\nu - j)$ **do**
      $[\mathbf{C}]_{i,j} = (\delta_{i-1}[\mathbf{C}]_{i-1,j-1} + (\gamma_i - \alpha_{j-1})[\mathbf{C}]_{i,j-1} + \delta_i[\mathbf{C}]_{i+1,j-1} - \beta_{j-2}[\mathbf{C}]_{i,j-2})/\beta_{j-1}$
  return $\mathbf{C} = \mathbf{C}_{\mu\to\nu}$

Remark 3.6. From Proposition 3.5 it is not hard to see that $[\mathbf{C}]_{:k,:k}$ can be computed using $[\mathbf{M}(\nu)]_{:k,:k}$ and $[\mathbf{M}(\mu)]_{:k,:k}$. Moreover, $[\mathbf{C}]_{0,:2k+1}$ can be computed using $[\mathbf{M}(\nu)]_{:k+1,:k}$ and $[\mathbf{M}(\mu)]_{:2k,:2k}$. In general $\mathbf{M}(\mu)$ will be known fully, and in such cases, the modified moments through degree $2k$ can be computed using the information generated by Lanczos run for k iterations. △

We can use Algorithm 3.3 in conjunction with Algorithm 3.2 and Algorithm 1.1 to compute modified moments with respect to μ. This is shown in Algorithm 3.4 and Algorithm 3.5 respectively.

Algorithm 3.4 Get modified moments wrt. μ of weighed CESM (via Chebyshev moments)

procedure GET-MOMENTS-FROM-CHEB($\mathbf{A}, \mathbf{v}, s, \mu, a, b$)
  $k = \lceil s/2 \rceil$
  $\{n_i\}_{i=0}^{2k} =$ GET-CHEBYSHEV-MOMENTS($\mathbf{A}, \mathbf{v}, k, a, b$)
  $\mathbf{C} =$ GET-CONNECTION-COEFFS($[\mathbf{M}(\mu)]_{:2k,:2k}, [\mathbf{M}(\mu_{a,b}^T)]_{:2k,:2k}$)
  for $i = 0, 1, \dots, s$ **do**
    $m_i = \sum_{j=0}^{i} [\mathbf{C}]_{j,i} n_j$
  return $\{m_i\}_{i=0}^{s}$

Algorithm 3.5 Get modified moments wrt. μ of weighed CESM (via Lanczos)

procedure GET-MOMENTS-FROM-LANCZOS($\mathbf{A}, \mathbf{v}, s, \mu$)
  $k = \lceil s/2 \rceil$
  $[\mathbf{T}]_{i:k+1,:k} =$ LANCZOS($\mathbf{A}, \mathbf{v}, k$)
  $\mathbf{C} =$ GET-CONNECTION-COEFFS($[\mathbf{M}(\mu)]_{:2k,:2k}, [\mathbf{T}]_{:k+1,:k}$)
  for $i = 0, 1, \dots, s$ **do**
    $m_i = [\mathbf{C}]_{0,i}$
  return $\{m_i\}_{i=0}^{s}$

Remark 3.7. Given the modified moments of Ψ with respect to μ, the tridiagonal matrix produced by Algorithm 1.1 can itself be obtained [SD71]. This is quite similar to s-step Lanczos methods designed to reduce communication on distributed memory computers. However, if implemented naively, such methods can be even more susceptible to the effects of finite precision arithmetic than the regular Lanczos method [CD15; Car20], so special care must be taken when implementing such an algorithm. △

3.2 Quadrature approximations for weighted spectral measures

We now discuss how to use the information extracted by the algorithms in the previous section to obtain quadrature rules for the weighted CESM Ψ. We begin with a discussion on quadrature by interpolation in Section 3.2.1 followed by a

discussion on Guassian quadrature in Section 3.2.2 and quadrature by approximation in Section 3.2.3. Finally, in Section 3.2.4, we describe how damping can be used to ensure the positivity of quadrature approximations.

3.2.1 Quadrature by interpolation

Our first class of quadrature approximations for Ψ is the degree s quadrature by interpolation $[\Psi]_s^{\mathrm{iq}}$ (i.e. $\circ = \mathrm{i}$) which is defined by the relation

$$\int f \, \mathrm{d}[\Psi]_s^{\mathrm{iq}} := \int [f]_s^{\mathrm{ip}} \, \mathrm{d}\Psi, \tag{3.1}$$

where $[f]_s^{\mathrm{ip}}$ is the degree s polynomial interpolating a function f at the zeros $\{\theta_j^{(s+1)}\}_{j=0}^s$ of p_{s+1}, the degree $s+1$ orthogonal polynomial with respect to μ. (3.1) implies that

$$[\Psi]_s^{\mathrm{iq}} = \sum_{j=0}^{s} \omega_j \mathbb{1}[\theta_j^{(s+1)} \leq x]$$

where the weights $\{\omega_j\}_{j=0}^s$ are chosen such that the moments of $[\Psi]_s^{\mathrm{iq}}$ agree with those of Ψ through degree s.

One approach to doing this is by solving the Vandermonde-like linear system of equations

$$\begin{bmatrix} p_0(\theta_0^{(s+1)}) & \cdots & p_0(\theta_s^{(s+1)}) \\ \vdots & & \vdots \\ p_s(\theta_0^{(s+1)}) & \cdots & p_s(\theta_s^{(s+1)}) \end{bmatrix} \begin{bmatrix} \omega_0 \\ \vdots \\ \omega_s \end{bmatrix} = \begin{bmatrix} \int p_0 \, \mathrm{d}\Psi \\ \vdots \\ \int p_s \, \mathrm{d}\Psi \end{bmatrix}. \tag{3.2}$$

which we denote by $\mathbf{P}\boldsymbol{\omega} = \mathbf{m}$. This will ensure that polynomials of degree up to s are integrated exactly.

While it is not necessary to restrict the test polynomials to be the orthogonal polynomials $\{p_i\}_{i=0}^\infty$ with respect to μ nor the interpolation nodes to be the zeros $\{\theta_j^{(s+1)}\}_{j=0}^s$ of p_{s+1}, doing so has several advantages. If arbitrary polynomials are used, the matrix $\mathbf{P}$ may be exponentially ill-conditioned; i.e. the condition number of the matrix could grow exponentially in s. This can cause numerical issues with solving $\mathbf{P}\boldsymbol{\omega} = \mathbf{m}$. If orthogonal polynomials are used, then as in Theorem 2.14 we see that the columns of $\mathbf{P}$ are eigenvectors of the Jacobi matrix $\mathbf{M}$. Since $\mathbf{M}$ is symmetric, this implies that $\mathbf{P}$ has orthogonal columns; i.e. $\mathbf{P}^{\mathsf{H}}\mathbf{P}$ is diagonal. Therefore, we can easily apply $\mathbf{P}^{-1}$ through a product with $\mathbf{P}^{\mathsf{H}}$ and

an appropriately chosen diagonal matrix. In particular, if $\mathbf{S}$ is the orthonormal matrix of eigenvectors, then $\mathbf{P} = \mathbf{S}\,\mathrm{diag}([\mathbf{S}]_{0,:})^{-1}$ so that $\mathbf{P}^{-1} = \mathrm{diag}([\mathbf{S}]_{0,:})\mathbf{S}^{\mathsf{H}}$. This yields Algorithm 3.6.

Algorithm 3.6 Quadrature by interpolation

procedure GET-IQ($\{m_i\}_{i=0}^{s}, \mu$)
  $\boldsymbol{\theta}, \mathbf{S} = \mathrm{EIG}([\mathbf{M}(\mu)]_{:s+1,:s+1})$ ▷ Eigenvectors normalized to unit length
  $\boldsymbol{\omega} = \mathrm{diag}([\mathbf{S}]_{0,:})\mathbf{S}^{\mathsf{H}}\mathbf{m}$
  return $[\Psi]_s^{\mathrm{iq}} = \sum_{j=0}^{s} [\boldsymbol{\omega}]_j \mathbb{1}[[\boldsymbol{\theta}]_j \le x]$

Remark 3.8. In certain cases, such as $\mu = \mu_{a,b}^T$, $\mathbf{P}^{-1}$ can be applied quickly and stably using fast transforms, such as the discrete cosine transform, without ever constructing $\mathbf{P}$. △

3.2.2 Gaussian quadrature

While interpolation-based quadrature rules supported on k nodes do not, in general, integrate polynomials of degree higher than $k-1$ exactly, if we allow the nodes to be chosen adaptively we can do better. The degree $2k-1$ Gaussian quadrature rule $[\Psi]_{2k-1}^{\mathrm{gq}}$ for Ψ is obtained by constructing an quadrature by interpolation rule at the roots $\{\theta_i^{(k)}\}_{i=1}^{k}$ of the degree k orthogonal polynomial p_k of Ψ (i.e. by taking $\mu = \Psi$).

Theorem 3.9. *If p is any polynomial of degree at most $2k-1$, then*

$$\int p \,\mathrm{d}\Psi = \int p \,\mathrm{d}[\Psi]_{2k-1}^{\mathrm{gq}}.$$

I.e., the Gaussian quadrature rule integrates polynomials of degree $2k-1$ exactly.

Proof. We can decompose p as

$$p = qp_k + r$$

where q and r are each polynomials of degree at most $k-1$. Since p_k is the k-th orthogonal polynomial with respect to $\mu = \Psi$, it is orthogonal to all polynomials of lower degree, including q. Thus,

$$\int p \,\mathrm{d}\Psi = \int qp_k \,\mathrm{d}\Psi + \int r \,\mathrm{d}\Psi = \int r \,\mathrm{d}\Psi.$$

On the other hand, since the interpolation nodes $\{\theta_j^{(k)}\}_{j=0}^{k-1}$ are the roots of p_k,

$$\int p \,\mathrm{d}[\Psi]_{k-1}^{\mathrm{iq}} = \sum_{j=0}^{k-1} \omega_j \big(q(\theta_j^{(k)}) p_k(\theta_j^{(k)}) + r(\theta_j^{(k)})\big) = \sum_{j=0}^{k-1} \omega_j r(\theta_j^{(k)}) = \int r \,\mathrm{d}[\Psi]_{k-1}^{\mathrm{iq}}.$$

Since the quadrature rule $[\Psi]_k^{\mathrm{iq}}$ is interpolatory of degree k, this implies

$$\int p \,\mathrm{d}\Psi = \int p \,\mathrm{d}[\Psi]_{k-1}^{\mathrm{iq}}. \qquad \square$$

Because the polynomials $\{p_i\}_{i=0}^{\infty}$ are orthogonal with respect to the probability distribution Ψ function, we have that, for all $i \geq 0$,

$$m_i = \mathbf{v}^{\mathsf{H}} p_i(\mathbf{A})\mathbf{v} = \int p_i p_0 \,\mathrm{d}\Psi(\mathbf{A}, \mathbf{v}) = \mathbb{1}[i = 0].$$

This means the right hand side $\mathbf{m}$ of (3.2) is the first canonical unit vector $\mathbf{e}_0 = [1, 0, \ldots, 0]^{\mathsf{H}}$. Thus, as in Algorithm 3.6, $\boldsymbol{\omega} = \mathrm{diag}([\mathbf{S}]_{0,:})[\mathbf{S}]_{0,:}$; that is, the quadrature weights are the squares of the first components of the unit length eigenvectors of $[\mathbf{T}]_{:k,:k}$. We then arrive at Algorithm 3.7 for obtaining a Gaussian quadrature rule for $\Psi(\mathbf{A}, \mathbf{v})$ from the tridiagonal matrix $[\mathbf{T}]_{:k,:k}$ generated by Algorithm 1.1.

Algorithm 3.7 Gaussian quadrature

1: **procedure** GET-GQ($[\mathbf{T}]_{:k,:k}$)
2: $\boldsymbol{\theta}, \mathbf{S} = \mathrm{EIG}([\mathbf{T}]_{:k,:k})$ ▷ Eigenvectors normalized to unit length
3: $\boldsymbol{\omega} = \mathrm{diag}([\mathbf{S}]_{0,:})[\mathbf{S}]_{0,:}$
4: **return** $[\Psi]_{2k-1}^{\mathrm{gq}} = \sum_{j=0}^{k-1} [\boldsymbol{\omega}]_j \mathbb{1}[[\boldsymbol{\theta}]_j \leq x]$

Remark 3.10. To construct a Gaussian quadrature rule, the three term recurrence for the orthogonal polynomials of Ψ must be determined. Thus, the main computational cost is computing the tridiagonal matrix giving this recurrence. However, due to orthogonality, we know that all but the degree zero modified moments are zero and do not need to compute the moments. This is in contrast to other schemes where the polynomial recurrence is known but the modified moments must be computed. $\triangle$

3.2.3 Quadrature by approximation

Rather than defining a quadrature approximation using an interpolating polynomial, we might instead define an approximation $[\Psi]_s^{\mathrm{aq}}$ by the relation

$$\int f \mathrm{d}[\Psi]_s^{\mathrm{aq}} := \int [f]_s^{\mathrm{ap}} \, \mathrm{d}\Psi$$

where $[f]_s^{\mathrm{ap}}$ is the projection of f onto the orthogonal polynomials with respect to μ through degree s in the inner product $\langle \cdot, \cdot \rangle_\mu$. That is,

$$[f]_s^{\mathrm{ap}} := \sum_{i=0}^{s} \langle f, p_i \rangle_\mu \, p_i = \sum_{i=0}^{s} \left(\int f p_i \mathrm{d}\mu \right) p_i.$$

Expanding the integral of $[f]_s^{\mathrm{ap}}$ against Ψ,

$$\int [f]_s^{\mathrm{ap}} \, \mathrm{d}\Psi = \int \sum_{i=0}^{s} \left(\int f p_i \, \mathrm{d}\mu \right) p_i \, \mathrm{d}\Psi = \int f \left(\sum_{i=0}^{s} \left(\int p_i \, \mathrm{d}\Psi \right) p_i \right) \mathrm{d}\mu.$$

This implies

$$\frac{\mathrm{d}[\Psi]_s^{\mathrm{aq}}}{\mathrm{d}\mu} = \sum_{i=0}^{s} \left(\int p_i \, \mathrm{d}\Psi \right) p_i = \sum_{i=0}^{s} m_i p_i$$

where $\mathrm{d}[\Psi]_s^{\mathrm{aq}}/\mathrm{d}\mu$ is the Radon–Nikodym derivative of $[\Psi]_s^{\mathrm{aq}}$ with respect to μ.

Supposing[2] that the Radon–Nikodym derivative $\mathrm{d}\Psi/\mathrm{d}\mu$ exists, we observe

$$m_i = \int p_i \, \mathrm{d}\Psi = \int p_i \frac{\mathrm{d}\Psi}{\mathrm{d}\mu} \, \mathrm{d}\mu$$

is the μ-projection of $\mathrm{d}\Psi/\mathrm{d}\mu$ onto p_i for $i = 0, 1, \ldots, s$. Thus $\mathrm{d}\Psi/\mathrm{d}\mu$ is approximated in a truncated orthogonal polynomial series as

$$\sum_{i=0}^{s} \left(\int p_i \frac{\mathrm{d}\Psi}{\mathrm{d}\mu} \, \mathrm{d}\mu \right) p_i = \sum_{i=0}^{s} \left(\int p_i \, \mathrm{d}\Psi \right) p_i.$$

This means the density $\mathrm{d}[\Psi]_s^{\mathrm{aq}}/\mathrm{d}x$ is, at least formally, the polynomial approximation to the Radon–Nikodym derivative $\mathrm{d}\Psi/\mathrm{d}\mu$ times the density $\mathrm{d}\mu/\mathrm{d}x$; i.e. $\mathrm{d}[\Psi]_s^{\mathrm{aq}}/\mathrm{d}x = [\mathrm{d}\Psi/\mathrm{d}\mu]_s^{\mathrm{ap}}$.

[2] If $\Psi = \Psi(\mathbf{A}, \mathbf{v})$ then Ψ is not absolutely continuous with respect to the Lebesgue measure (or any equivalent measure) so the Radon–Nikodym derivative does not exist. However, there are absolutely continuous distribution distributions with the same modified moments as Ψ up to arbitrary degree, so conceptually one can use such a distribution instead.

We can obtain an approximation to the density $\mathrm{d}\Psi/\mathrm{d}x$ by using the density $\mathrm{d}\mu/\mathrm{d}x$ and the definition of the Radon–Nikodym derivative:

$$\frac{\mathrm{d}[\Psi]_s^{\mathrm{aq}}}{\mathrm{d}x} = \frac{\mathrm{d}[\Psi]_s^{\mathrm{aq}}}{\mathrm{d}\mu}\frac{\mathrm{d}\mu}{\mathrm{d}x} = \frac{\mathrm{d}\mu}{\mathrm{d}x}\sum_{i=0}^{s} m_i p_i.$$

Converting this "density" to a distribution gives the approximation $[\Psi]_s^{\mathrm{aq}}$ shown in Algorithm 3.8.

Algorithm 3.8 Quadrature by approximation

1: **procedure** GET-AQ($\{m_i\}_{i=0}^s, \mu$)
2: **return** $[\Psi]_s^{\mathrm{aq}} = \left(x \mapsto \sum_{i=0}^s m_i \int_{-\infty}^x p_i \,\mathrm{d}\mu\right)$

Remark 3.11. When $\mu = \mu_{a,b}^T$, $\mathrm{d}[\Psi]_s^{\mathrm{aq}}/\mathrm{d}\mu$ can be evaluated quickly at Chebyshev nodes by means of the discrete cosine transform. This allows the density $\mathrm{d}[\Psi]_s^{\mathrm{aq}}/\mathrm{d}x$ to be evaluated quickly at these points. △

Evaluating spectral sums and the relation to quadrature by interpolation

We have written the output of Algorithm 3.8 as a distribution function for consistency. However, note that

$$\int f \,\mathrm{d}[\Psi]_s^{\mathrm{aq}} = \sum_{i=0}^{s} m_i \int f p_i \,\mathrm{d}\mu.$$

Thus, if used for the task of spectral sum approximation, the distribution function $[\Psi]_s^{\mathrm{aq}}$ need not be computed. Rather, the μ-projections of f onto the orthogonal polynomials with respect to μ can be used instead. In many cases, the values of these projections are known analytically, and even if they are unknown, computing them is a scalar problem independent of the matrix size n.

A natural approach to computing the μ-projections of f numerically is to use a quadrature approximation for μ. Specifically, we might use the d-point Gaussian quadrature rule $[\mu]_{2d-1}^{\mathrm{gq}}$ for μ to approximate $\int f p_i \,\mathrm{d}\mu$. This gives us the approximation

$$\sum_{i=0}^{s} m_i \int f p_i \,\mathrm{d}\mu \approx \sum_{i=0}^{s} m_i \int f p_i \,\mathrm{d}[\mu]_{2s+1}^{\mathrm{gq}} = \sum_{i=0}^{s} m_i \sum_{j=0}^{d-1} \omega_i^{(d)} f(\theta_j^{(d)}) p_i(\theta_j^{(d)})$$

where $\{\omega_i^{(d)}\}_{i=0}^{d-1}$ are the Gaussian quadrature weights for μ.

Similar to above, denote by $\mathbf{P}$ the $d \times d$ Vandermonde-like matrix of orthogonal polynomials with respect to μ evaluated at the zeros of p_{d+1}. If $\mathbf{S}$ is the orthonormal matrix of eigenvectors of the $d \times d$ Jacobi matrix $[\mathbf{M}]_{:d,:d}$ for μ, then recall that the Gaussian quadrature weights $\boldsymbol{\omega}^{(d)}$ are given by $\operatorname{diag}([\mathbf{S}]_{0,:s+1})([\mathbf{S}]_{:s+1,:})^{\mathsf{H}}$. This yields Algorithm 3.9.

Remark 3.12. In the case $d = s+1$, Algorithm 3.9 is equivalent to Algorithm 3.6. △

Algorithm 3.9 Approximate quadrature by approximation

1: **procedure** GET-AAQ($\{m_i\}_{i=0}^{s}, d, \mu$)
2: $\quad\boldsymbol{\theta}, \mathbf{S} = \text{EIG}([\mathbf{M}(\mu)]_{:d,:d})$ ▷ Eigenvectors normalized to unit length
3: $\quad\boldsymbol{\omega} = \operatorname{diag}([\mathbf{S}]_{0,:s+1})([\mathbf{S}]_{:s+1,:})^{\mathsf{H}}\mathbf{m}$
4: $\quad$**return** $[\Psi]_s^{\text{iq}} = \sum_{j=0}^{k-1}[\boldsymbol{\omega}]_j \mathbb{1}[[\boldsymbol{\theta}]_j \le x]$

3.2.4 Positivity by damping and the kernel polynomial method

While it is clear that that the Gaussian quadrature $[\Psi]_s^{\text{gq}}$ is a non-negative probability distribution function, neither $[\Psi]_s^{\text{iq}}$ nor $[\Psi]_s^{\text{aq}}$ are guaranteed to be weakly increasing. We now discuss how to use damping to enforce positivity.

Towards this end, define the damping kernel,

$$P_x(y) = \sum_{i=0}^{s} \rho_i p_i(x) p_i(y)$$

where $\{\rho_i\}_{i=0}^{s}$ are *damping coefficients* as in Theorem 2.18. Then the damped interpolant $[f]_s^{\text{d-ip}}$ and approximant $[f]_s^{\text{d-ap}}$ can be written in terms of P_x as

$$[f]_s^{\text{d-ip}}(x) = \int P_x f \, \mathrm{d}[\mu]_{2s+1}^{\text{gq}} \qquad \text{and} \qquad [f]_s^{\text{d-ap}}(x) = \int P_x f \, \mathrm{d}\mu.$$

Remark 3.13. If $\rho_i = 1$ for all i, then $[f]_s^{\text{d-ip}} = [f]_s^{\text{ip}}$ and $[f]_s^{\text{d-ap}} = [f]_s^{\text{ap}}$. △

These approximations induce $[\Psi]_s^{\text{d-iq}}$ and $[\Psi]_s^{\text{d-aq}}$ by

$$\int f \, \mathrm{d}[\Psi]_s^{\text{d-iq}} := \int [f]_s^{\text{d-ip}} \, \mathrm{d}\Psi \qquad \text{and} \qquad \int f \, \mathrm{d}[\Psi]_s^{\text{d-aq}} := \int [f]_s^{\text{d-ap}} \, \mathrm{d}\Psi.$$

Algorithmically, this is equivalent to replacing m_i by $\rho_i m_i$ in the expressions for quadrature by interpolation and approximation described in Sections 3.2.1 and 3.2.3.

Lemma 3.14. *If $\rho_0 = 1$, then $[\Psi]_s^{\text{d-iq}}$ and $[\Psi]_s^{\text{d-aq}}$ have unit mass, and if $P_x(y) \geq 0$ for all x, y, then $[\Psi]_s^{\text{d-iq}}$ and $[\Psi]_s^{\text{d-aq}}$ are weakly-increasing.*

Proof. The first part of the theorem follows from the fact that

$$\int P_x \mathrm{d}\mu = \sum_{i=0}^{s} \rho_i p_i(x) \int p_i \mathrm{d}\mu = \rho_0.$$

To prove the remainder, suppose f is non-negative. Then clearly

$$[f]_s^{\text{d-ap}}(x) = \int P_x f \, \mathrm{d}\mu \geq 0$$

so that $\int f \, \mathrm{d}[\Psi]_s^{\text{d-aq}} \geq 0$. A similar argument implies $\int f \, \mathrm{d}[\Psi]_s^{\text{d-iq}} \geq 0$ as well. Therefore the approximations are weakly increasing. □

Remark 3.15. While we have describe the damping procedure in terms of the damped interpolant and approximant, an equilvalent perspective is that we first smooth the distribution Ψ with the damping kernel $P_t(y)$ and subsequently use quadrature by interpolation or approximation with this new distribution function. △

One particularly important damping kernel for the case $\mu = \mu_{a,b}^T$ is given by the Jackson coefficients defined in Theorem 2.26. The associated damping kernel was used in the original proof of Jackson's theorem [Jac12] and leads to the Jackson damped KPM approximation, which is the most popular KPM variant [Wei+06; BKM22]. The rate of convergence of polynomial approximations using these damping coefficients is estimated in Theorem 2.27 below. For a discussion on other damping schemes we refer readers to [Wei+06; LSY16].

3.3 A priori error bounds on an interval

Most of the quadrature approximations we consider have the property that they integrate polynomials exactly. In this case, we can use this property to reduce

error bounds to the quality of the best uniform polynomial approximations to f, which we discussed in Section 2.3.

Lemma 3.16. *Suppose Υ_1 and Υ_2 are probability distribution functions whose moments are equal through degree s, each constant on $(-\infty, a)$ and (b, ∞). Then,*

$$\left| \int f \, \mathrm{d}\Upsilon_2 - \int f \, \mathrm{d}\Upsilon_2 \right| = (d_{\mathrm{TV}}(\Upsilon_1) + d_{\mathrm{TV}}(\Upsilon_2)) \min_{\deg(p) \leq s} \|f - p\|_{[a,b]}.$$

Proof. Let p be any polynomial of degree at most s and note that $\int p \, \mathrm{d}\Upsilon_1 = \int p \, \mathrm{d}\Upsilon_2$. Then, applying the triangle inequality,

$$\begin{aligned}
\left| \int f \, \mathrm{d}\Upsilon_1 - \int f \, \mathrm{d}\Upsilon_2 \right| &= \left| \int (f - p) \, \mathrm{d}\Upsilon_1 - \int (f - p) \, \mathrm{d}\Upsilon_2 \right| \\
&\leq \int |f - p| |\mathrm{d}\Upsilon_1| + \int |f - p| |\mathrm{d}\Upsilon_2| \\
&\leq \int \|f - p\|_{[a,b]} |\mathrm{d}\Upsilon_1| + \int \|f - p\|_{[a,b]} |\mathrm{d}\Upsilon_2| \\
&= (d_{\mathrm{TV}}(\Upsilon_1) + d_{\mathrm{TV}}(\Upsilon_2)) \|f - p\|_{[a,b]}.
\end{aligned}$$

The result follows by optimizing over p. □

Lemma 3.16 shows that the Lanczos-based Gaussian quadrature approximations always perform within a factor of two of the best polynomial approximation on $\mathcal{I}$. Intuitively, approaches based on explicit polynomial approximation will have error roughly equal to $\|f - [f]_s^{\circ \mathrm{p}}\|_{\mathcal{I}}$ as a large portion of mass of Ψ is likely in regions where $|f - [f]_s^{\circ \mathrm{p}}|$ is large. Thus, Gaussian quadrature should not be expected to perform significantly worse than explicit polynomial methods, at least in exact arithmetic. In fact, as we will discuss in Chapter 8, even in finite preicsion arithmetic, Lemma 3.16 is still morally correct.

For some quadrature approximations we consider, polynomials are not integrated exactly. In such cases, we turn to the following bound:

Lemma 3.17. *Suppose Υ_1 is a probability distribution function and Υ_2 is defined by $\int f \, \mathrm{d}\Upsilon_2 = \int \mathcal{O}[f] \mathrm{d}\Upsilon_1$ for some operator $\mathcal{O}[\,\cdot\,]$. Then, for any f,*

$$\left| \int f \, \mathrm{d}\Upsilon_1 - \int f \, \mathrm{d}\Upsilon_2 \right| = \|f - \mathcal{O}[f]\|_{[a,b]} d_{\mathrm{TV}}(\Upsilon_1)$$

Proof. The result follows by a simple application of the triangle inequality:

$$\left| \int f \, \mathrm{d}\Upsilon_1 - \int f \, \mathrm{d}\Upsilon_2 \right| = \left| \int (f - \mathcal{O}[f]) \, \mathrm{d}\Upsilon_1 \right| \leq \int \|f - \mathcal{O}[f]\|_{[a,b]} |\mathrm{d}\Upsilon_1|$$

$$= \|f - \mathcal{O}[f]\|_{[a,b]} d_{\mathrm{TV}}(\Upsilon_1). \qquad \square$$

The primary downside of this bound is that it requires being able to bound $\|f - \mathcal{O}[f]\|_{[a,b]}$. However, at least in the case that $\mathcal{O}$ corresponds to undamped or Jackson damped Chebyshev interpolation or approximation, we are able to derive bounds for $\|f - \mathcal{O}[f]\|_{[a,b]}$ directly.

3.4 Qualitative comparison of algorithms

In Sections 1.1 and 3.1 we described Algorithm 3.1 (GET-MOMENTS), Algorithm 3.2 (GET-CHEBYSHEV-MOMENTS), Algorithm 1.1 (LANCZOS), Algorithm 3.4 (GET-MOMENTS-FROM-CHEB), and Algorithm 3.5 (GET-MOMENTS-FROM-LANCZOS) which are used to compute the information required for the quadrature approximations described in Section 3.2 Since Algorithms 3.4 and 3.5 respectively call Algorithms 1.1 and 3.2, Algorithms 1.1, 3.1 and 3.2 constitute the bulk of the computational cost of all implementations of the protoalgorithm discussed in this paper.

In each iteration, Algorithms 1.1, 3.1 and 3.2 each require one matrix vector product with $\mathbf{A}$ along with several scalar multiplications, vector additions, and inner products. As such, the total computational cost of each algorithms is $O(k(T_{\mathrm{mv}} + n))$ where k is the number of iterations and T_{mv} is the cost of a matrix-vector product with $\mathbf{A}$. Here we ignore terms depending only on k (e.g. k^2) which are unimportant if we assume $k \ll n$. Each of the algorithms can also be implemented using just $O(n)$ storage; i.e. without storing the entire basis for the Krylov subspace which would cost $O(kn)$ storage.

While the algorithms are typically quite storage efficient, there are some situations in which it may be desirable to store the whole Krylov basis. First, Algorithm 1.1 is sometimes run with full reorthogonalization. This can improve numerical stability, but increases the computation cost to $O(k(T_{\mathrm{mv}} + kn))$. Next, by delaying all inner products to a final iteration (or using a non-blocking implementation), the number of global reductions required by Algorithm 3.1 and Algorithm 3.2 can be reduced. Since global communication can significantly slow down Krylov subspace methods, this may speed up computation on highly parallel machines [DHL15; Ber+08]. As mentioned earlier, there are

implementations of the Lanczos algorithm which aim to decrease the number of global communications [CD15; Car20]. Designing Krylov subspace methods for avoiding or reducing communication costs is a large area study, but further discussion is outside the scope of this paper.

3.5 Numerical experiments

In this section, we provide a range of numerical experiments to illustrate the behavior of the algorithms described above as well as the tradeoffs between algorithms. Our focus is primarily on quadrature approximations of the weighted CESM, as the approximation of the true CESM by the average of weighted CESMs is straightforward and well understood.

3.5.1 Comparison with classical quadrature

We begin with an example designed to illustrate some of the similarities and differences between the behavior of classical quadrature rules for continuous weight functions and the behavior of the matrix-free quadrature algorithms presented in this paper.

Throughout this example, we use the Runge function $f(x) = 1/(1 + 16x^2)$ and a vector $\mathbf{v}$ with uniform weight on each eigencomponent. We will compare the effectiveness of the Gaussian quadrature rule $[\Psi]^{\text{gq}}_{2k-1}$, the quadrature by interpolation rule $[\Psi]^{\text{iq}}_{2k}$ and the quadrature by approximation rule $[\Psi]^{\text{aq}}_{2k}$. For the latter approximations, we set $\mu = \mu^T_{-1,1}$, and for the quadrature by approximation rule, we use Algorithm 3.9 with enough quadrature nodes so that the involved integrals are computed to essentially machine precision. All three approximations can be computing using k matrix-vector products with $\mathbf{A}$, and since the approaches exactly integrate polynomials of degree $2k - 1$ and $2k$ respectively, we might expect that them to behave similarly. However, there are a variety of factors which prevent this from being the case.

In our first experiment, shown in Figure 3.1, the spectrum of $\mathbf{A}$ uniformly fills out the interval $[-1, 1]$; i.e., $\lambda_i = -1 + (2i + 1)/n$, $i = 0, 1, \dots, n - 1$. We take $n = 10^5$ so that $[\Psi]^{\text{gq}}_{2k-1}$ and $[\Psi]^{\text{iq}}_{2k}$ respectively approximate the k-point Gaussian quadrature and $(2k - 1)$-point Fejér quadrature rules for a uniform weight on

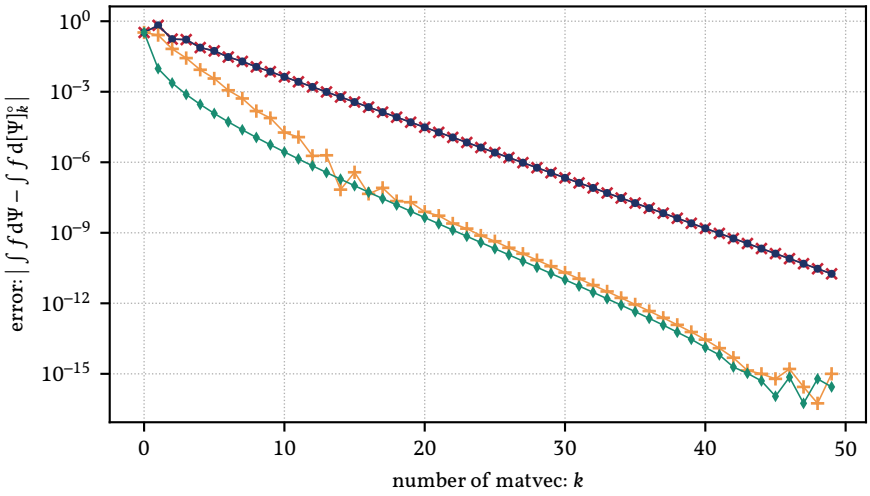


Figure 3.1: Errors for approximating $\int f\,d\Psi = \mathbf{v}^H f(\mathbf{A})\mathbf{v}$ when $f(x) = 1/(1 + 16x^2)$ for a spectrum uniformly filling $[-1, 1]$. *Legend*: Gaussian quadrature with (—•—) and without (—×—) reorthogonalization, quadrature by interpolation (—+—), and approximate quadrature by approximation (—♦—). *Takeaway*: Intuition about classical approximation theory informs our understanding of algorithms for matrix-free quadrature. In fact, in some cases, quadrature by interpolation or approximation can provably outperform Gaussian quadrature.

$[-1, 1]$. For many functions, certain quadrature by interpolation rules on $[-1, 1]$, including the Fejér rule, behave similarly to the Gaussian quadrature rule when the same number of nodes are used [Tre08]. For $f(x) = 1/(1 + 16x^2)$, this phenomenon is observed for some time until the convergence rate is abruptly cut in half [WT07]. In our setting, a fair comparison means that the number of matrix-vector products are equal, so we see that the quadrature by interpolation approximation initially converges twice as quickly as the Gaussian quadrature approximation! The rate of the quadrature by interpolation approximation is eventually cut in half to match the rate of the Gaussian quadrature approximation.

In our second experiment, shown in Figure 3.2, the spectrum of $\mathbf{A}$ uniformly fills out the disjoint intervals $[-1, -0.75] \cup [0.75, 1]$ with the same inter-point spacing as the first example; i.e. we remove the eigenvalues in the previous

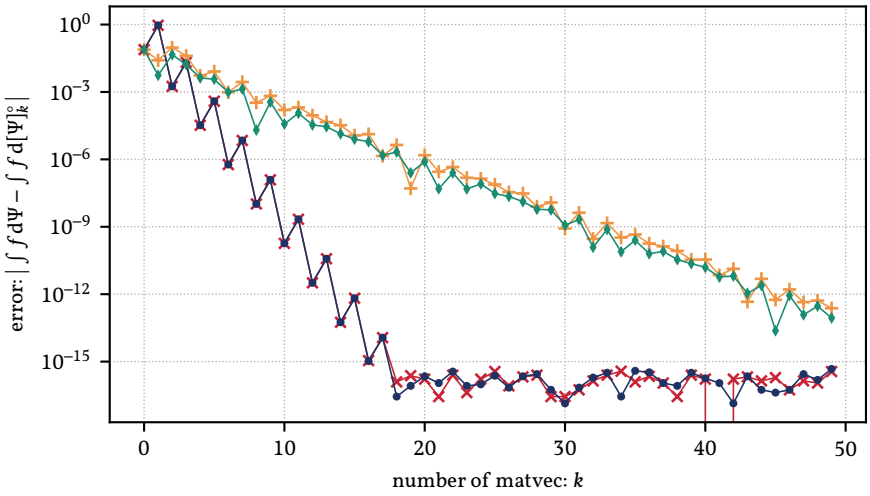


Figure 3.2: Errors for approximating $\int f\,d\Psi = \mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ when $f(x) = 1/(1+16x^2)$ for a spectrum uniformly filling $[-1,1]$ except for a gap around zero. *Legend*: Gaussian quadrature with (—●—) and without (—×—) reorthogonalization, quadrature by interpolation (—+—), and approximate quadrature by approximation (—♦—). *Takeaway*: The behavior of the algorithms are highly dependent on the eigenvalue distribution of $\mathbf{A}$, and Gaussian quadrature may perform significantly better that explicit methods when the spectrum of $\mathbf{A}$ has additional structure such as gaps.

example which fall between –0.75 and 0.75. Here we observe that the Gaussian quadrature rule converges significantly faster than in the previous experiment. This to be expected. Indeed, the Gaussian quadrature rule has its nodes near $[-1,-0.75] \cup [0.75,1]$, so the union of the support of Ψ and $[\Psi]_{2k}^{\text{iq}}$ is further from the poles of f located at $\pm\boldsymbol{i}/4$. We also note that the conditions which enabled accelerated convergence in the first experiment are no longer present, so the quadrature by interpolation approximation converges at its limiting rate [Tre08].

In the both experiments, the Lanczos based Gaussian quadrature approach behaves similar with and without reorthogonalization. In fact, it is easily verified that the Lanczos algorithm does not lose orthogonality and behaves nearly the same regardless of whether or not reorthogonalization is used. To the best of

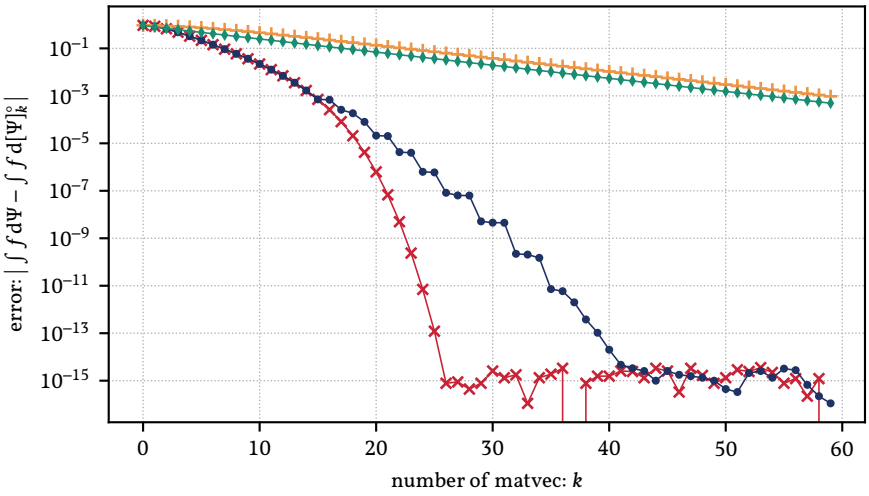


Figure 3.3: Errors for approximating $\int f \, d\Psi = \mathbf{v}^H f(\mathbf{A})\mathbf{v}$ when $f(x) = 1/x$ for model problem. *Legend*: Gaussian quadrature with (—●—) and without (—×—) reorthogonalization, quadrature by interpolation (—+—), and approximate quadrature by approximation (—◆—). *Takeaway*: Without reorthogonaliztion the convergence of Gaussian quadrature is slowed. However, the method still converges and can even outperform the other methods.

our knowledge, such a result has not been proved rigorously.

3.5.2 Finite precision convergence

In this example, we consider several experiments where orthogonality is lost and the effects of finite precision arithmetic are easily observed. In both experiments we use diagonal matrices scaled so $\|\mathbf{A}\|_2 = 1$ and set $\mathbf{v}$ to have uniform entries. We set a, b as the largest and smallest eigenvalues respectively and again use $\mu = \mu_{a,b}^T$ for the interpolatory and quadrature by approximations.

In the first experiment, shown in Figure 3.3, the eigenvalues of $\mathbf{A}$ are distributed according to the model problem (10.1) and $f(x) = 1/x$. Specifically, the eigenvalues are given by the model problem with selected parameters $n = 300, \kappa = 10^3$, and $\rho = 0.85$. In the second experiment, shown in the right panel Figure 3.4, we use the $n = 9664$ eigenvalues of the California matrix from the sparse matrix

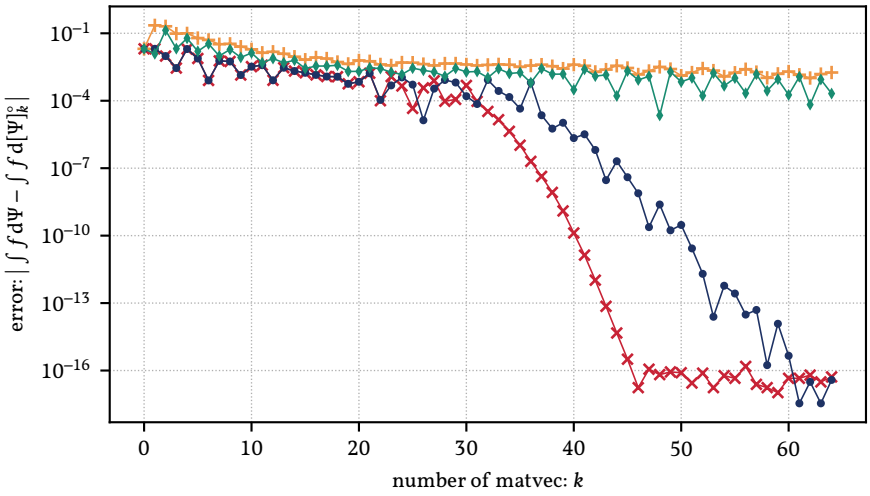


Figure 3.4: Errors for approximating $\int f \, d\Psi = \mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ when $f(x) = \mathbb{1}[x > c]$ for MNIST covariance matrix. *Legend*: Gaussian quadrature with (—●—) and without (—×—) reorthogonalization, quadrature by interpolation (—+—), and approximate quadrature by approximation (—♦—). *Takeaway*: Without reorthogonaliztion the convergence of Gaussian quadrature is slowed. However, the method still converges and can even outperform the other methods.

suite [DH11] and the function $f(x) = |x|$.

In both cases, the Jacobi matrices produced by Lanczos, with or without reorthogonalization, differ greatly; i.e. the difference of the matrices is on the order of $\|\mathbf{A}\|_2$. Even so, the modified moments for $\mu = \mu_{a,b}^T$ obtained by Algorithms 3.4 and 3.5 differ only in the 12th digit and 14th digits respectively. Using one approach in place of the other does not noticeably impact the convergence of the quadrature by interpolation approximations.

Chapter 4
Spectrum and spectral sum approximation

We now turn to the tasks of spectrum and spectral sum approximation. Specifically, we will show how the algorithms from the previous chapter can be used to produce approximations to the CESM Φ, the probability distribution function with unit mass at each eigenvalue of $\mathbf{A}$ which we defined in Theorem 1.2. This in turn induces approximations to $\operatorname{tr}(f(\mathbf{A}))$.

Towards this end, suppose $\mathbf{v}$ is a random vector satisfying $\mathbb{E}[\mathbf{v}\mathbf{v}^{\mathsf{H}}] = n^{-1}\mathbf{I}$; i.e., $\mathbf{v}$ is isotropic with appropriate scale. Then, using basic properties of the trace and expectation for any $x \in \mathbb{R}$, we have

$$\begin{aligned}\mathbb{E}[\Psi(x)] = \mathbb{E}[\mathbf{v}^{\mathsf{H}}\mathbb{1}[\mathbf{A} \preceq x]\mathbf{v}] &= \mathbb{E}[\operatorname{tr}(\mathbf{v}^{\mathsf{H}}\mathbb{1}[\mathbf{A} \preceq x]\mathbf{v})] \\ &= \mathbb{E}[\operatorname{tr}(\mathbb{1}[\mathbf{A} \preceq x]\mathbf{v}\mathbf{v}^{\mathsf{H}})] = \operatorname{tr}(\mathbb{E}[\mathbb{1}[\mathbf{A} \preceq x]\mathbf{v}\mathbf{v}^{\mathsf{H}}]) \\ &= \operatorname{tr}(\mathbb{1}[\mathbf{A} \preceq x]\mathbb{E}[\mathbf{v}\mathbf{v}^{\mathsf{H}}]) = n^{-1}\operatorname{tr}(\mathbb{1}[\mathbf{A} \preceq x]) = \Phi(x).\end{aligned}$$

That Ψ is an unbiased estimator for Φ at every point $x \in \mathbb{R}$ is illustrated in Figure 4.1. Further, almost by definition, we see that $\int f \, \mathrm{d}\Psi = \mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ is an unbiased estimator for $n^{-1}\operatorname{tr}(f(\mathbf{A}))$.

Let $\{\Psi_\ell\}_{\ell=0}^{n_{\mathrm{v}}-1}$ be independent and identically distributed (iid) copies of the weighted CESM Ψ corresponding to vectors $\{\mathbf{v}_\ell\}_{\ell=0}^{n_{\mathrm{v}}-1}$ which are iid copies of $\mathbf{v}$. Then the averaged weighted CESM

$$\langle \Psi_\ell \rangle := n_{\mathrm{v}}^{-1} \sum_{\ell=0}^{n_{\mathrm{v}}-1} \Psi_\ell$$

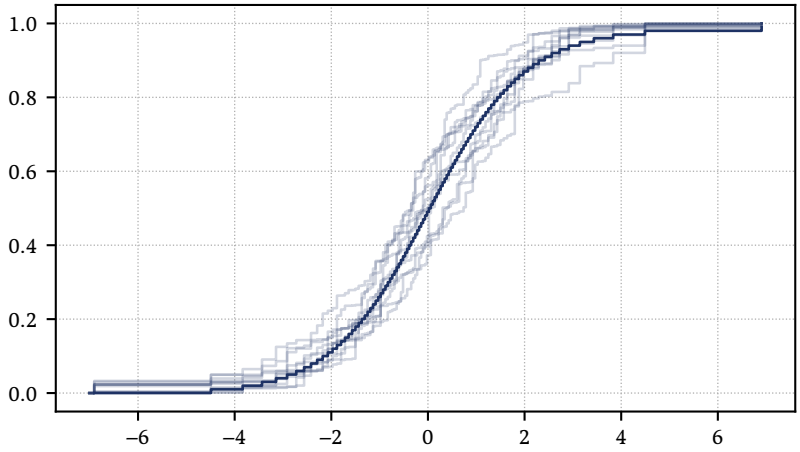


Figure 4.1: CESM Φ (——) and 10 independent samples of weighted CESM Ψ corresponding to random $\mathbf{v}$ (——). Each copy of Ψ is an unbiased estimator for Φ at each point $x \in \mathbb{R}$.

is also an unbiased estimator for the CESM at every point x. This implies

$$\left\langle \mathbf{v}_\ell^{\mathsf{H}} f(\mathbf{A}) \mathbf{v}_\ell \right\rangle := n_{\mathrm{v}}^{-1} \sum_{\ell=0}^{n_{\mathrm{v}}-1} \mathbf{v}_\ell^{\mathsf{H}} f(\mathbf{A}) \mathbf{v}_\ell = \int f \, \mathrm{d}\langle \Psi_\ell \rangle,$$

is an unbiased estimator for $n^{-1} \operatorname{tr}(f(\mathbf{A}))$. In both cases, the standard deviation of the averaged estimator decreases proportional to $1/\sqrt{n_{\mathrm{v}}}$, so the averaged estimators concentrate more sharply about the mean as n_{v} increases.

