BIT
200?, Vol. ??, No. ?, pp. ???–???

0006-3835/98/3804-0101 $12.00
© Swets & Zeitlinger

AN ARNOLDI METHOD FOR NONLINEAR EIGENVALUE PROBLEMS

H. Voss

Technical University of Hamburg – Harburg, Section of Mathematics D–21071 Hamburg, Germany. email: voss@tu-harburg.de

Abstract.

For the nonlinear eigenvalue problem $T(\lambda)x = 0$ we propose an iterative projection method for computing a few eigenvalues close to a given parameter. The current search space is expanded by a generalization of the shift-and-invert Arnoldi method. The resulting projected eigenproblems of small dimension are solved by inverse iteration. The method is applied to a rational eigenvalue problem governing damped vibrations of a structure.

AMS subject classification: 65F15, 49G05.

Key words: nonlinear eigenvalue problem, iterative projection method, residual inverse iteration

1 Introduction

In this paper we consider the nonlinear eigenvalue problem

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

where $T(\lambda) \in \mathbb{C}^{n\times n}$ is a family of matrices depending on a parameter $\lambda \in D \subset \mathbb{C}$. As in the linear case $T(\lambda) = \lambda I - A$ a parameter λ is called an eigenvalue of $T(\cdot)$ if problem (1.1) has a nontrivial solution $x \neq 0$ which is called a corresponding eigenvector. We assume that the matrices $T(\lambda)$ are large and sparse.

Iterative projection methods where approximations to the wanted eigenvalues and corresponding eigenvectors are obtained from projections to subspaces which are expanded in the course of the algorithm are very efficient for linear sparse eigenproblems. 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 [8] 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 [7], [9] and Hager and Wiberg [3], [2] generalized the rational Krylov approach to sparse nonlinear eigenvalue problems by nesting the linearization of problem (1.1) 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 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 (1.1). 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 (1.1) 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.

In this paper we suggest an iterative projection method for the nonlinear eigenproblem where the two subtasks mentioned in the last paragraph are handled separately. We order the eigenvalues in some way and determine them one after another. If V_k denotes the subspace of $\mathbb{C}^n$ of small dimension k constructed in the course of the algorithm we solve the projected nonlinear eigenvalue problem $V_k^H T(\lambda)V_k z = 0$ of dimension k by a dense solver to obtain approximations λ_k and $x_k = V_k z$ to an eigenvalue and eigenvector, respectively. Thereafter we expand the ansatz space V_k to $V_{k+1} = [V_k, v_{k+1}]$ and repeat the projection step. Similarly as in the Jacobi–Davidson method the direction v_{k+1} is chosen such that $x_k + \alpha v_{k+1}$ for some $\alpha \in \mathbb{C}$ has a high approximation potential for the eigenvector we are just aiming at.

It is well known that inverse iteration converges quadratically to simple eigenvalues. Therefore, the expansion $v_{k+1} = T(\lambda_k)^{-1}T'(\lambda_k)x_k$ would be a reasonable choice. However, in this case we would have to solve a high dimensional linear system in every iteration step where the coefficient matrix varies. The way out is the residual inverse iteration suggested by Neumaier [5], and given by $x_{k+1} = x_k - T(\sigma)^{-1}T(\lambda_k)x_k$ where σ is a fixed shift (not to far away from the eigenvalue targeted at) and λ_k is the current approximation.

Although derived in completely different ways the rational Krylov method and the Arnoldi method suggested here are closely related to each other. In particular both methods employ the same vector to expand the ansatz space. However, there are some differences. An advantage of Ruhe's approach is the fact that one only needs a procedure that yields the product $T(\lambda)x$ for given λ and given x whereas our methods needs the explicit form of the projected family of matrices $V_k^H T(\lambda)V_k$ to which we apply a nonlinear eigensolver. On the other hand, our method is much faster since the rational Krylov methods needs a great deal of inner iterations to get an accurate linear approximation to the nonlinear eigenproblem. Moreover, if the matrices $T(\lambda)$ are symmetric and the eigenvalues can be characterized as minmax values of a Rayleigh functional then this property is inherited by the projected matrices and can be utilized to solve the projected problems very efficiently. This is discussed further in [10], and in [1] where the subspaces were expanded by a Jacobi–Davidson type approach.

