Mathematical Modelling and Analysis
Volume 13 Number 2, 2008, pages 171–182
 ISSN 1392-6292 print, ISSN 1648-3510 online

RESTARTING PROJECTION METHODS FOR RATIONAL EIGENPROBLEMS ARISING IN FLUID-SOLID VIBRATIONS

M. M. BETCKE[1] and H. VOSS[2]

[1] *The University of Manchester, School of Mathematics*
Oxford Road, M13 9PL, Manchester, United Kingdom
[2] *Hamburg University of Technology, Institute of Numerical Simulation*
D-21071 Hamburg, Germany
E-mail: m.betcke@maths.manchester.ac.uk;voss@tu-harburg.de

Received September 1, 2005; revised October 15, 2005; published online May 1, 2008

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 limitations if a large number of eigenvalues is required. In this paper we discuss restart procedures which are able to cope with this problem, and we evaluate them for a rational eigenvalue problem governing vibrations of a fluid-solid structure.

Key words: nonlinear eigenvalue problem, iterative projection method, Arnoldi method, minmax characterization, restart, fluid-solid structure

1 Introduction

We consider the nonlinear eigenvalues problem

$$T(\lambda)x = 0, \tag{1.1}$$

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 are very efficient (cf. [4, 6] and the literature given therein). In particular, if the eigenvalues of $T(\cdot)$ satisfy a minmax characterization, all eigenvalues can be determined one after the other in a safe way. However, this approach hits its limitations if a large number of eigenvalues or eigenvalues in the interior of the spectrum of (1.1) are needed. In this case one has to project the problem under consideration onto a sequence of search spaces of growing dimensions requiring an excessive amount of storage and computing time. In [2] we presented a new restart technique which projects problem (1.1) only onto

search spaces of limited dimension. Here we generalize this approach to rational eigenvalue problems governing free vibrations of fluid-solid structures. 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 as well as and the safeguarded iteration method. Section 3 recalls the Arnoldi method for sparse, symmetric, and nonlinear eigenproblems. In Section 4 we present the local restart technique. In particular we discuss the problem of spurious eigensolutions. An example of a rational eigenproblem in Section 5 demonstrates the efficiency of the new restart procedure.

2 Variational Characterization of Eigenvalues

In this section we recall conditions under which the eigenvalues of $T(\cdot)$ can be characterized as minmax values of a Rayleigh functional. Let $J \subset \mathbb{R}$ be an open interval which may be unbounded, and assume that $T(\lambda) \in \mathbb{C}^{n\times n}$ is a family of Hermitean matrices. Suppose that for every $x \in \mathbb{C}^n \setminus \{0\}$ the real equation

$$f(\lambda, x) := x^H T(\lambda)x = 0 \tag{2.1}$$

has at most one solution $\lambda \in J$. Then equation (2.1) defines a functional p on some subset $D \subset \mathbb{C}^n$ which obviously generalizes the Rayleigh quotient for linear pencils $T(\lambda) = \lambda B - A$, and which is called the Rayleigh functional.

We assume further

$$(\lambda - p(x))f(\lambda, x) > 0 \quad \text{for every } x \in D \text{ and } \lambda \in J \setminus \{p(x)\}$$

generalizing the definiteness condition for linear pencils.

Under these conditions a minmax principle for the nonlinear eigenproblem (1.1) was proved in [7] if the eigenvalues are enumerated appropriately. A value $\lambda \in J$ is an eigenvalue of (1.1) if and only if $\mu = 0$ is an eigenvalue of the matrix $T(\lambda)$, and by Poincaré's maxmin principle there exists $m \in \mathbb{N}$ with

$$0 = \max_{\dim V = m} \min_{x\in V,\ x\neq 0} \frac{x^H T(\lambda)x}{\|x\|^2}.$$

Then we assign this m to λ as its number and call λ an mth eigenvalue of problem (1.1).

