

On restarting the tensor infinite Arnoldi method

Giampaolo Mele¹  · Elias Jarlebring¹

Received: 14 July 2016 / Accepted: 5 July 2017 / Published online: 13 July 2017
© The Author(s) 2017. This article is an open access publication

Abstract An efficient and robust restart strategy is important for any Krylov-based method for eigenvalue problems. The tensor infinite Arnoldi method (TIAR) is a Krylov-based method for solving nonlinear eigenvalue problems (NEPs). This method can be interpreted as an Arnoldi method applied to a linear and infinite dimensional eigenvalue problem where the Krylov basis consists of polynomials. We propose new restart techniques for TIAR and analyze efficiency and robustness. More precisely, we consider an extension of TIAR which corresponds to generating the Krylov space using not only polynomials, but also structured functions, which are sums of exponentials and polynomials, while maintaining a memory efficient tensor representation. We propose two restarting strategies, both derived from the specific structure of the infinite dimensional Arnoldi factorization. One restarting strategy, which we call semi-explicit TIAR restart, provides the possibility to carry out locking in a compact way. The other strategy, which we call implicit TIAR restart, is based on the Krylov–Schur restart method for the linear eigenvalue problem and preserves its robustness. Both restarting strategies involve approximations of the tensor structured factorization in order to reduce the complexity and the required memory resources. We bound the error introduced by some of the approximations in the infinite dimensional Arnoldi factorization showing that those approximations do not substantially influence the robustness of the restart approach. We illustrate the effectiveness of the approaches by

Communicated by Daniel Kressner.

✉ Giampaolo Mele
gmele@kth.se
Elias Jarlebring
eliasj@kth.se

¹ Department of Mathematics, Swedish e-science Research Center (SeRC), KTH Royal Institute of Technology, Lindstedtsvägen 25, Stockholm, Sweden

applying them to solve large scale NEPs that arise from a delay differential equation and a wave propagation problem. The advantages in comparison to other restart methods are also illustrated.

Keywords Nonlinear eigenvalue problem · Restart · Tensor infinite Arnoldi · Krylov subspace method · Krylov–Schur method

Mathematics Subject Classification 35P30 · 65H17 · 65F60 · 15A18 · 65F15

1 Introduction

We consider the *nonlinear eigenvalue problem* (NEP) defined as finding $(\lambda, v) \in \mathbb{C} \times \mathbb{C}^n \setminus \{0\}$ such that

$$M(\lambda)v = 0 \quad (1)$$

where $\lambda \in \Omega \subseteq \mathbb{C}$, Ω is an open disk centered in the origin and $M : \Omega \rightarrow \mathbb{C}^{n \times n}$ is analytic. The NEP has received a considerable attention in literature. See the review papers [26, 38] and the problem collection [7].

There is a large number of methods available in a large amount of numerical linear algebra literature for (1). There are specialized methods for solving different classes of NEPs such as polynomial eigenvalue problems (PEPs) see [18, 22, 23] and [2, Chapter 9], in particular quadratic eigenvalue problems (QEPs) [3, 24, 25, 33] and rational eigenvalue problems (REPs) [5, 6, 30, 36]. There are also methods that exploit the structure of the operator $M(\lambda)$ like Hermitian structure [31, 32] or low rank of the matrix-coefficients [34]. Methods for solving a more general class of NEP are also present in literature. There exist methods based on modification of the Arnoldi method [37], which can be restarted for certain problems, Jacobi–Davidson methods [8], Newton-like methods [9, 16, 28]. Finally, there is a class of methods (to which the presented method belongs) based on Krylov methods and rational Krylov methods that can be interpreted as either dynamically expanding an approximation of the NEP or applying a method on an infinite dimensional operator [11, 14, 35].

In principle, we do not assume any particular structure of the NEP except for the analyticity and the computability of certain quantities associated with $M(\lambda)$ (further described later). This is similar to the *infinite Arnoldi method* (IAR) [14], which is in the same line of reasoning as our approach. IAR is equivalent to the Arnoldi method applied to a linear operator. More precisely, under the assumption that zero is not an eigenvalue, the problem (1) can be reformulated as $\lambda B(\lambda)v = v$, where $B(\lambda) = M(0)^{-1}(M(0) - M(\lambda))/\lambda$. This problem is equivalent to the linear and infinite dimensional eigenvalue problem $\lambda \mathcal{B}\psi(\theta) = \psi(\theta)$, where $\psi : \mathbb{C} \rightarrow \mathbb{C}^n$ is an analytic function [14, Theorem 3]. The operator \mathcal{B} is linear, maps functions to functions, and it is defined as

$$\mathcal{B}\psi(\theta) := \int_0^\theta \psi(\hat{\theta}) d\hat{\theta} + \sum_{i=0}^{\infty} \frac{B^{(i)}(0)}{i!} \psi^{(i)}(0). \quad (2)$$

IAR has a particular structure such that it can be represented with a tensor and this was the basis for the *tensor infinite Arnoldi method* (TIAR) in [13]. TIAR is equivalent to IAR but computationally more attractive (in terms of memory and CPU-time) due to a memory efficient representation of the basis matrix.

A problematic aspect of any algorithm based on the Arnoldi method is that, when many iterations are performed, the computation time per iteration will eventually become large. Moreover, finite arithmetic aspects may restrict the accuracy. Fortunately, an appropriate restart of the algorithm can resolve these issues in many situations. There exist two main classes of restarting strategies: explicit restart and implicit restart. Most of the explicit restart techniques correspond to selecting a starting vector that generates an Arnoldi factorization with the wanted Ritz values. The implicit restart consists in computing a new Arnoldi factorization, without explicitly computing a starting vector, with the wanted Ritz values. This process can be done deflating the unwanted Ritz values as in, e.g., IRA [20] or extracting a proper subspace of the Krylov space by using the Krylov–Schur restart approach [29]. For reasons of numerical stability, the implicit restart is often considered more robust than explicit restart. See [27] for further discussions about the restart of the Arnoldi method for the linear eigenvalue problem.

In this work we propose two new restart techniques for TIAR:

- An implicit restart, which consists in an adaption of the Krylov–Schur restart,
- A semi-explicit restart, which consists in an explicit restart by imposing the structure on the converged locked Ritz pairs and on the starting function.

The derivation of our implicit restart procedure is based on Krylov–Schur restarting in infinite dimensions. We improve the procedure in several ways. The structure of the Arnoldi factorization allows us to perform approximations that reduce the complexity and the memory requirements. We prove that the coefficients matrix, representing the basis of the Krylov space, presents a fast decay in the singular values. This allows us to effectively use a low rank approximation of this matrix. Moreover, we prove that there is a fast decay in the coefficients of the polynomials representing the basis of the Krylov space. Therefore, the high order coefficients of the Krylov basis can be neglected if the power series coefficients of $M(\lambda)$ decay to zero. We give explicit bounds on the errors due to those approximations.

A semi-explicit restart for IAR was presented in [12]. We extend the procedure to TIAR. The feature of imposing the structure on the converged Ritz values and starting function is obtained by generating the Krylov space using a particular type of structured functions, which are sums of polynomial and exponential functions. We show that such functions can be included in the framework of TIAR. In particular we carry out a memory efficient representation of the structured functions, similar to [13].

There exist other Arnoldi-like methods combined with a companion linearization that use memory efficient representation of the Krylov basis matrix and that can be restarted. There are, for instance, TOAR [17, 39] and CORK [35], which are based on the concept of compact Arnoldi decompositions [21]. Similar to TIAR, the direct usage

of the Krylov–Schur restart for these methods does not decrease the complexity unless SVD-based approximations are used (which is indeed suggested in the implementation of the methods). More precisely, the coefficients that represent the Krylov basis are replaced with their low rank approximations. In contrast to those approaches, our specific setting, in particular the infinite dimensional function formulation and the representation of the basis with tensor structure functions, allows us to characterize the impact of the approximations.

The relationship with CORK [35] can be seen as follows. A variation of a special case of our approach (implicit restart without SVD-compression and without polynomial degree reduction) has similarities with a special case of a variation of CORK (single shift, with a particular companion linearization without SVD-compression). Our approach is based on a derivation involving infinite dimensionality which allows us to derive theory for the truncation and it allows us to restart with infinite dimensional objects. This strategy is effective since the invariant pairs are infinite dimensional objects. In contrast to this, CORK is derived from reasoning concerning the NEP linearization. This allows the usage of different types of companion linearizations, that correspond to different approximations of the nonlinearities of $M(\lambda)$, and leads to a rational Krylov approach, i.e., several shifts can be used in one run.

The paper is organized as follows: in Sect. 2 we extend TIAR to tensor structured functions. In Sect. 3 we present a derivation of the Krylov–Schur type restarting in an abstract and infinite dimensional setting. Section 4 contains the derivation of a semi-explicit restart for TIAR. In Sect. 5 we carry out the adaption of Krylov–Schur restart for TIAR. We analyze the complexity of the proposed methods in Sect. 6. Finally, in Sect. 7 we show the effectiveness of the restarting strategies with numerical simulations to large and sparse NEPs.

We denote by $a_{:, :, :}$ a three-dimensional tensor and by $a_{i, :, :}$, $a_{:, j, :}$ and $a_{:, :, \ell}$ the slices of the tensor with respect to the first, second and third dimension. The vector z_j denotes the j -th column of the matrix Z and e_j the j -th canonical unit vector. The matrix $I_{m,p}$ denotes the matrix obtained by extracting the first m rows and p columns of a larger square identity matrix. The matrix H_k denotes the square matrix obtained by removing the last row from the matrix $\underline{H}_k \in \mathbb{C}^{(k+1) \times k}$.

2 Tensor structured functions and TIAR factorizations

Our main algorithms are derived using particular types of functions. More precisely, we consider functions that can be expressed as $\psi(\theta) = q(\theta) + Y \exp(S\theta)c$ where $q : \mathbb{C} \rightarrow \mathbb{C}^n$ is a polynomial, $Y \in \mathbb{C}^{n \times p}$, $S \in \mathbb{C}^{p \times p}$, $c \in \mathbb{C}^p$ and $\exp(S\theta)$ denotes the matrix exponential. Such functions were also used in [12]. We now introduce a new memory-efficient representation of such functions involving tensors.

Definition 1 (*Tensor structured function*) The vector-valued function $\psi : \mathbb{C} \rightarrow \mathbb{C}^n$ is a tensor structured function if there exist $Y, W \in \mathbb{C}^{n \times p}$, $\bar{a} \in \mathbb{C}^{d \times r}$, $\bar{b} \in \mathbb{C}^{d \times p}$, $\bar{c} \in \mathbb{C}^p$, $S \in \mathbb{C}^{p \times p}$ and $Z \in \mathbb{C}^{n \times r}$ where $[Z, W]$ has orthonormal columns and $\text{span}(Y) = \text{span}(W)$, such that

$$\psi(\theta) = P_{d-1}(\theta) \left(\sum_{\ell=1}^r \bar{a}_{:, \ell} \otimes z_{\ell} + \sum_{\ell=1}^p \bar{b}_{:, \ell} \otimes w_{\ell} \right) + Y \exp_{d-1}(\theta S) \bar{c} \quad (3)$$

where

$$P_{d-1}(\theta) := (1, \theta, \dots, \theta^{d-1}) \otimes I_n \quad (4)$$

and $\exp_{d-1}(\theta S) := \sum_{i=d}^{\infty} \theta^i S^i / i!$ is the remainder of the Taylor series expansion of the exponential function.

The matrix-valued function $\Psi_k : \mathbb{C} \rightarrow \mathbb{C}^{n \times k}$ is a tensor structured function if it can be expressed as $\Psi_k(\theta) = (\psi_1(\theta), \dots, \psi_k(\theta))$, where each ψ_i is a tensor structured function. We denote the i -th column of Ψ_k by ψ_i . The structure induced by (3) is now, in a compact form

$$\Psi_k(\theta) = P_{d-1}(\theta) \left(\sum_{\ell=1}^r a_{:, :, \ell} \otimes z_{\ell} + \sum_{\ell=1}^p b_{:, :, \ell} \otimes w_{\ell} \right) + Y \exp_{d-1}(\theta S) C \quad (5)$$

where $a \in \mathbb{C}^{d \times k \times r}$, $b \in \mathbb{C}^{d \times k \times p}$ and $C \in \mathbb{C}^{p \times k}$. In particular, $\Psi_k(\theta)$ is represented by the matrices (Z, W, Y, S) and the coefficients (a, b, C) . We say that $\Psi_k(\theta)$ is orthogonal if the columns are orthonormal, i.e., $\langle \psi_i, \psi_j \rangle = \delta_{i,j}$ for $i, j = 1, \dots, k$. We use the scalar product consistent with the other papers about the infinite Arnoldi method [12, 14], i.e., if $\psi(\theta) = \sum_{i=0}^{\infty} \theta^i x_i$ and $\varphi(\theta) = \sum_{i=0}^{\infty} \theta^i y_i$, then

$$\langle \psi, \varphi \rangle = \sum_{i=0}^{\infty} y_i^H x_i.$$

The computation of this scalar product and the induced norm for the tensor structured functions (3) is further characterized in this section. We let $\|F\|$ denote the induced norm of $F : \mathbb{C} \rightarrow \mathbb{C}^{n \times p}$. Notice that the polynomials are also tensor structured functions, more precisely they are represented as (5) with $C = 0$. In particular, by definition of (4), we have the following relation between the induced norm of a polynomial and the Frobenius norm if its coefficients

$$\|P_{d-1}W\| = \|W\|_F \quad (6)$$

for any $W \in \mathbb{C}^{nd \times p}$.