Our paper is organized as follows: Section 2 derives the nonlinear Arnoldi method, and discusses a strategy how to update the shift σ when the convergence becomes to slow and a restart method to reduce the computational cost for solving the projected eigenproblems as the subspaces expand. Section 3 reviews methods for solving dense nonlinear eigenproblems, and Section 4 contains numerical experiments demonstrating the efficiency of the Arnoldi method when applied to a rational eigenvalue problem governing damped vibrations of a structure.

2 Arnoldi's method for nonlinear eigenproblems

Iterative projection methods (Lanczos, Arnoldi, Jacobi-Davidson, e.g.), where approximations of the wanted eigenvalues and corresponding eigenvectors are obtained from projections to subspaces which are expanded in the course of the algorithm a very efficient for linear sparse eigenproblems. In this section we propose a method of this type for the nonlinear eigenvalue problem

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

where $T(\lambda) \in \mathbb{C}^{n\times n}$ is a family of matrices the elements of which are continuously differentiable functions of $\lambda \in D \subset \mathbb{C}$.

We determine eigenvalues one after another and expand the approximating space V by a direction which has high approximation potential for the next wanted eigenvector. A suitable direction is given by inverse iteration $v = T(\lambda)^{-1}T'(\lambda)x$ where λ and x is the current approximation to the wanted eigenvalue and eigenvector, respectively. Inverse iteration is known to converge quadratically to simple eigenvalues, and for symmetric eigenproblems it converges even cubically. Its drawback however is that it is much to expensive for large problems since in every iteration step one has to solve a linear system where the system matrices vary.

Replacing v by a simplified version $v = T(\sigma)^{-1}T'(\lambda)x$ with a fixed shift σ leads to wrong convergence. It is easily seen that this iteration converges to an eigenpair of the linear problem $T(\sigma)x = \gamma T'(\tilde{\lambda})x$ ($\gamma \neq 0$ and $\tilde{\lambda}$ depending on the normalization condition) from which we cannot recover an eigenpair of the nonlinear problem (1.1).

Algorithm 2.1 Residual inverse iteration

```
Let $e$ be a normalization vector and start with an approximations $\sigma$ and $x_1$
  to an eigenvalue and corresponding eigenvector of (2.1) such that $e^H x_1 = 1$
for $\ell = 1, 2, \dots$ until convergence do
    solve $e^H T(\sigma)^{-1} T(\lambda_{\ell+1}) x_\ell = 0$ for $\lambda_{\ell+1}$
    compute the residual $r_\ell = T(\lambda_{\ell+1}) x_\ell$
    solve $T(\sigma) d_\ell = r_\ell$ for $d_\ell$
    set $y_{\ell+1} = x_\ell - d_\ell$
    normalize $x_{\ell+1} = y_{\ell+1}/e^H y_{\ell+1}$
end for
```

Neumaier [5] introduced a variant of inverse iteration given in Algorithm 2.1 which he called residual inverse iteration and which does not have this unpleasant property. The update of the eigenvalue approximation in step 3. is motivated by the fact that $e^H T(\sigma)^{-1}$ is an approximation to a left eigenvector of $T(\sigma)$ corresponding to the smallest eigenvalue in modulus obtained by one step of inverse iteration. Theorem 2.1 proved in [5] describes the convergence of this method.

THEOREM 2.1. *Let $T(\lambda)$ be twice continuously differentiable. Assume that $\hat\lambda$ is a simple eigenvalue of problem (2.1), and let $\hat x$ be a corresponding eigenvector normalized by $e^H\hat x = 1$ where $e \in \mathbb{C}^n$ denotes a fixed vector. Then the residual inverse iteration converges for all σ sufficiently close to $\hat\lambda$, and it holds*

$$\frac{\|x_{\ell+1}-\hat x\|}{\|x_\ell - \hat x\|} = \mathcal{O}(|\sigma - \hat\lambda|), \quad \textit{and} \quad |\lambda_{\ell+1} - \hat\lambda| = \mathcal{O}(\|x_\ell - \hat x\|). \tag{2.2}$$

The convergence properties of the residual inverse iteration method suggest to expand the ansatz space V in a projection method in the following way. If $\tilde\lambda$ is an eigenvalue of the projected problem $V^H T(\lambda) V z = 0$ 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 2.1.

