New block quadrature rules for the approximation of matrix functions

Lothar Reichel

Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA.

Giuseppe Rodriguez

Dipartimento di Matematica e Informatica, Università di Cagliari, viale Merello 92, 09123 Cagliari, Italy.

Tunan Tang

Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA.

Abstract

Golub and Meurant have shown how to use the symmetric block Lanczos algorithm to compute block Gauss quadrature rules for the approximation of certain matrix functions. We describe new block quadrature rules that can be computed by the symmetric or nonsymmetric block Lanczos algorithms and yield higher accuracy than standard block Gauss rules after the same number of steps of the symmetric or nonsymmetric block Lanczos algorithms. The new rules are block generalizations of the generalized averaged Gauss rules introduced by Spalević. Applications to network analysis are presented.

Keywords: Matrix functions, Gauss quadrature, block Lanczos algorithm, complex networks.
2010 MSC: 65F60, 65D32, 05C50, 05C82.

1. Introduction

The aim of this paper is to describe new methods for the approximation of expressions of the form

$$W^T f(A)W, \tag{1.1}$$

where $A \in \mathbb{R}^{m\times m}$ is a large symmetric matrix, $W \in \mathbb{R}^{m\times k}$ has a few orthonormal columns, i.e., $1 \le k \ll m$, and f is a function such that $f(A)$ is well defined. The superscript ^{T} denotes transposition. We also consider expressions of the type

$$W^T f(A)V, \tag{1.2}$$

in which $A \in \mathbb{R}^{m\times m}$ is a large possibly nonsymmetric matrix and the matrices $W, V \in \mathbb{R}^{m\times k}$, with $1 \le k \ll m$, are biorthogonal, i.e., $W^T V = I_k$. Throughout this paper I_k denotes the identity matrix of order k.

The matrix function $f(A)$ can be defined, e.g., by the spectral factorization of A, assuming that it exists; see, e.g., [22, 24] for discussions on several possible definitions of matrix functions. In the present paper, we assume that the matrix A is so large that it is unfeasible or impractical to evaluate its spectral factorization.

For symmetric matrices A, Golub and Meurant [20, 21] show how approximations of (1.1) can be conveniently computed by first carrying out $\ell \ll m/k$ steps with the symmetric block Lanczos algorithm applied to A with initial block vector W. This algorithm produces the decomposition

$$A[W_1, \ldots, W_\ell] = [W_1, \ldots, W_\ell]J_\ell + W_{\ell+1}\Gamma_\ell E_\ell^T, \tag{1.3}$$

Email addresses: reichel@math.kent.edu (Lothar Reichel), rodriguez@unica.it (Giuseppe Rodriguez), nickdon2006@gmail.com (Tunan Tang)

where the block vectors $W_j \in \mathbb{R}^{m\times k}$ are orthonormal, i.e.,

$$W_i^T W_j = \begin{cases} I_k, & i = j, \\ O_k, & i \neq j, \end{cases}$$

with $W_1 = W$. Here and below O_k denotes the zero matrix of order k. Moreover, $E_\ell = [O_k, \ldots, O_k, I_k]^T \in \mathbb{R}^{k\ell\times k}$ and the matrix

$$J_\ell = \begin{bmatrix} \Omega_1 & \Gamma_1^T & & & O \\ \Gamma_1 & \Omega_2 & \Gamma_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{\ell-2} & \Omega_{\ell-1} & \Gamma_{\ell-1}^T \\ O & & & \Gamma_{\ell-1} & \Omega_\ell \end{bmatrix} \in \mathbb{R}^{k\ell\times k\ell} \tag{1.4}$$

is block tridiagonal with symmetric diagonal blocks $\Omega_i \in \mathbb{R}^{k\times k}$. The subdiagonal blocks $\Gamma_i \in \mathbb{R}^{k\times k}$ may be chosen to be upper triangular, but this is not necessary. The remainder term in (1.3) contains the matrix Γ_ℓ, which is the last subdiagonal block in the symmetric block tridiagonal matrix $J_{\ell+1} \in \mathbb{R}^{k(\ell+1)\times k(\ell+1)}$ that would have been obtained if $\ell + 1$ steps with the symmetric block Lanczos algorithm were carried out. We assume that ℓ is chosen small enough so that the recursion relations of the symmetric block Lanczos method do not break down. Remedies for breakdown are commented on in Section 2.

Golub and Meurant [20, 21] show that

$$\mathcal{G}_\ell f = E_1^T f(J_\ell) E_1, \tag{1.5}$$

where $E_1 = [I_k, O_k, \ldots, O_k]^T \in \mathbb{R}^{k\ell\times k}$, can be used to approximate (1.1). Here we have used that W has orthonormal columns. In fact, $\mathcal{G}_\ell f$ can be interpreted as an ℓ-block Gauss quadrature rule, i.e.,

$$\mathcal{G}_\ell f = W^T f(A) W \qquad \forall f \in \mathbb{P}_{2\ell-1}, \tag{1.6}$$

where $\mathbb{P}_{2\ell-1}$ denotes the set of all polynomials of degree at most $2\ell - 1$; see [20, 21] or Section 2 for details. While the matrix A is assumed to be so large that it is difficult to evaluate $f(A)$, the number of steps ℓ of the symmetric block Lanczos algorithm typically can be chosen small enough so that $f(J_\ell)$ in (1.5) can be conveniently computed by one of the methods for evaluating functions of small to moderately sized matrices described by Higham [24].

The matrix Γ_ℓ in the decomposition (1.3) is not used by the block Gauss rule (1.5). We will present block quadrature rules that use all the blocks $\Omega_1, \ldots, \Omega_\ell$ and $\Gamma_1, \Gamma_1^T, \ldots, \Gamma_{\ell-1}, \Gamma_{\ell-1}^T$ of the matrix (1.4) as well as the matrices Γ_ℓ and Γ_ℓ^T, and are exact for all $f \in \mathbb{P}_{2\ell}$. We therefore can expect these rules to yield more accurate approximations of (1.1) than the block Gauss rule (1.5) for many functions f. The construction of the new block rules requires the same number of block Lanczos steps, and therefore the same number of matrix-block-vector product evaluations with the matrix A, as the construction of the block Gauss rule (1.5). When the matrix A is large, the dominating computational effort for evaluating (1.5) is the computation of these matrix-block-vector products. Therefore the new block quadrature rules of this paper require about the same computational effort as the Gauss rule (1.5), but they are exact for a larger class of polynomials.

This paper is organized as follows. We review results on block Gauss rules by Golub and Meurant [20, 21] in Section 2, where we also describe the new block quadrature rules mentioned above. The latter rules are particularly attractive to use when the matrix A is so large that the dominant computational work for the evaluation of the quadrature rule is the calculation of the ℓ matrix-block-vector products required to determine the decomposition (1.3) by the symmetric block Lanczos algorithm. Section 3 presents analogous quadrature rules for the approximation of expressions (1.2) with a nonsymmetric matrix A. Block Gauss quadrature rules for this approximation problem have been described in [17]. These rules are determined from decompositions computed by the nonsymmetric block Lanczos algorithm. We present new block quadrature rules that are exact for polynomials of higher degree than the associated block Gauss quadrature rule. Sections 4 and 5 present computed examples. We illustrate in the former section that the difference between the new block rules and associated block Gauss quadrature rules can be used to determine estimates for the error in the latter; Section 5 describes applications to the analysis of large networks. In these applications the matrix A is an adjacency matrix that defines the network. We remark that instead of using block quadrature rules, the entries of the matrices (1.1) and (1.2) can be approximated by evaluating $k(k + 1)/2$ and k^2 quadrature rules

with block size one, respectively. Computed examples in Section 5 show the application of block quadrature rules to be significantly faster. This depends on the fact that on many modern computers, the evaluation of a matrix-vector product and of a matrix-block-vector product requires about the same amount of time when the block vector does not have many columns. This is discussed in, e.g., [19]. Section 6 contains concluding remarks.

The new block quadrature rules of this paper are block generalizations of the averaged Gauss quadrature rules proposed by Spalević [30, 31] for the integration of real-valued functions on a real interval. When the block size is one, the standard symmetric or nonsymmetric Lanczos algorithms can be used to determine the quadrature rules described in Sections 2 and 3. Quadrature rules for this situation have recently been discussed in [28]. Other approaches, based on extrapolation, to estimate functionals of the form (1.1) and (1.2) when the block size is one have recently been described by Brezinski et al. [7] and Fika et al. [18].

2. The symmetric problem $W^T f(A)W$

We discuss the approximation of expressions of the form (1.1) by block quadrature rules. The matrix $A \in \mathbb{R}^{m\times m}$ is assumed to be symmetric throughout this section and $W \in \mathbb{R}^{m\times k}$ has orthonormal columns with $1 \le k \ll m$.

To justify the use of quadrature rules, we first show that the expression (1.1) can be written as a Stieltjes-type integral. This was first observed by Golub and Meurant [20]. A more recent and detailed discussion can be found in [21]; see also [17].

Consider the spectral factorization

$$A = Q\Lambda Q^T, \tag{2.1}$$

where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $\Lambda = \mathrm{diag}[\lambda_1, \ldots, \lambda_m] \in \mathbb{R}^{m\times m}$. Thus, the λ_i are eigenvalues of A. Substituting (2.1) into (1.1) yields

$$W^T f(A)W = W^T Q f(\Lambda) Q^T W = \sum_{i=1}^{m} f(\lambda_i)\boldsymbol{\alpha}_i \boldsymbol{\alpha}_i^T = \int f(\lambda)\,\mathrm{d}\sigma(\lambda) =: \mathcal{I} f, \tag{2.2}$$

where $[\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_m] = W^T Q$ and σ is a piecewise matrix-valued distribution with jump $\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^T \in \mathbb{R}^{k\times k}$ at the eigenvalue λ_i of A for $i = 1, 2, \ldots, m$. For future reference, we define the matrix moments

$$M_j := \mathcal{I}(\lambda^j), \qquad j = 0, 1, 2, \ldots\ . \tag{2.3}$$

In particular, $M_0 = W^T W = I_k$.

Introduce the bilinear form

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

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

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

see, e.g., [11, 20, 21, 29]. These polynomials satisfy a three-term recurrence relation of the form

$$\begin{aligned} \lambda p_{j-1}(\lambda) &= p_j(\lambda)\Gamma_j + p_{j-1}(\lambda)\Omega_j + p_{j-2}(\lambda)\Gamma_{j-1}^T, \quad j = 1, 2, \ldots, \\ p_0(\lambda) &:= I_k, \quad p_{-1}(\lambda) := O_k, \quad \Gamma_0 := O_k, \end{aligned} \tag{2.4}$$

where the matrices $\Omega_j \in \mathbb{R}^{k\times k}$ are symmetric. The matrices $\Gamma_j \in \mathbb{R}^{k\times k}$ may be chosen to be upper triangular, but this is not necessary.

Define the matrix

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

Then the recursion relations (2.4) for the polynomials $p_0, p_1, \ldots, p_\ell$ can be expressed in the form

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

where J_ℓ is given by (1.4).

Introduce the spectral factorization

$$J_\ell = Y_\ell \Theta_\ell Y_\ell^T, \tag{2.5}$$

where

$$Y_\ell = [\boldsymbol{y}_1^{(\ell)}, \ldots, \boldsymbol{y}_{k\ell}^{(\ell)}] \in \mathbb{R}^{k\ell \times k\ell}, \quad Y_\ell^T Y_\ell = I_{k\ell}, \quad \Theta_\ell = \mathrm{diag}[\theta_1^{(\ell)}, \ldots, \theta_{k\ell}^{(\ell)}] \in \mathbb{R}^{k\ell \times k\ell}.$$

Substituting (2.5) into (1.5) yields

$$\mathcal{G}_\ell f = \sum_{i=1}^{k\ell} f(\theta_i^{(\ell)}) \boldsymbol{u}_i^{(\ell)} (\boldsymbol{u}_i^{(\ell)})^T, \tag{2.6}$$

where the vector $\boldsymbol{u}_i^{(\ell)} \in \mathbb{R}^k$ consists of the first k elements of the eigenvector $\boldsymbol{y}_i^{(\ell)}$. Whether it is most advantageous to compute $\mathcal{G}_\ell f$ by evaluating (2.6) or by calculating (1.5) without computing the spectral factorization of J_ℓ depends on the function f. For instance, the squaring and scaling algorithm described by Higham [25] is a convenient way to compute $f(J_\ell)$ when f is the exponential function and does not require the spectral factorization (2.5).

