Proceedings of ALGORITMY 2005
pp. 212–221

A LOCAL RESTART PROCEDURE FOR ITERATIVE PROJECTION METHODS FOR NONLINEAR SYMMETRIC EIGENPROBLEMS

MARTA MARKIEWICZ* AND HEINRICH VOSS†

Abstract. For nonlinear eigenvalue problems $T(\lambda)x = 0$ satisfying a minmax characterization of its eigenvalues iterative projection methods combined with safeguarded iteration are suitable for computing all eigenvalues in a given interval. Such methods hit their limitation if a large number of eigenvalues (in the interior of the spectrum) are required. In this paper we propose a localized version of safeguarded iteration which is able to cope with this problem.

Key words. eigenvalue, nonlinear eigenproblem, Arnoldi method, restart technique, minmax characterization

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

1. Introduction. Acoustic simulation is increasingly becoming an important part of the automotive design process. The most costly part of the effort of optimizing noise and vibration performance is the computation of frequency response for very large finite element models because a very large number of modes (not necessarily at the end of the spectrum) may be needed for obtaining satisfactory accuracy over the frequency range of interest.

For instance, the major source of traffic noise at speed above 40 km/h for passenger cars and above 60 km/h for trucks is the sound radiation of rolling tires (cf. [6], [7]). Simulating the structural dynamics of the rolling tire by an Arbitrary Lagrangian Eulerian (ALE) approach one ends up with the conservative gyroscopic eigenvalue problem

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

Here K is the stiffness matrix modified by the presence of centrifugal forces, M is the mass matrix, and G is the gyroscopic matrix stemming from the Coriolis force. K and M are assumed to be symmetric and positive definite, and G is skew–symmetric.

More generally we consider the nonlinear eigenvalues problem

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

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. For this type of problems iterative projection methods were considered in [1, 5, 10, 11, 12, 13, 14, 15], and for the special case that the eigenvalues of (1.2) in J can be characterized as minmax values of a Rayleigh functional a combination with safeguarded iteration was discussed to determine a moderate number of consecutive eigenvalues in [1] and [12].

This approach hits it limitations if a large number of eigenvalues (in particular in the interior of the spectrum) of (1.2) is needed, since in this case one has to

*Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, England, (m.markiewicz@tu-harburg.de).

†Hamburg University of Technology, Institute of Mathematics, D-21071 Hamburg, Germany (voss@tu-harburg.de).

project the problem under consideration onto a sequence of search spaces of growing dimensions requiring an excessive amount of storage and computing time. In this paper we propose a new restart technique which projects problem (1.2) only to search spaces of limited dimension. Our presentation is restricted to the Arnoldi method, but the local restart technique applies to any other iterative projection method.

The paper is organized as follows. Section 2 outlines the variational characterization of eigenvalues for nonlinear and nonoverdamped eigenproblems and the safeguarded iteration method, and Section 3 recalls the Arnoldi method for sparse, symmetric, and nonlinear eigenproblems. In Section 4 we present the new restart technique, and we discuss the problem of spurious eigensolutions. An example of a gyroscopic eigenproblem in Section 5 demonstrates the efficiency of the new restart method.

2. Solving dense symmetric nonlinear eigenproblems. We consider the 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 for the linear case $T(\lambda) = \lambda I - A$ a parameter $\lambda \in J$ is called an eigenvalue of $T(\cdot)$ if problem (2.1) has a nontrivial solution $x \neq 0$ which is called an eigenvector corresponding to λ.

For a linear Hermitean problem $Ax = \lambda x$ all eigenvalues are real, and if they are ordered by magnitude $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ then it is well known that they can be characterized by the minmax principle of Poincaré.

