J. Austral. Math. Soc. Ser. B 30(1988), 230–249

AN APPLICATION OF THE HOMOTOPY METHOD TO THE GENERALISED SYMMETRIC EIGENVALUE PROBLEM

WEN-WEI LIN[1] AND GERHARD LUTZER[1]

(Received 20 November 1986; revised 27 May 1987 and 19 January 1988)

Abstract

The homotopy method is used to find all eigenpairs of a generalised symmetric eigenvalue problem $Ax = \lambda Bx$ with positive definite B. The determination of n eigenpairs (x, λ) is reduced to curve tracing of n disjoint smooth curves in $\mathbf{R}^n \times \mathbf{R} \times [0, 1]$. The method might be attractive if A and B are sparse symmetric. In this paper it is shown that the method will work for almost all symmetric tridiagonal matrices A and B.

1. Introduction

Solving a generalised symmetric eigenvalue problem with positive definite B,

$$Ax = \lambda Bx, \tag{1.1}$$

can be regarded as solving a nonlinear algebraic equation

$$f(x, \lambda) \triangleq [(Ax - \lambda Bx, (1 - x^{\mathsf{T}}x)/2] = 0, \tag{1.2}$$

where A, B are two $n \times n$ symmetric matrices in $\mathbf{R}$, $x \in \mathbf{R}^n$ and $\lambda \in \mathbf{R}$.

The classical Newton's method for solving (1.2) has two difficulties: (1) it is only locally convergent, (2) it is not obvious how to choose the starting vectors (x_0, λ_0). In this paper we present a homotopy method that is globally convergent and where the starting vectors are obvious. We construct a homotopy from a trivial function, whose roots are easy to determine, to f. It can be shown that there exist n disjoint smooth curves characterised by a differential equation. Therefore one can trace them by an ODE software solver.

[1]Institute of Applied Mathematics, National Tsing Hua University, Hsinchu, Taiwan 30043, R.O.C.

© Copyright Australian Mathematical Society 1988, Serial-fee code 0334-2700/88

https://doi.org/10.1017/S0334270000006184 Published online by Cambridge University Press

Chu in [3] has applied the homotopy method to the ordinary symmetric eigenvalue problems $Ax = \lambda x$. Recently, Li and Rhee [4] have given an elegant fast algorithm for this problem. Moreover, a paper by Li and Sauer [5] has proposed some important theoretical results by using the homotopy method for generalised eigenvalue problems. In this paper, we seek to generalise the approach of Chu [3] to (1.1), and to develop some numerical strategies. If both A and B are tridiagonal one has to reduce the generalised symmetric eigenvalue problem to the form $L^{-1}AL^{-\mathsf{T}}x = \lambda x$, where $B = LL^{\mathsf{T}}$ is the Cholesky decomposition. In general, we have fill-in in the matrix $L^{-1}AL^{-\mathsf{T}}$. We propose that applying the homotopy method to (1.1) together with sparse matrix techniques might be competitive to the QR-method [8] applied to $L^{-1}AL^{-\mathsf{T}}$.

In the next section we shall construct a special homotopy equation and show that this homotopy will work for almost all symmetric tridiagonal matrices A and B. In the third and fourth sections, we refer to the applicability of the method and present some numerical results.

2. Main theorems

Let A, B be two symmetric tridiagonal matrices and D be an arbitrary diagonal matrix with n distinct elements. Consider the mapping $H\colon \mathbf{R}^n \times \mathbf{R} \times [0,1] \to \mathbf{R}^n \times \mathbf{R}$ defined by

$$H(x,\lambda,t) = [((1-t)(D-\lambda I) + t(A-\lambda B))x, (1-x^{\mathsf{T}}x)/2], \qquad x \in \mathbf{R}^n, \quad \lambda \in \mathbf{R}, \quad t \in [0,1]. \tag{2.1}$$

Denote by e_i the ith unit vector in $\mathbf{R}^n$. Then we have the following homotopy with $z_0 = (e_i, d_i)$ and $F(z) = [Ax - \lambda Bx, (1-x^{\mathsf{T}}x)/2]$ for $z^{\mathsf{T}} = (x^{\mathsf{T}}, \lambda) \in \mathbf{R}^n \times \mathbf{R}$:

$$\begin{cases} H(z_0, 0) = 0 \\ H(z,1) = F(z). \end{cases} \tag{2.2}$$

If t runs from 0 to 1, then the trivial matrix pencil $D - \lambda I$ passes over to the matrix pencil $A - \lambda B$. Let

$$C = (1-t)(D-\lambda I) + t(A-\lambda B) \tag{2.3}$$

and Γ denote the graph of the homotopy equation, i.e.

$$\Gamma := \{(x,\lambda,t) \in \mathbf{R}^n \times \mathbf{R} \times [0,1] | H(x,\lambda,t) = 0\}. \tag{2.4}$$

Denote by $\delta_1 H$ the Jacobian matrix of H referring to the first variables (x,λ). We claim

THEOREM 1. *If* $\operatorname{rank}(C) = n-1$, *then* $\delta_1 H$ *is nonsingular for all* $(x,\lambda,t) \in \Gamma$.

PROOF. Observe that $\delta_1 H = \left[\begin{smallmatrix} C & -\widetilde{B}x \\ -x^{\mathsf{T}} & 0 \end{smallmatrix}\right]$ for $(x, \lambda, t) \in \Gamma$, where $Cx = 0$ and $\widetilde{B} := tB + (1-t)I$ is nonsingular for $t \in [0,1]$. Assume that $\delta_1 H \left[\begin{smallmatrix} y \\ a \end{smallmatrix}\right] = 0$ for $y \in \mathbf{R}^n$ and $a \in \mathbf{R}$, then $Cx = a\widetilde{B}x$ and $x^{\mathsf{T}} y = 0$. From $0 = x^{\mathsf{T}} C y = a x^{\mathsf{T}} \widetilde{B} x$ it follows that $a = 0$ (for $x^{\mathsf{T}} \widetilde{B} x > 0$), therefore $Cy = 0$. We then have y is contained in the nullspace of C. If $\operatorname{rank}(C) = n-1$, the nullspace of C is spanned by x. Because $x^{\mathsf{T}} y = 0$, the vector y must be the null vector.

