# Scilab Textbook Companion for Linear Algebra by K. Hoffman and R. Kunze[1]

Created by
Gagandeep Kaur
B.Tech
Computer Engineering
Rajasthan Technical University
College Teacher
None
Cross-Checked by
Spandana

July 31, 2019

[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

**Author:** K. Hoffman and R. Kunze

**Publisher:** Prentice-Hall (India)

**Edition:** 2

**Year:** 1986

**ISBN:** 978-0135367971

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

**Scilab code Exa 1.5** Elementary Row Operations

```
//page 8
//Example 1.5
clear;
close;
clc;
a = [2 -1 3 2; 1 4 0 -1; 2 6 -1 5];
disp(a,'a=');
disp('Applying row transformations:');
disp('R1 = R1-2*R2');
a(1,:) = a(1,:) - 2*a(2,:);
disp(a,'a = ');
disp('R3 = R3-2*R2');
a(3,:) = a(3,:) - 2*a(2,:);
disp(a,'a = ');
disp('R3 = R3/-2');
a(3,:) = -1/2*a(3,:);
disp(a,'a = ');
disp('R2 = R2-4*R3');
a(2,:) = a(2,:) - 4*a(3,:);
disp(a,'a = ');
disp('R1 = R1+9*R3');
```

```
a(1,:) = a(1,:) + 9*a(3,:);
disp(a,'a = ');
disp('R1 = R1*2/15');
a(1,:) = a(1,:) * 2/15;
disp(a,'a = ');
disp('R2 = R2+2*R1');
a(2,:) = a(2,:) + 2*a(1,:);
disp(a,'a = ');
disp('R3 = R3-R1/2');
a(3,:) = a(3,:) - 1/2*a(1,:);
disp(a,'a = ');
disp('We get the system of equations as:');
disp('2*x1 - x2 + 3*x3 + 2*x4 = 0');
disp('x1 + 4*x2 - x4 = 0');
disp('2*x1 + 6* x2 - x3 + 5*x4 = 0');
disp('and');
disp('x2 - 5/3*x4 = 0','x1 + 17/3*x4 = 0','x3 -
   11/3*x4 = 0');
disp('now by assigning any rational value c to x4 in
    system second, the solution is evaluated as:');
disp('(-17/3*c,5/3,11/3*c,c)');
//end
```

**Scilab code Exa 1.6** Elementary Row Operations

```
//page 9
//Example 1.6
clear;
close;
clc;
a=[-1 %i;-%i 3;1 2];
disp(a,'a = ');
disp('Applying row transformations:');
disp('R1 = R1+R3 and R2 = R2 + i *R3');
a(1,:) = a(1,:) +a(3,:);
```

```
a(2,:) = a(2,:) + %i * a(3,:);
disp(a,'a = ');
disp('R1 = R1 * (1/2+i)');
a(1,:) = 1/(2 + %i) * a(1,:);
disp(a,'a = ');
disp('R2 = R2-R1*(3+2i) and R3 = R3 - 2 *R1');
a(2,:) = round(a(2,:) - (3 + 2 * %i) * a(1,:));
a(3,:) = round(a(3,:) - 2 * a(1,:));
disp(a,'a = ');
disp('Thus the system of equations is:');
disp('x1 + 2*x2 = 0','-i*x1 + 3*x2 = 0','-x1+i*x2 =
    0');
disp('It has only trivial solution x1 = x2 = 0');
//end
```

**Scilab code Exa 1.7** Row reduced echelon Matrix

```
//page 9
//Example 1.7
clear;
close;
clc;
n = rand();
n = round(n*10);
disp(eye(n,n));
printf('This is an Identity matrix of order %d * %d'
    ,n,n);
disp('And It is a row reduced matrix.');
//end
```

**Scilab code Exa 1.8** Row reduced echelon Matrix

```
//page 12
```

```
//Example 1.8
clear;
close;
clc;
n = rand();
n = round(n*10);
disp(eye(n,n));
printf('This is an Identity matrix of order %d * %d'
    ,n,n);
disp('And It is a row reduced matrix.');
m = rand();
n = rand();
m = round(m*10);
n = round(n*10);
disp(zeros(m,n));
printf('This is an Zero matrix of order %d * %d',m,n
    );
disp('And It is also a row reduced matrix.');
a = [0 1 -3 0 1/2;0 0 0 1 2;0 0 0 0 0];
disp(a,'a = ');
disp('This is a non-trivial row reduced matrix.');
//end
```

**Scilab code Exa 1.9** System of Equations

```
//page 14
//Example 1.9
clear;
close;
clc;
A = [1 -2 1;2 1 1;0 5 -1];
disp(A,'A = ');
disp('Applying row transformations:');
disp('R2 = R2 - 2*R1');
A(2,:) = A(2,:) - 2*A(1,:);
```

```
disp(A,'A = ');
disp('R3 = R3 − R2');
A(3,:) = A(3,:) - A(2,:);
disp(A,'A = ');
disp('R2 = 1/5*R2');
A(2,:) = 1/5*A(2,:);
disp(A,'A = ');
disp('R1 = R1 − 2*R2');
A(1,:) = A(1,:) + 2*A(2,:);
disp(A,'A = ');
disp('The condition that the system have a solution
    is:');
disp('2*y1 − y2 + y3 = 0');
disp('where, y1,y2,y3 are some scalars');
disp('If the condition is satisfied then solutions
    are obtained by assigning a value c to x3');
disp('Solutions are:');
disp('x2 = 1/5*c + 1/5*(y2 − 2*y1)','x1 = −3/5*c +
    1/5*(y1 + 2*y2)');
//end
```

**Scilab code Exa 1.10** Product of Matrices

```
//page 17
//Example 1.10
clear;
close;
clc;
//Part a
a = [1 0;-3 1];
b = [5 -1 2;15 4 8];
disp(a,'a=');
disp(b,'b=');
disp(a*b,'ab = ');
disp('
```

```
--------------------------------------------------------------------------------
');
//Part b
a = [1 0;-2 3;5 4;0 1];
b = [0 6 1;3 8 -2];
disp(a,'a=');
disp(b,'b=');
disp(a*b,'ab = ');
disp('--------------------------------------------------------------------------------');
//Part c
a = [2 1;5 4];
b = [1;6];
disp(a,'a=');
disp(b,'b=');
disp(a*b,'ab = ');
disp('--------------------------------------------------------------------------------');
//Part d
a = [-1;3];
b = [2 4];
disp(a,'a=');
disp(b,'b=');
disp(a*b,'ab = ');
disp('--------------------------------------------------------------------------------');
//Part e
a = [2 4];
b = [-1;3];
disp(a,'a=');
disp(b,'b=');
disp(a*b,'ab = ');
disp('--------------------------------------------------------------------------------');
```

```
//Part f
a = [0 1 0;0 0 0;0 0 0];
b = [1 -5 2;2 3 4;9 -1 3];
disp(a,'a=');
disp(b,'b=');
disp(a*b,'ab = ');
disp('
--------------------------------------------------------------------------------
');
//Part g
a = [1 -5 2;2 3 4;9 -1 3];
b = [0 1 0;0 0 0;0 0 0];
disp(a,'a=');
disp(b,'b=');
disp(a*b,'ab = ');
//end
```

**Scilab code Exa 1.14** Inverse of a matrix

```
//page 22
//Example 1.14
clear;
close;
clc;
a = [0 1;1 0];
disp(a,'a = ');
disp(inv(a),'inverse a = ');
//end
```

**Scilab code Exa 1.15** Inverse of a matrix

```
//page 25
//Example 1.15
```

```
clear;
close;
clc;
a = [2 -1;1 3];
disp(a,'a = ');
b = a;                 //Temporary variable to store a
disp('Applying row tranformations');
disp('Interchange R1 and R2');
x = a(1,:);
a(1,:) = a(2,:);
a(2,:) = x;
disp(a,'a = ');
disp('R2 = R2 - 2 * R1');
a(2,:) = a(2,:) - 2 * a(1,:);
disp(a,'a = ');
disp('R2 = R2 *1/(-7)');
a(2,:) = (-1/7) * a(2,:);
disp(a,'a = ');
disp('R1 = R1 - 3 * R2');
a(1,:) = a(1,:) - 3 * a(2,:);
disp(a,'a = ');
disp('Since a  has become an identity matrix. So, a
   is invertible');
disp('inverse of a = ');
disp(inv(b)); //a was stored in b
//end
```

