NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 0000; **00**:1–23
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla

A rational Arnoldi process with applications

M. S. Pranić[1], L. Reichel[2], G. Rodriguez[3*], Z. Wang[4], and X. Yu[2]

[1]*Department of Mathematics and Informatics, University of Banja Luka, Faculty of Science, M. Stojanovića 2, 51000 Banja Luka, Bosnia and Herzegovina.*
[2]*Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA.*
[3]*Dipartimento di Matematica e Informatica, Università di Cagliari, viale Merello 92, 09123 Cagliari, Italy.*
[4]*Department of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing, P. R. China.*

SUMMARY

The rational Arnoldi process is a popular method for the computation of a few eigenvalues of a large non-Hermitian matrix $A \in \mathbb{C}^{n\times n}$ and for the approximation of matrix functions. The method is particularly attractive when the rational functions that determine the process have only few distinct poles $z_j \in \mathbb{C}$, because then few factorizations of matrices of the form $A - z_j I$ have to be computed. We discuss recursion relations for orthogonal bases of rational Krylov subspaces determined by rational functions with few distinct poles. These recursion formulas yield a new implementation of the rational Arnoldi process. Applications of the rational Arnoldi process to the approximation of matrix functions as well as to the computation of eigenvalues and pseudospectra of A are described. The new implementation is compared to several available implementations. Copyright © 0000 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: rational Arnoldi; recursion relation; matrix function; eigenvalue, pseudospectrum.

In memory of Axel Ruhe.

1. INTRODUCTION

Let $A \in \mathbb{C}^{n\times n}$ be a large, possibly sparse, non-Hermitian matrix, and let $\boldsymbol{v} \in \mathbb{C}^n$ be a vector. Throughout this paper $\|\boldsymbol{v}\|$ denotes the Euclidean vector norm. The standard inner product of vectors in $\mathbb{C}^n$ is written as $\langle\cdot,\cdot\rangle$. Thus, $\|\boldsymbol{v}\| = \langle \boldsymbol{v}, \boldsymbol{v}\rangle^{1/2}$. The rational Arnoldi process determines orthonormal rational functions with prescribed poles with respect to the sesquilinear form

$$\langle f, g\rangle := \boldsymbol{v}^*(f(A))^* g(A)\boldsymbol{v}, \tag{1}$$

where the superscript * denotes transposition and, if applicable, complex conjugation. The expression (1) is an inner product of functions f and g, when the functions live in a suitably restricted set. We are interested in the situation when they are rational functions with fixed poles of low enough order so that (1) is an inner product. The exact order allowed depends on the matrix A and vector $\boldsymbol{v}$. We note that the inner product (1) also can be represented as

$$\langle f, g\rangle = \frac{1}{4\pi^2}\int_\Gamma\int_\Gamma \overline{f(z)}g(\zeta)\boldsymbol{v}^*(\overline{z}I - A^*)^{-1}(\zeta I - A)^{-1}\boldsymbol{v}\,\overline{dz}d\zeta,$$

*Correspondence to: Dipartimento di Matematica e Informatica, Università di Cagliari, viale Merello 92, 09123 Cagliari, Italy. E-mail: rodriguez@unica.it.

where the contour of integration Γ contains the spectrum of A in its interior and the bar denotes complex conjugation; see, e.g., [15] for discussions of related representations. The rational Arnoldi process described by Algorithm 1 below was introduced by Ruhe [20] for the computation of selected eigenvalues and eigenvectors of a large matrix; see also [21, 22, 23]. The shifts z_i in the algorithm are the preselected poles of the orthogonal rational functions determined by the rational Arnoldi process. Applications of rational Arnoldi and Lanczos processes to the approximation of matrix functions are discussed in [2, 3, 4, 5, 12, 15, 16, 17].

Algorithm 1 The rational Arnoldi process

1: **Input:** $A \in \mathbb{C}^{n\times n}$, initial vector $\boldsymbol{v} \in \mathbb{C}^n\backslash\{\mathbf{0}\}$, shifts $z_i \in \mathbb{C}$, $i = 1, 2, \ldots, \ell$.
2: $\boldsymbol{v}_1 := \boldsymbol{v}/\|\boldsymbol{v}\|$
3: **for** $i = 1, \ldots, \ell$
4: Apply one step of rational Arnoldi with shift z_i. This gives

$$\begin{array}{l} \boldsymbol{w} := (A - z_i I)^{-1}\boldsymbol{v}_i - \sum_{j=1}^{i} h_{j,i}\boldsymbol{v}_j \\ h_{i+1,i} := ||\boldsymbol{w}||, \;\; \boldsymbol{v}_{i+1} := \boldsymbol{w}/h_{i+1,i} \end{array} \tag{2}$$

5: where I denotes the identity matrix and the coefficients $h_{j,i}$ are chosen
6: so that the vector $\boldsymbol{w}$ is orthogonal to $\boldsymbol{v}_j$ for $1 \le j \le i$. When $z_i = \infty$,
7: we replace the matrix $(A - z_i I)^{-1}$ by A.
8: **end for**

We assume that the coefficients $h_{i+1,i}$ determined in (2) are positive for $i = 1, 2, \ldots, m$. This is the generic situation. Otherwise the rational Arnoldi process is said to break down and the computations may simplify. We will not dwell on this rare situation. Define the matrices $V_m = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m] \in \mathbb{C}^{n\times m}$ and $V_{m+1} = [V_m, \boldsymbol{v}_{m+1}] \in \mathbb{C}^{n\times(m+1)}$, as well as the upper Hessenberg matrix $H_{m+1,m} = [h_{j,i}] \in \mathbb{C}^{(m+1)\times m}$ made up of the coefficients $h_{j,i}$ defined by Algorithm 1; the entries below the subdiagonal of $H_{m+1,m}$ vanish. The columns of V_m form an orthonormal basis for the rational Krylov subspace

$$\mathrm{span}\{\boldsymbol{v}, (A - z_1 I)^{-1}\boldsymbol{v}, (A - z_2 I)^{-1}(A - z_1 I)^{-1}\boldsymbol{v}, \ldots, \prod_{i=1}^{m-1}(A - z_i I)^{-1}\boldsymbol{v}\}. \tag{3}$$

When $z_i = \infty$, the factor $(A - z_i I)^{-1}$ is replaced by A.

In the special case when all shifts z_i are the same, the recursions (2) simplify to those of the standard Arnoldi process applied to the matrix $(A - z_\ell I)^{-1}$. The recursion formulas then can be expressed in the form

$$(A - z_\ell I)^{-1} V_m = V_{m+1} H_{m+1,m}. \tag{4}$$

Ruhe [23] has shown how one can determine a decomposition of the form (4) also when not all shifts z_i are equal. This is described by Algorithm 2. Each shift z_i is applied repeatedly (lines 15–24) until a stopping criterion is satisfied. Then a new shift is selected and the matrices $H_{m+1,m}$ and V_{m+1} are updated (lines 11–13). The matrix $H_{m+1,m}$ determined by this algorithm generally is not upper Hessenberg and differs from the upper Hessenberg matrix with entries $h_{j,i}$ generated by Algorithm 1. The application of Ruhe's updating method is most attractive when many consecutive shifts are the same, because only lines 15–24 have to be executed when the present shift is the same as the previous one. We remark that the transformation of $H_{m+1,m}$ in line 13 of Algorithm 2 is not orthogonal and this may lead to reduced accuracy. Another difficulty with Algorithm 2 is that using the same pole, say z_i, repeatedly without applying other poles z_k, $k \neq i$, in between may require the solution of severely ill-conditioned intermediate problems. This is illustrated by the following simple example.

Example 1.1. Let $A \in \mathbb{C}^{n\times n}$, $n \ge 21$, be a Hermitian matrix with its spectrum containing the points ± 1, and let $\boldsymbol{b} \in \mathbb{C}^n\backslash\{\mathbf{0}\}$. Define the poles $z_1 = 1 + \delta$ and $z_2 = -1 - \delta$ with the constant

DOI: 10.1002/nla

Algorithm 2 Ruhe's updating method for the rational Arnoldi process

Input: $A \in \mathbb{C}^{n\times n}$, initial vector $\boldsymbol{v} \in \mathbb{C}^n \backslash \{\boldsymbol{0}\}$, vector of finite shifts
$\boldsymbol{z} = [z_1, \ldots, z_\ell]^T \in \mathbb{C}^\ell$, convergence test, breakdown tolerance τ
Output: $H_{m+1,m} \in \mathbb{C}^{(m+1)\times m}$, $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$ with orthonormal
columns that span the rational Krylov subspace
$\boldsymbol{v}_1 := \boldsymbol{v}/\|\boldsymbol{v}\|$, $\boldsymbol{w} := (A - z_1 I)^{-1}\boldsymbol{v}_1$
$h_{11} := \langle \boldsymbol{v}_1, \boldsymbol{w}\rangle$, $\boldsymbol{w} := \boldsymbol{w} - h_{11}\boldsymbol{v}_1$
$h_{21} := \|\boldsymbol{w}\|$, $\boldsymbol{v}_2 := \boldsymbol{w}/h_{21}$
$m := 0$
for $i = 1, \ldots, \ell$
  if $i > 1$
    factor $Q_{m+1}R_{m+1,m} := I_{m+1,m} + (z_{i-1} - z_i)H_{m+1,m}$
    $V_{m+1} := V_{m+1}Q_{m+1}$
    $H_{m+1,m} := Q_{m+1}^T H_{m+1,m} R_{m,m}^{-1}$
  end if
  repeat
    $m := m + 1$
    $\boldsymbol{w} := (A - z_i I)^{-1}\boldsymbol{v}_m$
    for $j = 1, \ldots, m$
      $h_{j,m} := \langle \boldsymbol{v}_j, \boldsymbol{w}\rangle$, $\boldsymbol{w} := \boldsymbol{w} - h_{j,m}\boldsymbol{v}_j$
    end for
    $h_{m+1,m} := \|\boldsymbol{w}\|$
    if $h_{m+1,m} < \tau$ **then** exit for breakdown **end**
    $\boldsymbol{v}_{m+1} := \boldsymbol{w}/h_{m+1,m}$
  until convergence
end for

$\delta > 0$ close to zero. Assume that we would like to use the rational Krylov subspace

$$\begin{aligned}\mathrm{span}\{&\boldsymbol{b}, (A - z_1 I)^{-1}\boldsymbol{b}, (A - z_1 I)^{-2}\boldsymbol{b}, \ldots, (A - z_1 I)^{-10}\boldsymbol{b},\\ &(A - z_2 I)^{-1}\boldsymbol{b}, (A - z_2 I)^{-2}\boldsymbol{b}, \ldots, (A - z_2 I)^{-10}\boldsymbol{b}\}.\end{aligned}$$

The vector $\boldsymbol{b}$ is chosen so that no breakdown occurs, i.e., this space is of dimension 21. Let us first use the pole z_1 to determine a basis for

$$\mathrm{span}\{\boldsymbol{b}, (A - z_1 I)^{-1}\boldsymbol{b}, (A - z_1 I)^{-2}\boldsymbol{b}, \ldots, (A - z_1 I)^{-10}\boldsymbol{b}\} \tag{5}$$

before applying the pole z_2. The eigenvalues $1 - z_1 = -\delta$ and $-1 - z_1 = -2 - \delta$ of $A - z_1 I$ give the lower bound $(2/\delta + 1)^{10}$ for the condition number of $(A - z_1 I)^{10}$. This bound is very large when $\delta > 0$ is "tiny." Therefore, severe propoagation of round-off errors introduced during the computations may take place when seeking to compute an orthonormal basis for the space (5).

The rational Krylov method proposed in this paper applies the poles in a cyclic fashion $z_1, z_2, z_1, z_2, \ldots$. This circumvents the need to carry out computations with very ill-conditioned matrices in the present example. We note that the lower bound for the condition number of the matrix $((A - z_1 I)(A - z_2 I))^{10}$ determined by the eigenvalues ± 1 of A is one and, hence, much smaller than $(2/\delta + 1)^{10}$ for $\delta > 0$ tiny.