In projection methods the new direction is orthonormalized against the previous ansatz vectors. Since the Ritz vector $\tilde x$ is contained in the span of V we may choose the new direction $v = T(\sigma)^{-1}T(\tilde\lambda)\tilde x$ as well. For the linear problem $T(\lambda) = A - \lambda B$ this is exactly the Cayley transform with pole σ and zero $\tilde\lambda$, and since $(A-\sigma B)^{-1}(A - \tilde\lambda B) = I + (\lambda - \sigma)(A - \sigma B)^{-1}B$ and Krylov spaces are shift-invariant the resulting projection method expanding V by v is nothing else but the shift-and-invert 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 $v = MT(\tilde\lambda)\tilde x$ with $M \approx T(\sigma)^{-1}$, and for the linear problem we obtain an inexact Cayley transform or a preconditioned Arnoldi method. We therefore call the resulting iterative projection method given in Algorithm 2.2 nonlinear Arnoldi method.

Since we are interested in all eigenvalues in some region and since 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 for linear problems in [8], and the name nonlinear rational Krylov method would be appropriate, too. However, since Ruhe [9] already introduced a rational Krylov method for nonlinear problems which differs from our method quite a bit we prefer the name nonlinear Arnoli method. We will comment on the differences of Ruhe's and our approach at the end of this section.

A template for the preconditioned Arnoldi method for nonlinear eigenvalue problems with restarts and varying preconditioner is contained in Algorithm 2.2. In the following we comment on some of its steps.

1. Here preinformation such as known approximate eigenvectors of problem

Algorithm 2.2 Nonlinear Arnoldi Method

```
1: start with an initial pole $\sigma$ and an initial basis $V$, $V^H V = I$;
2: determine preconditioner $M \approx T(\sigma)^{-1}$, $\sigma$ close to first wanted eigenvalue
3: $k = 1$
4: while $m \leq$ number of wanted eigenvalues do
5:    compute appropriate eigenvalue $\mu$ and corresponding eigenvector $y$ of the
      projected problem $T_V(\mu)y := V^H T(\mu)Vy = 0$.
6:    determine Ritz vector $u = Vy$ and residual $r_k = T(\mu)u$
7:    if $\|r_k\|/\|u\| < \epsilon$ then
8:       PRINT $\lambda_m = \mu$, $x_m = u$,
9:       if $m ==$ number of wanted eigenvalues then
10:         STOP
11:      end if
12:      $m = m + 1$
13:      if $(k > 1)$ & $(\|r_{k-1}\|/\|r_k\| >$ tol$)$ then
14:         choose new pole $\sigma$
15:         determine new preconditioner $M \approx T(\sigma)^{-1}$
16:      end if
17:      restart if necessary
18:      choose approximations $\mu$ and $u$ to next eigenvalue and eigenvector
19:      determine residual $r = T(\mu)u$
20:      $k = 0$
21:   end if
22:   $v = Mr$
23:   $v = v - VV^H v$ ,$\tilde{v} = v/\|v\|$, $V = [V, \tilde{v}]$
24:   reorthogonalize if necessary
25:   update projected problem $T_V(\mu) = V^H T(\mu)V$
26:   $k = k + 1$
27: end while
```

(2.1) or eigenvectors of contiguous problems can be introduced into the algorithm.

If no information on eigenvectors is at hand, and we are interested in eigenvalues close to the parameter $\sigma \in D$, one can choose an initial vector at random, execute a few Arnoldi steps for the linear eigenproblem $T(\sigma)u = \theta u$ or $T(\sigma)u = \theta T'(\sigma)u$, and choose V by orthogonalizing eigenvectors corresponding to small eigenvalues in modulus. Starting with a random vector without this preprocessing usually will yield a value μ in step 5. which is far away from σ and will avert convergence.

2. In our numerical examples we used the LU factorization of $T(\sigma)$ if this could be determined inexpensively and otherwise an incomplete LU factorization, but every other preconditioner is fine.

3. k counts the number of iterations for fixed m. This is only needed to mea-

sure the speed of convergence and to decide whether a new preconditioner is recommended in condition 13.

4. Every other stopping criterion can replace the requirement to determine m eigenvalues.

5. Since the dimension of the projected problem usually is quite small one can solve it by inverse iteration or by residual inverse iteration. However, differently from the linear case there is no easy way to inhibit the algorithm to converge to the same eigenvalue repeatedly. This is the crucial point in the algorithm, and we discuss it in detail in Section 3.

13. Corresponding to Theorem 2.1 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 measured by the quotient of the last two residual norms has become too slow.

    In our numerical examples it happened that the condition in step 7. was fulfilled in the first step after having increased m. In this case the quotient of the last two residual norms does not say anything about the speed of convergence, and we do not update the preconditioner.