REMARK. If $\lambda \neq a_{i,i+1}/b_{i,i+1}$ in (2.3) for all $i = 1, \cdots, n-1$, where $a_{i,i+1}$ and $b_{i,i+1}$ denote the subdiagonal elements of A and B respectively, then $\operatorname{rank}(C) = n-1$. $\operatorname{Rank}(C) < n-1$ may occur only when $\lambda = a_{i,i+1}/b_{i,i+1}$ for some i.

THEOREM 2. *If* $\operatorname{rank}(C) = n-1$ *for all* $(x, \lambda, t) \in \Gamma$, *then the set* Γ *can be characterised by a differential equation*

$$[dx/dt\, d\lambda/dt] = -\delta_1 H(x, \lambda, t)^{-1} \delta_2 H(x, \lambda, t), \tag{2.5}$$

where

$$\delta_2 H(x, \lambda, t) = \begin{bmatrix} (D - A - \lambda I \times \lambda B)x \\ 0 \end{bmatrix}$$

denotes the derivative of H *with respect to* t.

PROOF. Along each component, we take the derivative with respect to the arclength s. The set Γ is then characterised by

$$\delta_1 H [dx/ds\, d\lambda/ds] + \delta_2 H \frac{dt}{ds} = 0. \tag{2.6}$$

Since $dt/ds \neq 0$, otherwise $\delta_1 H$ would be singular, Γ can be parametrised by the variable t and (2.6) becomes

$$[dx/dt\, d\lambda/dt] = -\delta_1 H(x, \lambda, t)^{-1} \delta_2 H(x, \lambda, t).$$

It is important to know whether the curve Γ will diverge to infinity or not. For this we claim

THEOREM 3. *If* $\operatorname{rank}(C) = n-1$ *for all* $(x, \lambda, t) \in \Gamma$, *then the set* Γ *is bounded, and the trivial solution of* $H(x, \lambda, 0) = 0$ *is connected with the desired solution* $H(x, \lambda, 1) = 0$ *by a smooth curve.*

PROOF. For any $(x, \lambda, t) \in \Gamma$, it is clear that $\|x(t)\| \leq 1$. The equation $(1-t)(D - \lambda I)x + t(A - \lambda B)x = 0$ can be written as

$$\lambda(t) x = [(1-t)I + tB]^{-1} [(1-t)D + tA] x.$$

Applying a lub-norm to the above equation leads to

$$|\lambda(t)| \leq \|\widetilde{B}^{-1}\| \, (\|D\| + \|A\|),$$

where $\widetilde{B} = (1-t)I+tB$ is positive definite. Since $\|\widetilde{B}^{-1}\|$ is bounded on the compact set $[0,1]$, so is $|\lambda(t)|$. From the rank condition on $\delta_1 H$ and the boundedness of λ it follows that Γ has no turning points and this yields the results.

If $\operatorname{rank}(C) = n-1$ for $t \in [0,1]$, the differential equation (2.5) satisfies a Lipschitz condition. Therefore existence and uniqueness of n distinct solutions are guaranteed if the starting vectors are linearly independent, and so, under the condition $\operatorname{rank}(C) = n-1$ for $t \in [0,1]$, all n eigenpairs are found.

3. Applicability

Choosing starting vectors $\begin{bmatrix} x(0) \\ \lambda(0) \end{bmatrix} = \begin{bmatrix} e_i \\ d_i \end{bmatrix}$, where e_i denotes the ith unit vector and d_i the ith diagonal entry of D, we can solve (2.5) by using an adequate ODE-solver, e.g. a predictor-corrector scheme consisting of Adams-Bashforth and Adams-Moulton formulae from Shampine and Gordon [7] can be used. When applying this solver, two function evaluations of the right side of (2.5) are necessary at each step, involving solving two linear bordered systems of the form

$$\begin{bmatrix} C & b \\ X^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} y \\ \gamma \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix} \quad \text{with } x^{\mathsf{T}}C = 0. \tag{3.1}$$

where b, d and $\gamma \in \mathbf{R}^n$, $y \in \mathbf{R}$ and C is a symmetric tridiagonal matrix. The complexity of performing the homotopy method depends strongly on how inexpensively and how stably these linear systems can be solved. As C is (nearly) singular, the obvious block elimination may produce large errors, also. Gaussian elimination with partial pivoting will destroy the sparse structure of the system. Therefore we use a slightly modified technique: row interchanges of a k by k band matrix are allowed only in the k preceding rows, producing a fill-in of k supplemental upper off-diagonals. Applying this technique, we have found some instability in the case of large systems ($n \geq 100$), but if $n < 100$ then this strategy works quite well in practice. Nevertheless, we think that a much better way to solve (3.1) is the so-called stabilised block-elimination developed by Chan [1], or a method proposed by Osborne [6], which works as follows.

We assume that we have a sparse factorisation $C = LU$ with L lower triangular and

$$U = \begin{array}{c} \\ \\ \end{array} \begin{array}{cc} n-1 & 1 \\ \left(\begin{matrix} U_{n-1} & u \\ 0 & u_{nn} \end{matrix}\right) & \begin{matrix} \}n-1 \\ \}1 \end{matrix} \end{array},$$

where $u_{nn} = 0$ or $u_{nn} = \mathrm{O}(\sigma_n)$ (σ_n denotes the smallest singular value of C). According to Chan [2], conventional pivoting strategy will usually be sufficient to produce such factorisation unless C^{-1} has a very skew distribution of the sizes of its elements. Otherwise, Chan's two-pass algorithm can be applied to force the element u_{nn} in U to be small. (Note that this last algorithm involves row and column permutation, which may destroy the sparse structure). If $u_{nn} = 0$, then the last row of L^{-1} is in the null space of C; this implies $x^{\mathsf{T}} = e_n^{\mathsf{T}} L^{-1}$ because $\operatorname{rank}(C) = n-1$ and $x^{\mathsf{T}} C = 0$. Since $x^{\mathsf{T}} b = x^{\mathsf{T}}(tB + (1-t)I)x > 0$, we can calculate γ in (3.1) by

$$\gamma = x^{\mathsf{T}} r / x^{\mathsf{T}} b = e_n^{\mathsf{T}} L^{-1} r / e_n^{\mathsf{T}} L^{-1} b \tag{3.2}$$

and then

$$y = \begin{bmatrix} U_{n-1}^{-1}\{-\gamma[L^{-1}b]_{n-1} + [L^{-1}r]_{n-1}\} \\ 0 \end{bmatrix} + \delta(U + e_n e_n^{\mathsf{T}})^{-1} e_n, \tag{3.3}$$

where δ is found from $x^{\mathsf{T}} y = 0$.