We refer to estimators of the form $\mathbf{v}^{\mathsf{H}}\mathbf{B}\mathbf{v}$, where $\mathbf{v}$ is an isotropic random vector, as quadratic trace estimators. Thus, we see that the quadratic trace estimator for the spectral sum $\operatorname{tr}(f(\mathbf{A}))$ is an integral against the weighted CESM, which is itself a quadratic trace estimator for the CESM at every point x. Moreover, since the quadratic form $\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ can be written as an integral of f against the weighted CESM, classical results about the convergence of quadrature rules for approximating this integral can be leveraged to obtain error estimates for the convergence of our Krylov subspace approximations of $\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$.

In order to approximate a sample of Ψ, and therefore integrals against such samples, we can simply use the algorithms from the previous chapter. Thus, we arrive at a prototypical algorithm for spectrum and spectral sum approx-

imation,Algorithm 4.1. The output $\langle [\Psi_\ell]_s^{\circ q} \rangle$ of Algorithm 4.1 is a distribution function which approximates the CESM Φ. For any function $f : \mathbb{R} \to \mathbb{R}$, this approximation naturally yields an approximation to the spectral sum $\operatorname{tr}(f(\mathbf{A}))$ by integration.

Algorithm 4.1 Prototypical randomized spectrum and spectral sum approximation

procedure SPEC-APPROX($\mathbf{A}, n_{\mathrm{v}}, k, \circ$)
  for $\ell = 0, 1, \dots, n_{\mathrm{v}} - 1$ **do**
    define (implicitly) $\Psi_\ell \overset{\text{iid}}{\sim} \Psi$ by sampling $\mathbf{v}_\ell \overset{\text{iid}}{\sim} \mathbf{v}$, $\mathbb{E}[\mathbf{v}\mathbf{v}^{\mathsf{H}}] = n^{-1}\mathbf{I}$
    compute moments of Ψ_ℓ through degree s by constructing $\mathcal{K}_{k+1}(\mathbf{A}, \mathbf{v}_\ell)$
    approximate Ψ_ℓ by $[\Psi_\ell]_s^{\circ q}$ induced by a polynomial operator $[\,\cdot\,]_s^{\circ p}$
  return $\langle [\Psi_\ell]_s^{\circ q} \rangle := n_{\mathrm{v}}^{-1} \sum_{\ell=0}^{n_{\mathrm{v}}-1} [\Psi_\ell]_s^{\circ q}$

4.1 Related work and context

Specific implementations of the prototypical algorithm, given in Algorithm 4.1, are by far the most common algorithms for spectral sum and spectrum approximation, and they have found widespread use in a range of disciplines [LSY16; UCS17]. As we have alluded to, the two key ingredients for such algorithms are (i) polynomial approximation and quadrature and (ii) quadratic trace estimation. The first of these ingredients has been studied for centuries [Tre19], so the popularization of the latter [Gir87; Hut89; Ski89] quickly lead to a variety algorithms fitting this framework. In this section we focus primarily on conceptual and theoretical advancements relating to the protoalgorithm. We hope our brief review of prior work will help tie together several clusters of literature which have remained largely disjoint.

Both [Gir87; Hut89] focus on estimating the trace of a large implicit matrix $\mathbf{B} = \mathbf{A}^{-1}$ for some matrix $\mathbf{A}$. While [Gir87] suggests the use of the conjugate gradient algorithm, neither paper discusses in detail how to approximate products with $\mathbf{B}$. Therefore, to the best of our knowledge, [Ski89] contains the first example of an algorithm which truly fits into the form considered in this paper. In [Ski89], an approximation to Φ based on an expansion in Chebyshev polynomials is described. This approximation is then used to approximate $\operatorname{tr}(\ln(\mathbf{A})) = \ln \det(\mathbf{A})$.

The Chebyshev based approach of [Ski89] was improved in [SR94] where a damping Kernel was introduced to avoid Gibbs oscillations. The connection to Jackson's damping and other classical ideas from approximation theory were subsequently considered in [Sil+96]. The resulting algorithm is now typically called the kernel polynomial method (KPM) and is widely used in the computational physical sciences; see [Wei+06] for a review.

Essentially in parallel, stochastic trace estimation was combined with quadrature explicitly. Typically, such approaches are based on the Lanczos algorithm which can be used in a straightforward way to compute certain quadrature rules for Ψ [GM09, Chapter 6], [Gau06]. In [BFG96], Gauss, Gauss-Lobatto, and Gauss-Radau quadrature rules are used to derive upper and lower bounds for $\int f \, \mathrm{d}\Psi$ when $f(x) = 1/x$ or $f = \ln(x)$. These bounds were in turn combined with stochastic trace estimation to provide probabilistic upper and lower bounds on the traces of the corresponding matrix functions. The Gaussian quadrature based approach is now typically referred to as stochastic Lanczos quadrature (SLQ).

Bounds on the number of samples n_{v} required so that the average of iid quadratic trace estimators is within ϵ of the true trace with at least probability $1 - \eta$ were derived in [AT11] and subsequently improved on in [RA14]. These bounds enabled a range of analyses which explicitly balanced the number of samples n_{v} with the approximation degree s. For instance, [Han+17] and [UCS17] respectively consider approximation of spectral sums corresponding to analytic functions by a Chebyshev based approach and SLQ. Later, [CK21] gives stronger bounds for quadratic trace estimators, and as in [UCS17], these bounds are used to analyze SLQ.

Around this time, spectrum approximation in Wasserstein distance was analyzed for KPM [BKM22] and SLQ [CTU21]. We remark that [BKM22; CTU21] both arrive at the conclusion that the number of samples required to approximate Φ in Wasserstein distance to accuracy ϵ actually *decreases* as the matrix size n increases, provided $\epsilon \gg n^{-1/2}$ as $n \to \infty$. While not stated explicitly, the analysis in [CK21] implies this same fact for the number of samples required to approximate $\int f \, \mathrm{d}\Phi = n^{-1} \operatorname{tr}(f(\mathbf{A}))$ to additive error $\pm\epsilon$. This fact was already known to the physics community [Wei+06], although, to the best of our knowledge, was not proved rigorously in the literature.

4.1.1 Note on history of stochastic quadratic trace estimators and their analysis

What we are calling a quadratic trace estimator is often called the *Hutchinson's trace estimator*, especially when $\mathbf{v}$ is chosen uniformly from the set of vectors with entries $\pm n^{-1/2}$. However, [Hut89] was not the first use of quadratic trace estimators for the task of approximating the trace of an implicit matrix; [Hut89] itself cites [Gir87] which addresses the same task by using samples of $\mathbf{v}$ drawn uniformly from the unit hypersphere. Algorithms based on the use of random vectors back at least to the mid 1970s [Alb+75; WW76; WW77; RV89].

In fact, such estimators are a special case of the concept of *typicality* in quantum physics. Typicality has its origins in work of Schrödinger [Sch27] and von Neumann [Neu29] from the late 1920s but was dismissed and/or forgotten until a resurgence in the mid 2000s [GMM09; Gol+06; PSW06; Rei07]; see [Gol+10] for a historical overview and discussion in a modern context and [Jin+21] for a review of algorithms based on typicality.

Likewise, while the first tail bounds for quadratic trace estimators are typically attributed to [AT11; RA14], quadratic trace estimators were analyzed before either of these papers. For instance, [Rei07] provides tail bounds based on Chebyshev's inequality for quadratic trace estimators used for the specific purpose of estimating the trace of a symmetric matrix. Sub-Gaussian concentration inequalities for quadratic trace estimators, similar to those in [AT11; RA14] are derived in [PSW06] using Levy's Lemma, a general result about concentration of measure [Led01]; see also [Gog10, Theorem 2.2.2].

There are also many earlier analyses of quadratic trace estimators outside of the specific context of trace estimation. For instance, [HW71] provides concentration bounds for quadratic trace estimators when the entries of $\mathbf{v}$ are independent symmetric sub-Gaussian random variables. In fact, some of the strongest bounds for quadratic trace estimators [Mey+21; PCK22] make use of so called *Hanson–Wright inequalities* [RV13] introduced in [HW71]. Earlier still, [GPS59] states as fact that the expectation of such estimators, when $\mathbf{v}$ has iid Gaussian entries, is the sum of the eigenvalues of the matrix in question, citing a book [Cra46] from the 1940s.

4.1.2 Other randomized trace estimation algorithms

As a consequence of the central limit theorem, the average of iid samples of quadratic trace estimators requires $O(\epsilon^{-2})$ samples to reach accuracy ϵ. In fact, any algorithm which returns a linear combination of estimators depending on vectors drawn independently of $\mathbf{A}$ requires $O(\epsilon^{-2})$ samples to obtain an approximation of the trace accurate to within a multiplicative factor $1 \pm \epsilon$ [WWZ14]. A number of papers aim to avoid this dependence on the number of samples by incorporating low-rank approximation to $f(\mathbf{A})$ [Lin16; GSO17; SAI17; Ada+18; LZ21; Mey+21; PCK22; CH22].

In [Mey+21] algorithm called Hutch++ in introduced and proved to output an estimate the trace of a positive definite matrix to relative error $1 \pm \epsilon$ using just $O(\epsilon^{-1})$ matrix-vector products. It is also shown that this ϵ dependence is nearly optimal in certain matrix-vector query models. The practicality of Hutch++ was improved in [PCK22] which describes a variant which outputs an (ϵ, δ) approximation to the trace. Such methods can be used to compute the trace of matrix functions by computing products with $f(\mathbf{A})$ (e.g. using black-box Krylov subspace methods).

A so-called *Krylov-aware* approach to estimating the trace of matrix functions was introduced in [CH22]. Rather than treating products with $f(\mathbf{A})$ as a black-box, [CH22] advocates a more careful approach in which products with $\mathbf{A}$ are viewed as the natural computational primative. This allows several efficincies not present in black-box versions of Hutch++ for matrix functions by producing better low-rank approximations. At least in terms of the total number of matrix-vector products used, the Krylov-aware approach always outperforms Hutch++ and related variants.

Finally, we note several more specialized techniques which may be of interest. Variance reduction techniques based on multi-level Monte Carlo methods are studied in [HT21; FKR21]. In [DM21], the problem of estimating the traces of a sequence of slowly-varying implicit matrices is studied. Such a setting occurs naturally in machine learning and physics. In physics, in order to compute important quantities for open quantum systems interacting strongly with their environment, [CC22] studies how to approximate the *partial trace* of matrix functions.

4.2 Analysis

A simple approach to analyzing the protoalgorithm is to separately analyze the errors due to randomness in quadratic trace estimators from the error in approximating quadratic forms. Specifically, we have

$$\begin{aligned}\left|\int f \,\mathrm{d}\big(\Phi - \langle [\Psi_\ell]_s^{\circ \mathrm{q}}\rangle\big)\right| &\le \left|\int f \,\mathrm{d}\big(\Phi - \langle \Psi_\ell\rangle\big)\right| + \left|\int f \,\mathrm{d}\big(\langle \Psi_\ell\rangle - \langle [\Psi_\ell]_s^{\circ \mathrm{q}}\rangle\big)\right| \\ &= \left|\int f \,\mathrm{d}\big(\Phi - \langle \Psi_\ell\rangle\big)\right| + \left|\int f \,\mathrm{d}\langle \Psi_\ell - [\Psi_\ell]_s^{\circ \mathrm{q}}\rangle\right| \\ &\le \left|\int f \,\mathrm{d}\big(\Phi - \langle \Psi_\ell\rangle\big)\right| + \left\langle\left|\int f \,\mathrm{d}\big(\Psi_\ell - [\Psi_\ell]_s^{\circ \mathrm{q}}\big)\right|\right\rangle. \qquad (4.1)\end{aligned}$$

The first term in (4.1) is controlled by the convergence of $\langle \Psi_\ell \rangle$ to Φ (as $n_{\mathrm{v}} \to \infty$). Since

$$\left|\int f \,\mathrm{d}\big(\Phi - \langle \Psi_\ell\rangle\big)\right| = \left|n^{-1} \operatorname{tr}(f(\mathbf{A})) - \langle \mathbf{v}_\ell^{\mathsf{H}} f(\mathbf{A}) \mathbf{v}_\ell \rangle\right|,$$

it can be analyzed in terms of bounds for quadratic trace estimators. Next, for each ℓ, the second term is controlled by the quality of the approximation of Ψ_ℓ by $[\Psi_\ell]_s^\circ$ (as $s \to \infty$). Since

$$\left|\int f \,\mathrm{d}\big(\Psi_\ell - [\Psi_\ell]_s^{\circ \mathrm{q}}\big)\right| = \left|\int (f - [f]_s^{\circ \mathrm{p}}) \,\mathrm{d}\Psi_\ell\right|,$$

we can analyze this term bounds for Krylov subspace methods for quadratic forms.

4.2.1 Uniform unit test vectors

Definition 4.1. *The complex unit hypersphere $\mathbb{S}^{n-1}$ is the set of unit vectors; i.e.*

$$\mathbb{S}^{n-1} := \{\mathbf{u} : \|\mathbf{w}\|_2 = 1\}.$$

In this section, we analyze the weighted CESM when $\mathbf{v}$ is drawn from the uniform distribution on $\mathbb{S}^{n-1}$. In the case that $\mathbf{A}$ is symmetric, similar results hold for uniform vectors drawn from the real unit hypersphere; see [CTU22].

Lemma 4.2. *Suppose $\mathbf{v} \sim \mathrm{Unif}(\mathbb{S}^{n-1})$ and, for any $t \in \mathbb{R}$, define $m(x) = n\Phi(x)$. Then,*

$$\Psi(x) \sim \mathrm{Beta}\big(m(x), n - m(x)\big).$$

Proof. Let $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_n]$, where $\mathbf{u}_i$ is the i-th normalized eigenvector of $\mathbf{A}$. Since $\mathbf{U}$ is unitary, by the invariance of $\mathrm{Unif}(\mathbb{S}^{n-1})$ under orthogonal transforms, we have that $\mathbf{U}^{\mathsf{H}}\mathbf{v} \sim \mathrm{Unif}(\mathbb{S}^{n-1})$.

We may therefore assume $\mathbf{U}^{\mathsf{H}}\mathbf{v} \overset{\text{dist.}}{=} \mathbf{x}/\|\mathbf{x}\|_2$, where $\mathbf{x} \sim \mathrm{ComplexNormal}(\mathbf{0}, \mathbf{I})$. Recall that the i-th weight of Ψ is given by $w_i = |\mathbf{v}^{\mathsf{H}}\mathbf{u}_i|^2$. Thus, the w_i have joint distribution given by,

$$w_i \overset{\text{dist.}}{=} \left|\frac{[\mathbf{x}]_i}{\|\mathbf{x}\|_2}\right|^2 = \frac{|[\mathbf{x}]_i|^2}{\sum_{i=0}^{n-1}|[\mathbf{x}]_i|^2},$$

for $i = 0, 1, \dots, n-1$.

Write, for notational convenience, $m = m(x) = n\Phi(x)$. Then,

$$\Psi(x) = \sum_{j=0}^{m-1} w_j \overset{\text{dist.}}{=} \frac{\sum_{i=0}^{m-1}|[\mathbf{x}]_i|^2}{\sum_{i=0}^{n-1}|[\mathbf{x}]_i|^2}.$$

It is well known that for independent chi-square random variables $Y \sim \chi^2_\alpha$ and $Z \sim \chi^2_\beta$ (see, for example, [JKB94, Section 25.2]),

$$\frac{Y}{Y+Z} \sim \mathrm{Beta}\left(\frac{\alpha}{2}, \frac{\beta}{2}\right).$$

Thus, since $\sum_{i=0}^{m-1}|[\mathbf{x}]_i|^2$ and $\sum_{i=m}^{n-1}|[\mathbf{x}]_i|^2$ are independent chi-square random variables with $2m$ and $2(n-m)$ degrees of freedom (because we are using complex normal random variables) respectively, $\Psi(x)$ is a beta random variable with parameters m and $n-m$. □

Definition 4.3. *A random variable X is σ^2-sub-Gaussian if*

$$\mathbb{E}\big[\exp(\lambda(X - \mathbb{E}[X]))\big] \le \exp\left(\frac{\lambda^2\sigma^2}{2}\right), \ \forall \lambda \in \mathbb{R}.$$

Lemma 4.4. *Suppose X is σ^2-sub-Gaussian. Let $X_0, \dots, X_{n_{\mathrm{v}}-1}$ be iid samples of X. Then for all $\epsilon \ge 0$,*

$$\mathbb{P}\big[|\langle X_i\rangle - \mathbb{E}[X]| > \epsilon\big] \le 2\exp\left(-\frac{n_{\mathrm{v}}}{2\sigma^2}\epsilon^2\right).$$

Proof. We follow a standard argument; see for instance [Ver18]. WLOG assume $\mathbb{E}[X] = 0$. Then,

$$\mathbb{P}[n_{\mathrm{v}}\langle X_i\rangle \ge n_{\mathrm{v}}\epsilon] = \mathbb{P}[\exp(\lambda n_{\mathrm{v}}\langle X_i\rangle) \ge \exp(\lambda n_{\mathrm{v}}\epsilon)]$$

$$
\begin{aligned}
&\leq \exp(-n_{\mathrm{v}}\lambda\epsilon)\mathbb{E}[\exp(\lambda n_{\mathrm{v}}\langle X_i\rangle)] && \text{(Markov)}\\
&= \exp(-n_{\mathrm{v}}\lambda\epsilon)\mathbb{E}[\exp(\lambda X)]^{n_{\mathrm{v}}} && \text{(iid)}\\
&\leq \exp(-n_{\mathrm{v}}\lambda\epsilon)\exp(n_{\mathrm{v}}\lambda^2\sigma^2/2) && \text{(sub-Gaussian)}\\
&= \exp(-n_{\mathrm{v}}\lambda\epsilon + n_{\mathrm{v}}\lambda^2\sigma^2/2).
\end{aligned}
$$

This expression is minimized when $\lambda = t/\sigma^2$ from which we obtain,

$$
\mathbb{P}[\langle X_i\rangle \geq t] \leq \exp\left(-\frac{n_{\mathrm{v}}}{2\sigma^2}t^2\right). \qquad \square
$$

Theorem 4.5. [MA17, Theorem 1] *Suppose $X \sim \mathrm{Beta}(\alpha,\beta)$. Then, $\mathbb{E}[X] = \alpha/(\alpha+\beta)$, and X is $(4(\alpha+\beta+1))^{-1}$-sub-Gaussian. If $\alpha = \beta$, then there is no smaller σ^2 such that X is σ^2-sub-Gaussian.*

With these results in place, the following theorem for spectrum approximation is straightforward.

Theorem 4.6. *Given a positive integer n_{v}, suppose $\{\mathbf{v}_\ell\}_{\ell=0}^{n_{\mathrm{v}}-1} \overset{\text{iid}}{\sim} \mathrm{Unif}(\mathbb{S}^{n-1})$. Then, for all $\varepsilon > 0$,*

$$
\begin{aligned}
\max_{x\in\mathbb{R}} \mathbb{P}\left[|\Phi(x) - \langle\Psi_\ell(x)\rangle| > \varepsilon\right] &\leq 2\exp\left(-2n_{\mathrm{v}}(n+1)\varepsilon^2\right).\\
\mathbb{P}\left[\max_{x\in\mathbb{R}} |\Phi(x) - \langle\Psi_\ell(x)\rangle| > \varepsilon\right] &\leq 2n\exp\left(-2n_{\mathrm{v}}(n+1)\varepsilon^2\right).
\end{aligned}
$$

Proof. First note that the maximums exist because Φ and $\langle\Psi_i\rangle$ are right continuous and piecewise constant except at $\{\lambda_i[\mathbf{A}]\}_{i=1}^n$.

For any t, let $m = m(x) = n\Phi(x)$. Using Theorems 4.2, 4.4 and 4.5 we have that for any t,

$$
\mathbb{P}\left[\,|\Phi(x) - \langle\Psi_i(x)\rangle| > \epsilon\right] \leq 2\exp\left(-\frac{n_{\mathrm{v}}}{2(4(m+(n-m)+1))^{-1}}\epsilon^2\right).
$$

We also have

$$
\sup_{x\in\mathbb{R}} |\Phi(x) - \langle\Psi_i(x)\rangle| = \max_{0\leq i<n-1} |\Phi(\lambda_i[\mathbf{A}]) - \langle\Psi_i(\lambda_i[\mathbf{A}])\rangle|.
$$

The second result follows by applying a union bound to the events that the maximum is attained at λ_i for each $i = 0, 1, \ldots, n-2$ (note since $\|\mathbf{v}\|_2 = 1$, Φ and Ψ_ℓ agree at λ_{n-1}). $\square$

This result can be used to obtain a bound for quadratic trace estimation.

Theorem 4.7. *Set* $n_{\mathrm{v}} \geq 1$ *and sample* $\{\mathbf{v}_\ell\}_{\ell=0}^{n_{\mathrm{v}}-1} \overset{\text{iid}}{\sim} \mathrm{Unif}(\mathbb{S}^{n-1})$. *Then*

$$\mathbb{P}\left[\left|n^{-1}\operatorname{tr}(\mathbf{A}) - \langle \mathbf{v}_\ell^{\mathsf{H}}\mathbf{A}\mathbf{v}_\ell\rangle\right| > \varepsilon(\lambda_{\max} - \lambda_{\min})\right] \leq 2n\exp(-2(n+1)n_{\mathrm{v}}\varepsilon^2).$$

Proof. Since $\langle\Psi_\ell\rangle$ and Φ are both constant on each of $(-\infty, \lambda_{\min})$ and $(\lambda_{\max}, \infty)$,

$$d_{\mathrm{W}}(\Phi, \langle\Psi_\ell\rangle) = \int |\Psi - \langle\Psi_\ell\rangle|\,\mathrm{d}x \leq (\lambda_{\max} - \lambda_{\min})\|\Psi - \langle\Psi_\ell\rangle\|_{\mathbb{R}}.$$

Using Theorem 4.6, we find that

$$\mathbb{P}\left[d_{\mathrm{W}}(\Phi, \langle\Psi_\ell\rangle) > \varepsilon(\lambda_{\max} - \lambda_{\min})\right] \leq 2n\exp(-2(n+1)n_{\mathrm{v}}\varepsilon^2).$$

Thus, using Lemma 2.9 and the fact that x is 1-Lipshitz,

$$\mathbb{P}\left[\left|\int x\,\mathrm{d}\Phi - \int x\,\mathrm{d}\langle\Psi_\ell\rangle\right| > \varepsilon(\lambda_{\max} - \lambda_{\min})\right] \leq 2n\exp(-2(n+1)n_{\mathrm{v}}\varepsilon^2).$$

Next, recall that $\int x\,\mathrm{d}\Phi = n^{-1}\operatorname{tr}(\mathbf{A})$ and $\int x\,\mathrm{d}\langle\Psi_\ell\rangle = \langle\mathbf{v}_\ell^{\mathsf{H}}\mathbf{A}\mathbf{v}_\ell\rangle$. Thus, we obtain a bound for the quadratic trace estimator:

$$\mathbb{P}\left[\left|n^{-1}\operatorname{tr}(\mathbf{A}) - \langle \mathbf{v}_\ell^{\mathsf{H}}\mathbf{A}\mathbf{v}_\ell\rangle\right| > \varepsilon(\lambda_{\max} - \lambda_{\min})\right] \leq 2n\exp(-2(n+1)n_{\mathrm{v}}\varepsilon^2). \qquad \square$$

This can be restated in terms of matrix functions.

Corollary 4.8. *Suppose* f *is bounded between* $f_{\min}$ *and* $f_{\max}$ *on the spectrum of* $\mathbf{A}$. *Set* $n_{\mathrm{v}} \geq \frac{1}{2}(f_{\max} - f_{\min})^2(n+1)^{-1}\varepsilon^{-2}\ln(2n\eta^{-1})$ *and sample* $\{\mathbf{v}_\ell\}_{\ell=0}^{n_{\mathrm{v}}-1} \overset{\text{iid}}{\sim} \mathrm{Unif}(\mathbb{S}^{n-1})$. *Then*

$$\mathbb{P}\left[\left|n^{-1}\operatorname{tr}(f(\mathbf{A})) - \int f\,\mathrm{d}\langle\Psi_\ell\rangle\right| > \varepsilon\right] \leq \eta.$$

As we remarked in Section 4.1, bounds similar to Theorem 4.7 have been studied for other distributions for $\mathbf{v}$. The best bounds are for Gaussian and Rademacher vectors, which have *independent* entries. For such distributions, the best bounds depend on $\|\mathbf{A}\|_{\mathsf{F}}^2$ rather than $n\|\mathbf{A}\|_2^2$ and are therefore significantly stronger than Theorem 4.7 when the stable rank $\|\mathbf{A}\|_{\mathsf{F}}^2/\|\mathbf{A}\|_2^2$ is small. It is likely that the bounds in Theorem 4.7 can be improved by a more careful analysis of Beta random variables. In particular, while the sub-Gaussian constant from Theorem 4.5 is sharp when $\alpha = \beta$, it can be improved when $\alpha \approx 0$ or $\beta \approx 0$ [ZZ20].

4.3 Numerical experiments

4.3.1 Approximating sparse spectra

If the spectrum of $\mathbf{A}$ is S-sparse; i.e., there are only S distinct eigenvalues, then the s-point Gaussian quadrature rule will be exactly equal to the weighted CESM for all $s \geq S$, at least in exact arithmetic. Thus, the runtime required by SLQ is determined by S and the number of samples of the weighted CESM which are required to get a good approximation to the true CESM. The interpolation and approximation based approaches, which are based on the orthogonal polynomials of some fixed distribution function μ, are unable to take advantage of such sparsity. Indeed, unless the eigenvalues of $\mathbf{A}$ are known a priori, such methods have fixed resolution $\sim s^{-1}$ due to the fixed locations of the zeros of the orthogonal polynomials with respect to μ. Moreover, quadrature by approximation methods suffer from Gibbs oscillations unless a damping kernel is used, in which case the resolution is further decreased.

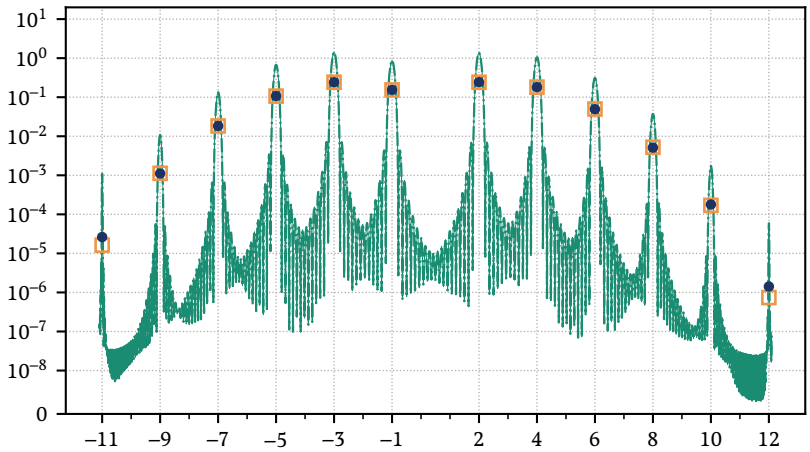


Figure 4.2: Approximations to a sparse spectrum with just 12 eigenvalues. *Legend*: true spectrum (□). Gaussian quadrature approximation: $k = 12$ (•). damped quadrature by approximation: $s = 500$ (—·—). *Takeaway*: The Gaussian quadrature produces an extremely good approximation using just 12 matrix-vector products. Even with many more matrix-vector products, quadrature by approximation does not have the same resolution.

In this example, we approximate the CESM of the adjacency matrix of a Kneser graph. The (N, K)-Kneser graph is the graph whose vertices correspond to size K subsets of $\{1, 2, \dots, N\}$ and whose edges connect vertices corresponding to disjoint sets. It is not hard to see that the number of vertices is $\binom{N}{K}$ and the number of edges is $\frac{1}{2}\binom{N}{K}\binom{N-K}{K}$. The spectrum of Kneser graphs is known as well. Specifically, there are $K+1$ distinct eigenvalues whose values and multiplicities are:

$$\lambda_i = (-1)^i \binom{N-K-i}{K-i}, \qquad m_i = \binom{N}{i} - \binom{N}{i-1}, \qquad i = 0, 1, \dots, K.$$

We conduct a numerical experiment with $N = 23$ and $K = 11$, the same values used in [Ada+18]. This results in a graph with 1,352,078 vertices and 8,112,468 edges. Thus, the adjacency matrix is highly sparse. We compare the Gaussian quadrature approximation with the damped quadrature by approximation. In both cases we use a single random test vector $\mathbf{v}$. For the Gaussian quadrature, we set $k = 12$. For the damped quadrature by approximation we set $s = 500$ and use Jackson damping with $\mu = \mu_{a,b}^T$, where $a = -11.1$ and $b = 12.1$. The results are shown in Figure 4.2. Note that the Gaussian quadrature matches almost exactly despite having used only $k = 12$ matrix-vector products. On the other hand, even after $k = 250$ matrix-vector products, the damped quadrature by approximation has a much lower resolution.

Remark 4.9. There are sublinear time algorithms for approximate matrix-vector products with the (normalized) adjacency matrix. Specifically, in a computational model where it is possible to (i) uniformly sample a random vertex in constant time, (ii) uniformly sample a neighbor of a vertex in constant time, and (iii) read off all neighbors of a vertex in linear time, then an ϵ_{mv}-accurate approximate to the a matrix-vector product with the adjacency matrix can be computed, with probability $1 - \eta$, in time $O(n(\epsilon_{\text{mv}})^{-2} \ln(\eta^{-1}))$. For dense graphs, this is sublinear in the input size $O(n^2)$ of the adjacency matrix. See [BKM22] for an analysis in the context of spectrum approximation. △

4.3.2 Approximating "smooth" densities

There are a range of settings in which the spectral density of $\mathbf{A}$ is close to a smooth slowly varying density. In such cases, we may hope that our approximation satisfies certain known criteria. For instance, that the approximation

is also a slowly varying density, that the behavior of the approximation at the endpoints of the support satisfies the right growth or decay conditions, etc. In this example, we consider how parameters in Algorithm 4.1 can be varied so that the resulting approximation enjoys certain desired properties.

One setting in which $\mathbf{A}$ may have a slowly varying density is when $\mathbf{A}$ is a large random matrix. We begin this example by considering a sample covariance matrix

$$\mathbf{A}_n = \frac{1}{m}\mathbf{\Sigma}^{1/2}\mathbf{X}\mathbf{X}^{\mathsf{H}}\mathbf{\Sigma}^{1/2}$$

where $\mathbf{X}$ is random and $\mathbf{\Sigma}$ is deterministic. Specifically, we fix constants $\sigma > 1$ and $d \in (0, 1)$, define $m = n/d$, and take $\mathbf{X}$ to be a $n \times m$ matrix with iid standard normal entries and $\mathbf{\Sigma}$ a diagonal matrix with $1/m$ as the first $n/2$ entries and σ/m as the last $n/2$ entries.

In the limit, as $n \to \infty$, the spectral density $\mathrm{d}\Phi_n/\mathrm{d}x$ of $\mathbf{A}_n$ is convergent to a density $\mathrm{d}\Psi_\infty/\mathrm{d}x$ supported on two disjoint intervals $[a_1, b_1] \cup [a_2, b_2]$, where $a_1 < b_1 < a_2 < b_2$, with equal mass on each [BS98]. The spectral edges are equal to the values at which

$$t \mapsto -\frac{1}{t} + \frac{d}{2}\left(\frac{1}{t+1} + \frac{1}{t+\sigma^{-1}}\right)$$

attains at its local extrema. Moreover, it is known that $\mathrm{d}\Psi_\infty/\mathrm{d}x$ has square root behavior at the spectral edges.

Because we know the support of the desired density, and because we know the behavior at the spectral edges, a natural choice is to use quadrature by approximation with

$$\mu = \frac{1}{2}\mu^U_{a_1,b_1} + \frac{1}{2}\mu^U_{a_2,b_2}$$

where $\mu^U_{a,b}$ is the weight function for the Chebyshev polynomials of the second kind given by

$$\frac{\mathrm{d}\mu^U_{a,b}}{\mathrm{d}x} = \frac{4}{\pi(b-a)}\sqrt{1 - \left(\frac{2}{b-a}x - \frac{b+a}{b-a}\right)}.$$

This will ensure that the Radon–Nikodym derivative $\mathrm{d}\Psi_\infty/\,\mathrm{d}\mu$ is of order 1 at the spectral edges which seems to result in better numerical behavior than if we were to use a KPM approximation corresponding to a density which explodes at the spectral edges.

To compute the Jacobi matrix for μ, we apply the Stieltjes procedure using a slight modification of the *Vandermonde with Arnoldi* approach [BNT21]. In

order to apply the Stieltjes procedure, we must be able to integrate polynomials against μ. Observe that the product

$$\int p \, d\mu = \frac{1}{2} \int p \, d\mu^U_{a_1,b_1} + \frac{1}{2} \int p \, d\mu^U_{a_2,b_2}$$

can be computed *exactly* by applying a sufficiently high degree quadrature rule to each of the right hand side integrals. If we aim to compute the $s \times s$ Jacobi matrix associated with μ the maximum degree polynomial we will integrate will be of degree $2s - 1$ when we orthogonalize xp_{s-1} against p_{s-1}. Therefore, it suffices to use the degree s Gaussian quadrature rules for $\mu^U_{a_1,b_1}$ and $\mu^U_{a_2,b_2}$ for all of the first s iterations of the Stieltjes procedure.

One simple approach to running the Stieltjes procedure in this manner is to place the quadrature nodes on the diagonal of a matrix $\mathbf{N}$ and the corresponding weights on a vector $\mathbf{w}$. Then the weighted CESM corresponding to $\mathbf{N}$ and $\mathbf{w}$ is a quadrature rule which integrate polynomials of degree up to $2s - 1$ against μ exactly. The tridiagonal matrix obtained by the Lanczos algorithm run for s iterations will be exactly the upper $s \times s$ block of the Jacobi matrix $\mathbf{M}(\mu)$. Some potentially more computationally efficient approaches are outlined in [FG91].

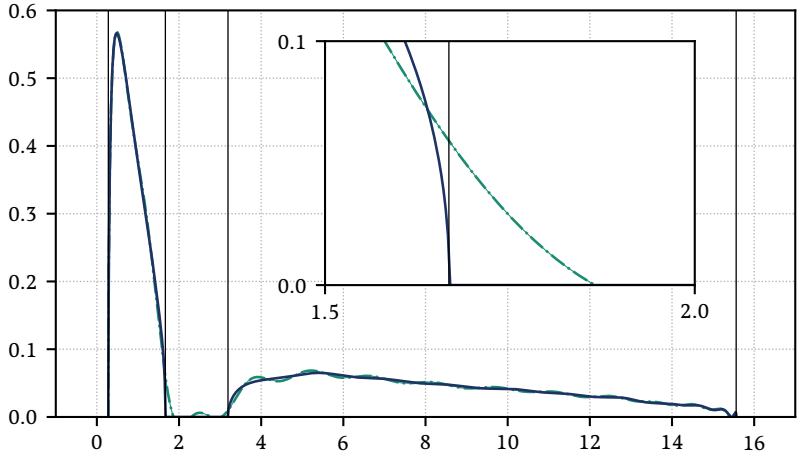


Figure 4.3: Approximations to a "smooth" spectrum using quadrature by approximation with various choices of μ. *Legend*: $\mu = \mu^U_{a_1,b_2}$ (—·—). $\mu = \frac{1}{2}\mu^U_{a_1,b_1} + \frac{1}{2}\mu^U_{a_2,b_2}$ (——). *Takeaway*: A priori knowledge about the spectrum allows for better choices of parameters such as μ.

We conduct a numerical experiment with $n = 10^4$ and $d = 0.3$. We use $s = 60$ and average over 10 trials, resampling $\mathbf{A}_n$ in each trial. To generate an approximation to the density we expand the support of the limiting density by 0.001 on endpoint to avoid eigenvalues of $\mathbf{A}_n$ lying outside the support of μ. In Figure 4.3 we show the approximations with $\mu = \mu^U_{a_1,b_2}$ and $\mu = \frac{1}{2}\mu^U_{a_1,b_1} + \frac{1}{2}\mu^U_{a_2,b_2}$. As shown in the inset image of Figure 4.3, we observe that the approximation with $\mu = \frac{1}{2}\mu^U_{a_1,b_1} + \frac{1}{2}\mu^U_{a_2,b_2}$ exhibits the correct square root behavior at the endpoints as well as fewer oscillations throughout the interior of the support of the density.

Remark 4.10. In recent work [DT21] it was shown how Lanczos performs on such a sample covariance matrix. In particular, one sample from stochastic Lanczos quadrature will converge almost surely, as $n \to \infty$, to the desired distribution. In this same work another density approximation scheme was proposed based on Stieltjes transform inversion. Analysis and comparison for this method is an interesting open problem. △

Smoothing by convolution

The Gaussian quadrature approximation is the sum of weighted Dirac delta functions. A simple approach to obtain a density function from a distribution function involving point masses is to approximate each point masses with some concentrated probability density function; e.g. Gaussians with a small variance [LSY16; GKX19]. This is simply convolution with this distribution, and if the smoothing distribution has small enough variance, the Wasserstein distance between the original and smoothed distributions will be small. Specifically, we have the following standard lemma:

Lemma 4.11. *Given a smooth positive probability distribution function G_σ, define the smoothed approximation Υ_σ to Υ by the convolution*

$$\Upsilon_\sigma(x) := \int_{-\infty}^{\infty} G_\sigma(t-y)\,\mathrm{d}\Upsilon(y).$$

Then, $d_{\mathrm{W}}(\Upsilon, \Upsilon_\sigma) \leq d_{\mathrm{W}}(\mathbb{1}[x < 0], G_\sigma) d_{\mathrm{TV}}(\Upsilon)$. Moreover, if G_σ has median zero and standard deviation σ, then $d_{\mathrm{W}}(\Upsilon, \Upsilon_\sigma) \leq \sigma\, d_{\mathrm{TV}}(\Upsilon)$.

It is well known that if G_σ is differentiable then the smoothed distribution function Υ_σ will also be differentiable. Thus, we can obtained a density function

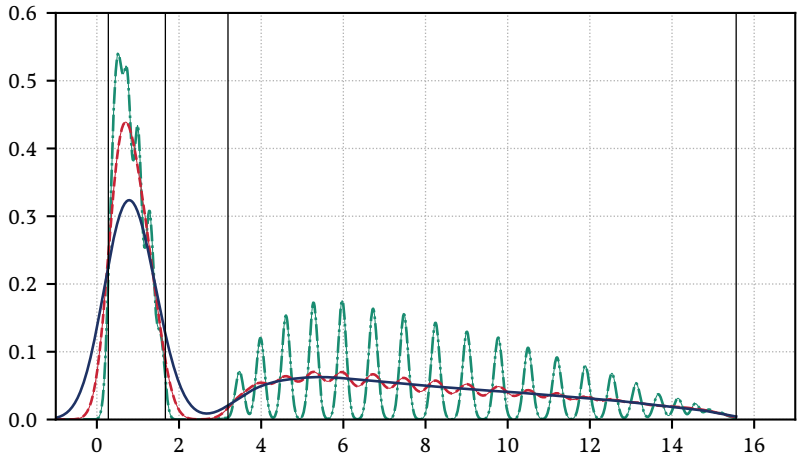


Figure 4.4: Approximations to a "smooth" spectrum using smoothed Gaussian quadrature for various smoothing parameters σ. *Legend*: $\sigma = 3/k$ (—·—). $\sigma = 8/k$ (- - -). $\sigma = 15/k$ (——). *Takeaway*: Gaussian quadrature is not always the best choice of algorithm. Here we observe that it is difficult to produce a density approximation using the specified smoothing scheme.

$d\Upsilon_\sigma/dx$ even if Υ has discontinuities. Moreover, the bounds obtained earlier can easily be extended to smoothed spectral density approximations obtained by convolution using the triangle inequality.

While the smoothing based approach has a simple theoretical guarantee in Wasserstein distance, it does not need to provide a good approximation to the density. Indeed, if the variance of the smoothing kernel is too small, then the smoothed distribution will still look somewhat discrete. On the other hand, if the variance of the smoothing kernel is too large, then the smooth distribution will become blurred out and lose resolution. As shown in Figure 4.4, this is particularly problematic if different parts of the spectrum would naturally require different amounts of smoothing.

There are of course many different smoothing schemes that could be used. These include adaptively choosing the variance parameter based on the position in the spectrum, using a piecewise constant approximation to the density, interpolating the distribution function with a low degree polynomial or splines,

etc. Further exploration of these approaches is beyond the scope of this thesis since they would likely be context dependent. For instance, in random matrix theory, it may be desirable to enforce square root behavior at endpoints whereas in other applications it may be desirable to have smooth tails.

We conclude with the remark that alternate metrics of closeness, such as the total variation distance, are likely better suited for measuring the quality of approximations to "smooth" densities. However, since the actual spectral density $\mathrm{d}\Psi/\mathrm{d}x$ is itself the sum of Dirac deltas, some sort of regularization is required to obtain a proper density [LSY16] which of course relates closely to what it actually means to be "close to a smooth slowly varying density". A rigorous exploration of this topic would be of interest.

Handling isolated spikes

In some situations one may encounter spectra which are nearly "smooth" except at a few points at which there are large jumps in the CESM (for instance, low rank matrices may have many repeated zero eigenvalues).

To model such a situation, we consider a matrix

$$\mathbf{A}_n := \begin{bmatrix} m^{-1}\mathbf{X}\mathbf{X}^{\mathsf{H}} & \mathbf{0} \\ \mathbf{0} & z\mathbf{I} + \sigma\mathbf{D} \end{bmatrix}$$

where $\mathbf{X}$ is a $n' \times m$ matrix standard normal entries and $\mathbf{D}$ is a $(n - n') \times (n - n')$ diagonal matrix with standard normal entries. In both cases, $m = n'/d$ for some fixed $d \in (0, 1)$. While this particular matrix is block diagonal, the protoalgorithm is mostly oblivious to this structure and would work similarly well if the matrix were conjugated by an arbitrary unitary matrix so that the block diagonal structure is lost.

When $n \to \infty$ and $\sigma \to 0$, the spectral density $\mathrm{d}\Phi_n/\mathrm{d}x$ is convergent to a density $\mathrm{d}\Phi_\infty/\mathrm{d}x$ equal to the sum of a scaled Marchenko–Pastur distribution and a weighted Dirac delta distribution. Thus, a natural approach would be to use quadrature by approximation with

$$\mu = (1 - p)\mu_{a,b}^{U} + p\,\delta(x - z).$$

As above, we can use a modified version of the Vandermonde with Arnoldi approach to compute the orthogonal polynomials with respect to μ. The resulting

approximation to the "smooth" part of the density $d\Phi/dx$ is shown in Figure 4.5.

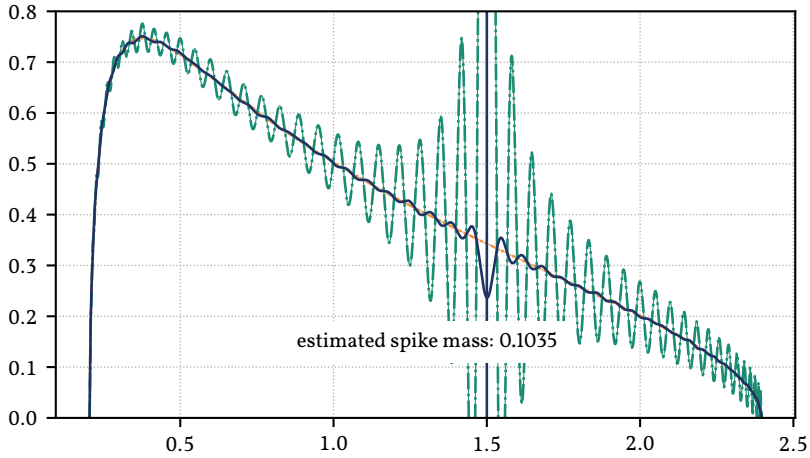


Figure 4.5: Approximations to a "smooth" spectrum with a spike using quadrature by approximation with various choices of μ. *Legend*: absolutely continuous part of true limiting density ( ). quadrature by approximation: $\mu = (1-p)\mu_{a,b}^U + p\,\delta(x-z)$ ( ). quadrature by approximation: $\mu = \mu_{a,b}^U$ ( ). *Takeaway*: A priori knowledge of the location of a singularity allows for a better approximation to the absolutely continuous part of the spectrum.

We set $n = 10^6$, $n' = n/10$, $d = 0.3$, $z = 1.5$, and $\sigma = 10^{-10}$. As before, we average of 10 trials where $\mathbf{A}_n$ is resampled in each trial. For each sample, we compute the quadrature by approximation with $s = 200$ for $\mu = (1-p)\mu_{a,b}^U + p\,\delta(x-z)$ with $p = 0.2$ and $\mu = \mu_{a,b}^U$. The results are show in Figure 4.5.

Clearly, accounting for the spike explicitly results in a far better approximation to the density. Note that this approach does not require that the mass of the spike is accurately matched. For instance, in our example, we estimate the spike mass to be 0.2 while the actual mass is 0.1. On the other hand, if the location of the spike is misestimated, then the approximation to the density may have massive oscillations. In our example the spike has width roughly 10^{-10} which does not cause issues for the value of s used. However, if s is increased, the width of the spike is increased, or the location of the estimate of the spike is offset significantly, then the existing oscillations become large. Approaches for adaptively finding the location of spikes would an interesting area of further

study.

4.3.3 Energy spectra of small spin systems

The quantum Heisenberg model can be used to study observables of magnetic systems [Wei+06; SS10; SRS20; Sch+21; SRS22]. For a system with N spins of spin number S, the Heisenberg spin Hamiltonian is an operator on a Hilbert space of dimension $(2S+1)^N$ given by

$$\mathbf{H} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left([\mathbf{J}^{\mathrm{x}}]_{i,j} \mathbf{s}_i^{\mathrm{x}} \mathbf{s}_j^{\mathrm{x}} + [\mathbf{J}^{\mathrm{y}}]_{i,j} \mathbf{s}_i^{\mathrm{y}} \mathbf{s}_j^{\mathrm{y}} + [\mathbf{J}^{\mathrm{z}}]_{i,j} \mathbf{s}_i^{\mathrm{z}} \mathbf{s}_j^{\mathrm{z}} \right).$$

Here $\mathbf{s}_i^{\sigma}$ gives the component spin operator for the i-th spin site and acts trivially on the Hilbert spaces associated with other spin sites but as the $(2S+1)\times(2S+1)$ component spin matrix $\mathbf{s}^{\sigma}$ on the i-th spin site. Thus, $\mathbf{s}_i^{\sigma}$ can be represented in matrix form as

$$\mathbf{s}_i^{\sigma} = \underbrace{\mathbf{I} \otimes \cdots \otimes \mathbf{I}}_{i \text{ terms}} \otimes \mathbf{s}^{\sigma} \otimes \underbrace{\mathbf{I} \otimes \cdots \otimes \mathbf{I}}_{N-i-1 \text{ terms}}.$$

The CESM of $\mathbf{H}$ gives the energy spectrum of the system and can be used to compute many important quantities. For instance, given an observable $\mathbf{O}$ (i.e. a Hermitian matrix), the corresponding thermodynamic expectation of the observable in thermal equilibrium at inverse temperature β is given by

$$\frac{\mathrm{tr}(\mathbf{O}\exp(-\beta\mathbf{H}))}{\mathrm{tr}(\exp(-\beta\mathbf{H}))}.$$

Quantities depending on observables which are matrix functions $\mathbf{H}$ can be written entirely in terms of matrix functions of $\mathbf{H}$. For instance, the system heat capacity is given by

$$\frac{C(T)}{k_B} = \frac{\mathrm{tr}\left((\beta\mathbf{H})^2 \exp(-\beta\mathbf{H})\right)}{\mathrm{tr}\left(\exp(-\beta\mathbf{H})\right)} - \left[\frac{\mathrm{tr}\left(\beta\mathbf{H}\exp(-\beta\mathbf{H})\right)}{\mathrm{tr}\left(\exp(-\beta\mathbf{H})\right)}\right]^2.$$

Thus, for fixed finite temperature, evaluating the heat capacity amounts to evaluating several matrix functions.

