SIAM J. NUMER. ANAL.
Vol. 10, No. 4, September 1973

ALGORITHMS FOR THE NONLINEAR EIGENVALUE PROBLEM*

AXEL RUHE†

Abstract. The following nonlinear eigenvalue problem is studied: Let $T(\lambda)$ be an $n \times n$ matrix, whose elements are analytical functions of the complex number λ. We seek λ and vectors x and y, such that $T(\lambda)x = 0$, and $y^H T(\lambda) = 0$.

Several algorithms for the numerical solution of this problem are studied. These algorithms are extensions of algorithms for the linear eigenvalue problem such as inverse iteration and the QR algorithm, and algorithms that reduce the nonlinear problem into a sequence of linear problems. It is found that this latter method can be extended into a global strategy, finding a complete basis of eigenvectors in the cases where it is proved that such a basis exists.

Numerical tests, performed in order to compare the different algorithms, are reported, and a few numerical examples illustrating their behavior are given.

1. Introduction and summary. We consider the following nonlinear eigenvalue problem: Let $T(\lambda)$ be an $n \times n$ matrix whose elements are analytical functions of the parameter λ. Determine those values of λ and those vectors x and y for which

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

$$y^H T(\lambda) = 0. \tag{1.2}$$

We shall call λ an eigenvalue, x a right, and y a left eigenvector of this nonlinear eigenvalue problem.

It is our aim to find useful algorithms for the solution of (1.1)–(1.2) in different cases. In the present contribution we describe some possible algorithms, and a few numerical tests we have performed in order to assess their usefulness.

The linear eigenvalue problem is a special case of (1.1) if we define

$$T(\lambda) = \lambda I - A \tag{1.3}$$

or

$$T(\lambda) = \lambda B - A.$$

The quadratic case

$$T(\lambda) = \lambda^2 A + \lambda B + C \tag{1.4}$$

arises from the solution of Lagrange's equations of motion for dissipative (nonconservative) mechanical systems. It has been studied by several authors. The reader is referred to the monograph by Lancaster [13] for references on that subject. In the case where $T(\lambda)$ is a polynomial,

$$T(\lambda) = \lambda^r A_r + \lambda^{r-1} A_{r-1} + \cdots + A_0, \tag{1.5}$$

* Received by the editors September 8, 1971, and in revised form May 22, 1972.

† Department of Information Processing, Institute of Mathematics and Statistics, University of Umeå, Umeå, Sweden.

the problem (1.1) can be reformulated as a linear problem in rn dimensions, provided either A_r or A_0 is nonsingular. In the former case the matrix

$$B = \begin{bmatrix} 0 & I & 0 & \cdots & 0 \\ 0 & 0 & I & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & I \\ B_0 & B_1 & B_2 & & B_r \end{bmatrix},$$

where $B_i = -A_r^{-1}A_i$.

A complete solution of (1.1) is obtained by solving the eigenproblem of B, and for moderate degree r this is certainly a very fast and accurate method. This method and methods of determinant evaluation are studied by Peters and Wilkinson [17] and Mavljanova [25]. We shall thus not consider this kind of algorithm further here, but instead study algorithms which directly apply to the formulation (1.1).

The mathematical theory of the general case, with special emphasis on symmetric matrices, has been studied by Rogers [18], and in a series of papers by Hadeler [5], [6], [7], [8]. We shall cite some of their results in § 2.

Different kinds of vector iterations have been proposed by Hadeler [5], who considers direct iteration, and Anselone and Rall [2], who consider inverse iteration. Both cases are rather simple extensions of the corresponding techniques for the linear problem. In § 3 we shall discuss inverse iteration methods and study their relations to the theory developed in § 2.

In the linear case (1.3) several of the most useful algorithms are based on similarity transformations. In the nonlinear case however it is no longer possible to transform $T(\lambda)$ into condensed form (e.g. Hessenberg or tridiagonal), and therefore the very successful LR and QR algorithms [22] have no counterpart for nonlinear eigenvalue problems. There is a variant of the QR algorithm that does not transform the matrix [21], and starting with an algorithm proposed by Kublanovskaja [11], [12], we develop this algorithm for the nonlinear case in § 4. We show that it is equivalent to a certain form of inverse iteration, as it has been shown by Parlett and Kahan [16] for the linear case. The main advantages of the QR algorithm, that is, the economy of computation that results from the invariance of Hessenberg form, and the convenient deflation process, are however not carried over.

If $T(\lambda)$ is *originally* of band structure or Hessenberg form, then all the algorithms considered here can be performed with a considerable economy of time and storage space. If on the other hand, $T(\lambda)$ is a full matrix, all algorithms need $O(n^3)$ arithmetical operations for *each* eigenvalue.

Since the complete solution of a linear eigenvalue problem also needs $O(n^3)$ operations, algorithms that reduce (1.1) into a sequence of linear problems can be considered. In § 5 we develop such an algorithm in a straightforward manner. It appears to be a very good algorithm, especially since it can be extended into a *global* strategy for computing a complete set of n eigenvectors to (1.1) or (1.2) spanning the whole space, in the cases when the existence of such a set is established theoretically.

In § 6 we discuss the results of the numerical tests we have performed on these algorithms, and give some indications of which algorithm to choose in a

specific case. This discussion is illustrated by an account of a few numerical examples in § 7.