14. The new pole should not be chosen to close to an eigenvalue of $T(\cdot)$ because this would hamper the construction of the preconditioner. A general strategy cannot be given, but the proper way to choose a new pole depends on the problem under consideration and on the method in step 5. for solving the projected problem.

    For instance, in Section 4. we consider a rational eigenproblem governing the damped vibrations of a structure. Due to the symmetry properties of eigenvalues and eigenvectors it is reasonable to determine only the eigenvalues with negative imaginary part, and to compute them one after another with decreasing imaginary part. In this case the new pole σ can be chosen as a moderate multiple of the last converged eigenvalue, e.g. $\sigma = 1.05\lambda_{m-1}$.

17. 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 just converged.

    Since some of the solvers of the nonlinear projected eigenproblems in 5. take advantage of some enumeration of the eigenvalues it is natural to keep the eigenvectors that have been converged in the course of the algorithm. Otherwise this enumeration would be perturbed. We therefore continue with an orthonormal basis of $X_m := \text{span}\{x_1, \dots, x_m\}$. If an approximation to an eigenvector wanted next is obtained cheaply (cf. 18.) we add it to X_m.

18. Some of the eigensolvers discussed in Section 3. can be used to get approximations to the eigenvector and eigenvalue wanted next. In this case we continue with these approximations. If no information on the next eigenvalue and eigenvector can be gained cheaply we continue with the current approximations.

23. v is orthogonalized with respect to the current search space V by classical Gram–Schmidt. In our implementation we replaced it by modified Gram–Schmidt for stability reasons.

24. If in statement 23. the norm of v is reduced in the (classical or modified) Gram–Schmidt process by more than a modest factor κ, say $\kappa = 0.25$, then it is appropriate to repeat the Gram–Schmidt method once.

25. Often problem (1.1) has the form $T(\lambda) = \sum_{j=1}^N f_j(\lambda)C_j$ with differentiable complex functions f_j and fixed matrices $C_j \in \mathbb{C}^{n\times n}$. Then the projected problem has the form

$$T_{V_k}(\lambda) = \sum_{j=1}^N f_j(\lambda)V_k^H C_j V_k =: \sum_{j=1}^N f_j(\lambda)C_{j,k},$$

and the matrices $C_{j,k}$ can be updated according to

$$C_{j,k} = \begin{pmatrix} C_{j,k-1} & V_{k-1}^H C_j \tilde{v} \\ \tilde{v}^H C_j V_{k-1} & \tilde{v}^H C_j \tilde{v} \end{pmatrix}.$$

Some comments on the relations between Ruhe's approach and ours are in order. Ruhe derived his method from Langrange interpolation

$$T(\lambda) = \frac{\lambda - \mu_k}{\sigma - \mu_k} T(\sigma) + \frac{\lambda - \sigma}{\mu_k - \sigma} T(\mu_k) + \text{higher order terms}$$

where σ is a fixed parameter und μ_k is close to the wanted eigenvalue. Neglecting the higher order terms one obtains the linear eigenproblem

$$T(\sigma)^{-1}T(\mu_k)w = \theta w \quad \text{where } \theta = \frac{\lambda - \mu_k}{\lambda - \sigma} \tag{2.3}$$

which predicts a new approximation $\mu_{k+1} = \mu_k + \theta(\mu_k - \sigma)/(1-\theta)$ to an eigenvalue of the nonlinear problem. Applying for each $k = 1, 2, \dots$ one step of Arnoldi's method to the linear problem (2.3) and updating the Hessenberg matrix in the Arnoldi process as μ_k varies one obtains a sequence of linear eigenvalue problems which approximate the projection of the nonlinear problem $T(\sigma)^{-1}T(\lambda)x = 0$ to a subspace of small dimension. To improve this approximation one has to introduce inner iterations based on the regula falsi method for every k.

An advantage of Ruhe's approach upon ours is the fact that the method accepts a function that evaluates the residual $r_k = T(\sigma)^{-1}T(\mu_k)x_k$ for given μ_k and x_k

but it does not need the explicit form of a projected problem. On the other hand the inner iterations which are necessary to adjust the linear approximation to the nonlinear problem $T(\sigma)^{-1}T(\lambda)x = 0$ and which are not needed in our approach are very expensive.