While this example is very simple, it nevertheless indicates that orthonormal rational Krylov subspace bases determined by Algorithm 2 may be severely contaminated by propagated round-off errors. Computed examples in Section 4 show that this indeed can be the case. □

DOI: 10.1002/nla

Algorithm 3 A cyclic rational Krylov subspace method

Input: $A \in \mathbb{C}^{n\times n}$, initial vector $\boldsymbol{v} \in \mathbb{C}^n \backslash \{\mathbf{0}\}$, vector of finite shifts
$\boldsymbol{z} = [z_1, \ldots, z_\ell]^T \in \mathbb{C}^\ell$, convergence test, breakdown tolerance τ
Output: $H_{m+1,m} \in \mathbb{C}^{(m+1)\times m}$, $K_{m+1,m} \in \mathbb{C}^{(m+1)\times m}$, $V_{m+1} \in \mathbb{C}^{n\times(m+1)}$ with
orthonormal columns that span the rational Krylov subspace
$\boldsymbol{v}_1 := \boldsymbol{v}/\|\boldsymbol{v}\|$
$m := 0$
repeat
  for $i = 1, \ldots, \ell$
    $m := m + 1$
    $\boldsymbol{w} := (A - z_i I)^{-1}\boldsymbol{v}_m$
    for $j = 1, \ldots, m$
      $h_{j,m} := \langle \boldsymbol{v}_j, \boldsymbol{w}\rangle$, $\boldsymbol{w} := \boldsymbol{w} - h_{j,m}\boldsymbol{v}_j$
    end for
    $h_{m+1,m} := \|\boldsymbol{w}\|$
    if $h_{m+1,m} < \tau$ **then** exit for breakdown **end**
    $\boldsymbol{v}_{m+1} := \boldsymbol{w}/h_{m+1,m}$
    for $j = 1, \ldots, m+1$
      $k_{j,m} := h_{j,m} z_i$
    end for
    $k_{m,m} := k_{m,m} + 1$
  end for
until convergence

Ruhe [21] also proposed another implementation of the rational Arnoldi process, described by Algorithm 3; it is referred to as Algorithm RKS in [21]. The algorithm determines a decomposition of the form

$$AV_{m+1}H_{m+1,m} = V_{m+1}K_{m+1,m}, \tag{6}$$

where the orthonormal columns of the matrix $V_{m+1} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{m+1}] \in \mathbb{C}^{n\times(m+1)}$ form a basis of the space (3) with m replaced by $m+1$, and the matrices $H_{m+1,m} = [h_{j,i}] \in \mathbb{C}^{(m+1)\times m}$ and $K_{m+1,m} = [k_{j,i}] \in \mathbb{C}^{(m+1)\times m}$ made up of the entries $h_{j,i}$ and $k_{j,i}$, respectively, generated by the algorithm are upper Hessenberg. Application of m suitably chosen Givens transformations to $H_{m+1,m}$ from the left (and to V_{m+1} from the right) transform $H_{m+1,m}$ to an upper triangular matrix $R_{m,m} \in \mathbb{C}^{m\times m}$ with an appended last row of zeros (and a new matrix $W_{m+1} = [W_m, \boldsymbol{w}_{m+1}] \in \mathbb{C}^{n\times(m+1)}$ with orthonormal columns). We obtain a decomposition of the form

$$AW_m R_{m,m} = W_{m+1}M_{m+1,m}, \tag{7}$$

where $M_{m+1,m} \in \mathbb{C}^{(m+1)\times m}$ is a dense matrix; see Ruhe [21] for details. Assuming that no breakdown occurs during the execution of Algorithm 3, the matrix $H_{m+1,m}$ in (6) is of full rank, and the matrix $R_{m,m}$ in (7) therefore is invertible. We obtain the decomposition

$$AW_m = W_{m+1}M_{m+1,m}R_{m,m}^{-1} = W_m N_{m,m} + \boldsymbol{w}_{m+1}\boldsymbol{g}_m^* \tag{8}$$

for a suitable matrix $N_{m,m} \in \mathbb{C}^{m\times m}$ and vector $\boldsymbol{g}_m \in \mathbb{C}^m$. A further, closely related, decomposition is described in [3]. We note that the computation of the matrix $N_{m,m}$ in the decomposition (8) requires the solution of $m+1$ linear systems of equations with the matrix $R_{m,m}^T$ (one system for each row of $M_{m+1,m}$), and if the matrix $R_{m,m}$ is ill-conditioned, then the computed results may be sensitive to round-off errors introduced during the computations.

DOI: 10.1002/nla

The decompositions determined by Algorithms 2 and 3 can be applied to computing a few selected eigenvalues or partial pseudospectra of a large matrix A, as well as to approximating matrix functions of the form

$$f(A)\boldsymbol{v}. \tag{9}$$

The former applications are described by Ruhe [21, 23] and the latter one by Güttel [11, Chapter 5]. Here we assume that the function f is such that $f(A)$ is well defined; see, e.g., [10, 14] for several ways of defining matrix functions.

The evaluation of matrix functions of the form (9) is of interest for entire functions such as

$$f(t) = \exp(t), \quad f(t) = (1 - \exp(t))/t, \quad f(t) = \cos(t), \quad f(t) = \sin(t),$$

with applications to the solution of ordinary and partial differential equations [4, 5, 8, 13, 15, 24] and network analysis [6, 7]. Other functions f of interest include $f(t) = \sqrt{t}$, with application to the solution of systems of stochastic differential equations [1], and $f(t) = \ln(t)$. Higham [14] describes many methods for the evaluation of $f(A)$ that can be applied when the matrix A is small enough to compute a factorization. When A is large, expressions (9) are commonly approximated by first reducing the matrix A to a small matrix by application of a fairly small number of steps of the standard or rational Arnoldi or Lanczos processes; see, e.g., [2, 3, 4, 5, 8, 11, 12, 15, 16, 17]. The techniques described in [14] can then be applied to evaluate the reduced problem so obtained.

The decomposition determined by applications of m steps of the standard Arnoldi process to the matrix A with initial vector $\boldsymbol{v}$ can be applied to find a polynomial approximation of f in (9) of degree at most $m-1$. Similarly, decompositions determined by Algorithms 2 and 3 can be used to compute approximations of f by rational functions with poles z_i. Rational functions may yield more accurate approximations of (9) than polynomials. However, the evaluation of the rational approximants requires the use of nonorthogonal transformations and this may lead to reduced accuracy due to severely propagated round-off errors introduced during the computations.

It is the purpose of this paper to describe a new approach to generating an orthonormal basis for a rational Krylov subspace. This approach determines a matrix analogous to $H_{m+1,m}$ in (4). The matrix generated is upper block Hessenberg. The block size, and thereby the lower bandwidth, are small when there are only few distinct poles z_i. We are particularly interested in the situation when the sequence of poles $z_1, z_2, \ldots, z_\ell$ is a cyclic enumeration of a few distinct poles. Recently, Simoncini [25] presented a rational Arnoldi method that is related to our scheme in the special situation when every other pole z_{2i-1} is at the origin and every other pole z_{2i} is at infinity.

This paper is organized as follows. Section 2 describes a Stieltjes-type procedure for the orthonormalization of a sequence of rational functions with a few distinct poles. We focus on the structure of the recursion relations. Section 3 discusses a rational Arnoldi process that is the linear algebra analogue of the Stieltjes-type procedure of Section 2. We provide an algorithm, Algorithm 5, and discuss implementation details. In particular, we show that the subdiagonal part of the matrix with the recursion coefficients has a block structure. Section 4 presents computed examples that compare the performance of Algorithms 2 and 3 with Algorithm 5 of Section 3. Applications to the approximation of matrix functions (9), the computation of a few eigenvalues of a large matrix, and the determination of ε-pseudospectra of a large matrix in a selected region of the complex plane are described. Concluding remarks are provided in Section 5.

2. A STIELTJES-TYPE PROCEDURE FOR THE ORTHONORMALIZATION OF RATIONAL FUNCTIONS

The Stieltjes procedure is a common approach to determine a sequence of orthonormal polynomials associated with an inner product defined on (part of) the real axis; see, e.g., Gautschi [9]. This section describes the recursion formulas of a Stieltjes-type procedure for determining a sequence of orthonormal rational functions with a few fixed distinct poles. We proceed similarly as in [19], where the situation when the inner product is defined by a nonnegative measure on the real axis and the poles are real or appear in complex conjugate pairs is considered. Here we consider a more

DOI: 10.1002/nla

general measure. Thus, let $d\mu$ be a nonnegative measure with infinitely many points of support in $\mathbb{C}$. In this section, we use the inner product

$$\langle f, g\rangle_\mu = \int \overline{f(z)} g(z) d\mu(z) \tag{10}$$

and the induced norm $\|f\|_\mu = \langle f, f\rangle_\mu^{1/2}$. The measure is assumed to be such that the inner product exists for all polynomials f and g. Let $\mathcal{P}$ denote the space of all polynomials and introduce the linear space of rational functions with distinct finite poles $z_1, z_2, z_3, \ldots$ that are bounded away from the support of $d\mu$,

$$\mathcal{Q} = \text{span}\left\{ \frac{1}{(z - z_k)^j} \ : \ j \in \mathbb{N}, \ \ z_k \in \mathbb{C}, \ \ |z_k| < \infty \right\},$$

where as usual $\mathbb{N}$ denotes the set of positive integers. Define the linear space

$$\mathcal{P} + \mathcal{Q} = \text{span}\left\{ 1, \ z^j, \ \frac{1}{(z - z_k)^j} \ : \ j \in \mathbb{N}, \ z_k \in \mathbb{C}, \ |z_k| < \infty \right\}.$$

Let $\Psi = \{\psi_0, \psi_1, \psi_2, \ldots\}$ denote an elementary basis for this space, i.e., $\psi_0(z) = 1$ and each $\psi_i(z)$ for $i = 1, 2, \ldots$ is one of the functions

$$z^j, \quad \frac{1}{(z - z_k)^j},$$

for some positive integers j and k. Application of the Gram–Schmidt process with respect to the inner product (10) to the basis Ψ yields a basis of orthonormal rational functions with the prescribed finite poles z_k,

$$\Phi = \{\phi_0, \phi_1, \phi_2, \ldots\}.$$

The recursion relations for the ϕ_j depend on the ordering of the basis functions ψ_j of Ψ. We write $\psi_j \prec \psi_k$ if the basis function ψ_j comes before ψ_k. The ordering of the basis Ψ is said to be *natural* if it satisfies the following conditions:

(i) $z^j \prec z^{j+1}$ for all integers $j \geq 0$,

(ii) $\dfrac{1}{(z - z_k)^j} \prec \dfrac{1}{(z - z_k)^{j+1}}$ for all positive integers j and every pole z_k.

We tacitly assume that there are no missing powers, i.e., if z^{j+1} is an elementary basis function for some $j \in \mathbb{N}$, then so is z^j. Similarly, if $(z - z_k)^{-j-1}$ is an elementary basis function, then so is $(z - z_k)^{-j}$.

We now will show that orthonormal rational functions with prescribed poles corresponding to any natural ordering satisfy two types of recurrence relations.

Theorem 1
Let the basis $\Psi = \{\psi_0, \psi_1, \psi_2, \ldots\}$ satisfy the requirements of natural ordering and assume that every sequence of m_1 consecutive basis functions $\psi_k, \psi_{k+1}, \ldots, \psi_{k+m_1-1}$ contains at least one power z^j. Then the orthonormal rational functions $\phi_0, \phi_1, \phi_2, \ldots$ with prescribed poles satisfy a recurrence relation of the form

$$z\phi_k(z) = \sum_{i=0}^{k+m_1} c_{k,i}\phi_i(z), \qquad k = 0, 1, 2, \ldots . \tag{11}$$

Proof
It follows from the ordering that

$$z\phi_k(z) \in \text{span}\{\phi_0, \phi_1, \ldots, \phi_{k+m_1-1}, \phi_{k+m_1}\}, \qquad k = 0, 1, 2, \ldots .$$

This shows (11). □

DOI: 10.1002/nla

Remark 2
If the set $\mathcal{Q}$ is empty, i.e., if all elementary basis functions ψ_j are monomials, then $m_1 = 1$, and we obtain the standard Arnoldi process.