A preliminary conclusion of these tests is that several strategies have fast *local* convergence. On the other hand, reliable *global* strategies exist only in quite special cases, mainly when $T(\lambda)$ is symmetric and $T'(\lambda)$ is definite. Algorithms that regard (1.1) as a matrix problem and use some kind of vector iteration in several cases work better than algorithms based on determinant evaluation. Even in well-behaved cases the algorithms need much better starting values than in the linear case, in order to be sure to converge to a certain (the closest) eigenvalue.

2. Rayleigh functionals and orthogonality. The Rayleigh quotient plays an important role both for the theory and the computation of eigenvalue problems, and it is therefore natural to seek a generalization of it applicable to nonlinear problems. Duffin [4] introduces the Rayleigh functional, defined implicitly by

$$p : x \in E^n \to p(x) \in W, \qquad W \text{ a real interval}, \tag{2.1}$$

$$p(\alpha x) = p(x), \qquad \alpha \neq 0, \tag{2.2}$$

$$x^H T(p(x))x = 0, \tag{2.3}$$

$$x^H T'(p(x))x > 0, \qquad x \neq 0. \tag{2.4}$$

The properties of $p(x)$ are studied by Rogers [18] and Hadeler [5], [6], [7]. They show that in the linear case

$$T(\alpha) = \alpha I - A;$$

then,

$$p(x) = x^H A x / x^H x,$$

the ordinary Rayleigh quotient.

The condition (2.4) is called the *overdamping condition* [18], since in the quadratic case it corresponds to an overdamped physical system. If T is overdamped it has n real eigenvalues, and much of the theory for linear symmetric eigenvalue problems is carried over to the nonlinear case [5].

To the n real eigenvalues there correspond n linearly independent eigenvectors, which however are not orthogonal. They can be made orthogonal with respect to a special, nonbilinear, scalar product defined by

$$[x, y] := y^H \Delta(p(x), p(y))x,$$

$$\Delta(\alpha, \beta) := \begin{cases} (T(\alpha) - T(\beta))/(\alpha - \beta), & \alpha \neq \beta, \\ T'(\alpha), & \alpha = \beta, \end{cases}$$

where p is defined by (2.1)–(2.4). Now the eigenvectors can be chosen to satisfy

$$[x_i, x_j] = \delta_{ij}.$$

It is difficult to use the scalar product $[\cdot, \cdot]$ when computing eigenvectors since orthogonalization processes of Gram–Schmidt type cannot be applied [5].

In the nonsymmetric case (2.1)–(2.3) also defines a continuous functional $p(x)$, in this case with W a part of the complex plane, if the overdamping condition (2.4) is replaced by the following condition having the same effect:

$$T(\alpha) = \alpha I - H(\alpha),$$

where H is bounded by

$$\begin{aligned} &U = \{\alpha : |\alpha| < \rho\}, \\ &\|H(\alpha)\| \leqq \rho, \qquad \alpha \in U, \\ &\|H(\alpha) - H(\beta)\| \leqq K|\alpha - \beta|, \quad K < 1, \quad \alpha, \beta \in U. \end{aligned} \tag{2.5}$$

This case is studied in [8].

In the nonsymmetric case the left and right eigenvectors are not identical, and it is therefore necessary to define the Rayleigh functional p as a function of two vectors x and y, satisfying the conditions

$$p(\alpha x, \beta y) = p(x, y), \qquad \alpha, \beta \neq 0,$$

$$y^H T(p(x, y))x = 0, \tag{2.6}$$

$$y^H T'(p(x, y))x \neq 0. \tag{2.7}$$

The eigenvalues are the stationary values of $p(x, y)$ and their computation amounts to seeking those stationary values.

Lancaster [13, p. 72] introduces another generalization of the Rayleigh quotient defined by

$$R(x, y, \lambda) := \lambda - \frac{y^H T(\lambda)x}{y^H T'(\lambda)x}. \tag{2.8}$$

When x, y and λ are eigenvectors and eigenvalue then certainly

$$\lambda = R(x, y, \lambda) = p(x, y),$$

and in other cases (2.8) describes one step of a Newton–Raphson iteration to solve (2.6) for $\lambda = p(x, y)$.

When no condition of the types (2.4) or (2.5) is fulfilled, the existence of the Rayleigh functional is not certain. If however the elements of $T(\lambda)$ are polynomials in λ, then complex solutions of (2.6) exist, since then (2.6) is an algebraical equation. If $T'(\lambda)$ is singular for some λ, we cannot be sure of finding n linearly independent eigenvectors.

3. Newton's method and inverse iteration. In order to apply Newton's method we formulate (1.1) as a nonlinear operator equation in $n + 1$ dimensions by adding a normalization condition

$$v^H x = 1. \tag{3.1}$$

We get the operator (see [2], [3])

$$P\begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} T(\lambda)x \\ v^H x - 1 \end{bmatrix}. \tag{3.2}$$

In order to solve the equation

$$P\begin{bmatrix} x \\ \lambda \end{bmatrix} = 0,$$

we differentiate, and obtain the Fréchet derivative

$$P' = \begin{bmatrix} T(\lambda) & T'(\lambda)x \\ v^H & 0 \end{bmatrix}. \tag{3.3}$$

Newton's method now amounts to solving an $(n+1)$-dimensional linear system in each iteration,

$$\begin{bmatrix} x_{s+1} \\ \lambda_{s+1} \end{bmatrix} = \begin{bmatrix} x_s \\ \lambda_s \end{bmatrix} - [P']^{-1}P\begin{bmatrix} x_s \\ \lambda_s \end{bmatrix}.$$