Similar to many restart strategies for linear eigenvalue problems, our approach is based on computation, representation and manipulation of an Arnoldi-type factorization. For our infinite dimensional operator, the analogous Arnoldi-type factorization is defined as follows.

Definition 2 (TIAR factorization) Let \mathcal{B} be the operator defined in (2). Let $\Psi_{k+1} : \mathbb{C} \rightarrow \mathbb{C}^{n \times (k+1)}$ be a tensor structured function with orthonormal columns and let

$\underline{H}_k \in \mathbb{C}^{(k+1) \times k}$ be a Hessenberg matrix with positive elements in the sub-diagonal. The pair $(\Psi_{k+1}, \underline{H}_k)$ is a TIAR factorization of length k if

$$\mathcal{B}\Psi_k(\theta) = \Psi_{k+1}(\theta)\underline{H}_k. \quad (7)$$

2.1 Action of \mathcal{B} on tensor structured functions

The TIAR factorization in Definition 2 involves the action of the operator \mathcal{B} . In order to characterize the action of the operator \mathcal{B} on tensor structured functions (3), we need the function $\mathbb{M}_d : \mathbb{C}^{n \times p} \times \mathbb{C}^{p \times p} \rightarrow \mathbb{C}^{n \times p}$, defined in [12] as

$$\mathbb{M}_d(Y, S) := \sum_{i=d+1}^{\infty} \frac{M_i Y S^i}{i!}, \quad (8)$$

where we introduced the notation $M_i := M^{(i)}(0)$.

Theorem 3 (Action of \mathcal{B}) *Let $(Z, W, Y, S) \in \mathbb{C}^{n \times r} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{p \times p}$ be the matrices and $(\bar{a}, \bar{b}, \bar{c}) \in \mathbb{C}^{d \times r} \times \mathbb{C}^{d \times p} \times \mathbb{C}^p$ be the coefficients that represent $\psi(\theta)$ given in (3). Suppose $\lambda(S) \subset \Omega \setminus \{0\}$, let $\tilde{c} := S^{-1}\bar{c}$ and*

$$\tilde{z} := -M_0^{-1} \left[\sum_{i=1}^d M_i \left(\sum_{\ell=1}^r \frac{\bar{a}_{i,\ell}}{i} z_\ell + \sum_{\ell=1}^p \frac{\bar{b}_{i,\ell}}{i} w_\ell \right) + \mathbb{M}_d(Y, S)\tilde{c} \right]. \quad (9)$$

Under the assumption that

$$\tilde{z} \notin \text{span}(z_1, \dots, z_r, w_1, \dots, w_p), \quad (10)$$

let z_{r+1} be the normalized result of the Gram–Schmidt orthogonalization of \tilde{z} against $z_1, \dots, z_r, w_1, \dots, w_p$ and $\tilde{a}_{1,\ell}$ and $\tilde{b}_{1,\ell}$ be the orthonormalization coefficients, i.e.,

$$\tilde{z} = \sum_{\ell=1}^{r+1} \tilde{a}_{1,\ell} z_\ell + \sum_{\ell=1}^p \tilde{b}_{1,\ell} w_\ell. \quad (11)$$

Then, the action of \mathcal{B} on the tensor structured function defined by (3) is

$$\mathcal{B}\psi(\theta) = P_d(\theta) \left(\sum_{\ell=1}^{r+1} \tilde{a}_{\cdot,\ell} \otimes z_\ell + \sum_{\ell=1}^p \tilde{b}_{\cdot,\ell} \otimes w_\ell \right) + Y \exp_d(\theta S) \tilde{c} \quad (12)$$

where $\tilde{a} \in \mathbb{C}^{(d+1) \times (r+1)}$ and $\tilde{b} \in \mathbb{C}^{(d+1) \times p}$ are defined as follows

$$\tilde{a}_{i,r+1} := 0, \quad i = 2, \dots, d+1, \quad (13a)$$

$$\tilde{a}_{i+1,\ell} := \tilde{a}_{i,\ell}/i, \quad i = 1, \dots, d; \ell = 1, \dots, r, \quad (13b)$$

$$\tilde{b}_{i+1,\ell} := \tilde{b}_{i,\ell}/i, \quad i = 1, \dots, d; \ell = 1, \dots, p. \quad (13c)$$

Proof With the notation

$$x_i := \sum_{\ell=1}^r \bar{a}_{i+1,\ell} z_\ell + \sum_{\ell=1}^p \bar{b}_{i+1,\ell} w_\ell, \quad i = 0, \dots, d-1, \quad (14)$$

and $x := \text{vec}(x_0, \dots, x_{d-1}) \in \mathbb{C}^{dn}$, $\psi(\theta)$ defined in (3) can be expressed as

$$\psi(\theta) = P_{d-1}(\theta)x + Y \exp_{d-1}(\theta S) \tilde{c}. \quad (15)$$

By invoking [12, theorem 4.2] and using (14), we can express the action of the operator as

$$\mathcal{B}\psi(\theta) = P_d(\theta)x_+ + Y \exp_d(\theta S) \tilde{c} \quad (16)$$

where $x_+ := \text{vec}(x_{+,0}, \dots, x_{+,d}) \in \mathbb{C}^{(d+1)n}$ with

$$x_{+,i} := \sum_{\ell=1}^r \frac{\bar{a}_{i,\ell}}{i} z_\ell + \sum_{\ell=1}^p \frac{\bar{b}_{i,\ell}}{i} w_\ell, \quad i = 1, \dots, d, \quad x_{+,0} := -M_0^{-1} \left(\sum_{i=1}^d M_i x_{+,i} + \mathbb{M}_d(Y, S) \tilde{c} \right).$$

Substituting $x_{+,i}$ into $x_{+,0}$ we obtain $x_{+,0} = \tilde{z}$ given in (9). Using (13) and (11) we can express x_+ in terms of \tilde{a} and \tilde{b} and we conclude by substituting this expression for x_+ in (16).

Remark 4 The assumption (10) can only be satisfied if $r + p \leq n$. This is the generic case that we consider in this paper. Our focus is on large-scale NEPs and, in Sect. 5.1, we introduce approximations that avoid r from being large. The hypothesis $\lambda(S) \subseteq \Omega$ is necessary in order to define $\mathbb{M}_d(Y, S)$ that is used to compute \tilde{z} in Eq. (9).

2.2 Orthogonality

The tensor structured functions in the TIAR factorization (Definition 2) are orthonormal. In order to impose the orthogonality in our algorithms, we now present a theory which characterizes the orthonormality of tensor structured functions in terms of their coefficients. In particular, we derive the theory necessary to carry out the Gram–Schmidt orthogonalization. Since most of the orthogonalization procedures involve linear combinations of vectors, we start with the observation that linear combinations of tensor structured functions carry over directly to the coefficients.

Observation 5 (Linearity with respect to coefficients) *Given the tensor structured function $\Psi_k(\theta)$ represented by the matrices $(Z, W, Y, S) \in \mathbb{C}^{n \times r} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{p \times p}$ with coefficients $(a, b, C) \in \mathbb{C}^{d \times k \times r} \times \mathbb{C}^{d \times k \times p} \times \mathbb{C}^{p \times k}$ and $\tilde{\Psi}_{\tilde{k}}(\theta)$ represented by the same matrices but with coefficients $(\tilde{a}, \tilde{b}, \tilde{C}) \in \mathbb{C}^{d \times \tilde{k} \times r} \times \mathbb{C}^{d \times \tilde{k} \times p} \times \mathbb{C}^{p \times \tilde{k}}$. Let $M \in \mathbb{C}^{k \times t}$ and $N \in \mathbb{C}^{\tilde{k} \times t}$, then the function $\hat{\Psi}_{\hat{k}}(\theta) := \Psi_k(\theta)M + \tilde{\Psi}_{\tilde{k}}(\theta)N$ is also a*

tensor structured function represented by the same matrices and coefficients $(\hat{a}, \hat{b}, \hat{C})$, where for $\ell = 1, \dots, r$ and $\ell' = 1, \dots, p$

$$\hat{a}_{::,\ell} = a_{::,\ell} M + \tilde{a}_{::,\ell} N, \quad \hat{b}_{::,\ell'} = b_{::,\ell'} M + \tilde{b}_{::,\ell'} N, \quad \hat{C} = CM + \tilde{C}N.$$

The above relation can be expressed also in terms of the unfolding in the first dimension, in particular for $i = 1, \dots, d$ it holds

$$\hat{a}_{i::}^T = a_{i::}^T M + \tilde{a}_{i::}^T N, \quad \hat{b}_{i::}^T = b_{i::}^T M + \tilde{b}_{i::}^T N.$$

Theorem 6 (Gram–Schmidt orthogonalization) *Let $(Z, W, Y, S) \in \mathbb{C}^{n \times r} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{p \times p}$ be the matrices and $(\bar{a}, \bar{b}, \bar{c}) \in \mathbb{C}^{d \times r} \times \mathbb{C}^{d \times p} \times \mathbb{C}^p$ and $(a, b, c) \in \mathbb{C}^{d \times k \times r} \times \mathbb{C}^{d \times k \times p} \times \mathbb{C}^{p \times k}$ be the coefficients that represent $\psi(\theta)$ given in (3) and $\Psi_k(\theta)$ given in (5). Assume that $\Psi_k(\theta)$ is orthogonal. Let*

$$h = \sum_{\ell=1}^r (a_{::,\ell})^H \bar{a}_{::,\ell} + \sum_{\ell=1}^p (b_{::,\ell})^H \bar{b}_{::,\ell} + \sum_{i=d}^{\infty} C^H \frac{(S^i)^H Y^H Y S^i}{(i!)^2} \bar{c}. \quad (17)$$

The normalized result of the Gram–Schmidt orthogonalization of $\psi(\theta)$ against the columns of $\Psi_k(\theta)$ is

$$\psi^\perp(\theta) = P_{d-1}(\theta) \left(\sum_{\ell=1}^r a_{::,\ell}^\perp \otimes z_\ell + \sum_{\ell=1}^p b_{::,\ell}^\perp \otimes w_\ell \right) + Y \exp_{d-1}(\theta S) c^\perp$$

where

$$c^\perp = \bar{c} - Ch, \quad (18a)$$

$$a_{::,\ell}^\perp = \bar{a}_{::,\ell} - a_{::,\ell} h, \quad \ell = 1, \dots, r, \quad (18b)$$

$$b_{::,\ell}^\perp = \bar{b}_{::,\ell} - b_{::,\ell} h, \quad \ell = 1, \dots, p. \quad (18c)$$

The vector h contains the orthogonalization coefficients, i.e., $h_j = \langle \psi, \psi_j \rangle$. Moreover

$$\|\psi^\perp\| := \beta = \sqrt{\|a^\perp\|_F^2 + \|b^\perp\|_F^2 + \sum_{i=d}^{\infty} \frac{(c^\perp)^H (S^i)^H Y^H Y S^i c^\perp}{(i!)^2}}. \quad (19)$$

Proof Let us define $h_j := \langle \psi, \psi_j \rangle$ for $j = 1, \dots, k$. The orthogonal complement, computed with the Gram–Schmidt process, is $\psi^\perp(\theta) = \psi(\theta) - \Psi_k(\theta)h$. Using the Observation 5 we obtain directly (18).

We express $\psi(\theta)$ as (15) and, the columns of Ψ_k as

$$\psi_j(\theta) = P_{d-1}(\theta) x^{(j)} + Y \exp_{d-1}(\theta S) c_j \quad (20)$$

where $x^{(j)} := \text{vec}(x_0^{(j)}, \dots, x_{d-1}^{(j)}) \in \mathbb{C}^{dn}$, with

$$x_i^{(j)} := \sum_{\ell=1}^r a_{i+1,j,\ell} z_\ell + \sum_{\ell=1}^p b_{i+1,j,\ell} w_\ell, \quad i = 0, \dots, d-1. \quad (21)$$

By applying [12, Equation (4.32)] we obtain

$$h_j = \sum_{i=0}^{d-1} (x_i^{(j)})^H x_i + c_j^H \sum_{i=d}^{\infty} \frac{(S^i)^H Y^H Y S^i}{(i!)^2} \bar{c}, \quad j = 1, \dots, k. \quad (22)$$

We now substitute (14) and (21) in (22) and, by using the orthonormality of the vectors $z_1, \dots, z_r, w_1, \dots, w_p$, we find that

$$h_j = \sum_{\ell=1}^r (a_{:,j,\ell})^H \bar{a}_{:, \ell} + \sum_{\ell=1}^p (b_{:,j,\ell})^H \bar{b}_{:, \ell} + \sum_{i=d}^{\infty} c_j^H \frac{(S^i)^H Y^H Y S^i}{(i!)^2} \bar{c}, \quad j = 1, \dots, k.$$

Those are the elements of the right-hand side of (17). Using that $\|\psi^\perp\|^2 = \langle \psi^\perp, \psi^\perp \rangle$, and repeating the same reasoning, we have

$$\|\psi^\perp\|^2 = \sum_{\ell=1}^r (a_{:, \ell}^\perp)^H a_{:, \ell}^\perp + \sum_{\ell=1}^p (b_{:, \ell}^\perp)^H b_{:, \ell}^\perp + \sum_{i=d}^{\infty} \frac{(c^\perp)^H (S^i)^H Y^H Y S^i c^\perp}{(i!)^2}$$

which proves (19).

