BIT Numerical Mathematics **44:** 387–401, 2004.

AN ARNOLDI METHOD FOR NONLINEAR EIGENVALUE PROBLEMS ⋆

H. VOSS[1]

[1] *Section of Mathematics, Hamburg University of Technology, 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 (2000): 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 can be 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, for more general problems. Taking advantage of shift-and-invert techniques in Arnoldi's method one gets approximate eigenvalues closest to the shift. Ruhe [13] 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 [12, 14] and Hager and Wiberg [5, 4] generalized the rational Krylov approach to sparse nonlinear eigenvalue problems by nesting the

⋆ Received May 2003. Accepted January 2004. Communicated by Axel Ruhe.

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 the projected nonlinear eigenproblems 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 search 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.

Inverse iteration with variable shifts converges quadratically to simple eigenvalues (cf. [11]). 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 [10], 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 a completely different way, the rational Krylov method can be interpreted as an iterative projection method (cf. [7]), and it is closely related to the Arnoldi method suggested here. Both methods expand the search space by the direction suggested by the residual inverse iteration. However, considering the projected problem $V_k^H T(\sigma)^{-1}T(\lambda)V_k z = 0$ for some fixed shift σ in Ruhe's approach potential symmetry properties (like symmetry or Hamiltonian structure) of the original problem are destroyed which may be useful when solving the projected problem. For instance, if the matrices $T(\lambda)$ are real and 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 by safeguarded iteration. This is discussed further in [15], and in [2] where the subspaces are expanded by a Jacobi–Davidson type approach. On the other hand, an advantage of Ruhe's approach is the fact that it does not require the explicit form of the matrices $T(\lambda)$ but only 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.

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 too 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. At least for this type of problems the Arnoldi method outperforms the rational Krylov method, and the same behaviour was observed in [15] for real and symmetric rational eigenproblems governing free vibrations of fluid–solid structures.

2 Arnoldi's method for nonlinear eigenproblems.

In this section we propose an Arnoldi type iterative projection method 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 depending on a parameter $\lambda \in D \subset \mathbb{C}$.

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 (cf. [11]), and for symmetric eigenproblems it converges even cubically. Its drawback however is that it is too expensive for large problems since in every iteration step one has to solve a linear system where the system matrices vary.

Differently from the case of a linear eigenproblem replacing v by the approximation $v = T(\sigma)^{-1}T'(\lambda)x$ 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

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

Neumaier [10] 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 $e^HT(\sigma)^{-1}$ 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 [10] describes the convergence of this method.

THEOREM 2.1. *Let $T(\lambda)$ be analytic. 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 locally 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 search space V in a projection method in the following way. If $\tilde\lambda$ is an eigenvalue of the projected problem $V^HT(\lambda)Vz = 0$ 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 2.1.

In projection methods the new direction is orthonormalized against the previous search 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 $(A-\sigma B)^{-1}(A-\tilde\lambda B) = I + (\sigma-\tilde\lambda)(A-\sigma B)^{-1}B$ 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, although no Krylov space is constructed and no Arnoldi recursion holds.

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 [13], and the name nonlinear rational Krylov method would be appropriate, too. However, since Ruhe [14] already introduced a rational Krylov method for nonlinear problems which differs from our method quite a bit we prefer the name nonlinear Arnoldi method. We will comment on the differences of Ruhe's and our approach at the end of this section.

ALGORITHM 2.2. Nonlinear Arnoldi Method

1: start with an initial pole σ and an initial basis V, $V^H V = I$;
2: determine preconditioner $M \approx T(\sigma)^{-1}$, σ close to first wanted eigenvalue
3: $k = 1$
4: **while** $m \leq$ number of wanted eigenvalues **do**
5: compute appropriate eigenvalue μ and corresponding normalized eigenvector y of the projected problem $V^H T(\mu) V y = 0$
6: determine Ritz vector $u = Vy$ and residual $r_k = T(\mu)u$
7: **if** $\|r_k\| < \epsilon$ **then**
8: accept eigenvalue $\lambda_m = \mu$, and eigenvector $x_m = u$,
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 σ
15: determine new preconditioner $M \approx T(\sigma)^{-1}$
16: **end if**
17: restart if necessary
18: choose approximations (μ, u) to eigenpair wanted next
19: determine residual $r_0 = T(\mu)u$
20: $k = 0$
21: **end if**
22: $v = Mr_k$
23: $v = v - VV^H v$, $\tilde{v} = v/\|v\|$, $V = [V, \tilde{v}]$
24: reorthogonalize if necessary
25: update projected matrix $V^H T(\mu) V$
26: $k = k + 1$
27: **end file**

