

R Textbook Companion for
Matrices and Linear Transformations
by Charles G. Cullen¹

Created by
Aswin S
B.Tech.
Mechanical Engineering
Government Engineering College, Barton Hill
Cross-Checked by
R TBC Team

September 21, 2021

¹Funded by a grant from the National Mission on Education through ICT
- <http://spoken-tutorial.org/NMEICT-Intro>. This Textbook Companion and R
codes written in it can be downloaded from the "Textbook Companion Project"
section at the website - <https://r.fossee.in>.

Book Description

Title: Matrices and Linear Transformations

Author: Charles G. Cullen

Publisher: Dover Publications, Inc., New York

Edition: 2

Year: 1990

ISBN: 978-0-486-66328-9

R numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means an R code whose theory is explained in Section 2.3 of the book.

Contents

| | |
|---------------------------------------|----|
| List of R Codes | 4 |
| 1 Matrices and Linear Systems | 5 |
| 2 Vector Spaces | 19 |
| 3 Determinants | 25 |
| 4 Linear Transformations | 30 |
| 5 Similarity Part I | 35 |
| 6 Polynomials and Polynomial Matrices | 38 |
| 8 Matrix Analysis | 40 |
| 9 Numerical Methods | 41 |

List of R Codes

| | | |
|----------|--|----|
| Exa 1.1 | Matrix multiplication | 5 |
| Exa 1.2 | Matrix multiplication | 6 |
| Exa 1.3 | Matrix multiplication | 7 |
| Exa 1.4 | Matrix multiplication | 8 |
| Exa 1.5 | Matrix addition | 9 |
| Exa 1.6 | Matrix addition | 9 |
| Exa 1.7 | Matrix addition | 10 |
| Exa 1.8 | Transposition | 11 |
| Exa 1.9 | Conjugate of complex matrix | 11 |
| Exa 1.10 | Triangular matrix | 12 |
| Exa 1.11 | Symmetric and skew symmetric | 13 |
| Exa 1.12 | Hermitian matrix | 14 |
| Exa 1.13 | Echelon | 16 |
| Exa 1.14 | Inverse of matrix | 16 |
| Exa 1.15 | Inverse of matrix | 17 |
| Exa 2.1 | Subspaces | 19 |
| Exa 2.2 | Subspaces | 20 |
| Exa 2.3 | Echelon | 20 |
| Exa 2.4 | Echelon | 21 |
| Exa 2.5 | Echelon | 21 |
| Exa 2.6 | Echelon | 22 |
| Exa 2.7 | Rank of a matrix | 23 |
| Exa 3.1 | Determinant | 25 |
| Exa 3.2 | Minor and cofactor | 25 |
| Exa 3.3 | Determinant | 26 |
| Exa 3.4 | Determinant | 26 |
| Exa 3.5 | Adjoins | 27 |
| Exa 3.6 | Laplace expansion | 28 |

| | | |
|---------|---|----|
| Exa 3.7 | Rank of a matrix | 28 |
| Exa 4.1 | w coordinate matrices | 30 |
| Exa 4.2 | Change of basis and similarity | 31 |
| Exa 4.3 | Change of basis and similarity | 31 |
| Exa 4.4 | Eigenvalues and eigenvectors | 32 |
| Exa 4.5 | Characteristic polynomial | 33 |
| Exa 4.6 | Characteristic polynomial | 33 |
| Exa 5.1 | Minimum polynomial | 35 |
| Exa 5.2 | Characteristic polynomial | 36 |
| Exa 6.1 | Left and right functional value | 38 |
| Exa 8.1 | Primary functions | 40 |
| Exa 9.1 | Echelon | 41 |
| Exa 9.2 | Echelon | 42 |
| Exa 9.3 | Inverse of matrix | 42 |
| Exa 9.4 | Eigenvalues and eigenvectors | 43 |

Chapter 1

Matrices and Linear Systems

R code Exa 1.1 Matrix multiplication

```
1 #page - 20
2 #section - 1.3 MATRICES
3 #example 1
4
5
6 #first matrix A
7 A <- matrix(c(2,-1,0,4,3,-2), 2, 3, byrow=TRUE)
8 A
9 #second matrix B
10 B <- matrix(c(-4,0,6,-4,1,6), 3, 2, byrow=TRUE)
11 B
12 #multiplying A and B to AB
13 AB <- A %*% B
14
15 #printing AB
16 AB
17
18 #multiplying B and A to BA
19 BA <- B %*% A
20
21 #printing BA
```

