Numer. Math.
DOI 10.1007/s00211-016-0804-3

Numerische Mathematik

# Restarting iterative projection methods for Hermitian nonlinear eigenvalue problems with minmax property

**Marta M. Betcke**[1] · **Heinrich Voss**[2]

Received: 27 October 2012 / Revised: 3 November 2015

**Abstract** In this work we present a new restart technique for iterative projection methods for nonlinear eigenvalue problems admitting minmax characterization of their eigenvalues. Our technique makes use of the minmax induced *local enumeration* of the eigenvalues in the inner iteration. In contrast to global numbering which requires including all the previously computed eigenvectors in the search subspace, the proposed local numbering only requires a presence of one eigenvector in the search subspace. This effectively eliminates the search subspace growth and therewith the super-linear increase of the computational costs if a large number of eigenvalues or eigenvalues in the interior of the spectrum are to be computed. The new restart technique is integrated into nonlinear iterative projection methods like the Nonlinear Arnoldi and Jacobi-Davidson methods. The efficiency of our new restart framework is demonstrated on a range of nonlinear eigenvalue problems: quadratic, rational and exponential including an industrial real-life conservative gyroscopic eigenvalue problem modeling free vibrations of a rolling tire. We also present an extension of the method to problems without minmax property but with eigenvalues which have a dominant either real or imaginary part and test it on two quadratic eigenvalue problems.

**Keywords** Nonlinear eigenvalue problem · Iterative projection method · Nonlinear Arnoldi method · Jacobi-Davidson method · Minmax characterization · Restart · Purge and lock

✉ Marta M. Betcke
m.betcke@ucl.ac.uk

[1] Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK

[2] Institute of Mathematics, Hamburg University of Technology, 21071 Hamburg, Germany

**Mathematics Subject Classification** 65F15 · 15A18 · 35P30 · 49R50 · 65N25

## 1 Introduction

In this work we consider a problem of computing a large number of eigenvalues in an open real interval $J \subset \mathbb{R}$ and the corresponding eigenvectors of the nonlinear eigenvalue problem (NEP)

$$T(\lambda)x = 0, \tag{1}$$

where $T(\lambda) \in \mathbb{C}^{n\times n}$ is a family of large and sparse Hermitian matrices for every $\lambda \in J$. We furthermore assume that the eigenvalues of (1) in $J$ can be characterized as minmax values of a Rayleigh functional [35]. Such problems routinely arise in simulation of acoustic properties of e.g.vehicles or their parts in order to minimize the noise exposure to the passengers as well as to the environment.

The problem of computing a moderate number of eigenpairs of a nonlinear eigenvalue problem at the beginning of the spectrum has been extensively studied. For minmax admitting problems a Nonlinear Arnoldi method was suggested in e.g. [29] and the Jacobi-Davidson method in [4]. For more general nonlinear eigenproblems iterative projection methods were considered in [1,7,11,14–17,19–21,25,31,32,34]. However, the approach in [4,29] hits its limitations if a large number of eigenvalues (in particular in the interior of the spectrum) of (1) is needed. To algorithmically exploit the minmax property, one has to project the problem under consideration onto a sequence of search spaces, which dimension is growing with the number of the targeted eigenvalue. For a large number of eigenvalues (or eigenvalues in the interior of the spectrum) this naturally requires an excessive amount of storage and computing time.

In this work we propose a new restart technique which allows to project the NEP (1) only onto search spaces of a fixed, relatively small dimension throughout the iteration. The new restart technique can be integrated with iterative projection methods such as the Nonlinear Arnoldi or Jacobi-Davidson method making them capable of computation of a large number of eigenpairs, possibly in the interior of the spectrum. A preliminary version of the local restart technique was published in [18].

The paper is organized as follows. In Sects. 2 and 3 we recapitulate the variational characterization for nonoverdamped nonlinear eigenproblems and the iterative projection methods for their solution. The new restart technique is presented in Sect. 4 along with a strategy for dealing with spurious eigensolutions which are an intrinsic part of the interior eigenvalue computation. The resulting framework for restarting of nonlinear iterative projection methods for interior eigenvalue computation is summarized in Sect. 5. The performance of the restarted methods is demonstrated in Sect. 6 on a range of nonlinear eigenvalue problems with and without minmax property including a real-life industrial gyroscopic eigenvalue problem arising in modeling of the noise radiation from rotating tires. Section 7 concludes the paper with a summary and outlook for future research.

## 2 Variational characterization of eigenvalues

**Definition 1** (*Hermitian Nonlinear Eigenvalue Problem*) Let $T(\lambda) \in \mathbb{C}^{n\times n}$ be a family of Hermitian matrices for every $\lambda$ in an open real interval $J$. As in the linear case, $T(\lambda) = \lambda I - A$, we call the parameter $\lambda \in J$ an *eigenvalue* of $T(\cdot)$, whenever Eq. (1)

$$T(\lambda)x = 0$$

has a nontrivial solution $x \neq 0$, which we call an *eigenvector* corresponding to $\lambda$.

It is well known that all eigenvalues of a linear Hermitian problem $Ax = \lambda x$ are real and if they are ordered, $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, it is possible to characterize them by the minmax principle of Poincaré

**Theorem 1** (Minmax principle of Poincaré) *Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the ordered eigenvalues of $Ax = \lambda x$ then*

$$\lambda_k = \min_{\mathcal{W}\in S_k} \max_{w\in\mathcal{W}^1} w^* A w, \tag{2}$$

*where $S_k$ denotes the set of all k dimensional subspaces of $\mathbb{C}^n$ and $\mathcal{W}^1 := \{w \in \mathcal{W} : \|w\|_2 = 1\}$ is the unit sphere in $\mathcal{W}$.*

It turns out, that a similar result holds also for a certain type of nonlinear eigenvalue problems.

**Definition 2** (*Rayleigh functional*) Let $f(\lambda; x) := x^* T(\lambda)x$ be a real function, continuous in J for every fixed $x \neq 0$. Assume that

$$f(\lambda; x) = 0 \tag{3}$$

has at most one solution $p(x) \in J$, then (3) implicitly defines a functional $p$ on some subset $D$ of $\mathbb{C}^n\backslash\{0\}$. We refer to $p$ as a *Rayleigh functional*, since it generalizes the notation of the Rayleigh quotient in the variational characterization of the eigenvalues of the linear problem.

We furthermore require that

$$f(\lambda; x)(\lambda - p(x)) > 0 \quad \text{for every } \lambda \in J \quad \text{and} \quad x \in D \quad \text{with } \lambda \neq p(x), \tag{4}$$

which is a natural generalization of the requirement that $B$ is positive definite for a linear pencil $(A, B)$.

Under these assumptions a variational characterization in terms of the Rayleigh functional has been considered by various authors. To mention a few, Duffin [9,10] and Rogers [24] proved the variational principle for the finite dimensional overdamped problems, i.e. problems for which the Rayleigh functional $p$ is defined on the entire space $\mathbb{C}^n\backslash\{0\}$. Nonoverdamped problems were considered by Werner and the second author [33,35].

The key to the variational principle is an adequate enumeration of the eigenvalues. In general, the natural enumeration, i.e. the first eigenvalue is the smallest one, followed by the second smallest one etc. is not appropriate (see [33,35]). Instead, the number of an eigenvalue $\lambda$ of the nonlinear problem (1) is inherited from the number of the eigenvalue 0 of the matrix $T(\lambda)$ based on the following consideration:

**Definition 3** (*Minmax induced numbering of eigenvalues*) Let $\lambda \in J$ be an eigenvalue of the nonlinear problem (1), then $\mu = 0$ is an eigenvalue of the linear problem $T(\lambda)x = \mu x$. Therefore there exists $k \in \mathbb{N}$ such that

$$0 = \max_{\mathcal{W} \in S_k} \min_{w \in \mathcal{W}^1} w^* T(\lambda) w$$

or equivalently that 0 is a $k$th largest eigenvalue of the matrix $T(\lambda)$. In this case we call $\lambda$ a *kth eigenvalue of* (1).

*Remark 1* For $T(\lambda) := \lambda B - A$, $B > 0$ it follows from the minmax characterization for linear eigenvalue problems that $\lambda$ is a $k$th eigenvalue of $T(\cdot)$ if and only if $\lambda$ is a $k$th smallest eigenvalue of the linear problem $Ax = \lambda Bx$.

*Remark 2* We note that if $T(\lambda)$ is differentiable w.r.t. $\lambda$ and $T'(\lambda)$ is positive definite, then replacing $T(\lambda)x = \mu x$ with the generalized eigenvalue problem $T(\lambda)x = \kappa T'(\lambda)x$ yields the same enumeration. This will be used later in Theorem 2.

With this enumeration the following minmax characterization of the eigenvalues of the nonlinear eigenproblem (1) was proved in [33,35]:

**Theorem 2** (Minmax characterization for eigenvalues of $T(\cdot)$) *For every $x \in D \subset \mathbb{C}^n$, $x \neq 0$ assume that the real Eq.* (3) *has at most one solution $p(x) \in J$, and let the definiteness condition* (4) *be satisfied.*

*Then the following assertions hold:*

(i) *For every $k \in \mathbb{N}$ there is at most one kth eigenvalue of problem* (1) *which can be characterized by*

$$\lambda_k = \min_{\substack{\mathcal{W} \in S_k, \\ \mathcal{W} \cap D \neq \emptyset}} \sup_{w \in \mathcal{W} \cap D} p(w). \tag{5}$$

*Hence, there are at most n eigenvalues of* (1) *in $J$.*

(ii) *If*

$$\lambda_k = \inf_{\substack{\mathcal{W} \in S_k, \\ \mathcal{W} \cap D \neq \emptyset}} \sup_{w \in \mathcal{W} \cap D} p(w) \in J$$

*then $\lambda_k$ is a kth eigenvalue of $T(\cdot)$ and* (5) *holds.*

(iii) *Assume that for $k < m$ the interval $J$ contains the kth and the mth eigenvalue $\lambda_k$ and $\lambda_m$, then $J$ contains all the eigenvalues $\lambda_j \in J$, $j = k, \dots, m$ and moreover it holds $\lambda_k \le \lambda_{k+1} \le \dots \le \lambda_m$.*

(iv) *If* $\lambda \in J$ *and* $k \in \mathbb{N}$ *such that problem* (1) *has a kth eigenvalue* $\lambda_k \in J$ *then it holds that*

$$\lambda \left\{ \begin{matrix} > \\ = \\ < \end{matrix} \right\} \lambda_k \quad \Longleftrightarrow \quad \mu_k(\lambda) := \max_{\mathcal{W} \in S_k} \min_{w \in \mathcal{W}^1} w^* T(\lambda) w \left\{ \begin{matrix} > \\ = \\ < \end{matrix} \right\} 0.$$

## 3 Iterative projection methods for nonlinear eigenproblems

For sparse linear eigenvalue problems, $Ax = \lambda x$, iterative projection methods are well-established and recognized as a very efficient tool. The key idea is to reduce the dimension of the eigenproblem by projecting it to a subspace of a much smaller dimension. The reduced problem is then handled by a fast technique for dense problems. Of course, this idea can only be successful if the subspace used for projection has good approximating properties w.r.t. some of the wanted eigenpairs, which translates to eigenvalues of the projected matrix being good approximations to the wanted eigenvalues of the large sparse matrix. In iterative projection methods the search subspace is expanded iteratively in a way promoting the approximation of the wanted eigenpairs. The generalizations of iterative projection methods to nonlinear eigenvalue problems were discussed in [1,4,7,11,15,17,19–21,25,29,31,32,34]. Two representative examples are the Nonlinear Arnoldi and Jacobi-Davidson methods. Both those methods extend the search subspace targeting a particular eigenvalue. In fact, there are no Krylov subspace methods (i.e. methods which as in linear case would admit a polynomial representation of the search subspace) working directly on the nonlinear eigenvalue problem without linearization. While applying iterative projection methods to general nonlinear eigenvalue problems with the objective to approximate more than one eigenpair, it is crucial to prevent the methods from converging to the same eigenpair repeatedly. In the linear case this is readily done by the Krylov subspace solvers or using partial Schur decomposition [12]. Unfortunately, a similar normal form does not exist for nonlinear eigenvalue problems. While this paper was in review, we became aware of a new approach to avoid repeated eigenpair convergence for general nonsymmetric eigenproblems based on minimal invariant pairs [11]. For nonlinear eigenvalue problems admitting a minmax characterization, in [4,29] it was proposed to use the induced eigenvalue ordering to remedy the problem. Algorithm 1 outlines a framework for iterative projection methods based on enumeration of the eigenvalues as discussed in Sect. 2.

There are many details that have to be considered when implementing an iterative projection method as outlined in Algorithm 1. The comprehensive review is out of scope of this work. Here, we restrict ourselves to only the essentials necessary for motivation and derivation of the local restart technique in Sect. 4. For more detailed discussion we refer the reader to [29,31,32].