Remark 3
If the set $\mathcal{Q}$ is nonempty, i.e., some of the elementary basis functions ψ_j have finite poles, then we can order the basis Ψ to get the smallest possible value of m_1. If every nonmonomial basis function ψ_j is followed by at least one monomial basis function, then $m_1 = 2$. This is the smallest possible m_1-value when $\mathcal{Q}$ is nonempty.

Theorem 4
Let the basis $\Psi = \{\psi_0, \psi_1, \psi_2, \dots\}$ satisfy the requirements of natural ordering and let z_j be a finite pole of this basis. Assume that every sequence of m_2 consecutive basis functions $\psi_k, \psi_{k+1}, \dots, \psi_{k+m_2-1}$ contains at least one power $(z - z_j)^{-t}$, $t \geq 1$. Then the basis of orthonormal rational functions $\phi_0, \phi_1, \phi_2, \dots$ with prescribed poles satisfy a recurrence relation of the form

$$\frac{1}{z - z_j}\,\phi_k(z) = \sum_{i=0}^{k+m_2} c_{k,i}^{(j)} \phi_i(z), \qquad k = 0, 1, 2, \dots .$$

Proof
The result can be shown similarly as Theorem 1. □

Algorithm 4 below is a Stieltjes-type procedure for generating an orthonormal basis $\phi_0, \phi_1, \phi_2, \dots$ for a subspace spanned by $\Psi = \{\psi_0, \psi_1, \psi_2, \dots\}$. We say that the orthonormal function ϕ_m "contains" the elementary basis function ψ_j if ϕ_m can be written as a linear combination with nonvanishing coefficients of elementary basis functions, one of which is ψ_j.

3. A RATIONAL ARNOLDI PROCESS

This section discusses in detail a new rational Arnoldi process that is analogous to the Stieltjes-type procedure of Section 2. The rational Arnoldi process determines an orthonormal basis for a rational Krylov subspace of the form (3). Algorithm 5 below describes an implementation. The algorithm computes an orthonormal basis for a rational Krylov subspace with ℓ distinct finite poles $z_1, z_2, \dots, z_\ell \in \mathbb{C}$ and a pole at infinity. In the following, for notational simplicity, we describe the algorithm for $\ell = 2$. Thus, given a matrix $A \in \mathbb{C}^{n\times n}$, an initial vector $\boldsymbol{v} \in \mathbb{C}^n \backslash \{\mathbf{0}\}$, and two finite distinct poles $z_1, z_2 \in \mathbb{C}$, Algorithm 5 computes a decomposition of the form

$$AV_m = V_{m+1} H_{m+1,m}, \tag{12}$$

in which the columns of the matrix $V_{m+1} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_{m+1}] \in \mathbb{C}^{n\times(m+1)}$, with $\boldsymbol{v}_1 = \boldsymbol{v}/\|\boldsymbol{v}\|$, make up an orthonormal basis for the rational Krylov subspace

$$\operatorname{span}\{\boldsymbol{v}, A\boldsymbol{v}, (A - z_1 I)^{-1}\boldsymbol{v}, (A - z_2 I)^{-1}\boldsymbol{v}, A^2 b, (A - z_1 I)^{-2}\boldsymbol{v}, (A - z_2 I)^{-2}\boldsymbol{v}, \dots\} \tag{13}$$

determined by the product of elementary rational functions in A with poles in the set $\{\infty, z_1, z_2\}$ and by the vector $\boldsymbol{v}$. The nontrivial entries of the Hessenberg-type matrix $H_{m+1,m} \in \mathbb{C}^{(m+1)\times m}$ are recursion coefficients for the orthonormal rational functions in A that form a basis for (13). We will for simplicity assume that a decomposition of the form (12) exists, i.e., that no rational Krylov subspace determined during the computation of the decomposition (12) is an invariant subspace of A. The latter situation is rare and in many situations simplifies the application of the computed decomposition. We will therefore not discuss this situation further.

In the remainder of this section, we let $\boldsymbol{m}_i$ denote the ith column of the matrix M with entries $m_{i,j}$, M_i stand for the submatrix made up of first i columns of M, and $M_{i,j}$ denote the leading principal $i \times j$ submatrix of M. Similarly, w_i stands for the ith element of the vector $\boldsymbol{w}$. Finally, the range of the matrix M is denoted by $\mathcal{R}(M)$.

DOI: 10.1002/nla

Algorithm 4 A Stieltjes-type procedure for the orthonormalization of rational functions.

1: **Input:** basis $\{\psi_0, \psi_1, \ldots, \psi_n\}$
2: **Output:** orthonormal basis $\{\phi_0, \phi_1, \ldots, \phi_n\}$
3: $\phi_0 := \psi_0/||\psi_0||_\mu$
4: **for** $k = 1, \ldots, n$
5: **if** $\psi_k(z) = z^s$ for some $s \in \mathbb{N}$ **then**
6: **if** $s = 1$ **then**
7: $m := 0$
8: **else**
9: let $m \leq k-1$ be an integer such that ϕ_m contains z^{s-1}
10: **end if**
11: $g(z) := z\phi_m(z)$
12: **for** $i = 0, \ldots, k-1$
13: $c_{k-1,i} := \langle g, \phi_i \rangle_\mu$, $g := g - c_{k-1,i}\phi_i$
14: **end if**
15: $\phi_k := g/||g||_\mu$
16: **end if**
17: **if** $\psi_k(z) = (z - z_j)^{-s}$ for some $s \in \mathbb{N}$ **then**
18: **if** $s = 1$ **then**
19: $m := 0$
20: **else**
21: let $m \leq k-1$ be an integer such that ϕ_m contains $(z - z_j)^{-(s-1)}$
22: **end if**
23: $g(z) := (z - z_j)^{-1}\phi_m(z)$
24: **for** $i = 0, \ldots, k-1$
25: $c_{k-1,i} := \langle g, \phi_i \rangle_\mu$, $g := g - c_{k-1,i}\phi_i$
26: **end for**
27: $\phi_k := g/||g||_\mu$
28: **end if**
29: **end for**

Algorithm 5 is an application of Theorems 1 and 4. We comment on some details in the case $\ell = 2$ and the poles z_1 and z_2 are distinct.

Lines 6–20 for $q = 0$: the algorithm generates the matrix $V_4 = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4]$; its columns form an orthonormal basis for

$$\operatorname{span}\{\boldsymbol{v}, A\boldsymbol{v}, (A - z_1 I)^{-1}\boldsymbol{v}, (A - z_2 I)^{-1}\boldsymbol{v}\}.$$

Also the matrix $H_{2,1} = \left[\begin{smallmatrix} h_{1,1} \\ h_{1,2} \end{smallmatrix}\right]$ is determined. It satisfies

$$AV_1 = V_2 H_{2,1}. \tag{14}$$

The last two columns of V_4 are not part of this decomposition. The column $\boldsymbol{v}_3$ is generated by orthogonalizing $(A - z_1 I)^{-1}\boldsymbol{v}$ against $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. Therefore $\boldsymbol{v}_3$ is a linear combination of $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $(A - z_1 I)^{-1}\boldsymbol{v}$. We have

$$\boldsymbol{v}_3 = V_2 \left[\begin{smallmatrix} \gamma_1 \\ \gamma_2 \end{smallmatrix}\right] + \gamma_3 (A - z_1 I)^{-1}\boldsymbol{v} \tag{15}$$

DOI: 10.1002/nla

Algorithm 5 A rational Arnoldi method.

```
1:  Input: A ∈ C^{n×n}, initial vector v ∈ C^n\{0}, vector of finite shifts
2:        z = [z_1, ..., z_ℓ]^T ∈ C^ℓ, convergence test, breakdown tolerance τ
3:  Output: H_{m+1,m} ∈ C^{(m+1)×m} Hessenberg-type matrix (m = p(ℓ+1)+1),
4:        V_{m+1} ∈ C^{n×(m+1)} with orthonormal columns that span the
5:        rational Krylov subspace
6:  v_1 := v/‖v‖, w := Av_1
7:  h_11 := ⟨v_1, w⟩, w := w − h_11 v_1
8:  h_21 := ‖w‖, v_2 := w/h_21
9:  q := 0, k := 2
10: repeat
11:    for i = 1, ..., ℓ  // augment rational Krylov subspace
12:       w := (A − z_i I)^{-1} v_k  // solve linear system of equations for w
13:       for j = 1, ..., k + i − 1
14:          w := w − ⟨v_j, w⟩ v_j
15:       end for
16:       ω := ‖w‖
17:       if ω < τ then exit for breakdown end
18:       v_{k+i} := w/ω
19:    end for
20:    w := Av_k  // augment rational Krylov subspace
21:    for j = 1, ..., k + ℓ
22:       h_jk := ⟨v_j, w⟩, w := w − h_jk v_j
23:    end for
24:    h_{k+ℓ+1,k} := ‖w‖
25:    if h_{k+ℓ+1,k} < τ then exit for breakdown end
26:    v_{k+ℓ+1} := w/h_{k+ℓ+1,k}
27:    for i = 1, ..., ℓ  // update the matrix H
28:       w := Av_{k+i}
29:       for j = 1, ..., k + ℓ + 1
30:          h_{j,k+i} := ⟨v_j, w⟩, w := w − h_{j,k+i} v_j
31:       end for
32:    end for
33:    q := q + 1, k := q(ℓ + 1) + 2
34: until convergence
```

for certain coefficients γ_i. We assume here the generic situation that $(A - z_1 I)^{-1}\boldsymbol{v} \notin \operatorname{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2\}$. Similarly, $\boldsymbol{v}_4$ is determined by orthogonalizing $(A - z_2 I)^{-1}\boldsymbol{v}$ against $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$ and normalizing the residual obtained.

Lines 21–34 for $q = 0$: We discuss the computations during the first pass through the q-loop, i.e., when $q = 0$ and $k = 2$. The algorithm augments the matrix V_4 with one column to obtain an orthonormal basis for the rational Krylov subspace

$$\operatorname{span}\{\boldsymbol{v}, A\boldsymbol{v}, (A - z_1 I)^{-1}\boldsymbol{v}, (A - z_2 I)^{-1}\boldsymbol{v}, A^2\boldsymbol{v}\}$$

and recursion coefficients stored in the matrix $H_{5,2}$. We carry out the following computations:

DOI: 10.1002/nla

a) Evaluate $A\boldsymbol{v}_2$ and orthogonalize against the columns of V_4. Assign the normalized residual to $\boldsymbol{v}_5$. The coefficients generated during orthogonalization and normalization are stored in the second column of the matrix $H_{5,2}$. This gives the decomposition

$$AV_2 = V_5 H_{5,2}, \qquad H_{5,2} = \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \\ 0 & h_{3,2} \\ 0 & h_{4,2} \\ 0 & h_{5,2} \end{bmatrix}.$$

Thus,

$$\mathcal{R}(AV_2) \subset \mathcal{R}(V_5) = \mathrm{span}\{\boldsymbol{v}, A\boldsymbol{v}, (A - z_1 I)^{-1}\boldsymbol{v}, (A - z_2 I)^{-1}\boldsymbol{v}, A^2\boldsymbol{v}\}.$$

b) Evaluate $A\boldsymbol{v}_3$ and orthogonalize against the columns of V_5. The following discussion shows that the orthogonalization process will break down and, therefore, no new columns of the matrix V_5 will be generated. It follows from (15) that

$$A\boldsymbol{v}_3 = AV_2 \left[\begin{smallmatrix} \gamma_1 \\ \gamma_2 \end{smallmatrix}\right] + \gamma_3 A(A - z_1 I)^{-1}\boldsymbol{v}, \tag{16}$$

where, by the discussion above, the first term in the right-hand side lives in $\mathcal{R}(V_5)$. The vector $A(A - z_1 I)^{-1}\boldsymbol{v}$ is a linear combination of $\boldsymbol{v}$ and $(A - z_1 I)^{-1}\boldsymbol{v}$. Therefore the second term in the right-hand side of (16) lives in $\mathcal{R}(V_3)$. In particular, $A\boldsymbol{v}_3 \in \mathcal{R}(V_5)$. A similar argument shows that $A\boldsymbol{v}_4 \in \mathcal{R}(V_5)$. When we arrive at line 33 with $q = 0$ and $k = 2$, we have determined the decomposition

$$AV_4 = V_5 H_{5,4}, \qquad H_{5,4} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} \\ 0 & h_{3,2} & h_{3,3} & h_{3,4} \\ 0 & h_{4,2} & h_{4,3} & h_{4,4} \\ 0 & h_{5,2} & h_{5,3} & h_{5,4} \end{bmatrix}. \tag{17}$$

