Numerical Algorithms **26:** 251–280, 2001.

Why Gaussian quadrature in the complex plane?

Paul E. Saylor [a,*] and Dennis C. Smolarski [b]

[a] *Department of Computer Science, University of Illinois, Urbana, IL 61801, USA*
E-mail: saylor@cs.uiuc.edu

[b] *Department of Mathematics and Computer Science, Santa Clara University, Santa Clara, CA 95053, USA*
E-mail: dsmolarski@scu.edu

Received 24 July 2000; revised 1 December 2000
Communicated by C. Brezinski

This paper synthesizes formally orthogonal polynomials, Gaussian quadrature in the complex plane and the bi-conjugate gradient method together with an application. Classical Gaussian quadrature approximates an integral over (a region of) the real line. We present an extension of Gaussian quadrature over an arc in the complex plane, which we call *complex Gaussian quadrature.* Since there has not been any particular interest in the numerical evaluation of integrals over the long history of complex function theory, complex Gaussian quadrature is in need of motivation. Gaussian quadrature in the complex plane yields approximations of certain sums connected with the bi-conjugate gradient method. The scattering amplitude $c^{\mathrm{T}}A^{-1}b$ is an example where A is a discretization of a differential–integral operator corresponding to the scattering problem and b and c are given vectors. The usual method to estimate this is to use $c^{\mathrm{T}}x^{(k)}$. A result of Warnick is that this is identically equal to the complex Gaussian quadrature estimate of $1/\lambda$. Complex Gaussian quadrature thereby replaces this particular inner product in the estimate of the scattering amplitude.

Keywords: scattering, scattering cross section, biconjugate gradient methods, constraint, signal processing

AMS subject classification: primary 41A55, 65D32; secondary 49M0, 65F10

1. Introduction

In this paper, we focus on an extension of Gaussian quadrature to the approximation of integrals along an arc in the complex plane, an extension that we call *complex Gaussian quadrature*. Quadrature formulas have never found a prominent role in the

* Effort sponsored by the Air Force Office Scientific Research, Air Force Materials Command, USAF, under grant number 1-5-20644 at UIUC. The US Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conslusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the US Government. This work also supported in part by NASA CAN NCC S5-153 ("Coalescing Binary Neutron Stars").

long history of complex function theory, even though contour integrals and the residue theorem are central to the field. A natural question, therefore, is this: Why would anyone care anything about complex Gaussian quadrature?

1.1. Motivation

Our answer to this question is the *scattering amplitude* [5–7,11,39,41,42]. The scattering amplitude is $c^{\mathrm{T}}A^{-1}b$, where $A = (a_{ij})_{N\times N}$ is a nonsingular matrix, b is a given right hand side vector of a system $Ax = b$, and c is a given vector. (More will be said below on the role of c.) The connection to complex Gaussian quadrature is through the bi-conjugate gradient iterative method (BCG) for solving $Ax = b$, a connection we explain next.

A popular method to solve $Ax = b$ is BCG, from which one may obtain the estimate $c^{\mathrm{T}}x^{(k)} \approx c^{\mathrm{T}}x = c^{\mathrm{T}}A^{-1}b$. BCG also generates a set of polynomials that yield complex Gaussian quadrature formulas [24,49]. Since the scattering amplitude can be interpreted as an integral, the result is that complex Gaussian quadrature yields an estimate of the scattering amplitude. An observation of Warnick [49] is that the Gaussian quadrature estimate is mathematically equivalent to $c^{\mathrm{T}}x^{(k)}$. In the case that A is Hermitian positive definite (HPD), this fact is implicit in an equality of Dahlquist et al. [14,16]. *Thus complex Gaussian quadrature avoids computing the inner product of $\overline{c}$ and $x^{(k)}$, which is notoriously inefficient on high performance processors.* The computational gain, as seen in section 3.1, results from avoiding the flop and communication cost of an inner product. This gain justifies a detailed presentation of complex Gaussian quadrature. Our approach is to present complex Gaussian quadrature as a mostly straightforward extension of classical Gaussian quadrature, especially as it relates to the connection between complex Gaussian quadrature and the scattering amplitude. The algorithm to compute the weights and nodes of a complex Gaussian quadrature formula extends one due to Golub and Welsch in the real case [28]; also see the recent text of Gautschi [22, pp. 69–74].

1.2. Other work

Our paper synthesizes Gaussian quadrature, integration in the complex plane, formally orthogonal polynomials, and BCG polynomials together with an important application to computing the scattering amplitude by means of the identity of Warnick. As an aid to the reader we have followed the traditional path to Gaussian quadrature with the slight changes necessary for the complex case. There are other approaches, though: In their paper [19], Freund and Hochbruck sketch complex Gaussian quadrature elegantly but concisely. Their work is related to the work of Golub et al., which we look at next.

Since the seventies, Golub, in a series of articles with various co-authors, has applied Gaussian quadrature to discrete integrals, which are objects less familiar than the integrals appearing on the pages of a calculus book but objects that are integrals nevertheless. Sums such as the inner products arising from the conjugate gradient (CG) method are an especially useful interpretation of discrete integrals. An example of how Gaussian quadrature can be used in this way is the estimation of the (square of the) A-norm of the

error, $x - x^{(k)}$, at each step. There are other ingenious examples [13,16,25,26] besides the norm of the error and, of interest to us, the scattering amplitude.

The classical CG method is restricted to symmetric positive definite (SPD) systems or more generally Hermitian positive definite (HPD) systems. (An SPD system is one for which the matrix is SPD and similarly for HPD systems.) The inner products are discrete integrals over the real line, suited for classical Gaussian quadrature. However, the scattering amplitude $c^{\mathrm{T}}A^{-1}b$ is not limited to HPD matrices. Moreover, for a non-Hermitian matrix, sums such as $c^{\mathrm{T}}A^{-1}b$ are integrals along arcs in the complex plane. We are thus led to complex Gaussian quadrature.

We note that, in at least one instance, the work of Golub and his colleagues on Gaussian quadrature and iterative methods leads up to complex Gaussian quadrature. In their paper [10], Calvetti et al. apply the method of moments with a complex measure, a non-classical version of the theory, to a real nonsymmetric matrix, thereby relating their approach implicitly to complex Gaussian quadrature.

We mention in passing that besides Gaussian quadrature, there are other approaches to quadrature in the complex plane as well as in higher dimensional spaces, notably *quadrature domains*. An example of a quadrature domain in N-space is the unit ball, Ω, which is such that for a harmonic function u, there is the quadrature-like formula, $u(\xi) = (\int_{\Omega} u(x)\,\mathrm{d}x)/|\Omega|$, where ξ is the center of Ω and $|\Omega|$ is the volume of Ω (see [45]). Warnick [49] and, separately, a reviewer of this paper have suggested that in the limit as N increases the spectrum may fill a quadrature domain and that the vanishing of an integral may relate to the breakdown of BCG. We do not pursue the topic any further here.

We also call attention to the paper of Warnick on using Gaussian quadrature to compute $b^{\mathrm{T}}A^{-1}b$ [49]. In his paper, Warnick assumes A is HPD and the iterative method is the CG method. He derives a power series expansion for $b^{\mathrm{T}}x$ along the lines of the power expansion of a Stieltjes transform [34, pp. 580ff]. The power series is then rewritten as a continued fraction. The even approximants of the continued fraction [34, (cf. p. 475)], for the case when the power series variable is zero, turn out to be $b^{\mathrm{T}}x^{(k)}$. In the general case, when A is an arbitrary nonsingular matrix, and the scattering amplitude is $c^{\mathrm{T}}x$, Warnick proved in [51] that the Gaussian quadrature estimate of $c^{\mathrm{T}}x$ is $c^{\mathrm{T}}x^{(k)}$. In the appendix of our paper, we state and prove this result, which we call Warnick's theorem.

1.3. Terminology and notation

The signal processing interpretation of given vector c is this: $Ax = b$ determines the field x from the signal b. The signal is received on an antenna, c, the right-hand side of an adjoint system $A^*\widetilde{x} = \bar{c}$. The signal received by the antenna is then $c^{\mathrm{T}}x$.

If b and c represent incoming and outgoing waves, respectively, and the operator A relates the incoming and scattered fields on the surface of an object, then $c^{\mathrm{T}}x$ is a scattering amplitude. The *scattering cross section*, $|c^{\mathrm{T}}A^{-1}b|^2$, is often sought in scattering computations as well as the amplitude. We note that the meaning of these terms

may vary among the applied disciplines, but the goals are the same, namely to compute $c^{\mathrm{T}}A^{-1}b$.

Not only does the scattering amplitude arise in signal processing, but it also arises in nuclear physics (cf. [1, p. 64]), quantum mechanics (cf. [37, p. 503]), and other disciplines (cf. Pierce and Giles [43, p. 247]).

Most of the paper is concerned with polynomials, which we use Greek letters to denote. Latin letters, such as f, g, h, denote general complex functions that could be polynomials. We also use g_i to denote the ith eigenvector of a matrix. Latin letters toward the end of the alphabet, such as u and v stand for vectors, which is customary notation. However, we use w to denote a weight function in either an integral or a discrete sum (and we interpret discrete sums to be integrals). Weight w may be real or complex. We use Greek letters in other ways, such as ω_i to denote the ith coefficient in a Gaussian quadrature formula. These coefficients are also called *weights* and are not to be confused with the weight function, w, in an integral. We use ξ as a real variable and ζ as complex.

Where necessary, the eigenvectors of matrix A are assumed to be complete.

1.4. Outline

In section 2 we look at the integrals and bilinear forms in the complex plane that are connected with Gaussian quadrature. In section 3 we review the theory of classical orthogonal polynomials and the conjugate gradient algorithm. In section 4 we discuss orthogonal polynomials in the complex plane. To extend Gaussian quadrature, we resort to formally orthogonal polynomials in place of orthogonal polynomials. We discuss formally orthogonal polynomials in section 5 and include an algorithm for computing formally orthogonal polynomials. Once we have identified the class of formally orthogonal polynomials, the extension of Gaussian quadrature to the complex plane, in section 6, is straightforward. Experiments are summarized in section 7. We discuss the BCG method in section 8. We make a simplifying assumption about BCG which is that each step is defined and no look ahead scheme is needed. The relation among BCG, Gaussian quadrature, and formally orthogonal polynomials is addressed in sections 9 and 10 with further comments and conclusions included in section 11. In addition, in section 11, there is an outline of the generalization of Gauss–Radau and Gauss–Lobatto quadrature to the complex plane. There is an appendix with a proof and statement of Warnick's theorem.