Using the partition (3.2) and (3.3) we get (denote $T(\lambda_s) = T_s$ and $T'(\lambda_s) = T_s'$)

$$\begin{bmatrix} T_s & T_s'x_s \\ v_s^H & 0 \end{bmatrix}\begin{bmatrix} x_{s+1} - x_s \\ \lambda_{s+1} - \lambda_s \end{bmatrix} = -\begin{bmatrix} T_s x_s \\ v_s^H x_s - 1 \end{bmatrix}.$$

Assuming x_s to be normalized (3.1), we get

$$\begin{aligned} T_s x_{s+1} &= -(\lambda_{s+1} - \lambda_s)T_s' x_s, \\ v_s^H x_{s+1} &= 1, \end{aligned}$$

and we see that Newton's method is equivalent to inverse iteration [20]. The algorithm is more conveniently formulated as

$$\begin{aligned} T_s u_{s+1} &= T_s' x_s, \\ \lambda_{s+1} &= \lambda_s - v_s^H x_s/(v_s^H u_{s+1}), \\ x_{s+1} &= C \cdot u_{s+1}, \end{aligned} \tag{3.4}$$

where C is an arbitrary normalization constant.

The vector v_s can be chosen in a number of ways. The simplest is to take one of the basis vectors e_i. This amounts to keeping one component of x_s constant [20], [3]. When computing several eigenvalues, a choice of v_s as a vector orthogonal to those eigenvectors already found should prevent convergence towards an eigenvalue already computed [2]. A third choice that comes naturally is

$$v_s = T_s^H y_s, \tag{3.5}$$

where y_s is an approximation to the left eigenvector y, since then (3.4) will read

$$\lambda_{s+1} = \lambda_s - \frac{y_s^H T_s x_s}{y_s^H T_s' x_s}, \tag{3.6}$$

the Rayleigh quotient (2.8) [13], and we arrive at a nonlinear version of the Rayleigh quotient iteration [15]. When $T(\lambda)$ is symmetrical, $y_s = x_s$, but in other cases they could also be taken identical. However the computation of two sequences of vectors x_s and y_s does not take much more time than computing just x_s, since in both cases only one factorization of the matrix T_s is needed in each step.

4. Methods of QR type. V. N. Kublanovskaya [11], [12] has proposed the following algorithm: Let λ_0 be a starting guess and compute the successive iterates λ_s by factorizing $T(\lambda_s)$ in the following way:

$$T_s P_s = Q_s R_s, \tag{4.1}$$

where Q_s is unitary, normally a product of $n-1$ elementary reflections (Householder transformations), R_s is upper triangular and P_s is a permutation matrix chosen so that the diagonal elements of R_s satisfy:

$$|r_{11}| \geqq |r_{22}| \geqq \cdots \geqq |r_{nn}|. \tag{4.2}$$

Considering the element r_{nn} as a function $f(\lambda)$, solving the problem (1.1) amounts to seeking a zero of $f(\lambda)$, and an application of Newton's method yields

$$\begin{aligned} f'(\lambda_s)/f(\lambda_s) &= e_n^T R_s' R_s^{-1} e_n \\ &= e_n^T Q_s^H T_s' P_s R_s^{-1} e_n, \\ \lambda_{s+1} &= \lambda_s - 1/(e_n^T Q_s^H T_s' P_s R_s^{-1} e_n). \end{aligned} \tag{4.3}$$

M. I. Mavljanova [24] has studied the conditions for the quadratic convergence of this algorithm. As approximations to the eigenvectors the factorization (4.1) suggests

$$y_s = Q_s e_n, \qquad x_s = P_s R_s^{-1} e_n$$

for the left and right vectors respectively. Inserting this into (3.6) we get, using (4.1),

$$\begin{aligned} y_s^H T_s x_s &= e_n^T Q_s^H (Q_s R_s P_s^T) P_s R_s^{-1} e_n \\ &= e_n^T e_n = 1, \\ y_s^H T_s' x_s &= e_n^T Q_s^H T_s' P_s R_s^{-1} e_n, \end{aligned}$$

and we see that (4.3) is equivalent to (3.6) with this choice of vectors.

Furthermore

$$\begin{aligned} y_s^H T_s &= e_n^T Q_s^H (Q_s R_s P_s^T) = e_n^T R_s P_s^T \\ &= r_{nn} \cdot e_k^T, \\ T_s x_s &= Q_s R_s P_s^T (P_s R_s^{-1} e_n) = Q_s e_n = y_s, \end{aligned}$$

and we see that the current approximation y_s has been obtained by one step of (3.4) transposed applied to the starting vector e_k, where k is the index of the column that is permuted to the last position by P_s, while x_s has been obtained by one step of (3.4) with y_s as starting vector; in both cases T' is set to the unit matrix. Consequently one iteration of the algorithm of Kublanovskaja is equivalent to one step of a Rayleigh quotient iteration starting with an appropriately chosen coordinate vector.