Lines 10–34 for $q = 1$: We briefly comment on the second pass through the q-loop of Algorithm 5. Lines 11–20 for $q = 1$ give the last columns, $\boldsymbol{v}_6$ and $\boldsymbol{v}_7$, of the matrix V_7. We then expand the decomposition (17) to obtain

$$AV_5 = V_8 H_{8,5}, \qquad H_{8,5} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} & h_{1,5} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} & h_{2,5} \\ 0 & h_{3,2} & h_{3,3} & h_{3,4} & h_{3,5} \\ 0 & h_{4,2} & h_{4,3} & h_{4,4} & h_{4,5} \\ 0 & h_{5,2} & h_{5,3} & h_{5,4} & h_{5,5} \\ 0 & 0 & 0 & 0 & h_{6,5} \\ 0 & 0 & 0 & 0 & h_{7,5} \\ 0 & 0 & 0 & 0 & h_{8,5} \end{bmatrix}$$

and finally arrive at

$$AV_7 = V_8 H_{8,7}, \qquad H_{8,7} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & h_{1,4} & h_{1,5} & h_{1,6} & h_{1,7} \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} & h_{2,5} & h_{2,6} & h_{2,7} \\ 0 & h_{3,2} & h_{3,3} & h_{3,4} & h_{3,5} & h_{3,6} & h_{3,7} \\ 0 & h_{4,2} & h_{4,3} & h_{4,4} & h_{4,5} & h_{4,6} & h_{4,7} \\ 0 & h_{5,2} & h_{5,3} & h_{5,4} & h_{5,5} & h_{5,6} & h_{5,7} \\ 0 & 0 & 0 & 0 & h_{6,5} & h_{6,6} & h_{6,7} \\ 0 & 0 & 0 & 0 & h_{7,5} & h_{7,6} & h_{7,7} \\ 0 & 0 & 0 & 0 & h_{8,5} & h_{8,6} & h_{8,7} \end{bmatrix}. \tag{18}$$

It is straightforward to adapt Algorithm 5 to the situation when there is no pole at infinity. We may, for instance, replace matrix-vector product evaluations with A by the solution of a system of equations with the matrix, say, $A - z_1 I$. In the computed examples, we will use an approach that is even simpler. Consider the situation when there are two finite poles z_1 and z_2, and no pole at infinity. We then make a change of variables that moves the pole z_2 to infinity, let $m = 2p + 1$, and execute the two loops "**for** $i = 1, 2$" only for $i = 1$. Consider

$$\begin{aligned} (A - z_2 I)^{-1} &= ((A - z_1 I) - (z_2 - z_1)I)^{-1} \\ &= \frac{1}{z_1 - z_2}(A - z_1 I)^{-1}\left((A - z_1 I)^{-1} - \frac{1}{z_2 - z_1} I\right)^{-1} \end{aligned}$$

DOI: 10.1002/nla

and let

$$S = (A - z_1 I)^{-1} \tag{19}$$

and $\gamma = \frac{1}{z_2 - z_1}$. Then

$$(A - z_2 I)^{-1} = (-\gamma) S (S - \gamma I)^{-1}.$$

This change of variables allows us to apply Algorithm 5 to the situation when the rational Krylov subspace is defined by two finite poles and no pole at infinity. Our code implements this extension of the algorithm for an arbitrary number of poles.

Algorithm 5 has to be modified when the shifts are not distinct. Thus, if $z_{i+1} = z_i$ and all other shifts are different from z_i, then line 13 of the algorithm should be $\boldsymbol{w} := (A - z_{i+1} I)^{-1} \boldsymbol{v}_{k+1}$ in order to produce an orthonormal basis. We remark that while the modification of line 13 to $\boldsymbol{w} := (A - z_i I)^{-1} \boldsymbol{v}_{k+i-1}$ would take care of this issue, this modification was found to sometimes give reduced accuracy due to considerable propagation of round-off errors in the columns of V_m. It is therefore not implemented.

4. COMPUTED EXAMPLES

This section illustrates the performance of Algorithm 5 when applied to the approximation of matrix functions, the determination of a few selected eigenvalues, and the computation of pseudospectra. We compare this algorithm to Algorithms 2 and 3. All computations were carried out in MATLAB with about 15 significant decimal digits.[†]

The reported results are obtained with reorthogonalization. We found Algorithm 2 for some examples, such as in the application to pseudospectrum computations reported in Subsection 4.3, to give very poor accuracy without reorthogonalization. The accuracy achieved with Algorithm 5 typically improved somewhat with reorthogonalization, but not dramatically so. In our experience reorthogonalization is more important for Algorithm 2 than for Algorithm 5. Nevertheless, to show the best possible performances of all algorithms, we carried out reorthogonalization in all examples.

4.1. Approximation of matrix functions

We would like to approximate expressions of the form (9) for $A \in \mathbb{C}^{n \times n}$, $\boldsymbol{v} \in \mathbb{C}^n$ a unit vector, and a function f. Let $H_{m,m} \in \mathbb{C}^{m \times m}$ be the leading submatrix of the matrix $H_{m+1,m}$ in (12), where $\boldsymbol{v}_1 = \boldsymbol{v}$. Then we obtain the approximation

$$f(A)\boldsymbol{v} \approx V_m f(H_{m,m}) \boldsymbol{e}_1, \tag{20}$$

where $\boldsymbol{e}_1 = [1, 0, \ldots, 0]^T$ is the first axis vector. When all shifts z_i are infinite, the columns of the matrix V_m form an orthonormal basis for a standard Krylov subspace and the right-hand side of (20) is a polynomial approximation of the left-hand side. If some or all shifts are bounded, then the columns of V_m form an orthonormal basis for a rational Krylov subspace and the right-hand side of (20) is a rational approximation of the left-hand side, where the rational function has poles at the shifts z_i. Rational approximants can give faster convergence and a smaller error than polynomial approximations when f has singularities close to the spectrum of A and the shifts are chosen at or close to these singularities, but also when f is an entire function; see [2, 3, 4, 11, 12, 17] for analyses and illustrations.

We construct a test matrix $A \in \mathbb{C}^{1000 \times 1000}$ by introducing 4 clusters, each of which has 250 "random" points $\lambda_j^{(k)} \in \mathbb{C}$, $k = 1, 2, 3, 4$ and $j = 1, 2, \ldots, 250$, close to $\pm 5 \pm 5i$, $i = \sqrt{-1}$. Define $D_k = \mathrm{diag}[\lambda_1^{(k)}, \ldots, \lambda_{250}^{(k)}]$ and compute $A_k = U_k D_k U_k^T$ for $k = 1, 2, 3, 4$. Here U_k is a random orthogonal matrix in Hessenberg form. Finally, we form the block diagonal matrix $A = \mathrm{diag}[A_1, A_2, A_3, A_4]$ and set, for $k = 1, 2, 3$, the $(250k + 1, 250k)$-entries of A to $1.5 \cdot 10^{-8}$ in order to couple the four diagonal blocks without significantly perturbing their eigenvalues.

[†]MATLAB code used for the experiments is available from the authors upon request.

DOI: 10.1002/nla

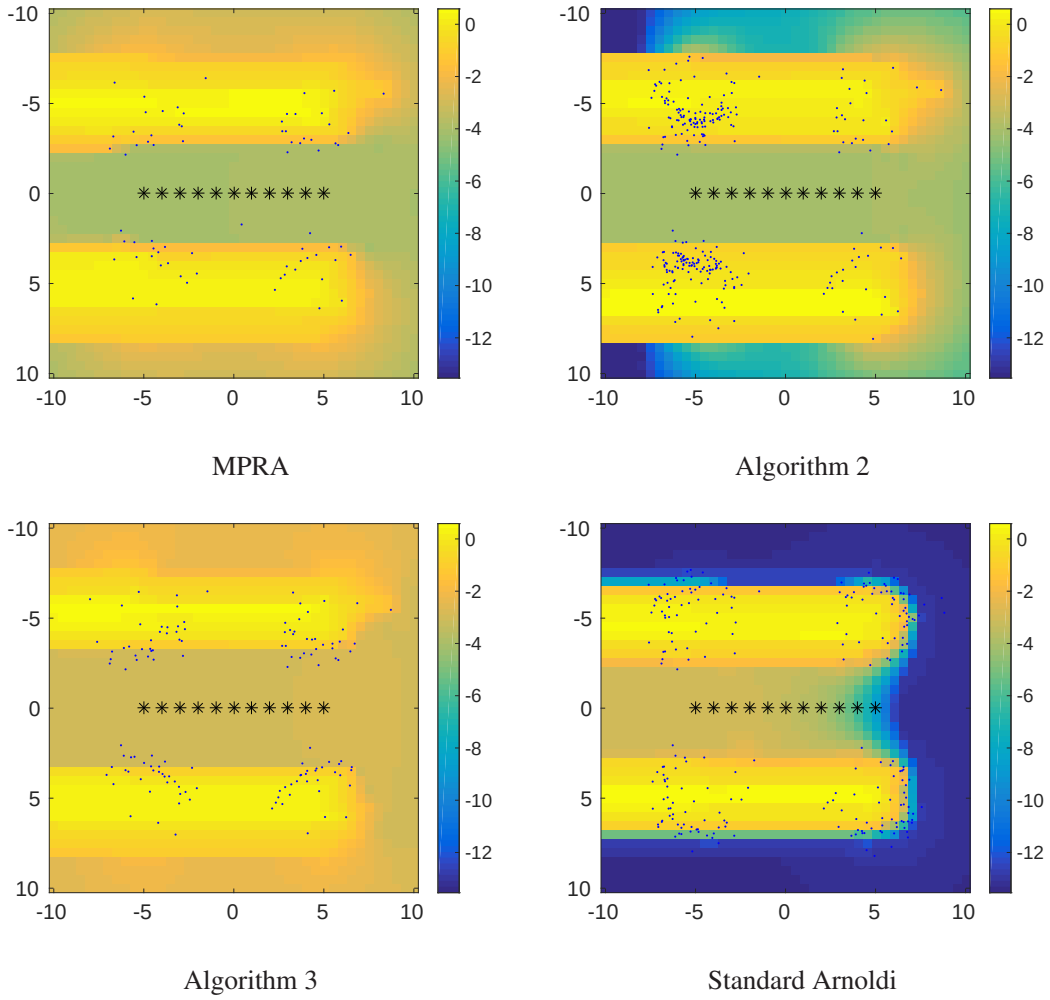


Figure 1. Relative errors in $\log_{10}$-scale for computed approximations of $\boldsymbol{F}(z) = f(zI - A)\boldsymbol{v}$ with $f(t) = \ln(t)$ measured using the Euclidean norm for MPRA, Algorithms 2 and 3, and the standard Arnoldi methods. The poles are displayed by stars. The MPRA method determined 73 eigenvalues (represented by dots) to satisfy the stopping criterion, Algorithm 2 determined 287 eigenvalues, Algorithm 3 computed 132 eigenvalues, and the standard Arnoldi method determined 238 eigenvalues. The number of eigenvalues computed equals the number of steps of the algorithm. Thus, the MPRA method requires fewer steps and less storage for the orthonormal basis generated than the other methods, and yields competitive accuracy.

