New matrix function approximations and quadrature rules based on the Arnoldi process

Nasim Eshghi

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

Thomas Mach

Institute for Mathematics, University of Potsdam, 14476 Potsdam, Germany

Lothar Reichel

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

Abstract

The Arnoldi process can be applied to inexpensively approximate matrix functions of the form $f(A)v$ and matrix functionals of the form $v^*(f(A))^*g(A)v$, where A is a large square non-Hermitian matrix, v is a vector, and the superscript * denotes transposition and complex conjugation. Here f and g are analytic functions that are defined in suitable regions in the complex plane. This paper reviews available approximation methods and describes new ones that provide higher accuracy for essentially the same computational effort by exploiting available, but generally not used, moment information. Numerical experiments show that in some cases the modifications of the Arnoldi decompositions proposed can improve the accuracy of $v^*(f(A))^*g(A)v$ about as much as performing an additional step of the Arnoldi process.

Keywords: Arnoldi process, matrix function approximation, quadrature rule

AMS classification: 65F60, 41A10, 15A16, 65D32

Email addresses: neshghi@kent.edu (Nasim Eshghi), mach@uni-potsdam.de (Thomas Mach), reichel@math.kent.edu (Lothar Reichel)

1. Introduction

Let $A \in \mathbb{C}^{N\times N}$ be a large, possibly sparse, non-Hermitian matrix, and let $v \in \mathbb{C}^N \backslash \{0\}$. Applying $1 \leq n \ll N$ steps of the Arnoldi process to the matrix A with initial vector v gives the Arnoldi decomposition

$$AV_n = V_n H_{n,n} + \hat{v}_{n+1} e_n^T, \tag{1}$$

where $V_n = [v_1, v_2, \ldots, v_n] \in \mathbb{C}^{N\times n}$ and $\hat{v}_{n+1} \in \mathbb{C}^N$ satisfy $V_n^* V_n = I_n$, $V_n^* \hat{v}_{n+1} = 0$, $v_1 = v/\| v \|$, and $H_{n,n} = [h_{i,j}]_{i,j=1}^n \in \mathbb{C}^{n\times n}$ is an upper Hessenberg matrix, i.e., entries $h_{i,j}$ below the subdiagonal are zero. Here and throughout this paper $I_n \in \mathbb{R}^{n\times n}$ denotes the identity matrix, e_j is the jth column of an identity matrix of suitable order, and $\| \cdot \|$ stands for the Euclidean vector norm. The superscript * denotes transposition and complex conjugation; the superscript ^{T} stands for transposition only. We assume that the number of Arnoldi steps, n, is small enough so that the decomposition (1) with the stated properties exists, and that the vector $\hat{v}_{n+1}$ is nonvanishing. This is the generic situation; see, e.g., Golub and Van Loan [26, Section 10.5.1] or Saad [36, Chapter 6] for discussions on the Arnoldi process. In applications of interest to us, n is much smaller that N. We will comment on the rare situation when the Arnoldi process breaks down below. An algorithm for the Arnoldi process is provided in Section 2. Here, we only note that the computation of the decomposition (1) requires n matrix-vector product evaluations with the matrix A, which is typically the dominating computational work for small n; see Section 2 for details.

We are concerned with the approximation of matrix functions of the form

$$f(A)v \tag{2}$$

and of positive semidefinite quadratic forms

$$\langle f, g \rangle = v^*(f(A))^* g(A) v. \tag{3}$$

Assume for the moment that the functions f and g are analytic in sufficiently large simply connected regions in the complex plane. Then $f(A)$ and $g(A)$ can be represented in terms of Cauchy integrals in the complex plane, see, e.g., [26, Section 9.2.7], [29, Section 1.2.3], and [30]. These representations show that (3) can be expressed as the double integral

$$\langle f, g \rangle = \frac{1}{4\pi^2} \int_\Gamma \int_\Gamma \overline{f(z_1)} g(z_2) v^* (\overline{z}_1 I - A^*)^{-1} (z_2 I - A)^{-1} v \, \overline{dz}_1 \, dz_2, \tag{4}$$

where the contour of integration Γ contains the spectrum of A in its interior and the bar denotes complex conjugation. The approximations of (3) that we will determine by using the Arnoldi decomposition (1) may be considered quadrature rules for the approximation of (4). We therefore refer to these approximations as *Arnoldi quadrature rules*.

The need to evaluate expressions of the forms (2) and (3) arises in many applications, such as in the solution of partial differential equations, network analysis, and the solution of linear discrete ill-posed problems; see [4, 9, 11, 12, 14, 24, 35]. We will discuss applications to network analysis in Section 4.

When the matrix A is large, the evaluation of (2) by first computing $f(A)$, or evaluating (3) by first computing $f(A)$ and $g(A)$, may be prohibitively expensive both in terms of computing time and computer memory. The memory requirement may be substantial, because even when the matrix A is sparse and requires little computer memory, the matrices $f(A)$ and $g(A)$, in general, are not. This is, for instance, the case, when $f(t) = \exp(t)$. In addition, the evaluation of $f(A)$ requires considerable computational effort when A is large. These difficulties can be circumvented by observing that neither $f(A)$ nor $g(A)$ are explicitly required to compute approximations of (2) and (3), only approximations of $f(A)v$ and $g(A)v$ are needed.

A commonly used approximation of (2) based on (1) is furnished by

$$f_n = V_n f(H_{n,n}) e_1 \| v \|. \tag{5}$$

This approximation requires that $f(H_{n,n})$ be well defined. For instance, it suffices that f, as well as appropriate derivatives, if $H_{n,n}$ has nontrivial Jordan blocks, are defined at the eigenvalues of $H_{n,n}$; see [29, Definition 1.1]. Alternatively, $f(H_{n,n})$ can be defined with a Cauchy integral analogously to (4).

Let $\mathbb{P}_{n-1}$ denote the set of all polynomials of degree at most $n-1$. It is well known that

$$f(A)v = V_n f(H_{n,n}) e_1 \| v \| \qquad \forall f \in \mathbb{P}_{n-1}; \tag{6}$$

see, e.g., [2, 21, 9, 13, 35]. This result can easily be established by observing that

$$A^i v = \| v \| V_n H_{n,n}^i e_1 \quad \text{for} \quad 1 \leq i \leq n-1,$$

which can be shown by induction over i. We remark that the evaluation of $f(H_{n,n})$ is much cheaper than the calculation of $f(A)$ when $n \ll N$; see, e.g., Higham [29] for the discussion of many methods for the evaluation of matrix functions.

Freund and Hochbruck [21], and more recently Calvetti et al. [9], considered the approximation of (3) by the Arnoldi quadrature rule

$$\langle f, g\rangle_n = \| v \|^2 e_1^*(f(H_{n,n}))^* g(H_{n,n}) e_1. \tag{7}$$

Properties of this and related approximations of (3) are provided in Section 3 as well as in [9, 21]. Freund and Hochbruck [21] showed by induction that the Arnoldi quadrature rule (7) is exact for $\{f, g\} \in \mathbb{W}_{n-1,n}$, where

$$\mathbb{W}_{n-1,n} = (\mathbb{P}_{n-1} \oplus \mathbb{P}_n) \cup (\mathbb{P}_n \oplus \mathbb{P}_{n-1});$$

a proof is also provided in [9]. Here $\mathbb{P}_{n-1} \oplus \mathbb{P}_n$ denotes the set of all pairs $\{f, g\}$, where $f \in \mathbb{P}_{n-1}$ and $g \in \mathbb{P}_n$. Hence, $W_{n-1,n}$ is the set of polynomial pairs, where one polynomial is of degree at most n and the other polynomial is of degree at most $n-1$.

