An Arnoldi Method for Nonlinear Symmetric Eigenvalue Problems

H. Voss*

1 Introduction

In this paper we consider the nonlinear eigenvalue problem

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

where $T(\lambda) \in \mathbb{R}^{n\times n}$ is a family of symmetric matrices depending on a parameter $\lambda \in J$, and $J \subset \mathbb{R}$ is an open interval which may be unbounded. As in the linear case $T(\lambda) = \lambda I - A$ a parameter λ is called an eigenvalue of $T(\cdot)$ if problem (1) has a nontrivial solution $x \neq 0$ which is called a corresponding eigenvector. We assume that the matrices $T(\lambda)$ are large and sparse.

For sparse linear eigenvalue problems most efficient methods are iterative projection methods, where approximations to the wanted eigenvalues and corresponding eigenvectors are obtained from projections of the eigenproblem to subspaces which are expanded in the course of the algorithm. Methods of this type are the Lanczos algorithm for symmetric problems, and Arnoldi's method and the Jacobi-Davidson method, e.g., for more general problems. Taking advantage of shift–and–invert techniques in Arnoldi's method one gets approximate eigenvalues closest to the shift. Ruhe [5] generalized this approach. He suggested the rational Krylov method using several shifts in one run, thus getting good approximations to all eigenvalues in a union of regions around the shifts chosen.

In some sense, Ruhe [6] and Hager and Wiberg [3], [2] generalized this approach to sparse nonlinear eigenvalue problems by nesting the linearization of problem (1) by Regula falsi and the solution of the resulting linear eigenproblem by Arnoldi's method, where the Regula falsi iteration and the Arnoldi recursion are knit together. Similarly as in the rational Krylov process they construct a sequence

*Technical University of Hamburg – Harburg, Section of Mathematics D–20171 Hamburg, Germany, voss @ tu-harburg.de

V_k of subspaces of $\mathbb{R}^n$, and at the same time they update Hessenberg matrices H_k which approximate the projection of $T(\sigma)^{-1}T(\lambda_k)$ to V_k. Here σ denotes a shift and λ_k an approximation to the wanted eigenvalue of (1). Then a Ritz vector of H_k corresponding to an eigenvalue of small modulus approximates an eigenvector of the nonlinear problem from which a (hopefully) improved eigenvalue approximation of problem (1) is obtained. Hence, in this approach the two numerical subtasks reducing the large dimension to a much smaller one and solving a nonlinear eigenproblem are attacked simultaneously.

A different approach where the two subtasks mentioned in the last paragraph are handled separately was suggested in [1] and [7]. If V_k is a subspace of $\mathbb{R}^n$ of small dimension from which we obtained approximations to an eigenvalue λ_k and corresponding eigenvector $x_k \in V_k$ then we expand V_k by a direction v_{k+1} obtained by a Jacobi-Davidson step in [1] and an Arnoldi step in [7]. Thereafter we set $V_{k+1} = [V_k, v_{k+1}]$ and solve the nonlinear projected eigenvalue problem $V_{k+1}^T T(\lambda) V_{k+1} z = 0$ to obtain new approximations λ_{k+1} and $x_{k+1} = V_{k+1} z$. This approach seems to have the disadvantage that solving a sequence of projected nonlinear eigenproblems is more expensive than solving the linear approximately projected eigenproblems for H_k in Ruhe's approach. However, it turned out in numerical examples that for large dimensions this subtask requires only a small share of the total CPU time.

In this paper we discuss a further benefit of our approach. If the underlying large problem is symmetric then the projected problems inherit the symmetry and thus can be solved more efficiently. Section 2 summarizes some useful properties of nonlinear symmetric eigenproblems and methods for solving dense nonlinear eigenproblems. Section 3 discusses the Arnoldi method from [7] with particular emphasis on symmetry, and in Section 3 we report on our numerical experience for a rational eigenvalue problem from fluid–structure interaction.

2 Solving symmetric nonlinear eigenproblems

We consider the nonlinear eigenvalue problem $T(\lambda)x = 0$ where $T(\lambda) \in \mathbb{R}^{n\times n}$ is a family of real symmetric matrices for every λ in an open real interval J which may be unbounded.

