Scilab Textbook Companion for Linear Algebra And Its Applications by G. Strang[1]

Created by
Sri Harsha Chillara
B.Tech (pursuing)
Electrical Engineering
NIT, Suratkal
College Teacher
NA
Cross-Checked by
Sonanya Tatikola, IIT Bombay

May 20, 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: G. Strang

Publisher: Cengage Learning

Edition: 4

Year: 2011

ISBN: 81-3150-172-8

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

Matrix Notation and Matrix Multiplication

Scilab code Exa 1.3.1 Breakdown of elimination

```
clear;
close;
clc;
a =[1 1 1;2 2 5;4 6 8]
disp('x=[u;v;w]')
disp('R2=R2-R1,R3=R3-4*R1')
a(2,:)=a(2,:)-2*a(1,:);
a(3,:)=a(3,:)-4*a(1,:);
disp(a);
disp('R2<->R3')
b=a(2,:);
a(2,:)=a(3,:);
a(3,:)=b;
disp(a);
disp('The system is now triangular and the equations
    can be solved by Back substitution');
//end
```

Scilab code Exa 1.3.2 Breakdown of elimination

```
clear ;
close ;
clc ;
a =[1 1 1;2 2 5;4 4 8];
disp(a,'a=');
disp('x=[u;v;w]');
disp('R2=R2-2*R1,R3=R3-4*R1');
a(2,:)=a(2,:)-2*a(1,:);
a(3,:)=a(3,:)-4*a(1,:);
disp(a);
disp('No exchange of equations can avoid zero in the
    second pivot positon ,therefore the equations
   are unsolvable');
//end
```

Scilab code Exa 1.4.1 Multiplication of Two Matrices

```
clear ;
close ;
clc ;
A=[2 3;4 0];
disp(A,'A=');
B=[1 2 0;5 -1 0];
disp(B,'B');
disp(A*B,'AB=')
//end
```

Scilab code Exa 1.4.2 Multiplication with Row exchange matrix

```
clear;
close;
clc;
A=[2 3;7 8];
disp(A,'A=');
P=[0 1;1 0];
disp(P,'P(Row exchange matrix)=')
disp(P*A,'PA=')
//end
```

Scilab code Exa 1.4.3 Multiplication with Identity Matrix

```
//page 24
clear;
close;
clc;
A=[1 2;3 4];
disp(A,'A=');
I=eye(2,2);
disp(I,'I=');
disp(I*A,'IA=')
//end
```

Scilab code Exa 1.4.4 Marix multiplication not commutative

```
//page 25
clear;
close;
clc;
E=eye(3,3);
E(2,:)=E(2,:)-2*E(1,:);
```

```
disp(E,'E=');
F=eye(3,3);
F(3,:)=F(3,:)+F(1,:);
disp(F,'F=');
disp(E*F,'EF=')
disp(F*E,'FE=')
disp('Here,EF=FE,so this shows that these two
   matrices commute')
//end
```

Scilab code Exa 1.4.5 Order of Elimination

```
//page 25
clear;
close;
clc;
E=eye(3,3);
E(2,:)=E(2,:)-2*E(1,:);
disp(E,'E')
F=eye(3,3);
F(3,:)=F(3,:)+F(1,:);
disp(F,'F=');
G=eye(3,3);
G(3,:)=G(3,:)+G(2,:);
disp(G,'G')
disp(G*E,'GE=')
disp(E*G,'EG=')
disp('Here EG is not equal to GE,Therefore these two
    matrices do not commute and shows that most
   matrices do not commute.')
disp(G*F*E,'GFE=')
disp(E*F*G,'EFG=')
disp('The product GFE is the true order of elimation
   .It is the matrix that takes the original A to
   the upper triangular U.')
```

```
//end
```

Scilab code Exa 1.5.1 Triangular factorization

```
//page 34
clear;
close;
clc;
A=[1 2;3 8];
disp(A,'A=')
[L,U]=lu(A);
disp(L,'L=');
disp(U,'U=')
disp('LU=')
disp(L*U)
disp('This shows that LU=A')
//end
```

Scilab code Exa 1.5.2 To check LU equals to A

```
//page 34
clear;
close;
clc;
A=[0 2;3 4];
disp(A,'A=')
disp('Here this cannot be factored into A=LU,(Needs
   a row exchange)');
//end
```