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 from the derivation of Ruhe. Further, the convergence result of Neumaier for the residual inverse iteration suggests a strategy when to change the shift σ, and finally, if the matrices $T(\lambda)$ are symmetric and the eigenvalues can be characterized as minmax values of a Rayleigh functional then this property is inherited by the projected matrices and can be utilized to solve the projected problems very efficiently.

3 Solving the projected problems

A crucial point in iterative projection methods for general nonlinear eigenvalue problems is to inhibit the method to converge to the same eigenvalue repeatedly. For linear eigenvalue problems this is no problem since Krylov subspace solvers construct the basis of the ansatz space without employing approximate eigenvalues, and 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. For nonlinear problems a similar normal form does not exist.

If $T(\lambda)$ is a family of real symmetric matrices and D is a real interval such that the eigenvalues of problem (1.1) can be characterized as minmax value of a Rayleigh functional (cf. [11]) then there is a close relation between the nonlinear problem (1.1) and the symmetric linear eigenproblem

$$T(\lambda)u = \mu u \qquad (\text{or } T(\lambda)u = \mu T'(\lambda)u \text{ if } T'(\lambda) \text{ is positive definite}). \tag{3.1}$$

In particular, if $\hat\lambda \in J$ is an eigenvalue of (1.1) then $\mu = 0$ is an eigenvalue of (3.1) with $\lambda = \hat\lambda$, and if $\mu = 0$ is the m-largest eigenvalue of (3.1) then the so called safeguarded iteration in Algorithm 3.1 converges locally and quadratically (or even cubically) to $\hat\lambda$.

Algorithm 3.1 Safeguarded iteration

```
Start with an approximation $\mu_1$ to the $m$-th eigenvalue of (1.1)
for $\ell = 1, 2, \dots$ until convergence do
    determine eigenvector $u$ corresponding to $m$-largest eigenvalue of (3.1)
    solve $u^H T(\mu_{\ell+1})u = 0$ for $\mu_{\ell+1}$
end for
```

Since there is at most one m-th eigenvalue in D this result suggests to solve the projected problem by safeguarded iteration. Arnoldi's method for symmetric nonlinear eigenproblems with safeguarded iteration as inner iteration to solve the projected problems was discussed in [10].

In the general case the following strategy is similar to safeguarded iteration. Assume that we want to determine all eigenvalues of problem (1.1) in the vicinity of a given parameter $\sigma_0 \in D$, and that already $m-1$ eigenvalues closest to σ_0 have been determined. Assume that $\tilde{\mu}$ is an approximation to the eigenvalue wanted next.

A first order approximation of problem (1.1) is

$$T(\lambda)x \approx (T(\tilde{\mu}) - \theta T'(\tilde{\mu}))x = 0, \ \theta = \tilde{\mu} - \lambda. \tag{3.2}$$

This suggests the method of successive linear problems introduced by Ruhe [6].

Algorithm 3.2 Method of successive linear problems

```
Start with an approximation $\mu_1$ to the $m$-th eigenvalue of (1.1)
for $\ell = 1, 2, \dots$ until convergence do
  solve the linear eigenproblem $T(\mu_\ell)u = \theta T'(\mu_\ell)u$
  choose the eigenvalue $\theta$ such $|\sigma_0 - (\mu_\ell - \theta)|$ is the $m$–smallest among the
  eigenvalues
  $\mu_{\ell+1} = \mu_\ell - \theta$
end for
```

Of course this method is not appropriate for the sparse problem (1.1), but the dimension of the projected problem in step 5. usually is quite small, and every standard solver for dense eigenproblems applies.

Quite often the nonlinear eigenvalue problem under consideration is a (small) perturbation of a linear eigenvalue problem. In the next section we will consider a finite element model of a vibrating structure with nonproportional damping. Using a viscoelastic constitutive relation to describe the behaviour of a material in the equations of motions yields a rational eigenvalue problem for the case of free vibrations. A finite element model obtains the form

$$T(\omega) := \left(\omega^2 M + K - \sum_{j=1}^{J} \frac{1}{1 + b_j \omega} \Delta K_j \right) x = 0 \tag{3.3}$$

where M is the consistent mass matrix, K is the stiffness matrix with the instantaneous elastic material parameters used in Hooke's law, J denotes the number of regions with different relaxation parameters b_j, and ΔK_j is an assemblage of element stiffness matrices over the region with the distinct relaxation constants. The real part of an eigenvalue is the exponential rate with which the motion described by the corresponding eigenvector x decays. The imaginary part is the (damped) angular frequency with which the motion described by x oscillates

It is well known that often the eigenmodes of the damped and the undamped problem do not differ very much although the eigenvalues do. Therefore, in step 5. of the algorithm it is reasonable to determine an eigenvector y of the undamped and projected problem $(\omega^2 V^H M V + V^H K V)y = 0$ corresponding to the m-largest eigenvalue ω_m^2, determine an approximate eigenvalue $\tilde{\omega}$ of the nonlinear projected problem from the complex equation $y^H V^H T(\omega) V y = 0$ or $e^H V^H T(\sigma)^{-1} T(\omega) V y = 0$, and correct it by (residual) inverse iteration.