The computation of the approximations (5) and (7) requires n steps of the Arnoldi process to be carried out and, therefore, demands the evaluation of n matrix-vector products with the matrix A; see Algorithm 1 below. If the matrix A is large and not very sparse, then each matrix-vector product evaluation is expensive. In addition, if the matrix A is very large, then each orthogonalization step in the algorithm is expensive, too. It is therefore advantageous to keep the number of Arnoldi steps as small as possible to determine approximations of (2) and (3) of desired accuracy, and to avoid unnecessarily many matrix-vector product evaluations and orthogonalization steps.

EXAMPLE 1.1. In this example we will demonstrate that matrix-vector products can be very expensive.

We are interested in finding the capacity of a capacitor. We will use the Laplace equation for the electric potential. To solve this equation efficiently a boundary element method can be employed. To this end we need the single layer potential

$$\phi(x) = \int_A \frac{\sigma(\xi)}{4\pi\epsilon_0 \| x - \xi \|}\, d\xi,$$

where ϕ is the electric potential, A the surface of all electrodes, $\sigma(\xi)$ the density of the charge on the surface A, and ϵ_0 the vacuum electric permittivity. Switching to a weak formulation and discretization leads to a symmetric, dense matrix

$$K|_{ij} = \int_A \int_A \frac{v_j(x) v_i(\xi)}{4\pi\epsilon_0 \| x - \xi \|}\, d\xi\, dx.$$

The entry $C_{k\ell}$ of the capacity matrix associated with the capacity between electrodes k and ℓ is $w_k^T f(K) w_\ell$, where $f(x) = \frac{1}{x}$ and w_k is the vector with

$$w_k|_i = \begin{cases} 1, & \text{supp}(v_i)(x) \subset \Omega_k, \\ 0, & \text{else}, \end{cases}$$

with Ω_k the surface of electrode k. This matrix can be expensive to store and handle. Employing a hierarchical compression with $\mathcal{H}^2$-matrices reduces the required storage to $O(N)$, with a large constant hidden in the $O(\cdot)$, and allows matrix-vector products in $O(N)$ flops [6].

A very fine discretization with 262,146 nodes results in a large, dense matrix of size $246,146 \times 246,146$. For our computations here we used the H2Lib library [28] and based this example on one of the standard examples provided with the library. Without compression 512 GB would be needed to store the matrix. On an Intel Core i710710U CPU with 16 GB of RAM it took 1103 s to assemble the matrix K in the compressed $\mathcal{H}^2$-matrix format. The matrix required 15.45 GB of storage. Thus almost all the available RAM. Performing one matrix-vector product required 1596 s, that is 44% more than for assembling the matrix. The reason is that the $O(N)$ flops require a significant amount of communication between faster and slower computer memory. □

This paper derives new expressions for approximating (2) and new quadrature rules for the approximation of (3) that require the same number of matrix-vector product evaluations as the expressions (5) and (7), and are exact for functions in larger sets than $\mathbb{P}_{n-1}$ and $\mathbb{W}_{n-1,n}$, respectively.

Generically, the vector $\hat{v}_{n+1}$ in (1) is nonvanishing.[1] Assume this to be the case. Then we can define the positive scalar $h_{n+1,n} = \| \hat{v}_{n+1} \|$, the normalized vector $v_{n+1} = \hat{v}_{n+1}/h_{n+1,n}$, as well as the matrices $V_{n+1} = [V_n, v_{n+1}] \in \mathbb{C}^{N\times(n+1)}$ and $H_{n+1,n} \in \mathbb{C}^{(n+1)\times n}$, where the latter matrix is obtained by appending the row $h_{n+1,n} e_n^T$ to $H_{n,n}$. The decomposition (1) then can be expressed as

$$AV_n = V_{n+1} H_{n+1,n}. \tag{8}$$

The matrix $H_{n+1,n}$ contains one more nontrivial entry, $h_{n+1,n}$ than $H_{n,n}$. This entry can be interpreted as a moment. We would like to use this moment when computing an approximation of $f(A)$, because using it may

[1] If $\hat{v}_{n+1}$ is zero, then v is a vector in an n-dimensional invariant subspace of A. We will comment on this situation in Proposition 1 below.

provide a more accurate approximation of $f(A)$ than $f(H_{n,n})$. We will show that adding a column of zeros to $H_{n+1,n}$ not only makes the matrix square, and thus allows the easy evaluation of f at this matrix, but also gives an approximation of $f(A)v$ that is more accurate than (1). Of course, we may extend $H_{n+1,n}$ to a square matrix by appending a nonvanishing column. This is discussed in Subsection 2.2.

An approach that has been advocated by Saad [35] is described in Subsection 2.1. Saad only considered the approximation of (2) when $f(t)$ is the exponential function. We discuss the approximation of more general functions and show that this approach is equivalent to zero-padding of $H_{n+1,n}$.

In the special case when $f(t) \equiv 1$, the functional (3) simplifies to

$$\mathbb{I}(g) = v^* g(A) v. \tag{9}$$

For $g(x) = 1/x$, this problem has been investigated by Strakoš and Tichý [38] and Fika et al. [20]. The approximation of expressions of the form (9) has received considerable attention when the matrix A is Hermitian; see, e.g., [1, 16, 19, 25] for methods that exploit the connection between the Hermitian Lanczos process, orthogonal polynomials, and Gauss quadrature rules. When the matrix A is non-Hermitian, the functional (9) can be approximated by methods that are based on the non-Hermitian Lanczos process [1, 19]. Techniques that use extrapolation are developed in [7, 8, 20]. A careful comparison of all these methods is outside the scope of the present paper. Here we only note that approximation methods that are based on the non-Hermitian Lanczos process require the evaluation of matrix-vector products with both the matrices A and A^*.

The methods considered in the present paper only demand the evaluation of matrix-vector products with A. This is beneficial when it is easy to compute matrix-vector products with A but not with A^*. For instance, this is the case when A approximates a Fredholm integral operator of the first or second kinds and matrix-vector products with A are evaluated by a multipole method. Then A is not explicitly formed and matrix-vector products with A^* are difficult to compute; see, e.g., [27] for a discussion on the multipole method.

This paper is organized as follows. Section 2 describes new approaches to approximate expressions of the form (2). New quadrature rules for the approximation of (3) are discussed in Section 3. A few computed examples are presented in Section 4 and concluding remarks can be found in Section 5.

2. New matrix function approximations based on the Arnoldi decomposition

This section describes new approaches to approximate expressions of the form (2). Subsection 2.1 shows how the Arnoldi decomposition (8) can be modified to yield higher accuracy. This approach has previously been advocated by Saad [35] for the matrix exponential. We consider more general functions f. Subsection 2.2 discusses appending a column to the matrix $H_{n+1,n}$ to obtain a square upper Hessenberg matrix, that allow us to determine more accurate approximations of $f(A)$ than (5).

For future reference we provide an algorithm for the Arnoldi process (Algorithm 1). We assume that the number of steps, n, is sufficiently small so that breakdown due to division by zero in line 9 does not occur. These events are rare but fortuitous; see Proposition 1 below.

The following implementation of the Arnoldi process is based on modified Gram–Schmidt orthogonalization of the columns of the matrix V_{n+1}.