Scilab code Exa 1.5.3 To check LU equals to A

```
//page 34
clear;
close;
clc;
disp('Given Matrix:')
A=[1 1 1;1 2 2;1 2 3];
disp(A,'A=');
[L,U]=lu(A);
disp(L,'L=');
disp(U,'U=');
disp(L*U,'LU=');
disp('Here LU=A,from A to U there are subtraction of
    rows.Frow U to A there are additions of rows');
//end
```

Scilab code Exa 1.5.4 If U equals to I then L equals to A

```
//page 34
clear;
close;
clc;
a=rand(1);
b=rand(1);
c=rand(1);
L=[1 0 0;a 1 0;b c 1];
disp(L,'L=');
U=eye(3,3);
disp(U,'U=');
E=[1 0 0;-a 1 0;0 0 1];
disp(E,'E=');
F=[1 0 0;0 1 0;-b 0 1];
disp(F,'F=');
G=[1 0 0;0 1 0;0 -c 1];
```

```
disp(G,'G=');
disp('A=inv(E)*inv(F)*inv(G)*U')
A=inv(E)*inv(F)*inv(G)*U;
disp(A,'A=');
disp('When U is identity matrix then L is same as A'
   );
//end
```

Scilab code Exa 1.5.5 Spilting A to L and U

```
//page 39
clear;
close;
clc;
A=[1 -1 0 0 ;-1 2 -1 0;0 -1 2 -1;0 0 -1 2];
disp(A,'A=');
[L,U]=lu(A);
disp(U,'U=');
disp(L,'L=');
disp('This shows how a matrix A with 3 diagnols has
   factors L and U with two diagnols.')
//end
```

Scilab code Exa 1.5.6 Solving for X using L and U

```
//page 36
clear;
close;
clc;
a=[1 -1 0 0;-1 2 -1 0;0 -1 2 -1;0 0 -1 2];
disp(a,'a=')
b=[1;1;1;1]
disp(b,'b=')
```

```
disp('Given Equation ,ax=b')
[L,U]=lu(a);
disp(U,'U=');
disp(L,'L=');
disp('Augmented Matrix of L and b=');
A=[L b];
disp(A)
c=zeros(4,1);
n=4;
//Algorithm Finding the value of c
c(1)=A(1,n+1)/A(1,1);
for i=2:n;
    sumk=0;
    for k=1:n-1
        sumk=sumk+A(i,k)*c(k);
    end
    c(i)=(A(i,n+1)-sumk)/A(i,i)
end
disp(c,'c=')
x=zeros(4,1);
disp('Augmented matrix of U and c=')
B=[U c];
disp(B)
//Algorithm for finding value of x
x(n)=B(n,n+1)/B(n,n);
for i=n-1:-1:1;
    sumk=0;
    for k=i+1:n
        sumk=sumk+B(i,k)*x(k);
    end
    x(i)=(B(i,n+1)-sumk)/B(i,i);
end
disp(x,'x=')
//end
```

Scilab code Exa 1.5.7 Elimination in a nutshell

```
//page 39
clear;
close;
clc;
A=[1 1 1;1 1 3;2 5 8];
disp(A,'A=');
[L,U,P]=lu(A);
disp(L,'L=');
disp(U,'U=');
disp(P,'P=');
disp(P*A,'PA=')
disp(L*U,'LU=')
disp('This shows that PA is the same as LU')
//end
```

Scilab code Exa 1.6.1 Gauss Jordon method

```
//page 47
clear;
close;
clc;
disp('Given matrix:')
A=[2 1 1;4 -6 0;-2 7 2];
disp(A);
[n,m]=size(A);
disp('Augmented matrix :')
a=[A eye(n,m)];
disp(a)
disp('R2=R2-2*R1,R3=R3-(-2)*R1');
a(2,:)=a(2,:)-2*a(1,:);
a(3,:)=a(3,:)-(-1)*a(1,:);
disp(a)
disp('R3=R3-(-1)*R2');
```

```
a(3,:)=a(3,:)-(-1)*a(2,:);
disp(a,'a=')
disp(a,'[U inv(L)] :')
disp('R2=R2-(-2)*R3,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)
a(1,:)=a(1,:)/a(1,1);
a(2,:)=a(2,:)/a(2,2);
disp('[I inv(A)]:')
a(3,:)=a(3,:)/a(3,3);
disp(a);
disp('inv(A):')
a(:,4:6);
disp(a(:,4:6))
```