A start on the error analysis might be provided by Osborne's paper [6] which considers the calculation of the right and left singular vectors and demonstrates the importance (and often the difficulty) in making $u_{nn} = 0$. At present it is not clear what method is best suited for problem (3.1). A compromise between preserving the sparse structure and achieving high stability needs to be found.

As we are only interested in $x(1)$ and $\lambda(1)$, it is not worth following the curves too closely. Most of the well-known ODE-solvers can manage the curve tracing problem if there are no singular points (that is, no t where $\operatorname{rank}(C) < n-1$). In the case of breakdown, nothing can be done except to jump over the singular point or to start a new homotopy with another diagonal D. Since singular points occur infrequently for randomly chosen A and B, we did not concentrate on these pathological cases. Instead, we tried to manage the problem in the case $\operatorname{rank}(C) = n-1$ for $t \in [0,1]$ by applying the following strategy.

(1) Choose an error bound *eps* for the ODE-solver, a bound *eps*1 to restrict the residual $r = \|H(x(t), \lambda(t), t)\|_2$ at each step and a positive number k representing the maximal number of iterations to refine $x(t)$ and $\lambda(t)$ by inverse iteration.

(2) Correct $x(t)$ and $\lambda(t)$ by inverse iteration, if $r \geq eps1$. For $t = 1$, correct the eigenpair as much as machine precision allows.

(3) Combine step-size control and order selection in the solver with the magnitude of r, which can be regarded as a measure of curve tracing quality.

It turns out that the overall efficiency of the homotopy method strongly depends on two decisions:

(a) choice of the error bounds *eps*1 and *eps*;

(b) choice of D, which corresponds to the initial conditions of the initial value problem (IVP).

In order to find all eigenpairs, we should prevent two curves "running together", else one eigenpair would be provided twice at $t = 1$. Choosing *eps* and *eps*1 too large results in the curves not being followed closely enough. In this case, an approximation $(x(t), \lambda(t))$ could be located between two neighbouring curves, that is, in the domain of attraction of another curve.

The importance of choosing a suitable diagonal D can be demonstrated by the following simple 3 by 3 example with $B = I$. Let

$$A = \begin{bmatrix} -1 & 1 & 0 \\ 1 & 198 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

with eigenvalues 198.0101, 0.99494 and −1.00504 and

$$M(t) \equiv \delta_1 H(x, \lambda, t) = \begin{bmatrix} D + t(A - D) - \lambda I & -x \\ -x^{\mathsf{T}} & 0 \end{bmatrix},$$

with the condition number

$$\kappa(M(t)) = |\lambda_{\max}(M(t))/\lambda_{\min}(M(t))|.$$

If $\kappa(M(t))$ becomes large, then $||(\dot{x}(t), \dot{\lambda}(t))^{\mathsf{T}}||$ will also be large, forcing the ODE-solver to take short step sizes. Consider the situation at $t = 1/2$ with $D = \operatorname{diag}(3, 2, 1)$:

$$\kappa(M(1/2)) \approx 20000 \quad \text{as } D + 0.5(A - D) \quad \text{becomes} \quad \begin{bmatrix} 1 & 0.5 & 0 \\ 0.5 & 100 & -0.5 \\ 0 & -0.5 & 1 \end{bmatrix}$$

with eigenvalues 99.01, −0.00505 and 0. The condition number has become much worse in comparison to $\kappa(M(1)) = \kappa(A) \approx 200$. This radical deterioration of the condition may have the undesired effect that many steps are necessary to prevent two curves "running together".

With the choice of $D = \operatorname{diag}(A)$, the homotopy method is able to calculate the three eigenpairs in not more than four steps from 0 to 1 with error bound $eps = 10^{-1}$ and $eps1 = 10^{-2}$. So it has to be emphasised that the choice of a suitable diagonal D plays an important role in making the homotopy method competitive. D should be chosen so that $\kappa(M(t)) \approx \kappa(A)$ for $t \in [0, 1]$. We have to concede that, so far, we do not know how to choose a suitable D for large n.

4. Numerical experience

In this section, some examples are given to illustrate the numerical behaviour, by performing the homotopy method and investigating the trace of the curves. All computations were done in FORTRAN 77 on a BASF 7/70 computer ($2^{-t} \approx 10^{-12}$) in single precision arithmetic.

https://doi.org/10.1017/S0334270000006184 Published online by Cambridge University Press

4.1. Improving the estimating accuracy of the homotopy method

According to our experience, it is not sufficient to pursue the curves using ODE-solvers without inverse iterative improvement in each step. In fact, the ε-tube of the solution curve in this case would be too wide and the results could stray outside the convergence range of Newton's method. Hence we have no chance to correct the computed eigenpairs at $t = 1$ using inverse iteration.

The following table gives a comparison between the homotopy method and the QR-method [8]. We perform one inverse iteration in each step in order to follow the exact trajectory sufficiently closely, and at $t = 1$ force the residual $\|Ax - \lambda Bx\|$ to be small (of the order of the machine precision using inverse iteration, if high accuracy is required).

EXAMPLE. A, B are two tridiagonal matrices of dimension 8 (see Table 1).

TABLE 1

diag. of A	subdiag. of A	diag. of B	subdiag. of B
− 0.82446	− 1.12084	1.06416	0.050622
1.80331	− 1.41511	1.04762	0.577722
0.61557	1.23429	1.53037	0.644825
− 0.27344	− 0.07309	1.89360	0.509867
− 1.60200	− 0.85724	1.88292	0.679965
− 0.55078	0.74460	1.77055	0.040127
1.10018	− 1.67464	1.01956	0.127609
− 1.99046		1.40009	

For the purpose of comparison between the QR and homotopy methods, we do not consider reiterations here at the end step ($t = 1$). We choose the initial matrix $D = \operatorname{diag}(A)$ for the homotopy method.

