R Textbook Companion for
Linear Algebra
by Jim Hefferon[1]

Created by
Ajay Ragh
B.Tech.
Information Technology
APJ Abdul Kalam Technological University
Cross-Checked by
R TBC Team

May 26, 2020

Book Description

Title: Linear Algebra

Author: Jim Hefferon

Publisher: Orthogonal Publishing L3C

Edition: 3

Year: 2017

ISBN: 978-1944325039

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

Chapter 1

Linear Systems

R code Exa 2.1 solution of system of linear equations

```r
1  #Example 2.1,chapter1,section 1.2,page 13
2  #package used matlib v0.9.1
3  #install package using command: install.packages("
     matlib")
4  #Github reposiory of matlib :https://github.com/
     friendly/matlib
5
6  #installation and loading library
7  #install.packages("matlib")
8  library("matlib")
9
10 #program
11 A <- matrix(c(2,1,3,0,-1,-1,1,-1,0),ncol=3)
12 b <- c(3,1,4)
13 Ab <- cbind(A,b)
14 Ab <- rowadd(Ab,1,2,-0.5)
15 Ab <- rowadd(Ab,1,3,-1.5)
16 Ab <- rowadd(Ab,2,3,-1)
17 Ab
18 #from the result we can see that the system doesnot
     have a unique solution.
```

```
19 #We can represent the solution set by representing
       the variables that lead(x,y)by the variable that
       does not lead(z).
20 #so solution set {((3/2)−(1/2)z,(1/2)−(3/2)z,z)|z
       belongs to R}
```

R code Exa 1.3 Solving Linear Systems

```
1 #Example 1.3
2 #Section I. Solving Linear Systems,page 3
3 A <- matrix(c(3,-1,2,1),ncol = 2)
4 b <- c(7,6)
5 x <- solve(A) %*% b
6 x
```

R code Exa 2.3 solution of system of linear equations

```
1 #Example 2.3,chapter1,section 1.2,page 13
2 #package used matlib v0.9.1
3 #install package using command: install.packages("
      matlib")
4 #Github reposiory of matlib :https://github.com/
      friendly/matlib
5
6 #installation and loading library
7 #install.packages("matlib")
8 library("matlib")
9
10 #program
11 A <- matrix(c(1,0,3,0,1,1,0,-1,1,-1,6,1,-1,1,-6,-1),
      ncol = 4)
12 b <- c(1,-1,6,1)
13 Ab <- cbind(A,b)
```

```
14  Ab <- rowadd(Ab,1,3,-3)
15  Ab <- rowadd(Ab,2,3,3)
16  Ab <- rowadd(Ab,2,4,1)
17  Ab
18  #from the reduced augmented matrix Ab,we can see
        that only x and y are leading ,while z and w are
        free .
19  #We represent the solution set by expressing the
        leading variables in terms of the free variables
20  #The solution set: {(2 − 2z + 2w,−1 + z − w,z ,w)|z ,w
         belongs to R}
```

R code Exa 3.3 system of homogeneous equations

```
 1  #Example 3.3 , section  1.3 , chapter  1 ,page  24
 2  #package  used  matlib  v0.9.1
 3  #install  package  using  command:  install.packages("
        matlib ")
 4  #Github  reposiory  of  matlib  : https ://github.com/
        friendly / matlib
 5
 6  #installation  and  loading  library
 7  #install.packages(" matlib ")
 8  library(" matlib ")
 9
10  #program
11  A <- matrix(c(3,2,4,-1),ncol = 2)
12  b <- c(3,1)
13  c <- c(0,0)
14  Ab <- cbind(A,b) # linear  system
15  Ac <- cbind(A,c) #system  of  homogeneous  equations
16  #reduction  of  original  system
17  Ab <- rowadd(Ab,1,2,-2/3)
18  #reduction  of  homogeneous  system
19  Ac <- rowadd(Ac,1,2,-2/3)
```

```
20  #comparing both
21  Ab
22  Ac
23  #Obviously the two reductions go in the same way.
24  #We can study how to reduce a linear systems by
        instead studying how to reduce the associated
        homogeneous system.
```

R code Exa 1.4 Solving Linear Systems

```
1   #Example 1.4,Section I. Solving Linear Systems
2   #page 3
3   #package used: matlib
4   #installation run command : install.packages("matlib
        ")
5   #package repo : https://github.com/friendly/matlib
6
7   #installing and loading library
8   #install.packages("matlib")
9   library("matlib")
10
11  #program
12  A <- matrix(c(0,1,1/3,0,5,2,3,-2,0),ncol = 3,nrow =
        3)
13  b <- c(9,2,3)
14  Ab <- cbind(A,b)
15  Ab <- rowswap(Ab,1,3)
16  Ab <- rowmult(Ab,1,3)
17  Ab <- rowmult(Ab,1,-1)
18  Ab <- rowadd(Ab,1,2,1)
19  Ab
20  #from Ab (X3=3)(since drom row3,3*X3=9)
21  #from Ab row2 -1*X2-2*x3=-7,so X2=1
22  #from Ab row1 X1+6*x2=9,so X1 = 3
```