We compute approximations of functions $\boldsymbol{F}(z) = f(zI - A)\boldsymbol{v}$ with

$$f(t) = \ln(t) \quad \text{or} \quad f(t) = t^{-1}, \tag{21}$$

and $\boldsymbol{v} = [1, \ldots, 1]^T/\sqrt{n}$ by the multipole rational Arnoldi (MPRA) method (Algorithm 5), by Algorithms 2 and 3 described by Ruhe [21, 23], and by the standard Arnoldi method. Here $\ln(t)$ denotes the natural logarithm with a branch cut along the negative real axis. The MPRA method delivers approximations of the form $\boldsymbol{R}_m(z) = V_m f(zI - H_{m,m})\boldsymbol{e}_1$, while Algorithm 2 gives the approximation

$$\boldsymbol{R}_m(z) = V_m f((z - z_\ell)I - H_{m,m}^{-1})V_m^T\boldsymbol{v} \tag{22}$$

(where $H_{m,m}$ differs from the matrix determined by the MPRA method), and Algorithm 3 yields the rational approximation $\boldsymbol{R}_m(z) = V_m f(zI - N_{m,m})\boldsymbol{e}_1$. We remark that the computations required for evaluating (22) are more demanding than those for the other methods. This is reflected by longer execution times, as is illustrated by Table I.

DOI: 10.1002/nla

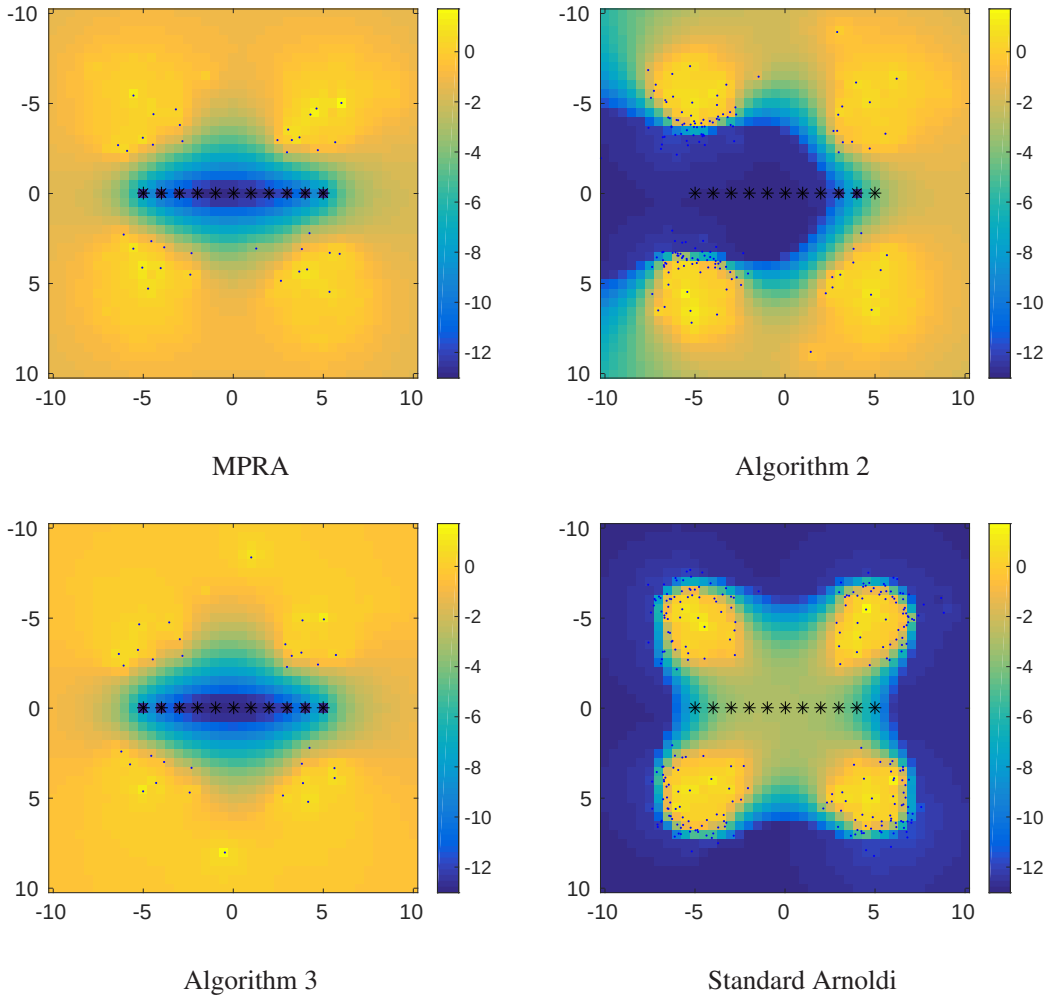


Figure 2. Relative errors in $\log_{10}$-scale for computed approximations of $\boldsymbol{F}(z) = f(zI - A)\boldsymbol{v}$ with $f(t) = t^{-1}$ measured using the Euclidean norm for MPRA, Algorithms 2 and 3, and the standard Arnoldi methods. The poles are displayed by stars. The MPRA method determined 37 eigenvalues (represented by dots) to satisfy the stopping criterion, Algorithm 2 computes 156 eigenvalues, Algorithm 3 determines 33 eigenvalues, and the standard Arnoldi method computes 270 eigenvalues. For this examples, MPRA and Algorithm 3 perform about equally well, while the other methods require many more steps.

We measure the approximation error at the grid points (with $i = \sqrt{-1}$)

$$w_{j,k} = -10 - 10i + \frac{1}{2}(j + ki), \qquad j, k = 0, 1, \ldots, 40.$$

The shifts z_j in the multipole methods are chosen in the interior of the convex hull of these grid points. Specifically, we let the shifts be equidistant points on the real interval $[-5, 5]$, i.e., $z_j = -5 + j$, $j = 0, 1, \ldots, 10$. This selection of shifts aims to produce fairly accurate approximations of eigenvalues of A in the convex hull of the grid points and of associated eigenvectors.

Figures 1, 2, 3, and 4 display results for the four methods in our comparison. The computation with each method is terminated when two consecutive approximations $\boldsymbol{R}_m$ and $\boldsymbol{R}_{m-1}$ are sufficiently close. Specifically, we terminate the computations as soon as

$$\max_j \frac{\|\boldsymbol{R}_m(\zeta_j) - \boldsymbol{R}_{m-1}(\zeta_j)\|}{\|\boldsymbol{R}_m(\zeta_j)\|} < \varepsilon \tag{23}$$

DOI: 10.1002/nla

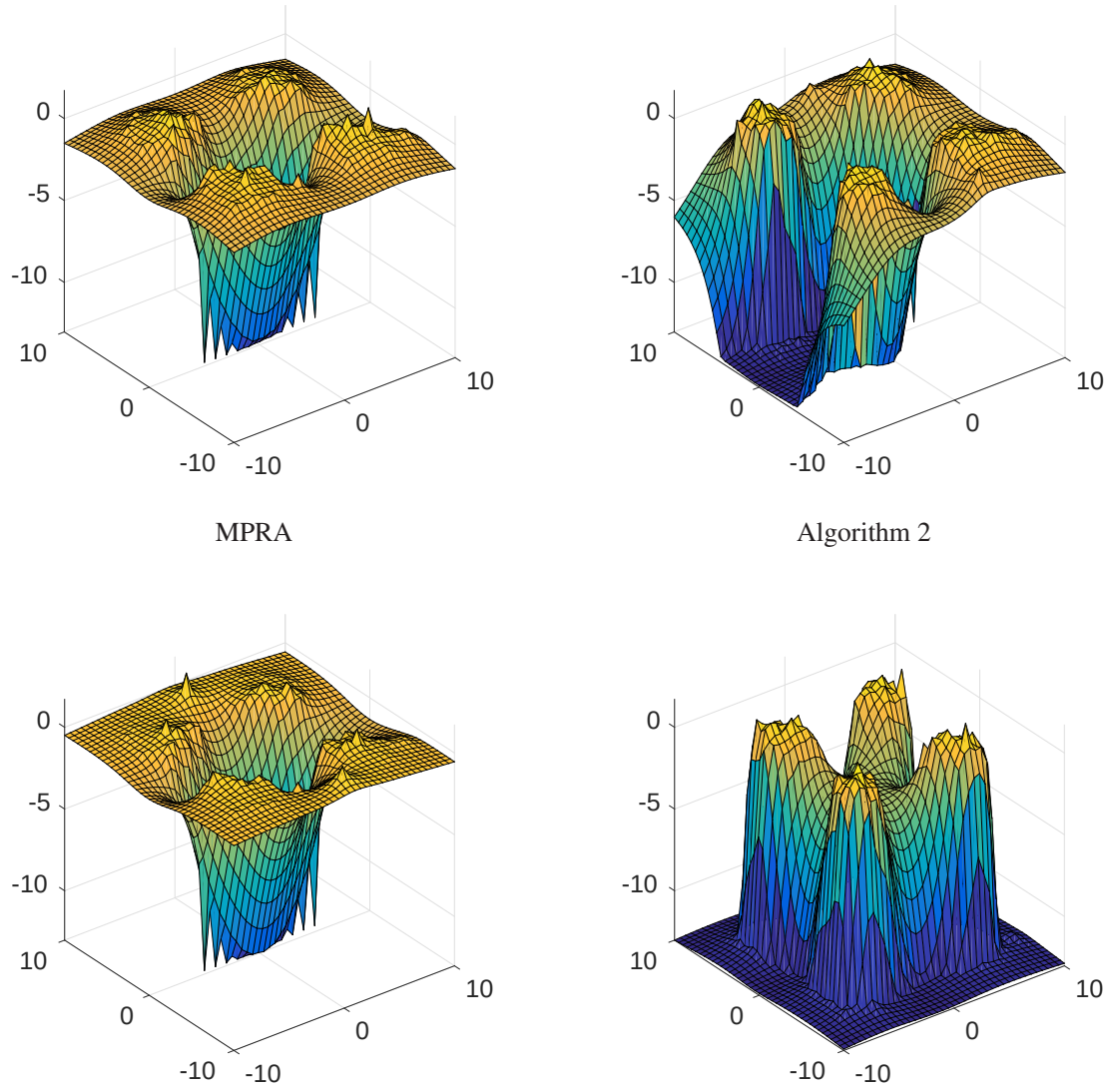


Figure 3. Relative errors in $\log_{10}$-scale for computed approximations of $\boldsymbol{F}(z) = f(zI - A)\boldsymbol{v}$ with $f(t) = t^{-1}$ measured using the Euclidean norm for MPRA, Algorithms 2 and 3, and the standard Arnoldi method. The graphs of Figure 2 are displayed here as 3D surfaces.

Table I. Number of iterations performed by the four algorithms in our comparison before satisfying the stopping criterion (23) in the experiments that produced the results reported in Figures 1, 2, 3, and 4. The numbers in parentheses denote execution times in seconds.

$f(t)$	MPRA	Algorithm 2	Algorithm 3	standard Arnoldi
$\ln(t)$	73 (5.5)	287 (373.3)	132 (9.8)	238 (269.4)
t^{-1}	37 (1.4)	156 (5.0)	33 (1.3)	270 (12.2)

at a set of prescribed points ζ_j, $j = 0, 1, \ldots, 10$. We let $\zeta_j = z_j$, and $\varepsilon = 10^{-3}$. Table I show MPRA and Algorithm 3 to satisfy this stopping criterion after fewer steps than Algorithm 2 and standard Arnoldi for both functions (21). Moreover, the graphs of Figures 1 and 2 illustrate that MPRA and Algorithm 3 give higher accuracy in a vicinity of the selected poles than the other two methods.

