EXTENSIONS OF GAUSS, BLOCK GAUSS, AND SZEGŐ QUADRATURE RULES, WITH APPLICATIONS

A dissertation submitted to
Kent State University in partial fulfillment of the requirements for the degree of Doctor of Philosophy

by
Tunan Tang
May, 2016
# TABLE OF CONTENTS

LIST OF FIGURES ........................................... v

LIST OF TABLES ........................................... vi

Acknowledgements ........................................ viii

1 Introduction ........................................... 1

2 Background and Notation ............................... 5

2.1 Orthogonal Polynomials ............................. 5

2.2 Quadrature Rules ..................................... 7

2.3 Moments and Modified Moments ..................... 10

3 Matrices, Polynomials and Quadrature ............... 13

3.1 The Problem .......................................... 13

3.2 Quadrature Rules for Hermitian Functionals ........ 13

3.3 Quadrature Rules for Non-Hermitian Functionals ... 22

4 Matrices, Matrix Polynomials and Block Quadrature .. 33

4.1 The Problem .......................................... 33

4.2 Block Quadrature Rules for Hermitian Matrices .... 34

4.3 Block Quadrature Rules for Non-Hermitian Matrices .. 41
LIST OF FIGURES
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Relative errors in computed approximations of $F(A) = u^T(I + A^2)^{-1}u$ with $A$ symmetric indefinite.</td>
<td>58</td>
</tr>
<tr>
<td>2</td>
<td>Relative error estimates for computed approximations of $F(A) = u^T(I + A^2)^{-1}u$ with $A$ symmetric indefinite.</td>
<td>58</td>
</tr>
<tr>
<td>3</td>
<td>Relative errors of $F(A) = u^T \exp(A)u$ with $A$ a banded nonsymmetric Toeplitz matrix.</td>
<td>59</td>
</tr>
<tr>
<td>4</td>
<td>Relative error estimates for computed approximations of $F(A) = u^T \exp(A)u$ with $A$ a banded nonsymmetric Toeplitz matrix.</td>
<td>59</td>
</tr>
<tr>
<td>5</td>
<td>Relative errors of $F(A) = u^T \log(A + cI)u$ with $A$ a banded nonsymmetric Toeplitz matrix.</td>
<td>60</td>
</tr>
<tr>
<td>6</td>
<td>Relative error estimates for computed approximations of $F(A) = u^T \log(A + cI)v$ with $A$ a banded nonsymmetric Toeplitz matrix.</td>
<td>60</td>
</tr>
<tr>
<td>7</td>
<td>Relative errors in computed approximations of $F(A) = u^T \log(A + cI)v$ with $A$ nonsymmetric.</td>
<td>61</td>
</tr>
<tr>
<td>8</td>
<td>Relative error estimates for computed approximations of $F(A) = u^T \log(A + cI)v$ with $A$ nonsymmetric.</td>
<td>61</td>
</tr>
<tr>
<td>9</td>
<td>Relative errors in computed approximations of (100) for $p = 2$.</td>
<td>63</td>
</tr>
<tr>
<td>10</td>
<td>Relative errors in computed approximations of $c^T \exp(A)c$, $c = [1, 1, \ldots, 1]^T$.</td>
<td>63</td>
</tr>
<tr>
<td>11</td>
<td>Errors in approximations of $F(A) = W^T \exp(A)W$ with $A$ symmetric indefinite.</td>
<td>65</td>
</tr>
</tbody>
</table>
12 Error estimates for computed approximations of \( F(A) = W^T \exp(A)W \) with \( A \) symmetric indefinite. .................................................. 65

13 Errors in approximations of \( F(A) = W^T A^{-1}W \) with \( A \) symmetric positive definite. .................................................. 66

14 Error estimates for computed approximations of \( F(A) = W^T A^{-1}W \) with \( A \) symmetric positive definite. .................................................. 67

15 Errors in approximations of \( F(A) = W^T A^{-1}V \) with \( A \) nonsymmetric. 68

16 Error estimates for computed approximations of \( F(A) = W^T A^{-1}V \) with \( A \) nonsymmetric. .................................................. 68

17 Errors in approximations of \( F(A) = W^T \exp(A)V \) with \( A \) nonsymmetric. 68

18 Error estimates for computed approximations of \( F(A) := W^T \exp(A)V \) with \( A \) nonsymmetric. .................................................. 69

19 Errors for several quadrature rules applied to the approximation of the integral (102) with integrand (103). .................................................. 70

20 Estimate of absolute error in the approximation furnished by \( S_1^{(n)}(T) \). 71

21 Errors for several quadrature rules applied to the approximation of the integral (102) with integrand (103). .................................................. 71

22 Estimate of absolute error in the approximation furnished by \( S_1^{(n)}(T) \). 72

23 Errors for several quadrature rules applied to the approximation of the integral (102) with integrand (103). .................................................. 72

24 Estimate of absolute error in the approximation furnished by \( S_1^{(n)}(T) \). 72
Acknowledgements

I would like to give my special thanks to Dr. Lothar Reichel for his patience and wisdom. Without his encouragement and instruction, I can never complete this. Also, I would like to thank my parents for their support and understanding along with my study.
CHAPTER 1

Introduction

With the advance of information technology, the volume of data that we generate explodes. Calculation with larger sets of data needs either a more powerful computer, or a better algorithm. In this dissertation, we mainly discuss the latter; specifically, we describe the recent development of practical numerical algorithms that calculate approximations of smooth functions applied to a large matrix, i.e., how to compute an estimate for $W^T f(A)V$, where $f$ is a smooth function, $A \in \mathbb{R}^{m \times m}$ and $V, W \in \mathbb{R}^{m \times k}$. When $k = 1$, $V, W$ become vectors and we denote the problem by $u^T f(A)v$.

There are many applications around the problem that require the evaluation of expression of the form $W^T f(A)V$ or $u^T f(A)v$; see [15, 17, 20, 29, 30, 42, 44, 64]. Expression $u^T f(A)v$ arise in the following problems.

1. Error estimates. In numerical linear algebra, a core problem is to solve a linear system of equations $Ax = b$, where $A$ is a large sparse symmetric positive definite matrix, $b$ is a given vector and we are trying to solve the equation for $x$. An algorithm gives the solution $\xi$. We can assume that $x = \xi + \epsilon$ for some perturbation $\epsilon$. The residual is $r = b - A\xi = A\epsilon$. Therefore the error is given by $\epsilon = A^{-1}r$. The square of the norm of the error is $\|\epsilon\|_2^2 = r^T A^{-2}r$, which is of the form $u^T f(A)v$ for $f(t) = t^{-2}$ and $u = v = r$; see [42] for more details.

2. Network analysis. In network analysis, an important topic is how to assess the
importance and accessibility of different nodes. Given a graph $G$ which is composed of a set of vertexes $V$ and a set of edges $E$. The adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{m \times m}$ associated with $G$ has the entry $a_{ij} = 1$ if there is an edge between node $i$ and node $j$; otherwise $a_{ij} = 0$. The function used in network analysis to measure the overall easiness of traveling from one node to another has the general form

$$f(A) = \sum_{j=0}^{\infty} c_j A^j$$

with positive coefficients $c_j$ chosen to guarantee convergence. The popular choice $c_j = 1/j!$ yields $f(A) = \exp(A)$. The easiness to move from node $i$ to node $j$, commonly referred to as the communicability between node $i$ and $j$, is of the form $u^T f(A) v$ for $f(t) = \exp(t)$ and $u = e_i$ and $v = e_j$, where $e_j$ denotes the $j$th axis vector.

When computing functions of a matrix, by definition,

$$W^T f(A)V = W^T Q f(\Lambda) Q^T V, \tag{1}$$

where $A = Q \Lambda Q^T$ is the spectral factorization of matrix $A$, i.e., $Q \in \mathbb{R}^{m \times m}$ and satisfies $Q^T Q = I_m$. It is known that for a matrix like $A$ with size $m \times m$, the order of complexity for such a factorization is $O(m^3)$, which could be prohibitively unaffordable for a larger matrix. Therefore, alternative methods have been developed in recent decades for the approximation of (1), among them Lanczos-type methods generate good approximate solutions to (1) by evaluating matrix-vector products with $A$. Their order of complexity is $O(m^2)$. Lanczos-type methods are attractive not only because of their fast calculation, but also because they allow the estimation of the accuracy of the computed approximation. The calculations can be stopped when a prescribed tolerance is met.

This dissertation is organized as follows. Chapter 2 introduces some background
material related to our problem, including orthogonal polynomials, quadrature rules and moments. The Lanczos-type methods are firmly rooted and intimately connected to orthogonal polynomials and quadrature rules.

In Chapter 3 we consider numerical methods based on the application of Gauss-type quadrature rules computed via Lanczos algorithms. The need to determine upper and lower bounds for matrix functionals of the form $u^T f(A)v$ arises in a large number of applications. Here $A \in \mathbb{R}^{m \times m}$ denotes a large matrix and $u, v \in \mathbb{R}^m$ are vectors satisfying $u^T v = 1$. Golub and collaborators have shown how upper and lower bounds for $u^T f(A)v$ can be computed by using Gauss and Gauss-Radau quadrature rules when $A$ is symmetric and the derivatives of $f$ are of constant sign in an interval that contains the spectrum of $A$. However, many matrix functionals of interest in applications are defined by functions $f$, whose derivatives do not have constant sign on the spectrum of $A$. Spalević [70, 71] recently presented generalized averaged Gauss quadrature rules; we describe how these rules can be applied to compute estimates of upper and lower bounds for $u^T f(A)v$ based on the application of pairs of Gauss and generalized averaged Gauss quadrature rules. This method does not require the sign of the derivatives of $f$ to be constant on an interval that contains the spectrum of $A$. Moreover, $A$ is not required to be symmetric.

In Chapter 4 we consider numerical methods based on extensions of generalized averaged Gauss quadrature rules discussed in Chapter 3, computed via block Lanczos algorithms. Block Lanczos methods have been widely used to compute approximations of matrix-valued functions of the form $W^T f(A)V$, where $A \in \mathbb{R}^{m \times m}, W, V \in \mathbb{R}^{m \times k}, k \ll m$ and $V^T W = I_k$. Golub and Meurant [41, 43] discuss how the application of a few steps of
the symmetric block Lanczos method to a symmetric matrix $A$ with initial block vector $W$ yields an estimate of $W^T f(A) W$, and show that this approximation can be interpreted as a Gauss-type quadrature rule with respect to a discrete matrix-valued measure. In [29], the Gauss and anti-Gauss quadrature rules computed via the symmetric and nonsymmetric block Lanczos algorithms were used to compute approximate bounds for the quantity $W^T f(A) V$. The block quadrature rules discussed in this chapter provide efficient and algorithmically simple ways of computing estimates of upper and lower bounds for each element of $W^T f(A) V$. We illustrate that when a $k \times k$ sub-matrix of $f(A)$ is desired, the one-time application of a block method with block size $k$ gives an efficient alternative to $O(k^2)$ applications of a standard method with block size 1 in terms of the required number of matrix-vector product evaluations.

In Chapter 5, inspired by the work of [12,14,55], we propose an analogue of generalized averaged Gauss-type quadrature rules for Szegő quadrature rules. Szegő quadrature rules are commonly applied to integrate periodic functions on the unit circle in the complex plane. However, often it is difficult to determine the quadrature error. We describe new quadrature rules for the unit circle that often yield higher accuracy than Szegő rules using the same moment information and may be used to estimate the error in Szegő quadrature rules.

In Chapter 6 we present various numerical examples that exhibit the accuracy of several of the quadrature rules discussed in prior chapters. Additionally, application of generalized averaged quadrature rules to complex network analysis is provided.
2.1 Orthogonal Polynomials

A sequence of orthogonal polynomials are a family of polynomials such that any two polynomials of different degrees are orthogonal to each other with respect to some inner product.

Consider the inner product

$$\langle f, g \rangle := \int f(x)g(x)d\mu(x),$$

where $\mu(x) \geq 0$ is a nondecreasing function, called a weight function or measure. Assume there exists a sequence of polynomials $\{p_j\}_{j=0}^{\infty}$, where $p_j$ is of degree $j$, such that

$$\langle p_i, p_j \rangle \begin{cases} \neq 0 & \text{if } i = j, \\ = 0 & \text{if } i \neq j. \end{cases}$$

These polynomials are called orthogonal polynomials. If in addition

$$\langle p_i, p_j \rangle = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

then they are said to be orthonormal polynomials. And if the leading coefficient of the polynomials $p_j$ is one, then they are called monic.

The orthogonal polynomials depend on the weight functions. In science and engineering, the most commonly used classical orthogonal polynomials are the Hermite
polynomials, the Laguerre polynomials and the Jacobi polynomials; see [74] for other widely used polynomials.

It often is convenient to represent a given function by an expansion in terms of orthogonal polynomials. The Gram-Schmidt orthogonalization process applied to the basis 1, x, x², ..., determines monic orthogonal polynomials as follows:

**Algorithm 2.1** The Gram-Schmidt algorithm

1: Initialization: \( p_0(x) = 1; \)
2: Iterate: for \( j = 1, 2, \ldots \) do

\[
p_j(x) = x^j - \frac{\langle x^j, p_0 \rangle}{\langle p_0, p_0 \rangle} p_0(x) - \cdots - \frac{\langle x^j, p_{j-1} \rangle}{\langle p_{j-1}, p_{j-1} \rangle} p_{j-1}(x)
\]

end \( j \)

It is easy to convert the above monic orthogonal polynomials to orthonormal polynomials by

\[
\hat{p}_j(x) = \frac{p_j(x)}{\langle p_j, p_j \rangle^{1/2}}.
\]

A well known result for orthonormal polynomials is that they satisfy a three-term recurrence relation; see [73] for a complete proof.

**Theorem 1.** Let \( \{p_i\}_{i=0}^\infty \) be a sequence of orthonormal polynomials, then there are two sequences of coefficients \( \{\alpha_i\}_{i=1}^\infty \) and \( \{\beta_i\}_{i=0}^\infty \) such that

\[
\beta_{j+1}p_{j+1}(x) = (x - \alpha_{j+1})p_j(x) - \beta_j p_{j-1}(x), \quad j = 0, 1, 2, \ldots,
\]

\[
p_{-1}(x) \equiv 0, \quad p_0(x) \equiv 1/\beta_0, \quad \beta_0 = \int \! d\mu(x)
\]

where

\[
\alpha_{j+1} = \langle xp_j, p_j \rangle, \quad j = 0, 1, \ldots
\]

and \( \beta_j \) is computed such that \( \|p_j\| = \langle p_j, p_j \rangle^{1/2} = 1. \)
2.2 Quadrature Rules

Calculating the integral of a function with respect to a given measure,

\[ I f := \int_a^b f(x) \text{d}\mu(x), \]  

is a classical problem. Usually, definite integrals are calculated by discretization. The Newton-Cotes formulas are traditional integral formulas that are still widely used; see [73] for more details about Newton-Cotes formulas, including the Trapezoidal rule, Simpson’s rule, the 3/8-rule, Milne’s rule, Weddle’s rule, etc.

Quadrature rules often are of the form

\[ \int_a^b f(x) \text{d}\mu(x) \approx \sum_{j=1}^{N} w_j f(t_j) + \sum_{k=1}^{M} v_k f(z_k), \]

where the weights \( \{w_j\}_{j=1}^{N}, \{v_k\}_{k=1}^{M} \) and the nodes \( \{t_j\}_{j=1}^{N} \) are unknowns and the nodes \( \{z_k\}_{k=1}^{M} \) are prescribed; see [23, 31, 34, 45].

Let us examine quadrature rules of the following form

\[ Q f = \sum_{j=1}^{\ell} w_j f(x_j) \]  

first; our aim is to choose the nodes \( \{x_j\}_{j=1}^{\ell} \) as well as their corresponding weights \( \{w_j\}_{j=1}^{\ell} \) such that the quadrature rule is exact for polynomials of as high degree as possible. This leads us to Gauss quadrature rules. Denote the \( \ell \)-point Gauss quadrature rule with respect to the measure \( \mu \) by

\[ G_{\ell} f := \sum_{i=1}^{\ell} \eta_{i}^{(\ell)} f(\theta_{i}^{(\ell)}), \]

with the Gauss weights \( \eta_{i}^{(\ell)} \) and Gauss nodes \( \theta_{i}^{(\ell)} \).

**Theorem 2.** The \( \ell \)-point Gauss quadrature rule has degree of exactness \( 2\ell - 1 \), i.e.,
\( \forall p \in \mathbb{P}_{2\ell-1}, \quad Qp = \mathcal{I}p. \) There is no quadrature rule with \( \ell \) nodes that is exact for all \( p \in \mathbb{P}_{2\ell}. \)

There is an intimate relation between orthogonal polynomials and Gauss quadrature. Here we briefly summarize and deliver the results. The complete proofs can be found in [73].

**Theorem 3.** Let \( \ell \in \mathbb{N} \). The roots of the orthonormal polynomial \( p_{\ell} \) with respect to the measure \( \mu \) are the nodes of the \( \ell \)-point Gauss quadrature rule (6).

**Theorem 4.** The weights of the \( \ell \)-point Gauss quadrature rule \( \eta_{i}^{(\ell)} \) are the solution of the (nonsingular) system of equations

\[
\sum_{i=1}^{\ell} \eta_{i}^{(\ell)} p_{k}(\theta_{i}^{(\ell)}) = \begin{cases} 
1 & \text{if } k = 0, \\
0 & \text{if } k = 1, 2, \ldots, \ell - 1.
\end{cases}
\]

Though the above two theorems give the theoretical methods to approximate the value (3) by using Gauss quadrature rules (6), there are more attractive characteristics of Gauss quadrature rules which can help us do the calculation.

First let us review some results. Recall the three-term recurrence relation in (2). For the \( \ell \)-point Gauss quadrature rule, define the symmetric tridiagonal matrix

\[
T_{\ell} = \begin{bmatrix}
\alpha_{0} & \beta_{1} & \quad & 0 \\
\beta_{1} & \alpha_{1} & \beta_{2} & \\
& \ddots & \ddots & \ddots \\
\beta_{\ell-2} & \alpha_{\ell-2} & \beta_{\ell-1} & \\
0 & \beta_{\ell-1} & \alpha_{\ell-1} & \\
\end{bmatrix} \in \mathbb{R}^{\ell \times \ell}
\] (7)
determined by the first $2\ell - 1$ nontrivial recursion coefficients. In matrix form, the three-term recursion can be written as

$$xp(x) = T_\ell p(x) + \beta_\ell p_\ell(x)e_\ell,$$

where $p(x)^T = [p_0(x) \ p_1(x) \ \cdots \ p_{\ell-1}(x)]$ and $e_\ell^T = [0 \ 0 \ \cdots \ 0 \ 1]$.

The following well-known result describes the important connection between orthogonal polynomials and Gauss quadrature. The proof can be found in [45, 73].

**Theorem 5.** Let $T_\ell = Y_\ell \Theta_\ell Y_\ell^T$ be a spectral factorization, where $Y_\ell Y_\ell^T = I$ and $\Theta_\ell = \text{diag}\{\theta^{(\ell)}_1, \theta^{(\ell)}_2, \ldots, \theta^{(\ell)}_\ell\}$ with $\theta^{(\ell)}_1 < \theta^{(\ell)}_2 < \cdots < \theta^{(\ell)}_\ell$. Then

1. The eigenvalues $\theta^{(\ell)}_i$ are the roots of orthonormal polynomial $p_\ell$, hence the Gauss nodes of the $\ell$-point Gauss quadrature rule with respect to $\mu$, where $\int d\mu(x) = 1$.

2. The corresponding Gauss weights $\eta^{(\ell)}_i$ are given by the first element of $(e_1^T Y_\ell e_i)^2$.

In addition to Gauss rules, there are related quadrature rules with prescribed nodes. If there is only one single node which is assigned at an endpoint of the interval of integration, the corresponding $\ell$-point quadrature rule

$$R_\ell f = \tilde{\eta}_1^{(\ell)} f(a) + \sum_{i=2}^\ell \tilde{\eta}_i^{(\ell)} f(\tilde{\theta}_i(\ell))$$

is known as a Gauss-Radau quadrature for (4). The degree of exactness for this rule is $2\ell - 2$. Moreover, if both of the endpoints are prescribed nodes, analogously the $\ell$-point quadrature rule

$$L_\ell f = \tilde{\eta}_1^{(\ell)} f(a) + \tilde{\eta}_2^{(\ell)} f(b) + \sum_{i=3}^\ell \tilde{\eta}_i^{(\ell)} f(\tilde{\theta}_i(\ell))$$

is known as a Gauss-Lobatto quadrature rule, whose degree of exactness is $2\ell - 3$. 

