COMPUTING INTERIOR EIGENVALUES OF NONLINEAR HERMITEAN EIGENVALUE PROBLEMS

MARTA M. BETCKE*, VERA LOCHMANN†, AND HEINRICH VOSS‡

Key words. Nonlinear eigenvalue problem, iterative projection method, Arnoldi method, min-max characterization, local restart

AMS subject classification. 65F15, 15A18, 35P30, 49R50, 65N25

Abstract. A nonlinear eigenvalue problem $T(\lambda)x = 0$, the eigenvalues of which satisfy a min-max characterization shares many valuable properties of linear Hermitean eigenvalue problems. For instance, its eigenvalues can be computed safely one after another by means of iterative projection methods, where the projected problems are solved by the safeguarded iteration. Though, such methods hit their limitations if a large number of eigenvalues possibly in the interior of the spectrum is required. In this paper we propose a localized version of safeguarded iteration which overcomes the problem of growing search subspace dimension by means of a local numbering of the eigenvalues. The efficiency of the new method is demonstrated on a real-life gyroscopic eigenvalue problem modeling free vibrations of a rolling tire.

1. Introduction. Acoustic simulations are increasingly gaining importance in the automotive design process. Usually, such simulations are a part of a larger optimization procedure with the objective of minimizing the noise exposure to the passengers as well as to the environment. The major effort associated with optimization of a noise and vibration performance is the frequency response computation. This is not only because the underlying finite element models are of formidable size but also due to the fact that in general a very large number of modes (not necessarily at the end of the spectrum) is required to obtain a satisfactory accuracy over the relevant frequency range.

For instance, according to Nackenhorst and von Estorff [9, 10] the major cause of noise originating from vehicles at cruising speeds (passenger cars: from 40km/h and trucks: from 60 km/h) is the sound radiation of their rolling tires. As described in [10] the simulation of the tire noise is performed in three steps. First, the nonlinear tire deflections under steady state conditions are computed using an Arbitrary Lagrangian Eulerian (ALE) approach. Next, the transient vibrations governed by the eigenpairs of a gyroscopic eigenvalue problem

$$Kx + i\lambda Gx - \lambda^2 Mx = 0 \tag{1.1}$$

are assumed to be superimposed onto the nonlinear deflections. Finally, the acoustic analysis is carried out solving Helmholtz's equation where the normal velocities at the wheel surface, extracted from the vibration analysis, are taken as boundary conditions.

More generally than the quadratic problem (1.1) we consider the nonlinear eigenvalue problem

$$T(\lambda)x = 0, \tag{1.2}$$

*née Markiewicz, School of Mathematics, The University of Manchester, Oxford Road, M13 9PL, Manchester, United Kingdom (m.betcke@maths.manchester.ac.uk).

†Institute of Numerical Simulation, Hamburg University of Technology, D-21071 Hamburg, Germany (lochmann@tu-harburg.de).

‡Institute of Numerical Simulation, Hamburg University of Technology, D-21071 Hamburg, Germany (voss@tu-harburg.de).

[1]A preliminary version of this paper appeared in the proceedings of ALGORITMY 2005 [8].

where $T(\lambda) \in \mathbb{C}^{n\times n}$ is a family of large and sparse Hermitean matrices for every λ in an open real interval J, and we assume that its eigenvalues can be characterized as minmax values of a Rayleigh functional [19]. For this type of problems a combination of iterative projection methods with safeguarded iteration was discussed in [2] and [15] to determine a moderate number of consecutive eigenvalues in J. For more general nonlinear eigenproblems iterative projection methods were considered in [1, 6, 7, 13, 14, 16, 17, 18].

However, the approach in [2, 15] hits it limitations if a large number of eigenvalues (in particular in the interior of the spectrum) of (1.2) is needed. In this case one has to project the problem under consideration onto a sequence of search spaces of growing dimensions. For a large number of eigenvalues 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 problem (1.2) only to search spaces of a limited dimension throughout the iteration. Our presentation is restricted to the Arnoldi method, but the local restart technique applies to any other iterative projection method that preserves the variational characterization of eigenvalues.

The paper is organized as follows. Section 2 outlines the variational characterization of eigenvalues for nonlinear and nonoverdamped eigenproblems, which naturally suggests the safeguarded iteration method for solving such problems. Section 3 briefly recalls the Arnoldi method for sparse Hermitean nonlinear eigenproblems. The new restart technique is presented in Section 4 along with the strategy for dealing with the spurious solutions which are an intrinsic part of the interior eigenvalue computation. An example of a large gyroscopic eigenproblem modelling free vibrations of a rotating tire in Section 5 demonstrates the efficiency of the new restart method.

2. Variational characterization of eigenvalues. We consider a nonlinear eigenvalue problem

$$T(\lambda)x = 0, \tag{2.1}$$

where $T(\lambda) \in \mathbb{C}^{n\times n}$ is a family of Hermitean matrices for every λ 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 (2.1) has a nontrivial solution $x \neq 0$, which we call an eigenvector corresponding to λ.

It is well known that all the eigenvalues of a linear Hermitean problem $Ax = \lambda x$ are real. Moreover, if they are ordered by magnitude,

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n,$$

it is possible to characterize them by the minmax principle of Poincaré.

