

Article

Error Estimators for a Krylov Subspace Iterative Method for Solving Linear Systems of Equations with a Symmetric Indefinite Matrix

Mohammed Alibrahim ¹, Mohammad Taghi Darvishi ² , Lothar Reichel ^{1,*}  and Miodrag M. Spalević ³ 

¹ Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA; malibra2@kent.edu

² Department of Mathematics, Razi University, Kermanshah 67149, Iran; darvishi@razi.ac.ir

³ Department of Mathematics, Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade 35, Serbia; mspalevic@mas.bg.ac.rs

* Correspondence: reichel@math.kent.edu

Abstract: This paper describes a Krylov subspace iterative method designed for solving linear systems of equations with a large, symmetric, nonsingular, and indefinite matrix. This method is tailored to enable the evaluation of error estimates for the computed iterates. The availability of error estimates makes it possible to terminate the iterative process when the estimated error is smaller than a user-specified tolerance. The error estimates are calculated by leveraging the relationship between the iterates and Gauss-type quadrature rules. Computed examples illustrate the performance of the iterative method and the error estimates.

Keywords: iterative method; Lanczos process; error norm estimate; Gauss-type quadrature rules

MSC: 65D30; 65D32



Academic Editor: Wen Li

Received: 27 January 2025

Revised: 11 February 2025

Accepted: 16 February 2025

Published: 28 February 2025

Citation: Alibrahim, M.; Darvishi, M.T.; Reichel, L.; Spalević, M.M. Error Estimators for a Krylov Subspace Iterative Method for Solving Linear Systems of Equations with a Symmetric Indefinite Matrix. *Axioms* **2025**, *14*, 179. <https://doi.org/10.3390/axioms14030179>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Consider the linear system of equations

$$Ax = b, \quad (1)$$

where $A \in \mathbb{R}^{m \times m}$ is a large, symmetric, nonsingular, and indefinite matrix and x and b are real vectors. Such systems arise in various areas of applied mathematics and engineering. When A is too large to make its factorization feasible or attractive, an iterative solution method has to be employed. Among the most well-known iterative methods for solving linear systems of the kind (1) are MINRES or SYMMLQ by Paige and Saunders (see [1,2]). However, none of these methods allow for easy estimation of the error in the computed iterates. This can make it difficult to decide when to terminate the iterations.

For a symmetric, positive definite matrix A , the conjugate gradient method is typically used to solve (1). Various techniques are available to estimate the A -norm of the error in the iterates determined by the conjugate gradient method. These techniques leverage the relationship between the conjugate gradient method and Gauss-type quadrature rules applied to integrate the function $f(t) = t^{-1}$. The quadrature rules are determined with respect to an implicitly defined non-negative measure defined by the matrix A , the right-hand side b , and the initial iterate x_0 (see, for example, Almutairi et al. [3], Golub and Meurant [4,5], Meurant and Tichý [6] and references therein).

Error estimation of iterates in cases where the matrix A is nonsingular, symmetric, and indefinite has not received much attention in the literature. We observe that the application of A -norm estimates of the error in the iterates is meaningful when A is symmetric and positive definite [7]. However, this is not the case when A is symmetric and indefinite. In this paper, we estimate the Euclidean norm of the error for each iterate produced by an iterative method, which is described below.

In their work, Calvetti et al. [8] introduced a Krylov subspace method for the iterative solution of (1). They proposed estimating the Euclidean norm of the error in the iterates generated by their method using pairs of associated Gauss and anti-Gauss quadrature rules. However, the quality of the error norm estimates determined in this manner is mixed. Examples of computed results demonstrate that some error norm estimates are significantly exaggerated compared to the actual error norm in the iterates.

This paper presents novel methods for calculating the Euclidean norm of the error in the iterates computed using the iterative method described in Section 2 and [8]. In particular, the anti-Gauss rule used in [8] is replaced by other quadrature rules.

For notational simplicity, we start with the initial approximate solution $\mathbf{x}_0 = \mathbf{0}$. Consequently, the k th approximate solution $\mathbf{x}_k$ determined by the iterative method discussed in this paper lives in the Krylov subspace:

$$\mathcal{K}_k(A, \mathbf{b}) = \text{span}\{\mathbf{b}, A\mathbf{b}, \dots, A^{k-1}\mathbf{b}\}, \quad k = 1, 2, \dots, \quad (2)$$

i.e.,

$$\mathbf{x}_k = p_{k-1}(A)\mathbf{b}, \quad k = 1, 2, \dots, \quad (3)$$

for a suitable iteration polynomial p_{k-1} in Π_{k-1} , where Π_{k-1} is the set of all polynomials of degree less than or equal to $k - 1$. We require that the iteration polynomials satisfy

$$p_{k-1}(0) = 0, \quad k = 1, 2, \dots. \quad (4)$$

Then

$$p_{k-1}(t) = tq_{k-2}(t) \quad (5)$$

fulfills condition (4) for a polynomial $q_{k-2} \in \Pi_{k-2}$.

We introduce the residual error related to $\mathbf{x}_k$ as

$$\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k \quad (6)$$

and let $\mathbf{x}_*$ denote the solution of (1). Then the error $\tilde{\mathbf{e}}_k = \mathbf{x}_* - \mathbf{x}_k$ in $\mathbf{x}_k$ can be expressed as

$$\tilde{\mathbf{e}}_k = A^{-1}\mathbf{r}_k. \quad (7)$$

Equations (6) and (7) yield

$$\tilde{\mathbf{e}}_k^t \tilde{\mathbf{e}}_k = \mathbf{r}_k^t A^{-2} \mathbf{r}_k = (\mathbf{b} - A\mathbf{x}_k)^t A^{-2} (\mathbf{b} - A\mathbf{x}_k) = \mathbf{b}^t A^{-2} \mathbf{b} - 2\mathbf{b}^t A^{-1} \mathbf{x}_k + \mathbf{x}_k^t \mathbf{x}_k, \quad (8)$$

where the superscript t denotes transposition. We may calculate the Euclidean norm of the vector $\tilde{\mathbf{e}}_k$ by using the terms found on the right-hand side of Equation (8). It is straightforward to compute the term $\mathbf{x}_k^t \mathbf{x}_k$, and using (5), the expression $2\mathbf{b}^t A^{-1} \mathbf{x}_k$ can be calculated as

$$2\mathbf{b}^t A^{-1} \mathbf{x}_k = 2\mathbf{b}^t q_{k-2}(A)\mathbf{b}.$$

Hence, the expression $2\mathbf{b}^t A^{-1} \mathbf{x}_k$ can be evaluated without using A^{-1} . The iterative method has to be chosen so that recursion formulas for the polynomials q_{k-2} , $k = 2, 3, \dots$

easily can be computed. Finally, we have to estimate $\mathbf{b}^t A^{-2} \mathbf{b}$, which by setting $f(t) = 1/t^2$, can be written as the following matrix functional

$$\mathcal{F}(A) = \mathbf{b}^t f(A) \mathbf{b}. \quad (9)$$

We use Gauss-type quadrature rules determined by the recurrence coefficients of the iterative method to approximate (9).

The structure of this paper is as follows. Section 2 outlines the iterative Krylov subspace method employed for the solution of (1). This method was discussed in [8]. We review the method for the convenience of the reader. Our presentation differs from that in [8]. The iterative method is designed to facilitate the evaluation of the last two terms on the right-hand side of (8). Section 3 explores various Gauss-type quadrature rules that are employed to estimate the first term on the right-hand side of (8). The quadrature rules used in this study include three kinds of Gauss-type rules, namely averaged and optimally averaged rules by Laurie [9] and Spalević [10], respectively, as well as Gauss-Radau rules with a fixed quadrature node at the origin. Additionally, we describe how to update the quadrature rules cost-effectively as the number of iterations grows. This section improves on the quadrature rules considered in [8]. Section 4 describes the use of these quadrature rules to estimate the Euclidean norm of the errors $\tilde{\mathbf{e}}_k$, $k = 1, 2, \dots$. Section 5 provides three computed examples and Section 6 contains concluding remarks.

The iterates determined by an iterative method often converge significantly faster when a suitable preconditioner is applied (see, e.g., Saad [11] for discussions and examples of preconditioners). We assume that the system (1) is preconditioned when this is appropriate.

2. The Iterative Scheme

This section revisits the iterative method considered in [8] for solving linear systems of Equation (1) with a nonsingular, symmetric, and indefinite matrix A . We begin by discussing some fundamental properties of the method. Subsequently, we describe updating formulas for the approximate solutions $\mathbf{x}_k$. This method can be seen as a variation of the SYMMLQ scheme discussed in [1,12].