Similar results hold for certain nonlinear eigenvalue problems, too. We assume that for every fixed $x \neq 0$ the real function $f(\lambda; x) := x^H T(\lambda)x$ is continuously differentiable on J, and that the equation

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

has at most one solution in J. Then equation (2.2) implicitly defines a functional p on some subset D of $\mathbb{C}^n \setminus \{0\}$ which replaces the Rayleigh quotient in the variational characterization of eigenvalues of problem (2.1), and which we call the Rayleigh functional.

For nonlinear eigenvalue problems variational properties using the Rayleigh functional were proved by Duffin [2, 3], and Rogers [9] for finite dimensional overdamped problems, i.e. if the Rayleigh functional p is defined in the entire space $\mathbb{C}^n \setminus \{0\}$. Nonoverdamped problems were considered by Werner and the second author [16].

In the general case the natural enumeration for which the smallest eigenvalue is the first one, the second smallest is the second one, etc. is not appropriate, but the number of an eigenvalue λ of the nonlinear problem (2.1) is inherited from the number of the eigenvalue 0 of the matrix $T(\lambda)$.

If $\lambda \in J$ is an eigenvalue of problem (2.1) then $\mu = 0$ is an eigenvalue of the linear problem $T(\lambda)y = \mu y$, and 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 k-th eigenvalue of (2.1).

With this enumeration the following minmax characterization of the eigenvalues of the nonlinear eigenproblem (2.1) was proved in [16]:

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*

$$x^H T'(p(x))x > 0 \quad \text{for every } x \in D.$$

Then the following assertions hold:

(i) For every $k \in \mathbb{N}$ there is at most one k-th 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 k-th eigenvalue $\lambda_k \in J$. Then it holds*

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

The correspondence between a k-th eigenvalue λ_k of $T(\cdot)$ and the k largest eigenvalue of the matrix $T(\lambda_k)$ suggests the safeguarded iteration for computing the k-th eigenvalue of a nonlinear problem given in Algorithm 1. Its convergence properties were proved in [15], and are collected in Theorem 2.2.

Algorithm 1 Safeguarded iteration

1: Start with an approximation μ_1 to the k-th 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**

THEOREM 2.2.

(i) If $\lambda_1 := \inf_{x \in D} p(x) \in J$, and if λ_1 is a simple eigenvalue of (2.1)*, then the safeguarded iteration converges globally and quadratically to λ_1.*

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

(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.

The safeguarded iteration is definitely not capable to solve large nonlinear eigenvalue problems. However, as an inner iteration in a projection method it is well suited since its convergence properties and for small dimension its complexity are similar to those of inverse iteration. As an advantage upon inverse iteration it aims at an eigenvalue with a specific number, and therefore it is less likely to miss an eigenvalue if one is interest in all eigenvalues in an interval.

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

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

iterative projection methods are very efficient. Here the dimension of the eigenproblem is reduced by projecting it to a subspace of much smaller dimension, and the reduced problem is handled by a fast technique for dense problems. The subspaces are expanded in the course of the algorithm in an iterative way with the aim that some of the eigenvalues of the reduced matrix become 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, 5, 10, 11, 12, 13, 14, 15]. A typical example is the nonlinear Arnoldi method in Algorithm 2, where we assume that problem (1.2) is symmetric, and the eigenvalues can be enumerated according to Section 2, and we are interested in computing the eigenvalues $\lambda_{m_{\min}}, \ldots, \lambda_{m_{\max}}$.

Algorithm 2 Nonlinear Arnoldi Method

start with an initial shift σ and an initial basis V, $V^H V = I$;
determine preconditioner $M \approx T(\sigma)^{-1}$, σ close to first wanted eigenvalue
for $m = m_{\min}, \ldots, m_{\max}$ **do**
  compute m smallest eigenvalue μ and corresponding eigenvector y of the projected problem $T_V(\mu)y := V^H T(\mu) V y = 0$ by safeguarded iteration
  determine Ritz vector $u = Vy$ and residual $r_k = T(\mu)u$
  if $\|r_k\|/\|u\| < \epsilon$ **then**
    accept approximate eigenpair $\lambda_m = \mu$, $x_m = u$,
    choose new shift σ and determine preconditioner $M \approx T(\sigma)^{-1}$ if indicated
    restart if necessary
    choose approximations μ and u to next eigenvalue and eigenvector
    determine residual $r = T(\mu)u$
  end if
  $v = Mr$
  $v = v - VV^H v$, $\tilde{v} = v/\|v\|$, $V = [V, \tilde{v}]$
  reorthogonalize if necessary
  update projected problem $T_V(\mu) = V^H T(\mu) V$
end for

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 and no Arnoldi recursion is determined 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 [8].