3 Restarting for TIAR in an abstract setting

3.1 A TIAR expansion algorithm in finite dimension

One algorithmic component common in many restart procedures is the expansion of an Arnoldi-type factorization. The standard way to expand Arnoldi-type factorizations (as, e.g., described in [29, Sect. 3]) requires the computation of the action of the operator/matrix and orthogonalization. We now show how we can carry out an expansion of the infinite dimensional TIAR factorization (7) by only using operations on matrices and vectors (of finite dimension).

In the previous section we characterized the action of the operator \mathcal{B} and orthogonalization of tensor structured functions. Notice that we have derived the orthogonalization procedure only for tensor structured functions that have the same degree in the polynomial part. The expansion of a TIAR factorization $(\Psi_k, \underline{H}_{k-1})$, involves the orthogonalization of the tensor structured function $\mathcal{B}\psi_k$ (computed using the Theorem 3) against the columns of Ψ_k . If the degree of Ψ_k is $d-1$ then the degree

of $\mathcal{B}\psi_k$ is d . Therefore, in order to perform the orthogonalization, we have to represent ψ_k as a tensor structured function with degree d . Starting from (5) we rewrite ψ_k as

$$\psi_k(\theta) = P_{d-1}(\theta) \left(\sum_{\ell=1}^r a_{:, :, \ell} \otimes z_\ell + \sum_{\ell=1}^p b_{:, :, \ell} \otimes w_\ell \right) + \frac{Y S^d C}{d!} \theta^d + Y \exp_d(\theta S) C. \quad (23)$$

We define

$$E := \frac{W^H Y S^d C}{d!}, \quad \begin{array}{ll} a_{d+1, j, \ell} := 0 & \ell = 1, \dots, r+1, \\ b_{d+1, j, \ell} := e_{\ell, j} & \ell = 1, \dots, p, \end{array} \quad (24)$$

for $j = 1, \dots, k$. Since $\text{span}(W) = \text{span}(Y)$ and, since W is orthogonal, we have that $Y = W W^H Y$. Hence, using this relation and (24), the function ψ_k in (23) can be expressed as

$$\psi_k(\theta) = P_d(\theta) \left(\sum_{\ell=1}^{r+1} a_{:, :, \ell} \otimes z_\ell + \sum_{\ell=1}^p b_{:, :, \ell} \otimes w_\ell \right) + Y \exp_d(\theta S) C.$$

These results can be directly combined to expand a TIAR factorization. The resulting algorithm is summarized in Algorithm 1. The action of the operator \mathcal{B} , described in Theorem 3, is expressed in Steps 2–4. Step 5 corresponds to increasing the degree of the TIAR factorization as described in (23) and (24). The orthogonalization of the new function, carried out by Theorem 6, is expressed in Steps 6–8 and the orthonormalization coefficients are then stored in the Hessenberg matrix $\underline{H}_{\tilde{k}}$. We can truncate the sum in (8), (19) and (17) analogously to [12]. Due to the representation of ψ_k as tensor structured function, the expansion with one column corresponds to an expansion of all the coefficients representing ψ_k . This expansion is visualized in Fig. 1.

3.2 The Krylov–Schur decomposition for TIAR factorizations

We briefly recall the reasoning for the Krylov–Schur type restarting [29]. This procedure can be carried out with operations on matrices and vectors of finite size. Let $(\psi_{m+1}, \underline{H}_m)$ be a TIAR factorization. Let $P \in \mathbb{C}^{m \times m}$ be such that $P^H \underline{H}_m P$ is triangular (ordered Schur factorization). Then,

$$\mathcal{B}\hat{\psi}_m = \hat{\psi}_{m+1} \begin{pmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ & R_{2,2} & R_{2,3} \\ a_1^H & a_2^H & a_3^H \end{pmatrix}, \quad (25)$$

where $\hat{\psi}_{m+1} = [\psi_m P, \psi_{m+1}]$. The matrix P is selected in a way that the matrix $R_{1,1} \in \mathbb{C}^{p_\ell \times p_\ell}$ contains the converged Ritz values, the matrix $R_{2,2} \in \mathbb{C}^{(p-p_\ell) \times (p-p_\ell)}$

Algorithm 1: Expand TIAR factorization (tensor structured functions)

input : A TIAR factorization $(\Psi_{k+1}, \underline{H}_k)$ represented by
 $(Z, W, Y, S) \in \mathbb{C}^{n \times r} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{p \times p}$ and
 $(a, b, C) \in \mathbb{C}^{d \times (k+1) \times r} \times \mathbb{C}^{d \times (k+1) \times p} \times \mathbb{C}^{p \times (k+1)}$ with S invertible and the final length m .

output: A TIAR factorization $(\Psi_{m+1}, \underline{H}_m)$ represented by
 $(Z, W, Y, S) \in \mathbb{C}^{n \times \tilde{r}} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{p \times p}$ and
 $(a, b, C) \in \mathbb{C}^{\tilde{d} \times (m+1) \times \tilde{r}} \times \mathbb{C}^{\tilde{d} \times (m+1) \times p} \times \mathbb{C}^{p \times (m+1)}$ where $\tilde{r} = r + m - k$ and $\tilde{d} = d + m - k$.

```

1 Set  $\tilde{r} = r, \tilde{d} = d$ 
  for  $\tilde{k} = \tilde{k} + 1, \dots, m + 1$  do
2   Compute  $\tilde{z}$  using (9), where  $\tilde{a} = a_{:, \tilde{k}, :}, \tilde{b} = b_{:, \tilde{k}, :}$  and  $\tilde{c} = c_{\tilde{k}}$ 
3   Compute  $z_{\tilde{r}+1}$  as in Theorem 3 and increase  $\tilde{r} = \tilde{r} + 1$ 
4   Set  $\tilde{a}, \tilde{b}$  as in (11) and (13) and  $\tilde{c} = S^{-1} \tilde{c}$ 
5   Compute  $E$  and expand the tensors  $a$  and  $b$  as (24) and increase  $\tilde{d} = \tilde{d} + 1$ 
6   Compute  $h$  using (17), where  $\tilde{a} = \tilde{a}, \tilde{b} = \tilde{b}$  and  $\tilde{c} = \tilde{c}$ 
7   Compute  $a^\perp, b^\perp, c^\perp$  using (18) and  $\beta$  using (19) and extend
                                     
$$\underline{H}_{\tilde{k}} = \begin{pmatrix} \underline{H}_{\tilde{k}-1} & h \\ 0 & \beta \end{pmatrix} \in \mathbb{C}^{(\tilde{k}+1) \times \tilde{k}}$$

8   Expand  $c_{\tilde{k}+1}^\perp := c^\perp / \beta$  and  $a_{:, \tilde{k}+1, :} := a^\perp / \beta$  and  $b_{:, \tilde{k}+1, :} := b^\perp / \beta$ .
  end
```

contains the wanted Ritz values and the matrix $R_{3,3} \in \mathbb{C}^{(m-p) \times (m-p)}$ contains the Ritz values that we want to purge. From (25) we find that

$$\mathcal{B}\tilde{\Psi}_p = \tilde{\Psi}_{p+1} \begin{pmatrix} R_{1,1} & R_{1,2} \\ a_1^H & a_2^H \end{pmatrix}, \quad (26)$$

where $\tilde{\Psi}_{p+1} := [\hat{\Psi}_m I_{m,p}, \psi_{m+1}] = [\hat{\Psi}_p, \psi_{m+1}]$. By using a product of Householder reflectors, we compute a matrix $Q \in \mathbb{C}^{p \times p}$ such that

$$\mathcal{B}\tilde{\Psi}_p = \tilde{\Psi}_{p+1} \begin{pmatrix} R_{1,1} & F \\ a_1^H & \beta e_{p-p_\ell}^H \end{pmatrix}, \quad (27)$$

where $\tilde{\Psi}_{p+1} = [\tilde{\Psi}_p Q, \psi_{m+1}]$ and H has an upper Hessenberg form. Since we want to lock the Ritz values in the matrix $R_{1,1}$, we replace in (27) the vector a_1 with zeros introducing an error $\mathcal{O}(\|a_1\|)$. With this approximation, (27) is the wanted TIAR factorization of length p .

Observation 7 In the TIAR factorization (27), $(\tilde{\Psi}_{p_\ell}, R_{1,1})$ is an approximation of an invariant pair, i.e., $\mathcal{B}\tilde{\Psi}_{p_\ell} = \tilde{\Psi}_{p_\ell} R_{1,1}$. Moreover $(\tilde{\Psi}_{p_\ell}(0), R_{1,1}^{-1})$ is an approximation of an invariant pair of the original NEP in the sense of [16, Definition 1], see [12, Theorem 2.2].

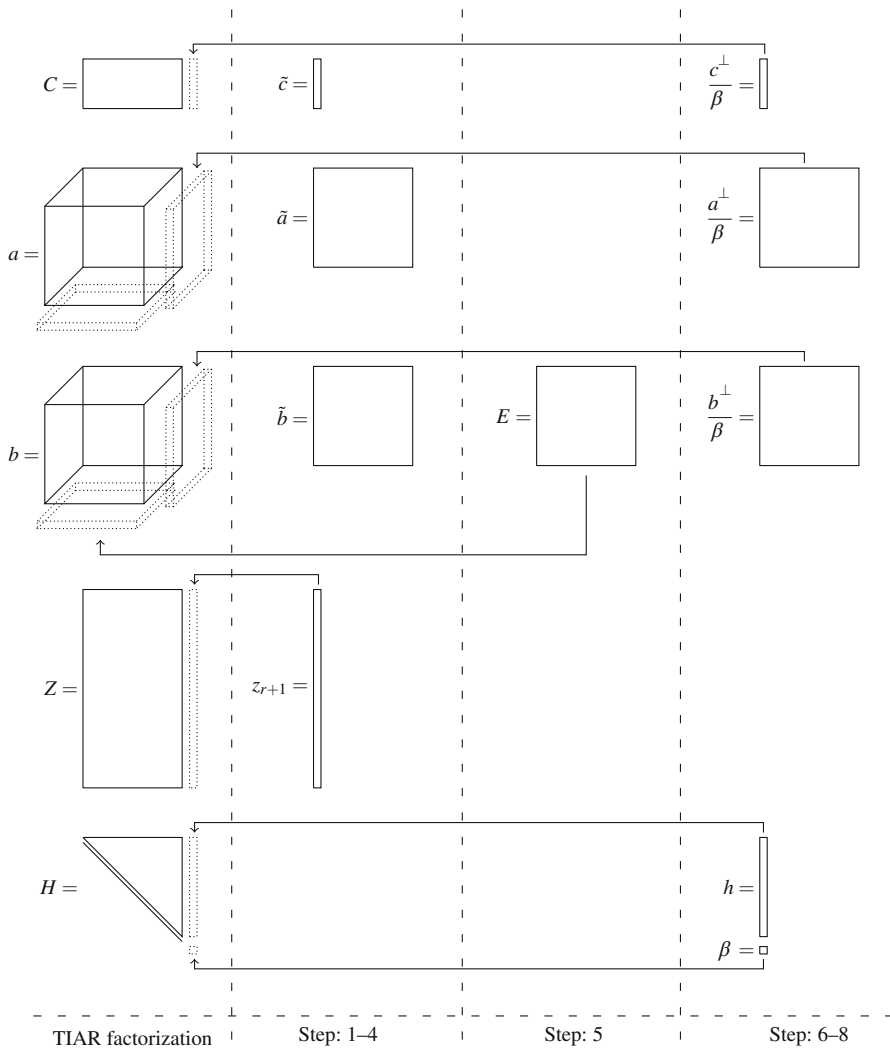


Fig. 1 Graphical illustration of the expansion of the tensor structured function that represents the TIAR factorization in Algorithm 1

3.3 Two structured restarting approaches

The standard restart approach for TIAR using Krylov–Schur type restarting, as described in the previous section, involves expansions and manipulations of the TIAR factorization. Due to the linearity of tensor structured functions with respect to the coefficients, described in Observation 5, the manipulations for Ψ_m leading to Ψ_p can be directly carried out on the coefficients representing Ψ_m . Unfortunately, due to the implicit representation of Ψ_m , the memory requirements are not substantially reduced

since the basis matrix $Z \in \mathbb{C}^{n \times r}$ is not modified in the manipulations. The size of the basis matrix Z is the same before and after the restart.