R code Exa 1.2 Matrix multiplication

```
1 #page - 21
2 #section - 1.3 MATRICES
3 #example 2
4
5
6 #first matrix A
7 A <- matrix(c(1,2,3,4), 2, 2, byrow=TRUE)
8 A
9 #second matrix B
10 B <- matrix(c(5,6,7,8), 2, 2, byrow=TRUE)
11 B
12 #multiplying A and B to AB
13 AB <- A %*% B
14
15 #printing AB
16 AB
17
18 #multiplying B and A to BA
19 BA <- B %*% A
20
21 #printing BA
22 BA
23
24 #function to compare two matrices
25 matequal <- function(x, y)
26     is.matrix(x) && is.matrix(y) && dim(x) == dim(y)
27     && all(x == y)
28
29 #checking if AB and BA are equal
30 matequal(AB, BA)
```

R code Exa 1.3 Matrix multiplication

```
1 #page - 23
2 #section - 1.3 MATRICES
3 #example 4
4
5
6 #first matrix A
7 A <- matrix(c(2,-1,1,-1,2,-1,1,-1,2), 3, 3, byrow=
  TRUE)
8 A
9 #second matrix B
10 B <- matrix(c(6,-5,5,-5,6,-5,5,-5,6), 3, 3, byrow=
  TRUE)
11 B
12 #multiplying A and B to AB
13 AB <- A %*% B
14
15 #printing AB
16 AB
17
18 #multiplying B and A to BA
19 BA <- B %*% A
20
21 #printing BA
22 BA
23
24 #function to compare two matrices
25 matequal <- function(x, y)
26   is.matrix(x) && is.matrix(y) && dim(x) == dim(y)
  && all(x == y)
27
28 #checking if AB and BA are equal
29 matequal(AB, BA)
```

R code Exa 1.4 Matrix multiplication

```
1 #page - 23
2 #section - 1.3 MATRICES
3 #example 4
4
5
6 #first matrix A
7 A <- matrix(c(3,1,0,1,-1,2,1,1,1), 3, 3, byrow=TRUE)
8 A
9 #second matrix B
10 B <- matrix(c(3/8,1/8,-2/8,-1/8,-3/8,6/8,-2/8,2/8,4/
11 8), 3, 3, byrow=TRUE)
12 B
13 #multiplying A and B to AB
14 AB <- A %*% B
15 AB
16 #printing AB
17 AB
18 #multiplying B and A to BA
19 BA <- B %*% A
20 BA
21 #printing BA
22 BA
23
24 #function to compare two matrices
25 matequal <- function(x, y)
26   is.matrix(x) && is.matrix(y) && dim(x) == dim(y)
27   && all(x == y)
28 #checking if AB and BA are equal
29 matequal(AB, BA)
30
```

```
31 #identity matrix I
32 I <-diag(3)
33 I
34
35 #checking if AB and I are equal
36 matequal(AB, I)
```

R code Exa 1.5 Matrix addition

```
1 #page - 27
2 #section - 1.4 MATRIX ADDITION AND SCALAR
  MULTIPLICATION
3 #example 5
4
5
6 #first matrix A
7 A <- matrix(c(3,-2,4,6,0,-3,-1,7,10,9,5,0), 3, 4,
  byrow=TRUE)
8 A
9 #second matrix B
10 B <- matrix(c(2,4,6,8,0,1,3,5,7,9,-2,-4), 3, 4,
  byrow=TRUE)
11 B
12 #adding A and B to C
13 C <- A + B
14
15 #printing C
16 C
```

R code Exa 1.6 Matrix addition

```
1 #page - 27
```

```

2 #section - 1.4 MATRIX ADDITION AND SCALAR
  MULTIPLICATION
3 #example 6
4
5
6 #first matrix M
7 M <- matrix(c(3,0,7,-2,4,0,5,6,0), 3, 3)
8 M
9 #second matrix S
10 S <- matrix(c(-3,0,-7,2,-4,0,-5,-6,0), 3, 3)
11 S
12 #adding M and S to C
13 C <- M + S
14
15 #printing C
16 C

```

R code Exa 1.7 Matrix addition

```

1 #page - 30
2 #section - 1.4 MATRIX ADDITION AND SCALAR
  MULTIPLICATION
3 #example 7
4
5
6 #first matrix A
7 A <- matrix(c(1,4,2,0,-3,5), 2, 3)
8 A
9 #second matrix B
10 B <- matrix(c(4,8,-7,7,3,0), 2,3)
11 B
12 #performing the action 2A-3B and equating it to C
13 C <- (A*2) - (B*3)
14
15