For a linear symmetric problem $Ax = \lambda x$ all eigenvalues are real, and if they are ordered by magnitude $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$ 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 the function $f(\lambda, x) := x^T T(\lambda) x$ is continuously differentiable on $J \times \mathbb{R}^n$, and that for every fixed $x \in \mathbb{R}^n \setminus \{0\}$ the real equation

$$f(\lambda, x) = 0 \tag{2}$$

has at most one solution in J. Then equation (2) implicitly defines a functional p on some subset D of $\mathbb{R}^n \setminus \{0\}$ which replaces the Rayleigh quotient in the variational characterization of eigenvalues of problem (1), and which we call the Rayleigh functional.

For nonlinear eigenvalue problems variational properties using the Rayleigh functional were proved by Duffin, Rogers, Hadeler, and Werner for overdamped

problems, i.e. if the Rayleigh functional p is defined in the entire space $\mathbb{R}^n \setminus \{0\}$, for nonoverdamped problems by Werner and the author (c.f. [8] and the literature given therein).

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 (1) is inherited from the number of the eigenvalue 0 of the matrix $T(\lambda)$.

If $\lambda \in J$ is an eigenvalue of problem (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

$$0 = \max_{V \in S_k} \min_{v \in V^1} v^T T(\lambda) v$$

where S_k denotes the set of all k–dimensional subspaces of $\mathbb{R}^n$ and $V^1 := \{v \in V : \|v\| = 1\}$ is the unit sphere in V. In this case we call λ a k-th eigenvalue of (1).

With this enumeration the following minmax characterization of the eigenvalues of the nonlinear eigenproblem (1) was proved in [8]:

Theorem 1. *Under the conditions given above the following assertions hold:*

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

$$\lambda_k = \min_{\substack{V \in S_k, \\ V \cap D \neq \emptyset}} \max_{v \in V \cap D} p(v). \tag{3}$$

The set of eigenvalues of (1) in J is at most countable.

(ii) If $\lambda \in J$ and $k \in \mathbb{N}$ such that (1) has a k-th eigenvalue $\lambda_k \in J$. Then it holds

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

The correspondence between a k-th eigenvalue λ_k of $T(\cdot)$ and the k largest eigenvalue of $T(\lambda_k)$ suggests the safeguarded iteration to determine the k-th eigenvalue of a nonlinear problem given in Algorithm 1 the convergence properties of which were proved in [9] and are collected in Theorem 2.

Algorithm 1 Safeguarded iteration

1: Start with an approximation μ_1 to the k-th eigenvalue of (1)
2: **for** $\ell = 1, 2, \ldots$ until convergence **do**
3: determine eigenvector u corresponding to the k-largest eigenvalue of $T(\mu_\ell)$
4: evaluate $\mu_{\ell+1} = p(u)$
5: **end for**

Theorem 2.

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

(ii) If $\lambda_k \in J$ is a k-th eigenvalue of (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 u in step 3. of Algorithm 1 is chosen to be an eigenvector corresponding to the k largest eigenvalue of the generalized eigenproblem $T(\mu_\ell)u = \kappa T'(\mu_\ell)u$ 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.

Since we want to determine all eigenvalues in an interval and since we approximate them one after another it is reasonable to expand the approximating space V by a direction which has high approximation potential for the next wanted eigenvector. Such direction is given by inverse iteration $v = T(\lambda)^{-1}T'(\lambda)x$ were λ and x is the current approximation to the wanted eigenvalue and eigenvector, respectively. However, solving a large linear system in each iteration step is much to expensive. Replacing v by a simplified version $v = T(\sigma)^{-1}T'(\lambda)x$ with a fixed shift σ leads to wrong convergence, namely to a solution of the linear eigenproblem $T(\sigma)x = \mu T'(\tilde{\lambda})x$ where $\tilde{\lambda}$ denotes the limit of the sequence λ_k.

Algorithm 2 Residual inverse iteration

1: Start with an approximation $x_1 \in D$ to an eigenvector of (1)
2: **for** $\ell = 1, 2, \ldots$ until convergence **do**
3: evaluate $\mu_{\ell+1} = p(x_\ell)$
4: compute the residual $r_\ell = T(\mu_{\ell+1})x_\ell$
5: solve $T(\sigma)d_\ell = r_\ell$
6: set $x_{\ell+1} = x_\ell - d_\ell$, $x_{\ell+1} = x_{\ell+1}/\|x_{\ell+1}\|$
7: **end for**

The variant of inverse iteration in Algorithm 2 called residual inverse iteration and introduced by Neumaier [4] does not have these unpleasant properties. Theorem 3 proved in [4] describes the convergence of this method.