In the algorithm of Kublanovskaja, the eigenvector iteration starts anew in each step. By replacing P_s in (4.1) by Q_{s-1} we obtain an algorithm that is equivalent to the QR algorithm with λ_s as the shift of origin:

$$\begin{aligned} &T_sQ_{s-1} = Q_sR_s, \\ &\lambda_{s+1} = \lambda_s - 1/(e_n^TQ_s^HT_s'Q_{s-1}R_s^{-1}e_n). \end{aligned} \tag{4.4}$$

Note that it is not generally possible to perform similarity transformations of a nonlinear eigenvalue problem, so that we have to use this unusual formulation of the QR algorithm [21], [16]. The analysis of the preceding paragraph can be carried through in this case too, and shows that the approximate left eigenvectors ($y_s = Q_se_n$) are the same as would have been obtained from the inverse iteration (3.4), while the right vector ($x_s = Q_{s-1}R_s^{-1}e_n$) is obtained by one step of (3.4) applied to y_s. Also now T' is replaced by the unit matrix in both cases.

We note that it is only the last column of Q_s that is really of interest. Instead of Q_{s-1} we could therefore use any unitary matrix with the same last column in (4.4). Taking an elementary reflection chosen in this manner, we reduce the amount of computation by about half in each iteration.

It is possible to take into account also the effect of T' in the vector iteration. We then get the following improved algorithm:

$$\begin{aligned} &\text{Determine } P_s = I - 2w_sw_s^H \text{ so that} \\ &P_se_n = K \cdot T_s'^HQ_{s-1}e_n, \qquad K \text{ constant}. \\ &\text{Compute } Q_s \text{ and } R_s \text{ from } T_sP_s = Q_sR_s, \\ &\lambda_{s+1} = \lambda_s - 1/(e_n^TQ_s^HT_s'P_sR_s^{-1}e_n). \end{aligned} \tag{4.5}$$

Note that the amount of work in each iteration is only marginally larger than in the original algorithm of Kublanovskaja.

5. The method of successive linear problems. We could linearize the problem by using Taylor's formula,

$$T(\lambda + h) = T(\lambda) + hT'(\lambda) + \frac{h^2}{2}R(\lambda, h), \tag{5.1}$$

where $R(\lambda)$ is a matrix whose norm is bounded by

$$\|R(\lambda, h)\| \leqq \sup_{|\xi| < |h|} \|T''(\lambda + \xi)\|.$$

Discarding R we get the following algorithm:

$$\begin{aligned} &\text{Choose } \lambda_0 \text{ appropriately.} \\ &\text{Solve } -T_sx_{s+1} = \mu_sT_s'x_{s+1}, \\ &\lambda_{s+1} = \lambda_s + \mu_s. \end{aligned} \tag{5.2}$$

If μ_s is chosen as the absolutely smallest eigenvalue, then the convergence is quadratic. In this algorithm it is possible to take advantage of the fact that $T(\lambda)$

is symmetric if the overdamping condition (2.4) is fulfilled, since then either $T'(\lambda)$ or $-T'(\lambda)$ is positive definite and a Cholesky decomposition of $T'(\lambda)$ can be performed.

Each iteration of the algorithm (5.2) takes much more time than any of the algorithms of §§ 3 or 4, but the convergence is faster in many cases. The most important advantage of (5.2), however, is that it is easily extended into a *global strategy* for the symmetric overdamped case (2.1)–(2.4) based on the following theorem.

THEOREM 1. *Let σ be either $+1$ or -1, and W be an interval and assume that $\sigma T'(\lambda)$ is positive definite for any $\lambda \in W$, and that $\sigma T(\lambda_0)$ is positive definite for one $\lambda_0 \in W$. Then the number of eigenvalues of $T(\lambda)$ greater than $\lambda_1 \in W$ is equal to the number of negative eigenvalues of the linear problem*

$$T(\lambda_1)x = \mu T'(\lambda_1)x. \tag{5.3}$$

Proof. If $T(\lambda_0)$ is positive definite, then the Rayleigh functional $p(x)$ satisfies

$$p(x) < \lambda_0$$

for any x and thus no eigenvalue of $T(\lambda)$ is greater than λ_0 (see [18]).

By proper ordering, each of the n eigenvalues μ_k of (5.3) is a differentiable function of λ and since $T'(\lambda)$ is positive definite $\mu_k(\lambda)$ is monotonically increasing [10]. Assuming

$$\mu_k(\lambda_1) < 0, \qquad k = 1, 2, \cdots, i,$$

$$\mu_k(\lambda_1) \geqq 0, \qquad k = i+1, \cdots, n,$$

each of $\mu_1(\lambda), \cdots, \mu_i(\lambda)$ must have a zero in the interval (λ_1, λ_0) since

$$\mu_k(\lambda_0) > 0, \qquad k = 1, 2, \cdots, n.$$

Each of these i zeros gives an eigenvalue of the nonlinear problem, if we count multiplicities. This completes the proof.

When one eigenvalue λ has been computed by the algorithm (5.2), the second smallest (or the next in the real case) eigenvalue of (5.3) can be added to λ to give a good first approximation to the next eigenvalue of the nonlinear problem. This strategy has given excellent results both in symmetric and general cases. Even when the conditions of Theorem 1 are not fulfilled it gets a maximal possible number of eigenvalues.

6. Numerical tests. Programs for the different algorithms discussed in this paper have been written and some numerical tests performed.

The local behavior of the inverse iteration algorithms of § 3 is studied in the report [1]. We have found that the algorithm (3.4) with the vectors v_s chosen as in (3.5) converges satisfactorily with quadratic convergence. However, as could be expected, the initial guess λ_0, x_0 has to be quite good, if we will be sure of getting convergence to the closest eigenvalue. With ad hoc guesses convergence occurs, but seldom to the eigenvalue closest to the initial guess.