It is well known that $\mathcal{G}_\ell$ is an ℓ-block Gauss rule associated with the measure $\mathrm{d}\sigma$, i.e.,

$$\mathcal{G}_\ell f = \mathcal{I} f \qquad \forall f \in \mathbb{P}_{2\ell-1},$$

which is equivalent to (1.6). Proofs can be found in [11, 20, 21, 29]. A particularly simple proof is provided in [17, Section 5].

We are now in a position to describe new block quadrature rules for the approximation of (1.1). Introduce for $1 \le r < \ell$ the reverse symmetric block tridiagonal matrices

$$\widetilde{J}_{\ell-r,r} = \begin{bmatrix} \Omega_{\ell-1} & \Gamma_{\ell-2}^T & & & O \\ \Gamma_{\ell-2} & \Omega_{\ell-2} & \Gamma_{\ell-3}^T & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{r+1} & \Omega_{r+1} & \Gamma_r^T \\ O & & & \Gamma_r & \Omega_r \end{bmatrix} \in \mathbb{R}^{k(\ell-r)\times k(\ell-r)} \tag{2.7}$$

as well as the concatenated symmetric block tridiagonal matrices

$$\widehat{J}_{2\ell-r,r} = \begin{bmatrix} J_{\ell-1} & \Gamma_{\ell-1}^T E_{\ell-1} & O \\ \Gamma_{\ell-1} E_{\ell-1}^T & \Omega_\ell & \Gamma_\ell^T E_1^T \\ O & \Gamma_\ell E_1 & \widetilde{J}_{\ell-r,r} \end{bmatrix} \in \mathbb{R}^{k(2\ell-r)\times k(2\ell-r)}. \tag{2.8}$$

The latter matrices induce our new block quadrature rules

$$\widehat{\mathcal{G}}_{2\ell-r,r} f = E_1^T f(\widehat{J}_{2\ell-r,r}) E_1, \qquad 1 \le r < \ell. \tag{2.9}$$

Similarly as the block Gauss rule (1.5), these rules can be evaluated when ℓ steps of the block Lanczos method applied to A with initial block W have been carried out; see below. The following theorem shows some properties.

Theorem 2.1. *The quadrature rules* (2.9) *are exact for all* $f \in \mathbb{P}_{2\ell}$. *If the measure* $\mathrm{d}\sigma$ *is such that all diagonal blocks* Ω_i *are equal, then the rules are exact for all* $f \in \mathbb{P}_{2\ell+1}$.

Proof. Fix $1 \le r < \ell$ and assume first that $\mathrm{d}\sigma$ is a general measure such that all matrix moments (2.3) exist. Our proof is based on the recursion formulas of the block Chebyshev algorithm. A modified block Chebyshev algorithm is described in, e.g., [9]; the block Chebyshev algorithm is a special case that uses moments (2.3) as input instead of modified moments. The latter algorithm determines recursion matrix coefficients Ω_j and Γ_j for the orthogonal matrix polynomials (2.4) from the moments (2.3). More precisely, the submatrices $\Omega_1, \ldots, \Omega_\ell$ and $\Gamma_1, \ldots, \Gamma_\ell$ of (2.8) are determined by the matrix moments $M_0, M_1, \ldots, M_{2\ell}$ in the order $\Omega_1, \Gamma_1, \Omega_2, \Gamma_2, \ldots, \Gamma_{\ell-1}, \Omega_\ell, \Gamma_\ell$. Thus, the block quadrature rule matches the first $2\ell + 1$ matrix moments. We conclude that the block quadrature rule (2.9) is exact for (at least) all matrix polynomials in $\mathbb{P}_{2\ell}$.

Now let the measure $\mathrm{d}\sigma$ be such that all diagonal blocks Ω_i are equal. This is, for instance, the case when all diagonal blocks vanish. Then the $(\ell+1)$st diagonal block entry of (2.8) can be thought of as having been determined by the moments $M_0, M_1, \ldots, M_{2\ell+1}$. If follows that the block quadrature rule (2.9) is exact for all $f \in \mathbb{P}_{2\ell+1}$. We remark that the requirement that all diagonal blocks be equal is sufficient for the quadrature rule (2.9) to be exact for all $f \in \mathbb{P}_{2\ell+1}$, but it is not necessary.

An alternative proof can be based on the recursions of the symmetric block Lanczos algorithm (Algorithm 1 below), in particular on how moment information is used to determine the decomposition (1.3).

Computed examples for block size $k = 1$ reported in [28] indicate that quadrature rules $\widehat{\mathcal{G}}_{2\ell-r,r} f$ with $r = 1$ generally yield higher accuracy than rules with $r > 1$.[1] Spalević [31] provides theoretical support for this observation under certain conditions. Computations reported in Sections 4 and 5 of this paper for block sizes $k > 1$ show that block methods may give as high accuracy when $r > 1$ as when $r = 1$; see, e.g., Figure 2. We remark that the computational effort decreases slightly when r is increased, but the reduction is negligible in comparison with the effort required to evaluate matrix-block-vector products with the matrix A when this matrix is large. Some timings are presented in Section 5.

Block quadrature rules that are different from the rules (2.9) and match the same matrix moments also can be derived. Consider, for instance, the block quadrature rule

$$\widehat{\mathcal{G}}'_{2\ell-r,r} f = E_1^T f(\widehat{J}'_{2\ell-r,r}) E_1, \qquad 1 \le r < \ell, \tag{2.10}$$

where

$$\widehat{J}'_{2\ell-r,r} = \begin{bmatrix} J_{\ell-1} & \Gamma_{\ell-1}^T E_{\ell-1} & O \\ \Gamma_{\ell-1} E_{\ell-1}^T & \Omega_\ell & \Gamma_\ell^T E_1^T \\ O & \Gamma_\ell E_1 & \widetilde{J}'_{\ell-r,r} \end{bmatrix} \in \mathbb{R}^{k(2\ell-r)\times k(2\ell-r)}$$

and the matrix

$$\widetilde{J}'_{\ell-r,r} = \begin{bmatrix} \Omega_{\ell-1} & \Gamma_{\ell-2} & & & O \\ \Gamma_{\ell-2}^T & \Omega_{\ell-2} & \Gamma_{\ell-3} & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{r+1}^T & \Omega_{r+1} & \Gamma_r \\ O & & & \Gamma_r^T & \Omega_r \end{bmatrix} \in \mathbb{R}^{k(\ell-r)\times k(\ell-r)} \tag{2.11}$$

is obtained by interchanging the sub- and super-diagonal blocks of (2.7).

Similarly as the block quadrature rule (2.9), the rule (2.10) can be evaluated after ℓ steps of the symmetric block Lanczos algorithm have been carried out. We have the following result.

Corollary 2.2. *The quadrature rules* (2.10) *are exact for all* $f \in \mathbb{P}_{2\ell}$. *If the measure* $\mathrm{d}\sigma$ *is such that all diagonal blocks* Ω_i *are equal, then the rules are exact for all* $f \in \mathbb{P}_{2\ell+1}$.

Proof. The result can be shown in the same way as Theorem 2.1.

Computed examples show the block quadrature rules (2.9) and (2.10) to yield essentially the same accuracy. We therefore in Sections 4 and 5 only report results for one of these rules.

The remainder $A[W_1, \ldots, W_\ell] - [W_1, \ldots, W_\ell] J_\ell$ in the block Lanczos decomposition (1.3) is of rank at most k. The following result shows a decomposition involving the block tridiagonal matrix $\widehat{J}'_{2\ell-1,1}$ in which the remainder is of rank at most $2k$. The decomposition uses the block permutation matrix

$$P_{\ell-1} = [E_{\ell-1}, \ldots, E_1] \in \mathbb{R}^{k(\ell-1)\times k(\ell-1)}. \tag{2.12}$$

Theorem 2.3. *Let* $U_\ell = [W_1, \ldots, W_\ell]$ *and* $\widehat{U}_{2\ell-1} = [U_\ell, U_{\ell-1} P_{\ell-1}] \in \mathbb{R}^{m\times k(2\ell-1)}$, *where* $P_{\ell-1}$ *is given by* (2.12). *Then* $A\widehat{U}_{2\ell-1} - \widehat{U}_{2\ell-1}\widehat{J}'_{2\ell-1,1}$ *is of rank at most* $2k$.

[1] The notation in this paper and in [28] differ. The case $r = 1$ in the present paper corresponds to $r = 0$ in [28].

PROOF. Define the matrix $\widetilde{U}_{\ell-1} = U_{\ell-1}P_{\ell-1}$ and note that $\widetilde{J}'_{\ell-1,1} = P_{\ell-1}J_{\ell-1}P_{\ell-1}$. We obtain from (1.3) that

$$\begin{aligned} A\widetilde{U}_{\ell-1} &= AU_{\ell-1}P_{\ell-1} = (U_{\ell-1}J_{\ell-1} + W_\ell\Gamma_{\ell-1}E_{\ell-1}^T)P_{\ell-1} \\ &= U_{\ell-1}P_{\ell-1}P_{\ell-1}J_{\ell-1}P_{\ell-1} + W_\ell\Gamma_{\ell-1}E_{\ell-1}^TP_{\ell-1} \\ &= \widetilde{U}_{\ell-1}\widetilde{J}'_{\ell-1,1} + W_\ell\Gamma_{\ell-1}E_1^T. \end{aligned}$$

It follows that

$$A[U_\ell, \widetilde{U}_{\ell-1}] = [U_\ell J_\ell, \widetilde{U}_{\ell-1}\widetilde{J}'_{\ell-1,1}] + [W_{\ell+1}\Gamma_\ell E_\ell^T, W_\ell\Gamma_{\ell-1}E_1^T].$$

Moreover,

$$[U_\ell, \widetilde{U}_{\ell-1}]\widehat{J}'_{2\ell-1,1} = [U_\ell J_\ell + W_{\ell-1}\Gamma_\ell E_\ell^T, \widetilde{U}_{\ell-1}\widetilde{J}'_{\ell-1,1} + W_\ell\Gamma_\ell^T E_1^T].$$

Therefore,

$$A[U_\ell, \widetilde{U}_{\ell-1}] - [W_{\ell+1}\Gamma_\ell E_\ell^T, W_\ell\Gamma_{\ell-1}E_1^T] = [U_\ell, \widetilde{U}_{\ell-1}]\widehat{J}'_{2\ell-1,1} - [W_{\ell-1}\Gamma_\ell E_\ell^T, W_\ell\Gamma_\ell^T E_1^T].$$

Rearranging the terms, we obtain

$$A\widehat{U}_{2\ell-1} = \widehat{U}_{2\ell-1}\widehat{J}'_{2\ell-1,1} + [(W_{\ell+1}\Gamma_\ell - W_{\ell-1}\Gamma_\ell)E_\ell^T, (W_\ell\Gamma_{\ell-1} - W_\ell\Gamma_\ell^T)E_1^T],$$

which shows the theorem.

We conclude this section with some comments on the computation of the matrix recursion coefficients in (2.4). They are computed with the symmetric block Lanczos algorithm described in, e.g., [20, 21]. Below we provide a modified Gram–Schmidt implementation. The initial block vector $W \in \mathbb{R}^{m\times k}$ is assumed to have orthonormal columns.