R code Exa 2.5 finding the type of solution set of a system of linear equation

```
1  #Example 2.5, section 1.2, page 15
2  #package used matlib v0.9.1
3  #install package using command: install.packages("
     matlib")
4  #Github reposiory of matlib :https://github.com/
     friendly/matlib
5
6  #installation and loading library
7  #install.packages("matlib")
8  library("matlib")
9
10 #program
11 A <- matrix(c(1,2,3,2,0,2,0,1,1,0,0,-1),ncol = 4)
12 b <- c(1,2,4)
13 Ab <- cbind(A,b) #augmented matrix
14 Ab <- rowadd(Ab,1,2,-2)
15 Ab <- rowadd(Ab,1,3,-3)
16 Ab <- rowadd(Ab,2,3,-1)
17 Ab
18 #The leading variables are x, y, and w. The variable
     z is free.
19 #although there are infinitely many solutions, the
     value of w doesn't vary but is constant w = −1.
```

R code Exa 3.5 gauss method to reduce system of linear equations

```
1  #Example 3.5, chapter 1, section 1.3, page 25
2  #package used matlib v0.9.1
```

```
 3  #install package using command: install.packages("
       matlib")
 4  #Github reposiory of matlib :https://github.com/
       friendly/matlib
 5
 6  #installation and loading library
 7  #install.packages("matlib")
 8  library("matlib")
 9
10  #program
11  A <- matrix(c(7,8,0,0,0,1,1,3,-7,-5,-3,-6,0,-2,0,-1)
       ,ncol=4)
12  b <- c(0,0,0,0)
13  Ab <- cbind(A,b)
14  Ab <- rowadd(Ab,1,2,-8/7)
15  Ab <- rowadd(Ab,2,3,-1)
16  Ab <- rowadd(Ab,2,4,-3)
17  Ab <- rowadd(Ab,3,4,-5/2)
18  Ab
```

R code Exa 1.7 solution of system of linear equations

```
 1  #Example 1.7,page nO:5
 2  #package used: matlib
 3  #installation run command : install.packages("matlib
       ")
 4  #package repo : https://github.com/friendly/matlib
 5
 6  #installing and loading library
 7  #install.packages("matlib")
 8  library("matlib")
 9
10  #program
11  A <- matrix(c(1,2,1,1,-1,-2,0,3,-1),ncol = 3)
12  b <- c(0,3,3)
```

```
13  Ab <- cbind(A,b)
14  Ab <- rowadd(Ab,1,2,-2)
15  Ab <- rowadd(Ab,1,3,-1)
16  Ab <- rowadd(Ab,2,3,-1)
17  Ab
```
18 #from row3 : $-4z=0$, so $z=0$
19 #from row2 : $-3y+3z=3$, so $y=-1$
20 #from row1 : $x+y=0$, so $x=1$

R code Exa 2.7 solution of system of linear equations

1 #Example 2.7, chapter one, section 1.2, page 16
2 #package used: matlib
3 #installation run command : install.packages("matlib")
4 #package repo : https://github.com/friendly/matlib
5
6 #installing and loading library
7 #install.packages("matlib")
8 `library("matlib")`
9
10 #program
```
11  A <- matrix(c(1,0,1,2,1,0,0,-1,2),ncol = 3)
12  b <- c(4,0,4)
```
13 #creating augmented matrix
```
14  Ab <- cbind(A,b)
15  Ab
```
16 #applying reduction techniques
```
17  Ab <- rowadd(Ab,1,3,-1)
18  Ab <- rowadd(Ab,2,3,2)
19  Ab
```
20 #second row: $y-z=0$
21 #first row: $x+2*y=4$
22 #so solution set is : $\{(4-2*z = 0)\}$

R code Exa 1.8 solution of system of linear equations

```
1  #Example 1.8, page 5
2  #package used : matlib
3  #installation run command : install.packages("matlib
       ")
4  #package repo : https://github.com/friendly/matlib
5
6  #installing and loading library
7  #install.packages("matlib")
8  library("matlib")
9
10 #program
11 A <- matrix(c(40,-50,15,25),ncol = 2)
12 b <- c(100,50)
13 Ab <- cbind(A,b)
14 Ab <- rowadd(Ab,1,2,1.25)
15 Ab
16 #from row2 : 43.75c = 175,so c= 4
17 #from row1 : 40h+15c=100,so h=1
```

