Scilab Textbook Companion for Linear Algebra and Its Applications by D. C. Lay[1]

Created by
Animesh Biyani
B.Tech (Pursuing)
Electrical Engineering
National Institute Of Technology, Karnataka
College Teacher
Prof. Murugan, NIT Suratkal
Cross-Checked by
Prashant Dave, IIT Bombay

May 16, 2016

[1]Funded by a grant from the National Mission on Education through ICT, http://spoken-tutorial.org/NMEICT-Intro. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website http://scilab.in

Book Description

Title: Linear Algebra and Its Applications

Author: D. C. Lay

Publisher: Addison Wesley

Edition: 3

Year: 2006

ISBN: 0-321-28066-0

Scilab 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)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

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

Contents

List of Scilab Codes

Chapter 1

LINEAR EQUATIONS IN LINEAR ALGEBRA

Scilab code Exa 1.1.1 Gaussian Elimination

```
disp('performing Gaussian elimination')
a=[1 5;-2 -7]
disp('the co-efficient matrix is:')
disp(a)
b=[7;-5]
c=[a b]
disp('the augmented matrix is:')
disp(c)
disp('R2=R2+2*R1')
c(2,:)=c(2,:)+2*c(1,:)
disp(c)
disp('R2=(1/3)*R2')
c(2,:)=(1/3)*c(2,:)
disp(c)
disp('R1=R1-5*R2')
c(1,:)=c(1,:)-5*c(2,:)
disp(c)
x1=c(1,3)/c(1,1)
x2=c(2,3)/c(2,2)
```

```
printf('the solution is:x1=%d x2=%d',x1,x2)
```

Scilab code Exa 1.1.7 Gaussian Elimination Singular case

```
disp('the augmented matrix is:')
a=[1 7 3 -4;0 1 -1 3;0 0 0 1;0 0 1 -2]
disp(a)
disp('interchange R3 and R4')
a([3,4],:)=a([4,3],:)
disp(a)
disp('from R4 we get 0=1')
disp('hence, no solution')
```

Scilab code Exa 1.1.13 Gaussian Elimination with row exchange

```
disp('the augmented matrix is')
a=[1 0 -3 8;2 2 9 7;0 1 5 -2]
disp(a)
disp('R2=R2-2*R1')
a(2,:)=a(2,:)-2*a(1,:)
disp(a)
disp('interchange R2 and R3')
a([2,3],:)=a([3,2],:)
disp(a)
disp('R3=R3-2*R2')
a(3,:)=a(3,:)-2*a(2,:)
disp(a)
disp('R3=(1/5)*R3')
a(3,:)=(1/5)*a(3,:)
disp(a)
disp('R2=R2-5*R3 and R1=R1+3*R3')
a(2,:)=a(2,:)-5*a(3,:)
a(1,:)=a(1,:)+3*a(3,:)
```

```
disp(a)
s=[a(1,4);a(2,4);a(3,4)]
disp('solution is')
disp(s)
```

Scilab code Exa 1.1.19 Condition for a solution to exist

```
disp('the augmented matrix for h=2')
a=[1 2 4;3 6 8]
disp(a)
disp('R2-2*R1')
a(2,:)=a(2,:)-3*a(1,:)
disp(a)
disp('from R3 we get 0=-4')
disp('hence, if h=2 no solution, else solution
    exists')
```

Scilab code Exa 1.1.25 Condition for a solution to exist

```
disp('the co-efficient matrix is:')
a=[1 -4 7;0 3 -5;-2 5 -9]
disp(a)
disp('let g,h,k be the constants on RHS')
disp('R3=R3+2*R1')
a(3,:)=a(3,:)+2*a(1,:)
disp(a)
disp('the constants on RHS are:g,h,k+2g')
disp('R3=R3+R2')
a(3,:)=a(3,:)+a(2,:)
disp(a)
disp('the constants on RHS are:g,h,k+2g+h')
disp('for solution to exist')
disp('from R3:k+2g+h=0')
```

Scilab code Exa 1.2.7 General solution of the system

```
disp('the augmented matrix is')
a=[1 3 4 7;3 9 7 6]
disp(a)
disp('R2=R2-3*R1')
a(2,:)=a(2,:)-3*a(1,:)
disp(a)
disp('(-1/5)*R2')
a(2,:)=(-1/5)*a(2,:)
disp(a)
disp('R1=R1-4*R2')
a(1,:)=a(1,:)-4*a(2,:)
disp('the row reduced form is:')
disp(a)
disp('corresponding equations are')
disp('x1+3*x2=-5 and x3=3')
disp('free variables:x2')
disp('general solution is:')
disp('x1=-5-3*x2, x2, x3=3')
```

Scilab code Exa 1.2.13 General solution of the system

```
disp('the augmented matrix is')
a=[1 -3 0 -1 0 -2;0 1 0 0 -4 1;0 0 0 1 9 4;0 0 0 0 0
    0]
disp(a)
disp('R1=R1+R3')
a(1,:)=a(1,:)+a(3,:)
disp(a)
disp('R1=R1+3*R2')
```

```
a(1,:)=a(1,:)+3*a(2,:)
disp(a)
disp('corresponding equations are:')
disp('x1-3*x5=5, x2-4*x5=1, x4+9*x5=4, and 0=0')
disp('free variables:x3, x5')
disp('general solution is:')
disp('x1=5+3*x5, x2=1+4*x5, x3, x4=4-9*x5, x5')
```

Scilab code Exa 1.2.34 Row reduced echelon form

```
disp('the augmented matrix is:')
a=[1 0 0 0 0 0 0;1 2 4 8 16 32 2.9;1 4 16 64 256
   1024 14.8;1 6 36 216 1296 7776 39.6;1 8 64 512
   4096 32768 74.3;1 10 10^2 10^3 10^4 10^5 119];
disp(a)
disp('performing row transformations')
for k=2:6
  a(k,:)=a(k,:)-a(1,:)
end
disp(a)
j=2;
for k=3:6
  a(k,:)=a(k,:)-j*a(2,:)
  j=j+1;
end
disp(a)
j=[0 0 0 3 6 10];
for k=4:6
  a(k,:)=a(k,:)-j(k)*a(3,:)
end
disp(a)
a(5,:)=a(5,:)-4*a(4,:)
a(6,:)=a(6,:)-10*a(4,:)
disp(a)
a(6,:)=a(6,:)-5*a(5,:)
```

```
disp(a)
a(6,:)=a(6,:)/a(6,6)
disp(a)
j=[0 32 960 4800 7680]
for k=1:5
  a(k,:)=a(k,:)-j(k)*a(6,:)
end
disp(a)
a(5,:)=a(5,:)/a(5,5)
j=[0 16 224 576]
for k=2:4
  a(k,:)=a(k,:)-j(k)*a(5,:)
end
a(4,:)=a(4,:)/48
a(2,:)=a(2,:)-8*a(4,:)
a(3,:)=a(3,:)-48*a(4,:)
a(3,:)=a(3,:)/8
a(2,:)=a(2,:)-4*a(3,:)
a(2,:)=a(2,:)/2
disp(a)
v=[a(1,7) a(2,7) a(3,7) a(4,7) a(5,7) a(6,7)]
p=poly(v,"t","coeff")
disp('p(t)=')
disp(p)
disp('p(7.5)=64.6 hundred lb')
```

Scilab code Exa 1.3.1 Linear combination of two vectors

```
u=[-1;2]
disp('u=')
disp(u)
v=[-3;-1]
disp('v=')
disp(v)
s=u-2*v
```

```
disp('u-2v=')
disp(s)
```

Scilab code Exa 1.3.11 Linear combination of vectors

```
disp('vectors a1 a2 a3 are:')
a1=[1 0 -2]
disp(a1')
a2=[-4 3 8]
disp(a2')
a3=[2 5 -4]
disp(a3')
disp('vector b=')
b=[3 -7 -3]
disp(b')
disp('the augmented matrix is:')
a=[1 -4 2 3;0 3 5 -7;-2 8 -4 -3]
disp(a)
a(3,:)=a(3,:)+2*a(1,:)
disp(a)
disp('from the entries of last row, the system is inconsistent')
disp('hence, b is not a linear combination of a1 a2 and a3')
```

