

Scilab Textbook Companion for
Linear Algebra
by K. Hoffman and R. Kunze¹

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

June 11, 2016

¹Funded 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 | 4 |
| 1 Linear Equations | 5 |
| 2 Vector Spaces | 15 |
| 3 Linear Transformations | 27 |
| 4 Polynomials | 36 |
| 5 Determinants | 41 |
| 6 Elementary Canonical Forms | 47 |
| 7 The Rational and Jordan Forms | 54 |
| 8 Inner Product Spaces | 58 |
| 10 Bilinear Forms | 66 |

List of Scilab Codes

| | | |
|----------|---|----|
| Exa 1.5 | Elementary Row Operations | 5 |
| Exa 1.6 | Elementary Row Operations | 6 |
| Exa 1.7 | Row reduced echelon Matrix | 7 |
| Exa 1.8 | Row reduced echelon Matrix | 7 |
| Exa 1.9 | System of Equations | 8 |
| Exa 1.10 | Product of Matrices | 9 |
| Exa 1.14 | Inverse of a matrix | 11 |
| Exa 1.15 | Inverse of a matrix | 11 |
| Exa 1.16 | Inverse of a matrix | 12 |
| Exa 2.8 | Vector Subspace | 15 |
| Exa 2.10 | Row space of matrix | 16 |
| Exa 2.11 | Space of polynomial function | 16 |
| Exa 2.12 | Linear Dependency | 17 |
| Exa 2.13 | Standard basis of Matrix | 18 |
| Exa 2.20 | Inverse of a matrix | 19 |
| Exa 2.21 | Standard basis of matrix | 19 |
| Exa 2.22 | Standard basis of Matrix | 22 |
| Exa 3.6 | Linear Transformation function | 27 |
| Exa 3.12 | Singular and onto linear transformation | 28 |
| Exa 3.14 | Standard Ordered Basis | 29 |
| Exa 3.15 | Matrix in Ordered basis | 29 |
| Exa 3.16 | Standard Ordered Basis | 30 |
| Exa 3.17 | Matrix in ordered basis | 31 |
| Exa 3.19 | Trace of a matrix | 32 |
| Exa 3.23 | Linear functional on vector space | 33 |
| Exa 3.24 | Linear functional on vector space | 34 |
| Exa 4.3 | Algebra of linear operators | 36 |
| Exa 4.7 | Ideal of a polynomial | 37 |

| | | |
|----------|---|----|
| Exa 4.8 | G C D of polynomials | 38 |
| Exa 4.9 | Ideal of a Polynomial | 39 |
| Exa 4.10 | Reducible Polynomial | 40 |
| Exa 5.3 | Two linear function | 41 |
| Exa 5.4 | Alternating 3 Linear Functions | 41 |
| Exa 5.5 | Determinant of a matrix | 42 |
| Exa 5.6 | Determinant of a matrix | 43 |
| Exa 5.7 | Inverse of a matrix | 44 |
| Exa 5.8 | Inverse of a matrix | 45 |
| Exa 6.1 | Characteristic Polynomial of a matrix | 47 |
| Exa 6.2 | Characteristic Polynomial of a matrix | 47 |
| Exa 6.3 | Characteristic Polynomial of a matrix | 48 |
| Exa 6.4 | Diagonalizable Operator | 50 |
| Exa 6.5 | Characteristic Polynomial of matrix | 52 |
| Exa 6.12 | Symmetric and skew symmetric matrix | 52 |
| Exa 7.3 | Linear operator annihilator | 54 |
| Exa 7.6 | Characteristic and minimal polynomial of matrix | 55 |
| Exa 7.7 | Characteristic and minimal polynomial of matrix | 56 |
| Exa 8.1 | Standard Inner Product | 58 |
| Exa 8.2 | Standard Inner Product | 58 |
| Exa 8.9 | Standard Inner Product | 59 |
| Exa 8.12 | Orthogonal Vectors | 60 |
| Exa 8.13 | Orthogonal Vectors | 61 |
| Exa 8.14 | Orthogonal Projection | 62 |
| Exa 8.15 | Orthogonal sets | 63 |
| Exa 8.17 | Inner product space and orthogonal projection | 63 |
| Exa 8.28 | Unitary matrix | 65 |
| Exa 10.4 | Bilinear Form of vectors | 66 |
| Exa 10.5 | Bilinear Form of vectors | 67 |