It is convenient to introduce the spectral factorization of the coefficient matrix (1),

$$A = U \Lambda U^t,$$

where $\Lambda \in \mathbb{R}^{m \times m}$ is a diagonal matrix with diagonal elements $\{\lambda_k\}_{k=1}^m$ and the matrix $U \in \mathbb{R}^{m \times m}$ is orthogonal. The spectral factorization is used for the description of the iterative method but does not have to be computed for the solution of (1). Defining $\mathbf{b}' = [b'_1, b'_2, \dots, b'_m]^t = U^t \mathbf{b}$, the functional (9) can be written as

$$\mathcal{F}(A) = \mathbf{b}'^t f(\Lambda) \mathbf{b}' = \sum_{k=1}^m f(\lambda_k) b_k'^2 = \int_{-\infty}^{\infty} f(x) dw(x), \quad (10)$$

where the measure dw has jump discontinuities at the eigenvalues λ_k of A .

Our iterative method is based on the Lanczos algorithm. Let $I_k \in \mathbb{R}^{k \times k}$ denote the identity matrix. By applying $1 \leq k \ll m$ steps of the Lanczos process to A with initial vector $\mathbf{b}$, the following Lanczos decomposition is obtained:

$$A \mathcal{V}_k = \mathcal{V}_k T_k + \mathbf{f}_k \mathbf{e}_k^t, \quad (11)$$

where $\mathbf{f}_k \in \mathbb{R}^m$ and $\mathcal{V}_k = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k] \in \mathbb{R}^{m \times k}$ are such that $\mathcal{V}_k^t \mathcal{V}_k = I_k$, $\mathcal{V}_k^t \mathbf{f}_k = \mathbf{0}$, and

$$\mathbf{v}_1 = \mathbf{b} / \|\mathbf{b}\|. \quad (12)$$

Additionally, T_k is a symmetric, tridiagonal $k \times k$ matrix, $\mathbf{e}_k$ denotes the k th column of the identity matrix, and $\|\cdot\|$ represents the Euclidean vector norm. The columns of the matrix $\mathcal{V}_k$ span the Krylov subspace (2), i.e.,

$$\text{range}(\mathcal{V}_k) = \mathcal{K}_k(A, \mathbf{b}). \quad (13)$$

It is assumed that all subdiagonal entries of T_k are nonvanishing; otherwise, the recursion formulas of the Lanczos process break down, and the solution of (1) can be formulated as a linear combination of the vectors $\mathbf{v}_j$ that are available at the time of breakdown. The recursion relation for the columns of $\mathcal{V}_k$ is established by Equation (11) and, in conjunction with (12), shows that

$$\mathbf{v}_j = S_{j-1}(A)\mathbf{b}, \quad j = 1, 2, \dots, k, \quad (14)$$

for certain polynomials S_j of degree j .

Theorem 1. The polynomials S_j determined by (14) are orthonormal with respect to the inner product

$$(g, h) = \mathcal{I}(gh) \quad (15)$$

induced by the operator $\mathcal{I}(f) = \int_{-\infty}^{\infty} f(x)dw(x)$.

Proof. We have

$$\begin{aligned} (S_{j-1}, S_{\ell-1}) &= \int_{-\infty}^{\infty} S_{j-1}(x)S_{\ell-1}(x)dw(x) \\ &= \mathbf{b}^t (US_{j-1}(\Lambda)S_{\ell-1}(\Lambda)U^t) \mathbf{b} \\ &= \mathbf{b}^t (US_{j-1}(\Lambda)U^t) (US_{\ell-1}(\Lambda)U^t) \mathbf{b} \\ &= \mathbf{b}^t S_{j-1}(A)S_{\ell-1}(A) \mathbf{b} \\ &= (S_{j-1}(A)\mathbf{b})^t (S_{\ell-1}(A)\mathbf{b}) \\ &= \mathbf{v}_j^t \mathbf{v}_\ell = \begin{cases} 0, & j \neq \ell, \\ 1, & j = \ell, \end{cases} \end{aligned}$$

because the columns $\mathbf{v}_j$, $1 \leq j \leq k$, of the matrix $\mathcal{V}_k$ are orthogonal and of unit norm (see (11)). $\square$

We also use the related decomposition to (11):

$$A\mathcal{V}_{k-1} = \mathcal{V}_k T_{k,k-1}, \quad (16)$$

where $T_{k,k-1}$ is the leading submatrix of T_k of order $k \times (k-1)$.

We use the QR factorization for T_k , that is,

$$T_k = Q_k R_k, \quad \text{for } Q_k, R_k \in \mathbb{R}^{k \times k}, \quad (17)$$

where $Q_k^t Q_k = I_k$ and the matrix $R_k = [r_{j\ell}^{(k)}]_{j,\ell=1}^k$ is upper triangular. Similarly, we introduce the factorization

$$T_{k,k-1} = Q_k \begin{bmatrix} \bar{R}_{k-1} \\ 0 \end{bmatrix} = Q_{k,k-1} \bar{R}_{k-1}, \quad (18)$$

where the $(k-1) \times (k-1)$ matrix $\tilde{R}_{k-1}$ is the leading submatrix of R_k and $Q_{k,k-1} \in \mathbb{R}^{k \times (k-1)}$ is the leading submatrix of Q_k .

Theorem 2. Combine the QR factorization (17) with the Lanczos decomposition (11). This defines a new iterative process with iteration polynomials that comply with (4).

Proof. Let $k > 1$. By applying the QR factorization (17) within the Lanczos decomposition (11), we obtain

$$A\mathcal{V}_k = \mathcal{V}_k Q_k R_k + \mathbf{f}_k \mathbf{e}_k^t. \quad (19)$$

Multiplying (19) by Q_k from the right-hand side, letting $\tilde{V}_k := \mathcal{V}_k Q_k$, and defining $\tilde{T}_k = R_k Q_k$, we obtain

$$A\tilde{V}_k = \tilde{V}_k \tilde{T}_k + \mathbf{f}_k \mathbf{e}_k^t Q_k. \quad (20)$$

The column vectors of the matrix expressed as $\tilde{V}_k = [\tilde{\mathbf{v}}_1^{(k)}, \tilde{\mathbf{v}}_2^{(k)}, \dots, \tilde{\mathbf{v}}_k^{(k)}]$ are orthonormal, and the matrix $\tilde{T}_k$ is symmetric and tridiagonal. To expose the relation between the first column $\mathbf{v}_1$ of $\mathcal{V}_k$ and the first column $\tilde{\mathbf{v}}_1^{(k)}$ of $\tilde{V}_k$, we multiply (19) by $\mathbf{e}_1$ from the right. This yields

$$A\mathcal{V}_k \mathbf{e}_1 = \tilde{V}_k R_k \mathbf{e}_1 + \mathbf{f}_k \mathbf{e}_k^t \mathbf{e}_1, \quad k > 1,$$

which simplifies to

$$A\mathbf{v}_1 = r_{11}^{(k)} \tilde{\mathbf{v}}_1^{(k)},$$

where $r_{11}^{(k)} \mathbf{e}_1 = R_k \mathbf{e}_1$. For a suitable choice of the sign of $r_{11}^{(k)}$, we have

$$\tilde{\mathbf{v}}_1^{(k)} = A\mathbf{b} / \|A\mathbf{b}\|. \quad (21)$$

Since T_k is tridiagonal, the orthogonal matrix Q_k in the QR factorization (17) has upper Hessenberg form. As a result, only the last two entries of the vector expressed as $\mathbf{e}_k^t Q_k$ are non-zero. Hence, the decomposition (20) differs from a Lanczos decomposition by potentially having non-zero entries in the last two columns of the matrix $\mathbf{f}_k \mathbf{e}_k^t Q_k$.