Scilab code Exa 1.3.31 Application of Gaussian elimination

```
disp('1 gram at (0,1), 1 gram at (8,1) and 1 gram at (2,4)')
cm=(1/3)*(1*[0;1]+1*[8;1]+1*[2;4])
disp('centre of mass is at')
disp(cm)
disp('the new weight of the system=9 grams')
```

```
disp('new centre of mass is at')
s=[2;2]
disp(s)
disp('let w1,w2 and w3 be the weights added at (0,1)
    ,(8,1) and (2,4) respectively')
disp('hence, w1+w2+w3=6')
disp('using the formula for the centre of mass, we
    get')
disp('8*w2+2*w3=8 and w1+w2+4*w3=12')
a=[1 1 1 6;0 8 2 8;1 1 4 12]
disp('the augmented matrix is:')
disp(a)
disp('R3=R3-R1')
a(3,:)=a(3,:)-a(1,:)
disp(a)
disp('R3=(1/3)*R3')
a(3,:)=(1/3)*a(3,:)
disp(a)
disp('R2=R2-2*R3 and R1=R1-R3')
a(2,:)=a(2,:)-2*a(3,:)
a(1,:)=a(1,:)-a(3,:)
disp(a)
disp('R1=R1-(1/8)*R2')
a(1,:)=a(1,:)-(1/8)*a(2,:)
disp(a)
disp('R2=(1/8)*R2')
a(2,:)=(1/8)*a(2,:)
disp(a)
printf('Add %.1f grams at (0,1), %.1f grams at (8,1)
     and %d grams at (2,4)',a(1,4),a(2,4),a(3,4))
```

Scilab code Exa 1.4.7 Vectors as columns of a matrix

```
disp('the three vectors are:')
u=[4;-1;7;-4]
```

```
v=[-5;3;-5;1]
w=[7;-8;0;2]
disp(w,v,u)
disp('u v and w form the columns of A')
A=[u v w]
disp(A)
disp('the augmented matrix is:')
c=[A [6 -8 0 -7]']
disp(c)
```

Scilab code Exa 1.4.13 Span of vectors

```
disp('the augmented matrix is:')
a=[3 -5 0;-2 6 4;1 1 4]
disp(a)
disp('interchange R1 and R3')
a([1,3],:)=a([3,1],:)
disp(a)
disp('R2=R2+2*R1 and R3=R3-3*R1')
a(2,:)=a(2,:)+2*a(1,:)
a(3,:)=a(3,:)-3*a(1,:)
disp(a)
disp('R3=R3+R2')
a(3,:)=a(3,:)+a(2,:)
disp(a)
disp('from the entries of last row, the system is
   consistent')
disp('hence, u is in the plane spanned by the
   columns of a')
```

Scilab code Exa 1.5.1 Free and pivot variables

```
disp('the augmented matrix is:')
```

```
a=[2 -5 8 0;-2 -7 1 0;4 2 7 0]
disp(a)
disp('R2=R2+2*R1 and R3=R3-2*R1')
a(2,:)=a(2,:)+a(1,:)
a(3,:)=a(3,:)-2*a(1,:)
disp(a)
disp('R3=R3+R2')
a(3,:)=a(3,:)+a(2,:)
disp(a)
disp('only two columns have non zero pivots')
disp('hence, one column is a free column and
   therefore there exists a non trivial solution')
```

Scilab code Exa 1.5.7 General solution of the system

```
disp('the augmented matrix is:')
a=[1 3 -3 7 0;0 1 -4 5 0]
disp(a)
disp('R1=R1-3*R2')
a(1,:)=a(1,:)-3*a(2,:)
disp(a)
disp('basic variables:x1 x2')
disp('free variables:x3 x4')
disp('x1=-9*x3+8*x4')
disp('x2=4*x3-5*x4')
disp('hence, solution is')
disp('[-9*x3+8*x4 4*x3-5*x4 x3 x4]')
```

Scilab code Exa 1.5.11 General solution of the system

```
disp('the augmented matrix is')
a=[1 -4 -2 0 3 -5 0;0 0 1 0 0 -1 0;0 0 0 0 -1 4 0;0
   0 0 0 0 0 0]
```

```
disp(a)
disp('R1=R1-3*R3')
a(1,:)=a(1,:)-3*a(3,:)
disp(a)
disp('R1=R1+2*R2')
a(1,:)=a(1,:)+2*a(2,:)
disp(a)
disp('the free variables are:x2, x4 and x6')
disp('the basic variables are:x1, x3 and x5')
disp('the solution is:')
disp('[4*x2-5*x6  x2  x6  x4  4*x6  x6]')
```

Scilab code Exa 1.7.1 Linear independence of vectors

```
disp('given vectors u, v and w are')
u=[5 0 0]'
disp(u)
v=[7 2 -6]'
disp(v)
w=[9 4 -8]'
disp(w)
disp('the augmented matrix is')
a=[5 7 9 0;0 2 4 0;0 -6 -8 0]
disp(a)
disp('R3=R3+3*R2')
a(3,:)=a(3,:)+3*a(2,:)
disp(a)
disp('there are no free variables')
disp('hence, the homogeneous equation has only
    trivial solution and the vectors are linearly
    independent')
```

Scilab code Exa 1.7.7 Linear independence of vectors

```
disp('the augmented matrix is')
A=[1 -3 3 -2 0;-3 7 -1 2 0;-4 -5 7  5 0]
disp(A)
disp('since there are three rows, the maximum number
    of pivots can be 3')
disp('hence, at least one of the four variable must
   be free')
disp('so the equations have non trivial solution and
    the columns of A are linearly independent')
```

Chapter 2

MATRIX ALGEBRA

Scilab code Exa 2.1.1 Matrix operations

```
A=[2 0 -1;4 -5 2];
disp('matrix A:')
disp(A)
disp('-2A=')
disp(-2*A)
disp('matrix B')
B=[7 -5 1;1 -4 -3];
disp(B)
disp('B-2A=')
disp(B-2*A)
```

Scilab code Exa 2.2.1 Inverse of a matrix

```
disp('given matrix:')
a=[8 6;5 4];
disp(a)
disp('inverse of the matrix is:')
disp(inv(a))
```

Scilab code Exa 2.2.7 Inverse of a matrix

```
disp('the co-efficient matrix is:')
a=[1 2;5 12]
disp(a)
disp('inverse of the matrix is:')
disp(inv(a))
disp('solution is:')
b=[-1;3];
c=inv(a);
disp(c*b)
```

Scilab code Exa 2.3.1 Invertibility of a matrix

```
disp('the given matrix is:')
a=[5 7;-3 -6];
disp(a)
disp('the columns are lineraly independent')
disp('hence, by invertible matrix theorem')
disp('the matrix A is invertible')
```

Scilab code Exa 2.3.33 Invertible matrix theorem

```
disp('matrix A corresponding to transformation T is:
   ')
A=[-5 9;4 -7];
disp(A)
disp('determinant of A is:')
disp(det(A))
```

```
disp('since det(A) is not equal to zero')
disp('by IMT, A is invertible')
disp('hence, the inverse of A exists')
disp('inverse of A is:')
disp(inv(A))
```

Scilab code Exa 2.4.25 Inverse using matrix partition

```
disp('given matrix is:')
a=[1 2 0 0 0;3 5 0 0 0;0 0 2 0 0;0 0 0 7 8;0 0 0 5
   6];
disp(a)
disp('partitioning the matrix into 4 submatrices')
A11=[a(1,1:2);a(2,1:2)]
disp(A11,'A11=')
A22=[a(3,3:5);a(4,3:5);a(5,3:5)]
disp(A22,'A22=')
A12=zeros(2,3)
disp(A12,'A12=')
A21=zeros(3,2)
disp(A21,'A21=')
disp('partitioning A22 into 4 submatrices')
A221=[2]
disp(A221)
B=[A22(2,2:3);A22(3,2:3)]
disp(B,'B=')
disp(zeros(1,2))
disp(zeros(2,1))
disp('determinant of B=')
disp(det(B))
disp('Hence, B is invertible')
disp('inverse of B is')
disp(inv(B))
disp('determinant of inverse of  B is:')
disp(det(inv(B)))
```

```
disp('hence the invese of A22 is:')
c=[det(inv(B)) zeros(1,2);0 3 -4;0 -2.5 3.5];
disp(c)
```