```

```
16 #printing C
17 C
```

R code Exa 1.8 Transposition

```
1 #page - 32
2 #section - 1.5 TRANSPOSITION
3 #example 8
4
5 #matrix A
6 A <- matrix(c(1,2,3,4,5,6), 2, 3, byrow=TRUE)
7 A
8
9 #matrix B
10 B <- matrix(c(5,-3,2), 3, 1)
11 B
12
13 #transpose of matrix A - AT
14 AT = t(A)
15 AT
16
17 #transpose of matrix B - BT
18 BT = t(B)
19 BT
```

R code Exa 1.9 Conjugate of complex matrix

```
1 #page - 34
2 #section - 1.5 TRANSPOSITION
3 #example 9
4
5 #complex matrix A
```

```

6 A <- matrix(c(3, 5+6*1i , 2, 0+3*1i, 2+3*1i , 2-1i )
, 3, 2, byrow=TRUE)
7 A
8
9 #conjugate of complex matrix A, A_
10 A_ = Conj(A)
11 A_
12
13 #transpose of conjugate of complex matrix A, Astar
14 Astar = Conj(t(A))
15 Astar

```

R code Exa 1.10 Triangular matrix

```

1 #page - 40
2 #section - 1.7 SPECIAL KINDS OF MATRICES
3 #example 10
4
5 #matrix L
6 L <- matrix(c(2,0,0,3,0,0,-1,0,4), 3, 3, byrow=TRUE)
7 L
8
9 #matrix U
10 U <- matrix(c(0,7,9,0,0,-1,0,0,0), 3, 3, byrow=TRUE)
11 U
12
13 #creating Lcheck to create a lower triangular matrix
    of L
14 Lcheck <- L
15 lower.tri(Lcheck)
16 Lcheck[upper.tri(Lcheck)] <- 0
17 Lcheck
18
19 #creating Ucheck to create a upper triangular matrix
    of U

```

```

20 Ucheck <- U
21 lower.tri(Ucheck)
22 Ucheck[lower.tri(Ucheck)] <- 0
23 Ucheck
24
25 #function to compare two matrices
26 matequal <- function(x, y)
27     is.matrix(x) && is.matrix(y) && dim(x) == dim(y)
28     && all(x == y)
29
30 #checking if L and U are lower and upper triangular
31 #matrix respectively
32 matequal(L, Lcheck)
33 matequal(U, Ucheck)

```

R code Exa 1.11 Symmetric and skew symmetric

```

1 #page - 41
2 #section - 1.7 SPECIAL KINDS OF MATRICES
3 #example 11
4
5 #matrix A
6 A <- matrix(c(1,2,3,4,2,5,-6,7,3,-6,8,-9,4,7,-9,0),
7             4, 4, byrow=TRUE)
8
9 #matrix B
10 B <- matrix(c
11             (0,1,2,3,-1,0,-4,5,-2,4,0,6,-3,-5,-6,-0), 4, 4,
12             byrow=TRUE)
13
14 A
15 AT = t(A)
16 AT

```

```

16 BT = t(B)
17 BT
18
19 #function to compare two matrices
20
21 #symmetric check function
22 matsym <- function(x, y)
23     is.matrix(x) && is.matrix(y) && dim(x) == dim(y)
24     && all(x == y)
25
26 #skew symmetric check function
27 matskew <- function(x, y)
28     is.matrix(x) && is.matrix(y) && dim(x) == dim(y)
29     && all(x == -y)
30
31 #condition check
32 if(matsym(AT, A)){
33     print("A is a symmetric matrix")
34 }else if(matskew(AT, A)){
35     print("A is a skew symmetric matrix")
36 }else{
37     print("none")
38 }
39
40 if(matsym(BT, B)){
41     print("B is a symmetric matrix")
42 }else if(matskew(BT, B)){
43     print("B is a skew symmetric matrix")
44 }else{
45     print("none")
46 }

```

R code Exa 1.12 Hermitian matrix

```

1 #page - 42
2 #section - 1.5 TRANSPOSITION

