DECUS
PROGRAM LIBRARY

DECUS NO.	8-80
TITLE	DETERMINATION OF REAL EIGENVALUES OF A REAL MATRIX
AUTHOR	A. E. Sapega
COMPANY	Trinity College Hartford, Connecticut
DATE	June 16, 1967
FORMAT	

ABSTRACT

This is a two-part program for determining the real eigenvalues of a real-valued matrix. The matrix does not have to be symmetric. Part I uses the power method of iterating on an eigenvector to determine the largest eigenvalue of the matrix. Part II then deflates the matrix using the results of Part I so as to produce a matrix of order one less than that solved for in Part I. Part I can then be reloaded, and the next eigenvalue in line may be calculated. In this all the real eigenvalues may be computed in order.

MINIMUM HARDWARE

PDP-8, ASR-33 (This is programmed for transfer between high-speed reader and the ASR-33, but the high-speed reader is not necessary to use the programs.)

OTHER PROGRAMS NEEDED

FORTRAN Systems

RESTRICTIONS

This will compute on an original matrix of up to size 7.

NOTES ON EIGENVALUES SOLUTIONS

The power method iterates on an eigenvector to calculate a new vector which is an improved value of the eigenvector. This process will converge providing the eigenvalue of the eigenvector is real. This particular program will calculate real eigenvalues which are very close to each other on the real axis and will also calculate repeated eigenvalues. A counter is used to terminate the iterations after 1000 have been performed. This usually indicates that the eigenvalues are complex, as this has been a sufficient number of iterations to be able to separate two eigenvalues which differ in the third place. (See for example the results of Leverrier's matrix as given in Faddeeva, COMPUTATIONAL METHODS OF LINEAR ALGEBRA.) This program starts arbitrarily with the vector (1 0 0 . . . . 0) as the first guess. By no-oping loc 0404 in the FORTRAN OTS, convergence will occur within 30 seconds (approx.), at which point the counter will terminate the calculation.

Part II of the program deflates the matrix based on the results of Part I. Dimension statements are identical in both parts so that the data remaining from the previous calculation is in the proper location in core storage to be used by the next program. Thus, only the first pass requires data input: each succeeding pass uses the data left by the previous pass.

The deflation procedure used is that given by Wilkinson, THE ALGEBRAIC EIGENVALUE PROBLEM, pages 587-594.

When the deflation procedure is applied for the n-th time, the resulting matrix is of size 1 and is the smallest eigenvalue of the original matrix.

If only one pair of complex eigenvalues exists, it will be possible to work to the matrix of size 2, at which point the iterations to an eigenvalue will not converge. At this point the characteristic equation of the matrix of size 2 can be solved by the quadratic formula for the complex eigenvalues.