It turns out, that a similar result holds also for a certain type of nonlinear eigenvalue problems. In the following we assume that, for every fixed $x \neq 0$, the real function $f(\lambda; x) := x^H T(\lambda)x$ is continuously differentiable in J. Suppose, that the equation

$$f(\lambda; x) = 0 \tag{2.2}$$

has at most one solution in J, then (2.2) implicitly defines a functional p on some subset D of $\mathbb{C}^n \setminus \{0\}$. 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 (2.1). We furthermore require that

$$\frac{\partial}{\partial\lambda} f(\lambda; x)\Big|_{\lambda=p(x)} > 0 \quad \text{for every } x \in D,$$

which is a natural generalization of the definiteness assumption for the linear pencils.

Under these assumptions a variational characterization in terms of the Rayleigh functional has been considered by various authors. To mention a few Duffin [3, 4] and Rogers [12] 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 $\mathbb{C}^n \setminus \{0\}$. Nonoverdamped problems were considered by Werner and the third author [19].

The key to the variational principle in [19] 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. Instead, the number of an eigenvalue λ of the nonlinear problem (2.1) is inherited from the number of the eigenvalue 0 of the matrix $T(\lambda)$ based on the following consideration:

Let $\lambda \in J$ be an eigenvalue of the nonlinear problem (2.1), then $\mu = 0$ is an eigenvalue of the linear problem $T(\lambda)y = \mu y$. Therefore there exists $k \in \mathbb{N}$ such that

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

where S_k denotes the set of all k dimensional subspaces of $\mathbb{C}^n$ and $\mathcal{W}^1 := \{w \in \mathcal{W} : \|w\| = 1\}$ is the unit sphere in $\mathcal{W}$. In this case we call λ a kth eigenvalue of (2.1).

With this enumeration the following minmax characterization of the eigenvalues of the nonlinear eigenproblem (2.1) was proved in [19]:

THEOREM 2.1. *For every $x \neq 0$ let the real equation* (2.2) *have at most one solution $p(x) \in J$, and assume that*

$$\frac{\partial}{\partial \lambda} f(\lambda; x)\bigg|_{\lambda = p(x)} > 0 \quad \textit{for every } x \in D.$$

Then the following assertions hold:

(i) For every $k \in \mathbb{N}$ there is at most one kth eigenvalue of problem (2.1) *which can be characterized by*

$$\lambda_k = \min_{\substack{\mathcal{W} \in S_k, \\ \mathcal{W} \cap D \neq \emptyset}} \sup_{w \in \mathcal{W} \cap D} p(w). \tag{2.3}$$

Hence, there are at most n eigenvalues of (2.1) *in J.*

(ii) If $\lambda \in J$ and $k \in \mathbb{N}$ such that (2.1) *has a kth eigenvalue $\lambda_k \in J$. Then it holds that*

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

The correspondence between the kth eigenvalue λ_k of $T(\cdot)$ and the k largest eigenvalue of the matrix $T(\lambda_k)$ suggests the safeguarded iteration for computing the kth eigenvalue of a nonlinear problem. An outline of the safeguarded iteration is given in Algorithm 1. Its convergence properties were proved in [18], and are recapitulated in Theorem 2.2.

THEOREM 2.2.

(i) If $\lambda_1 := \inf_{x \in D} p(x) \in J$, then the safeguarded iteration converges globally to λ_1.

(ii) If $\lambda_k \in J$, $k \geq 1$ is a kth eigenvalue of (2.1) *which is simple then the safeguarded iteration converges locally and quadratically to λ_k.*

Algorithm 1 Safeguarded iteration

1: Start with an approximation μ_1 to the kth eigenvalue of (2.1)
2: **for** $\ell = 1, 2, \ldots$ until convergence **do**
3: Determine eigenvector x corresponding to the k largest eigenvalue of $T(\mu_\ell)$
4: Evaluate $\mu_{\ell+1} = p(x)$
5: **end for**

(iii) If $T(\lambda)$ is positive definite for $\lambda \in J$ and x in step 3. of Algorithm 1 is chosen to be an eigenvector corresponding to the k largest eigenvalue of the generalized eigenproblem $T(\mu_\ell)x = \kappa T'(\mu_\ell)x$ then the convergence is even cubic.

Certainly, the safeguarded iteration, is not capable of solving large nonlinear eigenvalue problems. However, it is well suited as an inner iteration in a projection method since its convergence properties and for small dimension also its complexity are comparable to those of the inverse iteration. Moreover, it has a significant advantage over the inverse iteration, namely it aims at an eigenvalue with a specific number. As a consequence, it is less likely to miss an eigenvalue, which is desirable whenever the knowledge of all the eigenvalues in an interval is relevant for the underlying application.

3. Iterative projection methods for nonlinear eigenproblems. For sparse linear eigenvalue problems

$$Ax = \lambda x \tag{3.1}$$

iterative projection methods are acknowledged to be a very efficient tool. Here, the dimension of the eigenproblem is reduced by projecting it to a subspace of a much smaller dimension. The reduced problem is then handled by a fast technique for dense problems. The search subspace is expanded in the course of the algorithm in an iterative way with the objective of some of the eigenvalues of the reduced matrix becoming good approximations to some of the wanted eigenvalues of the given large matrix. Prominent representatives of this type are the Lanczos, Arnoldi, rational Krylov, and Jacobi-Davidson methods.