Algorithm 1 The symmetric block Lanczos algorithm.

Input: symmetric matrix $A \in \mathbb{R}^{m\times m}$, initial block vector $W \in \mathbb{R}^{m\times k}$,
number of steps ℓ.
$W_0 = O \in \mathbb{R}^{m\times k}$, $\Gamma_0 = O_k$, $W_1 = W$
for $j = 1$ **to** ℓ
  $T = AW_j - W_{j-1}\Gamma_{j-1}^T$
  $\Omega_j = W_j^T T$
  $R_j = T - W_j\Omega_j$
  $W_{j+1}\Gamma_j = R_j$
end for
Output: symmetric block Lanczos decomposition (1.3)

In the algorithm, the statement

$$W_{j+1}\Gamma_j = R_j \tag{2.13}$$

denotes the computation of a reduced QR factorization such that $W_{j+1} \in \mathbb{R}^{m\times k}$ has orthonormal columns and $\Gamma_j \in \mathbb{R}^{k\times k}$ is upper triangular. When the matrix A is large and ℓ is fairly small, which is the situation in most applications of interest, the dominant computational work in Algorithm 1 is the evaluation of the matrix-block-vector products AW_j for $j = 1, 2, \ldots, \ell$. The block Lanczos algorithm is said to break down at the jth step if R_j is (numerically) rank deficient. In this case, the computations can be continued by, during the computation of the QR factorization (2.13), replacing (numerically) linearly dependent columns of W_{j+1} by arbitrary columns of unit length that are orthogonal to the ranges of the matrices $W_1, \ldots, W_j$ and such that $W_{j+1}^T W_{j+1} = I_k$. The upper triangular matrix $\Gamma_j \in \mathbb{R}^{k\times k}$ so obtained is necessarily singular. We remark that the matrices Γ_j do not have to be triangular; it suffices that the blocks are made up of orthonormal columns, and that each block W_j is orthogonal to every other block. For ease of exposition, the handling of breakdown is not included in Algorithm 1.

3. The nonsymmetric problem $W^T f(A)V$

We extend the discussion of the previous section to the situation when the matrix $A \in \mathbb{R}^{m\times m}$ is nonsymmetric. The block vectors $V, W \in \mathbb{R}^{m\times k}$, with $1 \le k \ll m$, are assumed to satisfy $W^T V = I_k$. The case of general block vectors will be discussed in Section 5. We derive new quadrature rules for the approximation of expressions of the form (1.2). When expressing (1.2) as an integral, we assume the matrix A to have the spectral factorization

$$A = S\Lambda S^{-1}, \tag{3.1}$$

where $S \in \mathbb{C}^{m\times m}$ is nonsingular and $\Lambda = \mathrm{diag}[\lambda_1, \ldots, \lambda_m] \in \mathbb{C}^{m\times m}$. The eigenvalues λ_i are real or appear in complex conjugate pairs. Substituting (3.1) into (1.2) yields

$$W^T f(A)V = \widetilde{W} f(\Lambda)\widetilde{V}^H = \sum_{i=1}^{m} f(\lambda_i)\boldsymbol{\alpha}_i\boldsymbol{\beta}_i^H = \int f(\lambda)\,\mathrm{d}\sigma(\lambda) =: \mathcal{I} f, \tag{3.2}$$

where $\widetilde{W} = W^T S = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m] \in \mathbb{C}^{k\times m}$ and $\widetilde{V} = (S^{-1}V)^H = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m] \in \mathbb{C}^{k\times m}$. The superscript ^{H} denotes transposition and complex conjugation. The expression (3.2) is analogous to (2.2); it differs from the latter expression in that the measure $\mathrm{d}\sigma$ in (3.2) may have support in $\mathbb{C}\backslash\mathbb{R}$.

Introduce the bilinear form

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

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

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

These polynomials satisfy three-term recurrence relations

$$\begin{aligned} \lambda p_{j-1}(\lambda) &= p_j(\lambda)\Gamma_j + p_{j-1}(\lambda)\Omega_j + p_{j-2}(\lambda)\Delta_{j-1}^T, \\ \lambda q_{j-1}(\lambda) &= q_j(\lambda)\Delta_j + q_{j-1}(\lambda)\Omega_j^T + q_{j-2}(\lambda)\Gamma_{j-1}^T, \end{aligned} \qquad j = 1, 2, \ldots, \tag{3.3}$$

where

$$p_0(\lambda) := I_k, \quad q_0(\lambda) := I_k, \quad p_{-1}(\lambda) := O_k, \quad q_{-1}(\lambda) := O_k, \quad \Delta_0 := O_k, \quad \Gamma_0 := O_k.$$

The matrix recursion coefficients Γ_j, Ω_j, and Δ_j are real $k \times k$ matrices with Γ_j and Δ_j upper triangular. For now we assume that all required matrices Γ_j, Ω_j, and Δ_j exist. We will comment on the situation when this is not the case below.

Let

$$\begin{aligned} P_\ell(\lambda) &:= [p_0(\lambda), \ldots, p_{\ell-1}(\lambda)] \in \mathbb{R}^{k\times k\ell}, \\ Q_\ell(\lambda) &:= [q_0(\lambda), \ldots, q_{\ell-1}(\lambda)] \in \mathbb{R}^{k\times k\ell}. \end{aligned}$$

The recursion relations for the polynomials $p_1, p_1, \ldots, p_\ell$ and $q_0, q_1, \ldots, q_\ell$ can be expressed compactly as

$$\begin{aligned} \lambda P_\ell(\lambda) &= P_\ell(\lambda)J_\ell + p_\ell(\lambda)\Gamma_\ell E_\ell^T, \\ \lambda Q_\ell(\lambda) &= Q_\ell(\lambda)J_\ell^T + q_\ell(\lambda)\Delta_\ell E_\ell^T, \end{aligned}$$

where

$$J_\ell = \begin{bmatrix} \Omega_1 & \Delta_1^T & & & O \\ \Gamma_1 & \Omega_2 & \Delta_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{\ell-2} & \Omega_{\ell-1} & \Delta_{\ell-1}^T \\ O & & & \Gamma_{\ell-1} & \Omega_\ell \end{bmatrix} \in \mathbb{R}^{k\ell\times k\ell} \tag{3.4}$$

is a block tridiagonal matrix. The block entries can be determined by carrying out ℓ steps of the nonsymmetric block Lanczos method; see below.

Assume that the matrix J_ℓ has the spectral factorization $J_\ell = Y_\ell \Theta_\ell Y_\ell^{-1}$, where

$$Y_\ell = [\boldsymbol{y}_1^{(\ell)}, \ldots, \boldsymbol{y}_{k\ell}^{(\ell)}] \in \mathbb{C}^{k\ell \times k\ell}, \qquad \Theta_\ell = \mathrm{diag}[\theta_1^{(\ell)}, \ldots, \theta_{k\ell}^{(\ell)}] \in \mathbb{C}^{k\ell \times k\ell}.$$

Letting $Z_\ell = [\boldsymbol{z}_1^{(\ell)}, \ldots, \boldsymbol{z}_{k\ell}^{(\ell)}] := Y_\ell^{-H}$, we obtain

$$J_\ell = Y_\ell \Theta_\ell Z_\ell^H, \qquad J_\ell^T Z_\ell = Z_\ell \bar{\Theta}_\ell, \tag{3.5}$$

where the bar denotes complex conjugation. Since the matrices A, V, W have real entries only, so does J_ℓ and, therefore, $J^T = J^H$.

Introduce the expression

$$\mathcal{G}_\ell f = E_1^T f(J_\ell) E_1. \tag{3.6}$$

It is shown in [17, Section 5] that $\mathcal{G}_\ell$ satisfies

$$\mathcal{G}_\ell f = \mathcal{I} f \qquad \forall f \in \mathbb{P}_{2\ell-1}.$$

We therefore refer to (3.6) as an ℓ-block Gauss quadrature rule. Substituting (3.5) into (3.6) yields

$$\mathcal{G}_\ell f := \sum_{i=1}^{k\ell} f(\theta_i^{(\ell)}) \boldsymbol{u}_i^{(\ell)} (\boldsymbol{v}_i^{(\ell)})^H, \tag{3.7}$$

where each vector $\boldsymbol{u}_i^{(\ell)} \in \mathbb{C}^k$ consists of the first k elements of the (right) eigenvector $\boldsymbol{y}_i^{(\ell)}$ of J_ℓ, and each vector $\boldsymbol{v}_i^{(\ell)} \in \mathbb{C}^k$ is made up of the first k elements of the (right) eigenvector $\boldsymbol{z}_i^{(\ell)}$ of J_ℓ^T. Whether it is better to compute $\mathcal{G}_\ell f$ by using the representation (3.7) or by evaluating (3.6) without determining the spectral factorization of J_ℓ depends both on the function f and on the accuracy with which the spectral factorization (3.5) can be calculated.

We can define new block Gauss quadrature rules analogously as in Section 2. Introduce for $1 \le r < \ell$ the reverse matrices

$$\widetilde{J}_{\ell-r,r} = \begin{bmatrix} \Omega_{\ell-1} & \Delta_{\ell-2}^T & & & O \\ \Gamma_{\ell-2} & \Omega_{\ell-2} & \Delta_{\ell-3}^T & & \\ & \ddots & \ddots & \ddots & \\ & & \Gamma_{r+1} & \Omega_{r+1} & \Delta_r^T \\ O & & & \Gamma_r & \Omega_r \end{bmatrix} \in \mathbb{R}^{k(\ell-r)\times k(\ell-r)} \tag{3.8}$$

and the concatenated block tridiagonal matrices

$$\widehat{J}_{2\ell-r,r} = \begin{bmatrix} J_{\ell-1} & \Delta_{\ell-1}^T E_{\ell-1} & O \\ \Gamma_{\ell-1} E_{\ell-1}^T & \Omega_\ell & \Delta_\ell^T E_1^T \\ O & \Gamma_\ell E_1 & \widetilde{J}_{\ell-1,r} \end{bmatrix} \in \mathbb{R}^{k(2\ell-r)\times k(2\ell-r)}. \tag{3.9}$$

The matrices (3.8) and (3.9) are analogues of the matrices (2.7) and (2.8) of Section 2. We use the matrices (3.9) to define the block quadrature rules

$$\widehat{\mathcal{G}}_{2\ell-r,r} f = E_1^T f(\widehat{J}_{2\ell-r,r}) E_1, \qquad 1 \le r < \ell, \tag{3.10}$$

which generalize the rules (1.5) to measures $\mathrm{d}\sigma$ of the type (3.2).

Theorem 3.1. *The block quadrature rules* (3.10) *are exact for all* $f \in \mathbb{P}_{2\ell}$. *If the measure* $\mathrm{d}\sigma$ *defined by (3.2) is such that all diagonal blocks* Ω_i *are equal, then the quadrature rules* (3.10) *are exact for all* $f \in \mathbb{P}_{2\ell+1}$.

Proof. We can show the result similarly as Theorem 2.1 by replacing the symmetric block Chebyshev algorithm by the nonsymmetric block Chebyshev algorithm; see, e.g., [8] for a description of the nonsymmetric modified Chebyshev algorithm for block size $k = 1$. Generalization to larger block size is fairly straightforward. Alternatively, one can base a proof on the recursions of the nonsymmetric block Lanczos method (Algorithm 2 below), in particular on how moment information is used by the method.

The matrix recursion coefficients in (3.3) can be determined by the nonsymmetric Lanczos method. The implementation described by Algorithm 2 is proposed by Bai et al. [3]. The algorithm determines the block matrices required by the block Gauss rule (3.6). These block matrices also are required by the block quadrature rules (3.10). The matrices $V, W \in \mathbb{R}^{m\times k}$ used to initialize the algorithm are assumed to satisfy $V^T W = I_k$.

Algorithm 2 The nonsymmetric block Lanczos algorithm.

