

HOMOTOPY CONTINUATION METHOD FOR THE NUMERICAL SOLUTIONS OF GENERALISED SYMMETRIC EIGENVALUE PROBLEMS

D. C. DZENG and W. W. LIN¹

(Received 16 March 1989; revised 21 August 1990)

Abstract

We consider a generalised symmetric eigenvalue problem $Ax = \lambda Mx$, where A and M are real n by n symmetric matrices such that M is positive semidefinite. The purpose of this paper is to develop an algorithm based on the homotopy methods in [9, 11] to compute all eigenpairs, or a specified number of eigenvalues, in any part of the spectrum of the eigenvalue problem $Ax = \lambda Mx$. We obtain a special Kronecker structure of the pencil $A - \lambda M$, and give an algorithm to compute the number of eigenvalues in a prescribed interval. With this information, we can locate the lost eigenpair by using the homotopy algorithm when multiple arrivals occur. The homotopy maintains the structures of the matrices A and M (if any), and the homotopy curves are n disjoint smooth curves. This method can be used to find all/some isolated eigenpairs for large sparse A and M on SIMD machines.

1. Introduction

Consider a generalised symmetric eigenvalue problem

$$Ax = \lambda Mx, \quad (1.1)$$

where $\lambda \in \mathbb{R}$, $x \in \mathbb{R}^n$, A and M are real n by n symmetric matrices, and M is positive semidefinite. We shall always assume that the nullspace of A and M intersect trivially, i.e.

$$\mathcal{N}(A) \cap \mathcal{N}(M) = \{0\}. \quad (1.2)$$

This kind of problem, for instance, arises from vibration mode analysis when a finite element method is used. In that case A is the stiffness matrix and M is the mass matrix.

¹Institute of Applied Maths., Tsing Hua University, Hsinchu, Taiwan, Republic of China.

© Copyright Australian Mathematical Society 1991, Serial-fee code 0334-2700/91

Our goal is to find all or some eigenpairs (λ, x) of (1.1). In practice, the matrices A and M are large and sparse (e.g. band structure) so that it is not practical to perform equivalence transformations using the methods from [5], [13], [2] which destroy the band structure of A and M .

The homotopy continuation method has been used to solve eigenvalue problems for the past few years. Kalaba et al. [7, 8] suggested an algorithm to find all eigenpairs of a family of matrices $y(t)$, which are parameterised by t . Chu [3] first applied the homotopy method to the symmetric eigenvalue problem $Ax = \lambda x$. Recently, Li and Rhee [9] gave an elegant algorithm for this problem, while Lin and Lutzer [11] gave an algorithm for the problem (1.1) with positive definite M . Moreover, in [4] and [10] they proposed some important theoretical results by using a homotopy continuation method for generalised eigenvalue problems and λ -matrix problems. In [11] the approach was to solve an ODE by using an adequate ODE-solver, which is expensive when the eigenpath is ill-conditioned. In this paper we would like to extend the ideas of [9, 11] to solve the problem (1.1) without destroying the sparsity of A and M . The method is globally convergent and starting vectors are readily available [7, 8]. It can also be shown that there exist n disjoint smooth homotopy curves characterised by a system of differential equations. A standard approach is to use a predictor-corrector method [9] to follow the continuation curve $(x_1(t), \dots, x_n(t), \lambda(t), t)$ in $\mathbb{R}^{n+1} \times [0, 1]$ constructed by the homotopy equation $H(x, \lambda, t) = 0$ (see Section 2) from $t = 0$ to $t = 1$. We first predict an approximate eigenpair on the eigenpair curve, and then apply the inverse power method and Rayleigh quotient iteration to correct the eigenpair. The homotopy method has the order-preserving property. But, in practice, when we follow the k th eigenpath, it is possible to jump into the neighbouring eigenpath. To make sure we have obtained the correct number of eigenpairs at $t = 1$, a reliable checking algorithm is needed. Unfortunately, for the positive semidefinite matrix M , in contrast to a positive definite or identity matrix M , there is no convenient checking algorithm such as the Sturm sequence method used in [9]. We give the special Kronecker structure of the pencil $A - \lambda M$ and develop a useful checking algorithm to compute the number of eigenvalues in a given interval. With this information, we can locate the lost eigenpairs by using the homotopy method.

Each homotopy curve is completely independent of the others. As a consequence, curves can be followed simultaneously on SIMD machines. Hence the homotopy algorithm could be attractive for exploiting the advantages of parallel processing. The plan of this paper is as follows. In Section 2, we construct a special homotopy equation and show that the conditioning of the homotopy curve is independent of the size of the matrices. Section 3

describes a predictor-corrector technique. A checking algorithm and some numerical results are given in Sections 4 and 5, respectively.

Throughout this paper we denote a positive definite (semidefinite) matrix E by $E > 0$ (≥ 0), and the i -th column of a unit matrix I_n by e_i . We define by $\|x\|_E := (x^T E x)^{1/2}$ ($E > 0$) the E -norm of a vector $x \in \mathbb{R}^n$, and by $\|A\|_E := \max_{x \neq 0} \|Ax\|_E / \|x\|_E$ the matrix E -norm of a matrix $A \in \mathbb{R}^{n \times n}$. The set of eigenvalues of the pencil $A - \lambda M$ is denoted by $\sigma(A, M)$.

2. Some main theorems and conditioning of the eigenvalue curve

The problem in finding the eigenpairs in (1.1) can be expressed equivalently as the problem of solving the system of nonlinear equations

$$F(x, \lambda) = \begin{bmatrix} Ax - \lambda Mx \\ (1 - x^T E x)/2 \end{bmatrix} = 0, \tag{2.1}$$

where E is positive definite.

Let $A - \lambda M$ and $D - \lambda E$ be two symmetric pencils with $M \geq 0$ and $E > 0$. To solve (2.1) we consider the following homotopy equation:

$$\begin{aligned} H(x, \lambda, t) &= (1 - t) \begin{bmatrix} Dx - \lambda E x \\ (1 - x^T E x)/2 \end{bmatrix} + t \begin{bmatrix} Ax - \lambda M x \\ (1 - x^T E x)/2 \end{bmatrix} \\ &= \begin{bmatrix} A(t)x - \lambda M(t)x \\ (1 - x^T E x)/2 \end{bmatrix} = 0, \end{aligned} \tag{2.2}$$

where $x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}$, $t \in [0, 1]$, $A(t) = (1 - t)D + tA$ and $M(t) = (1 - t)E + tM$. The pencil $D - \lambda E$ is chosen such that it has n distinct real eigenvalues and all its eigenpairs are easily computed. If t runs from 0 to 1, then the initial pencil $D - \lambda E$ passes to the final pencil $A - \lambda M$. Let

$$\Gamma := \{(x, \lambda, t) \in \mathbb{R}^n \times \mathbb{R} \times [0, 1] \mid H(x, \lambda, t) = 0\}. \tag{2.3}$$

We claim

THEOREM 2.1. *If the pencil $A(t) - \lambda M(t)$ has no multiple eigenvalues for all $t \in [0, 1]$, then the point $0 \in \mathbb{R}^n \times \mathbb{R}$ is a regular point of H i.e., the Jacobian matrix*

$$D_{(x, \lambda)} H = \begin{bmatrix} A(t) - \lambda M(t) & -M(t)x \\ -x^T E & 0 \end{bmatrix} \tag{2.4}$$

is nonsingular for each $(x, \lambda, t) \in \Gamma$. Furthermore, the set Γ is a one dimensional smooth manifold and can be parameterised by the variable t .