Suppose that the matrix $\tilde{V}_{k-1}$ consists of the first $k-1$ columns of $\tilde{V}_k$. Then,

$$\tilde{V}_{k-1} = \mathcal{V}_k Q_{k,k-1}, \quad (22)$$

where $Q_{k,k-1}$ is defined by Equation (18). Typically, $\tilde{V}_{k-1} \neq \tilde{V}_{k-1}$; additional details can be found in Section 4. When the last column is removed from each term in (20), the following decomposition results:

$$A\tilde{V}_{k-1} = \tilde{V}_{k-1} \tilde{T}_{k-1} + \tilde{\mathbf{f}}_{k-1} \mathbf{e}_{k-1}^t. \quad (23)$$

In (23), the matrix $\tilde{T}_{k-1}$ is the $(k-1) \times (k-1)$ leading submatrix of $\tilde{T}_{k-1}$. Furthermore, $\tilde{V}_{k-1}^t \tilde{\mathbf{f}}_{k-1} = \mathbf{0}$, and $\tilde{V}_{k-1}^t \tilde{V}_{k-1} = I_{k-1}$. As a result, according to (21), we have that (23) is a Lanczos decomposition with the starting vector $\tilde{\mathbf{v}}_1^{(k)}$ of $\tilde{V}_{k-1}$ proportional to the vector $A\mathbf{b}$. Similarly to (13), we have

$$\text{range}(\bar{V}_{k-1}) = \mathcal{K}_{k-1}(A, Ab). \quad (24)$$

To determine the iteration polynomials (3) and the corresponding approximate solutions $\mathbf{x}_k$ of (3), we impose the following requirement for certain vectors $\mathbf{w}_{k-1} \in \mathbb{R}^{k-1}$:

$$\mathbf{x}_k = p_{k-1}(A)\mathbf{b} = \bar{V}_{k-1}\mathbf{w}_{k-1}. \quad (25)$$

It follows from (24) that any polynomial p_{k-1} determined by (25) fulfills (4). This completes the proof. $\square$

Remark 1. We chose $\mathbf{w}_{k-1}$ in (25) and, thereby, p_{k-1} in Π_{k-1} in a manner that guarantees the residual error (6) for the approximate solution $\mathbf{x}_k$ of (1) satisfies the Petrov-Galerkin condition, i.e.,

$$\mathbf{0} = \mathcal{V}_{k-1}^t \mathbf{r}_k = \mathcal{V}_{k-1}^t \mathbf{b} - \mathcal{V}_{k-1}^t A \bar{V}_{k-1} \mathbf{w}_{k-1}, \quad (26)$$

which, according to (12) and the factorization (22), simplifies to

$$\|\mathbf{b}\| \mathbf{e}_1 = (A\mathcal{V}_{k-1})^t \mathcal{V}_k Q_{k,k-1} \mathbf{w}_{k-1}. \quad (27)$$

Remark 2. Replacing the matrix $\bar{V}_{k-1}$ in (26) with $\mathcal{V}_{k-1}$ recovers the SYMMLQ method [1]. However, the iteration polynomial p_{k-1} associated with the SYMMLQ method typically does not satisfy condition (4). Our method implements a QR factorization of matrix T_k , akin to the SYMMLQ method implementation by Fischer ([12], Section 6.5). In contrast, Paige and Saunders' [1] implementation of the SYMMLQ method relies on an LQ factorization of T_k .

Remark 3. Equation (26) shows that the iterative method is a Petrov-Galerkin method. In each step of the method, the dimension of the solution subspace (24) is increased by one, and the residual error is required to be orthogonal to the subspace $(\mathcal{K}_{k-1}(A, \mathbf{b}))$, cf. (13). This secures convergence of the iterates (25) to the solution $\mathbf{x}_*$ of (1) as k increases.

Using Theorems 1 and 2, along with Remark 1, we can simplify the right-hand side of (7). First, using (16) and (18), we obtain

$$(A\mathcal{V}_{k-1})^t \mathcal{V}_k Q_{k,k-1} = T_{k,k-1}^t Q_{k,k-1} = \bar{R}_{k-1}^t. \quad (28)$$

Subsequently, by substituting (28) into (27), we obtain

$$\bar{R}_{k-1}^t \mathbf{w}_{k-1} = \|\mathbf{b}\| \mathbf{e}_1. \quad (29)$$

We can evaluate $\mathbf{w}_{k-1}$ by forward substitution using (29). The rest of this section focuses on evaluating the right-hand side of (8).

Section 4 presents iterative formulas to efficiently update the approximate solutions $\mathbf{x}_k$. The remainder of this section discusses the evaluation of the right-hand side of (8). From (24) and (25), it can be deduced that $\mathbf{x}_k$ lives in $\mathcal{K}_{k-1}(A, Ab)$. Consequently, there is a vector $\mathbf{y}_{k-1} \in \mathbb{R}^{k-1}$ such that

$$A^{-1} \mathbf{x}_k = \mathcal{V}_{k-1} \mathbf{y}_{k-1}. \quad (30)$$

Using the decomposition (16), the k th iterate generated by our iterative method can be written as

$$\mathbf{x}_k = A\mathcal{V}_{k-1} \mathbf{y}_{k-1} = \mathcal{V}_k T_{k,k-1} \mathbf{y}_{k-1}.$$

Furthermore, according to (22) and (25), we have

$$\mathbf{x}_k = \mathcal{V}_k Q_{k,k-1} \mathbf{w}_{k-1}.$$

This implies that

$$Q_{k,k-1} \mathbf{w}_{k-1} = T_{k,k-1} \mathbf{y}_{k-1}. \quad (31)$$

Multiplying (31) by $Q_{k,k-1}^t$ from the left and using (18) yields

$$\mathbf{w}_{k-1} = Q_{k,k-1}^t T_{k,k-1} \mathbf{y}_{k-1} = \bar{R}_{k-1} \mathbf{y}_{k-1}. \quad (32)$$

By successively applying (12), (29), (30), and (32), we obtain

$$\mathbf{b}^t A^{-1} \mathbf{x}_k = \mathbf{b}^t \mathcal{V}_{k-1} \mathbf{y}_{k-1} = \|\mathbf{b}\| \mathbf{e}_1^t \mathbf{y}_{k-1} = \|\mathbf{b}\| \mathbf{e}_1^t \bar{R}_{k-1}^{-1} \mathbf{w}_{k-1} = \mathbf{w}_{k-1}^t \mathbf{w}_{k-1}. \quad (33)$$

According to (25), it follows that $\mathbf{x}_k^t \mathbf{x}_k = \mathbf{w}_{k-1}^t \mathbf{w}_{k-1}$. Combining this with (33) shows that Equation (8) can be represented as

$$\tilde{\mathbf{e}}_k^t \tilde{\mathbf{e}}_k = \mathbf{r}_k^t A^{-2} \mathbf{r}_k = \mathbf{b}^t A^{-2} \mathbf{b} - \mathbf{w}_{k-1}^t \mathbf{w}_{k-1}.$$

The term $\mathbf{w}_{k-1}^t \mathbf{w}_{k-1}$ can easily be computed using (29). Section 3 describes several Gauss-type quadrature rules that are applied to compute estimates of $\mathbf{b}^t A^{-2} \mathbf{b}$ in Section 4.

3. Quadrature Rules

This section considers the approximation of integrals like

$$\mathcal{I}(f) = \int_a^b f(x) dw(x) \quad (34)$$

by Gauss-type quadrature rules, where $[a, b] \subseteq [-\infty, \infty]$ and dw denotes a non-negative measure with an infinite number of support points such that all moments $\mu_j = \int_a^b x^j dw(x)$ exist, for $j = 0, 1, 2, \dots$. In this section, we assume that $\mu_0 = 1$. Let

$$\mathcal{Q}_n(f) = \sum_{i=1}^n w_i^{(n)} f(\xi_i^{(n)})$$

denote an n -node quadrature rule to approximate (34). Then

$$\int_a^b f(x) dw(x) = \mathcal{Q}_n(f) + \mathcal{E}_n(f),$$

where $\mathcal{E}_n(f)$ is the remainder term. This term vanishes for all polynomials in Π_d for some non-negative integer d . The value of d is referred to as the degree of precision of the quadrature rule. It is well known that the maximum value of d for an n -node quadrature rule is $2n - 1$. This value is achieved by the n -node Gauss rule (see, e.g., [13] for a proof). The latter rule can be written as

$$\mathcal{G}_n(f) = \sum_{i=1}^n w_i^{(n)} f(\xi_i^{(n)}). \quad (35)$$

The nodes $\zeta_i^{(n)}$, $i = 1, 2, \dots, n$, are the eigenvalues of the matrix

$$T_n = \begin{bmatrix} \alpha_1 & \beta_1 & & & \mathbf{0} \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ \mathbf{0} & & & \beta_{n-1} & \alpha_n \end{bmatrix} \in \mathbb{R}^{n \times n}, \quad (36)$$

and the weights $w_i^{(n)}$ are the square of the first elements of normalized eigenvectors.

The entries α_i and β_i of T_n are obtained from the recursion formula for the sequence of monic orthogonal polynomials $\{P_i\}_{i=0}^\infty$ associated with the inner product (15):

$$P_{i+1}(x) = (x - \alpha_i)P_i(x) - \beta_i^2 P_{i-1}(x), \quad i = 0, 1, 2, \dots, \quad (37)$$

where $P_{-1}(x) \equiv 0$ and $P_0(x) \equiv 1$. The values of α_i and β_i in (37) can be determined from the following formulas (see, e.g., Gautschi [13] for details):

$$\alpha_i = \frac{(xP_i, P_i)}{(P_i, P_i)}, \quad \beta_i^2 = \frac{(P_i, P_i)}{(P_{i-1}, P_{i-1})};$$

They also can be computed by the Lanczos process, which is presented in Algorithm 1.