Generalizations to nonlinear eigenproblems are discussed in [1, 2, 6, 7, 13, 14, 15, 16, 17, 18]. A representative example is the nonlinear Arnoldi method in Algorithm 2 where we assume that the problem (1.2) is Hermitean and its eigenvalues can be enumerated according to Section 2 and furthermore we are interested in computing the eigenvalues $\lambda_{m_{\min}}, \ldots, \lambda_{m_{\max}}$.

Applying Algorithm 2 to the linear eigenproblem $T(\lambda) = \lambda B - A$ and choosing the preconditioner $M = (\sigma B - A)^{-1}$ the method is nothing else but the shift-and-invert Arnoldi method. This motivates the name nonlinear Arnoldi method despite the fact that differently from the linear case no Krylov space is generated and no Arnoldi recursion holds in the course of the algorithm. Similarly as in the Jacobi-Davidson method for linear problems the underlying idea is to expand the search space by a direction which has a high approximation potential for the eigenvector wanted next, namely the improvement by the residual inverse iteration [11].

There are many details that have to be considered when implementing the Arnoldi method according to Algorithm 2 concerning the choice of the initial basis, solving the projected problem, when to change and how to choose the preconditioner, when

Algorithm 2 Nonlinear Arnoldi Method

1: Start with an initial shift σ and an initial basis V, $V^H V = I$; set $m = m_{\min}$
2: Determine preconditioner $M \approx T(\sigma)^{-1}$, σ close to first wanted eigenvalue
3: **while** $m \leq m_{\max}$ **do**
4: Compute the m smallest eigenvalue μ and corresponding eigenvector y of the projected problem $T_V(\mu)y := V^H T(\mu)Vy = 0$ by safeguarded iteration
5: Determine Ritz vector $u = Vy$ and residual $r = T(\mu)u$
6: **if** $\|r\|/\|u\| < \epsilon$ **then**
7: Accept approximate eigenpair $\lambda_m = \mu$, $x_m = u$; increase $m \leftarrow m + 1$
8: Choose new shift σ and determine preconditioner $M \approx T(\sigma)^{-1}$ if indicated
9: Restart if necessary
10: Choose approximations μ and u to next eigenvalue and eigenvector
11: Determine residual $r = T(\mu)u$
12: **end if**
13: $v = Mr$
14: $v = v - VV^H v$, $\tilde{v} = v/\|v\|$, $V = [V, \tilde{v}]$
15: Reorthogonalize if necessary
16: Update projected problem $T_V(\mu) = V^H T(\mu)V$
17: **end while**

and how to restart, and how to continue after an eigenpair was accepted. A detailed discussion is contained in [15, 16]. Here we concentrate on the initialization of the algorithm and the restart technique for Hermitean problems allowing a minmax characterization of their eigenvalues.

While applying the iterative projection methods to general nonlinear eigenvalue problems with the objective to approximate more than one eigenvalue, it is crucial to prevent the method from converging to the same eigenvalue repeatedly. In the linear case this is readily done by the Krylov subspace solvers. The latter construct an orthogonal basis of the ansatz space not aiming at a particular eigenvalue. As a result one gets approximations to extreme eigenvalues without replication (at least after the reorthogonalization is employed). If several eigenvalues are to be computed by the Jacobi-Davidson method, an incomplete Schur factorization is determined in order to prevent the method from reapproaching an already previously computed eigenvalue (cf. [5]). Unfortunately, a similar normal form does not exist for nonlinear eigenvalue problems.

Obviously, to successfully apply the safeguarded iteration as an inner iteration in an iterative projection method, the projected problems have to allow a variational characterization of their eigenvalues as well. Let $T(\lambda)$ be a family of Hermitean matrices allowing a minmax characterization of their eigenvalues in an open interval J and let the columns of $V \in \mathbb{C}^n$ form a basis of the current search space $\mathcal{V}$ of $\mathbb{C}^n$, then it is easily seen that the projected problem

$$T_V(\lambda)y := V^H T(\lambda)Vy = 0 \tag{3.2}$$

inherits this 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 numeration of the eigenvalues of the original problem and the projected problem may differ.

If J contains the first eigenvalue $\lambda_1 = \min_{x \in D} \; p(x)$, then by Theorem 2.2 the safeguarded iteration for (3.2) converges globally, i.e. for any initial vector $x \in \mathcal{V} \cap D$,

to the smallest eigenvalue of (3.2). If x_j denotes an eigenvector corresponding to the jth eigenvalue λ_j of (1.2), and if $x_j \in \mathcal{V}$ for $j = 1, \ldots, k$, then λ_j is a jth eigenvalue of the projected problem (3.2), as well. Hence, expanding the search space $\mathcal{V}$ iteratively, and determining the $(k+1)$th eigenvalue of the projected problems, one gets a sequence of upper bounds of λ_{k+1} which (hopefully) converges to λ_{k+1}. Thus, the eigenvalues of (1.2) can be determined quite safely one after the other by the Nonlinear Arnoldi algorithm starting with an approximation to x_1.