R code Exa 1.9 gauss method to reduce system of linear equations

```
1  #Example 1.9, chapter1 linear systems page 6
2  #Example showing gauss method to reduce given system
        of linear equations
3  #package used matlib v0.9.1
4  #install package using command: install.packages("
       matlib")
5  #Github reposiory of matlib :https://github.com/
       friendly/matlib
6
```

```
 7  #installing  and  loading  library
 8  #install.packages("matlib")
 9  library("matlib")
10
11  #program
12  A <- matrix(c(1,2,3,1,4,6,1,-3,-5),ncol = 3)
13  b <-c(9,1,0)
14  Ab <- cbind(A,b)
15  Ab <-rowadd(Ab,1,2,-2)
16  Ab <-rowadd(Ab,1,3,-3)
17  Ab <-rowadd(Ab,2,3,-(3/2))
18  Ab
19  #from  Ab row3:  z=3
20  #from  Ab row2:  y=-1
21  #from  Ab row1:  x=7
```

R code Exa 1.11 gauss method to reduce system of linear equations

```
 1  #Example  1.11,page  6
 2  #package  used:  matlib
 3  #installation  run  command :  install.packages("matlib
        ")
 4  #package  repo  :  https://github.com/friendly/matlib
 5
 6  #installation  and  loading  library
 7  #install.packages("matlib")
 8  library("matlib")
 9
10  #program
11  A <- matrix(c(1,2,0,0,-1,-2,1,0,0,1,0,2,0,2,1,1),
        ncol=4)
12  b <- c(0,4,0,5)
13  Ab <- cbind(A,b)
14  Ab <- rowadd(Ab,1,2,-1)
15  Ab <- rowswap(Ab,2,3)
```

```
16  Ab <- rowadd(Ab,3,4,-2)
17  Ab
18  #Back−substitution  gives  w = 1,  z = 2  ,  y = −1,  and
        x = −1.
```

R code Exa 1.12 gaussian elimination technique

```
1  #Example  1.12  Section  I.  Solving  Linear  Systems
        page7
2  #package  used  matlib  v0.9.1
3  #install  package  using  command:  install.packages("
        matlib")
4  #Github  reposiory  of  matlib  :https://github.com/
        friendly/matlib
5
6  #installing  and  loading  library
7  #install.packages("matlib")
8  library("matlib")
9
10  #program
11  A <- matrix(c(1,2,2,3,1,2),ncol = 2)
12  b <- c(1,-3,-2)
13  #for  this  problem  we  cannot  use  normal  method
        because  the  number  of  equations  is  more  than
        number  of  variables
14  #so we use  gaussian  elimination  technique.
15  gaussianElimination(A, b, tol = sqrt(.Machine$double
        .eps), verbose = FALSE,
16                          latex = FALSE, fractions = FALSE
                            )
17  #  result  shows  that  one  of  the  equations  is
        redundant,  here  x=−2,y=1
```

R code Exa 2.12 vector addition

```
1  #Example 2.12, section 1.2, chapter 1, page 18.
2  #vector addition
3  a <- c(2,3,1)
4  b <- c(3,-1,4)
5  a+b
6  c <- c(1,4,-1,-3)
7  7*c
```

R code Exa 1.13 solution of system of linear equations

```
1   #Example 1.13, page 7
2   #package used matlib v0.9.1
3   #install package using command: install.packages("
       matlib")
4   #Github reposiory of matlib :https://github.com/
       friendly/matlib
5
6   #installation and loading library
7   #install.packages("matlib")
8   library("matlib")
9
10  #program
11  A <- matrix(c(1,2,2,3,1,2),ncol = 2)
12  b <- c(1,-3,0)
13  Ab <- cbind(A,b)
14  Ab <- rowadd(Ab,1,2,-2)
15  Ab <- rowadd(Ab,1,3,-2)
16  Ab <- rowadd(Ab,2,3,-(4/5))
17  Ab
18  #the echelon form shows that the system is
       inconsistent, hence the solution set is empty
```

Chapter 2

Vector Spaces

R code Exa 3.2 Linear Independence

```
1  #Example 3.2, Section III. Basis and Dimension, page
       127
2  #package used matlib v0.9.1
3  #install package using command: install.packages("
       matlib")
4  #Github reposiory of matlib :https://github.com/
       friendly/matlib
5
6  #installation and loading library
7  #install.packages("matlib")
8  library("matlib")
9
10 #program
11 A <- matrix(c(2,4,3,6),ncol = 2)
12 #Rowspace(A) is this subspace of the space of two-
       component row vectors
13 #{c1.(2 3) + c2.(4 6) | c1,c2 belongs to R}
14 #simplifying A
15 A <- rowadd(A,1,2,-2)
16 A
17 #From the simplified matrix,the second row vector is
```

```
      linearly  dependent  on  the  first  and  so  we  can
      simplify  the  above  description  to
18  #{ c . ( 2   3 ) | c   belongs  to  R}
```