---

**Scilab code Exa 1.16** Inverse of a matrix

```
//page 25
//Example 1.16
clear;
close;
clc;
a = [1 1/2 1/3;1/2 1/3 1/4;1/3 1/4 1/5];
```

```
disp(a,'a = ');
b = eye(3,3);
disp(b,'b = ');
disp('Applying row transformations on a and b
    simultaneously,');
disp('R2 = R2 - 1/2 * R1 and R3 = R3 - 1/3*R1');
a(2,:) = a(2,:) - 1/2 * a(1,:);
a(3,:) = a(3,:) - 1/3 * a(1,:);
b(2,:) = b(2,:) - 1/2 * b(1,:);
b(3,:) = b(3,:) - 1/3 * b(1,:);
disp(a,'a = ');
disp(b,'b = ');
disp('R3 = R3 - R2');
a(3,:) = a(3,:) -  a(2,:);
b(3,:) = b(3,:) -  b(2,:);
disp(a,'a = ');
disp(b,'b = ');
disp('R2 = R2 * 12 and R3 = R3 * 180');
a(2,:) = a(2,:) *12;
a(3,:) = a(3,:) * 180;
b(2,:) = b(2,:) * 12;
b(3,:) = b(3,:) * 180;
disp(a,'a = ');
disp(b,'b = ');
disp('R2 = R2 - R3 and R1 = R1 - 1/3*R3');
a(2,:) = a(2,:) - a(3,:);
a(1,:) = a(1,:) - 1/3 * a(3,:);
b(2,:) = b(2,:) - b(3,:);
b(1,:) = b(1,:) - 1/3 * b(3,:);
disp(a,'a = ');
disp(b,'b = ');
disp('R1 = R1 - 1/2 * R2');
a(1,:) = a(1,:) -  1/2 * a(2,:);
b(1,:) = b(1,:) -  1/2 * b(2,:);
disp(round(a),'a = ');
disp(b,'b = ');
disp('Since, a = identity matrix of order 3*3. So, b
     is inverse of a');
```

```
disp(b,'inverse(a) = ');
//end
```

# Chapter 2

# Vector Spaces

**Scilab code Exa 2.8** Vector Subspace

```
//page 37
//Example 2.8
clear;
clc;
close;
a1 = [1 2 0 3 0];
a2 =[0 0 1 4 0];
a3 = [0 0 0 0 1];
disp(a1,'a1 = ');
disp(a2,'a2 = ');
disp(a3,'a3 = ');
disp('By theorem 3, vector a is in subspace W of F^5
    spanned by a1, a2, a3');
disp('if and only if there exist scalars c1, c2, c3
   such that');
disp('a= c1a1 + c2a2 + c3a3');
disp('So, a = (c1,2*c1,c2,3c1+4c2,c3)');
c1 = -3;
c2 = 1;
c3 = 2;
a = c1*a1 + c2*a2 + c3*a3;
```

```
disp(c1,'c1 = ');
disp(c2,'c2 = ');
disp(c3,'c3 = ');
disp(a,'Therefore, a = ');
disp('This shows, a is in W');
disp('And (2,4,6,7,8) is not in W as there is no
    value of c1 c2 c3 that satisfies the equation');
//end
```

**Scilab code Exa 2.10** `Row space of matrix`

```
//page 38
//Example 2.10
clear;
clc;
close;
A = [1 2 0 3 0;0 0 1 4 0;0 0 0 0 1];
disp(A,'A = ');
disp('The subspace of F^5 spanned by a1 a2 a3(row
    vectors of A) is called row space of A.');
a1 = A(1,:);
a2 = A(2,:);
a3 = A(3,:);
disp(a1,'a1 = ');
disp(a2,'a2 = ');
disp(a3,'a3 = ');
disp('And, it is also the row space of B.');
B = [1 2 0 3 0;0 0 1 4 0;0 0 0 0 1;-4 -8 1 -8 0];
disp(B,'B = ');
//end
```

**Scilab code Exa 2.11** `Space of polynomial function`

```
//page 39
//Example 2.11
clear;
clc;
close;
disp('V is the space of all polynomial functions
   over F.');
disp('S contains the functions as:')
x = poly(0,"x");
n = round(rand()*10);
disp(n,'n = ');
for i = 0 : n
    f = x^i;
    printf('f%d(x) = ',i);
    disp(f);
end
disp('Then, V is the subspace spanned by set S.');
//end
```

**Scilab code Exa 2.12** Linear Dependency

```
//page 41
//Example 2.12
clear;
clc;
close;
a1 = [3 0 -3];
a2 = [-1 1 2];
a3 = [4 2 -2];
a4 = [2 1 1];
disp(a1, 'a1 = ');
disp(a2, 'a2 = ');
disp(a3, 'a3 = ');
disp(a4, 'a4 = ');
t = 2 * a1 + 2 * a2 - a3 + 0 * a4;
```

```
disp(' = 0',t,' Since, 2 * a1 + 2 * a2 - a3 + 0 * a4
    = ');
disp('a1,a2,a3,a4 are linearly independent');
e1 = [1 0 0];
e2 = [0 1 0];
e3 = [0 0 1];
disp(e1, 'Now, e1 = ');
disp(e2, 'e2 = ');
disp(e3, 'e3 = ');
disp('Also, e1,e2,e3 are linearly independent.');
//end
```

**Scilab code Exa 2.13** `Standard basis of Matrix`

```
//page 41
//Example 2.13
clear;
clc;
close;
disp('S is the subset of F^n consisting of n vectors
    .');
n = round(rand() *10 + 1);
disp(n,'n = ');
I = eye(n,n);
for i = 0 : n-1
    e = I(i+1,:);
    printf('e%d = ',i+1);
    disp(e);
end
disp('x1,x2,x3...xn are the scalars in F');
disp('Putting a = x1*e1 + x2*e2 + x3*e3 + .... + xn*
    en');
disp('So, a = (x1,x2,x3,...,xn)');
disp('Therefore, e1,e2..,en span F^n');
disp('a = 0 if x1 = x2 = x3 = .. = xn = 0');
```

```
disp('So,e1,e2,e3,..,en are linearly independent.');
disp('The set S = {e1,e2,..,en} is called standard
    basis of F^n');
//end
```

**Scilab code Exa 2.20** Inverse of a matrix

```
//page 54
//Example 2.20
clear;
clc;
close;
P = [-1 4 5; 0 2 -3; 0 0 8];
disp(P,'P = ');
disp(inv(P),'inverse(P) = ');
a1 = P(:,1);
a2 = P(:,2);
a3 = P(:,3);
disp('The vectors forming basis of F^3 are a1'', a2'
    ', a3''');
disp(a1','a1'' = ');
disp(a2','a2'' = ');
disp(a3','a3'' = ');
disp('The coordinates x1'',x2'',x3'' of vector a = [
    x1,x2,x3] is given by inverse(P)*[x1; x2; x3]');
t = -10*a1 - 1/2*a2 - a3;
disp(t,'And, -10*a1'' - 1/2*a2'' - a3'' = ');
//end
```