Under the assumptions above it was shown in [7] that for every $m \in \{1, \ldots, n\}$ problem (1.1) has at most one mth eigenvalue in J, which can be characterized by

$$\lambda_m = \min_{\dim V = m, D\cap V\neq\emptyset} \sup_{v\in D\cap V} p(v). \tag{2.2}$$

The minimum is attained by the invariant subspace of $T(\lambda_m)$ corresponding to its m largest eigenvalues, and the supremum is attained by any eigenvector of $T(\lambda_m)$ corresponding to $\mu = 0$.

Algorithm 1 Safeguarded Iteration

1: Start with an approximation σ_1 to the mth eigenvalue of (1.1)
2: **for** $k = 1, 2, \ldots$ until convergence **do**
3: Compute an eigenvector x_k corresponding to the m-largest eigenvalue of $T(\sigma_k)$
4: Solve $x_k^H T(\sigma_{k+1}) x_k = 0$ for σ_{k+1}
5: **end for**

The enumeration of eigenvalues and the fact that the eigenvectors of (1.1) are the stationary vectors of the Rayleigh functional suggests the Algorithm 1 called safeguarded iteration for computing the mth eigenvalue.

Safeguarded iteration has the following convergence properties (cf. [5]):

(i) If $\lambda_1 := \inf_{x \in D} p(x) \in J$ and $x_1 \in D$ then the safeguarded iteration converges globally to λ_1.

(ii) If $\lambda_m \in J$ is a mth eigenvalue of (1.1) which is simple, then the safeguarded iteration converges locally and quadratically to λ_m.

The safeguarded iteration is certainly not capable of solving large and sparse nonlinear eigenvalue problems. However, it is well suited as an inner iteration in a projection method, especially if one is interested in some particular eigenvalues, e.g. in an interval. Since it aims at an eigenvalue with a specific number missing out of eigenvalues becomes less likely.

3 Iterative Projection Methods

Iterative projection methods like Lanczos, Arnoldi or Jacobi-Davidson are very efficient for tackling sparse linear eigenvalue problems, and so are their generalizations for nonlinear ones. A typical example is the nonlinear Arnoldi method in Algorithm 2, where we assume that problem (1.1) is Hermitean, and the eigenvalues can be enumerated according to Section 2.

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, when to change and how to choose the preconditioner, when and how to restart. A detailed discussion is contained in [6]. Here we concentrate on the initialization and on restarts for symmetric problems allowing a minmax characterization of their eigenvalues.

If $T(\lambda)$ is a family of Hermitean 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 the projected problem

$$T_V(\lambda) y := V^H T(\lambda) V y = 0 \tag{3.1}$$

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.

Algorithm 2 Nonlinear Arnoldi Method

Start with initial basis V, $V^HV = I$; $m = 1$
Determine preconditioner $M \approx T(\sigma)^{-1}$, σ close to first wanted eigenvalue
while $m \leq$ number of wanted eigenvalues **do**
  Compute m smallest eigenvalue μ and corresponding eigenvector y of the projected problem $T_V(\mu)y := V^HT(\mu)Vy = 0$ by safeguarded iteration
  Determine Ritz vector $u = Vy$ and residual $r = T(\mu)u$
  if $\|r\|/\|u\| < \epsilon$ **then**
    Accept approximate eigenpair $\lambda_m = \mu$, $x_m = u$; increase $m \leftarrow m + 1$
    Choose new shift σ and determine preconditioner $M \approx T(\sigma)^{-1}$ if indicated
    Restart if necessary
    Find approximation (μ, u) to next eigenpair; determine residual $r = T(\mu)u$
  end if
  $v = Mr$
  $v = v - VV^Hv$, $\tilde{v} = v/\|v\|$
  Reorthogonalize $\tilde{v}$ if necessary
  Expand subspace: $V = [V, \tilde{v}]$
end while

If J contains a first eigenvalue $\lambda_1 = \min_{x\in D}\ p(x)$, then the safeguarded iteration for (3.1) converges globally for any initial vector $x \in \mathcal{V} \cap D$ to the smallest eigenvalue of (3.1). If x_j denotes an eigenvector corresponding to the jth eigenvalue λ_j of (1.1), and if $x_j \in \mathcal{V}$ for $j = 1, \ldots, k$, then λ_j is a jth eigenvalue of the projected problem (3.1), 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 (1.1) can be determined one after the other by the nonlinear Arnoldi algorithm.