PROOF. By a slight modification of the proofs in [11, 12].

Differentiating (2.2) with respect to t , using Theorem 2.1, we obtain

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{bmatrix} = \begin{bmatrix} C(t) & -M(t)x(t) \\ -x(t)^T E & 0 \end{bmatrix}^{-1} \begin{bmatrix} -B(t)x(t) \\ 0 \end{bmatrix}, \tag{2.5}$$

where $C(t) := A(t) - \lambda(t)M(t)$ and $B(t) := (A - D) - \lambda(t)(M - E)$.

In the following we want to estimate the bounds of $|\dot{\lambda}(t)|$ and $|\dot{x}(t)|$. If $\text{rank}(M) = n - k$, one can prove that (see Section 4) the pencil $A - \lambda M$ has r finite eigenvalues, where $n - 2k \leq r \leq n - k$. Hence there are r bounded curves $\lambda(t) := x^T(t)A(t)x(t)/x^T(t)M(t)x(t)$ in the homotopy equation (2.2). Since A and M have no common nullspace (by (1.2)), for these r bounded curves there exists a positive number 2μ such that $x^T(1)Mx(1) \geq 2\mu > 0$. Here μ is chosen as large as possible. Since $x(t)$ and $M(t)$ are continuous in t , there exists $t_0 \in [0, 1)$ such that $x^T(t)M(t)x(t) \geq \mu > 0$, for all $t \in [t_0, 1]$. Let $\sigma(M, E) = \{m_1, \dots, m_n\}$ with $0 \leq m_1 \leq \dots \leq m_n$. We have $\sigma(M(t), E) = \{(1 - t) + tm_1, \dots, (1 - t) + tm_n\}$. Define κ by

$$1/\kappa \equiv \min\{1, \mu, (1 - t_0) + t_0 m_1\} \cdot \min(\sigma(E, I)). \tag{2.6}$$

In order to simplify the notations in the following paragraph we use $\| \cdot \|$ for $\| \cdot \|_E$ (E is positive definite, defined in (2.2)). Now, for $t \in [0, 1)$ we have

$$\begin{aligned} |\dot{\lambda}(t)| &= \left| \frac{x^T B(t)x}{x^T M(t)x} \right| \\ &\leq \frac{\|E^{-1/2}x\|_2 \|E^{1/2}x\|_2}{|x^T M(t)x|} \|E^{1/2}B(t)E^{-1/2}\|_2 \\ &\leq \kappa(\|A - D\| + |\lambda(t)| \|M - E\|). \end{aligned}$$

Similarly,

$$|\lambda(t)| = \left| \frac{x^T A(t)x}{x^T M(t)x} \right| \leq \kappa(\|A\| + \|D\|) \equiv \tilde{K}.$$

So we obtain an upper bound of $|\dot{\lambda}(t)|$

$$|\dot{\lambda}(t)| \leq \kappa(\|A - D\| + \tilde{K}\|M - E\|). \tag{2.7}$$

Now we want to estimate the upper bound of $\|\dot{x}(t)\|$. For those curves $\lambda(t)$ not tending to infinity at $t = 1$, the matrix

$$C \equiv \begin{bmatrix} C(t) & -M(t)x(t) \\ -x(t)^T E & 0 \end{bmatrix}$$

is invertible for $t \in [0, 1]$. Given t fixed, let $\sigma_i \in \sigma(C(t), E)$ and y_i be the corresponding eigenvector with $y_i^T E y_i = 1$, for $i = 1, \dots, n$. By assumption the rank of $C(t)$ is $n - 1$, we let $\sigma_n = 0$ and $y_n = x$. In the following theorem we give a representation of the inverse of \tilde{C} and \dot{x} , which cannot be directly derived by the proof in [9].

THEOREM 2.2. *Let*

$$\Sigma^{-1} := \text{diag}(\sigma_1^{-1}, \dots, \sigma_{n-1}^{-1}), \quad Y := [y_1, \dots, y_{n-1}], \quad \alpha := \sqrt{x^\top M(t)x}$$

and

$$b := [y_1^\top M(t)x, \dots, y_{n-1}^\top M(t)x]^\top.$$

Then

$$\tilde{C}^{-1} = \left[\begin{array}{c|c} Y \Sigma^{-1} Y^\top - \alpha^{-2} Y \Sigma^{-1} b x^\top & -x \\ \hline -\alpha^{-2} x^\top & 0 \end{array} \right] \tag{2.8}$$

and

$$\dot{x} = -Y \Sigma^{-1} Y^\top B(t)x + \alpha^{-2} Y \Sigma^{-1} b x^\top B(t)x. \tag{2.9}$$

PROOF. Let

$$V = \left[\begin{array}{c|c} y_1, \dots, y_{n-1}, x & 0 \\ \hline 0 & 1 \end{array} \right] \quad \text{and} \quad \Phi = V^\top \tilde{C} V.$$

Then

$$\tilde{C}^{-1} = (V^{-\top} \Phi V^{-1})^{-1} = V \Phi^{-1} V^\top.$$

Compute Φ and Φ^{-1} directly by $\Phi = V^\top \tilde{C} V$. We have

$$\Phi = \left[\begin{array}{c|cc} \Sigma & 0 & -b_1 \\ & \vdots & \vdots \\ & 0 & -b_{n-1} \\ \hline 0 & 0 & -\alpha^2 \\ & -1 & 0 \end{array} \right] \quad \text{and} \quad \Phi^{-1} = \left[\begin{array}{c|cc} \Sigma^{-1} & p & 0 \\ \hline 0 & 0 & -1 \\ & -\alpha^{-2} & 0 \end{array} \right],$$

where $p = -\alpha^{-2} \Sigma^{-1} b$. Then

$$\begin{aligned} \tilde{C}^{-1} &= \left[\begin{array}{c|c} Y \ x & 0 \\ \hline 0 & 1 \end{array} \right] \left[\begin{array}{c|cc} \Sigma^{-1} & p & 0 \\ \hline 0 & 0 & -1 \\ & -\alpha^{-2} & 0 \end{array} \right] \left[\begin{array}{c|c} Y^\top & 0 \\ \hline x^\top & 1 \end{array} \right] \\ &= \left[\begin{array}{c|cc} Y \Sigma^{-1} & Y p & -x \\ \hline 0 & -\alpha^{-2} & 0 \end{array} \right] \left[\begin{array}{c|c} Y^\top & 0 \\ \hline x^\top & 1 \end{array} \right] \\ &= \left[\begin{array}{c|c} Y \Sigma^{-1} Y^\top - \alpha^{-2} Y \Sigma^{-1} b x^\top & -x \\ \hline -\alpha^{-2} x^\top & 0 \end{array} \right]. \end{aligned}$$

From (2.5) we have

$$\dot{x} = -Y \sum^{-1} Y^T B(t)x + \alpha^{-2} Y \sum^{-1} b x^T B(t)x.$$

Now, we define

$$\begin{cases} d(t) := \min(|\sigma_1|, \dots, |\sigma_{n-1}|), \\ \nu := [\|M(t)x\|_{E^{-1}}^2 - (x^T M(t)x)^2]^{1/2}, \\ K := \max(\tilde{K}, d(t)\|M - E\|). \end{cases} \tag{2.10}$$