Theorem 3. *Let $T(\lambda)$ be twice continuously differentiable. Assume that $\hat{\lambda}$ is a simple eigenvalue of problem (1), and let $\hat{x}$ be a corresponding eigenvector normalized by $\|\hat{x}\| = 1$. Then the residual inverse iteration converges for all σ sufficiently*

close to $\hat{\lambda}$*, and it holds*

$$\frac{\|x_{\ell+1}-\hat{x}\|}{\|x_\ell-\hat{x}\|}=\mathcal{O}(|\sigma-\hat{\lambda}|),\quad \textit{and}\quad |\lambda_{\ell+1}-\hat{\lambda}|=\mathcal{O}(\|x_\ell-\hat{x}\|^2).$$

3 Arnoldi's method

The convergence properties of the residual inverse iteration method suggests to expand the ansatz space V in a projection method in the following way. If $\tilde{\lambda}$ is an eigenvalue of the projected problem $V^TT(\lambda)Vz=0$ and $\tilde{x}=V\tilde{z}$ is a corresponding Ritz vector, then we choose as new direction $v=\tilde{x}-T(\sigma)^{-1}T(\tilde{\lambda})\tilde{x}$. With this expansion we may expect that the projection method has similar convergence properties as the residual inverse iteration given in Theorem 3.

In projection methods the new direction is orthonormalized against the previous ansatz vectors. Since the Ritz vector $\tilde{x}$ is contained in span V we may choose the new direction $v=T(\sigma)^{-1}T(\tilde{\lambda})\tilde{x}$. For the linear problem $T(\lambda)=\lambda I-A$ this is exactly the shifted-and-inverted Arnoldi method. If the linear system $T(\sigma)v=T(\tilde{\lambda})\tilde{x}$ is too expensive to solve for v we may choose as new direction $v=MT(\lambda)\tilde{x}$ with $M\approx T(\sigma)^{-1}$, and for the linear problem we obtain the preconditioned Arnoldi method. We therefore call the resulting iterative projection method given in Algorithm 3 nonlinear Arnoldi method.

Since we are interested in all eigenvalues in an interval J and the speed of convergence is expected to depend crucially on $|\sigma-\tilde{\lambda}|$ it will be advisable to change the shift or more generally the preconditioner M in the course of the algorithm if the convergence to the current eigenvalue becomes too slow. So actually we obtain a method which generalizes the rational Krylov method in [5], and the name nonlinear rational Krylov method would be more appropriate.

As mentioned in Section 1 Ruhe [6] designed a rational Krylov method for nonlinear eigenvalue problems which differs in some respects from our approach. Motivated by a linearization with Regula falsi the ansatz space is also expanded by $v=T(\sigma)^{-1}T(\tilde{\lambda})\tilde{x}$. However, in [6] (a linear approximation of) the projection of $T(\sigma)^{-1}T(\lambda)x=0$ to V is solved whereas we consider the projection of $T(\lambda)x=0$. Hence, in our approach symmetry properties of $T(\lambda)$ are conserved which can be used to solve the projected problem more efficiently, and which are destroyed in Ruhe's approach. Moreover, motivating the choice of the expansion v by the residual inverse iteration it is obvious that $T(\sigma)^{-1}$ can be replaced by a preconditioner $M\approx T(\sigma)^{-1}$ which is not clear in the derivation of Ruhe. Finally, the convergence result of Neumaier for the residual inverse iteration suggests a strategy when to change the shift σ.