It is possible to replace T' in (3.4) and (3.6) by a difference

$$T'(\lambda_s) \approx \frac{T(\lambda_s) - T(\lambda_{s-1})}{\lambda_s - \lambda_{s-1}} \tag{6.1}$$

and obtain a secant iteration. It is also tested in [1], and found to work well. The limiting accuracy is somewhat smaller since differences are taken, but even in quite ill-conditioned cases it gives a reasonable approximation. In cases when T' is not readily available this modified algorithm is certainly superior.

When comparing the original algorithm of Kublanovskaja (4.1)–(4.3) with the improved versions (4.4) and (4.5) all three converged quadratically, as could be expected. It was remarkable, however, that the improved versions (4.4) and (4.5) did not converge much faster than (4.1)–(4.3); often only one or two iterations were saved. In (4.1) a starting vector is chosen implicitly in each step by the permutations; this is enough to yield fast convergence. Tests of (4.1)–(4.3) are described in the report [14].

It is interesting to compare the algorithms of §§ 3 and 4 to the various algorithms based on determinant evaluation, described in [13] and [17]. In fact, having computed a factorization (4.1), we see that

$$\det(T(\lambda_s)) = (-1)^{\nu} \prod_{i=1}^{n} r_{ii}$$

and taking $f(\lambda) = \det(T(\lambda))$, Newton's method applied to the equation

$$\det(T(\lambda)) = 0$$

yields the iteration [13, p. 83]

$$\lambda_{s+1} = \lambda_s - 1/\mathrm{tr}\,(Q_s^H T_s' P_s R_s^{-1}) \tag{6.2}$$

which is quite similar to (4.3). However this algorithm did not give fewer iterations than the simpler (4.3). Especially in the cases when $T(\lambda)$ had multiple eigenvalues corresponding to linear elementary divisors, (4.3) still converged quadratically, and gave the correct annihilated subspace, while (6.2) converged only linearly.

When we have a defective problem, that is, an eigenvalue with fewer linearly independent eigenvectors than its multiplicity, convergence is very slow for all algorithms. Most algorithms gave geometric convergence, as predicted by the theory. It is worth noticing that when a linear problem is treated with the QR algorithm (or another transformation method) [22], rounding errors will successively perturb the matrix, so that a nondefective matrix occurs. After that a fast, quadratic convergence occurs towards the eigenvalues of this perturbed problem. In the nontransformation algorithms studied here, rounding errors also introduce perturbations, but now we get a new perturbation in each step since we start with the original matrix in each iteration. Thus we never get faster than geometrical convergence, and soon arrive at a "noise level" where no improvements occur. Another difficulty that does not occur in linear problems, is that $T'(\lambda)$ may be singular at a multiple eigenvalue effectively impairing the use of (5.2).

When we look for a *global method*, that is, an algorithm that finds either all eigenvalues in a prescribed domain, or a system of eigenvectors that spans the whole space, the situation is not as good as in the linear case, since we cannot perform explicit deflations. We have tried three other strategies.

When one root λ of the equation

$$f(\lambda) = 0$$

is found we divide it out as proposed by Lancaster [13, p. 85] and, when the $k-1$ eigenvalues $\lambda^{(1)}, \cdots, \lambda^{(k-1)}$ are computed, we get

$$g(\lambda) = f(\lambda) / \prod_{i=1}^{k-1} (\lambda - \lambda^{(i)})$$

giving a Newton–Raphson iteration with

$$g'(\lambda)/g(\lambda) = f'(\lambda)/f(\lambda) - \sum_{i=1}^{k-1} 1/(\lambda - \lambda^{(i)}). \tag{6.3}$$

The subtraction (6.3) is performed in each step. In our tests this strategy was used in the iterations (4.3), (4.4) and (4.5). It gave one or a few more eigenvalues in the cases we tried, but the second eigenvalue was seldom the one closest to the first, and in no nontrivial case did we get a complete system of eigenvalues.

We also tried the strategy proposed by Anselone and Rall [2] of taking the vector v_s in (3.4) orthogonal to the eigenvectors already computed. In fact, when computing $\lambda^{(k)}$, we orthogonalized the approximate left vector $T_s^H y_s$ against all computed right vectors $x^{(1)}, \cdots, x^{(k-1)}$, using modified Gram–Schmidt orthogonalization [19], so that we obtained a modified Rayleigh quotient iteration (3.6). This was found to be a much better strategy than (6.3); in well-conditioned cases it gave a complete system of eigenvectors.

For the overdamped case (2.4), Theorem 1 establishes a relation between the eigenvalues of the linear problem (5.3) and (1.1). Having found one eigenvalue of (1.1) we take the next eigenvalue of (5.3) and so we can go through all the n real eigenvalues. When the eigenvalues are complex we do not have such a simple way to order them as along the real line, but for all nondefective problems we have tried, the following strategy has worked well and given n different eigenvalues:

Introduce a partial ordering in the complex plane.

$$f : C \ni \alpha \to f(\alpha) \in R, \qquad \alpha \prec \beta \Leftrightarrow f(\alpha) \leqq f(\beta). \tag{6.4}$$

Order the eigenvalues μ_i of (5.3) so that

$$\mu_j \prec \mu_k : j < k. \tag{6.5}$$

In each iteration choose

$$\lambda_{s+1} = \lambda_s + \mu_{\text{iev}} \tag{6.6}$$

if eigenvalue iev is computed.