2. Inner products and bilinear forms

We have already introduced SPD and HPD to mean symmetric positive definite and Hermitian positive definite, respectively. If $A = (a_{ij})_{N\times N}$ is a matrix, the *transpose* is $A^{\mathrm{T}} = (a_{ji})$, and the *Hermitian* transpose is $A^{*} = (\overline{a}_{ji})$. For $u^{\mathrm{T}} = (u_1, \ldots, u_N)$,

$v^{\mathrm{T}} = (v_1, \ldots, v_N)$, let

$$\langle u, v\rangle := \sum_{i=1}^{N} u_i \overline{v}_i \tag{1}$$

be the familiar *Euclidean inner product*. It is *Hermitian bilinear*, also *sesquilinear*, in that

$$\big\langle \alpha u^{(1)} + \beta u^{(2)}, v\big\rangle = \alpha\big\langle u^{(1)}, v\big\rangle + \beta\big\langle u^{(2)}, v\big\rangle,$$

but

$$\big\langle u, \alpha v^{(1)} + \beta v^{(2)}\big\rangle = \overline{\alpha}\big\langle u, v^{(1)}\big\rangle + \overline{\beta}\big\langle u, v^{(2)}\big\rangle.$$

The square of the *Euclidean norm* is $\|u\|^2 := \langle u, u\rangle$.

A *bilinear form* on a vector space is a function $(u, v)_w$ on two vectors as is the Euclidean inner product (1). It is linear in each variable rather than Hermitian bilinear, i.e.,

$$\big(\alpha u^{(1)} + \beta u^{(2)}, v\big)_w = \alpha\big(u^{(1)}, v\big)_w + \beta\big(u^2, v\big)_w$$

and

$$\big(u, \alpha v^{(1)} + \beta v^{(2)}\big)_w = \alpha\big(u, v^{(1)}\big)_w + \beta\big(u, v^{(2)}\big)_w.$$

(The w-notation will be explained in a moment.) A simple example is the bilinear form $(u, v) := \langle u, \overline{v}\rangle$. This form is of course an inner product over a real vector space, but we assume vector spaces are complex in which case it is *not positive*, i.e., there is a nonzero complex vector u for which (u, u) is not positive and another vector $\widetilde{u}$, which could be the same vector, for which $(\widetilde{u}, \widetilde{u})$ is not negative. (A positive form is one for which $(u, u) > 0$ whenever $u \neq 0$.) In the case of a general bilinear form the same situation holds, namely that there is a nonzero vector u such that $(u, u)_w$ is not positive and a $\widetilde{u}$ such that $(\widetilde{u}, \widetilde{u})_w$ is not negative.

We do not discuss bilinear forms in a completely general way because the only bilinear form on vectors that we make use of is:

$$(u, v)_w = \sum_{i=1}^{N} u_i v_i w_i, \tag{2}$$

where $u^{\mathrm{T}} = (u_1, \ldots, u_N)$, $v^{\mathrm{T}} = (v_1, \ldots, v_N)$, and $w^{\mathrm{T}} = (w_1, \ldots, w_N)$ is a vector of coefficients, in this paper usually referred to as *weights*. A general treatment of bilinear forms appears in a paper of Freund [18].

Vectors u and v, $u \neq v$, are *orthogonal* if $\langle u, v\rangle = 0$, and *formally orthogonal* if $(u, v)_w = 0$.

3. Classical orthogonal polynomials

The standard reference for orthogonal polynomials is Szegő's classical monograph [48]; also see [12, pp. 104ff; 20, pp. 238ff; 22, pp. 69–74; 52, pp. 48–81]. The theory begins with a *weighted inner product of two real functions f and g*, defined on an interval of the real axis (a, b), $a < b$ by

$$\langle f, g\rangle_w := \int_a^b f(\xi)g(\xi)w(\xi)\,\mathrm{d}\xi, \tag{3}$$

where $w(\xi)$ is a positive *weight function* over (a, b). (We make some simplifications. Gaussian quadrature in principle approximates integrals on any measurable set. Thus we could use the union of intervals or more complicated sets as the domain of integration but, for our purpose, an interval suffices. Also, we will assume integrability.) In the discrete case, (3) becomes

$$\langle f, g\rangle_w := \sum_{i=1}^{M} f(\xi_i)g(\xi_i)w(\xi_i) \tag{4}$$

for some number M equal to the total number of discrete points.

3.1. Orthogonality

Orthogonality applies generally to functions in a function space with an inner product. Thus functions f and g are *orthogonal* if $\langle f, g\rangle_w = 0$. Since our concern is with polynomials only, henceforth we neglect the generality of many well-known concepts.

We shall summarize some well known facts about real orthogonal polynomials. Fuller discussions are in the references cited above, but see, in particular, [23,52] and [22, p. 164].

We shall use ϕ_i to denote an orthogonal polynomial of degree i. Real orthogonal polynomials ϕ_i, $i = 0, \ldots,$ satisfy a three-term recurrence, following the style in the recent text of Heath [33, p. 224],

$$\phi_i(\xi) = \big(\widehat{\alpha}_i\xi + \widehat{\beta}_i\big)\phi_{i-1}(\xi) - \widehat{\gamma}_i\phi_{i-2}(\xi), \quad 1 \leqslant i, \tag{5}$$

where $\widehat{\alpha}_i$, $\widehat{\beta}_i$ and $\widehat{\gamma}_i$ are real, $\phi_{-1}(\xi)$ is 0, and $\phi_0(\xi)$ is real and nonzero. The key to deriving the recurrence is that the independent variable transfers from one side to the other in the inner product:

$$\langle \xi\phi, \psi\rangle = \langle \phi, \xi\psi\rangle_w. \tag{6}$$

Remark 1. The transference property in (6) holds if f and g are complex valued functions and the inner product is either

$$\langle f, g\rangle_w := \int_a^b f(\xi)\overline{g(\xi)}w(\xi)\,\mathrm{d}\xi, \tag{7}$$

or

$$\langle f, g\rangle_w := \sum_{i=1}^{M} f(\xi_i)\overline{g(\xi_i)}w(\xi_i)$$

instead of (3) or (4). For a and b real, ξ and ξ_i are assumed to lie in $[a, b]$. The transference property holds because ξ in (6) is real.

Recurrence (5) is equivalent to the matrix–vector equation

$$\xi\Phi(\xi) = S_k\Phi(\xi) + \big(\phi_k(\xi)/\widehat{\alpha}_k\big)\mathbf{e}_k \tag{8}$$

where $\Phi(\xi) = (\phi_0(\xi), \phi_1(\xi), \ldots, \phi_{k-1}(\xi))^{\mathrm{T}}$, $\mathbf{e}_k$ is the kth unit vector $(0, 0, \ldots, 1)^{\mathrm{T}}$, and $S_k = (s_{ij})$ is the $k \times k$ tridiagonal matrix defined by $s_{i-1,i} = \widehat{\gamma}_i/\widehat{\alpha}_i$, $s_{ii} = -\widehat{\beta}_i/\widehat{\alpha}_i$ and $s_{i,i+1} = 1/\widehat{\alpha}_i$ (cf. [28]). This matrix is called the *Jacobi matrix*, [22, pp. 163–64; 28]. In practice, CG runs for a number of steps much less than the number, N, of unknowns. The cost of computing the eigen-structure of a Jacobi matrix is therefore much less than the flop and communication cost of an inner product such as $c^{\mathrm{T}}x^{(k)}$.

If $\widehat{\xi}$ is a root of the kth orthogonal polynomial, ϕ_k, then (8) yields

$$\widehat{\xi}\Phi\big(\widehat{\xi}\big) = S_k\Phi\big(\widehat{\xi}\big).$$

Therefore, $\widehat{\xi}$ is an eigenvalue of the matrix S_k with corresponding eigenvector $\Phi(\widehat{\xi})$, and conversely an eigenvalue of S_k is a root of ϕ_k.

3.2. Conventions for a three-term recursion

The two conditions that ϕ_i be orthogonal to ϕ_{i-1} and to ϕ_{i-2} determine the recurrence coefficients $\widehat{\alpha}_i$, $\widehat{\beta}_i$ and $\widehat{\gamma}_i$ in (5). Two conditions for three parameters yields an underdetermined system with an infinite number of solutions for the three parameters. One more (nontrivial) condition would determine the three parameters uniquely. The extra condition results from some special way in which orthogonal polynomials are being used. Our reason for bringing this up is that each one of these special ways gives rise to its own convention for writing the recurrence formula, which may not look like (5). To help sort through the confusion, we have listed here the different conditions and conventions, some of which the reader may have seen and be familiar with. Our remarks apply not only to orthogonal polynomials, to which our discussion has so far been restricted, but also to formally orthogonal polynomials, which are the basis for Gaussian quadrature in the complex plane.

Monic polynomials. Let $\phi_{-1} = 0$, $\phi_0 = 1$. Let $\widehat{\alpha}_i = 1$ in (5) so that (5) becomes

$$\phi_i(\xi) = \big(\xi + \widehat{\beta}_i\big)\phi_{i-1}(\xi) - \widehat{\gamma}_i\phi_{i-2}(\xi).$$

Stiefel uses this convention in [47] and Gautschi in [22, p. 164], but we do not use monic polynomials in our paper.

Residual polynomials. Residual polynomials [47] are orthogonal polynomials for which $\phi_i(0) = 1$. The new condition the recurrence parameters in (5) now have to satisfy is:

$$1 = \widehat{\beta}_i - \widehat{\gamma}_i. \tag{9}$$

By convention, a different symbol than ϕ_i, such as R_i, is used to denote a residual polynomial. Moreover, there are various ways to rewrite the recurrence relation depending on how (9) is used to eliminate one of the three parameters. Residual polynomials are used in sections 9 and 10.

Normalized polynomials. A sequence of orthogonal polynomials is *normalized* if $\langle \phi_i, \phi_i \rangle_w = 1$. In the case of formally orthogonal polynomials the inner product $\langle f, g \rangle_w$ is replaced with a bilinear form, where f and g are general complex valued functions that may or may not be polynomials. When the polynomials are normalized, the Jacobi matrix S_k turns out to be symmetric. In this case, we rewrite the recurrence relation (5) in different notation:

$$\xi \phi_i(\xi) = \frac{\sigma_i}{\widetilde{\nu}_i}\phi_{i-1}(\xi) + \frac{\nu_i}{\widetilde{\nu}_i}\phi_i(\xi) + \frac{1}{\widetilde{\nu}_i}\phi_{i+1}(\xi), \tag{10}$$

which is in the style of Wilf [52]. We shall use this convention in section 5.2, though for formally orthogonal polynomials instead of orthogonal polynomials.

3.3. Conjugate gradient method, orthogonal polynomials, and Gaussian quadrature

The conjugate gradient (CG) method of Hestenes and Stiefel solves the linear system $Ax = b$, where A is HPD. The standard algorithmic form of CG is:

Algorithm 1. The Conjugate Gradient Method.
Purpose: To solve $Ax = b$.
Assumption: A is HPD.
Initialization: Choose $x^{(0)}$. Set

$$r^{(0)} := b - Ax^{(0)}$$
$$d^{(0)} := r^{(0)}$$

Do $k = 0, \ldots,$ until convergence

$$\alpha_k := \frac{\|r^{(k)}\|^2}{\langle d^{(k)}, A\, d^{(k)} \rangle}$$
$$x^{(k+1)} := x^{(k)} + \alpha_k\, d^{(k)}$$
$$r^{(k+1)} := r^{(k)} - \alpha_k A\, d^{(k)}$$
$$\beta_k := \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$$
$$d^{(k+1)} := r^{(k+1)} + \beta_k\, d^{(k)}$$

End Do.