We propose two ways of further exploiting the structure of the functions in order to avoid a dramatic increase in the required memory resources.

- Semi-explicit restart (Sect. 4): An invariant pair can be completely represented by exponentials and therefore it does not contribute to the memory requirement for Z . The fact that invariant pairs are exponentials was exploited in the restart in [12]. We show how the ideas in [12] can be carried over to tensor structured functions. More precisely, the adaption of [12] involves restarting the iteration with a locked pair, i.e., only the first p_ℓ columns of (27), and a function f constructed in a particular way. The approach is outlined in Algorithm 2 with details specified in Sect. 4.
- Implicit restart (Sect. 5): By only representing polynomials, we show that the TIAR factorization has a particular structure such that it can be accurately approximated. This allows us to carry out a full implicit restart, and subsequently approximate the TIAR factorization reducing the size of the matrix Z . The adaption is given in Algorithm 3. The approximation of the TIAR factorization in Step 6 is specified in Algorithm 4 and derived in Sect. 5, including an error analysis.

Algorithm 2: Semi-explicit restarting for TIAR in operator setting

input : A normalized tensor structured function ψ represented by
 $(Z, W, Y, S) \in \mathbb{C}^{n \times r} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{p \times p}$ with S invertible,
 $(a, b, C) \in \mathbb{C}^{d \times 1 \times r} \times \mathbb{C}^{d \times 1 \times p} \times \mathbb{C}^p$ and the maximum length m of the TIAR factorization
 and number of wanted eigenvalues p_{\max}

output: p_{\max} eigenvalues of \mathcal{B}

- 1 Set $\Psi^{(1)} = [\psi], \underline{H}^{(1)}$ empty matrix of size 1×0 and $j = 1$
- while** $p_\ell \leq p_{\max}$ **do**
- 2 Expand the TIAR factorization $(\Psi^{(j)}, \underline{H}^{(j)})$ to length m using Algorithm 1
- 3 Compute the p_ℓ converged Ritz pairs and $P, R_{i,j}$ and a_i given in (25)
- 4 Compute the matrices Q, F and H given in (27)
- 5 Set $S := \begin{pmatrix} R_{1,1} & F \\ & H \end{pmatrix}^{-1}$ and compute \hat{Y} , as in (28), where $\Psi_m = \Psi^{(j)}$
- 6 Compute and impose the structure to the invariant pair $(\tilde{\Psi}, R_{1,1})$ by setting $C = I_{p,p_\ell}$
 and $\tilde{\Psi} = \hat{Y} \exp(S\theta)C$
- 7 Set $f = \hat{Y} \exp(\theta S) e_{p_\ell+1}$ and compute h using (17), where $\tilde{a} = \tilde{b} = 0$, and $\tilde{c} = e_{p_\ell+1}$
- 8 Set c^\perp using (18a) and $f^\perp = \hat{Y} \exp(\theta S) c^\perp$
- 9 Compute β using (19), where $a^\perp = b^\perp = 0$
- 10 Set $C = [C, c^\perp/\beta], \Psi^{(j+1)} = [\tilde{\Psi}, f^\perp/\beta] = \hat{Y} \exp(\theta S)C$ and $\underline{H}^{(j+1)} = \begin{pmatrix} R_{1,1} \\ 0 \end{pmatrix}$ and
 $j = j + 1$
- end**
- 11 Return the eigenvalues of $R_{1,1}$

Algorithm 3: Implicit restart for TIAR in operator setting

input : A normalized tensor structured function ψ represented by $(Z, 0, 0, I) \in \mathbb{C}^{n \times r} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{p \times p}$ and $(a, 0, 0) \in \mathbb{C}^{d \times 1 \times r} \times \mathbb{C}^{d \times 1 \times p} \times \mathbb{C}^p$ and the maximum length m of the TIAR factorization and number of wanted eigenvalues p_{\max}

output: p_{\max} eigenvalues of \mathcal{B} .

- 1 Set $\Psi^{(1)} = [\psi]$, $\underline{H}^{(1)}$ empty matrix of size 1×0 and $j = 1$
- while** $p_\ell \leq p_{\max}$ **do**
- 2 Expand the TIAR factorization $(\Psi^{(j)}, \underline{H}^{(j)})$ to length m using Algorithm 1
- 3 Compute the p_ℓ converged Ritz pairs and $P, R_{i,j}$ and a_i given in (25)
- 4 Compute the matrices Q, F, H and β given in (27)
- 5 Set $\Psi^{(j+1)} = \Psi^{(j)}[I_{m+1,m} P I_{m,p} Q, e_{m+1}]$, $\underline{H}^{(j+1)} = \begin{pmatrix} R_{1,1} & F \\ & H \\ & \beta e_{p-p_\ell} \end{pmatrix}$
- 6 Approximate the TIAR factorization, Algorithm 4
- end**
- 7 Return the eigenvalues of $R_{1,1}$

4 Tensor structure exploitation for the semi-explicit restart

The IAR restart approach in [12] is based on representing functions as sums of exponentials and polynomials. An attractive feature of that approach is that the invariant pairs can be exactly represented and locking can be efficiently incorporated. Due to the explicit storage of polynomial coefficients, the approach still requires considerable memory. We here show that, by representing the functions implicitly as a tensor structured functions (3), we can maintain all the advantages but improve performance (both in memory and CPU-time). This construction is equivalent to [12], but more efficient.

The expansion of the TIAR factorization with tensor structured functions (as described in Algorithm 1), combined with the locking procedure (as described in Sect. 3.2), and imposing the structure to the invariant pair as in [12], results in Algorithm 2. Steps 3–10 follow the procedure described in [12] adapted for tensor structured functions. In particular Steps 3–6 consist in extracting and imposing the structure on the invariant pair $(\bar{\Psi}, R_{1,1})$. In Steps 7–9 a new starting function f is selected and orthogonalized with respect to $\bar{\Psi}$ and in Step 10 the new TIAR factorization is defined.

The computation of the invariant pair $(\bar{\Psi}, R_{1,1})$ and of the new starting function f involves the matrix \hat{Y} [12, Equation (5.11)]. This matrix can be extracted from the tensor structured representation as follows. By using Observation 5 with $M := P I_{m,p} Q$, we obtain

$$\hat{Y} := \Psi_m(0)M = P_{d-1}(0) \left(\sum_{\ell=1}^r a_{:, :, \ell} M \otimes z_\ell + \sum_{\ell=1}^p b_{:, :, \ell} M \otimes w_\ell \right) + Y \exp_{d-1}(0) C M \quad (28a)$$

$$= \sum_{\ell=1}^r a_{1, :, \ell} M \otimes z_\ell + \sum_{\ell=1}^p b_{1, :, \ell} M \otimes w_\ell. \quad (28b)$$

5 Tensor structure exploitation for the implicit polynomial restart

In contrast to the procedure in Sect. 4, where the main idea was to do locking with exponentials and restart with a factorization of length p_ℓ , we now propose a fully implicit procedure involving a factorization of length p . In this setting we use $C = 0$, i.e., we only represent polynomials with the tensor structured functions. This allows us to derive theory for the structure of the coefficient matrix, which shows how to approximate the TIAR factorization. This procedure is summarized in Algorithm 3.

The approximation in Step 6 is done in order to reduce the growth in memory requirements for the representation. The approximation technique is derived in the following subsections and summarized in Algorithm 4.

Our approximation approach is based on an approximation with a truncated singular value decomposition and a degree reduction. A compression with a truncated singular value decomposition was also made for the compact representations in CORK [35] and TOAR [17]. Our specific setting allows us to prove bounds on the error introduced by the approximations (Sects. 5.1, 5.2). We also show the effectiveness by proving a bound on the decay of the singular values (Sect. 5.3).

Similar to the semi-explicit restart, the approximations that we use in the implicit restart are based on the fact that the invariant pairs of \mathcal{B} are represented by exponential functions. In particular, if (Ψ, Λ^{-1}) is an invariant pair of \mathcal{B} , then $\Psi(\theta) = Y \exp(\Lambda\theta)$, see [12, Theorem 2.2]. Hence, Ψ expressed in a monomial basis, corresponds to a Taylor series expansion with coefficients having a fast decay. In this section we illustrate that, under certain hypothesis, the functions generated by Algorithm 1 and Algorithm 3 have similar decay properties. We introduce a definition that describes the decay in the magnitude of the coefficients of the power series, represented by the tensor structured function Ψ_k (5), in terms of the representing polynomial coefficients a and b , i.e.,

$$C(\Psi_k) := \min \left\{ \beta \in \mathbb{R} : \|a_{i,:,:}\|_F + \|b_{i,:,:}\|_F \leq \frac{\beta}{(i-1)!} \text{ for } i = 1, \dots, d \right\}. \quad (29)$$

Observe that, if (Ψ, Λ^{-1}) is an invariant pair, due to the decay of the power series coefficients of the exponential function, we have

$$C(\Psi) = \|\Lambda\|_F. \quad (30)$$

Theorem 8 *Let ψ be a tensor structured function (3) such that $\bar{c} = 0$ and $\|\psi\| = 1$. Let $Z \in \mathbb{C}^{n \times r}$ be a matrix and $a \in \mathbb{C}^{(d+k+1) \times (k+1) \times r}$ be a set of coefficients that represent the tensor structured function Ψ_{k+1} and $\underline{H}_k \in \mathbb{C}^{(k+1) \times k}$ be such that $(\Psi_{k+1}, \underline{H}_k)$ is a TIAR factorization obtained by using ψ as a starting function. Then*

$$C(\Psi_{k+1}) \leq \frac{\sqrt{d-1}}{\kappa(L_k)} \|L_k\|_F C(\psi) + \kappa(L_k), \quad (31)$$

where $L_k := [v, C_{k+1}v, \dots, C_{k+1}^{k+1}v]$, C_{k+1} is defined in [14, Eq. 29] and $v = \sum_{\ell=1}^r \bar{a}_{:, \ell} \otimes z_\ell$ and κ is the condition number with respect to the Frobenius norm.

Proof Let $\Phi_{k+1}(\theta) = (\psi(\theta), \mathcal{B}\psi(\theta), \dots, \mathcal{B}^k\psi(\theta))$. By applying Theorem 3, we obtain

$$\Phi_{k+1}(\theta) = P_k(\theta) \left(\sum_{\ell=1}^r \hat{a}_{:, \ell} \otimes z_\ell \right)$$

where, for $\ell = 1, \dots, r$ we have $\hat{a}_{:, \ell} = DT_\ell \in \mathbb{C}^{(d+k+1) \times (k+1)}$ with $D \in \mathbb{C}^{d \times d}$ being a diagonal matrix with elements $D_{i,i} = 1/(i-1)!$ and $T_\ell \in \mathbb{C}^{(d+k+1) \times (k+1)}$ is a Toeplitz matrix with leading column $[\bar{a}_{1,\ell}, 1!\bar{a}_{2,\ell}, \dots, (d-1)!\bar{a}_{d,\ell}, 0, \dots, 0]$ and with leading row $[\bar{a}_{1,\ell}, \hat{a}_{1,2,\ell}, \dots, \hat{a}_{1,k+1,\ell}]$. The structure of $\hat{a}_{:, \ell}$ implies that

$$\begin{aligned} \|\hat{a}_{i, : \ell}\|_F^2 &= \frac{1}{(i-1)!^2} \left(\sum_{t=2}^{\min(d,i)} \bar{a}_{t,\ell}^2 (t-1)!^2 + \sum_{t=1}^{k-i+2} \hat{a}_{1,t,\ell}^2 \right) \\ &\leq \frac{(d-1)C(\psi)^2 + \|\hat{a}_{1, : \ell}\|_F^2}{(i-1)!^2} \end{aligned}$$

and by using the properties of the Frobenius norm we have

$$\|\hat{a}_{i, : \ell}\|_F^2 \leq \frac{(d-1)C(\psi)^2 + \|\hat{a}_{1, : \ell}\|_F^2}{(i-1)!^2}. \quad (32)$$

Since (Ψ_{k+1}, H_k) forms a TIAR factorization, it holds $\text{span}(\Phi_{k+1}) = \text{span}(\Psi_{k+1})$. Therefore it exists an invertible matrix $R \in \mathbb{C}^{(k+1) \times (k+1)}$ such that $\Phi_{k+1} = \Psi_{k+1}R$. By using Observation 5 and the sub-multiplicativity of the Frobenius norm we have that for $i = 1, \dots, k+1$

$$\|a_{i, : \ell}\|_F = \|\hat{a}_{i, : \ell} R^{-1}\|_F \leq \|\hat{a}_{i, : \ell}\|_F \|R^{-1}\|_F. \quad (33)$$

By combining (33), (32) and sub-additivity of the square root we obtain

$$\begin{aligned} \|a_{i, : \ell}\|_F &\leq \frac{\sqrt{(d-1)C(\psi)^2 + \|\hat{a}_{1, : \ell}\|_F^2}}{(i-1)!} \|R^{-1}\|_F \\ &\leq \frac{\sqrt{d-1}C(\psi) \|R^{-1}\|_F + \|a_{1, : \ell}\|_F \kappa(R)}{(i-1)!}. \end{aligned} \quad (34)$$

Since Ψ_k is orthonormal and (6) we have that $\|a_{1, : \ell}\|_F \leq 1$.