A template for the preconditioned Arnoldi method for nonlinear eigenvalue problems with restarts and varying preconditioner is contained in Algorithm 2.2. An iteration step has the following structure: Given the search space V we solve the projected eigenproblem by some dense solver and choose a Ritz pair approximating the eigenpair wanted next (steps 5–6). If its residual is small enough we accept this pair. If more eigenpairs are wanted we prepare the computation of the next pair choosing an initial approximation, and if advisable we update the preconditioner, and/or we reduce the search space (steps 7–21). Finally we extend the search space by residual inverse iteration (steps 22–26).

In the following we comment on some of its steps.

1 Here preinformation such as a small number of known approximate eigenvectors of problem (2.1) corresponding to eigenvalues close to σ or of eigenvectors of a contiguous problem can be introduced into the algorithm.
If no information on eigenvectors is at hand, and if 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 $T(\sigma)u = \theta T'(\sigma)u$ (3 to 6 steps were sufficient in our examples), and choose the eigenvector corresponding to the smallest eigenvalue in modulus or a small number of Schur vectors as initial basis of the search space. 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.

For certain rational eigenproblems governing free vibrations of fluid–solid structures, and of plates with elastically attached masses where the rational term is of small rank we discussed the choice of the initial space V in [16].

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 other preconditioners may be useful.

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

4 Any 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 from converging 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, i.e. we choose a new pole σ close to the next wanted eigenvalue, and determine a new preconditioner $M \approx T(\sigma)^{-1}$.

In our numerical examples it happened that the condition in step 7 was fulfilled in the first step after having increased m. Of course, in this case 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 vec-

tors. 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.

An obvious way to restart is to determine a Ritz pair (μ, u) from the projection to the current search space $\operatorname{span}(V)$ approximating an eigenpair wanted next, and to restart the Arnoldi method with this single vector u. However, this may discard too much valuable information contained in $\operatorname{span}(V)$, and may slowdown the speed of convergence too much. Therefore it is often better to restart with a subspace spanned by the Ritz vector u and a small number of eigenvector approximations obtained in previous steps which correspond to eigenvalues closest to μ.

To inhibit the algorithm from converging to the same eigenvalue repeatedly, some of the solvers of the nonlinear projected eigenproblems in step 5 take advantage of some enumeration of the eigenvalues. For instance, in order to compute all eigenvalues in the vicinity of a given pole σ we may order them by their distance to σ, and compute them one after the other by the method of successive linear problems (cf. Section 3). In this case we keep all eigenvectors x_j, $j = 1, \ldots, m$ that have been obtained in the course of the algorithm, since otherwise the enumeration would be perturbed after restart, and we continue with an orthonormal basis of $X_m := \operatorname{span}\{x_1, \ldots, x_m\}$. 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.

19 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. If $v = 0$ we continue with a random vector orthogonalized against V.

20 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 (see [3]).

21 Often problem (1.1) has the form $T(\lambda) = \sum_{j=1}^{N} f_j(\lambda) C_j$ with 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 Lagrange 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 $k = 1, 2, \ldots$ 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 $V^H T(\sigma)^{-1} T(\lambda) V y = 0$ 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 Arnoldi method as proposed here in many applications preserves symmetry properties of the underlying problem which can and should be exploited in its numerical solution for efficiency, stability and accuracy reasons. We only mention two examples. If $T(\lambda)$ is real and symmetric such that its eigenvalues allow a minmax characterization then the projected problem $V^H T(\lambda) V y = 0$ has this property too, and can be solved efficiently and safely by safeguarded iteration [15]. If $T(\lambda)$ is a symmetric/skew-symmetric polynomial (for instance quadratic and gyroscopic) then the projected matrix is of the same form, and the corresponding eigenproblem is equivalent to a linear skew-Hamiltonian eigenproblem which can be solved by a structure preserving algorithm (cf. [8, 9]). In both cases the structure is destroyed by the rational Krylov method.