**Scilab code Exa 2.21** Standard basis of matrix

```
//page 60
//Example 2.21
```

```
clear;
clc;
close;
a1 = [1 2 2 1];
a2 = [0 2 0 1];
a3 = [-2 0 -4 3];
disp('Given row vectors are:');
disp(a1,'a1 = ');
disp(a2,'a2 = ');
disp(a3,'a3 = ');
disp('The matrix A from these vectors will be:');
A = [a1; a2; a3];
disp(A,'A = ');
disp('Finding Row reduced echelon matrix of A that is given by R');
disp('And applying same operations on identity matrix Q such that R = QA');
Q = eye(3,3);
disp(Q,'Q = ');
T = A;                  //Temporary matrix to store A
disp('Applying row transformations on A and Q,we get');
disp('R1 = R1-R2');
A(1,:) = A(1,:) - A(2,:);
Q(1,:) = Q(1,:) - Q(2,:);
disp(A,'A = ');
disp(Q,'Q = ');
disp('R3 = R3 + 2*R1');
A(3,:) = A(3,:) + 2*A(1,:);
Q(3,:) = Q(3,:) + 2*Q(1,:);
disp(A,'A = ');
disp(Q,'Q = ');
disp('R3 = R3/3');
A(3,:) = 1/3*A(3,:);
Q(3,:) = 1/3*Q(3,:);
disp(A,'A = ');
disp(Q,'Q = ');
disp('R2 = R2/2');
```

```
A(2,:) = 1/2*A(2,:);
Q(2,:) = 1/2*Q(2,:);
disp(A,'A = ');
disp(Q,'Q = ');
disp('R2 = R2 - 1/2*R3');
A(2,:) = A(2,:) - 1/2*A(3,:);
Q(2,:) = Q(2,:) - 1/2*Q(3,:);
disp(A,'A = ');
disp(Q,'Q = ');
R = A;
A = T;
disp('Row reduced echelon matrix:');
disp(R,'R = ');
disp(Q,'Q =');
//part a
disp(rank(R),'rank of R = ');
disp('Since, Rank of R is 3, so a1, a2, a3 are independent');
//part b
disp('Now, basis for W can be given by row vectors of R i.e. p1,p2,p3');
disp('b is any vector in W. b = [b1 b2 b3 b4]');
disp('Span of vectors p1,p2,p3 consist of vector b with b3 = 2*b1');
disp('So,b = b1p1 + b2p2 + b4p3');
disp('And,[p1 p2 p3] = R = Q*A');
disp('So, b = [b1 b2 b3]* Q * A');
disp('hence, b = x1a1 + x2a2 + x3a3 where x1 = [b1 b2 b4] * Q(1) and so on'); //Equation 1
//part c
disp('Now, given 3 vectors a1'' a2'' a3'':');
c1 = [1 0 2 0];
c2 = [0 2 0 1];
c3 = [0 0 0 3];
disp(c1,'a1'' = ');
disp(c2,'a2'' = ');
disp(c3,'a3'' = ');
disp('Since a1'' a2'' a3'' are all of the form (y1
```

```
y2 y3 y4) with y3 = 2*y1, hence they are in W.');
disp('So, they are independent.');
//part d
c = [c1; c2; c3];
P = eye(3,3);
for i = 1:3
    b1 = c(i,1);
    b2 = c(i,2);
    b4 = c(i,4);
    x1 = [b1 b2 b4] * Q(:,1);
    x2 = [b1 b2 b4]*Q(:,2);
    x3 = [b1 b2 b4]*Q(:,3);
   P(:,i) = [x1; x2; x3];
end
disp('Required matrix P such that X = PX'' is:');
disp(P,'P = ');
//end
```

---

**Scilab code Exa 2.22** Standard basis of Matrix

```
//page 63
//Example 2.22
clear;
clc;
close;
A = [1 2 0 3 0;1 2 -1 -1 0;0 0 1 4 0;2 4 1 10 1;0 0 0 0 1];
disp(A,'A = ');
//part a
T = A;                        //Temporary storing A in T
disp('Taking an identity matrix P:');
P = eye(5,5);
disp(P,'P = ');
disp('Applying row transformations on P and A to get a row reduced echelon matrix R:');
```

```
disp('R2 = R2 - R1 and R4 = R4 - 2* R1');
A(2,:) = A(2,:) - A(1,:);
P(2,:) = P(2,:) - P(1,:);
A(4,:) = A(4,:) - 2 * A(1,:);
P(4,:) = P(4,:) - 2 * P(1,:);
disp(A,'A = ');
disp(P,'P = ');
disp('R2 = -R2 , R3 = R3 - R1 + R2 and R4  = R4 - R1
    + R2');
A(2,:) = -A(2,:);
P(2,:) = -P(2,:);
A(3,:) = A(3,:) - A(2,:);
P(3,:) = P(3,:) - P(2,:);
A(4,:) = A(4,:) - A(2,:);
P(4,:) = P(4,:) - P(2,:);
disp(A,'A = ');
disp(P,'P = ');
disp('Mutually interchanging R3, R4 and R5');
x = A(3,:);
A(3,:) = A(5,:);
y = A(4,:);
A(4,:) = x;
A(5,:) = y - A(3,:);
x = P(3,:);
P(3,:) = P(5,:);
y = P(4,:);
P(4,:) = x;
P(5,:) = y - P(3,:);
R = A;
A = T;
disp(R,'Row reduced echelon matrix R = ');
disp(P,'Invertible Matrix P = ');
disp('Invertible matrix P is not unique. There can
   be many that depends on operations used to reduce
    A');
disp('----------------------------------------------');
//part b
disp('For the basis of row space W of A, we can take
```

```
    the non-zero rows of R');
disp('It can be given by p1, p2, p3');
p1 = R(1,:);
p2 = R(2,:);
p3 = R(3,:);
disp(p1,'p1 = ');
disp(p2,'p2 = ');
disp(p3,'p3 = ');
disp('-----------------------------------------');
//part c
disp('The row space W consists of vectors of the
    form:');
disp('b = c1p1 + c2p2 + c3p3');
disp('i.e. b = (c1,2*c1,c2,3*c1+4*c2,c3) where, c1
    c2 c3 are scalars.');
disp('So, if b2 = 2*b1 and b4 = 3*b1 + 4*b3  =>  (b1
    ,b2,b3,b4,b5) = b1p1 + b3p2 + b5p3');
disp('then,(b1,b2,b3,b4,b5) is in W');
disp('-----------------------------------------');
//part d
disp('The coordinate matrix of the vector (b1,2*b1,
    b2,3*b1+4*b2,b3) in the basis (p1,p2,p3) is
    column matrix of b1,b2,b3 such that:');
disp('  b1');
disp('  b2');
disp('  b3');
disp('-----------------------------------------');
//part e
disp('Now, to write each vector in W as a linear
    combination of rows of A:');
disp('Let b = (b1,b2,b3,b4,b5) and if b is in W,
    then');
disp('we know,b = (b1,2*b1,b3,3*b1 + 4*b3,b5)  =>  [
    b1,b3,b5,0,0]*R');
disp('=> b = [b1,b3,b5,0,0] * P*A  =>  b = [b1+b3,-
    b3,0,0,b5] * A');
disp('if b = (-5,-10,1,-11,20)');
b1 = -5;
```

```
b2 = -10;
b3 = 1;
b4 = -11;
b5 = 20;
x = [b1 + b3,-b3,0,0,b5];
disp(']',A,'[','*',')',x,'(','b = ');
disp('------------------------------------------------------------');
//part f
disp('The equations in system RX = 0 are given by R * [x1 x2 x3 x4 x5]');
disp('i.e., x1 + 2*x2 + 3*x4');
disp('x3 + 4*x4');
disp('x5');
disp('so, V consists of all columns of the form');
disp('[','X=');
disp('  -2*x2 - 3*x4');
disp('  x2');
disp('  -4*x4');
disp('  x4');
disp('  0');
disp('where x2 and x4  are arbitrary',']');
disp('------------------------------------------------------------');
//part g
disp('Let x2 = 1,x4 = 0 then the given column forms a basis of V');
x2 = 1;
x4 = 0;
disp([-2*x2-3*x4; x2; -4*x4; x4; 0]);
disp('Similarly, if x2 = 0,x4 = 1 then the given column forms a basis of V');
x2 = 0;
x4 = 1;
disp([-2*x2-3*x4; x2; -4*x4; x4; 0]);
disp('------------------------------------------------------------');
//part h
disp('The equation AX = Y has solutions X if and only if');
disp('-y1 + y2 + y3 = 0');
```

```
disp('-3*y1 + y2 + y4 -y5 = 0');
disp('where, Y = (y1 y2 y3 y4 y5)');
//end
```