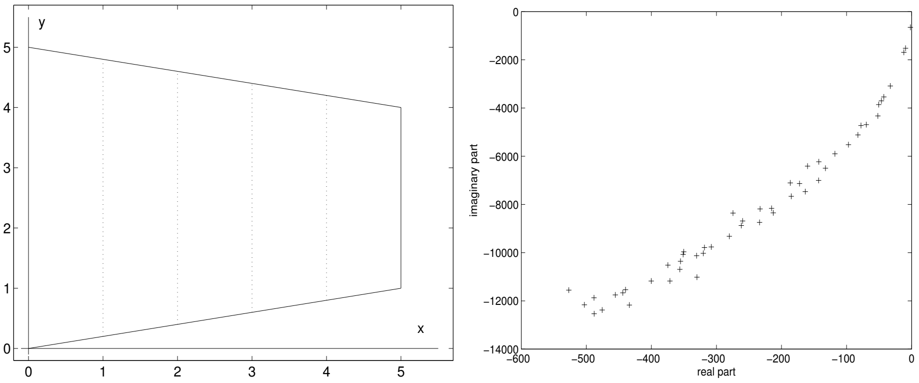

Fig. 1: Trapezoidal plate / Eigenvalues

4 Numerical experiments

To test the Arnoldi method we consider the rational eigenvalue problem governing damped vibrations of a structure which was mentioned in the last section.

A trapezoidal plate $\{(x, y) : 0 \le x \le 5,\ 0.2x \le y \le 5 - 0.2x\}$ (cf. Fig. 1) is subject to plane stress, and is clamped at the side given by $x = 0$. The instantaneous Young's modulus is set to $E = 2.10*10^{11}$, the instantaneous Poisson's rate is $\nu = 0.33$, and the density is set to $\rho = 7800$. For the nonproportional damping we use in addition the following parameters, $\Delta\nu = 0.28$, and $\Delta E = 7 * 10^{10}$ for $0 < x < 1$, $\Delta E = 6 * 10^{10}$ for $1 < x < 2$, $\Delta E = 5 * 10^{10}$ for $2 < x < 3$, $\Delta E = 4 * 10^{10}$ for $3 < x < 4$, and $\Delta E = 3 * 10^{10}$ for $4 < x < 5$. The relaxation constant is set to $b = 2 * 10^{-5}$.

Discretizing this problem by linear Lagrangean elements we obtained the rational eigenproblem (3.3) of dimension 9376. For symmetry reasons we determined only eigenvalues with negative imaginary part, and we computed 50 of them one after another with decreasing imaginary part. The nonlinear projected eigenproblems were solved by inverse iteration with an initial guess obtained from the corresponding undamped projected problem as explained at the end of Section 3.

The experiments were run under MATLAB 6.5 on a Pentium 4 processor with 2 GHz and 1 GB RAM. We preconditioned by the LU factorization of $T(\sigma)$, and terminated the iteration if the norm of the residual was less than 10^{-6}.

The algorithm without restarts needed 258 iteration steps, i.e. an average of 5 iterations per eigenvalue, and a CPU time of 559.6 seconds to approximate all 50 eigenvalues with maximal negative imaginary part. With the tolerance tol $= 10^{-1}$ in step 13. of the Arnoldi algorithm only 2 updates of the preconditioner were necessary. Fig.2 contains the convergence history.