In Table 2, we show eigenvalues with corresponding eigenvectors calculated by an error bound of 10^{-1}.

4.2. Multiple arrival of the same eigenpair

For the pursuit of the n space curves using an ODE-solver with variable step size, we have to choose an error bound *eps* in order to limit the local error in each step. When two curves come near at a point $t \in [0, 1]$, a jump to a neighbouring curve may occur if the ε-tube around the exact curve becomes too wide because of too large an error. This "running together" of two curves would not often occur for matrices of small dimension, if *eps* is chosen small enough (e.g. ≤ 0.01). The following 40×40 example will show the cause of a multiple arrival (see Table 3).

TABLE 2. QR-method

$\underline{4.6447}1364$	0.190373	$-\,\underline{0.7909}6387$	− 0.402404
	− 0.809692		− 0.006422
	0.542059		0.436272
	− 0.114076		− 0.460238
	0.032809		− 0.598198
	− 0.015193		− 0.264742
	− 0.002799		0.043626
	0.000747		− 0.077747
$\underline{1.98359}439$	0.000153	$-\,\underline{1.06747}536$	0.786015
	− 0.000368		0.229513
	− 0.000034		− 0.210384
	0.022878		0.341430
	− 0.085006		− 0.408140
	0.194407		− 0.042896
	0.905725		0.004852
	0.095725		0.004852
	− 0.366222		− 0.015054
$\underline{0.37757}980$	− 0.356161	$-\,\underline{1.34079}937$	0.424441
	0.383129		0.242807
	0.578821		0.518333
	0.609497		− 0.584663
	− 0.108956		0.387589
	0.080899		− 0.011760
	− 0.031166		0.000419
	0.021315		− 0.005558
$-\,\underline{0.18152}507$	− 0.004109	$-\,\underline{1.9363}9927$	0.001267
	0.002334		0.001531
	0.007037		0.015420
	− 0.002389		− 0.022044
	− 0.481013		0.039943
	0.824438		− 0.133845
	− 0.217044		0.446081
	0.206440		0.883614

TABLE 2 (continued). Homotopy method

$\underline{4.6447}0040$	− 0.190373	$-\underline{0.7909}7263$	0.402456
	0.809692		0.006437
	− 0.542060		− 0.436293
	0.114076		0.460269
	− 0.032809		0.598180
	0.015193		0.264731
	0.002799		− 0.043625
	− 0.000747		0.077743
$\underline{1.98359}770$	0.000153	$-\underline{1.06747}478$	0.786015
	− 0.000368		0.229513
	− 0.000033		− 0.210385
	0.022878		0.341431
	− 0.085008		− 0.408137
	0.194406		− 0.042896
	0.905723		0.004852
	− 0.366223		− 0.015054
$\underline{0.37757}799$	− 0.356158	$-\underline{1.34079}829$	0.424448
	0.383134		0.242809
	0.578831		0.518336
	0.609492		− 0.584666
	− 0.108955		0.387590
	0.080899		− 0.011760
	− 0.031166		0.000418
	0.021315		− 0.005558
$-\underline{0.18152}968$	0.004114	$-\underline{1.9363}8674$	0.001266
	− 0.002336		0.001531
	− 0.007046		0.015404
	0.002399		− 0.022025
	0.480115		0.039935
	− 0.824432		− 0.133830
	0.217040		0.446050
	− 0.206448		0.883625

TABLE 3

diag. of A		subdiag. of A	
1.03790	− 0.94870	− 1.57300	− 0.79227
− 0.39143	− 0.50191	− 0.72290	0.71835
0.61643	− 0.65524	0.81169	0.51495
− 1.95590	1.40716	− 1.36820	1.99538
1.26920	1.06806	− 1.91596	− 0.92466
0.34467	− 1.91614	− 1.22120	− 1.75060
− 1.65383	− 1.47243	− 0.45275	− 0.78361
− 1.85014	− 1.50767	− 1.72410	− 0.40233
− 1.20483	0.30503	1.98133	− 0.63900
1.15569	− 1.70789	− 0.40882	0.47224
0.05080	0.12880	1.06304	− 0.66552
− 1.81171	− 0.40209	0.78293	1.23641
0.26809	0.91812	0.13995	1.87632
− 1.29456	− 0.00381	− 0.79281	− 1.20339
1.15722	− 1.72923	− 1.77261	− 0.39009
− 1.18261	0.72568	1.55665	1.37069
1.33834	− 0.48764	− 1.54071	1.28342
− 0.65834	− 1.73680	− 1.94258	0.11292
1.82102	0.87320	0.39931	1.73372
1.01112	− 0.10050	1.50296	

diag. of B		subdiag. of B	
1.42369	1.56722	0.890872	0.866422
1.82736	1.38461	0.786782	0.180812
1.80066	1.01601	0.893630	0.820654
1.63709	1.67491	0.622052	0.161941
1.52247	1.40288	0.993529	0.730857
1.63151	1.80216	0.509042	0.356992
1.52916	1.66128	0.762349	0.480960
1.06769	1.53259	0.384084	0.736915
1.25593	1.51033	0.449564	0.069231
1.53228	1.43571	0.640229	0.970906
1.98110	1.63248	0.575375	0.111342
1.17838	1.55986	0.590335	0.237507
1.15058	1.22127	0.761321	0.961169
1.34477	1.45483	0.162205	0.457632
1.88009	1.48999	0.901615	0.225999
1.97487	1.58789	0.903999	0.325589
1.79800	1.23272	0.943198	0.187618
1.76634	1.13767	0.870167	0.921842
1.33776	1.56517	0.473603	0.489429
1.59123	1.79588	0.385359	

The following pairs mark the curves of index i with the initial vector e_i, which run together for $eps = 0.1$: (7, 14), (15, 20), (16, 23), (28, 32) and (27, 40).

Pursue all curves with $eps = 0.1$ and correct the eigenpairs using inverse iteration at the end (see Table 4).