### 3.1 Initialization

In order to preserve the numbering of the eigenvalues, the initial basis $V$ has to contain at least $j_{\min}$ linearly independent vectors. Let $\mathcal{W}$ be the invariant subspace of $T(\lambda_{j_{\min}})$

**Algorithm 1** Iterative Projection Method

```
Require:
1: First wanted eigenvalue number $j := j_{\min}$
2: Initial basis $V$, $V^*V = I$
3: Choose initial shift $\sigma$
4: Initial preconditioner $K \approx T(\sigma)^{-1}$, $\sigma$ close to first wanted eigenvalue, $\lambda_{j_{\min}}$
Execute:
5: while $j \le j_{\max}$ do
6:    Compute the eigenpair $(\tilde{\lambda}, y)$, $\tilde{\lambda}$ is the $j$th eigenvalue of the
      projected problem $T_V(\lambda)y := V^*T(\lambda)Vy = 0$
7:    Compute the Ritz pair $(\tilde{\lambda}, \tilde{x} := Vy)$ and its residual $r = T(\tilde{\lambda})\tilde{x}$
8:    if $(\tilde{\lambda}, \tilde{x})$ converged then
9:       return approximate eigenpair $(\lambda_j, x_j) := (\tilde{\lambda}, \tilde{x})$
10:      $j := j + 1$
11:      optionally choose a new shift $\sigma$ and recompute $K \approx T(\sigma)^{-1}$, if indicated
12:      optionally restart
13:      Choose approximation $(\tilde{\lambda}, \tilde{x})$ to the next eigenpair, and compute $r = T(\tilde{\lambda})\tilde{x}$
14:   end if
15:   Compute new direction $v$, e.g., $v = Kr$
16:   Orthonormalize and expand $V$, $v = v - VV^*v$, $v = v/\|v\|_2$, $V = [V, v]$
17:   Reorthogonalize $V$ if necessary
18:   Update projected problem $T_V(\lambda) := V^*T(\lambda)V$
19: end while
```

corresponding to its $j_{\min}$ largest eigenvalues then it holds that $z^*T(\lambda_{j_{\min}})z \ge 0$ for every $z \in \mathcal{W}$, and therefore by Theorem 2 $p(z) \le \lambda_{j_{\min}}$ for every $z \in \mathcal{W} \cap D$, and $\sup_{z \in \mathcal{W} \cap D} p(z) = \lambda_{j_{\min}}$. Hence, $\lambda_{j_{\min}}$ is a $j_{\min}$th eigenvalue of the orthogonal projection of $T(\cdot)$ onto $\mathcal{W}$, and a reasonable choice for the initial space $\mathcal{V}$ is the corresponding invariant subspace of $T(\tilde{\lambda})$ for some $\tilde{\lambda}$ close to $\lambda_{j_{\min}}$. Likewise, if $T(\cdot)$ is overdamped, then it holds that $z^*T(\lambda_{j_{\min}})z \ge 0$ for every $z \in \operatorname{span}\{x_1, \dots, x_{j_{\min}}\}$, where $x_j$ denotes the eigenvector of $T(\cdot)$ corresponding to $\lambda_j$, and the subspace spanned by $x_j$, $j = 1, \dots, j_{\min} - 1$ and additionally an approximation to $x_{j_{\min}}$ is also a reasonable choice for the initial search space.

## 3.2 Solution of a projected nonlinear eigenvalue problem (PNEP)

For nonlinear eigenvalue problem (1) let the columns of $V \in \mathbb{C}^n$ form a basis of the current search space $\mathcal{V} \subset \mathbb{C}^n$. Then it is easily seen that the projected problem

$$T_{\mathcal{V}}(\lambda)y := V^*T(\lambda)Vy = 0 \tag{6}$$

inherits the variational property, i.e. its eigenvalues in $J$ are minmax values of the restriction of the Rayleigh functional $p$ of $T(\cdot)$ to $D \cap \mathcal{V}$. Although, in general the enumeration of the eigenvalues of the original problem and the projected problem may differ.

There are many methods for solving small and dense nonlinear eigenvalue problems. For polynomial eigenvalue problems linearization is a natural choice, where the enumeration of eigenvalues in the sense of Sect. 2 can be deduced from the natural ordering of the real eigenvalues of the linearized problem. For general nonlinear eigenvalue problems safeguarded iteration [23] outlined in Algorithm 2 can be used for computing the $k$th eigenvalue of the nonlinear problem.

**Algorithm 2** Safeguarded Iteration

```
Require:
1: Initial approximation $\nu_1$ to the $k$th eigenvalue $\lambda_k$ of (1)
Execute:
2: for $i = 1, 2, \ldots$ until convergence do
3:     Determine eigenvector $\tilde{x}^i$ corresponding to the $k$th largest eigenvalue of $T(\nu_i)$
4:     Evaluate $\nu_{i+1} = p(\tilde{x}^i)$
5: end for
```

### 3.3 Subspace expansion

In general two approaches to subspace expansion can be found in the literature: Jacobi-Davidson [4] and Nonlinear Arnoldi [31] type expansion. Both schemes approximate inverse iteration, which is known to provide a direction with high approximating potential to the targeted eigenpair (cubical convergence for symmetric nonlinear eigenproblems if the eigenvalue approximation is updated with the Rayleigh functional).

Let $(\tilde{\lambda}_k, \tilde{x}_k)$ be a currently available approximation to the eigenpair and $r_k = T(\tilde{\lambda}_k)\tilde{x}_k$ its residual. In Jacobi-Davidson the search subspace is expanded by an orthogonal direction $t \perp \tilde{x}_k$ obtained from the following correction equation

$$\left(I - \frac{T'(\tilde{\lambda}_k)\tilde{x}_k\tilde{x}_k^*}{\tilde{x}_k^* T'(\tilde{\lambda}_k)\tilde{x}_k}\right) T(\tilde{\lambda}_k) \left(I - \frac{\tilde{x}_k\tilde{x}_k^*}{\tilde{x}_k^*\tilde{x}_k}\right) t = -r_k, \quad t \perp \tilde{x}_k. \tag{7}$$

If (7) is solved exactly we can expect asymptotically cubic convergence. The convergence rates of inexact Newton and Newton-like methods were studied in [27], and it is a common experience that even very coarse solution of (7) is sufficient to maintain a reasonably fast convergence.

The Nonlinear Arnodi method uses the direction of the residual inverse iteration [22]

$$v = T(\sigma)^{-1} T(\tilde{\lambda}_k)\tilde{x}_k,$$

where $\sigma$ is a fixed parameter close to the wanted eigenvalue $\lambda_k$. The Nonlinear Arnoldi method converges linearly, i.e. if $\tilde{x}_k^{i-1}$ and $\tilde{x}_k^i$ are two consecutive iterations with $\|\tilde{x}_k^{i-1}\| = \|\tilde{x}_k^i\| = 1$ and $\tau = \|T(\tilde{\lambda}_k^i)\tilde{x}_k^i\|_2 / \|T(\tilde{\lambda}_k^{i-1})\tilde{x}_k^{i-1}\|_2$ then $\tau = \mathcal{O}(|\lambda_k - \sigma|)$ (cf. [22]). For Hermitian problems if the eigenvalue approximations are updated with the value of the Rayleigh functional and $\sigma$ is updated with the previous approximation to $\lambda_k$, $\sigma = \tilde{\lambda}_k^{i-1}$, [26] the convergence is even quadratic. Moreover, if the linear system $T(\sigma)v = T(\tilde{\lambda}_k)\tilde{x}_k$ is too expensive to solve for $v$ we may choose as a new direction $v = KT(\tilde{\lambda}_k)\tilde{x}_k$ with $K \approx T(\sigma)^{-1}$.

## 3.4 Standard restarting based on global numbering

As the subspace expands in the course of the algorithm, the increasing storage and computational cost of the solution of the projected eigenvalue problem may make it necessary to restart the algorithm and purge some of the basis vectors. To be able to continue determining subsequent eigenpairs the correct enumeration has to be enforced at the restart.

If $J$ contains the first eigenvalue $\lambda_1 = \min_{x \in D} \; p(x)$, then e.g. the safeguarded iteration for the projected nonlinear problem (6) converges globally, i.e. for any initial vector $x \in \mathcal{V} \cap D$, to the smallest eigenvalue of (6) [23]. Furthermore, if the eigenvectors $x_j$ of the original problem (1) corresponding to the eigenvalues $\lambda_j, \; j = 1, \dots, k$, are contained in $\mathcal{V}$, then $\lambda_j$ is a $j$th eigenvalue of the projected problem (6), as well. Hence, expanding the search space $\mathcal{V}$ iteratively, and determining the $(k+1)$st eigenvalue of the projected problems, one gets a sequence of upper bounds to $\lambda_{k+1}$ which (hopefully) converges to $\lambda_{k+1}$. Thus, the eigenvalues of (1) can be determined quite safely one after the other by the iterative projection method starting with an approximation to $x_1$.

If $\inf_{x \in D} \; p(x) \notin J$ we can modify this approach in the following way. Let $k$ be the smallest integer such that $\lambda_k \in J$ where $k$ is chosen according to Definition 3. The minimum in (5) is attained by the invariant subspace $\mathcal{W}$ of $T(\lambda_k)$ spanned by the eigenvectors corresponding to its $k$ largest eigenvalues. Hence, if the current search space $\mathcal{V}$ satisfies $\mathcal{W} \subset \mathcal{V}$ then it is easily seen that $\lambda_k$ is the $k$th eigenvalue of the projected problem (6), i.e. again the numbering of the eigenvalues in the projected and in the original problem coincide, thus the eigenvalues can be determined successively.

In either case, for the numbering to be preserved, the search subspace after restart has to contain the eigenvectors corresponding to all the preceding eigenvalues in $J$ and if $\inf_{x \in D} \; p(x) \notin J$ also appropriate initial vectors, hence the restart requires global information. Notice that we only restart if an eigenvector has just converged since a restart destroys information on the eigenvectors and in particular on the currently iterated one.

## 3.5 Convergence criterion

In the course of our algorithm we accept an approximate eigenpair $(\tilde{\lambda}_k, \tilde{x}_k)$ as converged, if the residual norm $\|T(\tilde{\lambda}_k)\tilde{x}_k\|_2 / \|\tilde{x}_k\|_2$ is small enough. For a linear eigenvalue problem this is just the backward error of the eigenpair. For nonlinear holomorphic eigenvalue problems Szyld and Xue [28] performed a perturbation analysis of simple invariant pairs and derived an error estimate for their approximation. For algebraically simple eigenvalues (i.e. det $T'(\tilde{\lambda}_k) \neq 0$) their result essentially states that a small residual norm indicates a small backward error, as long as the Jacobian of the augmented system

$$\begin{bmatrix} T(\lambda)x \\ c^* x - 1 \end{bmatrix} = 0$$

is not ill-conditioned at the desired eigenpair $(\lambda_k, x_k)$. Here $c \in \mathbb{C}^n$ denotes a fixed vector with $c^* x_k \neq 0$.

## 4 A local restart technique

To overcome the problem of the search subspace dimension growing with the number of the sought eigenvalue inherent to global restarts (see Sect. 3.4) we propose to replace the global numbering of the eigenvalues by a local one. As the local numbering is obtained w.r.t. some chosen eigenvalue, only the corresponding eigenvector has to be included into the search subspace after a restart rather than the entire set of preceding eigenvectors or the invariant subspace of $T(\lambda_k)$.

### 4.1 Local numbering of eigenvalues

Assume that we are given an eigenpair $(\hat{\lambda}, \hat{x})$, $\hat{\lambda} \in J$ and $\hat{x} \in \mathbb{C}^n$, of the nonlinear eigenproblem (1). We refer to such an eigenpair as an anchor. In the following to avoid unnecessary technicalities we assume that $\hat{\lambda}$ is a simple eigenvalue, but all the results can be generalized to allow $\hat{\lambda}$ to be a multiple eigenvalue.

Let $\mathcal{V}$ be a subspace of $\mathbb{C}^n$ that contains $\hat{x}$, and let the columns of $V$ form a basis of $\mathcal{V}$. Then, along with the original family of matrices $T(\cdot)$, its projection $T_{\mathcal{V}}(\cdot) := V^* T(\cdot) V$ satisfies the conditions of Theorem 2. Therefore the Ritz values of $T(\cdot)$ with respect to $\mathcal{V}$, i.e. the eigenvalues of the projected eigenproblem (6)

$$T_{\mathcal{V}}(\lambda) y := V^* T(\lambda) V y = 0,$$

can be enumerated according to Definition 1. In particular, since $\hat{x} \in \mathcal{V}$, $\hat{\lambda}$ is also an eigenvalue of the projected problem (6), and $\hat{\lambda}$ can be assigned a local number $\ell = \ell(\mathcal{V})$ as follows:

$$\boxed{\begin{array}{c} \hat{\lambda} \text{ is the } \ell\text{th eigenvalue of the nonlinear problem } T_{\mathcal{V}}(\lambda) y = 0 \\ \Leftrightarrow \\ \mu(\hat{\lambda}) = 0 \text{ is the } \ell\text{th largest eigenvalue of the linear problem} \\ V^* T(\hat{\lambda}) V y = \mu(\hat{\lambda}) y. \end{array}}$$

The remaining eigenvalues of $T_{\mathcal{V}}(\cdot)$ (i.e. the Ritz values of $T(\cdot)$ with respect to $\mathcal{V}$) are given numbers relative to the anchor number, $\ell(\mathcal{V})$. We call such a relative numbering local.

*Example 1* Let $\mathcal{V} := \operatorname{span}\{x_1, x_3, x_7, x_8, x_{10}\}$ where $x_i$ is an eigenvector of (1) corresponding to the $i$th eigenvalue $\lambda_i$. Then the projected problem $T_{\mathcal{V}}(\lambda) y = 0$ has exactly the eigenvalues $\lambda_i$, $i = 1, 3, 7, 8, 10$ in $J$. For the anchor $\hat{x} := x_7$ it holds that $\ell = 3$, and the local numbers of the subsequent eigenvalues $\lambda_8$ and $\lambda_{10}$ are 4 and 5, respectively.