Input: matrix $A \in \mathbb{R}^{m\times m}$, initial block vectors $W, V \in \mathbb{R}^{m\times k}$,
number of steps ℓ.
$V_0\Delta_0^T = W_0\Gamma_0^T = O \in \mathbb{R}^{m\times k}$, $V_1 = V/\|V\|$, $W_1 = W/\|W\|$
for $j = 1$ **to** ℓ
  $T = AV_j - V_{j-1}\Delta_{j-1}^T$
  $\Omega_j = W_j^T T$
  $R_j = T - V_j\Omega_j$
  $S_j = A^T W_j - W_j\Omega_j^T - W_{j-1}\Gamma_{j-1}^T$
  $Q_R R_R = R_j$, $Q_S R_S = S_j$
  $U\Sigma Z^T = Q_S^T Q_R$
  $V_{j+1} = Q_R Z\Sigma^{-\frac{1}{2}}$, $W_{j+1} = Q_S U\Sigma^{-\frac{1}{2}}$
  $\Gamma_j = \Sigma^{\frac{1}{2}} Z^T R_R$, $\Delta_j = \Sigma^{\frac{1}{2}} U^T R_S$
end for
Output: nonsymmetric block Lanczos decomposition (3.11)

The statements $Q_R R_R = R_j$ and $Q_S R_S = S_j$ in Algorithm 2 denote the computation of reduced QR factorizations, where $Q_R, Q_S \in \mathbb{R}^{m\times k}$ have orthonormal columns and $R_R, R_S \in \mathbb{R}^{k\times k}$ are upper triangular. Further, $U\Sigma Z^T = Q_S^T Q_R$ denotes the evaluation of a singular value decomposition of the matrix in the right-hand side. Algorithm 2 determines the decompositions

$$\begin{aligned} A[V_1,\ldots,V_\ell] &= [V_1,\ldots,V_\ell]J_\ell + V_{\ell+1}\Gamma_\ell E_\ell^T, \\ A^T[W_1,\ldots,W_\ell] &= [W_1,\ldots,W_\ell]J_\ell^T + W_{\ell+1}\Delta_\ell E_\ell^T, \end{aligned} \tag{3.11}$$

with the matrix J_ℓ given by (3.4).

The description of Algorithm 2 assumes that the matrix products $S_j^T R_j$, $1 \le j \le \ell$, are (numerically) nonsingular. The algorithm is said to break down if one of these products is (numerically) singular. The problem of breakdown is more complicated for Algorithm 2 than for Algorithm 1. A breakdown at step j of Algorithm 2 is said to be *serious* if $S_j^T R_j$ is singular, but both the matrices S_j and R_j are of full rank. Bai et al. [3] provide a thorough discussion on breakdown and show that serious breakdown can be circumvented by restarting Algorithm 2 after introducing appropriate additional vectors in the initial block vectors W_1 and V_1, and thereby increasing the block size.

Similarly as in Section 2, we may modify the trailing principal $k(\ell-r)\times k(\ell-r)$ submatrix of the matrix (3.9) to obtain a new block quadrature rule for which an analogue of Theorem 3.1 holds. In particular, interchanging the sub- and super-diagonal block entries of the matrices (3.8) gives

$$\widetilde{J}'_{\ell-r,r} = \begin{bmatrix} \Omega_{\ell-1} & \Gamma_{\ell-2} & & & O \\ \Delta_{\ell-2}^T & \Omega_{\ell-2} & \Gamma_{\ell-3} & & \\ & \ddots & \ddots & \ddots & \\ & & \Delta_{r+1}^T & \Omega_{r+1} & \Gamma_r \\ O & & & \Delta_r^T & \Omega_r \end{bmatrix} \in \mathbb{R}^{k(\ell-r)\times k(\ell-r)}, \qquad 1 \le r < \ell,$$

and the associated concatenated matrices

$$\widehat{J}'_{2\ell-r,r} = \begin{bmatrix} J_{\ell-1} & \Delta_{\ell-1}^T E_{\ell-1} & O \\ \Gamma_{\ell-1}E_{\ell-1}^T & \Omega_\ell & \Delta_\ell^T E_1^T \\ O & \Gamma_\ell E_1 & \widetilde{J}'_{\ell-r,r} \end{bmatrix} \in \mathbb{R}^{k(2\ell-r)\times k(2\ell-r)}, \qquad 1 \le r < \ell,$$

which define the block quadrature rules

$$\widehat{\mathcal{G}}'_{2\ell-r,r} f = E_1^T f(\widehat{J}'_{2\ell-r,r}) E_1, \qquad 1 \le r < \ell. \tag{3.12}$$

The following result can be shown in the same way as Theorem 3.1.

Corollary 3.2. *The quadrature rules* (3.12) *are exact for all* $f \in \mathbb{P}_{2\ell}$. *If the measure* $\mathrm{d}\sigma$, *defined by (3.2), is such that all diagonal blocks* Ω_i *are equal, then the rules are exact for all* $f \in \mathbb{P}_{2\ell+1}$.

Let $\check{V}_\ell = [V_1, \ldots, V_\ell]$ and $\check{W}_\ell = [W_1, \ldots, W_\ell]$. The nonsymmetric block Lanczos decompositions (3.11) are such that the rank of $A\check{V}_\ell - \check{V}_\ell J_\ell$ and $A^T\check{W}_\ell - \check{W}_\ell J_\ell^T$ is at most k. The following result is analogous to Theorem 2.3 and shows that decompositions involving the block tridiagonal matrix $\widehat{J}'_{2\ell-1,1}$ have remainders of rank at most $2k$.

Theorem 3.3. *Let the block permutation matrix* P_ℓ *be defined by* (2.12) *and introduce the matrices*

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

Then both the matrices $A\widehat{V}_{2\ell-1} - \widehat{V}_{2\ell-1}\widehat{J}'_{2\ell-1,1}$ *and* $A^T\widehat{W}_{2\ell-1} - \widehat{W}_{2\ell-1}\widehat{J}'^T_{2\ell-1,1}$ *have rank at most* $2k$.

Proof. The result can be shown similarly as Theorem 2.3. We therefore omit the details.

4. Numerical examples

This and the following sections illustrate the performance of the block quadrature rules of this paper. Thus, we approximate expressions of the form

$$F(A) := W^T f(A) V, \tag{4.1}$$

where $A \in \mathbb{R}^{m\times m}$ is a large symmetric or nonsymmetric matrix and $W, V \in \mathbb{R}^{m\times k}$, $1 \le k \ll m$, are block vectors with $W^TV = I_k$. The examples of this section illustrate the reduction of the quadrature error with increasing number of "block nodes", ℓ, of the quadrature rules, as well as the possibility to estimate the error in the block Gauss rule (1.5) by computing the difference between this rule and the rules (2.9) or (3.10). All computations in this and the following section were carried out in MATLAB R2014a with about 15 significant decimal digits on an Intel Core i7 computer running Linux.

In the first examples of this section, we set $V = W$; in later examples $V \neq W$.

Table 1: Example 4.1: Errors in approximations of $F(A) = W^T \exp(A)W$ with A symmetric indefinite.

	$\ell = 3$	$\ell = 5$	$\ell = 7$
$R_\infty(\mathcal{G}_\ell f, F(A))$	3.02×10^{-4}	4.36×10^{-9}	5.34×10^{-14}
$R_\infty(\widehat{\mathcal{G}}_{2\ell-1,1} f, F(A))$	2.18×10^{-5}	6.88×10^{-10}	3.06×10^{-15}
$R_\infty(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, F(A))$	1.67×10^{-5}	6.94×10^{-10}	3.06×10^{-15}
$R_2(\mathcal{G}_\ell f, F(A))$	1.81×10^{-4}	4.17×10^{-9}	2.01×10^{-14}
$R_2(\widehat{\mathcal{G}}_{2\ell-1,1} f, F(A))$	1.21×10^{-5}	9.24×10^{-11}	8.62×10^{-16}
$R_2(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, F(A))$	1.60×10^{-5}	1.20×10^{-10}	5.37×10^{-16}

Example 4.1. Let $A \in \mathbb{R}^{100\times100}$ be a symmetric indefinite matrix with 50 equidistant eigenvalues in the interval $[-2, -1]$ and 50 equidistant eigenvalues in the interval $[\frac{1}{2}, 1]$. The eigenvector matrix is an orthogonal random matrix. The function f in (4.1) is the exponential, i.e., $f(A) = \exp(A)$. We let $k = 2$ and $W = V = E_1 \in \mathbb{R}^{100,2}$. The relative error in the computed approximations of (4.1) is measured with the spectral norm as well as entrywise. Thus, we compute the difference

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

Table 2: Example 4.1: Error estimates for computed approximations of $F(A) = W^T \exp(A)W$ with A symmetric indefinite.

	$\ell = 3$	$\ell = 5$	$\ell = 7$
$R_\infty(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	3.23×10^{-4}	4.70×10^{-9}	5.60×10^{-14}
$R_\infty(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	6.19×10^{-6}	2.93×10^{-11}	4.38×10^{-16}
$R_2(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	1.71×10^{-4}	4.11×10^{-9}	2.08×10^{-14}
$R_2(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	4.32×10^{-6}	2.77×10^{-11}	3.39×10^{-16}

where $\|\cdot\|$ denotes the spectral norm, as well as

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

where $X_{i,j}$ denotes the $\{i, j\}$th entry of the matrix X. The latter error measure may not be meaningful if an entry $[F(A)]_{i,j}$ vanishes. Thus, we measure the error in the approximation (1.5) using $R_2(\mathcal{G}_\ell f, F(A))$ and $R_\infty(\mathcal{G}_\ell f, F(A))$. The exact solution is

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

We remark that in all examples of this section, the exact solution is determined by evaluating $f(A)$ by using matrix functions that are available in MATLAB. For instance for the present example, we use the MATLAB function `expm`.

Table 1 displays the errors for several quadrature rules for the present example, and shows the elementwise errors to be roughly of the same size as the spectral norm errors for all quadrature rules. Moreover, the errors are seen to decrease rapidly as ℓ increases. The errors reported for $\ell = 7$ may be significantly affected by round-off errors introduced during the computations.

Table 2 shows error estimates obtained by evaluating the difference of approximations of $F(A)$ determined with different quadrature rules. The estimate $R_\infty(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$ is seen to give a fairly accurate approximation of $R_\infty(\mathcal{G}_\ell f, F(A))$, and $R_2(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$ can be seen to be close to $R_2(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$. Thus, Table 2 suggests that the difference $\mathcal{G}_\ell f - \widehat{\mathcal{G}}_{2\ell-1,1} f$ furnishes a useful estimate of the error in $\mathcal{G}_\ell f$. □

Table 3: Example 4.2: Errors in approximations of $F(A) = W^T A^{-1} W$ with A symmetric positive definite.

	$\ell = 15$	$\ell = 20$	$\ell = 25$	$\ell = 30$
$R_\infty(\mathcal{G}_\ell f, F(A))$	9.17×10^{-5}	2.60×10^{-7}	1.83×10^{-10}	1.61×10^{-14}
$R_\infty(\widehat{\mathcal{G}}_{2\ell-1,1} f, F(A))$	3.41×10^{-5}	6.21×10^{-8}	3.87×10^{-11}	1.73×10^{-15}
$R_\infty(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, F(A))$	3.58×10^{-5}	8.97×10^{-8}	2.21×10^{-11}	1.19×10^{-15}
$R_2(\mathcal{G}_\ell f, F(A))$	5.56×10^{-5}	1.58×10^{-7}	1.11×10^{-10}	9.61×10^{-15}
$R_2(\widehat{\mathcal{G}}_{2\ell-1,1} f, F(A))$	2.07×10^{-5}	3.77×10^{-8}	2.35×10^{-11}	1.06×10^{-15}
$R_2(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, F(A))$	2.17×10^{-5}	5.45×10^{-8}	1.34×10^{-11}	7.05×10^{-16}

Table 4: Example 4.2: Error estimates for computed approximations of $F(A) = W^T A^{-1} W$ with A symmetric positive definite.