Scilab code Exa 1.6.2 Symmetric products

```
//Caption :Symmetric Products
//Example:1.6.2-To Find the product of transpose(R)
   and R.
//page 51
clear;
close;
clc;
R=[1 2];
disp(R,'R=');
Rt=R';
disp(Rt,'Transpose of the given matrix is :')
disp(R*Rt,'The product of R & transpose(R)is :')
disp(Rt*R,'The product of transpose(R)& R is :' )
disp('Rt*R and R*Rt are not likely to be equal even
```

```
    if m==n.')
//end
```

Chapter 2

Vector Spaces

Scilab code Exa 2.1.1 Vector Spaces and subspaces

```
//page 70
clear;
close;
clc;
disp('Consider all vectors in R^2 whose components
   are positive or zero')
disp('The subset is first Quadrant of x-y plane,the
   co-ordinates satisfy x>=0 and y>=0.It is not a
   subspace.')
v=[1,1];
disp(v,'If the Vector=');
disp('Taking a scalar,c=-1')
c=-1; //scalar
disp(c*v,'c*v=')
disp('It lies in third Quadrant instead of first,
   Hence violating the rule(ii).')
//end
```

Scilab code Exa 2.1.2 Vector Spaces and subspaces

```
//page 71
clear;
close;
clc;
disp('Take vector space of 3X3 matrices')
disp('One possible subspace is the set of lower
   triangular matrices,Another is set of symmetric
   matrices')
disp('A+B,cA are both lower triangular if A and B
   are lower triangular ,and are symmetric if A and
   B are symmetric and Zero matrix is in both
   subspaces')
```

Scilab code Exa 2.3.1 Linear Independence

```
//page 92
clear;
close;
clc;
disp('For linear independence,C1V1+C2V2+......CkVk=0
   ')
disp('If we choose V1=zero vector,then the set is
   linearly dependent.We may choose C1=3 and all
   other Ci=0;this is a non-trival solution that
   produces zero.')
//end
```

Scilab code Exa 2.3.2 Linear Independence

```
//page 92
clear;
close;
clc;
```

```
A=[1 3 3 2;2 6 9 5;-1 -3 3 0];
disp('Given matrix:')
disp(A)
B=A;
disp('C2->C2-3*C1')
A(:,2)=A(:,2)-3*A(:,1);
disp(A)
disp('Here,C2=3*C1,Therefore the columns are
    linearly dependent.')
disp('R3->R3-2*R2+5*R1')
B(3,:)=B(3,:)-2*B(2,:)+5*B(1,:);
disp(B)
disp('Here R3=R3-2*R2+5*R1,therefore the rows are
    linearly dependent.')
//end
```

Scilab code Exa 2.3.3 Linear Independence

```
clear;
close;
clc;
A=[3 4 2;0 1 5;0 0 2];
disp(A,'A=');
disp('The columns of the triangular matrix are
    linearly independent,it has no zeros on the
    diagonal');
//end
```

Scilab code Exa 2.3.4 Linear Independence

```
//page 93
clear;
close;
```

```
clc;
disp('The columns of the nxn identity matrix are
   independent.')
n=input('Enter n:');
I=eye(n,n);
disp(I,'I=');
//end
```

Scilab code Exa 2.3.5 Linear Independence

```
//page 93
clear;
close;
clc;
disp('Three columns in R2 cannot be independent.')
A=[1 2 1;1 2 3];
disp(A,'Given matrix:')
[L,U]=lu(A);
disp(U,'U=');
disp('If c3 is 1 ,then back-substitution Uc=0 gives
   c2=-1,c1=1,With these three weights,the first
   column minus the second plus the third equals
   zero ,therefore linearly dependent.')
```

Scilab code Exa 2.3.6 Linear Independence

```
//page 93
clear;
close;
clc;
disp('The vectors w1=(1,0,0),w2=(0,1,0),w3=(-2,0,0)
   span a plane (x-y plane) in R3. The first two
```

```
   vectors also span this plane, whereas w1 and w3
   span only a line.');
//end
```