Algorithm 1 The Arnoldi process

```
1:  Input: A ∈ C^{N×N}, v ∈ C^n\{0}, number of steps n.
2:  v_1 := v/‖v‖
3:  for j = 1 to n
4:      w := Av_j
5:      for k = 1 to j
6:          h_{k,j} := v_k^* w
7:          w := w − v_k h_{k,j}
8:      end for
9:      h_{j+1,j} := ‖w‖; v_{j+1} := w/h_{j+1,j}
10: end for
11: Output: upper Hessenberg matrix H_{n+1,n} = [h_{k,j}] ∈ C^{(n+1)×n} matrix
12:         V_{n+1} = [v_1, v_2, ..., v_{n+1}] ∈ C^{N×(n+1)} with orthonormal
            columns
```

The methods described in this section are not required in the event that the Arnoldi process breaks down. This is discussed in the following proposition.

Proposition 1. *Assume that Algorithm 1 breaks down at step $\ell \geq 1$, that is $h_{j+1,j} > 0$ for $1 \leq j < \ell$, and $h_{\ell+1,\ell} = 0$. Let $H_{\ell,\ell} \in \mathbb{R}^{\ell\times\ell}$ be the upper Hessenberg matrix determined by Algorithm 1. Let $f(H_{\ell,\ell})$ and $g(H_{\ell,\ell})$ be*

well defined. Then

$$f(A)v = V_\ell f(H_{\ell,\ell})e_1 \| v \|, \tag{10}$$

$$\langle f, g \rangle = \langle f, g \rangle_\ell. \tag{11}$$

Proof. The relation (10) follows from the observations that any matrix function $f(A)$ is a polynomial in $A \in \mathbb{C}^{N\times N}$ of degree at most $N-1$, see, e.g., [29, Section 1.2], and that breakdown implies that the Krylov subspace spanned by the columns of V_ℓ is invariant under A. The relation (11) can be shown similarly. □

2.1. Modification of the function f

Let $t_0 \in \mathbb{C}$ be in the domain of the function f and express f as

$$f(t) = f(t_0) + (t - t_0)f_1(t), \qquad f_1(t) := \frac{f(t) - f(t_0)}{t - t_0} \tag{12}$$

for t in the domain of f; to permit $t = t_0$, we require f to be continuously differentiable at $t = t_0$. The expression (12) allows us to replace the determination of an approximation of f by computing an approximation of f_1. Our reason for doing this will become apparent shortly. Thus, we will approximate $f_1(A)v$ by using the right-hand side of (5) with f replaced by f_1. This gives

$$f(A)v \approx f(t_0)v + (A - t_0 I)V_n f_1(H_{n,n})e_1 \| v \|, \tag{13}$$

where

$$f_1(H_{n,n}) = (f(H_{n,n}) - f(t_0)I_n)(H_{n,n} - t_0 I_n)^{-1}.$$

We remark that if t_0 belongs to the spectrum of $H_{n,n}$, then we can use the Schur factorization of $H_{n,n}$ and define f_1 by continuity.

Theorem 1. *Let $f \in \mathbb{P}_n$. Then equality holds in* (13).

Proof. It suffices to show that equality holds in (13) for $f(t) = (t - t_0)^k$ for $k = 0, 1, \ldots, n$. Consider the case $k = n$. Then $f_1(t) = (t - t_0)^{n-1}$ and the right-hand side of (13) becomes, when substituting t by A and t_0 by $t_0 I$,

$$(A - t_0 I)V_n f_1(H_{n,n})e_1 \| v \| = (A - t_0 I)f_1(A)v = (A - t_0 I)^n v. \tag{14}$$

where the first equality follows from (6). The left-hand side of (14) equals the right-hand side of (13) because $f(t_0) = 0$. The result for $k < n$ can be shown similarly. □

Theorem 2. *Let the matrices $H_{n+1,n}$, V_n, and V_{n+1} be defined by the decomposition* (8)*, let the matrix $\widehat{H}_{n+1,n+1} \in \mathbb{C}^{(n+1)\times(n+1)}$ have the leading $(n+1)\times n$ submatrix $H_{n+1,n}$ and vanishing last column, and let the matrix $H_{n,n}$ be the leading $n \times n$ submatrix of $H_{n+1,n}$. Let $t_0 = 0$ and assume that f is defined at $\widehat{H}_{n+1,n+1}$ and t_0, and that the function f_1, given by* (12)*, is defined at $H_{n,n}$. Then*

$$V_{n+1} f(\widehat{H}_{n+1,n+1}) e_1 = f(t_0) v_1 + A V_n f_1(H_{n,n}) e_1.$$

Hence, using the approximation of f in the right-hand side of (13) *with $t_0 = 0$ is equivalent to extending the matrix $H_{n+1,n}$ by zero-padding.*

Proof. The expression $f(\widehat{H}_{n+1,n+1})$ is a polynomial in $\widehat{H}_{n+1,n+1}$ of degree at most n; see, e.g., [29, Section 1.2.2]. Using the power series representation $f(\widehat{H}_{n+1,n+1}) = \sum_{i=0}^{n} c_i \widehat{H}^i_{n+1,n+1}$, we obtain

$$V_{n+1} f(\widehat{H}_{n+1,n+1}) e_1 = c_0 v_1 + V_{n+1} \sum_{i=1}^{n} c_i \widehat{H}^i_{n+1,n+1} e_1, \tag{15}$$

where the vector v_1 is the first column of V_{n+1}. Substituting

$$\widehat{H}^i_{n+1,n+1} = \begin{bmatrix} H^i_{n,n} & 0 \\ h_{n+1,n} e_n^T H^{i-1}_{n,n} & 0 \end{bmatrix} \quad \text{for} \quad i = 1, 2, \ldots, n,$$

where "0" in the first row of the matrix denotes the zero vector in $\mathbb{C}^n$ and the "0" in the bottom row of the matrix is a scalar, into (15) gives

$$\begin{aligned} V_{n+1} f(\widehat{H}_{n+1,n+1}) e_1 &= c_0 v_1 + V_{n+1} \sum_{i=1}^{n} c_i \begin{bmatrix} H^i_{n,n} \\ h_{n+1,n} e_n^T H^{i-1}_{n,n} \end{bmatrix} e_1 \\ &= c_0 v_1 + V_{n+1} \begin{bmatrix} H_{n,n} \\ h_{n+1,n} e_n^T \end{bmatrix} \sum_{i=1}^{n} c_i H^{i-1}_{n,n} e_1 \\ &= c_0 v_1 + A V_n f_1(H_{n,n}) e_1, \end{aligned}$$

where the last equality follows from (8). This shows the theorem. □

2.2. Extension of the matrix $H_{n+1,n}$

In the last subsection we used zero-padding of $H_{n+1,n}$ to obtain a square matrix. However, performing $n+1$ steps of the Arnoldi process leads to a

matrix

$$H_{n+1,n+1} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} & h_{1,n+1} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n} & h_{2,n+1} \\ & \ddots & \ddots & \vdots & \vdots \\ & & h_{n,n-1} & h_{n,n} & h_{n,n+1} \\ \mathrm{O} & & & h_{n+1,n} & h_{n+1,n+1} \end{bmatrix} \in \mathbb{C}^{(n+1)\times(n+1)} \quad (16)$$

that in general has a non-zero last column. Thus, in this section we will investigate padding $H_{n+1,n}$ by a non-zero $(n+1)$st column. Several choices of vectors for the $(n+1)$st column will be discussed. Theorem 3 below shows that zero-padding is not necessary to achieve exact approximation for $f \in \mathbb{P}_n$.