See, for example, [27, p. 527; 44, p. 179].

Vectors $d^{(k)}$ and $r^{(k)}, k = 0, \ldots,$ are called the *direction vectors* and *residual vectors*, respectively. The direction vectors, satisfy the A-orthogonality condition: $\langle d^{(i)}, A\, d^{(j)} \rangle = 0$ if and only if $i \neq j$. The residual vectors satisfy the standard Euclidean orthogonality condition: $\langle r^{(i)}, r^{(j)} \rangle = 0$ if and only if $i \neq j$.

For A, the *Krylov subspace* is $V_k := \operatorname{span}\{s, As, \ldots, A^{k-1}s\}$. Usually it is not necessary to show the dependency of V_k on A and the *seed vector* s. Assume A and s are such that the dimension of V_k is k.

For any v in V_k, we have

$$v = P(A)s,$$

where $P(\lambda) = \sum_{i=0}^{k} \gamma_i \lambda^i$ is a polynomial of degree k, with real coefficients γ_i. We now assume the special choice of the initial residual vector for the seed vector: $s = r^{(0)}$. Since the CG residual and direction vectors $r^{(k)}$ and $d^{(k)}$ lie in V_k, they correspond in a one-to-one fashion to polynomials, $R_k(\lambda)$ and $D_k(\lambda)$, which are called the *residual* and *direction polynomials* respectively, [35, pp. 428–429].

Next we relate the inner product in CG to the inner product of polynomials. Expand the initial residual in terms of eigenvectors, $r^{(0)} = \sum_{i=1}^{N} \theta_i \lambda_i g_i$, with λ_i and g_i an eigenpair of A and with θ_i a real or complex coefficient. Since A is HPD, λ_i is also real. Let

$$w(\lambda) = \begin{cases} \left(|\theta_i|\lambda_i\right)^2 & \text{for } \lambda = \lambda_i,\ i = 1, \ldots, N, \\ 0 & \text{for } \lambda \neq \lambda_i,\ i = 1, \ldots, N. \end{cases}$$

Then the residual polynomials are orthogonal with respect to $w(\lambda)$. As noted in section 3.2, residual polynomials have the special property that $R_k(0) = 1$, which expresses the "consistency" of the algorithm, i.e., that the solution satisfies certain algorithmic formulas. They also satisfy the minimization property

$$\int \left|R_k(\lambda)\right|^2 \frac{w(\lambda)}{\lambda}\, \mathrm{d}\lambda \leqslant \int \left|P_k(\lambda)\right|^2 \frac{w(\lambda)}{\lambda}\, \mathrm{d}\lambda,$$

where P_k is any residual polynomial of degree k or less, i.e., $P_k(0) = 1$. This is a restatement of the fact that if $e^{(k)} := x - x^{(k)}$ is the error, then $e^{(k)} = R_k(A)e^{(0)}$ and $\langle e^{(k)}, Ae^{(k)} \rangle$ is minimized.

3.4. Classical connection to Gaussian quadrature

A *quadrature rule* is a finite sum that approximates an integral. The classical Gaussian quadrature rule has the form

$$\int_a^b f(\xi) w(\xi)\, \mathrm{d}\xi = \sum_{i=1}^{k} \omega_i f(\xi_i) + \varepsilon_k, \tag{11}$$

where ε_k is the error. It is *exact*, i.e., $\varepsilon_k = 0$, if $f(\xi)$ is a polynomial of degree $2k - 1$ or less. The ω_i's are real and are called *weights* and the ξ_i's are called *nodes*. The nodes are

real and the roots of the degree k polynomial from the family of polynomials orthogonal with respect to w (cf. [12, p. 302] or [22, pp. 157ff]). These orthogonal polynomials could come from CG, an observation Golub et al. [13,16,25,26] (an incomplete list of references) have exploited.

The weights and nodes may be computed from the Jacobi matrix S_k in (8). (The details are omitted, see [22, pp. 164–165]. We give a Jacobi matrix in this paper but, for formally orthogonal polynomials, see section 6.2.)

3.5. Classical connection to the Lanczos algorithm

We will make only passing reference to the Lanczos algorithm even though we focus on the BCG algorithm later which is derived from the Lanczos algorithm. The well-known, all-important property of the Lanczos algorithm is that it yields a tridiagonal matrix at each step, the eigenvalues of which approximate the eigenvalues of A [27, p. 473]. This tridiagonal matrix is, in fact, the Jacobi matrix of sections 3.1 and 3.4 above, as well as of section 6.2.

The algorithm generates an orthogonal basis of each Krylov subspace V_k at each step which the CG method also does, using seed vector $r^{(0)}$. (The orthogonal basis that CG generates is the set of $r^{(k)}$'s.) If the Lanczos algorithm uses the same seed vector, the two algorithms are equivalent, but have different goals: in one case, to solve a linear system, and, in the other, to estimate eigenvalues.

4. Orthogonal polynomials in the complex plane

Analogous to (3), the *weighted inner product for complex valued functions f and g* is defined as:

$$\langle f, g\rangle_w := \int_\gamma f(\zeta)\overline{g(\zeta)}w(\zeta)|\mathrm{d}\zeta| \tag{12}$$

for $w(\zeta)$ a positive weight function, and γ an arc [48, p. 365].

As before in section 3, if functions f and g satisfy $\langle f, g\rangle_w = 0$, they are said to be *orthogonal*. In the real case, a sequence of orthogonal polynomials $\phi_i, i = 0, \ldots,$ are related through a three-term recurrence relation. But, in general, in the complex case there is no three-term recurrence relation [38]. (An exception occurs on the unit circle in the case of so-called Szegő polynomials [4,31].) An informal explanation as to why there is no three-term recurrence is that the key transference property, $\langle \zeta f, g\rangle_w = \langle f, \zeta g\rangle_w$, no longer holds: In general, for an arc in the complex plane,

$$\langle \zeta f, g\rangle_w \neq \langle f, \zeta g\rangle_w,$$

in contrast to (6). We shall not pursue this avenue any further but instead turn next to formally orthogonal polynomials.

5. Formally orthogonal polynomials

We have seen that orthogonal polynomials in the complex plane (cf. [48]) do not in general satisfy a three-term recurrence [38]. For simplicity we strive to obtain three-term recurrences. More general recurrences, however, also yield Gaussian quadrature [19, p. 379].

In place of orthogonal polynomials, we resort to *formally orthogonal polynomials*, i.e., a set of polynomials $\phi_0, \ldots$, for which $(\phi_i, \phi_j)_w = 0$ if $i \neq j$, where $(\pi, \psi)_w$ is a bilinear form, examples of which are below.

Observe that we leave open the possibility that $(\phi_i, \phi_i)_w = 0$. In the rest of this paper, in order that algorithms execute, we will assume that $(\phi_i, \phi_i)_w \neq 0$ and that formally orthogonal polynomials satisfy a three-term recurrence. Such formally orthogonal polynomials are said to be *regular* in [19].

5.1. Continuous and discrete integrals in the complex plane

We present an overview of some of the ways in which bilinear forms arise.

Complex integrals. The integral

$$\int_\gamma h(\zeta) w(\zeta)\, \mathrm{d}\zeta, \tag{13}$$

where h is an analytic function, w is a complex valued weight function, and $\mathrm{d}\zeta = \mathrm{d}\xi + \mathrm{i}\mathrm{d}\eta$, is standard in complex function theory. The domain of integration, γ, is an arc. It is an elementary fact of complex function theory that the integral is independent of γ. Therefore, if γ connects a to z and a is constant then the integral defines a function of z the derivative of which is h, when $w(\zeta) = 1$.

Complex integrals yield the bilinear form

$$(\!(f, g)\!)_w := \int_\gamma f(\zeta) g(\zeta) w(\zeta)\, \mathrm{d}\zeta. \tag{14}$$

This bilinear form is symmetric but not positive and not an inner product. Symmetry means that $(\!(f, g)\!)_w = (\!(g, f)\!)_w$. Note that in general $(\!(f, f)\!)_w \not> 0$, $f \neq 0$. In general, $(\!(f, f)\!)_w$ is nonreal; moreover it may even happen that $(\!(f, f)\!)_w = 0$ for $f \neq 0$.

Two dimensional integrals. If we identify the complex plane with the real 2D plane, then integration and Gaussian quadrature over the complex plane connects with the potential theory interpretation of iterative methods [15].

Line integrals. In this case there is a slight though important change from (13). A line integral is:

$$\int_\gamma h(\zeta) w(\zeta) |\mathrm{d}\zeta|, \tag{15}$$

where $\zeta = \xi + \mathrm{i}\eta$ is a complex variable and $|\mathrm{d}\zeta|$ means arc length, i.e., $|\mathrm{d}\zeta| = \sqrt{\mathrm{d}\xi^2 + \mathrm{d}\eta^2}$.

Line integrals yield the bilinear form,

$$(f, g)_w := \int_\gamma f(\zeta)g(\zeta)w(\zeta)|\mathrm{d}\zeta|. \tag{16}$$

In contrast to the bilinear form $(\!(f, g)\!)_w$ in (14), this form yields an inner product

$$\langle f, g\rangle_w := \left(f(\zeta), \overline{g(\zeta)}\right)_w := \int_\gamma f(\zeta)\overline{g(\zeta)}w(\zeta)|\mathrm{d}\zeta|, \tag{17}$$

provided that w is positive on γ. Szegő's classic monograph on orthogonal polynomials [48] includes an extension of the real theory to the complex plane in which he uses the line integral in order to work with an inner product.

Discrete integrals. The discrete bilinear form,

$$(f, g)_w := \sum_{i=1}^{M} f(\zeta_i)g(\zeta_i)w(\zeta_i) \tag{18}$$

may be viewed as a line integral over an arc connecting the points $\zeta_1, \ldots, \zeta_N$ with a complex valued discrete weight, w, defined on the arc. It is a discrete version of either (14) or (16) and is the same as (4). We choose to interpret (18) as a line integral. Sums of this kind arise from BCG.

5.2. Generating formally orthogonal polynomials

In this subsection, we view $(\pi, \phi)_w$ as any symmetric bilinear form on polynomials. If they exist and $(\phi_k, \phi_k)_w \neq 0$, formally orthogonal polynomials satisfy a familiar coupled three-term recurrence relation. We state an algorithm next and then prove that the polynomials it generates *under the assumption that they exist* are normalized formally orthogonal polynomials. A formally orthogonal polynomial is said to be *normalized*, if $(\phi_k, \phi_k)_w = 1$ (cf. (10)).