Scilab code Exa 2.3.7 Linear Independence

```
//page 93
clear;
close;
clc;
disp('The column space of A is excatly the space
   that is spanned by its columns.The row space is
   spanned by the rows.The definition is made to
   order.Multiplying A by any x gives a combination
   of columns; it is a vector Ax in the column space
   . The coordinate vectors e_1,....e_n coming from
   the identity matrix span Rn. Every vector b=(b_1
   .....,b_n) is a combination of those columns.In
   this example the weights are the components b_i
   themselves:b=b_1e_1+.....+b_ne_n.But the columns
   of other matrices also span R_.')
//end
```

Scilab code Exa 2.3.8 Basis for a vector space

```
//page 93
clear;
close;
clc;
disp('Here,the vector v1 by itself is linearly
   independent , but it fails to span R2.The three
   vectors v1,v2,v3 certainly span R2, but are not
   independent. Any two of these vectors say v1 and
```

```
    v2 have both properties -they span and they are
    independent.So they form a basis.(A vector space
    does not have a unique basis)')
//end
```

Scilab code Exa 2.3.9 Basis for a vector space

```
//page 96
clear;
close;
clc;
disp('These four columns span the column space U,but
     they are not independent.')
U=[1 3 3 2;0 0 3 1;0 0 0 0];
disp(U,'U=');
disp('The columns that contains pivots (here 1st & 3
    rd) are a basis for the column space. These
    columns are independent, and it is easy to see
    that they span the space.In fact,the column space
     of U is just the x-y plane withinn R3. C(U) is
    not the same as the column space C(A) before
    elimination-but the number of independent columns
     did not change.')
```

Scilab code Exa 2.4.1 The four fundamental subspaces

```
//page 107
clear;
close;
clc;
A=[1 2;3 6];
disp(A,'A=');
[m,n]=size(A);
```

```
disp(m,'m=');
disp(n,'n=');
[v,pivot]=rref(A);
r=length(pivot);
disp(r,'rank=')
cs=A(:,pivot);
disp(cs,'Column space=');
ns=kernel(A);
disp(ns,'Null space=');
rs=v(1:r,:)';
disp(rs,'Row space=')
lns=kernel(A');
disp(lns,'Left null sapce=');
```

Scilab code Exa 2.4.2 Inverse of a mxn matrix

```
//page 108
clear;
close;
clc;
A=[4 0 0;0 5 0];
disp(A,'A=');
[m,n]=size(A);
disp(m,'m=');
disp(n,'n=')
r=rank(A);
disp(r,'rank=');
disp('since m=r=2 ,there exists a right inverse .');
C=A'*inv(A*A');
disp(C,'Best right inverse=')
//end
```

Scilab code Exa 2.5.1 Networks and discrete applied mathematics

```
//page 121
clear;
close;
clc;
disp('Applying current law A''y=f at nodes 1,2,3:')
A=[-1 1 0;0 -1 1; -1 0 1;0 0 -1;-1 0 0];
disp(A','A''=');
C=diag(rand(5,1)); //Taking some values for the resistances.
b=zeros(5,1);
b(3,1)=rand(1);//Taking some value of the battery.
f=zeros(3,1);
f(2,1)=rand(1);//Taking some value of the current source.
B=[b;f];
disp('The other equation is inv(C)y+Ax=b.The block form of the two equations is:')
C=[inv(C) A;A' zeros(3,3)];
disp(C);
X=['y1';'y2';'y3';'y4';'y5';'x1';'x2';'x3'];
disp(X,'X=')
X=C\B;
disp(X,'X=');
//end
```

Chapter 3

Orthogonality

Scilab code Exa 3.1.1 Orthogonal vectors

```
//page 143
clear;
close;
clc;
x1=[2;2;-1];
disp(x1,'x1=');
x2=[-1;2;2];
disp(x2,'x2=');
disp(x1'*x2,'x1''*x2=');
disp('Therefore,X1 is orthogonal to x2 .Both have
   length of sqrt(14).')
```