In some cases, symmetries of the system can be exploited to diagonalize or block diagonalize $\mathbf{H}$ [SS10]. Numerical diagonalization can be applied to blocks to obtain a full diagonalization. Even so, the exponential dependence of the size of

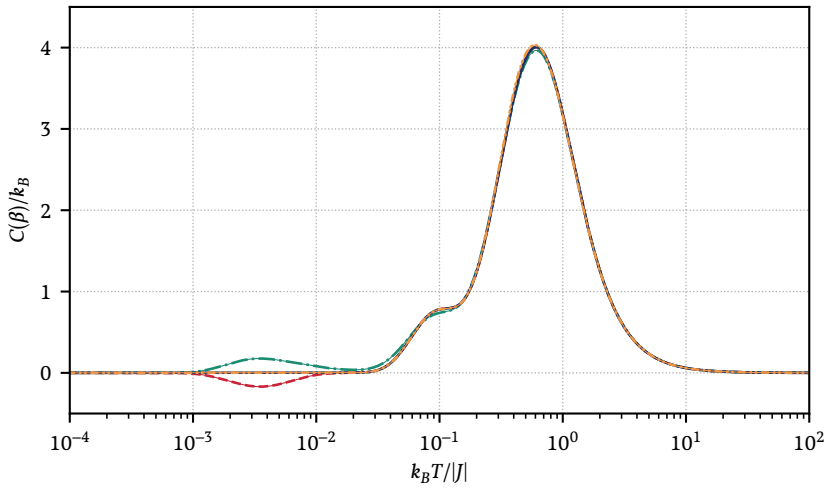


Figure 4.6: Heat capacity as a function of temperature for a small spin system. *Legend*: exact diagonalization ( ), Gaussian quadrature ( ), quadrature by approximation ( ), and damped quadrature by approximation ( ). *Takeaway*: While damping produces a physical result, the resulting ghost bump may be more difficult to identify than the nonphysical ghost dip obtained without damping.

H on the number of spin sites N limits the size of systems which can be treated in this way. Moreover, such techniques are not applicable to all systems. Thus, approaches based on Algorithm 4.1 are widely used; see [SRS20] for examples using a Lanczos based approach and [Sch+21] for examples using a Chebyshev based approach.

In this example, we consider a Heisenberg ring ($[\mathbf{J}^{\mathrm{x}}]_{i,j} = [\mathbf{J}^{\mathrm{y}}]_{i,j} = [\mathbf{J}^{\mathrm{z}}]_{i,j} = \mathbb{1}[|i-j| = 1 \pmod N]$) with $N = 12$ and $S = 1/2$. Similar examples, with further discussion in the context of the underlying physics, are considered in [SRS20; Sch+21]. We take $k = 50$ and $n_{\mathrm{v}} = 300$ and compute approximations to the heat capacity at many temperatures using Gaussian quadrature, quadrature by interpolation, and damped quadrature by interpolation. For the latter two approximations we use $\mu = \mu_{a,b}^T$ where a and b are chosen based on the nodes of the Gaussian quadrature. Note that averages over random vectors are computed for each trace rather than for $C(\beta)$, and that we use the same vectors for all four traces. The results are shown in Figure 4.6.

We note the presence of an nonphysical "ghost dip" in the quadrature by interpolation approximation. If the approximation to the CESM is non-decreasing, the Cauchy–Schwarz inequality guarantees a positive heat capacity. Thus, when we use Jackson's damping, the heat capacity remains positive for all temperatures. However, as noted in [Sch+21], this is not necessarily desirable as the ghost dip is easily identifiable while the ghost peak may be harder to identify.

We conclude with the remark that it would be interesting to provide bounds for the accuracy of the approximations to the quantity $\mathrm{tr}(\mathbf{O}\exp(-\beta\mathbf{H}))/\mathrm{tr}(\exp(-\beta\mathbf{H}))$ for all $\beta > 0$.

Chapter 5
Optimal rational matrix function approximation

We now shift gears a bit and consider methods for approximating $f(\mathbf{A})\mathbf{v}$ from Krylov subspace $K_k = \text{span}\{\mathbf{v}, \mathbf{A}\mathbf{v}, \dots, \mathbf{A}^{k-1}\mathbf{v}\}$. The most ideal Krylov subspace method for this task would output iterates alg_k satisfying

$$\text{alg}_k = \underset{\mathbf{x} \in K_k}{\text{argmin}} \|f(\mathbf{A})\mathbf{v} - \mathbf{x}\|.$$

The above condition guarantees the algorithm produces approximations with smaller error (in the given norm) than any other Krylov subspace method. Moreover, under the assumption $\|\cdot\|$ is induced by a positive definite matrix with the same eigenvectors as $\mathbf{A}$, Lemma 10.1 implies the iterates satisfy a bound

$$\|f(\mathbf{A})\mathbf{v} - \text{alg}_k\| \leq \min_{\deg(p)<k} \|f(\mathbf{A})\mathbf{v} - p(\mathbf{A})\mathbf{v}\| \leq \min_{\deg(p)<k} \|f - p\|_\Lambda \|\mathbf{v}\|.$$

In other words, the iterates satisfy a minimax bound *on the eigenvalues of* $\mathbf{A}$. As we saw in the introduction, a bound on the eigenvalues can be substantially stronger than a bound on an interval containing the eigenvalues.

A number of well-known algorithms, including the conjugate gradient (CG), minimum residual (MINRES), and quasi-minimum residual (QMR) algorithms, are standard methods for solving linear systems of equations $\mathbf{A}\mathbf{x} = \mathbf{v}$; i.e. for approximating $\mathbf{A}^{-1}\mathbf{v}$. Each of these methods is *optimal* for a certain norm and for certain classes of matrix $\mathbf{A}$. Moreover, such methods can be implemented in such a way that the amount of storage they use does not grow with the number of iterations k.

In this chapter, we describe the Lanczos method for optimal rational matrix function approximation (Lanczos-OR). When f is a rational function, Lanczos-OR outputs the *optimal* (in a certain norm) approximation to $f(\mathbf{A})\mathbf{v}$ from $\mathcal{K}_{k+1}$ using slightly more than k matrix-vector products. We provide a practical implementation of Lanczos-OR that only requires storing a number of vectors of length n proportional to the degree of the denominator in the rational function. Therefore, for a fixed rational function, the storage costs do not grow with the iteration k. The approach used to derive this implementation of Lanczos-OR can also be used for computing the Lanczos-FA approximations to rational matrix functions, avoiding storage costs growing with k in that widely used method.

Lanczos-OR is closely related to existing methods for linear systems. In particular, if $f = 1/(x - z)$, then, depending on the choice of z, the CG, MINRES, and QMR iterates can all be obtained as special cases of Lanczos-OR. The Lanczos-OR iterate is mathematically equivalent to an optimal Galerkin projection method as described in [LS06, Section 4], provided the denominator matrix is positive definite. However, this method was mostly viewed as of theoretical interest since it could be used to help explain the behavior of Lanczos-FA. On the other hand, we show that such approximations can be computed efficiently. Our approach is also somewhat more general in that it works with *any* rational function.

5.1 A bit of notation

To simplify analysis, it will be useful to consider the recurrence that would obtained if the Lanczos algorithm were run to completion. In exact arithmetic, for some $K \leq n$, $\beta_{K-1} = 0$ in which case the algorithm terminates. Then, the final basis $\widehat{\mathbf{Q}} := [\mathbf{q}_0, \ldots, \mathbf{q}_{K-1}]$ and symmetric tridiagonal $\widehat{\mathbf{T}}$ with diagonals $[\alpha_0, \ldots, \alpha_{K-1}]$ and off diagonals $[\beta_0, \ldots, \beta_{K-2}]$ satisfy a three-term recurrence

$$\mathbf{A}\widehat{\mathbf{Q}} = \widehat{\mathbf{Q}}\widehat{\mathbf{T}}. \tag{5.1}$$

We emphasize that the algorithms we discuss *do not* require Lanczos to be run to completion; the introduction of $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{T}}$ is for analysis purposes only. We note that $\mathbf{Q} = [\widehat{\mathbf{Q}}]_{:,:k}$ and $\mathbf{T} = [\widehat{\mathbf{T}}]_{:k,:k}$. Since the columns of $\widehat{\mathbf{Q}}$ are orthogonal, we have

that

$$\widehat{\mathbf{T}} = \widehat{\mathbf{Q}}^{\mathsf{H}}\mathbf{A}\widehat{\mathbf{Q}},$$

from which we easily see that, after any number of iterations k, $\mathbf{T} = \mathbf{Q}^{\mathsf{H}}\mathbf{A}\mathbf{Q}$. Note also that, for any shift $z \in \mathbb{C}$,

$$(\mathbf{A} - z\mathbf{I})\widehat{\mathbf{Q}} = \widehat{\mathbf{Q}}(\widehat{\mathbf{T}} - z\mathbf{I}).$$

In other words, the Krylov subspaces generated by $(\mathbf{A}, \mathbf{v})$ and $(\mathbf{A} - z\mathbf{I}, \mathbf{v})$ coincide, and the associated tridiagonal matrices are easily related by a diagonal shift. This shift invariance of Krylov subspaces is critical in a number of Krylov subspace methods.

5.2 Existing algorithms

It is well known that when $\mathbf{A}$ is positive definite, CG minimizes the $\mathbf{A}$-norm of the error over the Krylov subspace; i.e., the CG approximation cg_k is given by

$$\mathsf{cg}_k := \operatorname*{argmin}_{\mathbf{x} \in K_k} \|\mathbf{A}^{-1}\mathbf{v} - \mathbf{x}\|_{\mathbf{A}}.$$

Since $\mathbf{Q}$ is a basis for K_k, we can equivalently write

$$\mathsf{cg}_k = \mathbf{Q} \operatorname*{argmin}_{\mathbf{c} \in \mathbb{R}^k} \|\mathbf{A}^{1/2}(\mathbf{A}^{-1}\mathbf{v} - \mathbf{Q}\mathbf{c})\|_2 = \mathbf{Q} \operatorname*{argmin}_{\mathbf{c} \in \mathbb{R}^k} \|\mathbf{A}^{-1/2}\mathbf{v} - \mathbf{A}^{1/2}\mathbf{Q}\mathbf{c}\|_2.$$

The solution to this least squares problem is

$$\mathsf{cg}_k = \mathbf{Q}(\mathbf{Q}^{\mathsf{H}}\mathbf{A}^{1/2}\mathbf{A}^{1/2}\mathbf{Q})^{-1}\mathbf{Q}^{\mathsf{H}}\mathbf{A}^{1/2}\mathbf{A}^{-1/2}\mathbf{v} = \mathbf{Q}\mathbf{T}^{-1}\mathbf{e}_0.$$

Here we have used that $\mathbf{Q}^{\mathsf{H}}\mathbf{A}\mathbf{Q} = \mathbf{T}$ and that $\mathbf{Q}^{\mathsf{H}}\mathbf{v} = \|\mathbf{v}\|_2\mathbf{e}_0 = \mathbf{e}_0$ (since we are assuming $\|\mathbf{v}\|_2 = 1$).

If $\mathbf{A}$ is indefinite, then the $\mathbf{A}$-norm of the error is not well defined and the CG iterates need not be optimal. A common alternative to CG for indefinite systems is MINRES, which minimizes the $\mathbf{A}^2$-norm of the error (the 2-norm of the residual) over the Krylov subspace; i.e., the MINRES approximation mr_k is given by

$$\mathsf{mr}_k := \operatorname*{argmin}_{\mathbf{x} \in K_k} \|\mathbf{A}^{-1}\mathbf{v} - \mathbf{x}\|_{\mathbf{A}^2} = \operatorname*{argmin}_{\mathbf{x} \in K_k} \|\mathbf{v} - \mathbf{A}\mathbf{x}\|_2$$

Now note that

$$\mathbf{AQ} = [\mathbf{A}\widehat{\mathbf{Q}}]_{:,:k} = [\widehat{\mathbf{Q}}\widehat{\mathbf{T}}]_{:,:k} = [\widehat{\mathbf{Q}}]_{:,:k+1}[\widehat{\mathbf{T}}]_{:k+1,:k}.$$

Therefore, since $\widehat{\mathbf{T}}$ is symmetric and $\widehat{\mathbf{Q}}$ has orthonormal columns,

$$\mathsf{mr}_k = \mathbf{Q}(\mathbf{Q}^{\mathsf{H}}\mathbf{AAQ})^{-1}\mathbf{Q}^{\mathsf{H}}\mathbf{Av} = \mathbf{Q}([\widehat{\mathbf{T}}]_{:k,:k+1}[\widehat{\mathbf{T}}]_{:k+1,:k})^{-1}\mathbf{Te}_0.$$

More generally, suppose z is an arbitrary complex number. Then a special case of the quasi-minimum residual method (QMR) [Fre92] can be used to compute the optimal $(\mathbf{A}^2 + |z|^2\mathbf{I})$-norm approximation to $(\mathbf{A} - z\mathbf{I})^{-1}\mathbf{v}$; i.e., the QMR approximation $\mathsf{qmr}_k(z)$ is given by

$$\begin{aligned}\mathsf{qmr}_k(z) &:= \operatorname*{argmin}_{\mathbf{x}\in K_k} \|(\mathbf{A} - z\mathbf{I})^{-1}\mathbf{v} - \mathbf{x}\|_{(\mathbf{A}^2+|z|^2\mathbf{I})} \\ &= \operatorname*{argmin}_{\mathbf{x}\in K_k} \|(\mathbf{A} - \overline{z}\mathbf{I})^{1/2}(\mathbf{A} - z\mathbf{I})^{-1/2}\mathbf{v} - (\mathbf{A} - \overline{z}\mathbf{I})^{1/2}(\mathbf{A} - z\mathbf{I})^{1/2}\mathbf{x}\|_2. \end{aligned} \tag{5.2}$$

Here we have used that $\mathbf{A}^2+|z|^2\mathbf{I} = (\mathbf{A}-\overline{z}\mathbf{I})(\mathbf{A}-z\mathbf{I})$. Next, using the shift invariance of Krylov subspace, we see that

$$\begin{aligned}\mathsf{qmr}_k(z) &:= \mathbf{Q}(\mathbf{Q}^{\mathsf{H}}(\mathbf{A} - \overline{z}\mathbf{I})(\mathbf{A} - z\mathbf{I})\mathbf{Q})^{-1}\mathbf{Q}^{\mathsf{H}}(\mathbf{A} - z\mathbf{I})\mathbf{v}. \\ &= \mathbf{Q}(\mathbf{Q}(\mathbf{A}^2 + |z|^2\mathbf{I})\mathbf{Q}^{\mathsf{H}})^{-1}\mathbf{Q}^{\mathsf{H}}(\mathbf{A} - \overline{z}\mathbf{I})\mathbf{v} \\ &= \mathbf{Q}([\widehat{\mathbf{T}}]_{:k,:k+1}[\widehat{\mathbf{T}}]_{:k+1,:k} + |z|^2\mathbf{I})^{-1}(\mathbf{T} - \overline{z}\mathbf{I})\mathbf{e}_0.\end{aligned}$$

At first glance, CG, MINRES, and QMR all require the matrix $\mathbf{Q}$ which is of size nk. However, by taking advantage of the tridiagonal structure of $\widehat{\mathbf{T}}$, each of these algorithms can be implemented in a way which require storing just a few vectors of length n. In particular, the algorithms work by implicitly forming $\mathbf{LDL}^{\mathsf{H}}$ factorization of $\widehat{\mathbf{T}}$ [PS75; Fre92; LS13b; ŠT21]. The low-memory implementations for Lanczos-OR and Lanczos-FA that we derive in Section 5.5 are based on this idea.

5.3 Optimal rational function approximation

We now describe an optimal iterate for approximating $r(\mathbf{A})\mathbf{b}$ when r is an arbitrary fixed rational function. We will describe a low-memory implementation of this algorithm in Section 5.5. Our low-memory implementation can also be

used to compute the Lanczos-FA approximations to $r(\mathbf{A})\mathbf{b}$ without doubling the number of matrix-vector products.

Definition 5.1. *Let $r : \mathbb{R} \to \mathbb{R}$ be a rational function decomposed as $r = M/N$, where $N : \mathbb{R} \to \mathbb{R}$ is a polynomial with leading coefficient one and $M : \mathbb{R} \to \mathbb{R}$ is a polynomial which does not divide N. For any polynomial $R : \mathbb{R} \to \mathbb{R}$, define $\tilde{M} = MR$ and $\tilde{N} = NR$. Then the Lanczos-OR iterate is defined as*

$$\text{lan-OR}_k(r, R) := \mathbf{Q}([\tilde{N}(\widehat{\mathbf{T}})]_{:k,:k})^{-1}[\tilde{M}(\widehat{\mathbf{T}})]_{:k,:k}\mathbf{e}_0.$$

Those familiar with CG, MINRES, and the version of QMR for shifted Hermitian systems will note that these optimal algorithms are each obtained as special cases of Lanczos-OR. Specifically, when $\mathbf{A}$ is positive definite, CG is obtained with $r(x) = 1/x$ and $R(x) = 1$, MINRES is obtained with $r(x) = 1/x$ and $R(x) = x$, and QMR is obtained if $r(x) = 1/(x - z)$ and $R(x) = (x - \overline{z})$. In fact, we have a more general optimality result for Lanczos-OR:

Theorem 5.2. *Given a rational function $r(x) = M(x)/N(x)$ as in Theorem 5.1, choose a polynomial R so that $\mathbf{H} = \tilde{N}(\mathbf{A}) = N(\mathbf{A})R(\mathbf{A})$ is positive definite. Then $\text{lan-OR}_k(r, R)$ is the $\mathbf{H}$-norm optimal approximation to $r(\mathbf{A})\mathbf{v}$ from $\mathcal{K}_k$.*

A simple way to ensure $\mathbf{H}$ is positive definite is to take $R(x) = N(x)$ so that $\mathbf{H} = N(\mathbf{A})^2$. However, in some situations, we may be able to get away with a lower degree choice for R. For instance, in the case of symmetric linear systems, while one can always use MINRES ($r(x) = 1/xi, R(x) = x$), if $\mathbf{A}$ is positive definite, then one may hope to use CG ($r(x) = 1/x$, $R(x) = 1$). A simple way to obtain a lower degree choice of R is to only take the terms in N which are indefinite.

Definition 5.3. *Given a rational function $r = M/N$ as in Theorem 5.1, factor*

$$N(x) = \left(\prod_{i=0}^{d_1-1}(x - z_i)\right)\left(\prod_{i=0}^{d_2-1}(x - z_i')(x - \overline{z_i'})\right)$$

where $z_i \neq \overline{z}_j$ for all $i, j = 0, 1, \ldots, d_1 - 1$ with $j \neq i$. Then R^* is defined by

$$R^*(x) = \xi\prod_{i=0}^{d_1-1}(x - \overline{z}_i)^{\alpha_i}$$

where, for $i = 0, 1, \ldots, q - 1$, $\alpha_i = 0$ if $z_i \in \mathbb{R} \setminus \mathcal{I}$ and $\alpha_i = 1$ otherwise and $\xi \in \{\pm 1\}$ is chosen so that $R^*(\lambda_{\min})N(\lambda_{\min}) \geq 0$.

Lemma 5.4. *Given a rational function $r = M/N$ as in Theorem 5.1, choose R^* as in Theorem 5.3. Then $\mathbf{H} = N(\mathbf{A})R^*(\mathbf{A})$ is positive definite.*

Proof. Clearly $\prod_{i=0}^{q'-1}(x - z'_i)(x - \overline{z'}_i) \geq 0$ for all $x \in \mathbb{R}$, and for $z_i \in \mathbb{R} \setminus \mathcal{I}$, $(x - z_i)$ does not change signs over $\mathcal{I}$. The choice of ξ ensures that $\tilde{N}(\lambda_{\min}) > 0$, and by assumption $N(\lambda) \neq 0$ for all $\lambda \in \Lambda$ so that that $\tilde{N}(\lambda) \neq 0$ for all $\lambda \in \Lambda$. It follows that $\tilde{N}(\lambda) > 0$ for all $\lambda \in \Lambda$; i.e. that $\mathbf{H} = \tilde{N}(\mathbf{A})$ is positive definite. □

Proof of Theorem 5.2. Since $\mathbf{y} \in \mathcal{K}_k$, we have $\mathbf{y} = \mathbf{Q}\mathbf{c}$ for some vector $\mathbf{c}$. Thus, we can consider the problem

$$\operatorname*{argmin}_{\mathbf{c}\in\mathbb{R}^k} \|r(\mathbf{A})\mathbf{v} - \mathbf{Q}\mathbf{c}\|_{\mathbf{H}} = \operatorname*{argmin}_{\mathbf{c}\in\mathbb{R}^k} \|\mathbf{H}^{1/2}r(\mathbf{A})\mathbf{v} - \mathbf{H}^{1/2}\mathbf{Q}\mathbf{c}\|_2.$$

But this is just a standard least squares problem which has solution

$$\mathbf{c} = ((\mathbf{H}^{1/2}\mathbf{Q})^{\mathsf{H}}(\mathbf{H}^{1/2}\mathbf{Q}))^{-1}(\mathbf{H}^{1/2}\mathbf{Q})^{\mathsf{H}}(\mathbf{H}^{1/2}r(\mathbf{A})\mathbf{v}).$$

Thus, we see that the optimal iterate has the form

$$\mathbf{Q}(\mathbf{Q}^{\mathsf{H}}\mathbf{H}\mathbf{Q})^{-1}\mathbf{Q}^{\mathsf{H}}\mathbf{H}r(\mathbf{A})\mathbf{v}.$$

By definition, $\mathbf{H} = N(\mathbf{A})R(\mathbf{A})$ so $\mathbf{H}r(\mathbf{A}) = M(\mathbf{A})R(\mathbf{A}) = \tilde{M}(\mathbf{A})$. Thus,

$$\mathbf{Q}^{\mathsf{H}}\mathbf{H}r(\mathbf{A})\mathbf{v} = \mathbf{Q}^{\mathsf{H}}M(\mathbf{A})\mathbf{v} = \mathbf{Q}^{\mathsf{H}}\widehat{\mathbf{Q}}\tilde{M}(\widehat{\mathbf{T}})\widehat{\mathbf{Q}}^{\mathsf{H}}\mathbf{v} = [\tilde{M}(\widehat{\mathbf{T}})]_{:k,:k}\mathbf{e}_0.$$

Next, since $\mathbf{Q}$ consists of the first k columns $\widehat{\mathbf{Q}}$,

$$\mathbf{Q}^{\mathsf{H}}\mathbf{A}^q\mathbf{Q} = [\widehat{\mathbf{Q}}^{\mathsf{H}}\mathbf{A}^q\widehat{\mathbf{Q}}]_{:k,:k} = [\widehat{\mathbf{T}}^q]_{:k,:k}$$

so since $\mathbf{H}$ is a linear combination of powers of $\mathbf{A}$ we obtain

$$\mathbf{Q}^{\mathsf{H}}\mathbf{H}\mathbf{Q} = \mathbf{Q}^{\mathsf{H}}\tilde{N}(\mathbf{A})\mathbf{Q} = [\tilde{N}(\widehat{\mathbf{T}})]_{:k,:k}.$$

The result follows by combining and rearranging the above expressions. □

As we noted at the beginning of this chapter, the optimality of Lanczos-OR implies an a priori scalar polynomial error bound on the eigenvalues of $\mathbf{A}$ analogous to the well known minimax bounds for CG, MINRES, and QMR [Gre97].

Theorem 5.5. *Given a rational function $r = M/N$ as in Theorem 5.1 and a polynomial R so that $\mathbf{H} = \tilde{N}(\mathbf{A}) = N(\mathbf{A})R(\mathbf{A})$ is positive definite,*

$$\frac{\|r(\mathbf{A})\mathbf{v} - \text{lan-OR}_k(r, R)\|_{\mathbf{H}}}{\|\mathbf{v}\|_{\mathbf{H}}} \leq \min_{\deg(p)<k} \max_{\lambda\in\Lambda} |r(\lambda) - p(\lambda)|.$$

Proof. Since $\text{lan-OR}_k(r, R)$ is the $\mathbf{H}$-norm optimal approximation over the Krylov subspace, we have

$$\|r(\mathbf{A})\mathbf{v} - \text{lan-OR}_k(r, R)\|_{\mathbf{H}} = \min_{\mathbf{x}\in K_k} \|r(\mathbf{A})\mathbf{v} - \mathbf{x}\|_{\mathbf{H}} = \min_{\deg(p)<k} \|r(\mathbf{A})\mathbf{v} - p(\mathbf{A})\mathbf{v}\|_{\mathbf{H}}$$

Next, using the fact that $\mathbf{A}$ and $\mathbf{H}^{1/2}$ commute, we note that,

$$\|r(\mathbf{A})\mathbf{v} - p(\mathbf{A})\mathbf{v}\|_{\mathbf{H}} = \|(r(\mathbf{A}) - p(\mathbf{A}))\mathbf{H}^{1/2}\mathbf{v}\| \leq \|r(\mathbf{A}) - p(\mathbf{A})\|\|\mathbf{v}\|_{\mathbf{H}}.$$

Finally, using the definition of the spectral norm,

$$\|r(\mathbf{A}) - p(\mathbf{A})\| = \|(r - p)(\mathbf{A})\| = \max_{\lambda\in\Lambda} |r(\lambda) - p(\lambda)|.$$

The result follows. □

It is not yet apparent that the iterate can be computed efficiently. Indeed the expression involves the terms $[\tilde{N}(\widehat{\mathbf{T}})]_{:k,:k}$ and $[\tilde{M}(\widehat{\mathbf{T}})]_{:k,:k}$, and computing $\widehat{\mathbf{T}}$ requires running Lanczos to completion. However, since $\hat{T}$ is tridiagonal, these terms can be computed without much more information than is in $\mathbf{T}$ and the Lanczos-OR iterate can be computed with only a few more matrix vector products than required to compute the Lanczos-FA iterate.

Lemma 5.6. *Suppose p is a polynomial with $q := \deg(p) > 0$. Then $[p(\widehat{\mathbf{T}})]_{:k,:k}$ can be computed using the coefficients generated by $k + \lfloor (q-1)/2 \rfloor$ iterations of Lanczos. Moreover, if $k' := k + \lfloor q/2 \rfloor$, then*

$$[p(\widehat{\mathbf{T}})]_{:k,:k} = [p([\widehat{\mathbf{T}}]_{:k',:k'})]_{:k,:k}.$$

Proof. Note that after $k + d$ iterations of Lanczos, one obtains $[\widehat{\mathbf{T}}]_{:k+d+1,:k+d}$. The result then follows immediately as a special case of Corollary 10.4. □

5.3.1 Relation of Lanczos-OR to QMR on inverse quadratics

The Lanczos-OR approximation to $r(x) = 1/(x^2 + |z|^2)$ is closely related to the approximation of two linear systems by QMR.

Lemma 5.7. *Suppose $z \in \mathbb{R}$ and define $r = 1/(x^2 + |z|^2)$, $R^{\pm} = x \pm \boldsymbol{i}z$, and $r^{\pm} = 1/R^{\pm}$. Then, for all $k \geq 1$,*

$$\text{lan-OR}_k(r, 1) = \frac{1}{2\boldsymbol{i}z} \left(\text{qmr}_k(-\boldsymbol{i}z) - \text{qmr}_k(\boldsymbol{i}z)\right).$$

Proof. Observe that

$$\text{lan-OR}_k(r^{\pm}, R^{\mp}) = \mathbf{Q}([\widehat{\mathbf{T}}^2 + |z|^2\mathbf{I}]_{:k,:k})^{-1}[\mathbf{T} \mp \boldsymbol{i}z\mathbf{I}]_{:k,:k}\mathbf{e}_0$$

so that

$$\text{qmr}_k(-\boldsymbol{i}z) - \text{qmr}_k(\boldsymbol{i}z) = 2\boldsymbol{i}z\mathbf{Q}([\widehat{\mathbf{T}}^2 + |z|^2\mathbf{I}]_{:k,:k})^{-1}\mathbf{e}_0 = 2\boldsymbol{i}z\text{lan-OR}_k(r, 1).$$

The result is then obtained by rearranging the above expression. □

Whether it is better to use Lanczos-OR with r and $R = 1$ or with $r^{\pm}$ and $R^{\pm}$ (i.e. QMR) is somewhat unclear. The Lanczos-OR based approach avoids the need for complex arithmetic, which simplifies implementation slightly. However, since QMR has been studied longer, it is likely to have more practical low-memory implementations.

5.4 Error estimates for Lanczos-OR

We now describe an approach for estimating the Lanczos-OR error. This approach has been widely studied for estimating the $\mathbf{A}$-norm of the error in CG, and we refer to [ST02; MT18; EOS19; MPT21] and the references within for more details. Note that several of these works also study whether such estimates are still reasonable in finite precision arithmetic as well as how to derive estimates for other norms such as the 2-norm.

Theorem 5.8. *Let r be a rational function and R a polynomial. Write $r(x) = M/N$ where N has leading coefficient one and M does not divide N and define $\tilde{M} = MR$ and $\tilde{N} = NR$. Suppose $\mathbf{H} = \tilde{N}(\mathbf{A})$ is positive definite. Then the Lanczos-OR iterates satisfy,*

$$\|r(\mathbf{A})\mathbf{v} - \text{lan-OR}_k(r, R)\|_{\mathbf{H}}^2 = \sum_{i=k}^{n-1} \|\text{lan-OR}_i(r, R) - \text{lan-OR}_{i+1}(r, R)\|_{\mathbf{H}}^2.$$

Proof. Let $\{\mathbf{p}_i\}_{i=0}^{n-1}$ be an $\mathbf{H}$-orthonormal set satisfying

$$\operatorname{span}\{\mathbf{p}_0, \dots, \mathbf{p}_i\} = \mathcal{K}_{i+1}$$

for all $i = 0, 1, \dots, n-1$, and decompose $r(\mathbf{A})\mathbf{v}$ as

$$r(\mathbf{A})\mathbf{v} = \sum_{i=0}^{n-1} c_i \mathbf{p}_i.$$

Theorem 5.2 asserts that $\text{lan-OR}_k(r, R)$ is the $\mathbf{H}$-norm optimal approximation to $r(\mathbf{A})\mathbf{v}$ from $\mathcal{K}_k$. Thus, for all $j = 0, 1, \dots, n-1$,

$$\text{lan-OR}_j(r, R) = \sum_{i=0}^{j-1} c_i \mathbf{p}_i.$$

This implies that

$$r(\mathbf{A})\mathbf{v} - \text{lan-OR}_k(r, R) = r(\mathbf{A})\mathbf{v} - \sum_{i=0}^{k-1} c_i \mathbf{p}_i = \sum_{i=k}^{n-1} c_i \mathbf{p}_i.$$

so that, by the $\mathbf{H}$-orthonormality of the $\{\mathbf{p}_i\}_{i=0}^{n-1}$,

$$\|r(\mathbf{A})\mathbf{v} - \text{lan-OR}_k(r, R)\|_{\mathbf{H}}^2 = \sum_{i=k}^{n-1} c_i^2.$$

But we also have

$$\text{lan-OR}_i(r, R) - \text{lan-OR}_{i+1}(r, R) = -c_i \mathbf{p}_i$$

so that

$$\|\text{lan-OR}_i(r, R) - \text{lan-OR}_{i+1}(r, R)\|_{\mathbf{H}}^2 = c_i^2. \qquad \square$$

Note that Theorem 5.8 implies that

$$\|r(\mathbf{A})\mathbf{v} - \text{lan-OR}_k(r, R)\|_{\mathbf{H}}^2 \geq \sum_{i=k}^{k+d-1} \|\text{lan-OR}_i(r, R) - \text{lan-OR}_{i+1}(r, R)\|_{\mathbf{H}}^2. \tag{5.3}$$

While this is a lower bound, if we assume that

$$\|r(\mathbf{A})\mathbf{v} - \text{lan-OR}_{k+d}(r, R)\|_{\mathbf{H}}^2 = \sum_{i=k+d}^{n-1} \|\text{lan-OR}_i(r, R) - \text{lan-OR}_{i+1}(r, R)\|_{\mathbf{H}}^2$$

is negligible compared to $\|r(\mathbf{A})\mathbf{v} - \text{lan-OR}_k(r, R)\|_{\mathbf{H}}^2$, then (5.3) becomes an approximate equality.

Typically d can be taken as a small constant, say $d = 5$, so the extra work required to obtain these estimate is not too large.

Remark 5.9. As shown in [ER21], a similar approach can be used for general matrix functions. In particular,

$$\|r(\mathbf{A})\mathbf{v} - \mathrm{alg}_k\| \le \|r(\mathbf{A})\mathbf{v} - \mathrm{alg}_{k+d}\| + \|\mathrm{alg}_{k+d} - \mathrm{alg}_k\| \approx \|\mathrm{alg}_{k+d} - \mathrm{alg}_k\|$$

provided that $\|r(\mathbf{A})\mathbf{v} - \mathrm{alg}_{k+d}\| \ll \|r(\mathbf{A})\mathbf{v} - \mathrm{alg}_k\|$. Note, however, that in situations where the convergence of alg_k is oscillatory, it may be hard to guarantee $\|r(\mathbf{A})\mathbf{v} - \mathrm{alg}_{k+d}\| \ll \|r(\mathbf{A})\mathbf{v} - \mathrm{alg}_k\|$, even if d is large. △

5.4.1 Numerical experiment

We choose $\mathbf{A}$ with eigenvalues from the model problem (10.1) with $n = 300$, $\rho = 0.8$, and $\kappa = 1000$ and $\mathbf{b}$ with equal projection onto each eigencomponent. We set $r(x) = 1/(x^2 + 1)$, and run Lanczos-OR using $R = 1$ with and without reorthgonalization. In each case, we compute (5.3) with $d = 4$.

The resulting estimates, shown in Figure 5.1, are accurate for most iterations, with larger error in initial iterations where the true Lanczos-OR error is not decreasing as quickly as in later iterations. Interestingly, the bound seems to work well in finite precision arithmetic. Understanding this further is of interest. In particular, a unified analysis of Lanczos-OR could provide more information about MINRES, for which existing bounds in finite precision arithmetic are somewhat weaker than CG.

5.5 Implementing Lanczos-OR using low memory

We now describe a low-memory implementation of Lanczos-OR which is similar in spirit to CG, MINRES, and QMR.

For convenience, we will denote $\mathbf{M} := [\tilde{M}(\widehat{\mathbf{T}})]_{:k,:k}$ and $\mathbf{N} := [\tilde{N}(\widehat{\mathbf{T}})]_{:k,:k}$ so that the Lanczos-OR output is given by $\mathbf{Q}\mathbf{N}^{-1}\mathbf{M}\mathbf{e}_0$. Then, at a high level, our approach is to:

- Take one iteration of Lanczos to generate one more column of $\widehat{\mathbf{Q}}$ and $\widehat{\mathbf{T}}$
- Compute one more column of each of $\mathbf{M}$ and $\mathbf{N}$

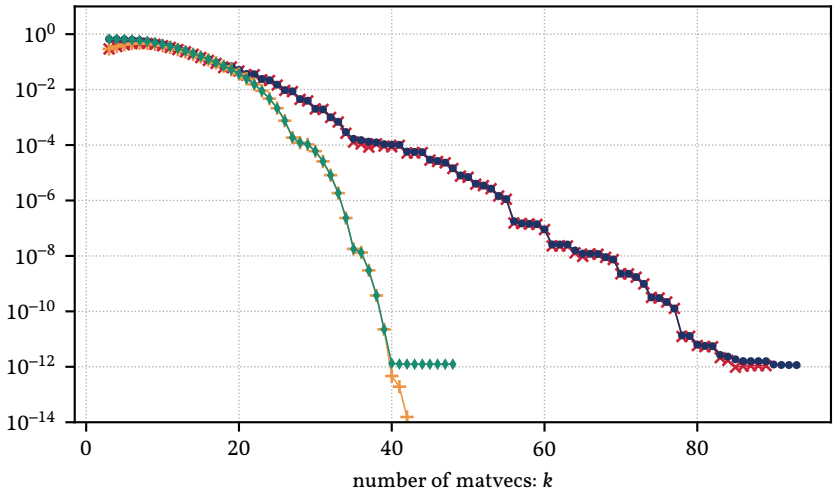


Figure 5.1: Error estimates for Lanczos-OR for $r(x) = 1/(x^2+1)$ and $R(x) = 1$. *Legend*: Lanczos-OR error with reorthogonalization ( ) and corresponding estimate (5.3) with $d = 4$ ( ). Lanczos-OR error without reorthogonalization ( ) and corresponding estimate (5.3) with $d = 4$ ( ). *Takeaway*: The error estimates are remarkably accurate, even in finite precision arithmetic.

- Compute one more factor of $\mathbf{L}^{-1} = \mathbf{L}_{k-1} \cdots \mathbf{L}_1 \mathbf{L}_0$ and one more entry of $\mathbf{D}$ where $\mathbf{L}$ and $\mathbf{D}$ are defined by the LDL factorization $\mathbf{N} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathsf{H}}$
- Compute one more term of the sum:

$$\mathbf{Q}\mathbf{N}^{-1}\mathbf{M}\mathbf{e}_0 = \mathbf{Q}\mathbf{L}^{-1}\mathbf{D}^{-1}\mathbf{L}^{-\mathsf{H}}\mathbf{M}\mathbf{e}_0 = \sum_{i=0}^{k-1} \frac{[\mathbf{L}^{-\mathsf{H}}\mathbf{M}\mathbf{e}_0]_i}{[\mathbf{D}]_{i,i}} [\mathbf{Q}\mathbf{L}^{-1}]_{:,i}$$

There are two critical observations which must be made in order to see that this gives a memory-efficient implementation. The first is that, since $\widehat{\mathbf{T}}$ is tridiagonal, $\mathbf{M}$, $\mathbf{N}$, and therefore $\mathbf{L}$ are all of half-bandwidth $q := \max(\deg(\tilde{M}), \deg(\tilde{N}))$. This means that it is possible to compute the entries of $\mathbf{D}$ and the factors of $\mathbf{L}^{-1} = \mathbf{L}_{k-1} \cdots \mathbf{L}_1 \mathbf{L}_0$ one by one as we get the entries of $\widehat{\mathbf{T}}$. The second is that because $\mathbf{L}$ is of bandwidth q, we can compute $[\mathbf{Q}\mathbf{L}^{-1}]_{:,i}$ without saving all of $\mathbf{Q}$. More specifically, $[\mathbf{L}^{-1}\mathbf{M}\mathbf{e}_0]_i$ and $[\mathbf{Q}\mathbf{L}^{-1}]_{:,i}$ can respectively be computed from $\mathbf{L}_{j-1} \cdots \mathbf{L}_1 \mathbf{L}_0 \mathbf{M}\mathbf{e}_0$ and $\mathbf{Q}\mathbf{L}_0^{\mathsf{H}}\mathbf{L}_1^{\mathsf{H}} \cdots \mathbf{L}_{k-1}^{\mathsf{H}}$ and can therefore be maintained iteratively as the factors of $\mathbf{L}^{-1}$ are computed. Moreover, because of the banded structure

of the factors $\mathbf{L}_i$, we need only maintain a sliding window of the columns of $\mathbf{Q}\mathbf{L}^{-1}$ which will allow us to access the relevant columns when we need them and discard them afterwards.

For clarity of exposition, we only describe how to compute $\mathbf{M}$ and $\mathbf{N}$ in the case that $\tilde{M}$ and $\tilde{N}$ are degree at most two. The rest of the subroutines are fully described for any degree. The syntax we use follows Python and other object oriented languages closely.

5.5.1 Computing LDL factorization

For the time being, we will assume that we can sequentially access the rows of $\mathbf{M}$ and $\mathbf{N}$. Our first step is to compute an LDL factorization of $\mathbf{N}$. A LDL factorization can be computed via a symmetrized version of Gaussian elimination and is guaranteed to exist if $\mathbf{N}$ is positive definite [Hig02]. Gaussian elimination can be viewed as transforming the starting matrix $\mathbf{N}_0 = \mathbf{N}$ to a diagonal matrix $\mathbf{N}_{k-1} = \mathbf{D}$ via a sequence of row and column operations

$$\mathbf{N}_{i+1} = \mathbf{L}_i \mathbf{N}_i \mathbf{L}_i^{\mathsf{H}}$$

where

$$\mathbf{L}_i := \mathbf{I}_k + \mathbf{l}_i \mathbf{e}_i^{\mathsf{H}}, \qquad \mathbf{l}_i := \Bigg[\underbrace{0, \cdots, 0}_{i+1 \text{ zeros}}, -\frac{[\mathbf{N}_i]_{i+1,i}}{[\mathbf{N}_i]_{i,i}}, \cdots, -\frac{[\mathbf{N}_i]_{k-1,i}}{[\mathbf{N}_i]_{i,i}} \Bigg]^{\mathsf{H}}.$$

Note that the entries of $\mathbf{L}_i$ are chosen to introduce zeros to the i-th row and column of $\mathbf{N}_i$ such that $[\mathbf{N}_{i+1}]_{:i+1,:i+1}$ is diagonal. Therefore, if the algorithm terminates successfully, we will have obtained a factorization

$$\mathbf{D} = (\mathbf{L}_{k-1} \cdots \mathbf{L}_1 \mathbf{L}_0) \mathbf{N} (\mathbf{L}_0^{\mathsf{H}} \mathbf{L}_1^{\mathsf{H}} \cdots \mathbf{L}_{n-1}^{\mathsf{H}})$$

where $\mathbf{D}$ is diagonal and each $\mathbf{L}_i$ is unit lower triangular. To obtain the factorization $\mathbf{N} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathsf{H}}$, simply define $\mathbf{L} := (\mathbf{L}_{k-1} \cdots \mathbf{L}_1 \mathbf{L}_0)^{-1}$ and note that

$$\mathbf{L} = \mathbf{I}_k - \sum_{i=0}^{k-1} \mathbf{l}_i \mathbf{e}_i^{\mathsf{H}}.$$

We remark that that $\mathbf{l}_{k-1}$ is the zeros vector and is only included in sums for ease of indexing later on. For further details on LDL factorizations, we refer readers to [Hig02].

To implement a LDL factorization, observe that the procedure above defines a recurrence

$$[\mathbf{D}]_{j,j} = [\mathbf{N}]_{j,j} - \sum_{\ell=0}^{j-1} [\mathbf{L}_{j,\ell}]^2 [\mathbf{D}]_{\ell,\ell}$$

$$[\mathbf{L}]_{i,j} = \frac{1}{[\mathbf{D}]_{j,j}} \left([\mathbf{N}]_{i,j} - \sum_{\ell=0}^{j-1} [\mathbf{L}]_{j,\ell} [\mathbf{L}]_{i,\ell} [\mathbf{D}]_{\ell,\ell} \right), \qquad i > j.$$

We therefore have Algorithm 5.1.

Algorithm 5.1 LDL factorization

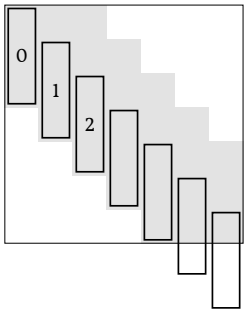
1: **procedure** LDL($\mathbf{N}$)
2: **for** $j = 0, 1, \dots, k-1$ **do**
3: $[\mathbf{D}]_{j,j} = [\mathbf{N}]_{j,j} - \sum_{\ell=0}^{j-1} [\mathbf{L}_{j,\ell}]^2 [\mathbf{D}]_{\ell,\ell}$
4: $[\mathbf{L}]_{j,j} = 1$
5: **for** $i = j+1, j+2, \dots, k-1$ **do**
6: $[\mathbf{L}]_{i,j} = (1/[\mathbf{D}]_{j,j})([\mathbf{N}]_{i,j} - \sum_{\ell=0}^{j-1} [\mathbf{L}]_{j,\ell} [\mathbf{L}]_{i,\ell} [\mathbf{D}]_{\ell,\ell})$
7: **return** $\mathbf{L}, \mathbf{D}$

Streaming version

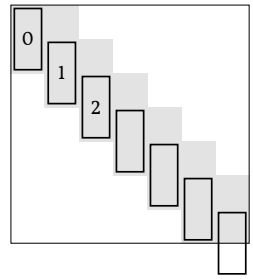
The fact that $\mathbf{L}$ has the same half bandwidth as $\mathbf{N}$ allows a more efficient implementation of Algorithm 5.1 where terms which are known to be zero are not computed and only the important diagonals of $\mathbf{L}$ are stored. Moreover, Algorithm 5.1 already only accesses $\mathbf{N}$ one column at a time so it is easily converted to a streaming algorithm. Making these changes gives the implementation Algorithm 5.2 which is fed a stream of the columns of $\mathbf{N}$ in order, as shown in Figure 5.2a. Here the diagonal of $\mathbf{D}$ is stored as $\mathtt{d}$ and the $(j+1)$-st diagonal of $\mathbf{L}$ is stored as $[\mathtt{L}]_{j,:}$. Thus, $\mathbf{L}_{i,j} = [\mathtt{L}]_{i-j-1,j}$ as long as $i - j \in 0 : q+1$.

5.5.2 Inverting the LDL factorization

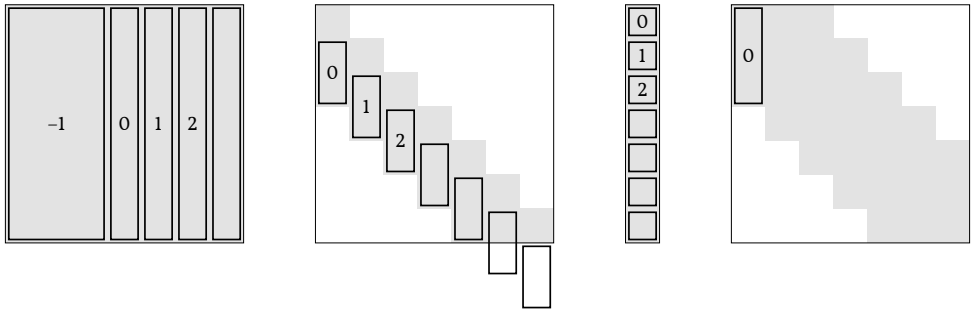
Once we have computed a a factorization $\mathbf{N} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathsf{H}}$, we can easily evaluate $\mathbf{Q}\mathbf{L}^{-1}\mathbf{D}^{-1}\mathbf{L}^{-\mathsf{H}}\mathbf{M}\mathbf{e}_1$ using the fact that $\mathbf{L}^{-1} = \mathbf{L}_{k-1} \cdots \mathbf{L}_1 \mathbf{L}_0$. Moreover, because the $\mathbf{L}_j$ can be computed one at a time, there is hope that we can derive a memory efficient implementation.



(a) Pattern for **N** in Algorithm 5.2.



(b) Pattern for **T** in Algorithm 5.5.



(c) Pattern for **Q**, **L**, **d**, **M** in Algorithm 5.3.