# Chapter 3

# Linear Transformations

**Scilab code Exa 3.6** Linear Transformation function

```
//page 70
//Example 3.6
clc;
clear;
close;
a1 = [1 2];
a2 = [3 4];
disp(a1,'a1 = ');
disp(a2,'a2 = ');
disp('a1 and a2 are linearly independent and hence
   form a basis for R^2');
disp('According to theorem 1, there is a linear
   transformation from R^2 to R^3 with the
   transformation functions as:');
Ta1 = [3 2 1];
Ta2 = [6 5 4];
disp(Ta1,'Ta1 = ');
disp(Ta2,'Ta2 = ');
disp('Now, we find scalars c1 and c2 for that we
   know T(c1a1 + c2a2) = c1(Ta1) + c2(Ta2))');
disp('if(1,0) = c1(1,2) + c2(3,4), then ');
```

```
c = inv([a1;a2]') * [1;0];
c1 = c(1,1);
c2 = c(2,1);
disp(c1,'c1 = ');
disp(c2,'c2 = ');
disp('The transformation function T(1,0) will be:');
T = c1*Ta1 + c2*Ta2;
disp(T,'T(1,0) = ');
//end
```

**Scilab code Exa 3.12** `Singular and onto linear transformation`

```
//page 81
//Example 3.12
clc;
clear;
close;
x = round(rand(1,2) * 10);
x1 = x(1);
x2 = x(2);
T = [x1+x2 x1];
disp(x1,'x1 = ');
disp(x2,'x2 = ');
printf('T(%d,%d) = ',x1,x2);
disp(T);
disp('If , T(x1,x2) = 0, then');
disp('x1 = x2 = 0');
disp('So, T is non-singular');
disp('z1,z2 are two scalars in F');
z1 = round(rand() * 10);
z2 = round(rand() * 10);
disp(z1,'z1 = ');
disp(z2,'z2 = ');
x1 = z2;
x2 = z1 - z2;
```

```
disp(x1,'So, x1 = ');
disp(x2,'x2 = ');
disp('Hence, T is onto.');
Tinv = [z2 z1-z2];
disp(Tinv,'inverse(T) = ');
//end
```

**Scilab code Exa 3.14** `Standard Ordered Basis`

```
//page 89
//Example 3.14
clc;
clear;
close;
disp('T is a linear operator on F^2 defined as:');
disp('T(x1,x2) = (x1,0)');
disp('B = {e1,e2} is a standard ordered basis for F
    ^2,then');
x1 = 1;
x2 = 0;
Te1 = [x1 0];
x1 = 0;
x2 = 1;
Te2 = [x1 0];
disp(Te1,'So, Te1 = T(1,0) = ');
disp(Te2,'So, Te2 = T(0,1) = ');
disp('so,matrix T in ordered basis B is: ');
T = [Te1; Te2];
disp(T,'T = ');
//end
```

**Scilab code Exa 3.15** `Matrix in Ordered basis`

```
//page 89
//Example 3.15
clc;
clear;
close;
disp('Differentiation operator D is defined as:');
D = zeros(4,4);
x = poly(0,"x");
for i= 1:4
    t= i-1;
    f = derivat(x^t);
    printf('(Df%d)(x) = ',i);
    disp(f);
    if ~(i == 1) then
    D(i-1,i) = i-1;
    end
end
disp('Matrix of D in ordered basis is:');
disp(D,'[D] = ');
//end
```

**Scilab code Exa 3.16** Standard Ordered Basis

```
//page 92
//Example 3.16
clc;
clear;
close;
disp('T is a linear operator on R^2 defined as T(x1, x2) = (x1,0)');
disp('So, the matrix T in standard ordered basis B = {e1,e2} is ');
T = [1 0 ;0 0];
disp(T,'[T]B = ');
disp('Let B'' is the ordered basis for R^2
```

```
consisting of vectors:');
E1 = [1 1];
E2 = [2 1];
disp(E1,'E1 = ');
disp(E2,'E2 = ');
P = [E1;E2]'
disp(P,'So, matrix P = ');
Pinv = inv(P);
disp(Pinv,'P inverse = ');
T1 = Pinv*T*P;
disp(T1,'So, matrix T in ordered basis B'' is [T]B'' = ');
//end
```

**Scilab code Exa 3.17** `Matrix in ordered basis`

```
//page 93
//Example 3.17
clc;
clear;
close;
t = poly(0,"t");
disp('g1 = f1');
disp('g2 = t*f1 + f2');
disp('g3 = t^2*f1 + 2*t*f2 + f3');
disp('g4 = t^3*f1 + 3*t^2*f2 + 3*t*f3 + f4');
P = [1 t t^2 t^3;0 1 2*t 3*t^2;0 0 1 3*t;0 0 0 1];
disp(P,'P = ');
disp(inv(P),'inverse P = ');
disp('Matrix of differentiation operator D in ordered basis B is:'); //As found in example 15
D = [0 1 0 0;0 0 2 0;0 0 0 3;0 0 0 0];
disp(D,'D = ');
disp('Matrix of D in ordered basis B'' is:');
disp(inv(P)*D*P,'inverse(P) * D * P = ');
```

```
//end
```

**Scilab code Exa 3.19** Trace of a matrix

```
//page 98
//Example 3.19
clc;
clear;
close;
function [tr] = trace_matrix(M,n)
    for i = 1 : n
    tr = tr + M(i,i);
    end
endfunction
n = round(rand() * 10 + 2);
disp(n,'n = ');
A = round(rand(n,n) * 10);
disp(A,'A = ');
tr = 0;
disp('Trace of A:');
tr1 = trace_matrix(A,n);
disp(tr1,'tr(A) = ');
disp('----------------------------------------');
c = round(rand() * 10 + 2);
disp(c,'c = ');
B = round(rand(n,n) * 10);
disp(B,'B = ');
disp('Trace of B:');
tr2 = trace_matrix(B,n);
disp(tr2,'tr(B) = ');
disp(c*tr1+tr2,'tr(cA + B) = ');
//end
```

**Scilab code Exa 3.23** Linear functional on vector space

```
//page 103
//Example 3.23
clc;
clear;
close;
disp('Matrix represented by given linear functionals
    on R^4:');
A = [1 2 2 1;0 2 0 1;-2 0 -4 3];
disp(A,'A = ');
T = A;                  //Temporary matrix to store A
disp('To find Row reduced echelon matrix of A given
   by R:')
disp('Applying row transformations on A,we get');
disp('R1 = R1-R2');
A(1,:) = A(1,:) - A(2,:);
disp(A,'A = ');
disp('R3 = R3 + 2*R1');
A(3,:) = A(3,:) + 2*A(1,:);
disp(A,'A = ');
disp('R3 = R3/3');
A(3,:) = 1/3*A(3,:);
disp(A,'A = ');
disp('R2 = R2/2');
A(2,:) = 1/2*A(2,:);
disp(A,'A = ');
disp('R2 = R2 - 1/2*R3');
A(2,:) = A(2,:) - 1/2*A(3,:);
disp(A,'A = ');
R = A;
A = T;
disp('Row reduced echelon matrix of A is:');
disp(R,'R = ');
disp('Therefore,linear functionals g1,g2,g3 span the
    same subspace of (R^4)* as f1,f2,f3 are given by
   :');
disp('g1(x1,x2,x3,x4) = x1 + 2*x3');
```

```
disp('g1(x1,x2,x3,x4) = x2');
disp('g1(x1,x2,x3,x4) = x4');
disp('The subspace consists of the vectors with');
disp('x1 = -2*x3');
disp('x2 = x4 = 0');
//end
```