Contrary to the above quadrature rules which are mostly used in direct approximation of (4), there are some other variants of Gauss rules that are used to estimate the error of the approximation. The strategy is, by adopting two different quadrature rules on the same set of nodes, the difference between these two approximations can be used to estimate the error of the approximation. The \((2\ell + 1)\)-point \textit{Gauss - Kronrod} quadrature rule

\[
\mathcal{K}_{2\ell + 1} f = \sum_{i=1}^{\ell} \tilde{\eta}_i^{(\ell)} f(\theta_i^{(\ell)}) + \sum_{j=1}^{\ell+1} \tilde{\eta}_j^{(\ell)} f(\tilde{\theta}_j^{(\ell)})
\]

is a good example, which re-uses the Gauss nodes \(\{\theta_i\}_{i=1}^\ell\) from the \(\ell\)-point Gauss quadrature rule (6). The degree of exactness is at least \(3\ell + 1\); see [62].

2.3 Moments and Modified Moments

Moments and quadrature rules are intimately connected. If the moments with respect to a measure are known, they provide an alternative method to determine the quadrature rules.

For the measure \(\mu\), we assume all \textit{moments}

\[
m_j := \int x^j d\mu(x), \quad j = 0, 1, 2, \ldots
\]

exist and are bounded.

For the general moment method, substitute \(f(x) = x^j, j = 0, 1, \ldots, n - 1\) into (5). This yields the system of \textit{moment equations}

\[
\sum_{k=1}^{\ell} w_k x_k^j = m_j, \quad j = 0, 1, \ldots, n - 1,
\]
a system of linear algebraic equations with the Vandermonde coefficient matrix

\[
V_n = \begin{bmatrix}
1 & 1 & \ldots & 1 \\
x_1 & x_2 & \ldots & x_n \\
x_1^2 & x_2^2 & \ldots & x_n^2 \\
\vdots & \vdots & \ddots & \vdots \\
x_1^{n-1} & x_2^{n-1} & \ldots & x_n^{n-1}
\end{bmatrix}. \tag{13}
\]

The nodes \(x_i\) are distinct, thus the coefficient matrix is nonsingular. The system of equations (12) contains \(\ell\) equations and \(\ell\) unknowns, so it is solvable, either by straightforward Gauss elimination or by some other algorithm that utilizes the structure of the Vandermonde matrix. This gives the weights for the corresponding \(n\) nodes.

Another related concept is \textit{modified moments}, which are more general than moments.

They are defined as follows, assuming the measure is still \(\mu\),

\[
\tau_j := \int \pi_j(x) d\mu(x), \quad j = 0, 1, 2, \ldots , \tag{14}
\]

where the polynomials \(\{\pi_j\}\) satisfy a three-term recurrence relation

\[
\pi_{-1} = 0, \quad \pi_0(x) = 1, \\
\pi_{j+1}(x) = (x - a_j)\pi_j(x) - b_j\pi_{j-1}(x), \quad j = 1, 2, \ldots , \tag{15}
\]

with known coefficients \(a_j \in \mathbb{R}, b_j \geq 0\). If \(\forall j \in \mathbb{Z}, a_j = b_j = 0\), the modified moments become the ordinary moments. The set of the polynomials \(\{\pi_j\}\) satisfy the three-term recurrence relation. Therefore, they are orthogonal polynomials with respect to some measure \(\nu\) other than \(\mu\). A common choice of the auxiliary polynomials \(\{\pi_j\}\) are the Chebyshev polynomials.
Modified moments are often used instead of moments, because the transformation from moments to orthogonal polynomial recursion coefficients is usually ill-posed; see [36] for examples. In order to achieve stable and reliable results, we may use modified moments to generate the recursion coefficients stably [32].

There is a well-known algorithm - the modified Chebyshev algorithm - that takes the first $2n$ modified moments and the first $2n - 1$ coefficients $a_j, b_j, j = 0, 1, \ldots, 2n - 2$, and generates the first $n$ recursion coefficients $\alpha_j, \beta_j$ in (7); see [33, 35, 40, 67, 75] for the detailed algorithm.

It is not the main purpose of this dissertation to utilize the moment equations (ordinary or modified) to get the quadrature nodes and weights. We introduce the moments here for the better understanding of the fact that there is a one-to-one mapping between moments and recursion coefficients.
CHAPTER 3

Matrices, Polynomials and Quadrature

3.1 The Problem

The need to compute expressions of the form \( u^T f(A)v \), where \( A \) is a large square matrix, \( u \) and \( v \) are vectors, and \( f \) is a function, arises in many applications, including in network analysis [8, 9, 29, 30], quantum chromodynamics [6], and the solution of linear discrete ill-posed problems [63].

The aim of this chapter is to summarize available and introduce the new methods for the approximation of expressions of the form

\[
 u^T f(A)v, \tag{16}
\]

where \( A \in \mathbb{R}^{m \times m} \) is a large Hermitian or non-Hermitian matrix, \( u, v \in \mathbb{R}^m \) are vectors, and the superscript \( T \) denotes transposition.

In the 1970’s, Gene Golub [39] and collaborators first developed elegant iterative methods to produce good approximations of \( u^T f(A)v \) in a few steps, where \( A \) is a real symmetric positive definite matrix, \( u \) and \( v \) are given vectors and \( f \) is some smooth (possibly \( \mathbb{C}^\infty \)) function on a given interval of the real line. The method replaces (16) by a quadrature rule as in Section 2.2.

3.2 Quadrature Rules for Hermitian Functionals

We assume in this section, unless explicitly stated otherwise, that \( A \in \mathbb{R}^{m \times m} \) is Hermitian, and we let \( v = u \). Introduce the spectral factorization of \( A \), which can be
written as

\[ A = Q\Lambda Q^T, \]

where \( Q \) is an orthonormal matrix whose columns are normalized eigenvectors of \( A \) and \( \Lambda \) is a diagonal matrix whose diagonal elements are the eigenvalues \( \lambda_i \) which we order as

\[ \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n. \]

By definition, we have

\[ f(A) = Q f(\Lambda) Q^T. \]

Therefore,

\[
\begin{align*}
    u^T f(A) u &= u^T Q f(\Lambda) Q^T u \\
    &= \alpha^T f(\Lambda) \alpha \\
    &= \sum_{i=1}^{n} f(\lambda_i) \alpha_i^2.
\end{align*}
\]

This last sum can be considered a Riemann-Stieltjes integral

\[
\mathcal{I} f = u^T f(A) u = \int_{a}^{b} f(x) d\mu(x), \quad (17)
\]

where the measure \( \mu \) is piecewise constant and defined by

\[
\mu(x) = \begin{cases} 
0, & \text{if } x < a = \lambda_1 \\
\sum_{j=1}^{i} \alpha_j^2, & \text{if } \lambda_i \leq x < \lambda_{i+1} \\
\sum_{j=1}^{n} \alpha_j^2, & \text{if } b = \lambda_n \leq x
\end{cases}
\]

A way to obtain estimates and bounds for the Stieltjes integrals (17) is to use Gauss, Gauss-Radau and Gauss-Lobatto quadrature formulas mentioned in Section 2.2, see [22, 38, 39].
Let us recall briefly how the nodes and weights are obtained in Gauss quadrature rules. Introduce the inner product
\[
\langle f, g \rangle := \int f(x)g(x)\,d\mu(x)
\]
and let \( \{p_j\}_{j=0}^{\infty} \) denote a family of orthonormal polynomials associated with this inner product. Thus, \( p_j \) is of degree \( j \) with positive leading coefficient and such that
\[
\langle p_j, p_k \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}.
\]
The \( p_j \) satisfy a recursion relation of the form
\[
\beta_{j+1} p_{j+1}(x) = (x - \alpha_j) p_j(x) - \beta_j p_{j-1}(x), \quad j = 0, 1, 2, \ldots ,
\]
where \( p_{-1}(x) := 0 \), \( p_0(x) := 1 \), and \( \beta_0 := 0 \). The recursion coefficients \( \alpha_j \) are real and the coefficients \( \{\beta_j\}_{j \geq 1} \), can be chosen real and nonnegative.

Recall Theorem 5 which shows that the \( \ell \)-point Gauss quadrature rule for approximating (17) can be written as
\[
G_{\ell} f = \|u\|^2 e_1^T f(T_{\ell}) e_1,
\]
where \( T_{\ell} \) is the symmetric tridiagonal matrix defined by (7).

Introduce for \( 0 \leq r < \ell \) the reverse matrices
\[
\tilde{T}_{\ell-r,r} = \begin{bmatrix}
\alpha_{\ell-1} & \beta_{\ell-1} & 0 \\
\beta_{\ell-1} & \alpha_{\ell-2} & \beta_{\ell-2} & \ddots & \ddots & \ddots \\
& \ddots & \ddots & \ddots & \ddots & \ddots \\
\beta_{r+2} & \alpha_{r+1} & \beta_{r+1} & 0 & \beta_{r+1} & \alpha_r
\end{bmatrix} \in \mathbb{R}^{(\ell-r) \times (\ell-r)}
\]
as well as the concatenated symmetric tridiagonal matrices

\[
\hat{T}_{2\ell+1-r,r} = \begin{bmatrix}
T_\ell & \beta_\ell e_\ell & 0 \\
\beta_\ell e_\ell^T & \alpha_\ell & \beta_{\ell+1} e_1 \\
0 & \beta_{\ell+1} e_1 & \tilde{T}_{-r,r}
\end{bmatrix} \in \mathbb{R}^{(2\ell+1-r) \times (2\ell+1-r)}.
\]  

(21)

Spalević [70, 71] considered the latter matrix for \( r = 0 \) and referred to the associated quadrature rule

\[
\hat{G}_{2\ell+1} f = e_1^T f(\hat{T}_{2\ell+1,0}) e_1
\]

(22)
as a generalized averaged Gauss formula. Substituting the spectral factorization of \( \hat{T}_{2\ell+1,0} \) into (22) shows that this quadrature rule has \( 2\ell+1 \) nodes. Special cases of this quadrature rule for Laguerre and Hermite measures have previously been considered by Ehrich [25], but without defining the associated matrices (21) for \( r = 0 \). We provide some properties of the matrix \( \hat{T}_{2\ell+1,0} \) and the corresponding quadrature rule.

**Proposition 1.** The quadrature rule (22) is exact for all \( f \in \mathbb{P}_{2\ell+2} \). If the measure \( d\mu \) is symmetric with respect to the origin, then the quadrature rule (22) is exact for all \( f \in \mathbb{P}_{2\ell+3} \). The nodes of the Gauss rule (19) form a subset of the nodes of the rule (22).

**Proof.** First assume that \( d\mu \) is a general, not necessarily symmetric, measure. Consider the Chebyshev algorithm described in, e.g., [37, Section 2.1.7]. It determines recursion coefficients \( \alpha_j \) and \( \beta_j \) of orthogonal polynomials from the moments (11). More precisely, the entries \( \alpha_0, \alpha_1, \ldots, \alpha_\ell \) and \( \beta_1, \beta_2, \ldots, \beta_{\ell+1} \) of (21) are determined by the moments \( m_0, m_1, \ldots, m_{2\ell+2} \) in the order \( \alpha_0, \beta_1, \alpha_1, \beta_2, \ldots, \beta_\ell, \alpha_\ell, \beta_{\ell+1} \). We conclude that the quadrature rule (22) is exact for (at least) all polynomials in \( \mathbb{P}_{2\ell+2} \). When the measure \( d\mu \) is symmetric with respect to the origin, all diagonal entries of the matrix (21) vanish.
Therefore, the \((\ell + 2)\)nd diagonal entry of (21) can be thought of having been determined by the moments \(m_0, m_1, \ldots, m_{2\ell+3}\). If follows that the quadrature rule (22) is exact for all \(f \in \mathbb{P}_{2\ell+3}\).

The leading and trailing principal \(\ell \times \ell\) submatrices of \(\hat{T}_{2\ell+1,0}\) have the same eigenvalues. By considering the characteristic polynomial for \(\hat{T}_{2\ell+1,0}\) and expanding the determinant along row \(\ell + 1\), one can show that the spectrum of the leading principal submatrix \(T_\ell\) of \(\hat{T}_{2\ell+1,0}\) is a subset of the spectrum of the latter matrix; see [62, Lemma 1] for details.

The property that the nodes of the Gauss rule (19) form a subset of the nodes of the rule (22) can be utilized in computations. Calvetti et al. [13] show how a divide-and-conquer method described in [10] can be used efficiently in the context of the computation of the nodes and weights of Gauss–Kronrod quadrature rules. The same approach can be applied to evaluate the nodes and weights of the rule (22). We note, however, that (22) can be evaluated for many functions \(f\) of interest without computing the spectral factorization of \(\hat{T}_{2\ell+1}\); see Higham [53] for algorithms and analysis of many methods for evaluating matrix functions with matrices of small to moderate size.

The actual accuracy achieved with the quadrature rule (22) may be higher than suggested by the above proposition. Spalević [70] showed that if the recursion coefficients \(\alpha_j\) and \(\beta_j\) approach limits when \(j\) increases, then the quadrature rules (22) integrate all polynomials in \(\mathbb{P}_{3\ell+1}\) nearly exactly.

We are also interested in the quadrature rules

\[
\hat{G}_{2\ell+1-r,r}f = e_1^T f(\hat{T}_{2\ell+1-r,r})e_1
\]  

(23)
for $1 \leq r < \ell$. The following result can be shown in the same manner as for Proposition 1.

**Corollary 1.** The quadrature rules (23), for $1 \leq r < \ell$, are exact for all $f \in \mathbb{P}_{2\ell+2}$. If the measure $d\mu$ is symmetric with respect to the origin, then they are exact for all $f \in \mathbb{P}_{2\ell+3}$.

We turn to the problem of computing approximations of matrix functions (16) with a large symmetric matrix $A \in \mathbb{R}^{m \times m}$. Assume that $v = u$. The computation of the entries of the matrix (21) requires $\ell + 1$ steps. We therefore describe the algorithm for $\ell + 1$ steps.

The Lanczos algorithm can be viewed as a simplified Arnoldi’s algorithm that applies to Hermitian matrices. It generates a similar tridiagonal matrix.

Given $A \in \mathbb{R}^{m \times m}$ a Hermitian matrix and an initial vector $u \in \mathbb{R}^m$, the Hermitian Lanczos algorithm is defined by the following recursion Algorithm 3.1.

**Algorithm 3.1 The Hermitian Lanczos algorithm**

1. Initialization: $u_1 := u/\|u\|; \ u_0 := 0; \ \beta_0 := 0;$
2. Iterate: for $j = 1, 2, \ldots, \ell$ do
   
   $w := Au_j - \beta_j u_{j-1}$;
   
   $\alpha_{j-1} := w^* u_j$;
   
   $w := w - \alpha_{j-1} u_j$;
   
   $\beta_j := \|w\|$;
   
   $u_{j+1} := w/\beta_j$;

   end $j$

It is well known that the Lanczos vectors $u_i$ form orthonormal bases of the successive Krylov subspaces associated with the matrix $A$ and the initial vector $u$, which are defined
by

\[ K_\ell(A, u) = \text{span}\{u, Au, \ldots, A^\ell u\}. \]

Therefore, it is equivalent to say that

\[ \text{span}\{u_1, \ldots, u_\ell\} = K_\ell(A, u). \]

Define the matrix \( U_\ell = [u_1, \ldots, u_\ell] \in \mathbb{R}^{\ell \times \ell} \) with orthonormal columns determined by Algorithm 3.1 and \( T_\ell \) is defined in (7). If \( \ell = m \), then we have the complete recursion and obtain the tridiagonal and Hermitian matrix that is similar to \( A \), i.e.,

\[ T_m = U_m^T A U_m. \]  \hspace{1cm} (24)

But usually we stop after \( \ell \ll m \) steps of the recursion to obtain an approximation,

\[ AU_\ell = U_\ell T_\ell + \beta_\ell u_{\ell+1} e_\ell^T. \]  \hspace{1cm} (25)

There is some possibility that the recursion will break down in the middle of the recursion. This breakdown happens only if \( Au_j \) is in the range of \( U_j \), which causes \( \beta_j = 0 \) and the algorithm cannot be continued because the denominator is zero. Luckily, it is very rare and even if it happens, the algorithm can still continue by defining \( u_{j+1} \) to be an arbitrary unit length vector orthogonal to the range of \( U_j \). Furthermore, the truth is, this kind of breakdown is a good sign, because it indicates we have obtained the factorization \( AU_j = U_j T_j \), and inserting the spectral decomposition \( T_j = Y_j D_j Y_j^T \), we have \( A(U_j Y_j) = (U_j Y_j) D_j \). That is, we have successfully obtained the eigenvalues and eigenvectors.

Introduce the matrix \( U_{\ell+1} = [u_1, \ldots, u_{\ell+1}] \in \mathbb{R}^{m \times (\ell+1)} \) with orthonormal columns determined by Algorithm 3.1, and let the scalars \( \alpha_0, \alpha_1, \ldots, \alpha_\ell \) and \( \beta_1, \beta_2, \ldots, \beta_\ell \), also
computed by the algorithm, define the symmetric tridiagonal matrix

\[
T_{\ell+1} = \begin{bmatrix}
\alpha_0 & \beta_1 & 0 \\
\beta_1 & \alpha_1 & \beta_2 \\
& \ddots & \ddots & \ddots \\
& & \beta_{\ell-1} & \alpha_{\ell-1} & \beta_\ell \\
0 & & \beta_\ell & \alpha_\ell
\end{bmatrix} \in \mathbb{R}^{(\ell+1) \times (\ell+1)}.
\] (26)

Thus, the matrix (7) is the leading $\ell \times \ell$ principal submatrix of $T_{\ell+1}$. Analogously to (25), we have

\[
AU_{\ell+1} = U_{\ell+1}T_{\ell+1} + \beta_{\ell+1}u_{\ell+1}e_{\ell+1}^T.
\] (27)

The entries of the matrices (21) and (26) can be determined from (27). We recall that the latter matrix determines an $(\ell + 1)$-point Gauss quadrature rule that is exact for all $f \in \mathbb{P}_{2\ell+1}$, while the quadrature rule associated with (21) is exact at least for all $f \in \mathbb{P}_{2\ell+2}$.

In each step of Algorithm 3.1, one matrix-vector product with the matrix $A$ is evaluated. When $A$ is large, this is the dominant computational work required by the algorithm. Therefore the computation of the rules (23) and the Gauss rule $G_{\ell+1}$ requires essentially the same computational work. Our interest in the former rules stems from the fact that they are exact for a larger class of polynomials than the latter.

We remark that Saad [66] proposed an interesting approach, referred to as a “corrected scheme”, to improve the accuracy when approximating the matrix exponential $\exp(A)u$ by a Krylov subspace method. The corrected scheme also can be applied to improve Krylov subspace approximations of expressions of the form $u^T \exp(A)u$. Similarly to our method, the corrected scheme uses the vector $u_{\ell+1}$ and coefficient $\beta_\ell$ in the Lanczos
decomposition (25). One row and one column are appended to the symmetric tridiagonal matrix \( T_\ell \) in (25) to obtain a non-Hermitian matrix. Hence, Saad’s corrected scheme and our approach are different. Moreover, the corrected scheme is only derived for the exponential function and uses properties of this function.

In many applications, the number of steps \( \ell \) can be chosen fairly small. Thus the loss of orthogonality of the vectors \( u_1, u_2, \ldots, u_{\ell+2} \) caused by propagated round-off errors introduced during the execution of Algorithm 3.1 is no longer a problem. When \( \ell \) is not small, the vectors \( u_j \) may have to be reorthogonalized to secure accuracy of the computed quadrature rule.

In the Lanczos decomposition (27), the remainder \( AU_{\ell+1} - U_{\ell+1}T_{\ell+1} \) is of rank one. The following result describes a decomposition involving the tridiagonal matrix (21) in which the remainder is of rank two. The decomposition uses the permutation

\[
P_\ell = [e_\ell, e_{\ell-1}, \ldots, e_1] \in \mathbb{R}^{\ell \times \ell}.
\]

**Theorem 6.** Let \( \tilde{U}_{2\ell+1} = [U_{\ell+1}, U_\ell P_\ell] \in \mathbb{R}^{m \times (2\ell+1)} \), where \( P_\ell \) is given by (28). Then \( A\tilde{U}_{2\ell+1} - \tilde{U}_{2\ell+1}\tilde{T}_{2\ell+1,0} \) is of at most rank two.