Algorithm 1: The Lanczos algorithm.

input : Symmetric matrix $A \in \mathbb{R}^{m \times m}$, right-hand-side vector $\mathbf{b} \in \mathbb{R}^m \setminus \{\mathbf{0}\}$,
number of steps, n .
output: Vectors $\{\alpha_k\}_{k=0}^{n-1}$, and $\{\beta_k\}_{k=1}^n$, nonzero elements of the matrix T_n in (36),
and columns of the matrix $\mathcal{V} = [\mathbf{v}_1, \mathbf{v}_1, \dots, \mathbf{v}_{n+1}]$ in (50).

```

1  $\mathbf{v}_1 := \mathbf{b} / \|\mathbf{b}\|_2$ ;
2 for  $k \leftarrow 1$  to  $n$  do
3    $\hat{\mathbf{w}} := A\mathbf{v}_k$ ;
4   if  $k > 1$ 
5      $\hat{\mathbf{w}} := \hat{\mathbf{w}} - \beta_{k-1}\mathbf{v}_{k-1}$ ;
6   end
7    $\alpha_k := \mathbf{v}_k^t \hat{\mathbf{w}}$ ;
8    $\hat{\mathbf{w}} := \hat{\mathbf{w}} - \alpha_k \mathbf{v}_k$ ;
9    $\beta_k := \|\hat{\mathbf{w}}\|_2$ ;
10   $\mathbf{v}_{k+1} := \hat{\mathbf{w}} / \beta_k$ ;

```

It is straightforward to demonstrate that

$$\mathcal{G}_n(f) = \mathbf{e}_1^t f(T_n) \mathbf{e}_1, \quad (38)$$

where $\mathbf{e}_1 = [1, 0, \dots, 0]^t$.

We are interested in measures dw with support in two real intervals $[a, b]$ and $[c, d]$, where $a < b < 0 < c < d$. The following result sheds light on how the nodes of the Gauss rule (35) are allocated for such measures.

Theorem 3. *Let dw be a non-negative measure with support on the union of bounded real intervals $[a, b]$ and $[c, d]$, where $a < b < c < d$. Then, the Gauss rule (35) has at most one node in the open interval (b, c) .*

Proof. The result follows from [14] (Theorem 3.41.1). $\square$

The following subsection reviews some Gauss-type quadrature rules that are used to estimate the error in approximate solutions $\mathbf{x}_k$ of (1) that are generated by the iterative method proposed in Section 2.

Selected Gauss-Type Quadrature Rules

In [9], Laurie presented *anti-Gauss quadrature rules*. A recent analysis of anti-Gauss rules was also carried out by Díaz de Alba et al. [15]. Related investigations can be found in [16,17]. The $(n + 1)$ -point anti-Gauss rule $\check{\mathcal{G}}_{n+1}$, which is associated with the Gauss rule (35), is defined by the property

$$(\mathcal{I} - \check{\mathcal{G}}_{n+1})(f) = -(\mathcal{I} - \mathcal{G}_n)(f), \quad \text{for all } f \in \Pi_{2n+1}. \quad (39)$$

The following tridiagonal matrix is used to determine the rule $\check{\mathcal{G}}_{n+1}$:

$$\check{T}_{n+1} = \begin{bmatrix} \alpha_1 & \beta_1 & & & & & \mathbf{0} \\ \beta_1 & \alpha_2 & \beta_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} & & \\ & & & \beta_{n-1} & \alpha_n & \sqrt{2}\beta_n & \\ \mathbf{0} & & & & \sqrt{2}\beta_n & \alpha_{n+1} & \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}.$$

Similarly to (38), we have

$$\check{\mathcal{G}}_{n+1}(f) = \mathbf{e}_1^t \check{T}_{n+1} \mathbf{e}_1.$$

Moreover,

$$\check{\mathcal{G}}_n(f) = \mathcal{I}(f), \quad \text{for all } f \in \Pi_{2n-1}.$$

Further, Laurie [9] introduced the *averaged Gauss quadrature rule* associated with $\mathcal{G}_n$. It has $2n + 1$ nodes and is given by

$$\mathcal{A}_{2n+1} = \frac{1}{2}(\mathcal{G}_n + \check{\mathcal{G}}_{n+1}). \quad (40)$$

The property (39) suggests that the quadrature error for $\mathcal{A}_{2n+1}$ is smaller than the error for $\mathcal{G}_n$. Indeed, it follows from (39) that the degree of precision of $\mathcal{A}_{2n+1}$ is no less than $2n + 1$. This implies that the difference

$$\mathcal{A}_{2n+1}(f) - \mathcal{G}_n(f) \quad (41)$$

can be used to estimate the quadrature error

$$\mathcal{I}(f) - \mathcal{G}_n(f). \quad (42)$$

Computed results in [18] illustrate that for numerous integrands and various values of n , the difference (41) provides fairly accurate approximation of the quadrature error (42). The accuracy of these estimates depends both on the integrand and the value of n .

In [10], Spalević presented *optimal averaged Gauss quadrature rules*, which usually have a higher degree of precision than averaged Gauss rules with the same number of nodes. The symmetric tridiagonal matrix for the optimal averaged Gauss quadrature rule $\hat{\mathcal{A}}_{2n+1}$ with $2n + 1$ nodes is defined as follows. Introduce the reverse matrix of T_n , which is given by

$$T'_n = \begin{bmatrix} \alpha_n & \beta_{n-1} & & & \mathbf{0} \\ \beta_{n-1} & \alpha_{n-1} & \beta_{n-2} & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_2 & \alpha_2 & \beta_1 \\ \mathbf{0} & & & \beta_1 & \alpha_1 \end{bmatrix} \in \mathbb{R}^{n \times n},$$

as well as the concatenated matrix

$$\hat{T}_{2n+1} = \begin{bmatrix} T_n & \beta_n \mathbf{e}_n & \mathbf{0} \\ \beta_n \mathbf{e}_n^t & \alpha_{n+1} & \beta_{n+1} \mathbf{e}_1^t \\ \mathbf{0} & \beta_{n+1} \mathbf{e}_1 & T'_n \end{bmatrix} \in \mathbb{R}^{(2n+1) \times (2n+1)}.$$

The nodes of the rule $\hat{\mathcal{A}}_{2n+1}$ are the eigenvalues, and the weights are the squared first components of normalized eigenvectors of the matrix $\hat{T}_{2n+1}$. It is worth noting that n of the nodes of $\hat{\mathcal{A}}_{2n+1}$ agree with the nodes of $\mathcal{G}_n$. Similarly to Equation (38), we have

$$\hat{\mathcal{A}}_{2n+1}(f) = \mathbf{e}_1^t f(\hat{T}_{2n+1}) \mathbf{e}_1.$$

The degree of precision for this quadrature rule is at least $2n + 2$. Analyses of the degree of precision of the rules $\hat{\mathcal{A}}_{2n+1}$ and the location of their largest and smallest nodes for several measures for which explicit expressions for coefficients α_i and β_i are known can be found in [19] and references therein. An estimate of the quadrature error in the Gauss rule (35) is given by

$$\hat{\mathcal{A}}_{2n+1}(f) - \mathcal{G}_n(f). \quad (43)$$

Numerical examples provided in [18] show this estimate to be quite accurate for a wide range of integrands. As the rule $\hat{\mathcal{A}}_{2n+1}$ typically has strictly higher degree of precision than Laurie's averaged Gauss rule (40), we expect the quadrature error estimate (43) to generally be more accurate than the estimate (41), particularly for integrands with high-order differentiability.

In the computations, we use the representation

$$\hat{\mathcal{A}}_{2n+1}(f) = \frac{\beta_{n+1}^2}{\beta_n^2 + \beta_{n+1}^2} \mathcal{G}_n(f) + \frac{\beta_n^2}{\beta_n^2 + \beta_{n+1}^2} \mathcal{G}_{n+1}^*(f),$$

where

$$\mathcal{G}_{n+1}^*(f) = \mathbf{e}_1^t f(T_{n+1}^*) \mathbf{e}_1$$

with

$$T_{n+1}^* = \begin{bmatrix} \alpha_1 & \beta_1 & & & \mathbf{0} \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n & \beta_n^* \\ \mathbf{0} & & & & \beta_n^* & \alpha_{n+1} \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}$$

and $\beta_n^* = \sqrt{\beta_n^2 + \beta_{n+1}^2}$.