Replacing the last column of $H_{n+1,n+1}$ by a vector

$$\widehat{h} = [\widehat{h}_{1,n+1}, \widehat{h}_{2,n+1}, \ldots, \widehat{h}_{n+1,n+1}]^T \in \mathbb{C}^{n+1}$$

gives the matrix

$$\widehat{H}_{n+1,n+1} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} & \widehat{h}_{1,n+1} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n} & \widehat{h}_{2,n+1} \\ & \ddots & \ddots & \vdots & \vdots \\ & & h_{n,n-1} & h_{n,n} & \widehat{h}_{n,n+1} \\ \mathrm{O} & & & h_{n+1,n} & \widehat{h}_{n+1,n+1} \end{bmatrix} \in \mathbb{C}^{(n+1)\times(n+1)}. \quad (17)$$

If the last column is *not* obtained through the Arnoldi process, then only n Arnoldi steps are required. This is the reason for our interest in this matrix. The following result generalizes (6).

Theorem 3. *Let the first n columns of the matrix* (17) *agree with the corresponding columns of* (16) *and let the last column of the matrix* (17) *be arbitrary. Then*

$$f(A)v = V_{n+1} f(\widehat{H}_{n+1,n+1}) e_1 \| v \| \qquad \forall f \in \mathbb{P}_n. \quad (18)$$

Proof. This result has been shown by, for instance, Paige et al. [31, Lemma 1] and later by van den Eshof et al. [17, Lemma 3] for the situation when the matrix A is Hermitian and the first n columns of the matrix (17) are generated by the Hermitian Lanczos process, and therefore form a tridiagonal matrix with an $n \times n$ Hermitian leading principal submatrix. The present theorem can be shown in the same manner. A more general version of the theorem has recently be shown by Frommer et al. [23, Theorem 2.7].

We provide a proof for completeness for exactly the statement needed here.

Equation (18) is equivalent to

$$A^k v_1 = V_{n+1} \widehat{H}_{n+1,n+1}^k e_1, \qquad 0 \le k \le n,$$

where v_1 is the first column of V_{n+1}. Using the Arnoldi decomposition (1) and the fact that the matrix $\widehat{H}_{n+1,n+1}$ is of upper Hessenberg form, we obtain

$$A^k v_1 = V_{n+1} \widehat{H}_{n+1,n+1}^k e_1, \qquad k = 1, 2, \ldots, n,$$

where v_1 is the first column of V_{n+1}. This shows (18). The entries of the vectors $\widehat{H}_{n+1,n+1}^k e_1$, $1 \le k \le n$, are independent of the last column of $\widehat{H}_{n+1,n+1}$. □

Remark 4. *The vector $H_{n+1,n+1}e_1$ lives in the subspace span$\{e_1, e_2\}$. Furthermore, $H_{n+1,n+1}^k e_1$ is an element of span$\{e_1, \ldots, e_{k+1}\}$ if $k < n$. Thus, for the first k powers of $H_{n+1,n+1}$ only the entries of the leading $(k+1) \times k$ submatrix are relevant for computing $H_{n+1,n+1}^k e_1$.*

The computation of the right-hand side of (18) requires the evaluation of the matrix function $f(\widehat{H}_{n+1,n+1})$. Typically, the matrix $\widehat{H}_{n+1,n+1}$ is fairly small in applications of interest to us. Assume that the spectral factorization

$$\widehat{H}_{n+1,n+1} = \widehat{S}_{n+1,n+1} \widehat{\Lambda}_{n+1,n+1} \widehat{S}_{n+1,n+1}^{-1}$$

exists. Thus, the eigenvalues of $\widehat{H}_{n+1,n+1}$ are the diagonal entries of $\widehat{\Lambda}_{n+1,n+1} = \mathrm{diag}[\widehat{\lambda}_1, \widehat{\lambda}_2, \ldots, \widehat{\lambda}_{n+1}]$ and the columns of $\widehat{S}_{n+1,n+1} \in \mathbb{C}^{(n+1)\times(n+1)}$ are the associated eigenvectors. Assume further that the matrix $\widehat{S}_{n+1,n+1}$ is not very ill-conditioned. Then it may be attractive to compute $f(\widehat{H}_{n+1,n+1})$ by using

$$f(\widehat{H}_{n+1,n+1}) = \widehat{S}_{n+1,n+1} \mathrm{diag}[f(\widehat{\lambda}_1), f(\widehat{\lambda}_2), \ldots, f(\widehat{\lambda}_{n+1})] \widehat{S}_{n+1,n+1}^{-1}.$$

Several choices of the last column of $\widehat{H}_{n+1,n+1}$ are possible. Using MATLAB-like notation, we denote this column by $\widehat{H}_{n+1,n+1}(1 : n+1, n+1)$. For instance, zero-padding yields

$$\widehat{H}_{n+1,n+1}(1 : n+1, n+1) = [0, \ldots, 0]^T. \tag{19}$$

Then (at least) one of the eigenvalues, say $\widehat{\lambda}_{n+1}$, of $\widehat{H}_{n+1,n+1}$ vanishes. Hence, this choice of the last column requires that $f(t)$ is well defined at

$t = 0$. In particular, this choice cannot be used when $f(t) = \ln(t)$. In this situation, we may be able to choose the last column

$$\widehat{H}_{n+1,n+1}(1:n+1,n+1) = [0,\ldots,0,\lambda]^T, \tag{20}$$

where $\lambda \in \mathbb{C}\backslash\{0\}$. Then the matrix $\widehat{H}_{n+1,n+1}$ has an eigenvalue λ.

The quality of the approximation (5) of (2) may improve by letting the last column of $\widehat{H}_{n+1,n+1}$ be an accurate approximation of the (unknown) last column of the matrix (16). The entries of the matrix (16) for many matrices A decrease smoothly with increasing column index and fixed row index. This suggests that the last column be a multiple of the penultimate column, i.e.,

$$\widehat{H}_{n+1,n+1}(1:n+1,n+1) = \gamma H_{n+1,n}(1:n+1,n)$$

for some scalar γ. We found the choice

$$\gamma = 0.9\frac{\| H_{n,n}(1:n,n) \|}{\| H_{n,n}(1:n,n-1) \|} \tag{21}$$

to give fairly accurate approximations of (3) for various analytic functions f and g, and matrices A. The matrix $\widehat{H}_{n+1,n+1}$ so defined is singular. If we prefer $\widehat{H}_{n+1,n+1}$ to have a specified eigenvalue $\lambda \neq 0$, then we may choose

$$\widehat{H}_{n+1,n+1}(1:n+1,n+1) = [\gamma\widehat{h}_{1,n},\ldots,\gamma\widehat{h}_{n-1,n},\gamma(\widehat{h}_{n,n}-\lambda),\gamma\widehat{h}_{n+1,n}+\lambda]^T.$$

We conclude this section with some comments on two problems that are somewhat related to the one discussed in this subsection. Let A be Hermitian and f be a function that is defined on the convex hull of the spectrum of A. Application of n steps of the Arnoldi process with initial vector v to A gives, assuming that breakdown does not occur, the decomposition (8). The matrix $H_{n+1,n} \in \mathbb{C}^{(n+1)\times n}$ in this decomposition is tridiagonal with a Hermitian leading principal $n \times n$ submatrix. We can append a column to $H_{n+1,n}$ to determine a Hermitian matrix $H_{n+1,n+1} \in \mathbb{C}^{(n+1)\times(n+1)}$. This matrix is uniquely defined except for the last diagonal entry. This entry can be chosen so that the matrix $H_{n+1,n+1}$ has a specified eigenvalue. This forms the basis for computing a Gauss–Radau quadrature rule with a specified node (which equals the specified eigenvalue) for the approximation of $v^*f(A)v$; see [25] for details. Recently, Frommer et al. [22] applied Gauss–Radau rules in the context of a restarted Hermitian Lanczos method. A discussion of the choice of the last diagonal entry in $H_{n+1,n+1}$ when this matrix is not required to have a specified eigenvalue can be found in [16].