Scilab code Exa 2.5.1 Application of LU decomposition

```
disp('the lower triangular matrix is:')
L=[1 0 0;-1 1 0;2 -5 1];
disp(L)
disp('the upper triangular matrix is:')
U=[3 -7 -2;0 -2 -1;0 0 -1];
disp(U)
disp('the RHS of the equations are')
b=[-7;5;2];
disp(b)
disp('combining matrices L and b')
c=[L b];
disp(c)
disp('performing row operations')
disp('R2=R2+R1')
c(2,:)=c(2,:)+c(1,:)
disp(c)
disp('R3=R3-2*R1')
c(3,:)=c(3,:)-2*c(1,:)
disp(c)
disp('R3=R3+5*R2')
c(3,:)=c(3,:)+5*c(2,:)
disp(c)
y=c(:,4)
disp(y,'y=')
disp('combining U and y')
d=[U y];
disp(d)
disp('performing row operations')
disp('R3=R3/-6')
```

```
d(3,:)=d(3,:)/(-1)
disp(d)
disp('R2=R2+R3 and R1=R1+2*R3')
d(2,:)=d(2,:)+d(3,:)
d(1,:)=d(1,:)+2*d(3,:)
disp(d)
disp('R1=R1-3.5*R2')
d(1,:)=d(1,:)-3.5*d(2,:)
disp(d)
disp('R1=R1/3 and R2=R2/-2')
d(1,:)=d(1,:)/3
d(2,:)=d(2,:)/(-2)
disp(d)
disp('the solution is:')
x=d(:,4)
disp(x,'x=')
```

Scilab code Exa 2.5.7 LU decomposition of a matrix

```
disp('given matrix is:')
a=[2 5;-3 -4]
d=a;
disp(a)
disp('performing row operations')
a(2,:)=a(2,:)-(a(2,1)/a(1,1))*a(1,:)
disp(a)
disp(a)
disp('thus, the upper triangular matrix is')
U=a;
disp(U,'U=')
disp('the lower triangular matrix is:')
L=[1 0;d(2,1)/d(1,1) 1];
disp(L,'L=')
```

Scilab code Exa 2.5.13 LU decomposition of a matrix

```
disp('given matrix is:')
a=[1 3 -5 -3;-1 -5 8 4;4 2 -5 -7;-2 -4 7 5]
d=a;
disp(a)
disp('performing row operations')
p21=a(2,1)/a(1,1);p31=a(3,1)/a(1,1);p41=a(4,1)/a
    (1,1);
a(2,:)=a(2,:)-p21*a(1,:)
a(3,:)=a(3,:)-p31*a(1,:)
a(4,:)=a(4,:)-p41*a(1,:)
disp(a)
p32=a(3,2)/a(2,2);p42=a(4,2)/a(2,2)
a(3,:)=a(3,:)-p32*a(2,:)
a(4,:)=a(4,:)-p42*a(2,:)
disp(a)
disp('thus, lower triangular matrix is:')
L=[1 0 0 0;p21 1 0 0;p31 p32 1 0;p41 p42 0 1]
disp(L,'L=')
disp('Upper triangular matrix is:')
disp(a,'U=')
```

Scilab code Exa 2.6.1 Application of matrix algebra

```
disp('the consumption matrix is:')
C=[.1 .6 .6;.3 .2 0;.3 .1 .1];
disp(C)
disp('Assuming that agriculture plans to produce 100
     units and other units produce nothing')
disp('the production vector is given by')
x=[0;100;0];
```

```
disp(x,'x=')
disp('thus the intermediate demand is:')
disp(C*x)
```

Scilab code Exa 2.6.7 Application of matrix algebra

```
disp('the consumption matrix is:')
C=[0 .5;.6 .2];
disp(C)
disp('the demand for 1 unit of output sector 1')
d1=[1;0]
disp(d1)
disp('the production required to satisfy demand d1
   is:')
x1=inv(eye(2,2)-C)*d1
disp(x1,'x1=')
disp('the final demand is:')
d2=[51;30]
disp(d2,'d2=')
disp('the production required to satisfy demand d2
   is:')
x2=inv(eye(2,2)-C)*d2
disp(x2,'x2=')
```

Scilab code Exa 2.7.1 Transformation using matrices

```
disp('consider the matrix')
a=[1 .25 0;0 1 0;0 0 1]
disp(a)
disp('consider a vector')
x=[6;8;0]
disp(x)
disp('the effect of the matric on the vector is:')
```

```
disp(a*x)
disp('now consider the matrix:')
b=[1 .25;0 1]
disp(b)
disp('considering the same vector')
x1=[6;8]
disp(x1)
disp('the effect of the new matrix on the vector is:
   ')
disp(b*x1)
disp('thus we can see that the two matrices have the
    same effect on vectors')
```

Scilab code Exa 2.7.7 Transformation using matrices

```
disp('the matrix in R2 to rotate a vector by 60
   degrees is:')
a=[cos(%pi/3) -sin(%pi/3);sin(%pi/3) cos(%pi/3)]
disp(a)
x=[6;8]
disp(x,'x=')
disp('so the 3X3 matrix for rotation about x is:')
y=[1 0 6;0 1 8;0 0 1]
z=[1 0 -6;0 1 -8;0 0 1]
a=[cos(%pi/3) -sin(%pi/3) 0;sin(%pi/3) cos(%pi/3)
   0;0 0 1]
R=y*(a*z)
disp(R)
```

Scilab code Exa 2.8.7 Column space of a matrix

```
disp('the given matrix is:')
A=[2 -3 -4;-8 8 6;6 -7 -7]
```

```
disp(A,'A=')
disp('the given vector is:')
p=[6;-10;11]
disp(p,'p=')
disp('combining A and p')
b=[A p]
disp(b)
disp('performing row operations')
b(2,:)=b(2,:)-(b(2,1)/b(1,1))*b(1,:)
b(3,:)=b(3,:)-(b(3,1)/b(1,1))*b(1,:)
disp(b)
b(3,:)=b(3,:)-(b(3,2)/b(2,2))*b(2,:)
disp(b)
if(b(3,3)==0 & b(3,4)==0)
    disp('p lies in column space of A')
  else
    disp('p does not lie in column space of A')
 end
```

Scilab code Exa 2.8.23 Pivot columns

```
disp('the given matrix is:')
a=[4 5 9 -2;6 5 1 12;3 4 8 -3]
disp(a)
disp('performing row operaions')
a(2,:)=a(2,:)-(a(2,1)/a(1,1))*a(1,:)
a(3,:)=a(3,:)-(a(3,1)/a(1,1))*a(1,:)
disp(a)
a(3,:)=a(3,:)-(a(3,2)/a(2,2))*a(2,:)
disp(a)
a(1,:)=a(1,:)/a(1,1)
a(2,:)=a(2,:)/a(2,2)
disp(a)
for i=1:3
  for j=i:4
```

```
    if(a(i,j)<>0)
       disp('is a pivot column',j,'column')
       break
    end
  end
end
```

Scilab code Exa 2.8.25 Pivot columns

```
disp('the given matrix is:')
a=[1 4 8 -3 -7;-1 2 7 3 4;-2 2 9 5 5;3 6 9 -5 -2]
disp(a)
disp('performing row operations')
a(2,:)=a(2,:)-(a(2,1)/a(1,1))*a(1,:)
a(3,:)=a(3,:)-(a(3,1)/a(1,1))*a(1,:)
a(4,:)=a(4,:)-(a(4,1)/a(1,1))*a(1,:)
disp(a)
a(3,:)=a(3,:)-(a(3,2)/a(2,2))*a(2,:)
a(4,:)=a(4,:)-(a(4,2)/a(2,2))*a(2,:)
disp(a)
a(4,:)=a(4,:)-(a(4,4)/a(3,4))*a(3,:)
disp(a)
for i=1:4
  for j=i:5
    if(a(i,j)<>0)
      disp('is a pivot column',j,'column')
      break
    end
  end
end
```

Scilab code Exa 2.9.13 Dimension of a matrix

```
disp('the given matrix is:')
a=[1 -3 2 -4;-3 9 -1 5;2 -6 4 -3;-4 12 2 7]
disp(a)
disp('performing row operations')
a(2,:)=a(2,:)-(a(2,1)/a(1,1))*a(1,:)
a(3,:)=a(3,:)-(a(3,1)/a(1,1))*a(1,:)
a(4,:)=a(4,:)-(a(4,1)/a(1,1))*a(1,:)
disp(a)
a(4,:)=a(4,:)-2*a(2,:)
disp(a)
a(4,:)=a(4,:)-a(3,:)
disp(a)
k=0
for i=1:4
  for j=i:4
    if(a(i,j)<>0)
      k=k+1
      break
    end
  end
end
disp(k,'dimension of the matrix=')
```