We finally consider the Gauss-Radau quadrature rule $\mathcal{R}_{n+1,0}(f)$, which has $n + 1$ nodes, with one node anchored at 0. This rule can be written as

$$\mathcal{R}_{n+1,0}(f) = w_0^{(n+1)} f(0) + \sum_{i=1}^n w_i^{(n+1)} f(\xi_{i,0}^{(n+1)}). \quad (44)$$

To maximize the degree of precision, which is $2n$, the n nodes $\zeta_{i,0}^{(n+1)}$, $i = 1, 2, \dots, n$, are suitably chosen. The rule $\mathcal{R}_{n+1,0}(f)$ can be expressed as

$$\mathcal{R}_{n+1,0}(f) = \mathbf{e}_1^t f(T_{n+1,0}) \mathbf{e}_1,$$

where

$$T_{n+1,0} = \begin{bmatrix} \alpha_1 & \beta_1 & & & & & \mathbf{0} \\ \beta_1 & \alpha_2 & \beta_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} & & \\ & & & \beta_{n-1} & \alpha_n & \beta_n & \\ \mathbf{0} & & & & \beta_n & \tilde{\alpha}_{n+1} & \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}, \quad (45)$$

and the entry $\tilde{\alpha}_{n+1}$ is chosen so that $T_{n+1,0}$ has an eigenvalue at the origin. Details on how to determine $\tilde{\alpha}_{n+1}$ are provided by Gautschi [13,20] and Golub [21].

Theorem 4. Let the nodes $\{\zeta_{i,0}^{(n+1)}\}_{i=1}^{n+1}$ of the rule (44) and the nodes $\{\zeta_i^{(n)}\}_{i=1}^n$ of the rule (35) be ordered according to increasing magnitude. Then $\zeta_{0,0}^{(n+1)} = 0$ and

$$|\zeta_{i,0}^{(n+1)}| > |\zeta_i^{(n)}|, \quad i = 1, \dots, n.$$

Proof. The last subdiagonal entry of the matrix (45) is non-vanishing since the measure dw in (34) has infinitely many support points. By the Cauchy interlacing theorem, the eigenvalues of the leading principal $n \times n$ submatrix (36) of the symmetric tridiagonal $(n+1) \times (n+1)$ matrix (45) strictly interlace the eigenvalues of the latter. Since one of the eigenvalues of (45) vanishes, the theorem follows. $\square$

The measure dw has no support at the origin. We therefore apply the Gauss-Radau rules with $f(0) = 0$ in (44).

Finally, we compute error-norm estimates by using the “minimum rule”:

$$\mathcal{A}_{4n+2}^{\min}(f) = \min\{\mathcal{A}_{2n+1}(f), \hat{\mathcal{A}}_{2n+1}(f)\}. \quad (46)$$

It typically has $3n+2$ distinct nodes. This rule is justified by the observation that the rules $\mathcal{A}_{2n+1}(f)$ and $\hat{\mathcal{A}}_{2n+1}(f)$ sometimes overestimate the error norm.

4. Error-Norm Estimation

This section outlines how the quadrature rules discussed in the previous section can be used to estimate the Euclidean norm of the error in the iterates $\mathbf{x}_k$, $k = 0, 1, \dots$, determined by the iterative method described in Section 2. The initial iterate is assumed to be $\mathbf{x}_0 = \mathbf{0}$. Then the iterate $\mathbf{x}_n$ lives in the Krylov subspace (cf. (24) and (25)),

$$\mathcal{K}_{n-1}(A, A\mathbf{b}) = \text{span}\{A\mathbf{b}, A^2\mathbf{b}, \dots, A^{n-1}\mathbf{b}\},$$

The residual corresponding to the iterate $\mathbf{x}_n$ is defined in (6). We use the relation (7) to obtain the Euclidean norm of the error, cf. (8). Our task is to estimate the first term in the right-hand side of (8). In this section, the measure is defined by (10). In particular, the measure has support in intervals on the negative and positive real axis. These intervals exclude an interval around the origin.

We turn to the computation of the iterates $\mathbf{x}_k$ described by (25). The computations can be structured in such a way that only a few m -vectors need to be stored. Let the $k \times k$ real matrix T_k be given by (36) with $n = k$, i.e.,

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & & & & & \mathbf{0} \\ & \beta_1 & \alpha_2 & \beta_2 & & & & & \\ & & \beta_2 & \alpha_3 & & & & & \\ & & & & \ddots & & \ddots & & \\ & & & & \ddots & \alpha_{k-2} & \beta_{k-2} & & \\ & & & & & \beta_{k-2} & \alpha_{k-1} & \beta_{k-1} & \\ & & & & & & \beta_{k-1} & \alpha_k & \\ \mathbf{0} & & & & & & & & \end{bmatrix}. \quad (47)$$

Based on the discussion following Equation (12), we may assume that the β_j are nonzero. This ensures that the eigenvalues of T_k are distinct. We can compute the QR factorization (17) of T_k by applying a sequence of $k - 1$ Givens rotations to T_k ,

$$G_k^{(j)} := \begin{bmatrix} I_{j-1} & & \\ & c_j & s_j \\ & -s_j & c_j \\ & & & I_{k-j-1} \end{bmatrix} \in \mathbb{R}^{k \times k}, \quad c_j^2 + s_j^2 = 1, \quad s_j \geq 0, \quad (48)$$

This yields an orthogonal matrix Q_k and an upper triangular matrix R_k given by

$$\begin{aligned} Q_k &:= G_k^{(1)T} G_k^{(2)T} \dots G_k^{(k-1)T}, \\ R_k &:= G_k^{(k-1)} G_k^{(k-2)} \dots G_k^{(1)} T_k. \end{aligned} \quad (49)$$

For a discussion on Givens rotations, see, e.g., ([2], Chapter 5). In our iterative method, the matrix Q_k is not explicitly formed; instead, we use the representation in (49). Since T_k is tridiagonal, the upper triangular matrix R_k has nonzero entries solely on the diagonal and the two adjacent superdiagonals.

Application of k steps of the Lanczos process to the matrix A with initial vector $\mathbf{b}$ results in the matrix T_k , as shown in (47). By performing one more step, analogous to (11), we obtain the following Lanczos decomposition:

$$A\mathcal{V}_{k+1} = \mathcal{V}_{k+1}T_{k+1} + \mathbf{f}_{k+1}\mathbf{e}_{k+1}^t. \quad (50)$$

We observe that the last subdiagonal element of the symmetric tridiagonal matrix T_{k+1} can be calculated as $\beta_k = \|\mathbf{f}_k\|$ right after the k th Lanczos step is completed. This is convenient when evaluating the tridiagonal matrices (45) associated with Gauss-Radau quadrature rules.

We can express the matrix T_{k+1} in terms of its QR factorization as follows:

$$T_{k+1} = Q_{k+1}R_{k+1}, \quad (51)$$

whose factors can be computed from Q_k and R_k in a straightforward manner. Indeed, we have

$$\begin{aligned} Q_{k+1} &= \begin{bmatrix} Q_k & \mathbf{0} \\ \mathbf{0}^t & 1 \end{bmatrix} G_{k+1}^{(k)T}, \\ Q_{k+1,k} &= \begin{bmatrix} Q_k & \mathbf{0} \\ \mathbf{0}^t & 1 \end{bmatrix} G_{k+1,k}^{(k)T}, \end{aligned} \quad (52)$$

wherein Q_{k+1} is a $(k+1) \times (k+1)$ real orthogonal matrix; $Q_{k+1,k}$ is a $(k+1) \times k$ real matrix; $G_{k+1}^{(k)}$ is defined by (48); and $G_{k+1,k}^{(k)}$ is a real $(k+1) \times k$ matrix, which is made up of the first k columns of $G_{k+1}^{(k)}$.

We express the matrices in terms of their columns to derive updating formulas for the computation of the triangular matrix R_{k+1} in (51) from R_k in (49):

$$R_k = [\mathbf{r}_1^{(k)}, \mathbf{r}_2^{(k)}, \dots, \mathbf{r}_k^{(k)}], \quad R_{k+1} = [\mathbf{r}_1^{(k+1)}, \mathbf{r}_2^{(k+1)}, \dots, \mathbf{r}_k^{(k+1)}, \mathbf{r}_{k+1}^{(k+1)}].$$

A comparison between (17) and (51) yields the following results:

$$\mathbf{r}_j^{(k+1)} = \begin{bmatrix} \mathbf{r}_j^{(k)} \\ 0 \end{bmatrix}, \quad 1 \leq j < k,$$

and

$$\begin{aligned} \mathbf{r}_k^{(k+1)} &= G_{k+1}^{(k)} \begin{bmatrix} \mathbf{r}_k^{(k)} \\ \beta_k \end{bmatrix}, \\ \mathbf{r}_{k+1}^{(k+1)} &= G_{k+1}^{(k)} G_{k+1}^{(k-1)} T_{k+1} \mathbf{e}_{k+1}. \end{aligned} \quad (53)$$

Thus, the elements of all the matrices $R_1, R_2, \dots, R_{k+1}$ can be calculated in just $O(k^2)$ arithmetic floating-point operations (flops).