**Proof.** Introduce the matrix \( \tilde{U}_\ell = U_\ell P_\ell \) and note that \( \tilde{T}_{\ell,0} = P_\ell T_\ell P_\ell \). We obtain from (25) that

\[
A\tilde{U}_\ell = AU_\ell P_\ell = (U_\ell T_\ell + g_\ell e_\ell^T)P_\ell
\]

\[
= U_\ell P_\ell T_\ell P_\ell + g_\ell e_\ell^T P_\ell = \tilde{U}_\ell \tilde{T}_{\ell,0} + g_\ell e_1^T.
\]

It follows that

\[
A[U_{\ell+1}, \tilde{U}_\ell] = [U_{\ell+1}T_{\ell+1}, \tilde{U}_\ell \tilde{T}_{\ell,0}] + [g_{\ell+1}e_{\ell+1}^T, g_\ell e_1^T].
\]
Moreover,

\[
[U_{\ell+1}, \tilde{U}_{\ell+1}] T_{2\ell+1,0} = [U_{\ell+1} T_{\ell+1} + \beta_{\ell+1} u_{\ell+1} e_{\ell+1}^T, \tilde{U}_{\ell} \tilde{T}_{\ell-1,0} + \beta_{\ell+1} u_{\ell+1} e_{1}^T].
\]

Therefore,

\[
A[U_{\ell+1}, \tilde{U}_{\ell}] - [g_{\ell+1} e_{\ell+1}^T, g e_1^T] = [U_{\ell+1}, \tilde{U}_{\ell}] T_{2\ell+1,0} - [\beta_{\ell+1} u_{\ell+1} e_{\ell+1}^T, \beta_{\ell+1} u_{\ell+1} e_1^T].
\]

Rearranging the terms, we obtain

\[
A\hat{U}_{2\ell+1} = \hat{U}_{2\ell+1} T_{2\ell+1,0} + [(g_{\ell+1} - \beta_{\ell+1} u_{\ell+1}) e_{\ell+1}^T, (g - \beta_{\ell+1} u_{\ell+1}) e_1^T],
\]

which shows the theorem.

We assumed above that \( v = u \) in (16). This restriction can be removed by writing (16) in the form

\[
u^T f(A) v = \frac{1}{4}(u + v)^T f(A)(u + v) - \frac{1}{4}(u - v)^T f(A)(u - v).
\]

Each term in the right-hand side can be approximated by quadrature rules in the manner described above. Since there are two terms, the work is doubled compared with the situation when \( u = v \). The following section discusses another approach to approximate the functional in the left-hand side. This approach also can be applied when the matrix \( A \) is nonsymmetric.

### 3.3 Quadrature Rules for Non-Hermitian Functionals

This section discusses quadrature rules that can be applied when \( A \in \mathbb{R}^{m \times m} \) is a general square matrix with spectral factorization \( A = SAS^{-1} \), where \( S \in \mathbb{C}^{m \times m} \) is nonsingular and \( \Lambda = \text{diag}[\lambda_1, \cdots, \lambda_m] \in \mathbb{C}^{m \times m} \). The vectors \( u, v \in \mathbb{R}^m \) in (16) are assumed to be scaled so that \( u^T v = 1 \).
Substituting the spectral factorization into (16) yields

$$\mathcal{F}(A) = u^T f(A)v = u^T S f(\Lambda) S^{-1} v = \sum_{j=1}^{n} f(\lambda_j) \mu_j \mu_j', \quad (30)$$

where $[\mu_1, \ldots, \mu_m] := u^T S$ and $[\mu'_1, \ldots, \mu'_m]^T := S^{-1} v$. The right-hand side of (30) can be written as a Stieltjes integral

$$\mathcal{I} f = \int f(x) d\mu(x), \quad (31)$$

with respect to the measure $d\mu(x)$ such that

$$d\mu(x) := \sum_{j=1}^{m} \delta(x - \lambda_j) \mu_j \mu'_j, \quad (32)$$

where $\delta(\cdot)$ denotes the Dirac $\delta$-function.

Introduce the bilinear form

$$\langle f, g \rangle := \int f(x) g(x) d\mu(x). \quad (33)$$

Generically, there are two sequences of polynomials $p_0, p_1, p_2, \ldots$ and $q_0, q_1, q_2, \ldots$ that are biorthonormal with respect to the bilinear form (33). These polynomials can be determined by applying the non-Hermitian Lanczos Algorithm 3.2 to $A$ with initial vectors $u, v$ such that $u^T v = 1$.

Given $A \in \mathbb{R}^{m \times m}$ a non-Hermitian matrix and two initial vectors $u \in \mathbb{R}^m$ and $v \in \mathbb{R}^m$, the non-Hermitian Lanczos algorithm is defined by the following Algorithm 3.2. The two initial vectors $u$ and $v$ determine two sequences of Krylov subspaces

$$\mathcal{K}_\ell(A, v) = \text{span}\{v, Av, ..., A^\ell v\}$$

and

$$\mathcal{K}_\ell(A^T, u) = \text{span}\{u, A^T u, ..., (A^T)^\ell u\}.$$
which can also be formed by the Lanczos vectors $u_i$ and $v_i$, i.e.,

$$\text{span}\{u_1, \ldots, u_\ell\} = \mathcal{K}_\ell(A^T, u)$$

and

$$\text{span}\{v_1, \ldots, v_\ell\} = \mathcal{K}_\ell(A, v).$$

**Algorithm 3.2** The non-Hermitian Lanczos algorithm

1: Initialization: $u_1 := u$; $v_1 := v$; $u_0 := 0$; $v_0 := 0$; $\beta_0 := 0$; $\gamma_0 := 0$
2: Iterate: for $j = 1, 2, \ldots, \ell + 1$ do
   
   $\alpha_{j-1} := u_j^T (Av_j - \gamma_{j-1}v_{j-1})$;
   $r := Av_j - \alpha_{j-1}v_j - \gamma_{j-1}v_{j-1}$;
   $s := A^T u_j - \alpha_{j-1} u_j - \beta_{j-1} u_{j-1}$;
   $\beta_j := |r^T s|^{1/2}$; $\gamma_j := r^T s / \beta_j$;
   $v_{j+1} := r / \beta_j$; $u_{j+1} := s / \gamma_j$;
   
end $j$

Introduce the matrices

$$U_{\ell+1} = [u_1, \ldots, u_{\ell+1}] \in \mathbb{R}^{m \times (\ell+1)}, \quad V_{\ell+1} = [v_1, \ldots, v_{\ell+1}] \in \mathbb{R}^{m \times (\ell+1)}.$$ 

The columns of these matrices are biorthonormal, i.e., $U_{\ell+1}^T V_{\ell+1} = I$. Define the tridiagonal matrix

$$T_{\ell+1} = \begin{bmatrix}
\alpha_0 & \gamma_1 & 0 \\
\beta_1 & \alpha_1 & \gamma_2 \\
& \ddots & \ddots & \ddots \\
& & \beta_{\ell-1} & \alpha_{\ell-1} & \gamma_\ell \\
0 & & & \beta_\ell & \alpha_\ell
\end{bmatrix} \in \mathbb{R}^{(\ell+1) \times (\ell+1)}. \quad (34)$$
If \( \ell = m \), we have the complete recursion and obtain the tridiagonal matrix \( T_m \) that is similar to \( A \), i.e.,

\[
T_m = U_m^T A V_m. \tag{35}
\]

Algorithm 3.2 is said to break down if some \( \beta_j \) vanishes. Breakdown is likely to take place when the matrix \( A \) is sparse and the initial vectors \( u \) and \( v \) only contain few nonvanishing entries. The breakdowns of this algorithm have been discussed by many authors; see [50]. They have been categorized into two cases: the first case is \( r = 0 \) or \( s = 0 \), and the second case is \( r \neq 0 \) and \( s \neq 0 \), but \( r^T s = 0 \). The first case is referred as premature breakdown and the second case is referred as serious breakdown.

For the first kind of breakdown, one may have to restart the algorithm with a pair of vectors that is biorthogonal to the vector sequences constructed so far, cf. [60, 61]. The second kind of breakdown is addressed in [51, 52]. A nice way to handle serious breakdown is described by Ye [76], who shows that one may replace one of the two Krylov subspaces by an appropriate sum of two Krylov subspaces.

We recall that a tridiagonal matrix is said to be unreduced if all its subdiagonal entries are nonvanishing. By the assumption of non-breakdown, all sub- and super-diagonal entries of \( T_{\ell+1} \) are nonvanishing. When we stop after \( \ell \ll m \) steps of the recursions, we have the decompositions

\[
AV_{\ell+1} = V_{\ell+1}T_{\ell+1} + \beta_{\ell+1}v_{\ell+2}e_{\ell+1}^T, \\
A^*U_{\ell+1} = U_{\ell+1}T_{\ell+1}^T + \gamma_{\ell+1}u_{\ell+2}e_{\ell+1}^T. \tag{36}
\]

Moreover, the recursion formulas of Algorithm 3.2 show that

\[
u_{j+1} = q_j(A^T)u_1, \quad v_{j+1} = p_j(A)v_1, \quad 0 \leq j \leq \ell + 1, \tag{37}\]
for some polynomials \( p_j \) and \( q_j \) of degree \( j \). It follows from the definition of the measure in (33) that

\[
\langle q_j, p_k \rangle = u^T q_j(A) p_k(A) v = u^T_{j+1} v_{k+1} = \begin{cases} 1 & j = k, \\ 0 & j \neq k. \end{cases}
\]

This shows that the Lanczos polynomials (37), which are implicitly defined by Algorithm 3.2, are biorthonormal polynomial with respect to the inner product (33). In particular, the recursion coefficients determined by the algorithm are recursion coefficients for the biorthonormal polynomials. It follows from (37) that the \( p_j \) and \( q_j \) satisfy the same recursion relations as the vectors \( u_j \) and \( v_j \), i.e.,

\[
\beta_{j+1} p_{j+1}(x) = (x - \alpha_j) p_j(x) - \gamma_j p_{j-1}(x), \quad j = 0, 1, \ldots, \ell + 1.
\]

\[
\gamma_{j+1} q_{j+1}(x) = (x - \alpha_j) q_j(x) - \beta_j q_{j-1}(x), \quad j = 0, 1, 2, \ldots, \ell + 1.
\]

Define the row vectors

\[
\tilde{p}_\ell(x) = [p_0(x), \ldots, p_{\ell-1}(x)], \quad \tilde{q}_\ell(x) = [q_0(x), \ldots, q_{\ell-1}(x)],
\]

and let \( T_\ell \) be the leading \( \ell \times \ell \) principal submatrix of (34), i.e.,

\[
T_\ell = \begin{bmatrix}
\alpha_0 & \gamma_1 & 0 \\
\beta_1 & \alpha_1 & \gamma_2 \\
\vdots & \ddots & \ddots \\
\beta_{\ell-2} & \alpha_{\ell-2} & \gamma_{\ell-1} \\
0 & \beta_{\ell-1} & \alpha_{\ell-1}
\end{bmatrix}.
\]

The recurrence relations (38) for \( 0 \leq j < \ell \) can be expressed as

\[
x \tilde{p}_\ell(x) = \tilde{p}_\ell(x) T_\ell + \beta_{q}_\ell(x) e_\ell^T,
\]

\[
x \tilde{q}_\ell(x) = \tilde{q}_\ell(x) T_\ell^T + \gamma_{q}_\ell(x) e_\ell^T.
\]
This shows that the zeros of $p_\ell$ are eigenvalues of $T_\ell$. Conversely, if $x$ is an eigenvalue of $T_\ell$, then $p_\ell(x) = 0$. We will assume that $T_\ell$ has $\ell$ distinct eigenvalues. Ammar et al. [1, Proposition 2.2] show that this is the case if and only if $T_\ell$ is diagonalizable (since the matrix is unreduced).

**Proposition 2.** Let the matrix (39) be diagonalizable. Then

$$G_\ell f = e_1^T f(T_\ell)e_1$$

is an $\ell$-point Gauss quadrature rule associated with the measure $d\mu(x)$ in (33), i.e.,

$$G_\ell f = I f \quad \forall f \in \mathbb{P}_{2\ell-1}$$

for the integral operator (31).

**Proof.** A proof can be found, e.g., in [18]. Our proof is simpler; it is a simplification of a proof presented in [29] for quadrature rules associated with the nonsymmetric block Lanczos process. We provide a proof for completeness.

The assumption that Algorithm 3.2 does not break down secures that the matrix $T_\ell$ exists, and the assumption that $T_\ell$ is diagonalizable guarantees that all eigenvalues are distinct. Substituting the spectral factorization of $T_\ell$ into (40) shows that $G_\ell$ is an $\ell$-point quadrature rule with real or complex conjugate nodes.

It follows by induction over the degree of the polynomials $p$ and $q$ that for fixed $\ell$,

$$p(A)v_1 = V_\ell p(T_\ell)e_1, \quad p \in \mathbb{P}_{\ell-1},$$

$$V_\ell^T q(A^T)u_1 = q(T_\ell^T)e_1, \quad q \in \mathbb{P}_\ell.$$

Let $f \in \mathbb{P}_{2\ell-1}$. Factoring $f = qp$, where $p \in \mathbb{P}_{\ell-1}$ and $q \in \mathbb{P}_\ell$, yields

$$I f = u^T q(A)p(A)v = u^T q(A)V_\ell p(T_\ell)e_1 = e_1^T q(T_\ell)p(T_\ell)e_1 = e_1^T f(T_\ell)e_1 = G_\ell f.$$
Introduce for $0 \leq r < \ell$ the reverse matrices

\[
\tilde{T}_{\ell-r,r} = \begin{bmatrix}
\alpha_{\ell-1} & \gamma_{\ell-1} & 0 \\
\beta_{\ell-1} & \alpha_{\ell-2} & \gamma_{\ell-2} & & \\
& \ddots & \ddots & \ddots & \\
& & \beta_{r+2} & \alpha_{r+1} & \gamma_{r+1} \\
0 & & & \beta_{r+1} & \alpha_r
\end{bmatrix} \in \mathbb{R}^{(\ell-r) \times (\ell-r)}
\] (41)

as well as the concatenated tridiagonal matrices

\[
\hat{T}_{2\ell+1-r,r} = \begin{bmatrix}
T_\ell & \gamma_\ell e_\ell & 0 \\
\beta_\ell e_\ell^T & \alpha_\ell & \gamma_{\ell+1} e_{1\ell}^T & \\
0 & \beta_{\ell+1} e_1 & \tilde{T}_{\ell-r,r}
\end{bmatrix} \in \mathbb{R}^{(2\ell+1-r) \times (2\ell+1-r)}.
\] (42)

The matrices (41) and (42) are analogues of the matrices (20) and (21) of Section 3.2.

We can use the matrix (42) to define the quadrature rule

\[
\hat{G}_{2\ell+1} f = e_1^T f (\hat{T}_{2\ell+1,0}) e_1,
\] (43)

which is a generalization of the rule (22) to measures $d\mu$ in (31). Another generalization is defined at the end of this section.

**Proposition 3.** Let all leading principal sub-matrices of $\hat{T}_{2\ell+1,0}$ be diagonalizable. Then (43) is an $(2\ell + 1)$-point quadrature rule that is exact for all $f \in \mathbb{P}_{2\ell+2}$. If the measure is symmetric with respect to the origin, then the quadrature rule (43) is exact for all $f \in \mathbb{P}_{2\ell+3}$. The nodes of the Gauss quadrature rule (40) form a subset of the nodes of rule (43).
Proof. Assume first that $d\mu$ in (33) is a complex-valued measure determined by (32), without any additional symmetry properties. Consider the matrix $T_{\ell+2}$, which analogous to the matrix (39). We will assume this matrix to be diagonalizable. Then it follows from Proposition 2 that

$$G_{\ell+2}f = e_1^T f(T_{\ell+2})e_1$$

is an $(\ell + 2)$-point Gauss quadrature rule. This rule is exact for all polynomials in $\mathbb{P}_{2\ell+3}$. If the last diagonal entry of $T_{\ell+2}$ is changed, then the resulting quadrature rule is exact for all polynomials in $\mathbb{P}_{2\ell+2}$. This is a consequence of the recursions of the Chebyshev algorithm, which from the moments $m_j = \int x^j d\mu(x), j = 0, 1, \ldots, 2\ell + 2$, computes all entries except the last diagonal entry of $T_{\ell+2}$; the last diagonal entry of $T_{\ell+2}$ is determined by $m_{2\ell+3}$. A modified Chebyshev algorithm is described in [14]; the (standard) Chebyshev algorithm is a special case. We remark that the moments determine the products $\beta_j \gamma_j$ of the off-diagonal entries. The scaling of the $\beta_j$ is arbitrary. In the Chebyshev algorithm described in [14], the $\beta_j$ are set to unity.

When the measure $d\mu$ is symmetric about the imaginary axis in the complex plane, all recursion coefficients $\alpha_j$ vanish. The leading $(\ell + 2) \times (\ell + 2)$ principal submatrix of (42) then is $T_{\ell+2}$. This means that the quadrature rule (43) is exact for all $f \in \mathbb{P}_{2\ell+3}$.

The fact that the nodes of the quadrature rule (40) form a subset of the nodes of rule (43) can be shown similarly as outlined in the proof of Proposition 1.

Ammar et al. [1] discuss the computation of nodes and weights of Gauss–Kronrod rules associated with a real nonsymmetric tridiagonal matrix. Some computational aspects carry over to the determination of the nodes and weights of the quadrature rule.
However, as already mentioned in Section 3.2, the evaluation of (43) can be done efficiently for many functions of interest without using the spectral factorization of $\hat{T}_{2\ell+1}$.

Similarly as in Section 3.2, we define the quadrature rules

$$\hat{G}_{2\ell+1-r,r}f = e_1^T f(\hat{T}_{2\ell+1-r,r})e_1, \quad 1 \leq r < \ell.$$  \hspace{1cm} (44)

**Corollary 2.** Under the conditions of Proposition 2, the quadrature rules (43) and (44) are exact for the same class of functions.

**Proof.** The leading $(\ell + 2) \times (\ell + 2)$ principal submatrix of the matrix $\hat{T}_{2\ell+1-r,r}$ in (44) is the same as the corresponding leading principal submatrix of the rule (43). Therefore, the proof of Proposition 2 carries over to the quadrature rules (44). \hfill \Box

The following result yields some properties of decompositions that involve the matrix (42).

**Theorem 7.** $T_\ell$ and $\tilde{T}_{\ell,0}$ share same eigenvalues. Let $T_\ell = S_\ell \Lambda_\ell S_\ell^{-1}$ and $\tilde{T}_{\ell,0} = \tilde{S}_\ell \Lambda_\ell \tilde{S}_\ell^{-1}$ be spectral factorization, let $V_{\ell+1}$ and $U_{\ell+1}$ be determined by the decompositions (36), and let $V_\ell$ and $U_\ell$ consist of the first $\ell$ columns of $V_{\ell+1}$ and $U_{\ell+1}$, respectively. Define the matrices $\tilde{V}_\ell = V_\ell S_\ell \tilde{S}_\ell^{-1}$ and $\tilde{U}_\ell = U_\ell S_\ell \tilde{S}_\ell^{-1}$. Define $\hat{V}_{2\ell+1} = [V_{\ell+1}, \tilde{V}_\ell]$ and $\hat{U}_{2\ell+1} = [U_{\ell+1}, \tilde{U}_\ell]$. Then the matrices $AV_{2\ell+1} - \hat{V}_{2\ell+1} \hat{T}_{2\ell+1,0}$ and $A\hat{U}_{2\ell+1} - \hat{U}_{2\ell+1} \hat{T}_{2\ell+1,0}$ have rank at most two.