The need to choose the last column of $\widehat{H}_{n+1,n+1}$ also arises in the pole placement problem in control theory. This problem is concerned with modifying a row or column of a square matrix so that all eigenvalues of the new matrix obtained have negative real part; see, e.g., [39]. Generically, the last column of $\widehat{H}_{n+1,n+1}$ can be chosen to make the matrix have desired eigenvalues. Discussions on the solvability and numerical aspects of the pole placement problem can be found in [10, 33, 34].

3. New quadrature rules based on the Arnoldi decomposition

We turn to the approximation of the bilinear form (3) and define the quadrature rule

$$\langle f, g\rangle_{n+1} = \| v \|^2 e_1^*(f(\widehat{H}_{n+1,n+1}))^* g(\widehat{H}_{n+1,n+1})e_1, \tag{22}$$

where $\widehat{H}_{n+1,n+1}$ is one of the matrices introduced above. While the result (18) holds independently of the choice of the last column of $\widehat{H}_{n+1,n+1}$, the difference between the right-hand side and left-hand side of (18) for functions $f \notin \mathbb{P}_n$ may depend on this choice. The choice (19) is possible when f is defined at the origin. The last column (20) typically also performs well when $|\lambda|$ is not very large and f is defined at λ. Independently of the choice of the last column of this matrix, we have the following result.

Corollary 1. *Let the first n columns of the matrix* (17) *agree with the corresponding columns of* (16) *and let the last column be arbitrary. Then the quadrature rule* (22) *satisfies*

$$\langle f, g\rangle = \langle f, g\rangle_{n+1} \qquad \forall f, g \in \mathbb{W}_{n,n},$$

where

$$\mathbb{W}_{n,n} = (\mathbb{P}_n \oplus \mathbb{P}_n) \cup (\mathbb{P}_n \oplus \mathbb{P}_n).$$

Proof. The result follows from Theorem 3. □

We note that the set $\mathbb{W}_{n,n}$ contains the set $\mathbb{W}_{n-1,n} = (\mathbb{P}_{n-1} \oplus \mathbb{P}_n) \cup (\mathbb{P}_n \oplus \mathbb{P}_{n-1})$ used in [9, 21].

Corollary 2. *Let $f, g \in \mathbb{P}_n$, let the function f_1 be defined by* (12)*, and let the function g_1 be defined analogously with the point t_1 playing the role of t_0. Then, for all $(f, g) \in \mathbb{W}_{n,n}$,*

$$\begin{aligned}\langle f, g\rangle = \| v \|^2 &(\overline{f(t_0)} g(t_1) \\ &+ e_1^*(f_1(H_{n,n}))^*(H_{n+1,n} - t_0 I_{n+1,n})^*(H_{n+1,n} - t_1 I_{n+1,n}) g_1(H_{n,n})e_1),\end{aligned}$$

where the matrix $I_{n+1,n}$ consists of the first n columns of I_{n+1}.

Proof. The result follows from Theorem 1. □

4. Numerical examples

This section presents a few computed examples that illustrate the approximations described. All computations were carried out in double precision arithmetic using MATLAB R2016b on a 64-bit Lenovo personal computer.

We first consider the quadrature rules of Section 3. At the end of this section we show errors for the approximations of Section 2. For the quadrature rules, we tabulate the relative errors

$$\text{Error} = \frac{|\langle f,g\rangle - \langle f,g\rangle_i|}{|\langle f,g\rangle|}, \qquad i \in \{n, n+1\},$$

where $\langle f,g\rangle$ denotes the exact value (3), $\langle f,g\rangle_n$ stands for the approximation (7) used in [9, 21], and $\langle f,g\rangle_{n+1}$ denotes approximations of the form (22) determined by several choices of the matrix $\widehat{H}_{n+1,n+1}$.

We compare four different methods: 1) We use n steps of the Arnoldi process. This is the baseline for a method requiring n matrix-vector products. 2) We use an $(n+1)\times(n+1)$ matrix obtained by adding a scaled copy of the last column of $H_{n+1,n}$ as $(n+1)$st column,

$$\widehat{H}_{n+1,n+1} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} & \gamma h_{1,n} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n} & \gamma h_{2,n} \\ & \ddots & \ddots & \vdots & \vdots \\ & & h_{n,n-1} & h_{n,n} & \gamma h_{n,n} \\ \mathrm{O} & & & h_{n+1,n} & \gamma h_{n+1,n} \end{bmatrix}, \tag{23}$$

with γ defined by (21). Experiments with $\gamma = 1$ gave worse accuracy for all examples and are not shown below. 3) We use the approximation described in (13), which is equivalent to zero padding $H_{n+1,n}$. 4) For comparison, we also display the error obtained after $n+1$ steps of the Arnoldi process. This is the only method that requires $n+1$ matrix-vector product evaluations with A.

Example 4.1. Let $A \in \mathbb{R}^{N\times N}$ with $N \in \{200, 2000, 10000\}$ be nonsymmetric Toeplitz matrices with first row $[1, 1/2, \ldots, 1/N]$ and first column $[1, 1/2^2, \ldots, 1/N^2]$. We apply n steps of the Arnoldi process to A with initial vector $v = [1, \ldots, 1]^T \in \mathbb{R}^N$. Table 1 shows results for the functions

Table 1: Example 4.1: Relative errors of computed approximations of $v^T f(A)g(A)v$ for $A \in \mathbb{R}^{N\times N}$ a nonsymmetric Toeplitz matrix, $f(t) = g(t) = \exp(t)$, and $v = [1, 1, \ldots, 1]^T$.

		Error	
N		$n = 5$	$n = 10$
200	n Arnoldi steps	$5.7852 \cdot 10^{-4}$	$6.1095 \cdot 10^{-9}$
	scaled nth column	$1.0360 \cdot 10^{-4}$	$4.0040 \cdot 10^{-10}$
	zero padding	$5.9115 \cdot 10^{-4}$	$6.1096 \cdot 10^{-9}$
	$n + 1$ Arnoldi steps	$7.3238 \cdot 10^{-5}$	$4.6439 \cdot 10^{-10}$
2000	n Arnoldi steps	$2.2440 \cdot 10^{-3}$	$2.6904 \cdot 10^{-7}$
	scaled nth column	$1.4752 \cdot 10^{-4}$	$2.1246 \cdot 10^{-8}$
	zero padding	$2.3146 \cdot 10^{-3}$	$2.6908 \cdot 10^{-7}$
	$n + 1$ Arnoldi steps	$4.5982 \cdot 10^{-4}$	$3.4749 \cdot 10^{-8}$
10000	n Arnoldi steps	$3.4127 \cdot 10^{-3}$	$1.1003 \cdot 10^{-6}$
	scaled nth column	$6.7299 \cdot 10^{-4}$	$8.4472 \cdot 10^{-8}$
	zero padding	$3.5232 \cdot 10^{-3}$	$1.1007 \cdot 10^{-6}$
	$n + 1$ Arnoldi steps	$8.5160 \cdot 10^{-4}$	$1.7492 \cdot 10^{-7}$

$f(t) = \exp(t)$ and $g(t) = \exp(t)$. We expect that the approximation obtained after $n + 1$ Arnoldi steps to be a more accurate approximation of $\langle f, g\rangle$ than what we get after n steps only. In the present example, the approximation using zero padding, gives a slightly larger error than just using $H_{n,n}$. The smallest error among the methods that require only n Arnoldi steps is achieved by the approximation (23), that is using a scaled copy of the nth column as $(n + 1)$st column. This holds for all three values of N tested. In fact, the error in these approximations is smaller than the error in the approximation based on $n + 1$ Arnoldi steps for $N \in \{2000, 10000\}$. As the following experiments show this could be a fluke. In any case we are not able to provide upper error bounds that show that (23) is superior to an additional Arnoldi step. □