As defined by (18), the matrix expressed as $\bar{R}_k = [\bar{r}_{j\ell}^{(k)}]_{j,\ell=1}^k$ is the leading principal submatrix of R_{k+1} of order k , and differs from $R_k = [r_{j\ell}^{(k)}]_{j,\ell=1}^k$ only in its last diagonal entry. From Equation (53) and the fact that β_k is nonzero, it follows that $\bar{r}_{kk}^{(k)} > r_{kk}^{(k)} \geq 0$. When T_k is nonsingular, we obtain that $r_{kk}^{(k)} > 0$. Here we assume that the diagonal entries of the upper triangular matrix in all QR factorizations are non-negative.

We turn to the computation of the columns of the matrix $\tilde{V}_{k+1}$, that is,

$$\tilde{V}_{k+1} = [\tilde{\mathbf{v}}_1^{(k+1)}, \tilde{\mathbf{v}}_2^{(k+1)}, \dots, \tilde{\mathbf{v}}_{k+1}^{(k+1)}] := \mathcal{V}_{k+1} Q_{k+1}, \quad (54)$$

from columns of $\tilde{V}_k$, where $\mathcal{V}_{k+1}$ is obtained by the modified Lanczos scheme (50), see Algorithm 1, and Q_{k+1} is determined by (52). Substituting (52) into the right-hand side of (54) yields

$$\begin{aligned} \tilde{V}_{k+1} &= [\mathcal{V}_k, \mathbf{v}_{k+1}] Q_{k+1} = [\tilde{V}_k, \mathbf{v}_{k+1}] G_{k+1}^{(k)T} \\ &= [\tilde{V}_{k-1}, c_k \tilde{\mathbf{v}}_k^{(k)} + S_k \mathbf{v}_{k+1}, -S_k \tilde{\mathbf{v}}_k^{(k)} + c_k \mathbf{v}_{k+1}]. \end{aligned} \quad (55)$$

As a result, the initial $k-1$ columns of the matrix $\tilde{V}_{k+1}$ correspond to those of $\tilde{V}_{k-1}$. The columns $\tilde{\mathbf{v}}_k^{(k+1)}$ and $\tilde{\mathbf{v}}_{k+1}^{(k+1)}$ in $\tilde{V}_{k+1}$ are linear combinations obtained from the last columns of both $\tilde{V}_k$ and $\mathcal{V}_{k+1}$.

Given the solution $\mathbf{w}_{k-1}$ of the linear system (29) and considering that $\bar{R}_k$ is upper triangular, with $\bar{R}_{k-1}$ as the leading principal submatrix of order $k-1$, the computation of the solution $\mathbf{w}_k = [\eta_1, \eta_2, \dots, \eta_k]^t$ of $\bar{R}_k^t \mathbf{w}_k = \|\mathbf{b}\| \mathbf{e}_1$ is inexpensive. We find that

$$\mathbf{w}_k = \begin{bmatrix} \mathbf{w}_{k-1} \\ \eta_k \end{bmatrix}, \quad \eta_k = -(\bar{r}_{k-2,k}^{(k)} \eta_{k-2} + \bar{r}_{k-1,k}^{(k)} \eta_{k-1}) / \bar{r}_{kk}^{(k)}.$$

Note that the computation of $\mathbf{w}_k$ from $\mathbf{w}_{k-1}$ only requires the last column of the matrix $\bar{R}_k$.

We are now in a position to compute $\mathbf{x}_{k+1}$ from $\mathbf{x}_k$. Using Equations (25) and (55), we obtain

$$\mathbf{x}_{k+1} = \tilde{V}_k \mathbf{w}_k = \tilde{V}_{k-1} \mathbf{w}_{k-1} + \eta_k \tilde{\mathbf{v}}_k^{(k+1)} = \mathbf{x}_k + \eta_k \tilde{\mathbf{v}}_k^{(k+1)},$$

Note that only the last few columns of $\mathcal{V}_k$ and $\tilde{V}_k$ are required to update the iterate $\mathbf{x}_{k+1}$.

Algorithm 1 shows pseudo-code for computing the nontrivial elements of the matrix T_n in (36) and the matrix $\mathcal{V}$ in (50). Each iteration requires the evaluation of one matrix-

vector product with the matrix A and a few operations with m -vectors. The latter only require $O(m)$ flops.

5. Computed Examples

This section illustrates the performance of the iterative method of Section 2 and the error estimates of Section 4 when applied to three linear systems of Equation (1). In all examples, the matrix A is symmetric, nonsingular, and indefinite. All computations were carried out on a laptop computer using MATLAB R2024b. The initial approximate solution was set to $x_0 = 0$, and iterations were terminated once the Euclidean norm of the error fell below 10^{-6} .

The matrix $A \in \mathbb{R}^{m \times m}$ in the computed examples were determined from their spectral factorization (2), where the eigenvector matrix U is a random orthogonal matrix and the eigenvalues are distributed on the positive and negative real axis. The exact solution for all examples is $x_* = [1, 1, \dots, 1]^t \in \mathbb{R}^m$. The approximation of the Euclidean norm of the errors in the iterates was done using the quadrature rules of Section 3.

Problem 1. In this example, we set $m = 491$. The spectrum of the symmetric matrix A is in the union of the intervals: $[-150, -10] \cup [1, 350]$. The eigenvalues of A are given by

$$\lambda_i = \begin{cases} i - 151 & \text{for } 1 \leq i \leq 141, \\ i - 141 & \text{for } 142 \leq i \leq 491. \end{cases}$$

We have $\text{cond}(A) = 350$. Figure 1 displays numerical results for this example. The plots show the convergence history of the Gauss, Gauss-Radau, optimal averaged Gauss, and minimum $\mathcal{A}_{4n+2}^{\min}(f)$ quadrature rules. The plots show all rules considered to be convergent. Furthermore, the Gauss-Radau and $\mathcal{A}_{4n+2}^{\min}(f)$ rules display the fastest convergence.

Problem 2. Let $m = 200$ and let the spectrum of the symmetric matrix A be in the union of two intervals: $[-7.3891, -2.7456] \cup [1.0513, 148.4132]$. The eigenvalues of A are given by

$$\lambda_i = \begin{cases} \exp(i/20) & \text{for } 1 \leq i \leq 100, \\ -\exp(i/100) & \text{for } 101 \leq i \leq 200. \end{cases}$$

Thus, the matrix A has 100 negative eigenvalues, and its condition number is $\text{cond}(A) \approx 147$. Figure 2 shows the convergence history for the Gauss, Gauss-Radau, optimal averaged Gauss, and $\mathcal{A}_{4n+2}^{\min}(f)$ rules.

Problem 3. In this example, we use the symmetric matrix $A \in \mathbb{R}^{100 \times 100}$ with eigenvalues

$$\lambda_i = \begin{cases} -i^{-2} & \text{for } 1 \leq i \leq 50, \\ i^{-3} & \text{for } 51 \leq i \leq 100. \end{cases}$$

Thus, the matrix A has 50 negative eigenvalues, with its spectrum in the intervals of $[-1, -4 \cdot 10^{-4}] \cup [10^{-6}, 7.5386 \cdot 10^{-6}]$ and $\text{cond}(A) = 1.25 \cdot 10^8$. Figure 3 displays the convergence history for the Gauss, Gauss-Radau, optimal averaged Gauss, and minimum $\mathcal{A}_{4n+2}^{\min}(f)$ rules.