3 Solving the projected problems.

A crucial point in iterative projection methods for general nonlinear eigenvalue problems is to prohibit the method from converging to the same eigenvalue repeatedly. For linear eigenproblems this is no problem. If a Ritz pair (μ, u) has converged and is an approximation to a wanted eigenvalue then it is locked, i.e. it is kept in the search space, and the computation of the remaining eigenvalues

is continued without altering μ and u. Stable and efficient transformations to implement locking which take advantage of the partial Schur factorization may be found in [1]. For nonlinear problems a Schur factorization or 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 values of a Rayleigh functional (cf. [17]) then there is a close relation between the nonlinear problem (1.1) and the symmetric linear eigenproblem

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

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 $\hat{\lambda}$ [16, 18].

ALGORITHM 3.1. Safeguarded iteration
1: Start with an approximation μ_1 to the m-th eigenvalue of (1.1)
2: **for** $\ell = 1, 2, \ldots$ until convergence **do**
3: determine eigenvector u corresponding to m-largest eigenvalue of (3.1)
4: solve $u^H T(\mu_{\ell+1})u = 0$ for $\mu_{\ell+1}$
5: **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 [15].

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

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

This suggests the method of successive linear problems introduced by Ruhe [11].

ALGORITHM 3.2. Method of successive linear problems
1: Start with an approximation μ_1 to the m-th eigenvalue of (1.1)
2: **for** $\ell = 1, 2, \ldots$ until convergence **do**
3: solve the linear eigenproblem $T(\mu_\ell)u = \theta T'(\mu_\ell)u$
4: choose the eigenvalue θ such $|\sigma_0 - (\mu_\ell - \theta)|$ is the m-smallest among the eigenvalues
5: $\mu_{\ell+1} = \mu_\ell - \theta$
6: **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 of the Arnoldi method usually is

quite small, and every standard solver for dense linear eigenproblems applies. There is no guarantee that the method detects all eigenvalues in the vicinity of σ_0, but since the already converged Ritz vectors are kept in the search space (even after a restart, cf. comment on step 17 of Algorithm 2.2) it is unlikely that the method converges to an eigenvector which has been found already in a previous step.

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

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

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 our experience that for this type of problems when the damping is not too large the eigenmodes of the damped and the undamped problem do not differ very much although the eigenvalues do. Therefore, in step 5 of the Arnoldi 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.

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. Figure 4.1) is subject to plane stress, and is clamped at the side given by $x = 0$. The instantaneous Young's modulus is set to $E = 2.10 \times 10^{11}$, 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 $\Delta E = 7 \times 10^{10}$ for $0 < x < 1$, $\Delta E = 6 \times 10^{10}$ for $1 < x < 2$, $\Delta E = 5 \times 10^{10}$ for $2 < x < 3$, $\Delta E = 4 \times 10^{10}$ for $3 < x < 4$, and $\Delta E = 3 \times 10^{10}$ for $4 < x < 5$. The relaxation constant is set to $b = 2 \times 10^{-5}$.

Discretizing this problem by linear Lagrangian elements we obtained the rational eigenproblem (3.3) of dimension 9376. For symmetry reasons we determined

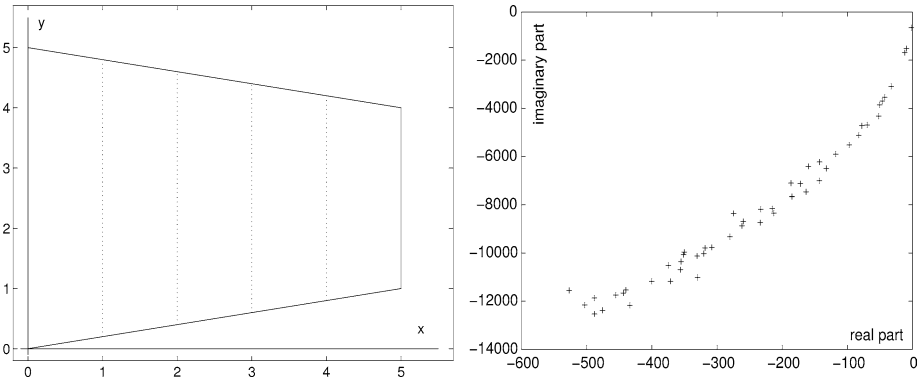

Figure 4.1: Trapezoidal plate/Eigenvalues.