**Proof.** The matrix (41) with $r = 0$ can be written as $\tilde{T}_{\ell,0} = \tilde{S}_\ell S_\ell^{-1} T_\ell \tilde{S}_\ell S_\ell^{-1}$. Using this property as well as the relation $T_\ell = S_\ell \tilde{S}_\ell^{-1} T_{\ell,0} \tilde{S}_\ell S_\ell^{-1}$, we obtain, with $h_j = \beta_j v_{j+1}$ for
\( j \in \{ \ell, \ell + 1 \} \) that
\[
A \tilde{V}_\ell = AV_\ell \tilde{S}_\ell^{-1} = (V_\ell T_\ell + h_\ell e_\ell^T)S_\ell \tilde{S}_\ell^{-1}
\]
\[
= (V_\ell S_\ell \tilde{S}_\ell^{-1} T_{\ell,0} \tilde{S}_\ell S_\ell^{-1} + h_\ell e_\ell^T)S_\ell \tilde{S}_\ell^{-1}
\]
\[
= V_\ell S_\ell \tilde{S}_\ell^{-1} \tilde{T}_{\ell,0} + h_\ell e_\ell^T S_\ell \tilde{S}_\ell^{-1}
\]
\[
= \tilde{V}_\ell \tilde{T}_{\ell,0} + h_\ell e_\ell^T S_\ell \tilde{S}_\ell^{-1}.
\]

The above relation combined with (36) gives

\[
A[V_{\ell+1}, \tilde{V}_\ell] = [V_{\ell+1} T_{\ell+1}, \tilde{V}_\ell \tilde{T}_{\ell,0}] + [h_{\ell+1} e_{\ell+1}^T, h_\ell e_\ell^T S_\ell \tilde{S}_\ell^{-1}].
\]

Moreover,

\[
[V_{\ell+1}, \tilde{V}_\ell] \tilde{T}_{2\ell+1,0} = [V_{\ell+1} T_{\ell+1} + \beta_{\ell+1} v_\ell S_\ell \tilde{S}_\ell^{-1} e_{\ell+1}^T, \gamma_{\ell+1} v_\ell e_1^T + \tilde{V}_\ell \tilde{T}_{\ell,0}].
\]

Therefore,

\[
A[V_{\ell+1}, \tilde{V}_\ell] - [h_{\ell+1} e_{\ell+1}^T, h_\ell e_\ell^T S_\ell \tilde{S}_\ell^{-1}] = [V_{\ell+1}, \tilde{V}_\ell] \tilde{T}_{2\ell+1,0} - [\beta_{\ell+1} v_\ell S_\ell \tilde{S}_\ell^{-1} e_{\ell+1}^T, \gamma_{\ell+1} v_\ell e_1^T],
\]

and rearranging terms yields

\[
A \tilde{V}_{2\ell+1} = \tilde{V}_{2\ell+1} \tilde{T}_{2\ell+1,0} + [(h_{\ell+1} - \beta_{\ell+1} v_\ell S_\ell \tilde{S}_\ell^{-1}) e_{\ell+1}^T, h_\ell e_\ell^T S_\ell \tilde{S}_\ell^{-1} - \gamma_{\ell+1} v_\ell e_1^T].
\] (45)

We can express \( \hat{U}_{2\ell+1} A^T \) in a similar fashion. This shows the theorem.

A different extension of the Gauss rule (40) is obtained by replacing the matrix (41) in (42) by its transpose. This gives the matrix

\[
\tilde{T}'_{2\ell+1-r,r} = \begin{bmatrix}
T_\ell & \gamma_\ell e_\ell & 0 \\
\beta_\ell e_\ell^T & \alpha_\ell & \gamma_{\ell+1} e_1^T \\
0 & \beta_{\ell+1} e_1 & \tilde{T}_{\ell-r,r}^T
\end{bmatrix}
\] (46)
and the associated quadrature rule

\[ \mathcal{G}'_{2t+1-r,r}f = e_1^T f(\hat{T}'_{2t+1-r,r})e_1. \]

When \( r = 0 \), we define

\[ \mathcal{G}'_{2t+1}f = e_1^T f(\hat{T}'_{2t+1,0})e_1. \]  \hspace{1cm} (47)

Analogues of Proposition 2, Corollary 2, and Theorem 7 can be shown. In our experience, the quadrature rules (44) and (47) yield about the same accuracy. The performance of these rules is illustrated in Chapter 6.
CHAPTER 4

Matrices, Matrix Polynomials and Block Quadrature

In many situations, we want to approximate the value \( u^T f(A) v \) for several vectors \( u \) and \( v \). It is possible to apply the methods described in Chapter 3 for each vector pair \( \{u, v\} \). This chapter describes an alternative approach, based on the application of the block Lanczos algorithm.

There are many connections between this chapter and prior chapter. The problem we investigate in this chapter is the approximation of \( W^T f(A) V \), where \( W, V \in \mathbb{R}^{m \times k} \) are biorthogonal. We will describe the Hermitian and non-Hermitian block Lanczos algorithms.

4.1 The Problem

The aim of this chapter is to summarize available methods and introduce new ones for the approximation of expressions of the form

\[
W^T f(A) W, \tag{48}
\]

where \( A \in \mathbb{R}^{m \times m} \) is a large symmetric matrix, \( W \in \mathbb{R}^{m \times k} \) has a few orthonormal columns, i.e., \( 1 \leq k \ll m \), and \( f \) is a matrix function. We also consider expressions of the type

\[
W^T f(A) V, \tag{49}
\]

in which \( A \in \mathbb{R}^{m \times m} \) is a large possibly nonsymmetric matrix and the matrices \( W, V \in \mathbb{R}^{m \times k} \), with \( 1 \leq k \ll m \), are biorthogonal, i.e., \( W^T V = I_k \).
4.2 Block Quadrature Rules for Hermitian Matrices

First we discuss the approximation of expressions of the form (48) by block quadrature rules. The matrix $A \in \mathbb{R}^{m \times m}$ is assumed to be symmetric throughout this section and $W \in \mathbb{R}^{m \times k}$ has orthonormal columns with $1 \leq k \ll m$.

To justify the use of quadrature rules, we first show that the expression (48) can be written as a Stieltjes-type integral. This was first observed by Golub and Meurant [41]. A more recent and detailed discussion can be found in [43]; see also [30].

Consider the spectral factorization

$$A = Q\Lambda Q^T,$$  \hspace{1cm} (50)

where $Q \in \mathbb{R}^{m \times m}$ is orthogonal and $\Lambda = \text{diag}[\lambda_1, \ldots, \lambda_m] \in \mathbb{R}^{m \times m}$. Thus, the $\lambda_i$ are eigenvalues of $A$. Substituting (50) into (48) yields

$$W^T f(A)W = W^T Q f(\Lambda) Q^T W = \sum_{i=1}^{m} f(\lambda_i) \alpha_i \alpha_i^T = \int f(\lambda) \, d\sigma(\lambda) =: \mathcal{I} f,$$  \hspace{1cm} (51)

where $[\alpha_1, \ldots, \alpha_m] = W^T Q$ and $\sigma$ is a piece-wise matrix-valued distribution with the jump $\alpha_i \alpha_i^T \in \mathbb{R}^{k \times k}$ at the eigenvalue $\lambda_i$ of $A$. For future reference, we define the matrix moments

$$M_j := \mathcal{I}(\lambda^j), \quad j = 0, 1, 2, \ldots .$$  \hspace{1cm} (52)

In particular, $M_0 = I_k$.

Introduce the bilinear form

$$\langle f, g \rangle := \mathcal{I}(fg).$$

There is a sequence of matrix polynomials $p_0, p_1, p_2, \ldots$ that are orthonormal with respect
to this bilinear form, i.e.,

$$\langle p_i, p_j \rangle = \begin{cases} I_k & i = j, \\ O_k & i \neq j, \end{cases}$$

see, e.g., [24,41,43,69]. These polynomials satisfy a three-term recurrence relation of the form

$$\lambda p_{j-1}(\lambda) = p_j(\lambda)\Gamma_j + p_{j-1}(\lambda)\Omega_j + p_{j-2}(\lambda)\Gamma_{j-1}^T, \quad j = 1, 2, \ldots, \quad (53)$$

where the matrices $\Omega_j \in \mathbb{R}^{k \times k}$ are symmetric and the matrices $\Gamma_j \in \mathbb{R}^{k \times k}$ are upper triangular.

Define the matrix

$$P_{\ell}(\lambda) := [p_0(\lambda), \ldots, p_{\ell-1}(\lambda)] \in \mathbb{R}^{k \times k\ell}.$$ 

Then recursion relations (53) for the polynomials $p_0, p_1, \ldots, p_{\ell}$ can be expressed in the form

$$\lambda P_{\ell}(\lambda) = P_{\ell}(\lambda)J_{\ell} + p_{\ell}(\lambda)\Gamma_{\ell}E_{\ell}^T,$$

where $E_{\ell} = [O_k, \ldots, O_k, I_k]^T \in \mathbb{R}^{m \times k\ell}$ and

$$J_{\ell} = \begin{bmatrix} \Omega_1 & \Gamma_1^T & O \\ \Gamma_1 & \Omega_2 & \Gamma_2^T \\ \vdots & \vdots & \vdots \\ \Gamma_{\ell-1} & \Omega_{\ell} & \Gamma_{\ell-1}^T \\ O & \Gamma_{\ell-1} & \Omega_{\ell} \end{bmatrix} \in \mathbb{R}^{k\ell \times k\ell} \quad (54)$$

is block tridiagonal with symmetric diagonal blocks $\Omega_i \in \mathbb{R}^{k \times k}$ and upper triangular subdiagonal blocks $\Gamma_i \in \mathbb{R}^{k \times k}$. 

35
Golub and Meurant \cite{41, 43} show that

\[ G_\ell f = E_1^T f(J_\ell) E_1, \tag{55} \]

where \( E_1 = [I_k, O_k, \ldots, O_k]^T \in \mathbb{R}^{k\ell \times k}. \)

Introduce the spectral factorization \( J_\ell = Y_\ell \Theta_\ell Y_\ell^T \), where

\[ Y_\ell = [y_1^{(\ell)}, \ldots, y_{k\ell}^{(\ell)}] \in \mathbb{R}^{k\ell \times k\ell}, \quad Y_\ell^T Y_\ell = I_{k\ell}, \]

\[ \Theta_\ell = \text{diag}[^\theta_1^{(\ell)}, \ldots, \theta_{k\ell}^{(\ell)}] \in \mathbb{R}^{k\ell \times k\ell}. \]

Substituting this factorization into (55) yields

\[ G_\ell f = \sum_{i=1}^{k\ell} f(\theta_i^{(\ell)}) u_i^{(\ell)} (u_i^{(\ell)})^T, \tag{56} \]

where the vector \( u_i^{(\ell)} \in \mathbb{R}^k \) is made up of the first \( k \) elements of the normalized eigenvector \( y_i^{(\ell)}. \) Whether it is most advantageous to compute \( G_\ell f \) by evaluating (56) or by calculating (55) without computing the spectral factorization of \( J_\ell \) depends on the function \( f. \) For instance, the squaring and scaling algorithm is a convenient way to compute \( f(J_\ell) \) when \( f \) is the exponential function; see \cite{54}.

\( G_\ell \) can be interpreted as an \( \ell \)-block Gauss rule associated with the measure \( d\sigma, \) i.e.,

\[ G_\ell f = W^T f(A) W = \mathcal{I} f \quad \forall f \in \mathbb{P}_{2\ell-1}, \]

where \( \mathbb{P}_{2\ell-1} \) denotes the set of all polynomials of degree at most \( 2\ell - 1. \) Proofs can be found in \cite{24, 41, 43, 69}. A simple proof is provided in \cite[Section 5\]. While the matrix \( A \) is assumed to be so large that it is difficult to evaluate \( f(A) \), the number of steps \( \ell \) of the symmetric block Lanczos algorithm typically can be chosen small enough so that \( f(J_\ell) \) in (55) can be conveniently computed by one of the methods for evaluating functions of small to moderately sized matrices described by Higham \cite{53}. 

36
We are in a position to describe new block quadrature rules for the approximation of (48). Introduce for $1 \leq r < \ell$ the reverse symmetric block tridiagonal matrices

$$\tilde{J}_{\ell-r, r} = \begin{bmatrix}
\Omega_{\ell-1} & \Gamma_{\ell-2} & O \\
\Gamma_{\ell-2} & \Omega_{\ell-2} & \Gamma_{\ell-3}^T \\
\vdots & \ddots & \ddots \\
O & \Gamma_{r+1} & \Omega_{r+1} & \Gamma_r^T \\
\Gamma_r & O & \Omega_r
\end{bmatrix} \in \mathbb{R}^{k(\ell-r) \times k(\ell-r)} \quad (57)$$

as well as the concatenated symmetric tridiagonal matrices

$$\hat{J}_{2\ell-r, r} = \begin{bmatrix}
J_{\ell-1} & \Gamma_{\ell-1}^T E_{\ell-1} & O \\
\Gamma_{\ell-1} E_{\ell-1}^T & \Omega_{\ell} & \Gamma_{\ell}^T E_1^T \\
O & \Gamma_{\ell} E_1 & \tilde{J}_{\ell-r, r}
\end{bmatrix} \in \mathbb{R}^{k(2\ell-r) \times k(2\ell-r)}. \quad (58)$$

The latter matrices induce the block quadrature rules

$$\hat{G}_{2\ell-r, r} f = E_1^T f(\hat{J}_{2\ell-r, r}) E_1, \quad 1 \leq r < \ell. \quad (59)$$

Similarly as the block Gauss rule (55), these rules can be evaluated when $\ell$ steps of the block Lanczos method with initial block $W$ have been carried out; see below. The rules (59) are block versions of quadrature rules discussed in Chapter 3. The following theorem shows some properties of the block rules (59).

**Theorem 8.** The quadrature rules (59) are exact for all $f \in \mathbb{P}_{2\ell}$. If the measure $d\sigma$ is such that all diagonal blocks $\Omega_i$ are equal, then the rules are exact for all $f \in \mathbb{P}_{2\ell+1}$.

**Proof.** Fix $1 \leq r < \ell$ and assume first that $d\sigma$ is a general measure such that all matrix moments (52) exist. Our proof is based on the recursion formulas of the block Chebyshev
algorithm. A modified block Chebyshev algorithm is described in, e.g., [16]; the block Chebyshev algorithm is a special case. The latter algorithm determines recursion matrix coefficients \( \Omega_j \) and \( \Gamma_j \) for the orthogonal matrix polynomials (53) from the moments (52). More precisely, the submatrices \( \Omega_1, \ldots, \Omega_\ell \) and \( \Gamma_1, \ldots, \Gamma_\ell \) of (58) are determined by the matrix moments \( M_0, M_1, \ldots, M_{2\ell} \) in the order \( \Omega_1, \Gamma_1, \Omega_2, \Gamma_2, \ldots, \Gamma_{\ell-1}, \Omega_\ell, \Gamma_\ell \). Thus, the block quadrature rule matches the first \( 2\ell \) matrix moments. We conclude that the block quadrature rule (59) is exact for (at least) all matrix polynomials in \( \mathbb{P}_{2\ell} \).

Now let the measure \( d\sigma \) be such that all diagonal blocks \( \Omega_i \) are the same. Then the \((\ell + 1)\)st diagonal block entry of (58) can be thought of having been determined by the moments \( M_0, M_1, \ldots, M_{2\ell+1} \). If follows that the block quadrature rule (59) is exact for all \( f \in \mathbb{P}_{2\ell+1} \).

Block quadrature rules that are different from the rules (59) and match the same matrix moments also can be derived. Consider, for instance, the block quadrature rule

\[
\tilde{G}_{2\ell-r,r}^f = E_1^T f(\tilde{J}_{2\ell-r,r}) E_1, \quad 1 \leq r < \ell,
\]

(60)

where

\[
\tilde{J}_{2\ell-r,r}^f = \begin{bmatrix}
J_{\ell-1} & \Gamma_{\ell-1}^T E_{\ell-1} & O \\
\Gamma_{\ell-1} E_{\ell-1}^T & \Omega_\ell & \Gamma_\ell E_1^T \\
O & \Gamma_\ell E_1 & \tilde{J}_{\ell-r,r}^f
\end{bmatrix} \in \mathbb{R}^{k(2\ell-r) \times k(2\ell-r)}.
\]

(61)
and the matrix

\[ \tilde{J}_{r-r}^\ell = \begin{bmatrix}
\Omega_{\ell-1} & \Gamma_{\ell-2} & O \\
\Gamma_{\ell-2}^T & \Omega_{\ell-2} & \Gamma_{\ell-3} \\
\Gamma_{\ell-3}^T & \Omega_{\ell-3} & \Gamma_{\ell-4} \\
\vdots & \ddots & \ddots \\
\Gamma_{r+1}^T & \Omega_{r+1} & \Gamma_r \\
O & \Gamma_r^T & \Omega_r 
\end{bmatrix} \in \mathbb{R}^{k(\ell-r) \times k(\ell-r)} \]  

(62)

is obtained by interchanging the sub- and super-diagonal blocks of (57).

Similarly as the block quadrature rule (59), the rule (60) can be evaluated after \( \ell \) steps of the symmetric block Lanczos algorithm have been carried out. Moreover, we have the following result.

**Corollary 3.** The quadrature rules (60) are exact for all \( f \in P_{2\ell} \). If the measure \( d\sigma \) is such that all diagonal blocks \( \Omega_i \) are equal, then the rules are exact for all \( f \in P_{2\ell+1} \).

**Proof.** The result can be shown in the same way as Theorem 8.

Computed examples show the block quadrature rules (59) and (60) to yield essentially the same accuracy. We therefore in Chapter 6 only report results for one of these rules.

The matrix recursion coefficients in (53) are computed with the symmetric block Lanczos algorithm described, e.g., in [41,43]. Below we provide a modified Gram–Schmidt implementation. The initial block vector \( W \in \mathbb{R}^{m \times k} \) is assumed to have orthonormal columns.

Here \( W_{j+1} \Gamma_j := R_j \) denotes the computation of a reduced QR factorization such that \( W_{j+1} \in \mathbb{R}^{m \times k} \) has orthonormal columns and \( \Gamma_j \in \mathbb{R}^{k \times k} \) is upper triangular. When the matrix \( A \) is large and \( \ell \) is fairly small, which is the situation in most applications
Algorithm 4.1 The symmetric block Lanczos algorithm

1: Initialization: $W_0 := O \in \mathbb{R}^{m \times k}$, $\Gamma_0 := O_k$, $W_1 := W$;
2: Iterate: for $j = 1, 2, \ldots, \ell$ do
   \[ \tilde{W} := AW_j - W_{j-1}\Gamma_{j-1}^T; \]
   \[ \Omega_j := W_j^T \tilde{W}; \]
   \[ R_j := \tilde{W} - W_j \Omega_j; \]
   \[ W_{j+1} \Gamma_j := R_j; \]
end $j$

of interest, the dominant computational work in Algorithm 4.1 is the evaluation of the matrix-block vector products $AW_j$. The block Lanczos algorithm is said to break down at the $j^{th}$ step if $R_j$ is (numerically) rank deficient. In this case, the computations can be continued by replacing (numerically) linearly dependent columns of $W_{j+1}$ by arbitrary columns of unit length that are orthogonal to the ranges of the matrices $W_1, \ldots, W_j$, and then computing the QR factorization $W_{j+1} \Gamma_j = R_j$. The upper triangular matrix $\Gamma_j \in \mathbb{R}^{k \times k}$ so obtained is necessarily singular. For ease of exposition, the handling of breakdown is not included in Algorithm 4.1.

The above block Lanczos algorithm yields the decomposition
\[
A[W_1, \ldots, W_\ell] = [W_1, \ldots, W_\ell] J_\ell + W_{\ell+1} \Gamma_\ell E_\ell^T, \tag{63}
\]
The remainder $A[W_1, \ldots, W_\ell] - [W_1, \ldots, W_\ell] J_\ell$ is of rank at most $k$. The following result shows a decomposition involving the block tridiagonal matrix $\hat{J}_{2\ell-1,1}$ in which the remainder is of rank $2k$. The decomposition uses the block permutation matrix
\[
P_{\ell-1} = [E_{\ell-1}, \ldots, E_1] \in \mathbb{R}^{k(\ell-1) \times k(\ell-1)}. \tag{64}
\]

Theorem 9. Let $U_\ell = [W_1, \ldots, W_\ell]$ and $\hat{U}_{2\ell-1} = [U_\ell, U_{\ell-1} P_{\ell-1}] \in \mathbb{R}^{m \times k(2\ell-1)}$, where $P_{\ell-1}$
is given by (64). Then \( A\hat{U}_{2\ell-1} - \hat{U}_{2\ell-1}\hat{J}_{2\ell-1,1} \) is of rank at most \( 2k \).