Chapter 3

DETERMINANTS

Scilab code Exa 3.1.1 Determinant of a matrix

```
disp('the given matrix is:')
A=[3 0 4;2 3 2;0 5 -1]
disp(A)
disp('calculating det(A) using cofactor expression
   along first row')
disp('det(A)=3 X (-1 X 3-5 X 2)+4 X (2 X 5-3 X 0)')
disp(det(A),'=')
```

Scilab code Exa 3.1.7 Determinant of a matrix

```
disp('given matrix is:')
A=[4 3 0;6 5 2;9 7 3]
disp(A)
disp('calculating det(A) using cofactor expression
   along first row')
disp('det(A)=4 X (5 X 3-7 X 2)-3 X (6 X 3-9 X 2)')
disp(det(A),'=')
```

Scilab code Exa 3.1.13 Determinant of a matrix

```
disp('the given matrix is:')
A=[4 0 -7 3 -5;0 0 2 0 0;7 3 -6 4 -8;5 0 5 2 -3;0 0
   9 -1 2]
disp(A,'A=')
P=A
disp('since row 2 has maximum zeros, using row 2 for
    cofactor expression')
A(2,:)=[]
A(:,3)=[]
disp('deleting second row and third column from A,
   we get')
disp(A)
disp(A,'det','det(A)=-2 X')
disp('for the 4X4 matrix obtained, using column 2
   for cofactor exansion')
disp('deleting second column and row from the 4X4
   matrix')
A(2,:)=[]
A(:,2)=[]
disp(A)
disp(A,'det','det(A)=-2 X 3 X')
disp('-6 X [4 X (4-3)-5 X (6-5)]','=')
disp(-6*det(A),'=')
```

Scilab code Exa 3.1.19 Property of determinants

```
disp('the given matrix is:')
disp('A=')
disp('a  b')
disp('c  d')
```

```
disp('det(A)=ad-bc')
disp('interchanging the rows of A, we get')
disp('B=')
disp('c  d')
disp('a  b')
disp('det(B)=bc-ad')
disp('-(ad-bc)','=')
disp('-det(A)','=')
disp('interchanging 2 rows reverses the sign of the
   determinant')
disp('at least for the 2X2 case')
```

Scilab code Exa 3.1.37 Property of determinants

```
A=[3 1;4 2]
disp('the given matrix is:')
disp(A)
disp(det(A),'det(A)=')
disp('5 X A = ')
disp(5*A)
disp(det(5*A),'det(5*A)=')
disp('thus, det(5A) is not equal to 5Xdet(A)')
disp('infact, the relation between det(rA) and det(A
   ) for a nxn matrix is:')
disp('det(rA)=(r^n)*det(A)')
```

Scilab code Exa 3.2.7 Determinant of a matrix

```
disp('the given matrix is:')
A=[1 3 0 2;-2 -5 7 4;3 5 2 1;1 -1 2 -3]
disp(A,'A=')
disp('performing row operations')
A(2,:)=A(2,:)-(A(2,1)/A(1,1))*A(1,:)
```

```
A(3,:)=A(3,:)-(A(3,1)/A(1,1))*A(1,:)
A(4,:)=A(4,:)-(A(4,1)/A(1,1))*A(1,:)
disp(A)
A(3,:)=A(3,:)-(A(3,2)/A(2,2))*A(2,:)
A(4,:)=A(4,:)-(A(4,2)/A(2,2))*A(2,:)
disp(A)
A(4,:)=A(4,:)-(A(4,3)/A(3,3))*A(3,:)
disp(A)
disp('det(A) is the product of diagonal entries')
disp(det(A),'det(A)=')
```

Scilab code Exa 3.2.13 Determinant of a matrix

```
disp('the given matrix is:')
a=[2 5 4 1;4 7 6 2;6 -2 -4 0;-6 7 7 0]
disp(a,'A=')
disp('performing row operations')
a(2,:)=a(2,:)-2*a(1,:)
disp(a)
disp('using cofactor expansion about fourth column')
a(1,:)=[]
a(:,4)=[]
disp(a,'det','det(A)= -1 X')
disp('performing row operations')
a(3,:)=a(3,:)+a(2,:)
disp(a)
disp('using cofactor expansion about first column')
a(2,:)=[]
a(:,1)=[]
disp(a,'det','det(A)= -1 X -6 X')
disp(6*det(a),'=')
```

Scilab code Exa 3.2.19 Determinant of a matrix

```
disp('the given matrix is:')
disp('A=')
disp(' a         b         c')
disp('2d+a     2e+b     2f+c')
disp(' g         h         i')
disp('B=')
disp('a  b  c')
disp('d  e  f')
disp('g  h  i')
disp('given, det(B)=7')
disp('performing row operations on A')
disp('R2=R2-R1')
disp('A=')
disp('a   b   c')
disp('2d  2e  2f')
disp('g   h   i')
disp('factoring 2 out of row 2')
disp('A=')
disp('2 X')
disp('a  b  c')
disp('d  e  f')
disp('g  h  i')
disp('therefore, det(A)=2 X det(B)')
disp('=2 X 7')
disp('= 14')
```

Scilab code Exa 3.2.25 Linear independency using determinants

```
disp('the given vectors are:')
v1=[7 -4 -6]'
v2=[-8 5 7]'
v3=[7 0 -5]'
disp(v3,'v3=',v2,'v2=',v1,'v1=')
disp('combining them as a matrix')
a=[v1 v2 v3]
```

```
disp(a,'A=')
disp('if det(A) is not equal to zero, then v1 v2 and
    v3 are linearly independent')
disp('expanding about third column')
disp('det(A)=7 X (-28+30) - 5 X (35-32)')
disp(det(a),'=')
disp('hence, v1 v2 and v3 are linearly independent')
```

Scilab code Exa 3.3.1 Cramers rule

```
disp('the co-efficient matrix is:')
a=[5 7;2 4]
disp(a,'A=')
disp('the RHS is:')
b=[3;1]
disp(b)
disp('applying cramers rule')
disp('replacing first column of matrix A by b')
A1=[3 7;1 4]
disp(A1,'A1=')
disp('replacing second column of matrix A by b')
A2=[5 3;2 1]
disp(A2,'A2=')
disp('x1=det(A1)/det(A)')
disp((det(A1)/det(a)),'=')
disp('x2=det(A2)/det(A)')
disp((det(A2)/det(a)),'=')
```

Scilab code Exa 3.3.13 Inverse of a matrix

```
disp('the given matrix is:')
a=[3 5 4;1 0 1;2 1 1]
disp(a,'A=')
```

```
disp('the cofactors are:')
C11=det([0 1;1 1])
disp(C11,'C11=')
C12=-det([1 1;2 1])
disp(C12,'C12=')
C13=det([1 0;2 1])
disp(C13,'C13=')
C21=-det([5 4;1 1])
disp(C21,'C21=')
C22=det([3 4;2 1])
disp(C22,'C22=')
C23=-det([3 5;2 1])
disp(C23,'C23=')
C31=det([5 4;0 1])
disp(C31,'C31=')
C32=-det([3 4;1 1])
disp(C32,'C32=')
C33=det([3 5;1 0])
disp(C33,'C33=')
B=[C11 C12 C13;C21 C22 C23;C31 C32 C33]'
disp('adj(A)=')
disp(B)
C=B/(det(a))
disp('inv(A)=')
disp(C)
```

Scilab code Exa 3.3.19 Application of determinant

```
disp('the points forming the parrallelogram are')
disp('(0,0),(5,2),(6,4),(11,6)')
disp('using the vertices adjacent to origin to form
   a matrix')
A=[5 6;2 4]
disp(A,'A=')
disp('Area of parallelogram = det(A)')
```

```
disp(det(A),'=')
```