Algorithm 2. Generating Formally Orthogonal Polynomials.
Purpose: To generate normalized formally orthogonal polynomials.
Assumption: No division by zero occurs: $(\phi_k, \phi_k)_w \neq 0, 0 \leqslant k$.
Initialization: Set $\phi_0(\zeta) := [(1, 1)_w]^{-1/2}$.
Choose integer Limit, $2 \leqslant$ Limit.
Set

$$\frac{\nu_0}{\widetilde{\nu}_0} := (\zeta\phi_0, \phi_0)_w \tag{19}$$

$$\widetilde{\nu}_0 := \left[\left(\zeta\phi_0 - \frac{\nu_0}{\widetilde{\nu}_0}\phi_0, \zeta\phi_0 - \frac{\nu_0}{\widetilde{\nu}_0}\phi_0\right)_w\right]^{-1/2} \tag{20}$$

$$\nu_0 := \widetilde{\nu}_0 \frac{\nu_0}{\widetilde{\nu}_0}$$
$$\phi_1(\zeta) := \widetilde{\nu}_0 \zeta \phi_0(\zeta) - \nu_0 \phi_0(\zeta)$$

Do $k := 1, \ldots,$ Limit $- 1$

$$\frac{\nu_k}{\widetilde{\nu}_k} := (\zeta\phi_k, \phi_k)_w \tag{21}$$
$$\frac{\sigma_k}{\widetilde{\nu}_k} := (\zeta\phi_k, \phi_{k-1})_w \tag{22}$$
$$P(\zeta) := \zeta\phi_k(\zeta) - \frac{\nu_k}{\widetilde{\nu}_k}\phi_k(\zeta) - \frac{\sigma_k}{\widetilde{\nu}_k}\phi_{k-1}(\zeta)$$
$$\widetilde{\nu}_k := \left[(P, P)_w\right]^{-1/2} \tag{23}$$
$$\nu_k := \widetilde{\nu}_k \frac{\nu_k}{\widetilde{\nu}_k}$$
$$\sigma_k := \widetilde{\nu}_k \frac{\sigma_k}{\widetilde{\nu}_k}$$
$$\phi_{k+1}(\zeta) := \widetilde{\nu}_k \zeta \phi_k(\zeta) - \nu_k \phi_k(\zeta) - \sigma_k \phi_{k-1}(\zeta) \tag{24}$$

End Do.

Theorem 1. Assume $(\phi_k, \phi_k) \neq 0,\ 0 \leqslant k \leqslant N$, in algorithm 2. Then algorithm 2 generates a sequence of normalized formally orthogonalpolynomials, $\phi_i,\ i = 1, \ldots, N$, such that ϕ_i is of exact degree i and $(\phi_i, \phi_j)_w = 0$ for $0 \leqslant i, j \leqslant k,\ i \neq j$.

Proof. The proof follows the real case [47, pp. 2–3] and is a simplification of the one in [46, pp. 49–50].

The proof is by induction on k. To establish the theorem for $k = 1$, observe that relation (19) together with $\phi_1(\zeta)/\widetilde{\nu}_0 = \zeta\phi_0(\zeta) - (\nu_0/\widetilde{\nu}_0)\phi_0(\zeta)$ yield $(\phi_1, \phi_0)_w = 0$. The normalization of ϕ_1 follows from (20).

Assume for $0 \leqslant i, j \leqslant k,\ i \neq j$, we have $(\phi_i, \phi_j)_w = 0$.

Conditions (21), (22) yield $(\phi_{k+1}, \phi_k)_w = 0$ and $(\phi_{k+1}, \phi_{k-1})_w = 0$, respectively, whereas (23) gives $(\phi_{k+1}, \phi_{k+1})_w = 1$.

It remains to show that ϕ_{k+1} is formally orthogonal to $\phi_i,\ i \leqslant k - 2$.

Since by assumption $(\phi_k, \phi_i)_w = 0$ and $\widetilde{\nu}_k(\phi_{k-1}, \phi_i)_w = 0$, we have

$$(\phi_{k+1}, \phi_i)_w = \widetilde{\nu}_k(\zeta\phi_k, \phi_i)_w.$$

From $(\zeta\pi, \psi)_w = (\pi, \zeta\psi)_w$, the fact that $\zeta\phi_i(\zeta) = \phi_{i+1}(\zeta)/\widetilde{\nu}_i + (\nu_i/\widetilde{\nu}_i)\phi_i(\zeta) + (\sigma_i/\widetilde{\nu}_i)\phi_{i-1}(\zeta)$ and the assumption that $(\phi_k, \phi_i)_w = 0$ for $i \leqslant k - 2$, it follows that $(\phi_k, \zeta\phi_i)_w = 0$. □

Remark 2. The existence of formally orthogonal polynomials is not equivalent to $(\phi_k, \phi_k)_w \neq 0,\ 0 \leqslant k$. For, if γ is an integrable, closed curve, such as a circle, then (see,

for example, (14)) $((\phi_k, \phi_k))_w = 0,\ 0 \leqslant k$, for any sequence of polynomials of exact degree.

Henceforth when we refer to a set of formally orthogonal polynomials, we continue to assume that they exist and that $(\phi_k, \phi_k) \neq 0,\ 0 \leqslant k \leqslant K$, for some K.

6. Complex Gaussian quadrature

The analog of (11), for the case that the domain of integration is an arc γ in the complex plane, is

$$\int_\gamma f(\zeta) w(\zeta) |\mathrm{d}\zeta| = \sum_{i=1}^{k} \omega_i f(\zeta_i) + \varepsilon_k, \tag{25}$$

a formula that is *exact*, i.e., $\varepsilon_k = 0$, when $f(\zeta)$ is a polynomial of degree $2k-1$ or less. As before in section 3.4, the ω_i's are called the *weights* and the ζ_i's are called the *nodes*.

6.1. Deriving nodes and weights

We shall sketch a derivation of expressions for the nodes ζ_i and the weights ω_i in (25).

Let $\phi_0, \ldots,$ be a sequence of formally orthogonal polynomials, i.e., $(\phi_i, \phi_j)_w = 0, i \neq j$. The derivation proceeds as in the real case but with formal orthogonality in place of orthogonality.

Assume that the integrand $f(\zeta)$ is a polynomial of degree $2k-1$. The division algorithm for polynomials gives

$$f(\zeta) = \phi_k(\zeta) Q_{k-1}(\zeta) + r_j(\zeta), \tag{26}$$

where Q_{k-1} is the quotient and r_j is the remainder of degree $k-1$ or less. This yields

$$\int_\gamma f(\zeta) w(\zeta) |\mathrm{d}\zeta| = \int_\gamma \phi_k(\zeta) Q_{k-1}(\zeta) w(\zeta) |\mathrm{d}\zeta| + \int_\gamma r_j(\zeta) w(\zeta) |\mathrm{d}\zeta|.$$

Polynomial ϕ_k is formally orthogonal to any polynomial of degree $k-1$ or less. Therefore,

$$\int_\gamma f(\zeta) w(\zeta) |\mathrm{d}\zeta| = \int_\gamma r_j(\zeta) w(\zeta) |\mathrm{d}\zeta|. \tag{27}$$

Let the k roots of $\phi_k(\zeta)$ be $\zeta_1, \ldots, \zeta_k$, and let

$$l_i(\zeta) = \prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{\zeta - \zeta_j}{\zeta_i - \zeta_j}$$

be the ith Lagrange basis polynomial which is such that $l_i(\zeta_j) = \delta_{ij}$, where δ_{ij} is the Dirac delta function. (Our notation for the Lagrange basis polynomial is standard but conflicts

with our convention elsewhere that a subscript denotes polynomial degree. Thus l_i is of degree $k-1$ rather than degree i.) Since $r_j(\zeta_i) = f(\zeta_i)$ for $1 \leqslant i \leqslant k$, we have $r_j(\zeta) = \sum_{i=1}^{k} f(\zeta_i) l_i(\zeta)$. From (27),

$$\int_\gamma f(\zeta) w(\zeta) |\mathrm{d}\zeta| = \int_\gamma \left(\sum_{i=1}^{k} f(\zeta_i) l_i(\zeta) \right) w(\zeta) |\mathrm{d}\zeta| = \sum_{i=1}^{k} f(\zeta_i) \int_\gamma l_i(\zeta) w(\zeta) |\mathrm{d}\zeta|.$$

Therefore quadrature form (25) is exact if

$$\omega_i = \int_\gamma l_i(\zeta) w(\zeta) |\mathrm{d}\zeta|. \tag{28}$$

We see that the nodes ζ_i are the roots of ϕ_k and conversely.

6.2. *Formulas for the weights continued: The Jacobi matrix*

We shall follow [52] to obtain modern formulas for the weights. We begin with an alternative expression for $l_i(\zeta)$, namely,

$$l_i(\zeta) = \frac{\phi_k(\zeta)}{(\zeta - \zeta_i)\phi_k{}'(\zeta_i)}.$$

From this and (28),

$$\omega_i \phi_k{}'(\zeta_i) = \int_\gamma \frac{\phi_k(\zeta) w(\zeta)}{\zeta - \zeta_i} |\mathrm{d}\zeta|. \tag{29}$$

Assume that $\phi_0, \ldots,$ is a normalized sequence of formally orthogonal polynomials. The recurrence for normalized formally orthogonal polynomials (cf. (10), (24)),

$$\frac{1}{\widetilde{\nu}_i} \phi_{i+1}(\zeta) = \zeta \phi_i(\zeta) - \frac{\nu_i}{\widetilde{\nu}_i} \phi_i(\zeta) - \frac{\sigma_i}{\widetilde{\nu}_i} \phi_{i-1}(\zeta),$$

is equivalent to the matrix–vector statement [52, (28), p. 55]

$$\zeta \begin{pmatrix} \phi_0(\zeta) \\ \phi_1(\zeta) \\ \vdots \\ \phi_{k-1}(\zeta) \end{pmatrix} = \begin{pmatrix} \frac{\nu_0}{\widetilde{\nu}_0} & \frac{1}{\widetilde{\nu}_1} & 0 & \ldots & 0 \\ \frac{\sigma_1}{\widetilde{\nu}_1} & \frac{\nu_1}{\widetilde{\nu}_1} & \frac{1}{\widetilde{\nu}_1} & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \frac{1}{\widetilde{\nu}_{k-2}} \\ 0 & \ldots & 0 & \frac{\sigma_{k-1}}{\widetilde{\nu}_{k-1}} & \frac{\nu_{k-1}}{\widetilde{\nu}_{k-1}} \end{pmatrix} \begin{pmatrix} \phi_0(\zeta) \\ \phi_1(\zeta) \\ \vdots \\ \phi_{k-1}(\zeta) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \frac{\phi_k(\zeta)}{\widetilde{\nu}_k} \end{pmatrix}. \tag{30}$$

In matrix–vector form, (30) is equivalent to

$$\zeta \Phi(\zeta) = S_k \Phi(\zeta) + \left(\frac{\phi_k(\zeta)}{\widetilde{\nu}_k} \right) \mathbf{e}_k, \tag{31}$$

where S_k is the Jacobi matrix (cf. (8)), $\mathbf{e}_k$ is the kth unit vector $(0, 0, \ldots, 1)^{\mathrm{T}}$, and

$$\Phi(\zeta) = \big(\phi_0(\zeta), \phi_1(\zeta), \ldots, \phi_{k-1}(\zeta)\big)^{\mathrm{T}}.$$

If ζ_i is a root of $\phi_k(\zeta)$ in (31), then

$$\zeta_i \Phi(\zeta_i) = S_k \Phi(\zeta_i). \tag{32}$$

As in the real case, we therefore see that $\Phi(\zeta_i)$ is an eigenvector of S_k, corresponding to the eigenvalue ζ_i.