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}.

Starting with the one-dimensional space V spanned by an eigenvector approximation of the linear problem $Kx = \lambda Mx$ corresponding to the smallest eigenvalue (which was obtained by 3 steps of the linear Arnoldi process), the nonlinear Arnoldi 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.

Figure 4.2 contains the convergence history. We show graphically the norm of the residual vector as a function of the iteration number. Every time when the norm is less than 10^{-6} (marked by $\times$) an eigenvalue has been found, and the iteration is continued to find the next eigenpair without changing the current search space V. The LU updates are marked by $\circ$. Notice that the method required only one step to determine the 33rd eigenvalue after the 32nd eigenvalue had been found.

The dominant share of the CPU time, namely 359.4 seconds, was consumed by the solver of the nonlinear eigenproblems. Figure 4.3 demonstrates the development of the CPU times of the entire iteration and of the share of the nonlinear eigensolvers as a function of the iteration number. It demonstrates the necessity of restarts since obviously the superlinear time consumption is mainly caused by the eigensolver.

We restarted the Arnoldi process if the dimension of the search space exceeded 80 after the method had found a new eigenvalue. To keep the enumeration

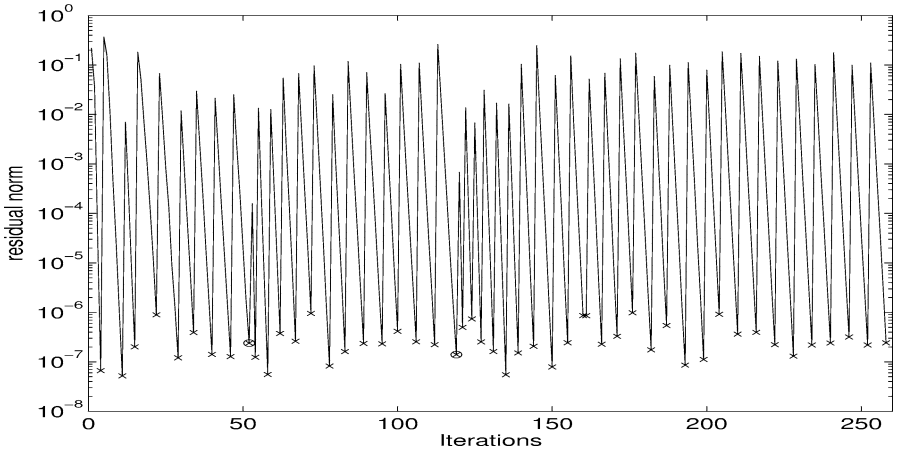

Figure 4.2: Convergence history without restarts.

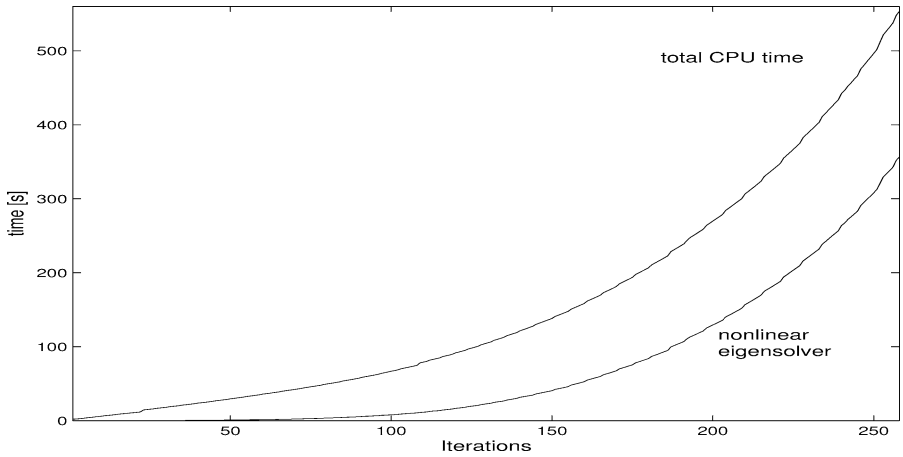

Figure 4.3: Development of CPU time consumption without restarts.

of the eigenvalues for the projected problems we restarted with the search space spanned by the already converged eigenvectors.