**Scilab code Exa 3.24** Linear functional on vector space

```
//page 104
//Example 3.24
clc;
clear;
close;
disp('W be the subspace of R^5 spanned by vectors:')
    ;
a1 = [2 -2 3 4 -1];
a2 = [-1 1 2 5 2];
a3 = [0 0 -1 -2 3];
a4 = [1 -1 2 3 0];
disp(a1,'a1 = ');
disp(a2,'a2 = ');
disp(a3,'a3 = ');
disp(a4,'a4 = ');
disp('Matrix A by the row vectors a1,a2,a3,a4 will
    be:');
A = [a1;a2;a3;a4];
disp(A,'A = ');
disp('After Applying row transformations, we get the
    row reduced echelon matrix R of A;');
T = A;                      //Temporary matrix to store
    A
//R1 = R1 - R4 and R2 = R2 + R4
A(1,:) = A(1,:) - A(4,:);
A(2,:) = A(2,:) + A(4,:);
```

```
//R2 = R2/2
A(2,:) = 1/2 * A(2,:);
//R3 = R3 + R2 and R4 = R4 - R1
A(3,:) = A(3,:) + A(2,:);
A(4,:) = A(4,:) - A(1,:);
//R3 = R3 - R4
A(3,:) = A(3,:) - A(4,:);
//R3 = R3/3
A(3,:) = 1/3 * A(3,:);
//R2 = R2 - R3
A(2,:) = A(2,:) - A(3,:);
//R2 = R2/2 and R4 = R4 - R2 - R3
A(2,:) = 1/2 * A(2,:);
A(4,:) = A(4,:) - A(2,:) - A(3,:);
//R1 = R1 - R2 + R3
A(1,:) = A(1,:) - A(2,:) + A(3,:);
R = A;
A = T;
disp(R,'R = ');
disp('Then we obtain all the linear functionals f by
    assigning arbitrary values to c2 and c4');
disp('Let c2 = a, c4 = b then c1 = a+b, c3 = -2b, c5
    = 0.');
disp('So, W0 consists all linear functionals f of
   the form');
disp('f(x1,x2,x3,x4,x5) = (a+b)x1 + ax2 -2bx3 + bx4'
   );
disp('Dimension of W0 = 2 and basis {f1,f2} can be
   found by first taking a = 1, b = 0. Then a = 0,b
   = 1');
//end
```

# Chapter 4

# Polynomials

**Scilab code Exa 4.3** `Algebra of linear operators`

```
//page 121
//Example 4.3
clc;
clear;
close;
disp('C is the field of complex numbers');
x = poly(0,"x");
f = x^2 + 2;
disp(f,'f = ');
//part a
disp('if a = C and z belongs to C, then f(z) = z^2 + 2');
disp(horner(f,2),'f(2) = ');
disp(horner(f,(1+%i)/(1-%i)),'f(1+%i/1-%i) = ');
disp('------------------------------------------------');
//part b
disp('If a is the algebra of all 2*2 matrices over C and');
B = [1 0;-1 2];
disp(B,'B = ');
disp(2*eye(2,2) + B^2,'then, f(B) = ');
```

```
disp('-----------------------------------------');
//part c
disp('If a is algebra of all linear operators on C^3');
disp('And T is element of a as:');
disp('T(c1,c2,c3) = (i*2^1/2*c1,c2,i*2^1/2*c3)');
disp('Then, f(T)(c1,c2,c3) = (0,3*c2,0)');
disp('-----------------------------------------');
//part d
disp('If a is the algebra of all polynomials over C');
g = x^4 + 3*%i;
disp(g,'And, g = ');
disp(horner(f,g),'Then f(g) = ');
//end
```

**Scilab code Exa 4.7** `Ideal of a polynomial`

```
//page 131
//Example 4.7
clc;
clear;
close;
x = poly(0,"x");
p1 = x + 2;
p2 = x^2 + 8*x + 16;
disp('M = (x+2)F[x] + (x^2 + 8x + 16)F[x]');
disp('We assert, M = F[x]');
disp('M contains:');
t = p2 - x*p1;
disp(t);
disp('And hence M contains:');
disp(t - 6*p1);
disp('Thus the scalar polynomial 1 belongs to M as well all its multiples.')
```

```
//end
```

**Scilab code Exa 4.8** `G C D of polynomials`

```
//page 133
//Example 4.8
clc;
clear;
close;
x = poly(0,"x");
//part a
p1 = x + 2;
p2 = x^2 + 8*x + 16;
disp(p1,'p1 = ');
disp(p2,'p2 = ');
disp('M = (x+2)F[x] + (x^2 + 8x + 16)F[x]');
disp('We assert , M = F[x]');
disp('M contains:');
t = p2 - x*p1;
disp(t);
disp('And hence M contains:');
disp(t - 6*p1);
disp('Thus the scalar polynomial 1 belongs to M as well all its multiples');
disp('So, gcd(p1,p2) = 1');
disp('----------------------------------------------------------------');
//part b
p1 = (x - 2)^2*(x+%i);
p2 = (x-2)*(x^2 + 1);
disp(p1,'p1 = ');
disp(p2,'p2 = ');
disp('M = (x - 2)^2*(x+%i)F[x] + (x-2)*(x^2 + 1');
disp('The ideal M contains p1 - p2 i.e.,');
disp(p1 - p2);
```

```
disp('Hence it contains (x-2)(x+i), which is monic
    and divides both,');
disp('So, gcd(p1,p2) = (x-2)(x+i)');
disp('----------------------------------------------------
    ');
//end
```

**Scilab code Exa 4.9** `Ideal of a Polynomial`

```
//page 133
//Example 4.9
clc;
clear;
close;
disp('M is the ideal in F[x] generated by:');
disp('(x-1)*(x+2)^2');
disp('(x+2)^2*(x+3)');
disp('(x-3)','and');
x = poly(0,"x");
p1 = (x-1)*(x+2)^2;
p2 = (x+2)^2*(x-3);
p3 = (x-3);
disp('M = (x-1)*(x+2)^2 F[x] + (x+2)^2*(x-3) + (x-3)
    ');
disp('Then M contains:');
t = 1/2*(x+2)^2*((x-1) - (x-3));
disp(t);
disp('i.e., M contains (x+2)^2');
disp('and since, (x+2)^2 = (x-3)(x-7) - 17');
disp('So M contains the scalar polynomial 1.');
disp('So, M = F[x] and given polynomials are
    relatively prime.');
//end
```

**Scilab code Exa 4.10** Reducible Polynomial

```
//page 135
//Example 4.10
clc;
clear;
close;
x = poly(0,"x");
P = x^2 + 1;
disp(P,'P = ');
disp('P is reducible over complex numbers as: ');
disp('=',P);
disp('(x-i)(x+i)');
disp('Whereas, P is irreducible over real numbers as
   :.');
disp('=',P);
disp('(ax + b)(a''x + b'')');
disp('For, a,a'',b,b'' to be in R,');
disp('aa'' = 1');
disp('ab'' + ba'' = 0');
disp('bb'' = 1');
disp('=> a^2 + b^2 = 0');
disp('=> a = b = 0');
//end
```

# Chapter 5

# Determinants

**Scilab code Exa 5.3** Two linear function

```
//page 143
//Example 5.3
clc;
clear;
close;
A = round(rand(2,2) *10 );
disp(A,'A = ');
D1 = A(1,1)*A(2,2);
D2 = - A(1,2)*A(2,1);
disp(D1,'D1(A) = ');
disp(D2,'D2(A) = ');
disp(D1 + D2,'D(A) = D1(A) + D2(A) = ');
disp('That is, D is a 2-linear function.');
//end
```

**Scilab code Exa 5.4** Alternating 3 Linear Functions

```
//page 145
```

```
//Example 5.4
clc;
clear;
close;
x = poly(0,"x");
A = [x 0 -x^2;0 1 0;1 0 x^3];
disp(A,'A = ');
disp('e1,e2,e3 are the rows of 3*3 identity matrix, then');
T = eye(3,3);
e1 = T(1,:);
e2 = T(2,:);
e3 = T(3,:);
disp(e1,'e1 = ');
disp(e2,'e2 = ');
disp(e3,'e3 = ');
disp('D(A) = D(x*e1 - x^2*e3, e2, e1 + x^3*e3)');
disp('Since, D is linear as a function of each row,');
disp('D(A) = x*D(e1,e2,e1 + x^3*e3) - x^2*D(e3,e2,e1 + x^3*e3)');
disp('D(A) = x*D(e1,e2,e1) + x^4*D(e1,e2,e3) - x^2*D(e3,e2,e1) - x^5*D(e3,e2,e3)');
disp('As D is alternating, So');
disp('D(A) = (x^4 + x^2)*D(e1,e2,e3)');
//end
```