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 and how to restart, and how to continue after an eigenpair was accepted. A detailed discussion is contained in [12, 13]. Here we concentrate on the start and restarts for symmetric problems allowing a minmax characterization of their eigenvalues.

A crucial point in iterative projection methods for general nonlinear eigenvalue problems when approximating more than one eigenvalue is to inhibit the method from

converging to the same eigenvalue repeatedly. In the linear case this is no problem. Krylov subspace solvers construct an orthogonal basis of the ansatz space not aiming at a particular eigenvalue, and one gets approximations to extreme eigenvalues without replication (at least if reorthogonalization is employed). If several eigenvalues are computed by the Jacobi–Davidson method then one determines an incomplete Schur factorization thus preventing the method from approaching an eigenvalue which was already obtained previously (cf. [4]). For nonlinear problems a similar normal form does not exist.

If $T(\lambda)$ is a family of symmetric matrices allowing a minmax characterization of its eigenvalues in an open interval J, and if 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 will differ.

If J contains a first eigenvalue $\lambda_1 = \min_{x \in D} \; p(x)$, then by Theorem 2.2 the safeguarded iteration for (3.2) converges globally 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 j-th eigenvalue λ_j of (1.2), and if $x_j \in \mathcal{V}$ for $j = 1, \ldots, k$, then λ_j is a j-th 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 one after the other by the Nonlinear Arnoldi algorithm starting with an approximation to x_1.

As the subspaces expand 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 numeration 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 the 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 [16] 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 k-th 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 Nonlinear Arnoldi Method as described in the last section hits its 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 eigenvalue, which results in increasing time consumed by the nonlinear solver and increasing storage requirement.

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}
1: $V = [x_i, v_1]$;
2: j = 1;
3: **while** Restart condition not satisfied **do**
4: **repeat**
5: Determine largest eigenvalues $\mu_1(\lambda_i) \geq \cdots \geq \mu_k(\lambda_i) > 0 \geq \mu_{k+1}(\lambda_i)$ of (4.2)
6: Set $\ell := k$ if $\mu_k \leq -\mu_{k+1}$, and else $\ell := k + 1$
7: Compute $(\ell + j)$-th eigenpair $(\tilde{\lambda}_{\ell+j}, y_{\ell+j})$ of $T_V(\cdot)$
8: Expand $V = [V, MT(\tilde{\lambda}_{\ell+j})Vy_{\ell+j}]$
9: **until** Eigenpair $(\tilde{\lambda}_{\ell+j}, Vy_{l+j}) =: (\lambda_{i+j}, x_{i+j})$ converged
10: j = j+1;
11: **end while**

We propose a way to overcome this difficulty by introducing 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 a 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

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

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 ℓ-th eigenvalue of problem (4.1) if $\mu(\hat{\lambda}) = 0$ is the ℓ largest eigenvalue of the linear problem

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

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 expanded in the same way as in Algorithm 2 aiming at the $(\ell(\mathcal{V}_k) + 1)$-th eigenvalue in the k-th iteration step. Notice that the number $\ell(\mathcal{V}_k)$ of the anchor may change in the course of the algorithm.

After convergence we may continue the Nonlinear Arnoldi 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 spaces 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.

Some comments are in order.