Figure 5.2: Access patterns for inputs to streaming functions used in low-memory implementations of Lanczos-OR and Lanczos-FA. Indices indicate what information should be streamed into the algorithm at the given iteration.

Algorithm 5.2 Streaming LDL factorization

class STREAMING-LDL(q, k)
  stream: $[\mathbf{N}]_{0,0:q+1}, [\mathbf{N}]_{1,1:q+2}, \dots, [\mathbf{N}]_{k-1,k-1:q+k-1}$
  $\mathtt{L} = \textsc{zeros}(q, k)$
  $\mathtt{d} = \textsc{zeros}(k)$
  $\mathtt{j} \leftarrow 0$
  procedure READ-STREAM($\mathbf{n}$)
    $[\mathtt{d}]_{\mathtt{j}} \leftarrow [\mathbf{n}]_0 - \sum_{\ell=\max(0,\mathtt{j}-q)}^{\mathtt{j}-1} [\mathtt{L}]_{\mathtt{j}-\ell-1,\ell}^2 [\mathtt{d}]_\ell$
    for $i = \mathtt{j}+1, \mathtt{j}+2, \dots, \min(\mathtt{j}-q, n-1)$ **do**
      $[\mathtt{L}]_{i-\mathtt{j}-1,\mathtt{j}} \leftarrow (1/[\mathtt{d}]_{\mathtt{j}})([\mathbf{n}]_{i-\mathtt{j}} - \sum_{\ell=\max(0,i-q)}^{i-1} [\mathtt{L}]_{i-\ell-1,\ell} [\mathtt{L}]_{\mathtt{j}-\ell-1,\ell} [\mathtt{d}]_\ell)$
    $\mathtt{j} \leftarrow \mathtt{j}+1$

Towards this end, define $\mathbf{y}_j := \mathbf{L}_{j-1} \cdots \mathbf{L}_1 \mathbf{L}_0 \mathbf{M} \mathbf{e}_0$ and $\mathbf{X}_j := \mathbf{Q} \mathbf{L}_0^{\mathsf{H}} \mathbf{L}_1^{\mathsf{H}} \cdots \mathbf{L}_{j-1}^{\mathsf{H}}$. Then, setting $\mathbf{y}_0 = \mathbf{M}\mathbf{e}_1$ we have that

$$\mathbf{y}_{j+1} = \mathbf{L}_j \mathbf{y}_j = (\mathbf{I} + \mathbf{l}_j \mathbf{e}_j^{\mathsf{H}}) \mathbf{y}_j = \mathbf{y}_j + (\mathbf{e}_j^{\mathsf{H}} \mathbf{y}_j) \mathbf{l}_j.$$

Similarly, setting $\mathbf{X}_0 = \mathbf{Q}$ we have that

$$\mathbf{X}_{j+1} = \mathbf{X}_j \mathbf{L}_j^{\mathsf{H}} = \mathbf{X}_j (\mathbf{I} + \mathbf{e}_j \mathbf{l}_j^{\mathsf{H}}) = \mathbf{X}_j + \mathbf{X}_j \mathbf{e}_j \mathbf{l}_j^{\mathsf{H}}.$$

Then $\mathbf{Q}\mathbf{L}^{-1}\mathbf{D}^{-1}\mathbf{L}^{-\mathsf{H}}\mathbf{M}\mathbf{e}_1 = \mathbf{X}_k \mathbf{D}^{-1} \mathbf{y}_k$ can be computed accessing $\mathbf{L}$, and therefore $\mathbf{N}$, column by column.

Streaming version

Recall that $[\mathbf{l}_i]_{:,\ell}$ is zero if $\ell \le i$ or $\ell > i + q$. Since $[\mathbf{l}_i]_{:i}$ is zero, we have

$$[\mathbf{y}_j]_j = [\mathbf{y}_j + (\mathbf{e}_j^{\mathsf{H}} \mathbf{y}_j) \mathbf{l}_j]_j = [\mathbf{y}_{j+1}]_j = \cdots = [\mathbf{y}_k]_j$$

and

$$[\mathbf{X}_j]_{:,j} = [\mathbf{X}_j + \mathbf{X}_j \mathbf{e}_j \mathbf{l}_j^{\mathsf{H}}]_{:,j} = [\mathbf{X}_{j+1}]_{:,j} = \cdots = [\mathbf{X}_k]_{:,j}.$$

We therefore have that

$$\mathbf{X}_k \mathbf{D}^{-1} \mathbf{y}_k = \sum_{j=0}^{k-1} \frac{[\mathbf{y}_k]_j}{[\mathbf{D}]_{j,j}} [\mathbf{X}_k]_{:,j} = \sum_{j=0}^{k-1} \frac{[\mathbf{y}_j]_j}{[\mathbf{D}]_{j,j}} [\mathbf{X}_j]_{:,j}.$$

Algorithm 5.3 Streaming banded product

```
1:  class STREAMING-BANDED-PROD(n, k, q)
2:      stream:
3:      X_ ← ZEROS(n, q + 1)
4:      y_ ← ZEROS(q + 1)
5:      out ← ZEROS(n)
6:      j ← −1
7:      procedure READ-STREAM(v, l, d, y_0)
8:          if j = −1 then
9:              [X_]_{:,:q} = v
10:         else
11:             if i = −1 then
12:                 y_ ← y_0
13:             out ← out + ([y_]_0/d)[X_]_{:,0}
14:             [y_]_{:q} ← [y_]_{1:} − [y_]_0 l
15:             [y_]_{−1} ← 0
16:             [X_]_{:,−1} ← v
17:             [X_]_{:,:q} ← [X_]_{:,1:} + [X_]_{:,0} l^H
18:         j ← j + 1
```

Similarly, since $[\mathbf{l}_i]_{i+q+1:}$ is zero,

$$[\mathbf{y}_j]_{j+q:} = [\mathbf{y}_{j-1} + (\mathbf{e}_{j-1}^{\mathsf{H}}\mathbf{y}_{j-1})\mathbf{l}_{j-1}]_{j+q:} = [\mathbf{y}_{j-1}]_{j+q:} = \cdots = [\mathbf{y}_0]_{j+q:}$$

and

$$[\mathbf{X}_j]_{:,j+q:} = [\mathbf{X}_{j-1} + \mathbf{X}_{j-1}\mathbf{e}_{j-1}\mathbf{l}_{j-1}^{\mathsf{H}}]_{:,j+q:} = [\mathbf{X}_{j-1}]_{:,j+q:} = \cdots = [\mathbf{X}_0]_{:,j+q:}.$$

By definition, $\mathbf{y}_0 = \mathbf{M}\mathbf{e}_0$ and $\mathbf{X}_0 = \mathbf{Q}$. Thus, we see that it is not necessary to know the later columns of $\mathbf{X}_j$ immediately.

We can define a streaming algorithm by maintaining only the relevant portions of the $\mathbf{X}_i$ and $\mathbf{y}_i$. Towards this end, define the length $q+1$ vector $\bar{\mathbf{y}}_j := [\mathbf{y}_j]_{j:j+q+1}$ the $n \times (q+1)$ matrix $\bar{\mathbf{X}}_j = [\mathbf{X}_j]_{:,j:j+q+1}$. Using the above observations, we see that these quantities can be maintained by the recurrences

$$\bar{\mathbf{y}}_j = \begin{bmatrix} [\bar{\mathbf{y}}_{j-1}]_{1:} \\ 0 \end{bmatrix} + [\bar{\mathbf{y}}_{j-1}]_0 [\mathbf{l}_j]_{j+1:j+q+1}$$

and

$$[\bar{\mathbf{X}}_j]_{:,:q} = [\bar{\mathbf{X}}_{j-1}]_{:,1:} + ([\bar{\mathbf{X}}_{j-1}]_{:,1})([\mathbf{l}_j]_{j+1:j+q+1})^{\mathsf{H}}, \qquad [\bar{\mathbf{X}}_j]_{:,q} = [\mathbf{Q}]_{j+q}.$$

Note then that,

$$\mathbf{X}_{k-1}\mathbf{D}^{-1}\mathbf{y}_{k-1} = \sum_{j=0}^{k-1} \frac{[\bar{\mathbf{y}}_j]_1}{[\mathbf{D}]_{j+1,j+1}} [\bar{\mathbf{X}}_j]_{:,1}.$$

This results in Algorithm 5.3 whose streaming pattern is outlined in Figure 5.2c.

Algorithm 5.4 Streaming banded inverse

```
1:  class STREAMING-BANDED-INV(n, k, q)
2:      stream:
3:      LDL ← STREAMING-LDL(k, q)
4:      Q0 ← ZEROS(n, q)
5:      j ← 0
6:      procedure READ-STREAM(q, n, y_0)
7:          if j < q then
8:              [Q0]_{:,j} ← v
9:              if j = q − 1 then
10:                 b-prod ← STREAMING-BANDED-PROD(n, k, q)
11:                 b-prod.READ-STREAM(V0, none, none, none)
12:         else
13:             LDL.READ-STREAM(n)
14:             b-inv.READ-STREAM(q, −[LDL.L]_{:,j−q}, [LDL.d]_{j−q}, y_0)
15:         j ← j + 1
```

5.5.3 Computing polynomials in T

The last major remaining piece is to construct $\mathbf{M} = \tilde{M}(\mathbf{T})$ and $\mathbf{N} = [\tilde{N}(\widehat{\mathbf{T}})]_{:k,:k}$. Recall that we have assumed $\tilde{M}$ and $\tilde{N}$ are of degree at most two for convenience. In iteration ℓ of Lanczos, we obtain α_ℓ and β_ℓ. Observe that $\mathbf{T}^2$ is symmetric and

that, defining $\beta_{-1} = \beta_k = 0$, the lower triangle is given by

$$[\mathbf{T}^2]_{i,j} = \begin{cases} \beta_{j-1}^2 + \alpha_j^2 + \beta_j^2 & j = i \\ (\alpha_j + \alpha_{j+1})\beta_i & j = i-1 \\ \beta_j \beta_{j+1} & j = i-2 \\ 0 & \text{o.w.} \end{cases}$$

We can use this to implement the streaming algorithm, Algorithm 5.5, for computing the entries of $\mathbf{T}^2$. Rather than being fed the entire tridiagonal matrix $\mathbf{T}$, Algorithm 5.5 is fed a stream of the columns of $\mathbf{T}$ in order, as shown in Figure 5.2b. The algorithm respectively stores the j-th diagonals of $\mathbf{T}$ and $\mathbf{T}^2$ as $[\mathtt{T}]_{j,:}$ and $[\mathtt{Tp2}]_{j,:}$.

Algorithm 5.5 Streaming tridiagonal square

```
1:  class STREAMING-TRIDIAG-SQUARE(k)
2:      stream: (α_0, β_0), ..., (α_{k-1}, β_{k-1})
3:      T ← ZEROS(2, k)
4:      Tp2 ← ZEROS(3, k)
5:      j ← 0
6:      procedure READ-STREAM(α, β)
7:          [T]_{0,j} = α
8:          [T]_{1,j} = β
9:          if i = 0 then
10:             [Tp2]_{0,j} ← [T]²_{0,j} + [T]²_{1,j}
11:         else
12:             [Tp2]_{0,j} ← [T]²_{0,j} + [T]²_{1,j} + [T]²_{1,j-1}
13:             [Tp2]_{1,j} ← ([T]_{0,j} + [T]_{0,j-1})[T]_{1,j-1}
14:             [Tp2]_{2,j} ← [T]_{1,j}[T]_{1,j-1}
15:         j ← j + 1
```

Since we maintain the columns of $\mathbf{T}^2$ with Algorithm 5.5, we can easily compute $\mathbf{M}$ and $\mathbf{N}$ using Algorithm 5.6.

Algorithm 5.6 Get polynomial of tridiagonal matrix

1: **procedure** GET-POLY(P, STp2, k, j)
2: $a, b, c = P(0), P'(0), P''(0)$
3: $\mathbf{p} \leftarrow$ ZEROS(3)
4: $[\mathbf{p}]_{:3} \leftarrow a[\texttt{STp2.Tp2}]_{:,j}$
5: $[\mathbf{p}]_{:2} \leftarrow b[\texttt{STp2.T}]_{:,j}$
6: $[\mathbf{p}]_{:1} \leftarrow c$

5.5.4 Putting it all together

With this algorithm in place, putting everything together is straightforward, and the full implementation is shown in Algorithm 5.7.

This can be incorporated into any Lanczos implementation and used to compute the Lanczoz-OR iterates. For concreteness, we show this with the implementation of Lanczos from Algorithm 1.1. We call the resulting implementation Lanczos-OR-lm.

We can easily obtain an implementation of Lanczos-FA, which we call Lanczos-FA-lm, by replacing β_{k-1} with 0 in the final iteration of the loop.

5.5.5 Some comments on implementation

Our main goal is to describe how to implement Lanczos-FA and Lanczos-OR in a way that requires k matrix-vector products and $O(n)$ storage, when M and N are each at most degree two. As mentioned, the approach can be extended to any constant degree. There are a range of improvements to our implementation which may be useful in practice.

First, the amount of storage used can be reduced somewhat. Indeed, the implementation described above saves $\mathbf{T}$, $\mathbf{T}^2$, $\mathbf{L}$, and $\mathbf{d}$, but only accesses a sliding window of these quantities. We have chosen to save them for convenience since they require only $O(k)$ storage. However, storing only the relevant information from these quantities would result in an implementation with storage costs independent of the number of iterations k. In this vein, a practical implementation would likely determine k adaptively by monitoring the residual or other

Algorithm 5.7 Streaming banded rational inverse

```
1:  class BANDED-RATIONAL(n, k, M̃, Ñ)
2:    b-inv ← BANDED-INV(n, k, 2)
3:    STp2 ← STREAMING-TRIDIAG-SQUARE(k)
4:    j ← 0
5:    procedure READ-STREAM(q, α, β)
6:      if j < k then
7:        STp2.READ-STREAM(α, β)
8:        b-inv.READ-STREAM(
9:          q,
10:         GETPOLY(Ñ, STp2, k, j − 1) if j ≥ 2 else none,
11:         GETPOLY(M̃, STp2, k, j − 1) if j = 2 else none,
12:       )
13:     LDL.READ-STREAM(n)
14:     j ← j + 1
15:   procedure FINISH-UP()
16:     for i = k, k + 1 do
17:       b-inv.READ-STREAM(none, GETPOLY(Ñ, STp2, k, j − 1), none)
18:   procedure GET-OUTPUT()
19:     return b-inv.b-prod.out
```

measures of the error. Improvements to the number of vectors of length n may be possible as well. For example, storage could possibly be reduced somewhat by incorporating the Lanczos iteration more explicitly with the inversion of the LDL facorization, much like the classical Hestenes and Stiefel implementation of CG [HS52].

As with other short-recurrence based Krylov subspace methods, the behavior of Lanczos-FA-lm and Lanzos-OR-lm in finite precision arithmetic may be different than in exact arithmetic. However, with the exception of the Lanczos algorithm, the other aspects of our algorithm are essentially backwards stable. It is therefore more or less clear that Lanczos-FA-lm and Lanczos-OR-lm will accurately compute the expressions $\mathbf{Q}N(\mathbf{T})^{-1}M(\mathbf{T})\mathbf{e}_0$ and $\mathbf{Q}([\tilde{N}(\widehat{\mathbf{T}})]_{:k,:k})^{-1}\tilde{M}(\mathbf{T})\mathbf{e}_0$ provided that $M(\mathbf{T})$ $N(\mathbf{T})$, $[\tilde{N}(\widehat{\mathbf{T}})]_{:k,:k}$ are reasonably well conditioned. Indeed, in

Algorithm 5.8 Lanczos-OR (low memory)

procedure LANCZOS-OR-LM($\mathbf{A}, \mathbf{v}, k, M, N$)
  $\mathbf{q}_{-1} = \mathbf{0}, \beta_{-1} = 0, \mathbf{q}_0 = \mathbf{v}$
  Set $\tilde{M}$ and $\tilde{N}$ as in Theorem 5.1
  `lam-lm` $\leftarrow$ BANDED-RATIONAL($n, k, \tilde{M}, \tilde{N}$)
  for $j = 0, 1, \ldots, k-1$ **do**
    $\tilde{\mathbf{q}}_{j+1} = \mathbf{A}\mathbf{q}_j - \beta_{j-1}\mathbf{q}_{j-1}$
    $\alpha_j = \langle \tilde{\mathbf{q}}_{j+1}, \mathbf{q}_j \rangle$
    $\tilde{\mathbf{q}}_{j+1} = \tilde{\mathbf{q}}_{j+1} - \alpha_j \mathbf{q}_j$
    optionally, reorthogonalize $\tilde{\mathbf{q}}_{j+1}$ against $\{\mathbf{q}_i\}_{i=0}^{j-1}$
    $\beta_j = \|\tilde{\mathbf{q}}_{j+1}\|$
    $\mathbf{q}_{j+1} = \tilde{\mathbf{q}}_{j+1}/\beta_j$
    `lam-lm.`READ-STREAM($\mathbf{q}_j, \alpha_j, \beta_j$)
  `lam-lm.`FINISH-UP()

practice solving linear systems by symmetric Gaussian elimination is accurate; see for instance [Hig02, Chapter 10]. Thus, such bounds and techniques can be applied to Lanczos-FA-lm and Lanczos-OR-lm.

Chapter 6
General matrix function approximation

We now consider Krylov subspace methods for approximating $f(\mathbf{A})\mathbf{v}$ for more general f. In Section 6.2 we discuss the Lanczos method for matrix function approximation (Lanczos-FA), which is probably the most widely used algorithm for this task. Then, in Section 6.3, we discuss how Lanczos-OR iterates can be used to generate approximations to a wide range of functions.

6.1 Explicit polynomial methods

A simple approach to approximating $f(\mathbf{A})\mathbf{v}$ is to compute

$$[f]_{k-1}^{\circ p}(\mathbf{A})\mathbf{v} \in \mathcal{K}_k$$

where $[f]_{k-1}^{\circ p}$ is some polynomial chosen to approximate f. For instance, the Chebyshev semi-iterative method mentioned in the example from Chapter 1 falls into this category of algorithms.

If $|f - [f]_{k-1}^{\circ p}|$ is not strongly correlated with the eigenvalues of $\mathbf{A}$, we might expect

$$\|f(\mathbf{A})\mathbf{v} - [f]_{k-1}^{\circ p}(\mathbf{A})\mathbf{v}\| > c\|f - [f]_{k-1}^{\circ p}\|_I\|\mathbf{v}\|$$

for some reasonable constant c.

Like mentioned in our discussion of algorithms for quadratic forms in Section 3.4, one nice property of explicit polynomial methods is the lack of inner products, which can be expensive on supercomputers. In addition, explicit

polynomial methods for $f(\mathbf{A})\mathbf{v}$ do not require storing a basis for Krylov subspace.

6.2 Lanczos-FA

Early uses of Lanczos-FA were focused primarily on computing matrix exponentials applied to a vector; i.e. $f = \exp(tx)$. As far as we can tell, Lanczos-FA was introduced in [NW83] and first used for general f in [Vor87]. Soon after Lanczos-FA was first used, a number of papers studying the algorithm and its convergence properties were published [PL86; DK88; DK89; GS92; Saa92]. These early works were followed by a number of papers demonstrating the effectiveness of Lanczos-FA in finite precision arithmetic [DK91; DK95; DGK98], a topic we discuss further in Chapter 8.

Definition 6.1. *The k-th Lanczos-FA approximation to $f(\mathbf{A})\mathbf{v}$ is*

$$\text{lan-FA}_k(f) := \mathbf{Q}f(\mathbf{T})\mathbf{e}_0.$$

Remark 6.2. If $\mathbf{A}$ is positive definite, then the Lanczos-FA approximation to $f = 1/x$ coincides with the CG iterate defined in Section 5.2. Even when $\mathbf{A}$ is indefinite, we can use the Lanczos-FA iterate as an approximation to $\mathbf{A}^{-1}\mathbf{v}$. However, the resulting algorithm is not guaranteed to be optimal, and if $\mathbf{T}$ has an eigenvalue near to or at zero then $\mathbf{T}^{-1}$ will be poorly conditioned or even undefined. Even so, the *overall* convergence of the Lanczos-FA iterate is closely related to the convergence of MINRES. We described this phenomenon in detail in Section 7.4. △

A basic property of the Lanczos-FA iterate is that polynomials are applied exactly. More precisely, we have the following, well known, theorem.

Theorem 6.3. *Suppose* $\deg(p) < k$. *Then,*

$$\text{lan-FA}_k(p) = p(\mathbf{A})\mathbf{v}.$$

Proof. Using Corollary 10.3 we have

$$\mathbf{A}^q\mathbf{v} = \mathbf{A}^q\widehat{\mathbf{Q}}\mathbf{e}_0 = \widehat{\mathbf{Q}}\widehat{\mathbf{T}}^q\mathbf{e}_0 = \mathbf{Q}\mathbf{T}^q\mathbf{e}_0.$$

□

This theorem implies that, when $\mu = \Psi$,

$$\text{lan-FA}_k(f) = [f]^{\text{ip}}_{k-1}(\mathbf{A})\mathbf{v}.$$

In other words, the Lanczos-FA iterate is obtained by interpolating f at the eigenvalues of $\mathbf{T}$ with a degree $k-1$ polynomial.

6.2.1 A priori error bounds on an interval

Akin to the bounds we saw in Section 3.3, we can derive a bound based on best approximation on an interval.

Theorem 6.4. *The Lanczos-FA iterate satisfies*

$$\frac{\|f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f)\|}{\|\mathbf{v}\|} \leq 2 \min_{\deg(p)<k} \|f - p\|_{\mathcal{I}}.$$

Proof. For any polynomial p with $\deg(p) < k$,

$$\begin{aligned}
\|f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f)\| &\leq \|f(\mathbf{A})\mathbf{v} - p(\mathbf{A})\mathbf{v}\| + \|\text{lan-FA}_k(p) - \text{lan-FA}_k(f)\| \\
&= \|(f(\mathbf{A}) - p(\mathbf{A}))\mathbf{v}\| + \|\mathbf{Q}(p(\mathbf{T}) - f(\mathbf{T}))\mathbf{Q}^{\mathsf{H}}\mathbf{v}\| \\
&\leq \|f(\mathbf{A}) - p(\mathbf{A})\|_2\|\mathbf{v}\| + \|\mathbf{Q}(p(\mathbf{T}) - f(\mathbf{T}))\mathbf{Q}^{\mathsf{H}}\|_2\|\mathbf{v}\| \\
&\leq (\|f(\mathbf{A}) - p(\mathbf{A})\|_2 + \|p(\mathbf{T}) - f(\mathbf{T})\|_2)\|\mathbf{v}\|. \\
&= (\|f - p\|_{\Lambda} + \|f - p\|_{\Lambda(\mathbf{T})})\|\mathbf{v}\|.
\end{aligned}$$

Then, optimizing over polynomials of degree less than k,

$$\|f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f)\| \leq \min_{\deg(p)<k} \left(\|f - p\|_{\Lambda} + \|f - p\|_{\Lambda(\mathbf{T})}\right) \|\mathbf{v}\|. \tag{6.1}$$

Finally, using that $\Lambda, \Lambda(\mathbf{T}) \subset \mathcal{I}$, we obtain the result. □

As we will discuss in Chapter 8, bounds for Lanczos-FA based on polynomial approximation on $\mathcal{I}$ still hold, to close approximation, in finite precision arithmetic.

6.2.2 Two-pass Lanczos-FA

A major downside of Lanczos-FA compared with explicit polynomial approaches is that a simple implementation requires that $\mathbf{Q}$ be stored. Fortunately,

this storage cost can be avoided by incurring additional computational cost. In particular, we can use an implementation called two pass Lanczos-FA [Bor00; FS08a]. On the first pass, the tridiagonal matrix $\mathbf{T}$ is computed using the short-recurrence version of Lanczos; i.e., without storing all of $\mathbf{Q}$. Once $\mathbf{T}$ has been computed, $f(\mathbf{T})\mathbf{e}_0$ can be evaluated using $O(k^2)$ storage. Lanczos is then run again and the product $\mathbf{Q}f(\mathbf{T})\mathbf{e}_0$ is computed as the columns of $\mathbf{Q}$ become available. Note that on the second run, the *exact same* Lanczos vectors (even in finite precision arithmetic) can be computed without any inner products by using the values computed in the first run and stored in $\mathbf{T}$.

Such an approach can be generalized by re-generating the Lanczos recurrence from multiple points simultaneously on the second pass [Li22]. Specifically, on the first pass, vectors $\mathbf{q}_j$ and $\mathbf{q}_{j-1}$ can be saved for $j = 0, d, 2d, \ldots$. Then, on the second pass, the rest of the Lanczos vectors can be constructed by continuing the three-term Lanczos recurrence (1.3) from each of the roughly n/d start points in parallel. Thus, the number of matrix-loads is reduced by a factor of roughly d at the cost of storing roughly $2n/d$ vectors. The case $d = n$ gives the original two-pass approach.

6.3 Lanczos-OR based methods

We can use integral representations of functions to derive algorithms based on Lanczos-OR iterates. For concreteness, we consider the case of the matrix-sign function and rational functions in partial fraction form. It's clear that a similar approach can be applied to other functions, and further study of the resulting algorithms would be interesting.

6.3.1 The matrix sign function

We begin by noting that, for any $a > 0$,

$$\frac{1}{\sqrt{a}} = \frac{2}{\pi}\int_0^\infty \frac{1}{a + z^2}\mathrm{d}z.$$

Thus, if $f = \text{sign} = x/|x| = x/\sqrt{x^2}$, we have

$$f(\mathbf{A})\mathbf{v} = \frac{2}{\pi}\int_0^\infty \mathbf{A}(\mathbf{A}^2 + z^2\mathbf{I})^{-1}\mathbf{v}\,\mathrm{d}z.$$

The Lanczos-OR approximation to $\mathbf{A}(\mathbf{A}^2+z^2\mathbf{I})^{-1}\mathbf{v}$ is $\mathbf{Q}([\widehat{\mathbf{T}}^2]_{:k,:k}+z^2\mathbf{I})^{-1}\mathbf{T}\mathbf{e}_0$, which is optimal over Krylov subspace in the $(\mathbf{A}^2+z^2\mathbf{I})$-norm. This yields the approximation

$$\frac{2}{\pi}\int_0^\infty \mathbf{Q}([\widehat{\mathbf{T}}^2]_{:k,:k}+z^2\mathbf{I})^{-1}\mathbf{T}\mathbf{e}_0\,\mathrm{d}z = \mathbf{Q}\left([\widehat{\mathbf{T}}^2]_{:k,:k}\right)^{-1/2}\mathbf{T}\mathbf{e}_0.$$

Thus, we can define the induced iterate as

$$\mathsf{sign-OR}_k := \mathbf{Q}\left([\widehat{\mathbf{T}}^2]_{:k,:k}\right)^{-1/2}\mathbf{T}\mathbf{e}_0 = \mathbf{Q}\left([\widehat{\mathbf{T}}]_{:k,:k+1}[\widehat{\mathbf{T}}]_{:k+1,:k}\right)^{-1/2}\mathbf{T}\mathbf{e}_0. \tag{6.2}$$

Relation to Lanczos-FA

The Lanczos-OR and Lanczos-FA iterates for a given rational matrix function are clearly related. In particular, $\tilde{N}(\mathbf{T})$ and $[\tilde{N}(\widehat{\mathbf{T}})]_{:k,:k}$ differ only in the bottom rightmost $(q-1)\times(q-1)$ principle submatrix, where $q = \deg(\tilde{N})$. Using this fact, it can be shown that the Lanczos-OR and Lanczos-FA iterates "tend to coalesce as convergence takes place" [LS06, Proposition 5.1]. We now show that a similar phenomenon occurs with the induced Lanczos-OR approximation to the sign function and the Lanczos-FA approximation.

Theorem 6.5. *The Lanczos-FA and induced Lanczos-OR approximations to the matrix sign function satisfy*

$$\|\mathsf{lan-FA}_k(\mathsf{sign}) - \mathsf{sign-OR}_k\|_2 \le \beta_{k-1}^2\frac{\sqrt{\alpha_0^2+\beta_0^2}}{2\sigma_{\min}(\mathbf{T})^3}.$$

where $\sigma_{\max}(\mathbf{T})$ and $\sigma_{\min}(\mathbf{T})$ are the largest and smallest singular values of $\mathbf{T}$ respectively.

Proof. Let $N = x^2+z^2$ and $M = x$. Note that $N(\mathbf{T}) = [N(\widehat{\mathbf{T}})]_{:k,:k} - \beta_{k-1}^2\mathbf{e}_{k-1}\mathbf{e}_{k-1}^{\mathsf{H}}$ so,

$$M(\mathbf{T})\mathbf{e}_0 = ([N(\widehat{\mathbf{T}})]_{:k,:k} - \beta_{k-1}^2\mathbf{e}_{k-1}\mathbf{e}_{k-1}^{\mathsf{H}})N(\mathbf{T})^{-1}M(\mathbf{T})\mathbf{e}_0.$$

Thus, rearranging terms and multiplying by $([N(\widehat{\mathbf{T}})]_{:k,:k})^{-1}$ we find that

$$\begin{aligned} N(\mathbf{T})^{-1}M(\mathbf{T})\mathbf{e}_0 - ([N(\widehat{\mathbf{T}})]_{:k,:k})^{-1}M(\mathbf{T})\mathbf{e}_0 \\ = \beta_{k-1}^2([N(\widehat{\mathbf{T}})]_{:k,:k})^{-1}\mathbf{e}_{k-1}\mathbf{e}_{k-1}^{\mathsf{H}}N(\mathbf{T})^{-1}M(\mathbf{T})\mathbf{e}_0 \end{aligned}$$

After left multiplying with $\mathbf{Q}$, this implies that

$$\mathsf{lan-FA}_k(r) - \mathsf{lan-OR}_k(r,1) = \beta_{k-1}^2\mathbf{Q}([N(\widehat{\mathbf{T}})]_{:k,:k})^{-1}\mathbf{e}_{k-1}\mathbf{e}_{k-1}^{\mathsf{H}}N(\mathbf{T})^{-1}M(\mathbf{T})\mathbf{e}_0.$$

Now, suppose that $f = \text{sign}$ and set $\text{diff}_k := \text{lan-FA}_k(\text{sign}) - \text{sign} - \text{OR}_k$ as above. Then, since the Lanczos-FA approximation can also be induced by an integral over $z \in [0, \infty)$, we have that,

$$\text{diff}_k = \beta_{k-1}^2 \frac{2}{\pi} \int_0^\infty \mathbf{Q}([\widehat{\mathbf{T}}^2]_{:k,:k} + z^2\mathbf{I})^{-1}\mathbf{e}_{k-1}\mathbf{e}_{k-1}^{\mathsf{H}}(\mathbf{T}^2 + z^2\mathbf{I})^{-1}\mathbf{T}\mathbf{e}_0 \mathrm{d}z.$$

Note that $[\widehat{\mathbf{T}}^2]_{:k,:k} - \mathbf{T}^2 = \beta_{k-1}^2\mathbf{e}_{k-1}\mathbf{e}_{k-1}^{\mathsf{H}}$ is positive semidefinite. Therefore, using that $\sigma_{\min}([\widehat{\mathbf{T}}^2]_{:k,:k}) \geq \sigma_{\min}(\mathbf{T}^2) = \sigma_{\min}(\mathbf{T})^2$,

$$\begin{aligned}
\|\text{diff}_k\|_2 &= \beta_{k-1}^2 \left\| \mathbf{Q}\left(\frac{2}{\pi} \int_0^\infty ([\widehat{\mathbf{T}}^2]_{:k,:k} + z^2\mathbf{I})^{-1}\mathbf{e}_{k-1}\mathbf{e}_{k-1}^{\mathsf{H}}(\mathbf{T}^2 + z^2\mathbf{I})^{-1}\mathrm{d}z \right)\mathbf{T}\mathbf{e}_0 \right\|_2 \\
&\leq \beta_{k-1}^2 \left(\frac{2}{\pi} \int_0^\infty \|([\widehat{\mathbf{T}}^2]_{:k,:k} + z^2\mathbf{I})^{-1}\|_2 \|(\mathbf{T}^2 + z^2\mathbf{I})^{-1}\|_2 \mathrm{d}z \right) \|\mathbf{T}\mathbf{e}_0\|_2 \\
&\leq \beta_{k-1}^2 \left(\frac{2}{\pi} \int_0^\infty |(\sigma_{\min}(\mathbf{T})^2 + z^2)^{-1}| \, |(\sigma_{\min}(\mathbf{T})^2 + z^2)^{-1}| \mathrm{d}z \right) \sqrt{\alpha_0^2 + \beta_0^2} \\
&= \beta_{k-1}^2 \frac{\sqrt{\alpha_0^2 + \beta_0^2}}{2\sigma_{\min}(\mathbf{T})^3}.
\end{aligned}$$

□

Since $|\beta_{k-1}|$ tends to decrease as the Lanczos method converges, this seemingly implies that the induced Lanczos-OR iterate and the Lanczos-FA iterate tend to converge in this limit. However, recall that $\mathbf{T} = [\widehat{\mathbf{T}}]_{:k,:k}$ changes at each iteration k. In particular, there is the difficulty that $\mathbf{T}$ may have an eigenvalue near zero, in which case the preceding bound could be useless.

It is known that $\mathbf{T}$ cannot have small eigenvalues in two consecutive iterations, provided the eigenvalues of $\mathbf{A}$ are not small [GDK99], a result we will recall in Theorem 7.16. Since β_{k-1} has little to do with the minimum magnitude eigenvalue of $\mathbf{T}$ (recall that the Lanczos recurrence is shift invariant), we expect that the "overall" convergence of the induced Lanczos-OR iterate and the Lanczos-FA iterate will be similar as Lanczos converges.

6.3.2 Rational function approximation

We can use a similar approach to derive (non-optimal) approximations to rational matrix functions $r(\mathbf{A})\mathbf{b}$. In many settings, particularly when the rational function r is used as a proxy for a function f, this approach is more natural than computing the Lanczos-OR approximation to the rational function directly. The

convergence of such methods is closely related to the quality of the (scalar) rational function approximation as well as the quality of the approximation to the rational matrix function. Specifically, for any output $\mathsf{alg}(r)$ meant to approximate $r(\mathbf{A})\mathbf{b}$, we have the following bound:

$$\begin{aligned}\|f(\mathbf{A})\mathbf{b} - \mathsf{alg}(r)\| &\leq \|f(\mathbf{A})\mathbf{b} - r(\mathbf{A})\mathbf{b}\| + \|r(\mathbf{A})\mathbf{b} - \mathsf{alg}(r)\| \\ &\leq \|f(\mathbf{A}) - r(\mathbf{A})\|_2 \|\mathbf{b}\| + \|r(\mathbf{A})\mathbf{b} - \mathsf{alg}(r)\| \\ &\leq \|\mathbf{b}\| \max_{\lambda \in \Lambda} |f(\lambda) - r(\lambda)| + \|r(\mathbf{A})\mathbf{b} - \mathsf{alg}(r)\| \\ &\leq \underbrace{\|\mathbf{b}\| \max_{\lambda \in I} |f(\lambda) - r(\lambda)|}_{\text{approximation error}} + \underbrace{\|r(\mathbf{A})\mathbf{b} - \mathsf{alg}(r)\|}_{\text{application error}}. \end{aligned} \tag{6.3}$$

In many cases, very good or even optimal scalar rational function approximations to a given function on a single interval are known or can be easily computed. Thus, the approximation error term can typically be made small with a rational function of relatively low degree [Tre19; NST18].

Of course, this bound is only meaningful if the approximation error term is small relative to the application error. Indeed, we also have

$$\|f(\mathbf{A})\mathbf{b} - \mathsf{alg}(r)\| \geq \big|\|f(\mathbf{A})\mathbf{b} - r(\mathbf{A})\mathbf{b}\| - \|r(\mathbf{A})\mathbf{b} - \mathsf{alg}(r)\|\big|. \tag{6.4}$$

This shows that the size of $\|f(\mathbf{A})\mathbf{b} - \mathsf{alg}(r)\|$ is rouhgly the size of $\|f(\mathbf{A})\mathbf{b} - r(\mathbf{A})\mathbf{b}\|$ when $\mathsf{alg}(r)$ is a good approximation to $r(\mathbf{A})\mathbf{b}$.

As we noted, rational function approximations commonly are obtained by discretizing an integral representation using a numerical quadrature approximation. For instance, the matrix sign function may be approximated as

$$r_q(\mathbf{A})\mathbf{v} = \sum_{i=0}^{q-1} \omega_i \mathbf{A}(\mathbf{A}^2 + z_i^2 \mathbf{I})^{-1}\mathbf{v} \tag{6.5}$$

where z_i and ω_i are appropriately chosen quadrature nodes and weights [HHT08].

We can of course write $r_q = M_q/N_q$, so it's tempting to set $R_q = 1$ and $\mathbf{H}_q = N_q(\mathbf{A})$ and then use Lanczos-OR to compute the $\mathbf{H}_q$-norm optimal approximation. However, while r_q is convergent to f as $q \to \infty$, $N_q := \prod_{i=0}^{q-1}(x^2 + z_i^2)$ is not convergent to any fixed function. In fact N_q will increase in degree and $\mathbf{H}_q$ will be increasingly poorly conditioned. This presents a numerical difficulty in

computing the Lanczos-OR iterate in this limit. More importantly, it is not clear that it is meaningful to approximate a function in this way. Indeed, it seems reasonable to expect that, for fixed k, as $q \to \infty$, our approximation should be convergent to something. However, we cannot guarantee lan-$\mathrm{OR}_k(M_q, N_q)$ is convergent in this limit.

On the other hand, akin to our approach for approximating the integral described in the previous subsection, we can compute the *term-wise optimal* approximations to each term in the sum representation of r_q and output

$$\sum_{i=0}^{q-1} \omega_i \mathbf{Q}([\widehat{\mathbf{T}}^2]_{:k,:k} + z_i^2 \mathbf{I})^{-1} \mathbf{T}\mathbf{e}_0.$$

In this case, as $q \to \infty$, the approximation is convergent to the integral output. It would be interesting to understand when a term-wise optimal approximation behaves nearly optimally.

6.4 Numerical experiments

6.4.1 The matrix sign function

We now provide several examples which illustrate various aspects of the convergence properties of Lanczos-OR and Lanczos-OR based algorithms, and show when these new methods can outperform more standard techniques like the classic Lanczos-FA.

As we noted in Section 6.3.1, Lanczos-OR can be used to obtain an approximation to the matrix sign function. A related approach, which interpolates the sign function at the so called "harmonic Ritz values", is described in [Esh+02, Section 4.3]. The harmonic Ritz values are characterized by the generalized eigenvalue problem

$$[\widehat{\mathbf{T}}^2]_{:k,:k}\mathbf{y} = \theta \mathbf{T}\mathbf{y}$$

and are closely related to MINRES in the sense that MINRES produces a polynomial interpolating $1/x$ at the harmonic Ritz values [PPV95]. Finally, a standard approach is using Lanczos-FA (or equivalently, Gaussian quadrature) as described in Chapters 3 and 4.

Spectrum approximation

In this example, we show the spectrum approximations induced by the algorithms described above. We now set $\mathbf{A}$ to be a diagonal matrix with 1000 eigenvalues set to the quantiles of a Chi-squared distribution with parameters $\alpha = 1$ and $\beta = 10$. We set $k = 10$ and consider approximations to the function $c \mapsto \mathbf{v}^{\mathsf{H}} \mathbb{1}[\mathbf{A} \leq c]\mathbf{v}$ for a range of values c. Here $\mathbb{1}[x \leq c] = (1 - \text{sign}(x - c))/2$ is one if $x \leq c$ and zero otherwise. We pick $\mathbf{v}$ as a unit vector with equal projection onto each eigencomponent so that $\mathbf{v}^{\mathsf{H}} \mathbb{1}[\mathbf{A} \leq c]\mathbf{v}$ gives the fraction of eigenvalues of $\mathbf{A}$ below c. In the $n \to \infty$ limit, this function will converge pointwise to the cumulative distribution of a Chi-squared random distribution with parameters $\alpha = 1$ and $\beta = 10$. The results are shown in Figure 6.1.

Note that the Lanczos-FA based approach is piecewise constant with jumps at each eigenvalue of $\mathbf{T}$. On the other hand, the harmonic Ritz value and Lanczos-OR based approaches produce continuous approximations to the spectrum. In this particular example, the spectrum of $\mathbf{A}$ is near to a smooth limiting density, so the harmonic Ritz value and Lanczos-OR based approaches seem to produce better approximations. Note that these approximations differ from the KPM approximations in Chapter 4 in that, like Gaussian quadrature, they adapt automatically to the spectrum of $\mathbf{A}$.

We note that it is not typically possible to pick $\mathbf{v}$ with equal projection onto each eigencomponent since the eigenvectors of $\mathbf{A}$ are unknown. However, by choosing $\mathbf{v}$ from a suitable distribution, it can be guaranteed that $\mathbf{v}$ has roughly equal projection onto each eigencomponent. This is discussed thoroughly in Chapter 4.

Quality of approximation

We now study how the number of matrix vector products impact the quality of approximation for a fixed sign function.

We construct a matrix with 400 eigenvalues, 100 of which are the negatives of the values of a model problem (10.1) with parameters $\kappa = 10^2$, $\rho = 0.9$, and $n = 100$ and 300 of which are the values of a model problem with parameters $\kappa = 10^3$, $\rho = 0.8$, $n = 300$. We then compute the Lanczos-OR induced

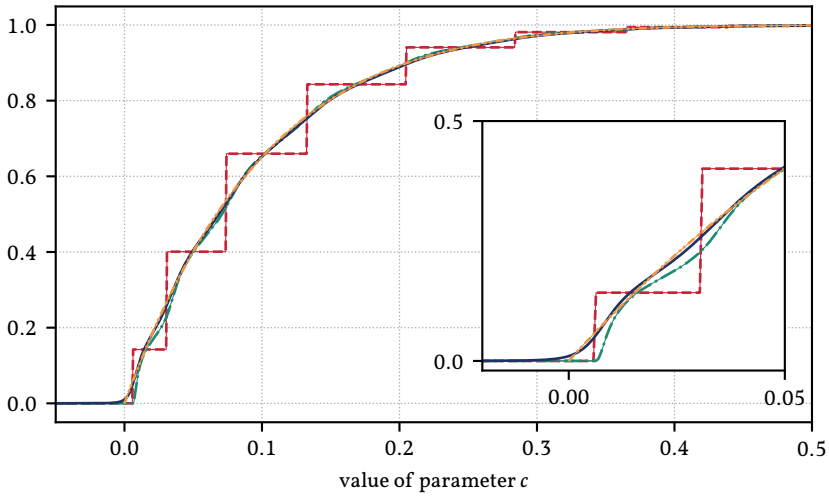


Figure 6.1: Comparison of Lanczos-based spectrum approximation algorithms. *Legend*: Lanczos-OR induced approximation (——), Lanczos-FA (GQ) (- - -), harmonic Ritz values based approximation (-·-), and limiting density (·····). *Takeaway*: The Lancos-OR and harmonic Ritz value based approximations produce smooth approximations to the spectral density.

approximation, the Lanczos-FA approximation, the harmonic Ritz value based approximation from [Esh+02], and the optimal $\mathbf{A}^2$-norm approximation to the matrix sign function. The results are shown in Figure 6.2. In all cases, we use the Lanczos algorithm with full reorthogonalization. Because eigenvalues of $\mathbf{T}$ may be near to zero, Lanczos-FA exhibits oscillatory behavior On the other hand, the Lanczos-OR based approach and the harmonic Ritz value based approach have much smoother convergence. Note that the Lanczos-OR induced approximation is not optimal, although it seems to perform close to optimally after a few iterations.

6.4.2 Rational matrix functions

We now illustrate the effectiveness of the Lanczos-OR based approach to approximating rational matrix functions described in Section 6.3.2.

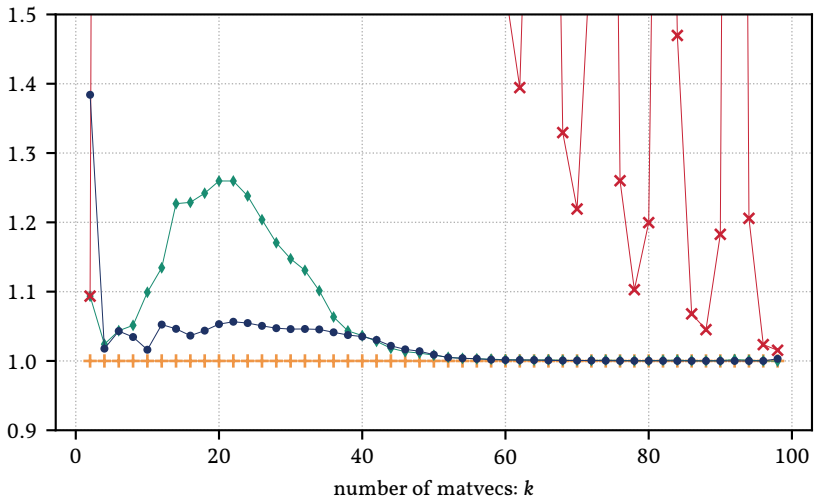


Figure 6.2: Optimality ratio for $\mathbf{A}^2$-norm errors for approximating sign(**A**)**v**. *Legend*: Lanczos-OR induced approximation (—•—), Lanczos-FA (GQ) (—×—), harmonic Ritz values based approximation (—♦—) optimal (—+—). *Takeaway*: The Lanczos-OR induced approximation to the matrix sign function performs well.

Sign function

In this example, we use the same spectrum as in the first example. However, rather than approximating the sign function directly, we instead use Lanczos-OR to approximate each term of a proxy rational function of the form (6.6). In particular, we consider the best uniform approximation[1] of degree $(39, 40)$ to the sign function on $[-10^3, 1] \cup [1, 10^3]$. Such an approximation is due to Zolotarev [Zol77], an can be derived from the more well known Zolotarev approximation to the inverse square root function on $[1, 10^6]$. Our implementation follows the partial fractions implementation in the Rational Krylov Toolbox [BEG20] and involves computing the sum of 20 terms of degree $(1, 2)$. The/ results are shown in Figure 6.3.

[1]Note that the eigenvalues of **A** live in $[-10^2, -1] \cup [1, 10^3]$, so we could have used an asymmetric approximation to the sign function. This would reduce the degree of the rational function required to obtain an approximation of given accuracy, but the qualitative behavior of Lanczos-OR-lm would not change substantially.

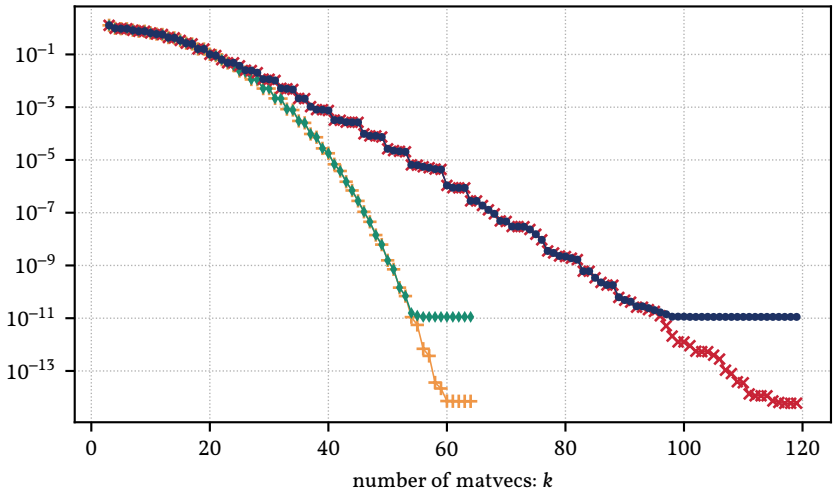


Figure 6.3: $\mathbf{A}^2$-norm error in Lanczos-OR-lm based rational approximation to matrix sign function. *Legend*: Lanczos-OR-lm based approximation of matrix sign function with (—♦—) and without (—●—), reorthogonalization. Lanczos-OR-lm based approximation of proxy rational matrix function with (—+—) and without (—×—) reorthogonalization *Takeaway*: Lanczos-OR-lm can be applied to each term of a proxy rational function approximation of the sign function.