*Remark 3* Numerically, the local number of the anchor $\hat{\lambda}$, can be determined as the number of the eigenvalue of the linear problem $T_{\mathcal{V}}(\hat{\lambda})y = \mu(\hat{\lambda})y$ with the smallest absolute value: if $\mu_1 \geq \mu_2 \geq \cdots$ are its eigenvalues then

$$\ell(\mathcal{V}) := \arg \min_{k=1,\dots,\dim \mathcal{V}} |\mu_k|.$$

## 4.2 Spurious eigenvalues

In Example 1, the search subspace $\mathcal{V}$ has been chosen to contain eigenvectors only, and therefore successive eigenvalues of $T(\cdot)$ with eigenvectors in $\mathcal{V}$ have consecutive local numbers. However, in a course of iteration, the search subspace will also contain additional vectors besides the eigenvectors which can adversely affect the local numbering. Only for the sake of the following argument let us assume that the nonlinear eigenvalue problem (1) is overdamped such that the Rayleigh functional $p$ is defined on $\mathbb{C}^n \backslash \{0\}$. Hence the eigenvectors $X := \{x_1, \dots, x_n\}$ of $T(\cdot)$ corresponding to the $n$ eigenvalues arranged in the ascending order $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ form a basis of $\mathbb{C}^n$.

Let the current search subspace be $\mathcal{V}_k$, and the anchor pair $(\hat{\lambda}, \hat{x})$, $\hat{x} \in \mathcal{V}_k$. Assume that from the last restart the method has already computed the eigenvalues $\lambda_\ell^{\mathcal{V}_k} := \hat{\lambda} < \lambda_{\ell+1}^{\mathcal{V}_k} \leq \cdots \leq \lambda_{\ell+j}^{\mathcal{V}_k}$ of $T(\cdot)$, which are consecutive eigenvalues of the projected eigenproblem $T_{\mathcal{V}_k}(\lambda)y = 0$ with local numbers $\ell, \dots, \ell + j$. After expanding $\mathcal{V}_k$ to $\mathcal{V}_{k+1} := \text{span}\{V_k, v\} =: \text{span}\{V_{k+1}\}$ by some vector $v$, each of $\lambda_\ell^{\mathcal{V}_k} < \lambda_{\ell+1}^{\mathcal{V}_k} \leq \cdots \leq \lambda_{\ell+j}^{\mathcal{V}_k}$ remains an eigenvalue of the new projected problem $T_{\mathcal{V}_{k+1}}(\lambda)y = 0$. However, it may happen that $T_{\mathcal{V}_{k+1}}(\lambda)y = 0$ has an additional eigenpair $(\theta, y_\theta)$ such that $\lambda_\ell^{\mathcal{V}_k} < \theta \leq \lambda_{\ell+j}^{\mathcal{V}_k}$.

If $\lambda_\ell^{\mathcal{V}_k}$ were the smallest eigenvalue of $T(\cdot)$ i.e. $\lambda_\ell^{\mathcal{V}_k} = \lambda_1$, then it would be clear that at least one eigenvalue is missing in the interval $(\lambda_\ell^{\mathcal{V}_k}, \lambda_{\ell+j}^{\mathcal{V}_k}]$. However, with an anchor in the interior of the spectrum it is possible for the additional Ritz vector, $x_\theta := V_{k+1}y_\theta$, that its representation with respect to the eigenbasis $X$ of $T(\cdot)$, $x_\theta = \sum_i \alpha_i x_i$, contains components $\alpha_i x_i$ such that some of the corresponding eigenvalues $\lambda_i$ are smaller than $\lambda_\ell^{\mathcal{V}_k}$ and others are larger than $\lambda_{\ell+j}^{\mathcal{V}_k}$ (or larger equal if $\lambda_{\ell+j}^{\mathcal{V}_k}$ is a multiple eigenvalue of $T(\cdot)$). We call such a Ritz value $\theta$ a spurious eigenvalue of $T(\cdot)$. The presence of a spurious eigenvalue obviously causes an increase of the local numbers of all the subsequent eigenvalues.

*Remark 4* (The case $\theta = \hat{\lambda}$) Note that even if $\theta = \hat{\lambda}$ (up to precision to which the eigenvalues are computed), $x_\theta \neq \hat{x}$. Hence we can identify such a spurious pair $(\theta, x_\theta)$ (recall we assumed the anchor $\hat{\lambda}$ to be simple) and enforce the ordering in which $\theta$ precedes $\hat{\lambda}$ so it does not interfere with the local ordering. Therefore, it is sufficient to consider the case $\hat{\lambda} < \theta$.

Our argument took advantage of the existence of an eigenbasis of $\mathbb{C}^n$, which is a consequence of assuming that the nonlinear eigenvalue problem (1) is overdamped. It is clear, that the same can happen for nonoverdamped problems. The additional complication for nonoverdamped problems is that the linear combination can also contain vectors not in the domain of definition of the Rayleigh functional $p$, which can have the same effect.

Occurrence of spurious eigenvalues is inherent to interior eigenvalue computation. It also happens for linear problems, when no transformation is applied to the eigenproblem to map the eigenvalues from the interior to the lower or upper end of the spectrum. Hence, in order to algorithmically exploit the local numbering we need to find a way to recognize when the local numbering has been obscured by spurious eigenvalues and how to effectively restore it.

### 4.3 Local restart framework

Algorithm 3 outlines one local restart cycle, which we explain in detail below.

**Algorithm 3** Restart Framework

```
Require:
 1: Preconditioner $K \approx T(\sigma)^{-1}$ for a suitable pole $\sigma$
 2: Anchor pair $(\lambda_i, x_i)$, an (approximate) eigenpair of $T(\cdot)$
 3: optionally $v$ an approximation to $x_{i+1}$ otherwise $v :=$ rand
Execute:
 4: $V := [x_i]$
 5: $j := 1$
 6: while restart condition not satisfied do
 7:   repeat
 8:     Expand $V := [V, v]$
 9:     Determine the local number of the anchor, $\ell(V)$
10:     Compute $(\ell + j)$th eigenpair $(\tilde{\lambda}_{\ell+j}, \tilde{y}_{\ell+j})$ of $T_{\mathcal{V}}(\cdot) := V^* T(\cdot) V$
11:     Compute new expansion direction $v$ aiming at the $(\ell + j)$th eigenpair $(\lambda_{\ell+j}, x_{\ell+j})$
12:   until eigenpair $(\tilde{\lambda}_{\ell+j}, V\tilde{y}_{l+j}) =: (\lambda_{i+j}, x_{i+j})$ converged
13:   if either $(\lambda_{i+j}, x_{i+j}) \in \{(\lambda_{i+j'}, x_{i+j'}),\ 0 < j' < j\}$ or $\lambda_{i+j} < \lambda_{i+j-1}$ then
14:     $m' := j - j'$, where $j'$ is an integer such that $\lambda_{i+j} \in (\lambda_{i+j'-1}, \lambda_{i+j'}],\ 0 < j' < j$
15:     $(\theta_s, x_s) := \emptyset$
16:     for $m = 1, \ldots, m'$ do
17:       Locate suspect eigenvalue $\theta^m$, and its Ritz vector $x_\theta^m$
18:       if $(\theta^m, x_\theta^m)$ converged then
19:         Recover missed out eigenpair $(\theta^m, x_\theta^m)$ and adjust numbering
20:         Increase local offset $j := j + 1$
21:       else if $(\theta_s, x_s) = \emptyset$ then
22:         $(\theta_s, x_s) := (\theta^m, x_\theta^m)$ {Record first suspect Ritz pair for subspace expansion}
23:       end if
24:     end for
25:     if $(\theta_s, x_s) \neq \emptyset$ then
26:       Compute new expansion direction $v$ aiming at the suspect Ritzpair $(\theta_s, x_s)$
27:     else
28:       Compute new expansion direction $v$ aiming at the $(\ell + j)$th eigenpair $(\lambda_{\ell+j}, x_{\ell+j})$
29:     end if
30:   else
31:     Increase local offset $j := j + 1$
32:     Compute new expansion direction $v$ aiming at the $(\ell + j)$th eigenpair $(\lambda_{\ell+j}, x_{\ell+j})$
33:   end if
34: end while
```

*Lines (1–5): Initialization* Here, the only difference between the local and and global restarts is the initialization subspace, $\mathcal{V}_0 := \text{span}\{\hat{x}, v\}$. For local restarts $\mathcal{V}_0$ contains only the eigenvector corresponding to the anchor, $\hat{x}$, along with $v \in \mathbb{C}^n$ an approximation to the next eigenvector such that $T_{\mathcal{V}_0}$ has an eigenvalue $\omega \in J$ with $\omega > \hat{\lambda}$. Starting with $\mathcal{V}_0$ we determine approximations to the eigenvalue subsequent to the anchor $\hat{\lambda}$ by projecting the nonlinear eigenvalue problem (1) to a sequence of subspaces $\mathcal{V}_0 \subset \mathcal{V}_1 \subset \mathcal{V}_2 \subset \ldots$.

*Lines (7–12): Computation of the $(\ell + j)$th eigenpair $(\lambda_{\ell+j}, x_{\ell+j})$* Let $\mathcal{V}_k$ be the current search space and $\ell(\mathcal{V}_k)$ the local number of $\hat{\lambda}$. Note that the number $\ell(\mathcal{V}_k)$ of the anchor may change in the course of the iteration hence its dependence on $\mathcal{V}_k$. Suppose that we have successively computed $j$ eigenvalues of $T(\cdot)$ in $J$,

$$\hat{\lambda} = \lambda_i = \lambda_\ell^{\mathcal{V}_k} < \lambda_{\ell+1}^{\mathcal{V}_k} \leq \cdots \leq \lambda_{\ell+j-q-1}^{\mathcal{V}_k} < \lambda_{\ell+j-q}^{\mathcal{V}_k} = \cdots = \lambda_{\ell+j-1}^{\mathcal{V}_k} = \lambda_{i+j-1} =: \check{\lambda},$$

and let $\check{X}$ be the $1 \leq q \leq j$ dimensional eigenspace of $T(\cdot)$ corresponding to $\check{\lambda}$. We are now aiming at the next eigenvalue of $T(\cdot)$. To this end, we compute the eigenpair $(\omega, y_\omega), \omega \in J$ of the current projected problem $T_{\mathcal{V}_k}(\cdot)$ with the local number $\ell(\mathcal{V}_k)+j$. We expand the search space, $\mathcal{V}_k$ to $\mathcal{V}_{k+1}$, by a new search direction $v$ aiming at the Ritz pair $(\omega, V_k y_\omega)$, e.g., $v = KT(\omega)V_k y_\omega$ for the Nonlinear Arnoldi method (we hope that, by using such a strategy, as the iteration progresses the search subspace will contain an increasingly significant component of the eigenvector $x_{\ell+j}$). We then solve the new projected eigenproblem $T_{\mathcal{V}_{k+1}}(\cdot)$ for the Ritz pair with the desired local number $\ell(\mathcal{V}_{k+1}) + j$ and we repeat this iterative process until the Ritz pair with the desired local number has converged (yields a sufficiently small eigenresidual norm, see Sect. 3.5).

*Remark 5* If $v \in \mathbb{C}^n$ in the initial subspace $\mathcal{V}_0$ is a poor approximation to $x_{\ell+1}$ and $T_{\mathcal{V}_0}$ has an eigenvalue $\omega \in J$, $\omega \leq \hat{\lambda}$ i.e. $\ell(\mathcal{V}_0) = \dim(\mathcal{V}_0)$ we return the eigenpair with the largest number in the subspace (here the anchor itself, $(\hat{\lambda}, \hat{x})$). As $(\hat{\lambda}, \hat{x})$ is accurate up to the computational tolerance, its residual $T(\hat{\lambda})\hat{x}$, similarly as a random vector, is expected to be a linear combination of many eigenvectors. The subspace expansion step, e.g. $v = KT(\hat{\lambda})\hat{x}$, then amplifies those eigendirections corresponding to the eigenvalues close to the pole $\sigma$ $(K \approx T^{-1}(\sigma))$ and within a few steps we expect the projected problem $T_{\mathcal{V}}$ to have an eigenvalue $\omega > \hat{\lambda}$.

*Line 13: Check if a new eigenpair has been computed* Ideally, the converged Ritz pair $(\omega, x_\omega)$ is a new eigenpair of the original problem (1). However, it may happen that the algorithm returns a replicate eigenvalue with number smaller than $i + j$ or a new eigenvalue $\hat{\lambda} < \bar{\lambda} < \check{\lambda} = \lambda_{i+j-1}$ (eigenvalue which has been previously missed out).

From the discussion in Sect. 4.2 we infer that such behaviour occurrs due to the local numbering being altered i.e. one or more additional eigenvalues exist in $(\hat{\lambda}, \check{\lambda}]$ correspondingly rising the local number of $\check{\lambda}$. Henceforth we will refer to such eigenvalues as "suspect". All such suspect eigenvalues can be identified and the missed out eigenvalues can be accepted while the spurious eigenvalues can be treated in the way described below one after the other.

*Lines (14–29): Restoring local numbering* For the sake of the following argument we assume that the algorithm returned a replicate eigenvalue $\omega = \check{\lambda}$, while any other case follows analogously. Such a repeated convergence of eigenvalues may happen in two cases: (1) $\check{\lambda}$ has a higher geometric multiplicity than $q$ (at least $q+1$), and (2) $\check{\lambda}$ is an eigenvalue with multiplicity $q$ and $1 \leq m' \leq q$ additional eigenvalues exist in $(\hat{\lambda}, \check{\lambda}]$.

If $\check{\lambda}$ has multiplicity at least $q+1$, which can be ascertained as described in Lemma 1, we simply accept $(\check{\lambda}, x_\omega)$ as a newly computed eigenpair and proceed to compute the next eigenvalue whose local number is by 1 larger than the largest local number of $\check{\lambda}$.