R code Exa 3.5 Basis for the column space of the given matrix

```
 1  #Example  3.5 , Section  III .  Basis  and  Dimension , page
       128
 2  #package  used  matlib  v0 . 9 . 1
 3  #install  package  using  command:  install . packages ("
       matlib ")
 4  #Github  reposiory  of  matlib  : https :// github . com/
       friendly / matlib
 5
 6  #installation  and  loading  library
 7  #install . packages (" matlib ")
 8  library (" matlib ")
 9
10  #program
11  A  <- matrix ( c ( 1 , 1 , 2 , 3 , 4 , 0 , 1 , 1 , 5 ) , ncol  = 3)
12  #From  any  matrix ,  we  can  produce  a  basis  for  the  row
        space  by
13  #performing  Gauss 's  Method  and  taking  the  nonzero
       rows  of  the  resulting  echelon  form  matrix
14  #simplifying  to  echelon  form
15  A  <- rowadd (A , 1 , 2 , -1)
16  A  <- rowadd (A , 1 , 3 , -2)
17  A  <- rowadd (A , 2 , 3 , 6)
18  A
19  #on  simplification : produces  the  basis  h(1  3  1);  (0  1
        0);  (0  0  3) i  for  the  row  space .  This  is  a  basis
20  #for  the  row  space  of  both  the  starting  and  ending
       matrices ,  since  the  two  row  spaces  are  equal .
```

R code Exa 1.6 Linear Independence

```
1  #Chapter 2.
2  #Section II. Linear Independence
3  #Example 1.6, page 103
4  #package used matlib v0.9.1
5  #install package using command: install.packages("
      matlib")
6  #Github reposiory of matlib :https://github.com/
      friendly/matlib
7
8  #installation and loading library
9  #install.packages("matlib")
10 library("matlib")
11
12 #program
13 #c1(40 15) + c2(-50 25) = (0 0),check if {(40 15)
      ,(-50 25)} is linearly independent
14 A <- matrix(c(40,15,-50,25),ncol = 2)
15 b <- c(0,0)
16 Ab <- cbind(A,b)
17 Ab <- rowadd(Ab,1,2,-15/40)
18 Ab
19 #from Ab,Both c1 and c2 are zero. So the only linear
       relationship between the two given row vectors
      is the trivial relationship.
```

R code Exa 3.7 Basis for the column space of the given matrix

```
1  #Example 3.7,Section III. Basis and Dimension ,page
      129
2  #package used matlib v0.9.1
```

19

```
3  #install package using command: install.packages("
      matlib")
4  #Github reposiory of matlib :https://github.com/
      friendly/matlib
5
6  #installation and loading library
7  #install.packages("matlib")
8  library("matlib")
9
10 #program
11 A <- matrix(c(1,2,0,4,3,3,1,0,7,8,2,4),ncol = 3)
12 #to get a basis for the column space, temporarily
      turn the columns into rows and reduce.
13 A <- t(A)
14 A <- rowadd(A,1,2,-3)
15 A <- rowadd(A,1,3,-7)
16 A <- rowadd(A,2,3,-2)
17 #Now turn the rows back to columns
18 A <- t(A)
19 A
20 #The result is a basis for the column space of the
      given matrix.
```

R code Exa 1.9 Linear Independence

```
1  #Chapter 2.
2  #Section II. Linear Independence
3  #Example 1.9,page 104
4  v1 <- c(3,4,5)
5  v2 <- c(2,9,2)
6  v3 <- c(4,18,4)
7  (0*v1)+(2*v2)-1*v3
8  #the set S = {v1,v2,v3} is linearly dependent
      because this is a relationship where not all of
      the scalars are zero
```

R code Exa 3.9 Finding a basis for the given span

```
1  #Example 3.5, Section III. Basis and Dimension, page
       130
2  #package used matlib v0.9.1
3  #install package using command: install.packages("
       matlib")
4  #Github reposiory of matlib :https://github.com/
       friendly/matlib
5
6  #installation and loading library
7  #install.packages("matlib")
8  library("matlib")
9
10 #program
11 #To get a basis for the span of {x^2 + x^4,2x^2 + 3x
       ^4,-x^2 - 3x^4 } in the space row4
12 A <- matrix(c(0,0,0,0,0,0,1,2,-1,0,0,0,1,3,-3),ncol
       = 5,nrow = 3)
13 #applying gauss method
14 A <- rowadd(A,1,2,-2)
15 A <- rowadd(A,1,3,1)
16 A <- rowadd(A,2,3,2)
17 A
18 #we get the basis (x^2 + x^4,x^4)
```

R code Exa 3.10 finding basis from reduced echelon form

```
1  #Example 3.10, Section III. Basis and Dimension, page
       131
2  #package used pracma
```

```
3  #install package using command: install.packages("
      pracma")
4
5  #installation and loading library
6  #install.packages("pracma")
7  library("pracma")
8
9  #program
10 A <- matrix(c(1,2,1,3,6,3,1,3,1,6,16,6),ncol = 4,
      nrow = 3)
11 #finding row reduced echelon form("using gauss-
      jordan reduction")
12 rref(A)
13 #Thus, for a reduced echelon form matrix we can find
       bases for the row and column spaces in
      essentially the same way, by taking the parts of
      the matrix, the rows or columns, containing the
      leading entries.
```