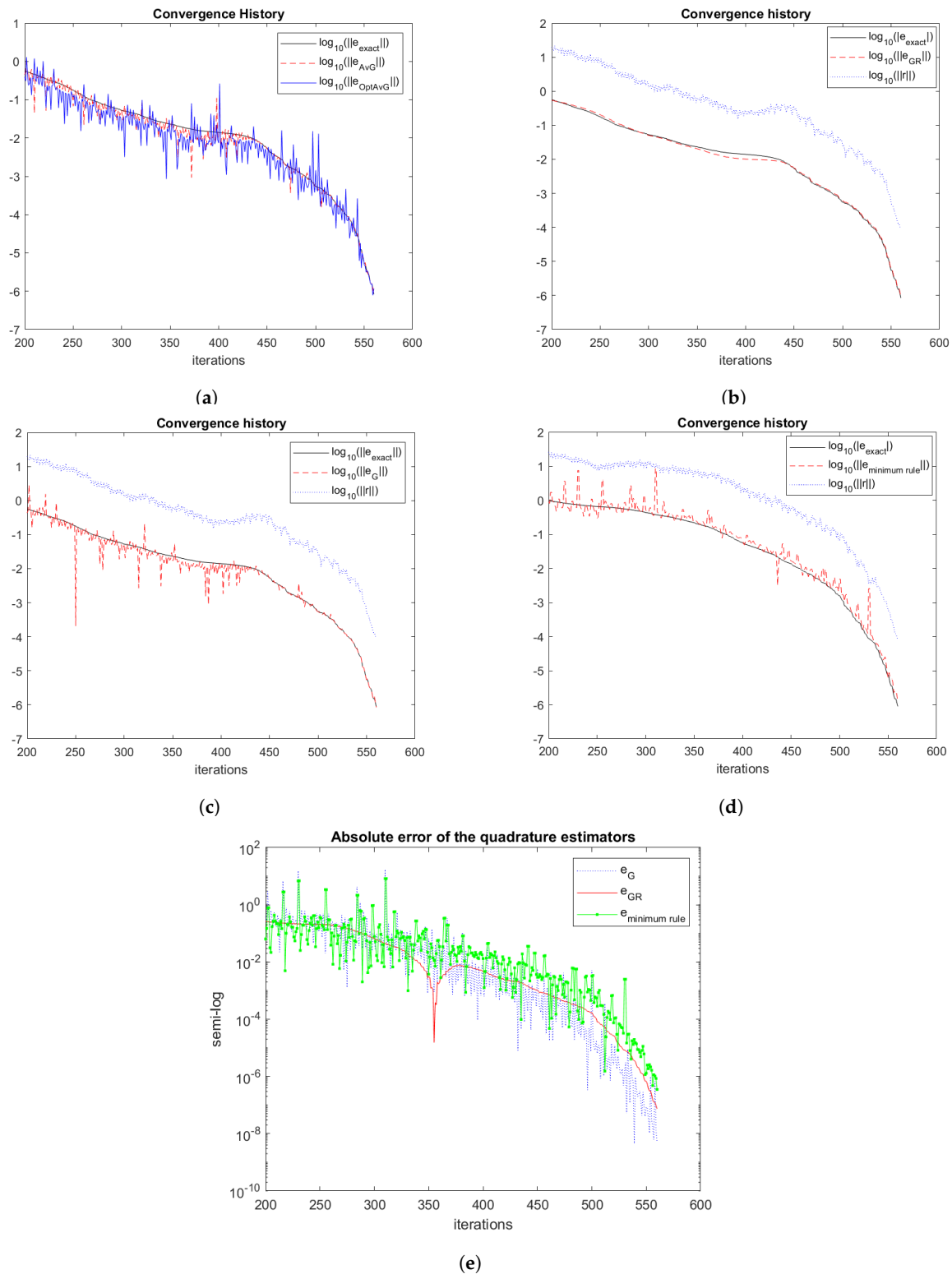


Figure 1. Problem 1. Plots (a–d): Convergence history of quadrature rules. (a) Convergence history of averaged (the dashed red curve) and optimal averaged (the blue curve) rules in comparison with the exact solution (the black curve). (b) Convergence history of Gauss-Radau (the dashed red curve) rule in comparison with the exact solution (the black curve) and the residual vector (the dotted blue curve). (c) Convergence history of Gauss rule (the dashed red curve) in comparison with the exact solution (the black curve) and the residual vector (the dotted blue curve). (d) Convergence history of the $\mathcal{A}_{4n+2}^{\min}(f)$ rule (the dashed red curve) in comparison with the exact solution (the black curve) and the residual vector (the dotted blue curve). (e) Absolute error of Gauss (the dotted blue curve), Gauss-Radau (the red curve), and $\mathcal{A}_{4n+2}^{\min}(f)$ (the green curve) rules, in a semi-logarithmic plot.

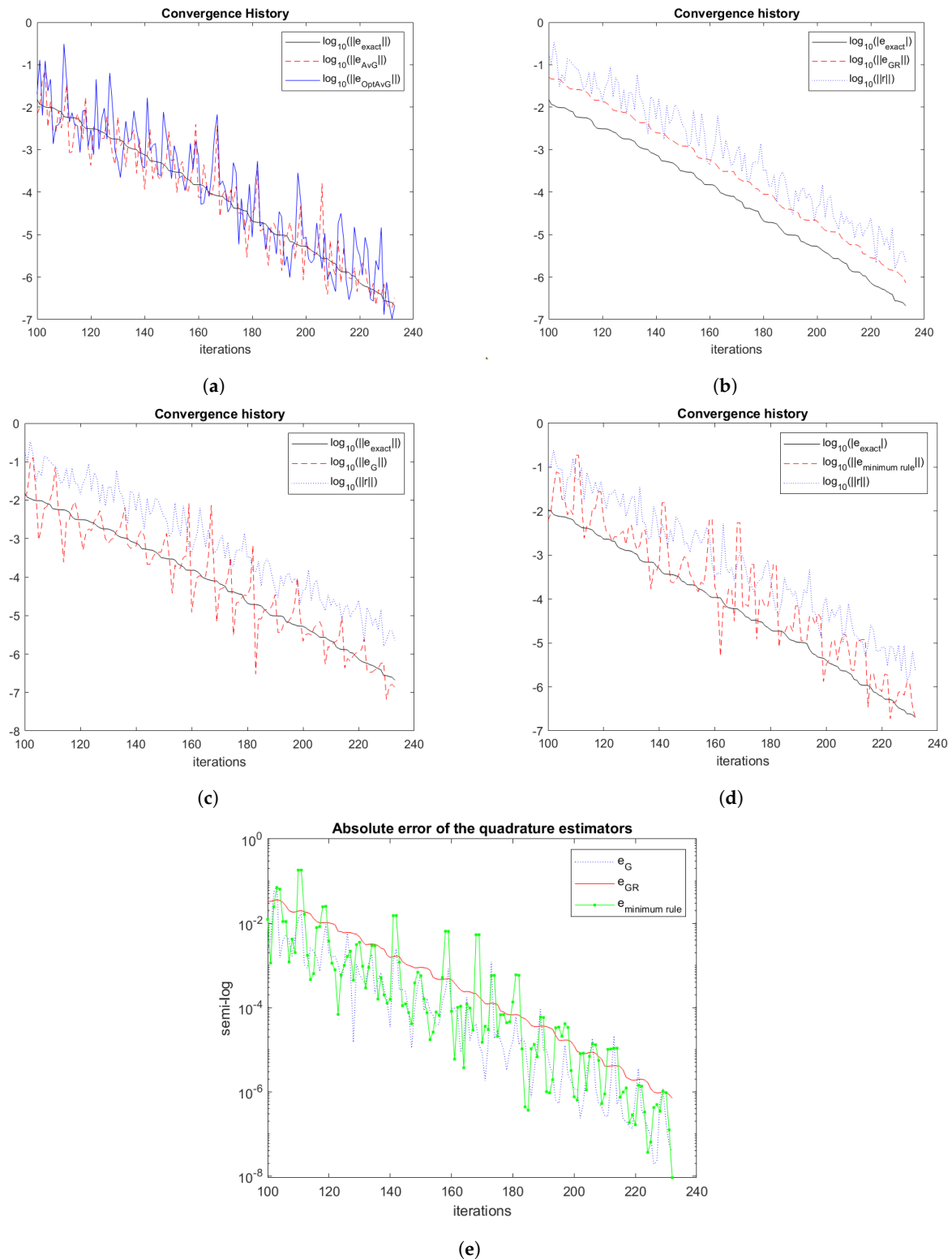


Figure 2. Problem 2. Plots (a–d): convergence history of quadrature rules. (a) Convergence history of averaged (the dashed red curve) and optimal averaged (the blue curve) rules in comparison with the exact solution (the black curve). (b) Convergence history of the Gauss–Radau rule (the dashed red curve) in comparison with the exact solution (the black curve) and the residual vector (the dotted blue curve). (c) Convergence history of the Gauss rule (the dashed red curve) in comparison with the exact solution (the black curve) and the residual vector (the dotted blue curve). (d) Convergence history of the $\mathcal{A}_{4n+2}^{\min}(f)$ rule (the dashed red curve) in comparison with the exact solution (the black curve) and the residual vector (the dotted blue curve). (e) Absolute error of the Gauss (the dotted blue curve), Gauss–Radau (the red curve), and $\mathcal{A}_{4n+2}^{\min}(f)$ (the green curve) rules in a semi-logarithmic plot.