1. In practice the search subspace $\mathcal{V}$ usually contains an approximation rather than the exact eigenvector, thus we assign the number ℓ of the eigenvalue $\mu(\lambda_i)$ of the linear problem (4.2) with minimal absolute value to the anchor λ_i.
2. 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 $k \geq 1$.
   If the angle between the eigenvectors x_{i+k} and x_{i+k+1} is different from 0 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.
   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:
   First, an eigenvalue of (1.2) in the interval $(\lambda_i, \lambda_{i+k})$ might have been missed out because the corresponding eigenvector $\hat{x}$ were not sufficiently present in the initial search space $\operatorname{span}\{x_i, v_1\}$ and might have not been amplified sufficiently in the course of the expansions of $\mathcal{V}$ until computing λ_{i+k}. Afterwards the component of $\hat{x}$ in the search space $\mathcal{V}$ was increased and became big enough to produce the additional eigenvalue approximation $\theta \in (\lambda_i, \lambda_{i+k})$, and Algorithm 3 yielded the eigenvalue approximation λ_{i+k} the second time.
   Secondly, 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 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 and its local number $\ell + j$, and we expand the search space $\hat{\mathcal{V}} = \operatorname{span}\{\mathcal{V}, MT(\theta)x_\theta\}$ by the direction of residual inverse iteration 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 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) or there are $k+2$ eigenvalues $\lambda_i, \ldots, \lambda_{i+k}, \hat{\theta} \in [\lambda_i, \lambda_{i+k}]$, and it holds $\hat{\theta} \leq \theta$.
   In the latter case we repeat the expansion by residual inverse iteration until the sequence of additional eigenvalues in $[\lambda_i, \lambda_{i+k}]$ has been moved out of the interval or has converged to an additional eigenvalues. We then adjust the numeration of the eigenvalues and continue the Arnoldi method.
3. Notice that more than one additional eigenvalue may exist in $[\lambda_i, \lambda_{i+k}]$ after we detected a replicate eigenvalue. They all can be treated in the same way as in the last item one after the other.

5. Numerical experiments. To evaluate the local restart technique we consider the conservative gyroscopic eigenvalue problem (1.1). It is well known that all 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 [3]

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

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 coarse finite element model of a rotating wheel of dimension 1728. 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 $[11780, 16820]$.

All the tests were run under MATLAB 7 on a 3.2 GHz Intel Xeon Processor with 2 GB RAM. 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 updated the LU factorization when the quotient of the last two residual norms before convergence of an eigenvalue exceeded a given threshold τ indicating that the convergence has become too slow.

In our first experiment, we computed all eigenpairs for positive eigenvalues less than 16820 by the Arnoldi method with safeguarded iteration without restart ending up with a search subspace of dimension 694. The total computing time was 3957 seconds, 3771 seconds of which were consumed solving the nonlinear projected eigenproblems.

To prevent the search subspace from getting arbitrarily large we restricted its dimension to 230 in our second experiment. We restarted the Arnoldi method with an orthonormal basis of the subspace spanned by the eigenvectors computed so far, every time when the subspace dimension exceeded this bound. This reduced the total computing time to 479 seconds, where 323 seconds were spent on solving the nonlinear projected problems.

In the third experiment we used the same restart technique, but this time the restarts were triggered whenever the dimension of the subspace exceeded the number of a currently converged eigenvalue by more than 30. This reduced the total computing time further to 347 seconds and 174 seconds for the nonlinear solver. Figure 5.1 shows the total computing time and the time consumed for solving the projected nonlinear problems indicating that the superlinear growth of the total CPU time is mainly caused by the solver of the projected eigenproblems.

Computing the smallest 200 eigenvalues with the local strategy from Section 4 restarting whenever the search space dimension exceeded 60 or when the convergence rate τ goes beyond 0.3 it took 199 seconds to compute all eigenvalues, where only 49 seconds were spent on solving the nonlinear problems. The elapsed computing times are shown in Figure 5.2.