It may happen that the algorithm converges to an eigenvalue that was obtained already in a previous step. If this eigenvalue is not a multiple one (which can be checked by computing the angle between the corresponding eigenvectors) this repeated convergence indicates that some eigenvalue has been missed out because the component of the corresponding eigenvector has not been sufficiently present in the initial vector and has now become large enough that the eigenvalue approximation enters the interval (λ_1, λ_j) where λ_j denotes the current iterate. In this case we detect the missed out eigenvalue by the following back-tracing procedure introduced in [5]. We reduce the number of the current iterate by 1, $j \leftarrow j - 1$ and compute the jth eigenpair again. Now, three possibilities arise:

- If it is a new eigenpair and $\lambda_j \geq \lambda_{j-1}$, then we have found the missed out eigenvalue. Hence, we set $j \leftarrow j+1$ and resume the outer iteration.
- If it is a new eigenpair but $\lambda_j < \lambda_{j-1}$ then we have found a missed out eigenvalue, but there are more such eigenvalues in the interval (λ_1, λ_j).

We then set $j \leftarrow j-1$ and compute the jth eigenvalue again.

- If it is a previously computed eigenpair and $\lambda_j \leq \lambda_{j-1}$ we have not yet found the missed eigenvalue. Thus, we again reduce $j \leftarrow j-1$ and compute the jth eigenvalue again.

We continue along these lines until $\lambda_j > \lambda_{j-1}$, and the numbering has been recovered.

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 numeration of the eigenvalues, and we can continue as above to determine the subsequent eigenpairs.

4 Local Restarts

The restart strategy described in the last section hits it 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 requirements. To overcome these difficulties we proposed an adjusted version of a local numbering [2], which does not require to include the entire set of preceding eigenvectors into the search subspace after a restart.

Assume that we are given an eigenvalue $\hat{\lambda} \in J$, 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

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

Moreover, $T_V(\cdot)$ satisfies the conditions of the minmax characterization. Thus, 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 $V^H T(\hat{\lambda}) V y = \mu(\hat{\lambda}) y$.

Starting with $\mathcal{V} =: \mathcal{V}_0$ we determine approximations to the eigenvalue subsequent to the anchor $\hat{\lambda}$, projecting problem (1.1) to a sequence of subspaces $\mathcal{V}_0 \subset \mathcal{V}_1 \subset \mathcal{V}_2 \subset \ldots$ which are generated in course of the Arnoldi method aiming at the $(\ell(\mathcal{V}_k)+1)$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 spaces

Algorithm 3 Restart Framework

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 projected problem (3.1)
    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

until the convergence has become too slow or the dimension exceeds a given bound. 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:

(i) the eigenvalue is a multiple (at least double) eigenvalue;

(ii) the algorithm converged to a single eigenvalue twice.

If λ_{i+k} is not a multiple eigenvalue (i.e. if the angle between x_{i+k} and x_{i+k+1} is close to 0), 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 (1.1) 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 $\mathrm{span}\{x_i, v_1\}$ and might not have been amplified sufficiently in the course of the expansions of $\mathcal{V}$ until computing λ_{i+k+1}. 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 $\theta \in (\lambda_i, \lambda_{i+k})$.

- It might be the case that no eigenvalue of (1.1) 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. it is a linear combination of eigenvectors of (1.1) 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 $\hat{\mathcal{V}}$ 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

$$T_{\hat{V}}(\lambda)\hat{y} = 0 \tag{4.2}$$

are less than or equal to the corresponding ones of $T_V(\lambda)y = 0$, and either problem (4.2) has exactly $k+1$ eigenvalues $\lambda_i, \dots, \lambda_{i+k} \in [\lambda_i, \lambda_{i+k}]$ (i.e. the additional eigenvalue has left the interval of interest) or there are still $k+2$ eigenvalues $\lambda_i, \dots, \lambda_{i+k}, \hat{\theta} \in [\lambda_i, \lambda_{i+k}]$, and it holds $\hat{\theta} \leq \theta$. In the latter case we repeat the expansion 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.