**Proof.** Introduce the matrix \( \tilde{U}_{\ell-1} = U_{\ell-1}P_{\ell-1} \) and note that \( \tilde{J}_{\ell-1,1} = P_{\ell-1}J_{\ell-1}P_{\ell-1} \). We obtain from (63) that

\[
A\tilde{U}_{\ell-1} = AU_{\ell-1}P_{\ell-1} = (U_{\ell-1}J_{\ell-1} + W_{\ell}\Gamma_{\ell-1}E_{\ell-1}^T)P_{\ell-1}
\]

\[
= U_{\ell-1}P_{\ell-1}P_{\ell-1}J_{\ell-1}P_{\ell-1} + W_{\ell}\Gamma_{\ell-1}E_{\ell-1}^T
\]

\[
= \tilde{U}_{\ell-1}\tilde{J}_{\ell-1,1} + W_{\ell}\Gamma_{\ell-1}E_{1}^T.
\]

It follows that

\[
A[U_{\ell}, \tilde{U}_{\ell-1}] = [U_{\ell}J_{\ell}, \tilde{U}_{\ell-1}\tilde{J}_{\ell-1,1}] + [W_{\ell+1}\Gamma_{\ell}E_{\ell}^T, W_{\ell}\Gamma_{\ell-1}E_{1}^T].
\]

Moreover,

\[
[U_{\ell}, \tilde{U}_{\ell-1}]\tilde{J}_{2\ell-1,1} = [U_{\ell}J_{\ell} + W_{\ell-1}\Gamma_{\ell}E_{\ell}^T, \tilde{U}_{\ell-1}\tilde{J}_{\ell-1,1} + W_{\ell}\Gamma_{\ell}^T E_{1}^T].
\]

Therefore,

\[
A[U_{\ell}, \tilde{U}_{\ell-1}] - [W_{\ell+1}\Gamma_{\ell}E_{\ell}^T, W_{\ell}\Gamma_{\ell-1}E_{1}^T] = [U_{\ell}, \tilde{U}_{\ell-1}]\tilde{J}_{2\ell-1,1} - [W_{\ell-1}\Gamma_{\ell}E_{\ell}^T, W_{\ell}\Gamma_{\ell}^T E_{1}^T].
\]

Rearranging the terms, we obtain

\[
A\hat{U}_{2\ell-1} = \hat{U}_{2\ell-1}\hat{J}_{2\ell-1,1} + [(W_{\ell+1}\Gamma_{\ell} - W_{\ell-1}\Gamma_{\ell})E_{\ell}^T, (W_{\ell}\Gamma_{\ell-1} - W_{\ell}\Gamma_{\ell})E_{1}^T],
\]

which shows the theorem.

\[\square\]

### 4.3 Block Quadrature Rules for Non-Hermitian Matrices

We extend the discussion of the previous section to the situation when \( A \in \mathbb{R}^{m \times m} \) is nonsymmetric and the block vectors \( V, W \in \mathbb{R}^{m \times k} \), with \( 1 \leq k \ll m \), may be different. They are assumed to satisfy \( W^TV = I_k \). This allows us to derive new quadrature
rules for the approximation of expressions of the form (49). We assume the matrix
$A$ to have the spectral factorization $A = Q \Lambda Q^{-1}$, where $Q \in \mathbb{C}^{m \times m}$ is nonsingular and
$\Lambda = \text{diag}[\lambda_1, \ldots, \lambda_m] \in \mathbb{C}^{m \times m}$. The eigenvalues $\lambda_i$ are real or appear in complex conjugate
pairs.

Substituting the spectral factorization of $A$ into (49) yields

$$W^T f(A) V = \tilde{W} f(\Lambda) \tilde{V}^H = \sum_{i=1}^{m} f(\lambda_i) \alpha_i \beta_i^H = \int f(\lambda) \, d\sigma(\lambda) =: \mathcal{I} f,$$  \hspace{1cm} (65)

where $\tilde{W} = W^T Q = [\alpha_1, \ldots, \alpha_m] \in \mathbb{C}^{k \times m}$ and $\tilde{V} = (Q^{-1} V)^H = [\beta_1, \ldots, \beta_m] \in \mathbb{C}^{k \times m}$. The superscript $^H$ denotes transposition and complex conjugation. The expression (65) is analogous to (51); it differs from the latter expression in that the measure $d\sigma$ in (65) may have support in $\mathbb{C} \setminus \mathbb{R}$.

Introduce the bilinear form

$$\langle f, g \rangle := \mathcal{I}(fg).$$ \hspace{1cm} (66)

There are two sequences of matrix polynomials $p_j$ and $q_j$, $j = 0, 1, \ldots$, that are biorthonormal with respect to this bilinear form, i.e.,

$$\langle p_i, q_j \rangle = \begin{cases} I_k & i = j, \\ O_k & i \neq j. \end{cases}$$

These polynomials satisfy three-term recurrence relations

$$\begin{align*}
\lambda p_{j-1}(\lambda) &= p_j(\lambda) \Gamma_j + p_{j-1}(\lambda) \Omega_j + p_{j-2}(\lambda) \Delta_{j-1}^T, \\
\lambda q_{j-1}(\lambda) &= q_j(\lambda) \Delta_j + q_{j-1}(\lambda) \Omega_j^T + q_{j-2}(\lambda) \Gamma_{j-1}^T,
\end{align*}$$ \hspace{1cm} (67)

where

$$p_0(\lambda) := I_k, \quad q_0(\lambda) := I_k, \quad p_{-1}(\lambda) := O_k, \quad q_{-1}(\lambda) := O_k, \quad \Delta_0 := O_k, \quad \Gamma_0 := O_k.$$
The matrix recursion coefficients $\Gamma_j, \Omega_j, \text{ and } \Delta_j$ are real $k \times k$ matrices; the matrices $\Gamma_j$ and $\Delta_j$ are upper triangular. For now we assume that all required matrices $\Gamma_j, \Omega_j, \text{ and } \Delta_j$ exist. We will comment on the situation when this is not the case later.

Let
\[
P_\ell(\lambda) := \begin{bmatrix} p_0(\lambda), \ldots, p_{\ell-1}(\lambda) \end{bmatrix} \in \mathbb{R}^{k \times k\ell},
\]
\[
Q_\ell(\lambda) := \begin{bmatrix} q_0(\lambda), \ldots, q_{\ell-1}(\lambda) \end{bmatrix} \in \mathbb{R}^{k \times k\ell}.
\]
The recursion relations for the polynomials $p_1, p_1, \ldots, p_\ell$ and $q_0, q_1, \ldots, q_\ell$ can be expressed compactly as
\[
\lambda P_\ell(\lambda) = P_\ell(\lambda)J_\ell + p_\ell(\lambda)\Gamma_\ell E_{\ell}^T,
\]
\[
\lambda Q_\ell(\lambda) = Q_\ell(\lambda)J_\ell^T + q_\ell(\lambda)\Delta_\ell E_{\ell}^T,
\]
where
\[
J_\ell = \begin{bmatrix}
\Omega_1 & \Delta_1^T & O \\
\Gamma_1 & \Omega_2 & \Delta_2^T \\
& \ddots & \ddots & \ddots \\
\Gamma_{\ell-2} & \Omega_{\ell-1} & \Delta_{\ell-1}^T \\
O & \Gamma_{\ell-1} & \Omega_\ell
\end{bmatrix} \in \mathbb{R}^{k\ell \times k\ell}
\]
is a block tridiagonal matrix. Its entries can be determined by $\ell$ steps of the nonsymmetric block Lanczos method; see below.

Assume that the matrix $J_\ell$ has the spectral factorization $J_\ell = Y_\ell\Theta_\ell Y_\ell^{-1}$, where
\[
Y_\ell = [y_1^{(\ell)}, \ldots, y_{k\ell}^{(\ell)}] \in \mathbb{C}^{k\ell \times k\ell}, \quad \Theta_\ell = \text{diag}[\theta_1^{(\ell)}, \ldots, \theta_{k\ell}^{(\ell)}] \in \mathbb{C}^{k\ell \times k\ell}.
\]
Letting $Z_\ell = [z_1^{(\ell)}, \ldots, z_{k\ell}^{(\ell)}] := Y_\ell^{-H}$, we obtain
\[
J_\ell = Y_\ell\Theta_\ell Z_\ell^H, \quad J_\ell^T Z_\ell = Z_\ell \tilde{\Theta}_\ell,
\]
(69)
where the bar denotes complex conjugation. Since the matrices $A, V, W$ have real entries only, so does $J$ and, therefore, $J^T = J^H$.

Introduce the expression

$$G_\ell f = E_1^T f(J_\ell)E_1.$$  \hspace{1cm} (70)

It is shown in [29, Section 5] that $G_\ell$ satisfies

$$G_\ell f = I f \quad \forall f \in \mathbb{P}_{2\ell - 1}.$$  

We therefore refer to (70) as an $\ell$-block Gauss quadrature rule associated with the operator $I$ defined by (65). Substituting (69) into (70) yields

$$G_\ell f := \sum_{i=1}^{k_\ell} f(\theta_i^{(\ell)}) u_i^{(\ell)}(v_i^{(\ell)})^H,$$  \hspace{1cm} (71)

where each vector $u_i^{(\ell)} \in \mathbb{C}^k$ consists of the first $k$ elements of the (right) eigenvector $y_i^{(\ell)}$ of $J_\ell$, and each vector $v_i^{(\ell)} \in \mathbb{C}^k$ is made up of the first $k$ elements of the (right) eigenvector $z_i^{(\ell)}$ of $J_\ell^T$. If it is better to compute $G_\ell f$ by using the representation (71) or by evaluating (70) without determining the spectral factorization of $J_\ell$ depends both on the function $f$ and on the accuracy with which the spectral factorization of $J_\ell$ can be calculated.

We can define block generalized averaged Gaussian quadrature rules analogously as in Section 4.2. Introduce for $1 \leq r < \ell$ the reverse matrices

$$\tilde{J}_{\ell-r,r} = \begin{bmatrix}
\Omega_{\ell-1} & \Delta_{\ell-2}^T & O \\
\Gamma_{\ell-2} & \Omega_{\ell-2} & \Delta_{\ell-3}^T \\
& \ddots & \ddots & \ddots \\
\Gamma_{r+1} & \Omega_{r+1} & \Delta_r^T \\
O & \Gamma_r & \Omega_r
\end{bmatrix} \in \mathbb{R}^{k(\ell-r) \times k(\ell-r)}$$  \hspace{1cm} (72)
and the concatenated symmetric tridiagonal matrices

\[
\hat{J}_{2\ell-r,r} = \begin{bmatrix}
J_{\ell-1} & \Delta_{\ell-1}^T E_{\ell-1} & O \\
\Gamma_{\ell-1} E_{\ell-1}^T & \Omega_{\ell} & \Delta_{\ell}^T E_{1}^T \\
O & \Gamma_{\ell} E_{1} & \hat{J}_{\ell-1,r}
\end{bmatrix} \in \mathbb{R}^{k(2\ell-r) \times k(2\ell-r)}. \tag{73}
\]

The matrices (72) and (73) are analogues of the matrices (57) and (58) of Section 4.2. We use the matrices (73) to define the quadrature rules

\[
\hat{G}_{2\ell-r,r} f = E_{1}^T f(\hat{J}_{2\ell-1,1}) E_{1}, \quad 1 \leq \ell < r, \tag{74}
\]

which generalize the rules (55) to measures \(d\sigma\) of the type (65).

**Theorem 10.** The block quadrature rules (74) are exact for all \(f \in \mathbb{P}_{2\ell}\). If the measure \(d\sigma\) defined by (65) is such that all diagonal blocks \(\Omega_{\ell}\) are equal, then the quadrature rules (74) are exact for all \(f \in \mathbb{P}_{2\ell+1}\).

**Proof.** We can show the result similarly as Theorem 8 by replacing the symmetric block Chebyshev algorithm by the nonsymmetric block Chebyshev algorithm. Alternatively, one can base the proof on the recursions of the nonsymmetric block Lanczos method (Algorithm 4.2 below), in particular on how moment information is used by the method. 

The matrix recursion coefficients in (67) can be determined by the nonsymmetric block Lanczos method. The implementation described by Algorithm 4.2 is proposed by Bai et al. [5]. The algorithm determines the block matrices required by the block Gauss rule (70). These block matrices also are required by the block quadrature rules (74). The matrices \(V, W \in \mathbb{R}^{m \times k}\) used to initialize the algorithm are assumed to satisfy \(V^T W = I_k\).
Algorithm 4.2 The nonsymmetric block Lanczos algorithm

1: Initialization: \( V_0 \Delta_0^T := W_0 \Gamma_0^T := O \in \mathbb{R}^{m \times k} \), \( V_1 := V/\|V\| \), \( W_1 := W/\|W\| \);

2: Iterate: for \( j = 1, 2, \ldots, \ell \) do

\[
\begin{align*}
\Omega_j &:= W_j^T (AV_j - V_{j-1} \Delta_{j-1}^T); \\
R_j &:= AV_j - V_j \Omega_j - V_{j-1} \Delta_{j-1}^T; \\
S_j &:= A^T W_j - W_j \Omega_j^T - W_{j-1} \Gamma_{j-1}^T; \\
Q_R R_R &:= R_j; \quad Q_S R_S := S_j; \\
W \Sigma V^T &:= Q_S^T Q_R; \\
V_{j+1} &:= Q_R V \Sigma^{-\frac{1}{2}}; \quad W_{j+1} := Q_S W \Sigma^{-\frac{1}{2}}; \\
\Gamma_j &:= \Sigma \frac{1}{2} V^T R_R; \quad \Delta_j := \Sigma \frac{1}{2} W^T R_S;
\end{align*}
\]

end \( j \)

In the above algorithm \( Q_R R_R := R_j \) and \( Q_S R_S := S_j \) indicate the computation of reduced QR factorizations, where \( Q_R, Q_S \in \mathbb{R}^{m \times k} \) have orthonormal columns and \( R_R, R_S \in \mathbb{R}^{k \times k} \) are upper triangular. Further, \( W \Sigma V^T := Q_S^T Q_R \) denotes the evaluation of a singular value decomposition of the right-hand side matrix. Algorithm 4.2 determines the decompositions

\[
\begin{align*}
A[V_1, \ldots, V_\ell] &= [V_1, \ldots, V_\ell] J_\ell + V_{\ell+1} \Gamma_\ell E_\ell^T, \\
A^T[W_1, \ldots, W_\ell] &= [W_1, \ldots, W_\ell] J_\ell^T + W_{\ell+1} \Delta_\ell E_\ell^T,
\end{align*}
\]

with the matrix \( J_\ell \) given by (68).

The description of Algorithm 4.2 assumes that the matrix products \( S_j^T R_j \), \( 1 \leq j \leq \ell \), are (numerically) nonsingular. The algorithm is said to break down if one of these products is (numerically) singular. The problem of breakdown is more complicated for Algorithm 4.2 than for Algorithm 4.1. A breakdown at step \( j \) of Algorithm 4.2 is said to be serious if \( S_j^T R_j \) is singular, but both the matrices \( S_j \) and \( R_j \) are of full rank. Bai et al. [5] provide a thorough discussion on breakdown and show that serious breakdown
can be circumvented by restarting Algorithm 4.2 after introducing appropriate additional vectors in the initial block vectors \( W_1 \) and \( V_1 \), and thereby increasing the block size.

Similarly as in Section 4.2, we may modify the trailing principal \( k(\ell - r) \times k(\ell - r) \) submatrix of the matrix (73) to obtain a new block quadrature for which an analogue of Theorem 10 holds. In particular, interchanging the sub- and super-diagonal block entries of the matrices (72) gives the matrices

\[
\tilde{J}_{\ell-r,r} = \begin{bmatrix}
\Omega_{\ell-1} & \Gamma_{\ell-2} & O \\
\Delta^T_{\ell-2} & \Omega_{\ell-2} & \Gamma_{\ell-3} \\
\ddots & \ddots & \ddots \\
\Delta^T_{r+1} & \Omega_{r+1} & \Gamma_r \\
O & \Delta^T_r & \Omega_r
\end{bmatrix} \in \mathbb{R}^{k(\ell-r) \times k(\ell-r)}, \quad 1 \leq r < \ell,
\]

and the associated concatenated matrices

\[
\tilde{J}_{2\ell-r,r} = \begin{bmatrix}
J_{\ell-1} & \Delta^T_{\ell-1} E_{\ell-1} & O \\
\Gamma_{\ell-1} E^T_{\ell-1} & \Omega_\ell & \Delta^T_\ell E^T_1 \\
O & \Gamma_\ell E_1 & \tilde{J}_{\ell-r,r}
\end{bmatrix} \in \mathbb{R}^{k(2\ell-r) \times k(2\ell-r)}, \quad 1 \leq r < \ell, \quad (76)
\]

which yield the block quadrature rules

\[
\tilde{G}^r_{2\ell-r,r} f = E^T_1 f(\tilde{J}_{2\ell-r,r}) E_1, \quad 1 \leq r < \ell. \quad (77)
\]

The following result can be shown in the same way as Theorem 10.

**Corollary 4.** The quadrature rules (77) are exact for all \( f \in \mathbb{P}_{2\ell} \). If the measure \( d\sigma \), defined by (65), is such that all diagonal blocks \( \Omega_i \) are equal, then the rules are exact for all \( f \in \mathbb{P}_{2\ell+1} \).
Let $\tilde{V}_\ell = [V_1, \ldots, V_\ell]$ and $\tilde{W}_\ell = [W_1, \ldots, W_\ell]$. The nonsymmetric block Lanczos decompositions (75) are such that the rank of $A\tilde{V}_\ell - \tilde{V}_\ell J_\ell$ and $A^T\tilde{W}_\ell - \tilde{W}_\ell J_\ell^T$ is at most $k$.

The following result is analogous to Theorem 9 and shows that decompositions involving the block tridiagonal matrix $\tilde{J}_{2\ell-1,1}$ have remainders of rank at most $2k$.

**Theorem 11.** Let the block permutation matrix $P_\ell$ be defined by (64) and introduce the matrices

$$
\tilde{V}_{2\ell-1} = [\tilde{V}_\ell, \tilde{V}_{\ell-1}P_{\ell-1}] \in \mathbb{R}^{m \times k(2\ell-1)}, \quad \tilde{W}_{2\ell-1} = [\tilde{W}_\ell, \tilde{W}_{\ell-1}P_{\ell-1}] \in \mathbb{R}^{m \times k(2\ell-1)}.
$$

Then both the matrices $A\tilde{V}_{2\ell-1} - \tilde{V}_{2\ell-1} \tilde{J}_{2\ell-1,1}$ and $A^T\tilde{W}_{2\ell-1} - \tilde{W}_{2\ell-1} \tilde{J}_{2\ell-1,1}^T$ have rank at most $2k$.

**Proof.** The result can be shown similarly as Theorem 9. We therefore omit the details.

$\Box$
5.1 Szegő Quadrature Rules

Szegő quadrature rules are commonly applied to integrate periodic functions on the unit circle in the complex plane. They are designed to approximate integrals of the following form

$$I(f) := \frac{1}{2\pi} \int_{-\pi}^{\pi} f(e^{i\theta}) d\mu(\theta), \quad (78)$$

where the measure $\mu(\theta)$ is nondecreasing and has infinitely many points of increase, and is such that all moments

$$\mu_j := \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-ij\theta} d\mu(\theta), \quad j = 0, \pm 1, \pm 2, \ldots , \quad (79)$$

exist and are finite. For notational convenience, we let $\mu$ be scaled so that $\mu_0 = 1$. The integrand $f$ is assumed to be continuous on the interval $[-\pi, \pi]$.

Introduce for polynomials $f$ and $g$ the inner product

$$(f, g) := \frac{1}{2\pi} \int_{-\pi}^{\pi} f(e^{i\theta}) \overline{g(e^{i\theta})} d\mu(\theta), \quad (80)$$

where $i := \sqrt{-1}$ and the bar denotes complex conjugation.

There is an infinite sequence of monic orthogonal polynomials $\{\psi_j\}_{j=0}^{\infty}$ with respect to the inner product (80). The $\psi_j$ are known as Szegő polynomials and satisfy the following
recursion relations

\[ \psi_0(z) = \psi_0^*(z) = 1, \]
\[ \psi_j(z) = z\psi_{j-1}(z) + \gamma_j \psi_{j-1}^*(z), \quad j = 1, 2, 3, \ldots, \]
\[ \psi_j^*(z) = \bar{\gamma}_j z\psi_{j-1}(z) + \psi_{j-1}^*(z), \]

where \( \psi_j^*(z) := z^j \bar{\psi}_j(z^{-1}) \). Thus, if \( \psi_j(z) = \sum_{k=0}^j \beta_k^{(j)} z^k \), then \( \psi_j^*(z) = \sum_{k=0}^j \bar{\beta}_k^{(j)} z^k \).