TABLE 4

Eigenvalues		ith initial vector e_i		Required steps from 0 to 1	
15.04731	− 0.89598	5	9	17	9
14.33334	− 1.08116	19	37	18	13
4.38383	$-\underline{1.30467}$	1	23	11	23
2.67272	$-\underline{1.30467}$	24	16	11	18
2.20794	$-\underline{1.34743}$	17	28	14	20
1.79434	$-\underline{1.34743}$	10	32	8	16
1.40901	$-\underline{1.59300}$	33	14	10	14
1.23301	$-\underline{1.59300}$	39	7	10	12
$\underline{1.21427}$	− 1.66087	15	21	15	19
$\underline{1.21427}$	− 2.33812	20	4	10	8
1.07317	− 2.71028	36	35	9	19
0.66271	− 3.03039	29	26	13	19
0.44396	− 3.18289	3	8	8	13
0.41900	− 5.14978	13	30	7	16
0.40880	− 5.65133	31	38	10	11
0.27985	− 8.73588	25	12	15	16
0.15848		22		14	
0.12585		11		7	
− 0.12008		6		9	
− 0.14825		34		15	
$-\underline{0.39705}$		40		14	
$-\underline{0.39705}$		27		16	
− 0.56775		2		8	
− 0.72685		18		8	

Multiple attainable eigenvalues are underlined.

Pursue all curves with $eps = 0.01$ and use inverse iteration at the end (see Table 5).

From Figures 1 and 2 it is clear that the derivatives (dx/dt, $d\lambda/dt$) have large norm, exactly at the point at which the curves run together. In fact, (2.5) provides $d\lambda/dt = x^{\mathsf{T}}(A - D)x$, i.e. the derivative $d\lambda/dt$ is almost constant over a broad range in the interval $[0, 1]$, so that most eigenvalue curves, chosen by random examples, run piecewise linearly. Numerical difficulty occurs only in the computation of eigenvectors, since their components are subjected to large fluctuation (see next example).

In the 20×20 example with $B = I$, large derivatives occur following curves 6, 7, 11 and 15, which force the solver to small step sizes (see Table 6).

TABLE 5

Eigenvalues		ith initial vector e_i		Required steps from 0 to 1	
15.04731	− 0.39705	19	40	46	38
14.33334	− 0.48492	5	27	40	33
4.38383	− 0.56775	1	2	19	12
2.67272	− 0.72685	24	18	18	21
2.20794	− 0.89598	17	9	26	18
1.79434	− 1.02530	10	32	14	47
1.40901	− 1.08116	33	28	16	48
1.23301	− 1.08116	39	37	18	69
1.21427	− 1.54785	15	23	29	60
1.07317	− 1.59300	36	16	14	55
0.94140	− 1.66087	20	21	21	60
0.66271	− 1.66087	25	14	30	47
0.44396	− 1.71097	3	7	14	30
0.41900	− 2.33812	13	4	11	14
0.40880	− 2.71028	29	35	37	40
0.27985	− 3.03039	31	26	30	40
0.15848	− 3.18289	22	12	23	26
0.12585	− 5.14978	11	30	15	27
− 0.12008	− 5.65133	6	38	19	22
− 0.14825	− 8.73588	34	8	19	33

TABLE 6

diag. of A		subdiag. of A	
1.76527	1.78285	4.46870	− 4.64719
− 3.34318	− 2.95864	− 0.13243	− 1.38682
4.16146	0.39343	1.95458	− 2.88848
4.92607	3.73302	4.65391	0.84243
3.59465	− 0.72845	2.46035	− 3.02567
3.54024	− 2.83381	− 4.74122	1.29526
2.62480	1.34617	4.62402	− 3.13650
3.90966	− 3.85757	2.37385	3.65834
− 1.95242	− 1.79869	− 3.76831	2.03203
− 0.69723	− 0.34191	4.22387	

Eigenvalues		ith initial vector e_i		Required steps from 0 to 1	
10.63040	0.80643	4	13	25	98
9.10130	− 0.86434	3	20	52	73
7.00736	− 1.26542	8	10	90	78
5.52085	− 1.84721	14	15	61	175
4.78417	− 3.59012	5	19	84	75
4.35742	− 5.07960	6	9	115	31
3.78543	− 5.15356	7	16	184	25
3.58174	− 5.93749	17	2	80	26
1.63356	− 7.58358	1	12	93	38
0.96612	− 7.58775	11	18	141	27

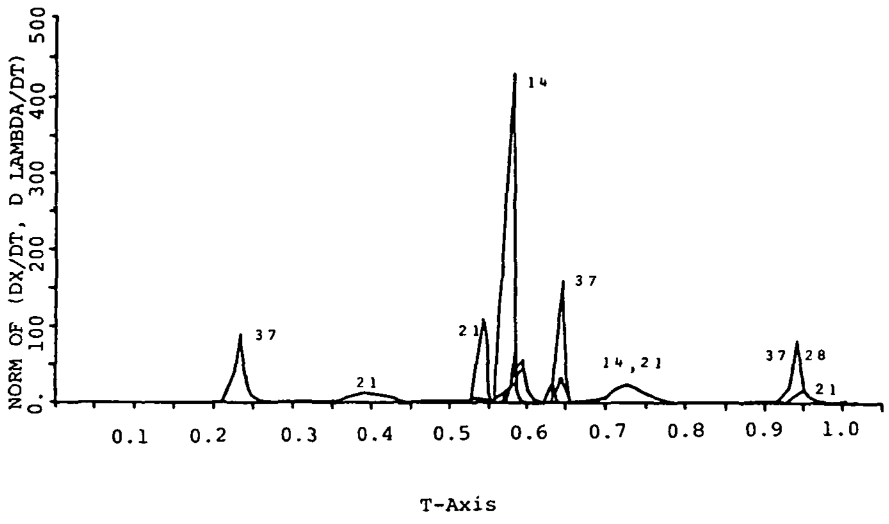

FIGURE 1. Norm of derivatives of x and λ of 40×40 example with $eps = 0.01$ for all curves.