DOI: 10.1002/nla

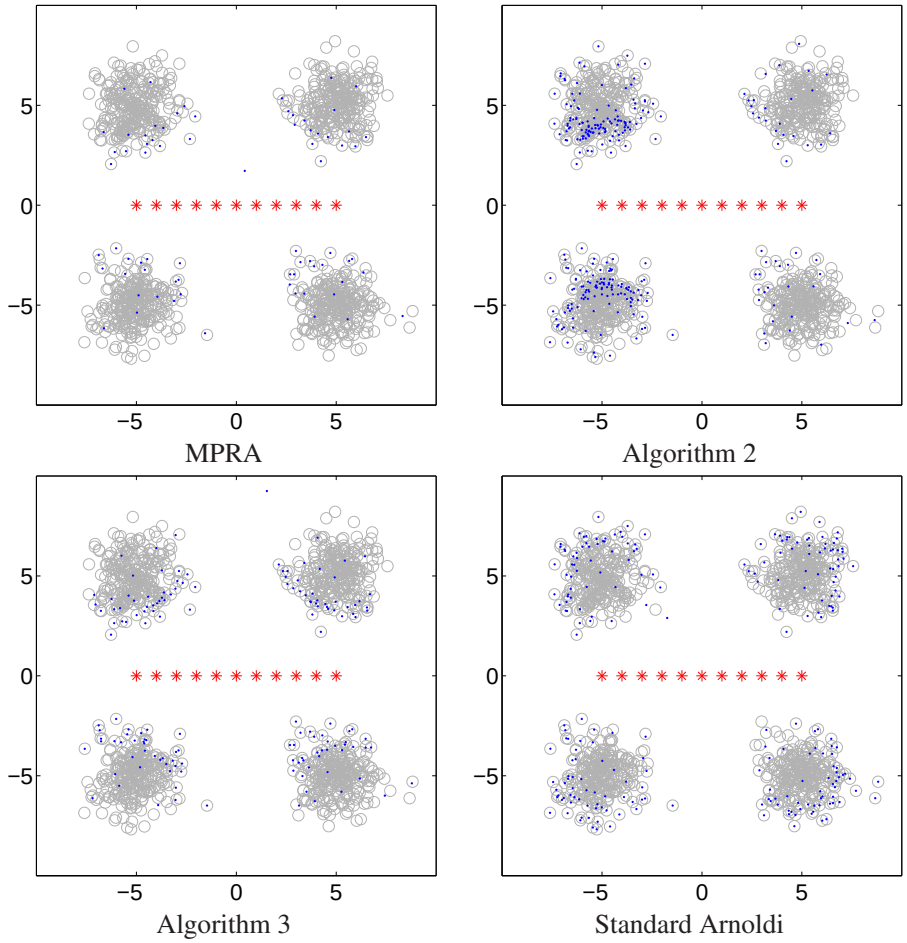


Figure 4. Location of the eigenvalues of A (circles) and of approximations of eigenvalues (dots) determined by MPRA, Algorithms 2 and 3, and the standard Arnoldi method. We applied these methods to approximate $\boldsymbol{F}(z) = f(zI - A)\boldsymbol{v}$ with $f(t) = \ln(t)$. MPRA determined 73 approximate eigenvalues, Algorithm 2 computed 287, Algorithm 3 determined 132, and standard Arnoldi determined 238 approximate eigenvalues. Note the different distribution of the computed eigenvalues for the four methods.

We observe that for the function $f(t) = \ln(t)$ the convergence of the MPRA method is substantially faster than of Algorithm 3. Very similar results are obtained for $f(t) = \sqrt{t}$. We do not report the latter in order not to make this section too long. The results produced by Algorithm 2 are strongly influenced by the ordering of the poles. This is due to the fact that Algorithm 2 sweeps across the poles only once, while the other two multipole methods use the poles cyclically during the iterations.

To be able to better observe the rate of convergence of the three multipole methods, we applied them to a test matrix of smaller size $A \in \mathbb{C}^{400\times 400}$ for the function $f(t) = \ln(t)$. For this size, we are able to evaluate $f(A)$ by the standard MATLAB function `logm`$(\cdot)$ and determine the error at each iteration. Figure 5 reports the relative error versus the iteration number for each method. While MPRA and Algorithm 3 yield monotonic convergence, Algorithm 2 exhibits instability and slower convergence.

4.2. Approximation of eigenvalues

Let $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_n]^T \in \mathbb{C}^n$ and define for each pole z_j, $j = 1, 2, \ldots, \ell$, the vector $\boldsymbol{\lambda}^{[j]} = [\lambda_1^{[j]}, \ldots, \lambda_n^{[j]}]^T \in \mathbb{C}^n$, whose entries are those of $\boldsymbol{\lambda}$ ordered according to increasing distance from

DOI: 10.1002/nla

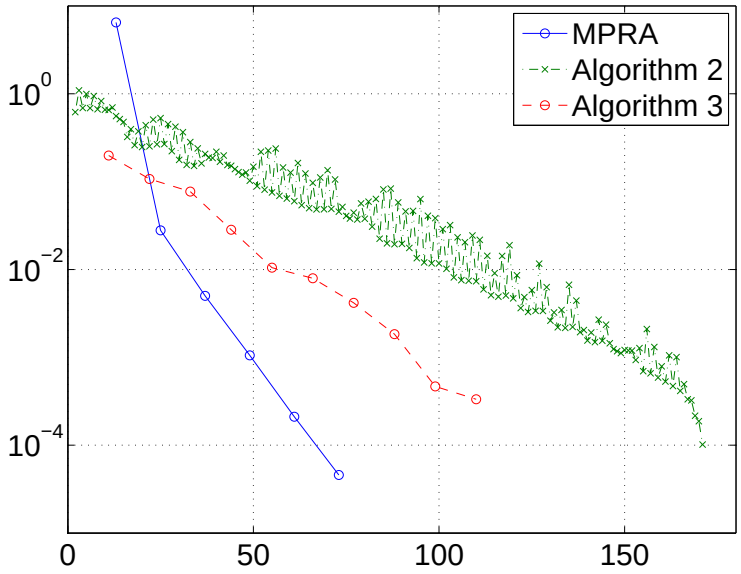


Figure 5. Relative error in the Euclidean norm measured at each iteration of MPRA, Algorithm 2, and Algorithm 3 when applied to the approximation of $\boldsymbol{F}(z) = f(zI - A)\boldsymbol{v}$ with $f(t) = \ln(t)$ and $A \in \mathbb{R}^{400\times 400}$. MPRA determined 73 approximate eigenvalues, Algorithm 2 computed 171, and Algorithm 3 determined 110 approximate eigenvalues.

z_j, i.e.,

$$|\lambda_1^{[j]} - z_j| \leq \cdots \leq |\lambda_n^{[j]} - z_j|.$$

We define, similarly, $\boldsymbol{\mu} \in \mathbb{C}^n$ and $\boldsymbol{\mu}^{[j]} = [\mu_1^{[j]}, \ldots, \mu_n^{[j]}]^T \in \mathbb{C}^n$ for $j = 1, 2, \ldots, \ell$, and introduce for the vector pair $\{\boldsymbol{\lambda}, \boldsymbol{\mu}\}$ the distance function

$$\mathcal{E}_{\bar{n}}(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{\max\{|\lambda_i^{[j]} - \mu_i^{[j]}| : j = 1, \ldots, \ell,\ i = 1, \ldots, \bar{n}\}}{\max\{1, |\lambda_i^{[j]}| : j = 1, \ldots, \ell,\ i = 1, \ldots, \bar{n}\}}. \tag{24}$$

To approximate the $\bar{n}$ eigenvalues closest to each pole, we apply Algorithm 5 and, at each iteration $m > \bar{n}$, we compute the eigenvalues $\lambda_1^{(m)}, \ldots, \lambda_m^{(m)}$ of $H_{m,m}$. Figure 6 (a) shows a snapshot of the convergence of the MPRA method when applied to a 100×100 matrix with equispaced eigenvalues on the unit circle. The 4 poles, depicted by stars in the graph, are allocated equidistantly on a circle of radius 1.2. The circles show the computed eigenvalues after 31 iteration steps. Figure 6 (b) displays the result produced by Algorithm 2 after the same number of steps. The different behavior of the two methods is evident. The eigenvalues determined by the MPRA method cluster at all the four poles, with roughly the same number of eigenvalues in each cluster, while the eigenvalues determined by Algorithm 2 converge sequentially to each single pole.

Table II compares Algorithms 2 and 3 to the MPRA method for computing the eigenvalues of a 1000×1000 matrix of the kind described in Subsection 4.1. The standard formulation of the latter with a pole at infinity is applied, and so is a variant without a pole at infinity. This variant is described at the end of Section 3. Computations with the algorithms are terminated as soon as the convergence test $\mathcal{E}_3(\boldsymbol{\lambda}^{(m)}, \boldsymbol{\lambda}^{(m-1)}) < 10^{-3}$ is satisfied; see (24). For Algorithm 2, this test is applied only to the subset of poles that have been used at that time. Each row of Table II shows average values for 10 random realizations of the matrix. The poles are the first 2, 4, or 8 elements, as specified in the second column of the table, of the set

$$\{8 + 5i, -5 + 8i, -8 - 5i, 5 - 8i, 5 + 8i, -8 + 5i, -5 - 8i, 8 - 5i\}.$$

DOI: 10.1002/nla

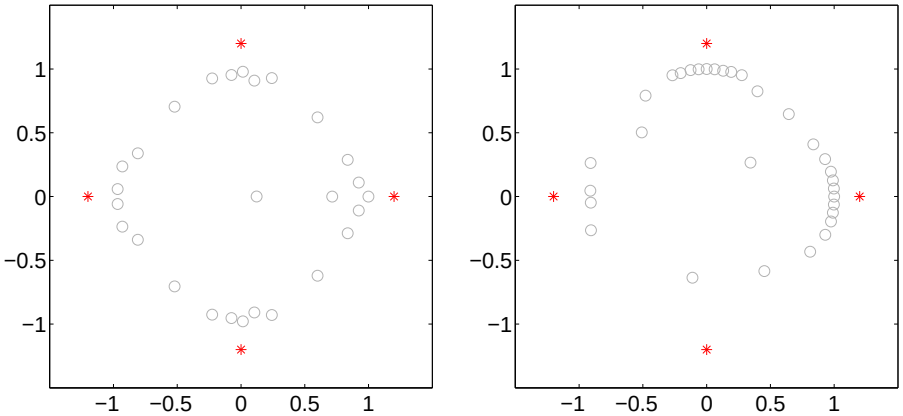


Figure 6. Snapshot of the convergence of the MPRA method and Algorithm 2 after 31 steps. The matrix has 100 eigenvalues equispaced on the unit circle. The poles are depicted by stars, the computed eigenvalues by circles.

Table II. Comparison of MPRA and Algorithms 2 and 3 for computing the eigenvalues of a matrix of order $n = 1000$. The iterations are terminated as soon as 3 eigenvalues for each pole are determined with prescribed accuracy. Each row of the table reports the average of 10 random realizations of the test matrix.

	poles	k	time	residual	error
MPRA	2	42	0.51	2.40e-14	2.11e-02
	4	79	1.08	2.37e-14	2.06e-02
	8	139	2.51	5.15e-14	3.31e-02
MPRA without a	2	26	0.45	2.67e-15	3.48e-02
pole at infinity	4	63	1.11	2.62e-15	4.49e-02
	8	128	2.44	4.92e-15	5.63e-02
Algorithm 2	2	25	0.41	6.61e-15	3.72e-02
	4	51	0.87	4.41e-15	6.92e-02
	8	92	1.97	3.32e-14	1.45e-01
Algorithm 3	2	28	0.42	2.87e-15	2.11e-02
	4	59	0.85	3.13e-15	3.41e-02
	8	125	1.96	3.00e-15	4.34e-02

The third column of Table II reports the dimension of the Krylov subspace required to satisfy the stopping criterion, the fourth column the computing time in seconds, and the fifth column the norm of the residual error of the factorization, that is $\|AV_m - V_{m+1}H_{m+1,m}\|$ for MPRA (the matrix A is replaced by $(A - z_1 I)^{-1}$ when the node at infinity is removed), and $\|(A - z_\ell I)^{-1}V_m - V_{m+1}H_{m+1,m}\|$ for Algorithm 2. The fact that the residual is close to machine precision ensures that the factorization process is numerically effective. The last column displays the approximation error measured by $\mathcal{E}_3(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(m)})$, where the vector $\boldsymbol{\lambda}$ contains the exact eigenvalues. The table shows the standard formulation of MPRA with a pole at infinity to give the highest accuracy. Algorithm 2 requires somewhat shorter computing time than MPRA. Removing the pole at infinity in the MPRA method slightly reduces the size of the projected matrix at convergence without significantly degrading the accuracy.