5 Restarts for Rational Eigenvalue Problems

We consider a rational eigenvalue problem

$$Kx = \lambda Mx + \sum_{j=1}^{k} \frac{\lambda}{\sigma_j - \lambda} C_j C_j^T x, \tag{5.1}$$

where $K, M \in \mathbb{R}^{n\times n}$ are symmetric and positive definite, $C_j \in \mathbb{R}^{n\times r_j}$ 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 as finite element models of rational eigenproblems governing free vibrations of tube bundles immersed in a slightly compressible fluid [3].

It is easily seen that in each of the intervals $J_j := (\sigma_{j-1}, \sigma_j)$, $j = 1, \dots, k+1$ with $\sigma_0 = 0$, $\sigma_{k+1} = \infty$, problem (5.1) satisfies the conditions of the minmax characterization (2.2). In J_1 the eigenvalues are numerated in the natural way. For the following intervals the numeration follows from Theorem 1 which was proved in [5].

Theorem 1. *Let n_j be the number of eigenvalues λ_i of the reduced linear eigenvalue problem*

$$\Big(K + \sum_{i=1}^{j-1} \frac{\sigma_j}{\sigma_j - \sigma_i} C_i C_i^T\Big)x = \lambda\Big(M + \sum_{i=j+1}^{k} \frac{1}{\sigma_i - \sigma_j} C_i C_i^T\Big)x, \quad C_j x = 0$$

satisfying $\lambda_i \leq \sigma_j$. Then the rational eigenvalue problem (5.1) *has exactly $n_{j+1} - n_j + r_{j+1}$ eigenvalues in $(\sigma_j, \sigma_{j+1}]$ enumerated by $n_j+1, n_j+2, \dots, n_{j+1}+r_{j+1}$.*

This theorem suggests a global strategy how for determining all eigenvalues in a given interval $I \subset \mathbb{R}_+$ by the nonlinear Arnoldi method, which is similar to the approach at the end of Section 3.

For $J_1 = (0, \sigma_1)$ the infimum of the Rayleigh functional is contained in J_1, and due to the global convergence of the safeguarded iteration we can start with any vector v such that $p(v) \in J_1$, and compute the eigenvalues in J_1 one after the other until the method leaves the interval J_1.

Assume that we have already determined all eigenvalues in $J_1, \dots J_j$, and let N_j be the number (according to the enumeration in Section 2 for problem (5.1) in J_j) of the largest eigenvalue λ_{N_j} found in J_j. To start the Arnoldi method for $J_{j+1} := (\sigma_j, \sigma_{j+1})$ we choose $\hat{\mu} = \sigma_j + \varepsilon$, $\varepsilon > 0$ small, and determine the eigenvalues of the linear problem

$$\Big(K + \sum_{i=1}^{j} \frac{\hat{\mu}}{\hat{\mu} - \sigma_i} C_i C_i^T\Big) x = \lambda \Big(M + \sum_{i=j+1}^{k} \frac{1}{\sigma_i - \hat{\mu}} C_i C_i^T\Big) x, \tag{5.2}$$

which are less than $\hat{\mu}$. We assume that these are $\tilde{n}_j$ of these eigenvalues. If $\tilde{n}_j + r_j = N_j$ then all eigenvalues in $(0, \sigma_j)$ have been found. No eigenvalue exists in $(\sigma_j, \hat{\mu})$, and we can initiate the Arnoldi method for J_{j+1} with an orthonormal basis V_{j+1} of the eigenspace of problem (5.2) corresponding to the $n_j + 1$ smallest eigenvalues. Otherwise we have to explore the interval $(\lambda_{N_j}, \hat{\mu})$ for further eigenvalues.