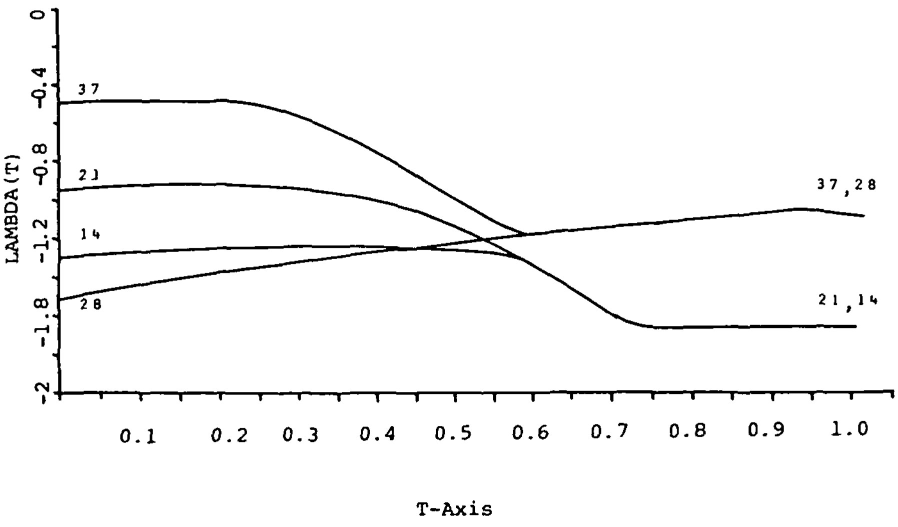

FIGURE 2. Eigenvalue curves 14, 20, 28 and 37 of 40×40 example. Because of large derivatives, the curves 14, 21 and 28, 37 run together.

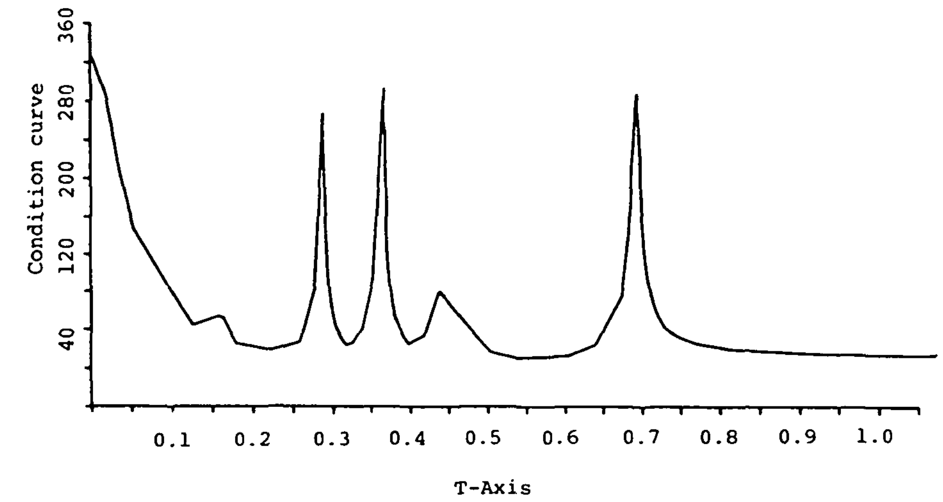

FIGURE 3. Condition number of matrices of linear system in curve 11 for the 20×20 example.

Figures 3 and 4 show the relation between the condition number of appearing matrices corresponding to linear equations and the nearest distance of two eigenvalues of $D+t(A-D)$ for $t \in [0,1]$. Figure 5 illustrates the curves of components of the demonstrative eigenvector $x(t)$ (see (2.1)). We see that abrupt alternations occur on the position of large condition in the individual components.

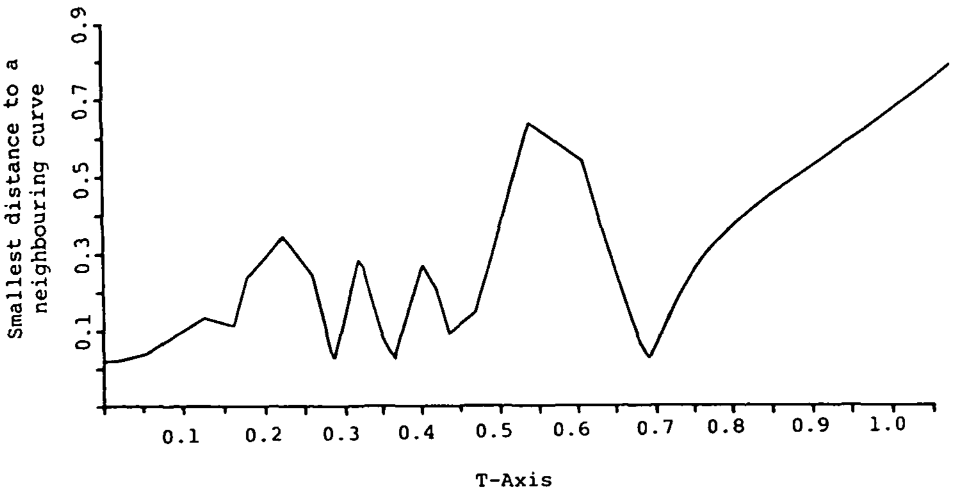

FIGURE 4. The smallest distance of eigenvalue curve 11 of 20×20 example to the other curves.