```



```

3 #example 12
4
5 #complex matrix A
6 A <- matrix(c(2, 5*1i , 2-3*1i, -5*1i, 3 , 4, 2+3*1i
  , 4, 0), 3, 3, byrow=TRUE)
7 A
8
9 #complex matrix B
10 B <- matrix(c(1i, 3 , 5*1i, -3, 0 , 2+3*1i, 5*1i, ,
  0), -2+3*1i, 3*1i), 3, 3, byrow=TRUE)
11 B
12
13 #conjugate transpose of complex matrix A, A_
14 Astar = Conj(t(A))
15 Astar
16
17 #conjugate transpose of complex matrix B, B_
18 Bstar = Conj(t(B))
19 Bstar
20
21 #function to compare two matrices
22
23 #Hermitian matrix check function
24 matHer <- function(x, y)
25     is.matrix(x) && is.matrix(y) && dim(x) == dim(y)
  && all(x == y)
26
27 #skew Hermitian matrix check function
28 matskewHer <- function(x, y)
29     is.matrix(x) && is.matrix(y) && dim(x) == dim(y)
  && all(x == -y)
30
31 #condition check
32 if(matHer(Astar, A)){
33     print("A is a Hermitian matrix")
34 }else if(matskewHer(Astar, A)){
35     print("A is a skew Hermitian matrix")
36 }else{

```

```

37     print("none")
38 }
39 if(matHer(Bstar, B)){
40     print("B is a Hermitian matrix")
41 }else if(matskewHer(Bstar, B)){
42     print("B is a skew Hermitian matrix")
43 }else{
44     print("none")
45 }

```

R code Exa 1.13 Echelon

```

1 #page - 45
2 #section - 1.8 ROW EQUIVALENCE
3 #example 13
4
5 #included package - matlib
6
7 #for echelon function
8 library(matlib)
9
10 #matrix A
11 A <- matrix(c(1,2,3,1,3,2,1,1,0,2,4,1,1,1,1,1), 4,
12             4, byrow=TRUE)
13 #column matrix k
14 K <- c(3,7,1,4)
15
16 #reduced row-echelon form
17 echelon(A, K, verbose=TRUE, fractions=TRUE)

```

R code Exa 1.14 Inverse of matrix

```

1 #page - 53
2 #section - 1.9 ELEMENTARY MATRICES AND MATRIX
  INVERSES
3 #example 14
4
5 #included package - matlib
6
7 #for inverse functions
8 library(matlib)
9
10 #matrix A
11 A <- matrix(c(1,-1,2,3,2,-1,0,2,4,1,-11,-1,1,2,3,83)
  , 4, 4, byrow=TRUE)
12 A
13
14 #determinant of A
15 det(A)
16
17 #inverse matrix of A, AI
18 (AI <- inv(A))

```

R code Exa 1.15 Inverse of matrix

```

1 #page - 55
2 #section - 1.9 ELEMENTARY MATRICES AND MATRIX
  INVERSES
3 #example 15
4
5 #included package - matlib
6
7 #for inverse functions
8 library(matlib)
9
10 #matrix B
11 B <- matrix(c(3,-1,2,2,1,1,1,-3,0), 3, 3, byrow=TRUE)

```

```
    )  
12 B  
13  
14 #determinant of matrix B  
15 det(B)  
16  
17 #inverse of matrix B, BI  
18 (BI <- inv(B))
```

Chapter 2

Vector Spaces

R code Exa 2.1 Subspaces

```
1 #page - 74
2 #section - 2.2 SUBSPACES
3 #example 1
4
5 #let us represent 'a' with a value as we cannot
  multiply character matrix with numerical matrix
6 a = 7
7
8 #let unit vectors of F4*1 be E1, E2, E3, E,4
9 E1 <- matrix(c(1,0,0,0), 4, 1)
10 E2 <- matrix(c(0,1,0,0), 4, 1)
11 E3 <- matrix(c(0,0,1,0), 4, 1)
12 E4 <- matrix(c(0,0,0,1), 4, 1)
13
14 #thus vector A is
15 A = (E1 %*% a) + (E2 %*% a) + (E3 %*% a) + (E4 %*% a
  )
16 A
```

R code Exa 2.2 Subspaces

```
1 #page - 75
2 #section - 2.2 SUBSPACES
3 #example 2
4
5 #included package - matlib
6
7 #for echelon function
8 library(matlib)
9
10 #matrix A
11 A <- matrix(c(1,-1,-3,-1,2,5,1,2,6), 3, 3, byrow=
    TRUE)
12
13 #column matrix k
14 K <- c(-6,10,15)
15
16 #reduced row-echelon form
17 echelon(A, K, reduced=FALSE, verbose=TRUE, fractions
    =TRUE)
```