A template for the complete preconditioned Arnoldi method for nonlinear symmetric eigenvalue problems with restarts and varying preconditioner is given in Algorithm 3. In the following we comment on some of its steps:

1. If $\inf_{x\in D}p(x)$ is contained in J then the safeguarded iteration converges globally, and the algorithm can be started with any random vector. However, with

Algorithm 3 Arnoldi Method for Nonlinear Symmetric Eigenproblems

Start with number m of first wanted eigenvalue and initial basis V, $V^TV = I$;
determine preconditioner $M \approx T(\sigma)^{-1}$, σ close to first wanted eigenvalue
$k = 1$
while $m \leq$ number of wanted eigenvalues **do**
  compute m-th eigenvalue μ and corresponding eigenvector y of $V^TT(\mu)Vy = 0$ with guarded iteration.
  determine $u = Vy$, $r_k = T(\mu)u$
  if $\|r_k\|/\|u\| < \epsilon$ **then**
    PRINT $\lambda_m = \mu$, $x_m = u$,
    if $m ==$ number of wanted eigenvalues **then**
      STOP
    end if
    $m = m + 1$
    if $(k > 1)$ & $(\|r_k\|/\|r_{k-1}\| > \text{tol})$ **then**
      choose new pole σ, and determine new preconditioner $M \approx T(\sigma)^{-1}$
    end if
    restart if necessary
    choose approximations μ and u to next eigenvalue and eigenvector
    determine $r = T(\mu)u$
    $k = 0$
  end if
  $v = Mr$
  $v = v - VV^Tv$,$\tilde{v} = v/\|v\|$, $V = [V, \tilde{v}]$
  reorthogonalize if necessary
  $k = k + 1$
end while

v an approximation to an eigenvector corresponding to the maximal eigenvector of $T(\mu)$ and μ an approximation to the smallest eigenvalue of $T(\cdot)$ one gets much faster convergence. On the other hand starting with a random vector the methods collects information on the higher eigenvectors while iterating for the first one, which speeds up the convergence of the method for higher eigenvalues. In our numerical experiments we did not observe essential differences in the overall convergence behaviour.

If $m > 1$ the algorithm can be started with an orthonormal basis of the invariant subspace of $T(\mu)$ corresponding to the m largest eigenvalues of $T(\mu)$ with μ an approximation to the first wanted eigenvalue of $T(\cdot)$.

One can profit from previous knowledge on wanted eigenvectors introducing approximations to these eigenvectors in the initial basis V as additional columns.

2. In the case study in the next section we chose M by the complete or an incomplete LU factorization of $T(\sigma)$.

3. k counts the number of iterations for fixed m. This is only needed to measure the speed of convergence and to decide in statement 13. whether a new preconditioner is recommended.

5. Any other method for determining the m-th eigenvalue of the nonlinear projected problem is suitable. However then one has to take care (for instance by Theorem 1, (ii) and Sylvester's inertia rule) that one got an approximation to the m-th eigenvalue.

13. Corresponding to Theorem 3 the residual inverse iteration with fixed pole σ converges linearly, and the contraction rate satisfies $\mathcal{O}(|\sigma-\lambda_m|)$. We therefore update the preconditioner if the convergence has become too slow. The new pole is chosen close to the eigenvalue wanted next.

16. 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. Since a restart destroys information on the eigenvectors and particularly on the one the method is just aiming at we restart only if an eigenvector has converged.

    By the proof of the minmax characterization (3) in [8] the minimum is attained by the invariant subspace $\tilde{V}$ of $T(\lambda_k)$ corresponding to the k largest eigenvalues of $T(\lambda_k)$. We therefore restart with $\tilde{V} = VZ$ where Z denotes a basis of the invariant subspace of $V^TT(\mu)V$ corresponding to the m largest eigenvalues of $V^TT(\mu)T$, or to retain more valuable information the invariant subspace corresponding to the $\tilde{m}$ largest eigenvalues where $\tilde{m}$ is slightly bigger than m.

    If $T'(\lambda)$ is positive definite then following the proof of the minmax characterization it can be shown that $\tilde{V}$ in the last paragraph can be replaced by the subspace spanned by the eigenvectors of $T(\lambda_k)u = \mu T'(\lambda_k)u$ corresponding to the k largest eigenvalues. In this case we therefore restart with VZ where Z is spanned by eigenvectors of $V^TT(\mu)Vz = \mu V^TT'(\mu)Vz$ corresponding to the m or $\tilde{m}$ largest eigenvalues of the generalized eigenproblem.