R code Exa 1.16 finding the coordinates of that vector with respect to the basis

```
1  #Example 1.16, Section III. Basis and Dimension, page
      118
2  #package used matlib v0.9.1
3  #install package using command: install.packages("
      matlib")
4  #Github reposiory of matlib :https://github.com/
      friendly/matlib
5
6  #installation and loading library
7  #install.packages("matlib")
8  library("matlib")
9
10 #program
```

```r
11  v <- c(3,2)
12  A <- matrix(c(1,1,0,2),ncol = 2)
13  Av <- cbind(A,v)
14  Av <- echelon(A,v,reduced = TRUE)
15  Av
16  #from Av, c1 = 3 and c2 = −1=2.
```

Chapter 3

Maps Between Spaces

R code Exa 1.1 correspondence between vectors

```
1  #Example 1.1, Section I. page 166
2  a <- c(1,2)
3  a <-t(a) #a becomes row vector
4  b <-t(a) #b becomes column vector
5  c <-c(3,4)
6  c <-t(c)
7  d <-t(c)
8  a+c
9  b+d
10 #these corresponding vectors add to corresponding
      totals
11 5*a
12 5*b
13 #correspondence respecting scalar multiplication
```

R code Exa 4.1.1 Matrix Operations

```
1  #Example1.1, section IV:Matrix Operations, chapter3,
      page 224
```

```
2  #Let f : V -> W be a linear function represented
      with respect to some bases by this matrix.
3  f <- matrix(c(1,1,0,1),ncol = 2)
4  #find the map that is the scalar multiple 5f: V -> W
      .
5  5*f
6  #Changing from the map f to the map 5f has the
      effect on the representation of the output vector
       of multiplying each entry by 5.
7  #Therefore, going from the matrix representing f to
      the one representing 5f means multiplying all the
       matrix entries by 5.
```

R code Exa 4.1.2 Matrix Operations

```
1  #Example1.2, section IV: Matrix Operations, chapter3,
      page 225
2  f <- matrix(c(1,2,3,0),ncol = 2)
3  g <- matrix(c(-2,2,-1,4),ncol = 2)
4  #two linear maps with the same domain and codomain f
      ; g: R^2 -> R^2 are represented with respect to
      bases B and D by these matrices.
5  #find effect of f + g on the map
6  f+g
7  #if f does v -> u and g does v -> w then f + g is
      the function whose action is v -> u + w.
```

R code Exa 3.1.11 Computing linear maps

```
1  #Example1.11, section III: Computing linear maps,
      chapter3, page 211
2  a <- matrix(c(1,2,0,0,-1,3),ncol=3)
3  b <- c(2,-1,1)
```

```
 4  a %*% b
 5  #the above result can also be obtained by the method
 6  x <- matrix(c(1,2),ncol = 1)
 7  y <- matrix(c(0,0),ncol = 1)
 8  z <- matrix(c(-1,3),ncol = 1)
 9  #splitting the matrix a into component columns and
       now multiplying
10  2*x-1*y+1*z
11  #we see that both methods are equal.
```

R code Exa 4.2.4 Matrix Operations

```
 1  #Example2.4,section IV:Matrix Operations,chapter3,
       page 229
 2  A <- matrix(c(2,4,8,0,6,2),ncol = 2)
 3  B <- matrix(c(1,5,3,7),ncol = 2)
 4  A %*% B
```

R code Exa 4.2.5 Matrix Operations

```
 1  #Example2.5,section IV:Matrix Operations,chapter3,
       page 229
 2  A <- matrix(c(1,3,2,4),ncol = 2)
 3  B <- matrix(c(-1,2,0,-2),ncol=2)
 4  A %*% B
```

R code Exa 4.2.6 Matrix Operations

```
 1  #Example2.6,section IV:Matrix Operations,chapter3,
       page 230
```

```
2  A <- matrix(c(1,0,1,1,1,0),ncol=2)
3  B <- matrix(c(4,5,6,7,8,9,2,3),ncol = 4)
4  A %*% B
```

R code Exa 4.3.15 Mechanics of Matrix Multiplication

```
1  #Example3.15, section IV.3:Mechanics of Matrix
        Multiplication , chapter3 , page 241
2  #A permutation matrix is square and is all 0's
        except for a single 1 in each row and column.
3  A <- matrix(c(0,1,0,0,0,1,1,0,0),ncol = 3) #
        permutation matrix
4  B <- matrix(c(1,4,7,2,5,8,3,6,9),ncol = 3)
5  #From the left these matrices permute rows.
6  A %*% B
7  #From the right they permute columns.
8  B %*% A
```