Let L_k^\dagger denote the pseudo-inverse of L_k . In order to show (31), we now show that $\|R^{-1}\|_F = \|L_k^\dagger\|_F$ and $\kappa(R) = \kappa(L_k)$ where $\kappa(L_k) = \|L_k\|_F \|L_k^\dagger\|_F$. Due to the equivalence of TIAR and IAR and the companion matrix interpretation of IAR [14, theorem 6], we have that TIAR is equivalent to using the Arnoldi method on the matrix C_{k+1} with starting vector $v = \sum_{\ell=1}^r \bar{a}_{:, \ell} z_\ell$. More precisely, the relation $\Phi_{k+1} = \Psi_{k+1}R$ can be written in terms of vectors as $L_{k+1} = V_{k+1}R$ where the first column of V_{k+1} and L_{k+1} is $v = \sum_{\ell=1}^r \bar{a}_{:, \ell} \otimes z_\ell$ and $L_k = [v, C_{k+1}v, \dots, C_{k+1}^k v]$.

By using the orthogonality of V_{k+1} we conclude that $\|R^{-1}\|_F = \|L_k^\dagger\|_F$ and $\kappa(R) = \kappa(L_k)$.

The approximations that we introduce in the next sections (required in Step 6 of Algorithm 3) are based on the assumption that the tensor structured functions $\Psi^{(j)}$ are such that the decay constant $C(\Psi^{(j)})$ is small. Theorem 8 shows that this constant remains small after the TIAR expansion in Step 2 if $\kappa(L_k)$ is not large. However, the condition number of the Krylov matrix L_k can be large, see [4]. This does not necessarily imply that the decay constant is large. Notice that if (Ψ_k, H_k) is an invariant pair, L_k has linearly dependent columns and $\kappa(L_k)$ is infinite. Analogously, if (Ψ_k, H_k) is (in this sense) close to an invariant pair, we expect $\kappa(L_k)$ to be large. Hence, in these situations, the right-hand side of (31) is expected to be large. However, the decay constant is not expected to be large, since the decay constant for an invariant pair is given by (30). Note that the decay is also preserved in the operations associated with the restart. After the Ritz-value selection (Step 3–4) the new TIAR factorization is computed in Step 5. Since $\Psi^{(j+1)}$ is obtained through a unitary transformation from $\Psi^{(j)}$, by using the properties of the Frobenius norm, we get $C(\Psi^{(j+1)}) \leq \sqrt{p + 1} C(\Psi^{(j)})$.

5.1 Approximation by SVD compression

Given a TIAR factorization with basis function Ψ_k we now show (in the following theorem) how we can approximate the basis function with less memory, by using a thinner basis matrix $Z \in \mathbb{C}^{n \times \tilde{r}}$ where \tilde{r} should be selected as small as possible. The theorem also shows how this approximation influences the approximation Ψ_k as well as the residual of the TIAR factorization. It turns out that the residual error is small if $\sigma_{\tilde{r}}$ is small, where $\sigma_1, \dots, \sigma_r$ are singular values associated with the coefficient tensor. This implies that \tilde{r} can be chosen small if we have a fast singular value decay. We characterize the decay of the singular values in Sect. 5.3.

Theorem 9 *Let $a \in \mathbb{C}^{(d+1) \times (k+1) \times r}$, $Z \in \mathbb{C}^{n \times r}$ be the coefficients and the matrix that represent the tensor structured function Ψ_{k+1} and let $\underline{H}_k \in \mathbb{C}^{(k+1) \times k}$ be such that $(\Psi_{k+1}, \underline{H}_k)$ is a TIAR factorization. Suppose $\{|z| \leq R\} \subseteq \Omega$ with $R > 1$. Let $A := [A_1, \dots, A_{d+1}] \in \mathbb{C}^{r \times (d+1) \times (k+1)}$ be the unfolding of the tensor a in the sense that $A_i = (a_{i, :, :})^T$. Given the singular value decomposition of A*

$$\begin{aligned} A &= [U_1, U] \text{diag}(\Sigma_1, \Sigma) [V_1^H, \dots, V_{d+1}^H] \\ \Sigma_1 &= \text{diag}(\sigma_1, \dots, \sigma_{\tilde{r}}), \quad \Sigma = \text{diag}(\sigma_{\tilde{r}+1}, \dots, \sigma_r), \end{aligned} \quad (35)$$

let

$$\tilde{Z} := ZU_1, \quad \tilde{A}_i := \Sigma_1 V_i^H \quad i = 1, \dots, d+1. \quad (36)$$

The tensor structured function $\tilde{\Psi}_{k+1}$ is defined by $\tilde{a} \in \mathbb{C}^{(d+1) \times (k+1) \times \tilde{r}}$ and $\tilde{Z} \in \mathbb{C}^{n \times \tilde{r}}$, with $\tilde{a}_{i, :, :} = \tilde{A}_i^T$. Then,

$$\|\Psi_{k+1} - \tilde{\Psi}_{k+1}\| \leq \sqrt{(d+1)(k+1)}\sigma_{\tilde{r}+1} \quad (37a)$$

$$\|\mathcal{B}\tilde{\Psi}_k - \tilde{\Psi}_{k+1}\underline{H}_k\| \leq \sqrt{k}(C_d + C_s)\sigma_{\tilde{r}+1} \quad (37b)$$

with $C_d := \gamma + \log(d+1) + (d+1)\|\underline{H}_k\|_2$ and $C_s := \|M_0^{-1}\|_2[(\gamma + \log(s+1))\max_{1 \leq i \leq s} \|M_i\|_2 + \max_{|\lambda|=R} \|M(\lambda)\|_2]$ where $\gamma \approx 0.57721$ is the Euler–Mascheroni constant and $s := \min\{s \in \mathbb{N} : C(\Psi_k)\sqrt{k}(d-s)/R^s \leq \sigma_{\tilde{r}+1}\}$.

Proof The proof of (37a) is based on the construction of a difference function $\hat{\Psi}_{k+1} = \Psi_{k+1} - \tilde{\Psi}_{k+1}$ as follows. We define

$$\begin{aligned} \hat{Z} &:= ZU, & \hat{A}_i &:= \Sigma V_i^H, \\ X_i &:= ZA_{i+1}, & \hat{X}_i &:= \hat{Z}\hat{A}_{i+1}, & \tilde{X}_i &:= \tilde{Z}\tilde{A}_{i+1}, \\ X &:= [X_0^H \dots X_d^H]^H, & \hat{X} &:= [\hat{X}_0^H, \dots, \hat{X}_d^H]^H, & \tilde{X} &:= [\tilde{X}_0^H, \dots, \tilde{X}_d^H]^H. \end{aligned}$$

Hence, we can express $\Psi_{k+1}(\theta) = P_d(\theta)X$, $\tilde{\Psi}_{k+1}(\theta) = P_d(\theta)\tilde{X}$ and $\hat{\Psi}_{k+1}(\theta) = P_d(\theta)\hat{X}$. By using (6) and $\|\hat{X}_i\|_F^2 \leq (k+1)\|\hat{X}_i\|_2^2 = (k+1)\|\hat{Z}\hat{A}_{i+1}\|_2^2 = (k+1)\|\hat{A}_{i+1}\|_2^2 = (k+1)\|\Sigma V_{i+1}^H\|_2^2 \leq (k+1)\|\Sigma\|_2^2 = (k+1)\sigma_{\tilde{r}+1}^2$ we obtain

$$\|\hat{\Psi}_{k+1}\|^2 = \sum_{i=0}^d \|\hat{X}_i\|_F^2 \leq (d+1)(k+1)\sigma_{\tilde{r}+1}^2$$

which proves (37a).

In order to show (37b) we first use that, since $(\Psi_{k+1}, \underline{H}_k)$ is a TIAR factorization, it holds $\|\mathcal{B}\tilde{\Psi}_k - \tilde{\Psi}_{k+1}\underline{H}_k\| = \|\mathcal{B}\hat{\Psi}_k - \hat{\Psi}_{k+1}\underline{H}_k\|$ and subsequently we use the decay of A_i and analyticity of M as follows. For notational convenience we define

$$Y_i := \hat{X}_i I_{k+1,k}, \quad \text{for } i = 0, \dots, d \quad (38)$$

and $Y := [Y_0^H \dots Y_d^H]^H$ such that we can express $\hat{\Psi}_k(\theta) = P_{d-1}(\theta)Y$. Using [12, theorem 4.2] for each column of $\hat{\Psi}_k(\theta)$, we get $\mathcal{B}\hat{\Psi}_k(\theta) = P_d(\theta)Y_+$ with

$$Y_{+,i+1} := \frac{Y_i}{i+1} \quad \text{for } i = 0, \dots, d-1 \quad \text{and} \quad Y_{+,0} := -M_0^{-1} \sum_{i=1}^d M_i Y_{+,i}.$$

By definition and (6) we have

$$\|\mathcal{B}\hat{\Psi}_k - \hat{\Psi}_{k+1}\underline{H}_k\| = \|P_d(\theta)Y_+ - P_d(\theta)\hat{X}\underline{H}_k\| = \|Y_+ - \hat{X}\underline{H}_k\|_F.$$

Moreover, by using the two-norm bound of the Frobenius norm, (38) and that $\|\hat{X}_i\|_2 \leq \sigma_{\tilde{r}+1}$,

$$\|Y_+ - \hat{X}\underline{H}_k\|_F \leq \sum_{i=0}^d \|Y_{+,i} - \hat{X}_i \underline{H}_k\|_F \leq \sqrt{k} \sum_{i=0}^d (\|Y_{+,i}\|_2 + \|\hat{X}_i\|_2 \|\underline{H}_k\|_2) \quad (39a)$$

$$= \sqrt{k} \left(\|Y_{+,0}\|_2 + \sum_{i=1}^d \|Y_{+,i}\|_2 + \sum_{i=0}^d \|\hat{X}_i\|_2 \|\underline{H}_k\|_2 \right) \quad (39b)$$

$$\leq \sqrt{k} \left(\|Y_{+,0}\|_2 + \sum_{i=1}^d \frac{\|\hat{X}_{i-1} I_{k+1,k}\|_2}{i} + \sum_{i=0}^d \|\hat{X}_i\|_2 \|\underline{H}_k\|_2 \right) \quad (39c)$$

$$\leq \sqrt{k} \left(\|Y_{+,0}\|_2 + \sum_{i=1}^d \frac{\sigma_{\tilde{r}+1}}{i} + \sum_{i=0}^d \sigma_{\tilde{r}+1} \|\underline{H}_k\|_2 \right) \quad (39d)$$

$$\leq \sqrt{k} [\|Y_{+,0}\|_2 + \sigma_{\tilde{r}+1} (\gamma + \log(d+1) + (d+1) \|\underline{H}_k\|_2)]. \quad (39e)$$

In the last inequality we use the Euler–Mascheroni inequality where γ is defined in [1, Formula 6.1.3]. It remains to bound $\|Y_{+,0}\|_2$. By using the definition of $Y_{+,0}$ and again applying the Euler–Mascheroni inequality we have that

$$\begin{aligned} \|Y_{+,0}\|_2 &\leq \|M_0^{-1}\|_2 \sum_{i=1}^d \|M_i\|_2 \frac{\|\hat{X}_{i-1} I_{k+1,k}\|_2}{i} \leq \|M_0^{-1}\|_2 \sum_{i=1}^d \|M_i\|_2 \frac{\|\hat{X}_{i-1}\|_2}{i} \\ &= \|M_0^{-1}\|_2 \left(\sum_{i=1}^s \|M_i\|_2 \frac{\|\hat{X}_{i-1}\|_2}{i} + \sum_{i=s+1}^d \|M_i\|_2 \frac{\|\hat{X}_{i-1}\|_2}{i} \right) \\ &\leq \|M_0^{-1}\|_2 \left(\sigma_{\tilde{r}+1} (\gamma + \log(s+1)) \max_{1 \leq i \leq s} \|M_i\|_2 + \sum_{i=s+1}^d \|M_i\|_2 \frac{\|\hat{X}_{i-1}\|_2}{i} \right). \end{aligned} \quad (40)$$

As consequence of the Cauchy integral formula

$$\|M_i\|_2 \frac{\|\hat{X}_{i-1}\|_2}{i} \leq \|M_i\|_2 \frac{\|A_i\|_2}{i} \leq C(\Psi_k) \sqrt{k} \frac{\|M_i\|_2}{i!} \leq C(\Psi_k) \sqrt{k} \frac{\max_{|\lambda|=R} \|M(\lambda)\|_2}{R^i}. \quad (41)$$

By substituting (41) in (40) we obtain

$$\begin{aligned} \|Y_{+,0}\|_2 &\leq \|M_0^{-1}\|_2 \left(\sigma_{\tilde{r}+1} (\gamma + \log(s+1)) \max_{1 \leq i \leq s} \|M_i\|_2 \right. \\ &\quad \left. + \max_{|\lambda|=R} \|M(\lambda)\|_2 C(\Psi_k) \sqrt{k} \frac{d-s}{R^s} \right) \\ &\leq \sigma_{\tilde{r}+1} \|M_0^{-1}\|_2 \left((\gamma + \log(s+1)) \max_{1 \leq i \leq s} \|M_i\|_2 + \max_{|\lambda|=R} \|M(\lambda)\|_2 \right). \end{aligned} \quad (42)$$

We reach the conclusion (37b) from the combination of (42) in (39).