At least while while the application error for the Lanczos-OR approximation to the proxy rational matrix function is large relative to the approximation error, then as seen in (6.3), the error in approximating the matrix sign function is similar to the error in approximating the proxy rational matrix function. However, as seen in (6.4), the final accuracy of approximating the matrix sign function is limited by the quality of the scalar approximation.

We also note that it really only makes sense to use Lanczos-OR-lm with a short-recurrence version of Lanczos, in which case the effects of a perturbed Lanczos recurrence are prevalent. In particular, as we noted in Chapter 1 and discuss in detail in Chapter 8, the algorithm encounters a delay of convergence as compared to what would happen with reorthogonalization.

6.4.3 Lanczos-FA vs Lanczos-OR vs CG

We now illustrate the effectiveness of the Lanczos-OR based approach to approximating rational matrix functions described in Section 6.3.2. Then we compare an existing low-memory approach, called multishift CG, to the analogous approaches based on Lanczos-OR-lm and Lanczos-FA-lm.

Throughout this example, we will assume that r is a rational function of the form

$$r = \sum_{i=1}^{m} \frac{A_i x^2 + B_i x + C_i}{a_i x^2 + b_i x + c_i}. \tag{6.6}$$

so that $r(\mathbf{A})\mathbf{b}$ has the form

$$r(\mathbf{A})\mathbf{b} = \sum_{i=1}^{m} (A_i \mathbf{A}^2 + B_i \mathbf{A} + C_i \mathbf{I})\mathbf{x}_i,$$

where $\mathbf{x}_i$ is obtained by solving the linear system of equations $(a_i\mathbf{A}^2 + b_i\mathbf{A} + c_i\mathbf{I})\mathbf{x}_i = \mathbf{b}$. This is relatively general since any real valued rational function $r : \mathbb{R} \to \mathbb{R}$ with numerator degree smaller than denominator degree and only simple poles can be written in this form (in fact, this would be true even if $A_i = 0$). A range of rational functions of this form appear naturally; for instance by a quadrature approximation to a Cauchy integral formula representation of f [HHT08]. Similar rational functions are seen in [Esh+02; FS09] and in the rational approximation tot the sign function given described in (6.5).

In certain cases, the shift invariance of Krylov subspace can be used to simultaneously compute all of the $\mathbf{x}_i$ using the same number of matrix-vector products as would be required to approximate a single $\mathbf{x}_i$. Specifically, suppose $(a_i\mathbf{A}^2+b_i\mathbf{A}+c_i\mathbf{I})^{-1}\mathbf{v}$ can be written as $\mathbf{B}+z_i\mathbf{I}$ for all $i = 1, \dots, m$. Then $K_k(\mathbf{B}-z_i\mathbf{I}, \mathbf{v})) = K_k(\mathbf{B}, \mathbf{v})$, so by constructing a single Krylov subspace $K_k(\mathbf{B}, \mathbf{v})$, one can compute all of the $\mathbf{x}_i$ and therefore $r(\mathbf{A})\mathbf{v}$. The resulting algorithms are typically called multishift-CG or multishift-MINRES [Esh+02; FS08a; FS08a; GS21; Ple+20]. However, such an approach only works when $a_i = 0$ or $a_i = a$ and $b_i = b$, and in the latter case, matrix-vector products with $\mathbf{B}$ require two matrix-vector products with $\mathbf{A}$ and convergence depends only on the properties of $\mathbf{A}^2$ rather than $\mathbf{A}$.

Instead, we might apply Lanczos-FA-lm to compute individual terms of $r(\mathbf{A})\mathbf{v}$. However, if $(a_i\mathbf{A}^2 + b_i\mathbf{A} + c_i\mathbf{I})$ is indefinite, then Lanczos-FA may exhibit oscillatory behavior due to eigenvalues of $(a_i\mathbf{T}^2 + b_i\mathbf{T} + c_i\mathbf{I})$ near zero. This may

result in a breakdown of Lanczos-FA-lm similar to the breakdown which may be encountered by standard implementations of CG on indefinite linear systems. Lanczos-OR-lm avoids such issues.

To highlight some of the tradeoffs between the algorithms, we construct several test problems by placing eigenvalues uniformly throughout the specified intervals. In all cases, $\mathbf{v}$ has uniform weight onto each eigencomponent. The outputs are computed using standard Lanczos, but we note that the spectrum and number of iterations are such that the behavior is quite similar to if full reorthgonalization were used. In particular, orthogonality is not lost since no Ritz value converges. The results of our experiments are shown in Figure 6.4.

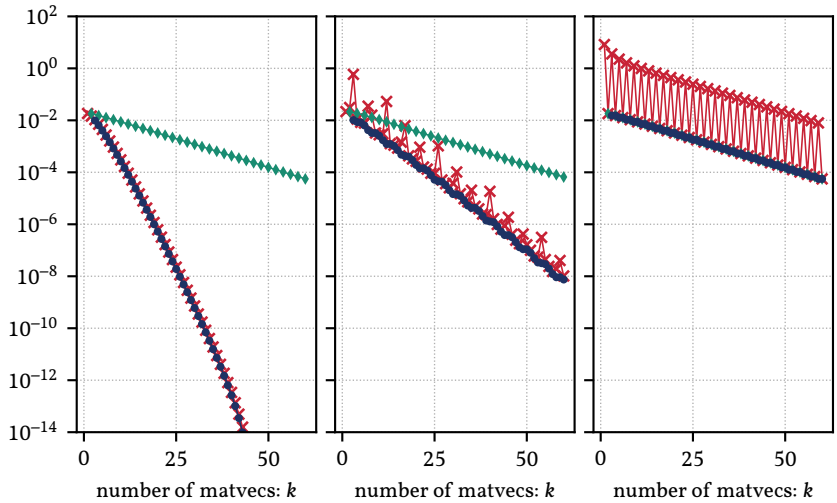


Figure 6.4: Comparison of $(\mathbf{A}^2 + c\mathbf{I})$-norm errors for CG and Lanczos-FA for computing $(\mathbf{A}^2 + c\mathbf{I})^{-1}\mathbf{v}$ with $c = 0.05$. Here CG works with $\mathbf{A}^2 + c\mathbf{I}$ and requires two matrix-vector products per iteration whereas Lanczos-FA works with $\mathbf{A}$ and requires just one. *Legend*: Lanczos-OR (—●—), Lanczos-FA (—×—), and CG on squared system (—♦—). *Left*: eigenvalues on $[1, 10]$. *Middle*: eigenvalues on $[-1.5, -1] \cup [1, 10]$. *Right*: eigenvalues on $[-10, -1] \cup [1, 10]$. *Legend*: Optimal algorithms have many nice convegence properties.

We consider approximations to $r = 1/(x^2 + 0.05)$ with eigenvalues spaced with increments of 0.005 throughout $[1, 10]$, $[-1.5, -1] \cup [1, 10]$, and $[-10, -1] \cup [1, 10]$ respectively. For each example, the condition number of $\mathbf{A}^2 + 0.05\mathbf{I}$ is roughly 100 and the eigenvalues of $\mathbf{A}^2 + 0.05\mathbf{I}$ fill out the interval $[1, 100.05]$.

As such we observe that multishift CG converges at a rate (in terms of matrix products with $\mathbf{A}$) of roughly $\exp(-k/\sqrt{\kappa(\mathbf{A}^2)}) = \exp(-k/\sqrt{100})$ on all of the examples.

In the first example, $\mathbf{A}$ is positive definite. Here Lanczos-FA and Lanczos-OR converge similarly to CG on $\mathbf{A}$ at a rate of roughly $\exp(-2k/\sqrt{10})$, where k is the number of matrix-vector products with $\mathbf{A}$.

In the next example $\mathbf{A}$ is indefinite. The convergence of CG is unchanged, because CG acts on $\mathbf{A}^2+c\mathbf{I}$, it is unable to "see" the asymmetry in the eigenvalues of $\mathbf{A}$. While the convergence of Lanczos-FA and Lanczos-OR is slowed considerably, both methods converges more quickly than CG due to the asymmetry in the intervals to the left and the right of the origin. The convergence of these methods is at a rate of roughly $\exp(-k/\sqrt{15})$, although the exact rate is more complicated to compute [Fis96; Sch11]. We also note the the emergence of oscillations in the error curve of Lanczos-FA.

In the third example, the asymmetry in the eigenvalue distribution about the origin is removed, and Lanczos-FA and Lanczos-OR converge at a rate very similar to that of multishift CG. Note that Lanczos-FA displays larger oscillations, since the symmetry of the eigenvalue distribution of $\mathbf{A}$ ensures that $\mathbf{T}$ has an eigenvalue at zero whenever k is odd. However, the size of the oscillations is regularized by the fact that $c > 0$.

Chapter 7
Spectrum dependent bounds and a posteriori error estimates

Thus far, we have not rigorously justified why Lanczos-FA and other non-optimal Lanczos-based methods typically outperform explicit polynomial methods. In Sections 7.1 to 7.3, we describe a general technique for bounding the error of Lanczos-based methods for matrix functions via a reduction to the error of Lanczos-FA used to solve a certain linear system of equations. Since the error of Lanazos-FA on linear systems is well studied, this approach can be used to derive a priori error bounds as well as a posteriori error bounds and estimates for general functions. The effectiveness of our approach is demonstrated by a range of numerical experiments. Finally, in Section 7.4, we discuss the error of Lanczos-FA on *indefinite* linear systems where Lanczos-FA is not optimal. These bounds explain why Lanczos-FA performs well in theory.

7.1 An integral representation of the Lanczos-FA error

Assuming $f : \mathbb{C} \to \mathbb{C}$ is analytic in a neighborhood of the eigenvalues of $\mathbf{A}$ and Γ is a simple closed curve or union of simple closed curves inside that neighborhood and enclosing the eigenvalues of $\mathbf{A}$, the Cauchy integral formula states that

$$f(\mathbf{A})\mathbf{v} = -\frac{1}{2\pi \boldsymbol{i}} \oint_{\Gamma} f(z)(\mathbf{A} - z\mathbf{I})^{-1}\mathbf{v}\, \mathrm{d}z. \tag{7.1}$$

If Γ also encloses the eigenvalues of $\mathbf{T}$ we can similarly write the Lanczos-FA approximation as

$$\text{lan-FA}_k(f) = -\frac{1}{2\pi \boldsymbol{i}} \oint_\Gamma f(z)\mathbf{Q}(\mathbf{T} - z\mathbf{I})^{-1}\mathbf{Q}^{\mathsf{H}}\mathbf{v}\, \mathrm{d}z. \tag{7.2}$$

Observing that the integrand of (7.1) contains the solution to the shifted linear system $(\mathbf{A} - z\mathbf{I})\mathbf{x} = \mathbf{v}$ while (7.2) contains the Lanczos-FA approximation to the solution, we make the following definition.

Definition 7.1. *For $z \in \mathbb{C}$, define the k-th Lanczos-FA error and residual for the linear system $(\mathbf{A} - z\mathbf{I})\mathbf{x} = \mathbf{v}$ as,*

$$\begin{aligned}\text{err}_k(z, \mathbf{A}, \mathbf{v}) &:= (\mathbf{A} - z\mathbf{I})^{-1}\mathbf{v} - \mathbf{Q}(\mathbf{T} - z\mathbf{I})^{-1}\mathbf{Q}^{\mathsf{H}}\mathbf{v},\\ \text{res}_k(z, \mathbf{A}, \mathbf{v}) &:= \mathbf{v} - (\mathbf{A} - z\mathbf{I})\mathbf{Q}(\mathbf{T} - z\mathbf{I})^{-1}\mathbf{Q}^{\mathsf{H}}\mathbf{v}.\end{aligned}$$

As with the Lanczos-FA approximation, we will typically omit the arguments $\mathbf{A}$ and $\mathbf{v}$, and in the case $z = 0$, we will often write err_k and res_k.

With Theorem 7.1 in place, the error of the Lanczos-FA approximation to $f(\mathbf{A})\mathbf{v}$ can be written as

$$f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f) = -\frac{1}{2\pi \boldsymbol{i}} \oint_\Gamma f(z)\, \text{err}_k(z)\, \mathrm{d}z. \tag{7.3}$$

Therefore, if for every $z \in \Gamma$ we are able to understand the convergence of Lanczos-FA on the linear system $(\mathbf{A} - z\mathbf{I})\mathbf{x} = \mathbf{v}$, then this formula lets us understand the convergence of Lanczos-FA for $f(\mathbf{A})\mathbf{v}$. To simplify bounding (7.3), we will write $\text{err}_k(z)$ for all $z \in \Gamma$ in terms of the error in solving a single shifted linear system.

To do this, we use the fact that the Lanczos factorization (1.3) can be shifted, even for complex z, to obtain

$$(\mathbf{A} - z\mathbf{I})\mathbf{Q} = \mathbf{Q}(\mathbf{T} - z\mathbf{I}) + \beta_{k-1}\mathbf{q}_k\mathbf{e}_{k-1}^{\mathsf{T}}. \tag{7.4}$$

That is, Lanczos applied to $(\mathbf{A}, \mathbf{v})$ for k steps produces output $\mathbf{Q}$ and $\mathbf{T}$ satisfying (1.3) while Lanczos applied to $(\mathbf{A} - z\mathbf{I}, \mathbf{v})$ for k steps produces output $\mathbf{Q}$ and $\mathbf{T} - z\mathbf{I}$ satisfying (7.4). Using this fact, we have the following well known lemma.

Lemma 7.2. *For all z where $\mathbf{T} - z\mathbf{I}$ is invertible,*

$$\text{res}_k(z) = \|\mathbf{v}\|_2 \left(\frac{(-1)^k}{\det(\mathbf{T} - z\mathbf{I})} \prod_{j=0}^{k-1} \beta_j \right) \mathbf{q}_k.$$

Proof. From (7.4), and the fact that $\mathbf{Q}$'s first column is $\mathbf{v}/\|\mathbf{v}\|_2$, it is clear that,

$$\begin{aligned}(\mathbf{A}-z\mathbf{I})\mathbf{Q}(\mathbf{T}-z\mathbf{I})^{-1}\mathbf{Q}^{\mathsf{H}}\mathbf{v} &= (\mathbf{A}-z\mathbf{I})\mathbf{Q}(\mathbf{T}-z\mathbf{I})^{-1}\|\mathbf{v}\|_2\mathbf{e}_0\\ &= \mathbf{Q}\|\mathbf{v}\|_2\mathbf{e}_0+\beta_k\mathbf{q}_k\mathbf{e}_{k-1}^{\mathsf{H}}(\mathbf{T}-z\mathbf{I})^{-1}\|\mathbf{v}\|_2\mathbf{e}_0\\ &= \mathbf{v}+\beta_k\mathbf{q}_k\mathbf{e}_{k-1}^{\mathsf{H}}(\mathbf{T}-z\mathbf{I})^{-1}\|\mathbf{v}\|_2\mathbf{e}_0.\end{aligned}$$

Using the formula $(\mathbf{T}-z\mathbf{I})^{-1} = (1/\det(\mathbf{T}-z\mathbf{I}))\operatorname{adj}(\mathbf{T}-z\mathbf{I})$, we see that

$$\mathbf{e}_{k-1}^{\mathsf{H}}(\mathbf{T}-z\mathbf{I})^{-1}\mathbf{e}_0 = \frac{(-1)^{k-1}}{\det(\mathbf{T}-z\mathbf{I})}\prod_{j=0}^{k-2}\beta_j.$$

The result then follows by combining these expressions. □

We use Lemma 7.2 to relate $\mathsf{err}_k(z)$ to $\mathsf{err}_k(w)$ for any $z, w \in \mathbb{C}$.

Definition 7.3. *For $w, z \in \mathbb{C}$ define $h_{w,z} : \mathbb{R} \to \mathbb{C}$ and $h_z : \mathbb{R} \to \mathbb{C}$ by*

$$h_{w,z}(x) := \frac{x-w}{x-z}, \qquad h_z(x) := \frac{1}{x-z}.$$

Corollary 7.4. *For all $z, w \in \mathbb{C}$, where $\mathbf{A}-z\mathbf{I}$ and $\mathbf{A}-w\mathbf{I}$ are both invertible,*

$$\begin{aligned}\mathsf{err}_k(z) &= \det(h_{w,z}(\mathbf{T}))\,h_{w,z}(\mathbf{A})\,\mathsf{err}_k(w)\\ \mathsf{res}_k(z) &= \det(h_{w,z}(\mathbf{T}))\,\mathsf{res}_k(w).\end{aligned}$$

Proof. By Lemma 7.2,

$$\det(\mathbf{T}-z\mathbf{I})\,\mathsf{res}_k(z) = \det(\mathbf{T}-w\mathbf{I})\,\mathsf{res}_k(w).$$

Thus,

$$\mathsf{res}_k(z) = \frac{\det(\mathbf{T}-w\mathbf{I})}{\det(\mathbf{T}-z\mathbf{I})}\,\mathsf{res}_k(w) = \det(h_{w,z}(\mathbf{T}))\,\mathsf{res}_k(w).$$

Noting that $\mathsf{res}_k(z) = (\mathbf{A}-z\mathbf{I})\,\mathsf{err}_k(z)$ and $\mathsf{res}_k(w) = (\mathbf{A}-w\mathbf{I})\,\mathsf{err}_k(w)$, we obtain the relation between the errors,

$$\begin{aligned}\mathsf{err}_k(z) &= \det(h_{w,z}(\mathbf{T}))(\mathbf{A}-z\mathbf{I})^{-1}(\mathbf{A}-w\mathbf{I})\,\mathsf{err}_k(w)\\ &= \det(h_{w,z}(\mathbf{T}))\,h_{w,z}(\mathbf{A})\,\mathsf{err}_k(w).\end{aligned}$$

□

In summary, combining (7.3) and Corollary 7.4 we have the following corollary. This result is by no means new, and appears throughout the literature; see for instance [FS09] and [FGS14b, Theorem 3.4].

Corollary 7.5. *Suppose* $\mathbf{A}$ *is a Hermitian matrix and* $f : \mathbb{C} \to \mathbb{C}$ *is a function analytic in a neighborhood of the eigenvalues of* $\mathbf{A}$ *and* $\mathbf{T}$*, where* $\mathbf{T}$ *is the tridiagonal matrix output by Lanczos run on* $\mathbf{A}, \mathbf{v}$ *for* k *steps. Then, if* Γ *is a simple closed curve or union of simple closed curves inside this neighborhood and enclosing the eigenvalues of* $\mathbf{A}$ *and* $\mathbf{T}$ *and* $w \in \mathbb{C}$ *is such that* $w \notin \Lambda(\mathbf{T}) \cup \Lambda$*,*

$$f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f) = \left(-\frac{1}{2\pi \boldsymbol{i}} \oint_\Gamma f(z) \det(h_{w,z}(\mathbf{T}))\, h_{w,z}(\mathbf{A})\, \mathrm{d}z\right) \mathsf{err}_k(w).$$

7.1.1 A reduction to linear system error

Our main result is a flexible bound for the Lanczos-FA error, obtained by bounding the integral in the right-hand side of Corollary 7.5. As we will see in Section 7.2, we can instantiate this theorem to obtain effective a priori and a posteriori error bounds in many settings.

Theorem 7.6. *In the setting of Corollary 7.5, if for some* $S, S_0, \ldots, S_{k-1} \subset \mathbb{R}$ *we have* $\Lambda \subset S_0$ *and* $\lambda_i(\mathbf{T}) \in S_i$ *for* $i = 0, \ldots, k-1$*, then*

$$\|f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f)\| \le \underbrace{\left(\frac{1}{2\pi} \oint_\Gamma |f(z)| \left(\prod_{i=0}^{k-1} \|h_{w,z}\|_{S_i}\right) \|h_{w,z}\|_S |\mathrm{d}z|\right)}_{\textit{integral term}} \underbrace{\|\mathsf{err}_k(w)\|}_{\textit{linear system error}}.$$

Analogously, we have a bound for Gaussian quadrature

Theorem 7.7. *In the setting of Corollary 7.5, if for some* $S, S_0, \ldots, S_{k-1} \subset \mathbb{R}$ *we have* $\Lambda \subset S_0$ *and* $\lambda_i(\mathbf{T}) \in S_i$ *for* $i = 0, \ldots, k-1$*, then*

$$|\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v} - \int f \,\mathrm{d}[\Psi]_{2k-1}^{\mathrm{gq}}| \le \underbrace{\left(\frac{1}{2\pi} \oint_\Gamma |f(z)| \left(\prod_{i=0}^{k-1} \|h_{w,z}\|_{S_i}^2\right) \|h_z\|_S |\mathrm{d}z|\right)}_{\textit{integral term}} \underbrace{\|\mathsf{res}_k(w)\|_2^2}_{\textit{linear system error}}.$$

The above bounds depend on our choices of Γ, w, and the sets $S, S_0, \ldots, S_{k-1}$, which must contain the eigenvalues of $\mathbf{A}$ and $\mathbf{T}_k$. The sets $S, S_0, \ldots, S_{k-1}$ should be chosen based on the information we have about $\mathbf{A}$ and $\mathbf{T}_k$. For example, we could take all these sets to be the eigenvalue range $I(\mathbf{A})$. If we have more information a priori about the eigenvalues of $\mathbf{A}$, we can obtain a tighter bound by choosing smaller S, with correspondingly lower $\|h_{w,z}\|_S$. For an a posteriori bound, we can simply

set $S_i = \{\lambda_i(\mathbf{T}_k)\}$, for $i = 0, \ldots, k-1$. This gives an optimal value for $\|h_{w,z}\|_S$. Both approaches are detailed in Section 7.2.

We emphasize that, in both bounds, the integral term and linear system error term in the theorem are entirely decoupled. Thus, once the integral term is computed, bounding the error of Lanczos-FA for $f(\mathbf{A})\mathbf{v}$ is reduced to bounding $\|\mathrm{err}_k(w)\|$, and if the integral term can be bounded independently of k, Theorem 7.6 implies that, up to a constant factor, the Lanczos-FA approximation to $f(\mathbf{A})\mathbf{v}$ converges at least as fast as $\|\mathrm{err}_k(w)\|$.

Note that Theorem 7.6 depends on $\|\mathrm{err}_k(w)\|$ whereas Theorem 7.7 depends on $\|\mathrm{res}_k(w)\|_2^2$. Thus, heuristically, we can expect the quadratic form to converge at a rate twice that of the norm of the error of the matrix function. This is exacted as Gaussian quadrature is exact for polynomials of degree $2k-1$ whereas Lanczos-FA is exact for polynomials of degree $k-1$. In the case that the contour Γ does not pass through $\mathcal{I}$, the bound of Theorem 7.7 is essentially as easy to compute as that of Theorem 7.6. However, if the contour passes through $\mathcal{I}$ at w, to ensure that S does not contain points in the contour, it must be chosen as a set other than $\mathcal{I}$. This set must contain all of $\mathbf{A}$'s eigenvalues and we must bound its distance to the contour (in particular, to w).

Proof of Theorem 7.6. We begin by taking the norm on both sides of Corollary 7.5. Applying the triangle inequality for integrals and using the fact that $\|\cdot\|$ is induced by a matrix with the same eigenvectors as $\mathbf{A}$ (see Lemma 10.1) we have

$$\|f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f)\| \leq \left(\frac{1}{2\pi}\oint_\Gamma |f(z)||\det(h_{w,z}(\mathbf{T}))|\|h_{w,z}(\mathbf{A})\|_2|dz|\right)\|\mathrm{err}_k(w)\|. \quad (7.5)$$

Next, since $\Lambda \subseteq S$ then

$$\|h_{w,z}(\mathbf{A})\|_2 = \max_{i=0,\ldots,n-1} |h_{w,z}(\lambda_i(\mathbf{A}))| \leq \|h_{w,z}\|_S,$$

and similarly, if $\lambda_i(\mathbf{T}) \in S_i$ for $i = 0, \ldots, k-1$, then

$$|\det(h_{w,z}(\mathbf{T}))| = \left|\prod_{i=0}^{k-1} h_{w,z}(\lambda_i(\mathbf{T}))\right| \leq \prod_{i=0}^{k-1} \|h_{w,z}\|_{S_i}. \quad (7.6)$$

Combining these inequalities yields the result. □

Proof of Theorem 7.7. Recall

$$\mathbf{v}^{\mathsf{H}}\mathsf{lan\text{-}FA}_k(f) = \mathbf{v}^{\mathsf{H}}\mathbf{Q}f(\mathbf{T})\mathbf{Q}^{\mathsf{H}}\mathbf{v} = \|\mathbf{v}\|_2^2\, \imath : \mathbf{e}_0^{\mathsf{H}} f(\mathbf{T})\mathbf{e}_0 = \int f \,\mathrm{d}[f]_{2k-1}^{\mathrm{gq}}.$$

Since $\mathbf{A}$ is Hermitian, $(\mathbf{A} - z\mathbf{I})^{\mathsf{H}} = \mathbf{A} - \overline{z}\mathbf{I}$. Thus, since

$$\mathbf{v}^{\mathsf{H}}(\mathbf{A} - z\mathbf{I})^{-1} = ((\mathbf{A} - \overline{z}\mathbf{I})^{-1}\mathbf{v})^{\mathsf{H}} = (\mathsf{lan\text{-}FA}_k(h_{\overline{z}}) + \mathsf{err}_k(\overline{z}))\mathbf{v})^{\mathsf{H}}$$

we can expand the quadratic form error as

$$\mathbf{v}^{\mathsf{H}}\mathsf{err}_k(z) = \mathbf{v}^{\mathsf{H}}(\mathbf{A} - z\mathbf{I})^{-1}\mathsf{res}_k(z) = (\mathsf{lan\text{-}FA}_k(h_{\overline{z}})) + \mathsf{err}_k(\overline{z}))^{\mathsf{H}}\,\mathsf{res}_k(z).$$

Now, by definition, $\mathsf{lan\text{-}FA}_k(h_{\overline{z}}(x)) = \mathbf{Q}h_{\overline{z}}(\mathbf{T})\mathbf{Q}^{\mathsf{H}}\mathbf{v}$ and by Lemma 7.2 $\mathsf{res}_k(z)$ is proportional to $\mathbf{q}_{k+1}$. Thus, since, at least in exact arithmetic, $\mathbf{q}_{k+1}$ is orthogonal to $\mathbf{Q}$,

$$\mathbf{v}^{\mathsf{H}}\mathsf{err}_k(z) = \mathsf{err}_k(\overline{z})^{\mathsf{H}}\mathsf{res}_k(z) = ((\mathbf{A} - \overline{z}\mathbf{I})^{-1}\mathsf{res}_k(\overline{z}))^{\mathsf{H}}\mathsf{res}_k(z).$$

Next, using Corollary 7.4 and the fact that $h_{w,z}(x)h_{w,\overline{z}}(x) = |h_{w,z}(x)|^2$ for $w, x \in \mathbb{R}$,

$$\mathbf{v}^{\mathsf{H}}\mathsf{err}_k(z) = |\det(h_{w,z}(\mathbf{T}))|^2\mathsf{res}_k(w)^{\mathsf{H}}(\mathbf{A} - z\mathbf{I})^{-1}\mathsf{res}_k(w).$$

We then have,

$$|\mathbf{v}^{\mathsf{H}}\mathsf{err}_k(z)| \leq |\det(h_{w,z}(\mathbf{T}))|^2\|(\mathbf{A} - z\mathbf{I})^{-1}\|_2\|\mathsf{res}_k(w)\|_2^2.$$

Applying the Cauchy integral formula we therefore obtain a bound for the quadratic form error analogous to Theorem 7.6 we obtain the result. □

7.1.2 Comparison with previous work

Our framework for analyzing Lanczos-FA has four properties which differentiate it from past work: (i) it is applicable to a wide range of functions, (ii) it yields a priori bounds dependent on fine-grained properties of the spectrum of $\mathbf{A}$ such as clustered or isolated eigenvalues, (iii) it can be used a posteriori as a practical stopping criterion, and (iv) it is applicable when computations are carried out in finite precision arithmetic. To the best of our knowledge, no existing analysis satisfies more than two of these properties simultaneously. In this section, we provide a brief overview of the most relevant past work.

Most directly related to our framework is a series of works which also make use of the shift-invariance of Krylov subspaces when f is a Stieltjes function[1] [FGS14a; FS15; ITS09] or a certain type of rational function [Fro+13; FS08b; FS09]. These analyses are applicable a priori and a posteriori and in fact allow for corresponding error *lower bounds* as well. However, these bounds cannot be applied to more general functions, and the impact of a perturbed Lanczos recurrence in finite precision is not considered.

The most detailed generally applicable analysis is [MMS18], which extends [DK91; DK95] and studies Theorem 6.4, the classical bound for Lanczos-FA based on polynomial approximation on $\mathcal{I}$, when Lanczos is run in finite precision arithmetic. However, as we have seen throughout this thesis, Theorem 6.4 is often too pessimistic in practice as it does not depend on the fine-grained properties about the distribution of eigenvalues. Another generally applicable analysis is [HLS98], which suggests replacing $\mathsf{err}_k(z)$ with $\mathsf{res}_k(z)$ in (7.3). Since $\mathsf{res}_k(z)$ can be computed once the outputs of Lanczos have been obtained, the resulting integral can be computed (or at least approximated by a quadrature rule). However, this approach does not take into account the actual relationship between $\mathsf{res}_k(z)$ and $\mathsf{err}_k(z)$, and therefore gives only an estimate of the error, not a true bound. Another Cauchy integral formula based approach is [HL97] which shows that Lanczos-FA exhibits superlinear convergence for the matrix exponential and certain other specific analytic functions.

There are a variety of other bounds specialized to individual functions. For example, it is known that if $\mathbf{A}$ is nonnegative definite and $t > 0$, then the error in the Lanczos-FA approximation for the matrix exponential $\exp(t\mathbf{A})\mathbf{v}$ can be related to the maximum over $s \in [0, t]$ of the error in the optimal approximation to $\exp(s\mathbf{A})\mathbf{v}$ over a Krylov space of slightly lower dimension [DGK98]. More recent work involving the matrix exponential are [JL14; JAK19; Jaw21]. There is also a range of work which analyzes the convergence of Lanczos-FA and related methods for computing the square root and sign functions [Bor99; Bor03; Esh+02].

[1] A function f defined on the positive real axis is a Stieltjes function if and only if $f(x) \geq 0$ for all $x \in \mathbb{R}$ and f has an analytic extension to the cut plane $\mathbb{C} \setminus (-\infty, 0]$ satisfying $\mathrm{Im}(f(x)) \leq 0$ for all x in the upper half plane [Ber07, Theorem 3.2] [AK65, p. 127 attributed to Krein].

7.2 Applying our framework

We proceed to show how to effectively bound the integral term of Theorem 7.6, to give a priori and a posteriori bounds on the Lanczos-FA error, assuming accurate bounds on $\|\mathsf{err}_k(w)\|$ are available. Throughout this chapter, we assume $w \in \mathbb{R}$ and we do not discuss in detail how to bound this linear system error – there are many known approaches, both a priori and a posteriori, and the best bounds to use are often context dependent. Some of these approaches are similar to those used for Lanczos-OR in Section 5.4.

To use Theorem 7.6, we must evaluate or bound $\|h_{w,z}\|_{S_i}$. Towards this end, we introduce the following lemmas, which apply when S_i is an interval. These lemmas are also useful when S_i is a union of intervals – in that case $\|h_{w,z}\|_{S_i}$ is bounded by the maximum bound on any of these intervals. i

Lemma 7.8. *For any interval $[a,b] \subset \mathbb{R}$, if $z \in \mathbb{C} \setminus [a,b]$ and $w \in \mathbb{R}$, we have*

$$\|h_{w,z}\|_{[a,b]} = \max\left\{\left|\frac{a-w}{a-z}\right|, \left|\frac{b-w}{b-z}\right|, \left(\left|\frac{z-w}{\mathrm{Im}(z)}\right| \text{ if } x^* \in [a,b] \text{ else } 0\right)\right\}$$

where

$$x^* := \frac{\mathrm{Re}(z)^2 + \mathrm{Im}(z)^2 - \mathrm{Re}(z)w}{\mathrm{Re}(z) - w}.$$

Proof. Note that for $x \in \mathbb{R}$,

$$|h_{w,z}(x)|^2 = \left|\frac{x-w}{x-z}\right|^2 = \frac{(x-w)^2}{(x-\mathrm{Re}(z))^2 + \mathrm{Im}(z)^2},$$

and

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(|h_{w,z}(x)|^2\right) = \frac{[(x-\mathrm{Re}(z))^2 + \mathrm{Im}(z)^2]2(x-w) - (x-w)^2 2(x-\mathrm{Re}(z))}{[(x-\mathrm{Re}(z))^2 + \mathrm{Im}(z)^2]^2}.$$

Aside from $x = w$, where $h_{w,z}(x) = 0$, the only value $x \in \mathbb{R}$ for which $\frac{\mathrm{d}}{\mathrm{d}x}\left(|h_{w,z}(x)|^2\right) = 0$ is x^*. This implies that the only possible local extrema of $|h_{w,z}(x)|$ on $[a,b]$ are a, b, and x^* if $x^* \in [a,b]$. Substituting the expression for x^* into that for $|h_{w,z}(x^*)|$, one finds, after some algebra, that $|h_{w,z}(x^*)| = |z-w|/|\mathrm{Im}(z)|$. □

Lemma 7.9. *Fix $r > 0$, let $\mathcal{D}(c,t)$ be the disc in the complex plane centered at c with radius $t \geq 0$, and define*

$$X_r = \bigcup_{x\in[a,b]} \mathcal{D}\left(x, \frac{|x-w|}{r}\right).$$

Then for $z \in \mathbb{C} \setminus X_r$, we have

$$\|h_{w,z}\|_{[a,b]} \leq r.$$

In particular, if z is on the boundary of X_r, then $\|h_{w,z}\|_{[a,b]} = r$.

Proof. Let $z \in \mathbb{C} \setminus X_r$ and pick any $x \in [a,b]$. Since $z \notin \mathcal{D}(x, |x-w|/r)$ it follows that $|z-x| > |x-w|/r$ and therefore $|h_{w,z}(x)| = |x-w|/|x-z| < r$. Maximizing over x yields the result.

If z is on the boundary of X_r, then for some $x \in [a,b]$, $|z-x| = |x-w|/r$, which means that for this x, $|h_{w,z}(x)| = r$. □

Note that if $r \leq 1$ and $w \in \mathbb{R} \setminus [a,b]$, then the region described in Lemma 7.9 is simply a disc about b if $w < a$ or a disc about a if $w > b$. If $r > 1$ and w is real, then the region described is that in the discs about a and b and between the two external tangents to these two discs.

Similar to Lemma 7.8 we have the following bound on $\|h_z\|_{S_i}$ when S_0 is an interval. This allows a bound on Theorem 7.7 analogous to (7.5).

Lemma 7.10. *For any interval $[a,b] \subset \mathbb{R}$, if $z \in \mathbb{C} \setminus [a,b]$, we have*

$$\|h_z\|_{[a,b]} = \begin{cases} 1/|\operatorname{Im}(z)| & \operatorname{Re}(z) \in \mathcal{I} \\ 1/|a-z| & \operatorname{Re}(z) < a \\ 1/|b-z| & \operatorname{Re}(z) > b \end{cases}$$

7.2.1 A priori bounds

We can use Theorem 7.6 to give a priori bounds, as long as we choose S and S_i, $i = 0, \dots, k-1$ independently of $\mathbf{b}$ (and in turn $\mathbf{T}$).

The simplest possibility is to take $S = S_i = \mathcal{I}$. In this case, as an immediate consequence of Theorem 7.6 and Lemma 7.9 we have the following a priori bound,

Corollary 7.11. *Suppose that for some $w < \lambda_{\min}$, f is analytic in a neighborhood of $\mathcal{D}(\lambda_{\max}, \lambda_{\max} - w)$. Then, taking Γ to be the boundary of this disk,*

$$\begin{aligned}\|f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f)\| &\leq \left(\frac{1}{2\pi}\oint_\Gamma |f(z)||\mathrm{d}z|\right)\|\text{err}_k(w)\| \\ &\leq \left((\lambda_{\max} - w)\max_{z\in\Gamma}|f(z)|\right)\|\text{err}_k(w)\|.\end{aligned}$$

Proof. To obtain the first inequality observe that Lemma 7.9 with $[a,b] = I$ implies $\|h_{w,z}\|_I = 1$ on this contour. The second inequality follows since the length of Γ is $2\pi(\lambda_{\max} - w)$. □

This bound is closely related to [FGS14a, Theorem 6.6] which bounds the error in Lanczos-FA for Stieltjes functions in terms of the error in the Lanczos approximation for a certain linear system.

Corollary 7.11 provides simple reductions to the error of solving a positive definite linear system involving $\mathbf{A} - w\mathbf{I}$ using Lanczos. However, these bounds may be a significant overestimate in practice. In particular, for any $k > 1$, (7.6) cannot be sharp due to the fact that $\|h_{w,z}\|_I = \sup_{x\in I}|h_{w,z}(x)|$ cannot be attained at every eigenvalue of $\mathbf{T}$. In fact, for most values $\lambda_i(\mathbf{T})$ and most points $z \in \Gamma$, we expect $|h_{w,z}(\lambda_i(\mathbf{T}))| \ll \|h_{w,z}\|_I$. Figure 7.1 shows sample level curves for $\|h_{w,z}\|_I/|\det(h_{w,z}(\mathbf{T}))|^{1/k}$ which illustrate the slackness in the bound.

To derive sharper a priori bounds, there are several approaches.

First, if more information is known about the eigenvalue distribution of $\mathbf{A}$, then the S_i can be chosen based on this information. For example, it is possible to exploit the interlacing property of the eigenvalues of $\mathbf{T}$.

Example 7.12. Suppose $\mathbf{A}$ has eigenvalues in $[0,1]$ with a single eigenvalue at $\kappa > 1$. Assume $w \leq 0$. Then there is at most one eigenvalue of $\mathbf{T}$ in $[1,\kappa]$ so in Theorem 7.6 we can pick $S_i = [0,1]$ for $i = 0,\ldots,k-2$ and $S_{k-1} = [0,\kappa]$. We have

$$|\det(h_{w,z}(\mathbf{T}))| = \left|\prod_{i=0}^{k-1} h_{w,z}(\lambda_i(\mathbf{T}))\right| \leq \left(\|h_{w,z}\|_{[0,1]}\right)^{k-1}\|h_{w,z}\|_{[0,\kappa]}.$$

If z is near to κ then $\|h_{w,z}\|_{[0,1]}$ may be much smaller than $\|h_{w,z}\|_{[0,\kappa]}$.

Second, the contour Γ can be chosen to try to reduce the slackness in (7.6). Intuitively, the slackness is exacerbated when $z \in \Gamma$ is close to S_i but far from

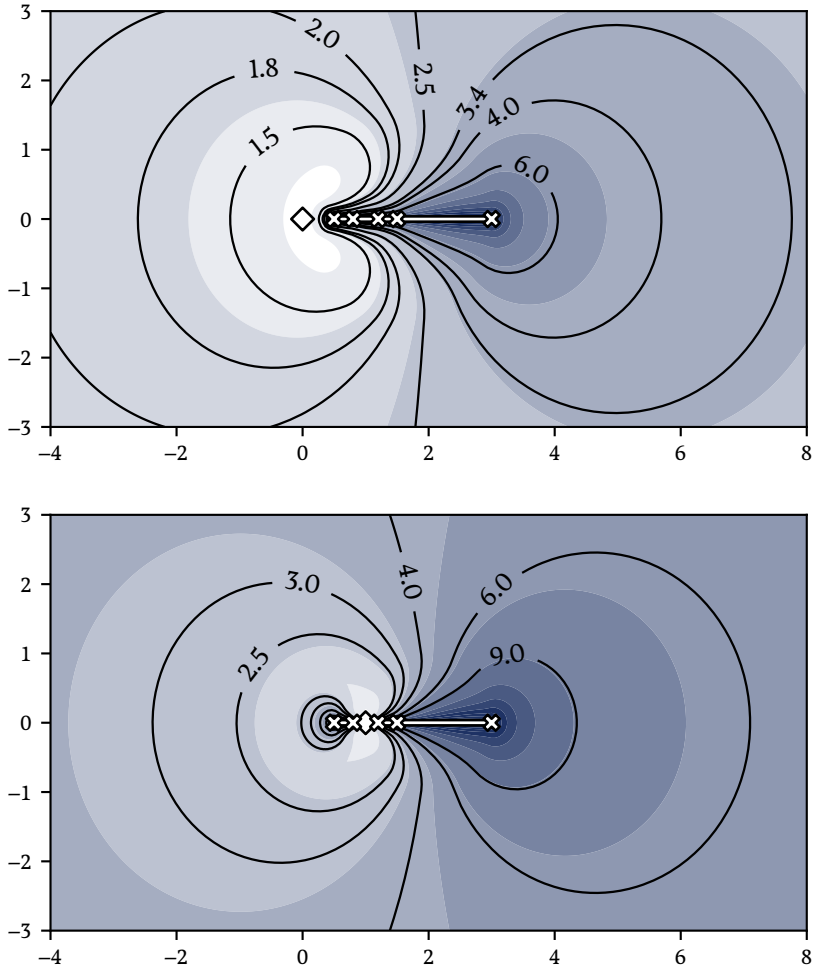


Figure 7.1: Contour plot of $\|h_{w,z}\|_I/|\det(h_{w,z}(\mathbf{T}))|^{1/k}$ as a function of $z \in \mathbb{C}$ for a synthetic example with $w = 0$ (top) and $w = 1$ (bottom), $I = [0.5, 3]$, and $\Lambda(\mathbf{T}) = \{0.5, 0.8, 1.2, 1.5, 3\}$ $(k = 5)$. Larger slackness in (7.6) corresponds to darker regions. *Legend*: Here w is indicated by the white diamond (◇). and the eigenvalues of $\mathbf{T}$ are indicated by white x's (⊠). *Takeaway*: Slackness exchibits structure; in particular, it is lower far from Λ.

$\lambda_i(\mathbf{T})$. For instance, for any $k > 1$,

$$\lim_{|z|\to\infty} \frac{\|h_{w,z}\|_{\mathcal{I}}^k}{|\det(h_{w,z}(\mathbf{T}))|} \to 1, \qquad \text{and} \qquad \forall \lambda \in \mathcal{I},\ \lim_{z\to\lambda} \frac{\|h_{w,z}\|_{\mathcal{I}}^k}{|\det(h_{w,z}(\mathbf{T}))|} \to \infty.$$

This behavior is also observed in Figure 7.1.

These observations suggest that we should pick Γ to be far from the spectrum of $\mathbf{A}$. Of course, we are constrained by properties of f such as branch cuts and singularities. Moreover, certain contours may increase the slackness in Theorem 7.6 itself. These considerations are discussed further in Section 7.3.1.

7.2.2 A posteriori error bounds

After the Lanczos factorization (1.3) has been computed, $\mathbf{T}$ is known and $\Lambda(\mathbf{T})$ can be cheaply computed. Thus, in Theorem 7.6 we can take $S_i = \{\lambda_i(\mathbf{T})\}$ for $i = 0, \dots, k-1$, which is the best possible choice. In this case (7.6) is an equality and $\det(h_{w,z}(\mathbf{T})) = \det(\mathbf{T}-w)/\det(\mathbf{T}-z)$ can be computed via tridiagonal determinant formulas rather than using the eigenvalues of $\mathbf{T}$.

If $\mathcal{I}$ is not known, the extreme Ritz values $\lambda_{\min}(\mathbf{T})$ and $\lambda_{\max}(\mathbf{T})$ can be used to estimate the extreme eigenvalues of $\mathbf{A}$ [KW92; PSS82]. All together, this means that it is not difficult to efficiently obtain accurate estimates of the bound from Theorem 7.6.

7.2.3 Numerical computation of integrals

Typically, to produce an a priori or a posteriori error bound, the integral term in Theorem 7.6 must be computed numerically. Consider a discretization of the integral

$$f(\mathbf{A}) = -\frac{1}{2\pi \boldsymbol{i}} \oint_\Gamma f(z)(\mathbf{A} - z\mathbf{I})^{-1} \mathrm{d}z$$

using nodes z_i and weights w_i, $i = 0, 1, \dots, q-1$. This yields a rational matrix function

$$r_q(\mathbf{A}) := -\frac{1}{2\pi \boldsymbol{i}} \sum_{i=0}^{q-1} w_i f(z_i)(\mathbf{A} - z_i\mathbf{I})^{-1}.$$

Using the triangle inequality, we can write

$$\|f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f)\|$$

$$\leq \|f(\mathbf{A})\mathbf{v} - r_q(\mathbf{A})\mathbf{v}\| + \|r_q(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(r_q))\| + \|\text{lan-FA}_k(r_q) - \text{lan-FA}_k(f)\|$$
$$\leq 2\left(\max_{x\in\Lambda\cup\Lambda(\mathbf{T})} |f(x) - r_q(x)|\right)\|\mathbf{v}\| + \|r_q(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(r_q)\|. \tag{7.7}$$

Now, observe that analogous to Theorem 7.6,

$$\|r_q(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(r_q)\|$$
$$\leq \left(\frac{1}{2\pi}\sum_{i=0}^{q-1} w_i\,|f(z_i)|\left(\prod_{i=0}^{k-1}\|h_{w,z}\|_{S_i}\right)\|h_{w,z}\|_{S_0}\right)\|\text{err}_k(w)\|. \tag{7.8}$$

If we use the same nodes and weights to evaluate the integral term in Theorem 7.6, we obtain exactly the expression on the right hand side of (7.8). Thus, this discretization of Theorem 7.6 is a true upper bound for the Lanczos-FA error to within an additive error of size equal to twice the approximation error of $r(x)$ to $f(x)$ on $\Lambda \cup \Lambda(\mathbf{T})$ times $\|\mathbf{v}\|$. In many cases, we expect exponential convergence of r_q to f, which implies that this term can be made less than any desired value $\epsilon > 0$ using a number of quadrature nodes that grows only as the logarithm of ϵ^{-1} [HHT08; TW14].

We note that fast convergence of r_q to f suggests that, instead of applying Lanczos-FA, we can approximate $f(\mathbf{A})\mathbf{v}$ by first finding r_q and then solving a small number of linear systems $(\mathbf{A} - z_i\mathbf{I})\mathbf{x}_i = \mathbf{v}$ to compute $r_q(\mathbf{A})\mathbf{v}$. Solving these systems with any fast linear system solver yields an algorithm for approximating $f(\mathbf{A})\mathbf{v}$ inheriting, up to logarithmic factors in the error tolerance, the same convergence guarantees as the linear system solvers used. A recent example of this approach is found in [JS19] which uses a modified version of stochastic variance reduced gradient (SVRG) to obtain a nearly input sparsity time algorithm for $f(\mathbf{A})\mathbf{v}$ when f corresponds to principal component projection or regression.

A range of work suggests using a Krylov subspace method and the shift invariance of the Krylov subspace to solve these systems and compute $r_q(\mathbf{A})\mathbf{b}$ explicitly. This was studied in [Fro+13; FS09] for the Lanczos method, and in [Ple+20] for MINRES, the latter of which uses the results of [HHT08] to determine the quadrature nodes and weights. However, as the above argument demonstrates, the limit of the Lanczos-based approximation as the discretization becomes finer is simply the Lanczos-FA approximation to $f(\mathbf{A})\mathbf{v}$. Therefore, there is no clear advantage to such an approach over Lanczos-FA in terms of the convergence properties, unless preconditioning is used. .