It can be verified that $\nu = \|b\|_2$. From (2.9) and $Y^T E Y = I$ it follows that

$$\begin{aligned} \|\dot{x}\| &\leq \left\| \sum^{-1} Y^T E^{1/2} E^{-1} E^{1/2} B(t) E^{-1/2} E^{1/2} x \right\|_2 \\ &\quad + \left\| \sum^{-1} \right\|_2 \|b\|_2 \|x^T E^{-1/2}\|_2 \|B(t)\| \\ &\leq \left\| \sum^{-1} \right\|_2 \|B(t)\| \|E^{-1}\|_2 + \kappa \nu \left\| \sum^{-1} \right\|_2 \|B(t)\| \\ &\leq d(t)^{-1} [\min(\sigma(E, I))^{-1} + \kappa \nu] \|B(t)\| \\ &\leq d(t)^{-1} [\min(\sigma(E, I))^{-1} + \kappa \nu] (\|A - D\| + K \|M - E\|). \end{aligned}$$

Furthermore, from the definition of κ in (2.6), we have

$$\|\dot{x}\| \leq d(t)^{-1} \kappa (1 + \nu) (\|A - D\| + K \|M - E\|). \tag{2.11}$$

Let

$$\eta := \max(1, \kappa(1 + \nu) \|M(t)\|). \tag{2.12}$$

From (2.7), (2.10), (2.11) and the definition of η we then have

$$\|\dot{x}\| \leq d(t)^{-1} \eta (\|A - D\| + K \|M - E\|) \tag{2.13}$$

and

$$\|\dot{\lambda}\| \leq \eta (\|A - D\| + K \|M - E\|). \tag{2.14}$$

By using (2.13), (2.14) and the representation of \tilde{C}^{-1} in (2.8) and by repeatedly differentiating (2.5) with respect to t , we obtain the bounds of the m -th derivatives of $\lambda(t)$ and $x(t)$, which are good measures of the local conditionings of eigenvalue and eigenvector curves.

THEOREM 2.3.

$$|\lambda^{(m)}(t)| \leq q(m) \eta^{2m-1} [(\|A - D\| + K \|M - E\|)^m / d(t)^{m-1}] \tag{2.15}$$

$$\|x^{(m)}(t)\| \leq q(m) \eta^{2m-1} [(\|A - D\| + K \|M - E\|)^m / d(t)^m], \tag{2.16}$$

where $q(m)$ is a constant depending on m .

REMARK 2.1. (1) The bound of $|\lambda^{(m)}(t)|$ indicates the smoothness of the eigenvalue curve $\lambda(t)$. Therefore, the poorer the separation of the nonzero eigenvalues $\{\sigma_i\}_{i=1}^{n-1} \subset \sigma(C(t), E)$ from zero, the poorer the local conditioning of the eigenvalue curve can be. The closer D is to A and E is to M , the better the local conditioning of the eigenvalue curve can be. Furthermore, the bound for $|\lambda^{(m)}(t)|$ is independent of the size of the matrix, so growth in matrix size does not imply that the eigenvalue curve becomes ill-conditioned.

(2) For the special case $M = E = I$ the quantity η in (2.12) is equal to 1, and $d(t)$ in (2.10) is just the separation of the eigenvalues of $C(t)$. Hence the bounds for $|\lambda^{(m)}|$ and $\|x^{(m)}\|$ in (2.15), (2.16) can be reduced to the bounds in [9] for the special homotopy (2.2) with $M = E = I$.

3. Prediction and correction

In this section we shall use an approach similar to the Li-Rhee algorithm [9]. Suppose that we have found $(\lambda(\nu), x(\nu))$ for $\nu \in [0, 1)$. We compute the eigenpair $(\lambda(\nu + h), x(\nu + h))$ with the stepsize $h > 0$ by following the eigenvalue curves and the corresponding eigenvector curves.

3.1 Eigenvalue prediction

As in [9], for a given $\nu \in [0, 1)$ and a given stepsize $h > 0$ such that $0 < \nu + h \leq 1$, we compute $\dot{\lambda}(\nu)$ by using (2.5) and predict $\lambda_0(\nu + h)$ by Hermite interpolation $P(t)$ at $\{\lambda(\mu), \dot{\lambda}(\mu), \lambda(\nu), \dot{\lambda}(\nu)\}$ (μ is the previous step). For the case $\nu = 0$ we use the third-order Taylor expansion to predict $\lambda_0(h)$. Here $\lambda^{(2)}(0)$ and $\lambda^{(3)}(0)$ are easily computed by differentiating (2.5).

3.2 Stepsize updating

The stepsize prediction is somewhat difficult to handle in the homotopy method. Here, we give another method to predict the stepsize, which is different from the method in [9]. Although the eigenvalue curves have the order preserving property, our final purpose is to find all/some eigenvalues at $t = 1$. It is not necessary to cut the stepsize by half as does the method in [9] and spend much more computational time by using the prediction and correction checking algorithm to follow the eigenpair curve and preserve its order (for A and M with general structure, the prediction and correction are always the most expensive step in the whole homotopy algorithm). Indeed, when two curves come near at some point $t \in [0, 1)$, a jump to a neighboring curve may occur (see Section 5 Fig. 1, 2, 3). In practice, we only perform the checking algorithm at $t = 1$ to check whether multiple arrival of eigenpair curves occur (see Section 5 for details).

The difference between the eigenvalue curve $\lambda(t)$ and the Hermite interpolation $P(t)$ is $(h+(\nu-\mu))^2 h^2 \lambda^{(4)}(\xi)/24$, where $\xi \in [\mu, \nu]$ (see [16, page 52]). For a given upper bound of this difference, we can compute h by a standard stepsize control technique [16, page 428] with the initial value $|\lambda^{(4)}(0)|$, and update the approximate $|\lambda^{(4)}(\xi)|$ by $24|\lambda(\nu+h) - P(\nu+h)|/(h+(\nu-\mu))^2 h^2$ for the next step, where $\lambda(\nu+h)$ is available by the following correction algorithm.

3.1 Correction

By using a similar method to that in [9] we can correct the predicted eigenvalue to the desired eigenpair. Suppose that $(\lambda(\nu), x(\nu))$ is an eigenpair of the pencil $A(\nu) - \lambda M(\nu)$ ($0 \leq \nu < 1$). After the eigenvalue prediction we have an approximate eigenvalue $\lambda_0(\nu+h)$ of the pencil $A(\nu+h) - \lambda M(\nu+h)$ and $\dot{x}(\nu)$ computed by (2.5) at $t = \nu$. In fact, $x(\nu)$ satisfies $x(\nu)^T E x(\nu) = 1$ and $\dot{x}(\nu)^T E x(\nu) = 0$. So if $\|\dot{x}(\nu)\|$ is not too large, then $(x(\nu) + h\dot{x}(\nu))^T E (x(\nu) + h\dot{x}(\nu)) \approx 1$, i.e. $x(\nu) + h\dot{x}(\nu)$ is a good approximation of the eigenvector of $A(\nu+h) - \lambda M(\nu+h)$. In this case, we perform the generalised Rayleigh quotient iteration (GRQI) [14, page 317], starting from the approximation $x(\nu) + h\dot{x}(\nu)$. If $\|\dot{x}(\nu)\|$ is large, then we prefer to consider the approximation $(\lambda_0(\nu+h), x(\nu))$. We first update the eigenvector by the generalised inverse iteration [15]; secondly, we correct the updated eigenvector by GRQI starting from the updated eigenvector.

