Thomas Penick 452 80 6040
M 340L-C April 2, 1998

Computer Assignment 7

beginning on page 142

Chapter 7, section 1, problem 1: Eigenvalues

Matlab Input:

```
A = [2 0;1.5 0.5]
[Eigenvectors, EigenvaluesOnDiag] = eig(A)
B = [-2 1;1 -2]
[Eigenvectors, EigenvaluesOnDiag] = eig(B)
C = [-1 2;-2 4]
[Eigenvectors, EigenvaluesOnDiag] = eig(C)
D = [1 -1;1 1]
[Eigenvectors, EigenvaluesOnDiag] = eig(D)
```

Matlab Output:

```
A =
    2.0000         0
    1.5000    0.5000
Eigenvectors =
         0    0.7071
    1.0000    0.7071
EigenvaluesOnDiag =
    0.5000         0
         0    2.0000
B =
    -2     1
     1    -2
Eigenvectors =
    0.7071    0.7071
   -0.7071    0.7071
EigenvaluesOnDiag =
   -3.0000         0
         0   -1.0000
C =
    -1     2
    -2     4
Eigenvectors =
   -0.8944   -0.4472
   -0.4472   -0.8944
EigenvaluesOnDiag =
     0     0
     0     3
D =
     1    -1
     1     1
Eigenvectors =
  -0.7071            -0.7071
        0 + 0.7071i        0 - 0.7071i
EigenvaluesOnDiag =
   1.0000 + 1.0000i        0
        0             1.0000 - 1.0000i
```

Thomas Penick 452 80 6040
M 340L-C April 2, 1998

$$A = \begin{bmatrix} 2 & 0 \\ 1.5 & 0.5 \end{bmatrix}$$

(a) **x** and *A***x** lined up 4 times in one revolution.

(b) *A* has real, distinct eigenvalues, 2 positive

(d) Eigenvectors are $\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}$. They are linearly independent. Since *A* is a 2 × 2 matrix and there are 2 distinct eigenvalues, the matrix is diagonalizable.

(e) $\lambda = 2,\ 0.5$

$$B = \begin{bmatrix} -2 & 1 \\ 1 & -2 \end{bmatrix}$$

(a) **x** and *B***x** lined up 4 times in one revolution.

(b) *B* has real, distinct eigenvalues, 2 negative

(d) Eigenvectors are $\begin{bmatrix} \sqrt{2}/2 \\ -\sqrt{2}/2 \end{bmatrix}, \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}$. They are linearly independent. Since *B* is a 2 × 2 matrix and there are 2 distinct eigenvalues, the matrix is diagonalizable.

(e) $\lambda = -1,\ -3$

$$C = \begin{bmatrix} -1 & 2 \\ -2 & 4 \end{bmatrix}$$

(a) **x** and *C***x** lined up 2 times in one revolution.

(b) *C* has real, distinct eigenvalues, 1 positive, 1 zero

(d) Eigenvectors are $\begin{bmatrix} -0.894 \\ -0.447 \end{bmatrix}, \begin{bmatrix} -0.447 \\ -0.894 \end{bmatrix}$. They are linearly independent. Since *C* is a 2 × 2 matrix and there are 2 distinct eigenvalues, the matrix is diagonalizable.

(e) $\lambda = 0,\ 3$

$$D = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

(a) **x** and *D***x** never lined up.

(b) *D* has no real eigenvalues.

(d) Since the matrix has no real eigenvalues, the matrix is not diagonalizable.

(e) No real eigenvalues.