The weights result from the first components of the eigenvectors of S_k [28]. Also see [21, p. 79; 29,30]. The derivation begins with the Christoffel–Darboux identities.

6.3. *Christoffel–Darboux identities*

The derivation of the Christoffel–Darboux identities is an extension of the real case [52, pp. 55–56]. We include it for the convenience of the reader; also see [8, p. 127ff].

We shall need the fact that the tridiagonal Jacobi matrix S_k is symmetric (but note that in general it is non-Hermitian).

Theorem 2.

$$S_k = \begin{pmatrix} \dfrac{\nu_0}{\widetilde{\nu}_0} & \dfrac{1}{\widetilde{\nu}_1} & 0 & \ldots & 0 \\ \dfrac{\sigma_1}{\widetilde{\nu}_1} & \dfrac{\nu_1}{\widetilde{\nu}_1} & \dfrac{1}{\widetilde{\nu}_1} & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \dfrac{1}{\widetilde{\nu}_{k-2}} \\ 0 & \ldots & 0 & \dfrac{\sigma_{k-1}}{\widetilde{\nu}_{k-1}} & \dfrac{\nu_{k-1}}{\widetilde{\nu}_{k-1}} \end{pmatrix} \tag{33}$$

is symmetric.

Proof. Since

$$\phi_{i+1}(\zeta) = \widetilde{\nu}_i \zeta \phi_i(\zeta) - \nu_i \phi_i(\zeta) - \sigma_i \phi_{i-1}(\zeta)$$

for $0 \leqslant i \leqslant k$ we have that

$$1 = (\phi_{i+1}, \phi_{i+1})_w = \widetilde{\nu}_i (\zeta \phi_i, \phi_{i+1})_w, \tag{34}$$

which, together with (22), implies that

$$\frac{\sigma_i}{\widetilde{\nu}_i} = \frac{1}{\widetilde{\nu}_{i-1}}. \qquad \square$$

We now return to deriving the Christoffel–Darboux identities. Replace ζ by ρ in (31), to obtain

$$\rho\Phi(\rho) = S_k\Phi(\rho) + \left(\frac{1}{\widetilde{\nu}_k}\right)\phi_k(\rho)\mathbf{e}_k. \tag{35}$$

Take the dot product of (31) on the left with $\Phi(\rho)$ on the right. Then take the dot product of $\Phi(\zeta)$ on the left with (35) on the right, and subtract the second resulting equation from the first. The symmetry of S_k gives

$$(\zeta - \rho)\big[\Phi(\zeta)\cdot\Phi(\rho)\big] = \left(\frac{1}{\widetilde{\nu}_k}\right)\big[\phi_{k-1}(\rho)\phi_k(\zeta) - \phi_{k-1}(\zeta)\phi_k(\rho)\big],$$

the first Christoffel–Darboux formula for ζ and ρ complex.

Dividing by $\zeta - \rho$ and rewriting the dot product yields

$$\sum_{i=0}^{k-1}\phi_i(\zeta)\phi_i(\rho) = \left(\frac{1}{\widetilde{\nu}_k}\right)\left[\frac{\phi_{k-1}(\rho)\phi_k(\zeta) - \phi_{k-1}(\zeta)\phi_k(\rho)}{\zeta - \rho}\right]. \tag{36}$$

If $\rho \to \zeta$ in (36) then a complex version of the second Christoffel–Darboux formula results:

$$\sum_{i=0}^{k-1}\big[\phi_i(\zeta)\big]^2 = \left(\frac{1}{\widetilde{\nu}_k}\right)\big[-\phi_k(\zeta)\phi'_{k-1}(\zeta) + \phi_{k-1}(\zeta)\phi'_k(\zeta)\big]. \tag{37}$$

Let $\zeta = \zeta_j$ in (37), where ζ_j is a root of $\phi_k(\zeta)$. Then

$$\sum_{i=0}^{k-1}\big[\phi_i(\zeta_j)\big]^2 = \left(\frac{1}{\widetilde{\nu}_k}\right)\phi_{k-1}(\zeta_j)\phi_k{}'(\zeta_j). \tag{38}$$

It follows that if, in (36), $\rho = \zeta_j$, then

$$\sum_{i=0}^{k-1}\phi_i(\zeta)\phi_i(\zeta_j) = \left(\frac{1}{\widetilde{\nu}_k}\right)\frac{\phi_k(\zeta)\phi_{k-1}(\zeta_j)}{\zeta - \zeta_j}. \tag{39}$$

We integrate both sides of (39) over arc γ with respect to the weight function $w(\zeta)$. The right side can be written as

$$\left(\frac{1}{\widetilde{\nu}_k}\right)\phi_{k-1}(\zeta_j)\int_\gamma \frac{\phi_k(\zeta)}{\zeta - \zeta_j}w(\zeta)|\mathrm{d}\zeta|,$$

and the left as

$$\int_\gamma \sum_{i=0}^{k-1}\phi_i(\zeta)\phi_i(\zeta_j)w(\zeta)|\mathrm{d}\zeta| = \sum_{i=0}^{k-1}\int_\gamma \phi_i(\zeta)\phi_i(\zeta_j)w(\zeta)|\mathrm{d}\zeta|.$$

But $\phi_i(\zeta_i)$ is a constant for each i and the polynomials ϕ_i, $i = 1, \ldots, k-1$, are formally orthogonal to constants. In addition, since ϕ_0 is normalized,

$$\int_\gamma \phi_0(\zeta)\phi_0(\zeta_j)w(\zeta)|\mathrm{d}\zeta| = 1.$$

Thus, the left side reduces to the constant 1, and

$$1 = \left(\frac{1}{\widetilde{\nu}_k}\right)\phi_{k-1}(\zeta_j)\int_\gamma \frac{\phi_k(\zeta)}{\zeta - \zeta_j}w(\zeta)|\mathrm{d}\zeta|.$$

Isolating the integral, we get

$$\frac{\widetilde{\nu}_k}{\phi_{k-1}(\zeta_j)} = \int_\gamma \frac{\phi_k(\zeta)}{\zeta - \zeta_j}w(\zeta)|\mathrm{d}\zeta|.$$

Equating the left side of this equation with the left side of (29) above gives

$$\omega_j \phi_k{}'(\zeta_j) = \frac{\widetilde{\nu}_k}{\phi_{k-1}(\zeta_j)}.$$

Solving for ω_j yields $\omega_j = \widetilde{\nu}_k/(\phi_{k-1}(\zeta_j)\phi_k{}'(\zeta_j))$, which is the reciprocal of the right side of (38). It follows that

$$1 = \omega_j \sum_{i=0}^{k-1}\left[\phi_i(\zeta_j)\right]^2 = \omega_j \left\|\Phi(\zeta_j)\right\|^2 \tag{40}$$

since the sum in the middle expression is merely the square of the norm of the eigenvector $\Phi(\zeta_j)$ of S_k (see (32)).

Let $\mathbf{q}_j$ denote the normalized eigenvector:

$$\mathbf{q}_j := \frac{\Phi(\zeta_j)}{\|\Phi(\zeta_j)\|}.$$

Denote the ith component of $\mathbf{q}_j$ by $q_{i,j}$. Using (40), we obtain

$$\omega_j = \frac{q_{1,j}^2}{\phi_0^2(\zeta_j)}. \tag{41}$$

This provides a method to obtain the weights from the eigenvalues and eigenvectors of the Jacobi matrix in (30).

An algorithm to obtain quadrature formulas results.

Algorithm 3. Gaussian quadrature.
Purpose: To compute nodes and weights for Gaussian quadrature in the complex plane.
Input: The number of terms, k, in the Gaussian quadrature formula. If the integral is continuous, an arc γ must be provided as part of the integral. If the integral is discrete, an arc γ is implicit; evaluation of an integral is equivalent to an inner product of two vectors.

1. Generate a set of polynomials formally orthogonalalong the given arc γ. This requires evaluating integrals. (Below in section 7 we do this with a composite midpoint approximation to the integral using a large number of fixed, determined [non-Gaussian] nodes.)
2. Derive the Jacobi matrix S_k (cf. (30)) from the three-term recurrence relation. The eigenvalues of this matrix are the nodes of the quadrature rule.
3. Generate the weights ω_i from the eigenvectors of S_k and formula (41).

We conclude by mentioning that Gauss–Radau and Gauss–Lobatto may also be developed in the same way by extending the standard theory. Further details are in section 11.

7. Experiments

Our experiments are based on an implementation of algorithm 3. We approximate the normalized integral

$$\frac{1}{L}\int_{\gamma} f(z)|\mathrm{d}z|, \tag{42}$$

where L is the length of the arc γ. Thus $(1/L)\int_{\gamma} f(z)|\mathrm{d}z| = 1$ when $f(z) = 1$.

Three node points and weights were computed for several integrals along either a straight line or a piecewise straight line. To compute the inner products needed in algorithm 3, we evaluated integrals with a composite midpoint rule rather than analytically, with a large number of panels, so as to assure full precision. We used 2^{17} subdivisions of the arc and double precision arithmetic. We compared the Gaussian values with analytic values of the integral in example 1 and with values obtained from the composite midpoint rule in example 2.

Example 1. For the straight line between the origin and 2i (along the imaginary axis), algorithm 3 gave the following nodes and weights:

Nodes	Weights
1.77457667i	0.277777777
i	0.444444444
0.225403331i	0.277777777

The values for the Gaussian quadrature rule matched those numerically computed for the integrals along the arc for test integrals of 1, ζ, and ζ^2 to at least 11 significant digits. The analytic values of the integral were 1, i, and $-4/3$, respectively. Note that in this case Gaussian quadrature in the complex plane reduces to classical quadrature since the line is a rotation and translation of an interval on the real axis.

Example 2. For the dogleg arc going from the origin to i and then along a 45° line in the first quadrant to $1 + 2\mathrm{i}$, algorithm 3 gave the following nodes and weights:

Nodes	Weights
$0.104489430 + 0.230172537\mathrm{i}$	$0.233255045 - 0.106638568\mathrm{i}$
$0.345707195 + 1.01610238\mathrm{i}$	$0.510352297 + 4.65830837 \times 10^{-2}\mathrm{i}$
$0.865188790 + 1.78088764\mathrm{i}$	$0.256392657 + 6.00554844 \times 10^{-2}\mathrm{i}$

The values for the Gaussian quadrature rule matched the composite midpoint rule values of the test integrals of 1, ζ, and ζ^2 to at least 15 significant digits. The results from the composite midpoint rule were 1, $0.2928932 + 1.085786\mathrm{i}$ and $-1.309644 + 0.9763107\mathrm{i}$, respectively.