5.2 Approximation by reducing the degree

Another approximation which reduces the complexity can be done by truncating the polynomial in Ψ_k . The following theorem illustrates the approximation properties of this approach.

Theorem 10 Let $a \in \mathbb{C}^{(d+1) \times (k+1) \times r}$, $Z \in \mathbb{C}^{n \times r}$ be the coefficients and the matrix that represent the tensor structured function Ψ_{k+1} and let $\underline{H}_k \in \mathbb{C}^{(k+1) \times k}$ be such that $(\Psi_{k+1}, \underline{H}_k)$ is a TIAR factorization. For $\tilde{d} \leq d$ let

$$\tilde{\Psi}_{k+1}(\theta) := P_{\tilde{d}}(\theta) \left(\sum_{\ell=1}^r \tilde{a}_{:, :, \ell} \otimes z_{\ell} \right) \quad (43)$$

where $\tilde{a}_{i,j,\ell} = a_{i,j,\ell}$ for $i = 1, \dots, \tilde{d}$, $j = 1, \dots, k+1$ and $\ell = 1, \dots, r$. Then

$$\|\tilde{\Psi}_{k+1} - \Psi_{k+1}\| \leq C(\Psi_{k+1}) \frac{(d - \tilde{d})}{\tilde{d}!} \quad (44)$$

$$\|\mathcal{B}\tilde{\Psi}_k - \tilde{\Psi}_{k+1}\underline{H}_k\| \leq C(\Psi_{k+1}) \left(\max_{\tilde{d}+1 \leq i \leq d} \|M_i\|_F \right) \|M_0^{-1}\|_F \frac{d - \tilde{d}}{(\tilde{d} + 1)!}. \quad (45)$$

Proof We define $X_i := ZA_{i+1}$ for $i = 0, \dots, d$ and $X := [X_0^T, \dots, X_d^T]$ and $\tilde{X} := [X_0^T, \dots, X_{\tilde{d}}^T]$ such that $\Psi_{k+1}(\theta) = P_d(\theta)X$ and $\tilde{\Psi}_{k+1}(\theta) = P_{\tilde{d}}(\theta)\tilde{X}$. We have

$$\|\Psi_{k+1}(\theta) - \tilde{\Psi}_{k+1}(\theta)\|^2 = \sum_{i=\tilde{d}+1}^d \|X_i\|_F^2 = \sum_{i=\tilde{d}+1}^d \|A_i\|_F^2.$$

By using the definition of $C(\Psi_{k+1})$ we obtain (44).

By definition $\Psi_k(\theta) = \Psi_{k+1}(\theta)I_{k+1,k}$ and $\tilde{\Psi}_k(\theta) = \tilde{\Psi}_{k+1}(\theta)I_{k+1,k}$, using the Observation 5, if we define $Y_i := X_i I_{k+1,k}$ for $i = 0, \dots, d-1$ and $Y := [Y_0^H \dots Y_{d-1}^H]^H$ and $\tilde{Y} := [Y_0^H \dots Y_{\tilde{d}-1}^H]^H$ we can express $\Psi_k(\theta) = P_{d-1}(\theta)Y$ and $\tilde{\Psi}_k(\theta) = P_{\tilde{d}-1}(\theta)\tilde{Y}$.

Using [12, theorem 4.2] for each column of $\Psi_k(\theta)$ and $\tilde{\Psi}_k(\theta)$, we get $\mathcal{B}\Psi_k(\theta) = P_d(\theta)Y_+$ and $\mathcal{B}\tilde{\Psi}_k(\theta) = P_{\tilde{d}}(\theta)\tilde{Y}_+$ with

$$\begin{aligned} Y_{+,i+1} &:= \frac{Y_i}{i+1} & \text{for } i=0, \dots, d-1 & \quad \text{and} \quad Y_{+,0} := -M_0^{-1} \sum_{i=1}^d M_i Y_{+,i} \\ \tilde{Y}_{+,i+1} &:= Y_{+,i+1} & \text{for } i=0, \dots, \tilde{d}-1 & \quad \text{and} \quad \tilde{Y}_{+,0} := -M_0^{-1} \sum_{i=1}^{\tilde{d}} M_i Y_{+,i} \end{aligned}$$

In our notation, the fact that $(\Psi_{k+1}, \underline{H}_k)$ is a TIAR factorization, can be expressed as $P_d(\theta)Y_+ = P_d(\theta)X\underline{H}_k$, which implies that the monomial coefficients are equal, i.e.,

$$Y_{+,i} = X_i \underline{H}_k \quad \text{for } i = 0, \dots, d. \quad (46)$$

Hence, from (6) we have

$$\begin{aligned}
 \|\mathcal{B}\tilde{\Psi}_k - \tilde{\Psi}_{k+1}\underline{H}_k\|^2 &= \|P_d(\theta)\tilde{Y}_+ - P_d(\theta)\tilde{X}\underline{H}_k\|^2 \\
 &= \|\tilde{Y}_+ - \tilde{X}\underline{H}_k\|_F^2 \\
 &= \|\tilde{Y}_{+,0} - X_0\underline{H}_k\|_F^2 + \sum_{i=1}^{\tilde{d}} \|Y_{+,i} - X_i\underline{H}_k\|_F^2 \\
 &= \|\tilde{Y}_{+,0} - X_0\underline{H}_k\|_F^2.
 \end{aligned}$$

In the last step we applied (46). Moreover, by again using (46), we have

$$\begin{aligned}
 Y_{+,0} - X_0\underline{H}_k &= -M_0^{-1} \sum_{i=1}^d M_i Y_{+,i} - X_0\underline{H}_k \\
 &= -M_0^{-1} \sum_{i=1}^{\tilde{d}} M_i \tilde{Y}_{+,i} - M_0^{-1} \sum_{i=\tilde{d}+1}^d M_i Y_{+,i} - X_0\underline{H}_k \\
 &= \tilde{Y}_{+,0} - X_0\underline{H}_k - M_0^{-1} \sum_{i=\tilde{d}+1}^d M_i \frac{X_{i-1}I_{k+1,k}}{i}.
 \end{aligned}$$

Therefore $\|\tilde{Y}_{+,0} - X_0\underline{H}_k\|_F \leq \|M_0^{-1}\|_F \sum_{i=\tilde{d}+1}^d \frac{\|M_i\|_F \|A_i\|_F}{i}$. We obtain (45) by using the properties of the Frobenius norm and the definition of $C(\Psi_{k+1})$.

Remark 11 The approximation given in Theorem 10 can only be effective under the condition that $(\max_{\tilde{d}+1 \leq i \leq d} \|M_i\|_F) / (\tilde{d} + 1)!$ is small. In particular this condition is satisfied if the Taylor coefficients $M_i / i!$ present a fast decay. This condition corresponds to having the coefficients of the power series expansion of $M(\lambda)$ that are decaying to zero.

The final approximation procedure is summarized in Algorithm 4. In particular Step 1–2 correspond to the approximation by SVD compression, described in Sect. 5.1, whereas Step 3–4 correspond to the approximation by reducing the degree, described in Sect. 5.2.

5.3 The fast decay of singular values

Finally, as a further justification for our approximation procedure, we now show how fast the singular values decay. The fast decay in the singular values illustrated below justifies the effectiveness of the truncation in Sect. 5.1.

Lemma 12 Let $a \in \mathbb{C}^{(d+1) \times (k+1) \times r}$, $Z \in \mathbb{C}^{n \times r}$ be the coefficients and the matrix that represent the tensor structured function Ψ_{k+1} and let $\underline{H}_k \in \mathbb{C}^{(k+1) \times k}$ be such that $(\Psi_{k+1}, \underline{H}_k)$ is a TIAR factorization. Then, the tensor a is generated by $d + 1$ vectors,

Algorithm 4: Approximation of TIAR factorization

input : An approximated TIAR factorization $(\Psi_{k+1}, \underline{H}_k)$ where the tensor structured function is represented by the matrices $(Z, 0, 0, I) \in \mathbb{C}^{n \times r} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{p \times p}$ and $(a, 0, 0) \in \mathbb{C}^{(d+1) \times (k+1) \times r} \times \mathbb{C}^{(d+1) \times (k+1) \times p} \times \mathbb{C}^p$ and the tolerance ε .

output: An approximated TIAR factorization $(\Psi_{k+1}, \underline{H}_k)$ where the tensor structured function is represented by the matrices $(\tilde{Z}, 0, 0, I) \in \mathbb{C}^{n \times \tilde{r}} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{n \times p} \times \mathbb{C}^{p \times p}$ and $(\tilde{a}, 0, 0) \in \mathbb{C}^{(\tilde{d}+1) \times (k+1) \times \tilde{r}} \times \mathbb{C}^{(\tilde{d}+1) \times (k+1) \times p} \times \mathbb{C}^p$.

- 1 Compute the SVD decomposition given in (35) partitioned such that $\sigma_{\tilde{r}+1} \leq \varepsilon$
- 2 Compute \tilde{Z} and \tilde{A}_i according to (36) and set $a_{i,:} = \tilde{A}_i^T$ for $i = 1, \dots, d+1$
- 3 Compute \tilde{d} such that

$$\left(\max_{\tilde{d}+1 \leq i \leq d} \|M_i\|_F \right) \|M_0^{-1}\|_F \frac{d - \tilde{d}}{(\tilde{d} + 1)!} < \varepsilon$$

- 4 Reduce the size of the tensor $\tilde{a}_{i,:} = a_{i,1:\tilde{d},:}$.

in the sense that each vector $a_{i,j,:}$ for $i = 1, \dots, d+1$ and $j = 1, \dots, k+1$ can be expressed as linear combination of the vectors $a_{i,1,:}$ and $a_{1,j,:}$ for $i = 1, \dots, d-k$ and $j = 1, \dots, k+1$.

Proof The proof is based on induction over the length k of the TIAR factorization. The result is trivial if $k = 1$. Suppose the result holds for some k . Let $Z \in \mathbb{C}^{n \times (r-1)}$, $a \in \mathbb{C}^{d \times k \times (r-1)}$ represent the tensor structured function Ψ_k and let $\underline{H}_{k-1} \in \mathbb{C}^{k \times (k-1)}$ be an upper Hessenberg matrix such that $(\Psi_k, \underline{H}_{k-1})$ is a TIAR factorization. If we expand the TIAR factorization $(\Psi_k, \underline{H}_{k-1})$ by using the Algorithm 1, more precisely by using (13b) and (18b), we obtain $\beta a_{i+1,k+1,:} = a_{i,k,:}/i - \sum_{j=1}^k h_j a_{i,j,:}$ for $i = 1, \dots, d$. We reach the conclusion by induction.

Theorem 13 Under the same hypothesis of Lemma 12, let A be the unfolding of the tensor a in the sense that $A = [A_1, \dots, A_{d+1}]$ such that $A_i := (a_{i,:})^T$. We have the following decay in the singular values

$$\sigma_i \leq C(\Psi_{k+1}) \frac{d - (R - k - 1)}{(R - k)!} \quad i = R + 1, \dots, d + 1,$$

where $k \leq R \leq d + k + 1$.

Proof We define the matrix $\tilde{A} := [A_1, \dots, A_{R-k}, 0, \dots, 0] \in \mathbb{C}^{r \times (k+1)(d+1)}$. Notice that the columns of the matrices A and \tilde{A} correspond to the vectors $a_{i,j,:}^T$. In particular, using the Lemma 12, we have that $\text{rank}(A_1) = k+1$ whereas $\text{rank}(A_j) = 1$ if $j \leq d-k$ otherwise $\text{rank}(A_j) = 0$. Then we have that $\text{rank}(A) = d+1$ and $\text{rank}(\tilde{A}) = R$. Using Weyl's theorem [10, Corollary 8.6.2] and Theorem 8 we have for $i \geq R+1$

$$\sigma_i \leq \|A - \tilde{A}\|_F \leq \sum_{i=R-k+1}^{d+1} \|A_i\|_F \leq \sum_{i=R-k+1}^{d+1} \frac{C(\Psi_{k+1})}{(i-1)!} \leq C(\Psi_{k+1}) \frac{d - (R - k - 1)}{(R - k)!}.$$

□

6 Complexity analysis

We presented above two different restarting strategies: the structured semi-explicit restart and the implicit restart. They have different performance for different problems and we have not been able to conclusively determine if one is better than the other. The best choice of the restarting strategy appears to depend on many problem properties. It may be convenient to test both methods on the same problem. We now discuss the general performances, in terms of complexity and stability. The complexity discussion is based on the assumption that the complexity of the action of M_0^{-1} is neglectable in comparison to the other parts of the algorithm.