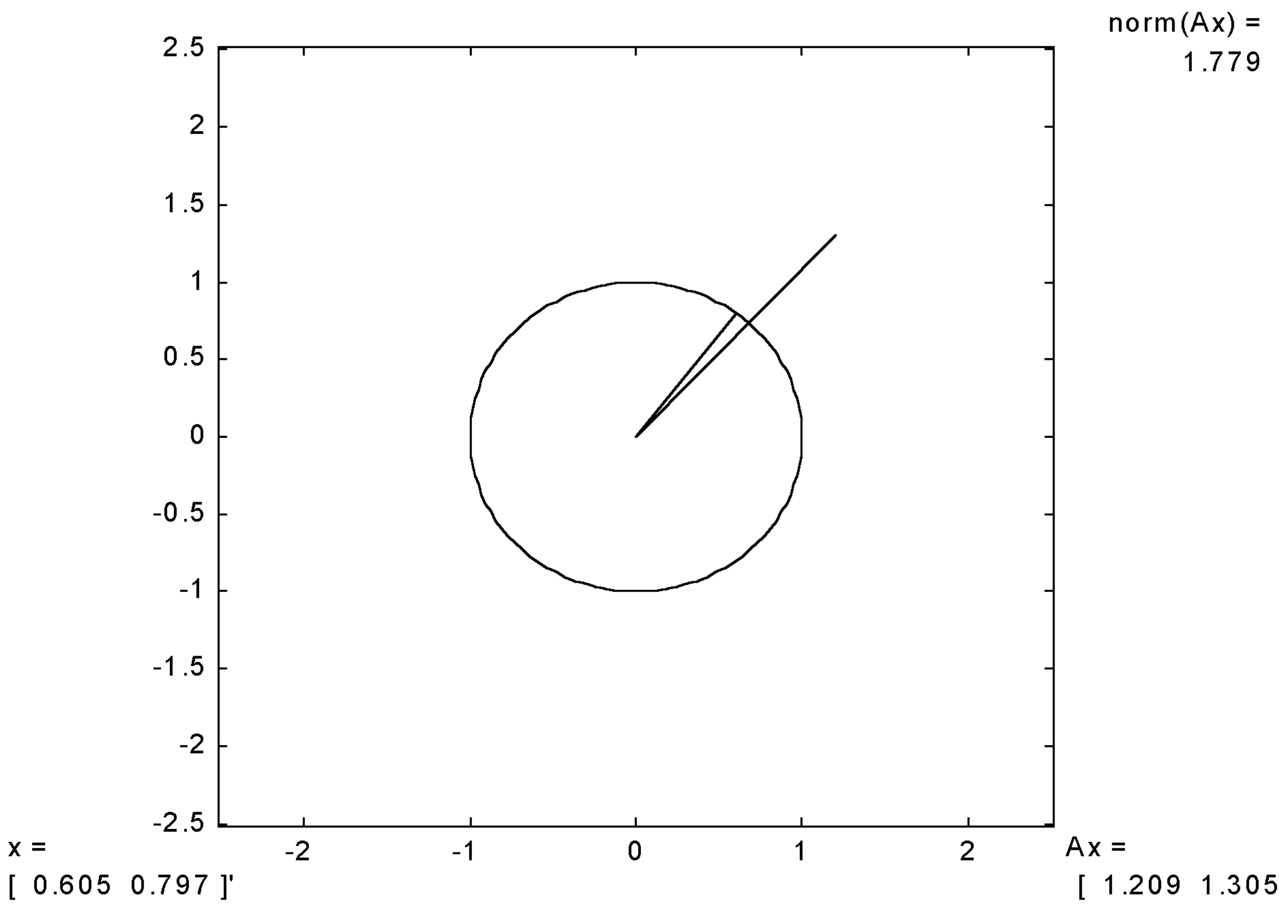
Thomas Penick 452 80 6040
M 340L-C April 2, 1998

Matlab Input:

```
A = [2 0; 1.5 0.5]
eigshow(A)
```

Matlab Output:

```
A =
    2.0000         0
    1.5000    0.5000
```



Thomas Penick 452 80 6040
M 340L-C April 2, 1998

Chapter 7, section 1, problem 1: Eigenvalues

Matlab Input:

```
A = gallery(5)
Eigenvalues = eig(A)
A5 = A^5
Eigenvalues = eig(A5)
CoefficientsOfPolynomial = poly(A)
RootsOfPolynomial = roots(CoefficientsOfPolynomial)
IntegerCoefficientsOfPolynomial = round(CoefficientsOfPolynomial)
RootsOfPolynomial = roots(IntegerCoefficientsOfPolynomial)
RREF = rref(A)
NullityIsNumOfColumns = null(A)
```