Chapter 4

VECTOR SPACES

Scilab code Exa 4.1.13 Subspace of vectors

```
disp('the given vectors are:')
v1=[1;0;-1]
disp(v1,'v1=')
v2=[2;1;3]
disp(v2,'v2=')
v3=[4;2;6]
disp(v3,'v3=')
w=[3;1;2]
disp(w,'w=')
disp('It is clear that w is not one of the three
    vectors in v1,v2 and v3')
disp('The span of v1,v2 and v3 contains infinitely
    many vectors.')
disp('To check if w is in the subspace of v1,v2 and
    v3,')
disp('we form an augmented matrix.')
a=[1 2 4 3;0 1 2 1;-1 3 6 2]
disp(a)
disp('performing row operations')
disp('R3=R3+R1')
a(3,:)=a(3,:)+a(1,:)
```

```
disp(a)
disp('R3=R3−5xR2')
a(3,:)=a(3,:)-5*a(2,:)
disp(a)
disp('there is no pivot in the augmented column,')
disp('hence the vector equation is consistent and w
   is in span{v1 v2 v3}.')
```

Scilab code Exa 4.2.1 Null space of a matrix

```
disp('the given matrix is:')
a=[3 -5 -3;6 -2 0;-8 4 1]
disp(a,'A=')
disp('the vector x is:')
x=[1;3;-4]
disp(x,'x=')
disp('To check if x is in nullspace of A')
disp('Ax=')
disp([0;0;0],'=')
disp('hence, x is in the null space of A')
```

Scilab code Exa 4.3.13 Column space of a matrix

```
disp('the given matrix is:')
a=[1 0 6 5;0 2 5 3;0 0 0 0]
p=a
disp(a,'A=')
disp('Reducing A to echelon form')
disp('R2=R2/2')
a(2,:)=a(2,:)/2
disp(a)
disp('the pivot columns are column 1 and 2 of A')
disp('hence column space of A is:')
```

```
disp('span')
disp(a(:,1),'and',a(:,2))
```

Scilab code Exa 4.4.7 Gaussian Elimination

```
disp('vector x=')
x=[8;-9;6]
disp(x)
disp('the given basis is:')
b1=[1;-1;-3]
b2=[-3;4;9]
b3=[2;-2;4]
disp(b1,'b1=')
disp(b2,'b2=')
disp(b3,'b3=')
disp('to solve the vector equation')
disp('an augmented matrix is formed')
a=[1 -3 2 8;-1 4 -2 -9;-3 9 4 6]
disp(a,'A=')
disp('performing row operations')
a(2,:)=a(2,:)-(a(2,1)/a(1,1))*a(1,:)
a(3,:)=a(3,:)-(a(3,1)/a(1,1))*a(1,:)
disp(a)
a(3,:)=a(3,:)/a(3,3)
a(1,:)=a(1,:)-2*a(3,:)
disp(a)
a(1,:)=a(1,:)+3*a(2,:)
disp(a)
disp('Xb=')
disp(a(:,4))
```

Scilab code Exa 4.4.27 Linear independence of vectors

```
disp('to check if vectors v1 v2 and v3 are linearly
   independent')
v1=[1;0;0;1]
v2=[3;1;-2;0]
v3=[0;-1;3;-1]
disp(v3,'v3=',v2,'v2=',v1,'v1=')
disp('forming an augmented matrix')
a=[1 3 0 0;0 1 -1 0;0 -2 3 0;1 0 -1 0]
disp(a,'A=')
disp('performing row operations')
a(4,:)=a(4,:)-a(1,:)
disp(a)
a(3,:)=a(3,:)+2*a(2,:)
a(4,:)=a(4,:)+3*a(2,:)
disp(a)
a(4,:)=a(4,:)+4*a(3,:)
disp(a)
disp('since the vector equation has only the trivial
    solution')
disp('vectors v1 v2 and v3 are linearly independent'
   )
```

Scilab code Exa 4.4.31a Span of vectors

```
disp('to check if the polynomials span R3')
disp('placing the coordinate vectors of the
   polynomial into the columns of a matrix')
a=[1 -3 -4 1;-3 5 5 0;5 -7 -6 1]
disp(a,'A=')
disp('performing row operations')
a(2,:)=a(2,:)+3*a(1,:)
a(3,:)=a(3,:)-5*a(1,:)
disp(a)
a(3,:)=a(3,:)+2*a(2,:)
disp(a)
```

```
disp('the four vectors DO NOT span R3 as there is no pivot in row 3')
```

Scilab code Exa 4.4.31b Span of vectors

```
disp('to check if the polynomials span R3')
disp('placing the coordinate vectors of the polynomial into the columns of a matrix')
a=[0 1 -3 2;5 -8 4 -3;1 -2 2 0]
disp(a,'A=')
disp('performing row operations')
a([1 3],:)=a([3 1],:)
disp(a)
a(2,:)=a(2,:)-5*a(1,:)
disp(a)
a(3,:)=a(3,:)-.5*a(2,:)
disp(a)
disp('the four vectors SPAN R3 as there is a pivot in each row')
```

Scilab code Exa 4.5.3 Dimension of a vector space

```
disp('to find the dimension of subspace H, which is the set of linear combination of vectors v1 v2 and v3')
v1=[0;1;0;1]
v2=[0;-1;1;2]
v3=[2;0;-3;0]
disp(v3,'v3=',v2,'v2=',v1,'v1=')
disp('Clearly , v1 is not equal to zero')
disp('and v2 is not a multiple of v1 as third element of v1 is zero whereas that of v2 is 1.')
```

```
disp('Also, v3 is not a linear combination of v1 and
    v2 as the first element of v1 and v2 is zero but
    that of v3 is 2')
disp('Hence, v1 v2 and v3 are linearly independent
   and dim(H)=3')
```

Scilab code Exa 4.6.1 Rank of a matrix

```
disp('to find the rank of matrix A')
a=[1 -4 9 -7;-1 2 -4 1;5 -6 10 7]
p=a
disp(a,'A=')
disp('performing row operations')
a(2,:)=a(2,:)+a(1,:)
a(3,:)=a(3,:)-5*a(1,:)
disp(a)
a(3,:)=a(3,:)+7*a(2,:)
disp(a)
disp('It is clear that matrix A has 2 pivot columns'
   )
disp('Hence, rank(A)=2')
disp('COlumns 1 and 2 are pivot columns')
disp(p(:,1),'and',p(:,2),'Hence, basis for C(A) is:'
   )
disp('Basis for row space of A is:')
disp(a(1,:),'and',a(2,:))
disp('To find the basis of N(A), solve Ax=0')
disp('on solving, we get the basis of N(A) as:')
u=[1;2.5;1;0]
v=[-5;-3;0;1]
disp(v,'and',u)
```

Chapter 5

EIGENVALUES AND EIGENVECTORS

Scilab code Exa 5.1.1 Eigenvalue of a matrix

```
disp('to check if 2 is an eigenvalue of matrix A')
a=[3 2;3 8]
disp(a,'A=')
disp('A-2I=')
b=a-2*eye(2,2)
disp(b)
disp('The columns of A are clearly independent,')
disp('hence (A-2I)x=0 has a non trivial solution and
    2 is an eigenvalue of matrix A')
```

Scilab code Exa 5.1.7 Eigenvalue of a matrix

```
disp('To check if 4 is an eigenvalue of matrix A')
a=[3 0 -1;2 3 1;-3 4 5]
disp(a,'A=')
disp('Therefore')
```

```
disp('A-4I=')
disp(a-4*eye(3,3))
b=a-4*eye(3,3)
disp('to check the invertibility of A-4I, form an
   augmented matrix')
c=[b [0;0;0]]
disp(c)
disp('performing row operations')
c(2,:)=c(2,:)+2*c(1,:)
c(3,:)=c(3,:)-3*c(1,:)
disp(c)
c(3,:)=c(3,:)+4*c(2,:)
disp(c)
disp('We can see that there exists a non trivial
   solution.')
disp('Hence, 4 is an eigenvalue of A.')
disp('For the eigenvector, -x1-x3=0 and -x2-x3=0')
disp('If x3=1,')
x=[-1;-1;1]
disp(x,'x=')
```