REMARK 3.1. (1) The linear system of GRQI can be solved by the usual sparse solver. When the eigenvalue $\lambda(\nu+h)$ is acceptable, we can use the current LU -factorisation of $A(\nu+h) - \lambda(\nu+h)M(\nu+h)$ and the formula described in [11] to compute $\dot{x}(\nu+h)$ and $\dot{\lambda}(\nu+h)$ for the starting vector of the next step.

(2) The asymptotic rate of convergence of GRQI is cubic for a generalised symmetric positive definite pencil $A(\nu) - \lambda M(\nu)$ ($0 \leq \nu < 1$) [14, page 317].

(3) When $\nu = 1$, convergence is no longer cubic because $M(1) = M$ is positive semidefinite. But we still have quadratic convergence with the generalised Rayleigh quotient $x^T M C A x / x^T M C M x$ for a given vector x , introduced by Geltner [6], where C is positive definite. Suppose that x is in the range of M (i.e. $Mx \neq 0$). If we choose $C := U \begin{pmatrix} \Gamma^{-1} & 0 \\ 0 & I \end{pmatrix} U^T > 0$, where U is unitary and satisfies $U^T M U = \begin{pmatrix} \Gamma & 0 \\ 0 & 0 \end{pmatrix}$ with $\Gamma = \text{diag}(\gamma_i) > 0$, then we have the generalised Rayleigh quotient $x^T A x / x^T M x$. By Theorem 2 in [6], GRQI converges quadratically, if x is sufficiently close to an eigenvector of $A - \lambda M$.

4. Checking algorithm

Up to now, there are no effective theorems and algorithms to specify the sign-preserving property of the eigenvalues of $A - \lambda M$ by congruent transformations. In this section, we first show that the pencil $A - \lambda M$ has the following special Kronecker structure, and then give a checking algorithm to check the number of eigenvalues of $A - \lambda M$ in a prescribed interval.

THEOREM 4.1. *Let $A - \lambda M$ be a $n \times n$ symmetric pencil as in (1.1) with $\text{rank}(M) = n - k$ and $\mathcal{N}(A) \cap \mathcal{N}(M) = \{0\}$. Then $A - \lambda M$ is a regular pencil and is real congruent to the pencil $A_c - \lambda M_c$ with*

$$A_c = \text{diag} \left\{ \lambda_1, \dots, \lambda_r, \overbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}^{k_3}, \overbrace{1, \dots, 1}^{k_1}, \overbrace{-1, \dots, -1}^{k_2} \right\} \tag{4.1}$$

and

$$M_c = \text{diag} \left\{ 1, \dots, 1, \overbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \dots, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}}^{k_3}, \overbrace{0, \dots, 0}^{k_1}, \overbrace{0, \dots, 0}^{k_2} \right\}, \tag{4.2}$$

where $k_1 + k_2 + k_3 = k$ and $k_1 + k_2 + 2k_3 + r = n$.

PROOF. Since $\mathcal{N}(A) \cap \mathcal{N}(M) = \{0\}$ and $\text{rank}(M) = n - k$, by the Fix-Heiberger reduction algorithm [14, page 311] there exists a real nonsingular matrix S_1 such that $S_1^T(A - \lambda M)S_1 = A_1 - \lambda M_1$ with the forms

$$A_1 = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \Theta \\ & A_{22} & A_{23} & 0 \\ \text{sym.} & & \Phi & 0 \\ & & & 0 \end{bmatrix} \quad \text{and} \quad M_1 = \begin{bmatrix} I & 0 & 0 & 0 \\ & I & 0 & 0 \\ \text{sym.} & 0 & 0 & 0 \\ & & & 0 \end{bmatrix} \begin{matrix} \} k_3 \\ \} r \\ \} k_1 + k_2 \\ \} k_3 \end{matrix}$$

where Θ is a positive diagonal matrix, Φ is a nonsingular diagonal matrix with k_1 positive and k_2 negative diagonal entries, respectively.

Next, we zero out the blocks $A_{23}, A_{23}^T, A_{13}, A_{13}^T$ and A_{12}, A_{12}^T, A_{11} by row-column transformations using Φ and Θ as pivots, respectively. Now, we reduce the diagonal entries of Φ and Θ to ± 1 and preserve their signs. Finally, transform A_{22} to $\text{diag}\{\lambda_1, \dots, \lambda_r\}$ and perform some suitable permutations; we then have a pencil $A_c - \lambda M_c$ of the form (4.1), (4.2) which is real congruent to $A - \lambda M$.

For a given real number $\alpha \in \mathbb{R}$ we denote the number of finite eigenvalues of $A - \lambda M$ which are larger and smaller than α by $\pi(A - \alpha M)$ and

$\omega(A - \alpha M)$, respectively; the number of positive and negative eigenvalues of a symmetric matrix A by $\# \text{pos}(A)$ and $\# \text{neg}(A)$, respectively; and the dimension of the nullspace of A by $\#n(A)$.

THEOREM 4.2. *Let $A - \lambda M$ be the symmetric pencil as in (1.1). If $A - \alpha M$ has a triangular factorisation $A - \alpha M = L_\alpha \Delta_\alpha L_\alpha^T$, where $\alpha \in \mathbb{R}$ and Δ_α is diagonal, then*

$$\pi(A - \alpha M) = \# \text{pos}(\Delta_\alpha) - (k_1 + k_3), \tag{4.3}$$

$$\omega(A - \alpha M) = \# \text{neg}(\Delta_\alpha) - (k_2 + k_3), \tag{4.4}$$

where k_1, k_2 , and k_3 are defined in (4.1) and (4.2).

PROOF. By Theorem 4.1 the pencil $A - \lambda M$ is real congruent to $A_c - \lambda M_c$. There is a real nonsingular matrix S such that $S^T(A - \lambda M)S = A_c - \lambda M_c$. Since $A - \alpha M = L_\alpha \Delta_\alpha L_\alpha^T$, we have $A_c - \alpha M_c = S^T(L_\alpha \Delta_\alpha L_\alpha^T)S$. By the well-known Sylvester's inertia theorem for symmetric matrices we have the following equalities

$$\# \text{pos}(\Delta_\alpha) = \# \text{pos}(A - \alpha M) = \# \text{pos}(A_c - \alpha M_c) \tag{4.5}$$

and

$$\# \text{neg}(\Delta_\alpha) = \# \text{neg}(A - \alpha M) = \# \text{neg}(A_c - \alpha M_c). \tag{4.6}$$

The matrix

$$A_c - \alpha M_c = \text{diag} \left\{ \lambda_1 - \alpha, \dots, \lambda_r - \alpha, \right.$$

$$\left. \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & -\alpha \end{bmatrix}, \dots, \begin{bmatrix} 0 & 1 \\ 1 & -\alpha \end{bmatrix}}_{k_3}, \underbrace{1, \dots, 1}_{k_1}, \underbrace{-1, \dots, -1}_{k_2} \right\}$$

has the eigenvalues

$$\left\{ \lambda_1 - \alpha, \dots, \lambda_r - \alpha, \underbrace{\mu_1, \dots, \mu_1}_{k_3}, \underbrace{\mu_2, \dots, \mu_2}_{k_3}, \underbrace{1, \dots, 1}_{k_1}, \underbrace{-1, \dots, -1}_{k_2} \right\}$$

where $\mu_1 = (-\alpha + (\alpha^2 + 4)^{1/2})/2$ and $\mu_2 = (-\alpha - (\alpha^2 + 4)^{1/2})/2$.