R code Exa 2.3 Echelon

```
1 #page - 80
2 #section - 2.3 LINEAR INDEPENDENCE AND BASES
3 #example 3
4
5 #included package - matlib
6
7 #for echelon function
8 library(matlib)
9
10 #matrix A
11 A <- matrix(c(1,-2,1,2,-5,0,-1,3,1,2,0,10), 4, 3,
```

```

    byrow=TRUE)
12
13 #column matrix k
14 K <- c(0,0,0,0)
15
16 #reduced row-echelon form
17 echelon(A, K, reduced=FALSE, verbose=TRUE, fractions
    =TRUE)

```

R code Exa 2.4 Echelon

```

1 #page - 81
2 #section - 2.3 LINEAR INDEPENDENCE AND BASES
3 #example 4
4
5 #included package - matlib
6
7 #for echelon function
8 library(matlib)
9
10 #matrix A
11 A <- matrix(c(-1,-2,2,0,1,3,1,1,1), 3, 3, byrow=TRUE
    )
12
13 #column matrix k
14 K <- c(0,0,0)
15
16 #reduced row-echelon form
17 echelon(A, K, reduced=FALSE, verbose=TRUE, fractions
    =TRUE)

```

R code Exa 2.5 Echelon

```

1 #page - 82
2 #section - 2.3 LINEAR INDEPENDENCE AND BASES
3 #example 5
4
5 #included package - matlab
6
7 #for echelon function
8 library(matlab)
9
10 #matrix A
11 A <- matrix(c(1,2,3,3,2,1,0,2,4,1,1,1), 4, 3, byrow=
      TRUE)
12
13 #column matrix k
14 K <- c(1,1,1,1)
15
16 #reduced row-echelon form
17 B = echelon(A, K, reduced=TRUE, verbose=TRUE,
      fractions=TRUE)

```

R code Exa 2.6 Echelon

```

1 #page - 82
2 #section - 2.3 LINEAR INDEPENDENCE AND BASES
3 #example 6
4
5 #included package - matlab
6
7 #for echelon function
8 library(matlab)
9
10 #matrix A
11 A <- matrix(c(3,-1,-1,-2,2,-2,-1,-1,3), 3, 3, byrow=
      TRUE)
12

```



```

13 #column matrix k
14 K <- c(-3,2,1)
15
16 #reduced row-echelon form
17 B = echelon(A, K, reduced=TRUE, verbose=TRUE,
             fractions=TRUE)

```

R code Exa 2.7 Rank of a matrix

```

1 #page - 88
2 #section - 2.4 THE RANK OF A MATRIX
3 #example 7
4
5 #included package - matlib
6 #included package - matrixcalc
7
8 #for echelon function
9 library(matlib)
10
11 #for rank calculation
12 library(matrixcalc)
13
14 #matrix A
15 A <- matrix(c(2,1,1,1,-2,1,0,5,-1), 3, 3, byrow=TRUE
             )
16
17 #column matrix k
18 K <- c(2,-3,8)
19
20 #reduced row-echelon form
21 B = echelon(A, K, reduced=TRUE, verbose=TRUE,
             fractions=TRUE)
22
23 #rank of A
24 matrix.rank(A)

```


Chapter 3

Determinants

R code Exa 3.1 Determinant

```
1 #page - 103
2 #section - 3.1 DEFINITION OF THE DETERMINANT
3 #example 1
4
5 #matrix A
6 A <- matrix(c(1,2,3,2,4,1,1,3,0), 3, 3, byrow=TRUE)
7 A
8
9 #determinant of A
10 det(A)
```

R code Exa 3.2 Minor and cofactor

```
1 #page - 106
2 #section - 3.2 THE LAPLACE EXPANSION
3 #example 2
4
5 #matrix A
```

```

6 A <- matrix(c(1,2,3,4,5,6,7,8,9), 3, 3, byrow=TRUE)
7 A
8
9 # Minor and cofactor functions
10 minor <- function(A, i, j) A[-i,-j]
11 cofactor <- function(A, i, j) (-1)^(i+j) * det(minor
      (A,i,j))
12
13 #calculating Minor and cofactor
14
15 minor(A, 1, 2)
16 cofactor(A, 1, 2)
17
18 minor(A, 3, 3)
19 cofactor(A, 3, 3)