	$\ell = 15$	$\ell = 20$	$\ell = 25$	$\ell = 30$
$R_\infty(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	1.26×10^{-4}	1.98×10^{-7}	2.22×10^{-10}	1.77×10^{-14}
$R_\infty(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	6.99×10^{-5}	2.76×10^{-8}	6.07×10^{-11}	2.92×10^{-15}
$R_2(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	7.63×10^{-5}	1.21×10^{-7}	1.35×10^{-10}	1.07×10^{-14}
$R_2(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	4.24×10^{-5}	1.68×10^{-8}	3.69×10^{-11}	1.75×10^{-15}

Example 4.2. This example is similar to an example in [20]. Let the symmetric positive definite matrix $A \in \mathbb{R}^{m\times m}$ be determined by the standard 5-point finite difference discretization of the Laplace operator on the unit square with n mesh points along each coordinate direction. Then $m = n^2$ and

$$A = \begin{bmatrix} T_n & -I_n & & & O \\ -I_n & T_n & -I_n & & \\ & \ddots & \ddots & \ddots & \\ & & -I_n & T_n & -I_n \\ O & & & -I_n & T_n \end{bmatrix}$$

is block tridiagonal with tridiagonal diagonal blocks

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

We set $n = 10$, and let $V = W = E_1 \in \mathbb{R}^{100\times 2}$ and $f(t) = t^{-1}$ in (4.1). Thus, $F(A)$ is the leading principal 2×2 submatrix of A^{-1}. We have

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

Table 3 shows the errors in computed approximations of $F(A)$ using several quadrature rules. Error estimates analogous to those reported in Table 2 are displayed in Table 4. Similarly as in Table 2, the estimate $R_\infty(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$ is a quite accurate approximation of $R_\infty(\mathcal{G}_\ell f, F(A))$, and $R_2(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$ can be seen to be fairly close to $R_2(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$. □

The following two examples are concerned with functions of real nonsymmetric matrices.

Table 5: Example 4.3: Errors in approximations of $F(A) = W^T A^{-1} V$ with A nonsymmetric.

	$\ell = 3$	$\ell = 4$
$R_\infty(\mathcal{G}_\ell f, F(A))$	1.93×10^{-7}	4.42×10^{-13}
$R_\infty(\widehat{\mathcal{G}}_{2\ell-1,1} f, F(A))$	2.53×10^{-8}	1.11×10^{-13}
$R_\infty(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, F(A))$	2.52×10^{-8}	1.12×10^{-13}
$R_2(\mathcal{G}_\ell f, F(A))$	1.48×10^{-10}	7.61×10^{-16}
$R_2(\widehat{\mathcal{G}}_{2\ell-1,1} f, F(A))$	1.98×10^{-11}	6.07×10^{-16}
$R_2(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, F(A))$	1.97×10^{-11}	6.02×10^{-16}

Table 6: Example 4.3: Error estimates for computed approximations of $F(A) = W^T A^{-1} V$ with A nonsymmetric.

	$\ell = 3$	$\ell = 4$
$R_\infty(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	2.19×10^{-7}	4.10×10^{-13}
$R_2(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	1.67×10^{-10}	5.49×10^{-16}

Example 4.3. We would like to determine an approximation of the function (4.1), where $f(t) = t^{-1}$ and $A \in \mathbb{R}^{300\times300}$ is a nonsymmetric matrix with uniformly distributed nonnegative random entries. The matrix is generated with the MATLAB command $A = \texttt{rand}(300)/100$. Matrices so obtained typically have one large real eigenvalue and many eigenvalues close to the origin, but none at the origin. The block vectors $W, V \in \mathbb{R}^{300\times2}$ are given by

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

where $e_j = [0, \ldots, 0, 1, 0, \ldots, 0]^T \in \mathbb{R}^{300}$ denotes the jth axis vector. Thus, $V^T W = I_2$. The value of (4.1) is

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

and, of course, depends on the random matrix A. The quadrature errors obtained when approximating $F(A)$ are reported in Table 5 and error estimates are shown in Table 6. The latter are accurate approximations of the errors in $\mathcal{G}_\ell f$. □

Table 7: Example 4.4: Errors in approximations of $F(A) = W^T \exp(A)V$ with A nonsymmetric.

	$\ell = 3$	$\ell = 4$
$R_\infty(\mathcal{G}_\ell f, F(A))$	8.50×10^{-7}	1.33×10^{-12}
$R_\infty(\widehat{\mathcal{G}}_{2\ell-1,1} f, F(A))$	8.89×10^{-8}	6.11×10^{-14}
$R_\infty(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, F(A))$	8.90×10^{-8}	4.93×10^{-14}
$R_2(\mathcal{G}_\ell f, F(A))$	1.40×10^{-8}	2.64×10^{-14}
$R_2(\widehat{\mathcal{G}}_{2\ell-1,1} f, F(A))$	1.50×10^{-9}	1.48×10^{-15}
$R_2(\widehat{\mathcal{G}}_{\ell+1,\ell-1} f, F(A))$	1.50×10^{-9}	1.23×10^{-15}

Table 8: Example 4.4: Error estimates for computed approximations of $F(A) := W^T \exp(A)V$ with A nonsymmetric.

	$\ell = 3$	$\ell = 4$
$R_\infty(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	9.39×10^{-7}	1.26×10^{-12}
$R_2(\mathcal{G}_\ell f, \widehat{\mathcal{G}}_{2\ell-1,1} f)$	1.55×10^{-8}	2.50×10^{-14}

Example 4.4. This example differs from the previous one only in that the inverse is replaced by the exponential function. Thus, $F(A) = W^T \exp(A)V$, and the matrices A, W, and V are the same as in Example 4.3. We have

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

Quadrature errors and errors estimates are reported in Tables 7 and 8. □

5. Applications to network theory

This section presents computed examples in which block quadrature rules are applied to the analysis of large networks. The quadrature rules of this paper are compared to the use of pairs of block Gauss and anti-Gauss rules described in [17].

A network is identified by a graph $G = \{\mathcal{V}, \mathcal{E}\}$ that consists of a set of vertices $\mathcal{V}$, also referred to as nodes, and a set of edges $\mathcal{E}$; the edges connect the nodes. The graph G is assumed to be unweighted with m nodes, and have no loops of length one or multiple edges. We consider both undirected graphs, in which travel can occur in both directions along each edge, and directed graphs, in which some or all edges are "one way streets." We are concerned with the analysis of large and sparse graphs, i.e., graphs that have many nodes m and much fewer than $O(m^2)$ edges between the nodes. This kind of graphs arise in numerous scientific and industrial applications, including genetics, epidemiology, energy distribution, and telecommunication; see, e.g., [12, 14, 27]. The notions of walks and paths in a graph are important. A walk is a sequence of vertices $v_1, v_2, \ldots, v_k$ such that there is an edge from vertex v_i to vertex v_{i+1} for $i = 1, 2, \ldots, k-1$. Vertices and edges may be repeated. A path is a walk with all vertices distinct.

It is common to represent a graph by its associated adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ with entry $a_{ij} = 1$ if there is an edge connecting node i to node j, and $a_{ij} = 0$ otherwise. The adjacency matrix is symmetric if and only if the

graph is undirected. The entry $a_{ij}^{(\ell)}$ of the matrix A^ℓ is equal to the number of walks of length ℓ starting at node i and ending at node j; see, e.g., Estrada [12] or Newman [27] for detailed discussions on graphs.

Estrada and his collaborators [12, 13, 14, 15] proposed to quantify the importance of the nodes in a network by using matrix functions of the form

$$f(A) = \sum_{\ell=0}^{\infty} c_\ell A^\ell, \tag{5.1}$$

where the nonnegative coefficients c_ℓ are assumed to decay fast enough in magnitude to make the series convergent. The decay of the coefficients signifies that short walks are more important than long ones. For instance, when all walks of length up to k are equally important and no walk of length longer than k is of interest, then one should choose $c_\ell = 1$ for $1 \leq \ell \leq k$, and $c_\ell = 0$ for $\ell > k$.

Particular choices of coefficients c_ℓ that have been considered in the literature include $c_\ell = 1/\ell!$, which leads to the matrix exponential $f(A) = \exp(A)$, and $c_\ell = \mu^\ell$, where $0 < \mu < 1/\rho(A)$ and $\rho(A)$ is the spectral radius of A. The latter choice of coefficients corresponds to the resolvent $f(A) = (I - \mu A)^{-1}$.

The diagonal entry $[f(A)]_{ii}$ is referred to as the *f-subgraph centrality* of node i, and its value is a measure of the connectivity of node i to the rest of the graph and, therefore, a measure of the importance of the node. Node i is considered more important the larger $[f(A)]_{ii}$ is. The off-diagonal entry $[f(A)]_{ij}$, $i \neq j$, is known as the *f-subgraph communicability* between nodes i and j, and quantifies the ease of traveling between these two nodes. A large value indicates that traveling from node i to node j is easy; see [12, 13, 14, 15].

The use of matrix functions (5.1) to analyze networks was originally conceived for undirected networks, and subsequently has been applied to directed networks [5, 10] and to time-dependent networks [1, 23]. The problem of the evaluation of desired entries of matrix functions (1.1) and (1.2) for large adjacency matrices A has been addressed in [4, 16] for undirected networks and in [2, 5] for directed ones. In [17] block algorithms were developed for both cases.

In this section we compare the performance of the block quadrature rules of Sections 2 and 3, to block quadrature rules described in [17] when applied to estimate the f-subgraph centrality of a small subset of nodes in large real-world networks. We also compute the f-communicability between these nodes. Both undirected networks (Email, Autobahn, Yeast, Power, Internet, Collaboration, Facebook) and directed networks (Airlines, Celegans, Air500, Twitter, Wikipedia, Slashdot) are considered. Properties of these networks are described in Table 9. The networks are available on the internet; see [2, 17] for more details. An additional complex nonsymmetric matrix (Vfem), which arises from the discretization of a partial differential problem in electromagnetics [32], has been included to investigate a large example that is not from network theory.

Table 9: Properties of networks used in the computed examples. The matrix Vfem is not an adjacency matrix for a network, but stems from the discretization of a partial differential problem in electromagnetics.

Undirected networks			Directed networks		
matrix	nodes	edges	matrix	nodes	edges
Email	1133	10902	Airlines	235	2101
Autobahn	1168	2486	Celegans	306	2345
Yeast	2114	4480	Air500	500	24009
Power	4941	13188	Twitter	3656	188712
Internet	22963	96872	Wikipedia	49728	941425
Collab.	40421	351384	Slashdot	82168	948464
Facebook	63731	1634180	Vfem	93476	1434636

Laurie [26] introduced so-called anti-Gauss quadrature rules for the approximation of integrals

$$\mathcal{I} f = \int f(t) \mathrm{d}\sigma(t)$$

of a real-valued function and a nonnegative measure $\mathrm{d}\sigma$ on the real axis with infinitely many points of support. The $(\ell + 1)$-point anti-Gauss quadrature rule, $\mathcal{H}_{\ell+1}$, associated with the ℓ-point Gauss rule for this measure, $\mathcal{G}_\ell$, is

characterized by

$$(\mathcal{I} - \mathcal{H}_{\ell+1})p = -(\mathcal{I} - \mathcal{G}_\ell)p \qquad \forall p \in \mathbb{P}^{2\ell+1}.$$

Let $\{p_\ell\}_{\ell=0}^{\infty}$ be a sequence of orthonormal polynomials associated with the measure $\mathrm{d}\sigma$. Thus, p_ℓ is of degree ℓ. If the coefficients d_ℓ in the expansion

$$f(t) = \sum_{\ell=0}^{\infty} d_\ell p_\ell(t)$$

converge sufficiently rapidly to zero, then $\mathcal{G}_\ell f$ and $\mathcal{H}_{\ell+1} f$ bracket $\mathcal{I} f$. Thus, for this kind of "nice" integrands $\min\{\mathcal{G}_\ell f, \mathcal{H}_{\ell+1} f\}$ and $\max\{\mathcal{G}_\ell f, \mathcal{H}_{\ell+1} f\}$ provide upper and lower bounds for $\mathcal{I} f$. Laurie [26] also showed that the average

$$\frac{1}{2}(\mathcal{G}_\ell f + \mathcal{H}_{\ell+1} f) \tag{5.2}$$

is a quadrature rule that is exact for all $p \in \mathbb{P}^{2\ell+1}$.