It has been shown in Theorem 4.1 that $\{\lambda_1, \dots, \lambda_r\}$ are the finite eigenvalues of $A - \lambda M$ and from above, μ_1 has the opposite sign of μ_2 . It follows from (4.5), (4.6) that

$$\# \text{pos}(\Delta_\alpha) = \# \text{pos}(A_c - \alpha M_c) = \pi(A - \alpha M) + (k_1 + k_3) \tag{4.5}$$

and

$$\# \text{neg}(\Delta_\alpha) = \# \text{neg}(A_c - \alpha M_c) = \omega(A - \alpha M) + (k_2 + k_3).$$

COROLLARY 4.3. *Let (α, β) be an open interval in \mathbb{R} . Suppose that $A - \alpha M$ and $A - \beta M$ have the following triangular factorisations*

$$A - \alpha M = L_\alpha \Delta_\alpha L_\alpha^T, \quad A - \beta M = L_\beta \Delta_\beta L_\beta^T, \quad (4.7)$$

where Δ_α and Δ_β are diagonal. Then the pencil $A - \lambda M$ has exactly $\# \text{pos}(\Delta_\alpha) - \# \text{pos}(\Delta_\beta) - \#n(\Delta_\beta)$ finite eigenvalues in (α, β) .

PROOF. It is clear that the pencil $A - \lambda M$ has exactly $\pi(A - \alpha M) - \pi(A - \beta M) - \dim \mathcal{N}(A - \beta M)$ finite eigenvalues in (α, β) . From (4.3) this quantity is equal to $\# \text{pos}(\Delta_\alpha) - \# \text{pos}(\Delta_\beta) - \#n(\Delta_\beta)$.

REMARK 4.1. (1) We frequently choose $\eta = n^* \text{eps}$ or $\sqrt{\text{eps}}$, where eps is the machine precision, as a criterion for negligibility of the diagonal entries of Δ_α and Δ_β in the triangular factorisations (4.7).

(2) If some diagonal entries of $A - \alpha M$ and $A - \beta M$ in (4.7) are close to zero, then the factors can have arbitrarily large magnitudes. Bunch and Kaufman [1] proposed a stable method for calculating the inertia of a symmetric matrix H , that is, there exists a permutation π , such that $\pi H \pi^T = LDL^T$, where D is block diagonal with 1 by 1 and 2 by 2 blocks and L is unit lower triangular. In practice, one can use this stable diagonal pivoting method to modify the unstable triangular factorisations in (4.7).

(3) If A and M in Corollary 4.3 are both tridiagonal, then we can use a Sturm sequence [14, page 131] to calculate the inertia of a symmetric tridiagonal matrix, which is much cheaper than the above triangular factorisation.

Now, we assume that the initial eigenpairs (δ_i, z_i) for $i = 1, \dots, n$ of the pencil $D - \lambda E$ in (2.2) are arranged in the order $\delta_1 < \dots < \delta_n$. As we mentioned in the stepsize updating algorithm in Section 3, for a given interval (α, β) we want to check all eigenvalues in (α, β) and locate the lost eigenvalues by using Corollary 4.3.

Checking algorithm

Given an open interval (α, β) , where α, β are not eigenvalues of $A - \lambda M$.

Given a number $0 < n_0 < n$.

Comment: In general, the final order (at $t = 1$) of an eigenvalue is close to its initial order (at $t = 0$), while some jumps occur. If we check some $\lambda_j > \beta$ or $\lambda_j < \alpha$, we guarantee that $\lambda_i > \beta$ or $\lambda_i < \alpha$ for $i \geq j + n_0$ or $i \leq j - n_0$. Calculate the number of eigenvalues of $A - \lambda M$ in (α, β) : $r_0 := \# \text{pos}(\Delta_\alpha) - \# \text{pos}(\Delta_\beta)$ as in Corollary 4.3.

Choose an index $m \in \{1, \dots, n\}$ ($0 < m < 1$) (e.g. bisect the set $\{1, \dots, n\}$) so that $\lambda_m \in (\alpha, \beta)$, where (λ_m, x_m) is achieved by follow-

ing the homotopy (2.2) using the method in Section 3 with initial vector (δ_m, z_m) .

Continue.

For $j = m + k$ and $j = m - k$,

If $l_{\min} \leq j \leq l_{\max}$, then follow the homotopy curve in (2.2) until $t = 1$ using the method in Section 3 with initial vector (δ_j, z_j) . Let (λ_j, x_j) be the acceptable eigenpair of $A - \lambda M$. If $\lambda_j \in (\alpha, \beta)$ then $r := r + 1$.

Else if $\lambda_j > \beta$ then $l_{\max} := \min\{n, j + n_0\}$.

Else if $\lambda_j < \alpha$ then $l_{\min} := \max\{1, j - n_0\}$.

$k := k + 1$, go to continue.

Arrange all eigenvalues $\lambda_i \in (\alpha, \beta)$ in the order $\lambda_1 \leq \dots \leq \lambda_r$ (say!).

If $r = r_0$ then stop.

Else comment: Multiple arrival of the same eigenpairs occurs.

For $i = 1, \dots, r$

Compute the triangular factorization $A - \lambda_i M = L_i \Delta_i L_i^T$.

For $i = 1, \dots, r - 1$

If $(\#\text{pos}(\Delta_i) - \#\text{pos}(\Delta_{j+1}) - \#n(\Delta_{j+1})) \geq 1$ then use bisection technique on $(\lambda_i, \lambda_{i+1})$ to approach the lost eigenvalues between λ_i and λ_{i+1} , and then compute the corresponding eigenvectors by generalised inverse iteration starting from a randomly chosen vector q satisfying $q^T M x_i = q^T M x_{i+1} = 0$.

5. Numerical results

A program based on the methods developed in the last two sections has been implemented on a CDC Cyber 184/840 with machine precision 2^{-49} . A typical example is first given to illustrate how our algorithm follows these homotopy curves (see Tables 1, 2, 3). Then we present two tables, which record the execution times needed to solve pencils $A - \lambda M$ with various dimensions by our algorithm and the *QZ* algorithm [13], respectively. Finally, we give a short discussion about storage and parallel processing.

EXAMPLE. A and M are tridiagonal generated by a random number generating program. In addition, M is positive semidefinite. Let

$$A = \begin{bmatrix} a_1 & & & & \\ & \alpha_2 & & & \\ & & \ddots & & \\ \alpha_2 & & & a_2 & & \alpha_{40} \\ & \ddots & & & \ddots & \\ & & & & & \alpha_{40} & a_{40} \end{bmatrix}, \quad M = \begin{bmatrix} m_1 & & & & \\ & \beta_2 & & & \\ & & \ddots & & \\ \beta_2 & & & m_2 & & \beta_{40} \\ & \ddots & & & \ddots & \\ & & & & & \beta_{40} & m_{40} \end{bmatrix}.$$

Initial matrices D and E are taken as follows.