Scilab code Exa 3.1.2 Orthogonal vectors

```
//page 144
clear;
close;
clc;
```

```
disp('Suppose V is a plane spanned by v1=(1,0,0,0)
   and v2=(1,1,0,0).If W is the line spanned by w
   =(0,0,4,5),then w is orthogonal to both v''s.The
   line W will be orthogonal to the whole plane V.')
```

Scilab code Exa 3.1.3 Orthogonal vectors

```
//page 145
clear;
close;
clc;
A=[1 3;2 6;3 9];
disp(A,'A=');
ns=kernel(A);
disp(ns,'Null space=');
disp(A(1,:)*ns,'A(1,:)*ns=');
disp(A(2,:)*ns,'A(2,:)*ns=');
disp(A(3,:)*ns,'A(3,:)*ns=');
disp('This shows that the null space of A is
   orthogonal to the row space.');
//end
```

Scilab code Exa 3.2.1 Projections onto a line

```
//page 155
clear;
close;
clc;
b=[1;2;3];
disp(b,'b=');
a=[1;1;1];
disp(a,'a=')
x=(a'*b)/(a'*a)
```

```
disp(x*a,'Projection p of b onto the line through a
    is x^*a=');
disp((a'*b)/(sqrt(a'*a)*sqrt(b'*b)),'cos(thetha)=');
//end
```

Scilab code Exa 3.2.2 Projections onto a line

```
//page 156
clear;
close;
clc;
a=[1;1;1];
disp(a,'a=');
P=(a*a')/(a'*a);
disp(P,'Matrix that projects onto a line through a
    =(1,1,1) is');
//end
```

Scilab code Exa 3.2.3 Projections onto a line

```
//page 156
clear;
close;
clc;
thetha=45; //Taking some value for thetha
a=[cos(thetha);sin(thetha)];
disp(a,'a=');
P=(a*a')/(a'*a);
disp(P,'Projection of line onto the thetha-direction
    (thetha taken as 45) in the x-y plane passing
    through a is ');
//end
```

Scilab code Exa 3.3.1 Projection matrices

```
//page 165
clear;
close;
clc;
A=rand(4,4);
disp(A,'A=');
P=A*inv(A'*A)*A';
disp('P=A*inv(A''*A)*A');
disp(P,'Projection of a  invertible 4x4 matrix on to
    the whole space is:');
disp('Its identity matrix.')
//end
```

Scilab code Exa 3.3.2 Least squares fitting of data

```
//page 166
clear;
close;
clc;
disp('b=C+Dt');
disp('Ax=b');
A=[1 -1;1 1;1 2];
disp(A,'A=');
b=[1;1;3];
disp(b,'b=');
disp('If Ax=b could be solved then they would be no
   errors, they can''t be solved because the points
   are not on a line.Therefore they are solved by
   least squares.')
disp('so,A''Ax^=A''b');
```

```
x=zeros(1,2);
x=(A'*A)\(A'*b);
disp(x(1,1),'C^ =');
disp(x(2,1),'D^=');
disp('The best line is 9/7+4/7t')
//end
```

Scilab code Exa 3.4.1 Orthogonal matrices

```
//page 175
clear;
close;
clc;
thetha=45;//Taking some value for thetha.
Q=[cos(thetha) -sin(thetha);sin(thetha) cos(thetha)
   ];
disp(Q,'Q=');
disp(Q','Q''=inv(Q)=');
disp('Q rotates every vector through an angle thetha
   , and Q'' rotates it back through -thetha.The
   columns are clearly orthogonal and they are
   orthonormal because sin^2(theta)+cos^2(thetha)=1.
   ');
//end
```

Scilab code Exa 3.4.2 Orthogonal matrices

```
//page 175
clear;
close;
clc;
disp('Any permutation matrix is an orthogonal matrix
   .The columns are certainly unit vectors and
```

```
    certainly orthogonal−because the 1 appears in a
    differnt place in each column')
P=[0 1 0;0 0 1;1 0 0];
disp(P,'P=');
disp(P','inv(P)=P''=');
disp(P'*P,'And,P''*P=');
//end
```