As the subspace expands in the course of the algorithm the increasing storage or the computational cost for solving the projected eigenvalue problems may make it necessary to restart the algorithm and purge some of the basis vectors. Restarting with a subspace $\mathcal{V}$ which contains the already converged eigenvectors $x_1, \ldots, x_k$ then obviously keeps the enumeration of the eigenvalues, and we can continue as above to determine the subsequent eigenpairs. Notice that we only restart if an eigenvector has just converged since a restart destroys information on the eigenvectors and particularly on that one the method is just aiming at.

If $\lambda_1 = \inf_{x \in D} \; p(x) \notin J$ we can modify this approach in the following way. The proof of the minmax characterization (2.3) in [19] shows that the minimum 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 the kth eigenvalue of the projected problem (3.2) is λ_k, i.e. again the numeration of the eigenvalues is not altered in the projected problem, and the eigenvalues can be determined successively.

4. A local restart technique. The iterative projection methods described in the last section hit their limitations if a large number of eigenvalues or a set of some subsequent eigenvalues in the interior of the spectrum is required. In order to preserve the numbering the dimension of the search space has to be at least as large as the number of eigenvalues in J preceding the sought one. Therefore the size of the projected problem is growing with the number of the wanted eigenvalues, which results in increasing time consumed by the nonlinear solver and increasing storage requirements.

To overcome these difficulties we propose a local numbering, which does not require to include the entire set of preceding eigenvectors or the invariant subspace of $T(\lambda_k)$ mentioned in the last paragraph of Section 3 into the search subspace after a restart.

Assume that we are given an eigenvalue $\hat{\lambda} \in J$ of the nonlinear eigenproblem (1.2), which we call an anchor, and the corresponding eigenvector $\hat{x}$. 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 $\hat{\lambda}$ is also an eigenvalue of the projected problem

$$T_V(\lambda) := V^H T(\lambda) V y = 0, \tag{4.1}$$

and since $T_V(\cdot)$ satisfies the conditions of Theorem 2.1 we can assign to $\hat{\lambda}$ a local number $\ell = \ell(\mathcal{V})$ in the following way: $\hat{\lambda}$ is an ℓth eigenvalue of problem (4.1) if $\mu(\hat{\lambda}) = 0$ is the ℓ largest eigenvalue of the linear problem

$$V^H T(\hat{\lambda}) V y = \mu(\hat{\lambda}) y. \tag{4.2}$$

Starting with $\mathcal{V} =: \mathcal{V}_0$ we determine approximations to the eigenvalue subsequent to the anchor $\hat{\lambda}$ projecting problem (1.2) to a sequence of subspaces $\mathcal{V}_0 \subset \mathcal{V}_1 \subset \mathcal{V}_2 \subset \ldots$ which are generated in course of an iterative projection method aiming at the

Algorithm 3 Restart framework

Require: Preconditioner $M \approx T(\sigma)^{-1}$ for a suitable pole σ,
Require: (λ_i, x_i) an (approximate) eigenpair of $T(\cdot)$
Require: v_1 an approximation to x_{i+1}
$V = [x_i, v_1]$;
j = 1;
while Restart condition not satisfied **do**
  repeat
    Determine largest eigenvalues $\mu_1(\lambda_i) \geq \cdots \geq \mu_k(\lambda_i) > 0 \geq \mu_{k+1}(\lambda_i)$ of (4.2)
    Set $\ell := k$ if $\mu_k \leq -\mu_{k+1}$, and else $\ell := k+1$
    Compute $(\ell + j)$th eigenpair $(\tilde{\lambda}_{\ell+j}, y_{\ell+j})$ of $T_V(\cdot)$
    Expand V aiming at $(\lambda_{\ell+j}, x_{\ell+j})$
  until Eigenpair $(\tilde{\lambda}_{\ell+j}, Vy_{l+j}) =: (\lambda_{i+j}, x_{i+j})$ converged
  j = j+1;
end while

$(\ell(\mathcal{V}_k) + 1)$th eigenvalue in the kth iteration step. Explicitly stating the dependence of ℓ on $\mathcal{V}_k$ we emphasize that the number $\ell(\mathcal{V}_k)$ of the anchor may change in the course of the iteration.

After convergence we may continue the iterative projection method aiming at the $(\ell(\mathcal{V}_k) + 2)$th eigenvalue or we may replace the anchor by the newly converged eigenpair. Since the current search space contains useful information about further eigenvalues it is advisable to continue expanding the search space until the convergence has become too slow or the dimension exceeds a given bound.

Once we have the local numbering there is no necessity any more to include all the eigenvectors corresponding to the preceding eigenvalues in J or the invariant subspace of $T(\hat{\lambda})$ corresponding to its nonnegative eigenvalues into the search space after a restart. All that we need to set up the new search subspace is an eigenvector $\hat{x}$ corresponding to an anchor $\hat{\lambda}$ and an approximation v_1 to the next eigenvector (or a random vector if such an approximation is not at hand). This leads to the restart framework in Algorithm 3.