```

R code Exa 3.3 Determinant

```

1 #page - 108
2 #section - 3.2 THE LAPLACE EXPANSION
3 #example 3
4
5 #matrix A
6 A <- matrix(c(2,4,6,1,2,3,1,4,9), 3, 3, byrow=TRUE)
7 A
8
9 #determinant of A
10 det(A)

```

R code Exa 3.4 Determinant

```

1 #page - 108
2 #section - 3.2 THE LAPLACE EXPANSION

```

```

3 #example 4
4
5 #matrix A
6 A <- matrix(c(1,-1,2,3,2,2,0,2,4,1,-1,-1,1,2,3,0),
7             4, 4, byrow=TRUE)
8
9 #determinant of A
10 det(A)

```

R code Exa 3.5 Adjoints

```

1 #page - 111
2 #section - 3.3 ADJOINTS AND INVERSES
3 #example 5
4
5 #matrix A
6 A <- matrix(c(2,-1,1,4,2,4,6,3,9), 3, 3)
7
8 # Minor and cofactor functions
9 minor <- function(A, i, j) det(A[-i,-j])
10 cofactor <- function(A, i, j) (-1)^(i+j) * minor(A,i
11             ,j)
12
13 #Adjoint functions
14 adjoint <- function(x) {
15   n <- nrow(x)
16   B <- matrix(NA, n, n)
17   for( i in 1:n )
18     for( j in 1:n )
19       B[j,i] <- cofactor(x, i, j)
20 }
21 adjoint(A)

```

R code Exa 3.6 Laplace expansion

```
1 #page - 114
2 #section - 3.2 THE LAPLACE EXPANSION
3 #example 6
4
5 #column matrices
6 c1 <- matrix(c(1,1,2), 3, 1, byrow=TRUE)
7 c2 <- matrix(c(2,-1,3), 3, 1, byrow=TRUE)
8 c3 <- matrix(c(1,1,-1), 3, 1, byrow=TRUE)
9 c4 <- matrix(c(4,5,1), 3, 1, byrow=TRUE)
10
11 A <- matrix(c(c1,c2,c3), 3, 3)
12 B <- matrix(c(c4,c2,c3), 3, 3)
13 C <- matrix(c(c1,c4,c3), 3, 3)
14 D <- matrix(c(c1,c2,c4), 3, 3)
15 A
16 B
17 C
18 D
19 #solution of r,s and t
20 r = det(B)/det(A)
21 r
22
23 s=det(C)/det(A)
24 s
25
26 t=det(D)/det(A)
27 t
```

R code Exa 3.7 Rank of a matrix

```
1 #page - 115
2 #section - 3.4 DETERMINANTS AND RANK
3 #example 7
4
5 #included package - matrixcalc
6
7 #for rank calculation
8 library(matrixcalc)
9
10 #matrix A
11 A <- matrix(c
12             (1,-1,1,1,1,2,-1,-1,2,-2,1,-1,0,-3,-1,-1), 4, 4,
13             byrow=TRUE)
14
15 #matrix N
16 N <- matrix(c(0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0), 4,
17             4, byrow=TRUE)
18
19 #rank of A
20 matrix.rank(A)
21
22 #rank of N
23 matrix.rank(N)
```

Chapter 4

Linear Transformations

R code Exa 4.1 w coordinate matrices

```
1 #page - 123
2 #section - 4.2 MATRIX REPRESENTATION
3 #example 1
4
5 #w-coordinate matrices
6 Crdwb1 <- matrix(c(-2,5,6,-4), 4, 1, byrow=TRUE)
7 Crdwb2 <- matrix(c(-5,12,11,-5), 4, 1, byrow=TRUE)
8 Crdwb3 <- matrix(c(3,-5,-5,3), 4, 1, byrow=TRUE)
9
10 #matrix T
11 Tmat <- matrix(c(Crdwb1,Crdwb2,Crdwb3), 4, 3)
12 Tmat
13
14 Crdvw <- matrix(c(3,-5,7), 3, 1)
15 Crdvw
16
17 CrdwTv = Tmat %*% Crdvw
18 CrdwTv
```