REFERENCES

1. THE ALGEBRAIC EIGENVALUE PROBLEM, J. H. Wilkinson, Oxford University Press, 1965.

2. COMPUTATIONAL METHODS OF LINEAR ALGEBRA, V. N. Faddeeva, Dover Books, 1959.

```
TYPE 1
1;FORMAT(/,"EIGENVALUE SOLUTION, PART 1,
ITERATIONS TO OBTAIN EIGENVALUE")
C;TRANSFER FROM HIGH SPEED READER TO TELETYPE
TYPE 150
150;FORMAT(/,"DROP BIT 1, HIT CONT")
PAUSE
PAUSE 237
C;PASS 1 READS DATA, COMPUTES LARGEST EIGENVALUE
C;PASS 2,3,ETC. COMPUTES EIGENVALUES OF DEFLATED MATRIX
C;DEFLATED MATRIX LEFT IN CORE BY PART 2
TYPE 151
151;FORMAT(/,"PASS NO.  ")
DIMENSION A(49),X(7),XNEW(7),ASV(7),NPIG(1),N(1),LAST(1)
ACCEPT 3,IPS
152;FORMAT(I)
IF(IPS-1) 153,153,156
C;TRANSFER BACK TO HIGH SPEED READER
153;TYPE 154
154;FORMAT(/,"LIFT BIT 1, LOAD DATA TAPE,HIT CONT")
PAUSE
PAUSE 237
GO TO 155
155;TYPE 2
2;FORMAT(/,/,"SIZE OF ORIGINAL MATRIX  ")
ACCEPT 3,N
3;FORMAT(I)
TYPE 3,N
TYPE 4
4;FORMAT(/,/,"DATA INPUT - A MATRIX",/)
LAST = N*N
DO 5 I = 1,LAST
ACCEPT 6, A(I)
6;FORMAT(E)
5;CONTINUE
C;PRINT OUT MATRIX
IP = 4
DO 8 I = 1,LAST
TYPE 6, A(I)
IF(IP-I) 9,9,8
9;IP=IP+4
TYPE 10
10;FORMAT(/)
8;CONTINUE
C;SET UP FIRST GUESS
156;DO 102 I =1,N
X(I) = 0
102;CONTINUE
X(1) = 1
C;MULTIIPLY MATRIX BY X VECTOR
ITER = 1
100;DO 11 I = 1,N
XNEW(I) = 0
11;CONTINUE
N1 = 1
DO 12 J = 1,LAST,N
DO 13 I = 1,N
XNEW(N1)  = X(I)*A(J+I-1) + XNEW(N1)
13;CONTINUE
```

```
N1 = N1+1
12;CONTINUE
C;NORMALIZE NEW X VECTOR
BIGE = XNEW(1)
NBIG = 1
IF(BIGE) 39,4Ø,4Ø
39;BIGE = -BIGE
4Ø;DO 16 I = 2,N
ETST = XNEW(I)
IF(ETST) 41,16,42
41;ETST = -ETST
42;IF(BIGE-ETST)17,16,16
17;BIGE = ETST
C;NBIG IS INDEX OF LARGEST VALUE IN EIGENVECTOR
NBIG = I
16;CONTINUE
DO 13Ø I = 1,NBIG
BIGE = XNEW(I)
13Ø;CONTINUE
DO 18 I = 1,N
XNEW(I) = XNEW(I)/BIGE
18;CONTINUE
C;TEST FOR CONVERGENCE, NCOV CHECKS THAT ALL ENTRIES WITHIN TOLERANCE
NCOV = Ø
DO 21 I = 1,N
COVG = XNEW(I) - X(I)
IF(COVG) 23,22,22
23;COVG = -COVG
22;IF(COVG - 1.E-Ø5) 24,21,21
24;NCOV = NCOV + 1
IF(NCOV-N) 21,3Ø,3Ø
C;TRANSFER TO 3Ø MEANS VECTOR HAS CONVERGED TO EIGENVALUE
21;CONTINUE
C;NO CONVERGENCE, REPEAT ITERATION
DO 25 I = 1,N
X(I) = XNEW(I)
25;CONTINUE
ITER = ITER + 1
IF(ITER-1ØØØ) 26,26,27
27;TYPE 28
28;FORMAT(/,/,"CALCULATION HAS NOT CONVERGED IN 1ØØØ ITERATIONS,
EIGENVALUE MAY BE COMPLEX")
GO TO 29
26;GO  TO 1ØØ
3Ø;TYPE 31,IPS,BIGE
31;FORMAT(/,/,/,"EIGENVALUE OBTAINED ON PASS  ",I," IS  ",F)
35;TYPE 34
34;FORMAT(/,"LOAD PROGRAM TAPE FOR PART 2 TO DEFLATE MATRIX")
29;STOP
END
```

PART II

```
TYPE 1
1;FORMAT(/,"PART 2 OF EIGENVALUE SOLUTION:
DEFLATION OF MATRIX")
DIMENSION A(49),X(7),XNEW(7),ASV(7),NBIG(1),N(1),LAST(1)
6;FORMAT(E)
10;FORMAT(/)
C;IF LARGEST VALUE IS IN FIRST POSITION, NO INTERCHANGE NEEDED
IF(NBIG-1) 211,211,200
C;INTERCHANGE ROWS OF A
200;IST = ((NBIG-1)*N)+1
IFIN = IST+N-1
ISS = 1
DO 201 I = IST,IFIN
ASV(ISS) = A(I)
ISS = ISS+1
201;CONTINUE
DO 202 I = 1,N
A(I+IST-1) = A(I)
A(I) = ASV(I)
202;CONTINUE
C;INTERCHANGE COLS OF A
IST = NBIG
IFIN = IST+(N*(N-1))
ISS = 1
DO 203 I = IST,IFIN,N
ASV(ISS) = A(I)
ISS = ISS+1
203;CONTINUE
LC = LAST-N+1
ISS = 1
DO 204 I = 1,LC,N
A(I+IST-1) = A(I)
A(I) = ASV(ISS)
ISS = ISS+1
204;CONTINUE
C;INTERCHANGE EIGENVECTOR
XSV = XNEW(1)
DO 205 I = 1,NBIG
XCHG = XNEW(I)
205;CONTINUE
XNEW(1) = XCHG
DO 206 I = 1,NBIG
IF(I-NBIG) 206,207,207
207;XNEW(I) = XSV
206;CONTINUE
C;BEGIN DEFLATION CALCULATION
211;J2=N
J1 = N-1
DO 212 J = 1,J1
J3 = J+1
DO 213 I = 2,N
A(I+J2) = A(I+J2)-A(I)*XNEW(J3)
213;CONTINUE
J2 = J2+N
212;CONTINUE
```

```
C;REDUCE A MATRIX PRIOR TO DETERMINING EIGENVALUE
N = N-1
LAST = N*N
NE = N+2
NED=N
DO 214 I = 1,LAST
A(I) = A(NE+1)
IF(I-NED) 214,215,215
215;NE = NE+1
NED = NED+N
214;CONTINUE
TYPE 216
216;FORMAT(/,"DEFLATED MATRIX WHOSE EIGENVALUE IS TO BE FOUND:",/)
IP = 4
DO 217 I= 1,LAST
TYPE 6,A(I)
IF(IP-I) 218,218,217
218;IP=IP+4
TYPE 10
217;CONTINUE
TYPE 219
219;FORMAT(/,/,"LOAD TAPE FOR PART I TO DETERMINE NEXT EIGENVALUE")
STOP
END
```