**Lemma 1** *If the angle between the eigenspace $\check{X}$, dim $\check{X} = q$, and $x_\omega$ is different from* 0 *(in the numerical practice, larger than a prescribed small threshold) or if $\check{\lambda}$ is the $(\ell(\mathcal{V}^\perp) + j - q)$th eigenvalue of the projected problem*

$$V^{\perp *} T(\lambda) V^\perp y = 0, \tag{8}$$

*where $\mathcal{V}$ is the current search space, $V^\perp$ denotes a basis of $\mathcal{V}^\perp$ the orthogonal complement of $\check{X}$ in $\mathcal{V}$, and $\ell(\mathcal{V}^\perp)$ the local number of $\hat{\lambda}$, then $\check{\lambda}$ is a multiple eigenvalue* (*with multiplicity at least* $q+1$).

Notice, that the number of columns of $V^\perp$, $\dim(\mathcal{V}) - q$, is usually quite small and therefore it can be easily verified with safeguarded iteration whether $\check{\lambda}$ is a $(\ell(\mathcal{V}^\perp) + j - q)$th eigenvalue of the projected eigenproblem (8) or not.

In the second case, there are two possible reasons for the current projected problem having an additional eigenvalue $\theta \in (\hat{\lambda}, \check{\lambda}]$ such that the corresponding Ritz pair $(\theta, x_\theta) \neq (\lambda_{i+j'}, x_{i+j'}),\ j' = 1, \ldots, j-1$:

1. *Missed out eigenvalue* An eigenvalue of the original problem (1) in the interval $(\hat{\lambda}, \check{\lambda}]$ might have been previously missed out because the corresponding eigenvector $x_\theta$ was not sufficiently present in the initial search space $\mathcal{V}_0$ and might not have been amplified sufficiently in the course of the expansions of $\mathcal{V}$ until computing $\check{\lambda}$. Afterwards the component of $x_\theta$ in the search space $\mathcal{V}$ has grown large enough to produce the additional eigenvalue $\theta \in (\hat{\lambda}, \check{\lambda}]$, and Algorithm 3 yields the eigenvalue $\check{\lambda}$ the $(q+1)$st time with a different local number.
2. *Spurious eigenvalue* It might be the case that no eigenvalue of (1) is missing in $(\hat{\lambda}, \check{\lambda}]$ but the newly produced eigenvalue $\theta$ of the projected problem (6) is a spurious one, i.e. the corresponding Ritz vector $x_\theta$ is a linear combination of eigenvectors of (1) corresponding to eigenvalues less than $\hat{\lambda}$ and eigenvalues greater than $\check{\lambda}$ (or greater equal if $\check{\lambda}$ has a higher geometrical multiplicity than computed so far) and possibly some vectors outside of the domain of definition of the Rayleigh functional if the problem is not overdamped.

In both cases we identify the additional eigenvalue $\theta$ and its local number $\ell + j_\theta$, and we expand the search space aiming at $(\theta, x_\theta)$ (in other words, we add a new search direction $v$, which is either $KT(\theta)x_\theta$ for the Nonlinear Arnoldi method, or the approximate solution of the Jacobi–Davidson correction Eq. (7) with right–hand side $-T(\theta)x_\theta$). Then for the projected problem on such extended subspace $\mathcal{V}_\theta$

$$T_{\mathcal{V}_\theta}(\lambda)y = 0 \tag{9}$$

either of the following holds:

- Problem (9) has exactly $j-1$ eigenvalues in $(\hat{\lambda}, \check{\lambda}]$, i.e. the additional eigenvalue has left the interval of interest and the numbering in $[\hat{\lambda}, \check{\lambda}]$ has been restored.
- There are still $j$ eigenvalues in $(\hat{\lambda}, \check{\lambda}]$. In this case we repeat the expansion of the subspace until the additional eigenvalue has been moved out from the interval $[\hat{\lambda}, \check{\lambda}]$ or the sequence of additional Ritz values has converged to a previously missed out regular eigenvalue, in which case we adjust the enumeration of the eigenvalues and increase $j$ by 1.

After the enumeration has been restored we continue with the iterative projection method targeting the eigenvalue with the local number $\ell + j$.

*Remark 6* In particular if more than one additional eigenvalue exist in $(\hat{\lambda}, \check{\lambda}]$, the Algorithm 3 will first identify and recover all missed out eigenvalues. Then the first of the found spurious eigenvalues (i.e. with the smallest local number) will be targeted.

*Lines 31–32: Targeting next eigenvalue* After convergence of the eigenvalue we may continue the iterative projection method aiming at the $(\ell(\mathcal{V}_k)+j+1)$st eigenvalue or we may replace the anchor with the newly converged eigenpair and target the eigenvalues subsequent to the new anchor. Since the current search space contains useful information about further eigenvalues it is advisable to continue expanding the search space until the convergence becomes too slow (notice that for the residual inverse iteration the convergence factor $\tau$ depends on the distance between the shift and the wanted eigenvalue) or the dimension exceeds a given bound.

## 4.4 Automated local restart

For certain problems, the cost to set up a restart, i.e. time for computing the preconditioner, generating the new search space and the projected problem, is relatively high in comparison to the remaining computations. We can further improve the performance allowing the algorithm to balance those time-consuming tasks automatically.

Let $t_r$ denote the time for the setup of a restart, and let $t_e^j$ be the time needed for computing the $(\ell + j)$th eigenvalue of problem (1), i.e. $j$ denotes the offset of the eigenvalue with respect to the anchor after a restart. Then the total time for computing the first $j$ eigenvalues after the restart is $t_t^j = t_r + \sum_{k=1}^{j} t_e^k$, and hence the running average time for computing one eigenvalue since last restart is $\bar{t}_e^j = t_t^j/j$. Notice, that as we compute more and more eigenvalues the setup time per eigenvalue decreases.

Let $\alpha \geq 1$ and $N_\alpha \in \mathbb{N}_0$ be parameters depending on the given problem, and we initialize $n_\alpha := N_\alpha$. After computing the $j$th eigenpair since a restart we adjust $n_\alpha$ in the following way

$$n_\alpha \leftarrow \begin{cases} \min\{N_\alpha, n_\alpha + 1\} & \text{if} \quad t_e^j \le \alpha \cdot \bar{t}_e^j \\ n_\alpha - 1 & \text{else} \end{cases}$$

Whenever $n_\alpha < 0$ we restart the method. The presented strategy compares the time required for convergence of an eigenvalue with the running average time and triggers a restart when the eigenvalue convergence is repeatedly slower by factor $\alpha$ than in average. In particular, if $N_\alpha = 0$ and $\alpha = 1$ we restart the algorithm straightaway when the time for convergence to an eigenvalue exceeds the average time for computing the eigenvalues since the last restart.

## 5 Framework for restarting nonlinear iterative projection methods

Integrating the local restart with the iterative projection methods, we arrive at the framework for restarting of nonlinear iterative projection methods summarized in Algorithm 4.

An initial anchor pair can be determined for instance with an iterative projection method expanding the search space by $KT(\tilde{\sigma})\tilde{x}_k$ where $\tilde{\sigma} \neq \sigma$ are both fixed shifts close to the wanted eigenvalues, $K \approx T(\sigma)^{-1}$ is a preconditioner, and $\tilde{x}_k$ are the iterates of the projection method aiming at the eigenvalue closest to $\tilde{\sigma}$. Alternatively we could use a direction suggested by the Jacobi Davidson method for the linear eigenproblem $T(\sigma)x = \lambda T'(\tilde{\sigma})x$ aiming at its smallest eigenvalue in modulus. Obviously, no anchor eigenpair is required if $\inf_{x\in D} p(x) \in J$ and one is looking for eigenvalues at the lower end of the spectrum as the natural enumeration can be used in the first interval. After a restart one of the just computed eigenpairs can serve as an anchor.

More general, for nonlinear eigenvalue problems where the minmax induced ordering and the natural (here ascending) ordering coincide on $[a, b] \subset J$ (e.g. minmax admitting quadratic eigenvalue problem), it is also possible to use one of the bounds of the interval of interest $[a, b]$ and enumerate the eigenvalues relatively to this bound instead of relatively to an anchor. In such case, we compute the eigenvalues of the projected nonlinear problem $T_{\mathcal{V}}(\cdot)$ larger or equal $a$ until the first restart, when the anchor is reset to an already computed eigenpair. Corollary 1 is a direct consequence of Theorem 2 and it shows how to locate the first eigenvalue of the projected nonlinear problem $T_{\mathcal{V}}(\cdot)$ larger or equal $a$.

**Corollary 1** *Let $(\lambda_m^{\mathcal{V}}, y_m)$ be the first eigenvalue of the projected nonlinear problem $T_{\mathcal{V}}(\cdot)$ in the interval $[a, b]$. Then by assumption $\lambda_{m-1}^{\mathcal{V}} < a \le \lambda_m^{\mathcal{V}}$ and from Theorem 2 it follows*

$$a > \lambda_{m-1}^{\mathcal{V}} \quad \Leftrightarrow \quad \mu_{m-1}(a) = \max_{\mathcal{W}\in S_{m-1}} \min_{y\in\mathcal{W}^1} y^* T_{\mathcal{V}}(a) y > 0$$
$$a \le \lambda_m^{\mathcal{V}} \quad \Leftrightarrow \quad \mu_m(a) = \max_{\mathcal{W}\in S_m} \min_{y\in\mathcal{W}^1} y^* T_{\mathcal{V}}(a) y \le 0.$$

*Thus, the local number, $m$, of the first eigenpair of $T_{\mathcal{V}}(\cdot)$ in the interval $[a, b]$ is the number of the largest nonpositive eigenvalue, $\mu(a) \le 0$, of $T_{\mathcal{V}}(a)$.*

**Algorithm 4** Framework for Restarting of Nonlinear Iterative Projection Methods

```
Require:
1: Eigenvalue interval $[a,b]$, **optionally** $a = \lambda_i, b = \lambda_{i_{\max}}$
2: **optionally** Anchor vector $\hat{x}$ : $(\hat{\lambda} := a, \hat{x}) \approx (\lambda_i, x_i)$, $i$th eigenpair of $T(\cdot)$
3: **optionally** Initial basis $V$, $V^*V = I$
4: Choose initial shift $\sigma$ close to the lower interval bound, $a$
5: Compute initial preconditioner $K \approx T(\sigma)^{-1}$
6: **optionally** $v$ an approximation to $x_{i+1}$, otherwise $v :=$ rand
Execute:
7: **if** no anchor vector $\hat{x}$ **then**
8:     anchor_exists := $false$
9:     Initial search subspace $V := [\hat{x} :=$ rand$]$
10:    Local offset $j := 0$
11: **else**
12:    anchor_exists := $true$
13:    Initial search subspace $V := [\hat{x} \approx x_i]$
14:    **if** anchor pair $(\hat{\lambda}, \hat{x})$ sufficiently accurate **then**
15:        Local offset $j := 1$
16:    **else**
17:        Local offset $j := 0$
18:        anchor_exists = $false$
19:    **end if**
20: **end if**
21: **while** $\lambda_{i+j} < b$ **do**
22:    **while** restart condition not satisfied **do**
23:        **repeat**
24:            Expand $V := [V, v]$
25:            **if** anchor_exists **then**
26:                $\ell(V) :=$ local number of the anchor $\hat{\lambda}$
27:            **else**
28:                $\ell(V) :=$ local number of the first eigenvalue in the interval $[a,b]$
29:            **end if**
30:            Compute $(\ell + j)$th eigenpair $(\tilde{\lambda}_{\ell+j}, \tilde{y}_{\ell+j})$ of $T_V(\cdot)$
31:            Compute new expansion direction $v$ aiming at the $(\ell + j)$th eigenpair $(\lambda_{\ell+j}, x_{\ell+j})$
32:        **until** eigenpair $(\tilde{\lambda}_{\ell+j}, V\tilde{y}_{\ell+j}) =: (\lambda_{i+j}, x_{i+j})$ converged
33:        **if either** $(\lambda_{i+j}, x_{i+j}) \in \{(\lambda_{i+j'}, x_{i+j'}),\ 0 < j' < j\}$ **or** $\lambda_{i+j} < \lambda_{i+j-1}$ **then**
34:            $m' := j - j'$, where $j'$ is an integer such that $\lambda_{i+j} \in (\lambda_{i+j'-1}, \lambda_{i+j'}],\ 0 < j' < j$
35:            $(\theta_s, x_s) := \emptyset$
36:            **for** $m = 1, \ldots, m'$ **do**
37:                Locate suspect eigenvalue $\theta^m$, and its Ritz vector $x_\theta^m$
38:                **if** $(\theta^m, x_\theta^m)$ converged **then**
39:                    Recover missed out eigenpair $(\theta^m, x_\theta^m)$ and adjust numbering
40:                    Increase local offset $j := j + 1$
41:                **else if** $(\theta_s, x_s) = \emptyset$ **then**
42:                    $(\theta_s, x_s) := (\theta^m, x_\theta^m)$ {Record first suspect Ritz pair for subspace expansion}
43:                **end if**
44:            **end for**
45:            **if** $(\theta_s, x_s) \neq \emptyset$ **then**
46:                Compute new expansion direction $v$ aiming at the suspect Ritzpair $(\theta_s, x_s)$
47:            **else**
48:                Compute new expansion direction $v$ aiming at the $(\ell + j)$th eigenpair $(\lambda_{\ell+j}, x_{\ell+j})$
49:            **end if**
50:        **else**
51:            Increase local offset $j := j + 1$
52:            Compute new expansion direction $v$ aiming at the $(\ell + j)$th eigenpair $(\lambda_{\ell+j}, x_{\ell+j})$
53:        **end if**
54:    **end while**{restart condition not satisfied}
55:    Global anchor number becomes $i := i + j - 1 - n_{locked}$
56:    Reset the anchor to a recently computed eigenvalue, $\hat{\lambda} := \lambda_i$
57:    anchor_exists := $true$
58:    Reset local offset $j := n_{locked} + 1$
59:    Choose new shift $\sigma$ close to the approximation of the next eigenvalue $\tilde{\lambda}_{i+j}$
60:    Recompute preconditioner $K \approx T(\sigma)^{-1}$
61:    Reset search subspace $V :=$ orthonormalize$([x_i, \ldots, x_{i+n_{locked}}, \tilde{x}_{i+j}])$, where $\tilde{x}_{i+j}$ is an approximation to the $x_{i+j}$th eigen-
       vector
62:    Compute new expansion direction $v$ aiming at the $(\ell + j)$th eigenpair $(\lambda_{\ell+j}, x_{\ell+j})$
63: **end while**$\{\lambda_{i+j} < b\}$
```