R code Exa 4.2 Change of basis and similarity

```
1 #page - 131
2 #section - 4.4 CHANGE OF BASIS AND SIMILARITY
3 #example - 2
4
5 #included package - matlib
6
7 #for inverse functions
8 library(matlib)
9
10 #matrix M
11 M <- matrix(c(2,4,6,-1,2,3,1,4,9), 3, 3, byrow=TRUE)
12 M
13
14 #matrix P
15 P <- matrix(c(1,0,1,1,1,1,1,1,0), 3, 3, byrow=TRUE)
16 P
17
18 #inverse of matrix P, PI
19 (PI <- inv(P))
20
21 Mtx = PI %*% M %*% P
22 Mtx
```

R code Exa 4.3 Change of basis and similarity

```
1 #page - 132
2 #section - 4.4 CHANGE OF BASIS AND SIMILARITY
3 #example - 3
4
5 #included Package - matlib
6
7 #for inverse functions
8 library(matlib)
```

```

9
10 #matrix M
11 M <- matrix(c(17,12,18,-16,-9,-24,-5,-4,-4), 3, 3,
              byrow=TRUE)
12 M
13
14 #matrix Q
15 Q <- matrix(c(10,-3,-3,-8,3,2,-3,1,1), 3, 3, byrow=
              TRUE)
16 Q
17
18 #inverse of matrix Q, QI
19 (QI <- inv(Q))
20
21 Mtx = QI %*% M %*% Q
22 Mtx

```

R code Exa 4.4 Eigenvalues and eigenvectors

```

1 #page - 135
2 #section - 4.5 CHARACTERISTIC VECTORS AND
  CHARACTERISTIC VALUES
3 #example - 4
4
5 #matrix A
6 A <- matrix(c(7,-8,-8,9,-16,-18,-5,11,13), 3, 3,
              byrow=TRUE)
7 A
8
9 #matrix X
10 X <- matrix(c(1,3,-2), 3, 1, byrow=TRUE)
11 X
12
13 A %*% X
14

```

```
15 #eigenvalues and eigenvectors
16 eigen(A)
```

R code Exa 4.5 Characteristic polynomial

```
1 #page - 137
2 #section - 4.5 CHARACTERISTIC VECTORS AND
  CHARACTERISTIC VALUES
3 #example - 5
4
5 #included package - pracma
6
7 #for charpoly function
8 library(pracma)
9
10 #matrix A
11 A <- matrix(c(2,1,1,2,3,2,1,1,2), 3, 3, byrow=TRUE)
12 A
13
14 #characteristic polynomial
15 charpoly(A, info = FALSE)
```

R code Exa 4.6 Characteristic polynomial

```
1 #page - 156
2 #section - 4.8 SCHUR'S THEOREM AND NORMAL MATRICES
3 #example - 6
4
5 #included package - pracma
6
7 #for charpoly function
8 library(pracma)
9
```

```
10 #matrix A
11 B <- matrix(c(4,-1,1,-1,4,-1,1,-1,4), 3, 3, byrow=
    TRUE)
12 B
13
14 #characteristic polynomial
15 charpoly(B, info = FALSE)
```

Chapter 5

Similarity Part I

R code Exa 5.1 Minimum polynomial

```
1 #page - 167
2 #section - 5.1 THE CAYLEY-HAMILTON THEOREM
3 #example - 1
4
5 #included package - polynom
6
7 #for minimum polynomial function
8 library(polynom)
9
10 #matrix B
11 B <- matrix(c
      (17, -8, -12, 14, 46, -22, -35, 41, -2, 1, 4, -4, 4, -2, -2, 3),
      4, 4, byrow = TRUE)
12 B
13
14 eigVals <- eigen(B)$values
15 multEig <- table(eigVals)
16 k <- length(multEig)
17 minPoly <- 1
18 for(i in 1:k){
19   poly.i <- polynomial(c(-as.numeric(names(multEig)[
```

```

        i]), 1))
20   minPoly <- (minPoly*poly.i)
21 }
22
23 #minimum polynomial
24 minPoly

```

R code Exa 5.2 Characteristic polynomial

```

1 #page - 168
2 #section - 5.1 THE CAYLEY-HAMILTON THEOREM
3 #example - 2
4
5 #included packages - pracma, polynom
6
7 #for charpoly function
8 library(pracma)
9
10 #for minimum polynomial function
11 library(polynom)
12
13 #matrix S
14 S <- matrix(c(4,-1,1,-1,4,-1,1,-1,4), 3, 3, byrow =
      TRUE)
15 S
16
17 eigVals <- eigen(S)$values
18 multEig <- table(eigVals)
19 k <- length(multEig)
20 minPoly <- 1
21 for(i in 1:k){
22   poly.i <- polynomial(c(-as.numeric(names(multEig)[
      i]), 1))
23   minPoly <- (minPoly*poly.i)
24 }