On the other hand, there are some advantages to these approaches in terms of computation. Indeed, Krylov solvers for symmetric/Hermitian linear systems require just $O(n)$ storage; i.e. they do not require more storage as more iterations are taken. A naive implementation of Lanczos-FA requires $O(kn)$ storage, and while Lanczos-FA can be implemented to use $O(n)$ storage by taking two passes, this has the effect of doubling the number of matrix-vector products required. See [GS21] for a recent overview of limited-memory Krylov subspace methods.

7.3 Examples and numerical verification

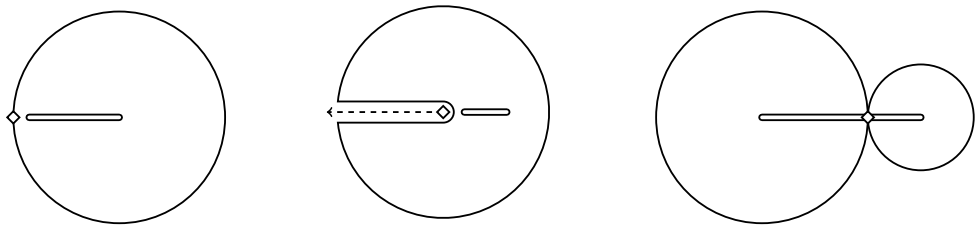
We next present examples in which we apply Theorem 7.6 to give a posteriori and a priori error bounds for approximating common matrix functions with Lanczos-FA. These examples illustrate the general approaches to applying Theorem 7.6 described in Section 7.2. All integrals are computed either analytically or using SciPy's `integrate.quad` which is a wrapper for QUADPACK routines.

In all cases, we exactly compute the $\|\mathrm{err}_k(w)\|$ term in the bounds. In practice, one would bound this quantity a priori or a posteriori using existing results on bounding the Lanczos error for linear system solves. By computing the error exactly, we separate any looseness due to our bounds from any looseness due to an applied bound on $\|\mathrm{err}_k(w)\|$.

7.3.1 Choice of contour

Let $\mathbf{A}$ be positive definite and $f(x) = \sqrt{x}$. Perhaps the simplest bound is obtained by using Theorem 7.6 with $w = 0$, $S_i = I$ and Γ chosen as the boundary of the disk $\mathcal{D}(\lambda_{\max}, \lambda_{\max})$.We then obtain a bound via Corollary 7.11. However, this bound may be loose – note that except through $\|\mathrm{err}_k(w)\|$, it does not depend on the number of iterations k. Thus it cannot establish convergence at a rate faster than that of solving a linear system with coefficient matrix $\mathbf{A}$.

Keeping $w = 0$, we can obtain tighter bounds by letting Γ be a "Pac-Man" like contour that consists of a large circle about the origin of radius R with a small circular cutout of radius r that excludes the origin and a small strip cutout to exclude the negative real axis. That is, as shown in Figure 7.2b, the boundary of



(a) circle contour **(b)** Pac-Man contour **(c)** double circle contour

Figure 7.2: Circle, Pac-Man and double circle contours described in Sections 7.3.1 and 7.3.2 respectively. All three figures show $\mathcal{I}$ (⟺) and w (◇).

the set,

$$\mathcal{D}(0,R) \setminus (\{z : \mathrm{Re}(z) \le 0, |\,\mathrm{Im}(z)| < r\} \cup \mathcal{D}(0,r)).$$

As the outer radius $R \to \infty$, the integral over the large circular arc goes to 0 since $\|h_{w,z}\|_{\mathcal{I}} = O(R^{-1})$, $|f(z)| = O(R^{1/2})$, and the length of the circular arc is on the order of R. Thus, the product $f(z)(\|h_{w,z}\|_{\mathcal{I}})^{k+1}$ goes to 0 as $R \to \infty$, for all $k \ge 1$. Similarly, as $r \to 0$, the length of the small arc goes to zero. Therefore, we need only consider the contributions to the integral on $[-R \pm ir, \pm ir]$ in the limit $R \to \infty, r \to 0$.

In this case, when $S_i = \mathcal{I}$ for all i, we can compute the value of the integral term in Theorem 7.6 analytically. We have

$$\begin{aligned}
\|f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f)\| &\le \left(\frac{1}{2\pi}\int_{-\infty}^{0} |(x \pm 0i)^{1/2}|\, \|h_{w,x\pm 0i}\|_{\mathcal{I}}^{k+1}\, \mathrm{d}x\right) \|\mathrm{err}_k\| \\
&= \left(\frac{1}{2\pi}\int_{-\infty}^{0} |x \pm 0i|^{1/2} \frac{\lambda_{\max}(\mathbf{A})^{k+1}}{(\lambda_{\max}(\mathbf{A}) - x)^{k+1}}\, \mathrm{d}x\right) \|\mathrm{err}_k\| \\
&= \left(\frac{1}{\pi}\lambda_{\max}(\mathbf{A})^{k+1}\int_{0}^{\infty} \frac{y^{1/2}}{(\lambda_{\max}(\mathbf{A}) + y)^{k+1}}\mathrm{d}y\right) \|\mathrm{err}_k\| \\
&= \left(\frac{\lambda_{\max}^{3/2}}{2\sqrt{\pi}} \frac{\Gamma(k-1/2)}{\Gamma(k+1)}\right) \|\mathrm{err}_k\|,
\end{aligned}$$

where we have made the change of variable $y = -x$. Note that

$$\lim_{k\to\infty} k^{3/2} \frac{\Gamma(k-1/2)}{\Gamma(k+1)} = 1.$$

This proves that lan-FA$_k(\sqrt{\cdot})$ converges somewhat faster than the Lanczos algorithm applied to the corresponding linear system $\mathbf{Ax} = \mathbf{v}$.

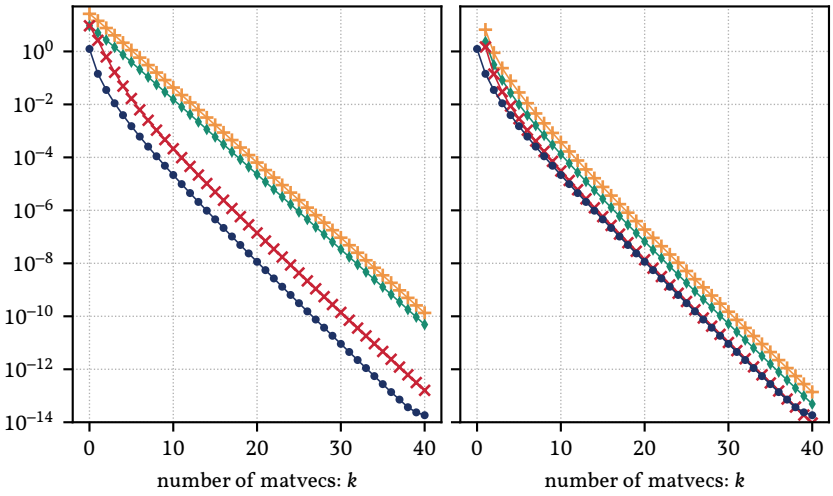


Figure 7.3: $\mathbf{A}$-norm error bounds for $f(x) = \sqrt{x}$ where $\mathbf{A}$ has $n = 1000$ eigenvalues spaced uniformly in $[10^{-2}, 10^{2}]$ and Γ is a circular contour (left) or Pac-Man contour (right). *Legend*: Lanczos-FA error (—•—), a priori bounds obtained by using Theorem 7.6 with $S = S_i = \mathcal{I}$ (—+—) and $S = S_i = \tilde{\mathcal{I}}(\mathbf{A}) = [\lambda_{\min}/2, 2\lambda_{\max}]$ (—♦—), a posteriori bounds obtained by using Theorem 7.6 with $S = \tilde{\mathcal{I}}$ (—×—). *Takeaway*: The a posteriori bounds tend to be quite accurate. The choice of contour impacts the quality of the bounds, particularly the a priori bounds.

In Figure 7.3 we plot the bounds from Theorem 7.6 for the circular and Pac-Man contours described above. For both contours we consider $S_i = \mathcal{I}$ for all i, as well as bounds based on an overestimate of this interval, $S_i = \tilde{\mathcal{I}}(\mathbf{A})$ where $\tilde{\mathcal{I}}(\mathbf{A}) = [\lambda_{\min}/2, 2\lambda_{\max}]$. This provides some sense of how sensitive the bounds are to the choice of S_i when S_i is a single interval. For a posteriori bounds, we set S_i to $\{\lambda_i(\mathbf{T}_k)\}$ for $i > 0$.

We remark that the bounds from Theorem 7.6 are upper bounds for (7.5) which implies that the slackness of (7.5) is relatively small. This suggests that the roughly 2 orders of magnitude improvement in Theorem 7.6 when moving from the circular contour to the Pac-Man contour is primarily due to reducing the slackness in (7.6), aligning with our intuition.

7.3.2 Piecewise analytic functions

We now discuss the application of Theorem 7.6 to piecewise analytic functions. Functions of this class have found widespread use throughout scientific computing and data science but have proven particularly difficult to analyze using existing approaches [NPS16; Fro+16; JS19; Esh+02].

Let $f(x)$ be one of $|x-a|$, $\text{step}(x-a)$, or $\text{step}(x-a)/x$ for $a \in \mathcal{I}$, where, for $z \in \mathbb{C}$ we define $\text{step}(z) := 0$ for $\text{Re}(z) < 0$ and $\text{step}(z) := 1$ for $\text{Re}(z) \geq 0$. Note that the latter two functions correspond to principle component projection and principle component regression respectively. In the case of principle component regression, we assume $\mathbf{A}$ is positive semi-definite. The step function is also closely related to the sign function, which is widely used in quantum chromodynamics to compute the overlap operator [Esh+02].

We take $w = a$ and define Γ_1 and Γ_2 as the boundaries of the disks

$$\mathcal{D}_1 := \mathcal{D}(\lambda_{\min}, w - \lambda_{\min} - \varepsilon) \quad \text{and} \quad \mathcal{D}_2 := \mathcal{D}(\lambda_{\max}, \lambda_{\max} - w - \varepsilon),$$

for some sufficiently small $\varepsilon > 0$. To extend $|x-a|$ to the complex plane, we replace $|x-a|$ by $z-a$ if $\text{Re}(z) > a$ and by $a-z$ if $\text{Re}(z) \leq a$. Then f is analytic in a neighborhood of the union of these two disks, so assuming none of the eigenvalues of $\mathbf{A}$ or $\mathbf{T}$ are equal to a, we can apply Lemma 7.9.

$f(x)$	$f(z), z \in \Omega_1$	$f(z), z \in \Omega_2$	$\frac{1}{2\pi}\sum_{j=1}^{2} \lvert\Gamma_j\rvert \max_{z\in\Gamma_j} \lvert f(z)\rvert$
$\lvert x-a\rvert$	$a-z$	$z-a$	$2(a-\lambda_{\min})^2 + 2(\lambda_{\max}-a)^2$
$\text{step}(x-a)$	0	1	$(\lambda_{\max}-a)$
$\text{step}(x-a)/x$	0	$1/z$	$(\lambda_{\max}-a)/a$

Table 7.1: Values of the factor in parentheses on the right-hand side of (7.9) (ignoring ε) for several common piecewise analytic functions.

Note that $\|h_{w,z}\|_{\mathcal{I}} \to 1$ as $z \to w$ from outside $[a,b]$, avoiding a potential singularity which would occur if the contour Γ passed through $\mathcal{I}$ at any other points. In fact, ignoring the contribution of ϵ, $\|h_{w,z}\|_{\mathcal{I}} = 1$ for all $z \in \Gamma_1$ and for

all $z \in \Gamma_2$. Thus, Corollary 7.11 can be written as

$$\|f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f)\| \leq \left(\frac{1}{2\pi} \sum_{j=1}^{2} |\Gamma_j| \max_{z \in \Gamma_j} |f(z)| \right) \|\text{err}_k(w)\|. \tag{7.9}$$

The values of this bound for all three functions are summarized in Table 7.1.

If $w \in \mathcal{I}$ we note that $\|\text{err}_k(w)\|$ corresponds to the indefinite linear system $(\mathbf{A} - w\mathbf{I})\mathbf{x} = \mathbf{v}$, so standard results for the Conjugate Gradient algorithm are not applicable. However, the residual of this system can still be computed exactly once the Lanczos factorization (1.3) has been obtained, and as we discuss in Section 7.4, a priori bounds for the convergence of MINRES [CG96] can be extended to the Lanczos algorithm for indefinite systems. It is also clear that, at the cost of having to compare against the error of multiple different linear systems, functions which are piecewise analytic on more than two regions can be handled.

In Figure 7.4, we plot the bounds from Theorem 7.6 for the contour described above for principle component regression with $f(x) = \text{step}(x - a)/x$. Here we use the same model as in Section 4.3.2. In particular, we set $n = 2000$, $d = 0.3$, and $\sigma = 8$ and take

$$\mathbf{A} = \frac{1}{m}\mathbf{\Sigma}^{1/2}\mathbf{X}\mathbf{X}^{\mathsf{H}}\mathbf{\Sigma}^{1/2},$$

where $m = n/d$, $\mathbf{X}$ is a $n \times m$ matrix with iid standard normal entries, and $\mathbf{\Sigma}$ a diagonal matrix with $1/m$ as the first $n/2$ entries and σ/m as the last $n/2$ entries. As discussed in Section 4.3.2, in the large n limit, the spectrum of such matrices is supported on intervals $[a_1, b_1] \cup [a_2, b_2]$, so we take $a = (b_1 + a_2)/2$ and $S = S_i = [a_1 - 0.1, b_2 + 0.1]$ for a priori bounds and $S = [a_1 - 0.1, b_2 + 0.1$ for a posteriori bounds. Thus, $f(x)$ corresponds to solving a linear system involving the eigenmodes of the right cluster of eigenvalues supported on $[a_2, b_2]$.

7.3.3 Quadratic forms

Let $f(x) = \text{step}(x - a)$ for $a \in \mathcal{I}$, and set $w = a$. Similarly to the previous example we use Theorem 7.7 to obtain a bound for the quadratic form error $|\mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v} - \mathbf{v}^{\mathsf{H}}\text{lan-FA}_k(f)|$. However, since $\|h_z\|_{S_i}$ has singularities at each point in S_i, we must have S_i avoid where Γ crosses the real axis.

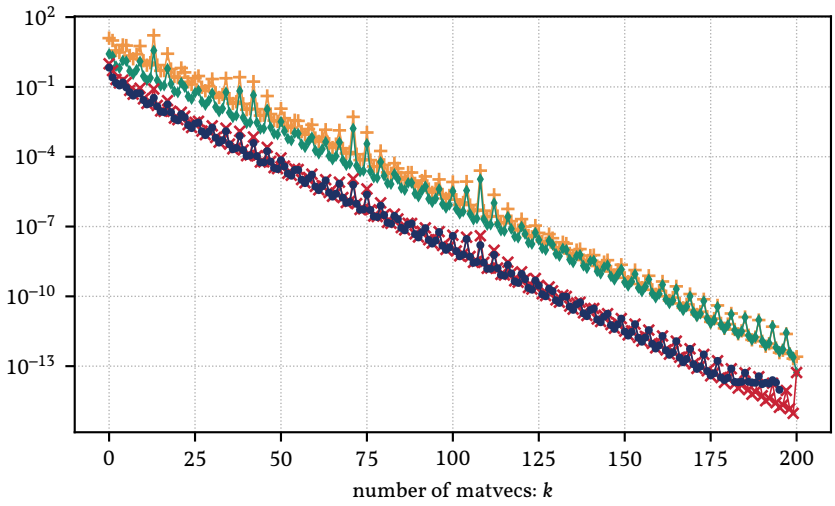


Figure 7.4: $(\mathbf{A}-w\mathbf{I})^2$-norm error bounds for $f(x) = \text{step}(x-a)/x$ where $\mathbf{A}$ is a random matrix whose limiting density is supported on $[a_1, b_1] \cup [a_2, b_2]$, $a = (b_1+a_2)/2$, and Γ is a double circle contour. *Legend*: Lanczos-FA error (—•—), a priori bounds obtained by using Theorem 7.6 with $S = S_i = [a_1 - 0.1, b_2 + 0.1]$ (—♦—) and (7.9) with the values from Table 7.1 (—+—) a posteriori bounds obtained by using Theorem 7.6 with $S = [a_1 - 0.1, b_2 + 0.1]$ (—×—). *Takeaway*: The bounds work well, even for pieceiwise analytic functions.

Suppose $\lambda^{\text{l},w}_{\text{max}}(\mathbf{A})$ and $\lambda^{\text{r},w}_{\text{min}}(\mathbf{A})$ are consecutive eigenvalues of $\mathbf{A}$ so that $\lambda^{\text{l},w}_{\text{max}}(\mathbf{A}) < w < \lambda^{\text{r},w}_{\text{min}}(\mathbf{A})$. Then we can define

$$\mathcal{I}_w(\mathbf{A}) := [\lambda_{\text{min}}, \lambda^{\text{l},w}_{\text{max}}(\mathbf{A})] \cup [\lambda^{\text{r},w}_{\text{min}}(\mathbf{A}), \lambda_{\text{max}}].$$

In this case, $\|h_z\|_{\mathcal{I}_w(\mathbf{A})} = \max\{\|h_z\|_{[\lambda_{\text{min}}, \lambda^{\text{l},w}_{\text{max}}]}, \|h_z\|_{[\lambda^{\text{r},w}_{\text{min}}, \lambda_{\text{max}}]})\}$ can be computed using Lemma 7.10. We can then apply Theorem 7.7 to obtain a bound for the quadratic form error $|\mathbf{v}^{\text{H}} f(\mathbf{A})\mathbf{v} - \mathbf{v}^{\text{H}}\text{lan-FA}_k(f)|$. A priori bounds are obtained with $S_0, S_i = \mathcal{I}_w(\mathbf{A})$ while a posteriori bounds are obtained with $S = \mathcal{I}_w(\mathbf{A})$ and $S_i = \{\lambda_i(\mathbf{T})\}$.

In Figure 7.5 we use the same matrix as in Section 7.3.2. This time, however, we use $S = S_i = [a_1 - 0.1, b_1 + 0.1] \cup [a_2 - 0.1, b_2 + 0.1]$ for a priori bounds and $S = [a_1 - 0.1, b_1 + 0.1] \cup [a_2 - 0.1, b_2 + 0.1]$ for a posteriori bounds. Note that the squared error $\|f(\mathbf{A})\mathbf{v} - \text{lan-FA}_k(f)\|^2$ is close to that of the quadratic form error $|\mathbf{v}^{\text{H}} f(\mathbf{A})\mathbf{v} - \int f \, \text{d}[\Psi]^{\text{gq}}_{2k-1}|$.

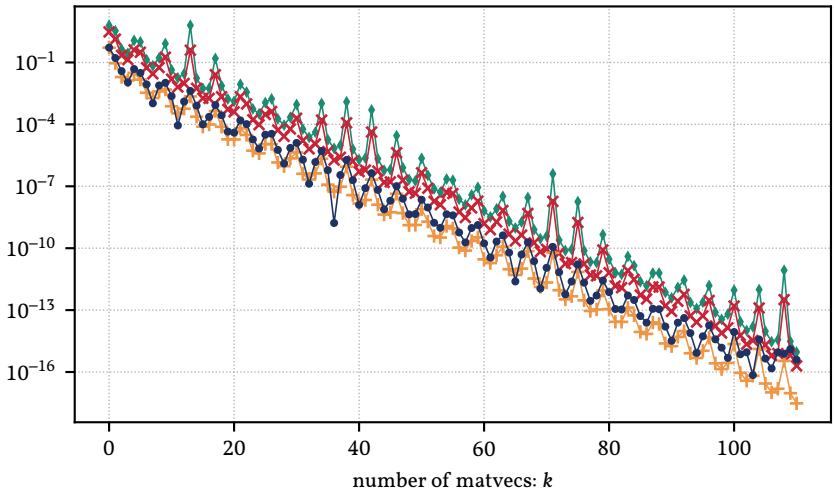


Figure 7.5: Quadratic form error bounds for $f(x) = \text{step}(x-a)$ where $\mathbf{A}$ is a random matrix whose limiting density is supported on $[a_1, b_1] \cup [a_2, b_2], a = (b_1+a_2)/2$, and Γ is a double circle contour. *Legend*: Lanczos-FA quadratic form error (—•—), squared Lanczos-FA 2-norm (—+—), a priori bounds obtained by using Theorem 7.6 with $S = S_i = [a_1-0.1, b_1+0.1] \cup [a_2-0.1, b_2+0.1]$ (—♦—) a posteriori bounds obtained by using Theorem 7.6 with $S = [a_1-0.1, b_1+0.1] \cup [a_2-0.1, b_2+0.1]$ (—×—). *Legend*: The bounds are applicable to quadratic forms.

7.4 Error bounds for Lanczos-FA on indefinite systems

In this section, we review several results which rigorously justify the claim that, for any choice of w with $\mathbf{A} - w\mathbf{I}$ invertible, $\|\text{err}_k(w)\|$ satisfies a spectrum-dependent error bound.

Note that on indefinite problems, the standard implementation of CG (or the LDL version on which Lanczos-OR is based) may fail in such situations since $\mathbf{T}$ can be singular in which case the inversion will break down. As a result, in such situations, it is standard practice to use MINRES or other related algorithms. On the other hand, the Lanczos algorithm does not break down if $\mathbf{T}$ is singular, and so the Lanczos-FA approximation to $\mathbf{A}^{-1}\mathbf{v}$ can be computed whenever $\mathbf{T}$ is non-singular, even if it was singular at earlier iterations. Interestingly, however, the "overall" convergence of the algorithm tends to be comparable to MINRES in

the sense that at many iterations the error is quite similar.

The first result we remark on was first proved in [CG96] compares the residual norms of Lanczos-FA and MINRES.

Theorem 7.13. *Let* $\mathbf{A}$ *be a nonsingular Hermitian matrix and define* $\mathbf{r}_k^M$ *as the MINRES residual at step* k*; i.e.*

$$\mathbf{r}_k^M := \mathbf{v} - \mathbf{A}\hat{\mathbf{y}}, \qquad \hat{\mathbf{y}} = \operatorname*{argmin}_{\mathbf{y} \in \mathcal{K}_k} \|\mathbf{v} - \mathbf{A}\mathbf{y}\|_2.$$

Then, assuming that the initial residuals in the two procedures are the same,

$$\frac{\|\mathsf{res}_k\|_2}{\|\mathsf{res}_0\|_2} = \frac{\|\mathbf{r}_k^M\|_2 / \|\mathbf{r}_0^M\|_2}{\sqrt{1 - \left(\|\mathbf{r}_k^M\|_2 / \|\mathbf{r}_{k-1}^M\|_2\right)^2}}.$$

Therefore, we see that if MINRES makes good progress at step k (i.e. $\|\mathbf{r}_k^M\|_2 / \|\mathbf{r}_{k-1}^M\|_2$ is small), then Theorem 7.13 implies $\|\mathsf{res}_k\|_2 / \|\mathsf{res}_0\|_2 \approx \|\mathbf{r}_k^M\|_2 / \|\mathbf{r}_0^M\|_2$. Thus, since MINRES converges at a linear rate, there must be iterations where the MINRES residual norm decreases enough that the Lanczos-FA residual norm is similarly small. This is made precise in [Che+22, Corollary A.2] which demonstrates the iteration complexity of Lanczos-FA on indefinite systems is nearly the same as that of MINRES.

In fact, stronger results are known. In particular, it is known that Lanczos-FA process iterates whose residuals satisfy a minimax bound on the eigenvalues of $\mathbf{A}$, at least at every other iteration [GDK99]. While not well known, the argument proving this claim is amazingly simple, so we provide proofs for the exact arithmetic case. These results hold to close approximation in finite precision arithmetic; see [GDK99] for the statements and proofs in this setting.

We begin with several lemmas.

Lemma 7.14. *Suppose* θ *is an eigenvalue of* $\mathbf{T}$ *with eigenvector* $\mathbf{s}$ *and* μ *is an eigenvalue of* $[\mathbf{T}]_{:k-1,:k-1}$ *with eigenvector* $\mathbf{v}$*. Then,*

$$\theta - \mu = \frac{\beta_{k-2}[\mathbf{s}]_{k-1}[\mathbf{v}]_{k-2}}{\mathbf{s}^{\mathsf{H}}\hat{\mathbf{v}}}$$

where $\hat{\mathbf{v}}$ *is* $\mathbf{v}$ *with a zero appended at the bottom.*

Proof. Observe that

$$\theta \mathbf{s}^{\mathsf{H}}\hat{\mathbf{v}} = \mathbf{s}^{\mathsf{H}}\mathbf{T}\hat{\mathbf{v}} = \mathbf{s}^{\mathsf{H}}(\mu\hat{\mathbf{v}} + \beta_{k-2}[\mathbf{v}]_{k-2}\mathbf{e}_{k-1}) = \mu\mathbf{s}^{\mathsf{H}}\hat{\mathbf{v}} + \beta_{k-2}[\mathbf{s}]_{k-1}[\mathbf{v}]_{k-2}.$$

The result follows by rearranging the above expression. □

Lemma 7.15. *Suppose θ is an eigenvalue of $\mathbf{T}$ with eigenvector $\mathbf{s}$. Then*

$$\min_{0\le i<n} |\theta - \lambda_i| \le |\beta_{k-1}[\mathbf{s}]_{k-1}|.$$

Proof. Using the Lanczos recurrence (1.3) and the eigendecomposition $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathsf{H}}$ we find,

$$\beta_{k-1}\mathbf{q}_k\mathbf{e}_{k-1}^{\mathsf{H}}\mathbf{s} = \mathbf{A}\mathbf{Q}\mathbf{s} - \mathbf{Q}\mathbf{T}\mathbf{s} = \mathbf{A}\mathbf{Q}\mathbf{s} - \theta\mathbf{Q}\mathbf{s} = \mathbf{U}(\mathbf{\Lambda} - \theta\mathbf{I})\mathbf{U}^{\mathsf{H}}\mathbf{Q}\mathbf{s}.$$

Rearranging and taking norms on both sides we then have

$$1 = \|\mathbf{Q}\mathbf{s}\| = \|\beta_{k-1}\mathbf{U}(\Lambda - \theta\mathbf{I})^{-1}\mathbf{q}_k\mathbf{e}_{k-1}^{\mathsf{H}}\mathbf{s}\|_2 \le |\beta_{k-1}[\mathbf{s}]_{k-1}|\|(\mathbf{\Lambda} - \theta\mathbf{I})^{-1}\|_2$$

where we have used that $\|\mathbf{U}\|_2 = 1$ and $\|\mathbf{Q}\mathbf{s}\|_2 = 1$. The result then follows from the fact that

$$\|(\mathbf{\Lambda} - \theta\mathbf{I})^{-1}\|_2^{-1} = \big(\max_{0\le i<n} |\lambda_i - \theta|^{-1}\big)^{-1} = \min_{0\le i<n} |\lambda_i - \theta|.$$ □

The first main result of [GDK99] asserts that eigenvalues of $\mathbf{T}$ are bounded away from zero at least at every other iteration, provided $\mathbf{A}$ is not singular.

Theorem 7.16. *Suppose θ is an eigenvalue of $\mathbf{T}$ and μ is an eigenvalue of $[\mathbf{T}]_{:k-1,:k-1}$. Then*

$$\frac{\max\{|\theta|, |\mu|\}}{\|\mathbf{A}\|} \ge \frac{\kappa^2}{2+\sqrt{3}}.$$

Proof. Applying Lemma 7.15 to $\mathbf{T}$ and $[\mathbf{T}]_{:k-1,:k-1}$ and then using Lemma 7.14 and the fact that $\|\mathbf{s}\|_2 = \|\mathbf{v}\|_2 = 1$ we have

$$|\theta||\mu| \le \beta_{k-2}\beta_{k-1}[\mathbf{s}]_{k-1}[\mathbf{v}]_{k-2} = \beta_{k-1}|\theta - \mu|\mathbf{s}^{\mathsf{H}}\hat{\mathbf{v}} \le \beta_{k-1}|\theta - \mu|.$$

Let $\tau := \max\{|\theta|, |\mu|\}$. Then $|\theta - \mu| \le 2\tau$, $|\theta| \ge \sigma_{\min} - \tau$, and $|\mu| \ge \sigma_{\min} - \tau$. This implies that

$$(\sigma_{\min} - \tau)^2 \le |\theta||\mu| \le \beta_{k-1}|\theta - \mu| \le 2\beta_{k-1}\tau.$$

Solving for τ we find

$$\tau \geq \frac{\sigma_{\min}^2}{\sigma_{\min} + \beta_{k-1} + \sqrt{\beta_{k-1}^2 + 2\beta_{k-1}\sigma_{\min}}} \geq \frac{\kappa}{2+\sqrt{3}}$$

where, in the final inequality, we have use the fact that both β_{k-1} and $\sigma_{\min}$ are bounded above by $\|\mathbf{A}\|_2$. □

We are nearly ready to show that the Lanczos-FA iterate satisfies a minimax bound on Λ. First, however, we require the following lemma relating the bottom left entry of $\mathbf{T}^{-1}$ to the norm of $\mathbf{T}^{-1}$.

Lemma 7.17.

$$|\mathbf{e}_{k-1}^{\mathsf{H}} \mathbf{T}^{-1} \mathbf{e}_0| \leq \|\mathbf{T}^{-1}\|_2 \min_{\deg(p)<k, p(0)=1} \|p(\mathbf{T})\mathbf{e}_1\|_2.$$

Proof. Write $p = 1 - xq$ for some q with $\deg(q) < k-1$. Then $\mathbf{e}_{k-1}^{\mathsf{H}} q(\mathbf{T})\mathbf{e}_0 = 0$ so

$$\mathbf{e}_{k-1}^{\mathsf{H}} \mathbf{T}^{-1} \mathbf{e}_0 = \mathbf{e}_{k-1}^{\mathsf{H}} (\mathbf{T}^{-1} - q(\mathbf{T}))\mathbf{e}_0 = \mathbf{e}_{k-1}^{\mathsf{H}} \mathbf{T}^{-1} p(\mathbf{T})\mathbf{e}_0.$$

Now, applying a submultiplicative bound, we find

$$|\mathbf{e}_{k-1}^{\mathsf{H}} \mathbf{T}^{-1} \mathbf{e}_0| \leq \|\mathbf{T}^{-1}\|_2 \|p(\mathbf{T})\mathbf{e}_1\|_2.$$

The result follows by optimizing over p. □

Proving a minimax bound for the Lanczos-FA residual is now straightforward.

Theorem 7.18. *At least at every other iteration,*

$$\|\mathrm{res}_k\|_2 \leq \frac{\kappa^2}{2+\sqrt{3}} l \min_{\deg(p)<k, p(0)=1} \|p\|_\Lambda$$

Proof. From Lemma 7.2 we see have that

$$\|\mathrm{res}_k\|_2 = |\beta_{k-1} \mathbf{e}_0^{\mathsf{H}} \mathbf{T}^{-1} \mathbf{e}_0|.$$

The result then follows immediately from Lemma 7.17 and Theorem 7.16. □

Interestingly, it is known that $\mathbf{T}$ cannot have eigenvalues near zero in two successive iterations, at least assuming that the eigenvalues of $\mathbf{A}$ are not too close to zero. Specifically, [GDK99, Equation 3.10] asserts that

$$\max\{\sigma_{\min}([\mathbf{T}]_{:k-1,:k-1}), \sigma_{\min}(\mathbf{T})\} > \frac{\sigma_{\min}(\mathbf{A})^2}{(2+\sqrt{3})\|\mathbf{A}\|_2}. \tag{7.10}$$

Thus, as noted in Theorem 6.2, we might still use the Lanczos-FA approximation to $\mathbf{A}^{-1}\mathbf{v}$.

Chapter 8
Finite precision arithmetic

As mentioned in the introduction and observed in the many numerical experiments throughout this thesis, even in finite precision arithmetic, Lanczos-based methods for matrix functions tend to perform at least as well as their explicit polynomial counterparts. In fact, they often perform significantly better. This is in direct conflict with the widespread notion that costly reorthogonalization schemes are necessary for Lanczos-based methods [JP94; Aic+03; Wei+06; UCS17; GWG19].

In this chapter, we provide an overview of several existing theoretically rigorous results which explain this phenomenon. We also prove that the reduction technique from Chapter 7 still works in finite precision arithmetic. It is our hope that our treatment of this topic will provide an accessible starting point for those outside of numerical analysis to better understand the impact of finite precision arithmetic on Lanczos-based methods. Thus, while this chapter consists mostly of exposition on existing results, we believe it to be it to be one of the more important contributions of this thesis.

In this chapter, we will use $\|\cdot\|$ rather than $\|\cdot\|_2$ to denote the operator 2-norm of matrices and Euclidean norm of vectors. We will also make the simplifying assumption that $\mathbf{A}$ is real symmetric.

8.1 Preliminaries

Almost all of modern scientific computing involves computations in finite precision arithmetic, and specifically, floating point arithmetic. The introduction of rounding errors can have potentially large impacts on the output of an algorithm, so accounting for these errors is important. In fact, this is one of the main goals of the field of numerical analysis.

There are many possible implementations of floating point arithmetic and other low-level math kernels. For instance, the number of bits of precision may vary, the rounding scheme may vary, the way basic functions such as the square root and logarithm are implemented may vary, etc. To avoid the need for a separate analysis of each implementation, it is standard to work in a model of computation which captures the essential qualities of a broad number of basic math routines.

Perhaps the most commonly studied model of finite precision computing assumes that basic operations are carried out to relative accuracy ϵ_{mach}, a constant referred to as the machine precision. For floating point numbers α and β and standard binary arithmetic operations $\circ \in \{+, -, \times, \div\}$, these assumptions take the form

$$|\text{fp}(\alpha \circ \beta) - \alpha \circ \beta| \leq \epsilon_{\text{mach}} |\alpha \circ \beta|.$$

Similar assumptions are also made for unary operations such as the square root,

$$|\text{fp}(\sqrt{\alpha}) - \sqrt{\alpha}| \leq \epsilon_{\text{mach}} |\sqrt{\alpha}|.$$

Assuming overflow and underflow do not occur, the above assumptions are satisfied for IEEE 754 floating point arithmetic [**ieee_19**]. Since the vast majority of modern computers use IEEE 754 floating point arithmetic, such assumptions are relatively safe.[1]

Under the above assumptions, the accuracy of basic linear algebraic primitives can be bounded. For instance, for floating point vectors $\mathbf{x}$, $\mathbf{y}$, floating point

[1] It is worth noting, however, that the above bounds do not necessarily hold for other number systems. Notable examples include Cray supercomputers prior to the mid 1990s as well as a number of other early computers [Hig02]. In fact, with the recent rise in low precision number formats and custom hardware acceleration methods, the above bounds cannot be universally assumed [Fas+21].

number α, and floating point matrix $\mathbf{A}$, reasonable implementations of basic linear algebraic tasks [Hig02] will satisfy

$$\begin{aligned} \|\mathrm{fp}(\mathbf{x}+\alpha\mathbf{y})-(\mathbf{x}+\alpha\mathbf{y})\| &\leq \epsilon_{\mathrm{mach}}\left(\|\mathbf{x}\|+2|\alpha|\|\mathbf{y}\|\right) \\ \|\mathrm{fp}(\langle\mathbf{x},\mathbf{y}\rangle)-\langle\mathbf{x},\mathbf{y}\rangle\| &\leq \epsilon_{\mathrm{mach}}\, n\, \|\mathbf{x}\|\|\mathbf{y}\| \\ \|\mathrm{fp}(\mathbf{A}\mathbf{x})-\mathbf{A}\mathbf{x}\| &\leq \epsilon_{\mathrm{mach}}\, c\, \|\mathbf{A}\|\|\mathbf{x}\|. \end{aligned}$$

Here $c \leq n^{3/2}$ is a dimensional constant depending on the method of matrix multiplication and the sparsity of $\mathbf{A}$ which is often written in terms of the ratio of the norm of the absolute value of $\mathbf{A}$ and the norm of $\mathbf{A}$. These results can then be applied to analyze linear algebra routines.

8.2 Three term recurrences

The Lanczos algorithm as well as the explicit polynomial methods Algorithms 1.1 and 3.2 from Chapter 3 compute $\mathbf{q}_{i+1}$ by a symmetric three term recurrence of the form

$$\mathbf{q}_{i+1} = \frac{1}{\beta_i}\left(\mathbf{A}\mathbf{q}_i - \alpha_i\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1}\right).$$

In Algorithms 3.1 and 3.2 the coefficients are predetermined, whereas Lanczos chooses the coefficients adaptively in order to enforce orthogonality. Regardless, in finite precision arithmetic, we will instead have a perturbed recurrence

$$\mathbf{q}_{i+1} = \frac{1}{\beta_i}\left(\mathbf{A}\mathbf{q}_i - \alpha_i\mathbf{q}_i - \beta_{i-1}\mathbf{q}_{i-1}\right) + \mathbf{f}_{i+1}$$

where $\mathbf{f}_{i+1}$ accounts for local rounding errors made in the computation of $\mathbf{q}_{i+1}$. These simple arithmetic computations are all stable in the sense described above, so $\mathbf{f}_{i+1}$ is small (on the order of $\epsilon_{\mathrm{mach}}\|\mathbf{A}\|$) relative to the involved quantities.

While $\mathbf{q}_{i+1} = p_{i+1}(\mathbf{A})\mathbf{v}$ in exact arithmetic, this is no longer the case in finite precision arithmetic. Indeed, the difference between $\mathbf{q}_{i+1}$ and $p_{i+1}(\mathbf{A})\mathbf{v}$ depends on the *associated polynomials* of the recurrence applied to the $\mathbf{f}_j$'s; see for instance [Meu06]. In the case of the Chebyshev polynomials of the first kind, the associated polynomials are well behaved on $[-1,1]$[2], so it can be shown that

[2] In fact, the associated polynomials are the Chebyshev polynomials of the second kind.

$\mathbf{q}_{i+1} \approx p_{i+1}(\mathbf{A})\mathbf{v}$. Here "$\approx$" means the error has a polynomial dependence on k and a linear dependence on the machine precision, along with other reasonable dependence on the dimension and matrix norm. This can be easily seen by performing an analysis similar to that in the proof of Theorem 8.4 found later in this section.

As such, the computed modified moments for $\mu = \mu_{a,b}^T$ can be expected to be near to the true modified moments and Lemma 3.16 can be expected to hold to close degree as long as $I \subset [a,b]$. On the other hand, for different $\{\alpha_i\}_{i=0}^{\infty}$ and $\{\beta_i\}_{i=0}^{\infty}$, for instance those generated by the Lanczos algorithm, the associated polynomials may grow exponentially in I and the modified moments obtained from the finite precision computation may differ greatly from their exact arithmetic counterparts unless very high precision is used. In fact, this situation includes $\mu_{a,b}^T$ if a and b are not chosen so that $I \subset [a,b]$.

8.2.1 The Lanczos algorithm

In the case of Lanczos, the coefficients are computed adaptively and therefore depend on $\mathbf{q}_{i-1}$, $\mathbf{q}_i$, and $\mathbf{q}_{i+1}$. It is well known that even if the $\mathbf{f}_j$'s are small, the coefficients produced by Lanczos run in finite precision arithmetic may differ *greatly* from what would be obtained in exact arithmetic and the Lanczos vectors $\{\mathbf{q}_j\}_{j=1}^{k+1}$ need not be orthogonal. Moreover, the tridiagonal matrix $\mathbf{T}$ from the finite precision computation may have multiple "ghost" eigenvalues near certain eigenvalues of $\mathbf{A}$, even when the eigenvalues of $\mathbf{T}$ would have been well separated in exact arithmetic. In this sense, the algorithm is notoriously unstable, and such instabilities can appear even after only a few iterations.

A great deal is known about the Lanczos algorithm in finite precision arithmetic; see for instance [Gre97; MS06; Meu06]. In this section, we summarize the content needed to understanding why Lanczos-based methods for matrix functions still work in finite precision arithmetic. A fully rigorous and self-contained treatment of the topic would be very long and exceedingly technical, and such a treatment would not improve understanding of the big-picture (in fact, it might do exactly the opposite). As such, we aim to provide an overview of existing theory which provides an intuitive understanding.

Throughout the next several sections the symbol "$\lesssim$" suppresses absolute con-

stants (e.g. 5, 12, and 1/10) and higher order terms in the machine precision, ϵ_{mach} (e.g. ϵ_{mach}^2). The results in the literature typically state explicit constants for terms linear in ϵ_{mach}, but such a precise analysis is not necessary to understand the intuition we wish to convey.

We will write the matrix-form of the perturbed Lanczos recurrence as

$$\mathbf{AQ} = \mathbf{QT} + \beta_{k-1}\mathbf{q}_k\mathbf{e}_{k-1}^\top + \mathbf{F}. \tag{8.1}$$

We denote by $\mathbf{R}$ the strictly upper triangular part of $\mathbf{Q}^\top\mathbf{Q}$; i.e.

$$\mathbf{Q}^\top\mathbf{Q} = \mathbf{R} + \mathbf{R}^\top + \mathbf{D}.$$

It can then be shown that

$$\mathbf{TR} = \mathbf{RT} - \beta_{k-1}\mathbf{Q}^\top\mathbf{q}_k\mathbf{e}_{k-1}^\top + \mathbf{H}$$

for some upper triangular perturbation term $\mathbf{H}$ which is $\mathbf{0}$ in exact arithmetic; see for instance [Pai76, Equation 41]. We will also denote by η the minimum value so that

$$\Lambda(\mathbf{T}) \subset [\lambda_{\min} - \eta, \lambda_{\max} + \eta].$$

The first work which truly explained why the Lanczos algorithm was still useful as an eigenvalue algorithm was the PhD thesis [Pai71] of Paige (and the technical reports leading up to the thesis). The results in [Pai71] were subsequently simplified and extended in [Pai72; Pai76; Pai80]. The main result we need is [Pai76, Theorem 1] which we have simplified to the needs of this chapter.

Theorem 8.1. *Suppose the implementation of the Lanczos algorithm given in Algorithm 1.1 is run in finite precision arithmetic with machine precision ϵ_{mach}. Then, for $i < k$, under some mild technical assumptions on ϵ_{mach}, the following quantities*

$$\|\mathbf{D} - \mathbf{I}\|, \qquad \|\mathbf{F}\|, \qquad \|\mathbf{H}\|, \qquad \eta$$

are bounded by $O(k^\alpha n^\beta \|\mathbf{A}\| \epsilon_{\text{mach}})$ for small constants α, β.

Remark 8.2. The full version of Theorem 8.1 from [Pai76] is significantly more precise than our above statement might suggest. In particular, the above quantities (and several others) are each explicitly bounded to first order in ϵ_{mach}, with the constants and the dependence on k, n, and $\|\mathbf{A}\|$ stated explicitly for each quantity. △

In exact arithmetic, Lanczos-based approaches such as Gaussian quadrature and Lanczos-FA apply sufficiently low degree polynomials exactly. We will now show that, even in finite precision arithmetic, these approaches apply (appropriately scaled) Chebyshev polynomials accurately. For functions which have a Chebyshev expansion with bounded coefficients, this implies that Lemma 3.16 and Theorem 6.4 holds to close degree. Thus, Lanczos-based approaches should be expected to perform at least as well as explicit polynomial approaches for most reasonable functions f.

For convenience, from this point to the end of the chapter, we will assume that $\mathbf{A}$ has been shifted and scaled so that Λ and $\Lambda(\mathbf{T})$ are each contained in $[-1, 1]$.

Recall that the Chebyshev polynomials of the first kind satisfy the recurrence

$$T_i = 2xT_{i-1} - T_{i-2}, \qquad T_1 = x, \qquad T_0 = 1.$$

and that the Chebyshev polynomials of the second kind satisfy the recurrence

$$U_i = 2xU_{i-1} - U_{i-2}, \qquad U_1 = 2x, \qquad U_0 = 1.$$

We have the following, well known, bound.

Lemma 8.3. *For all $j \geq 0$, $\|T_j\|_{[-1,1]} \leq 1$ and $\|U_j\|_{[-1,1]} \leq j + 1$.*

For notational brevity, for $i \geq 1$, introduce the vectors

$$\mathbf{t}_i = T_i(\mathbf{A})\mathbf{v}, \qquad \tilde{\mathbf{t}}_i = T_i(\mathbf{T})\mathbf{e}_0, \qquad \boldsymbol{\psi}_i = \mathbf{t}_i - \mathbf{Q}\tilde{\mathbf{t}}_i.$$

8.3 Lanczos-FA

To the best of our knowledge, the result in this section first appeared in [DK91, Section 4]; see also [MMS18, Lemma 10].

Theorem 8.4. *For all $i = 0, 1, \ldots, k-1$,*

$$\|T_i(\mathbf{A})\mathbf{v} - \mathbf{Q}T_i(\mathbf{T})\mathbf{e}_0\|_2 \leq k^2\|\mathbf{F}\|.$$

Proof. Using (8.1) and Corollary 10.3, observe that for $i > 1$, the $\mathbf{d}_i$ satisfy a perturbed three term recurrence

$$\boldsymbol{\psi}_i = (2\mathbf{A}\mathbf{t}_{i-1} - \mathbf{t}_{i-2}) - (2\mathbf{Q}\mathbf{T}\tilde{\mathbf{t}}_{i-1} - \mathbf{Q}\tilde{\mathbf{t}}_{i-2})$$

$$
\begin{aligned}
&= 2(\mathbf{A}\mathbf{t}_{i-1} - (\mathbf{A}\mathbf{Q}\tilde{\mathbf{t}}_{i-1} + \beta_k \mathbf{q}_{k-1}\mathbf{e}_{k-1}^\top \tilde{\mathbf{t}}_{i-1} + \mathbf{F}\tilde{\mathbf{t}}_{i-1})) - \boldsymbol{\psi}_{i-2} \\
&= 2\mathbf{A}\boldsymbol{\psi}_{i-1} - \boldsymbol{\psi}_{i-2} + 2\mathbf{F}\tilde{\mathbf{t}}_{i-1}.
\end{aligned}
$$

By direct computation, we also have $\boldsymbol{\psi}_0 = \mathbf{0}$ and $\boldsymbol{\psi}_1 = \mathbf{F}\tilde{\mathbf{t}}_0$. Then, it's easy to see that

$$
\mathbf{d}_i = U_{i-1}(\mathbf{A})\mathbf{F}\tilde{\mathbf{t}}_0 + 2\sum_{j=1}^{i} U_{i-j-1}(\mathbf{A})\mathbf{F}\tilde{\mathbf{t}}_j.
$$

We now use Theorem 8.3 to obtain

$$
\|\mathbf{d}_i\| \leq 2\sum_{j=0}^{i-1} \|U_{i-j-1}(\mathbf{A})\|\|\mathbf{F}\|\|\tilde{\mathbf{t}}_j\| \leq 2\sum_{j=0}^{i-1}(i-j)\|\mathbf{F}\| \leq 2i^2\|\mathbf{F}\|. \qquad \square
$$

8.4 Gaussian quadrature

The results in this section are summarized from [Kni96]. Interestingly, this work seems to be relatively unknown, despite its significance given the widespread use of Lanczos-based quadrature methods. It is our hope that the resurfacing of these results will help assuage some of the hesitancy to use Lanczos-based quadrature methods without reorthogonalization.

We begin with several lemmas.