**Scilab code Exa 5.5** Determinant of a matrix

```
//page 147
//Example 5.5
clc;
clear;
close;
function [E1 , E2 , E3] = determinant(A)
```

```
    E1 = A(1,1)*det([A(2,2) A(2,3);A(3,2) A(3,3)]) -
        A(2,1)*det([A(1,2) A(1,3);A(3,2) A(3,3)]) +
       A(3,1)*det([A(1,2) A(1,3);A(2,2) A(2,3)]);
    E2 = -A(1,2)*det([A(2,1) A(2,3);A(3,1) A(3,3)])
       + A(2,2)*det([A(1,1) A(1,3);A(3,1) A(3,3)]) +
        A(3,2)*det([A(1,1) A(1,3);A(2,1) A(2,3)]);
    E3 = A(1,3)*det([A(2,1) A(2,2);A(3,1) A(3,2)]) -
        A(2,3)*det([A(1,1) A(1,2);A(3,1) A(3,2)]) +
       A(3,3)*det([A(1,1) A(1,2);A(2,1) A(2,2)]);
endfunction

//part a
x = poly(0,"x");
A = [x-1 x^2 x^3;0 x-2 1;0 0 x-3];
disp(A,'A = ');
[E1, E2, E3] = determinant(A);
disp(E1,'E1(A) = ');
disp(E2,'E2(A) = ');
disp(E3,'E3(A) = ');
disp('----------------------------------------------------');
//part b
A = [0 1 0;0 0 1;1 0 0];
disp(A,'A = ');
[E1, E2, E3] = determinant(A);
disp(E1,'E1(A) = ');
disp(E2,'E2(A) = ');
disp(E3,'E3(A) = ');
//end
```

**Scilab code Exa 5.6** Determinant of a matrix

```
//page 158
//Example 5.6
clc;
clear;
```

```
close;
disp('Given Matrix:');
A = [1 -1 2 3; 2 2 0 2; 4 1 -1 -1;1 2 3 0];
disp(A,'A = ');
disp('After, Subtracting muliples of row 1 from rows 2 3 4');
disp('R2 = R2 - 2*R1');
A(2,:) = A(2,:) - 2 * A(1,:);
disp('R3 = R3 - 4*R1');
A(3,:) = A(3,:) - 4 * A(1,:);
disp('R4 = R4 - R1');
A(4,:) = A(4,:) - A(1,:);
disp(A,'A = ');
T = A;                    //Temporary matrix to store A
disp('We obtain the same determinant as before.');
disp('Now, applying some more row transformations as :');
disp('R3 = R3 - 5/4 * R2');
T(3,:) = T(3,:) - 5/4 * T(2,:);
disp('R4 = R4 - 3/4 * R2');
T(4,:) = T(4,:) - 3/4 * T(2,:);
B = T;
disp('We get B as:');
disp(B,'B = ');
disp('Now,determinant of A and B will be same');
disp(det(B),'det A = det B = ');
//end
```

**Scilab code Exa 5.7** Inverse of a matrix

```
//page 160
//Example 5.7
clc;
clear;
```

```
close;
x = poly(0,"x");
A = [x^2+x x+1;x-1 1];
B = [x^2-1 x+2;x^2-2*x+3 x];
disp(A,'A = ');
disp(B,'B = ');
disp(det(A),'det A = ');
disp(det(B),'det B = ');
disp('Thus, A is not invertible over K whereas B is
    invertible');
disp(inv(A)*det(A),'adj A = ');
disp(inv(B)*det(B),'adj B = ');
disp('(adj A)A = (x+1)I');
disp('(adj B)B = -6I');
disp(inv(B),'B inverse = ');
//end
```

**Scilab code Exa 5.8** Inverse of a matrix

```
//page 161
//Example 5.8
clc;
clear;
close;
A = [1 2;3 4];
disp(A,'A = ');
d = det(A);
disp(d,'det A = ','Determinant of A is:');
ad = (det(A) * eye(2,2)) / A;
disp(ad,'adj A = ','Adjoint of A is:');
disp('Thus, A is not invertible as a matrix over the
    ring of integers.');
disp('But, A can be regarded as a matrix over field
    of rational numbers.');
in = inv(A);
```

```
//The A inverse matrix given in book has a wrong
    entry of 1/2. It should be −1/2.
disp(in,'inv(A) = ','Then, A is invertible and
    Inverse of A is:');
//end
```

# Chapter 6

# Elementary Canonical Forms

**Scilab code Exa 6.1** Characteristic Polynomial of a matrix

```
//page 184
//Example 6.1
clc;
clear;
close;
disp('Standard ordered matrix for Linear operator T
   on R^2 is:');
A = [0 -1;1 0];
disp(A,'A = ');
disp('The characteristic polynomial for T or A is:')
x = poly(0,"x");
p = detr(x*eye(2,2)-A);
disp(p);
disp('Since this polynomial has no real roots,T has
   no characteristic values.');
//end
```

**Scilab code Exa 6.2** Characteristic Polynomial of a matrix

```
//page 184
//Example 6.2
clc;
clear;
close;
A = [3 1 -1; 2 2 -1;2 2 0];
disp(A,'A = ');
disp('Characteristic polynomial for A is:');
p = poly(A,"x");
disp(p);
disp('or');
disp('(x-1)(x-2)^2');
r = roots(p);
[m,n] = size(A);
disp('The characteristic values of A are:');
disp(round(r));
B = A-eye(m,n);
disp(B,'Now, A-I = ');
disp(rank(B),'rank of A - I= ');
disp('So, nullity of T-I = 1');
a1 = [1 0 2];
disp(a1,'The vector that spans the null space of T-I
    = ');
B = A-2*eye(m,n);
disp(B,'Now,A-2I = ');
disp(rank(B),'rank of A - 2I= ');
disp('T*alpha = 2*alpha if alpha is a scalar
   multiple of a2');
a2 = [1 1 2];
disp(a2,'a2 = ');
//end
```

**Scilab code Exa 6.3** Characteristic Polynomial of a matrix

```
//page 187
```

```
//Example 6.3
clc;
clear;
close;
disp('Standard ordered matrix for Linear operator T
   on R^3 is:');
A = [5 -6 -6; -1 4 2; 3 -6 -4];
disp(A,'A = ');
disp('xI - A = ');
B = eye(3,3);
x = poly(0,"x");
P = x*B - A;
disp(P);
disp('Applying row and column transformations:');
disp('C2 = C2 - C3');
P(:,2) = P(:,2) - P(:,3);
disp('=>');
disp(P);
disp('Taking (x-2) common from C2');
c = x-2;
P(:,2) = P(:,2) / (x-2);
disp('=>');
disp(' * ', c);
disp(P);
disp('R3 = R3 + R2');
P(3,:) = P(3,:) + P(2,:);
disp('=>');
disp(' * ', c);
disp(P);
P = [P(1,1) P(1,3); P(3,1) P(3,3)];
disp('=>');
disp(' * ', c);
disp(P);
disp('=>');
disp(' * ',c);
disp(det(P));
disp('This is the characteristic polynomial');
disp(A-B,'Now, A - I = ');
```

```
disp(A-2*B,'And, A- 2I = ');
disp(rank(A-B),'rank(A-I) = ');
disp(rank(A-2*B),'rank(A-2I) = ');
disp('W1,W2 be the spaces of characteristic vectors
    associated with values 1,2');
disp('So by theorem 2, T is diagonalizable');
a1 = [3 -1 3];
a2 = [2 1 0];
a3 = [2 0 1];
disp(a1,'Null space of (T- I) i.e basis of W1 is
    spanned by a1 = ');
disp('Null space of (T- 2I) i.e. basis of W2 is
    spanned by vectors x1,x2,x3 such that x1 = 2x1 +
    2x3');
disp('One example are;');
disp(a2,'a2 = ');
disp(a3,'a3 = ');
disp('The diagonal matrix is:');
D = [1 0 0 ;0 2 0;0 0 2];
disp(D,'D = ');
disp('The standard basis matrix is denoted as:');
P = [a1;a2;a3]';
disp(P,'P = ');
disp(A*P,'AP = ');
disp(P*D,'PD = ');
disp('That is, AP = PD');
disp('=>  inverse(P)*A*P = D');
//end
```