R code Exa 4.4.9 Inverses

```
1  #Example4.9, section IV.4:Inverses , chapter3 , page 251
2  #package used matlib v0.9.1
3  #install package using command: install.packages("
        matlib")
4  #Github reposiory of matlib :https://github.com/
        friendly/matlib
5
6  #installation and loading library
7  #install.packages("matlib")
8  library("matlib")
9
10 #program
11 # Augmented matrix
```

```
12  A <- matrix(c(0,1,1,3,0,-1,-1,1,0),ncol = 3)
13  B <- matrix(c(1,0,0,0,1,0,0,0,1),ncol = 3)
14  AB <- cbind(A,B)
15  echelon(A,B)
```

R code Exa 2.5 Computing linear maps

```
1  #Example2.5, section III:Computing linear maps,
       chapter3 , page 218
2  #package used pracma
3  #install package using command: install.packages("
       pracma")
4
5  #installation and loading library
6  #install.packages("pracma")
7  library("pracma")
8
9  #program
10 A <- matrix(c(1,1,0,0,2,2,0,0,2,1,3,2),ncol = 3)
11 Rank(A)
12 #Any map represented by above matrix must have three
       -dimensional domain and a four-dimensional
       codomain.
13 #Since the rank of this matrix is found to be 2 by
       above code;
14 #Any map represented by this matrix has a two-
       dimensional range space.
```

R code Exa 1.9 automorphism

```
1  #Example 1.9, Section I. page 170
2  #space P5 of polynomials of degree 5 or less and the
       map f that sends a polynomial p(x) to p(x - 1).
```

```
3 #under this map x^2 ->(x-1)^2 = x^2-2x+1 and x^3+2x
     -> (x-1)^3+2(x-1) = x^3-3x^2+5x-3.
4 curve(x^2,from = -1000,to=1000)
5 curve((x-1)^2,from = -1000,to=1000)
6 curve(x^3,from = -1000,to=1000)
7 curve((x-1)^3,from = -1000,to=1000)
8 #from these plots we can say that this map is an
     automorphism of this space.
```

R code Exa 2.10 Computing linear maps

```
1 #Example2.10,section III:Computing linear maps,
     chapter3,page 219
2 #package used pracma
3 #install package using command: install.packages("
     pracma")
4
5 #installation and loading library
6 #install.packages("pracma")
7 library("pracma")
8
9 #program
10 A <- matrix(c(1,0,2,3),ncol = 2)
11 Rank(A)
12 #Any map from R^2 to P1 represented with respect to
     any pair of bases by the above matrix;
13 #is nonsingular because this matrix has rank two.
```

R code Exa 1.11 value of h on the basis vectors

```
1 #Example 1.11,Section II. page 186
2 #given map: h(1,0)=(-1,1) and h(0,1)=(-4,4)
3 #h(3,-2)=h(3*(1,0)-2*(0,1))=3*h(1,0)-2*h(0,1)
```

29

```
4  a <- c(-1,1) # h(1,0)
5  b <- c(-4,4) # h(0,1)
6  3*a-2*b
7  #the value of h on this argument is a direct
       consequence of the value of h on the basis
       vectors.
```

R code Exa 2.11 Computing linear maps

```
1  #Example2.11, section III: Computing linear maps,
       chapter3, page 220
2  A <- matrix(c(1,3,2,6),ncol=2)
3  det(A)
4  #map g: V -> W represented by above matrix is
       singular because this matrix is singular.
```

Chapter 4

Determinants

R code Exa 3.1 Determinants

```
1 #Example3.1 , chapter4 , section1 , page 329
2 X <- matrix ( c (4 , -2 ,2 ,6) , ncol = 2)
3 #scalars come out of each row separately , not from
      the entire matrix at once .
4 #so X will become
5 A <- matrix ( c (2 , -1 ,1 ,3) , ncol = 2)
6 #with scalars 2 & 2 coming out of rows 1 and 2
      respectively
7 c <- det ( X )
8 d <- 4* det ( A )
9 all . equal ( c , d )
```

R code Exa 1.4 Laplace Formula to find determinant

```
1 #Example 1.4 , chapter 4 , Section III . Laplace 's
      Formula , page 354
2 #package used matlib v0 .9.1 and pracma
3 #Github reposiory of matlib : https :// github . com /
      friendly / matlib
```

31

```
 4
 5  #installation and loading library
 6  #install.packages("matlib")
 7  library("matlib")
 8
 9  #program
10  T <- matrix(c(1,4,7,2,5,8,3,6,9),ncol = 3)
11  T12 <- cofactor(T,1,2)
12  T22 <- cofactor(T,2,2)
13  T12
14  T22
```

R code Exa 3.4 Determinants

```
 1  #Example3.4,chapter 4,page 330
 2  A <- matrix(c(2,4,1,3),ncol = 2)
 3  #using multinlinearity property to break up the
        matrix
 4  a <- matrix(c(2,4,0,0),ncol = 2)
 5  b <- matrix(c(2,0,0,3),ncol = 2)
 6  c <- matrix(c(0,4,1,0),ncol = 2)
 7  d <- matrix(c(0,0,1,3),ncol = 2)
 8  #verifying the property
 9  x <- det(A)
10  y <- det(a)+det(b)+det(c)+det(d)
11  all.equal(x,y)
```