6.1 Complexity of expanding the TIAR factorization

The main computational effort of the Algorithm 2 and Algorithm 3 is the expansion of a TIAR factorization described in Algorithm 1, independent of which restarting strategy is used. The essential computational effort of Algorithm 1 is the computation of \tilde{z} , given in Eq. (9). This operation has complexity $\mathcal{O}(drn)$ for each iteration. In the implicit restart, Algorithm 3, r and d are not large in general, due to the way they are automatically selected in the approximation procedure in Algorithm 4. In the semi-explicit restart, Algorithm 2, we have instead $r, d \leq m$.

6.2 Complexity of the restarting strategies

After an implicit restart we obtain a TIAR factorization of length p , whereas after a semi-explicit restart, we obtain a TIAR factorization of length p_ℓ . This means that the semi-explicit restart requires a re-computation phase, i.e., after the restart we need to perform extra $p - p_\ell$ steps in order to have a TIAR factorization of length p . If $p - p_\ell$ is large, i.e., if not many Ritz values converged in comparison to the restarting parameter p , then the re-computation phase is the essential computational effort of the algorithm. Notice that this is hard to predict since we do not know how fast the Ritz values will converge in advance.

6.3 Stability of the restarting strategies

We will illustrate in Sect. 7 that the restarting approaches have different stability properties. The semi-explicit restart tends to be efficient if only a few eigenvalues are wanted, i.e., if p_{\max} is small. This is due to the fact that we impose the structure in the starting function. On the other hand, the implicit restart requires a thick restart in order to be stable in several situations, see corresponding discussions for the linear case in [19, chapter 8]. Then p has to be large enough in a sense that at each restart the p wanted Ritz values have the corresponding residual not small. This leads to additional computational and memory resources.

If we use the semi-explicit restart, then the computation of \tilde{z} , in Eq. (9), involves the term $\mathbb{M}_d(Y, S)$. This quantity can be computed in different ways. In the simulations we

must choose between (8) or [12, Equation (4.8)]. The choice influences the stability of the algorithm. In particular if one eigenvalue of S is close to $\partial\Omega$ and $M(\lambda)$ is not analytic on $\partial\Omega$, the series (8) may converge slowly and in practice overflow can occur. In such situations, [12, Equation (4.8)] is preferable. Notice that it is not always possible to use [12, Equation (4.8)] since many problems cannot be formulated as a short sum of product of matrices and functions.

6.4 Memory requirements of the restarting strategies

From a memory point of view, the essential part of the semi-explicit restart is the storage of the matrices Z and Y , that is $\mathcal{O}(nm + np)$. In the implicit restart the essential part is the storage of the matrix Z and requires $\mathcal{O}(nr_{\max})$ where r_{\max} denotes the maximum value that the variable r takes in the Algorithm 1. The size of r_{\max} is not predictable since it depends on the SVD-approximation introduced in Algorithm 4. Since in each iteration of the Algorithm 1 the variable r is increased, it holds $r_{\max} \geq m - p$. Therefore, in the optimal case where r_{\max} takes the lower value, the two methods are comparable in terms of memory requirements. Notice that, the semi-explicit restart in general requires less memory and has the advantage that the required memory is problem independent.

7 Numerical experiments

7.1 Delay eigenvalue problem

In order to illustrate properties of the proposed restart methods and advantages in comparison to other approaches, we carried out numerical simulations¹ for solving a delay eigenvalue problem (DEP). More precisely, we consider the DEP obtained by discretizing the characteristic equation of the delay differential equation defined in [15, sect 4.2, Eq (22a)] with $\tau = 1$. By using a standard second order finite difference discretization, the DEP is formulated as $M(\lambda) = -\lambda^2 I + \lambda T_1 + T_0 + e^{-\lambda} T_2$. We show how the proposed methods perform in terms of m , the maximum length of the TIAR factorization, and the restarting parameter p .

Table 1a, b show the advantages of our semi-explicit restart approach in comparison to the equivalent method described in [12]. Our new approach is faster in terms of CPU-time and can solve larger problems due to the memory efficient representation of the Krylov basis.

Table 2a, b show the effectiveness of approximations introduced in Sects. 5.1 and 5.2 in comparison to the corresponding restart procedure without approximations. In particular, in Algorithm 4 we consider a drop tolerance $\varepsilon = 10^{-14}$. Since the DEP is defined by entire functions, the power series coefficients decay to zero and, according to Remark 11, the approximation by reducing the degree is expected to be effective.

¹ All simulations were carried out with Intel octa core i7-3770 CPU 3.40GHz and 16 GB RAM using MATLAB.

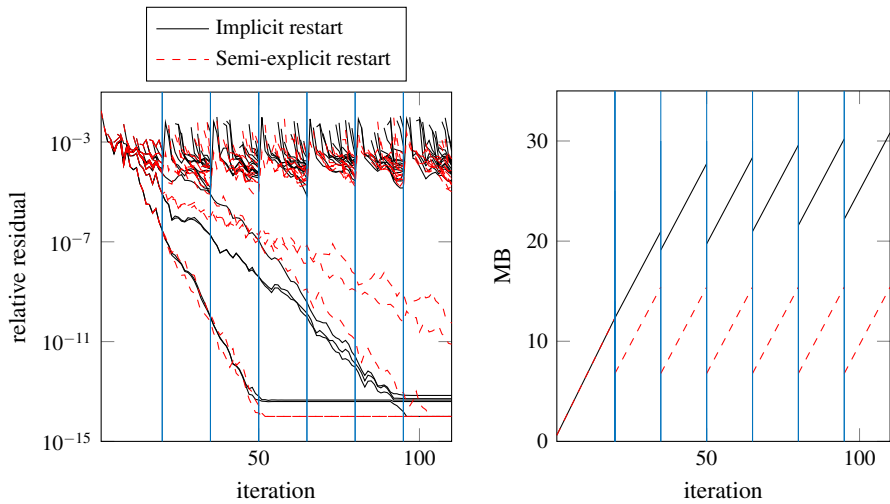


Fig. 2 Implicit and semi-explicit restart for DEP of size $n = 40401$ with $m = 20$, $p = 5$ and resting 7 times (restart=7)

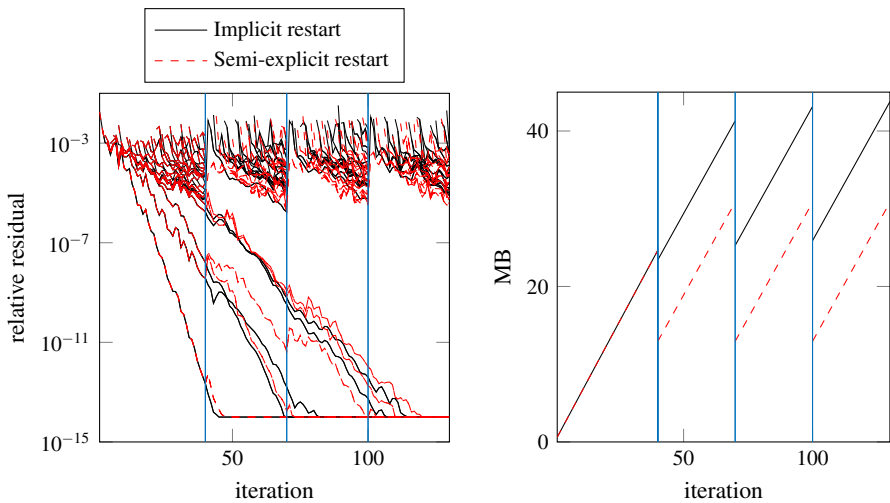


Fig. 3 Implicit and semi-explicit restart for DEP of size $n = 40401$ with $m = 40$, $p = 10$ and resting 4 times (restart=4)

By approximating the TIAR factorization, the implicit restart requires less resources in terms of memory and CPU-time and can solve larger problems.

We now illustrate the differences between the semi-explicit and the implicit restart. More precisely, we show how the parameters m and p influence the convergence of the Ritz values with respect to the number of iterations. The accuracy of an eigenpair (λ, v) of the nonlinear eigenvalue problem is measured with the relative residual

$$\frac{\|M(\lambda)v\|_2}{\|v\|_2 \sum_{i=1}^q \|T_i\|_\infty |f_i(\lambda)|},$$

Table 1 DEP, Semi-explicit restart

Size	Tensor struct. Functions		Original approach [12]	
	CPU	Memory	CPU	Memory
(a) $m = 20$, $p = 5$, restart = 7				
10,201	19.07 s	3.7 MB	31.41 s	65.38 MB
40,401	30.14 s	14.8 MB	1 m 30 s	258.92 MB
160,801	1 m 47 s	58.9 MB	6 m 04	1.01 GB
641,601	7 m 30 s	235.0 MB	24 m 27 s	4.02 GB
1,002,001	12 m 01 s	366.9 MB	–	–
(b) $m = 40$, $p = 10$, restart = 4				
10,201	13.47 s	7.62 MB	1 m 05 s	255.27 MB
40,401	41.81 s	30.20 MB	4 m	1 GB
160,801	144.79 s	120.23 MB	15 m 54 s	3.93 GB
641,601	10 m 43 s	479.71 MB	–	–
1,002,001	16 m 21 s	749.18 MB	–	–

where $M(\lambda) = \sum_{i=1}^q T_i f_i(\lambda)$. The convergence of the semi-explicit restart appears to be slower when p is not sufficiently large. See Fig. 2. The convergence speed of both restarting strategies is comparable for a larger m and p . See Fig. 3.

In practice, the performance of the two restarting strategies corresponds to a trade-off between CPU-time and memory. In particular, due to the fact that we impose the structure, the semi-explicit restart does not have a growth in the polynomial part at each restart and therefore requires less memory. On the other hand, for this problem, the semi-explicit restart appears to be slower in terms of CPU-time. See Tables 1 and 2. For completeness we provide the total number of accumulated inner iterations: for Table 1a 134, for Table 1b 154, for Table 2a 110 and for Table 2b 130. An illustration of the CPU-time for a fixed accuracy is given in Table 3.

7.2 Waveguide eigenvalue problem

In order to illustrate how the performance depends on the problem properties, we now consider a NEP defined by functions with branch point and branch cut singularities. More precisely, we consider the waveguide eigenvalue problem (WEP) described in [13, Section 5.1] after the Cayley transformation with pole removal, i.e.,

$$M(\lambda) = \begin{pmatrix} F_A(\lambda) & F_{C_1}(\lambda) \\ (1 - \lambda)C_2^T & \tilde{P}(\lambda) \end{pmatrix}.$$

The matrix C_2^T and the second degree polynomials $F_A(\lambda)$ and $F_{C_1}(\lambda)$ are sparse. The matrix $\tilde{P}(\lambda)$ is defined by nonlinear functions of λ involving square roots of polynomials and it is dense. However, in our framework, an explicit storage of this matrix is not necessary since the matrix–vector product $\tilde{P}(\lambda)w$ is efficiently computed

Table 2 DEP, Implicit restart

Size	Approximation		No approximation	
	CPU	Memory	CPU	Memory
(a) $m = 20$, $p = 5$, restart=7				
10,201	6.82 s	7.8 MB	11.95 s	17.1 MB
40,401	21.96 s	30.8 MB	37.63 s	67.8 MB
160,801	1 m20 s	120.2 MB	2 m21 s	269.9 MB
641,601	5 m24 s	469.9 MB	9 m33 s	1.1 GB
1,002,001	8 m36 s	733.9 MB	15 m16 s	1.6 GB
(b) $m = 40$, $p = 10$, restart=4				
10,201	9.54 s	11.1 MB	16.61 s	20.2 MB
40,401	30.48 s	43.8 MB	50.66 s	80.1 MB
160,801	1 m54 s	174.2 MB	3 m11 s	319.0 MB
641,601	8 m05 s	695.1 MB	13 m14 s	1.2 GB
1,002,001	12 m17 s	1.1 GB	20 m57 s	1.9 GB

Table 3 DEP, Implicit and semi-explicit restart, stopped when residual of $p_{\max} = 10$ Ritz values is less than 10^{-10}

Size	Implicit restart			Semi-explicit restart		
	Nr. restarts	CPU	Memory	Nr. restarts	CPU	Memory
40,401	6	13.8 s	29.59 MB	7	28.97 s	14.79 MB
160,801	4	37.02 s	115.32 MB	6	1 m 29 s	58.89 MB
641,601	4	2 m 31 s	450.34 MB	6	6 m 23 s	234.96 MB
1,002,001	4	3 m 58 s	703.30 MB	5	8 m 29 s	366.94 MB
2,563,201	–	–	–	5	21 m10 s	938.67 MB

using two Fast Fourier Transforms (FFTs) and a multiplication with a diagonal matrix. See [13] for a full description of the problem. In this NEP, Ω is the unit disc and there are branch point singularities in $\partial\Omega$. Thus, due to the slow convergence of the power series of $M(\lambda)$, in the semi-explicit restart we have to use [12, Equation (4.8)] in order to compute $\mathbb{M}_d(Y, S)$. This also implies that the approximation by reducing the degree is not expected to be effective since the power series coefficients of $M(\lambda)$ are not decaying to zero.