Matlab Output:

```
A =
          -9          11         -21          63        -252
          70         -69         141        -421        1684
        -575         575       -1149        3451      -13801
        3891       -3891        7782      -23345       93365
        1024       -1024        2048       -6144       24572
Eigenvalues =
  -0.0328 + 0.0243i
  -0.0328 - 0.0243i
   0.0130 + 0.0379i
   0.0130 - 0.0379i
   0.0396
A5 =
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
Eigenvalues =
     0
     0
     0
     0
     0
CoefficientsOfPolynomial =
    1.0000    0.0000    0.0000    0.0000    0.0000    0.0000
RootsOfPolynomial =
  -0.0328 + 0.0243i
  -0.0328 - 0.0243i
   0.0130 + 0.0379i
   0.0130 - 0.0379i
   0.0396
IntegerCoefficientsOfPolynomial =
     1     0     0     0     0     0
RootsOfPolynomial =
     0
     0
     0
     0
     0
RREF =
    1.0000         0         0         0         0
         0    1.0000         0         0   -0.0820
         0         0    1.0000         0    0.5547
         0         0         0    1.0000   -3.8008
         0         0         0         0         0
NullityIsNumOfColumns =
    0.0000
   -0.0207
    0.1397
   -0.9574
   -0.2519
```

Thomas Penick 452 80 6040
M 340L-C April 2, 1998

(a) The eigenvalues of A^5 are all zeros. Since A^5 is a zero matrix, I would expect the eigenvalues to be zeros, while those of A would be non-zero.

(b) The eigenvalues computed using the commands poly(A) and roots(p) are identical to those computed in part A.

(c) After using the command round(p) and computing the roots again, the result is all zeros.

(e) The nullity of A is 1. A is not diagonalizable because it doesn't have 5 distinct eigenvalues.

Thomas Penick 452 80 6040
M 340L-C April 2, 1998

ABOUT THE MATLAB COMMANDS USED

---

eigshow Eigenvalue/eigenvector demo.

Synopsis: `eigshow(A)`

Graphical demonstration of geometric eigenvalue/eigenvector relationship. A is a square matrix.

---

gallery Small test matrices.

Synopsis: `g = gallery(5)`

returns a 5-by-5 matrix that poses an interesting eigenvalue problem. Using exact arithmetic, gallery(5) has a single eigenvalue at zero with algebraic multiplicity 5; and a single eigenvector. Numerical calculations, however, yield five distinct eigenvalues: two pairs of complex eigenvalues and a real eigenvalue.

For any input other than 3 or 5, gallery returns an empty matrix.

---

eig Eigenvalues.

Synopsis:
```
e = eig(A)
[V,D] = eig(A)
```

`E = eig(A)` is a vector containing the eigenvalues of a square matrix A.

`[V,D] = eig(A)` produces a diagonal matrix *D* of eigenvalues and a full matrix *V* whose columns are the corresponding eignvectors.

---

poly Small test matrices.

Synopsis: `p = poly(A)`

If A is an n-by-n matrix, poly(A) is an n+1 element row vector whose elements are the coefficients of the characteristic polynomial, det(sI - A). The coefficients are ordered in descending powers: if a vector c has n+1 components, the polynomial it represents is $c_1s^n + \ldots + c_ns + c_{n+1}$.

---

roots Polynomial roots.

Synopsis: `r = roots(p)`

Polynomial coefficients are ordered in descending powers. If c is a row vector containing the coefficients of a polynomial, roots(c) is a column vector whose elements are the roots of the polynomial.

If r is a column vector containing the roots of a polynomial, poly(r) returns a row vector whose elements are the coefficients of the polynomial.

---

round Polynomial derivative.

Synopsis: `R = round(X)`

rounds the elements of X to the nearest integers.

---

null Null space of a matrix.

Synopsis: `Q = null(X)`

Returns an orthonormal basis for the null space of *X*. The number of columns of *Q* is the nullity of *X*.

---

rref Reduced row echelon form.

Synopsis: `R = rref(A)`

Produces the reduced row echelon form of matrix *A* using Gauss Jordan elimination.