The method can be restarted if the search space has grown too large where the current initial basis V_j has to be complemented by the eigenvectors corresponding to eigenvalues found in the current interval J_j so far. The method (which we call GLOBAL 1) is safe, but costly if the number of poles σ_j less than $\sup I$ (i.e. the number of linear eigenvalue problems (5.2) to be solved) is large or if there are many eigenvalues in $(0, \sup I)$.

Another variant (called GLOBAL 2) of the iterative projection method can be devised on the basis of Theorem 1. Again we address one interval after the other and within each interval one eigenvalue after the other adapting the number of the eigenvalue for each interval as given by Theorem 1, i.e. every time when a pole is being crossed the number of the eigenvalue is reduced by the rank r_j of the matrix C_j. This variant fully utilizes the information acquired while computing the eigenvalues in the former intervals, since the computation continuous over multiple intervals and no linear problems are being solved. However, there is no guarantee that all eigenvalues in the interval treated so far have been found. Moreover, no restart strategy is implemented and therefore the method is not suitable for a large number of eigenvalues, since the search space is growing prohibitively large.

However, this can be cured by combining the two variants in a way that we compute eigenvalues in several intervals corresponding to method GLOBAL 2, and restart corresponding to method GLOBAL 1 as soon as the search subspace has grown too large. Obviously, this method (called GLOBAL 3) can be restarted at any point $\hat{\mu}$. On restart we can check again, whether eigenvalues have been missed in the computation so far.

Finally, the local restart strategy can be applied to the rational eigenvalue problem in the following way. Let $\hat{\lambda}$ denote an anchor, and $\hat{x}$ the corresponding eigenvector of (1.1), and let ℓ denote the local number of the anchor in the current search subspace $\mathcal{V}$. According to Theorem 1 the number of an eigenvalue is not unique on the real axis and therefore always needs to be considered with the interval containing the eigenvalue. Therefore we denote by $\ell + k_j$ the number of the eigenvalue $\lambda_{\ell+k_j}$ in the interval (σ_j, σ_{j+1}).

As long as the anchor $\hat{\lambda}$ and the currently iterated eigenvalue $\lambda_{\ell+k_j}$ are in

the same interval (σ_j, σ_{j+1}) the local enumeration holds exactly as in Section 4. However, if a pole has been crossed, i.e. $\hat{\lambda} \in (\sigma_j, \sigma_{j+1})$ and $\lambda_{\ell+k_{j+1}} \in (\sigma_{j+1}, \sigma_{j+2})$ the situation is different.

According to Theorem 1 if $\lambda_{\ell+k_j}$ is the largest eigenvalue smaller than σ_{j+1} then the number of the next eigenvalue is $\ell + k_{j+1} = \ell + k_j + 1 - r_{j+1}$, where the offset r_{j+1} is equal to the rank of the corresponding matrix C_{j+1}. For the iterative projection methods it means that at the lth step of the iteration with the search subspace $\mathcal{V}_l$, the offset for the projected problem is $r^l_{j+1} = \operatorname{rank} V_l^H C_{j+1}$. This is bad news, since the offset r^l_{j+1} can vary in the course of the iteration. This can be helped in three ways:

- The simplest but unsatisfactory solution is to reset the anchor every time a pole is being crossed. However, such a procedure violates the condition that the anchor is an eigenvalue of (1.1) since the anchor would have to be set to an approximation to the eigenvalue obtained so far. In this way missing out eigenvalues becomes more probable;

- A better solution is to recompute the offset r^l_{j+1} on every iteration as long as $r^l_{j+1} = \operatorname{rank} V_l^H C_{j+1} < \operatorname{rank} C_{j+1}$. Once $\operatorname{rank} V_l^H C_{j+1} = \operatorname{rank} C_{j+1}$ the offset remains the same until we discard a part of the subspace $\mathcal{V}_l$. However, in the worst case we have to compute the rank at every iteration;

- In our view the best solution is to introduce the absent vectors by including $\operatorname{span} C_{j+1}$ corresponding to the $(j+1)$th pole lying between the anchor and the currently iterated eigenvalue. Clearly, this strategy is not restricted to the two neighboring intervals but can be used over multiple neighboring intervals.