Exactly as before we can target the eigenvalue with the local number $m$ and after it converged, the eigenvalue with the local number $m+1$, etc. Theoretically, after the eigenvalue with the $m$th local number converged this could be used as an anchor straight away. However, there is a danger of accepting $\lambda_m^{\mathcal{V}}$ as an anchor (hence the first eigenvalue in $[a, b]$) prematurely i.e. $\lambda_m^{\mathcal{V}}$ is not the first eigenvalue in $[a, b]$ of the original problem (1) because eigenvalues in $[a, \lambda_m^{\mathcal{V}})$ have been missed out. In this case the algorithm would continue to compute only the eigenvalues larger or equal $\lambda_m^{\mathcal{V}}$ permanently missing out the eigenvalues in $[a, \lambda_m^{\mathcal{V}})$. This is less likely to happen if the enumeration w.r.t. the bound $a$ is used until the first restart until when the search subspace is large and hence hopefully it includes the first and further consecutive eigenvalues of (1) in $[a, b]$.

We remark, that the missed out eigenvalues and the spurious eigenvalues have exactly the same effect regardless whether the interval bound $a$ or the anchor $\hat{\lambda}$ is used to relatively enumerate the eigenvalues i.e. the local number of the eigenvalues following such missed out/spurious eigenvalue is raised. Hence, they can be dealt with in the same way as described in Sect. 4.3.

Obviously, a very similar strategy can be applied when the anchor pair is not available to the required precision, i.e. its residual norm is above a set tolerance. We incorporated both those important cases in the pseudocode in Algorithm 4.

We might want to keep $n_{locked}$ eigenpairs in addition to the anchor pair at the restart, to minimize the occurrence of spurious eigenvalues. However the benefits have to be traded off against increased cost of the solution of the projected problems due to larger search subspace dimensions.

## 6 Numerical experiments

In this section we demonstrate the performance of the local restarts on a range of nonlinear eigenvalue problems. All the tests were performed with the QARPACK MATLAB package [2] on a desktop machine with two quadcore Intel Xeon X5550, 2.67GHz processors and 48 GB RAM. The LU decompositions and the subsequent system solves for small problems (small gyroscopic problem, “wiresaw1(2000)”, “wiresaw2(2000)”) were performed using MATLAB built-in LU and for large problems (large gyroscopic problem, delay problem, rational problem, “acoustic wave 1d” problem) with the MATLAB’s LU routine with five outputs. For all quadratic problems the projected problems were solved by linearization while for general nonlinear problems with safeguarded iteration.

The results are uniformly presented in terms of elapsed CPU times. We preconditioned the Nonlinear Arnoldi method with the LU factorization of the real part of $T(\sigma)$ where $\sigma$ is a shift not too far away from the wanted eigenvalues. We chose to neglect the imaginary part of $T(\sigma)$ since its influence on the action of the preconditioner is small in our examples, not justifying the extra effort of using complex arithmetic. We updated the LU factorization at each restart.

Table 1 holds the details of the behavior of the Nonlinear Arnoldi method with the local restart strategy described in Sect. 4—Nonlinear Restarted Arnoldi (NRA)—for each of the solved nonlinear eigenvalue problems listed in the first column. The other columns of Table 1 from left to right denote: `dim`: dimension of the eigenvalue problem; `type`: type of the eigenvalue problem: `gyroscopic`, general `quadratic`, `exponential`, `rational`; $\mathcal{R}(\lambda) \in [a, b]$: interval containing the real part of the wanted eigenvalues; #$\lambda$: number of computed eigenvalues; `CPU[s]`: CPU time for solution of the nonlinear eigenvalue problem in seconds; `PNEP CPU[s]`: CPU time for solution of the projected nonlinear eigenvalue problems (PNEPs) in seconds; `#iter`: number of iterations; `#rest`: number of restarts. Values for all problems except for the large tire problem are averaged over 10 runs.

## 6.1 A conservative gyroscopic eigenvalue problem

We consider a conservative gyroscopic eigenvalue problem

$$T(\lambda)x = \lambda^2 Mx - i\lambda Gx - Kx = 0 \tag{10}$$

describing for instance the free vibrations of a rolling tire. It is well known that all its eigenvalues are real and occur in pairs $\pm\lambda$, the corresponding eigenvectors are complex conjugate, and the positive eigenvalues $0 < \lambda_1 \leq \cdots \leq \lambda_n$ satisfy the minmax characterization [10]

$$\lambda_i = \min_{\mathcal{W}\in S_i} \max_{w\in\mathcal{W}^1} p(w),$$

where $p(x)$ is the positive solution of the quadratic equation

$$x^*T(\lambda)x = \lambda^2 x^*Mx - i\lambda x^*Gx - x^*Kx = 0.$$

### *6.1.1 Qualitative properties of the method*

We start with showing some qualitative behavior of our method on a small example of a wheel, composed of solid elements with Mooney-Rivlin material, see Fig. 1. The wheel is pressed on the track and is rotating at a rate of 50Hz. It is discretized with 450 brick elements with 720 nodes yielding after application of boundary conditions, 1728 degrees of freedom.

For the purpose of comparison we computed all the eigenpairs in the interval [0,16,820] by a globally restarted Nonlinear Arnoldi method. This corresponds to the smallest 200 eigenvalues. In all experiments an eigenvalue was regarded as converged if its relative residual norm was smaller than $10^{-4}$. The preconditioner was recomputed, whenever the ratio $\tau = \|T(\tilde{\lambda}_k^s)\tilde{x}_k^s\|/\|T(\tilde{\lambda}_k^{s-1})\tilde{x}_k^{s-1}\|$ with $\|\tilde{x}_k^{s-1}\| = \|\tilde{x}_k^s\| = 1$, of two successive residual norms in the last two step $s-1, s$ before convergence of the eigenpair $(\lambda_k, x_k)$ exceeded 0.1. Note, that large $\tau$ indicates that $|\sigma - \lambda_k|$ is to large (see Sect. 3.3). To prevent the search subspace from getting arbitrarily large we used global restarts which were triggered whenever the search subspace dimension exceeded 230, an absolute threshold on the subspace dimension. In the global restart technique we restart the Nonlinear Arnoldi method with an orthogonal basis of the subspace spanned by all eigenvectors computed so far. The total computing time, and

**Table 1** Nonlinear Restarted Arnoldi (NRA) performance for a range of nonlinear eigenvalue problems

| NEP | dim | type | $\Re(\lambda) \in [a, b]$ | $\#\lambda$ | CPU [s] | PNEP CPU [s] | #iter | #rest |
|---|---|---|---|---|---|---|---|---|
| wheel | 1728 | gyro | (11,748,16,820] | 100 | 135.2 | 71.9 | 1083.6 | 26.7 |
| wiresaw1 | 2000 | gyro | [317,629] | 100 | 496.7 | 111.2 | 1197.4 | 21.9 |
| tire | 124,992 | gyro | [0, 20,000] | 388 | 12.028 [h] | 2.87 [h] | 5165 | 22 |
| delay | 39,601 | exp | [150,250] | 75 | 247.5 | 58.2 | 485.1 | 8.7 |
| fluid-str. | 36,040 | rat | [10,20] | 84 | 275.6 | 99.1 | 464 | 11.6 |
| wiresaw2 | 2000 | quad | [317,629] | 100 | 850.7 | 217.1 | 1060.1 | 12.4 |
| acoust.1d | 30,000 | quad | [0, 50] | 100 | 219.5 | 67.4 | 618.1 | 11.1 |

Values for all problems except for the large tire problem are averaged over 10 runs

**Fig. 1** Solid rubber wheel

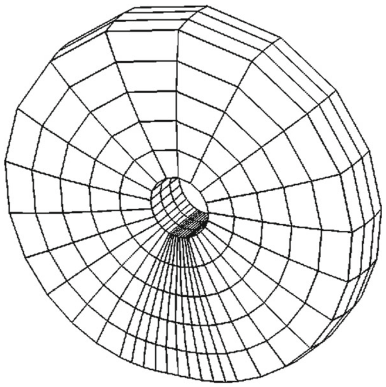

**Table 2** Global restarts with absolute and relative threshold

| | CPU [s] | PNEP CPU [s] | #LU | #rest |
|---|---|---|---|---|
| gl. rest. abs. thres. | 11,883 | 11,721 | 8 | 5 |
| gl. rest. rel. thres. | 2941 | 2782 | 39 | 41 |

Comparison of total CPU time, CPU time for solution of the projected nonlinear eigenvalue problems (PNEPs), number of LU decompositions and restarts

the time spent on solving the projected nonlinear problems for the wheel problem are summarized in Table 2.

In the next experiment we used the same global restart technique, but this time the restart was triggered whenever the dimension of the subspace exceeded the number of the last converged eigenvalue by 30, a relative threshold on the subspace dimension. In this way the average dimension of the search spaces and therefore the time for solving the projected problems were reduced, see Table 2. Plotting the total computing time and the time for solving the projected nonlinear problems in Fig. 2 reveals a super-linear growth. In fact, the CPU time spent on solving the projected eigenproblems itself grows super-linearly, determining the general trend.

Next, we computed the smallest 200 eigenvalues with Nonlinear Restarted Arnoldi. A restart was triggered whenever the search space dimension exceeded 80 or the convergence rate $\tau$ became larger than 0.5. Only the anchor and the current approximation were kept in the search subspace at restart ($n_{locked} = 0$). The experiment was repeated 10 times and the averaged elapsed computing times are shown in Fig. 2. The super-linear time growth has been effectively eliminated through the local restart strategy. The zoom into the lower end of the spectrum reveals that the global restart can outperform the local restart with fixed search space dimension limit in the initial phase when all the eigenvalues from the beginning of the spectrum are computed (Fig. 3). However, as it can be seen in Figs. 4 and 5 the local restart with automatic balancing outperforms the global restart also in the initial phase.

The outstanding advantage of the local restart strategy is its ability of computing eigenvalues in an interval in the interior of the spectrum without the need of computing all the preceding eigenpairs. Using the same local restart strategy we computed all the

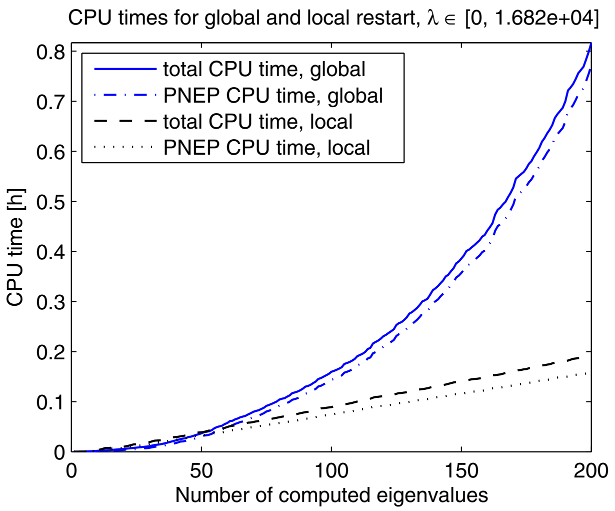

**Fig. 2** Global versus local restart for first 200 eigenvalues of the gyroscopic wheel problem

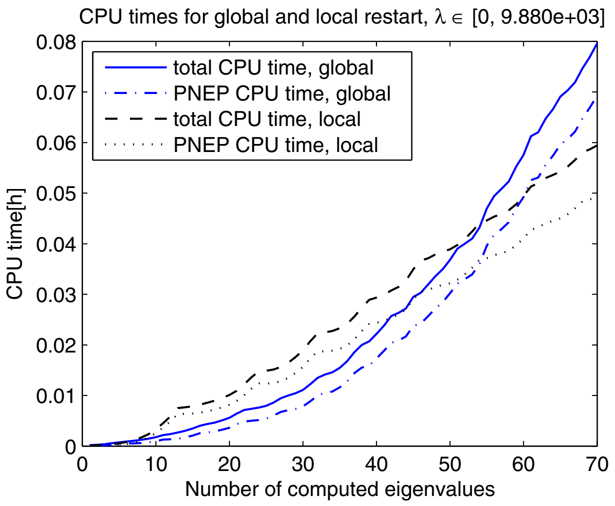

**Fig. 3** Global versus local restart—zoom into the *lower end* of the spectrum in Fig. 2. When computing the eigenvalues at the beginning of the spectrum in the initial phase the global restart can outperform the local restart with fixed search subspace dimension limit

eigenpairs in $(\lambda_{100} = 11{,}748,\ 16{,}820]$ which corresponds to $\lambda_{101}, \ldots, \lambda_{200}$ using the eigenpair $(\lambda_{100}, x_{100})$ as the initial anchor. Again for reproducibility of the results we repeated the experiment 10 times. As expected the averaged computing time has been approximately halved from 700 s to 350 s, Fig. 6. To illustrate typical behavior of the method in more detail, in Fig. 7 for just one run of the experiment we plotted histograms of the computed eigenvalues and of the occurrences of the spurious eigenvalues in each of the intervals between the consecutive restarts. The corresponding eigenvalue