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. Figure 4.4 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 about a search space appropriate for the next eigenpair and to converge.

Figure 4.5 demonstrates the history of CPU time consumption for the entire Arnoldi method, the nonlinear eigensolver and the LU factorizations where we

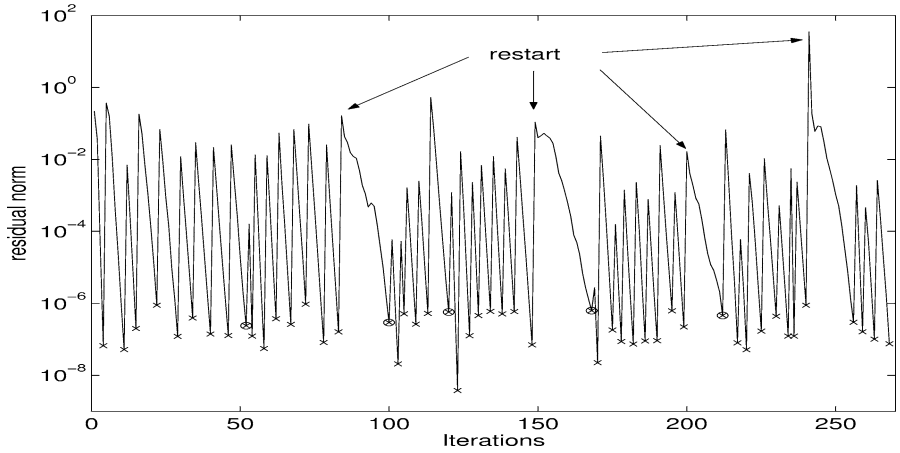

Figure 4.4: Convergence history with restarts.

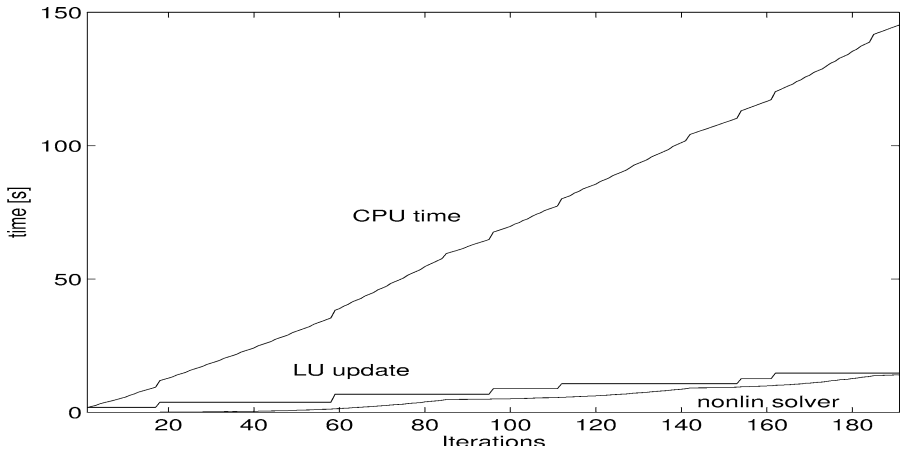

Figure 4.5: Development of CPU time consumption with restarts.

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.

Replacing the LU factorization of $T(\sigma)$ as preconditioner by an incomplete LU factorization the Arnoldi method still converges although its speed deteriorates considerably. If no fill-in is allowed the average number of iteration steps to reduce the residual to 10^{-6} is 171, and preconditioning with incomplete LU factorization with drop tolerances 0.01 and 0.001, respectively, the average number

of iterations for convergence is 47.62, and 9.84, and the CPU time to determine all 50 eigenvalues is 1333.7, and 313.4 seconds, respectively.