Recently, block anti-Gauss rules were presented in [17] and it was shown that they have the same properties, suitably modified, as real-valued anti-Gauss rules. In particular, for sufficiently "nice" integrands, the ℓ-block Gauss rule and the associated $(\ell + 1)$-block anti-Gauss rule provide elementwise upper and lower bounds for the matrix-valued integral. Their computation for expressions of the forms (1.1) and (1.2) requires the execution of $\ell + 1$ steps of the symmetric or nonsymmetric block Lanczos process, respectively; see [17] for details.

Table 10: Computation time and relative errors obtained performing 7 iterations of the symmetric Lanczos method (E_7) using both a pair of Gauss and anti-Gauss quadrature formulas, and averaged Gauss quadrature rules (AGQ). The error is computed with respect to `expm` for the first four networks. For the last three networks we let the Gauss/anti-Gauss method iterate until the stopping tolerance 10^{-12} is reached.

	Gauss/anti-Gauss		AGQ ($r = 6$)		AGQ ($r = 1$)	
matrix	time	E_7	time	E_7	time	E_7
Email	1.18e-02	5.47e-06	1.04e-02	8.38e-07	1.65e-02	1.19e-07
Autobahn	7.70e-03	4.91e-12	9.62e-03	4.87e-11	1.79e-02	1.93e-11
Yeast	9.23e-03	3.22e-06	1.11e-02	5.12e-07	1.70e-02	1.56e-07
Power	1.27e-02	2.62e-07	1.51e-02	2.46e-08	2.09e-02	3.09e-10
Internet	4.07e-02	3.36e-03	4.56e-02	2.63e-04	5.09e-02	1.09e-04
Collab.	8.87e-02	5.29e-04	9.77e-02	1.05e-04	1.25e-01	4.26e-05
Facebook	2.56e-01	9.20e-03	2.70e-01	1.51e-05	2.74e-01	8.22e-04

Our first experiment concerns the undirected test networks listed, together with the number of nodes, m, and edges, in the first three columns of Table 10. We would like to compute accurate approximations of the expression (1.1) for $f(A) = \exp(A)$, where W consists of $k = 5$ columns of the $m \times m$ identity matrix corresponding to 5 selected nodes, in order to determine the f-subgraph centrality of these nodes and the f-subgraph communicability between them. To avoid the occurrence of breakdown in the first few iterations of the Lanczos method, we append to W an additional column of ones, thus obtaining a measure proportional to the *total subgraph communicability* for a node. This measure is discussed in detail by Benzi and Klymko [6]. It has also been described in [17] and elsewhere in the literature. The proper inclusion of an extra column with many nonvanishing entries reduces the likelihood of a breakdown. This approach to reduce breakdown is discussed by Bai et al. [3] for the nonsymmetric Lanczos method. We first focus on the symmetric Lanczos method and comment on the nonsymmetric method below. The reasons why breakdown of the symmetric Lanczos method is common when applied to network analysis is that both A and the first five columns of W, which are axis vectors, are very sparse. While these columns are orthonormal, the column of ones in $W \in \mathbb{R}^{m\times 6}$ results in that $W^T W \neq I_6$. To proceed, we compute the singular value decomposition $W = U_W \Sigma_W V_W^T$, where $U_W \in \mathbb{R}^{m\times 6}$ has orthonormal columns and $\Sigma_W \in \mathbb{R}^{6\times 6}$ is diagonal. Then

$$W^T f(A) W = V_W \Sigma_W [U_W^T f(A) U_W] \Sigma_W V_W^T \tag{5.3}$$

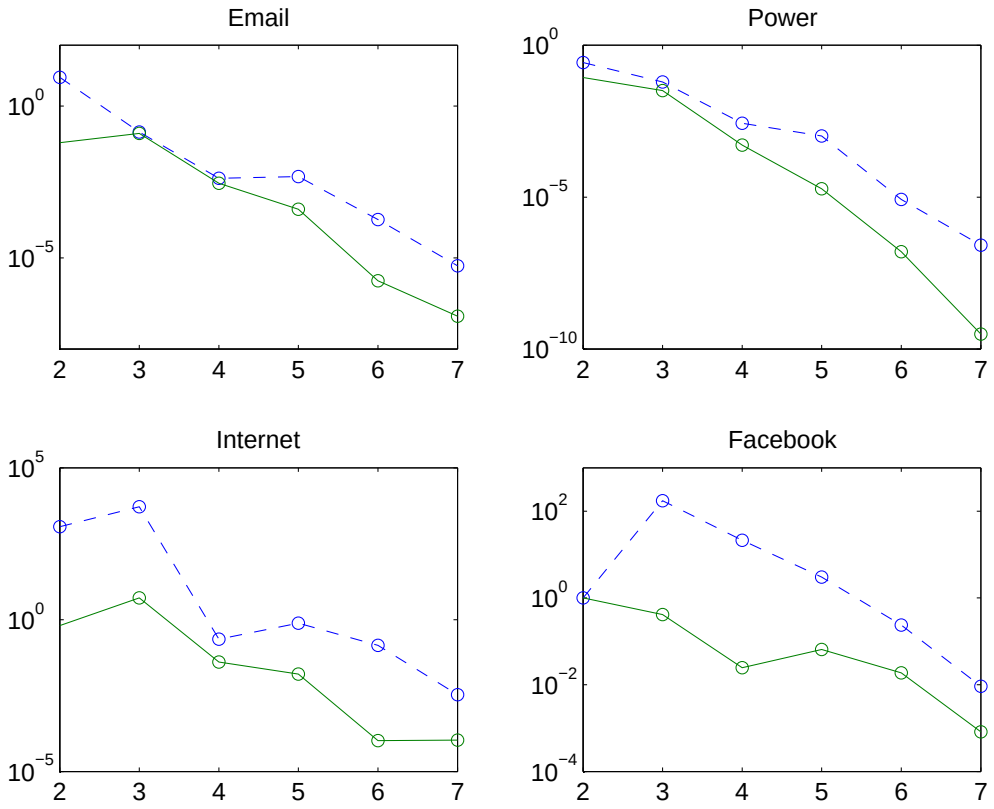

Figure 1: Relative errors versus number of iterations ℓ with the symmetric block Lanczos method for averaged Gauss quadrature $\widehat{\mathcal{G}}_{2\ell-1,1}f$ (continuous curve) and Gauss/anti-Gauss quadrature (5.2) (dashed curve) for $\ell = 2, 3, \ldots, 7$.

and we can compute approximations of the expression in brackets on the right-hand side by the methods of this paper. The symmetric Lanczos method is unlikely to break down during these computations because the matrix U_W has very few, if any, nonvanishing entries.

We carry out $\ell = 7$ steps with the symmetric block Lanczos algorithm and approximate the expression (1.1) by $\widehat{\mathcal{G}}_{2\ell-r,r}f$ for $r = 1$ and $r = \ell - 1$ (see (2.9)), and by the average of the Gauss rule $\mathcal{G}_{\ell-1}f$ and the anti-Gauss rule $\mathcal{H}_\ell f$; cf. (5.2). Table 10 reports the execution time (in seconds) and the relative error in the matrix infinity norm for quadrature rules; the columns labeled "AGQ" refer to the averaged Gauss quadrature rules of Section 2. The relative error in the computed approximation X_ℓ of the exact value $F(A)$, cf. (4.1), obtained after ℓ steps of the symmetric block Lanczos method is measured as in [17] by

$$E_\ell = \frac{\|X_\ell - F(A)\|_\infty}{\|F(A)\|_\infty}. \tag{5.4}$$

In the tables of this section, we use the notation $XeY = X \cdot 10^Y$. We consider the result produced by the MATLAB function `expm` to be exact and let $F(A) = W^T \texttt{expm}(A)W$ for networks with fewer than 5000 nodes. However, `expm` requires too much computer time to be useful for the 3 largest networks of Table 10. For these networks, we increase the number of block Lanczos steps ℓ until the componentwise difference between the block Gauss rule $\mathcal{G}_{\ell-1}f$ and the associated block anti-Gauss rule $\mathcal{H}_\ell f$ is at most 10^{-12}. This requires $\ell = 13$ block Lanczos steps for the Internet and Collaboration networks, and $\ell = 16$ steps for Facebook. We then form the average of these rules, cf. (5.2). The value so obtained is considered "exact" and used to compute the errors for the 3 largest networks reported in Table 10. The columns labeled "Gauss/anti-Gauss" display the computing times and the errors for the approximation (5.2) for $\ell = 7$. The table shows the averaged block Gauss rules $\widehat{\mathcal{G}}_{2\ell-r,r}f$ for $r \in \{1, \ell - 1\}$, defined by (2.9), to produce more accurate results and require essentially the same computing times as the rule (5.2). For all networks, except one, $\widehat{\mathcal{G}}_{2\ell-r,r}f$ can

be seen to furnish more accurate approximations for $r = 1$ than for $r = \ell - 1$. The computing times for $r = 1$ is slightly longer than for $r = \ell - 1$. The accuracy achieved with $\widehat{\mathcal{G}}_{2\ell-1,1} f$ is further illustrated in Figure 1, which displays the errors E_ℓ obtained with $\widehat{\mathcal{G}}_{2\ell-1,1} f$ and the rule (5.2) for each step ℓ of the symmetric block Lanczos method for four test networks.

For the same four networks, Figure 2 displays the error E_7 produced by the quadrature rules $\widehat{\mathcal{G}}_{2\ell-r,r} f$ when r ranges from 1 to 6, after 7 iterations of the symmetric block Lanczos method. For the first two networks, all values of $1 \leq r \leq 5$ give small errors of about the same size. In the Internet example, the errors are small and of about the same size for $1 \leq r \leq 3$. The errors for the other test networks, except for Facebook, behave similarly, i.e., letting r be a small integer larger than one gives about the same accuracy as $r = 1$. The Facebook network is the only example for which the error in the quadrature rule is larger for r close to unit than for a larger r-value.