8. Biconjugate gradient algorithm

Standard references for BCG are [3, p. 494; 36, p. 47; 44, p. 212]. We state here the usual form of the BCG algorithm, known as the *Omin* form, under the assumption that the sequences exist, i.e., no division by zero occurs. Then we derive a three-term recursion for the residual vectors. We also derive a recursion for the residual vectors. This recursion generates the Jacobi matrix, from which we obtain complex Gaussian quadrature.

8.1. Omin algorithm for BCG

This form of the BCG algorithm is patterned after the most familiar form of the CG algorithm.

This algorithm generates sequences of *biorthogonal vectors*, $\widetilde{r}^{\,(i)}$ and $r^{\,(i)}$, that is, vectors such that $\langle \widetilde{r}^{\,(i)}, r^{(j)} \rangle \neq 0$ when $i = j$. (For further discussion on biorthogonality, see [9].)

Algorithm 4. The Biconjugate Gradient Method (BCG).
Purpose: To solve $Ax = b$ through generating sequences of biorthogonal vectors.
Assumption: No division by zero occurs.
Initialization: Choose $x^{(0)}$. Set

$$r^{(0)} := b - Ax^{(0)}.$$

Choose $\widetilde{r}^{\,(0)}$ such that $\langle \widetilde{r}^{\,(0)}, r^{(0)} \rangle \neq 0$. Set

$$d^{(0)} := r^{(0)} \tag{43}$$

$$\widetilde{d}^{(0)} := \widetilde{r}^{\,(0)} \tag{44}$$

Do $k = 0, 1, \ldots,$ until convergence

$$\alpha_k := \frac{\langle \widetilde{r}^{\,(k)}, r^{(k)} \rangle}{\langle \widetilde{d}^{(k)}, Ad^{(k)} \rangle}$$

$$x^{(k+1)} := x^{(k)} + \alpha_k d^{(k)}$$

$$r^{(k+1)} := r^{(k)} - \alpha_k A d^{(k)} \tag{45}$$

$$\widetilde{r}^{(k+1)} := \widetilde{r}^{(k)} - \overline{\alpha}_k A^* \widetilde{d}^{(k)} \tag{46}$$

$$\beta_k := \frac{\langle \widetilde{r}^{(k+1)}, r^{(k+1)} \rangle}{\langle \widetilde{r}^{(k)}, r^{(k)} \rangle}$$

$$d^{(k+1)} := r^{(k+1)} + \beta_k d^{(k)} \tag{47}$$

$$\widetilde{d}^{(k+1)} := \widetilde{r}^{(k+1)} + \overline{\beta}_k \widetilde{d}^{(k)} \tag{48}$$

End Do.

We will call both $d^{(k)}$ and $\widetilde{d}^{(k)}$ direction vectors.

We note without proof that $r^{(k)} = b - Ax^{(k)}$. Also, for j not necessarily equal to k, $\widetilde{r}^{(j)}$ is a residual vector even though in the case of $\widetilde{r}^{(j)}$ there is no explicitly given system $A^*\widetilde{x} = \widetilde{b}$ to be solved. There do exist, however, $\widetilde{b}$ and $\widetilde{x}^{(0)}$ for which $\widetilde{r}^{(0)} = \widetilde{b} - A^*\widetilde{x}^{(0)}$. If the algorithm also computed $\widetilde{x}^{(k+1)} := \widetilde{x}^{(k)} + \alpha_k \widetilde{d}^{(k)}$, then, as it may be shown, $\widetilde{x}^{(0)}, \ldots$, would be a sequence of approximations to the solution of $A^*\widetilde{x} = \widetilde{b}$ for which $\widetilde{r}^{(k)} = \widetilde{b} - A^*\widetilde{x}^{(k)}$. This will justify references to $\widetilde{r}^{(k)}$ below as a residual vector.

8.2. Recursion for the residual vectors

We derive a recursion for the normalized residual vectors. This will yield a recursion for a set of normalized formally orthogonal polynomials from which a symmetric Jacobi matrix results.

To obtain a three-term recursion, we first have to rewrite the BCG algorithm. We shall follow the derivation in [2p. 1551], with the necessary changes.

Set $r^{(0)} = d^{(0)} = b$ and $\widetilde{r}^{(0)} = \widetilde{d}^{(0)} = \overline{c}$ in (43), (44) of the BCG algorithm. Define $\alpha_{-1} = 1$, $\beta_{-1} = 0$.

Since $\alpha_k d^{(k)} = x^{(k+1)} - x^{(k)}$, it follows from (47) that

$$x^{(k+)} = \alpha_k r^{(k)} + \left(1 + \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}}\right) x^{(k)} - \frac{\alpha_k \beta_{k-1}}{\alpha_{k-1}} x^{(k-1)}, \quad 0 \leqslant k,$$

which is of the form

$$x^{(k+1)} = \rho_k \left(x^{(k)} + \mu_k r^{(k)}\right) + (1 - \rho_k) x^{(k-1)}, \quad 0 \leqslant k, \tag{49}$$

where $\mu_0 = \alpha_0$, $\rho_0 = 1$.

Note that

$$x = \rho_k x + (1 - \rho_k) x. \tag{50}$$

Multiply (49), (50) by A and subtract to obtain

$$r^{(k+1)} = \rho_k \left(r^{(k)} - \mu_k A r^{(k)}\right) + (1 - \rho_k) r^{(k-1)} \tag{51}$$

(cf. [2, equation (4.4e), p. 1551]).

Rearranging terms, we have

$$Ar^{(k)} = \frac{1-\rho_k}{\rho_k\mu_k} r^{(k-1)} + \frac{1}{\mu_k} r^{(k)} - \frac{1}{\rho_k\mu_k} r^{(k+1)}, \tag{52}$$

where

$$\mu_k = \frac{\alpha_k\alpha_{k-1}}{\alpha_{k-1} + \alpha_k\beta_{k-1}} \quad (\text{with } \mu_0 = \alpha_0)$$

and

$$\rho_k = 1 + \frac{\alpha_k\beta_{k-1}}{\alpha_k} \quad (\text{with } \rho_1 = 1).$$

8.3. Normalized residual vectors

Thus far we are working with residual vectors only, not with normalized polynomials. Ultimately we will reach the topic of normalized polynomials, in fact normalized *residual* polynomials, from which a symmetric Jacobi matrix results (see (8)). Symmetry is equivalent to normalizing the residual polynomials.

We proceed in a straightforward way as follows.

Let $\eta_k = \sqrt{\langle \widetilde{r}^{(k)}, r^{(k)} \rangle}$, and

$$\boldsymbol{\Psi}^{(k)} = \frac{r^{(k)}}{\eta_k}.$$

Then (52) is equivalent to

$$A\boldsymbol{\Psi}^{(k)} = \frac{1-\rho_k}{\rho_k\mu_k}\frac{\eta_{k-1}}{\eta_k}\boldsymbol{\Psi}^{(k-1)} + \frac{1}{\mu_k}\boldsymbol{\Psi}^{(k)} - \frac{1}{\rho_k\mu_k}\frac{\eta_{k+1}}{\eta_k}\boldsymbol{\Psi}^{(k+1)}. \tag{53}$$

9. BCG polynomials and complex Gaussian quadrature

In our application to sums from BCG such as the scattering amplitude, we make use of certain polynomials that BCG generates, which yield Gaussian quadrature formulas. In this section we describe these polynomials, called *BCG polynomials*, show that they are formally orthogonal and link them to complex Gaussian quadrature. BCG polynomials are well-known [32], see also [17]. Our approach simplifies the biorthogonal polynomials derived in [46].

9.1. BCG residual vectors and polynomials

Vectors $d^{(k)}$ and $r^{(k)}$ belong to the Krylov subspace

$$V_k := \operatorname{span}\{r^{(0)}, \ldots, A^{k-1}r^{(0)}\}$$

whereas $\widetilde{d}^{(k)}$ and $\widetilde{r}^{(k)}$ belong to the Krylov subspace

$$\widetilde{V}_k := \operatorname{span}\{\widetilde{r}^{(0)}, \ldots, (A^*)^{k-1}\widetilde{r}^{(0)}\}.$$

Assume V_k and $\widetilde{V}_k$ are of dimension k. Vectors in these subspaces correspond to polynomials R_k, D_k, $\widetilde{R}_k$ and $\widetilde{D}_k$, which are such that $r^{(k)} := R_k(A)r^{(0)}$, $d^{(k)} := D_k(A)r^{(0)}$, $\widetilde{r}^{(k)} := \widetilde{R}_k(A^*)\widetilde{r}^{(0)}$, $\widetilde{d}^{(k)} := \widetilde{D}_k(A^*)\widetilde{r}^{(0)}$.

Below we shall use $\overline{\psi}$ to mean that polynomial obtained from polynomial ψ by conjugating the coefficients of ψ, i.e., $\overline{\psi}(\lambda) := \overline{\psi(\overline{\lambda})}$.

Theorem 3. $\overline{\widetilde{R}}_k(\zeta) = R_k(\zeta)$.

Proof. The proof is an easy induction based on the fact that the coefficients in either (45), (46) or (47), (48) are conjugates of one another. □

9.2. Formal orthogonality of BCG polynomials

Next we turn to the relation between BCG residual vectors and BCG residual polynomials and from this point on ignore D_k.

Assume that the eigenvectors of A are complete. Let

$$r^{(0)} := \sum_{i=1}^{N} \theta_i \lambda_i g_i \tag{54}$$

and

$$\widetilde{r}^{(0)} := \sum_{i=1}^{N} \widetilde{\theta}_i \overline{\lambda}_i \widetilde{g}_i, \tag{55}$$

where λ_i, g_i and $\overline{\lambda}_i$, $\widetilde{g}_i$ are eigenvalues and eigenvectors of A and A^*, respectively (cf. section 3.3). Recall that $x^{(0)} = 0$ from which it follows that

$$b = r^{(0)} \quad \text{and} \quad \bar{c} = \widetilde{r}^{(0)}.$$

(The vector $\bar{c}$ corresponds to the vector $\widetilde{b}$ in algorithm 4.) Moreover,

$$c^{\mathrm{T}} b = \left\langle \widetilde{r}^{(0)}, r^{(0)} \right\rangle. \tag{56}$$

We have

$$r^{(k)} = \sum_{i=1}^{N} \theta_i \lambda_i R_k(\lambda_i) g_i$$

and

$$\widetilde{r}^{(k)} = \sum_{i=1}^{N} \widetilde{\theta}_i \overline{\lambda}_i \overline{R_k(\lambda_i)} \widetilde{g}_i.$$

These are biorthogonal vectors, $\langle r^{(j)}, \widetilde{r}^{(k)} \rangle = 0$ if $j \neq k$. We observe without proof that the eigenvectors of A and A^* are also biorthogonal: $\langle g_j, \widetilde{g}_k \rangle = 0$ if $j \neq k$. Therefore,

$$\left\langle r^{(j)}, \widetilde{r}^{(k)} \right\rangle = \sum_{i=1}^{N} \theta_i \overline{\overline{\theta}}_i \lambda_i^2 R_j(\lambda_i) \overline{R_k\left(\overline{\lambda}_i\right)} = 0 \quad \text{for } j \neq k. \tag{57}$$