EXAMPLE 4.2. We now choose $f(t) = g(t) = \sqrt{t + 1}$. The matrix A, vector v, and orders N, and steps n are the same as in Example 4.1. Table 2 lists the relative errors for the different approximations of $\langle f, g\rangle$. Also for this example, the approximations (23) perform well. □

EXAMPLE 4.3. This example uses the same functions f and g as Example 4.1, and the same initial vector v, but a different matrix. The matrix A of the present example is a nearly symmetric Toeplitz matrix with first row $[1/2, 1/2, 1/3, \ldots, 1/N]$ and first column $[1/2, 1/3, \ldots, 1/(N + 1)]$. Results are shown in Table 3. The approximations (23) of $\langle f, g\rangle$ are seen to give smaller errors than the approximations based on n Arnoldi steps, but not

Table 2: Example 4.2: Relative errors of computed approximations of $v^T f(A)g(A)v$ for $A \in \mathbb{R}^{N\times N}$ a nonsymmetric Toeplitz matrix, $f(t) = g(t) = \sqrt{1+t}$, and $v = [1, 1, \ldots, 1]^T$.

		Error	
N		$n = 5$	$n = 10$
200	n Arnoldi steps	$3.3922 \cdot 10^{-6}$	$5.7095 \cdot 10^{-9}$
	scaled nth column	$2.2259 \cdot 10^{-7}$	$1.9204 \cdot 10^{-10}$
	zero padding	$3.3680 \cdot 10^{-6}$	$5.7098 \cdot 10^{-9}$
	$n+1$ Arnoldi steps	$8.9522 \cdot 10^{-7}$	$1.6797 \cdot 10^{-9}$
2000	n Arnoldi steps	$2.3013 \cdot 10^{-6}$	$1.0501 \cdot 10^{-8}$
	scaled nth column	$1.4437 \cdot 10^{-7}$	$2.7235 \cdot 10^{-10}$
	zero padding	$2.2726 \cdot 10^{-6}$	$1.0503 \cdot 10^{-8}$
	$n+1$ Arnoldi steps	$7.1245 \cdot 10^{-7}$	$3.9296 \cdot 10^{-9}$
10000	n Arnoldi steps	$1.3860 \cdot 10^{-6}$	$8.5499 \cdot 10^{-9}$
	scaled nth column	$7.3021 \cdot 10^{-8}$	$1.7912 \cdot 10^{-10}$
	zero padding	$1.3672 \cdot 10^{-6}$	$8.5531 \cdot 10^{-9}$
	$n+1$ Arnoldi steps	$4.4929 \cdot 10^{-7}$	$3.4425 \cdot 10^{-9}$

Table 3: Example 4.3: Relative errors of computed approximations of $v^T f(A)g(A)v$ for $A \in \mathbb{R}^{N\times N}$ a nonsymmetric Toeplitz matrix, $f(t) = g(t) = \exp(t)$, and $v = [1, 1, \ldots, 1]^T$.

		Error	
N		$n = 5$	$n = 10$
200	n Arnoldi steps	$1.1236 \cdot 10^{-5}$	$9.7413 \cdot 10^{-11}$
	scaled nth column	$8.8070 \cdot 10^{-6}$	$8.7963 \cdot 10^{-12}$
	zero padding	$1.1310 \cdot 10^{-5}$	$9.7413 \cdot 10^{-11}$
	$n+1$ Arnoldi steps	$1.8919 \cdot 10^{-6}$	$5.7866 \cdot 10^{-12}$
2000	n Arnoldi steps	$8.4251 \cdot 10^{-6}$	$1.4688 \cdot 10^{-9}$
	scaled nth column	$2.5821 \cdot 10^{-5}$	$1.1130 \cdot 10^{-9}$
	zero padding	$7.9549 \cdot 10^{-6}$	$1.4694 \cdot 10^{-9}$
	$n+1$ Arnoldi steps	$8.3296 \cdot 10^{-8}$	$1.0640 \cdot 10^{-10}$
10000	n Arnoldi steps	$3.3744 \cdot 10^{-5}$	$1.6263 \cdot 10^{-9}$
	scaled nth column	$7.4965 \cdot 10^{-5}$	$1.1720 \cdot 10^{-9}$
	zero padding	$3.2586 \cdot 10^{-5}$	$1.6281 \cdot 10^{-9}$
	$n+1$ Arnoldi steps	$2.6019 \cdot 10^{-6}$	$5.5610 \cdot 10^{-10}$

as small as achieved by $n+1$ Arnoldi steps. The results of this example are more in line with what we expect in general. □

In our next example, the matrix $A = [a_{i,j}] \in \mathbb{R}^{N\times N}$ is an adjacency matrix for a directed unweighted graph with N nodes and without multiple

edges and self-loops. Then $a_{i,j} = 1$ if there is an edge from node i to node j, and $a_{i,j} = 0$ otherwise. Since the graph is directed the adjacency matrix is not symmetric. Typically, the number of edges is much smaller than N^2. This makes the adjacency matrix A sparse. A walk of length k in a graph is a sequence of vertices $\nu_{i_1}, \nu_{i_2}, \ldots, \nu_{i_{k+1}}$ such that there is an edge from vertex ν_{i_j} to vertex $\nu_{i_{j+1}}$ for $j = 1, 2, \ldots, k$. Vertices and edges in a walk may be repeated. The entry $[a_{i,j}^{(\ell)}]$ of the matrix $A^\ell = [a_{i,j}^{(\ell)}]$ is equal to the number of walks of length ℓ starting at node i and ending at node j. Short walks are considered more important than long walks. This motivates the use of matrix functions in network analysis; see [3, 18] for nice introductions. The exponential

$$f(A) = \sum_{\ell=0}^{\infty} \frac{A^\ell}{\ell!}$$

is commonly used. The *total communicability* is defined as $v^* f(A) v$, where $v = [1, 1, \ldots, 1]^T$. A large value indicates that it is easy to communicate or travel within the network that is represented by the graph; see [4] for details. We will compute approximations of the total communicability for a graph that models air traffic.

The matrix

$$\widehat{H}_{n+1,n+1} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} & 0 \\ h_{2,1} & h_{2,2} & \cdots & h_{2,n} & 0 \\ & \ddots & \ddots & \vdots & \vdots \\ & & h_{n,n-1} & h_{n,n} & h_{n+1,n} \\ \mathrm{O} & & & h_{n+1,n} & 0 \end{bmatrix} \tag{24}$$

is an extension of $H_{n+1,n}$ using the last row also as last column. This idea is inspired by the treatment of undirected graphs with symmetric adjacency matrices [16]. A closest Hermitian matrix in $\mathbb{C}^{(n+1)\times(n+1)}$ with leading $(n+1) \times n$ submatrix $H_{n+1,n}$ and arbitrary last column $v \in \mathbb{C}^{n+1}$ in the matrix Frobenius norm $\|\cdot\|_F$ is obtained by solving the minimization problem

$$\min_{v \in \mathbb{C}^{n+1}} \| [H_{n+1,n}, v] - [H_{n+1,n}, v]^* \|_F.$$

Choosing the last entry of v to be zero, we obtain the solution

$$v = [0, \ldots, 0, \bar{h}_{n+1,n}, 0]^T \in \mathbb{C}^{n+1}.$$

In the present example $H_{n+1,n} \in \mathbb{R}^{(n+1)\times n}$.