17. If we solve the projected eigenproblem by safeguarded iteration, then for fixed ν we solve the eigenproblem for the full matrix $V^TT(\nu)V$ by some standard eigensolver, and for the eigenvector y corresponding to the m largest eigenvalue we evaluate the Rayleigh functional at y, i.e. we solve $y^TV^TT(\lambda)Vy = 0$ to obtain a new approximation $\tilde{\nu}$. After convergence we have at hand the eigenvector $\tilde{y}$ corresponding to the $(m+1)$-th eigenvalue of $V^TT(\nu)V$, and we can easily obtain the root of $\tilde{y}^TV^TT(\lambda)V\tilde{y} = 0$ as an approximation μ to the next eigenvalue and the Ritz vector $u = V\tilde{y}$ as an approximation to the corresponding eigenvector. If $T'(\lambda)$ is positive definite then of course we may use the generalized eigenproblem $V^TT(\nu)Vy = \mu V^TT'(\nu)Vy$ instead, and get faster convergence.

    If no information on the next eigenvalue and eigenvector can be gained cheaply we continue with the current approximations.

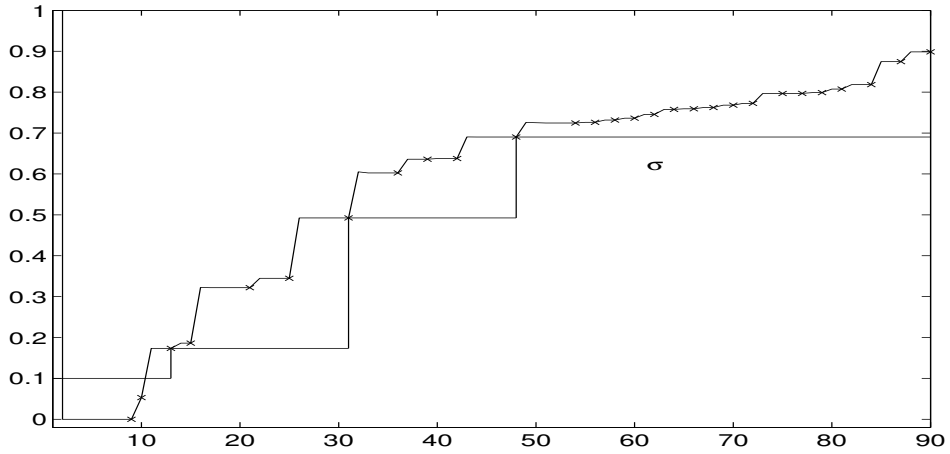

Fig. 1: Approximation history for $[0, 1)$

4 Numerical experiments

To test the Arnoldi method we consider a rational eigenvalue problem governing free vibrations of a tube bundle immersed in a slightly compressible fluid which is described in detail in [1]. A finite element model results in a rational matrix eigenproblem

$$T(\lambda)x := -Kx + \lambda Mx + \frac{\lambda}{1-\lambda}Cx = 0 \tag{4}$$

where K, M and C are symmetric matrices, K and C are positive semidefinite, and M is positive definite.

In our experiments we considered a discretization of dimensions 36040. Problem (4) has 28 eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{28} < 1$ in $[0, 1)$ and a large number of eigenvalues $1 < \tilde{\lambda}_{11} \leq \tilde{\lambda}_{12} \leq \ldots$ in $(1, \infty)$, 20 of which are contained in in the interval $(1, 3)$. Notice that the linear eigenvalue problem $Kx = \lambda Mx$ contains only 12 eigenvalues in $(0, 1)$. Thus, the rational eigenvalue problem (4) is not just a small perturbation of the linear problem which is obtained by neglecting the rational term.

The experiments were run under MATLAB 6.5 on a Pentium 4 processor with 2 GHz and 1 MB RAM. We preconditioned by the LU factorization of $T(\sigma)$, and terminated the iteration if the norm of the residual was less than 10^{-8}. By the approximation properties of the Rayleigh functional (as in the linear case the eigenvectors of $T(\cdot)$ are the stationary points of p) then the eigenvalues were determined with full accuracy.

Starting with the initial shift $\sigma = 0.1$ and a random vector the algorithm without restarts needed 90 iteration steps and a CPU time of 87.7 seconds to approximate all 28 eigenvalues in the interval $[0, 1)$, and with the tolerance tol $= 10^{-1}$ only 3 updates of the preconditioner were necessary. Fig.1 contains the approximation history and the shift σ which was used in the LU factorization. Crosses mark the iterations when the method has found an eigenvalue.