```

```
25
26 #characteristic polynomial
27 charpoly(S, info = FALSE)
28
29 #minimum polynomial
30 minPoly
```

Chapter 6

Polynomials and Polynomial Matrices

R code Exa 6.1 Left and right functional value

```
1 #page - 208
2 #section - 6.4 MATRICES WITH POLYNOMIAL ELEMENTS
3 #example 1
4
5 #included package - expm
6
7 #for power of a matrix
8 library(expm)
9
10 #constant matrices
11 c1 <- matrix(c(0,0,1,1), 2, 2, byrow=TRUE)
12 c2 <- matrix(c(0,1,0,1), 2, 2, byrow=TRUE)
13 c3 <- matrix(c(1,1,0,1), 2, 2, byrow=TRUE)
14
15 #matrix A
16 A <- matrix(c(1,0,0,2), 2, 2)
17 A
18
19 #square of A
```



```
20 A2 = A %^% 2
21 A2
22
23 #right functional value
24 Pr = c1%*%A2 + c2%*%A + c3
25 Pr
26
27 #left functional value
28 Pi = A2%*%c1 + A%*%c2 + c3
29 Pi
30
31 #function to compare two matrices
32 matequal <- function(x, y)
33   is.matrix(x) && is.matrix(y) && dim(x) == dim(y) &
34     & all(x == y)
35 matequal(Pr, Pi)
```

Chapter 8

Matrix Analysis

R code Exa 8.1 Primary functions

```
1 #page - 242
2 #section - 8.2 PRIMARY FUNCTIONS
3 #example 1
4
5
6 #first matrix A
7 A <- matrix(c(2,1,1,2,3,2,1,1,2), 2, 3, byrow=TRUE)
8 A
9
10 fn <- function(z)
11   sin((pi/2)*z)
12
13 fn(A)
```

Chapter 9

Numerical Methods

R code Exa 9.1 Echelon

```
1 #page - 253
2 #section - 9.2 EXACT METHODS FOR SOLVING AX = K
3 #example 1
4
5 #included package - matlib
6
7 #for echelon function
8 library(matlib)
9
10 #matrix A
11 A <- matrix(c(2,1,-1,2,1,3,2,-3,-1,2,1,-1,2,-3,-1,4)
12             , 4, 4, byrow=TRUE)
13
14 #column matrix k
15 K <- c(1,0,1,0)
16
17 #reduced row-echelon form
18 echelon(A, K, reduced=TRUE, verbose=TRUE, fractions=
19         FALSE)
```

R code Exa 9.2 Echelon

```
1 #page - 254
2 #section - 9.2 EXACT METHODS FOR SOLVING AX = K
3 #example 2
4
5 #included package - matlib
6
7 #for echelon function
8 library(matlib)
9
10 #matrix A
11 A <- matrix(c
      (7,9,2,-1,4,-5,-7,2,3,-2,-5,-1,1,6,-4,-3), 4, 4,
      byrow=TRUE)
12
13 #column matrix k
14 K <- c(1,2,4,3)
15
16 #reduced row-echelon form
17 echelon(A, K, reduced=TRUE, verbose=TRUE, fractions=
      TRUE)
```

R code Exa 9.3 Inverse of matrix

```
1 #page - 259
2 #section - 9.2 EXACT METHODS FOR SOLVING AX = K
3 #example 3
4
5 #included package - matlib
6
7 #for inverse functions
```

```

8 library(matlib)
9
10 #matrix A
11 A <- matrix(c
      (1,0,2,-1,4,5,3,-1,0,1,8,5,-3,1,4,6,2,0,0,1,0,1,4,2,0)
      , 5, 5, byrow=TRUE)
12 A
13
14 #determinant of matrix A
15 det(A)
16
17 #inverse of matrix A, AI
18 (AI <- inv(A))

```

R code Exa 9.4 Eigenvalues and eigenvectors

```

1 #page - 265
2 #section - 9.4 CHARACTERISTIC VALUES AND VECTORS
3 #example - 4
4
5 #matrix A
6 A <- matrix(c(1,1,3,1,-2,1,3,1,3), 3, 3, byrow=TRUE)
7 A
8
9 #matrix X
10 X <- matrix(c(1,1,1), 3, 1, byrow=TRUE)
11 X
12
13 #eigenvalues and eigenvectors
14 eigen(A)

```