R code Exa 2.5 Determinants

```
 1  #Example 2.5,chapter4,Section I. Definition,page 325
 2  #package used matlib v0.9.1
 3  #install package using command: install.packages("
        matlib")
```

```
 4 #Github reposiory of matlib :https://github.com/
      friendly/matlib
 5
 6 #installation and loading library
 7 #install.packages("matlib")
 8 library("matlib")
 9
10 #program
11 A <- matrix(c(2,4,0,2,4,-3,6,3,5),ncol=3)
12 b <- det(A)
13 #determinant by normal ,ethod
14 A <- rowadd(A,1,2,-2)
15 A <- rowswap(A,2,3)
16 A
17 #reducing with gaussian reduction now multiplying
      the diagonal terms,keeping in mind the sign
      change due to row swap to find determinant.
18 c <- -1*A[1,1]*A[2,2]*A[3,3]
19 all.equal(b,c)
20 #so the determinant by both techniques are the same
```

R code Exa 1.6 Laplace Formula to find determinant

```
 1 #Example 1.6,chapter 4,Section III. Laplace's
      Formula,page 355\
 2 #package used matlib v0.9.1 and pracma
 3 #Github reposiory of matlib :https://github.com/
      friendly/matlib
 4
 5 #installation and loading library
 6 #install.packages("matlib")
 7 library("matlib")
 8
 9 #program
10 t <- matrix(c(1,4,7,2,5,8,3,6,9),ncol = 3)
```

```
11  x <- det(t)
12  #computing the determinant by expanding along the
        first row,
13  y <- t[1,1]*cofactor(t,1,1)+t[1,2]*cofactor(t,1,2)+t
        [1,3]*cofactor(t,1,3)
14  #computing the determinant by expanding along the
        second row,
15  z <- t[2,1]*cofactor(t,2,1)+t[2,2]*cofactor(t,2,2)+t
        [2,3]*cofactor(t,2,3)
16  all.equal(x,y)
17  all.equal(x,z)
```

R code Exa 2.6 Determinants

```
1   #Example 2.6, chapter4, Section I. Definition, page 326
2   #package used matlib v0.9.1 and pracma
3   #Github reposiory of matlib :https://github.com/
        friendly/matlib
4
5   #installation and loading library
6   #install.packages("matlib")
7   library("matlib")
8
9   #program
10  A <- matrix(c(1,0,0,0,0,1,0,1,1,1,0,0,3,4,5,1),ncol
        = 4)
11  A <- rowadd(A,2,4,-1)
12  A <- rowswap(A,3,4)
13  A
14  #multiplying diagonal terms and multiplying it with
        -1 to compensate for rowswap
15  -1*A[1,1]*A[2,2]*A[3,3]*A[4,4]
```

R code Exa 1.7 Laplace Formula to find determinant

```
1  #Example 1.7,chapter 4,Section III. Laplace's
        Formula,page 355
2  #package used matlib v0.9.1 and pracma
3  #Github reposiory of matlib :https://github.com/
        friendly/matlib
4
5  #installation and loading library
6  #install.packages("matlib")
7  library("matlib")
8
9  #program
10 t <- matrix(c(1,2,3,5,1,-1,0,1,0),ncol = 3)
11 #computing the determinant by expanding along the
        third column.
12 y <- t[1,3]*cofactor(t,1,3)+t[2,3]*cofactor(t,2,3)+t
        [3,3]*cofactor(t,3,3)
13 y
```

R code Exa 1.10 Laplace Formula to find inverse

```
1  #Example 1.10,chapter 4,Section III. Laplace's
        Formula,page 356
2  #package used matlib v0.9.1 and pracma
3  #Github reposiory of matlib :https://github.com/
        friendly/matlib
4
5  #installation and loading library
6  #install.packages("matlib")
7  library("matlib")
8
9  #program
10 t <- matrix(c(1,2,1,0,1,0,4,-1,1),ncol = 3)
11 #finding inverse: if T has an inverse, if |T| != 0,
```

```
         then  T^-1 = (1/|T|)    adj(T)
12  a  <-  adjoint(t)
13  b  <-  1/det(t)
14  i  <-  b*a
15  i
```

Chapter 5

Similarity

R code Exa 2.2 Complex Vector Spaces

```
1  #Example 2.2,chapter 5,Section I. Complex Vector
       Spaces,page 387
2  a <- complex(real = 1,imaginary = 1)
3  b <- complex(real = 0,imaginary = 1)
4  c <- complex(real = 2,imaginary = -0)
5  d <- complex(real = -2,imaginary = 3)
6  e <- complex(real = 1,imaginary = 0)
7  f <- complex(real = 0,imaginary = 3)
8  g <- complex(real = 1,imaginary = -0)
9  h <- complex(real = 0,imaginary = -1)
10 A <- matrix(c(a,b,c,d),ncol = 2)
11 B <- matrix(c(e,f,g,h),ncol = 2)
12 A %*% B
```