In most cases we have chosen

$$f(\alpha) := \operatorname{Re}(\alpha) + \operatorname{Im}(\alpha) \tag{6.7}$$

and thus got the eigenvalues from left to right and upwards. In stability problems of the type considered by Lancaster [13] it would be appropriate to take

$$f(\alpha) := \operatorname{Im}(\alpha).$$

We have also tried to take the eigenvalue of smallest absolute value in (5.2), but then we often got convergence to the same eigenvalue several times; in some cases, however, this strategy gave a somewhat faster local convergence.

The method of successive linear problems described in § 5, (5.2) needs much more computational work in each iteration—in the symmetric overdamped case about 6 times, and in the nonsymmetric case about 40 times more, than inverse iteration (3.4). In both cases no regard is paid to the time needed to compute the elements of $T(\lambda)$ and $T'(\lambda)$. If that time dominates the computation, then the algorithm (5.2) is clearly motivated, since in most cases it gives the fewest iterations. However the algorithms of §§ 3 and 4 have the same asymptotic rate of convergence, so in most cases they are sufficiently effective. A combined strategy of first taking one or two steps of (5.2) followed by (3.4) or (4.1)–(4.3) seems to give reasonable results with not too peculiar problems. Our tests indicate that two steps of (5.2) are sufficient to assure that (4.1)–(4.3) shall converge towards the correct eigenvalue; with only one step of (5.2) we sometimes missed it.

Some tests have also been performed with an algorithm based on minimizing the functional

$$f(x, \lambda) := \|T(\lambda)x\|_2$$

by means of quasi-Newton methods. The results, reported in [27], show that this approach gives much poorer results than any of the algorithms treated here.

7. Numerical examples. To illustrate the behavior of the algorithms described here and the conclusions drawn in the preceding section, we list the results on some numerical examples. The programs have been run on a CD 3200 computer, using a 36 bit mantissa.

We have used the procedures of *Handbook of Linear Algebra* [9] when solving (5.2) in the real symmetric and complex cases. The QZ algorithm of Moler and Stewart [26], which was not available when the tests were performed, is specially devised for this kind of problem and should be preferred.

First we consider a typical problem, where the matrix $T(\lambda)$ is symmetric, and where the overdamping condition (2.4) is fulfilled, so that a complete system of eigenvectors exists. The matrix is given in [5] where also some calculations are described. The problem is defined by

$$T(\lambda) = (e^{\lambda} - 1)B_1 + \lambda^2 B_2 - B_0, \tag{7.1}$$

where

$$B_0 = b_0 I,$$
$$B_1 = (b_{jk}^{(1)}), \qquad b_{jk}^{(1)} = [n + 1 - \max(j, k)] \cdot j \cdot k,$$
$$B_2 = (b_{jk}^{(2)}), \qquad b_{jk}^{(2)} = n\delta_{jk} + 1/(j + k).$$

We present the results for $n = 8$ and $b_0 = 100$. We see that

$$T'(\lambda) = e^{\lambda} B_1 + 2\lambda B_2$$

is positive definite for $\lambda \geqq 0$, and negative definite for large negative λ. Certainly there also exists $\lambda_0 > 0$ such that $T(\lambda_0)$ is positive definite while $T(0)$ is negative definite, so by Theorem 1 there are n eigenvalues in the interval $0 < \lambda < \lambda_0$. There is also a negative interval that contains n eigenvalues.

We list in Table 1 a summary of the results of the eigenvalue calculations. We list the eigenvalues in ascending order in each group. The algorithm (5.2) really computed them in that order, while the other algorithms gave them in an arbitrary order. After the eigenvalues we list the number of iterations needed by different algorithms.

TABLE 1

i	Eigenvalue λ_i	Algorithm 1 it'ns	Algorithm 2 order	Algorithm 2 it'ns	Algorithm 3 it'ns	Algorithm 4 it'ns
1	0.217461384	4	1	5	1 + 4	2 + 3
2	0.884961520	4	4	6	—	2 + 4
3	1.394724184	4	3	6	1 + 6	2 + 4
4	1.726304141	3	2	6	1 + 3	2 + 3
5	2.007943631	3	7	14	1 + 4	2 + 2
6	2.335424784	3	6	14	1 + 4	2 + 3
7	2.731077006	3	5	13	1 + 4	2 + 3
8	3.182595890	3	—	—	1 + 4	2 + 3
1	−7.642558349	3	1	5		
2	−4.521556148	4	2	6		
3	−3.968169057	3	4	7		
4	−3.801274897	2	3	6		
5	−3.702761577	2	5	5		
6	−3.627468151	2	6	7		
7	−3.571755851	2	8	6		
8	−3.491852633	2	7	7		

Algorithm 1 is the method of successive linear problems (5.2), which computed the eigenvalues in ascending order. *Algorithm 2* is (4.5) combined with the strategy of Anselone and Rall [2], described in § 6, to get all eigenvalues. Here we also list the order in which the eigenvalues were computed. *Algorithm 3* is a combined strategy with one step of (5.2) followed by (4.1)–(4.3), and *Algorithm 4* is two steps of (5.2) combined with (4.1)–(4.3). When no number of iterations is given in the table, convergence occurred but to the wrong eigenvalue. Divergence never occurred, except for defective problems. We see that Algorithm 1 has the fewest iterations, but in this case the other algorithms might be faster, because they need less work for each iteration. The eigenvalues are relatively close together, which however causes no trouble with these algorithms; ad hoc guesses or the strategy (6.3) on the other hand did not find more than some of the eigenvalues.