In the course of the computation it may happen that the algorithm converges to an eigenvalue twice, i.e. it returns $\lambda_i < \lambda_{i+1} < \cdots < \lambda_{i+k} \approx \lambda_{i+k+1}$ for some $k \geq 1$. Then two possibilities arise:

- The eigenvalue is a multiple (at least double) eigenvalue;
- The algorithm converged to a single eigenvalue twice.

Which one applies can be checked as follows. If the angle between the eigenvectors x_{i+k} and x_{i+k+1} is different from 0 (i.e. in the numerical reality larger than a prescribed small threshold) or if λ_{i+k} is the $(\ell+k)$th eigenvalue of the projected problem

$$\tilde{V}^H T(\lambda)\tilde{V}y = 0,$$

where $\tilde{V}$ denotes a basis of the orthogonal complement of x_{i+k+1} in $\mathcal{V}$, then λ_{i+k} is a multiple (at least a double) eigenvalue, and we continue Algorithm 3 to compute the $(i+k+2)$th eigenvalue.

However, if in this way λ_{i+k} is not shown to be a double eigenvalue, then for the current search space $\mathcal{V}$ the projected problem (4.1) possesses an additional eigenvalue $\theta \in (\lambda_i, \lambda_{i+k})$ such that $\theta \neq \lambda_{i+j}$ for $j = 0, \ldots, k$. Therefore the local number of λ_{i+k} is raised by 1, and λ_{i+k} is accepted as an $(i+k+1)$th eigenvalue. This may have

happened for one of the following two reasons:

- An eigenvalue of (1.2) in the interval $(\lambda_i, \lambda_{i+k})$ might have been missed out because the corresponding eigenvector $\tilde{x}$ was not sufficiently present in the initial search space $\operatorname{span}\{x_i, v_1\}$ and might not have been amplified sufficiently in the course of the expansions of $\mathcal{V}$ until computing λ_{i+k}. Afterwards the component of $\tilde{x}$ in the search space $\mathcal{V}$ has increased and has become large enough to produce the additional eigenvalue approximation $\theta \in (\lambda_i, \lambda_{i+k})$, and Algorithm 3 yields the eigenvalue approximation λ_{i+k} the second time with a different local number.
- It might be the case that no eigenvalue of (1.2) is missing in $(\lambda_i, \lambda_{i+k})$ but the newly produced eigenvalue of the projected problem (4.1) is a spurious one, i.e. a linear combination of eigenvectors of (1.2) corresponding to eigenvalues less than λ_i and of eigenvectors corresponding to eigenvalues greater than λ_{i+k}.

In both cases we determine the additional eigenvalue θ (henceforth we shall refer to such eigenvalues as "suspects") and its local number $\ell + j$, and we expand the search space $\hat{\mathcal{V}}$ aiming at (θ, x_θ), where x_θ denotes the Ritz vector corresponding to θ. Then by the minmax principle all eigenvalues of the projected problem

$$T_{\hat{V}}(\lambda)\hat{y} = 0 \tag{4.3}$$

are less than or equal to the corresponding ones of $T_V(\lambda)y = 0$, and either of the following happens:

- Problem (4.3) has exactly $k+1$ eigenvalues $\lambda_i, \ldots, \lambda_{i+k} \in [\lambda_i, \lambda_{i+k}]$ (i.e. the additional eigenvalue has left the interval of interest).
- There are still $k+2$ eigenvalues $\lambda_i, \ldots, \lambda_{i+k}, \hat{\theta} \in [\lambda_i, \lambda_{i+k}]$, and it holds that $\hat{\theta} \leq \theta$. In this case we repeat the expansion of the subspace until the sequence of additional eigenvalues has been moved out of the interval $[\lambda_i, \lambda_{i+k}]$ or has converged to a previously missed out regular eigenvalue. We then adjust the enumeration of the eigenvalues and continue the iterative projection method.

We would like to point out that if more than one additional eigenvalue exist in $[\lambda_i, \lambda_{i+k}]$ after we detected a replicate eigenvalue they all can be treated in the way described above one after the other.

So far, we have not commented on two important issues. First, how do we identify the suspect eigenvalues θ in the search subspace $\mathcal{V}_k$ and the second how do we decide, whether a suspect eigenvalue θ is a proper or a spurious one? In fact, the latter can be easily checked by computing the residual of the corresponding Ritz pair of the original problem (1.2). If it is small enough then the eigenvalue is genuine. Otherwise we proceed with the subspace expansion described above and if necessary test its residual again.

The identification of the suspect eigenvalue θ in the search subspace is also straight forward if the projected problem is solved by linearization. In this case we have all the eigenvalues of the projected nonlinear problem at hand and can simply compare them with already computed nonlinear eigenvalues. The nonmatching eigenvalues are suspect.

However, this is not possible if the inner solve is done by the safeguarded iteration when we only have the just computed eigenvalue λ_{i+k+1}. In this case we proceed as follows. We identify the numbers of the so far computed nonlinear eigenvalues $\lambda_{i+j},\ j = 1, \ldots, k$ in the search subspace $\check{\mathcal{V}}$, which is the subspace, for which the algorithm recorded the repeatable convergence to the eigenvalue λ_{i+k}. To this end for

each nonlinear eigenvalue $\lambda_{i+j},\ j = 1, \ldots, k$ we solve the projected linear eigenvalue problem $T_{\check{V}}(\lambda_{i+j})w = \mu w$ and assign λ_{i+j} the number of the zero eigenvalue of this linear problem. Since, there were $k+1$ eigenvalues of $T_{\check{V}}(\cdot)$ in the interval $(\lambda_i, \lambda_{i+k}]$ we are left with one not matching eigenvalue, which is exactly the suspect eigenvalue.