Lemma 8.5. *For all* $i = 0, 1, \ldots, k-1$,

$$
\|\mathbf{R}T_i(\mathbf{T})\mathbf{e}_0\| \leq k^2\|\mathbf{H}\|.
$$

Proof. Write $\boldsymbol{\Delta}_i = \mathbf{R}T_i(\mathbf{T})\mathbf{e}_0$. Using Section 8.2.1 and Corollary 10.3, for $i > 1$, analogous to Theorem 8.4, the $\boldsymbol{\Delta}_i$ satisfy the perturbed three term recurrence

$$
\begin{aligned}
\boldsymbol{\Delta}_i &= 2\mathbf{R}\mathbf{T}\tilde{\mathbf{t}}_{i-1} - \mathbf{R}\tilde{\mathbf{t}}_{i-2} \\
&= 2(\mathbf{T}\mathbf{R}\tilde{\mathbf{t}}_{i-1} + (\beta_{k-1}\mathbf{Q}^\top \mathbf{q}_k \mathbf{e}_{k-1}^\top \tilde{\mathbf{t}}_{i-1} + \mathbf{H}\tilde{\mathbf{t}}_{i-1}) - \mathbf{R}\boldsymbol{\Delta}_{i-2} \\
&= 2\mathbf{T}\boldsymbol{\Delta}_{i-1} - \boldsymbol{\Delta}_{i-2} + 2\mathbf{H}\tilde{\mathbf{t}}_{j-1}
\end{aligned}
$$

Since $\mathbf{R}$ is strictly upper triangular we have $\boldsymbol{\Delta}_0 = \mathbf{0}$ and, by direct computation, $\boldsymbol{\Delta}_1 = \mathbf{H}\mathbf{e}_0$. This implies that

$$
\boldsymbol{\Delta}_i = U_{i-1}(\mathbf{T})\mathbf{H}\mathbf{e}_0 + 2\sum_{j=1}^{i-1} U_{i-j-1}(\mathbf{T})\mathbf{H}\tilde{\mathbf{t}}_j.
$$

We again use Theorem 8.3 to obtain

$$\|\Delta_i\| \leq 2\sum_{j=0}^{i-1} \|U_{i-j-1}(\mathbf{T})\|\|\mathbf{H}\|\|\tilde{\mathbf{t}}_j\| \leq 2\sum_{j=0}^{i-1}(i-j)\|\mathbf{H}\| \leq 2i^2\|\mathbf{H}\|. \qquad \square$$

We now state the main result.

Theorem 8.6. *For all* $i = 0, 1, \ldots, 2k-2$

$$|\mathbf{v}^\mathsf{T} T_i(A) - \mathbf{e}_0^\mathsf{T} T_i(\mathbf{T})\mathbf{e}_0| \leq k^2(\|\mathbf{F}\| + \|\mathbf{H}\|) + \|\mathbf{D}-\mathbf{I}\|.$$

Proof. Recall that the Chebyshev polynomial satisfy the identities

$$T_{2i} = 2(T_i)^2 - 1, \qquad T_{2i+1} = 2T_iT_{i+1} - x.$$

It therefore suffices to bound $|\mathbf{v}^\mathsf{T} T_i(\mathbf{A})T_j(\mathbf{A})\mathbf{v} - \mathbf{e}_0^\mathsf{T} T_i(\mathbf{T})T_j(\mathbf{T})\mathbf{e}_0|$.

By definition,

$$\mathbf{t}_i^\mathsf{T}\mathbf{t}_j = (\tilde{\mathbf{t}}_i\mathbf{Q}^\mathsf{T} + \boldsymbol{\psi}_i^\mathsf{T})(\boldsymbol{\psi}_j + \mathbf{Q}\tilde{\mathbf{t}}_j)$$

Thus,

$$|\mathbf{t}_i^\mathsf{T}\mathbf{t}_j - \tilde{\mathbf{t}}_i^\mathsf{T}\tilde{\mathbf{t}}_j| \leq |\tilde{\mathbf{t}}_i\mathbf{Q}^\mathsf{T}\mathbf{Q}\tilde{\mathbf{t}}_j - \tilde{\mathbf{t}}_i^\mathsf{T}\tilde{\mathbf{t}}_j| + \|\boldsymbol{\psi}_i\|\|\mathbf{Q}\tilde{\mathbf{t}}_j\| + \|\boldsymbol{\psi}_j\|\|\mathbf{Q}\tilde{\mathbf{t}}_i\| + \|\boldsymbol{\psi}_i\|\|\boldsymbol{\psi}_j\|$$

By definition of $\mathbf{R}$,

$$\tilde{\mathbf{t}}_i^\mathsf{T}\mathbf{Q}^\mathsf{T}\mathbf{Q}\tilde{\mathbf{t}}_j = \tilde{\mathbf{t}}_i^\mathsf{T}(\mathbf{R} + \mathbf{R}^\mathsf{T} + \mathbf{I} + (\mathbf{D}-\mathbf{I}))\tilde{\mathbf{t}}_j.$$

Thus, applying Theorems 8.1, 8.3 and 8.5 we have

$$|\tilde{\mathbf{t}}_i\mathbf{Q}^\mathsf{T}\mathbf{Q}\tilde{\mathbf{t}}_j - \tilde{\mathbf{t}}_i^\mathsf{T}\tilde{\mathbf{t}}_j| \leq \|\tilde{\mathbf{t}}_j\|\|\mathbf{R}\tilde{\mathbf{t}}_i\| + \|\tilde{\mathbf{t}}_i\|\|\mathbf{R}\tilde{\mathbf{t}}_j\| + \|\mathbf{D}-\mathbf{I}\|\|\tilde{\mathbf{t}}_i\|\|\tilde{\mathbf{t}}_j\| \leq k^2\|\mathbf{H}\| + \|\mathbf{D}-\mathbf{I}\|.$$

Now, observe that by Theorem 8.3

$$\|\mathbf{Q}\tilde{\mathbf{t}}_j\| = \|\boldsymbol{\psi}_j - \mathbf{t}_j\| \leq \|\boldsymbol{\psi}_j\| + \|\mathbf{t}_j\| \leq 1 + k^2\|\mathbf{F}\|.$$

Then, dropping higher order terms in ϵ_{mach},

$$\|\tilde{\mathbf{t}}_i\|\|\mathbf{Q}\tilde{\mathbf{t}}_j\| \leq k^2\|\mathbf{F}\| \quad \text{and} \quad \|\tilde{\mathbf{t}}_j\|\|\mathbf{Q}\tilde{\mathbf{t}}_i\| \leq k^2\|\mathbf{F}\|.$$

The result follows by combining the above expressions. $\square$

Remark 8.7. In fact, when Lanczos is run for k iterations, Theorem 8.6 holds for $i = 0, 1, \ldots, 2k-1$. However, in the context of this thesis, the additional work required to prove this more general statement is not warranted. See [Kni96] for details. $\triangle$

8.5 Backwards stability of the Lanczos algorithm

The Lanczos method is clearly far from forward stable in the sense that the $\mathbf{Q}$ and $\mathbf{T}$ output are far from what would be produced in exact arithmetic. The work of Greenbaum [Gre89] shows that the matrix $\mathbf{T}$ in a perturbed Lanczos recurrence of the form (8.1) can be viewed as the output of the Lanczos algorithm run in exact arithmetic on a certain "nearby" problem, provided the conclusions of Theorem 8.1 are satisfied; i.e. it shows that Lanczos is backwards stable. In particular, [Gre89] shows that if these conditions are satisfied, there exists a $N \times N$ matrix $\bar{\mathbf{A}}$ and vector $\bar{\mathbf{v}}$ such that Lanczos run on $\bar{\mathbf{A}}$, $\bar{\mathbf{v}}$ in exact arithmetic for k steps produces $\mathbf{T}$ (i.e., in the notation from Chapter 3, that $\Psi_{\mathbf{T},\mathbf{e}_0} = [\Psi_{\bar{\mathbf{A}},\bar{\mathbf{v}}}]_{2k-1}^{\text{gq}}$) and , that

(i) Eigenvalues of $\bar{\mathbf{A}}$ are clustered near to those of $\mathbf{A}$: for any $j \in 0, 1, \ldots, N-1$, there exists $i \in 0, 1, \ldots, n-1$ such that

$$\lambda_j(\bar{\mathbf{A}}) \approx \lambda_i(\mathbf{A}).$$

(ii) The sum of squares of first components of eigenvectors corresponding to eigenvalues or clusters of eigenvalues of $\bar{\mathbf{A}}$ approximately equal tot he squares of the projections of $\mathbf{v}$ onto the eigenvectors of $\mathbf{A}$: for an eigenvalue $\lambda_i(\mathbf{A})$

$$w_i \approx \sum_{j \in S} \bar{w}_j$$

where S_i is the set of indices such that $\lambda_j(\bar{\mathbf{A}}) \approx \lambda_i(\mathbf{A})$ for all $j \in S$.

Together, these conditions imply that

$$\Psi_{\mathbf{A},\mathbf{v}} \approx \Psi_{\bar{\mathbf{A}},\bar{\mathbf{v}}}. \tag{8.2}$$

8.5.1 A new approach

While the analysis of [Gre89] provides a backwards stability result, the proofs are highly technical. We now show how the result of [Kni96] implies a simple backwards stability result.

Towards this end, denote by $\{\sigma_m\}_{m=0}^{\infty}$ and $\{\rho_m\}_{m=0}^{\infty}$ the Chebyshev moments of $\Psi_{\mathbf{A},\mathbf{v}}$ and $\Psi_{\mathbf{T},\mathbf{e}_0}$ respectively; i.e.

$$\sigma_m = \int T_m \, \mathrm{d}\Psi_{\mathbf{A},\mathbf{v}} \quad \text{and} \quad \rho_m = \int T_m \, \mathrm{d}\Psi_{\mathbf{T},\mathbf{e}_0}.$$

Now define the distribution function

$$\widehat{\Psi} = \Psi_{\mathbf{A},\mathbf{v}} + \Xi, \quad \text{where} \quad \frac{\mathrm{d}\Xi}{\mathrm{d}x} = \frac{\mathrm{d}\mu_{-1,1}^{T}}{\mathrm{d}x} \sum_{m=0}^{2k-1} (\rho_m - \sigma_m) T_m.$$

By construction, for $m = 1, 2, \ldots, 2k-1$,

$$\begin{aligned} \int_{-1}^{1} T_m \widehat{\Psi} &= \int_{-1}^{1} T_m \, \mathrm{d}\Psi_{\mathbf{A},\mathbf{v}} + \int T_m \, \mathrm{d}\Xi \\ &= \sigma_m + \sum_{\ell=0}^{2k-1} (\rho_\ell - \sigma_\ell) \int T_m T_\ell \, \mathrm{d}\mu_{-1,1}^{T} \\ &= \sigma_m + (\rho_m - \sigma_m). \end{aligned}$$

Thus, the modified moments $\{\hat{\sigma}_m\}_{m=0}^{\infty}$ of $\widehat{\Psi}$ satisfy

$$\hat{\sigma}_m = \int_{-1}^{1} T_m \, \mathrm{d}\widehat{\Psi} = \begin{cases} \rho_m & m = 0, 1, \ldots, 2k-1 \\ \sigma_m & m = 2k, 2k+1, \ldots \end{cases}.$$

Now, observe that

$$d_{\mathrm{W}}(\Psi, \widehat{\Psi}) = \int_{-1}^{1} |\Xi(x)| \, \mathrm{d}x \leq \sum_{m=0}^{2k-1} |\rho_m - \sigma_m| \int_{-1}^{1} \left| \int_{-1}^{x} T_m \, \mathrm{d}\mu_{-1,1}^{T} \right| \mathrm{d}x.$$

The Chebyshev polynomials are bounded by one on $[-1, 1]$, so

$$\left| \int_{-1}^{x} T_m \, \mathrm{d}\mu_{-1,1}^{T} \right| \leq \int_{-1}^{x} |T_m \, \mathrm{d}\mu_{-1,1}^{T}| \leq \int_{-1}^{1} \mathrm{d}\mu_{-1,1}^{T} = 1.$$

Thus, assuming $|\rho_m - \sigma_m| \leq \epsilon(k)$,

$$d_{\mathrm{W}}(\Psi, \widehat{\Psi}) \leq \sum_{m=0}^{2k-1} 2|\rho_m - \sigma_m| \leq 4k\epsilon(k).$$

The distribution function $\widehat{\Psi}$ is near to $\Psi_{\mathbf{A},\mathbf{v}}$ in the sense of Wasserstein distance, and $\mathbf{T}_k$ is produced when the Stieltjes algorithm is applied. However, there are two shortcomings. First, the support of $\widehat{\Psi}$ is all of $[-1, 1]$ as Ξ is absolutely continuous on $(-1, 1)$. Second Ξ is not necessarily increasing, and so $\widehat{\Psi}$ is not necessarily increasing.

8.6 CIF bounds Finite precision

While the tridiagonal matrix $\mathbf{T}$ and the matrix $\mathbf{Q}$ of Lanczos vectors produced in finite precision arithmetic may be very different from those produced in exact arithmetic, we now show that our error bounds, based on the $\mathbf{T}$ and $\mathbf{Q}$ actually produced by the finite precision computation, still hold to a close approximation. First, we argue that Lemma 7.2 holds to a close degree provided $\mathbf{F}$ is not too large. Towards this end, note that we have the shifted perturbed recurrence,

$$(\mathbf{A} - z\mathbf{I})\mathbf{Q} = \mathbf{Q}(\mathbf{T} - z\mathbf{I}) + \beta_{k-1}\mathbf{q}_k\mathbf{e}_{k-1}^{\mathsf{H}} + \mathbf{F}. \tag{8.3}$$

From (8.3), it is then clear that,

$$(\mathbf{A} - z\mathbf{I})\mathbf{Q}(\mathbf{T} - z\mathbf{I})^{-1}\mathbf{e}_0 = \mathbf{Q}\mathbf{e}_1 + \beta_{k-1}\mathbf{q}_k\mathbf{e}_{k-1}^{\mathsf{H}}(\mathbf{T} - z\mathbf{I})^{-1}\mathbf{e}_0 + \mathbf{F}(\mathbf{T} - z\mathbf{I})^{-1}\mathbf{e}_0.$$

This implies that Corollary 7.4 also holds closely. More specifically,

$$\begin{aligned}\mathsf{res}_k(z) &= \det(h_{w,z}(\mathbf{T}))\mathsf{res}_k(w) + \mathbf{f}_k(w,z)\\ \mathsf{err}_k(z) &= \det(h_{w,z}(\mathbf{T}))h_{w,z}(\mathbf{A})\mathsf{err}_k(w) + (\mathbf{A} - z\mathbf{I})^{-1}\mathbf{f}_k(w,z)\end{aligned}$$

where

$$\mathbf{f}_k(w,z) := \mathbf{F}\left((\mathbf{T} - z\mathbf{I})^{-1} - \det(h_{w,z}(\mathbf{T}))(\mathbf{T} - w\mathbf{I})^{-1}\right)\mathbf{e}_0.$$

Using this we have,

$$f(\mathbf{A})\mathbf{v} - \mathsf{lan\text{-}FA}_k(f) = -\frac{1}{2\pi\boldsymbol{i}}\oint_\Gamma f(z)\mathsf{err}_k(z)\mathrm{d}z - \frac{1}{2\pi\boldsymbol{i}}\oint_\Gamma f(z)(\mathbf{A} - z\mathbf{I})^{-1}\mathbf{f}_k(w,z)\,\mathrm{d}z$$

which we may bound using the triangle inequality as

$$\|f(\mathbf{A})\mathbf{v} - \mathsf{lan\text{-}FA}_k(f)\| \le \frac{1}{2\pi}\left\|\oint_\Gamma f(z)\mathsf{err}_k(z)\,\mathrm{d}z\right\| + \frac{1}{2\pi}\left\|\oint_\Gamma f(z)(\mathbf{A} - z\mathbf{I})^{-1}\mathbf{f}_k(w,z)\,\mathrm{d}z\right\|.$$

This expression differs from Theorem 7.6 only by the presence of the term involving $\mathbf{f}_k(w,z)$ (and, of course, by the fact that $\mathsf{err}_k(z)$ now denotes the error in the finite precision computation). If we take $\|\cdot\|$ as the $(\mathbf{A} - w\mathbf{I})^2$-norm, then this additional term can be bounded by using that

$$\begin{aligned}\left\|\oint_\Gamma f(z)(\mathbf{A} - z\mathbf{I})^{-1}\mathbf{f}_k(w,z)\,\mathrm{d}z\right\| &\le \oint_\Gamma |f(z)|\|(\mathbf{A} - w\mathbf{I})(\mathbf{A} - z\mathbf{I})^{-1}\|_2\|\mathbf{f}_k(w,z)\|_2|\mathrm{d}z|\\ &\le \oint_\Gamma |f(z)|\|h_{w,z}\|_{S_0}\|\mathbf{f}_k(w,z)\|_2|\mathrm{d}z|.\end{aligned} \tag{8.4}$$

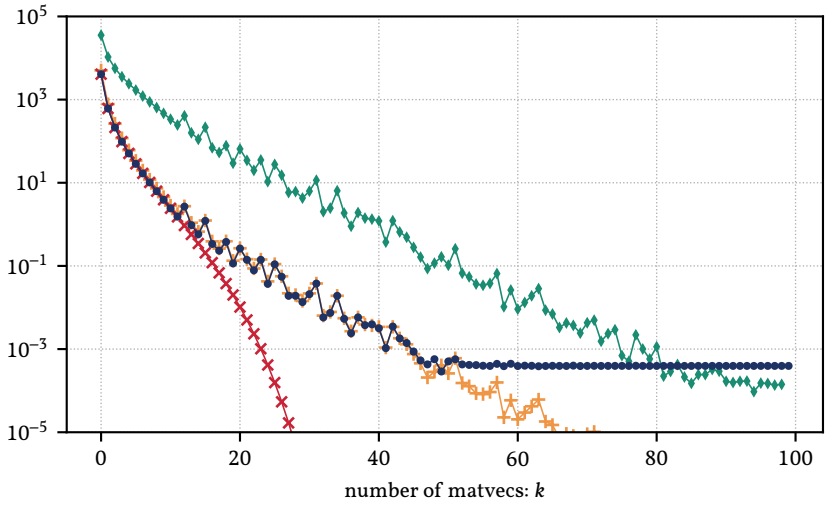


Figure 8.1: $\mathbf{A}^2$-norm error bounds for $f(x) = \sqrt{x}$ where $\mathbf{A}$ has $n = 50$ eigenvalues spaced according to the model problem with $\rho = 0.8$ and $\kappa = 10^3$. We take Γ as a circular contour of radius $\lambda_{\max}$ centered at $\lambda_{\max}$ and $w = 0$. Computations without reorthogonalization are run in single precision arithmetic, and error bounds are computed using Theorem 7.6 using the finite precision quantities. *Legend*: Lanczos-FA error with (—×—) and without (—•—) reorthogonalization, a priori bounds with $S = S_i = \mathcal{I}$ (—•—), and a posteriori bounds obtained by using Theorem 7.6 with $S = \mathcal{I}$ (—+—). *Takeaway*: The bounds in finite precision arithmetic are accurate until near the ultimately attainable accuracy.

Note that $1/(2\pi)$ times (8.4) can be viewed as an upper bound of the ultimate obtainable accuracy of Lanczos-FA in finite precision after convergence. If the inequalities do not introduce too much slack, this upper bound will also produce a reasonable estimate. Since $\|\mathbf{F}\|$ is small, one may simply ignore the contribution of (8.4), at least provided the Lanczos-FA error is not near the final accuracy. Finally, we have worked in the $(\mathbf{A} - w\mathbf{I})^2$ norm as it simplifies some of the analysis, but in principle, a similar approach could be used with other norms. This is straightforward, but would involve bounding something other than $\|h_{w,z}\|_{S_0}$.

8.6.1 Numerical experiment

To illustrate the point of above analysis, we use a setup similar to what was used to produce Figure 7.3. However, we now use the model problem and run Lanczos without reorthgonalization. We use the $\mathbf{T}$ produced in finite precision arithmetic in our computation of the error bounds from Theorem 7.6 and report the results in Figure 8.1. Note that we use Theorem 7.6 and therefore do not account for the roundoff term analyzed above. However, since this term can be expected to be on the order of machine precision, the absence of this term in our computed bound does not significantly impact the bounds until near the ultimately attainable accuracy of Lanczos-FA. In other words, Theorem 7.6 still holds until the Lanczos-FA error is small.

Chapter 9
Outlook

So, what's next for algorithms for computing expressions involving matrix functions? While this is far too vague a question to provide any sort of definitive answer, I will gladly discuss several directions which I hope will be pursued further in the near future. These topics are simply a collection of directions for future work I personally find interesting, and they should not be viewed as any sort of statement regarding the direction the field as a whole should move in. Indeed, many important topics, such as algorithms for high performance computing and the use of mixed precision arithmetic [Abd+21] are not discussed.

9.1 Randomization

It is now widely recognized that randomization is an extremely powerful algorithmic tool in numerical linear algebra [HMT11; MT20]. While a number of topics have "matured", the use of randomization in Krylov subspace methods and related algorithms remains ripe for further study.

A big question is how to compute a low-rank approximation to $f(\mathbf{A})$, given access to products with $\mathbf{A}$. Some progress on this question has been made, primarily with the end goal of estimating the spectral sum $\operatorname{tr}(f(\mathbf{A}))$ [Lin16; GSO17; SAI17; LZ21; CH22], but a general theoretical understanding of randomized Krylov subspace methods for approximating $f(\mathbf{A})$ is an open problem. A natural starting point is the analysis of block Krylov subspace methods applied to a set of random vectors [MM15; MT20; Tro21]. However, block Lanczos methods are

perhaps even more susceptible to the effects of finite precision arithmetic than the standard Lanczos methods, and not much is known theoretically about their behavior in finite precision arithmetic.

Another interesting question is how randomization can be used to speed up computations of $f(\mathbf{A})\mathbf{v}$.

For overdetermined linear systems $\mathbf{A}\mathbf{x} = \mathbf{v}$, methods such as the randomized Kaczmarz algorithm [SV08; NSW14], accelerated coordinate descent [LS13a; All+16], and stochastic heavy ball momentum [BCW22] can all outperform applying CG to the normal equations $\mathbf{A}^{\mathsf{H}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathsf{H}}\mathbf{v}$. In fact, for positive definite systems, accelerated coordinate descent methods can outperform CG applied directly to the system of interest.

A natural way to extend these fast linear system solvers to matrix functions is by applying them to a proxy rational function whose individual terms are each positive definite linear systems. This technique was used in [JS19] to obtain a fast algorithm for approximating products with the matrix sign function and related quantities. However, this approach treats each term in the proxy rational function as independent, despite the fact that there is significant shared structure. From a theoretical perspective this is acceptable as long as the number of terms in the proxy rational function is logarithmic in the accuracy tolerance, which is typically the case [GT19]. However, this is likely somewhat wasteful in practice, so it would be worthwhile to study how such ideas can be implemented more efficiently.

9.2 Typicality

Recall that *typicality*, discussed in Section 4.1.1, is essentially the physics version of concentration of quadratic trace estimators. I find typicality fascinating for a number of reasons. First, typicality provides a physical meaning to quadratic trace estimators, which have become one of the most widely studied methods in randomized numerical linear algebra. Second, the literature on typicality has a rich history, with the earliest works dating back nearly a century. This not only means the popular opinion on typicality has evolved, but it makes typicality an interesting case study in the fragmentation of knowledge between

disciplines. While several review papers have been published recently in the physics literature [Gol+10; Jin+21], I believe a review from the perspective of numerical linear algebra would yield many interesting historical insights. Indeed, applied mathematicians have seemingly overlooked several important lines of literature on this topic.

9.3 Accessibility to non-experts

As mentioned in the introduction, it is my sense that practitioner knowledge of Lanczos based methods for matrix functions is limited by the lack of resources providing easy to understand background for such methods. While I hope that this thesis provides a more accessible introduction to the topic, by nature, a thesis emphasizes the author's own work and only touches on the important work of others. A more balanced treatment of methods for matrix functions, with a treatment of methods for non-symmetric problems as well as a further emphasis on the important case of linear systems would be of general interest.

Separately, I hope that easy-to-use black-box versions of some of the algorithms studied in this thesis are eventually implemented. A natural starting point would be implementing the integral based bounds from Chapter 7 in such a way that they could be easily integrate into existing codes. In order for such a tool to be truly black-box would require additional study into how to choose parameters such as the contour of integration. However, even if some user input is required, such a tool would help ensure more efficient resource allocation.

Chapter 10
Notation and other reference sheets

10.1 Basic notation

Here we provide a reference for some common notation. The page number is the first page on which the notation is used. In many cases, some of the parameters will be suppressed for notational convince. for instance, while the i-th eigenvalue of a matrix $\mathbf{B}$ is denoted $\lambda_i(\mathbf{B})$, we will often write λ_i for the i-th eigenvalue of $\mathbf{A}$

notation	description	page
$\mathbf{A}$	$n \times n$ Hermitian matrix	1
λ_i, $\lambda_i(\mathbf{A})$	i-th eigenvalue of $\mathbf{A}$	1
Λ, $\Lambda(\mathbf{A})$	set of eigenvalues of $\mathbf{A}$	1
$\mathcal{I}, \mathcal{I}(\mathbf{A})$	smallest interval containing $\Lambda(\mathbf{A}$	6
$f(\mathbf{A})$	matrix function	1
$\mathrm{tr}(f(\mathbf{A}))$	spectral sum	2
Φ, $\Phi_{\mathbf{A}}$	cumulative empirical spectral measure (CESM)	2
Ψ, $\Psi_{\mathbf{A},\mathbf{v}}$	weighted CESM	28
K_k, $K_k(\mathbf{A}, \mathbf{v})$	dimension k Krylov subspace	3
$\mathbf{Q}, \mathbf{T}$	Lanczos vectors and coefficients after k iterations	4

continues on next page

notation	description	page
$\widehat{\mathbf{Q}}, \widehat{\mathbf{T}}$	Lanczos vectors and coefficients after completion	71
$\mathbb{1}$	indicator function	1
$\|g\|_S$	supremum of $g : \mathbb{C} \to \mathbb{C}$ on $S \subset \mathbb{C}$	5
$\|\cdot\|$	norm induced by matrix with same eigenvectors as $\mathbf{A}$	5
μ	non-negative unit-mass distribution function	15
$\langle \cdot, \cdot \rangle_\mu$	inner product induced by μ	15
p_i	degree i orthogonal polynomial of μ	15
$\theta_j^{(s+1)}$	j-th zero of p_s	17
$\mathbf{M}, \mathbf{M}(v)$	Jacobi matrix for μ, v	16
$\mu_{a,b}^T$	Chebyshev distribution function on $[a, b]$	18
m_i	degree i modified moment with respect to μ	16
$[f]_s^{\text{ap}}$	degree s projection of f in $\langle \cdot, \cdot \rangle_\mu$	18
$[f]_s^{\text{ip}}$	degree s interpolation of f at zeros of p_s	18
$[f]_s^{\text{d-ap}}$	degree s damped projection of f in $\langle \cdot, \cdot, \rangle_\mu$	18
$[f]_s^{\text{d-ip}}$	degree s damped interpolation of f at zeros of p_s	18
$\{\rho_i\}_{i=0}^s$	damping coefficients	19
$\{\rho_i^J\}_{i=0}^s$	Jackson's damping coefficients	21
$\mathbf{C}_{\mu \to v}$	connection coefficient matrix	31
$\langle \cdot \rangle$	average over $\ell = 0, \ldots, n_{\text{v}} - 1$	49
$\mathbb{S}^{n-1}$	unit hypersphere on $\mathbb{C}^n$	55
cg_k	CG iterate at step k	72
mr_k	MINRES iterate at step k	73
qmr_k	QMR iterate at step k	73
$\text{lan-OR}_k(r, R)$	Lanczos-OR iterate	74
$\text{lan-FA}_k(f)$	Lanczos-FA iterate	92
sign – OR	Lanczos-OR induced iterate to matrix sign	95
$\text{err}_k(z)$	Lanczos-FA error at step k for for $\mathbf{A}^{-1}\mathbf{v}$	107
$\text{res}_k(z)$	Lanczos-FA residual at step k for for $\mathbf{A}^{-1}\mathbf{v}$	107
$h_{w,z}$	$h_{w,z} = (x - w)/(x - z)$	108
h_z	$h_z = 1/(x - z)$	108

continues on next page

notation	description	page
$S, S_0, \ldots, S_{k-1}$	$\Lambda \subset S, \lambda_i(\mathbf{T}) \subset S_i$	109

10.2 Indexing for matrices

Given a matrix $\mathbf{B}$, we use $[\mathbf{B}]_{r:r',c:c'}$ to denote the submatrix matrix of $\mathbf{B}$ consisting of rows r up to (but not including) row r' and columns c up to (but not including) c'. Thus, the dimension of $[\mathbf{B}]_{r:r',c:c'}$ is $(r'-r)\times(c'-c)$. *Indexing of matrices starts at zero.* If any of these indices are equal to 0 or the corresponding max dimension of $\mathbf{B}$, they may be omitted. If $r' = r+1$ or $c' = c+1$, then we will simply write r or c.

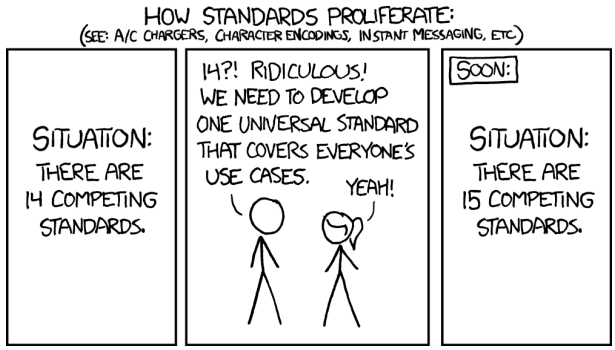
As an example, suppose

$$\mathbf{B} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

Then

$$[\mathbf{B}]_{:,:2} = \begin{bmatrix} 1 & 2 \\ 5 & 6 \\ 9 & 10 \end{bmatrix}, \quad [\mathbf{B}]_{1,:} = \begin{bmatrix} 5 & 6 & 7 & 8 \end{bmatrix}, \quad \text{and} \quad [\mathbf{B}]_{0,3} = 4.$$

I am sure many readers are wondering why I would use such a notation. Perhaps some are even thinking of xkcd number 927, Standards.



While I do not expect this notation to become standard in linear algebra, it was chosen intentionally after much consideration. The two primary motivations are as follows:

- Much of this thesis relies on the theory of orthogonal polynomials, and it is natural for orthogonal polynomials to be indexed by their degree (which starts at zero). It then makes sense that matrices involving orthogonal polynomials are indexed in a way which matches the polynomials.
- This thesis is written to be as accessible as possible to practitioners, particularly those in physics and data science. Zero-indexed programming languages are more common than one-indexed languages in these fields. In such languages, non-inclusive endpoints are idiomatic, so that the number of objects in a range is equal to the difference of the endpoints.[1] In fact, our notation is identical to that of Python/NumPy, the language of choice in many disciplines from these fields.

In my opinion, it would have been nice if sums were also indexed in a similar way. However, using a notation like $\sum_{r\le i<r'}$ was deemed too verbose, and modifying the standard notation $\sum_{i=r}^{r'-1}$ would have caused too much confusion.

10.3 The model problem

The model problem [Str91; SG92] is a standard class of problems used in the analysis of the finite precision behavior of Lanczos based algorithms, especially in the context of solving linear systems of equations. This is because the exponential spacing of the eigenvalues is favorable to Lanczos based linear system solvers in exact arithmetic yet simultaneously causes the Lanczos algorithm to rapidly lose orthogonality in finite precision arithmetic. The model problem is parameterized by the dimension n, the condition number κ, and a parameter ρ controlling the rate of growth of eigenvalues. Specifically,

$$\lambda_0 = 1, \quad \lambda_{n-1} = \kappa, \quad \lambda_i = \lambda_1 + \left(\frac{i}{n-1}\right)\cdot(\kappa-1)\cdot\rho^{n-i-1}, \qquad i = 1,\dots,n-1. \tag{10.1}$$

[1] The reason many languages use such conventions is perhaps due in part to Dijkstra's 1982 letter, *Why numbering should start at zero*, which advocates for indexing to start at zero and for ranges to include the start point but not the end point.

10.4 Some basic properties

Lemma 10.1. *Let $\|\cdot\|$ be a norm induced by a positive definite matrix with the same eigenvectors as* $\mathbf{A}$*. Then*

$$\|g(\mathbf{A})\mathbf{v}\| \leq \|g(\mathbf{A})\|_2\|\mathbf{v}\|.$$

Proof. Let $\mathbf{B}^2$ be the matrix inducing $\|\cdot\|$. By assumption $\mathbf{A}$ and $\mathbf{B}$ commute, so

$$\|g(\mathbf{A})\mathbf{v}\| = \|\mathbf{B}g(\mathbf{A})\mathbf{v}\|_2 = \|g(\mathbf{A})\mathbf{B}\mathbf{v}\|_2 \leq \|g(\mathbf{A})\|_2\|\mathbf{B}\mathbf{v}\|_2 \leq \|g(\mathbf{A})\|_2\|\mathbf{v}\|. \qquad \square$$

We now provide a number of useful facts about powers of tridiagonal matrices. To simplify our proofs, we recall the following fact.

Lemma 10.2. *For any $q > 0$ and $k_0, k_q = 0, 1, \ldots, n-1$,*

$$[\mathbf{A}^q]_{k_0,k_q} = \sum_{k_1=0}^{n-1}\sum_{k_2=0}^{n-1}\cdots\sum_{k_{q-1}=0}^{n-1}[\mathbf{A}]_{k_0,k_1}[\mathbf{A}]_{k_1,k_2}\cdots[\mathbf{A}]_{k_{q-1},k_q}.$$

Proof. This is the definition of matrix multiplication applied q times. $\square$

Note that if $\mathbf{T}$ is tridiagonal, then $[\mathbf{T}]_{k_\ell,k_{\ell+1}} = 0$ whenever $|k_\ell - k_{\ell+1}| > 1$. Thus, the product

$$[\mathbf{T}]_{k_0,k_1}[\mathbf{T}]_{k_1,k_2}\cdots[\mathbf{T}]_{k_{q-1},k_q} \tag{10.2}$$

is nonzero, if and only if $|k_\ell - k_{\ell+1}| \leq 1$ for all ℓ. Thus, assuming (10.2) is nonzero, we can view $\{k_\ell\}$ as a walk on $\{0, 1, \ldots, n-1\}$, starting from k_0 and ending at k_q, where, at each iteration ℓ, we stay put or move to an adjacent index. Clearly

$$|k_0 - k_1| + |k_1 - k_2| + \cdots + |k_{q-1} - k_q| \leq q.$$

In other words, the total distance moved during the walk is at most q.

Using this perspective, we immediately find that powers of tridiagonal matrices are banded.

Corollary 10.3. *Suppose* $\mathbf{T}$ *is a tridiagonal matrix and $q \geq 0$ an integer. Then, for all i, j with $|i - j| > q$,*

$$[\mathbf{T}^q]_{i,j} = 0.$$

Proof. Consider a nonzero term (10.2). If $|i-j| > q$, then it is not possible to move from i to j in q steps. □

More generally, we find that the entries of powers of a tridiagonal matrix depend only on nearby entries of the base tridiagonal matrix. We consider the symmetric case for simplicity.

Corollary 10.4. *Suppose* $\mathbf{T}$ *is a symmetric tridiagonal matrix and* $q \geq 0$ *an integer. Then, for any* i, j *with* $|j - i| \leq q$,

$$[\mathbf{T}^q]_{i,j}$$

is determined entirely by $[\mathbf{T}]_{:k,:k}$, *where* $k = \max(i, j) + \lfloor (q - |j - i|)/2 \rfloor$. *In fact, if* $q - |j - i|$ *is even, then there is no dependence on* $[\mathbf{T}]_{k-1,k-1}$.

Proof. Without loss of generality, we may assume $i \leq j$.

Consider a nonzero term (10.2). We require at least $j - i$ of our allocated q movements to move from i to j. Since we must end at j, we could move past j at most $\lfloor (q - (j - i))/2 \rfloor$ indices before returning.

If $(q - (j - i))/2$ is an integer, then when we reach the maximum point $j + (q - (j - i))/2$ we must immediately return towards j. Thus, $k_\ell \neq k_{\ell+1}$ for any ℓ, and in particular, for $\ell = j + (q - (j - i))/2$. □

10.5 List of algorithms

name	description	reference
LANCZOS	Lanczos algorithm	Algorithm 1.1
STIELTJES	Stieltjes algorithm (naive)	Algorithm 2.1
STIELTJES	Stieltjes algorithm	Algorithm 2.2
GET-MOMENTS	Get modified moments of Ψ wrt. μ	Algorithm 3.1
GET-CHEBYSHEV-MOMENTS	Get modified moments of Ψ wrt. $\mu^T_{a,b}$	Algorithm 3.2
GET-CONNECTION-COEFFS	Get connection coefficients	Algorithm 3.3
GET-MOMENTS-FROM-CHEB	Get modified moments wrt. μ of weighed CESM (via Chebyshev moments)	Algorithm 3.4
GET-MOMENTS-FROM-LANCZOS	Get modified moments wrt. μ of weighed CESM (via Lanczos)	Algorithm 3.5
GET-IQ	Quadrature by interpolation	Algorithm 3.6
GET-GQ	Gaussian quadrature	Algorithm 3.7
GET-AQ	Quadrature by approximation	Algorithm 3.8
GET-AAQ	Approximate quadrature by approximation	Algorithm 3.9
SPEC-APPROX	Prototypical randomized spectrum and spectral sum approximation	Algorithm 4.1
LDL	LDL factorization	Algorithm 5.1
STREAMING-LDL	Streaming LDL factorization	Algorithm 5.2
STREAMING-BANDED-PROD	Streaming banded product	Algorithm 5.3
STREAMING-BANDED-INV	Streaming banded inverse	Algorithm 5.4
STREAMING-TRIDIAG-SQUARE	Streaming tridiagonal square	Algorithm 5.5
GET-POLY	Get polynomial of tridiagonal matrix	Algorithm 5.6
BANDED-RATIONAL	Streaming banded rational inverse	Algorithm 5.7
LANCZOS-OR-LM	Lanczos-OR (low memory)	Algorithm 5.8

10.6 List of figures

1.1 CG convergence (with and without reorthgonalization) and error bounds. 8

2.1 Sample unit mass distribution functions. 13

3.1 Errors for approximating $\int f \, d\Psi = \mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ when $f(x) = 1/(1 + 16x^2)$ for a spectrum uniformly filling $[-1, 1]$. 45

3.2 Errors for approximating $\int f \, d\Psi = \mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ when $f(x) = 1/(1 + 16x^2)$ for a spectrum uniformly filling $[-1, 1]$ except for a gap around zero. 46

3.3 Errors for approximating $\int f \, d\Psi = \mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ when $f(x) = 1/x$ for model problem. 47

3.4 Errors for approximating $\int f \, d\Psi = \mathbf{v}^{\mathsf{H}} f(\mathbf{A})\mathbf{v}$ when $f(x) = \mathbb{1}[x > c]$ for MNIST covariance matrix. 48

4.1 CESM Φ and independent 10 samples of weighted CESM Ψ 50

4.2 Approximations to a sparse spectrum with just 12 eigenvalues. . . 59

4.3 Approximations to a "smooth" spectrum using quadrature by approximation with various choices of μ. 62

4.4 Approximations to a "smooth" spectrum using smoothed Gaussian quadrature for various smoothing parameters σ. 64

4.5 Approximations to a "smooth" spectrum with a spike using quadrature by approximation with various choices of μ. 66

4.6 Heat capacity as a function of temperature for a small spin system. 68

5.1 Error estimates for Lanczos-OR for $r(x) = 1/(x^2 + 1)$ and $R(x) = 1$. 80

5.2 Access patterns for inputs to streaming functions used in low-memory implementations of Lanczos-OR and Lanczos-FA. Indices indicate what information should be streamed into the algorithm at the given iteration. 83

6.1 Comparison of Lanczos-based spectrum approximation algorithms. 100

6.2 Optimality ratio for $\mathbf{A}^2$-norm errors for approximating $\operatorname{sign}(\mathbf{A})\mathbf{v}$. . 101

6.3 $\mathbf{A}^2$-norm error in Lanczos-OR-lm based rational approximation to matrix sign function. 102

6.4 Comparison of $(\mathbf{A}^2 + c\mathbf{I})$-norm errors for CG and Lanczos-FA for computing $(\mathbf{A}^2 + c\mathbf{I})^{-1}\mathbf{v}$ with $c = 0.05$. 104

7.1 Contour plot of $\|h_{w,z}\|_I / |\det(h_{w,z}(\mathbf{T}))|^{1/k}$ as a function of $z \in \mathbb{C}$ 116

7.2 Circle, Pac-Man and double circle contours 120

7.3 $\mathbf{A}$-norm error bounds for $f(x) = \sqrt{x}$ where $\mathbf{A}$ has $n = 1000$ eigenvalues spaced uniformly in $[10^{-2}, 10^2]$ and Γ is a circular contour (left) or Pac-Man contour (right). 121

7.4 $(\mathbf{A} - w\mathbf{I})^2$-norm error bounds for $f(x) = \text{step}(x-a)/x$ where $\mathbf{A}$ is a random matrix whose limiting density is supported on $[a_1, b_1] \cup [a_2, b_2]$, $a = (b_1 + a_2)/2$, and Γ is a double circle contour. 124

7.5 Quadratic form error bounds for $f(x) = \text{step}(x - a)$ where $\mathbf{A}$ is a random matrix whose limiting density is supported on $[a_1, b_1] \cup [a_2, b_2]$, $a = (b_1 + a_2)/2$, and Γ is a double circle contour. 125

8.1 $\mathbf{A}^2$-norm error bounds for $f(x) = \sqrt{x}$ where $\mathbf{A}$ has $n = 50$ eigenvalues spaced according to the model problem with $\rho = 0.8$ and $\kappa = 10^3$ 141

Chapter 11
Bibliography

[Abd+21] A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, M. Gates, N. J. Higham, X. S. Li, J. Loe, P. Luszczek, S. Pranesh, S. Rajamanickam, T. Ribizel, B. F. Smith, K. Swirydowicz, S. Thomas, S. Tomov, Y. M. Tsai, and U. M. Yang. "A survey of numerical linear algebra methods utilizing mixed-precision arithmetic". In: *The International Journal of High Performance Computing Applications* 35.4 (Mar. 2021), pp. 344–369. DOI: 10.1177/10943420211003313 (cited on page 143).

[Ach92] N. I. Achieser. *Theory of approximation*. Dover Publications, 1992. ISBN: 9780486671291 (cited on page 20).

[Ada+18] R. P. Adams, J. Pennington, M. J. Johnson, J. Smith, Y. Ovadia, B. Patton, and J. Saunderson. *Estimating the Spectral Density of Large Implicit Matrices*. 2018. arXiv: 1802.03451 [stat.ML] (cited on pages 54, 60).

[Aic+03] M. Aichhorn, M. Daghofer, H. G. Evertz, and W. von der Linden. "Low-temperature Lanczos method for strongly correlated systems". In: *Physical Review B* 67.16 (Apr. 2003). DOI: 10.1103/physrevb.67.161103 (cited on pages 6, 130).

[AK65] N. I. Aheizer and N. Kemmer. *The classical moment problem and some related questions in analysis*. Oliver & Boyd Edinburgh, 1965 (cited on page 112).

[Alb+75] R. Alben, M. Blume, H. Krakauer, and L. Schwartz. "Exact results for a three-dimensional alloy with site diagonal disorder: comparison with the coherent potential approximation". In: *Physical Review B* 12.10 (Nov. 1975), pp. 4090–4094. DOI: 10.1103/physrevb.12.4090 (cited on page 53).

[All+16] Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan. "Even Faster Accelerated Coordinate Descent Using Non-Uniform Sampling". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 2016, pp. 1110–1119. arXiv: 1512.09103 [math.OC] (cited on page 144).

[AT11] H. Avron and S. Toledo. "Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix". In: *Journal of the ACM* 58.2 (Apr. 2011), pp. 1–34. DOI: 10.1145/1944345.1944349 (cited on pages 52, 53).

[Avr10] H. Avron. "Counting Triangles in Large Graphs using Randomized Matrix Trace Estimation". In: *Proceedings of KDD-LDMTA*. 2010 (cited on page 2).

[BB20] M. Benzi and P. Boito. "Matrix functions in network analysis". In: *GAMM-Mitteilungen* 43.3 (2020), e202000012. DOI: 10.1002/gamm.202000012. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/gamm.202000012 (cited on page 2).

[BCW22] R. Bollapragada, T. Chen, and R. Ward. *On the fast convergence of minibatch heavy ball momentum*. 2022. arXiv: 2206.07553 [cs.LG] (cited on page 144).

[BEG20] M. Berljafa, S. Elsworth, and S. Güttel. *A Rational Krylov Toolbox for MATLAB*. 2020 (cited on page 101).

[Ber+08] K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, K. Hill, J. Hiller, et al. "Exascale computing study: Technology challenges in achieving exascale systems". In: *Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep* 15 (2008) (cited on page 43).

[Ber07] C. Berg. *Stieltjes-Pick-Bernstein-Schoenberg and their connection to complete monotonicity*. 2007. eprint: http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.142.3872 (cited on page 112).

[BFG96] Z. Bai, G. Fahey, and G. Golub. "Some large-scale matrix computation problems". In: *Journal of Computational and Applied Mathematics* 74.1-2 (Nov. 1996), pp. 71–89. DOI: 10.1016/0377-0427(96)00018-0 (cited on page 52).

[BKM22] V. Braverman, A. Krishnan, and C. Musco. "Sublinear time spectral density estimation". In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. arXiv cs.DS 2104.03461. ACM,

June 2022. DOI: 10.1145/3519935.3520009. arXiv: 2104.03461 [cs.DS] (cited on pages 41, 52, 60).

[BNT21] P. D. Brubeck, Y. Nakatsukasa, and L. N. Trefethen. "Vandermonde with Arnoldi". In: *SIAM Review* 63.2 (Jan. 2021), pp. 405–415. DOI: 10.1137/19m130100x (cited on page 61).

[Bor00] A. Boriçi. "Fast Methods for Computing the Neuberger Operator". In: Springer Berlin Heidelberg, 2000, pp. 40–47. DOI: 10.1007/978-3-642-58333-9_4 (cited on page 94).

[Bor03] A. Boriçi. "Computational methods for UV-suppressed fermions". In: *Journal of Computational Physics* 189.2 (Aug. 2003), pp. 454–462. DOI: 10.1016/s0021-9991(03)00227-4 (cited on page 112).

[Bor99] A. Boriçi. "On the Neuberger overlap operator". In: *Physics Letters B* 453.1-2 (Apr. 1999), pp. 46–53. DOI: 10.1016/s0370-2693(99)00318-4 (cited on page 112).

[BP99] R. P. Barry and R. K. Pace. "Monte Carlo estimates of the log determinant of large sparse matrices". In: *Linear Algebra and its Applications* 289.1-3 (Mar. 1999), pp. 41–54. DOI: 10.1016/s0024-3795(97)10009-x (cited on page 2).

[BS22] K. Bergermann and M. Stoll. "Fast computation of matrix function-based centrality measures for layer-coupled multiplex networks". In: *Physical Review E* 105.3 (Mar. 2022). ISSN: 2470-0053. DOI: 10.1103/physreve.105.034305 (cited on page 2).

[BS98] Z. D. Bai and J. W. Silverstein. "No eigenvalues outside the support of the limiting spectral distribution of large-dimensional sample covariance matrices". In: *The Annals of Probability* 26.1 (Jan. 1998). DOI: 10.1214/aop/1022855421 (cited on page 61).