$$D = \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & \ddots & \\ & & & D_{10} \end{bmatrix} \quad \text{and} \quad E = \begin{bmatrix} E_1 & & & \\ & E_2 & & \\ & & \ddots & \\ & & & E_{10} \end{bmatrix},$$

where

$$D_i = \begin{bmatrix} a_j & \alpha_j & & \\ \alpha_j & a_k & \alpha_k & \\ & \alpha_k & a_l & \alpha_l \\ & & \alpha_l & a_n \end{bmatrix} \quad \text{and} \quad E_i = \begin{bmatrix} \tilde{m}_j & \beta_j & & \\ \beta_j & \tilde{m}_k & \beta_k & \\ & \beta_k & \tilde{m}_l & \beta_l \\ & & \beta_l & \tilde{m}_n \end{bmatrix}.$$

Subscripts j, k, l and n are defined by $j = 4(i - 1) + 1, k = 4(i - 1) + 2, l = 4(i - 1) + 3, n = 4i$ and $\tilde{m}_s = m_s + \zeta, s \in [j, k, l, n]$. ζ is chosen to avoid E being singular and the initial pencil $D - \lambda E$ having multiple eigenvalues. The error tolerance ϵ in stepsize updating is given by $\max(0.05, 0.05\lambda(\nu))$. The approximate eigenpair (λ, x) is corrected until

$$\frac{\|Ax - \lambda Mx\|}{\|A\| + |\lambda| \|M\|} \leq 10^{-3} \quad \text{for } t < 1$$

and

$$\frac{\|Ax - \lambda Mx\|}{\|A\| + |\lambda| \|M\|} \leq 10^{-5} \quad \text{for } t = 1.$$

The results of the homotopy method are shown in Tables 1 and 2. Table 3 gives the eigenvalues obtained by the QZ algorithm with the error tolerance 10^{-10} (for eigenvalues).

TABLE 1. The computational results by the homotopy algorithm to $H(\lambda, x, t)$, as in (2.2).

| 00 | 01 | EV ($T = 0$) | EV ($T = 1$) | S | L | M | CPU |
|----|----|--------------------|--------------------|---|---|---|------|
| 1 | 2 | -.2814061774E + 01 | -.1632964183E + 05 | 4 | 5 | N | .160 |
| 2 | 4 | -.1706459761E + 01 | -.1750285450E + 01 | 1 | 1 | N | .056 |
| 3 | 3 | -.1434691579E + 01 | -.2260152378E + 01 | 4 | 7 | N | .186 |
| 4 | 5 | -.1404593349E + 01 | -.1404013320E + 01 | 3 | 5 | N | .145 |
| 5 | 8 | -.1225833240E + 01 | -.1187072310E + 01 | 1 | 5 | N | .114 |
| 6 | 6 | -.1134789968E + 01 | -.1333516093E + 01 | 3 | 5 | N | .145 |
| 7 | 9 | -.9179918930E + 00 | -.1105816505E + 01 | 3 | 4 | N | .141 |
| 8 | 11 | -.8954478872E + 00 | -.9154988779E + 00 | 1 | 1 | N | .051 |
| 9 | 10 | -.8511456577E + 00 | -.9328022950E + 00 | 3 | 3 | N | .116 |
| 10 | 12 | -.6299535768E + 00 | -.6380542267E + 00 | 1 | 1 | N | .051 |
| 11 | 7 | -.5103918388E + 00 | -.1187072310E + 01 | 3 | 8 | Y | .185 |
| 12 | 14 | -.4224461853E + 00 | -.4293582916E + 00 | 1 | 2 | Y | .072 |
| 13 | 13 | -.4170917332E + 00 | -.6013359554E + 00 | 3 | 5 | N | .146 |
| 14 | 19 | -.4098700913E + 00 | -.2347505495E + 00 | 3 | 5 | N | .146 |
| 15 | 15 | -.4005574983E + 00 | -.4293582916E + 00 | 2 | 3 | N | .102 |
| 16 | 17 | -.3285497166E + 00 | -.3341348708E + 00 | 1 | 2 | N | .068 |
| 17 | 16 | -.3117386178E + 00 | -.3670701949E + 00 | 3 | 5 | N | .147 |
| 18 | 18 | -.1575037581E + 00 | -.3223850586E + 00 | 3 | 6 | N | .158 |
| 19 | 21 | -.8954804091E - 01 | -.5805549131E - 01 | 3 | 5 | N | .145 |
| 20 | 20 | -.8652378417E - 01 | -.8915593977E - 01 | 2 | 4 | N | .120 |
| 21 | 24 | -.5894779789E - 01 | .2072323002E + 00 | 3 | 6 | N | .166 |
| 22 | 22 | .1147081411E + 00 | .9818751310E - 01 | 1 | 2 | N | .078 |
| 23 | 23 | .1557098669E + 00 | .1135588505E + 00 | 1 | 2 | N | .070 |
| 24 | 25 | .2590070041E + 00 | .2668755993E + 00 | 1 | 3 | N | .091 |
| 25 | 26 | .3235740702E + 00 | .3218884632E + 00 | 1 | 3 | N | .081 |
| 26 | 27 | .4143842806E + 00 | .3978408154E + 00 | 2 | 2 | N | .085 |
| 27 | 28 | .4695831546E + 00 | .4808380128E + 00 | 3 | 4 | N | .129 |
| 28 | 30 | .4940888909E + 00 | .7060273880E + 00 | 3 | 4 | N | .128 |
| 29 | 29 | .6486366274E + 00 | .6670577309E + 00 | 1 | 1 | N | .058 |
| 30 | 32 | .8272829203E + 00 | .8417961699E + 00 | 1 | 1 | N | .051 |
| 31 | 31 | .8471829777E + 00 | .7110036092E + 00 | 2 | 6 | N | .151 |

Table 1. (Continued)

| 00 | 01 | EV ($T = 0$) | EV ($T = 1$) | S | L | M | CPU |
|--|----|-------------------|--------------------|----|-----|---|-------|
| 32 | 35 | .9266450995E + 00 | .1077463067E + 01 | 1 | 2 | N | .073 |
| 33 | 33 | .9575987247E + 00 | .1059138240E + 01 | 1 | 2 | Y | .072 |
| 34 | 34 | .1136005187E + 01 | .1059138240E + 01 | 5 | 11 | N | .263 |
| 35 | 39 | .1167424686E + 01 | .1928652319E + 01 | 4 | 8 | N | .199 |
| 36 | 37 | .1318443953E + 01 | .1501230584E + 01 | 3 | 4 | N | .128 |
| 37 | 36 | .1333273350E + 01 | .1366191023E + 01 | 1 | 1 | N | .053 |
| 38 | 38 | .1692829722E + 01 | .1719356313E + 01 | 1 | 1 | N | .055 |
| 39 | 40 | .2022759455E + 01 | .2127556976E + 01 | 1 | 1 | N | .051 |
| 40 | 1 | .7243980046E + 01 | -.7887234032E + 05 | 4 | 5 | N | .166 |
| TOTAL | | | | 88 | 151 | 3 | 4.602 |
| AVERAGE STEPS/CURVE = 2.200 AVERAGE L-SYSTEM/STEP = 1.716 CPU OF SORTING AND REMOVING = .010 CPU OF LOCATING LOST E.V. = .360 | | | | | | | |

TABLE 2. The lost eigenvalues computed by the bisection method.

| BIS | II | RLQI | EV |
|-----|----|------|--------------------|
| 5 | 4 | 1 | -.1515427871E + 01 |
| 5 | 4 | 1 | -.5489866188E + 00 |
| 6 | 4 | 1 | .9378133725E + 00 |

CPU OF COMPUTING LOST FINITE E.V. = .340

NOTATIONS.

00 :The order of increasing eigenvalues at $t = 0$.