The presented strategy is flexible to restart whenever necessary keeping the size of the basis moderate. Furthermore the restart does not require expensive solving of a generalized linear eigenvalue problem (5.2) of a dimension growing with the number of the eigenvalue.

The described local enumeration clearly does not interfere with the strategies of dealing with spectral pollution described in Section 4. The only issue is to keep track of which intervals the anchor and the currently iterated eigenvalue are contained in and correspondingly adjust the offset of the currently iterated eigenvalue from the anchor. For other strategies of dealing with spectral pollution see [1, 2].

6 Numerical Experiments

In this section we evaluate the four different variants of iterative projection method for the computation of the interior eigenvalues of (5.1). We consider a rational eigenproblem (5.1) of dimension $n = 36040$ with poles $\sigma_j := j$ and $\operatorname{rank} C_j = 2$ for $j = 1, \ldots, 9$ which is a finite element model of an elliptic cavity with 9 immersed tubes. Details about the model can be found in [1].

As a benchmark we compute all the eigenvalues in the intervals $(0, 10)$, $(7, 10)$ and $(9, 10)$ by the four restart variants of the nonlinear Arnoldi iteration

Table 1. CPU times for nonlinear Arnoldi with global restarts.

strategy	interval	total CPU [s]	nonlin. CPU [s]	init. CPU [s]
GLOBAL 1	(0,10)	511	12	157
	(7,10)	183	6.7	83.3
	(9,10)	65.8	2.6	32.2
GLOBAL 2	(0,10)	323	128	3.1
	(7,10)	126	11.2	36.7
GLOBAL 3	(0,10)	295	60.3	19.1

discussed in the last section. We report CPU times as obtained under MATLAB 2006b on a Intel Pentium D processor with 3.2 GHz and 4 GB RAM.

Notice that the way of starting and initialization after a restart with invariant subspaces of the linear eigenvalue problem (5.2) corresponding to the correct number of its largest eigenvalues guarantees the correct enumeration of the eigenvalues. Hence, no spurious eigenvalues occur in any of the three global variants of the nonlinear Arnoldi method. Though, it might happen that an eigenvalue is missed in the course of a global variant of the algorithm (in particular in GLOBAL 2), but this never occurred in our numerical tests. So, back-tracing was never necessary.

Table 1 shows the total CPU times for computing all eigenvalues with the times consumed by the nonlinear solvers and the times necessary for the initialization. For the interval $(9, 10)$ all three variants are identical since no restart was necessary for this relatively short interval which contains only eight eigenvalues. Similarly for the interval $(7, 10)$ a restart of GLOBAL 2 does not pay off since the time for initializing the algorithm (i.e. the time for solving the linear eigenproblem (5.2)) is higher than the gain of CPU time when solving smaller projected nonlinear eigenproblems after a restart. Hence, for this interval GLOBAL 2 and GLOBAL 3 are identical.

For GLOBAL 1 the initialization requires 30% to 50% of the total computing time since many intervals and eigenvalues are involved.

The total CPU times can be reduced considerably by GLOBAL 2 for the interval $(0, 10)$ since then with GLOBAL 1 a large number of initializations is necessary, and the combination of GLOBAL 1 and GLOBAL 2 in GLOBAL 3 reduces the total CPU time even further. We gain about 37% of the total CPU time by GLOBAL 2 for $(0, 10)$ and about 44% of the CPU time by GLOBAL 3. Nevertheless, as we infer from the time consumed by the nonlinear solver in GLOBAL 2, this strategy is not suitable for the computation of a large number of eigenvalues since the search subspace is growing too large.

For the interval $(7, 10)$ we do not have to determine all eigenvalues of the nonlinear eigenproblem (5.1) in the interval $(0, 7)$ which reduces the CPU time for all GLOBAL variants. The gain of GLOBAL 2 upon GLOBAL 1 is still 31% although only 2 additional solves of the linear eigenproblem (5.2) in GLOBAL 1 are necessary for restarts after a pole has been crossed.