In practice the search subspace $\mathcal{V}$ usually contains approximations rather than the exact eigenvectors. Thus while identifying the anchor or the other already computed nonlinear eigenvalues we assign the number of the eigenvalue $\mu(\lambda_i)$ of the linear problem $T_V(\lambda_i)$ with minimal absolute value. Further, since we are only interested in identifying the eigenvalues in the interval $(\lambda_i, \lambda_{i+k}]$ it is sufficient to project instead on $\check{V}$ on the subspace $Z = [z_{\ell+1}, \ldots, z_{\ell+k+1}]$, where $z_j, j = \ell+1, \ldots, \ell+k+1$ are the corresponding eigenvectors of $T_{\check{V}}(\lambda_{\ell+k+1})z = \mu z$.

5. Numerical experiments. To evaluate the local restart technique we consider the conservative gyroscopic eigenvalue problem (1.1). It is well known that all its eigenvalues are real and occur in pairs $\pm\lambda$, that the corresponding eigenvectors are complex conjugate, and that the positive eigenvalues $0 < \lambda_1 \leq \cdots \leq \lambda_n$ satisfy the minmax characterization [4]

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

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

$$x^H T(\lambda) x = -\lambda^2 x^H M x + i\lambda x^H G x + x^H K x = 0.$$

We consider a tire model with 39204 brick elements, 124992 degrees of freedom and 20 different material groups, rotating corresponding to 50 km/h vehicle speed. To demonstrate the efficiency of the local restart technique we compute the eigenvalues $\lambda_{101}, \ldots, \lambda_{200}$ and the associated eigenvectors. This corresponds to the interval $[8250, 13367]$.

All the tests were run on an SGI Altix machine with 160 Itanium 2 Madison 9M processors (1.6 GHz, 6MB L3, single core) sharing 448 GB RAM. However, our implementation uses only single processor. We implemented the algorithm in FORTRAN 90 using the standard linear algebra packages BLAS and LAPACK from SCSL hardware optimized library and SPARSKIT 2. The LU decompositions and subsequent system solves are computed with the PARDISO routines from the Intel MKL library. The projected problems were solved by linearization.

The results are uniformly presented in terms of elapsed CPU times. We preconditioned the Arnoldi method by the LU factorization of $K - \sigma^2 M$ where σ is a shift not too far away from the wanted eigenvalues (we neglected the gyroscopic term in $T(\sigma)$ since its influence is small, and since $K - \sigma^2 M$ can be obtained in real arithmetic). We updated the LU factorization whenever the quotient of the last two residual norms before convergence of an eigenvalue exceeded the threshold $\tau = 0.65$ indicating that the convergence has become too slow.

In our first experiment, we computed all eigenpairs for positive eigenvalues less than 13367 by the Arnoldi method. To prevent the search subspace from getting arbitrarily large we restricted its dimension to 230. We restarted the Arnoldi method with an orthogonal basis of the subspace spanned by the eigenvectors computed so far, every time when the dimension of the search space exceeded this bound. The total computing time was 21,138 seconds (5.87 hours), 13,912 seconds (3.86 hours) of which were spent solving the nonlinear projected eigenvalue problems.

In the next experiment we used the same global restart technique, but this time the restarts were triggered whenever the subspace exceeded the number of a currently converged eigenvalue by 60. This reduced the average dimension of the search spaces and therefore the time for solving the projected problems decreased to 8,731 seconds (2.42 hours). The total computing time was 15,864 seconds (4.41 hours). Figures 5.1 and 5.2 show the total computing time and the time consumed for solving the projected nonlinear problems for these two experiments. Obviously in both cases the superlinear growth of the total CPU time is mainly caused by the solver of the projected eigenproblems.

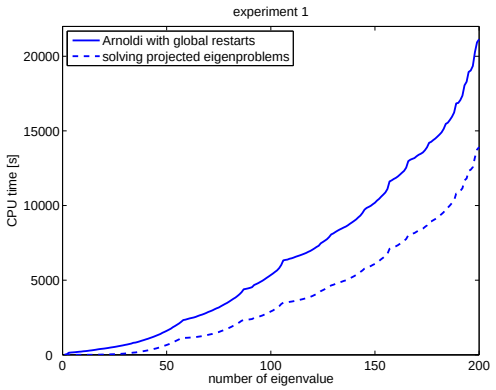

FIG. 5.1. *Global restarts; experiment 1*

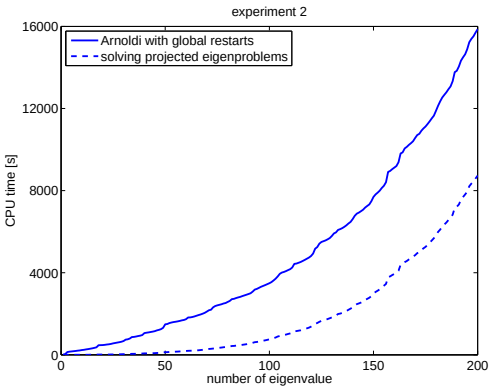

FIG. 5.2. *Global restarts; experiment 2*

Computing the smallest 200 eigenvalues (experiment 3) with the local strategy from Section 4 restarting whenever the search space dimension exceeded 60 or when the convergence rate τ goes beyond 0.65 it took 12,676 seconds (3.52 hours) to compute all eigenvalues, where only 295 seconds were spent on solving the nonlinear problems. The elapsed computing times are shown in Figure 5.3.