Scilab code Exa 5.1.13 Eigenvectors

```
disp('To find a basis for the eigenspace')
disp('Matrix A=')
a=[4 0 1;-2 1 0;-2 0 1]
disp(a)
disp('for lambda=1')
disp('A-1I=')
b=a-eye(3,3)
disp(b)
disp('solving (A-I)x=0, we get')
disp('-2*x1=0 and 3*x1+x3=0')
disp('therefore, x1=x3=0')
disp('which leaves x2 as a free variable')
```

```
disp('Hence a basis for the eigen space is:')
disp([0;1;0])
disp('for lambda=2')
disp('A-2I=')
b=a-2*eye(3,3)
disp(b)
disp('performing row operations on the augmented matrix')
c=[b [0;0;0]]
disp(c)
c(2,:)=c(2,:)+c(1,:)
c(3,:)=c(3,:)+c(1,:)
disp(c)
c(1,:)=c(1,:)/c(2,2)
disp(c)
disp('We can see that x3 is a free variable')
disp('x2=x3 and x1=-.05*x3')
disp('Hence, a basis for the eigenspace is:')
disp([-.5;1;1])
disp('for lambda=3')
disp('A-3I=')
b=a-3*eye(3,3)
disp(b)
disp('performing row operations on the augmented matrix')
c=[b [0;0;0]]
disp(c)
c(2,:)=c(2,:)+2*c(1,:)
c(3,:)=c(3,:)+2*c(1,:)
disp(c)
c(2,:)=c(2,:)/2
disp(c)
disp('Again x3 is a free variable')
disp('x1=-x3 and x2=x3')
disp('Hence, a basis for the eigenspace is:')
disp([-1;1;1])
```

Scilab code Exa 5.1.19 Property of non invertible matrices

```
disp('The given matrix is:')
a=[1 1 1;2 2 2;3 3 3]
disp(a,'A=')
disp('A is not invertible because its columns are
   linearly dependent.')
disp('Hence, 0 is an eigenvalue of matrix A.')
```

Scilab code Exa 5.2.1 Eigenvalue of a matrix

```
disp('To find the eigenvalue of matrix A')
disp('A=')
a=[2 7;7 2]
disp(a)
disp('Eigen values of A are:')
disp(spec(a))
```

Scilab code Exa 5.2.7 Complex eigenvalues

```
disp('To find the eigenvalues of matrix A.')
disp('A=')
a=[5 3;-4 4]
disp(a)
disp('Eigen values of A are:')
disp(spec(a))
disp('Hence, A has no real eigenvalues.')
```

Scilab code Exa 5.2.13 Eigenvalues of a matrix

```
disp('To find the eigenvalues of the matrix A')
disp('A=')
a=[6 -2 0;-2 9 0;5 8 3]
disp(a)
disp('Eigenvalues of A are:')
disp(spec(a))
```

Scilab code Exa 5.2.25 Eigenvectors

```
disp('Matrix A=')
a=[.6 .3;.4 .7]
disp(a)
disp('Eigenvector v1=')
v1=[3/7;4/7]
disp(v1)
disp('vector Xo=')
Xo=[.5;.5]
disp(Xo)
disp('Eigenvalues of A are:')
c=spec(a)
disp(c)
disp('To verify if v1 is an eigenvector of A:')
disp('A*v1=')
disp(a*v1)
disp('=')
disp('1*v1')
disp('Hence v1 is an eigenvector of A corresponding to eigenvalue 1.')
disp('for lambda=.3')
disp('A-.3I=')
b=a-.3*eye(2,2)
disp(b)
disp('performing row operations on the augmented
```

```
    matrix')
c=[b [0;0]]
disp(c)
c(2,:)=c(2,:)-(c(2,1)/c(1,1))*c(1,:)
disp(c)
disp('hence, x1+x2=0')
disp('Eigenvector corresponding to eigenvalue .3 is:
    ')
disp([-1;1])
```

Scilab code Exa 5.3.1 Diagonalization of a matrix

```
disp('The given eigenvector matrix is:')
p=[5 7;2 3]
disp(p,'P=')
disp('The diagonal matrix is:')
d=[2 0;0 1]
disp(d,'D=')
disp('Therefore, matrix A=PD(p^-1)')
s=inv(p)
disp(p*d*s)
disp('Hence, A^4=P(D^4)(P^-1)')
disp(p*(d^4)*s)
```

Scilab code Exa 5.3.7 Diagonalization of a matrix

```
disp('the given matrix is:')
a=[1 0;6 -1]
disp(a,'A=')
disp('Since A is triangular, eigenvalues are the
    diagonal entries.')
disp(a(2,2),a(1,1),'Eigenvalues are:')
disp('for lambda=1')
```

```
disp('A-1I=')
b=a-eye(2,2)
disp(b)
disp('Hence, x1=(1/3)x2 with x2 as free variable.')
disp('Eigenvector corresponding to lambda=1 is:')
u1=[1;3]
disp(u1)
disp('for lambda=-1')
disp('A-(-1)I=')
b=a+eye(2,2)
disp(b)
disp('Hence, x1=0 with x2 as free variable.')
disp('Eigenvector corresponding to lambda=-1 is:')
u2=[0;1]
disp(u2)
disp('Thus, matrix P=')
disp([u1 u2])
disp('and matrix D=')
disp([1 0;0 -1])
```

Scilab code Exa 5.3.13 Diagonalization of a matrix

```
disp('Given matrix A=')
a=[2 2 -1;1 3 -1;-1 -2 2]
disp(a)
disp('Given its eigen values are 5 and 1')
disp('for lambda=5')
disp('A-5I=')
b=a-5*eye(3,3)
disp(b)
disp('performing row operations')
c=[b [0;0;0]]
disp(c)
c([1 2],:)=c([2 1],:)
disp(c)
```

```
c(2,:)=c(2,:)+3*c(1,:)
c(3,:)=c(3,:)+c(1,:)
disp(c)
c(3,:)=c(3,:)-c(2,:)
disp(c)
c(2,:)=c(2,:)/c(2,2)
disp(c)
disp('With x3 as free variable, x1=-x3 and x2=-x3')
disp('Hence, for lambda=5 eigenvector is:')
u1=[-1;-1;1]
disp(u1)
disp('for lambda=1')
disp('A-I=')
b=a-eye(3,3)
disp(b)
disp('performing row operations')
c=[b [0;0;0]]
disp(c)
c(2,:)=c(2,:)-c(1,:)
c(3,:)=c(3,:)+c(1,:)
disp(c)
disp('With x2 and x3 as free variables, eigen vectors corresponding to lambda=1 are')
u2=[-2;1;0]
u3=[1;0;1]
disp(u3,u2)
disp('Hence, matrix P=')
disp([u1 u2 u3])
disp('and matrix D=')
disp([5 0 0;0 1 0;0 0 1])
```

Scilab code Exa 5.4.31 PD decomposition of a matrix

```
disp('Given matrix A=')
a=[-7 -48 -16;1 14 6;-3 -45 -19]
```

```
disp(a)
disp('and matrix P=')
p=[-3 -2 3;1 1 -1;-3 -3 0]
disp(p)
disp('Hence, marix D=')
s=inv(p)
disp(s*a*p)
```

Scilab code Exa 5.5.1 Complex eigenvectors

```
disp('Matrix A=')
a=[1 -2;1 3]
disp(a)
disp('Eigen values of A are')
eig=spec(a)
disp(eig)
disp('for lambda=2+i')
i=sqrt(-1)
disp('A-(2+i)I=')
b=a-(2+i)*eye(2,2)
disp(b)
disp('With x2 as free variable, x1=-(1-i)x2')
disp('Hence, eigenvector corresponding to lambda=2+i
    is:')
disp([i-1;1])
disp('for lambda=2-i, eigenvector is:')
disp([-1-i;1])
```

Scilab code Exa 5.5.7 Scale factor of transformation

```
disp('Matrix A=')
a=[sqrt(3) -1;1 sqrt(3)]
disp(a)
```

```
disp('Eigenvalues of A are:')
eig=spec(a)
disp(eig)
disp('The scale factor associated with the
    transformation x to Ax is:')
disp(abs(eig(1,1)))
```

Chapter 6

ORTHOGONALITY AND LEAST SQUARES

Scilab code Exa 6.1.1 Dot product of vectors

```
disp('Vectors u an v are:')
u=[-1;2]
v=[4;6]
disp(v,u)
disp('Projection of v on u=(u.v)/(v.v)')
a=u'*v
b=u'*u
p=a/b
disp(p,'=')
```

Scilab code Exa 6.1.7 Norm of a vector

```
disp('w=')
w=[3;-1;-5]
disp(w)
disp('||w||=sqrt(9+1+25)')
```

```
disp(sqrt(35))
```