The \( \psi_j^* \) are sometimes referred to as reversed polynomials. We have \( \psi_j^*(0) = 1 \) for all \( j \geq 0 \) and, therefore, by (82), the recursion coefficients satisfy \( \gamma_j = \psi_j(0), \quad j = 1, 2, 3, \ldots. \)

They are sometimes referred to as Schur parameters or reflection coefficients and satisfy \(|\gamma_j| < 1 \) for all \( j \). The recursion coefficients can be computed by combining (82) with

\[ \gamma_j = -(1, z\psi_{j-1})/\delta_{j-1}, \quad j = 1, 2, 3, \ldots, \]
\[ \delta_j = \delta_{j-1}(1 - |\gamma_j|^2), \]

with \( \delta_0 = 1 \). Many properties of Szegő polynomials are described in, e.g., \([68,74]\).

We are interested in approximating the integral (78) by quadrature rules of the form

\[ S_r^{(n)}(f) = \sum_{k=1}^n \omega_k^{(n)} f(\lambda_k^{(n)}), \quad \omega_k^{(n)} > 0, \quad \lambda_k^{(n)} \in \Gamma, \]

where \( \Gamma := \{ z \in \mathbb{C} : |z| = 1 \} \) denotes the unit circle in the complex plane. The \( \lambda_k^{(n)} \) are nodes and the \( \omega_k^{(n)} \) weights of the quadrature rule. The nodes depend on the parameter \( \tau \), which will be discussed in later sections.

Let \( \Lambda_{(n-1),n-1} \) denote the set of Laurent polynomials

\[ L_{n-1}(z) = \sum_{k=-(n-1)}^{n-1} c_k z^k, \quad c_k \in \mathbb{C}, \]

of order at most \( n - 1 \). The quadrature rule (85) is said to be a Szegő quadrature rule if

\[ S_r^{(n)}(p) = \mathcal{I}(p) \quad \forall p \in \Lambda_{(n-1),n-1}, \]
where the integral $I$ is defined by (78). This requirement defines the $n$-point Szegő rule uniquely up to the location of one node, say $\lambda^{(n)}_1$, which can be chosen arbitrarily on the unit circle; see, e.g., [47, 56]. There are no quadrature rules of the form (85) that are exact for all Laurent polynomials of order $n$. The parameter $\tau$ will be used to fix the node $\lambda^{(n)}_1$.

Laurent polynomials (86) with $z = \exp(i\theta)$, $\theta \in \mathbb{R}$, are trigonometric polynomials in $\theta$. For instance,

$$L_{n-1}(\exp(i\theta)) = a_0 + \sum_{k=1}^{n-1} (a_k \cos(k\theta) + b_k \sin(k\theta)),$$

for appropriate coefficients $a_j, b_j \in \mathbb{C}$. Szegő quadrature rules are quadrature rules for trigonometric polynomials of maximal order. This makes them attractive to use for the integration of periodic functions.

Gragg [47] and Jones et al. [57] show that the nodes $\lambda^{(n)}_m$ of an $n$-point Szegő quadrature rule (85) are the zeros of the para-orthogonal polynomial

$$B_n(z; \tau) := \psi_n(z) + \tau \psi_n^*(z),$$

where $\tau \in \Gamma$ is an arbitrary but fixed parameter. The weights can be expressed as

$$\omega^{(n)}_k = I(\ell_k), \quad k = 1, 2, \ldots, n,$$

where the $\ell_k$ are Lagrange polynomials determined by the nodes,

$$\ell_k(z) := \prod_{j=1}^{n} \frac{z - \lambda^{(n)}_j}{\lambda^{(n)}_k - \lambda^{(n)}_j}.$$
Introduce the upper Hessenberg matrix

\[ \tilde{H}_n(\tau) := \begin{bmatrix}
-\tilde{\gamma}_0\gamma_1 & -\tilde{\gamma}_0\gamma_2 & \cdots & -\tilde{\gamma}_0\gamma_{n-1} & -\tilde{\gamma}_0\tau \\
1 - |\gamma_1|^2 & -\gamma_1\gamma_2 & \cdots & -\gamma_1\gamma_{n-1} & -\gamma_1\tau \\
0 & 1 - |\gamma_2|^2 & \cdots & -\gamma_2\gamma_{n-1} & -\gamma_2\tau \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 1 - |\gamma_{n-1}|^2 & -\gamma_{n-1}\tau \\
\end{bmatrix} \in \mathbb{C}^{n \times n}, \quad (89) \]

where \( \tilde{\gamma}_0 := 1 \) and \( \tau \in \Gamma \), and define the diagonal matrix \( D_n := \text{diag}[\delta_0, \delta_1, \ldots, \delta_{n-1}] \).

Gragg [47] considered the matrix \( H_n(\tau) := D_n^{-1/2} \tilde{H}_n(\tau) D_n^{1/2} \), \( (90) \) which is similar to \( \tilde{H}_n(\tau) \). The following results are due to Gragg [47].

**Proposition 4.** The matrix \((90)\) is unitary, its eigenvalues are the zeros of the para-orthogonal polynomial \((88)\) and therefore are the nodes \( \lambda_k^{(n)}, k = 1, 2, \ldots, n, \) of the Szegő rule \((85)\), and the magnitude squared of the first components of the unit eigenvectors of \( H_n(\tau) \) are the weights \( \omega_k^{(n)}, k = 1, 2, \ldots, n, \) of the Szegő rule.

The matrix \((90)\) is determined by the parameters: \( \gamma_1, \gamma_2, \ldots, \gamma_{n-1}, \tau \). The Levinson or Schur algorithms can be used to compute the \( \gamma_1, \gamma_2, \ldots, \gamma_{n-1} \) from the moments \( \{\mu_j\}_{j=0}^{n-1} \); see, e.g., [47,55,58]. Several fast algorithms for computing the nodes and weights of the Szegő rule \((85)\) from \( \gamma_1, \gamma_2, \ldots, \gamma_{n-1}, \tau \), without explicitly forming the matrix \((90)\), are discussed in [2,4,19,46,48,49,72].
5.2 Anti-Szegő Quadrature Rules

Szegő quadrature rules are used to define anti-Szegő quadrature rules. The $n$-node anti-Szegő rules are of the form

$$A_{\tau}^{(n)}(f) = \sum_{k=1}^{n} \tilde{\omega}_k^{(n)} f(\tilde{\lambda}_k^{(n)}), \quad \tilde{\omega}_k^{(n)} > 0, \quad \tilde{\lambda}_k^{(n)} \in \Gamma. \quad (91)$$

They are characterized by the property that the quadrature error for Laurent polynomials of order at most $n$ is a specified negative multiple of the quadrature error obtained with the $n$-node Szegő rule (85). Thus, for some specified constant $c > 0$,

$$I(p) - A_{\tau}^{(n)}(p) = -c(I(p) - S_{\tau}^{(n)}(p)), \quad \forall p \in \Lambda_{-n,n}. \quad (92)$$

It follows from (87) that

$$I(p) - A_{\tau}^{(n)}(p) = 0, \quad \forall p \in \Lambda_{-(n-1),n-1},$$

however, $A_{\tau}^{(n)}(p) \neq I(p)$ for some $p \in \Lambda_{-n-n} \setminus \Lambda_{-(n-1),n-1}$. Solving (92) for $I(p)$ yields

$$I(p) = \left( \frac{1}{c+1} \right) A_{\tau}^{(n)}(p) + \left( \frac{c}{c+1} \right) S_{\tau}^{(n)}(p), \quad \forall p \in \Lambda_{-n,n},$$

which suggests the definition of the averaged Szegő rule

$$M_{\tau}^{(n)}(f) = \left( \frac{1}{c+1} \right) A_{\tau}^{(n)}(f) + \left( \frac{c}{c+1} \right) S_{\tau}^{(n)}(f). \quad (93)$$

Since this averaged rule is exact for $f \in \Lambda_{-n,n}$, while the $n$-node Szegő and anti-Szegő rules (85) and (91) are not, we can expect the averaged Szegő rule to yield higher accuracy for many functions $f$. This can be verified by computations; see [59] for examples.

The construction of the anti-Szegő rule (91) requires that the Schur parameter $\gamma_n$ be known; see [59]. The construction is particularly simple when the auxiliary parameter
$\tau \in \Gamma$ is chosen to be

$$\tau := \begin{cases} \frac{\gamma_n}{|\gamma_n|}, & \text{if } \gamma_n \neq 0, \\ 1, & \text{if } \gamma_n = 0. \end{cases}$$  \hspace{1cm} (94)$$

For this choice the unitary upper Hessenberg matrix $H_n(\tau)$ is a closest unitary matrix, in any unitarily invariant norm, to the upper Hessenberg matrix $H_n(\gamma_n)$; see, e.g., [65] for a proof. We have

$$c = \begin{cases} \frac{1 + |\gamma_n|}{1 - |\gamma_n|}, & 0 < |\gamma_n| < 1, \\ 1, & \gamma_n = 0. \end{cases}$$  \hspace{1cm} (95)$$

5.3 Generalized Averaged Szegö Quadrature Rules

From last two chapters, we know that the nodes and weights of an $(\ell+1)$-point Gauss quadrature rule are the eigenvalues and squares of the first components of suitably normalized eigenvectors of a symmetric tridiagonal matrix $T_{\ell+1}$. We have proposed that the last diagonal entry of $T_{\ell+1}$ be replaced by the matrix $\tilde{T}_{\ell,0}$ obtained by flipping the leading principal $\ell \times \ell$ submatrix of $T_{\ell+1}$ upside-down and right-left. This yields a symmetric tridiagonal matrix of order $2\ell + 1$ that defines a new quadrature rule that Spalević [71] refers to as a generalized averaged Gauss rule. Properties and applications are described in [70] and Chapters 3 and 4. This section describes an extension of the rules (85). This extension is inspired by the work of Spalević [70,71]. We refer to the extended quadrature rules obtained as generalized averaged Szegö rules.

We define the generalized averaged Szegö quadrature rule

$$\hat{S}_r^{(2n-2)}(f) = \sum_{k=1}^{2n-2} \hat{\omega}_k^{(2n-2)} f(\hat{\lambda}_k^{(2n-2)}), \quad \hat{\omega}_k^{(2n-2)} > 0, \quad \hat{\lambda}_k^{(2n-2)} \in \Gamma,$$  \hspace{1cm} (96)$$

by using the recursion coefficients $\gamma_1, \gamma_2, \ldots, \gamma_{n-1}$ that define the Szegö rule (85), first
forwards and then backwards. Thus, introduce the recursion coefficients

\[
\hat{\gamma}_j = \gamma_j, \quad j = 1, 2, \ldots, n - 1,
\]

\[
\hat{\gamma}_j = \gamma_{2n-2-j}, \quad j = n, n + 1, \ldots, 2n - 3,
\]

as well as the auxiliary coefficients

\[
\hat{\delta}_j = \hat{\delta}_{j-1}(1 - |\hat{\gamma}_j|^2), \quad j = 1, 2, \ldots, 2n - 3,
\]

with \(\hat{\delta}_0 = 1\). The recursion coefficients \(\{\hat{\gamma}_j\}_{j=0}^{2n-3}\) and the parameter \(\tau \in \Gamma\) define the matrix \(\hat{H}_{2n-2}\) similarly as the matrix (89) is determined by \(\gamma_1, \gamma_2, \ldots, \gamma_{n-1}\) and \(\tau\). Expressing \(\hat{H}_{2n-2}\) in terms of the latter recursion coefficients, we obtain

\[
\hat{H}_{2n-2}(\tau) := \begin{bmatrix}
-\gamma_0 \gamma_1 & -\gamma_0 \gamma_2 & \cdots & -\gamma_0 \gamma_{n-1} & -\gamma_0 \gamma_{n-2} & \cdots & -\gamma_0 \gamma_1 & -\gamma_0 \\
1 - |\gamma_1|^2 & -\gamma_1 \gamma_2 & \cdots & -\gamma_1 \gamma_{n-1} & -\gamma_1 \gamma_{n-2} & \cdots & -\gamma_1 \gamma_1 & -\gamma_1 \\
0 & 1 - |\gamma_2|^2 & \cdots & -\gamma_2 \gamma_{n-1} & -\gamma_2 \gamma_{n-2} & \cdots & -\gamma_2 \gamma_1 & -\gamma_2 \\
\vdots & 0 & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & 1 - |\gamma_{n-2}|^2 & -\gamma_{n-2} \gamma_{n-1} & -\gamma_{n-2} \gamma_{n-2} & \cdots & -\gamma_{n-2} \gamma_1 & -\gamma_{n-2} \\
\end{bmatrix}
\]

Introduce

\[
\hat{D}_{2n-2} = \text{diag}[\hat{\delta}_0, \hat{\delta}_1, \ldots, \hat{\delta}_{2n-2}].
\]

Similarly as in Section 5.1, the upper Hessenberg matrix

\[
\hat{H}_{2n-2}(\tau) := \hat{D}_{2n-2}^{-1/2} \hat{H}_{2n-2}(\tau) \hat{D}_{2n-2}^{1/2}
\]

(98)
is unitary. Its eigenvalues \(\hat{\lambda}_k^{(2n-2)}\), \(k = 1, 2, \ldots, 2n - 2\), are the nodes of the quadrature rule (96) and the magnitude squared of the first components of associated unit eigenvectors are the weights \(\hat{\omega}_k^{(2n-2)}\).
Proposition 5. The generalized averaged Szegő quadrature rule (96) with the nodes and weights determined as described above is exact for all Laurent polynomial in $\Lambda_{-n+1,n-1}$.

Proof. The result follows because the first recursion coefficients $\hat{\gamma}_j$, $j = 0, 1, \ldots, n-1$, that define (96) agree with the recursion coefficients $\gamma_j$, $j = 0, 1, \ldots, n-1$, that determine (85), and the latter rule is exact for all Laurent polynomial in $\Lambda_{-n+1,n-1}$. The exactness of the latter rule follows from the fact that the Levinson or Schur algorithms determine the recursion coefficients $\gamma_j$, $j = 0, 1, \ldots, n-1$, so that the quadrature rule (85) integrates the moments $\mu_j$, $j = 0, \pm 1, \ldots, \pm(n-1)$ exactly.

The generalized averaged Szegő rule may be more accurate than indicated by the above proposition.
CHAPTER 6

Numerical Applications

In this chapter, we present applications with the new quadrature rules that we have developed in prior chapters. All computations were carried out using MATLAB R2010b on a 64-bit DELL personal computer with about 15 significant decimal digits.

6.1 Approximation of $u^T f(A)v$

In this section, we use four examples to illustrate the performance of the quadrature rules when applied to several integrands $f$ and matrices $A$. The examples we choose are very generic and the reader need no further background knowledge to understand them. Additionally we will present two examples which are related to network analysis.

Example 6.1.1. We would like to determine an approximation of the functional $F(A) := u^T(I+A^2)^{-1}u$, where $A \in \mathbb{R}^{200 \times 200}$ has randomly generated uniformly distributed real eigenvalues in the interval $[-5, 5]$ and a random orthogonal eigenvector matrix, and $u$ is a random vector of unit norm. The exact value of $F(A)$ is 0.2942. Table 1 displays the relative errors achieved with quadrature rules (23) for $r \in \{0, \ell/2, \ell-1\}$ and for several values of $\ell$. We note that $r = 0$ and $r = \ell - 1$ are the smallest and largest possible $r$-values. For all $\ell$-values, except for $\ell = 12$, the most accurate approximations are achieved for $r = 0$.

Table 2 shows the relative differences $|(G_{\ell+1} - \hat{G}_{2\ell+1})f|/F(A)$ and $|(G_{4\ell+1, 5} - \hat{G}_{2\ell+1})f|/F(A)$. The former differences provide fairly accurate estimate of the magnitude of the relative
error in $G_{\ell+1}f$, but overestimate the magnitude of the relative error in $\hat{G}_{2\ell+1}f$. The relative differences $|(G_{\ell+1} - \hat{G}_{2\ell+1})f|/F(A)$ are larger than $|(\hat{G}_{2\ell+1}f - \hat{G}_{2\ell+1})f|/F(A)$ for all $\ell$-values. This suggests that one may use the latter as an estimate of the magnitude of the relative error in $\hat{G}_{2\ell+1}f$, but this may give an underestimate of the actual error.

We note that the derivatives of $f(x) = (1 + x^2)^{-1}$ change sign on $[-5, 5]$. Therefore the technique described in [41,43] for bounding $F(A)$ based on evaluating pairs of Gauss and Gauss–Radau quadrature rules is not guaranteed to yield lower and upper bounds.

Table 1: Relative errors in computed approximations of $F(A) = u^T(I + A^2)^{-1}u$ with $A$ symmetric indefinite.

<table>
<thead>
<tr>
<th>$m = 200$</th>
<th>$\ell = 6$</th>
<th>$\ell = 10$</th>
<th>$\ell = 12$</th>
<th>$\ell = 14$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_{\ell+1}f$</td>
<td>$8.4003 \times 10^{-2}$</td>
<td>$1.4300 \times 10^{-2}$</td>
<td>$6.9629 \times 10^{-3}$</td>
<td>$1.2098 \times 10^{-3}$</td>
</tr>
<tr>
<td>$\hat{G}_{2\ell+1}f$</td>
<td>$5.5165 \times 10^{-3}$</td>
<td>$8.7300 \times 10^{-4}$</td>
<td>$2.1057 \times 10^{-3}$</td>
<td>$4.0794 \times 10^{-4}$</td>
</tr>
<tr>
<td>$\hat{G}_{2\ell+1}f$</td>
<td>$4.2586 \times 10^{-2}$</td>
<td>$8.8065 \times 10^{-4}$</td>
<td>$1.6913 \times 10^{-3}$</td>
<td>$5.5927 \times 10^{-4}$</td>
</tr>
<tr>
<td>$\hat{G}_{2\ell+1}f$</td>
<td>$6.6907 \times 10^{-2}$</td>
<td>$4.5364 \times 10^{-3}$</td>
<td>$1.9222 \times 10^{-3}$</td>
<td>$1.9274 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Table 2: Relative error estimates for computed approximations of $F(A) = u^T(I + A^2)^{-1}u$ with $A$ symmetric indefinite.

<table>
<thead>
<tr>
<th>$m = 200$</th>
<th>$\ell = 6$</th>
<th>$\ell = 10$</th>
<th>$\ell = 12$</th>
<th>$\ell = 14$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>(G_{\ell+1} - \hat{G}_{2\ell+1})f</td>
<td>/F(A)$</td>
<td>$8.9519 \times 10^{-2}$</td>
<td>$1.3427 \times 10^{-2}$</td>
</tr>
<tr>
<td>$</td>
<td>(\hat{G}<em>{2\ell+1}f - \hat{G}</em>{2\ell+1})f</td>
<td>/F(A)$</td>
<td>$3.7069 \times 10^{-2}$</td>
<td>$1.7537 \times 10^{-3}$</td>
</tr>
</tbody>
</table>

Example 6.1.2. This example determines approximations of $F(A) := u^T \exp(A)u$, where $A = [a_{i-j}] \in \mathbb{R}^{1000 \times 1000}$ is a pentadiagonal nonsymmetric Toeplitz matrix with $a_0 = 1, a_1 = 3/2, a_2 = 2, a_{-1} = 2$, and $a_{-2} = 3$. We let $u = [1/\sqrt{1000}, \ldots, 1/\sqrt{1000}]^T \in \mathbb{R}^{1000}$. The exact value of $F(A)$ is $1.3273 \times 10^4$.

Table 3 displays the relative errors achieved with the different quadrature rules. We
choose \( r = \ell/2 \) when \( \ell \) is even, otherwise \( r = (\ell + 1)/2 \). The quadrature error for \( \hat{G}_{2\ell+1}f \) is the smallest for all values of \( \ell \) in this example.

Table 3: Relative errors of \( F(A) = u^T \exp(A)u \) with \( A \) a banded nonsymmetric Toeplitz matrix.