The outstanding advantage of the local strategy is the fact that we do not have to determine the leading eigenpairs if we are only interested in eigenvalues in a given interval. All that we need is an anchor which can be determined by residual inverse iteration with a shift close to the left bound. In this way (experiment 4) the total computing time (shown in Figure 5.4) was reduced further to 8125 seconds (2.26 hours), while 126 seconds were consumed by the nonlinear solver, for computing all eigenvalues in $[8250, 13376]$.

Bearing in mind that for large problems the setup time for a restart, i.e. the cost for determining the preconditioner and generating the new search space and the projected problem, can be relatively high in comparison to the remaining computations, we can further improve the performance admitting the algorithm to balance these expenses automatically.

Let t_r denote the setup time of a restart, and let t_e^i be the time needed for computing the ith eigenvalue of problem (1.2), where i denotes the local number after the restart. Then the total time for computing the first i eigenvalues is $t_t^i = t_r + \sum_{j=1}^{i} t_e^j$, and the average time for computing one eigenvalue in this loop is $\bar{t}_e^i = t_t^i/i$.

Let $\alpha \geq 1$ and $N_v \in \mathbb{N}_0$ be parameters depending on the given problem, and let

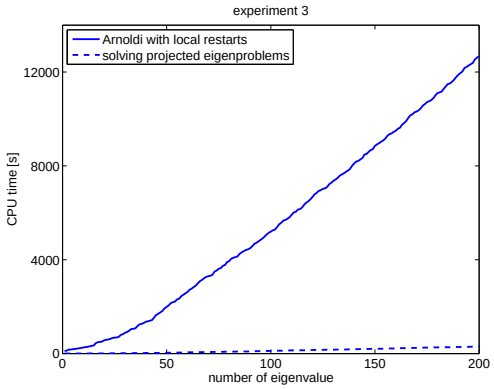

FIG. 5.3. *Local restarts; experiment 3*

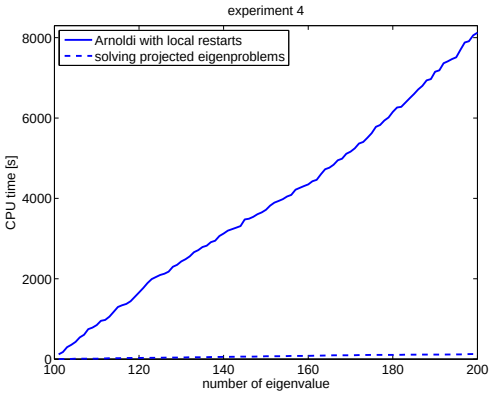

FIG. 5.4. *Local restarts; experiment 4*

$n_v = N_v$. We adjust n_v in the ith step after a restart in the following way

$$n_v \leftarrow \begin{cases} \min\{N_v, n_v + 1\} & \text{if} \quad t_e^i \leq \alpha \cdot \bar{t}_e^i \\ n_v - 1 & \text{else} \end{cases}$$

and we restart the method again if $n_v < 0$. Hence, we do not allow too often that the time required for convergence to an eigenvalue is bigger than the average time for convergence including the setup time. In particular, if $N_v = 0$ and $\alpha = 1$ we restart the algorithm straightaway when the time for convergence to an eigenvalue is bigger than the average time for computing the previous eigenvalues since the last restart.

With $\alpha = 1$ and $N_v = 3$, restarting whenever the search space dimension exceeded 100 this strategy reduced the total time for computing all eigenvalues in $[8250, 13367]$ to 6,813 seconds with only 548 seconds spent on solving all projected nonlinear eigenproblems. The elapsed computation times are pictured in Figure 5.5.

The plots in Figures 5.4 and 5.5 show that the cost for computing one eigenvalue is approximately the same, no matter what its number is. Thus the new restart technique effectively eliminates the superlinear growth with the number of eigenvalues and constitutes an efficient method for computing eigenvalues in the interior of the spectrum.

6. Conclusion. We presented a local restart strategy for iterative projection methods for nonlinear Hermitean eigenvalue problems satisfying a minmax characterization of its eigenvalues. Properly initialized, the method can be employed for computing eigenvalues in the interior of the spectrum. We proposed a strategy for dealing with the spurious Ritz values of the projected problems, which intrinsically arise in the higher frequencies computations. The efficiency of the method was demonstrated for a large gyroscopic eigenvalue problem modelling the dynamic behavior of a rotating tire.