**Scilab code Exa 6.4** Diagonalizable Operator

```
//page 193
//Example 6.4
clc;
clear;
```

```
close;
x = poly(0,"x");
A = [5 -6 -6; -1 4 2; 3 -6 -4];      //Matrix given in Example 3
disp(A,'A = ');
f = (x-1)*(x-2)^2;
disp('Characteristic polynomial of A is:');
disp('f = (x-1)(x-2)^2');
disp(f,'i.e., f = ');
p = (x-1)*(x-2);
disp((A-eye(3,3))*(A-2 * eye(3,3)),'(A-I)(A-2I) = ');
disp('Since, (A-I)(A-2I) = 0. So, Minimal polynomial for above is:');
disp(p,'p = ');
disp('----------------------------------------------');
A = [3 1 -1; 2 2 -1;2 2 0];      //Matrix given in Example 2
disp(A,'A = ');
f = (x-1)*(x-2)^2;
disp('Characteristic polynomial of A is:');
disp('f = (x-1)(x-2)^2');
disp(f,'i.e., f = ');
disp((A-eye(3,3))*(A-2 * eye(3,3)),'(A-I)(A-2I) = ');
disp('Since, (A-I)(A-2I) is not equal to 0. T is not diagonalizable. So, Minimal polynomial cannot be p.');
disp('----------------------------------------------');
A = [0 -1;1 0];
disp(A,'A = ');
f = x^2 + 1;
disp('Characteristic polynomial of A is:');
disp(f,'f = ');
disp(A^2 + eye(2,2),'A^2 + I = ');
disp('Since, A^2 + I = 0, so minimal polynomial is');
p = x^2 + 1;
```

```
disp(p,'p = ');
//end
```

**Scilab code Exa 6.5** Characteristic Polynomial of matrix

```
//page 197
//Example 6.5
clc;
clear;
close;
A = [0 1 0 1;1 0 1 0;0 1 0 1;1 0 1 0];
disp(A,'A = ');
disp('Computing powers on A:');
disp(A*A,'A^2 = ');
disp(A*A*A,'A^3 = ');
deff('[p] = p(x)','p = x^3 - 4*x');
disp('if p = x^3 - 4x, then');
disp(p(A),'p(A) = ');
x = poly(0,"x");
f = x^3 - 4*x;
disp(f,'Minimal polynomial for A is: ');
disp(roots(f),'Characteristic values for A are:');
disp(rank(A),'Rank(A) = ');
disp(round(poly(A,"x")),'So, from theorem 2,
    characteristic polynomial for A is:');
//end
```

**Scilab code Exa 6.12** Symmetric and skew symmetric matrix

```
//page 210
//Example 6.12
clc;
clear;
```

```
close;
A = round(rand(3,3) * 10);
disp(A,'A = ');
disp('A transpose is:');
disp(A','A'' = ');
if A' == A then
    disp('Since, A'' = A, A is a symmetric matrix.')
        ;
else
    disp('Since, A'' is not equal to A, A is not a
        symmetric matrix.');
end
if A' == -A then
    disp('Since, A'' = -A, A is a skew-symmetric
        matrix.');
else
    disp('Since, A'' is not equal to -A, A is not a
        skew-symmetric matrix.');
end
A1 = 1/2*(A + A');
A2 = 1/2*(A - A');
disp('A can be expressed as sum of A1 and A2');
disp('i.e., A = A1 + A2');
disp(A1,'A1 = ');
disp(A2,'A2 = ');
disp(A1 + A2,'A1 + A2 = ');
//end
```

# Chapter 7

# The Rational and Jordan Forms

**Scilab code Exa 7.3** Linear operator annihilator

```
//page 239
//Example 7.3
clc;
clear;
close;
A = [5 -6 -6;-1 4 2;3 -6 -4];
disp(A,'A = ');
f = poly(A,"x");
disp('Characteristic polynomial for linear operator
    T on R^3 will be:');
disp(f,'f = ');
disp('or');
disp('(x-1)(x-2)^2');
x = poly(0,"x");
disp('The minimal polynomial for T is:');
p = (x-1)*(x-2);
disp(p,'p = ');
disp('or');
disp('p = (x-1)(x-2)');
disp('So in cyclic decomposition of T, a1 will have
    p as its T-annihilator.');
```

```
disp('Another vector a2 that generate cyclic
    subspace of dimension 1 will have its T-
    annihilator as p2.');
p2 = x-2;
disp(p2,'p2 = ');
disp(p*p2,'pp2 = ');
disp('i.e., pp2 = f');
disp('Therefore, A is similar to B');
B = [0 -2 0;1 3 0;0 0 2];
disp(B,'B = ');
disp('Thus, we can see thet Matrix of T in ordered
    basis is B');
//end
```

**Scilab code Exa 7.6** `Characteristic and minimal polynomial of matrix`

```
//page 247
//Example 7.6
clc;
clear;
close;
disp('A = ');
disp('2   0   0');
disp('a   2   0');
disp('b   c   -1');
a = 1;
b = 0;
c = 0;
A = [2 0 0;a 2 0;b c -1];
disp(A,'A = ');
disp('Characteristic polynomial for A is:');
disp(poly(A,"x"),'p = ');
disp('In this case, minimal polynomial is same as
    characteristic polynomial.');
disp('-------------------------------------------');
```

```
a = 0;
b = 0;
c = 0;
A = [2 0 0;a 2 0;b c -1];
disp(A,'A = ');
disp('Characteristic polynomial for A is:');
disp(poly(A,"x"),'p = ');
disp('In this case, minimal polynomial is:');
disp('(x-2)(x+1)');
disp('or');
x = poly(0,"x");
s = (x-2)*(x+1);
disp(s);
disp('(A-2I)(A+I) = ');
disp('0    0    0');
disp('3a   0    0');
disp('ac   0    0');
disp('if a = 0, A is similar to diagonal matrix.')
//end
```

**Scilab code Exa 7.7** Characteristic and minimal polynomial of matrix

```
//page 247
//Example 7.7
clc;
clear;
close;
disp('A = ');
disp('2    0    0    0');
disp('1    2    0    0');
disp('0    0    2    0');
disp('0    0    a    2');
disp('Considering a = 1');
A = [2 0 0 0;1 2 0 0;0 0 2 0;0 0 1 2];
p = poly(A,"x");
```

```
disp('Characteristic polynomial for A is:');
disp(p,'p = ');
disp('or');
disp('(x-2)^4');
disp('Minimal polynomial for A =');
disp('(x-2)^2');
disp('For a = 0 and a = 1, characteristic and
    minimal polynomial are same.');
disp('But for a=0, the solution space of (A - 2I)
    has 3 dimension whereas for a = 1, it has 2
    dimension. ')
//end
```

# Chapter 8

# Inner Product Spaces

**Scilab code Exa 8.1** `Standard Inner Product`

```
//page 271
//Example 8.1
clc;
clear;
close;
n = round(rand() * 10 + 2);
a = round(rand(1,n) * 10)
b = round(rand(1,n) * 10)
disp(n,'n = ');
disp(a,'a = ');
disp(b,'b = ');
disp(a*b','Then, (a|b) = ');
//end
```

---

**Scilab code Exa 8.2** `Standard Inner Product`

```
//page 271
//Example 8.2
```

```
clc;
clear;
close;
a = round(rand(1,2) * 10)
b = round(rand(1,2) * 10)
disp(a,'a = ');
disp(b,'b = ');
x1 = a(1);
x2 = a(2);
y1 = b(1);
y2 = b(2);
t = x1*y1 - x2*y1 - x1*y2 + 4*x2*y2;
disp(t,'Then, a|b = ');
//end
```