<table>
<thead>
<tr>
<th>( m = 1000 )</th>
<th>( \ell = 2 )</th>
<th>( \ell = 3 )</th>
<th>( \ell = 6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( G_{\ell+1}f )</td>
<td>( 1.9025 \times 10^{-4} )</td>
<td>( 2.1698 \times 10^{-5} )</td>
<td>( 5.7383 \times 10^{-9} )</td>
</tr>
<tr>
<td>( \hat{G}_{2\ell+1}f )</td>
<td>( 2.3086 \times 10^{-5} )</td>
<td>( 2.2268 \times 10^{-7} )</td>
<td>( 6.7648 \times 10^{-12} )</td>
</tr>
<tr>
<td>( \hat{G}_{\ell+2,\ell-1}f )</td>
<td>( 2.3251 \times 10^{-5} )</td>
<td>( 1.8832 \times 10^{-6} )</td>
<td>( 2.3465 \times 10^{-10} )</td>
</tr>
<tr>
<td>( \hat{G}_{2\ell+1-r,r}f )</td>
<td>( 2.9705 \times 10^{-4} )</td>
<td>( 1.9623 \times 10^{-6} )</td>
<td>( 3.8471 \times 10^{-11} )</td>
</tr>
</tbody>
</table>

The magnitudes of relative differences between quadrature errors are reported in Table 4. Similarly as in Table 2, the relative differences \( |(G_{\ell+1} - \hat{G}_{2\ell+1})f|/F(A) \) are fairly accurate estimates of the relative errors in \( G_{\ell+1}f \). The relative differences \( |(G_{2\ell+1-r,r} - \hat{G}_{2\ell+1})f|/F(A) \) are smaller for \( \ell \geq 3 \). Since \( A \) is nonsymmetric pairs of Gauss and Gauss–Radau rules are not guaranteed to provide lower and upper bounds for \( F(A) \). □

Table 4: Relative error estimates for computed approximations of \( F(A) = u^T \exp(A)u \) with \( A \) a banded nonsymmetric Toeplitz matrix.

<table>
<thead>
<tr>
<th>( m = 1000 )</th>
<th>( \ell = 2 )</th>
<th>( \ell = 3 )</th>
<th>( \ell = 6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>(G_{\ell+1} - \hat{G}_{2\ell+1})f</td>
<td>/F(A) )</td>
<td>( 1.6716 \times 10^{-4} )</td>
</tr>
<tr>
<td>(</td>
<td>(G_{2\ell+1-r,r} - \hat{G}_{2\ell+1})f</td>
<td>/F(A) )</td>
<td>( 3.2014 \times 10^{-4} )</td>
</tr>
</tbody>
</table>

Example 6.1.3. This example approximate the value of \( F(A) := u^T \log(A+cI)u \), where \( A = [a_{i-j}] \in \mathbb{R}^{100 \times 100} \) is a pentadiagonal nonsymmetric Toeplitz matrix with \( a_0 = 1 \), \( a_1 = 3/2 \), \( a_2 = 2 \), \( a_{-1} = 2 \), and \( a_{-2} = 3 \). The constant \( c \) is chosen as 4.3. Then all eigenvalues of \( A+cI \) have positive real part and some of them are close to the origin, the singularity of the logarithm function. We let \( u = [1/10, \ldots, 1/10]^T \in \mathbb{R}^{100} \). The exact value of \( F(A) \) is 2.6127.
Table 5 displays the relative errors achieved with the different quadrature rules. We choose \( r = \ell/2 \) when \( \ell \) is even, otherwise \( r = (\ell + 1)/2 \). Table 6 reports the magnitudes of relative differences between quadrature errors. The relative differences 
\[
| (\mathcal{G}_{\ell+1} - \hat{\mathcal{G}}_{2\ell+1}) f | / F(A)
\]
are fairly accurate estimates of the relative errors in \( \mathcal{G}_{\ell+1}f \).

\[\square\]

**Table 5:** Relative errors of \( F(A) = u^T \log(A + cI)u \) with \( A \) a banded nonsymmetric Toeplitz matrix.

<table>
<thead>
<tr>
<th>( m = 100 )</th>
<th>( \ell = 4 )</th>
<th>( \ell = 14 )</th>
<th>( \ell = 19 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{G}_{\ell+1}f )</td>
<td>4.5839 ( \times 10^{-6} )</td>
<td>1.7569 ( \times 10^{-7} )</td>
<td>5.1568 ( \times 10^{-9} )</td>
</tr>
<tr>
<td>( \hat{\mathcal{G}}_{2\ell+1}f )</td>
<td>1.4315 ( \times 10^{-7} )</td>
<td>1.5645 ( \times 10^{-7} )</td>
<td>3.2633 ( \times 10^{-9} )</td>
</tr>
<tr>
<td>( \hat{\mathcal{G}}<em>{2\ell+1}f</em>{\ell+1} )</td>
<td>1.9361 ( \times 10^{-6} )</td>
<td>1.6364 ( \times 10^{-7} )</td>
<td>4.2304 ( \times 10^{-9} )</td>
</tr>
<tr>
<td>( \hat{\mathcal{G}}_{2\ell+1-r,r}f )</td>
<td>1.8950 ( \times 10^{-6} )</td>
<td>1.1401 ( \times 10^{-7} )</td>
<td>3.6063 ( \times 10^{-9} )</td>
</tr>
</tbody>
</table>

**Table 6:** Relative error estimates for computed approximations of \( F(A) = u^T \log(A + cI)u \) with \( A \) a banded nonsymmetric Toeplitz matrix.

<table>
<thead>
<tr>
<th>( m = 100 )</th>
<th>( \ell = 4 )</th>
<th>( \ell = 14 )</th>
<th>( \ell = 19 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>(\mathcal{G}<em>{\ell+1} - \hat{\mathcal{G}}</em>{2\ell+1}) f</td>
<td>/ F(A) )</td>
<td>4.4408 ( \times 10^{-6} )</td>
</tr>
<tr>
<td>(</td>
<td>(\hat{\mathcal{G}}<em>{2\ell+1-r,r} - \hat{\mathcal{G}}</em>{2\ell+1}) f</td>
<td>/ F(A) )</td>
<td>1.7518 ( \times 10^{-6} )</td>
</tr>
</tbody>
</table>

The number of steps of the symmetric or nonsymmetric Lanczos process required to determine an approximation of a matrix functional depends on the function \( f \) and the location of the eigenvalues of the matrix \( A \). For instance, the following example requires more steps than the example above to yield comparable accuracy.

**Example 6.1.4.** In this example, we approximate the value \( F(A) := u^T \log(A + cI)v \), where \( u = e_2 \), \( v = e_2 + e_3 \). The result is the sum of the matrix entries \( [\log(A + cI)]_{2,2} \) and \( [\log(A + cI)]_{2,3} \). The matrix \( A \in \mathbb{R}^{200 \times 200} \) has random entries and the constant \( c \) is chosen so that all eigenvalues of \( A + cI \) have positive real part and some are close to the origin. The exact value of \( F(A) \) is 0.229.
Table 7 displays the relative errors achieved with the different quadrature rules. We choose \( r = \ell/2 \) when \( \ell \) is even, otherwise \( r = (\ell + 1)/2 \). The Table 8 reports the magnitudes of relative differences between quadrature errors. □

**Table 7: Relative errors in computed approximations of** \( F(A) = u^T \log(A + cI)v \) **with** \( A \) **nonsymmetric.**

<table>
<thead>
<tr>
<th>( m = 200 )</th>
<th>( \ell = 9 )</th>
<th>( \ell = 19 )</th>
<th>( \ell = 24 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{G}_{\ell+1}f )</td>
<td>3.1529 \times 10^{-5}</td>
<td>1.7917 \times 10^{-8}</td>
<td>8.7185 \times 10^{-9}</td>
</tr>
<tr>
<td>( \hat{G}_{2\ell+1}f )</td>
<td>3.1256 \times 10^{-5}</td>
<td>4.3359 \times 10^{-9}</td>
<td>8.5918 \times 10^{-9}</td>
</tr>
<tr>
<td>( \hat{G}_{\ell+2,\ell-1} )</td>
<td>3.1322 \times 10^{-5}</td>
<td>1.2428 \times 10^{-8}</td>
<td>8.5948 \times 10^{-9}</td>
</tr>
<tr>
<td>( \hat{G}_{2\ell+1-r,r} )</td>
<td>3.1276 \times 10^{-5}</td>
<td>2.2239 \times 10^{-8}</td>
<td>8.5687 \times 10^{-9}</td>
</tr>
</tbody>
</table>

**Table 8: Relative error estimates for computed approximations of** \( F(A) = u^T \log(A + cI)v \) **with** \( A \) **nonsymmetric.**

<table>
<thead>
<tr>
<th>( m = 200 )</th>
<th>( \ell = 9 )</th>
<th>( \ell = 19 )</th>
<th>( \ell = 24 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( | (\hat{G}<em>{\ell+1} - \hat{G}</em>{2\ell+1}) f |/F(A) )</td>
<td>2.7347 \times 10^{-7}</td>
<td>2.2253 \times 10^{-8}</td>
<td>1.2668 \times 10^{-10}</td>
</tr>
<tr>
<td>( | (\hat{G}<em>{2\ell+1-r,r} - \hat{G}</em>{2\ell+1}) f |/F(A) )</td>
<td>2.0600 \times 10^{-8}</td>
<td>2.6575 \times 10^{-8}</td>
<td>2.3078 \times 10^{-11}</td>
</tr>
</tbody>
</table>

6.2 Application of \( u^T f(A)v \) to the Analysis of Large Networks

In this subsection, we briefly summarize how quadrature rules can be applied to large network calculations and apply our new quadrature rules to some problems.

A network is defined by a graph \( G \), which is made up of a set of \( m \) vertexes \( V \) and a set of edges \( E \). We assume that \( G \) is an unweighted connected graph with simple edges, and with no loop of length one. This kind of graphs arise in many applications, including genetics, epidemiology, and telecommunication; see, e.g., Estrada and Higham [26, 28]. The adjacency matrix \( A = [a_{ij}] \in \mathbb{R}^{m \times m} \) associated with \( G \) has the entry \( a_{ij} = 1 \) if there is an edge between node \( i \) and node \( j \); otherwise \( a_{ij} = 0 \). The graph \( G \) in the examples below is undirected. Then \( A \) is symmetric. The size of the entries of \( \exp(A) \) is commonly used to measure properties of vertices of a network. For instance, when the diagonal
entry $[\exp(A)]_{ii}$ is large in comparison with other diagonal entries, the vertex $i$ is well connected and, therefore, important. A large off-diagonal entry $[\exp(A)]_{ij}$, $i \neq j$, signals that communication between nodes $i$ and $j$ is easy; see [8, 9, 26, 28–30] for discussions.

The *communicability betweenness* of vertex $p$ is defined by

$$1 \left( m - 1 \right) \left( m - 2 \right) \sum_{i \neq p} \sum_{j \neq p} \sum_{j \neq i} \frac{[\exp(A)]_{ij} - [\exp(A_p)]_{ij}}{[\exp(A)]_{ij}} ,$$

(99)

where $A_p$ is the adjacency matrix of the graph obtained by removing from $G$ all edges involving node $p$. This is a measure of the amount of communication passing through node $p$; see Estrada et al. [27].

The quantity (99) is cumbersome to compute. Therefore, Fenu et al. [29] introduced the *alternative communicability betweenness* of node $p$, given by

$$\frac{c_p^T \exp(A)c_p - c_p^T \exp(A_p)c_p}{c_p^T \exp(A)c_p} ,$$

(100)

where $c_p \in \mathbb{R}^m$ is the vector with all entries one, except for the $p$th entry, which vanishes. The quotient (100) is related to (99), but differs from the latter in that it takes into consideration the effect of removing node $p$ on the diagonal elements of $\exp(A)$.

**Example 6.2.1.** We consider a network that describes the protein interaction for yeast. Each edge represents the interaction of two proteins; see Sun et al. [11]. The data set is available at [7]. The graph has 2114 vertices, 4480 edges, and is undirected.

Table 9 shows the magnitude of relative errors in computed approximations of alternative communicability betweenness (100) of node $p = 2$ determined by quadrature rules discussed in this thesis. Only few quadrature nodes are required to yield approximations with higher accuracy than what typically is required in applications. □
Table 9: Relative errors in computed approximations of $\text{(100)}$ for $p = 2$.  

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>$\ell = 2$</th>
<th>$\ell = 3$</th>
<th>$\ell = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{G}_{\ell+1}$</td>
<td>$2.0738\times10^{-1}$</td>
<td>$1.0574\times10^{-2}$</td>
<td>$3.6370\times10^{-5}$</td>
</tr>
<tr>
<td>$\tilde{G}_{2\ell+1}$</td>
<td>$6.4565\times10^{-2}$</td>
<td>$4.0547\times10^{-2}$</td>
<td>$1.1034\times10^{-5}$</td>
</tr>
<tr>
<td>$\tilde{G}_{\ell+2,\ell-1}$</td>
<td>$5.3863\times10^{-2}$</td>
<td>$2.6791\times10^{-2}$</td>
<td>$8.5372\times10^{-6}$</td>
</tr>
</tbody>
</table>

Example 6.2.2. This example displays estimates of the total communicability determined by the quadrature rules considered in this thesis for the yeast network; see the previous example for a description. Table 10 shows the averaged Gaussian rule $\tilde{G}_{2\ell-1}$ to give approximations with the smallest relative errors for $\ell = 3$ and $\ell = 6$ Gaussian nodes. Quite high accuracy is achieved with very few nodes. □

Table 10: Relative errors in computed approximations of $c^T \exp(A)c$, $c = [1, 1, \ldots, 1]^T$.  

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>$\ell = 2$</th>
<th>$\ell = 3$</th>
<th>$\ell = 6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tilde{G}_{\ell+1}$</td>
<td>$1.7370\times10^{-1}$</td>
<td>$2.8278\times10^{-2}$</td>
<td>$1.8624\times10^{-5}$</td>
</tr>
<tr>
<td>$\tilde{G}_{2\ell+1}$</td>
<td>$3.0256\times10^{-2}$</td>
<td>$6.5685\times10^{-4}$</td>
<td>$1.1743\times10^{-6}$</td>
</tr>
<tr>
<td>$\tilde{G}_{\ell+2,\ell-1}$</td>
<td>$2.0679\times10^{-3}$</td>
<td>$4.7213\times10^{-3}$</td>
<td>$2.0722\times10^{-6}$</td>
</tr>
</tbody>
</table>

6.3 Approximation of $W^T f(A)V$

In this section, we are going to illustrate the performance of several block quadrature rules. Thus, we approximate expressions of the form

$$F(A) := W^T f(A)V,$$  

(101)

where $A \in \mathbb{R}^{m \times m}$ is a large symmetric or nonsymmetric matrix and $W, V \in \mathbb{R}^{m \times k}$ are block vectors with $1 \leq k \ll m$.

Example 6.3.1. Let $A \in \mathbb{R}^{100 \times 100}$ be a symmetric indefinite matrix with 50 equidistant eigenvalues in the interval $[-2, -1]$ and 50 equidistant eigenvalues in the interval $[\frac{1}{2}, 1]$. The eigenvector matrix is an orthogonal random matrix. The function $f$ in (101) is the
exponential, i.e., \( f(A) = \exp(A) \). We let \( k = 2 \) and \( W = V = E_1 \in \mathbb{R}^{100 \times 2} \). The error in the computed approximations of (101) is measured with the spectral norm as well as entry-wise. Let \( X, Y \in \mathbb{R}^{k \times k} \). Then we compute the relative differences

\[
R_2(X, Y) := \frac{\| X - Y \|}{\| F(A) \|},
\]

where \( \| \cdot \| \) denotes the spectral norm, and

\[
R_{\infty}(X, Y) := \max_{1 \leq i, j \leq k} \left| \frac{[X - Y]_{i,j}}{[F(A)]_{i,j}} \right|
\]

where \([X]_{i,j}\) denotes the \(\{i, j\}\)th entry of the matrix \(X\). For instance, we measure the error in the approximation (55) using \( R_2(G_\ell f, F(A)) \) and \( R_{\infty}(G_\ell f, F(A)) \). Table 11 displays the errors for several quadrature rules for the present example. We have

\[
F(A) = \begin{bmatrix} 1.243 & 0.063 \\ 0.063 & 1.259 \end{bmatrix}.
\]

The table shows the element-wise errors to be roughly of the same size as the spectral norm errors for all quadrature rules. Moreover, the errors are seen to decrease rapidly as \( \ell \) increases. The errors reported for \( \ell = 7 \) may be significantly affected by round-off errors introduced during the computations.

Table 12 displays error estimates obtained by evaluating the difference of approximations of \( F(A) \) determined with different quadrature rules. The estimate \( R_{\infty}(G_\ell f, \hat{G}_{2\ell-1,1} f) \) is seen to furnish fairly accurate approximations of \( R_{\infty}(G_\ell f, F(A)) \) and \( R_2(G_\ell f, \hat{G}_{2\ell-1,1} f) \) can be seen to be close to \( R_2(G_\ell f, \hat{G}_{2\ell-1,1} f) \). Thus, Table 12 suggests that the difference \( G_\ell f - \hat{G}_{2\ell-1,1} f \) furnished a useful estimate of the error in \( G_\ell f \). \( \square \)
Table 11: Errors in approximations of $F(A) = W^T \exp(A)W$ with $A$ symmetric indefinite.

<table>
<thead>
<tr>
<th></th>
<th>$\ell = 3$</th>
<th>$\ell = 5$</th>
<th>$\ell = 7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_\infty(G_{\ell f}, F(A))$</td>
<td>$3.016 \times 10^{-4}$</td>
<td>$4.356 \times 10^{-9}$</td>
<td>$5.338 \times 10^{-14}$</td>
</tr>
<tr>
<td>$R_\infty(G_{2\ell-1,1 f}, F(A))$</td>
<td>$2.177 \times 10^{-5}$</td>
<td>$6.879 \times 10^{-10}$</td>
<td>$3.063 \times 10^{-15}$</td>
</tr>
<tr>
<td>$R_\infty(G_{\ell+1,\ell-1 f}, F(A))$</td>
<td>$1.672 \times 10^{-5}$</td>
<td>$6.943 \times 10^{-10}$</td>
<td>$3.063 \times 10^{-15}$</td>
</tr>
</tbody>
</table>

Table 12: Error estimates for computed approximations of $F(A) = W^T \exp(A)W$ with $A$ symmetric indefinite.