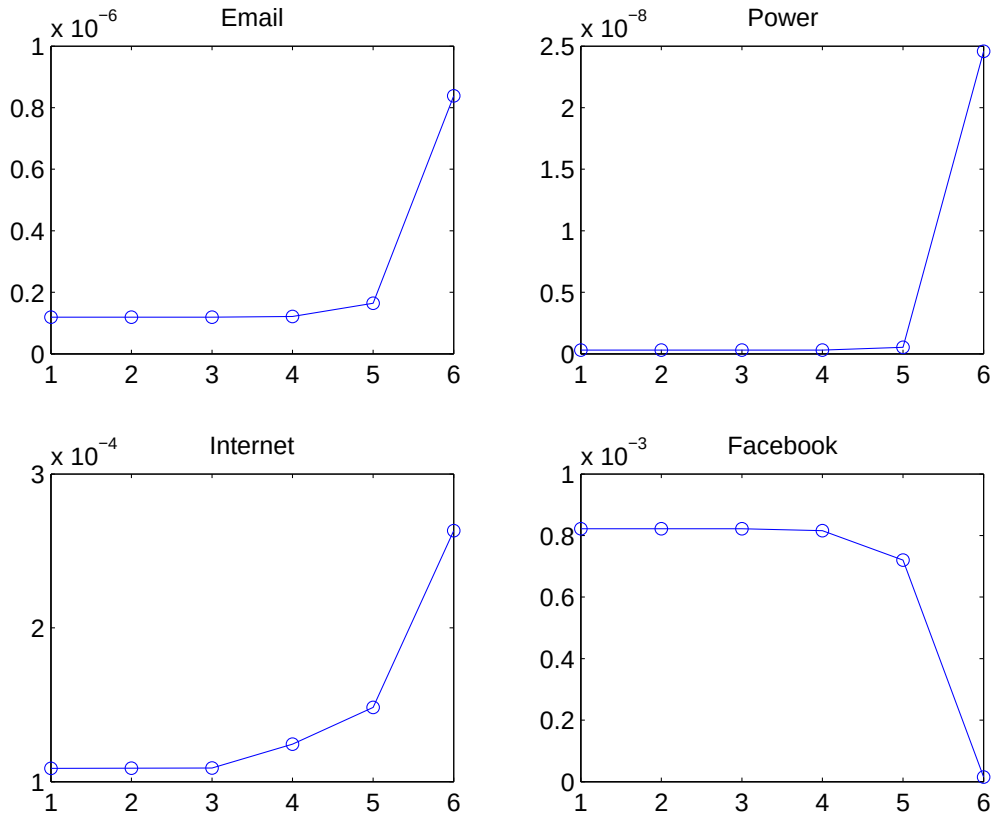

Figure 2: Relative errors for averaged Gauss quadrature rule $\widehat{\mathcal{G}}_{2\ell-r,r} f$, with $r = 1, 2, \ldots, 6$, obtained by 7 iterations of the symmetric block Lanczos method.

To investigate the influence of the block size on the execution time, we compare in Figure 3 the computing times of the block and scalar algorithms when evaluating (1.1) with a block $W \in \mathbb{R}^{m\times k}$ for $k = 1, 2, \ldots, 20$. The test is performed on the network Facebook, the largest network in our test set. It can be seen that the speedup produced by the block method with respect to the scalar algorithm grows linearly with the block size k.

We turn to quadrature rules for the approximation of expressions of the form (1.2) with a nonsymmetric adjacency matrix $A \in \mathbb{R}^{m\times m}$. The nonsymmetric block averaged Gauss rules $\widehat{\mathcal{G}}_{2\ell-r,r} f$ for $r \in \{1, \ell - 1\}$ defined by (3.10) are compared to averages of nonsymmetric block Gauss and block anti-Gauss rules (5.2) for $\ell = 7$ when applied to several directed networks. We apply these rules to approximate expressions (1.2) with

$$f(A) = (I - \mu A)^{-1}$$

for $\mu = 0.9/\rho(A)$. The matrix W is the same as in the above examples, and $V = W$. Similarly as above, we use the singular value decomposition of $W = U_W \Sigma_W V_W^T$ and evaluate an expression of the form (5.3) with A nonsymmetric.

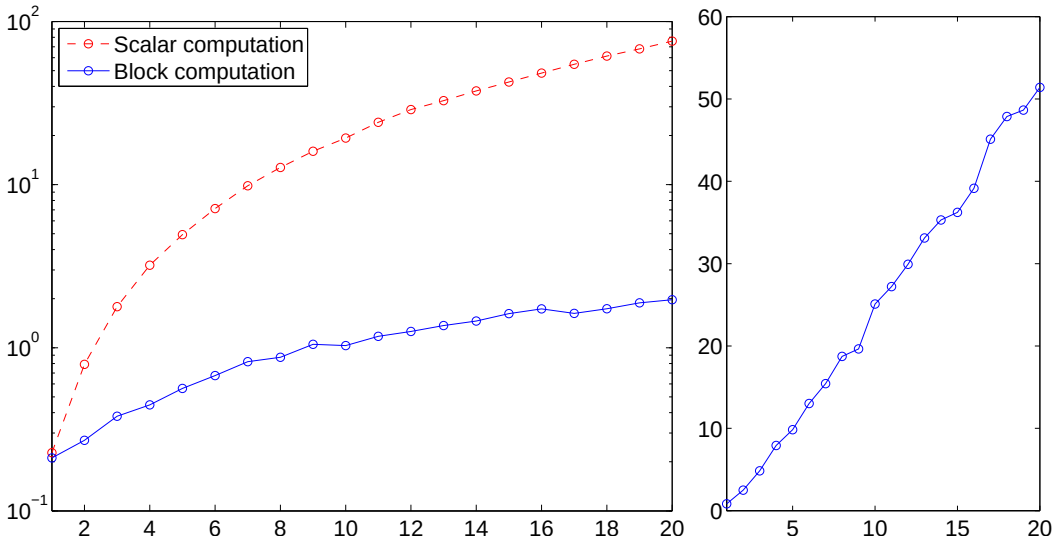

Figure 3: In the graph on the left, the execution time for computing a $k \times k$ block of communicabilities for the Facebook network (63731 nodes) is depicted with respect to the block size k. The continuous curve corresponds to block averaged Gauss quadrature $\widehat{\mathcal{G}}_{13,1} f$, the dashed curve to the scalar algorithm. The graph on the right displays the ratio between the two computing times.

Table 11: Computation time and relative errors obtained performing 7 iterations of the nonsymmetric Lanczos method (E_7) using both a pair of Gauss and anti-Gauss quadrature formulas, and averaged Gauss quadrature rules (AGQ). The error is computed with respect to `inv` for the first four networks. For the last three we let the Gauss/anti-Gauss method iterate until the stopping tolerance 10^{-12} is reached.

	Gauss/anti-Gauss		AGQ ($r = 6$)		AGQ ($r = 1$)	
matrix	time	E_7	time	E_7	time	E_7
Airlines	6.03e-03	5.12e-10	6.09e-03	2.10e-12	6.60e-03	1.72e-12
Celegans	6.74e-03	1.04e-05	1.00e-02	1.00e-06	7.01e-03	5.94e-07
Air500	8.46e-03	3.65e-12	8.70e-03	6.03e-13	9.19e-03	5.71e-13
Twitter	3.42e-02	1.09e-08	4.68e-02	1.16e-12	4.07e-02	1.17e-12
Wikipedia	3.11e-01	1.09e-05	3.53e-01	1.89e-07	3.47e-01	3.16e-08
Slashdot	4.67e-01	2.87e-08	5.22e-01	5.45e-09	5.15e-01	2.20e-09
Vfem	1.39e+00	1.90e-14	1.55e+00	7.78e-16	1.49e+00	9.34e-16

Thus, we apply the the nonsymmetric Lanczos method and quadrature rules to determine estimates of the expression $U_W^T f(A) U_W$.

The block quadrature rules are compared for adjacency matrices defined by the 7 directed networks listed in Table 11, which reports, for each network, the execution time and the error E_7 obtained when executing $\ell = 7$ steps of the nonsymmetric block Lanczos method. The "exact" solution for networks with fewer than 5000 nodes is computed by the MATLAB function `inv`, that is, by inverting the matrix $I - \mu A$ by means of its LU factorization. When the number of nodes is larger, we determine an "exact solution" similarly as above. Thus, we increase the number of steps with the nonsymmetric Lanczos method until the componentwise difference between the approximations delivered by the nonsymmetric block Gauss and anti-Gauss rules is bounded by 10^{-12}. The average of these rules is considered the "exact" value and used to compute the error. Table 11 reports execution times in seconds and errors. The notation for Table 11 is the same as for Table 10. We see that the nonsymmetric block averaged rule $\widehat{\mathcal{G}}_{13,1} f$ of Section 3 for most networks gives higher accuracy than $\widehat{\mathcal{G}}_{8,6} f$ and requires slightly more computing time. Both quadrature rules $\widehat{\mathcal{G}}_{13,1} f$ and $\widehat{\mathcal{G}}_{8,6} f$ demand about the same computing time as the evaluation of the average (5.2) for $\ell = 7$, but yield higher accuracy. Figure 4 displays the errors obtained with $\widehat{\mathcal{G}}_{2\ell-1,1} f$ and the average rule (5.2) versus the iterations

$\ell = 2, 3, \ldots, 7$ for four of the networks. The figure is analogous to Figure 1.

Figure 5 replicates the experiment of Figure 2 for four directed networks. For two test networks, Wikipedia and Slashdot, a moderately small value of r gives higher accuracy in the computed approximations than larger r-values. This is the typical situation. In two of the examples, Twitter and Vfem, the error increases slightly, but does not vary significantly, when r decreases.

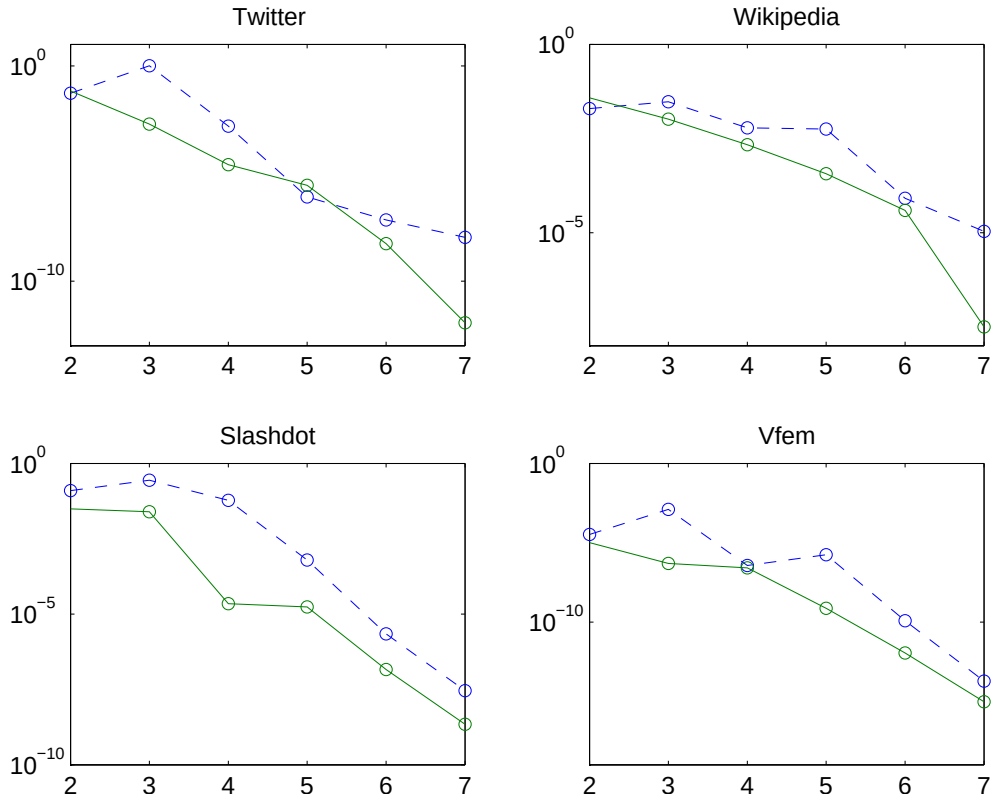

Figure 4: Relative errors versus number of iterations ℓ with the nonsymmetric block Lanczos method for nonsymmetric averaged Gauss quadrature $\widehat{\mathcal{G}}_{2\ell-1,1} f$ (continuous curve) and nonsymmetric Gauss/anti-Gauss quadrature (5.2) (dashed curve) for $\ell = 2, 3, \ldots, 7$.