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. This way the total computing time was reduced further to 102 seconds, while 21.7 seconds was consumed by the nonlinear solver, for computing all eigenvalues in $[11780, 16820]$.

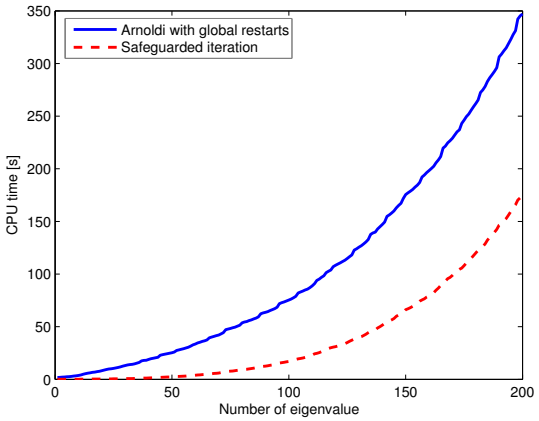

FIG. 5.1. *CPU time consumption for global restarts*

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 i-th 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 $n_v = N_v$. We adjust n_v after a restart in the i-th step 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 $N_v = 0$ this restart strategy reduced the total time for computing all eigenvalues in $[11780, 16820]$ to 81.5 seconds with 5.4 seconds only spent on solving all projected nonlinear eigenproblems. The elapsed computation times are pictured in Figure 5.3.

The plots in Figures 5.2 and 5.3 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.

Acknowledgements. The first author gratefully acknowledges financial support by the Rotary Foundation which granted the Rotary Ambassadorial Scholarship for a study visit at the University of Oxford.

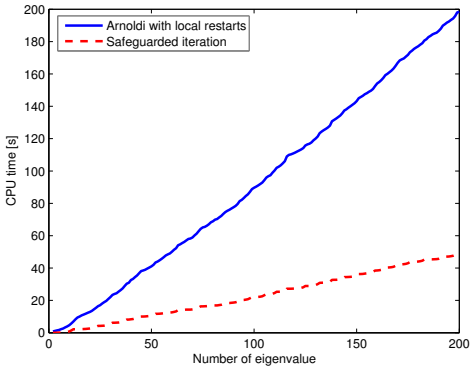

FIG. 5.2. *Local restarts*

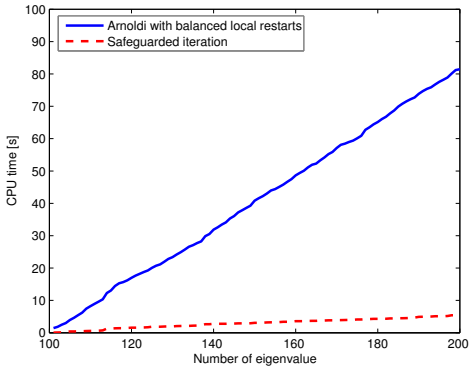

FIG. 5.3. *Balanced local restarts*

REFERENCES

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

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

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

[4] 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.

[5] E. Jarlebring and H. Voss. Rational Krylov for nonlinear eigenproblems, an iterative projection method. Technical Report 69, Section of Mathematics, Hamburg University of Technology, 2003. To appear in Appl.Math.

[6] U. Nackenhorst. Rollkontaktdynamik – Numerische Analyse der Dynamik rollender Körper mit der Finite Elemente Methode. Habilitationsschrift, Institut für Mechanik, Universität der Bundeswehr, Hamburg, 2000.

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

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

[9] E.H. Rogers. A minmax theory for overdamped systems. *Arch.Rat.Mech.Anal.*, 16:89 – 96, 1964.

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

[11] A. Ruhe. Rational Krylov for large nonlinear eigenproblems. Technical report, Royal Institute of Technology, Stockholm, Sweden, 2004. To appear in Proceedings of PARA'04, Lyngby, Denmark, 2004.

[12] 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.

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

[14] 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.

[15] 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.

[16] 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.