Scilab code Exa 3.4.3 Projection onto a plane

```
//page 175
clear;
close;
clc;
disp('If we project b=(x,y,z) onto the x−y plane
    then its projection is p=(x,y,0),and is the sum
    of projection onto x− any y−axes.')
b=rand(3,1);
q1=[1;0;0];
disp(q1,'q1=');
q2=[0;1;0];
disp(q2,'q2=');
P=q1*q1'+q2*q2';
disp(P,'Overall projection matrix,P=');
disp('and,P[x;y;z]=[x;y;0]')
disp('Projection onto a plane=sum of projections
    onto orthonormal q1 and q2.')
//end
```

Scilab code Exa 3.4.4 Least squares fitting of data

```
//page 166
clear;
```

```
close;
clc;
disp('y=C+Dt');
disp('Ax=b');
A=[1 -3;1 0;1 3];
disp(A,'A=');
y=rand    (3,1);
disp(y,'y=');
disp('the columns of A are orthogonal,so')
x=zeros(1,2);
disp(([1 1 1]*y)/(A(:,1)'*A(:,1)),'C^ =');
disp(([-3 0 3]*y)/(A(:,2)'*A(:,2)),'D^ =')
disp('C^ gives the besy fit ny horizontal line, whereas D^t is the best fit by a straight line through the origin.')
//end
```

Scilab code Exa 3.4.5 Gram Schmidt process

```
//page 166
clear;
close;
clc;
A=[1 0 1;1 0 0;2 1 0];//independent vectors stored in columns of A
disp(A,'A=');
[m,n]=size(A);
for k=1:n
    V(:,k)=A(:,k);
    for j=1:k-1
        R(j,k)=V(:,j)'*A(:,k);
        V(:,k)=V(:,k)-R(j,k)*V(:,j);
    end
    R(k,k)=norm(V(:,k));
    V(:,k)=V(:,k)/R(k,k);
```

```
end
disp(V,'Q=')
```

Chapter 4

Determinants

Scilab code Exa 4.3.1 Determinant of a matrix is the product of its pivots

```
clear;
close;
clc;
n=input('Enter the value of n:');
for i=1
    for j=i;
        a(i,j)=2;
        a(i,j+1)=-1;
    end
end
for i=2:n-1
    for j=i
        a(i,j-1)=-1;
        a(i,j)=2;
        a(i,j+1)=-1;
    end
end
for i=n
    for j=i
        a(i,j-1)=-1;
        a(i,j)=2;
```

```
    end
end
disp(a,'a=');
[L,D,U]=lu(a)
determinant=1;
for i=1:n
        determinant=determinant*D(i,i);
end
disp(determinant,'Determinant=')
//end
```

Scilab code Exa 4.3.2 Calculation of determinant of a matrix by using cofactors

```
clear;
close;
clc;
disp('For a 3*3 matrix:')
disp('det A=a11(a22a33-a23a32)+a12(a23a31-a21a33)+
   a13(a21a32-a22a31)')
//end
```

Scilab code Exa 4.3.3 Calculation of determinant of a matrix by using cofactors

```
//page 214
clear;
close;
clc;
A=[2 -1 0 0;-1 2 -1 0;0 -1 2 -1;0 0 -1 2];
disp(A,'A=');
[m,n]=size(A)
a=A(1,:);
```

```
c=[];
for l=1:4
    B=A([1:0,2:4],[1:l-1,l+1:4]);
    c1l=(-1)^(1+l)*det(B);
    c=[c;c1l];
end
d=a*c;
disp(d)
```

Scilab code Exa 4.4.1 Inverse of a sum matrix is a difference matrix

```
//282
clear;
close;
clc;
A=[1 1 1;0 1 1;0 0 1];
disp(A,'A=')
n=size(A,1); d=1:n-1;
B=zeros(n); AA=[A,A;A,A]';
for j=1:n
   for k=1:n
       B(j,k)=det(AA(j+d,k+d));
   end
end
disp(B,'Adjoint of A:');
disp(B/det(A),'inv(A):');
//end
```

Scilab code Exa 4.4.2 Cramers rule

```
//page 222
clear;
close;
```

```
clc;
//x1+3x2=0
//2x1+4x2=6
A=[1 3;2 4];
b=[0;6];
X1=[0 3;6 4];
X2=[1 0;2 6];
disp(det(X1)/det(A),'x1=');
disp(det(X2)/det(A),'x2=');
//end
```