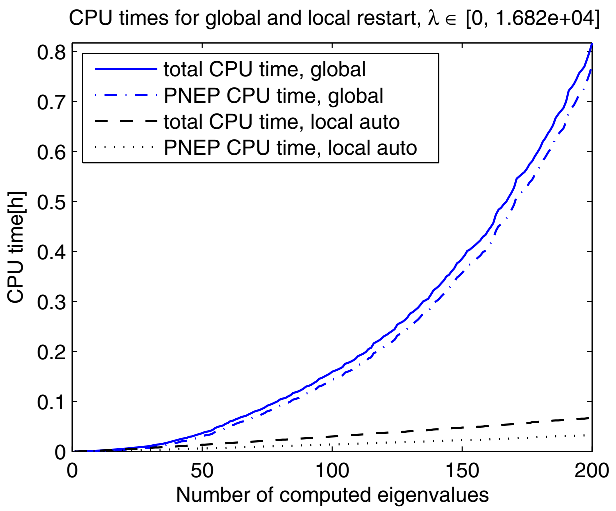

**Fig. 4** Global versus local restart with automatic balancing for first 200 eigenvalues of the gyroscopic wheel problem

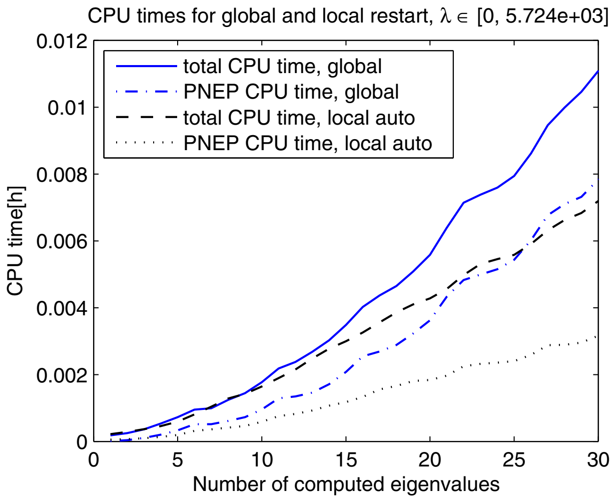

**Fig. 5** Global versus local restart with automatic balancing—zoom into the *lower end* of the spectrum in Fig. 4. We observe that the balancing effectively restores the superior performance of the local restart over the global restart also in the initial phase

convergence history throughout first 500 iterations is depicted in Fig. 8. The dots not encircled pin down the occurrence of the spurious values during the iteration e.g. in iterations 98, 159, 213 or 312 in Fig. 8 (cf. histogram in Fig. 7).

The automated restart strategy described in Sect. 4.4, can be used to let the algorithm balance the limit on the search subspace size on fly. Using the automated restart with $\alpha = 1$ and $N_\alpha = 1$ further reduced the average CPU time to about 130 s, see Fig. 6.

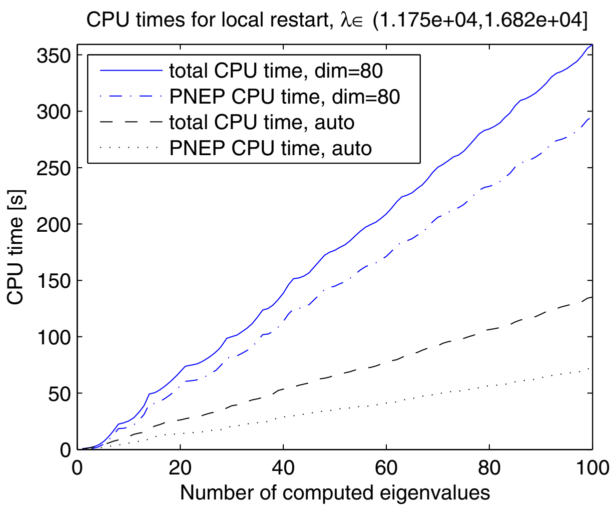

**Fig. 6** Local restart with automatic balancing for eigenvalues in ($\lambda_{100} = 11{,}748,\ 16{,}820$] of the gyroscopic wheel problem (eigenvalues $\lambda_{101}, \ldots, \lambda_{200}$)

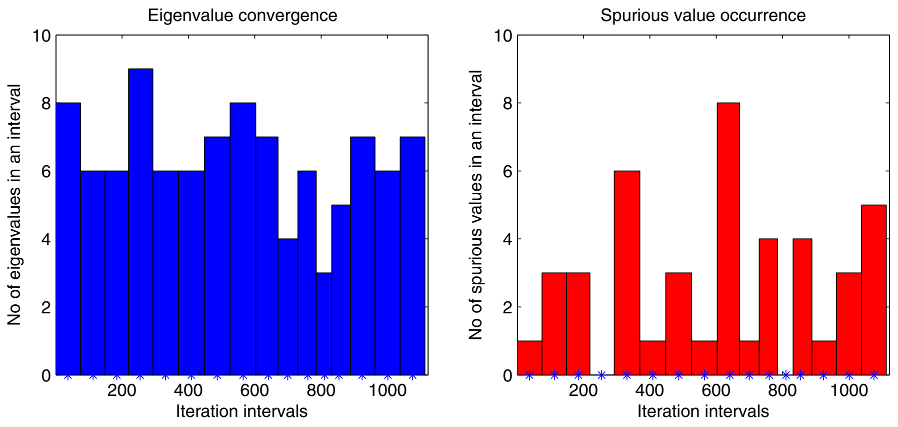

**Fig. 7** Histogram of *left* eigenvalue convergence, *right* spurious value occurrence per interval between consecutive restarts in one run of computation of eigenvalues in (11,748, 16,820] of the gyroscopic wheel problem

Figures 2, 3, 4, 5 and 6 demonstrate that using the local restart technique, the cost for computing one eigenvalue is approximately the same throughout the iteration, no matter what is the eigenvalue number. Thus we conclude that the new local restart technique effectively eliminates the super-linear CPU time growth with the number of computed eigenvalues and hence constitutes an efficient method for computing eigenvalues in the interior of the spectrum.

For the purpose of performance comparison of Nonlinear Arnoldi (NA) and Jacobi-Davidson (JD) type subspace expansions, we computed all the eigenpairs of the wheel problem in the interval ($\lambda_{100} = 11{,}748,\ 16{,}820$] using both subspace expansions.

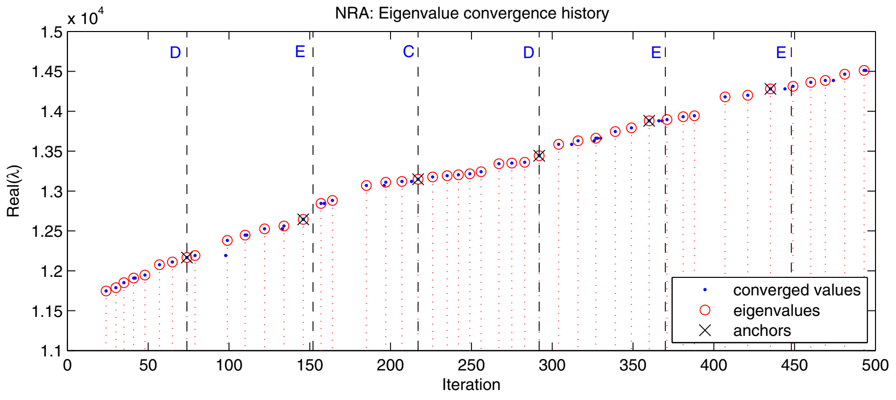

**Fig. 8** Eigenvalue convergence history throughout first 500 iterations of NRA while computing eigenvalues in (11,748, 16,820] of the gyroscopic wheel problem

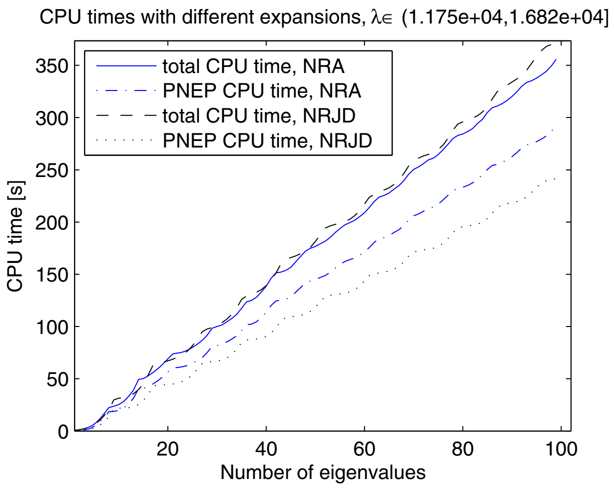

**Fig. 9** CPU time of NRA and NRJD for computation of eigenvalues in ($\lambda_{100}$ = 11,748, 16,820] of the gyroscopic wheel problem without automatic balancing

Each of the experiments was repeated 10 times and the performance figures were averaged through 10 runs. The automatic balancing was switched off, to focus on the effect of subspace expansion only. In both cases the methods consistently found all eigenvalues. Nonlinear Restarted Arnoldi (NRA) needed on average 1082.6 iterations and 14 restarts (15 LU factorizations), while Nonlinear Restarted Jacobi-Davidson (NRJD) 876.9 and 11 (12), respectively. Nonetheless, the NRA variant is still slightly faster in terms of the total CPU time, see Fig. 9. This is due to an JD expansion step being more expensive than an NA expansion step.

The here used preconditioner (LU decomposition of $K - \sigma^2 M$) remains of reasonable quality in the spectrum of interest. The results are in line with our general experience that Nonlinear Arnoldi method is faster whenever a good quality precondi-

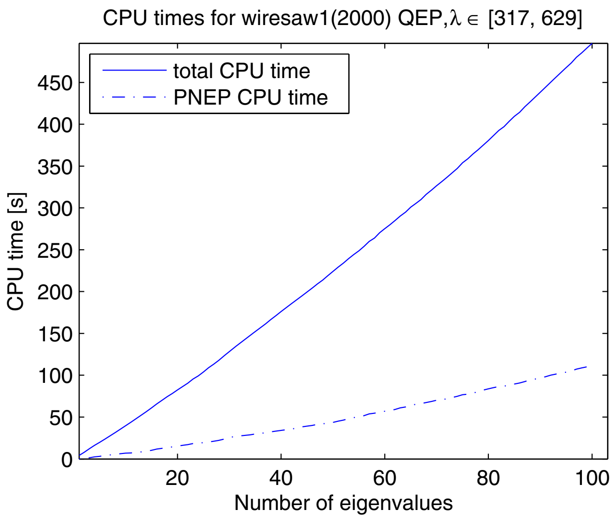

**Fig. 10** CPU time of NRA for computation of eigenvalues in [317, 629] of the NLEVP "wiresaw1" problem of dimension 2000

tioner is available, while Jacobi-Davidson method is more robust with respect to poor preconditioning [34].

### *6.1.2 NLEVP "wiresaw1" gyroscopic QEP*

As a second example we solve the gyroscopic problem arising from the vibration analysis of a wiresaw, "wiresaw1" from the NLEVP collection [6] of dimension 2000 and with the NLEVP default value of the wire speed parameter $v = 0.01$. The gyroscopic matrix $G$ for this problem is not sparse, hence the moderate choice of problem dimension. In formulation (10) all the eigenvalues are real, and are growing by approximately $\pi$ increment from one eigenvalue to the next.

We computed all 100 eigenvalues in the interval [317, 629]. The algorithm was initialized using the lower bound of the interval rather than an anchor. The relative residual tolerance was chosen to $10^{-4}$, the maximal subspace dimension to 120, the number of locked eigenvectors after the restart $n_{locked} = 0$ and the slowest admissible convergence rate $\tau = 0.5$. We used automated restarts with $\alpha = 1$ and $N_\alpha = 1$. Figure 10 shows the CPU time and the time for solution of the projected nonlinear problems averaged over 10 runs. The method took on average 1197 iterations with 22 restarts (23 LU decompositions) to compute the 100 eigenvalues. The average total CPU time was 497 s and the time for solution of the nonlinear projected problems 111 s (q.v. Table 1).

### *6.1.3 Large sparse gyroscopic QEP*

Our third example is a tire 205/55R16-91H cf. [19] (see Fig. 11) provided by Continental AG. The tire is discretized with 39,204 brick elements and 42,876 nodes. The nodes at the wheel rim are fixed resulting in 124,992 degrees of freedom. To account

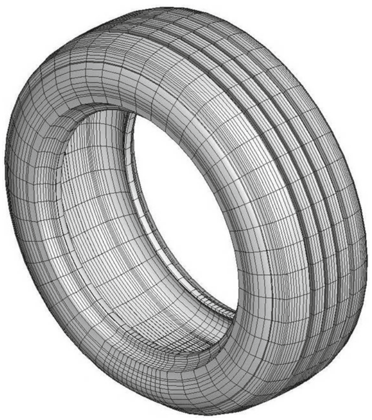
**Fig. 11** Continental AG 205/55R16-91H tire

for a complex structure of the tire, 19 different materials are used in the finite element model. The model includes the stress approximating the air pressure in the tire. The tire is pressed on the track and is rotating at a rate corresponding to a vehicle speed of 50 km/h.