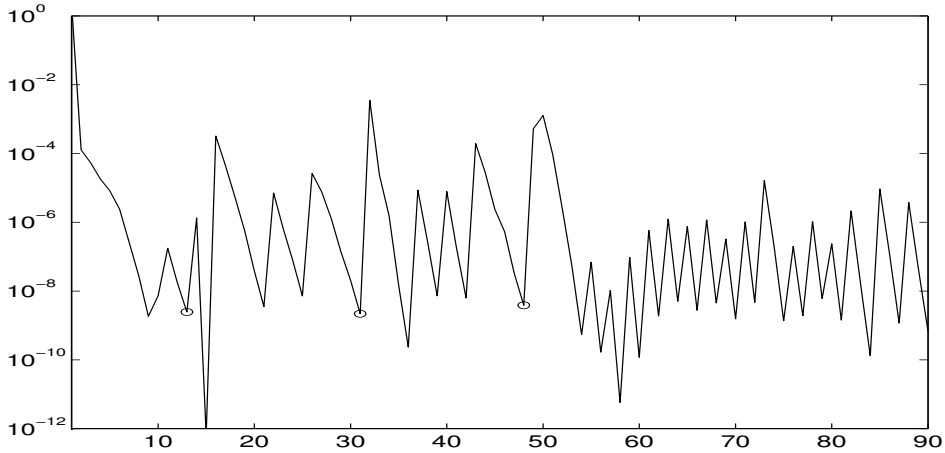

Fig. 2: Convergence history for $[0, 1)$

Fig.2 shows the convergence history. We marked by circles the 3 updates of the LU factorizations Since we did not use preinformation on eigenvectors but started with a random vector 9 iterations were required to evaluate the smallest eigenvalue, however for the remaining 27 eigenvalue the method needed only 81 iterations, i.e. an average of 3 iterations.

Starting with the known eigenvector $(1, 1, \ldots, 1)^T$ of problem (4) corresponding to the first eigenvalue $\lambda_1 = 0$ only 2 updates of the LU factorization were necessary, but the method needed 102 iterations and 101.1 seconds to approximate all 28 eigenvalues. Finally, starting with an initial space spanned by the eigenvectors corresponding to the 5 minimal eigenvalues of the linear problem $Kx = \lambda Mx$ the algorithm needed 90 iterations, 91.9 seconds and 2 LU updates, and for 10 eigenvectors it needed 86 iterations, 86.5 seconds and 1 LU update. Hence, introducing approximate eigenvectors as preinformation into the algorithm (at least in this example) does not pay.

Restarting the method if the dimension of the ansatz exceeds 50 and adding 3 additional basis vectors the method behaved similarly. It needed 2 restarts (after convergence to the 12th and the 22nd eigenvalue had appeared), 116 iterations, 97.7 seconds and 4 updates of the LU factorization, two of them just after the restarts.

For the interval $(1, 3]$ the method acted in a similar manner. Because the smallest eigenvalue in this interval is an 11th one we had to start with an ansatz space of dimension 11 (otherwise the nonlinear eigensolver could not be successful). Motivated by the proof of the minmax characterization in Theorem 1 we chose a basis of the invariant subspace of the linear eigenproblem

$$T(\sigma)x = (-K + \sigma M + \frac{\sigma}{1-\sigma}C)x = \mu(M + \frac{1}{(1-\sigma)^2}C)x = \mu T'(\sigma)x \qquad (5)$$

corresponding to the 11 largest eigenvalues. For $\sigma = 1.1$ in (5) and as initial pole the algorithm needed 99 iteration steps and 101 seconds to approximate all 30 eigenvalues in the interval $(1, 3)$. 2 updates of the LU factorization were necessary. Fig. 3