We now establish formal orthogonality between polynomials R_j and R_k, meaning that in place of an inner product there is a nonpositive bilinear form. Equation (57) expresses this formal orthogonality if there is a weight function that makes it possible to write the BCG inner products as integrals of polynomials. To find the appropriate weight function, start with (54), (55)

$$\left\langle r^{(0)}, \widetilde{r}^{(0)} \right\rangle = \sum_{i=1}^{N} \theta_i \overline{\overline{\theta}}_i \lambda_i^2. \tag{58}$$

The weight function therefore is

$$w(\lambda) = \begin{cases} \theta_i \overline{\overline{\theta}}_i \lambda_i^2 & \text{for } \lambda = \lambda_i,\ i = 1, \ldots, N, \\ 0 & \text{for } \lambda \neq \lambda_i,\ i = 1, \ldots, N, \end{cases} \tag{59}$$

since

$$0 = \left\langle r^{(j)}, \widetilde{r}^{(k)} \right\rangle = \int_{\gamma} R_j(\zeta) \overline{\overline{R}_k\left(\overline{\zeta}\right)} w(\zeta) |\mathrm{d}\zeta| = \int_{\gamma} R_j(\zeta) R_k(\zeta) w(\zeta) |\mathrm{d}\zeta|. \tag{60}$$

For polynomials π and ψ, let

$$(\pi, \psi)_w := \pi(\zeta) \psi(\zeta) w(\zeta) |\mathrm{d}\zeta|. \tag{61}$$

We now have a symmetric bilinear form, i.e., $(\pi, \psi)_w = (\psi, \pi)_w$. Note that symmetry still does not mean that $(\pi, \psi)_w$ is an inner product, however, since $(\psi, \psi)_w = 0$ is possible for nonzero ψ. Thus, a symmetric bilinear form allows for, but does not guarantee, a sequence of *formally orthogonal* polynomials [8], which is a sequence of polynomials $\phi_0, \ldots,$ such that $(\phi_j, \phi_k)_w = 0$, $j \neq k$.

From (60), it follows that

$$(R_j, R_k)_w = 0, \quad j \neq k. \tag{62}$$

9.3. Jacobi matrix

Given a set of formally orthogonal polynomials, there is a corresponding set of complex Gaussian quadrature formulas, derived from section 6. If the Omin form of BCG is used, however, direct evaluation of the formulas for the symmetric tridiagonal matrix S_k in (33) are not convenient. Instead we derive S_k from the parameters of the Omin form of BCG by using the Omin parameters to express the coefficients of the three-term recursion for the normalized residual vectors, (53). Let ϕ_i be the polynomial such that $r^{(i)} = \eta_i \phi_i(A) r^{(0)}$. Recursion (53) yields

$$\zeta\phi_i(\zeta) = \frac{1-\rho_i}{\rho_i\mu_i}\frac{\eta_{i-1}}{\eta_i}\phi_{i-1}(\zeta) + \frac{1}{\mu_i}\phi_i(\zeta) - \frac{1}{\rho_i\mu_i}\frac{\eta_{i+1}}{\eta_i}\phi_{i+1}(\zeta), \quad 0 \leqslant i \leqslant k-1 \leqslant N. \tag{63}$$

It follows that (cf. (30))

$$\begin{aligned}
\zeta &\begin{pmatrix} \phi_0(\zeta) \\ \phi_1(\zeta) \\ \vdots \\ \phi_{k-1}(\zeta) \end{pmatrix} \\
&= \begin{pmatrix}
\frac{1}{\mu_0} & \frac{1}{-\rho_0\mu_0}\frac{\eta_1}{\eta_0} & 0 & \dots & 0 \\
\frac{1-\rho_1}{\rho_1\mu_1}\frac{\eta_0}{\eta_1} & \frac{1}{\mu_1} & \frac{1}{-\rho_1\mu_1}\frac{\eta_2}{\eta_1} & \dots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \frac{1}{-\rho_{k-2}\mu_{k-2}}\frac{\eta_{k-1}}{\eta_{k-2}} \\
0 & \dots & 0 & \frac{1-\rho_{k-1}}{\rho_{k-1}\mu_{k-1}}\frac{\eta_{k-2}}{\eta_{k-1}} & \frac{1}{\mu_{k-1}}
\end{pmatrix} \\
&\quad \times \begin{pmatrix} \phi_0(\zeta) \\ \phi_1(\zeta) \\ \vdots \\ \phi_{k-1}(\zeta) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \frac{1}{-\rho_{k-1}\mu_{k-1}}\frac{\eta_k}{\eta_{k-1}}\phi_k(\zeta) \end{pmatrix}.
\end{aligned} \tag{64}$$

10. Relating BCG to Gaussian quadrature

The idea of approximating the scattering amplitude by Gaussian quadrature is due to Golub [40] and, independently, Warnick [50]. The details are as follows.

At any step of the BCG iteration, the eigenvalues and eigenvectors of S_k give the node points of the quadrature rule and the weights, ω_i, respectively. Recall (54), (55) from section 9.1: If $x^{(0)} = \widetilde{x}^{(0)} = 0$, then

$$(b =)\, r^{(0)} = \sum_{i=1}^{N} \theta_i \lambda_i g_i \quad \text{and} \quad (\overline{c} =)\, \widetilde{r}^{(0)} = \sum_{i=1}^{N} \widetilde{\theta}_i \overline{\lambda}_i \widetilde{g}_i,$$

where $g_1, \dots$ and $\widetilde{g}_1, \dots$ are eigenvectors of A and A^*, respectively.

If v and $\widetilde{v}$ are any two vectors,

$$v \in V_k := \operatorname{span}\{r^{(0)}, \ldots, A^{k-1}r^{(0)} = A^{k-1}b\}$$

and

$$\widetilde{v} \in \widetilde{V}_k := \operatorname{span}\{\widetilde{r}^{(0)}, \ldots, (A^*)^{k-1}\widetilde{r}^{(0)}\}$$

then there are polynomials P_k and $\widetilde{P}_k$ such that

$$v = P_k(A)r^{(0)} = P_k(A)\sum_{i=1}^{N}\theta_i\lambda_i g_i = \sum_{i=1}^{N}\theta_i\lambda_i P_k(\lambda_i)g_i$$

and

$$\widetilde{v} = \widetilde{P}_k(A^*)\widetilde{r}^{(0)} = \widetilde{P}_k(A^*)\sum_{i=1}^{N}\widetilde{\theta}_i\overline{\lambda}_i\widetilde{g}_i = \sum_{i=1}^{N}\widetilde{\theta}_i\overline{\lambda}_i\widetilde{P}_k(\overline{\lambda}_i)\widetilde{g}_i.$$

Thus,

$$\langle v, \widetilde{v}\rangle = \sum_{i=1}^{N}\theta_i\overline{\widetilde{\theta}}_i\lambda_i^2 P_k(\lambda_i)\overline{\widetilde{P}_k(\overline{\lambda}_i)}.$$

If $w(\lambda)$ is the discrete measure defined by (59) and if $\langle g_i, \widetilde{g}_i\rangle = 1$, then

$$\langle v, \widetilde{v}\rangle = \sum_{i=1}^{N}\theta_i\overline{\widetilde{\theta}}_i\lambda_i^2 P_k(\lambda_i)\overline{\widetilde{P}_k(\overline{\lambda}_i)} = \int_\gamma P_k(\lambda)\overline{\widetilde{P}_k(\overline{\lambda})}w(\lambda)|\mathrm{d}\lambda|,$$

where γ is an arc connecting the eigenvalues of A.

In particular, if P_0 and $\widetilde{P}_0$ are residual polynomials, i.e., $P_0(0) = \widetilde{P}_0(0) = 1$, we have

$$r^{(0)} = b, \qquad \widetilde{r}^{(0)} = \overline{c}$$

and (cf. (56))

$$c^{\mathrm{T}}b = \langle \overline{c}b\rangle = \int w(\lambda)|\mathrm{d}\lambda|.$$

In general, for any polynomial or nonpolynomial function f, Gaussian quadrature gives

$$\int_\gamma f(\lambda)w(\lambda)|\mathrm{d}\lambda| \approx \sum_{i=1}^{k}\omega_i f(\zeta_i).$$

But

$$\int_\gamma f(\lambda)\cdot 1\cdot w(\lambda)|\mathrm{d}\lambda| = \sum_{i=1}^{N}\theta_i\overline{\widetilde{\theta}}_i\lambda_i^2 f(\lambda_i)\cdot 1 = \langle f(A)b, \overline{c}\rangle.$$

For $f(A) = A^{-1}$, we have

$$\int_\gamma f(\lambda) \cdot w(\lambda)|\mathrm{d}\lambda| = \int_\gamma \frac{1}{\lambda} w(\lambda)|\mathrm{d}\lambda| = \left\langle A^{-1}b, \overline{c} \right\rangle = c^{\mathrm{T}} A^{-1} b,$$

and

$$c^{\mathrm{T}} A^{-1} b \approx \sum_{i=1}^{k} \frac{\omega_i}{\zeta_i}.$$

11. Further comments

We briefly mention two important variants of Gaussian quadrature. For details we refer the reader to the work of Gander and Hřebíček in [20; pp. 254–60] and of Golub et al. in [23; pp. 332–335; 25, pp. 108–109].

The Gauss–Radau quadrature rule is a variation of the general Gaussian quadrature rule for which one node point is predetermined:

$$\int_\gamma f(\zeta) w(\zeta)|\mathrm{d}\zeta| \approx \sum_{i=1}^{k} \omega_i f(\zeta_i) + \omega_{k+1} f(a),$$

where a is a predetermined (end-)point on the arc γ. In this form, there is one more weight to determine, namely ω_{k+1}.

The Gauss–Lobatto quadrature rule is a variation of the general Gaussian quadrature rule for which two points in the formula are predetermined:

$$\int_\gamma f(\zeta) w(\zeta)|\mathrm{d}\zeta| \approx \sum_{i=1}^{k} \omega_i f(\zeta_i) + \omega_{k+1} f(a) + \omega_{k+2} f(b).$$

Two additional weights have to be determined.

In both cases, the extra weights can be computed from the eigensystem of a tridiagonal matrix just as in the real case, with some necessary changes.

Acknowledgement

We are indebted to Karl Warnick for pointing out to us the equivalence of the Gaussian quadrature estimate to $c^{\mathrm{T}} x^{(k)}$. We are also indebted to Karl for many references.

Appendix: Warnick's theorem

For completeness, we include in this appendix a statement and proof of Warnick's theorem on the equivalence of $x^{(k)}$ and the Gaussian quadrature estimate to the scattering amplitude.