The choice of the parameter μ is important for the computation of the resolvent. To understand how this parameter influences the accuracy and the convergence, we let $\mu = \xi/\rho(A)$ for $\xi = 0.10, 0.15, \ldots, 0.95$. The number of steps ℓ of the nonsymmetric block Lanczos algorithm is increased until the relative difference between the block Gauss rule $\mathcal{G}_{\ell-1} f$ and the associated block anti-Gauss rule $\mathcal{H}_\ell f$ is less than 10^{-3}; the difference is measured analogously to (5.4). We use the network Slashdot. The upper graph in Figure 6 shows the number of steps ℓ performed for each value of ξ. In the lower graph, we display by a dashed curve the error produced by the average (5.2), while the continuous curve shows the error obtained performing the same number of steps with the averaged Gauss quadrature formula $\widehat{\mathcal{G}}_{2\ell-1,1} f$. The accuracy of the latter is between a factor 4 and 280 higher than for the alternative.

6. Conclusion

This paper introduces new block quadrature rules for the approximation of expressions of the form (1.1) and (1.2). Some properties of these rules are shown. Computed examples illustrate that they can be applied to estimate the error in block Gauss quadrature rule, and that they can give a smaller error for essentially the same computational effort as the use of pairs of block Gauss and anti-Gauss rules.

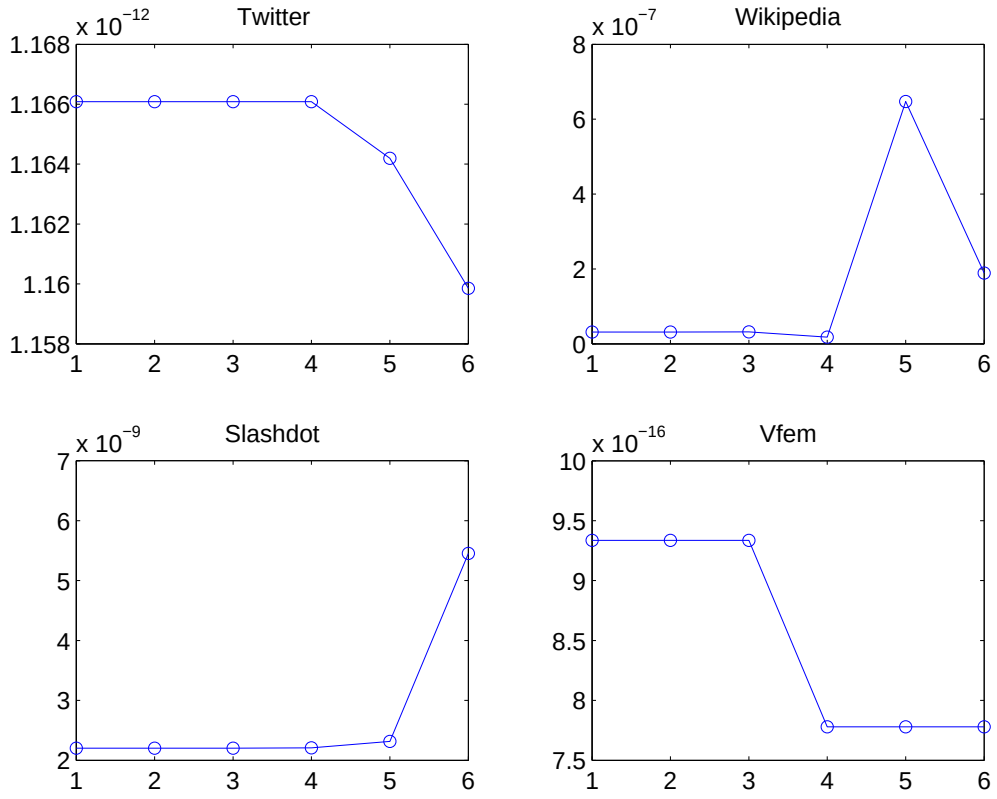

Figure 5: Relative errors for nonsymmetric averaged Gauss quadrature rule $\widehat{\mathcal{G}}_{2\ell-r,r}\, f$, with $r = 1, 2, \ldots, 6$, obtained by 7 iterations of the nonsymmetric block Lanczos method.

Acknowledgements

The authors would like to thank the referees for comments. Work of L.R. was supported by the University of Cagliari RAS Visiting Professor Program and by NSF grant DMS-1115385. Work of G.R. was partially supported by INdAM-GNCS.

[1] A. Alsayed and D. J. Higham, *Betweenness in time dependent networks*, Chaos, Solitons, Fract., 72 (2015), pp. 35–48.

[2] J. Baglama, C. Fenu, L. Reichel, and G. Rodriguez, *Analysis of directed networks via partial singular value decomposition and Gauss quadrature*, Linear Algebra Appl., 456 (2014) pp. 93–121.

[3] Z. Bai, D. Day, and Q. Ye, *ABLE: An adaptive block Lanczos method for non-Hermitian eigenvalue problems*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1060–1082.

[4] M. Benzi and P. Boito, *Quadrature rule-based bounds for functions of adjacency matrices*, Linear Algebra Appl., 433 (2010), pp. 637–652.

[5] M. Benzi, E. Estrada, and C. Klymko, *Ranking hubs and authorities using matrix functions*, Linear Algebra Appl., 438 (2013), pp. 2447–2474.

[6] M. Benzi and C. Klymko, *Total communicability as a centrality measure*, Journal of Complex Networks, 1 (2013), pp. 1–26.

[7] C. Brezinski, P. Fika, and M. Mitrouli, *Estimations of the trace of powers of self-adjoint operators by extrapolation of the moments*, Electron. Trans. Numer. Anal., 39 (2012), pp. 144–159.

[8] 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), pp. 21–40.

[9] D. Calvetti and L. Reichel, *Application of a block modified Chebyshev algorithm to the iterative solution of symmetric linear systems with multiple right hand side vectors*, Numer. Math., 68 (1994), pp. 3–16.

[10] J. J. Croft, E. Estrada, D. J. Higham, and A. Taylor, *Mapping directed networks*, Electron. Trans. Numer. Anal., 37 (2010), pp. 337–350.

[11] A. J. Duran and P. Lopez–Rodriguez, *Orthogonal matrix polynomials: zeros and Blumenthal's theorem*, J. Approx. Theory, 84 (1996), pp. 96–118.

[12] E. Estrada, *The Structure of Complex Networks*, Oxford University Press, Oxford, 2012.

[13] E. Estrada, N. Hatano, and M. Benzi, *The physics of communicability in complex networks*, Physics Reports, 514 (2012), pp. 89–119.

[14] E. Estrada and D. J. Higham, *Network properties revealed through matrix functions*, SIAM Rev., 52 (2010), pp. 696–714.

[15] E. Estrada and J. A. Rodríguez-Velázquez, *Subgraph centrality in complex networks*, Phys. Rev. E, 71 (2005), 056103

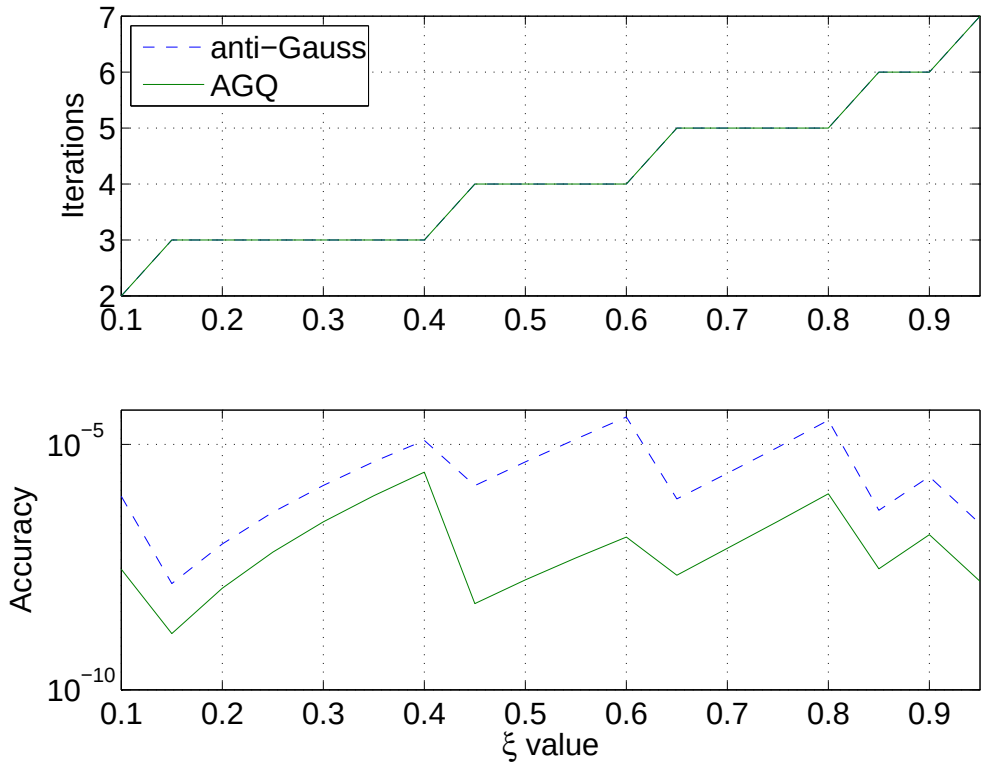

Figure 6: Upper graph: iterations performed by the nonsymmetric Lanczos method to compute the resolvent with accuracy 10^{-3} when $\xi = \mu\rho(A)$ varies. Lower graph: relative errors for averaged Gauss quadrature (continuous curve) and nonsymmetric Gauss/anti-Gauss quadrature (dashed curve).

[16] C. Fenu, D. Martin, L. Reichel, and G. Rodriguez, *Network analysis via partial spectral factorization and Gauss quadrature*, SIAM J. Sci. Comput., 35 (2013), pp. A2046–A2068.

[17] C. Fenu, D. Martin, L. Reichel, and G. Rodriguez, *Block Gauss and anti-Gauss quadrature with application to networks*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 1655–1684.

[18] P. Fika, M. Mitrouli, and P. Roupa, *Estimates for the bilinear form $x^TA^{-1}y$ with applications to linear algebra problems*, Electron. Trans. Numer. Anal., 43 (2014), pp. 70–89.

[19] K. Gallivan, M. Heath, E. Ng, B. Peyton, R. Plemmons, J. Ortega, C. Romine, A. Sameh, and R. Voigt, *Parallel Algorithms for Matrix Computations*, SIAM, Philadelphia, 1990.

[20] G. H. Golub and G. Meurant, *Matrices, moments and quadrature*, in Numerical Analysis 1993, eds. D. F. Griffiths and G. A. Watson, Longman, Essex, England, 1994, pp. 105–156.

[21] G. H. Golub and G. Meurant, *Matrices, Moments and Quadrature with Applications*, Princeton University Press, Princeton, 2010.

[22] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, 2013.

[23] P. Grindrod and D. J. Higham, *A matrix iteration for dynamic network summaries*, SIAM Rev., 55 (2013), pp. 118–128.

[24] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.

[25] N. J. Higham, *The scaling and squaring method for the matrix exponential revisited*, SIAM Rev., 51 (2009), pp. 747–764.

[26] D. P. Laurie, *Anti-Gaussian quadrature formulas*, Math. Comp., 65 (1996), pp. 739–747.

[27] M. E. J. Newman, *Networks: An Introduction*, Oxford University Press, Oxford, 2010.

[28] L. Reichel, M. M. Spalević, and T. Tang, *Generalized averaged Gauss quadrature rules for the approximation of matrix functionals*, submitted for publication.

[29] A. Sinap and W. Van Assche, *Polynomial interpolation and Gaussian quadrature for matrix-valued functions*, Linear Algebra Appl., 207 (1994), pp. 71–114.

[30] M. M. Spalević, *On generalized averaged Gaussian formulas*, Math. Comp., 76 (2007), pp. 1483–1492.

[31] M. M. Spalević, *A note on generalized averaged Gaussian formulas*, Numer. Algorithms, 46 (2007), pp. 253–264.

[32] The University of Florida Sparse Matrix Collection, `http://www.cise.ufl.edu/research/sparse/matrices/`.