The dominant share of the CPU time, namely 359.4 seconds, was consumed by the solver of the nonlinear eigenproblems. Fig. 3 demonstrates the development of the CPU times of the entire iteration and of the share of the nonlinear

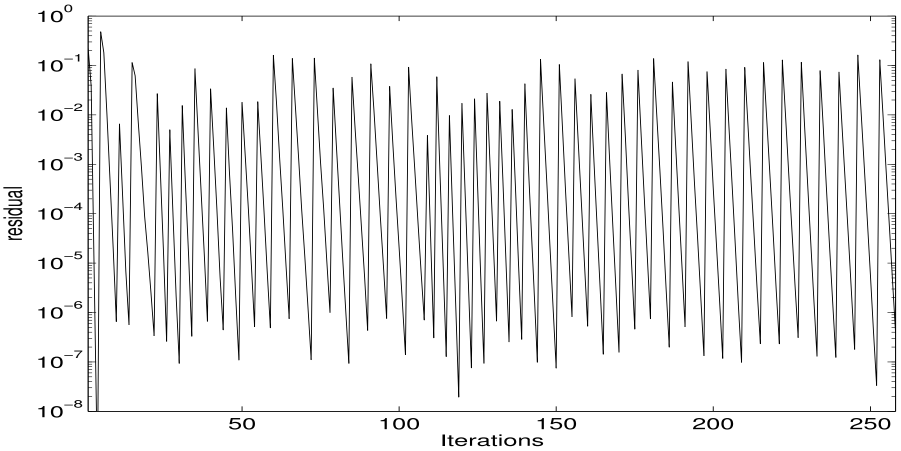

Fig. 2: Convergence history without restarts

eigensolvers. It demonstrates the necessity of restarts since the superlinear time consumption is mainly caused by the eigensolver.

We restarted the Arnoldi process if the dimension of the ansatz space exceeded 80. Again all 50 eigenvalues were found by the method requiring 272 iterations and 197.7 seconds where 19.4 seconds were needed to solve the nonlinear projected eigenproblems and 12.8 seconds to determine the 6 LU factorizations necessary in this run. Fig. 4 contains the convergence history. It looks very similar to the one without restarts, however, it demonstrates that after a restart the speed of convergence is reduced. Typically, as for the method without restarts an average of 5 iterations was needed to find the next eigenvalue, after a restart, however, an average of 14 iterations was needed to gather enough information

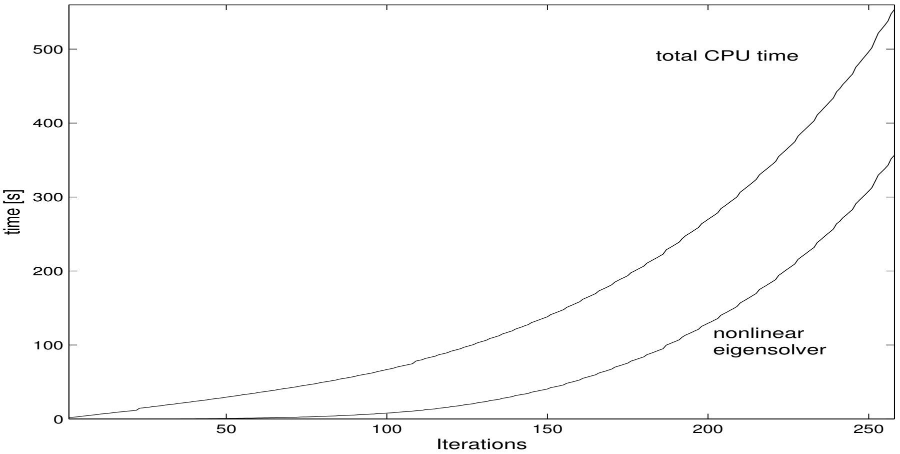

Fig. 3: Development CPU time consumption without restarts

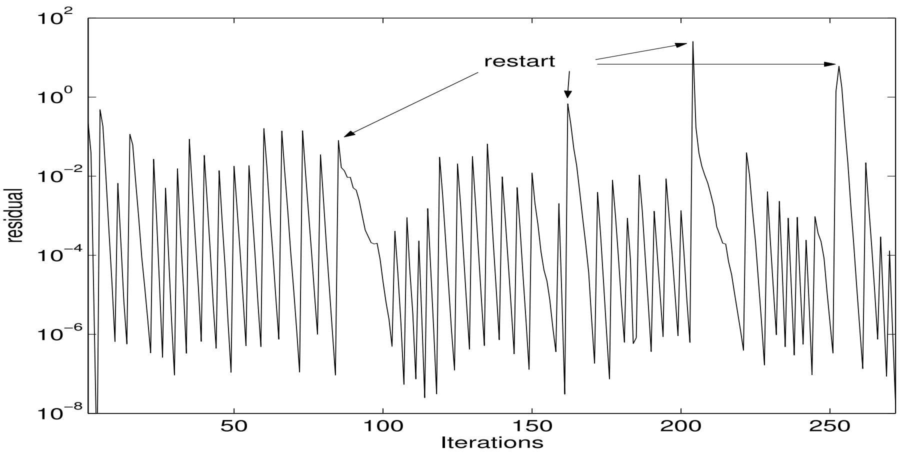

Fig. 4: Convergence history with restarts

about an ansatz space and to converge.