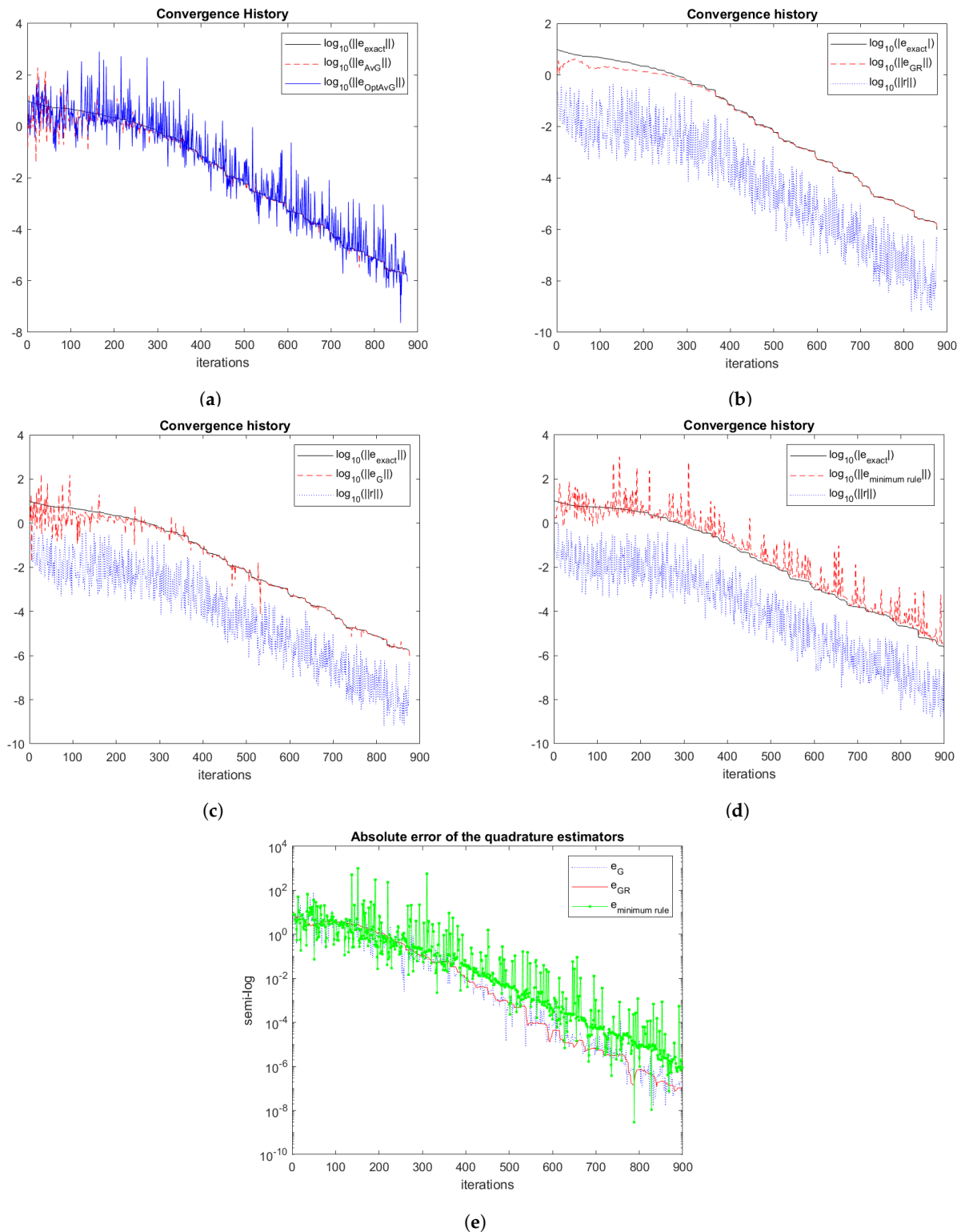


Figure 3. Problem 3. Plots (a–d): convergence history of quadrature rules. (a) Convergence history of averaged (the dashed red curve) and optimal averaged (the blue curve) rules in comparison with the exact solution (the black curve). (b) Convergence history of the Gauss–Radau rule (the dashed red curve) in comparison with the exact solution (the black curve) and the residual vector (the dotted blue curve). (c) Convergence history of the Gauss rule (the dashed red curve) in comparison with the exact solution (the black curve) and the residual vector (the dotted blue curve). (d) Convergence history of the $\mathcal{A}_{4n+2}^{\min}(f)$ rule (the dashed red curve) in comparison with the exact solution (the black curve) and the residual vector (the dotted blue curve). (e) Absolute error of the Gauss (the dotted blue curve), Gauss–Radau (the red curve), and $\mathcal{A}_{4n+2}^{\min}(f)$ (the green curve) rules in a semi-logarithmic plot.

6. Concluding Remarks

This paper describes novel techniques for estimating the Euclidean norm of the error in iterates obtained with a Krylov subspace iterative method. The method is designed to solve linear systems of equations with a symmetric, nonsingular, indefinite matrix. The error norm estimates are obtained by using the relation between the Krylov subspace iterative method and Gauss-type quadrature rules. The computed results demonstrate the performance of the iterative method and the error norm estimates. Among the considered quadrature rules, the rules (44) and (46) provide the most accurate error-norm estimates.

Author Contributions: Investigation, M.A., M.T.D., L.R. and M.M.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research by M.M.S. was supported, in part, by the Serbian Ministry of Science, Technological Development, and Innovations according to Contract 451-03-65/2024-03/200105 dated 5 February 2024.

Data Availability Statement: The data that support the findings of the study are available from the authors upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest.

References

1. Paige, C.C.; Saunders, M.A. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.* **1975**, *12*, 617–629. [\[CrossRef\]](#)
2. Golub, G.H.; Loan, C.F.V. *Matrix Computations*, 4th ed.; Johns Hopkins University Press: Baltimore, MD, USA, 2013.
3. Almutari, H.; Meurant, G.; Reichel, L.; Spalević, M.M. New error estimates for the conjugate gradient method. *J. Comput. Appl. Math.* **2025**, *459*, 116357. [\[CrossRef\]](#)
4. Golub, G.H.; Meurant, G. Matrices, moments and quadrature. In *Numerical Analysis 1993*; Griffiths, D.F., Watson, G.A., Eds.; Longman: Essex, UK, 1994; pp. 105–156.
5. Golub, G.H.; Meurant, G. *Matrices, Moments and Quadrature with Applications*; Princeton University Press: Princeton, NJ, USA, 2010.
6. Meurant, G.; Tichý, P. *Error Norm Estimation in the Conjugate Gradient Algorithm*; SIAM: Philadelphia, PA, USA, 2024.
7. Meurant, G.; Tichý, P. On computing quadrature-based bounds for the A -norm of the error in conjugate gradients. *Numer. Algorithms* **2013**, *62*, 163–191. [\[CrossRef\]](#)
8. Calvetti, D.; Morigi, S.; Reichel, L.; Sgallari, F. An iterative method with error estimators. *J. Comput. Appl. Math.* **2001**, *127*, 93–119. [\[CrossRef\]](#)
9. Laurie, D.P. Anti-Gaussian quadrature formulas. *Math. Comp.* **1996**, *6*, 739–747. [\[CrossRef\]](#)
10. Spalević, M.M. On generalized averaged Gaussian formulas. *Math. Comput.* **2007**, *76*, 1483–1492. [\[CrossRef\]](#)
11. Saad, Y. *Iterative Methods for Sparse Linear Systems*, 2nd ed.; SIAM: Philadelphia, PA, USA, 2003.
12. Fischer, B. *Polynomial Based Iteration Methods for Symmetric Linear Systems*; Teubner-Wiley: New York, NY, USA, 1996.
13. Gautschi, W. *Orthogonal Polynomials, Computation and Approximation*; Oxford University Press: Oxford, UK, 2004.
14. Szegő, G. *Orthogonal Polynomials*, 4th ed.; American Mathematical Society: Providence, RI, USA, 1975.
15. Díaz de Alba, P.; Fermo, L.; Rodriguez, G. Solution of second kind Fredholm integral equations by means of Gauss and anti-Gauss quadrature rules. *Numer. Math.* **2020**, *146*, 699–728. [\[CrossRef\]](#)
16. Hascelik, A.I. Modified anti-Gauss and degree optimal average formulas for Gegenbauer measure. *Appl. Numer. Math.* **2008**, *58*, 171–179. [\[CrossRef\]](#)
17. Notaris, S.E. Anti-Gaussian quadrature formulae based on the zeros of Stieltjes polynomials. *BIT Numer. Math.* **2018**, *58*, 179–198. [\[CrossRef\]](#)
18. Reichel, L.; Spalević, M.M. Averaged Gauss quadrature formulas: Properties and applications. *J. Comput. Appl. Math.* **2022**, *410*, 114232. [\[CrossRef\]](#)
19. Djukić, D.L.; Mutavdžić Djukić, R.M.; Reichel, L.; Spalević, M.M. Weighted averaged Gaussian quadrature rules for modified Chebyshev measures. *Appl. Numer. Math.* **2024**, *200*, 195–208. [\[CrossRef\]](#)

-
20. Gautschi, W. The interplay between classical analysis and (numerical) linear algebra—A tribute to Gene H. Golub. *Electron. Trans. Numer. Anal.* **2002**, *13*, 119–147.
 21. Golub, G.H. Some modified matrix eigenvalue problems. *SIAM Rev.* **1973**, *15*, 318–334. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.