Scilab code Exa 6.1.13 Distance between two points

```
disp('Vector x and y are:')
x=[10;-3]
y=[-1;-5]
disp(y,x)
disp('||x-y||=sqrt(121+4)')
disp(sqrt(125),'=')
```

Scilab code Exa 6.2.1 Orthogonality of vectors

```
disp('To verify if u v and w are orthogonal')
u=[-1;4;-3]
v=[5;2;1]
w=[3;-4;-7]
disp(w,v,u)
disp('u.v=')
disp(v'*u)
disp('u.w=')
disp(u'*w)
disp('Since u.w is not equal to zero, the set {u v w
    } is not orthogonal.')
```

Scilab code Exa 6.2.7 Orthogonal basis

```
disp('vectors u1 u2 and x are:')
u1=[2;-3]
u2=[6;4]
```

```
x=[9;-7]
disp(x,u2,u1)
disp('u1.u2=')
disp(u1'*u2)
disp('u1.u2=0, {u1 u2} is an orthogonal set')
disp('Hence {u1 u2} forms a basis of R2')
disp('x can be written as: x=a*u1+b*u2')
disp('where a=(x.u1)/(u1.u1)')
a1=x'*u1
a2=u1'*u1
a=a1/a2
disp(a,'=')
disp('and b=(x.u2)/(u2.u2)')
b1=x'*u2
b2=u2'*u2
b=b1/b2
disp(b,'=')
```

Scilab code Exa 6.2.13 Projection of vectors

```
disp('Vectors y and u are:')
y=[2;3]
u=[4;-7]
disp(u,y)
disp('The orthogonal projection of y on u=((y.u)/(u.u))*u')
a=y'*u
b=u'*u
c=(a/b)*u
disp(c,'=')
disp('The component of y orthogonal to u is:')
disp(y-c)
```

Scilab code Exa 6.2.19 Orthonormal vectors

```
disp('given vectors u and v are:')
u=[-.6;.8]
v=[.8;.6]
disp(v,u)
disp('u.v=')
disp(u'*v)
disp('Hence, {u v} is an orthogonal set.')
disp('||u||=1 and ||v||=1')
disp('Thus, {u v} is an orthonormal set')
```

Scilab code Exa 6.3.1 Orthogonal projection

```
disp('Given vectors are:')
u1=[0;1;-4;-1]
u2=[3;5;1;1]
u3=[1;0;1;-4]
u4=[5;-3;-1;1]
x=[10;-8;2;0]
disp(x,'x=',u4,'u4=',u3,'u3=',u2,'u2=',u1,'u1=')
disp('The vector in span{u4}=((x.u4)/(u4.u4))*u4')
a1=x'*u4
a2=u4'*u4
disp((a1/a2)*u4)
disp('Therefore, the vector in span{u1 u2 u3}=x-2*u4
    ')
disp(x-2*u4)
```

Scilab code Exa 6.3.7 Orthogonal projection

```
disp('Vectors u1 u2 and y are')
u1=[1;3;-2]
```

```
u2=[5;1;4]
y=[1;3;5]
disp(y,'y=',u2,'u2=',u1,'u1=')
disp('u1.u2=')
a=u1'*u2
disp(a,'=')
disp('Hence, {u1 u2} form an orthogonal basis.')
disp('Let W=span{u1 u2}')
disp('Therefore, projection of y on W is:')
disp('((y.u1)/(u1.u1))*u1+((y.u2)/(u2.u2))*u2')
a1=y'*u1
a2=u1'*u1
b1=y'*u2
b2=u2'*u2
disp((b1/b2)*u2,'+',(a1/a2)*u1,'=')
```

Scilab code Exa 6.3.13 Orthogonal projection

```
disp('Given vectors are:')
v1=[2;-1;-3;1]
v2=[1;1;0;-1]
z=[3;-7;2;3]
disp(z,'z=',v2,'v2=',v1,'v1=')
a=v1'*v2
disp(a,'v1.v2=')
if(a==0)
  disp('v1 and v2 are orthogonal')
end
disp('By best spproximation theorem, closest point
   in span{v1 v2} to z is the orthogonal projection'
   )
disp('=((z.v1)/(v1.v1))*v1+((z.v2)/(v2.v2))*v2')
a1=z'*v1
a2=v1'*v1
b1=z'*v2
```

```
b2=v2'*v2
disp((a1/a2)*v1,'+',(b1/b2)*v2,'=')
disp((a1/a2)*v1+(b1/b2)*v2,'=')
```

Scilab code Exa 6.3.19 Orthogonal decomposition theorem

```
disp('By orthogonal decomposition theorem,')
disp('u3 is the sum of a vector in W=span{u1 u2} and
    a vector v orthogonal to W')
disp('To find v, given u1 and u2')
u1=[1;1;-2]
u2=[5;-1;2]
disp(u2,'u2=',u1,'u1=')
disp('Projection of u3 on W')
disp('= (-1/3)*u1+(1/15)*u2')
disp((-1/3)*u1+(1/15)*u2,'=')
disp('v= u3-(projection of u3 on W)')
disp((-1/3)*u1+(1/15)*u2,'-',[0;0;1],'=')
disp([0;0;1]-((-1/3)*u1+(1/15)*u2),'=')
```

Scilab code Exa 6.4.1 Gram Schimdt Orthogonalisation

```
disp('to orthogonalise the given vectors using Gram-
   Schimdt orthogonalisation')
x1=[3;0;-1]
x2=[8;5;-6]
disp(x2,'x2=',x1,'x1=')
disp('Let v1=x1')
v1=x1
disp('v2=x2-((x2.v1)/(v1.v1))*v1')
a1=x2'*v1
a2=v1'*v1
p=(a1/a2)*v1
```

```
v2=x2-p
disp(p,'-',x2,'=')
disp(v2,'=')
disp('Thus, an orthogonal basis is:')
disp(v2,v1)
```

Scilab code Exa 6.4.7 Gram Schimdt Orthogonalisation

```
disp('to orthogonalise the given vectors using Gram-
   Schimdt orthogonalisation')
x1=[2;-5;1]
x2=[4;-1;2]
disp(x2,'x2=',x1,'x1=')
disp('Let v1=x1')
v1=x1
disp('v2=x2-((x2.v1)/(v1.v1))*v1')
a1=x2'*v1
a2=v1'*v1
p=(a1/a2)*v1
v2=x2-p
disp(p,'-',x2,'=')
disp(v2,'=')
disp('Thus, an orthogonal basis is:')
disp(v2,v1)
disp('Normalizing v1 and v2, we get')
s1=sqrt(v1(1,1)^2+v1(2,1)^2+v1(3,1)^2)
s2=sqrt(v2(1,1)^2+v2(2,1)^2+v2(3,1)^2)
disp(v2/s2,v1/s1)
```

Scilab code Exa 6.4.13 QR decomposition of a matrix

```
disp('QR decomposition of a matrix')
disp('given matrix A=')
```

```
a=[5 9;1 7;-3 -5;1 5]
disp(a)
disp('given matrix Q=')
q=(1/6)*[5 -1;1 5;-3 1;1 3]
disp(q)
disp('Therefore, R=')
s=q'*a
disp(s)
```

Scilab code Exa 6.5.1 Least square solution

```
disp('The co-efficient matrix is:')
a=[-1 2;2 -3;-1 3]
disp(a,'A=')
disp('The RHS is:')
b=[4;1;2]
disp(b)
disp('Product of transpose of A and A=')
p1=a'*a
disp(p1)
disp('Product of transpose of A and b=')
p2=a'*b
disp(p2)
disp('Hence, the solution is:')
p=inv(p1)*p2
disp(p)
```