As a second example we consider the following quadratic problem, which is studied in [13]:

$$T(\lambda) = B_0 + \lambda B_1 + \lambda^2 B_2, \tag{7.2}$$

$$B_0 = \begin{bmatrix} 121 & 18.9 & 15.9 \\ 0 & 2.7 & 0.145 \\ 11.9 & 3.64 & 15.5 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 7.66 & 2.45 & 2.1 \\ 0.23 & 1.04 & 0.223 \\ 0.6 & 0.756 & 0.658 \end{bmatrix},$$

$$B_2 = \begin{bmatrix} 17.6 & 1.28 & 2.89 \\ 1.28 & 0.824 & 0.413 \\ 2.89 & 0.413 & 0.725 \end{bmatrix}.$$

This problem has three well-separated complex conjugate pairs of eigenvalues. We list the results of Algorithms 1, 3 and 4 (see the preceding example) in Table 2. We started at

$$\lambda_0 = -1 + i$$

and used the ordering function (6.7) and the strategy (6.4)–(6.6) to find all eigenvalues with positive imaginary parts. We note that one iteration of (5.2) was not enough to give convergence of (4.1)–(4.3) to the second eigenvalue.

TABLE 2

i	Eigenvalue λ_i	Algorithm 1 it'ns	3 it'ns	4 it'ns
1	$-0.917998172 + i \cdot 1.760584204$	5	6	1 + 5
2	$0.094721726 + i \cdot 2.522876588$	5		2 + 4
3	$-0.884830246 + i \cdot 8.441512159$	6	1 + 5	2 + 4

Finally we list results from a test problem which has eigenvalues whose closeness depends on a parameter, in order to study the behavior in ill-conditioned and degenerate cases. This problem can also be found in [13], and is of the form (7.2), with

$$B_0 = \begin{bmatrix} -1 + 2\alpha^2 & \alpha(1 - \alpha^2 - 2\beta^2) & 2\alpha^2\beta^2 & -\alpha\beta^2(\alpha^2 + \beta^2) \\ 2\alpha & -(\alpha^2 + 2\beta^2) & 2\alpha\beta^2 & -\beta^2(\alpha^2 + \beta^2) \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 3\alpha & -(1 + \alpha^2 + 2\beta^2) & \alpha(1 + 2\beta^2) & -\beta^2(\alpha^2 + \beta^2) \\ 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix},$$

$$B_2 = I.$$

Here $\alpha \geqq 0$ is a parameter and $\beta = 1 + \alpha$. The eigenvalues are given as crosses in Fig. 1. We see that when $\alpha = 0$ we have triple eigenvalues at $\pm i$ and double at 0. The algorithm (5.2) with the global strategy (6.4)–(6.7) consistently gave the roots enclosed in circles if we started somewhere in the lower left part of the complex plane.

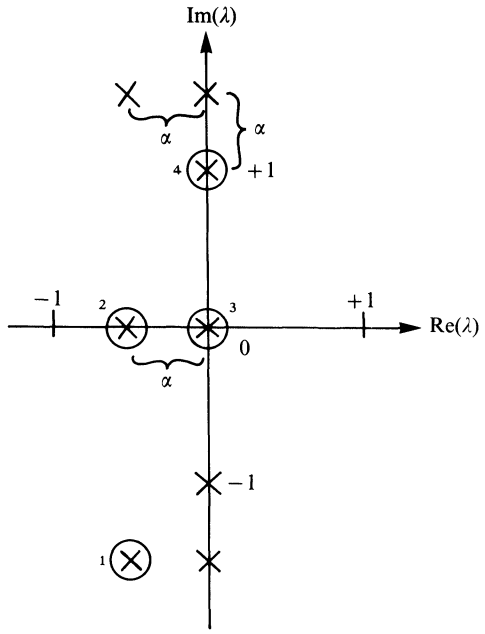

FIG. 1

× = eigenvalue, ○ = value computed by Algorithm 1

In Table 3, we list results for Algorithms 1, 3 and 4 considered earlier. We list starting points and final error in the respective eigenvalues, together with the number of iterations needed. Full accuracy means that the 9 decimals we listed were correct.

TABLE 3

Parameter α	Starting point	Eigenvalue	Algorithm 1 it'ns	Algorithm 1 error	Algorithm 3 it'ns	Algorithm 4 it'ns
0.5	$-1-2i$	1	7	Full	1 + 5	2 + 4
		2	6	Full	—	2 + 5
		3	6	Full	1 + 7	2 + 5
		4	7	Full	—	2 + 5
0.1	$-1-i$	1	10	Full	1 + 9	1 + 9
		2	5	Full	—	2 + 5
		3	12	Full	—	—
		4	9	$2_{10}-8$	—	2 + 6
0.02	$-0.1-1.1i$	1	8	$1_{10}-8$	1 + 6	2 + 5
		2	5	Full	—	2 + 4
		3	16	Full	—	2 + 12
		4	12	$7_{10}-9$	1 + 17	2 + 8
0	$-0.01-1.01i$	1	12	$1_{10}-7$	$19(2_{10}-5)$	
	$0.1+0i$	2 + 3	30	$5_{10}-6$	$1+19(2_{10}-7)$	

The case $\alpha = 0$ needs a special comment. The convergence of Algorithm 1 towards the root $-i$ was rather peculiar. First it converged geometrically as could be expected then in one step it came close to the correct value only to jump far away in the next. This happened, because $T'(-i)$ is singular giving an indeterminate step in (5.2). For the other root, and the other algorithms, convergence was geometrical. We list in parentheses the limiting accuracy.