<table>
<thead>
<tr>
<th></th>
<th>$\ell = 3$</th>
<th>$\ell = 5$</th>
<th>$\ell = 7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_\infty(\hat{G}_{\ell f}, F(A))$</td>
<td>$3.234 \times 10^{-4}$</td>
<td>$4.700 \times 10^{-9}$</td>
<td>$5.601 \times 10^{-14}$</td>
</tr>
<tr>
<td>$R_2(G_{2\ell-1,1 f}, F(A))$</td>
<td>$1.213 \times 10^{-5}$</td>
<td>$9.235 \times 10^{-11}$</td>
<td>$8.628 \times 10^{-16}$</td>
</tr>
<tr>
<td>$R_2(G_{\ell+1,\ell-1 f}, F(A))$</td>
<td>$1.604 \times 10^{-5}$</td>
<td>$1.196 \times 10^{-10}$</td>
<td>$5.374 \times 10^{-16}$</td>
</tr>
</tbody>
</table>

Example 6.3.2. This example is similar to an example in [41]. Let the symmetric positive definite matrix $A \in \mathbb{R}^{m \times m}$ be determined by the standard 5-point finite difference discretization of the Laplace operator on the unit square with $n$ mesh points along each coordinate direction. Then $m = n^2$ and

$$A = \begin{bmatrix}
T_n & -I_n & O \\
-I_n & T_n & -I_n \\
& \ddots & \ddots & \ddots \\
& & -I_n & T_n & -I_n \\
O & & -I_n & T_n
\end{bmatrix}$$
is block tridiagonal with tridiagonal diagonal blocks

\[ T_n = \begin{bmatrix} 4 & -1 & O \\ -1 & 4 & -1 \\ & & \ddots & \ddots & \ddots \\ & & -1 & 4 & -1 \\ O & -1 & 4 \end{bmatrix} \in \mathbb{R}^{n \times n}. \]

We set \( n = 10 \), and let \( V = W = E_1 \) and \( f(t) = t^{-1} \) in (101). Thus, \( F(A) \) is the leading principal \( 2 \times 2 \) submatrix of \( A^{-1} \). We have

\[ F(A) = \begin{bmatrix} 0.302 & 0.105 \\ 0.105 & 0.344 \end{bmatrix}. \]

Table 13 shows the errors in computed approximations of \( F(A) \) using several quadrature rules. Error estimates analogous to those reported in Table 12 are displayed in Table 14. Similarly as in Table 12, the estimates \( R_\infty(G_\ell f, \hat{G}_{2\ell-1,1} f) \) are quite accurate approximations of \( R_\infty(G_\ell f, F(A)) \), and \( R_2(G_\ell f, \hat{G}_{2\ell-1,1} f) \) can be seen to be fairly close to \( R_2(G_\ell f, \hat{G}_{2\ell-1,1} f) \). □

<table>
<thead>
<tr>
<th>( \ell )</th>
<th>( \ell = 15 )</th>
<th>( \ell = 20 )</th>
<th>( \ell = 25 )</th>
<th>( \ell = 30 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_\infty(G_\ell f, F(A)) )</td>
<td>9.165×10^{-5}</td>
<td>2.603×10^{-7}</td>
<td>1.828×10^{-10}</td>
<td>1.606×10^{-14}</td>
</tr>
<tr>
<td>( R_\infty(G_{2\ell-1,1} f, F(A)) )</td>
<td>3.412×10^{-5}</td>
<td>6.213×10^{-8}</td>
<td>3.869×10^{-11}</td>
<td>1.725×10^{-15}</td>
</tr>
<tr>
<td>( R_\infty(G_{\ell+1,\ell-1} f, F(A)) )</td>
<td>3.577×10^{-5}</td>
<td>8.972×10^{-8}</td>
<td>2.205×10^{-11}</td>
<td>1.194×10^{-15}</td>
</tr>
<tr>
<td>( R_2(G_\ell f, F(A)) )</td>
<td>5.564×10^{-8}</td>
<td>1.582×10^{-7}</td>
<td>1.112×10^{-10}</td>
<td>9.608×10^{-15}</td>
</tr>
<tr>
<td>( R_2(G_{2\ell-1,1} f, F(A)) )</td>
<td>2.067×10^{-5}</td>
<td>3.774×10^{-8}</td>
<td>2.353×10^{-11}</td>
<td>1.058×10^{-15}</td>
</tr>
<tr>
<td>( R_2(G_{\ell+1,\ell-1} f, F(A)) )</td>
<td>2.173×10^{-5}</td>
<td>5.451×10^{-8}</td>
<td>1.341×10^{-11}</td>
<td>7.053×10^{-16}</td>
</tr>
</tbody>
</table>

The following two examples are concerned with functionals of real nonsymmetric matrices.
Example 6.3.3. We would like to determine an approximation of the functional (101), where \( f(t) = t^{-1} \) and \( A \in \mathbb{R}^{300 \times 300} \) is a nonsymmetric matrix with uniformly distributed nonnegative random entries. The matrix is generated with the MATLAB command \( A = \text{rand}(300)/100 \). Matrices so obtained typically have one large real eigenvalue and many eigenvalues close to the origin, but none at the origin. The block vectors \( W, V \in \mathbb{R}^{300 \times 2} \) are given by

\[
W = [e_1, 2e_1 + 3e_2], \quad V = [e_1 - \frac{2}{3} e_2, \frac{1}{3} e_2],
\]

where \( e_j = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^{300} \) denotes the \( j \)th axis vector. Then \( V^T W = I_2 \).

The value of (101) is

\[
F(A) = \\
\begin{bmatrix}
-0.154 & -1.616 \times 10^{-5} \\
-7.550 \times 10^{-4} & -0.154
\end{bmatrix}
\]

and, of course, depends on the random matrix \( A \). The quadrature errors obtained when approximating \( F(A) \) are reported in Table 15 and error estimates are shown in Table 16. The latter are accurate approximations of the errors in \( \mathcal{G}_{\ell} f \). □

Example 6.3.4. This example differs from the previous one only in that the inverse is replaced by the exponential function. Thus, \( F(A) = W^T \exp(A)V \), and the matrices \( A \),
Table 15: Errors in approximations of \( F(A) = W^T A^{-1} V \) with \( A \) nonsymmetric.

<table>
<thead>
<tr>
<th>( R_\infty(G_\ell f, F(A)) )</th>
<th>( \ell = 3 )</th>
<th>( \ell = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_\infty(G_{2\ell-1,1} f, F(A)) )</td>
<td>( 1.932 \times 10^{-7} )</td>
<td>( 4.420 \times 10^{-13} )</td>
</tr>
<tr>
<td>( R_\infty(G_{\ell+1,\ell-1} f, F(A)) )</td>
<td>( 2.528 \times 10^{-8} )</td>
<td>( 1.113 \times 10^{-13} )</td>
</tr>
<tr>
<td>( R_2(G_\ell f, F(A)) )</td>
<td>( 1.760 \times 10^{-10} )</td>
<td>( 7.611 \times 10^{-16} )</td>
</tr>
<tr>
<td>( R_2(G_{2\ell-1,1} f, F(A)) )</td>
<td>( 1.976 \times 10^{-11} )</td>
<td>( 6.067 \times 10^{-16} )</td>
</tr>
<tr>
<td>( R_2(G_{\ell+1,\ell-1} f, F(A)) )</td>
<td>( 1.970 \times 10^{-11} )</td>
<td>( 6.023 \times 10^{-16} )</td>
</tr>
</tbody>
</table>

Table 16: Error estimates for computed approximations of \( F(A) = W^T A^{-1} V \) with \( A \) nonsymmetric.

<table>
<thead>
<tr>
<th>( R_\infty(G_\ell f, \tilde{G}_{2\ell-1,1} f) )</th>
<th>( \ell = 3 )</th>
<th>( \ell = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_2(G_\ell f, \tilde{G}_{2\ell-1,1} f) )</td>
<td>( 2.185 \times 10^{-7} )</td>
<td>( 4.095 \times 10^{-13} )</td>
</tr>
</tbody>
</table>

\( W \), and \( V \) are the same as in Example 4.3. We have

\[
F(A) = \begin{bmatrix}
1.010 & 0.002 \\
0.041 & 1.018
\end{bmatrix}.
\]

Quadrature errors and errors estimates are reported in Tables 17 and 18. □

Table 17: Errors in approximations of \( F(A) = W^T \exp(A)V \) with \( A \) nonsymmetric.

<table>
<thead>
<tr>
<th>( R_\infty(G_\ell f, F(A)) )</th>
<th>( \ell = 3 )</th>
<th>( \ell = 4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_\infty(G_{2\ell-1,1} f, F(A)) )</td>
<td>( 8.503 \times 10^{-7} )</td>
<td>( 1.325 \times 10^{-12} )</td>
</tr>
<tr>
<td>( R_\infty(G_{\ell+1,\ell-1} f, F(A)) )</td>
<td>( 8.994 \times 10^{-8} )</td>
<td>( 6.114 \times 10^{-14} )</td>
</tr>
<tr>
<td>( R_2(G_\ell f, F(A)) )</td>
<td>( 1.401 \times 10^{-8} )</td>
<td>( 2.639 \times 10^{-14} )</td>
</tr>
<tr>
<td>( R_2(G_{2\ell-1,1} f, F(A)) )</td>
<td>( 1.500 \times 10^{-9} )</td>
<td>( 1.479 \times 10^{-15} )</td>
</tr>
<tr>
<td>( R_2(G_{\ell+1,\ell-1} f, F(A)) )</td>
<td>( 1.501 \times 10^{-9} )</td>
<td>( 1.232 \times 10^{-15} )</td>
</tr>
</tbody>
</table>

6.4 Approximation of \( \frac{1}{2\pi} \int_{-\pi}^{\pi} T(t)d\mu(t) \)

Szegö quadrature rules are well suited for the integration of periodic functions. Let \( T(\theta) \) be a \( 2\pi \)-periodic function. We illustrate the performance of the quadrature rules discussed in Chapter 5 when applied to approximate integrals

\[
\mathcal{I}(T) = \frac{1}{2\pi} \int_{-\pi}^{\pi} T(t)d\mu(t).
\]
Table 18: Error estimates for computed approximations of $F(A) := W^T \exp(A)V$ with $A$ nonsymmetric.

\[
\begin{array}{c|cc}
 & \ell = 3 & \ell = 4 \\
R_\infty(G_\ell f, G_{2\ell-1,1} f) & 9.393 \times 10^{-7} & 1.264 \times 10^{-12} \\
R_2(G_\ell f, G_{2\ell-1,1} f) & 1.551 \times 10^{-8} & 2.499 \times 10^{-14} \\
\end{array}
\]

With a slight abuse of notation, we express the Szegö rule (85) as

$S^{(n)}(T) = \sum_{k=1}^{n} \psi_k^{(n)} T(\theta_k^{(n)}), \quad \theta_m^{(n)} = \text{arg}(\lambda_k^{(n)}).$

The other quadrature rules are applied in a similar fashion.

**Example 6.4.1.** Define the measure $d\mu(t) = 2 \sin^2(t/2) \, dt$. The associated monic Szegö polynomials are given by

$$
\psi_j(z) = \frac{1 - (j + 2)z^{j+1} + (j + 1)z^{j+2}}{(j + 1)(1 - z)^2}, \quad j = 0, 1, 2, \ldots ,
$$

and have recursion coefficients $\gamma_j = \psi_j(0) = 1/(j + 1)$; see [12].

Let $\tau = 1$. The nodes and weights of the $n$-node Szegö rule are determined by the unitary upper Hessenberg matrix $H_n(1) \in \mathbb{C}^{n \times n}$ given by (90) and the nodes and weights of the $(2n - 2)$-node generalized averaged Szegö rule (96) are determined by the unitary upper Hessenberg matrix $\tilde{H}_{2n-2}(1)$ defined by (98). Both the $n$-node Szegö and the $(2n - 2)$-node generalized averaged Szegö rules require only the recursion coefficients $\gamma_1, \gamma_2, \ldots , \gamma_{n-1}$. We also determine the $n$-node anti-Szegö rule (91), defined by $H_n(-1)$, and the rule (93) obtained by averaging the $n$-node Szegö and anti-Szegö rules. The latter rule requires that the recursion coefficient $\gamma_n$ be known because it is used to determine the coefficient $c$ in (92), defined by (95). We have for the present example $c = 1 + 2/n$.

We would like to compute approximations of the integral (102) with the integrand

$$
T(t) := \frac{1}{2} \log(5 + 4 \cos(t)).
$$

(103)
The value of the integral can be shown to be $I(T) \approx 0.44314718055995$.

Table 19 displays the error for the quadrature rules discussed in this paper. The Szegő and associated anti-Szegő rules are supposed to give quadrature errors of opposite sign. Therefore the quadrature error for $M_1^{(n)}(T)$ is smaller than the error for $S_1^{(n)}(T)$ and $A_1^{(n)}(T)$. The computation of the rule $M_1^{(n)}(T)$ requires the coefficient (95), whose computation requires knowledge of $\gamma_n$. The generalized averaged rule $\hat{S}_1^{(2n-2)}(T)$ can be seen to yield the highest accuracy for all choices of $n$. This rule does not require the evaluation of $\gamma_n$.

It is important to be able to estimate the quadrature error in a computed approximation of an integral. However, the computation of reliable error estimates can be difficult. Table 20 shows differences between values obtained with the different quadrature rules. These differences can be used as error estimates. The table shows the rule $\hat{S}_1^{(2n-2)}(T)$ to give the most accurate error estimate. The estimate is more accurate than those obtained with the rules $M_1^{(n)}(T)$ and $S_1^{(n+1)}(T)$, both of which require knowledge of the recursion coefficient $\gamma_n$. Tables 19 and 20 suggest that we use $\hat{S}_1^{(2n-2)}(T)$ as an approximation of the integral (102) and use the difference $|S_1^{(n)}(T) - \hat{S}_1^{(2n-2)}(T)|$ as an estimate of the magnitude of the quadrature error. □

Table 19: Errors for several quadrature rules applied to the approximation of the integral (102) with integrand (103).

<table>
<thead>
<tr>
<th>Rule</th>
<th>$n = 12$</th>
<th>$n = 15$</th>
<th>$n = 18$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1^{(n)}(T)$</td>
<td>$-2.2 \times 10^{-5}$</td>
<td>$2.2 \times 10^{-6}$</td>
<td>$-2.3 \times 10^{-7}$</td>
</tr>
<tr>
<td>$A_1^{(n)}(T)$</td>
<td>$2.3 \times 10^{-5}$</td>
<td>$-2.3 \times 10^{-6}$</td>
<td>$2.4 \times 10^{-7}$</td>
</tr>
<tr>
<td>$M_1^{(n)}(T)$</td>
<td>$-1.1 \times 10^{-6}$</td>
<td>$9.0 \times 10^{-8}$</td>
<td>$-7.9 \times 10^{-9}$</td>
</tr>
<tr>
<td>$\hat{S}_1^{(2n-2)}(T)$</td>
<td>$-1.5 \times 10^{-7}$</td>
<td>$9.2 \times 10^{-9}$</td>
<td>$-6.7 \times 10^{-10}$</td>
</tr>
</tbody>
</table>

70
Table 20: Estimate of absolute error in the approximation furnished by \( S_1^{(n)}(T) \).

<table>
<thead>
<tr>
<th>Difference</th>
<th>( n = 12 )</th>
<th>( n = 15 )</th>
<th>( n = 18 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>(</td>
<td>S_1^{(n)}(T) - M_1^{(n)}(T)</td>
<td>2.1 \times 10^{-5}</td>
<td>2.1 \times 10^{-6}</td>
</tr>
<tr>
<td>(</td>
<td>S_1^{(n)}(T) - S_1^{(n+1)}(T)</td>
<td>3.2 \times 10^{-5}</td>
<td>3.2 \times 10^{-6}</td>
</tr>
<tr>
<td>(</td>
<td>S_1^{(n)}(T) - \hat{S}_1^{(2n-2)}(T)</td>
<td>2.2 \times 10^{-5}</td>
<td>2.2 \times 10^{-6}</td>
</tr>
</tbody>
</table>

Example 6.4.2. The integrand in the integral (102) is the same as in the previous example, but the measure is \( d\mu(t) = 2\cos^2(t/2)dt \). The monic Szegő polynomials associated with this measure are given by

\[
\psi_j(z) = \frac{(-1)^j + (j + 2)z^{j+1} + (j + 1)z^{j+2}}{(j + 1)(1 + z)^2}, \quad j = 0, 1, 2, \ldots ,
\]

and have recursion coefficients \( \gamma_j = (-1)^j/(j + 1) \); see [12].

The value of the integral (102) is \( \mathcal{I}(T) \approx 0.94314718055995 \). Table 21 is analogous to Table 19 and lists the errors in the quadrature rules. The absolute value of the differences between the \( n \)-node Szegő rule and other rules are shown in Table 22. These differences can be used as estimates of the quadrature error for \( S_1^{(n)}(T) \). The generalized averaged Szegő quadrature rules can be seen to yield the highest accuracy as well as accurate estimates of the error in \( S_1^{(n)}(T) \). □

Table 21: Errors for several quadrature rules applied to the approximation of the integral (102) with integrand (103).

<table>
<thead>
<tr>
<th>Rule</th>
<th>( n = 12 )</th>
<th>( n = 15 )</th>
<th>( n = 18 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_1^{(n)}(T) )</td>
<td>-3.0 \times 10^{-6}</td>
<td>-2.9 \times 10^{-7}</td>
<td>-2.9 \times 10^{-8}</td>
</tr>
<tr>
<td>( A_1^{(n)}(T) )</td>
<td>4.5 \times 10^{-6}</td>
<td>4.1 \times 10^{-7}</td>
<td>4.0 \times 10^{-8}</td>
</tr>
<tr>
<td>( M_1^{(n)}(T) )</td>
<td>4.9 \times 10^{-7}</td>
<td>3.8 \times 10^{-8}</td>
<td>3.3 \times 10^{-9}</td>
</tr>
<tr>
<td>( \hat{S}_1^{(2n-2)}(T) )</td>
<td>-1.8 \times 10^{-7}</td>
<td>-1.1 \times 10^{-8}</td>
<td>-7.7 \times 10^{-10}</td>
</tr>
</tbody>
</table>

Example 6.4.3. In the last two examples, the \( \{ \gamma_j \} \) are real numbers. In this example, we choose the measure \( d\mu(t) = e^{t\pi} \sinh(t\pi)dt \). Then the recursion coefficients \( \gamma_j \) are complex and can be computed by Levinson’s algorithm; see [3,21].
The integral is the same as (102) and the integrand is the same as (103), thus $I(T) \approx 0.41271658497332$. The error and error estimates for this example are displayed in Table 23 and Table 24 below. The generalized averaged Szegő quadrature rules can be seen to yield the highest accuracy as well as accurate estimates of the error in $S_1^{(n)}(T)$. □

Table 23: Errors for several quadrature rules applied to the approximation of the integral (102) with integrand (103).

<table>
<thead>
<tr>
<th>Rule</th>
<th>$n = 12$</th>
<th>$n = 15$</th>
<th>$n = 18$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_1^{(n)}(T)$</td>
<td>$-5.1 \times 10^{-6}$</td>
<td>$-5.5 \times 10^{-7}$</td>
<td>$-5.4 \times 10^{-8}$</td>
</tr>
<tr>
<td>$A_1^{(n)}(T)$</td>
<td>$8.2 \times 10^{-6}$</td>
<td>$7.9 \times 10^{-7}$</td>
<td>$7.2 \times 10^{-8}$</td>
</tr>
<tr>
<td>$M_1^{(n)}(T)$</td>
<td>$9.8 \times 10^{-7}$</td>
<td>$7.4 \times 10^{-8}$</td>
<td>$5.5 \times 10^{-9}$</td>
</tr>
<tr>
<td>$\hat{S}_1^{(2n-2)}(T)$</td>
<td>$1.9 \times 10^{-7}$</td>
<td>$-1.4 \times 10^{-8}$</td>
<td>$-2.3 \times 10^{-9}$</td>
</tr>
</tbody>
</table>

Table 24: Estimate of absolute error in the approximation furnished by $S_1^{(n)}(T)$.

<table>
<thead>
<tr>
<th>Difference</th>
<th>$n = 12$</th>
<th>$n = 15$</th>
<th>$n = 18$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>S_1^{(n)}(T) - M_1^{(n)}(T)</td>
<td>$</td>
<td>$6.1 \times 10^{-6}$</td>
</tr>
<tr>
<td>$</td>
<td>S_1^{(n)}(T) - S_1^{(n+1)}(T)</td>
<td>$</td>
<td>$8.7 \times 10^{-6}$</td>
</tr>
<tr>
<td>$</td>
<td>S_1^{(n)}(T) - \hat{S}_1^{(2n-2)}(T)</td>
<td>$</td>
<td>$5.3 \times 10^{-6}$</td>
</tr>
</tbody>
</table>
CHAPTER 7

Conclusion

Matrix functionals (16) with a Hermitian matrix $A$ and $v = u$ are commonly approximated by the use of Gauss quadrature rules. We propose that the generalized averaged Gauss quadrature rules (22) or (23) be used instead of the associated $(\ell + 1)$-point Gauss rule, because the latter typically yield higher accuracy and require essentially the same computational effort when the matrix $A$ is so large that the dominating computational work in Algorithm 3.1 is the evaluation of matrix-vector products.

New quadrature rules (43) and (44) are developed for the approximation of matrix functionals (30) with a non-Hermitian matrix. The advantages of the rules (43) and (44) over the corresponding $(\ell + 1)$-point Gauss-type rule are the same as of the rules (22) or (23) for a Hermitian matrix $A$.

We extend the generalized averaged Gauss quadrature rule to the block quadrature rules for the approximation of expressions of the form (48) and (49). Some properties of these rules are shown. Computed examples illustrate that they can be applied to estimate the error in block Gauss quadrature rules, and that they can give a smaller error for essentially the same computational effort as the use of pairs of block Gauss and anti-Gauss rules.

At last, inspired by the intimate relation between the quadrature rules on the interval $[-1,1]$ and complex unit circle, we describe new quadrature rules for the approximation of integrals of periodic functions. The rules are extensions of standard Szegő quadrature
rules that are analogous to the generalized averaged Gauss rules. Computed examples show the extended rules to give higher accuracy than Szegő quadrature rules defined by the same number of recursion coefficients. This can be of significance when the recursion coefficients are not explicitly known, but have to be computed from the moments. Application of the generalized averaged Szegő rules to the estimation of the quadrature error in Szegő rules is illustrated.
BIBLIOGRAPHY


77