4.3. Application to pseudospectrum computation

We consider the application of the MPRA method to the computation of ε-pseudospectra of large matrices, and compare this method to Algorithms 2 and 3. The ε-pseudospectrum of a matrix

DOI: 10.1002/nla

$A \in \mathbb{C}^{n\times n}$, for fixed $\varepsilon > 0$, is the set

$$\Lambda_\varepsilon(A) := \{z \in \mathbb{C} : \sigma_{\min}(A - zI) \leq \varepsilon\},$$

where $\sigma_{\min}(M)$ denotes the smallest singular value of the matrix M. Thus, the spectrum of A is a subset of $\Lambda_\varepsilon(A)$. An insightful discussion of properties and many applications of the ε-pseudospectrum are presented by Trefethen and Embree [27].

The computation of the ε-pseudospectrum for a large matrix is very expensive. Therefore Toh, Trefethen, and Wright [26, 28, 29] discuss how the standard Arnoldi process and ARPACK [18], which is based on this process, can be used to reduce the matrix A to a small Hessenberg matrix H, and how an approximation of the ε-pseudospectrum of A can be determined by computing the ε-pseudospectrum of H. This approach may require the evaluation of a large number of matrix-vector products with A in order to be able to determine the ε-pseudospectrum of A with desired accuracy in selected parts of the complex plane. It therefore may be expensive. We are interested in investigating the performance of the MPRA method for this application. Toh and Trefethen [26] discuss the application of the Arnoldi process to A^{-1} and illustrate with an example that accurate ε-pseudospectra of A can be determined around the origin. The MPRA methods allows us to choose poles in areas in the complex plane, where we would like to gain knowledge of the ε-pseudospectrum. We refer to these computations as partial pseudospectrum computations.

MPRA allows us to compute, to high accuracy, an invariant subspace of the matrix A determined by eigenvalues in selected areas of the complex plane. The following result shows that such a subspace determines a subset of the pseudospectrum of A.

Proposition 5
Let the columns of the matrix $W \in \mathbb{C}^{n\times\ell}$ with $\ell \leq n$ form an orthonormal basis for an invariant subspace of the matrix $A \in \mathbb{C}^{n\times n}$. Then $AW = WM$ for some matrix $M \in \mathbb{R}^{\ell\times\ell}$ and

$$\Lambda_\varepsilon(M) \subseteq \Lambda_\varepsilon(A) \tag{25}$$

for any $\varepsilon \geq 0$.

Proof
The relation (25) follows from

$$\sigma_{\min}(A - zI) \leq \sigma_{\min}((A - zI)W) = \sigma_{\min}(W(M - zI)) = \sigma_{\min}(M - zI).$$

□

The following special case of the above proposition is of particular interest.

Corollary 6
Let λ be an eigenvalue of A. The disk

$$D_\varepsilon(\lambda) = \{z \in \mathbb{C} : |z - \lambda| \leq \varepsilon\}$$

is a subset of $\Lambda_\varepsilon(A)$ for any $\varepsilon \geq 0$.

Proof
Let $\boldsymbol{w} \in \mathbb{C}^n$ be an eigenvector of A of unit length associated with the eigenvalue λ and assume that $z \in D_\varepsilon(\lambda)$. Then we obtain similarly as in the proof of Proposition 5 that $\sigma_{\min}(A - zI) \leq |z - \lambda|$ and the desired result follows. □

We turn to the application of rational Arnoldi methods to the computation of pseudospectra of the matrix A and compare the MPRA method to Algorithms 2 and 3. When a standard Arnoldi decomposition

$$AV_m = V_{m+1}H_{m+1,m}$$

is available, where the orthonormal columns of V_{m+1} form a basis for the (standard) Krylov subspace $\mathcal{K}_{m+1}(A, \boldsymbol{v}) = \mathrm{span}\{\boldsymbol{v}, A\boldsymbol{v}, \ldots, A^m\boldsymbol{v}\}$ and $H_{m+1,m} \in \mathbb{C}^{(m+1)\times m}$ is an upper

DOI: 10.1002/nla

Hessenberg matrix, Toh and Trefethen [26] drop the last row of $H_{m+1,m}$ to obtain the square matrix $H_{m,m} \in \mathbb{C}^{m\times m}$ and use the ε-pseudospectrum of $H_{m,m}$ as an approximation of the ε-pseudospectrum of A, or determine ε-pseudospectra of $H_{m+1,m}$; see also Wright and Trefethen [28] for discussions on the approximation of pseudospectra of a large matrix by applying ARPACK. The latter is an implementation of the implicitly restarted Arnoldi method described in [18].

We outline how decompositions determined by the MPRA method (Algorithm 5) can be applied to compute approximations of pseudospectra. Assume that the decomposition (18) is available. Then we compute the ε-pseudospectrum of $H_{8,7}$ as an approximation of the ε-pseudospectrum of A. The computed ε-pseudospectrum is a subset of the ε-pseudospectrum of A. The following result is analogous to Theorem 1 by Toh and Trefethen [26] for the standard Arnoldi decomposition.

Proposition 7
Application of n steps of the rational Arnoldi process of Section 3 (Algorithm 5) to $A \in \mathbb{C}^{n\times n}$ gives a sequence of upper Hessenberg-like matrices $H^{(k)}$ until we obtain $H_{n,n} \in \mathbb{C}^{n\times n}$ provided that possible breakdowns are handled properly. The matrix $H^{(k)}$ has k columns; the number of rows depends on the number of finite poles. In particular, we have $H^{(1)} = H_{2,1}$ in (14) and $H^{(n)} = H_{n,n}$. Then

$$\Lambda_\varepsilon(H^{(1)}) \subset \Lambda_\varepsilon(H^{(2)}) \subset \cdots \subset \Lambda_\varepsilon(A). \tag{26}$$

Proof
If Algorithm 5 does not break down prematurely, then n steps of the algorithm gives the decomposition

$$AV_n = V_n H_{n,n} \tag{27}$$

with $H_{n,n} \in \mathbb{C}^{n\times n}$ of upper Hessenberg-type and $V_n \in \mathbb{C}^{n\times n}$ unitary. If the algorithm breaks down at step k, then we restart the computations with an arbitrary unit vector that is orthogonal to the vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$ already computed. We therefore may assume that the decomposition (27) exists.

Since the matrices $H_{n,n}$ and A are unitarily similar, they have the same singular values, and so do the matrices $A - zI$ and $H_{n,n} - zI$ for $z \in \mathbb{C}$. The smallest singular value of a matrix obtained by removing a row or column of a given matrix is larger than or equal to the smallest singular value of the original matrix. Therefore,

$$\sigma_{\min}(H^{(1)} - zI) \geq \sigma_{\min}(H^{(2)} - zI) \geq \cdots \geq \sigma_{\min}(H^{(n)} - zI), \tag{28}$$

where I denotes a truncated identity matrix with ones on the diagonal, zeros everywhere else, and with the same number of rows and columns as the corresponding matrix $H^{(j)}$. The inclusions (26) now follow from (28). □

Algorithm 5 assumes that one pole is at infinity. If we only are interested in pseudospectra near points in the finite complex plane, then we can use the transformation (19). This may make it possible to carry out fewer steps than when using an additional pole at infinity. For instance, when we have two finite poles and use the transformation (19) to move one of them to infinity, we have to transform the problem back before computing the ε-pseudospectrum. Specifically, we have, for some $m > 0$,

$$SV_m = V_{m+1}H_{m+1,m} \approx V_m H_{m,m}.$$

Substituting (19) into the left-hand side yields after some manipulations

$$AV_m \approx V_m(H_{m,m}^{-1} + z_1 I).$$

We therefore use the ε-pseudospectra of $H_{m,m}^{-1} + z_1 I$ as approximations of the ε-pseudospectra of A. Note that the pseudospectra of $H_{m,m}^{-1} + z_1 I$ are not related to pseudospectra of A by inclusions of the kind (26).

DOI: 10.1002/nla

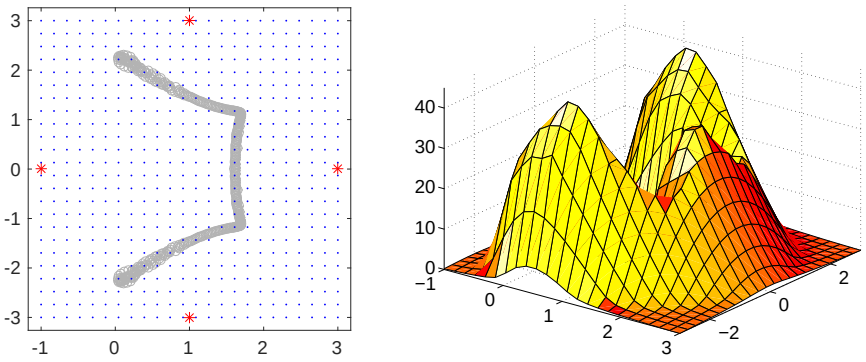
Let

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & & 0 \\ -1 & \ddots & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \ddots & 1 \\ & & \ddots & \ddots & \ddots & 1 \\ & & & \ddots & \ddots & 1 \\ 0 & & & & -1 & 1 \end{bmatrix} \in \mathbb{R}^{400\times 400}$$

be a Grcar matrix. These matrices are banded nonsymmetric Toeplitz matrices with -1 on the subdiagonal and 1 on the diagonal and three superdiagonals. We choose the finite complex poles $\{-1, 1-3i, 3, 1+3i\}$, where $i = \sqrt{-1}$. Algorithm 5 is applied with the stopping criterion

$$\frac{\max_j |P_m(\zeta_j) - P_{m-1}(\zeta_j)|}{\max_j |P_m(\zeta_j)|} < 10^{-3}, \tag{29}$$

where $P_m(z) = \sigma_{\min}(H_{m+1,m} - zI_{m+1,m})$ gives approximations of pseudospectra of A at iteration m. Here $I_{m+1,m} \in \mathbb{C}^{(m+1)\times m}$ denotes the matrix obtained by removing the last column from the identity matrix of order $m+1$. The evaluation points ζ_j are the complex numbers $\{0, 1, 2, i, -i\}$. Figure 7 shows the exact eigenvalues of A, as computed by the `eig` function of MATLAB, the poles, the approximation grid, as well as the function $G(z) = -\log_{10} P(z)$ evaluated on the grid, where $P(z)$ yields the exact pseudospectra of A.



(a) (b)

Figure 7. (a): Eigenvalues of the Grcar matrix of order $n = 400$ (circles), poles z_1, z_2, z_3, z_4 (stars) ordered counterclock-wise with z_1 having the largest real part, approximation grid (dots). (b): Exact pseudospectrum on the grid.

Figure 8 displays reciprocal pseudospectra computed by the MPRA and Algorithm 2 (top row) and the corresponding relative error

$$\frac{|G(z) - G_m(z)|}{|G(z)|}$$

(bottom row) on the grid, where $G_m(z) = -\log_{10} P_m(z)$. The stopping criterion (29) was satisfied by MPRA after $k = 321$ steps and by Algorithm 2 after $k = 309$ steps.

Neither algorithm gives an accurate approximation close to the eigenvalues. This is somehow expected, as the stopping criterion (29) is checked at points at a reasonable distance from the eigenvalues. The error obtained when using MPRA is more symmetric and significantly smaller in parts of the complex plane than the error produced by Algorithm 2.