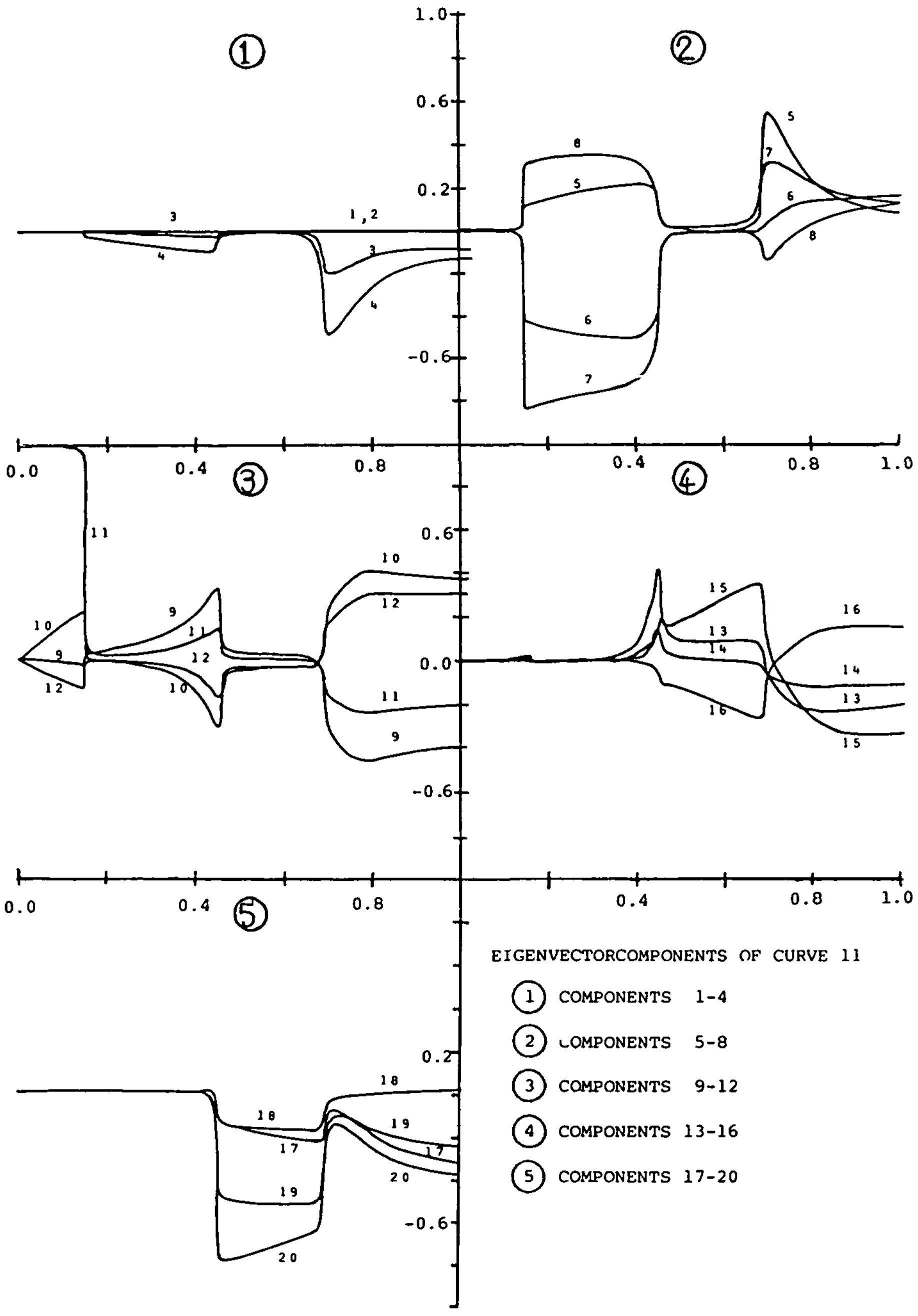
1.0
0.6
0.2
-0.6
0.0
0.4
0.8
1.0
①
②
③
④
⑤
EIGENVECTORCOMPONENTS OF CURVE 11
① COMPONENTS 1-4
② COMPONENTS 5-8
③ COMPONENTS 9-12
④ COMPONENTS 13-16
⑤ COMPONENTS 17-20

FIGURE 5

4.3. Possible strategy for the prevention of the multiple arrival of the same eigenpair

Denote by (x_l, λ_l) an eigenpair, which was computed by the homotopy method with $0 \leq t \leq 1$. The starting vector was the l-th entry on the diagonal D with the l-th unit vector. Let

$$\text{Step}(l) := \text{Required number of steps from 0 to 1 for the computation of } (x_l(1), \lambda_l(1)).$$

$$Doub := \text{The set of the indices of the curves which run together.}$$

$$d\colon Doub \to Doub,\ d(l) = k,\ \text{where } l \neq k \text{ with}$$

$$(x_k(1), \lambda_k(1)) = (x_l(1), \lambda_l(1)).$$

We propose a strategy consisting of two parts which can be combined.

In the predictor step, a jump to another curve can be detected by observation of rotations in the eigenvectors $x_l(t)$, $x_l(t+h)$ of two consecutive steps. Let s be a bound given by the user, with $0 < s < 1$. Then $x_l(t+h)^{\mathsf{T}} x_l(t) \leq s$ indicates that a jump on a different curve has occurred. The predictor step must be repeated with a smaller step size (e.g. $h/2$).

In the case of matrices with clustered eigenvalues, it is often not sufficient to observe rotations of eigenvectors (the bound s cannot be chosen optimally). Curves which have run together should be pursued once again with a smaller error tolerance. At first, we try to pursue all curves with a large error bound in order to minimize the number of steps. If multiple arrival of the same eigenpair occurs ($l, k \in Doub$), we reduce the error bound of the curve with index l satisfying step$(l) \leq$ step(k).

The strategy will be repeated until no double pairs occur (see Figure 6).

Consider the 20×20 example given in Table 7.

TABLE 7

diag. of A		subdiag. of A	
– 0.92090	– 0.24239	0.58790	– 1.26938
1.34215	0.91212	– 1.02382	– 0.44064
– 1.88659	– 1.42630	– 0.50036	1.68493
0.39088	– 1.92657	– 1.77122	0.64265
1.00548	– 0.99159	1.94549	– 0.55770
1.61127	– 0.02780	1.16990	0.76313
– 0.05334	– 0.08227	0.61511	– .35795
0.02240	1.75703	0.52738	1.57235
0.92464	0.26430	0.88681	– 0.15158
0.74512	1.38202	1.49588	

TABLE 7 (continued)

diag. of B		subdiag. of B	
1.74959	1.86182	0.482148	0.066360
1.77554	1.15285	0.453783	0.078613
1.70061	1.29831	0.872928	0.650783
1.39198	1.07308	0.945517	0.265925
1.38120	0.98102	0.640771	0.657949
1.74952	1.30884	0.420097	0.254484
1.95976	1.94632	0.137067	0.837674
1.28791	1.05313	0.926086	0.937862
1.12338	1.95318	0.346523	0.155763
1.35598	1.68701	0.879626	