Table 4: Example 4.4: Relative error of computed approximations of $v^*f(A)v$ for $A \in \mathbb{R}^{500\times 500}$ of the Air500 network, $f(x) = exp(x)$, and $v = [1, 1, \ldots, 1]^T$.

		Error	
N		$n = 5$	$n = 10$
500	n Arnoldi steps	$2.3853 \cdot 10^{-2}$	$8.5168 \cdot 10^{-7}$
	scaled nth column	$7.8598 \cdot 10^{-2}$	$5.1654 \cdot 10^{-7}$
	transposed $(n+1)$st row	$6.0149 \cdot 10^{-4}$	$8.0304 \cdot 10^{-7}$
	zero padding	$2.3691 \cdot 10^{-2}$	$8.5168 \cdot 10^{-7}$
	$n+1$ Arnoldi steps	$8.0514 \cdot 10^{-4}$	$7.1425 \cdot 10^{-8}$

The extension (24) of the matrix $H_{n+1,n}$ is meaningful when the latter matrix has a leading $n \times n$ principal submatrix that is nearly symmetric. The determination of the entries of the matrix (24) requires the evaluation of n steps of the Arnoldi process. This matrix delivers approximations of $\langle f, g\rangle$ of higher accuracy for Example 4.4 than any of the Hessenberg matrices that can be determined with n steps of the Arnoldi process and were used in the previous examples. We remark that the matrix (24) does not outperform the other matrices that require n Arnoldi steps in the previous computed examples.

EXAMPLE 4.4. Let the nonsymmetric matrix $A = [a_{i,j}] \in \mathbb{R}^{500\times 500}$ be the adjacency matrix for the Air500 network that describes flight connections between the top 500 airports within one year from July 1, 2007, to June 30, 2008; see [5, 32]. Thus, the airports are nodes and the flights are edges in the graph determined by the network. The matrix A has the entry $a_{i,j} = 1$ if there is a flight from airport i to airport j. Generally, but not always, $a_{i,j} = 1$ implies that $a_{j,i} = 1$. This makes A close to symmetric. Table 4 displays computed approximations of the total communicability for the network. The approximation of the total communicability determined with the matrix (24) is more accurate than the approximations determined by the other approaches that require the evaluation of n steps of the Arnoldi process. □

The computed examples above illustrate that for several matrices A and functions f and g, more accurate approximations of $\langle f, g\rangle$ than those obtained by using the matrix $H_{n,n}$ in (1) can be determined with the same number of steps with the Arnoldi process. Numerous computed examples, some of which are shown above, suggest that the matrix (23) often yields good results, except when the matrix A is very close to symmetric.

In the remainder of this section, we consider approximations of matrix functions of the form (2) described in Section 2. We measure the relative

Table 5: Example 4.5: Relative errors of computed approximations of $f(A)v$ for $A \in \mathbb{R}^{N\times N}$ a nonsymmetric Toeplitz matrix, $f(t) = \exp(t)$, and $v = [1, 1, \ldots, 1]^T$.

		Error	
N		$n = 5$	$n = 10$
200	n Arnoldi steps	$5.03510 \cdot 10^{-3}$	$3.13885 \cdot 10^{-7}$
	scaled nth column	$1.95280 \cdot 10^{-3}$	$6.37350 \cdot 10^{-8}$
	zero padding	$1.76493 \cdot 10^{-3}$	$6.02077 \cdot 10^{-8}$
	$n+1$ Arnoldi steps	$9.80516 \cdot 10^{-4}$	$3.05590 \cdot 10^{-8}$
2000	n Arnoldi steps	$1.40923 \cdot 10^{-2}$	$8.40692 \cdot 10^{-6}$
	scaled nth column	$7.21887 \cdot 10^{-3}$	$2.53102 \cdot 10^{-6}$
	zero padding	$6.70142 \cdot 10^{-3}$	$2.49285 \cdot 10^{-6}$
	$n+1$ Arnoldi steps	$4.06182 \cdot 10^{-3}$	$1.38556 \cdot 10^{-6}$
10000	n Arnoldi steps	$1.95631 \cdot 10^{-2}$	$2.81242 \cdot 10^{-5}$
	scaled nth column	$1.11112 \cdot 10^{-2}$	$9.91392 \cdot 10^{-6}$
	zero padding	$1.05464 \cdot 10^{-2}$	$1.00081 \cdot 10^{-5}$
	$n+1$ Arnoldi steps	$6.55416 \cdot 10^{-3}$	$5.68982 \cdot 10^{-6}$

error

$$\text{Error} = \frac{\| f(A)v - f_{\text{approx}}(A)v \|}{\| f(A)v \|}, \tag{25}$$

where $f(A)v$ is the exact value (2) and $f_{\text{approx}}(A)v$ stands for one of the approximants described in Section 2.

EXAMPLE 4.5. Let $A \in \mathbb{R}^{N\times N}$ for $N \in \{200, 2000, 10000\}$ be the nonsymmetric Toeplitz matrices defined in Example 4.1, let $v = [1, 1, \ldots, 1]^T$ and $f(t) = \exp(t)$. Table 5 displays the relative errors (25) achieved by some of the approximations of $f(A)v$ described in of Section 2. Among the methods that require the evaluation of n steps of the Arnoldi process, the method equivalent to zero padding is seen to yield the most accurate approximations of $f(A)v$ for both $n = 5$ and $n = 10$ Arnoldi steps and all but the largest value of N. The method based on the matrix (23) determines approximations of about the same accuracy. Both these methods give approximations of higher accuracy than the standard approximation method that uses the matrix (1), but of lower accuracy compared to an additional Arnoldi step. □

EXAMPLE 4.6. This example is concerned with an approximation problem that arises in network analysis. Let $A \in \mathbb{R}^{500\times 500}$ be the adjacency matrix for the graph of Example 4.4. The importance of a node as a receiver and broadcaster of information can be determined by evaluation the entries of $\exp(A)v$ and $\exp(A^*)v$, respectively, for a suitable vector $v \in \mathbb{R}^{500}$;

Table 6: Example 4.6: Relative errors of computed approximations of $f(A)v$ for the adjacency matrix $A \in \mathbb{R}^{500\times 500}$ of the Air500 network, $f(t) = \exp(t)$, and $v = [1, 1, \ldots, 1]^T$.

		Error	
N		$n = 5$	$n = 10$
500	n Arnoldi steps	$2.23385 \cdot 10^{-2}$	$1.51927 \cdot 10^{-6}$
	scaled nth column	$3.97069 \cdot 10^{-2}$	$4.01892 \cdot 10^{-7}$
	transposed $n+1$st row	$4.75809 \cdot 10^{-3}$	$4.60743 \cdot 10^{-7}$
	zero padding	$1.49107 \cdot 10^{-2}$	$5.22552 \cdot 10^{-7}$
	$n+1$ Arnoldi steps	$3.16756 \cdot 10^{-3}$	$2.17088 \cdot 10^{-7}$

see [4, 12]. The choice $v = [1, 1, \ldots, 1]^T$ is commonly used, and we use it in this example. Node j of the graph is an important receiver of information in the network if the jth entry of the vector $\exp(A)v$ is relatively large. We approximate this vector by using the techniques described in Section 2. Table 6 shows the relative errors in these approximations. The approximation of $f(A)v$ determined by the matrix (24) gives the highest accuracy among all methods that require n Arnoldi steps. □

The performance of the Arnoldi process when applied to a large non-Hermitian matrix $A \in \mathbb{C}^{N\times N}$ depends on the structure of the matrix, its spectrum, and on the initial vector $v \in \mathbb{C}^N$. The Arnoldi process has been studied in detail in the context of the FOM and GMRES iterative methods for the solution of large linear systems of equations; see Du et al. [15] and Schweitzer [37] for recent discussions and references. In particular, it is difficult to predict how quickly the iterates determined by FOM and GMRES will converge to the desired solution when these methods are applied to the solution of a linear system of equations with a fairly general non-Hermitian matrix.