The approximation determined by Algorithm 2 turns out to be sensitive to the ordering of the poles. This is illustrated in Figure 9, which reports the error corresponding to three different permutations of the poles. Note that the error in the third graph is roughly equivalent to the error

DOI: 10.1002/nla

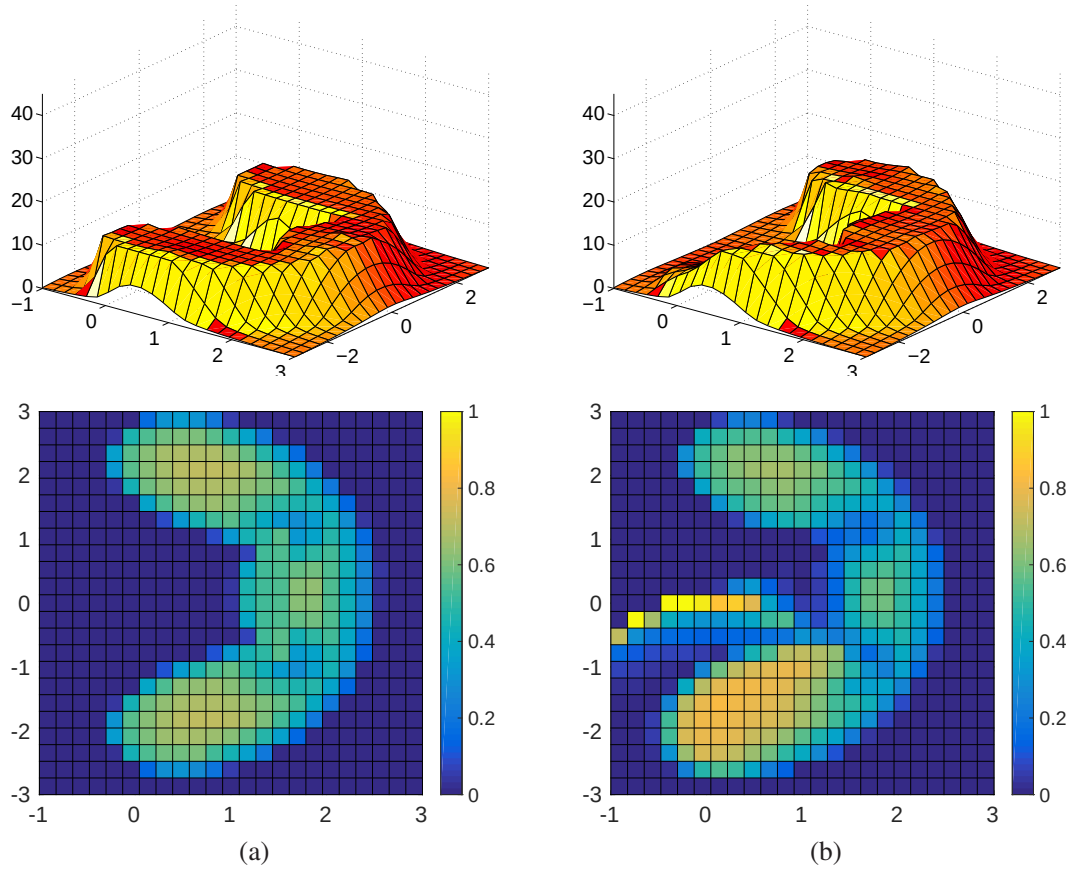


Figure 8. Top: Approximation of the pseudospectrum on the grid; bottom: magnitude of the relative error of the computed pseudospectrum. (a): MPRA; (b): Algorithm 2 with poles ordered as described in the caption of Figure 7.

produced with MPRA method. This dependence of Algorithm 2 on the ordering of the poles makes it difficult to compare the algorithm with MPRA. Here we only note that it is not clear a priori which is the best pole ordering for Algorithm 2.

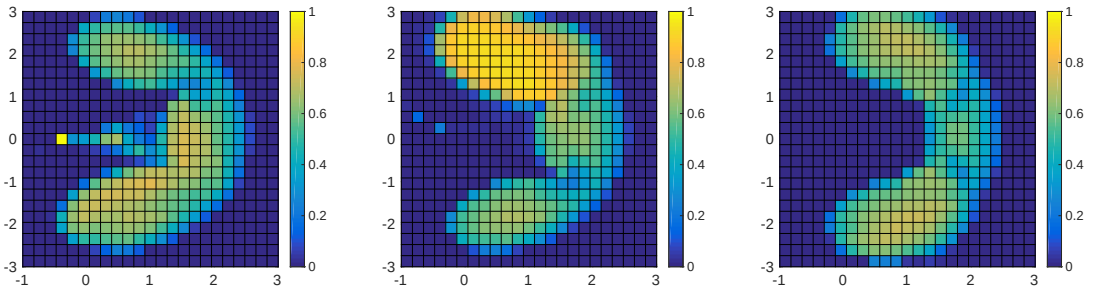


Figure 9. Algorithm 2: Errors in the computed pseudospectra of the Grcar matrix of order $n = 400$ using three permutations of the poles. The permutations are, from left to right: $z_2 z_3 z_4 z_1$, $z_3 z_4 z_1 z_2$, $z_4 z_1 z_2 z_3$.

In this example, the rational Arnoldi approximations determined by Algorithm 3 proved to be extremely unstable. We show in Figure 10 approximations of pseudospectra computed for Grcar matrices of orders $n = 80, 100, 120$. In all cases, Algorithm 3 carried out the maximum number of iterations, that is, the factorizations (6) were computed for $m = n$. The results can be seen to be meaningless for the larger n-values.

DOI: 10.1002/nla

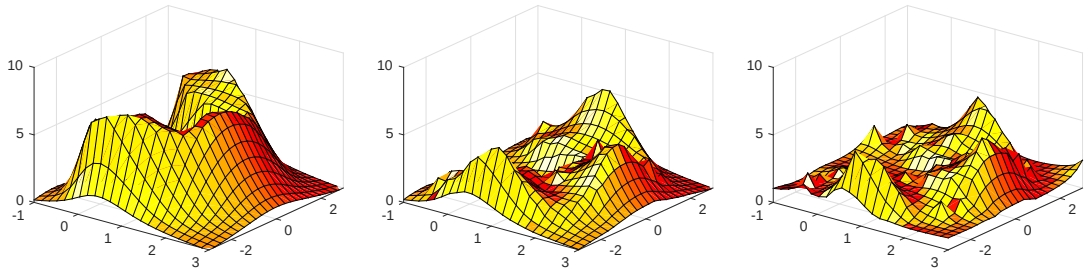


Figure 10. Algorithm 3: Approximations of the pseudospectrum of the Grcar matrix of orders $n = 80, 100, 120$. The algorithm carried out n iterations in the three examples.

Our tests show that the MPRA method is promising for the computation of approximations of pseudospectra in selected parts of the complex plane. We believe that more work is required to obtain an effective and robust implementation. In particular, an adaptive strategy for the dynamic choice of the poles should be developed, in order to obtain high accuracy in the domain of interest without requiring excessive computation time.

5. CONCLUSION

A new implementation of the multipole rational Arnoldi method is described. Properties of the reduced matrix are investigated, and computed examples that compare this implementation to other schemes proposed by Ruhe are presented. The new implementation can give higher accuracy and is less sensitive to the ordering of the poles.

ACKNOWLEDGEMENTS

We thank the referees for the time they devoted to read our paper and for their comments. MP was supported in part by the Serbian Ministry of Education and Science (research project "Methods of Numerical and Nonlinear Analysis with Applications" no. 174002) and by the Ministry of Science and Technology of Republika Srpska (research project "Generalizations of Gauss quadratures and their applications"). LR would like to thank GR making an enjoyable stay in Cagliari possible during which part of this work was carried out. This stay was supported by the University of Cagliari RAS Visiting Professor Program. LR and XY were supported in part by NSF grant DMS-1115385. GR was partially supported by INdAM-GNCS. ZW was partially supported by the Natural Science Foundation of the Jiangsu Province of China under grant No. BK20141408, the National Natural Science Foundation of China under grant No.61473148, the Jiangsu Overseas Research & Training Program for Prominent Young & Middle-Aged University Teachers and Presidents, and the Fundamental Research Funds for the Central Universities under grant No. NZ2014101.

REFERENCES

1. E. J. Allen, J. Baglama, and S. K. Boyd, Numerical approximation of the product of the square root of a matrix with a vector, Linear Algebra and its Applications, 310 (2000), pp. 167–181.
2. B. Beckermann and S. Güttel, Superlinear convergence of the rational Arnoldi method for the approximation of matrix functions, Numerische Mathematik, 121 (2012), pp. 205–236.
3. B. Beckermann and L. Reichel, Error estimation and evaluation of matrix functions via the Faber transform, SIAM Journal on Numerical Analysis, 47 (2009), pp. 3849–3883.
4. V. Druskin and L. Knizhnerman, Extended Krylov subspace approximations of the matrix square root and related functions, SIAM Journal on Matrix Analysis and Applications, 19 (1998), pp. 755–771.
5. V. Druskin, L. Knizhnerman, and M. Zaslavsky, Solution of large scale evolutionary problems using rational Krylov subspaces with optimized shifts, SIAM Journal on Scientific Computing, 31 (2009), pp. 3760–3780.
6. E. Estrada and D. J. Higham, Network properties revealed through matrix functions, SIAM Review, 52 (2010), pp. 696–714.
7. C. Fenu, D. Martin, L. Reichel, and G. Rodriguez, Network analysis via partial spectral factorization and Gauss quadrature, SIAM Journal on Scientific Computing, 35 (2013), pp. A2046–A2068.
8. E. Gallopoulos and Y. Saad, Efficient solution of parabolic equations by Krylov approximation methods, SIAM Journal on Scientific and Statistical Computing, 13 (1992), pp. 1236–1264.

DOI: 10.1002/nla

9. W. Gautschi, Orthogonal Polynomials: Computation and Approximation, Oxford University Press, Oxford, 2004.
10. G. H. Golub and C. F. Van Loan, Matrix Computations, 4th ed., Johns Hopkins University Press, Baltimore, 2013.
11. S. Güttel, Rational Krylov Methods for Operator Functions, Ph.D. thesis, Fakultät für Mathematik und Iformatik, Technische Universität Bergakademi Freiberg, Freiberg, Germany, 2010.
12. S. Güttel and L. Knizhnerman, A black-box rational Arnoldi variant for Cauchy-Stietljes matrix functions, BIT Numerical Mathematics, 53 (2013), pp. 595–616.
13. G. I. Hargreaves and N. J. Higham, Efficient algorithms for the matrix cosine and sine, Numerical Algorithms, 40 (2005), pp. 383–400.
14. N. J. Higham, Functions of Matrices: Theory and Computation, SIAM, Philadelphia, 2008.
15. M. Hochbruck and C. Lubich, On Krylov subspace approximations to the matrix exponential operator, SIAM Journal on Numerical Analysis, 34 (1997), pp. 1911–1925.
16. C. Jagels and L. Reichel, Recursion relations for the extended Krylov subspace method, Linear Algebra and its Applications, 434 (2011), pp. 1716–1732.
17. L. Knizhnerman and V. Simoncini, A new investigation of the extended Krylov subspace method for matrix function evaluations, Numerical Linear Algebra with Applications, 17 (2010), pp. 615–638.
18. R. B. Lehoucq, D. C. Sorensen, and C. Yang, ARPACK Users Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods, SIAM, Philadelphia, 1998.
19. M. S. Pranić and L. Reichel, Recurrence relations for orthogonal rational functions, Numerische Mathematik, 123 (2013), pp. 629–642.
20. A. Ruhe, Rational Krylov sequence methods for eigenvalue computation, Linear Algebra and its Applications, 58 (1984), pp. 391–405.
21. A. Ruhe, Rational Krylov algorithms for nonsymmetric eigenvalue problems, in Recent Advances in Iterative Methods, eds. G. Golub, A. Greenbaum, and M. Luskin, Springer, New York, 1994, pp. 149–164.
22. A. Ruhe, Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils, SIAM Journal on Scientific Computing, 19 (1998), pp. 1535–1551.
23. A. Ruhe, Rational Krylov subspace methods, in Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, eds. Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, SIAM, Philadelphia, 2000.
24. R. B. Sidje, Expokit: a software package for computing matrix exponentials, ACM Transactions on Mathematical Software, 24 (1998), pp. 130–156.
25. V. Simoncini, A new iterative method for solving large-scale Lyapunov matrix equations, SIAM Journal on Scientific Computing, 29 (2007), pp. 1268–1288.
26. K.-C. Toh and L. N. Trefethen, Calculation of pseudospectra by the Arnoldi iteration, SIAM Journal on Scientific Computing, 17 (1996), pp. 1–15.
27. L. N. Trefethen and M. Embree, Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators, Princeton University Press, Princeton, 2005.
28. T. G. Wright and L. N. Trefethen, Large-scale computation of pseudospectra using ARPACK and eigs, SIAM Journal on Scientific Computing, 23 (2001), pp. 591–605.
29. T. G. Wright and L. N. Trefethen, Pseudospectra of rectangular matrices, IMA Journal of Numerical Analysis, 22 (2002), pp. 501–519.

DOI: 10.1002/nla