For comparison we determined 50 eigenvalues with maximal negative imaginary part of the nonlinear eigenproblem (3.3) by the rational Krylov method introduced by Ruhe [14] and worked out in detail by Hager [4] and Jarlebring [6]. Accepting a Ritz pair, if the residual was less than 10^{-6}, and updating the LU factorization, if the method needed more than 30 inner iterations to converge to an eigenvalue, the rational Krylov method found all 50 eigenvalue requiring 1248 inner and 582 outer iterations, a total CPU time of 1172.5 seconds and 8 LU updates. Details about the implementation of the rational Krylov method are contained in [6].

Acknowledgement.

The author gratefully acknowledges the assistance of Elias Jarlebring who provided the MATLAB implementation of the rational Krylov method. He is also grateful to the referees who made many valuable suggestions.

REFERENCES

1. Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst (eds.), *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.
2. T. Betcke and H. Voss, *A Jacobi–Davidson-type projection method for nonlinear eigenvalue problems*, Future Generation Computer Systems, 20 (2004), pp. 363–372.
3. J. Daniel, W. Gragg, L. Kaufman, and G. Stewart, *Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization*, Math. Comp., 30 (1976), pp. 87–94.
4. P. Hager, *Eigenfrequency analysis. FE-adaptivity and a nonlinear eigenvalue problem*, Ph.D. thesis, Chalmers University of Technology, Göteborg, 2001.
5. P. Hager and N. Wiberg, *The rational Krylov algorithm for nonlinear eigenvalue problems*, in B. Topping (ed.), Computational Mechanics for the Twenty-First Century, Saxe–Coburg Publications, Edinburgh, 2000, pp. 379–402.
6. E. Jarlebring, *Krylov methods for nonlinear eigenvalue problems*, Master's thesis, Royal Institute of Technology, Dept. Numer. Anal. Comput. Sci., Stockholm, 2003.
7. E. Jarlebring and H. Voss, *Rational Krylov for nonlinear eigenproblems, an iterative projection method*, Tech. Report 69, Dept. Math., Hamburg University of Technology, 2003. Submitted to Appl. Math.
8. V. Mehrmann and D. Watkins, *Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils*, SIAM J. Sci. Comput., 22 (2001), pp. 1905–1925.
9. V. Mehrmann and D. Watkins, *Polynomial eigenvalue problems with Hamiltonian structure*, Electronic Transactions on Numerical Analysis, 13 (2002), pp. 106–118.
10. A. Neumaier, *Residual inverse iteration for the nonlinear eigenvalue problem*, SIAM J. Numer. Anal., 22 (1985), pp. 914–923.
11. A. Ruhe, *Algorithms for the nonlinear eigenvalue problem*, SIAM J. Numer. Anal., 10 (1973), pp. 674–689.
12. A. Ruhe, *Computing nonlinear eigenvalues with spectral transformation Arnoldi*, ZAMM, 76 (1996), pp. S2: 17–20.
13. A. Ruhe, *Rational Krylov, a practical algorithm for large sparse nonsymmetric matrix pencils*, SIAM J. Sci. Comput., 19 (1998), pp. 1535–1551.
14. A. Ruhe, *A rational Krylov algorithm for nonlinear matrix eigenvalue problems*, Zapiski Nauchnyh Seminarov POMI, 268 (2000), pp. 176–180.

15. H. Voss, *An Arnoldi method for nonlinear symmetric eigenvalue problems*, in Online Proceedings of the SIAM Conference on Applied Linear Algebra, Williamsburg, http://www.siam.org/meetings/laa03/, 2003.
16. H. Voss, *Initializing iterative projection methods for rational symmetric eigenproblems*, in 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, 2003.
17. H. Voss and B. Werner, *A minimax principle for nonlinear eigenvalue problems with applications to nonoverdamped systems*, Math. Meth. Appl. Sci., 4 (1982), pp. 415–424.
18. H. Voss and B. Werner, *Solving sparse nonlinear eigenvalue problems*, Tech. Report 82/4, Inst. f. Angew. Mathematik, Univ. Hamburg, 1982.