In analogy to the previous subsection, we carried out numerical simulations in order to compare the semi-explicit and the implicit restart. With Figs. 4 and 5, we illustrate the performance of the two restarting approaches with respect to the choice of the parameters m and p . When p is sufficiently large, the residual in the semi-explicit restart appears to stagnate after the first restart whereas it decreases in a regular way in the implicit restart. See Fig. 4. This is due to the fact that semi-explicit restart imposes the structure on p vectors which is not beneficial when they do not contain eigenvector approximations. On the other hand, when p is small, the behavior of the residual appear

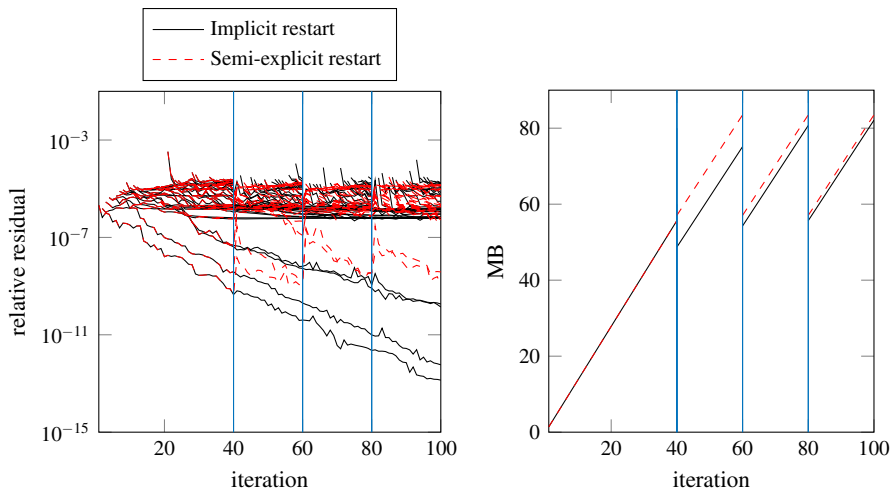


Fig. 4 Implicit and semi-explicit restart for WEP of size $n = 40803$ with $m = 40$, $p = 20$ and $\text{restart} = 4$

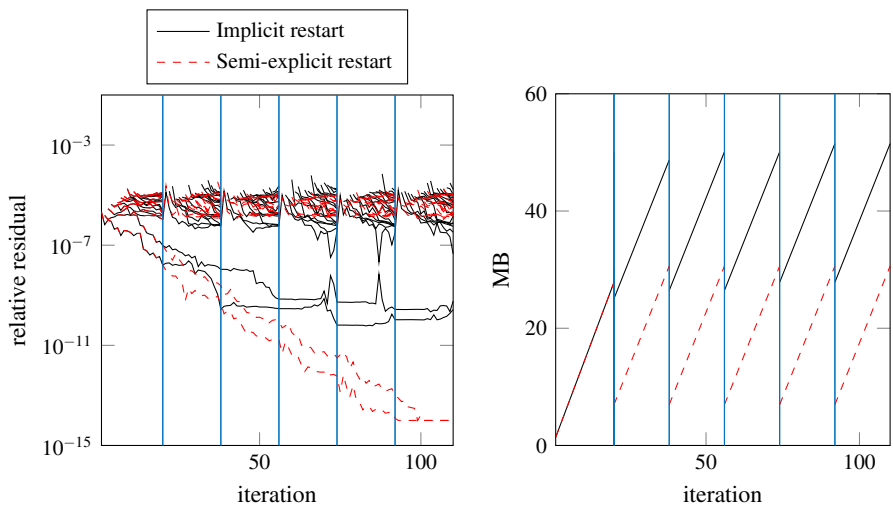


Fig. 5 Implicit and semi-explicit restart for WEP of size $n = 91,203$ with $m = 20$, $p = 4$ and $\text{restart} = 6$

to be specular. See Fig. 5. This is a consequence of the fact that, already after the first restart, the Krylov subspace is almost an invariant subspace (since p Ritz pairs are quite accurate). This is consistent with the linear case where implicit restarting with a Krylov subspace which is almost an invariant subspace is known to suffer from numerical instabilities. It is known that this specific problem has two eigenvalues. Therefore, in order to reduce the CPU-time and the memory resources, the restarting parameter p should be selected small. As consequence of the above discussion, we conclude that the semi-explicit restart is the best restarting strategy for this problem.

8 Concluding remarks and outlook

In this work we have derived an extension of the TIAR algorithm and two restarting strategies. Both restarting strategies are based on approximating the TIAR factorization. In other works on the IAR-method it has been proven that the basis matrix contains a structure that allows exploitations, e.g. for NEPs with low rank structure in the coefficients [34]. An investigation about the combination of the approximations of the TIAR factorization with such structures of the NEP seems possible but deserve further attention.

Although the framework of TIAR and restarted TIAR is general, a specialization of the methods to the NEP is required in order to efficiently solve the problem. More precisely, an efficient computation procedure for computing (9) is required. This is a nontrivial task for many application and requires problem specific research.

Acknowledgements We gratefully acknowledge the support of the Swedish Research Council under Grant No. 621-2013-4640.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Abramowitz, M., Stegun, I.: Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables, vol. 55. Courier Corporation, North Chelmsford (1964)
2. Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H.: Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, vol. 11. Siam, New Delhi (2000)
3. Bai, Z., Su, Y.: SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J. Matrix Anal. Appl.* **26**(3), 640–659 (2005)
4. Beckermann, B.: The condition number of real Vandermonde, Krylov and positive definite Hankel matrices. *Numer. Math.* **85**(4), 553–577 (2000)
5. Betcke, M.M., Voss, H.: Restarting projection methods for rational eigenproblems arising in fluid-solid vibrations. *Math. Model. Anal.* **13**(2), 171–182 (2008)
6. Betcke, M.M., Voss, H.: Restarting iterative projection methods for Hermitian nonlinear eigenvalue problems with minmax property. *Numer. Math.* **135**(2), 397–430 (2017)
7. Betcke, T., Higham, N.J., Mehrmann, V., Schröder, C., Tisseur, F.: NLEVP: a collection of nonlinear eigenvalue problems. In: Technical Report, Manchester Institute for Mathematical Sciences (2011)
8. Betcke, T., Voss, H.: A Jacobi–Davidson-type projection method for nonlinear eigenvalue problems. *Future Gener. Comput. Syst.* **20**(3), 363–372 (2004)
9. Effenberger, C.: Robust solution methods for nonlinear eigenvalue problems. Ph.D. thesis, École polytechnique fédérale de Lausanne (2013)
10. Golub, G.H., Van Loan, C., Charles, F.: Matrix Computations, vol. 3. JHU Press, Baltimore (2012)
11. Güttel, S., Van Beeumen, R., Meerbergen, K., Michiels, W.: NLEIGS: a class of fully rational Krylov methods for nonlinear eigenvalue problems. *SIAM J. Sci. Comput.* **36**(6), A2842–A2864 (2014)
12. Jarlebring, E., Meerbergen, K., Michiels, W.: Computing a partial Schur factorization of nonlinear eigenvalue problems using the infinite Arnoldi method. *SIAM J. Matrix Anal. Appl.* **35**(2), 411–436 (2014)
13. Jarlebring, E., Mele, G., Runborg, O.: The waveguide eigenvalue problem and the tensor infinite Arnoldi method. *SIAM J. Sci. Comput.* **39**(3), A1062–A1088 (2017)
14. Jarlebring, E., Michiels, W., Meerbergen, K.: A linear eigenvalue algorithm for the nonlinear eigenvalue problem. *Numer. Math.* **122**(1), 169–195 (2012)

15. Jarlebring, E., Poloni, F.: Iterative methods for the delay Lyapunov equation with T-Sylvester preconditioning. In: Technical Report (2015). [ArXiv:1507.02100](https://arxiv.org/abs/1507.02100)
16. Kressner, D.: A block Newton method for nonlinear eigenvalue problems. *Numer. Math.* **114**(2), 355–372 (2009)
17. Kressner, D., Roman, J.E.: Memory-efficient Arnoldi algorithms for linearizations of matrix polynomials in Chebyshev basis. *Numer. Linear Algebr. Appl.* **21**(4), 569–588 (2014)
18. Lancaster, P., Psarrakos, P.: On the pseudospectra of matrix polynomials. *SIAM J. Matrix Anal. Appl.* **27**(1), 115–129 (2005)
19. Lehoucq, R.B.: Analysis and implementation of an implicitly restarted Arnoldi iteration. Ph.D. thesis, Rice University (1995)
20. Lehoucq, R.B., Sorensen, D.C.: Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Anal. Appl.* **17**(4), 789–821 (1996)
21. Lu, D., Su, Y., Bai, Z.: Stability analysis of the two-level orthogonal Arnoldi procedure. *SIAM J. Matrix Anal. Appl.* **37**(1), 195–214 (2016)
22. Mackey, D.S., Mackey, N., Mehl, C., Mehrmann, V.: Structured polynomial eigenvalue problems: good vibrations from good linearizations. *SIAM J. Matrix Anal. Appl.* **28**(4), 1029–1051 (2006)
23. Mackey, D.S., Mackey, N., Tisseur, F.: Polynomial eigenvalue problems: Theory, computation, and structure. In: *Numerical Algebra, Matrix Theory, Differential-Algebraic Equations and Control Theory*, pp. 319–348. Springer (2015)
24. Meerbergen, K.: Locking and restarting quadratic eigenvalue solvers. *SIAM J. Sci. Comput.* **22**(5), 1814–1839 (2001)
25. Meerbergen, K.: The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM J. Matrix Anal. Appl.* **30**(4), 1463–1482 (2008)
26. Mehrmann, V., Voss, H.: Nonlinear eigenvalue problems: a challenge for modern eigenvalue methods. *GAMM Mitt.* **27**(2), 121–152 (2004)
27. Morgan, R.: On restarting the Arnoldi method for large nonsymmetric eigenvalue problems. *Math. Comput.* **65**(215), 1213–1230 (1996)
28. Neumaier, A.: Residual inverse iteration for the nonlinear eigenvalue problem. *SIAM J. Numer. Anal.* **22**(5), 914–923 (1985)
29. Stewart, G.W.: A Krylov–Schur algorithm for large eigenproblems. *SIAM J. Matrix Anal. Appl.* **23**(3), 601–614 (2002)
30. Su, Y., Bai, Z.: Solving rational eigenvalue problems via linearization. *SIAM J. Matrix Anal. Appl.* **32**(1), 201–216 (2011)
31. Szyld, D., Vecharynski, E., Xue, F.: Preconditioned eigensolvers for large-scale nonlinear Hermitian eigenproblems with variational characterizations. II. Interior eigenvalues. *SIAM J. Sci. Comput.* **37**(6), A2969–A2997 (2015)
32. Szyld, D., Xue, F.: Preconditioned eigensolvers for large-scale nonlinear Hermitian eigenproblems with variational characterizations. I. Extreme eigenvalues. *Math. Comput.* **85**(302), 2887–2918 (2016)
33. Tisseur, F., Meerbergen, K.: The quadratic eigenvalue problem. *SIAM Rev.* **2**, 235–286 (2001)
34. Van Beeumen, R., Jarlebring, E., Michiels, W.: A rank-exploiting infinite Arnoldi algorithm for nonlinear eigenvalue problems. *Numer. Linear Algebra Appl.* **23**(4), 607–628 (2016)
35. Van Beeumen, R., Meerbergen, K., Michiels, W.: Compact rational Krylov methods for nonlinear eigenvalue problems. *SIAM J. Matrix Anal. Appl.* **36**(2), 820–838 (2015)
36. Voss, H.: A maxmin principle for nonlinear eigenvalue problems with application to a rational spectral problem in fluid-solid vibration. *Appl. Math.* **48**(6), 607–622 (2003)
37. Voss, H.: An Arnoldi method for nonlinear eigenvalue problems. *BIT* **44**(2), 387–401 (2004)
38. Voss, H.: Nonlinear eigenvalue problems. In: L. Hogben (ed.) *Handbook of Linear Algebra*, Second Edition, no. 164 in *Discrete Mathematics and Its Applications*. Chapman and Hall/CRC (2013)
39. Zhang, Y., Su, Y.: A memory-efficient model order reduction for time-delay systems. *BIT* **53**(4), 1047–1073 (2013)