We used Nonlinear Restarted Arnoldi method with MATLAB's five output LU routine as a preconditioner and set the tolerance for the relative residual norm to be $10^{-6}$ and the maximal search subspace dimension to 300. After each restart, only the anchor vector and the next approximation were kept in the subspace (i.e. $n_{locked} = 0$). The preconditioner was recomputed after at most 300 iterations, subject to residual norm ratio of at most $\tau = 0.9$ and automatic balancing with $N_\alpha = 1$ and $\alpha = 1$.

We computed all the eigenvalues in the interval [0, 20,000]. NRA needed 5165 iterations and 22 restarts (23 LU factorizations) to find all 388 eigenvalues in this interval. Figure 12 shows the total CPU time and the CPU time for solving projected nonlinear eigenvalue problems. We observe a slight increase in CPU time per eigenvalue, while we compute the eigenvalues at the lower end of the spectrum, which saturates at about 150th eigenvalue. Here, the reason is an increasing occurrence of spurious eigenvalues in proportion to the number of computed eigenvalues in the initial phase. For the eigenvalues higher in spectrum this effect settles, resulting in approximately constant time per eigenvalue computation. All but one restart were triggered through our automatic balancing strategy, demonstrating its effectiveness.

In this example we observed an increased occurrence of spurious values after the restarts. This leads us to believe that retaining some of the previously computed subspace after the restart may help to alleviate this effect, like for instance keeping further $n_{locked}$ eigenvectors along with the anchor in the local basis. Any benefits however, have to be traded off against increased computational cost due to larger search space dimensions.

We believe that the key to the optimal performance is to balance the subspace growth with the occurrence of spurious eigenvalues. An optimal strategy may include adapting the number of eigenvectors kept in the local basis along with the anchor, $n_{locked}$, in dependence of e.g. frequency of occurrence of spurious eigenvalues in the previous interval.

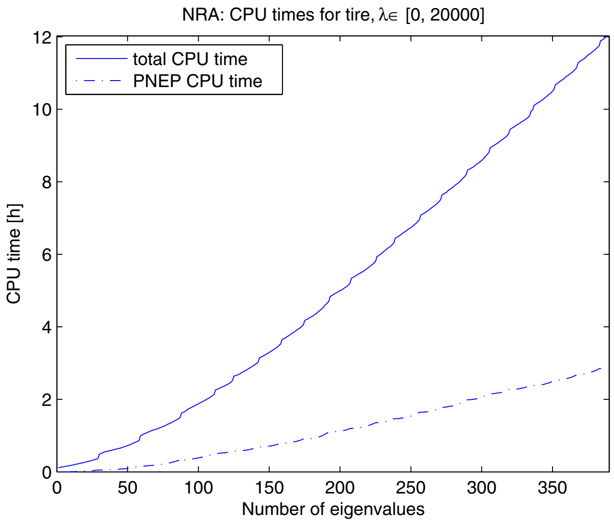

**Fig. 12** CPU time for NRA for eigenvalues in [0, 20,000] of the gyroscopic tire 205/55R16-91H problem

## 6.2 General nonlinear eigenvalue problems

The following two problems are non-quadratic nonlinear eigenvalue problems. Thus the projected problems are solved by the safeguarded iteration (Algorithm 2). For a general nonlinear function, we do not have an explicit formula for its zeros (and hence for the Rayleigh functional) as it was the case for the quadratic eigenvalue problem but we have to revert to Newton iteration.

In both cases the method was initialized using the lower bound of the interval. The relative residual tolerance was $10^{-6}$, maximal subspace dimension 80, the slowest admissible convergence rate $\tau = 0.5$ and the automatic restart parameters $\alpha = 1$ and $N_\alpha = 1$. We report average performance values over 10 runs.

### *6.2.1 Delay exponential NEP*

We consider the following delay differential equation on a square domain [13]

$$u_t(\xi, t) = \Delta u(\xi, t) + a(\xi)u(\xi, t) - b(\xi)u(\xi, t-2), \quad \xi \in \Omega := [0, \pi]^2, \quad t \geq 0$$

with Dirichlet boundary conditions $u(\xi, t) = 0$, $\xi \in \partial\Omega$, $t \geq 0$ and $a(\xi) = 8\sin(\xi_1)\sin(\xi_2)$ and $b(\xi) = 100|\sin(\xi_1 + \xi_2)|$. Using the ansatz $u(\xi, t) = e^{\lambda t}v(\xi)$ and discretizing the Laplace operator on a uniform grid with step size $\pi/200$ by the 5-point stencil finite difference approximation, we obtain the nonlinear eigenvalue problem

$$T(\lambda)x = \lambda x + Ax + e^{-2\lambda}Bx = 0$$

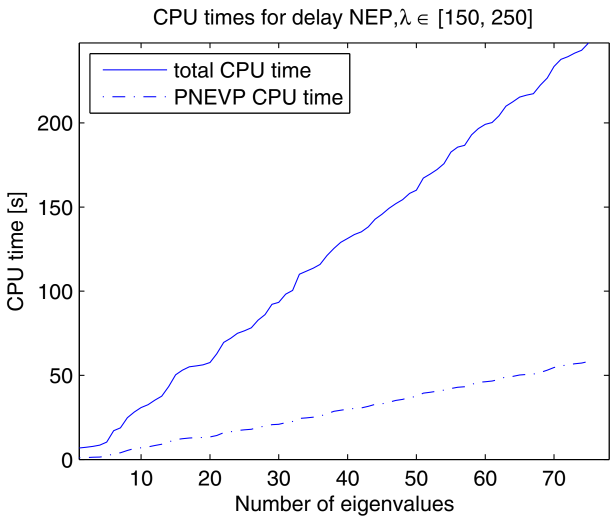

**Fig. 13** CPU time of NRA for computation of eigenvalues in [150, 250] of the exponential delay problem of dimension 39,601

of dimension 39601. $B$ is a diagonal matrix corresponding to values of the function $b(\xi_1, \xi_2)$ and $A$ is the negative sum of a diagonal matrix with entries corresponding to values of the function $a(\xi_1, \xi_2)$ and the 2-D discrete Laplacian.

Due to the symmetry of the problem in $\xi_1, \xi_2$ (Laplacian is symmetric on a square domain $\Omega$ and $a(\xi_1, \xi_2) = a(\xi_2, \xi_1)$, $b(\xi_1, \xi_2) = b(\xi_2, \xi_1)$) the problem has double eigenvalues. To avoid missing out eigenpairs, at each restart we locked the preceding eigenvector along with the anchor in the search subspace, $n_{locked} = 1$. We computed all 75 eigenvalues in the interval $[150, 250]$. Figure 13 shows the linear dependence of the CPU times on the number of computed eigenvalues. On average NRA method took 485.1 iterations with 8.7 restarts (9.7 LU decompositions) over 247.5 s, 58.2 s of which were spent on solution of the projected problems, Table 1.

For problems with double or higher multiplicity eigenvalues, it may be beneficial to consider extension of the restart strategy to block versions of the nonlinear Arnoldi and Jacobi-Davidson methods. Block methods by design are well suited for problems with multiple or clustered eigenvalues, as an entire subspace is iterated simultaneously. In principle, the local restarts can be used within block methods, when we simply iterate a number of eigenpairs, $q$, with consecutive local numbers, say $k, k+1, \dots, k+q-1$ instead of one. At each iteration, all the spurious eigenvalues which disturb the local ordering in the entire interval $(\lambda_\ell, \lambda_{\ell+k+q-1}]$ would have to be removed which again could be done using block operations. A large number of numerical tests would be necessary to access whether such restarted block method has a significant advantage over the single vector version. A serious discussion of such extension is beyond the scope of the current paper.

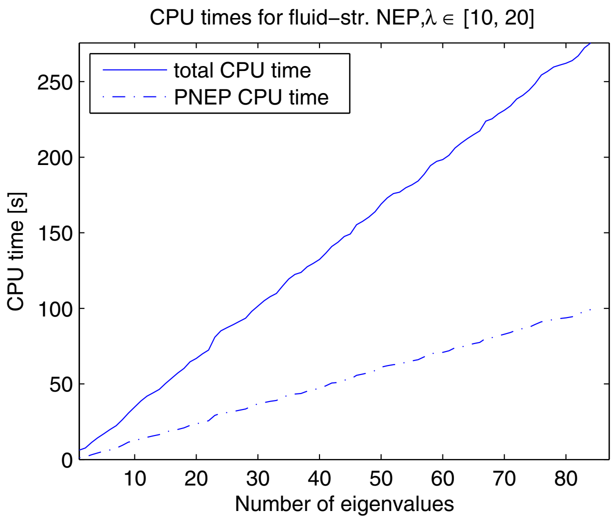

**Fig. 14** CPU time of NRA for computation of eigenvalues in [10, 20] of the rational fluid structure interaction problem of dimension 36,040

### *6.2.2 Fluid structure interaction rational NEP*

Our second problem is a rational eigenvalue problem

$$Kx = \lambda Mx + \sum_{j=1}^{k} \frac{\lambda}{\sigma_j - \lambda} C_j x, \tag{11}$$

where $K, M \in \mathbb{R}^{n\times n}$ are symmetric and positive definite, $C_j \in \mathbb{R}^{n\times n}$ are matrices of small rank $r_j$, and $0 < \sigma_1 < \sigma_2 < \cdots < \sigma_k$ are given poles. Problems of this type arise for example in free vibrations of tube bundles immersed in a slightly compressible fluid [8].

In each of the intervals $J_j := (\sigma_{j-1}, \sigma_j)$, $j = 1, \ldots, k+1$ with $\sigma_0 = 0, \sigma_{k+1} = \infty$, problem (11) satisfies the conditions of the minmax characterization and in each interval the eigenvalues have consecutive numbers [30].

The considered matrix problem is a finite element discretization of an elliptic cavity with 9 emerged tubes with 36040 degrees of freedom, it has 9 poles, $\sigma_j = j,\ j = 1, \ldots, 9$ and rank $C_j = 2,\ j = 1, \ldots, 9$ [3].

Using the search subspace with only the anchor locked i.e. $n_{locked} = 0$, we computed all 84 eigenvalues in the interval [10, 20]. Figure 14 shows the average total CPU time and CPU time for solution of the projected nonlinear problems with safeguarded iteration. The algorithm took on average 464 iterations with 11.6 restarts (12.6 LU decompositions) over average total CPU time of 275.6 s, 99.1 s of which were spent on solution of projected nonlinear problems.

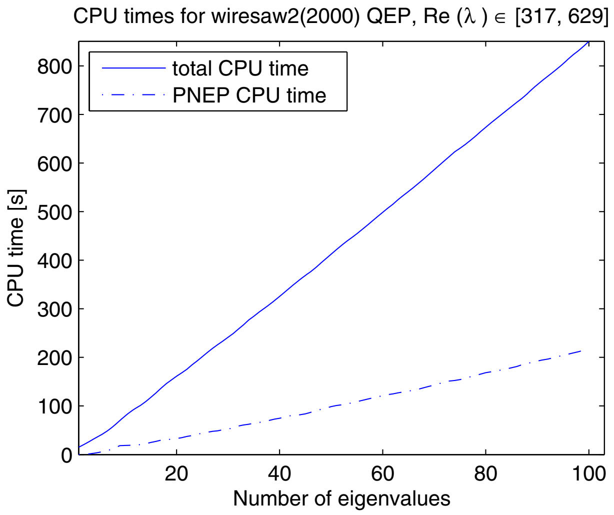

**Fig. 15** CPU time of NRA for computation of eigenvalues with real part in [317, 629] of the NLEVP "wiresaw2" problem of dimension 2000

## 6.3 Quadratic eigenvalue problems with eigenvalues with dominant real part

The described local restart procedure hinges upon the minmax property of the nonlinear eigenvalue problem. However, observe that if $(\lambda, x)$ is an eigenpair of the nonlinear problem $T(\lambda)x = 0$ with a complex eigenvalue, $\lambda \in \mathbb{C}$, it holds that $\mu = 0$ is an eigenvalue of the linear problem $T(\lambda)x = \mu x$ and also of $T_{\mathcal{V}}(\lambda)y = \mu y$ if $x \in \mathcal{V}$. As there is no natural order in the complex plain, in general we cannot infer the number of the eigenvalue. However, if the eigenvalues have a dominant real part $\Re(\lambda) \gg \Im(\lambda)$ (or equivalently up to a transformation dominant imaginary part), they can be ordered with respect to the dominant part. Furthermore, this ordering is inherited by the projected problem. If we can solve the projected nonlinear problem for the complex eigenpair $(\lambda, y)$ (e.g. well known issues with convergence of Newton method for complex zeros) we can proceed as in the real case but where the eigenvalues are enumerated w.r.t. the ascending real part. In particular in the polynomial case, the projected polynomial problems can be effectively solved by linearization.

We used such ordering to solve two quadratic eigenvalue problems from the NLEVP collection [5] which eigenvalues have a dominant either real or imaginary part. The projected problems were solved using linerization and the method was initialized using the lower bound of the interval containing the dominant part of the wanted eigenvalues.

### *6.3.1 NLEVP "wiresaw2" QEP*

We consider a quadratic eigenvalue problem arising from the vibration analysis of a wiresaw including the effect of viscous damping, "wiresaw2" problem from NLEVP collection [6]

$$T(\lambda)x = \lambda^2 Mx - i\lambda Cx - Kx.$$

We chose the dimension of 2000 and NLEVP default values of the wire speed and damping parameters, $v = 0.01$ and $\eta = 0.8$, respectively. For this problem both the matrices $C$ and $K$ are not sparse, hence the relatively small problem dimension. The real parts of the eigenvalues are approximately equal to the corresponding eigenvalues of the "wiresaw1" problem with the same dimension and value of the parameter $v$, and the imaginary part of all eigenvalues is a constant equal to 0.8.