Similarly, the quality of the approximations of (2) and (3) determined by the expressions in the right-hand side of (18) and (22), respectively, depends on the structure of the matrix A, its spectrum, the initial vector v, the function f, and the choice of the last columns of the Hessenberg matrix $\widehat{H}_{n+1,n+1}$. A detailed analysis is difficult and outside the scope of the present paper. Numerous numerical examples, some of which are reported above, showed the approximation (13), which is equivalent to zero padding, and the approximations obtained when using the matrix (23) to perform well. For matrices that are close to symmetric, that is $\| A - A^* \|_F$ is small, the approximation determined by using the last row as last column, (24), typically also gave high accuracy.

5. Conclusion

The paper discusses the approximation of matrix functions and quadrature rules based on the Arnoldi process. New methods are proposed that provide more accurate approximations, in the sense that more moments are matched for essentially the same computational effort, as available methods. When the moments matched dominate the approximation, the new methods proposed are more accurate than the available approximation schemes based on the use of the matrix $H_{n,n}$ in (1). In addition, we generalize a method proposed by Saad [35] and show its equivalence to zero-padding of the rectangular matrix $H_{n+1,n}$ in the Arnoldi decomposition (8).

Acknowledgment

The authors would like to thank the referees for comments that lead to an improved presentation. Research by LR was supported in part by NSF grant DMS-1720259.

References

[1] H. Alqahtani and L. Reichel, Simplified anti-Gauss quadrature rules with applications in linear algebra, Numer. Algorithms, 77 (2018), pp. 577–602.

[2] B. Beckermann and L. Reichel, Error estimation and evaluation of matrix functions vis the Faber transform, SIAM J. Numer. Anal., 47 (2009), pp. 3848–3883.

[3] M. Benzi and P. Boito, Matrix functions in network analysis, GAMM-Mitteilungen, 43 (2020), e202000012.

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

[5] Biological Networks Data Sets of Newcastle University. Available at http://www.biological-networks.org/

[6] S. Börm, Efficient numerical methods for non-local operators: $\mathcal{H}^2$-matrix compression, algorithms and analysis. Vol. 14. European Mathematical Society, 2010.

[7] C. Brezinski, P. Fika, and M. Mitrouli, Moments of a linear operator on a Hilbert space, with applications to the trace of the inverse of matrices and the solution of equations, Numer. Linear Algebra Appl., 19 (2012), pp. 937–953.

[8] C. Brezinski, P. Fika, and M. Mitrouli, Estimations of the trace of powers of positive self-adjoint operators by extrapolation of the moments, Electron. Trans. Numer. Anal., 39 (2012), pp. 144–155.

[9] D. Calvetti, S. Kim, and L. Reichel, Quadrature rules based on the Arnoldi process, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 765–781.

[10] D. Calvetti, B. Lewis, and L. Reichel, On the selection of poles in the single input pole placement problem, Linear Algebra Appl., 302-303 (1999), pp. 331–345.

[11] D. Calvetti and L. Reichel, Lanczos-based exponential filtering for discrete ill-posed problems, Numer. Algorithms, 29 (2002), pp. 45–65.

[12] O. De la Cruz Cabrera, M. Matar, and L. Reichel, Analysis of directed networks vis the matrix exponential, J. Comput. Appl. Math., 355 (2019), pp. 182–192.

[13] V. L. Druskin and L. A. Knizhnerman, Two polynomial methods for the computation of functions of symmetric matrices, USSR Comput. Math. Math. Phys., 29 (1989), pp. 112–121.

[14] V. Druskin, L. Knizhnerman, and M. Zaslavsky, Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts, SIAM J. Sci. Comput., 31 (2009), pp. 3760–3780.

[15] K. Du, J. Duintjer Tebbens, and G. Meurant, Any admissible harmonic Ritz value set is possible for GMRES, Electron. Trans. Numer. Anal., 47 (2017), pp. 37–56.

[16] N. Eshghi, L. Reichel, and M. Spalević, Enhanced matrix function approximation, Electron. Trans. Numer. Anal., 47 (2017), pp. 197–205.

[17] J. van den Eshof, A. Frommer, T. Lippert, K. Schilling, and H. A. van der Vorst, Numerical methods for the QCD overlap operator, I. Sign-function and error bounds, Comput. Phys. Commun., 146 (2002), pp. 203–224.

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

[19] 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.

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

[21] R. W. Freund and M. Hochbruck, Gauss quadratures associated with the Arnoldi process and the Lanczos algorithm, in Linear Algebra for Large Scale and Real Time Application, eds. M. S. Moonen, G. H. Golub, and B. L. R. De Moor, Kluwer, Dordrecht, 1993, pp. 377–380.

[22] A. Frommer, K. Lund, M. Schweitzer, and D. B. Szyld, The Radau–Lanczos method for matrix functions, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 710–732.

[23] A. Frommer, K. Lund, and D. B. Szyld, Block Krylov subspace methods for functions of matrices II: Modified block FOM, SIAM J. Matrix Anal. Appl., 41 (2020), pp. 804–837.

[24] E. Gallopoulos and Y. Saad, Efficient solution of parabolic equations by Krylov approximation methods, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1236–1264.

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

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

[27] L. Greengard and V. Rokhlin, A new version of the fast multipole method for the Laplace equation in three dimensions, Acta Numer., 6 (1997), pp. 229–269.

[28] H2Lib, http://www.h2lib.org/, 2015–2020.

[29] N. J. Higham, Functions of Matrices, SIAM, Philadelphia, 2008.

[30] M. Hochbruck and C. Lubich, On Krylov subspace approximations to the matrix exponential operator, SIAM. J. Numer. Anal., 34 (1997), pp. 1911–1925.

[31] C. C. Paige, B. N. Parlett, and H. A. Van der Vorst, Approximate solutions and eigenvalue bounds from Krylov subspaces, Numer. Linear Algebra Appl., 2 (1995), pp. 115–133.

[32] J. Marcelino and M. Kaiser, Critical paths in a metapopulation model of H1N1: Effiently delaying influenza spreading through flight cancellation, PLoS Curr., 4 (2012), e4f8c9a2e1fca8.

[33] V. Mehrmann and H. Xu, An analysis of the pole placement problem. I. The single-input case, Electron. Trans. Numer. Anal., 4 (1996), pp. 89–105.

[34] Y. Saad, Projection and deflation methods for partial pole assignment in linear state feedback, IEEE Trans. Autom. Control, AC-33 (1988), pp. 290–297.

[35] Y. Saad, Analysis of some Krylov subspace approximations to the matrix exponential operator, SIAM J. Numer. Anal., 29 (1992), pp. 209–228.

[36] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd ed., SIAM, Philadelphia, 2003.

[37] M. Schweitzer, Any finite convergence curve is possible in the initial iterations of restarted FOM, Electron. Trans. Numer. Anal., 45 (2016), pp. 133–145.

[38] Z. Strakoš and P. Tichý, On efficient numerical approximation of the bilinear form $c^*A^{-1}b$, SIAM J. Sci. Comput., 33 (2011), pp. 565–587.

[39] W. M. Wonham, Linear Multivariate Control: A Geometric Approach, 3rd ed., Springer, New York, 1985.