R code Exa 2.2.2 Checking if matrix is diagonalizable

```
1  #Example 2.2,chapter 5,section II.2
       Diagonalizability ,page 393
```

```
2 #matrix: T is diagonalizable if there is a
     nonsingular P such that PTP-1 is diagonal.
3 T <- matrix(c(4,1,-2,1),ncol = 2)
4 P <- matrix(c(-1,1,2,-1),ncol = 2)
5 #checking whether diagonizable
6 P %*% T %*% Inverse(P) # diagonal matrix
7 # so matrix P is diagonizable
```

R code Exa 1.3 Similar Matrix

```
1 #Example 1.3,chapter 5,Section II. Similarity ,page
     390
2 #package used matlib v0.9.1
3 #Github reposiory of matlib :https://github.com/
     friendly/matlib
4
5 #installation and loading library
6 #install.packages("matlib")
7 library("matlib")
8 P <- matrix(c(2,1,1,1),ncol = 2)
9 T <- matrix(c(2,1,-3,-1),ncol = 2)
10 #finding similar matrix of T
11 T1 <- P %*% T %*% Inverse(P)
12 T1
```

R code Exa 2.4 finding null spaces

```
1 #Example 2.4,chapter 5,section IV.2,page 432
2 #package used pracma v1.9.9
3 #installing and loading library
4 #install.packages("pracma")
5 library("pracma")
6 T <- matrix(c(2,1,-1,4),ncol=2)
```

```
7  a <- eigen(T)
8  a$values
9  #so T has only the single eigenvalue 3.
10 I <- matrix(c(1,0,0,1),ncol = 2)
11 T-(3*I)
12 # so for this,the only eigenvalue is 0 and T -3I is
      nilpotent.
13 #to ease this computation we find nulspaces
14 x <- nullspace(T)
15 x
```

R code Exa 3.6 Eigenvalues and Eigenvectors

```
1  #Example3.6,chapter 5,section II.3 Eigenvalues and
      Eigenvectors,page 398
2  T <- matrix(c(2,0,0,0),ncol = 2)
3  a <-eigen(T)
4  a$values
5  a$vectors
```

R code Exa 2.10 Nilpotent matrix

```
1  #Example 2.10,chapter 5,scetion III.2,page 414
2  #package used matlib v0.9.1
3  #Github reposiory of matlib :https://github.com/
      friendly/matlib
4
5  #installation and loading library
6  #install.packages("matlib")
7  library("matlib")
8  N <- matrix(c(0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0),ncol
      =4)
```

```
 9  P <- matrix(c(1,0,1,0,0,2,1,0,1,1,1,0,0,0,0,1),ncol
       = 4)
10  A <- P %*% N %*% Inverse(P)
11  A
12  #The new matrix,A is nilpotent; its fourth power is
        the zero matrix.
13  x <- P %*% N^4 %*% Inverse(P)
14  #since (PNP^-1)^4 = P * N^4 *P^-1
15  y <- det(x)
16  all.equal(y,0)
```

R code Exa 2.17 Nilpotence index

```
 1  #Example 2.17,chapter 5,scetion III.2,page 419
 2  #package used matlib v0.9.1
 3  #Github reposiory of matlib :https://github.com/
       friendly/matlib
 4
 5  #installation and loading library
 6  #install.packages("matlib")
 7  library("matlib")
 8  M <- matrix(c(1,1,-1,-1),ncol = 2)
 9  #finding nilpotent index
10  A <- matrix(c(0,0,0,0),ncol = 2)
11  count <- 1
12  Y <- M
13  repeat{
14    Y <- Y %*% M
15    if (all.equal(Y,A)){
16      print(count+1)
17      break()
18    }
19   count= count+1
20  }
```

R code Exa 2.18 nilpotence index

```
1  #Example 2.18,chapter 5,scetion III.2,page 420
2  #package used matlib v0.9.1
3  #Github reposiory of matlib :https://github.com/
       friendly/matlib
4
5  #installation and loading library
6  #install.packages("matlib")
7  library("matlib")
8  M <- matrix(c
       (0,1,-1,0,1,0,0,1,1,0,0,0,1,0,-1,0,0,-1,0,1,0,0,1,0,-1)
       ,ncol = 5)
9  #finding nilpotent index
10 A <- matrix(c(0),ncol = 5,nrow = 5)
11 count <- 1
12 Y <- M
13 repeat{
14    Y <- Y %*% M
15    if (all.equal(Y,A)== TRUE){
16       print(count+1)
17       break()
18    }
19    count=count+1
20 }
```

R code Exa 3.19 Eigenvalues and Eigenvectors

```
1  #Example3.19,chapter 5,section II.3 Eigenvalues and
       Eigenvectors,page 404
2  T <- matrix(c(2,0,-4,-2,1,8,2,1,3),ncol = 3)
3  a <- eigen(T)
```

41

4 a$values