Scilab code Exa 6.5.7 Least square solution

```
disp('The co-efficient matrix is:')
a=[1 -2;-1 2;0 3;2 5]
disp(a,'A=')
disp('The RHS is:')
```

```
b=[3;1;-4;2]
disp(b,'b=')
disp('Product of transpose of A and A=')
p1=a'*a
disp(p1)
disp('Product of transpose of A and b=')
p2=a'*b
disp('Forming an augmented matrix to solve the
   normal equations')
p=[p1 p2]
disp(p)
disp('performing row operations')
disp('R2=R2-R1')
p(2,:)=p(2,:)-p(1,:)
disp(p)
disp('R1=R1/6 and R2=R2/36')
p(1,:)=p(1,:)/6
p(2,:)=p(2,:)/36
disp(p)
disp('R1=R1-R2')
p(1,:)=p(1,:)-p(2,:)
disp(p)
disp('Hence, the solution is:')
disp(p(:,3))
x=p(:,3)
disp('The least square error is=||Ax-b||')
disp('Ax-b=')
disp(a*x-b)
c=a*x-b
s=0
for i=1:4
  s=s+c(i,1)^2
end
disp('||Ax-b||=')
disp(sqrt(s))
```

Scilab code Exa 6.5.13 Least square solution

```
disp('To determine if u is the least square solution
    to Ax=b')
disp('Given')
a=[3 4;-2 1;3 4]
disp(a,'A=')
b=[11;-9;5]
disp(b,'b=')
u=[5;-1]
v=[5;-2]
disp(v,'v=',u,'u=')
disp('Au=')
disp(a*u)
c=b-a*u
disp(c,'b-Au=')
disp('||b-Au||=')
disp(sqrt(c(1,1)^2+c(2,1)^2+c(3,1)^2))
disp('Av=')
disp(a*v)
d=b-a*v
disp(d,'b-Av=')
disp('||b-Av||=')
disp(sqrt(d(1,1)^2+d(2,1)^2+d(3,1)^2))
disp('Since Av is more closer to A than Au, u is not
    the least square solution.')
```

Scilab code Exa 6.6.1 Least squares line

```
disp('To obtain a least sqaure line from the given
   data')
```

```
disp('Placing the x coordinates of the data in
    second column of matrix X we get:')
x=[1 0;1 1;1 2;1 3]
disp(x,'X=')
disp('Placing the y coordinates in y vector')
y=[1;1;2;2]
disp(y,'y=')
disp('Product of transpose of X and X=')
p1=x'*x
disp(p1)
disp('Product of transpose of X and y=')
p2=x'*y
disp(p2)
disp('The least square solution =')
disp(inv(p1)*p2)
p=inv(p1)*p2
disp('Hence, the least square line is:')
disp('x',p(2,1),'+',p(1,1),'=','y')
```

Chapter 7

SYMMETRIC MATRICES AND QUADRATIC FORMS

Scilab code Exa 7.1.1 Symmetric matrices

```
disp('To check if the given 2X2 matrix is symmetric'
  )
a=[3 5;5 -7]
disp(a,'A=')
if(a(1,2)==a(2,1))
  disp('A is a symmetric matrix because the (1,2)
     and(2,1) entries match.')
else
  disp('A is not a symmetric matrix')
end
```

Scilab code Exa 7.1.7 Orthogoanl matrix

```
disp('To show that the given matrix P is orthogonal.
  ')
p=[.6 .8;.8 -.6]
```

```
disp(p,'P=')
disp('P is composed of two vectors.')
p1=[.6;.8]
p2=[.8;-.6]
disp(p2,'p2=',p1,'p1=')
disp('To show that the columns are orthonormal')
disp('p1.p2=')
s=p1'*p2
r=p1'
disp(p2,'*',r,'=')
disp(s,'=')
if(s==0)
  disp('The columns of P are othonormal')
end
disp('||p1||=')
disp(sqrt(p(1,1)^2+p(2,1)^2))
disp('||p2||=')
disp(sqrt(p(1,2)^2+p(2,2)^2))
disp('Hence, ||p1||=||p2||=1. Thus P is an
   orthogonal matrix')
```

Scilab code Exa 7.1.13 PD decomposition of a matrix

```
disp('To diagonalize the given matrix A')
a=[3 1;1 3]
disp(a,'A=')
eig=spec(a)
disp('Eigen values of A are:')
disp(eig)
disp('for lambda=4')
disp('A-4I=')
disp(a-4*eye(2,2))
b=a-4*eye(2,2)
disp('To find the eigenvector, form an augmented
   matrix.')
```

```
c=[b [0;0]]
disp('performing row operations')
disp(c)
c(2,:)=c(2,:)+c(1,:)
disp(c)
disp('With x2 as free variable, x1=x2')
disp('Hence a basis for the eigenspace is:')
d=[1;1]
disp(d)
disp('Upon normalizing')
disp(d/(sqrt(2)))
u1=d/(sqrt(2))
disp('for lambda=2')
disp('A-2I=')
b=a-2*eye(2,2)
disp(b)
disp('To find the eigenvector, form an augmented
    matrix.')
c=[b [0;0]]
disp('performing row operations')
disp(c)
c(2,:)=c(2,:)-c(1,:)
disp(c)
disp('With x2 as free variable, x1=-x2')
disp('Hence a basis for the eigenspace is:')
d=[-1;1]
disp(d)
disp('Upon normalizing')
disp(d/(sqrt(2)))
u2=d/(sqrt(2))
disp('Matrix P=')
p=[u1 u2]
disp(p)
disp('The corresponding matrix D=')
disp([eig(2,1) 0;0 eig(1,1)])
```

Scilab code Exa 7.1.19 PD decomposition of a matrix

```
disp('PD decomposition of a matrix A')
a=[3 -2 4;-2 6 2;4 2 3]
disp(a,'A=')
disp('Eigenvalues of A are')
eig=spec(a)
disp(eig)
disp(eig(2,1),'for lambda =')
disp('A-(lambda)I=')
b=a-eig(2,1)*eye(3,3)
disp(b)
disp('To find eigenvector, form an augmented matrix'
    )
c=[b [0;0;0]]
disp(c)
disp('performing row operations')
c(2,:)=c(2,:)-(c(2,1)/c(1,1))*c(1,:)
c(3,:)=c(3,:)-(c(3,1)/c(1,1))*c(1,:)
disp(c)
disp('With x2 and x3 as free variables, we get two
    vectors.')
disp('x1=-.5x2+x3')
disp('Thus, the two vectors are')
v1=[-1;2;0]
v2=[1;0;1]
disp(v2,v1)
disp('Orthogonalizing v1 and v2')
disp('Let x1=v1')
disp('x2=v2-((v2.v1)/(v1.v1))*v1')
x1=v1
a1=v2'*v1
a2=v1'*v1
x2=v2-(a1/a2)*v1
```

```
x1=x1/(sqrt(x1(1,1)^2+x1(2,1)^2+x1(3,1)^2))
x1=x2/(sqrt(x2(1,1)^2+x2(2,1)^2+x2(3,1)^2))
disp('An orthonormal basis is:')
disp(x2,x1)
disp(eig(1,1),'for lambda=')
disp('A-(lambda)I=')
b=a-eig(1,1)*eye(3,3)
disp(b)
disp('To find eigenvector, form an augmented matrix'
    )
c=[b [0;0;0]]
disp(c)
disp('performing row operations')
c(2,:)=c(2,:)-(c(2,1)/c(1,1))*c(1,:)
c(3,:)=c(3,:)-(c(3,1)/c(1,1))*c(1,:)
disp(c)
c(3,:)=c(3,:)-(c(3,2)/c(2,2))*c(2,:)
disp(c)
c(1,:)=c(1,:)/c(1,1)
c(2,:)=c(2,:)/c(2,2)
disp(c)
c(1,:)=c(1,:)-(c(1,2)/c(2,2))*c(2,:)
disp(c)
disp('With x3 as free variable')
disp('x1=x3 and x2=-.5x3')
disp('Thus a basis for the eigenspace is:')
v3=[1;-.5;1]
disp(v3)
disp('upon normalizing')
v3=v3/(sqrt(v3(1,1)^2+v3(2,1)^2+v3(3,1)^2))
disp(v3)
disp('Thus, matrix P=')
disp([x1 x2 v3])
disp('Corresponding matrix D=')
disp([eig(2,1) 0 0;0 eig(3,1) 0;0 0 eig(1,1)])
```

Scilab code Exa 7.2.1 Quadratic form

```
disp('given matrix A and vector x')
a=[5 (1/3);(1/3) 1]
disp(a,'A=')
x=[6;1]
disp(x,'x=')
disp('Product of transpose of x and A and x=')
p=x'*a*x
disp(p)
disp('New value of vector x=')
x=[1;3]
disp(x)
disp('Product of transpose of x and A and x=')
p=x'*a*x
disp(p)
```