Fig.5 demonstrates the history of CPU time consumption for the entire Arnoldi method, the nonlinear eigensolver and the LU factorizations where we terminated the iteration if the norm of the residual was less than 10^{-4} (for 10^{-6} the graph for the nonlinear eigensolvers and the LU factorization could not have been distinguished from the x-axis).

The nonlinear Arnoldi algorithm showed a similar behaviour if the projected eigenproblems are solved by the method of successive linear problems. To determine 20 eigenvalues in the vicinity of $\sigma_0 = -200 - 2000i$ it needed 101 iterations and 82.4 seconds, and the LU factorization was updated once.

For comparison we determined 50 eigenvalues with maximal negative imag-

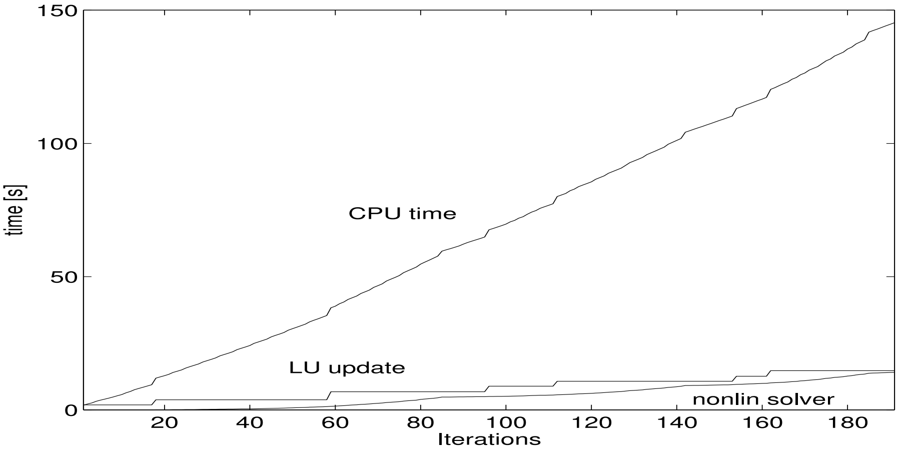

Fig. 5: Development CPU time consumption with restarts

inary part of the nonlinear eigenproblem (3.3) by the rational Krylov method introduced by Ruhe [9] and worked out in detail by Hager [2] and Jarlebring [4]. The method found all 50 eigenvalue requiring 2160 (inner and outer) iterations, a total CPU time of 1323.5 seconds and 8 LU updates.

Acknowledgement The author gratefully acknowledges the assistance of Elias Jarlebring who provided the MATLAB implementation of the rational Krylov method.

REFERENCES

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. E. Jarlebring. Krylov methods for nonlinear eigenvalue problems. Master's thesis, Royal Institute of Technology, Dept.Numer.Anal.Comput.Sci., Stockholm, 2003.
5. A. Neumaier. Residual inverse iteration for the nonlinear eigenvalue problem. *SIAM J.Numer.Anal.*, 22:914–923, 1985.
6. A. Ruhe. Algorithms for the nonlinear eigenvalue problem. *SIAM J. Numer. Anal.*, 10:674 – 689, 1973.
7. A. Ruhe. Computing nonlinear eigenvalues with spectral transformation Arnoldi. *ZAMM*, 76:S2: 17 – 20, 1996.
8. A. Ruhe. Rational Krylov, a practical algorithm for large sparse nonsymmetric matrix pencils. *SIAM J.Sci.Comput.*, 19(5):1535–1551, 1998.
9. A. Ruhe. A rational Krylov algorithm for nonlinear matrix eigenvalue problems. *Zapiski Nauchnyh Seminarov POMI*, 268:176 – 180, 2000.
10. H. Voss. An Arnoldi method for nonlinear symmetric eigenvalue problems. Technical Report 57, Dept.Math., TU Hamburg-Harburg, 2003. Submitted to Proceedings of the SIAM Conference on Applied Linear Algebra, Williamsburg, VA, 2003.
11. 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.