**Scilab code Exa 8.9** Standard Inner Product

```
//page 278
//Example 8.9
clc;
clear;
close;
a = round(rand(1,2) * 10);
x = a(1);
y = a(2);
b = [-y x];
disp(a,'(x,y) = ');
disp(b,'(-y,x) = ');
disp('Inner product of these vectors is:');
t = -x*y + y*x;
disp(t,'(x,y)|(-y,x) = ');
disp('So, these are orthogonal.');
disp('----------------------------------------------');
disp('If inner product is defined as:');
disp('(x1,x2)|(y1,y2) = x1y1- x2y1 - x1y2 + 4x2y2');
```

```
disp('Then, (x,y)|(-y,x) = -x*y+y^2-x^2+4*x*y = 0 if
    ,');
disp('y = 1/2(-3 + sqrt(13))*x');
disp('or');
disp('y = 1/2(-3 - sqrt(13))*x');
disp('Hence,');
if y == 1/2*(-3 + sqrt(13))*x | y == 1/2*(-3 - sqrt
    (13))*x then
disp(a);
disp('is orthogonal to');
disp(b);
else
disp(a);
disp('is not orthogonal to');
disp(b);
end
//end
```

**Scilab code Exa 8.12** Orthogonal Vectors

```
//page 282
//Example 8.12
clc;
clear;
close;
b1 = [3 0 4];
b2 = [-1 0 7];
b3 = [2 9 11];
disp(b1,'b1 = ');
disp(b2,'b2 = ');
disp(b3,'b3 = ');
disp('Applying the Gram-Schmidt process to b1,b2,b3:
    ');
a1 = b1;
disp(a1,'a1 = ');
```

```
a2 = b2-((b2*b1')'/25*b1);
disp(a2,'a2 = ');
a3 = b3-((b3*b1')'/25*b1) - ((b3*a2')'/25*a2);
disp(a3,'a3 = ');
disp('{a1,a2,a3} are mutually orthogonal and hence forms orthogonal basis for R^3');
disp('Any arbitrary vector {x1,x2,x3} in R^3 can be expressed as:');
disp('y = {x1,x2,x3} = (3*x1 + 4*x3)/25*a1 + (-4*x1 + 3*x3)/25*a2 + x2/9*a3');
x1 = 1;
x2 = 2;
x3 = 3;
y = (3*x1 + 4*x3)/25*a1 + (-4*x1 + 3*x3)/25*a2 + x2/9*a3;
disp(x1,'x1 = ');
disp(x2,'x2 = ');
disp(x3,'x3 = ');
disp(y,'y = ');
disp('i.e. y = [x1 x2 x3], according to above equation.');
disp('Hence, we get the orthonormal basis as:');
disp(',',1/5*a1);
disp(',',1/5*a2);
disp(1/9*a3);
//end
```

**Scilab code Exa 8.13** `Orthogonal Vectors`

```
//page 283
//Example 8.13
clc;
clear;
close;
A = rand(2,2);
```

```
A(1,:) = A(1,:) + 1; //so b1 is not equal to zero
a = A(1,1);
b = A(1,2);
c = A(2,1);
d = A(2,2);
b1 = A(1,:);
b2 = A(2,:);
disp(A,'A = ');
disp(b1,'b1 = ');
disp(b2,'b2 = ');
disp('Applying the orthogonalization process to b1,
    b2:');
a1 = [a b];
a2 = (det(A)/(a^2 + b^2))*[-b' a'];
disp(a1,'a1 = ');
disp(a2,'a2 = ');
disp('a2 is not equal to zero if and only if b1 and
    b2 are linearly independent.');
disp('That is, if determinant of A is non-zero.');
//end
```

**Scilab code Exa 8.14** Orthogonal Projection

```
//page 286
//Example 8.14
clc;
clear;
close;
v = [-10 2 8];
u = [3 12 -1]
disp(v,'v = ');
disp(u,'v = ');
disp('Orthogonal projection of v1 on subspace W
    spanned by v2 is given by:');
a = ((u*v')')/(u(1)^2 + u(2)^2 + u(3)^2) * u;
```

```
disp(a);
disp('Orthogonal projection of R^3 on W is the
   linear transformation E given by:');
printf('(x1,x2,x3) -> (3*x1 + 12*x2 - x3)/%d * (3 12
    -1)',(u(1)^2 + u(2)^2 + u(3)^2));
disp('Rank(E) = 1');
disp('Nullity(E)  = 2');
//end
```

**Scilab code Exa 8.15** Orthogonal sets

```
//page 288
//Example 8.15
clc;
clear;
close;
//part c
disp('f = (sqrt(2)*cos(2*pi*t) + sqrt(2)*sin(4*pi*t)
   )^2');
disp('Integration (f dt) in limits 0 to 1 = ');
x0 = 0;
x1 = 1;
X = integrate('(sqrt(2)*cos(2*%pi*t) + sqrt(2)*sin
   (4*%pi*t))^2','t',x0,x1);
disp(X);
//end
```

**Scilab code Exa 8.17** Inner product space and orthogonal projection

```
//page 294
//Example 8.17
//Equation given in example 14 is used.
clc;
```

```
clear;
close;
function [m] = transform(x,y,z)
    x1 = 3*x;
    x2 = 12*y;
    x3 = -z;
    m = [x1 x2 x3];
endfunction

disp('Matrix of projection E in orthonormal basis is :');
t1 = transform(3,3,3);
t2 = transform(12,12,12);
t3 = transform(-1,-1,-1);
A = [t1; t2; t3];
disp(A,'A = 1/154  *  ');
A1 = (conj(A))';
disp(A1,'A* = ');
disp('Since, E = E* and A = A*, then A is also the matrix of E*');
a1 = [154 0 0];
a2 = [145 -36 3];
a3 = [-36 10 12];
disp(a1,'a1 = ');
disp(a2,'a2 = ');
disp(a3,'a3 = ');
disp('{a1,a2,a3} is the basis.');
Ea1 = [9 36 -3];
Ea2 = [0 0 0];
Ea3 = [0 0 0];
disp(Ea1,'Ea1 = ');
disp(Ea2,'Ea2 = ');
disp(Ea3,'Ea3 = ');
B = [-1 0 0;-1 0 0;0 0 0];
disp('Matrix B of E in the basis is:');
disp(B,'B = ');
B1 = (conj(B))';
disp(B1,'B* = ');
```

```
disp('Since, B is not equal to B*, B is not the
    matrix of E*');
//end
```

**Scilab code Exa 8.28** Unitary matrix

```
//page 307
//Example 8.28
clc;
clear;
close;
disp('x1 and x2  are two real nos. i.e., x1^2 + x2^2
    = 1');
x1 = rand();
x2 = sqrt(1 - x1^2);
disp(x1,'x1 = ');
disp(x2,'x2 = ');
B = [x1 x2 0;0 1 0;0 0 1];
disp(B,'B = ');
disp('Applying Gram-Schmidt process to B:')
a1 = [x1 x2 0];
a2 = [0 1 0] - x2 * [x1 x2 0];
a3 = [0 0 1];
disp(a1,'a1 = ');
disp(a2,'a2 = ');
disp(a3,'a3 = ');
U = [a1;a2/x1;a3];
disp(U,'U = ');
M = [1 0 0;-x2/x1 1/x1 0;0 0 1];
disp(M,'M = ')
disp(inv(M) * U,'inverse(M) * U = ');
disp('So, B = inverse(M) * U');
//end
```

# Chapter 10

# Bilinear Forms

**Scilab code Exa 10.4** Bilinear Form of vectors

```
//page 363
//Example 10.4
clc;
clear;
close;
disp('a = [x1 x2]');
disp('b = [y1 y2]');
disp('f(a,b) = x1*y1 + x1*y2 + x2*y1 + x2*y2');
disp('so, f(a,b) = ');
disp('[x1  x2]  *  |1   1|  *   |y1|');
disp('              |1   1|      |y2|');
disp('So the matrix of f in standard order basis B =
    {e1,e2} is:');
fb = [1 1;1 1];
disp(fb,'[f]B = ');
P = [1 1;-1 1];
disp(P,'P = ');
disp('Thus, [f]B'' = P''*[f]B*P');
fb1 = P' * fb * P;
disp(fb1,'[f]B'' = ');
//end
```

**Scilab code Exa 10.5** `Bilinear Form of vectors`

```
//page 365
//Example 10.5
clc;
clear;
close;
n = round(rand() * 10 + 2);
a = round(rand(1,n) * 10);
b = round(rand(1,n) * 10);
disp(n,'n = ');
disp(a,'a = ');
disp(b,'b = ');
f = a * b';
disp(f,'f(a,b) = ');
disp('f is non-degenerate billinear form on R^n.');
//end
```