01 :The order of increasing eigenvalues at $t = 1$.

E.V. :Eigenvalues.

S :The total number of steps for finding an eigenpair.

L :The total number of linear systems solved.

M :Multiple arrival (same eigenpairs)

CPU :Execution time.

BIS :The total number of bisections used.

II :The total number of inverse power iterations

RLQI :The total number of generalised Rayleigh quotient iterations.

In this example, we have an average of 2.2 steps for one curve and solving 1.716 linear systems for one step. Checking and locating all the lost

TABLE 3. The eigenvalues of $A - \lambda M$ computed by the QZ -algorithm.

| | | | |
|----------------------|----------------------|----------------------|----------------------|
| $-.5917182688E + 05$ | $-.1238075896E + 05$ | $-.1160152379E + 01$ | $-.1750285450E + 01$ |
| $-.1515427871E + 01$ | $-.1404013320E + 01$ | $-.1333516093E + 01$ | $-.1187072310E + 01$ |
| $-.1105816505E + 01$ | $-.9328022950E + 00$ | $-.9154988779E + 00$ | $-.6380542267E + 00$ |
| $-.6013359554E + 00$ | $-.5489866188E + 00$ | $-.4293582916E + 00$ | $-.3670701949E + 00$ |
| $-.3341348708E + 00$ | $-.3223850586E + 00$ | $-.2347505495E + 00$ | $-.8915593977E - 01$ |
| $.5805549131E - 01$ | $.9818751310E - 01$ | $.1135588404E + 00$ | $.2072323002E + 00$ |
| $.2668755993E + 00$ | $.3218884632E + 00$ | $.3978408154E + 00$ | $.4808380128E + 00$ |
| $.6670577309E + 00$ | $.7060273880E + 00$ | $.7110036092E + 00$ | $.8417961699E + 00$ |
| $.9378133725E + 00$ | $.1059138240E + 01$ | $.1077463067E + 01$ | $.1366191023E + 01$ |
| $.1501230584E + 01$ | $.1719356313E + 01$ | $.1928652319E + 01$ | $.2127556976E + 01$ |

eigenvalues requires 0.36 sec. This is about one-twelfth of the total execution time of following all homotopy curves. Figures 1, 2 and 3 illustrate, for homotopy curves 7–11, 9–23 and 35–38, that order preserving is not recommended, because there are many places in which $d(t)$ is very small (for some $0 \leq t < 1$) and consequently, a large number of steps are needed to follow these places if order preserving is required. In contrast to order preserving, our algorithm will let the stepsize pass over the interval in which $d(t)$ is very small. Figures 1, 2 and 3 show how this idea is carried out. From Table 1 and Figures 1, 2, 3, we see that homotopy curves 8, 10, 37 and 38 lead, respectively, to curves 10, 11, 35 and 37 and the corresponding eigenpairs are obtained by solving only one linear system. Curves 22 and 23 lead, respectively, to curves 21 and 22, and the eigenpairs are obtained one step which involves solving two linear systems.

That some eigenvalues may be lost is a disadvantage of not preserving the order. In this example we lose three eigenvalues. Fortunately, from Tables 1 and 2, we see that almost the same execution time of following one homotopy curve is needed to compute one lost eigenvalue by using the checking algorithm. In addition, numerical experience shows that the ratio of the number of lost eigenvalues and the dimension for tridiagonal and five-diagonal matrices is in the range $[0, 0.1]$. This shows that the bisection method applied to inverse iteration and GRQI is a good method to compute the lost eigenvalues, by using the computed eigenvalues as lower and upper bounds. Tables 1 and 3 show that the eigenvalues obtained by our method and by the QZ algorithm coincide up to about 10 significant digits. In general, the order of the eigenvalues at $t = 0$ and $t = 1$ for each homotopy curve may not be the same, but they are very close. This fact is very useful when we want to compute a specified number of eigenvalues in an interval (α, β) . We introduce briefly this procedure, here. We first determine the number of eigenvalues in (α, β) and their orders, r_1 to r_2 , by the checking

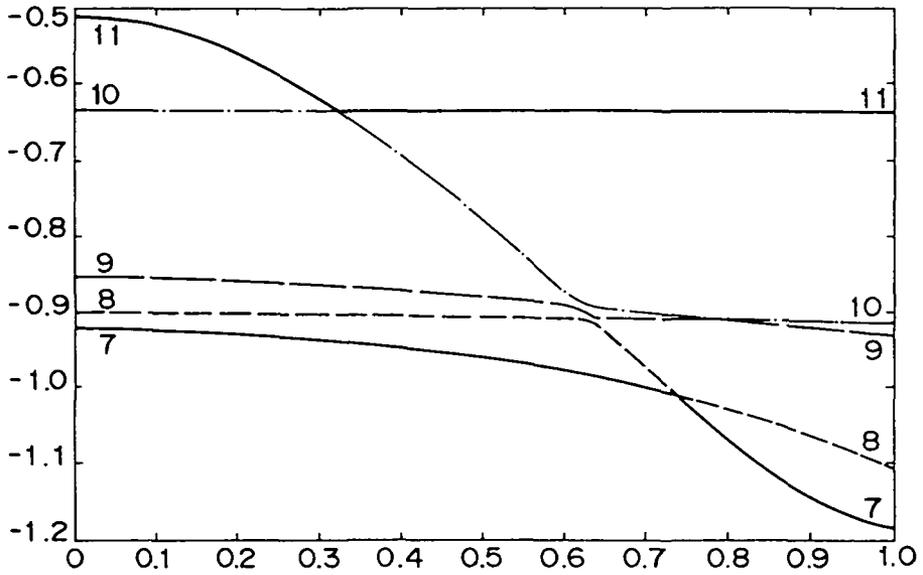


FIGURE 1. Curves 7 to 11.

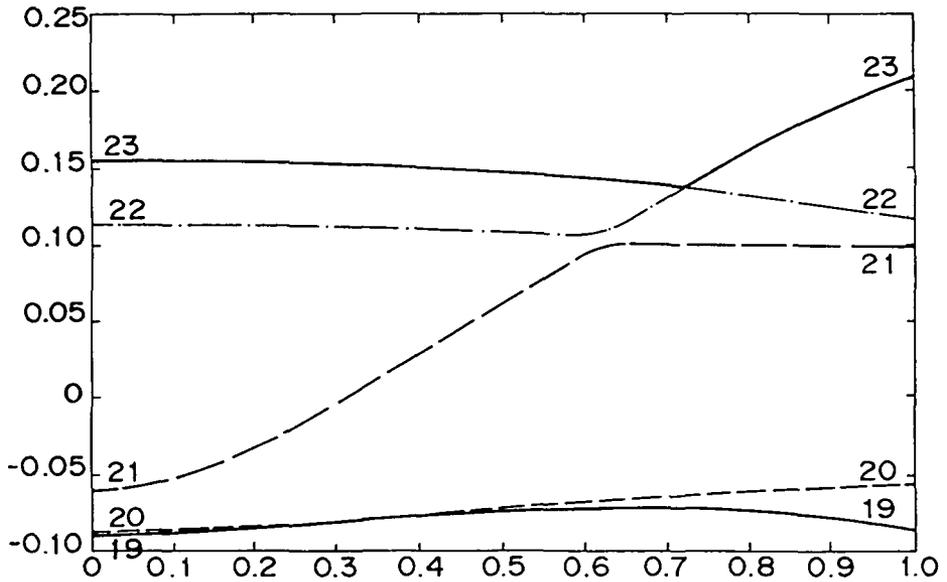


FIGURE 2. Curves 19 to 23.