Chapter 5

Eigenvalues and Eigenvectors

Scilab code Exa 5.1.1 Eigenvalues and eigenvectors

```
//page 238
clear;
close;
clc;
A=[3 0;0 2];
eig=spec(A);
[V,Val]=spec(A);
disp(eig,'Eigen values:')
x1=V(:,1);
x2=V(:,2);
disp(x1,x2,'Eigen vectors:');
//end
```

Scilab code Exa 5.1.2 Eigenvalues and eigenvectors

```
//page 238
clear;
close;
```

```
clc;
disp('The eigen values of a projection matrix are 1
    or 0.')
P=[1/2 1/2;1/2 1/2];
eig=spec(P);
[V,Val]=spec(P);
disp(eig,'Eigen values:')
x1=V(:,1);
x2=V(:,2);
disp(x1,x2,'Eigen vectors:');
//end
```

Scilab code Exa 5.2.1 Diagonalization

```
//page 238
clear;
close;
clc;
A=[1/2 1/2;1/2 1/2];
[V,Val]=spec(A);
disp(Val,'Eigenvalue matrix:');
disp(V,'S=');
disp(A*V,'AS=S*eigenvaluematrix')
disp('Therefore inv(S)*A*S=eigenvalue matrix')
//end
```

Scilab code Exa 5.2.2 Diagonalization

```
//page 238
clear;
close;
clc;
```

```
disp('The eigenvalues themselves are not so clear
   for a rotation.')
disp('90 degree rotation')
K=[0 -1;1 0];
disp(K,'K=')
eig=spec(K);
[V,Val]=spec(K);
disp(eig,'Eigen values:')
x1=V(:,1);
x2=V(:,2);
disp(x1,x2,'Eigen vectors:');
//end
```

Scilab code Exa 5.2.3 Powers and Products

```
//page 249
clear;
close;
clc;
disp('K is rotation through 90 degree,then K^2 is
   rotation through 180 degree and inv(k is rotation
    through -90 degree)')
K=[0 -1;1 0];
disp(K,'K=')
disp(K*K,'K^2=')
disp(K*K*K,'K^3=')
disp(K^4,'K^4=')
[V,D]=spec(K);
disp('K^4 is a complete rotation through 360 degree.
   ')
disp(D,'Eigen value matrix,D of K:');
disp(D^4,'and also D^4=')
//end
```

Scilab code Exa 5.3.1 Difference equations

```
//page 249
clear;
close;
clc;
A=[0 4;0 1/2];
disp(A,'A=');
eig=spec(A);
disp(eig,'Eigen values:')
[v,D]=spec(A);
u0=[v(:,1)];//Taking u0 as the 1st eigen Vector.
for k=0:5
    disp(k,'k=');
    u=A*u0;
    disp(u,'U(k+1)(K from 0 to 5)')
    u0=u;
end
u0=[v(:,2)];//Taking u0 as the 2nd eigen vector.
for k=0:5
    disp(k,'k=');
    u=A*u0;
    disp(u,'U(k+1)=')
    u0=u;
end
```

Scilab code Exa 5.5.1 Complex matrices

```
//page282
clear;
close;
clc;
```

```
i=sqrt(-1);
x=3+4*i;
disp(x,'x=');
x_=conj(x);
disp(x*x_,'xx_=');
r=sqrt(x*x_);
disp(r,'r=')
//end
```

Scilab code Exa 5.5.2 Inner product of a complex matrix

```
//282
clear;
close;
clc;
i=sqrt(-1);
x=[1 i]';
y=[2+1*i 2-4*i]';
disp(x'*x,'Length of x squared:');
disp(y'*y,'Length of y squared:');
//end
```

Chapter 6

Positive Definite Matrices

Scilab code Exa 6.1.1 Definite versus indefinite

```
//313
clear;
close;
clc;
disp('f(x,y)=x^2-10*x*y+y^2');
a=1;
c=1;
deff('[f]=f(x,y)','f=x^2-10*x*y+y^2');
disp(f(1,1),'f(1,1)=');
disp('The conditions a>0 and c>0 ensure that f(x,y)
    is positive on the x and y axes. But this
    function is negative on the line x=y,because b
    =-10 overwhelms a and c. ');
//end
```