Acknowledgments. The author is indebted to several of his students for writing programs and performing tests, reported as term papers by Tore Ahlström and Thomas Nilsson [1], Bengt Enqvist [23], Per Lindström [14] and Johnny Widén [27].

REFERENCES

[1] T. AHLSTRÖM AND T. NILSSON, *Three algorithms for solving the nonlinear eigenvalue problem*, Tech. Rep. UMINF-2.71, Dept. of Information Processing, Univ. of Umeå, Umeå, Sweden.

[2] P. M. ANSELONE AND L. B. RALL, *The solution of characteristic value-vector problems by Newton's method*, Numer. Math., 11 (1968), pp. 38–45.

[3] L. COLLATZ, *Functional Analysis and Numerical Mathematics*, Academic Press, New York and London, 1966.

[4] R. J. DUFFIN, *A minimax theory for overdamped networks*, Arch. Rational Mech. Anal., 4 (1955), pp. 221–233.

[5] K. P. HADELER, *Mehrparametrige und nichtlineare Eigenwertaufgaben*, Ibid., 27 (1967), pp. 306–328.

[6] ———, *Variationsprinzipien bei nichtlinearen Eigenwertaufgaben*, Ibid., 30 (1968), pp. 297–307.

[7] ———, *Ritzsches Verfahren bei nichtlinearen Eigenwertaufgaben*, Z. Angew. Math. Mech., 48 (1968), pp. T75–76.

[8] ———, *Anwendung von Fixpunktsätzen auf nichtlineare Eigenwertaufgaben*, Math. Z., 112 (1969), pp. 181–189.

[9] *Handbook for Automatic Computation*, vol. II: *Linear Algebra*, J. H. Wilkinson and C. Reinsch, eds., Springer-Verlag, Berlin, Heidelberg, New York, 1971.

[10] T. KATO, *Perturbation Theory for Linear Operators*, Springer-Verlag, Berlin, Heidelberg, New York, 1966.

[11] V. N. KUBLANOVSKAJA, *On an application of Newton's method to the determination of eigenvalues of λ-matrices*, Dokl. Akad. Nauk SSSR, 188 (1969), pp. 1004–1005 = Soviet Math. Dokl., 10 (1969), pp. 1240–1241.

[12] ———, *On an approach to the solution of the generalized latent value problem for λ-matrices*, this Journal, 7 (1970), pp. 532–537.

[13] P. LANCASTER, *Lambda-Matrices and Vibrating Systems*, Pergamon Press, Oxford, 1966.

[14] P. LINDSTRÖM, *Kublanovskaya's algorithm for solving the nonlinear eigenvalue problem*, Tech. Rep. UMINF-5.71, Dept. of Information Processing, Univ. of Umeå, Umeå, Sweden.

[15] A. M. OSTROWSKI, *On the convergence of the Rayleigh quotient iteration for the computation of characteristic roots and vectors*, Arch. Rational Mech. Anal., 1, pp. 233–241; 2, pp. 423–428; 3, pp. 325–340; 3, pp. 341–347; 3, pp. 472–481; 4, pp. 153–165; (1958–59).

[16] B. N. PARLETT AND W. KAHAN, *On the convergence of a practical QR algorithm*, Proc. IFIP Congress, Edinburgh, 1968, pp. 25–30.

[17] G. PETERS AND J. H. WILKINSON, *$Ax = \lambda Bx$ and the generalized eigenproblem*, this Journal, 7 (1970), pp. 479–492.

[18] E. H. ROGERS, *A minimax theory for overdamped systems*, Arch. Rational Mech. Anal., 16 (1964), pp. 89–96.

[19] H. RUTISHAUSER, *Orthonormierung von Vektoren nach E. Schmidt*, Computing, 2 (1966), pp. 159–161.

[20] H. UNGER, *Nichtlineare Behandlung von Eigenwertaufgaben*, Z. Angew. Math. Mech., 30 (1950), pp. 281–282.

[21] J. H. WILKINSON, *Convergence of the LR, QR, and related algorithms*, Comput. J., 8 (1965), pp. 77–84.

[22] ———, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965.

[23] B. ENQVIST, *The method of successive linear problems applied to a nonlinear eigenvalue problem*, Tech. Rep. UMINF-6.71, Dept. of Information Processing, Univ. of Umeå, Umeå, Sweden.

[24] M. I. MAVLJANOVA, *Solution of the special eigenvalue problem of a polynomial matrix*, Zap. Nauchn. Seminarov Leningrad Otdel. Mat. Inst. Steklov (LOMI), 18 (1970), pp. 116–124.

[25] ———, *On a method of constructing a matrix solution of a polynomial matrix*, Ibid., 18 (1970), pp. 125–137.

[26] C. B. MOLER AND G. W. STEWART, *An algorithm for the generalized matrix eigenvalue problem* $Ax = \lambda Bx$, Tech. Rep. CNA-32, Center for Numerical Analysis, Univ. of Texas, Austin, 1971.

[27] J. WIDÉN, *Quasi-Newton algorithms applied to eigenvalue problems*, Tech. Rep. UMINF-23.72, Dept. of Information Processing, Univ. of Umeå, Umeå, Sweden.