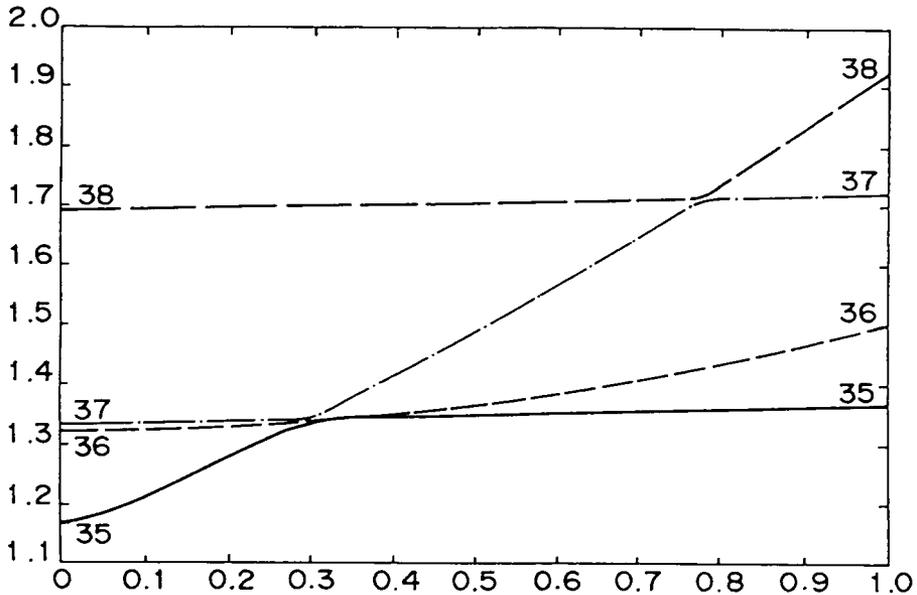


FIGURE 3. Curves 35 to 38.

algorithm, and then follow homotopy curves whose initial orders are close to the range r_1 to r_2 . Finally, we locate the lost eigenvalues by the checking algorithm and compute them by the bisection method.

Tables 4 and 5 record, respectively, execution times spent by our algorithm and *QZ* algorithm [13] to find all eigenpairs for tridiagonal and five-diagonal pencils. A and M are generated in the same way as the last example, but A , here, is diagonally dominant. The other parameters are the same as in the example. These two tables show that as the dimension gets larger, the homotopy algorithm gets better.

TABLE 4. The ratio of execution times between the homotopy method and the *QZ*-algorithm.

| N | IE | IBS | CPU (HM) | CPU (QZ) | RATIO(QZ/HM) |
|-----|----|-----|----------|----------|--------------|
| 30 | 2 | 3 | 2.215 | 2.300 | 1.04 |
| 60 | 3 | 4 | 10.125 | 17.290 | 1.71 |
| 120 | 3 | 4 | 50.984 | 138.420 | 2.71 |
| 180 | 4 | 4 | 151.918 | 448.383 | 2.95 |
| 220 | 4 | 4 | 252.716 | ***** | **** |

NOTATIONS.

N : The dimension.

IE : The total number of infinite eigenvalues.

IBS : The dimension of initial submatrix.

CPU(HM) : Execution time spent by homotopy method.

CPU(QZ) : Execution time spent by QZ algorithm.

RATIO(HM/QZ) : Ratio of execution times.

TABLE 5. The ratio of execution times between the homotopy method and the QZ-algorithm.

| N | IE | IBS | CPU (HM) | CPU (QZ) | RATIO(QZ/HM) |
|-----|----|-----|----------|----------|--------------|
| 60 | 3 | 4 | 14.225 | 18.232 | 1.28 |
| 120 | 4 | 4 | 77.652 | 138.218 | 1.78 |
| 180 | 4 | 4 | 210.418 | 447.300 | 2.13 |
| 220 | 4 | 4 | 351.041 | ***** | **** |

Since the storage for the QZ algorithm is at least $3n^2$, the case $n > 180$ can not be dealt with on our machine. But for our algorithm, $2n^2 + O(n)$ storage is needed. Moreover, only $n^2 + O(n)$ storage is necessary if the computed eigenvectors are stored in the second storage. Essentially, the complexity of following one homotopy curve is $O(n)$. Hence the complexity of finding all eigenpairs of a pencil by our algorithm is $O(n^2)$. Because checking and bisection procedures can also be performed in parallel processing, the complexity becomes $O(n)$ if there are n parallel processors.

References

- [1] J. R. Bunch and L. Kaufman, "Some stable methods for calculating inertia and solving symmetric linear systems", *Math. Comp. J.* **31** (1977) 163–179.
- [2] A. Bunse-Gerstner, "An algorithm for the symmetric generalized eigenvalue problem", *Lin. Alg. Appl.* **58** (1984) 43–68.
- [3] M. T. Chu, "A simple application of the homotopy method to symmetric eigenvalue problems", *Lin. Alg. Appl.* **59** (1984) 85–90.
- [4] M. T. Chu, T. Y. Li and T. Sauer, "Homotopy method for general λ -matrix problem", *SIAM J. Matrix Anal. Appl.* **9** (1988) 528–536.
- [5] G. Fix and R. Heiberger, "An algorithm for the ill-conditioned generalized eigenvalue problem", *SIAM J. Numer. Anal.* **9** (1972) 78–88.
- [6] P. B. Geltner, "General Rayleigh quotient iteration", *SIAM J. Numer. Anal.* **18** (1981) 839–843.
- [7] R. Kalaba, K. Spingarn, and L. Tesfatsion, "Variational equations for the eigenvalues and eigenvectors of non-symmetric matrices", *J. Optimiz. Theory and Appl.* **33** (1981) 1–8.

- [8] R. Kalaba, K. Spingarn, and L. Tesfatsion, "Individual tracking of an eigenvalue and eigenvector of a parameterize matrix", *Nonlinear Analysis, Theory, Methods and Applications* **5** (1981) 337–340.
- [9] T. Y. Li and N. Rhee, "Homotopy algorithm for symmetric eigenvalue problems", *Numer. Math.* **55** (1989) 265–280.
- [10] T. Y. Li and T. Sauer, "Homotopy method for generalized eigenvalue problems", *Lin. Alg. Appl.* **91** (1987) 65–74.
- [11] W. W. Lin and G. Lutzer, "An application of the homotopy method to the generalized symmetric eigenvalue problem", *J. Austral. Math. Soc. Ser. B*, **30** (1987) 232–249.
- [12] J. Milnor, *Topology from the differentiable viewpoint*, (Univ. of Virginia Press, 1965).
- [13] C. B. Moler and G. W. Stewart, "An algorithm for generalised matrix eigenvalue problems", *SIAM J. Numer. Anal.* **10** (1973) 241–256.
- [14] B. N. Parlett, *The Symmetric Eigenvalue Problem*, (Prentice-Hall, Inc., Englewood Cliffs, 1980).
- [15] G. Peters and J. H. Wilkinson, "Inverse iteration, ill-conditioned equations and Newton's method", *SIAM Review*, **21** (1979) 339–360.
- [16] J. Stoer and R. Bulirsch, "Introduction to numerical analysis", Translated by R. Bartels, W. Gautschi and C. Witzgall (1980).