We continue to assume $\widetilde{x}^{(0)} = x^{(0)} = 0$. Then $\widetilde{r}^{(0)} = \widetilde{b}$ and $r^{(0)} = b$. We follow convention and use $\bar{c}$ as the right side of the adjoint system $A^*\widetilde{x} = \widetilde{b}$, rather than $\widetilde{b}$. Thus, $A^*\widetilde{x} = \bar{c}$.

We shall make use of the following decomposition for residual polynomials. Since $R_k(0) = 1$, we have

$$R_k(\lambda) = \lambda\left[\frac{1}{\lambda} - C_{k-1}(\lambda)\right], \tag{A.1}$$

where $C_{k-1}(\lambda)$ is a polynomial of degree $k-1$. Observe that $C_{k-1}(\lambda)$ is such that $x^{(k)} = C_{k-1}(A)r^{(0)}$.

We shall denote the Gaussian quadrature formula in (25) by

$$G(f) := \sum_{i=1}^{k} \omega_i f(\zeta_i).$$

The family of formally orthogonal polynomials is the family of residual polynomials, for which the weight function is w defined in (59).

Theorem 4. (Karl Warnick). If $x^{(k)}$ is the BCG iterate at step k,

$$G\left(\frac{1}{\lambda}\right) = c^{\mathrm{T}}x^{(k)}.$$

Proof. Gaussian quadrature with weight function w applied to (A.1) yields

$$G\left(\frac{R_k(\lambda)}{\lambda}\right) = G\left(\frac{1}{\lambda}\right) - G\big(C_{k-1}(\lambda)\big).$$

Since Gaussian quadrature is exact for the polynomial,

$$\begin{aligned} G\big(C_{k-1}(\lambda)\big) &= \int_{\gamma} C_{k-1}(\lambda)\cdot 1\cdot w(\lambda)|\mathrm{d}\lambda| \\ &= \big\langle \widetilde{R}_0\big(A^*\big)\widetilde{r}^{(0)}C_{k-1}(A)r^{(0)}\big\rangle = \big\langle \overline{c}x^{(k)}\big\rangle = c^{\mathrm{T}}x^{(k)}. \end{aligned}$$

Therefore,

$$G\left(\frac{R_k(\lambda)}{\lambda}\right) = G\left(\frac{1}{\lambda}\right) - \big\langle \overline{c}, x^{(k)}\big\rangle.$$

The Gaussian quadrature node points are the roots of R_k. It follows that

$$G\left(\frac{R_k(\lambda)}{\lambda}\right) = 0.$$

Therefore

$$0 = G\left(\frac{1}{\lambda}\right) - c^{\mathrm{T}}x^{(k)}. \qquad \square$$

References

[1] D. Arnett, *Supernovae and Nucleosynthesis* (Princeton Univ. Press, Princeton, 1996).

[2] S.F. Ashby, T.A. Manteuffel and P.E. Saylor, A taxonomy for conjugate gradient methods, SIAM J. Numer. Anal. 27(6) (1990) 1542–1568.

[3] O. Axelsson, *Iterative Solution Methods* (Cambridge Univ. Press, Cambridge, 1996).

[4] T.L. Barth and T.A. Manteuffel, Conjugate gradient algorithms using multiple recursions, in: *Linear and Nonlinear Conjugate Gradient Related Methods*, eds. L. Adams and J.L. Nazareth (SIAM, Philadelphia, PA 1996).

[5] W.E. Boyse, D.R. Lynch, K.D. Paulsen and G.N. Minerbo, Nodal based finite element modeling of Maxwell's equations in three dimensions, IEEE Trans. Antennas and Propagation 40(6) (1992) 642–651.

[6] W.E. Boyse, D.R. Lynch, K.D. Paulsen and G.N. Minerbo, Scalar and vector potential formulation for finite element solutions to Maxwell's equations, in: *Proc. IEEE AP–S International Symposium*, Vol. 1 (Springer, Berlin, 1992) pp. 516–519.

[7] W.E. Boyse, G.N. Minerbo, K.D. Paulsen and D.R. Lynch, Applications of potentials to finite element modeling of Maxwell's equations, IEEE Trans. Magn. 29 (1993) 1333–1336.

[8] C. Brezinski, *Padé-Type Approximation and General Orthogonal Polynomials* (Birkhäuser, Basel, 1980).

[9] C. Brezinski, *Biorthogonality and Its Applications to Numerical Analysis* (Marcel Dekker, New York, 1992).

[10] D. Calvetti, G.H. Golub and L. Reichel, An adaptive Chebyshev iterative method for nonsymmetric linear systems based on modified moments, Numer. Math. 67 (1994) 21–40.

[11] W.C. Chew, *Waves and Fields in Inhomogeneous Media* (Oxford, New York, 1996).

[12] G. Dahlquist and Å. Björck, *Numerical Methods* (Prentice-Hall, Englewood Cliffs, NJ, 1974).

[13] G. Dahlquist, S.C. Eisenstat and G.H. Golub, Bounds for the error of linear systems of equations using the theory of moments, J. Math. Anal. Appl. 37 (1972) 151–166.

[14] G. Dahlquist, G.H. Golub and S. Nash, Bounds for the error in linear systems, in: *Proc. of the Workshop on Semi-Infinite Programming*, ed. R. Hettich (Springer, New York) pp. 154–172.

[15] T.A. Driscoll, K.-C. Toh and L.N. Trefethen, From potential theory to matrix iterations in six steps, SIAM Rev. 40 (1998) 547–581.

[16] B. Fischer and G.H. Golub, On the error computation for polynomial based iteration methods, in: *Recent Advances in Iterative Methods* (Springer, New York, 1993) pp. 59–69.

[17] R. Freund, M. Gutknecht and N.M. Nachtigal, An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices, SIAM J. Sci. Statist. Comput. 14 (1993) 137–158.

[18] R.W. Freund, The look-ahead lanczos process for large nonsymmetric matrices and related algorithms, in: *Linear Algebra for Large Scale and Real Time Applications* (Kluwer Academic, Dordrecht, 1993) pp. 137–163.

[19] R.W. Freund and M. Hochbruck, Gauss quadrature associated with the Arnoldi process and the Lanczos algorithm, in: *Linear Algebra for Large Scale and Real Time Applications* (Kluwer Academic, Dordrecht, 1993) pp. 377–380.

[20] W. Gander and J. Hřebíček, *Solving Problems in Scientific Computing using Maple and Matlab* (Springer, Berlin, 1995).

[21] W. Gautschi, *Orthogonal Polynomials: Applications and Computation*, Vol. 5 (Cambridge Univ. Press, Cambridge, 1996) pp. 45–119.

[22] W. Gautschi, *Numerical Analysis: An Introduction* (Birkhäuser, Boston, 1997).

[23] G.H. Golub, Some modified matrix eigenvalue problems, SIAM Rev. 15(2) (1973) 318–334.

[24] G.H. Golub, Private communication, 1996.

[25] G.H. Golub and G. Meurant, Matrices, moments and quadrature, in: *Numerical Analysis 1993: Proc. of the 15th Dundee Conference, June–July 1993*, eds. D.F. Griffiths and G.A. Watson, Essex, England, Pitman Research Notes in Mathematics Series, Vol. 303 (Longman, Harlow, 1994) pp. 105–156.

[26] G.H. Golub and G. Meurant, Matrices, moments and quadrature II or how to compute the norm of the error in iterative methods, BIT 37(3) (1997) 687–705.
[27] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 3rd edn. (Johns Hopkins Univ. Press, Baltimore, MD, 1996).
[28] G.H. Golub and J.H. Welsch, Calculation of Gauss quadrature rules, Math. Comp. 23 (1969) 221–230.
[29] R.G. Gordon, Error bounds in equilibrium statistical mechanics, J. Math. Phys. 9(5) (1967) 655–663.
[30] R.G. Gordon, Error bounds in spectroscopy and nonequilibrium statistical mechanics, J. Math. Phys. 9(7) (1967) 1087–1092.
[31] W.B. Gragg, Positive definite Toeplitz matrices, the Arnoldi process for isometric operators, and gaussian quadrature on the unit circle, J. Comput. Appl. Math. 46 (1993) 183–198; English transl. of 1982 Russian article.
[32] M. Gutknecht, *Lanczos Type Solvers for Nonsymmetric Linear Systems of Equations* (Cambridge Univ. Press, Cambridge, 1997) pp. 271–398.
[33] M.T. Heath, *Scientific Computing: An Introductory Survey* (McGraw-Hill, New York, 1997).
[34] P. Henrici, *Applied and Computational Complex Analysis*, Vol. 2 (Wiley, New York, 1977).
[35] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. of NBS 49 (1952) 409–435.
[36] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM Frontiers Series, Vol. 16 (SIAM, Philadelphia, PA, 1995).
[37] L.D. Landau and E.M. Lifshitz, *Quantum Mechanics* (*Non-relativistic Theory*), 3rd edn. (Pergamon, Oxford, 1977).
[38] L. Lempert, Recursion for orthogonal polynomials on complex domains, Colloq. Math. Soc. János Bolyai 19 (1976) 481–494.
[39] F.E. Low, *Classical Field Theory, Electromagnetism and Gravitation* (Wiley, New York, 1998).
[40] G.N. Minerbo, G.H. Golub and P.E. Saylor, Nine ways to solve a scattering problem I, Technical Report, Stanford University (2001), in preparation.
[41] R.G. Newton, *Scattering Theory of Waves and Particles*, 2nd edn. (Springer, New York, 1982).
[42] K.D. Paulsen, Finite element solution of Maxwell's equations with Helmholtz forms, Optical Society of America A: Special Issue on Scattering by Three-dimensional Objects 11 (1994) 1434–1444.
[43] N.A. Pierce and M.B. Giles, Adjoint recovery of superconvergent functionals from pde approximations, SIREV 42(2) (2000) 247–264.
[44] Y. Saad, *Iterative Methods for Sparse Linear Systems* (PWS, Boston, 1996).
[45] H.S. Shapiro, *The Schwarz Function and Its Generalization to Higher Dimensions* (Wiley, New York, 1992).
[46] D.C. Smolarski, Optimum semi-iterative methods of the solution of any linear algebraic system with a square matrix, Ph.D. thesis, Department of Computer Science, University of Illinois, Urbana (December 1981). Available as Technical Report 1077.
[47] E. Stiefel, Kernel polynomials in linear algebra and their numerical applications, Nat. Bureau Standards Math. Ser. 49 (1958) 1–22.
[48] G. Szegő, *Orthogonal Polynomials*, 4th edn. (Amer. Math. Soc., Providence, RI, 1975).
[49] K.F. Warnick, Private communication (1998).
[50] K.F. Warnick, Continued fraction error bound for conjugate gradient method, Technical Report ECEN Department Report No. TR-L100-98.3, Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602 (9 June 1997).
[51] K.F. Warnick, Gaussian quadrature and scattering amplitude (exact title tbd), Technical Report No. TBD, Department of Electrical and Computer Engineering, Center for Computational Electromagnetics, University of Illinois, Urbana–Champaign (2000).
[52] H.S. Wilf, *Mathematics for the Physical Sciences* (Wiley, New York, 1962).