As for "wiresaw1" problem we computed all 100 eigenvalues with the real part in the interval [317, 629] using the same initialization and solver parameters. Figure 15 shows the total CPU time and the CPU time for solution of the projected quadratic problems. While the solution for each complex eigenvalue takes longer than for the corresponding eigenvalue of the real problem, the qualitative property that the method needs approximately equal CPU time per eigenvalue regardless of its location in the spectrum is preserved. On average the solver took 1060 iterations in 851 s, 217 s of which were spent solving projected quadratic problems with 11.1 restarts corresponding to 12.1 LU factorizations.

### *6.3.2 NLEVP "acoustic wave 1D" QEP*

We consider a quadratic eigenvalue problem

$$T(\lambda)x = \lambda^2 Mx + \lambda Cx + Kx \tag{12}$$

arising from a finite element discretization of a 1D acoustic wave equation with mixed Dirichlet and impedance boundary conditions, "acoustic wave 1d" problem from NLEVP [6]. The matrices $K$, $M$ are real symmetric and $C$ is a low rank complex diagonal matrix dependent on the impedance parameter. For the formulation (12) all the eigenvalues lie in the upper half of the complex plane and have a dominant real part.

Using NLEVP default value of the impedance parameter $\zeta = 1$ we generated a matrix problem of dimension 30,000. We computed all 100 eigenvalues with the real part in the interval [0, 50] (see Fig. 16). The relative residual tolerance was $10^{-6}$, the maximal subspace dimension 120, the slowest admissible convergence rate $\tau = 0.5$, $n_{locked} = 0$ and the automated restart parameters $\alpha = 1$ and $N_\alpha = 1$. Figure 17 shows the total CPU time and the time for solving of the projected quadratic eigenvalue problems. On average NRA method took 618.1 iterations and 11.1 restarts (12.1 LU factorizations) in 219.5 s, 67.4 s of which were spent on the solution of the projected linearized problems.

## 7 Conclusions

We presented a local restart technique for iterative projection methods for solution of nonlinear Hermitian eigenvalue problems admitting a minmax characterization of their eigenvalues. We showed how the proposed technique can effectively eliminate a

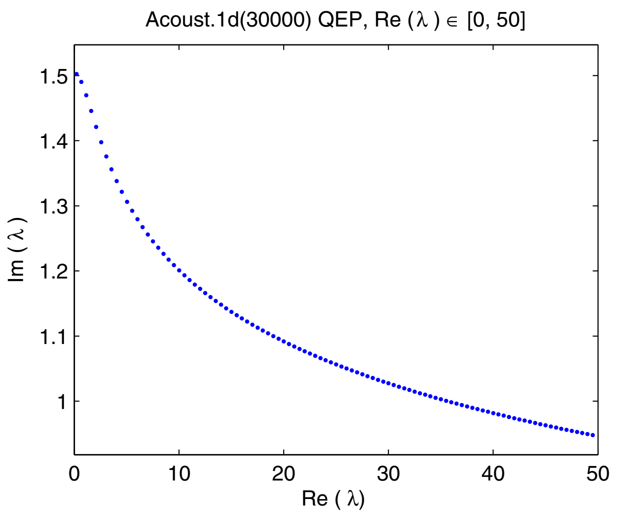

**Fig. 16** Eigenvalues with the real part in [0, 50] of the NLEVP "acoustic wave 1d" quadratic eigenvalue problem of dimension 30,000

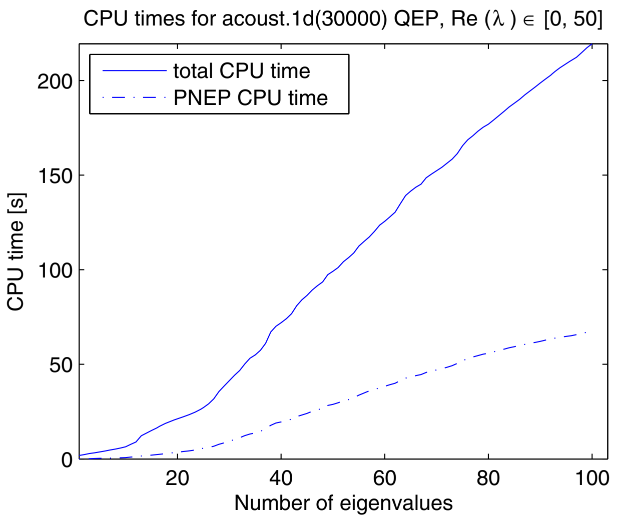

**Fig. 17** CPU time of NRA for computation of eigenvalues with real part in [0, 50] of the NLEVP "acoustic wave 1d" quadratic eigenvalue problem of dimension 30,000

super-linear search subspace growth experienced when computing a large number of eigenvalues. Properly initialized, the method can be employed for computing eigenvalues in the interior of the spectrum. Iterative projection methods here considered work directly on the nonlinear eigenvalue problem without increasing its size and possibly destroying its structure by prior linearization. In this setting we do not have a transformation, like shift-invert for linear problems, mapping the eigenvalues close to a chosen shift to the exterior of the spectrum. In the absence of such transformation, spurious eigenvalues are intrinsic to interior eigenvalue computations and we proposed an effective strategy for dealing with such spurious values. We incorporated the proposed technique in the nonlinear iterative projection methods like the Nonlinear Arnoldi and Jacobi-Davidson methods. We illustrated various aspects of the local restart technique on numerical examples. The efficiency of the new restart framework was demonstrated on a range of nonlinear eigenvalue problems: three gyroscopic problems including a large gyroscopic eigenvalue problem modeling the dynamic behavior of a rotating tire, one exponential and one rational eigenvalue problem. Furthermore, we showed on two quadratic eigenvalue problems how the local restart technique can be extended to problems with complex eigenvalues with a dominant part (either real or imaginary). All the examples in this paper were solved using MATLAB toolbox QARPACK [2] containing an exemplary implementation of the locally restarted iterative methods (`qra`: quadratic, `nra`: general nonlinear solver). In the future we intend to extend the local restart technique to problems with more general distributions of the eigenvalues in the complex plane, close to a line or an a priori known curve.

**Acknowledgments** The authors would like to thank the anonymous referees for their invaluable help in improving the manuscript. Marta M. Betcke also acknowledges the financial support through EPSRC Postdoctoral Fellowship (Grant Number EP/H02865X/1).

## References

1. Betcke, M.M.: Iterative projection methods for symmetric nonlinear eigenvalue problems with applications. Ph.D. thesis, Institute of Numerical Simulation, Hamburg University of Technology (2007)
2. Betcke, M.M.: QARPACK: Quadratic Arnoldi Package. University College London (2011). http://www.cs.ucl.ac.uk/staff/M.Betcke/codes/qarpack
3. Betcke, M.M., Voss, H.: Restarting projection methods for rational eigenproblems arising in fluid-solid vibrations. Math. Model. Anal. **13**, 171–182 (2008)
4. Betcke, T., Voss, H.: A Jacobi-Davidson-type projection method for nonlinear eigenvalue problems. Future Gen. Comput. Syst. **20**(3), 363–372 (2004)
5. Betcke, T., Higham, N.J., Mehrmann, V., Schröder, C., Tisseur, F.: NLEVP: A collection of nonlinear eigenvalue problems. http://www.mims.manchester.ac.uk/research/numerical-analysis/nlevp.html
6. Betcke, T., Higham, N.J., Mehrmann, V., Schröder, C., Tisseur, F.: NLEVP: A collection of nonlinear eigenvalue problems. ACM Trans. Math. Softw. **39**(2), 7:1–7:28 (2013)
7. Chou, S.-H., Huang, T.-M., Huang, W.-Q., Lini, W.-W.: Efficient Arnoldi-type algorithms for rational eigenvalue problems arising in fluid-solid systems. J. Comput. Phys. **230**(5), 2189–2206 (2011)
8. Conca, C., Planchard, J., Vanninathan, M.: Fluid and periodic structures. research in applied mathematics, vol. 24. Masson, Paris (1995)
9. Duffin, R.J.: A minimax theory for overdamped networks. J. Rat. Mech. Anal. **4**, 221–233 (1955)
10. Duffin, R.J.: The Rayleigh-Ritz method for dissipative and gyroscopic systems. Q. Appl. Math. **18**, 215–221 (1960)
11. Effenberger, C.: Robust successive computation of eigenpairs for nonlinear eigenvalue problems. SIAM J. Matrix Anal. Appl. **34**(3), 1231–1256 (2013)
12. Fokkema, D.R., Sleijpen, G.L.G., van der Vorst, H.A.: Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils. SIAM J. Sci. Comput. **20**, 94–125 (1998)
13. Jarlebring, E.: The Spectrum of delay-differential equations: numerical methods, stability and perturbation. Ph.D. thesis, Carl-Friedrich-Gauß-Fakultät, TU Braunschweig (2008)

14. 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)
15. Jarlebring, E., Voss, H.: Rational Krylov for nonlinear eigenproblems, an iterative projection method. Appl. Math. **50**, 543–554 (2005)
16. Jarlebring, E., Michiels, W., Meerbergen, K.: A linear eigenvalue algorithm for the nonlinear eigenvalue problem. Numer. Math. **122**(1), 169–195 (2012)
17. Liao, B.-S., Bai, Z., Lee, L.-Q., Ko, K.: Nonlinear Rayleigh-Ritz iterative methods for solving large scale nonlinear eigenvalue problems. Taiwan. J. Math. **14**, 869 (2010)
18. Markiewicz, M., Voss, H.: A local restart procedure for iterative projection methods for nonlinear symmetric eigenproblems. In: Handlovicova, A., Kriva, Z., Mikula, K., Sevcovic, D. (eds.) Algoritmy 2005, 17th Conference on Scientific Computing, Vysoke Tatry—Podbanske, Slovakia 2005, pp. 212–221. Slovak University of Technology, Bratislava, Slovakia (2005)
19. Meerbergen, K.: Locking and restarting quadratic eigenvalue solvers. SIAM J. Sci. Comput. **22**, 1814–1839 (2001)
20. Meerbergen, K.: The Quadratic Arnoldi method for the solution of the quadratic eigenvalue problem. SIAM. J. Matrix Anal. Appl. **30**, 1463–1482 (2008)
21. Meerbergen, K., Schröder, C., Voss, H.: A Jacobi-Davidson method for two-real-parameter nonlinear eigenvalue problems arising from delay-differential equations. Numer. Linear Algebra Appl. **20**(5), 852–868 (2013)
22. Neumaier, A.: Residual inverse iteration for the nonlinear eigenvalue problem. SIAM J. Numer. Anal. **22**, 914–923 (1985)
23. Niendorf, V., Voss, H.: Detecting hyperbolic and definite matrix polynomials. Linear Algebra Appl. **432**, 1017–1035 (2010)
24. Rogers, E.H.: A minimax theory for overdamped systems. Arch. Ration. Mech. Anal. **16**, 89–96 (1964)
25. Ruhe, A.: Rational Krylov for large nonlinear eigenproblems. In: Dongarra, J., Madsen, K., Wasniewski, J. (eds.) Applied Parallel Computing. State of the Art in Scientific Computing, vol. 3732 of Lecture Notes on Computer Science, pp. 357–363. Springer, Berlin (2006)
26. Schreiber, K.: Nonlinear Eigenvalue Problems: Newton-type Methods and Nonlinear Rayleigh Functionals. Ph.D. thesis, Technische Universität Berlin (2008)
27. Szyld, D., Xue, F.: Local convergence analysis of several inexact Newton-type algorithms for general nonlinear eigenvalue problems. Numer. Math. **123**, 333–362 (2013)
28. Szyld, D.B., Xue, F.: Several properties of invariant pairs of nonlinear algebraic eigenvalue problems. IMA J. Numer. Anal. (2013)
29. Voss, H.: An Arnoldi method for nonlinear symmetric eigenvalue problems. In: Online Proceedings of the SIAM Conference on Applied Linear Algebra. Williamsburg (2003). http://www.siam.org/meetings/la03/proceedings/VossH.pdf
30. Voss, H.: Initializing iterative projection methods for rational symmetric eigenproblems. In: Online Proceedings of the Dagstuhl Seminar Theoretical and Computational Aspects of Matrix Algorithms, Schloss Dagstuhl (2003). ftp://ftp.dagstuhl.de/pub/Proceedings/03/03421/03421.VoszHeinrich.Other.pdf
31. Voss, H.: An Arnoldi method for nonlinear eigenvalue problems. BIT Numer. Math. **44**, 387–401 (2004)
32. Voss, H.: A Jacobi-Davidson method for nonlinear and nonsymmetric eigenproblems. Comput. Struct. **85**, 1284–1292 (2007)
33. Voss, H.: A minmax principle for nonlinear eigenproblems depending continuously on the eigenparameter. Numer. Lin. Algebra Appl. **16**, 899–913 (2009)
34. Voss, H.: Iterative projection methods for large-scale nonlinear eigenvalue problems. In: Topping, B.H.V., Adams, J.M., Pallarés, F.J., Bru, R., Romero, M.L. (eds.) Computational Technology Reviews, vol. 1, pp. 187–214. Saxe-Coburg Publications, Stirlingshere (2010)
35. Voss, H., Werner, B.: A minimax principle for nonlinear eigenvalue problems with applications to nonoverdamped systems. Math. Meth. Appl. Sci. **4**, 415–424 (1982)