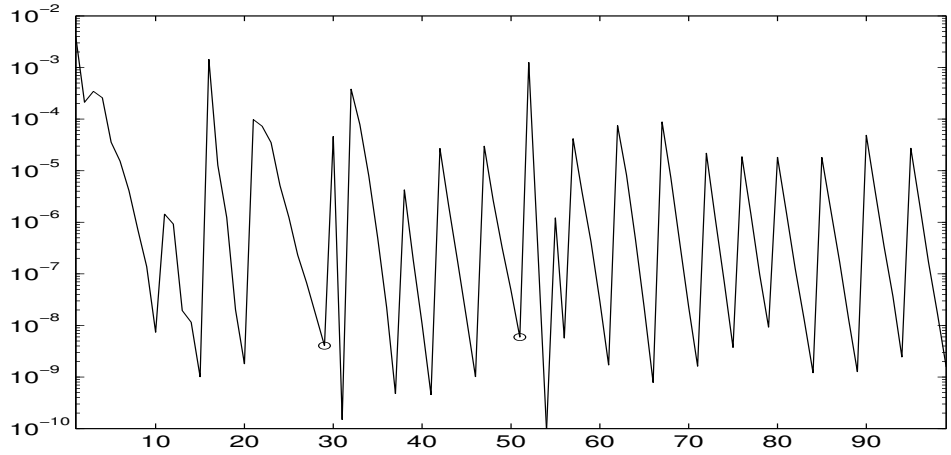

Fig. 3: Convergence history for $(1, 3)$

Table 1. *Share of solver for nonlinear eigenproblem on total CPU time*

	without restarts		with restarts	
dimension	total CPU	nonlin. evp.	total CPU	nonlin. evp
2350	10.8	4.8	6.4	1.6
9144	28.4	2.9	25.3	1.5
36040	87.7	2.5	97.7	1.2

shows the convergence history for the interval $(1, 3)$. Again the updates of LU factorizations are marked by circles. Restarts or introducing further initial information into the start space did not change the convergence behaviour essentially.

We stressed the fact that solving the projected nonlinear eigenproblems requires only a small portion of the total CPU time for large dimensions. Table 1 shows the total CPU time for computing all 28 eigenvalues in $[0, 1)$ for 3 FEM models of different dimensions without and with restarts if the dimension of the projected problems exceeded 50.

Our final experiment demonstrates, that the method works if it is preconditioned by less accurate preconditioners than exact LU factorization, although the convergence is much slower. Preconditioning by incomplete LU factorization with drop tolerance 0.01 and restarting and updating the preconditioner whenever an eigenvalue has converged the algorithm finds all 28 eigenvalues in the $[0, 1)$ requiring 1722 iterations and 1112 seconds, 4% of which are required to solve the projected nonlinear eigenproblems and 2.5% to update the preconditioner.

The problem was also solved by the Jacobi–Davidson method [1] and the rational Krylov method [6]. Jacobi–Davidson found all eigenvalue in $[0, 1)$ and $(1, 3)$ requiring 1006.6 and 758.9 seconds, respectively. The rational Krylov method needed 847.0 seconds to find all eigenvalues in $[0, 1)$. Due to sensitive dependence on the initial approximation μ and the shift σ it was only able to locate 2 or 3 eigenvalues in $(1, 3)$ before it diverged in several tests with different μ and σ.

Bibliography

[1] T. Betcke and H. Voss. A Jacobi–Davidson–type projection method for nonlinear eigenvalue problems. Technical Report 47, Dept.Math., TU Hamburg-Harburg, 2002. submitted to Future Generation Computer Systems.

[2] P. Hager. *Eigenfrequency Analysis. FE-Adaptivity and a Nonlinear Eigenvalue Problem.* PhD thesis, Chalmers University of Technology, Göteborg, 2001.

[3] P. Hager and N.E. Wiberg. The rational Krylov algorithm for nonlinear eigenvalue problems. In B.H.V. Topping, editor, *Computational Mechanics for the Twenty-First Century*, pages 379 – 402. Saxe–Coburg Publications, Edinburgh, 2000.

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

[5] A. Ruhe. Rational Krylov, a practical algorithm for large sparse nonsymmetric matrix pencils. *SIAM J.Sci.Comput.*, 19(5):1535–1551, 1998.

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

[7] H. Voss. An Arnoldi method for nonlinear eigenvalue problems. Technical Report 56, Dept.Math., TU Hamburg-Harburg, 2003. submitted to BIT Numerical Mathematics.

[8] 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.

[9] H. Voss and B. Werner. Solving sparse nonlinear eigenvalue problems. Technical Report 82/4, Inst. f. Angew. Mathematik, Univ. Hamburg, 1982.