Finally, we apply the nonlinear Arnoldi method with local restarts. Here spurious eigensolutions may appear and it is much more likely than for the global variants that eigenvalues are missed in the course of the Arnoldi algo-

rithm. The histogram in Figure 1 shows the behaviour of the locally restarted Arnoldi method for the interval $(7, 10)$. The centers of the black block bars at the bottom mark the eigenvalues whereas the height of the block indicates how often the algorithm has converged to this eigenvalue. Most of the eigenvalues are obtained only once, but there are eigenvalues to which the method converged up to seven times. This happened for instance for the eigenvalue next to 9. The white block bars indicate how often the Arnoldi method converged to eigenvalues in the corresponding interval.

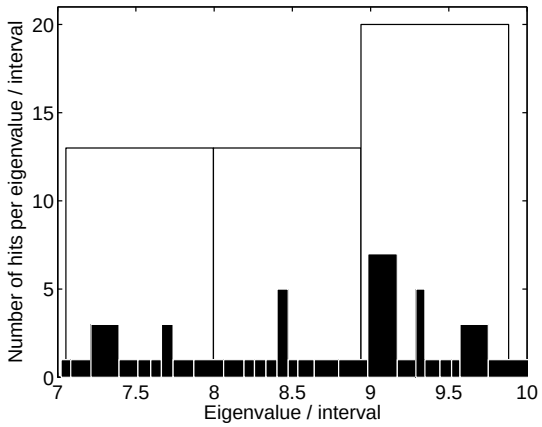

Figure 1. Number of hits of the nonlinear Arnoldi method with local restarts for the interval $(7, 10)$ per eigenvalue and interval, respectively.

The computation times for the nonlinear Arnoldi method with local restarts are shown in Table 2. Comparing to the best of the global variants we gain about 10% for the interval $(0, 10)$, about 12% for $(9, 10)$, and 20% for $(7, 10)$. It means, that for large intervals where the search subspace would have grown large or for a single interval where the effects of crossing a pole do not arise, the locally restarted variant performs slightly better. For intervals of moderate size and a small number of pole crossings the gain of the local variant is even more pronounced.

Table 2. CPU times for nonlinear Arnoldi with local restarts.

	total CPU [s]	nonlin. CPU [s]
(0,10)	266	17.0
(7,10)	100	7.3
(9,10)	58	0.3

The main advantage of the local restart strategies is that we do not require all the preceding eigenvectors to be in the search subspace. Thus we can initialize the problem by computing the eigenvalue with the smallest magnitude of the shifted linear problem (5.2). Moreover, in contrast to the first two variants

we do not encounter orthogonality problems since the search subspace remains of moderate size.

7 Conclusions

We proposed three global and one local restart strategy for the nonlinear Arnoldi method for computing a large number of eigenvalues of a nonlinear eigenvalue problem allowing for a minmax characterization of its eigenvalues. The occurrence of spurious eigenvalues and the problem of missing out eigenvalues are discussed. For the global restart strategies these do not cause problems, but they may occur if local restarts are used. However, they are usually detected if the method is not restarted too often, i.e. if the dimensions of the search spaces are not kept too small.

References

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

[2] M.M. Betcke, V. Lochmann and H. Voss. *Computing interior eigenvalues of nonlinear symmetric eigenvalue problems*. Technical Report, Institute of Numerical Simulation, Hamburg University of Technology, 2007.

[3] C. Conca, J. Planchard and M. Vanninathan. *Fluid and Periodic Structures*. Masson, Research in Applied Mathematics 24, Paris, 1995.

[4] V. Mehrmann and H. Voss. Nonlinear eigenvalue problems: A challenge for modern eigenvalue methods. *GAMM Mitteilungen*, **27**:121–152, 2004.

[5] H. Voss. Initializing iterative projection methods for rational symmetric eigenproblems. *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

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

[7] 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.