REFERENCES

[1] M.M. Betcke. *Iterative Projection Methods for Symmetric Nonlinear Eigenvalue Problems with Applications*. PhD thesis, Institute of Numerical Simulation, Hamburg University of Technology, 2007.

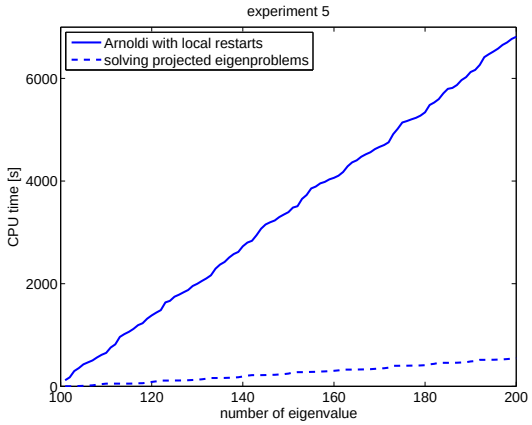

FIG. 5.5. *Balanced local restarts*

[2] T. Betcke and H. Voss. A Jacobi–Davidson–type projection method for nonlinear eigenvalue problems. *Future Generation Comput. Syst.*, 20(3):363 – 372, 2004.

[3] R.J. Duffin. A minimax theory for overdamped networks. *J. Rat. Mech. Anal.*, 4:221 – 233, 1955.

[4] R.J. Duffin. The Rayleigh–Ritz method for dissipative and gyroscopic systems. *Quart. Appl. Math.*, 18:215 – 221, 1960.

[5] D.R. Fokkema, G.L.G. Sleijpen, and H.A. van der Vorst. Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils. *SIAM J. Sci. Comput.*, 20:94 – 125, 1998.

[6] E. Jarlebring and H. Voss. Rational Krylov for nonlinear eigenproblems, an iterative projection method. *Applications of Mathematics*, 50:543 – 554, 2005.

[7] B.-S. Liao, Z. Bai, L.-Q. Lee, and K. Ko. Solving large scale nonlinear eigenvalue problems in next-generation accelerator design. Technical Report CSE-TR, University of California, Davis, 2006.

[8] M. Markiewicz and H. Voss, A local restart procedure for iterative projection methods for nonlinear symmetric eigenproblems. *Algoritmy 2005,* 17^{th} *Conference on Scientific Computing, Vysoke Tatry - Podbanske, Slovakia 2005*, 212 – 221.

[9] U. Nackenhorst. The ALE–formulation of bodies in rolling contact. Theoretical foundations and finite element approach. *Comput. Meth. Appl. Mech. Engrg.*, 193:4299 – 4322, 2004.

[10] U. Nackenhorst and O. von Estorff. Numerical analysis of tire noise radiation – a state of the art review. In *Inter-noise 2001. The 2001 International Congress and Exhibition on Noise Control Engineering*, The Hague, The Netherlands, 2001.

[11] A. Neumaier. Residual inverse iteration for the nonlinear eigenvalue problem. *SIAM J. Numer. Anal.*, 22:914 – 923, 1985.

[12] E.H. Rogers. A minimax theory for overdamped systems. *Arch. Ration. Mech. Anal.*, 16:89 – 96, 1964.

[13] A. Ruhe. A rational Krylov algorithm for nonlinear matrix eigenvalue problems. *Zapiski Nauchnyh Seminarov POMI*, 268:176 – 180, 2000.

[14] A. Ruhe. Rational Krylov for large nonlinear eigenproblems. In J. Dongarra, K. Madsen, and J. Wasniewski, editors, *Applied Parallel Computing. State of the Art in Scientific Computing*, volume 3732 of *Lecture Notes on Computer Science*, pages 357 – 363, Berlin, 2006. Springer Verlag.

[15] H. Voss. An Arnoldi method for nonlinear symmetric eigenvalue problems. In *Online Proceedings of the SIAM Conference on Applied Linear Algebra, Williamsburg.*, `http://www.siam.org/meetings/laa03/`, 2003.

[16] H. Voss. An Arnoldi method for nonlinear eigenvalue problems. *BIT Numerical Mathematics*, 44:387 – 401, 2004.

[17] H. Voss. A Jacobi–Davidson method for nonlinear eigenproblems. In M. Buback, G.D. van Albada, P.M.A. Sloot, and J.J. Dongarra, editors, *Computational Science – ICCS 2004, 4th International Conference, Kraków, Poland, June 6–9,2004,Proceedings, Part II*, volume 3037 of *Lecture Notes in Computer Science*, pages 34–41, Berlin, 2004. Springer Verlag.

[18] H. Voss. Numerical methods for sparse nonlinear eigenproblems. In Ivo Marek, editor, *Proceedings of the XV-th Summer School on Software and Algorithms of Numerical Mathematics, Hejnice, 2003*, pages 133 – 160, University of West Bohemia, Pilsen, Czech Republic, 2004. Available at `http://www.tu-harburg.de/mat/Schriften/rep/rep70.pdf`.

[19] H. Voss and B. Werner. A minimax principle for nonlinear eigenvalue problems with applications to nonoverdamped systems. *Math. Meth. Appl. Sci.*, 4:415–424, 1982.