Scilab code Exa 6.1.3 Maxima Minima And Saddle points

```
//315
```

```
clear;
close;
clc;
disp('f(x,y)=2*x^2+4*x*y+y^2');
A=[2 2;2 1];
a=1;
c=1;
b=2;
disp(a*c,'ac=');
disp(b^2,'b^2=');
disp('Saddle point,as ac<b^2');
```

Scilab code Exa 6.1.4 Maxima Minima And Saddle points

```
//315
clear;
close;
clc;
disp('f(x,y)=2*x^2+4*x*y+y^2');
A=[2 2;2 1];
a=0;
c=0;
b=1;
disp(a*c,'ac=');
disp(b^2,'b^2=');
disp('Saddle point,as ac<b^2');
```

Scilab code Exa 6.2.2 Maxima Minima And Saddle points

```
//313
clear;
close;
clc;
```

```
disp('f(x,y)=x^2+4*x*y+y^2');
a=1;
c=1;
deff('[f]=f(x,y)','f=x^2+4*x*y+y^2');
disp(f(0,0),'f(0,0)=')
disp('Here 2b=4  it still does not ensure a minimum
    ,the sign of b is of no importance.Neither F nor
    f has a minimum at(0,0) because f(1,-1)=-1.')
//end
```

Scilab code Exa 6.3.1 Singular value decomposition

```
//332
clear;
close;
clc;
A=[-1 2 2]';
disp(A,'A=');
[U diagnol V]=svd(A);
disp(U,'U=');
disp(diagnol,'diagnol=');
disp(V','V''=');
disp(U*diagnol*V','A=U*diagnol*V''')
//end
```

Scilab code Exa 6.3.2 Singular value decomposition

```
//332
clear;
close;
clc;
A=[-1 1 0;0 -1 1];
disp(A,'A=');
```

```
[U diagnl V]=svd(A);
disp(U,'U=');
disp(diagnl,'Diagonal=');
disp(V','V''=');
disp(U*diagnl*V','A=U*diagonal*V''=')
//end
```

Scilab code Exa 6.3.3 Polar decomposition

```
//332
clear;
close;
clc;
A=[1 -2;3 -1];
disp(A,'A=');
[U S V]=svd(A);
Q=U*V';
S=V*S*V';
disp(Q,'Q=');
disp(S,'S=');
disp(Q*S,'A=SQ=')
//end
```

Scilab code Exa 6.3.4 Reverse polar decomposition

```
//332
clear;
close;
clc;
A=[1 -2;3 -1];
disp(A,'A=');
[U diag1 V]=svd(A);
Q=U*V';
```

```
S=[2 1;1 3];
disp(Q,'Q=');
disp(S,'S=')
disp(S'*Q,'A=S''Q=')
//end
```

Chapter 7

Computations with Matrices

Scilab code Exa 7.4.1 Jacobi Method

```
//page 238
clear;
close;
clc;
A=[2 -1;-1 2];
S=[2 0;0 2];
T=[0 1;1 0];
p=inv(S)*T;
b=[2 2]';
x=zeros(2,1);
disp(x,'intial v & w:')
x_1=zeros(1,2);
for k=0:25
    x_1=p*x+inv(S)*b;
    x=x_1;
    disp(k,'k=')
    disp(x_1,'v(k+1) & w(k+1)=');
end
```

Scilab code Exa 7.4.2 Gauss Seidel method

```
//page 238
clear;
close;
clc;
A=[2 -1;-1 2];
S=[2 0;-1 2];
T=[0 1;0 0];
b=rand(2,1);
p=inv(S)*T;
x=zeros(2,1);
disp(x,'intial v & w:')
x_1=zeros(1,2);
for k=0:25
    x_1=p*x+inv(S)*b;
    x=x_1;
    disp(k,'k=')
    disp(x_1,'v(k+1) & w(k+1)=');
end
```

Chapter 8

Linear Programming and Game Theory

Scilab code Exa 8.2.2 Minimize cx subject to x greater than or equal to zero and Ax equals to b

```
//page 238
clear;
close;
clc;
A=[1 0 1 6 2;0 1 1 0 3];
b=[8 9]';
c=[0 0 7 -1 -3]';
lb=[0 0 0 0 0]'
ub=[];
[x,lagr,f]=linpro(c,A,b,lb,ub);
disp(x,'New corner:');
disp(f,'Minimum cost:');
//end
```