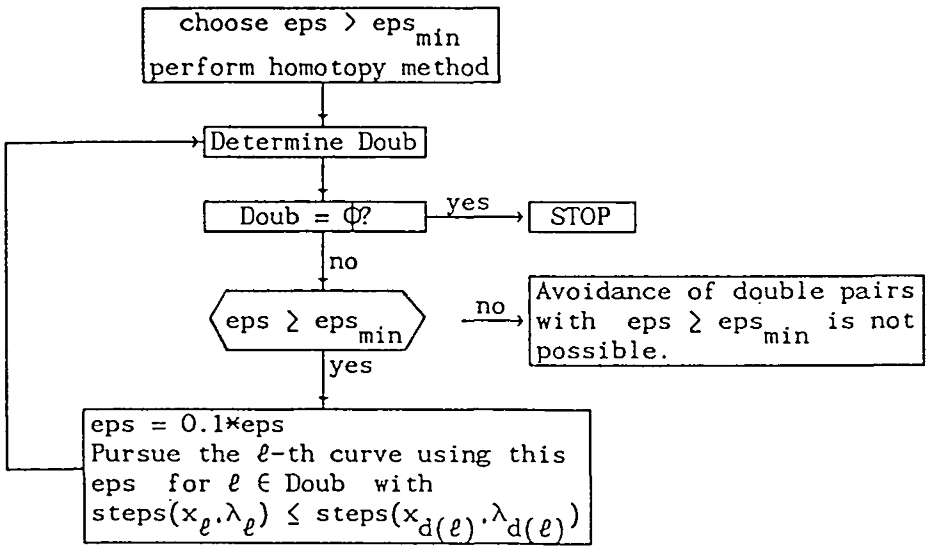
choose eps > eps_min
perform homotopy method
Determine Doub
Doub = Φ?
yes
STOP
no
eps ≥ eps_min
no
Avoidance of double pairs with eps ≥ eps_min is not possible.
yes
eps = 0.1*eps
Pursue the ℓ-th curve using this eps for ℓ ∈ Doub with steps(x_ℓ,λ_ℓ) ≤ steps(x_d(ℓ),λ_d(ℓ))

FIGURE 6

TABLE 8

Eigenvalues		i-th initial vector e_i		Required steps from 0 to 1	
28.46983	− 0.01865	6	16	33	44
2.21768	− 0.40977	18	8	12	48
$\underline{1.44173}$	− 0.60235	$\underline{9}$	17	$\underline{39}$	47
$\underline{1.44173}$	− 0.67347	$\underline{2}$	7	$\underline{37}$	61
1.18556	− 0.75602	5	1	50	37
0.91963	− 1.24384	12	15	29	59
0.84275	− $\underline{1.50296}$	20	$\underline{13}$	9	$\underline{60}$
0.56178	− $\underline{1.50296}$	19	$\underline{3}$	35	$\underline{27}$
0.33365	− 2.62570	10	11	21	80
0.09003	− 7.09735	47	14	31	19

For *eps* = 0.01 the curves 3 and 13, 2 and 9 run together (see Table 8). Since 3 and 2 are pursued with a fewer number of steps, their error bound will be reduced to 0.001; we then have 9 and 2 run together (see Table 9).

TABLE 9

Eigenvalues	i-th initial vector e_i	Required steps from 0 to 1
$\underline{1.44173}$	$\underline{9}$	39
$\underline{1.44173}$	$\underline{2}$	69
1.18556	5	50
−1.50296	13	60
−4.28759	3	63

We now reduce the error bound for curve 9 to 0.001; then the curves 9 and 5 run together (see Table 10).

TABLE 10

Eigenvalues	i-th initial vector e_i	Required steps from 0 to 1
1.44173	2	69
$\underline{1.18556}$	$\underline{9}$	83
$\underline{1.18556}$	$\underline{5}$	50
−1.50296	13	60
−4.28759	3	63

Finally, we reduce *eps* for curve 5 to 0.001 (see Table 11).

TABLE 11

Eigenvalues	i-th initial vector e_i	Required steps from 0 to 1
1.44173	2	69
1.24247	5	101
1.18556	9	83
−1.50296	13	60
−4.28759	3	63

5. Conclusion

The above examples clearly show that some numerical difficulties exist when using the curve-pursuit method to solve the eigenvalue problem. In some cases during curve pursuit, the condition of the matrices of linear systems, whose solutions are necessary for the homotopy method, becomes suddenly worse. Therefore, an appropriate choice of the initial diagonal D plays an important role for the homotopy method, especially for large dimensions of A and B. Numerical experience tell us that the choice $D = \text{diag}(A)$ is favourable, although this choice cannot completely avoid the occurrence of large condition numbers. It is remarkable that an ODE-solver with step size control and variable order can be applied to recover the lost eigenpair (because of multiple arrival) using a reduced error bound.

References

[1] T. F. Chan, "Deflated decomposition of solutions of nearly singular systems", *SIAM J. Numer. Anal.* **21** (1984) 738–754.
[2] T. F. Chan, "On the existence and computation of LU-factorization with small pivots", *Math. Comp.* **42** (1984) 535–547.
[3] M. T. Chu, "A simple application of the homotopy method to symmetric eigenvalue problem", *Linear Algebra Appl.* **59** (1984) 85–90.
[4] T. Y. Li and N. Rhee, "Homotopy algorithm for symmetric eigenvalue problems", (to appear).
[5] T. Y. Li and T. Sauer, "Homotopy method for generalized eigenvalue problems", *Linear Algebra Appl.* **91** (1987) 65–74.
[6] M. R. Osborne, "Numerical methods for hydrodynamic stability problems", *SIAM J. Appl. Math.* **15** (1967) 539–557.
[7] L. F. Shampine and M. K. Gordon, *Computer solution of ordinary differential equations* (W. H. Freeman and Company, San Francisco 1975).
[8] J. H. Wilkinson and C. Reinch, *Linear Algebra* (Springer-Verlag, New York, Heidelberg, Berlin 1971).