[Car20] E. C. Carson. "An adaptive *s*-step conjugate gradient algorithm with dynamic basis updating". In: *Applications of Mathematics* 65.2 (Feb. 2020), pp. 123–151. DOI: 10.21136/am.2020.0136-19 (cited on pages 34, 44).

[CC22] T. Chen and Y.-C. Cheng. "Numerical computation of the equilibrium-reduced density matrix for strongly coupled open quantum systems". In: *The Journal of Chemical Physics* 157.6 (Aug. 2022), p. 064106. DOI: 10.1063/5.0099761. arXiv: 2204.08147 [quant-ph] (cited on page 54).

[CD15] E. C. Carson and J. W. Demmel. "Accuracy of the *s*-Step Lanczos Method for the Symmetric Eigenproblem in Finite Precision". In: *SIAM Journal on Matrix Analysis and Applications* 36.2 (Jan. 2015), pp. 793–819. DOI: 10.1137/140990735 (cited on pages 34, 44).

[CD71] F. Cyrot-Lackmann and F. Ducastelle. “Binding Energies of Transition-Metal Atoms Adsorbed on a Transition Metal”. In: *Physical Review B* 4.8 (Oct. 1971), pp. 2406–2412. DOI: 10.1103/physrevb.4.2406 (cited on page 29).

[CG96] J. Cullum and A. Greenbaum. “Relations between Galerkin and Norm-Minimizing Iterative Methods for Solving Linear Systems”. In: *SIAM Journal on Matrix Analysis and Applications* 17.2 (1996), pp. 223–247. DOI: 10.1137/S0895479893246765. eprint: https://doi.org/10.1137/S0895479893246765 (cited on pages 123, 126).

[CH22] T. Chen and E. Hallman. *Krylov-aware stochastic trace estimation*. 2022. arXiv: 2205.01736 [math.NA] (cited on pages 54, 143).

[Che+22] T. Chen, A. Greenbaum, C. Musco, and C. Musco. “Error Bounds for Lanczos-Based Matrix Function Approximation”. In: *SIAM Journal on Matrix Analysis and Applications* 43.2 (May 2022), pp. 787–811. DOI: 10.1137/21m1427784. arXiv: 2106.09806 [math.NA] (cited on page 126).

[Che00] E. W. Cheney. *Introduction to approximation theory*. 2. ed., repr. AMS Chelsea Publ, 2000. ISBN: 9780821813744 (cited on page 20).

[CK21] A. Cortinovis and D. Kressner. “On Randomized Trace Estimates for Indefinite Matrices with an Application to Determinants”. In: *Foundations of Computational Mathematics* (July 2021). DOI: 10.1007/s10208-021-09525-9 (cited on page 52).

[Cor22] A. Cortinovis. “Fast deterministic and randomized algorithms for low-rank approximation, matrix functions, and trace estimation”. en. PhD thesis. Lausanne, EPFL, 2022. DOI: 10.5075/EPFL-THESIS-9967 (cited on page 9).

[Cra46] H. Cramér. *Mathematical methods of statistics*. Princeton landmarks in mathematics and physics. Princeton University Press, 1946. ISBN: 9780691005478 (cited on page 53).

[CTU21] T. Chen, T. Trogdon, and S. Ubaru. “Analysis of stochastic Lanczos quadrature for spectrum approximation”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 1728–1739. arXiv: 2105.06595 [cs.DS] (cited on page 52).

[CTU22] T. Chen, T. Trogdon, and S. Ubaru. *Randomized matrix-free quadrature for spectrum and spectral sum approximation*. 2022. arXiv: 2204.01941 [math.NA] (cited on page 55).

[Cyr67] F. Cyrot-Lackmann. "On the electronic structure of liquid transitional metals". In: *Advances in Physics* 16.63 (July 1967), pp. 393–400. DOI: 10.1080/00018736700101495 (cited on page 29).

[Cyr69] F. Cyrot-Lackmann. "On the calculation of surface tension in transition metals". In: *Surface Science* 15.3 (July 1969), pp. 535–548. DOI: 10.1016/0039-6028(69)90140-x (cited on page 29).

[DBB19] K. Dong, A. R. Benson, and D. Bindel. "Network Density of States". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, July 2019. DOI: 10.1145/3292500.3330891 (cited on page 2).

[DC70] F. Ducastelle and F. Cyrot-Lackmann. "Moments developments and their application to the electronic charge distribution of d bands". In: *Journal of Physics and Chemistry of Solids* 31.6 (June 1970), pp. 1295–1306. DOI: 10.1016/0022-3697(70)90134-4 (cited on page 29).

[DC71] F. Ducastelle and F. Cyrot-Lackmann. "Moments developments: II. Application to the crystalline structures and the stacking fault energies of transition metals". In: *Journal of Physics and Chemistry of Solids* 32.1 (Jan. 1971), pp. 285–301. DOI: 10.1016/s0022-3697(71)80031-8 (cited on page 29).

[DEG72] G. Dahlquist, S. C. Eisenstat, and G. H. Golub. "Bounds for the error of linear systems of equations using the theory of moments". In: *Journal of Mathematical Analysis and Applications* 37.1 (Jan. 1972), pp. 151–166. DOI: 10.1016/0022-247x(72)90264-8 (cited on page 3).

[DGK98] V. Druskin, A. Greenbaum, and L. Knizhnerman. "Using Nonorthogonal Lanczos Vectors in the Computation of Matrix Functions". In: *SIAM Journal on Scientific Computing* 19.1 (1998), pp. 38–54. DOI: 10.1137/S1064827596303661. eprint: https://doi.org/10.1137/S1064827596303661 (cited on pages 92, 112).

[DH11] T. A. Davis and Y. Hu. "The university of Florida sparse matrix collection". In: *ACM Transactions on Mathematical Software* 38.1 (Nov. 2011), pp. 1–25. DOI: 10.1145/2049662.2049663 (cited on page 48).

[DHL15] J. Dongarra, M. A. Heroux, and P. Luszczek. "High-performance conjugate-gradient benchmark: A new metric for ranking high-performance computing systems". In: *The International Journal of High Performance Computing Applications* 30.1 (Aug. 2015), pp. 3–10. DOI: 10.1177/1094342015593158 (cited on page 43).

[DK88] V. Druskin and L. Knizhnerman. "Spectral Differential-Difference Method for Numeric Solution of Three-Dimensional Nonstationary Problems of Electric Prospecting". In: *Physics of the Solid Earth* 24 (Jan. 1988), pp. 641–648 (cited on page 92).

[DK89] V. Druskin and L. Knizhnerman. "Two polynomial methods of calculating functions of symmetric matrices". In: *USSR Computational Mathematics and Mathematical Physics* 29.6 (Jan. 1989), pp. 112–121. DOI: 10.1016/s0041-5553(89)80020-5 (cited on page 92).

[DK91] V. L. Druskin and L. A. Knizhnerman. "Error Bounds in the Simple Lanczos Procedure for Computing Functions of Symmetric Matrices and Eigenvalues". In: *Comput. Math. Math. Phys.* 31.7 (July 1991), pp. 20–30. ISSN: 0965-5425 (cited on pages 8, 92, 112, 135).

[DK95] V. Druskin and L. Knizhnerman. "Krylov subspace approximation of eigenpairs and matrix functions in exact and computer arithmetic". In: *Numerical Linear Algebra with Applications* 2.3 (May 1995), pp. 205–217. DOI: 10.1002/nla.1680020303 (cited on pages 92, 112).

[DM21] P. Dharangutte and C. Musco. "Dynamic Trace Estimation". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 30088–30099. arXiv: 2110.13752 [cs.DS] (cited on page 54).

[DT21] 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*. 2021. arXiv: 2112.12354 [math.PR] (cited on page 63).

[EOS19] R. Estrin, D. Orban, and M. Saunders. "Euclidean-Norm Error Bounds for SYMMLQ and CG". In: *SIAM Journal on Matrix Analysis and Applications* 40.1 (Jan. 2019), pp. 235–253. DOI: 10.1137/16m1094816 (cited on page 77).

[ER21] N. Eshghi and L. Reichel. "Estimating the error in matrix function approximations". In: *Advances in Computational Mathematics* 47.4 (Aug. 2021). DOI: 10.1007/s10444-021-09882-7 (cited on page 79).

[Esh+02] J. van den Eshof, A. Frommer, T. Lippert, K. Schilling, and H. van der Vorst. "Numerical methods for the QCDd overlap operator. I. Sign-function and error bounds". In: *Computer Physics Communications* 146.2 (2002), pp. 203–224. ISSN: 0010-4655. DOI: 10.1016/S0010-4655(02)00455-1 (cited on pages 2, 98, 100, 103, 112, 122).

[Est00] E. Estrada. "Characterization of 3D molecular structure". In: *Chemical Physics Letters* 319.5-6 (Mar. 2000), pp. 713–718. DOI: 10.1016/s0009-2614(00)00158-5 (cited on page 2).

[Fan+19] L. Fan, D. I. Shuman, S. Ubaru, and Y. Saad. "Spectrum-adapted Polynomial Approximation for Matrix Functions". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2019. DOI: 10.1109/icassp.2019.8683179 (cited on page 3).

[Fas+21] M. Fasi, N. J. Higham, M. Mikaitis, and S. Pranesh. "Numerical behavior of NVIDIA tensor cores". In: *PeerJ Computer Science* 7 (Feb. 2021), e330. DOI: 10.7717/peerj-cs.330 (cited on page 131).

[FG91] B. Fischer and G. H. Golub. "On generating polynomials which are orthogonal over several intervals". In: *Mathematics of Computation* 56.194 (1991), pp. 711–730. DOI: 10.1090/s0025-5718-1991-1068818-5 (cited on pages 31, 62).

[FGS14a] A. Frommer, S. Güttel, and M. Schweitzer. "Convergence of Restarted Krylov Subspace Methods for Stieltjes Functions of Matrices". In: *SIAM Journal on Matrix Analysis and Applications* 35.4 (Jan. 2014), pp. 1602–1624. DOI: 10.1137/140973463 (cited on pages 112, 115).

[FGS14b] A. Frommer, S. Güttel, and M. Schweitzer. "Efficient and Stable Arnoldi Restarts for Matrix Functions Based on Quadrature". In: *SIAM Journal on Matrix Analysis and Applications* 35.2 (Jan. 2014), pp. 661–683. DOI: 10.1137/13093491x (cited on page 108).

[Fis96] B. Fischer. *Polynomial Based Iteration Methods for Symmetric Linear Systems*. Vieweg+Teubner Verlag, 1996. ISBN: 9783663111092. DOI: 10.1007/978-3-663-11108-5 (cited on page 105).

[FKR21] A. Frommer, M. N. Khalil, and G. Ramirez-Hidalgo. *A Multilevel Approach to Variance Reduction in the Stochastic Estimation of the Trace of a Matrix*. 2021. arXiv: 2108.11281 [math.NA] (cited on page 54).

[Fre92] R. W. Freund. "Conjugate Gradient-Type Methods for Linear Systems with Complex Symmetric Coefficient Matrices". In: *SIAM Journal on Scientific and Statistical Computing* 13.1 (Jan. 1992), pp. 425–448. DOI: 10.1137/0913023 (cited on page 73).

[Fro+13] A. Frommer, K. Kahl, T. Lippert, and H. Rittich. "2-Norm Error Bounds and Estimates for Lanczos Approximations to Linear Systems and Rational Matrix Functions". In: *SIAM Journal on Matrix Analysis and Applications* 34.3 (2013), pp. 1046–1065. DOI: 10.1137/110859749 (cited on pages 112, 118).

[Fro+16] R. Frostig, C. Musco, C. Musco, and A. Sidford. "Principal component projection without principal component analysis". In: *International Conference on Machine Learning*. 2016, pp. 2349–2357. arXiv: 1602.06872 [cs.DS] (cited on page 122).

[FS08a] A. Frommer and V. Simoncini. "Matrix Functions". In: *Mathematics in Industry*. Springer Berlin Heidelberg, 2008, pp. 275–303. DOI: 10.1007/978-3-540-78841-6_13 (cited on pages 94, 103).

[FS08b] A. Frommer and V. Simoncini. "Stopping Criteria for Rational Matrix Functions of Hermitian and Symmetric Matrices". In: *SIAM Journal on Scientific Computing* 30.3 (Jan. 2008), pp. 1387–1412. DOI: 10.1137/070684598 (cited on page 112).

[FS09] A. Frommer and V. Simoncini. "Error Bounds for Lanczos Approximations of Rational Functions of Matrices". In: *Numerical Validation in Current Hardware Architectures*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 203–216. ISBN: 978-3-642-01591-5 (cited on pages 103, 108, 112, 118).

[FS15] A. Frommer and M. Schweitzer. "Error bounds and estimates for Krylov subspace approximations of Stieltjes matrix functions". In: *BIT Numerical Mathematics* 56.3 (Dec. 2015), pp. 865–892. DOI: 10.1007/s10543-015-0596-3 (cited on page 112).

[FS50] D. A. Flanders and G. Shortley. "Numerical Determination of Fundamental Modes". In: *Journal of Applied Physics* 21.12 (Dec. 1950), pp. 1326–1332. DOI: 10.1063/1.1699598 (cited on page 7).

[Gau06] W. Gautschi. "Orthogonal Polynomials, Quadrature, and Approximation: Computational Methods and Software (in Matlab)". In: *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 2006, pp. 1–77. DOI: 10.1007/978-3-540-36716-1_1 (cited on page 52).

[GDK99] A. Greenbaum, V. Druskin, and L. A. Knizhnerman. "On solving indefinite symmetric linear systems by means of the Lanczos method". In: *Zh. Vychisl. Mat. Mat. Fiz.* 39.3 (1999), pp. 371–377 (cited on pages 96, 126, 127, 129).

[Gir87] D. Girard. *Un algorithme simple et rapide pour la validation croisée généralisée sur des problèmes de grande taille*. 1987 (cited on pages 51, 53).

[GKX19] B. Ghorbani, S. Krishnan, and Y. Xiao. "An Investigation into Neural Net Optimization via Hessian Eigenvalue Density". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 2232–2241. arXiv: 1901.10159 [cs.LG] (cited on pages 3, 63).

[GM09] G. H. Golub and G. Meurant. *Matrices, moments and quadrature with applications*. Vol. 30. Princeton series in applied mathematics. Princeton University Press, 2009. ISBN: 9780691143415 (cited on pages 3, 9, 28, 52).

[GM94] G. H. Golub and G. Meurant. "Matrices, moments and quadrature". In: *Pitman Research Notes in Mathematics Series* (1994), pp. 105–105 (cited on pages 3, 9, 28).

[GMM09] J. Gemmer, M. Michel, and G. Mahler. *Quantum Thermodynamics*. Springer Berlin Heidelberg, 2009. DOI: 10.1007/978-3-540-70510-9 (cited on page 53).

[GO89] G. H. Golub and D. P. O'Leary. "Some History of the Conjugate Gradient and Lanczos Algorithms: 1948 – 1976". In: *SIAM Review* 31.1 (Mar. 1989), pp. 50–102. DOI: 10.1137/1031003 (cited on page 9).

[Gog10] C. Gogolin. "Pure State Quantum Statistical Mechanics". MA thesis. Julius-Maximilians-Universität Würzburg, 2010. arXiv: 1003.5058 [quant-ph] (cited on page 53).

[Gol+06] S. Goldstein, J. L. Lebowitz, R. Tumulka, and N. Zanghì. "Canonical Typicality". In: *Physical Review Letters* 96.5 (Feb. 2006). DOI: 10.1103/physrevlett.96.050403 (cited on page 53).

[Gol+10] S. Goldstein, J. L. Lebowitz, C. Mastrodonato, R. Tumulka, and N. Zanghì. "Normal typicality and von Neumann's quantum ergodic theorem". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 466.2123 (May 2010), pp. 3203–3224. DOI: 10.1098/rspa.2009.0635 (cited on pages 53, 145).

[GPS59] U. Grenander, H. O. Pollak, and D. Slepian. "The Distribution of Quadratic Forms in Normal Variates: A Small Sample Theory with Applications to Spectral Analysis". In: *Journal of the Society for Industrial and Applied Mathematics* 7.4 (Dec. 1959), pp. 374–401. DOI: 10.1137/0107032 (cited on page 53).

[GR02] M. H. Gutknecht and S. Röllin. "The Chebyshev iteration revisited". In: *Parallel Computing* 28.2 (Feb. 2002), pp. 263–283. DOI: 10.1016/s0167-8191(01)00139-9 (cited on page 7).

[Gre89] A. Greenbaum. "Behavior of slightly perturbed Lanczos and conjugate-gradient recurrences". In: *Linear Algebra and its Applications* 113 (1989), pp. 7–63. ISSN: 0024-3795. DOI: 10.1016/0024-3795(89)90285-1 (cited on pages 8, 138).

[Gre97] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1997. ISBN: 0-89871-396-X (cited on pages 9, 75, 133).

[GS21] S. Güttel and M. Schweitzer. "A Comparison of Limited-memory Krylov Methods for Stieltjes Functions of Hermitian Matrices". In: *SIAM Journal on Matrix Analysis and Applications* 42.1 (Jan. 2021), pp. 83–107. DOI: 10.1137/20m1351072 (cited on pages 103, 119).

[GS92] E. Gallopoulos and Y. Saad. "Efficient Solution of Parabolic Equations by Krylov Approximation Methods". In: *SIAM Journal on Scientific and Statistical Computing* 13.5 (Sept. 1992), pp. 1236–1264. DOI: 10.1137/0913071 (cited on page 92).

[GS94] G. H. Golub and Z. Strakoš. "Estimates in quadratic formulas". In: *Numerical Algorithms* 8.2 (Sept. 1994), pp. 241–268. DOI: 10.1007/bf02142693 (cited on page 3).

[GSO17] A. S. Gambhir, A. Stathopoulos, and K. Orginos. "Deflation as a Method of Variance Reduction for Estimating the Trace of a Matrix Inverse". In: *SIAM Journal on Scientific Computing* 39.2 (Jan. 2017), A532–A558. DOI: 10.1137/16m1066361 (cited on pages 54, 143).

[GT19] A. Gopal and L. N. Trefethen. "Solving Laplace Problems with Corner Singularities via Rational Functions". In: *SIAM Journal on Numerical Analysis* 57.5 (Jan. 2019), pp. 2074–2094. DOI: 10.1137/19m125947x (cited on page 144).

[GWG19] D. Granziol, X. Wan, and T. Garipov. *Deep Curvature Suite*. 2019. arXiv: 1912.09656 [stat.ML] (cited on pages 3, 6, 130).

[Hal21] E. Hallman. "Faster stochastic trace estimation with a Chebyshev product identity". In: *Applied Mathematics Letters* 120 (Oct. 2021), p. 107246. DOI: 10.1016/j.aml.2021.107246 (cited on page 31).

[Han+17] I. Han, D. Malioutov, H. Avron, and J. Shin. "Approximating Spectral Sums of Large-Scale Matrices using Stochastic Chebyshev Approximations". In: *SIAM Journal on Scientific Computing* 39.4 (Jan. 2017), A1558–A1585. DOI: 10.1137/16m1078148 (cited on page 52).

[Hay+72] R. Haydock, V. Heine, M. J. Kelly, and J. B. Pendry. "Electronic Density of States at Transition-Metal Surfaces". In: *Physical Review Letters* 29.13 (Sept. 1972), pp. 868–871. DOI: 10.1103/physrevlett.29.868 (cited on page 29).

[HHK72] R. Haydock, V. Heine, and M. J. Kelly. "Electronic structure based on the local atomic environment for tight-binding bands". In: *Journal of Physics C: Solid State Physics* 5.20 (Oct. 1972), pp. 2845–2858. DOI: 10.1088/0022-3719/5/20/004 (cited on page 29).

[HHK75] R. Haydock, V. Heine, and M. J. Kelly. "Electronic structure based on the local atomic environment for tight-binding bands. II". In: *Journal of Physics C: Solid State Physics* 8.16 (Aug. 1975), pp. 2591–2605. DOI: 10.1088/0022-3719/8/16/011 (cited on page 29).

[HHT08] N. Hale, N. J. Higham, and L. N. Trefethen. "Computing A^α, $\log(A)$, and Related Matrix Functions by Contour Integrals". In: *SIAM Journal on Numerical Analysis* 46.5 (2008), pp. 2505–2523. DOI: 10.1137/070700607. eprint: https://doi.org/10.1137/070700607 (cited on pages 97, 103, 118).

[Hig02] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Jan. 2002. DOI: 10.1137/1.9780898718027 (cited on pages 81, 90, 131, 132).

[Hig08] N. J. Higham. *Functions of Matrices*. Society for Industrial and Applied Mathematics, 2008. DOI: 10.1137/1.9780898717778. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9780898717778 (cited on pages 2, 9).

[HL97] M. Hochbruck and C. Lubich. "On Krylov Subspace Approximations to the Matrix Exponential Operator". In: *SIAM Journal on Numerical Analysis* 34.5 (Oct. 1997), pp. 1911–1925. DOI: 10.1137/S0036142995280572 (cited on pages 2, 112).

[HLS98] M. Hochbruck, C. Lubich, and H. Selhofer. "Exponential Integrators for Large Systems of Differential Equations". In: *SIAM Journal on Scientific Computing* 19.5 (1998), pp. 1552–1574. DOI: 10.1137/S1064827595295337. eprint: https://doi.org/10.1137/S1064827595295337 (cited on page 112).

[HMT11] N. Halko, P. G. Martinsson, and J. A. Tropp. "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions". In: *SIAM Review* 53.2 (Jan. 2011), pp. 217–288. DOI: 10.1137/090771806 (cited on page 143).

[HS52] M. R. Hestenes and E. Stiefel. *Methods of conjugate gradients for solving linear systems*. Vol. 49. NBS Washington, DC, 1952 (cited on pages 7, 89).

[HT21] E. Hallman and D. Troester. *A Multilevel Approach to Stochastic Trace Estimation*. 2021. arXiv: 2103.10516 [math.NA] (cited on page 54).

[Hut89] M. Hutchinson. "A Stochastic Estimator of the Trace of the Influence Matrix for Laplacian Smoothing Splines". In: *Communications in Statistics - Simulation and Computation* 18.3 (Jan. 1989), pp. 1059–1076. DOI: 10.1080/03610918908812806 (cited on pages 51, 53).

[HW71] D. L. Hanson and F. T. Wright. "A Bound on Tail Probabilities for Quadratic Forms in Independent Random Variables". In: *The Annals of Mathematical Statistics* 42.3 (June 1971), pp. 1079–1083. DOI: 10.1214/aoms/1177693335 (cited on page 53).

[ITS09] M. D. Ilic, I. W. Turner, and D. P. Simpson. "A restarted Lanczos approximation to functions of a symmetric matrix". In: *IMA Journal of Numerical Analysis* 30.4 (June 2009), pp. 1044–1061. DOI: 10.1093/imanum/drp003 (cited on page 112).

[Jac12] D. Jackson. "On Approximation by Trigonometric Sums and Polynomials". In: *Transactions of the American Mathematical Society* 13.4 (1912), pp. 491–515. ISSN: 00029947 (cited on page 41).

[JAK19] T. Jawecki, W. Auzinger, and O. Koch. "Computable upper error bounds for Krylov approximations to matrix exponentials and associated φ-functions". In: *BIT Numerical Mathematics* 60.1 (Sept. 2019), pp. 157–197. DOI: 10.1007/s10543-019-00771-6 (cited on page 112).

[Jaw21] T. Jawecki. "A study of defect-based error estimates for the Krylov approximation of φ-functions". In: *Numerical Algorithms* (Nov. 2021). DOI: 10.1007/s11075-021-01190-x (cited on page 112).

[Jin+21] F. Jin, D. Willsch, M. Willsch, H. Lagemann, K. Michielsen, and H. De Raedt. "Random State Technology". In: *Journal of the Physical Society of Japan* 90.1 (Jan. 2021), p. 012001. ISSN: 1347-4073. DOI: 10.7566/jpsj.90.012001 (cited on pages 2, 3, 53, 145).

[JKB94] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous univariate distributions*. 2nd ed. Wiley series in probability and mathematical statistics. Wiley, 1994. ISBN: 9780471584957 (cited on page 56).

[JL14] Z. Jia and H. Lv. "A posteriori error estimates of Krylov subspace approximations to matrix functions". In: 69.1 (June 2014), pp. 1–28. DOI: 10.1007/s11075-014-9878-0 (cited on page 112).

[Joz94] R. Jozsa. "Fidelity for Mixed Quantum States". In: *Journal of Modern Optics* 41.12 (Dec. 1994), pp. 2315–2323. DOI: 10.1080/09500349414552171 (cited on page 2).

[JP94] J. Jaklič and P. Prelovšek. "Lanczos method for the calculation of finite-temperature quantities in correlated systems". In: *Physical Review B* 49.7 (Feb. 1994), pp. 5065–5068. DOI: 10.1103/physrevb.49.5065 (cited on pages 6, 130).

[JS19] Y. Jin and A. Sidford. "Principal Component Projection and Regression in Nearly Linear Time through Asymmetric SVRG". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 3868–3878. arXiv: 1910.06517 [cs.DS] (cited on pages 2, 118, 122, 144).

[Kni96] L. A. Knizhnerman. "The Simple Lanczos Procedure: Estimates of the Error of the Gauss Quadrature Formula and Their Applications". In: *Comput. Math. Math. Phys.* 36.11 (Jan. 1996), pp. 1481–1492. ISSN: 0965-5425 (cited on pages 136–138).

[KW92] J. Kuczyński and H. Woźniakowski. "Estimating the Largest Eigenvalue by the Power and Lanczos Algorithms with a Random Start". In: *SIAM Journal on Matrix Analysis and Applications* 13.4 (Oct. 1992), pp. 1094–1122. DOI: 10.1137/0613066 (cited on page 117).

[Lan50] C. Lanczos. "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators". In: *Journal of research of the National Bureau of Standards* 45 (1950), pp. 255–282 (cited on page 4).

[Led01] M. Ledoux. *The concentration of measure phenomenon*. Nachdr. Mathematical surveys and monographs. American Mathematical Society, 2001. ISBN: 9780821837924 (cited on page 53).

[Li+19] R. Li, Y. Xi, L. Erlandson, and Y. Saad. "The Eigenvalues Slicing Library (EVSL): Algorithms, Implementation, and Software". In: *SIAM Journal on Scientific Computing* 41.4 (Jan. 2019), pp. C393–C415. DOI: 10.1137/18m1170935 (cited on page 3).

[Li22] H. Li. personal communication. 2022 (cited on page 94).

[Lin16] L. Lin. "Randomized estimation of spectral densities of large matrices made accurate". In: *Numerische Mathematik* 136.1 (Aug. 2016), pp. 183–213. DOI: 10.1007/s00211-016-0837-7 (cited on pages 54, 143).

[LS06] L. Lopez and V. Simoncini. "Analysis of Projection Methods for Rational Function Approximation to the Matrix Exponential". In: *SIAM Journal on Numerical Analysis* 44.2 (Jan. 2006), pp. 613–635. DOI: 10.1137/05062590 (cited on pages 71, 95).

[LS13a] Y. T. Lee and A. Sidford. "Efficient Accelerated Coordinate Descent Methods and Faster Algorithms for Solving Linear Systems". In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science* (Oct. 2013). DOI: 10.1109/focs.2013.24 (cited on page 144).

[LS13b] J. Liesen and Z. Strakoš. *Krylov subspace methods: principles and analysis*. 1st ed. Numerical mathematics and scientific computation. Oxford University Press, 2013. ISBN: 9780199655410 (cited on pages 9, 73).

[LSY16] L. Lin, Y. Saad, and C. Yang. "Approximating Spectral Densities of Large Matrices". In: *SIAM Review* 58.1 (Jan. 2016), pp. 34–65. DOI: 10.1137/130934283 (cited on pages 41, 51, 63, 65).

[LZ21] H. Li and Y. Zhu. "Randomized block Krylov subspace methods for trace and log-determinant estimators". In: *BIT Numerical Mathematics* 61.3 (Mar. 2021), pp. 911–939. DOI: 10.1007/s10543-021-00850-7 (cited on pages 54, 143).

[MA17] O. Marchal and J. Arbel. "On the sub-Gaussianity of the Beta and Dirichlet distributions". In: *Electronic Communications in Probability* 22.0 (2017). ISSN: 1083-589X. DOI: 10.1214/17-ecp92 (cited on page 57).

[Meu06] G. Meurant. *The Lanczos and Conjugate Gradient Algorithms*. Society for Industrial and Applied Mathematics, 2006. DOI: 10.1137/1.9780898718140. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9780898718140 (cited on pages 9, 132, 133).

[Mey+21] R. A. Meyer, C. Musco, C. Musco, and D. P. Woodruff. "Hutch++: Optimal Stochastic Trace Estimation". In: *Symposium on Simplicity in Algorithms (SOSA)*. Society for Industrial and Applied Mathematics, Jan. 2021, pp. 142–155. DOI: 10.1137/1.9781611976496.16 (cited on pages 53, 54).

[MM15] C. Musco and C. Musco. *Randomized Block Krylov Methods for Stronger and Faster Approximate Singular Value Decomposition*. Montreal, Canada, 2015. arXiv: 1504.05477 [cs.DS] (cited on page 143).

[MMS18] C. Musco, C. Musco, and A. Sidford. "Stability of the Lanczos Method for Matrix Function Approximation". In: Society for Industrial and Applied Mathematics, Jan. 2018, pp. 1605–1624. DOI: 10.1137/1.9781611975031.105 (cited on pages 8, 112, 135).

[MPT21] G. Meurant, J. Papež, and P. Tichý. "Accurate error estimation in CG". In: *Numerical Algorithms* (Apr. 2021). DOI: 10.1007/s11075-021-01078-w (cited on page 77).

[MS06] G. Meurant and Z. Strakoš. "The Lanczos and conjugate gradient algorithms in finite precision arithmetic". In: *Acta Numerica* 15 (May 2006), pp. 471–542. DOI: 10.1017/s096249290626001x (cited on pages 9, 133).

[MT18] G. Meurant and P. Tichý. "Approximating the extreme Ritz values and upper bounds for the A-norm of the error in CG". In: *Numerical Algorithms* 82.3 (Nov. 2018), pp. 937–968. DOI: 10.1007/s11075-018-0634-8 (cited on page 77).

[MT20] P.-G. Martinsson and J. A. Tropp. "Randomized numerical linear algebra: Foundations and algorithms". In: *Acta Numerica* 29 (May 2020), pp. 403–572. DOI: 10.1017/s0962492920000021 (cited on page 143).

[Neu29] J. von Neumann. "Beweis des Ergodensatzes und des$H$-Theorems in der neuen Mechanik". In: *Zeitschrift für Physik* 57.1-2 (Jan. 1929). English translation https://arxiv.org/abs/1003.2133, pp. 30–70. DOI: 10.1007/bf01339852 (cited on page 53).

[NPS16] E. D. Napoli, E. Polizzi, and Y. Saad. "Efficient estimation of eigenvalue counts in an interval". In: *Numerical Linear Algebra with Applications* 23.4 (Mar. 2016), pp. 674–692. DOI: 10.1002/nla.2048 (cited on page 122).

[NST18] Y. Nakatsukasa, O. Sète, and L. N. Trefethen. "The AAA Algorithm for Rational Approximation". In: *SIAM Journal on Scientific Computing* 40.3 (Jan. 2018), A1494–A1522. DOI: 10.1137/16m1106122 (cited on page 97).

[NSW14] D. Needell, N. Srebro, and R. Ward. "Stochastic Gradient Descent, Weighted Sampling, and the Randomized Kaczmarz Algorithm". In: NIPS'14. Montreal, Canada: MIT Press, 2014, pp. 1017–1025 (cited on page 144).

[NW83] A. Nauts and R. E. Wyatt. "New Approach to Many-State Quantum Dynamics: The Recursive-Residue-Generation Method". In: *Physical Review Letters* 51.25 (Dec. 1983), pp. 2238–2241. DOI: 10.1103/physrevlett.51.2238 (cited on page 92).

[Pai71] C. C. Paige. "The computation of eigenvalues and eigenvectors of very large sparse matrices." PhD thesis. University of London, 1971 (cited on pages 9, 134).

[Pai72] C. C. Paige. "Computational Variants of the Lanczos Method for the Eigenproblem". In: *IMA Journal of Applied Mathematics* 10.3 (1972), pp. 373–381. DOI: 10.1093/imamat/10.3.373 (cited on page 134).

[Pai76] C. C. Paige. "Error Analysis of the Lanczos Algorithm for Tridiagonalizing a Symmetric Matrix". In: *IMA Journal of Applied Mathematics* 18.3 (Dec. 1976), pp. 341–349. ISSN: 0272-4960. DOI: 10.1093/imamat/18.3.341 (cited on page 134).

[Pai80] C. C. Paige. "Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem". In: *Linear Algebra and its Applications* 34 (1980), pp. 235–258. ISSN: 0024-3795. DOI: 10.1016/0024-3795(80)90167-6 (cited on page 134).

[Pap19] V. Papyan. *The Full Spectrum of Deepnet Hessians at Scale: Dynamics with SGD Training and Sample Size*. 2019. arXiv: 1811.07062 [cs.LG] (cited on page 3).

[PCK22] D. Persson, A. Cortinovis, and D. Kressner. "Improved Variants of the Hutch++ Algorithm for Trace Estimation". In: *SIAM Journal on Matrix Analysis and Applications* 43.3 (July 2022), pp. 1162–1185. DOI: 10.1137/21m1447623 (cited on pages 53, 54).

[PL04] R. Pace and J. P. LeSage. "Chebyshev approximation of log-determinants of spatial weight matrices". In: *Computational Statistics & Data Analysis* 45.2 (Mar. 2004), pp. 179–196. DOI: 10.1016/s0167-9473(02)00321-3 (cited on page 2).

[PL86] T. J. Park and J. C. Light. "Unitary quantum time evolution by iterative Lanczos reduction". In: *The Journal of Chemical Physics* 85.10 (Nov. 1986), pp. 5870–5876. DOI: 10.1063/1.451548 (cited on page 92).

[Ple+20] G. Pleiss, M. Jankowiak, D. Eriksson, A. Damle, and J. Gardner. "Fast Matrix Square Roots with Applications to Gaussian Processes and Bayesian Optimization". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 22268–22281. arXiv: 2006.11267 [cs.LG] (cited on pages 2, 103, 118).

[Pol09] E. Polizzi. "Density-matrix-based algorithm for solving eigenvalue problems". In: *Physical Review B* 79.11 (Mar. 2009). DOI: 10.1103/physrevb.79.115112 (cited on page 3).

[PPV95] C. C. Paige, B. N. Parlett, and H. A. van der Vorst. "Approximate solutions and eigenvalue bounds from Krylov subspaces". In: *Numerical Linear Algebra with Applications* 2.2 (Mar. 1995), pp. 115–133. DOI: 10.1002/nla.1680020205 (cited on page 98).

[PS75] C. C. Paige and M. A. Saunders. "Solution of Sparse Indefinite Systems of Linear Equations". In: *SIAM Journal on Numerical Analysis* 12.4 (Sept. 1975), pp. 617–629. DOI: 10.1137/0712047 (cited on page 73).

[PSS82] B. N. Parlett, H. Simon, and L. M. Stringer. "On estimating the largest eigenvalue with the Lanczos algorithm". In: *Mathematics of Computation* 38.157 (Jan. 1982), pp. 153–153. DOI: 10.1090/s0025-5718-1982-0637293-9 (cited on page 117).

[PSW06] S. Popescu, A. J. Short, and A. Winter. "Entanglement and the foundations of statistical mechanics". In: *Nature Physics* 2.11 (Oct. 2006), pp. 754–758. ISSN: 1745-2481. DOI: 10.1038/nphys444 (cited on page 53).

[RA14] F. Roosta-Khorasani and U. Ascher. "Improved Bounds on Sample Size for Implicit Matrix Trace Estimators". In: *Foundations of Computational Mathematics* 15.5 (Sept. 2014), pp. 1187–1212. DOI: 10.1007/s10208-014-9220-1 (cited on pages 52, 53).

[Rei07] P. Reimann. "Typicality for Generalized Microcanonical Ensembles". In: *Physical Review Letters* 99.16 (Oct. 2007). DOI: 10.1103/physrevlett.99.160404 (cited on page 53).

[Riv81] T. J. Rivlin. *An introduction to the approximation of functions*. Unabridged and corr. republication of the 1969 ed. Dover books on advanced mathematics. Dover, 1981. ISBN: 9780486640693 (cited on pages 21, 23, 24).

[RV13] M. Rudelson and R. Vershynin. "Hanson-Wright inequality and sub-gaussian concentration". In: *Electronic Communications in Probability* 18.none (Jan. 2013). DOI: 10.1214/ecp.v18-2865 (cited on page 53).

[RV89] H. D. Raedt and P. de Vries. "Simulation of two and three-dimensional disordered systems: Lifshitz tails and localization properties". In: *Zeitschrift für Physik B Condensed Matter* 77.2 (June 1989), pp. 243–251. DOI: 10.1007/bf01313668 (cited on page 53).

[Saa11] Y. Saad. *Numerical Methods for Large Eigenvalue Problems: Revised Edition*. en. Society for Industrial and Applied Mathematics, Jan. 2011. ISBN: 9781611970722. DOI: 10.1137/1.9781611970739 (cited on page 9).

[Saa92] Y. Saad. "Analysis of Some Krylov Subspace Approximations to the Matrix Exponential Operator". In: *SIAM Journal on Numerical Analysis* 29.1 (1992), pp. 209–228. DOI: 10.1137/0729014. eprint: https://doi.org/10.1137/0729014 (cited on pages 2, 92).

[SAI17] A. K. Saibaba, A. Alexanderian, and I. C. F. Ipsen. "Randomized matrix-free trace and log-determinant estimators". In: *Numerische Mathematik* 137.2 (Apr. 2017), pp. 353–395. DOI: 10.1007/s00211-017-0880-z (cited on pages 54, 143).

[Sch+21] H. Schlüter, F. Gayk, H.-J. Schmidt, A. Honecker, and J. Schnack. "Accuracy of the typicality approach using Chebyshev polynomials". In: *Zeitschrift für Naturforschung A* 76.9 (June 2021), pp. 823–834. DOI: 10.1515/zna-2021-0116 (cited on pages 67–69).

[Sch11] K. Schiefermayr. "Estimates for the asymptotic convergence factor of two intervals". In: *Journal of Computational and Applied Mathematics* 236.1 (Aug. 2011), pp. 28–38. ISSN: 0377-0427. DOI: 10.1016/j.cam.2010.06.008 (cited on page 105).

[Sch16] M. Schweitzer. "Restarting and error estimation in polynomial and extended Krylov subspace methods for the approximation of matrix functions". PhD thesis. Universität Wuppertal, Fakultät für Mathematik und Naturwissenschaften, 2016. eprint: http://elpub.bib.uni-wuppertal.de/servlets/DocumentServlet?id=5590 (cited on page 9).

[Sch27] E. Schrödinger. "Energieaustausch nach der Wellenmechanik". In: *Annalen der Physik* 388.15 (1927), pp. 956–968. DOI: 10.1002/andp.19273881504 (cited on page 53).

[SD71] R. A. Sack and A. F. Donovan. "An algorithm for Gaussian quadrature given modified moments". In: *Numerische Mathematik* 18.5 (Oct. 1971), pp. 465–478. DOI: 10.1007/bf01406683 (cited on page 34).

[SG92] Z. Strakos and A. Greenbaum. "Open questions in the convergence analysis of the Lanczos process for the real symmetric eigenvalue problem". In: University of Minnesota, 1992. eprint: https://conservancy.umn.edu/handle/11299/1838 (cited on page 149).

[Sil+96] R. Silver, H. Roeder, A. Voter, and J. Kress. "Kernel Polynomial Approximations for Densities of States and Spectral Functions". In: *Journal of Computational Physics* 124.1 (Mar. 1996), pp. 115–130. DOI: 10.1006/jcph.1996.0048 (cited on pages 6, 52).

[Ski89] J. Skilling. "The Eigenvalues of Mega-dimensional Matrices". In: *Maximum Entropy and Bayesian Methods*. Springer Netherlands, 1989, pp. 455–466. DOI: 10.1007/978-94-015-7860-8_48 (cited on pages 31, 51, 52).

[SR94] R. Silver and H. Röder. "Densities of states of mega-dimensional Hamiltonian matrices". In: *International Journal of Modern Physics C* 05.04 (Aug. 1994), pp. 735–753. DOI: 10.1142/s0129183194000842 (cited on pages 31, 52).

[SRS20] J. Schnack, J. Richter, and R. Steinigeweg. "Accuracy of the finite-temperature Lanczos method compared to simple typicality-based estimates". In: *Physical Review Research* 2.1 (Feb. 2020). DOI: 10.1103/physrevresearch.2.013186 (cited on pages 2, 67, 68).

[SRS22] H. Schlüter, J. Richter, and J. Schnack. *Melting of magnetization plateaus for kagome and square-kagome lattice antiferromagnets*. 2022. arXiv: 2206.08710 [cond-mat.str-el] (cited on page 67).

[SS10] R. Schnalle and J. Schnack. “Calculating the energy spectra of magnetic molecules: application of real- and spin-space symmetries”. In: *International Reviews in Physical Chemistry* 29.3 (July 2010), pp. 403–452. DOI: 10.1080/0144235x.2010.485755 (cited on pages 2, 67).

[ST02] Z. Strakoš and P. Tichý. “On error estimation in the conjugate gradient method and why it works in finite precision computations.” In: *ETNA. Electronic Transactions on Numerical Analysis [electronic only]* 13 (2002), pp. 56–80. eprint: https://etna.ricam.oeaw.ac.at/vol.13.2002/pp56-80.dir/pp56-80.pdf (cited on page 77).

[ŠT21] D. Šimonová and P. Tichý. *When does the Lanczos algorithm compute exactly?* 2021. arXiv: 2106.02068 [math.NA] (cited on page 73).

[Str91] Z. Strakos. “On the real convergence rate of the conjugate gradient method”. In: *Linear Algebra and its Applications* 154-156 (1991), pp. 535–549. ISSN: 0024-3795. DOI: 10.1016/0024-3795(91)90393-B (cited on page 149).

[SV08] T. Strohmer and R. Vershynin. “A Randomized Kaczmarz Algorithm with Exponential Convergence”. In: *Journal of Fourier Analysis and Applications* 15.2 (Apr. 2008), pp. 262–278. DOI: 10.1007/s00041-008-9030-4 (cited on page 144).

[Tre08] L. N. Trefethen. “Is Gauss Quadrature Better than Clenshaw–Curtis?” In: *SIAM Review* 50.1 (Jan. 2008), pp. 67–87. DOI: 10.1137/060659831 (cited on pages 45, 46).

[Tre19] L. N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Society for Industrial and Applied Mathematics, Jan. 2019. DOI: 10.1137/1.9781611975949 (cited on pages 17, 19, 51, 97).

[Tro21] J. A. Tropp. “Randomized block Krylov methods for approximating extreme eigenvalues”. In: *Numerische Mathematik* 150.1 (Dec. 2021), pp. 217–255. ISSN: 0945-3245. DOI: 10.1007/s00211-021-01250-3 (cited on page 143).

[TW14] L. N. Trefethen and J. A. C. Weideman. “The Exponentially Convergent Trapezoidal Rule”. In: *SIAM Review* 56.3 (Jan. 2014), pp. 385–458. DOI: 10.1137/130932132 (cited on page 118).

[TWO17] A. Townsend, M. Webb, and S. Olver. “Fast polynomial transforms based on Toeplitz and Hankel matrices”. In: *Mathematics of Computation* 87.312 (Nov. 2017), pp. 1913–1934. DOI: 10.1090/mcom/3277 (cited on page 33).

[UCS17] S. Ubaru, J. Chen, and Y. Saad. "Fast Estimation of tr($f(A)$) via Stochastic Lanczos Quadrature". In: *SIAM Journal on Matrix Analysis and Applications* 38.4 (Jan. 2017), pp. 1075–1099. DOI: 10.1137/16m1104974 (cited on pages 6, 51, 52, 130).

[Ver18] R. Vershynin. *High-Dimensional Probability*. Cambridge University Press, Sept. 2018. DOI: 10.1017/9781108231596 (cited on page 56).

[Vor87] H. V. D. Vorst. "An iterative solution method for solving f(A)x = b, using Krylov subspace information obtained for the symmetric positive definite matrix A". In: *Journal of Computational and Applied Mathematics* 18.2 (May 1987), pp. 249–263. DOI: 10.1016/0377-0427(87)90020-3 (cited on page 92).

[Wan+21] S. Wang, Y. Sun, C. Musco, and Z. Bao. "Public Transport Planning". In: *Proceedings of the 2021 International Conference on Management of Data*. Association for Computing Machinery, June 2021. DOI: 10.1145/3448016.3457247 (cited on page 2).

[WB72] J. C. Wheeler and C. Blumstein. "Modified Moments for Harmonic Solids". In: *Physical Review B* 6.12 (Dec. 1972), pp. 4380–4382. DOI: 10.1103/physrevb.6.4380 (cited on page 29).

[Wei+06] A. Weiße, G. Wellein, A. Alvermann, and H. Fehske. "The kernel polynomial method". In: *Reviews of Modern Physics* 78.1 (Mar. 2006), pp. 275–306. DOI: 10.1103/revmodphys.78.275 (cited on pages 2, 3, 6, 9, 26, 31, 41, 52, 67, 130).

[WO21] M. Webb and S. Olver. "Spectra of Jacobi Operators via Connection Coefficient Matrices". In: *Communications in Mathematical Physics* 382.2 (Feb. 2021), pp. 657–707. DOI: 10.1007/s00220-021-03939-w (cited on pages 31, 32).

[WT07] J. A. C. Weideman and L. N. Trefethen. "The kink phenomenon in Fejér and Clenshaw–Curtis quadrature". In: *Numerische Mathematik* 107.4 (Aug. 2007), pp. 707–727. DOI: 10.1007/s00211-007-0101-2 (cited on page 45).

[WW76] D. Weaire and A. R. Williams. "New numerical approach to the Anderson localization problem". In: *Journal of Physics C: Solid State Physics* 9.17 (Sept. 1976), pp. L461–L463. DOI: 10.1088/0022-3719/9/17/004 (cited on page 53).

[WW77] D. Weaire and A. R. Williams. "The Anderson localization problem. I. A new numerical approach". In: *Journal of Physics C: Solid State Physics* 10.8 (Apr. 1977), pp. 1239–1245. DOI: 10.1088/0022-3719/10/8/025 (cited on page 53).

[WWZ14] K. Wimmer, Y. Wu, and P. Zhang. "Optimal Query Complexity for Estimating the Trace of a Matrix". In: *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*. Ed. by J. Esparza, P. Fraigniaud, T. Husfeldt, and E. Koutsoupias. Vol. 8572. Lecture Notes in Computer Science. Springer, 2014, pp. 1051–1062. DOI: 10.1007/978-3-662-43948-7_87 (cited on page 54).

[Yao+20] Z. Yao, A. Gholami, K. Keutzer, and M. Mahoney. *PyHessian: Neural Networks Through the Lens of the Hessian*. 2020. arXiv: 1912.07145 [cs.LG] (cited on page 3).

[Zol77] E. Zolotarev. "Application of elliptic functions to questions of functions deviating least and most from zero". In: *Zap. Imp. Akad. Nauk. St. Petersburg* 30.5 (1877), pp. 1–59 (cited on page 101).

[ZZ20] A. R. Zhang and Y. Zhou. "On the non-asymptotic and sharp lower tail bounds of random variables". In: *Stat* 9.1 (Oct. 2020). DOI: 10.1002/sta4.314 (cited on page 58).