Chapter 1

Linear Equations

Scilab code Exa 1.5 Elementary Row Operations

```
1 //page 8
2 //Example 1.5
3 clear;
4 close;
5 clc;
6 a = [2 -1 3 2; 1 4 0 -1; 2 6 -1 5];
7 disp(a, 'a=');
8 disp('Applying row transformations:');
9 disp('R1 = R1-2*R2');
10 a(1,:) = a(1,:) - 2*a(2,:);
11 disp(a, 'a = ');
12 disp('R3 = R3-2*R2');
13 a(3,:) = a(3,:) - 2*a(2,:);
14 disp(a, 'a = ');
15 disp('R3 = R3/-2');
16 a(3,:) = -1/2*a(3,:);
17 disp(a, 'a = ');
18 disp('R2 = R2-4*R3');
19 a(2,:) = a(2,:) - 4*a(3,:);
20 disp(a, 'a = ');
21 disp('R1 = R1+9*R3');
```

```

22 a(1,:) = a(1,:) + 9*a(3,:);
23 disp(a, 'a = ');
24 disp('R1 = R1*2/15 ');
25 a(1,:) = a(1,:) * 2/15;
26 disp(a, 'a = ');
27 disp('R2 = R2+2*R1 ');
28 a(2,:) = a(2,:) + 2*a(1,:);
29 disp(a, 'a = ');
30 disp('R3 = R3-R1/2 ');
31 a(3,:) = a(3,:) - 1/2*a(1,:);
32 disp(a, 'a = ');
33 disp('We get the system of equations as:');
34 disp('2*x1 - x2 + 3*x3 + 2*x4 = 0 ');
35 disp('x1 + 4*x2 - x4 = 0 ');
36 disp('2*x1 + 6* x2 - x3 + 5*x4 = 0 ');
37 disp('and');
38 disp('x2 - 5/3*x4 = 0 ', 'x1 + 17/3*x4 = 0 ', 'x3 -
      11/3*x4 = 0 ');
39 disp('now by assigning any rational value c to x4 in
      system second, the solution is evaluated as:');
40 disp('(-17/3*c,5/3,11/3*c,c) ');
41 //end

```

Scilab code Exa 1.6 Elementary Row Operations

```

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

```



```

11 a(2,:) = a(2,:) + %i * a(3,:);
12 disp(a, 'a = ');
13 disp('R1 = R1 * (1/2+i)');
14 a(1,:) = 1/(2 + %i) * a(1,:);
15 disp(a, 'a = ');
16 disp('R2 = R2-R1*(3+2i) and R3 = R3 - 2 *R1');
17 a(2,:) = round(a(2,:) - (3 + 2 * %i) * a(1,:));
18 a(3,:) = round(a(3,:) - 2 * a(1,:));
19 disp(a, 'a = ');
20 disp('Thus the system of equations is:');
21 disp('x1 + 2*x2 = 0', '-i*x1 + 3*x2 = 0', '-x1+i*x2 =
    0');
22 disp('It has only trivial solution x1 = x2 = 0');
23 //end

```

Scilab code Exa 1.7 Row reduced echelon Matrix

```

1 //page 9
2 //Example 1.7
3 clear;
4 close;
5 clc;
6 n = rand();
7 n = round(n*10);
8 disp(eye(n,n));
9 printf('This is an Identity matrix of order %d * %d'
    ,n,n);
10 disp('And It is a row reduced matrix. ');
11 //end

```

Scilab code Exa 1.8 Row reduced echelon Matrix

```

1 //page 12

```

```

2 //Example 1.8
3 clear;
4 close;
5 clc;
6 n = rand();
7 n = round(n*10);
8 disp(eye(n,n));
9 printf('This is an Identity matrix of order %d * %d'
        ,n,n);
10 disp('And It is a row reduced matrix. ');
11 m = rand();
12 n = rand();
13 m = round(m*10);
14 n = round(n*10);
15 disp(zeros(m,n));
16 printf('This is an Zero matrix of order %d * %d',m,n
        );
17 disp('And It is also a row reduced matrix. ');
18 a = [0 1 -3 0 1/2;0 0 0 1 2;0 0 0 0 0];
19 disp(a, 'a = ');
20 disp('This is a non-trivial row reduced matrix. ');
21 //end

```

Scilab code Exa 1.9 System of Equations

```

1 //page 14
2 //Example 1.9
3 clear;
4 close;
5 clc;
6 A = [1 -2 1;2 1 1;0 5 -1];
7 disp(A, 'A = ');
8 disp('Applying row transformations: ');
9 disp('R2 = R2 - 2*R1 ');
10 A(2,:) = A(2,:) - 2*A(1,:);

```

```

11 disp(A, 'A = ');
12 disp('R3 = R3 - R2');
13 A(3,:) = A(3,:) - A(2,:);
14 disp(A, 'A = ');
15 disp('R2 = 1/5*R2');
16 A(2,:) = 1/5*A(2,:);
17 disp(A, 'A = ');
18 disp('R1 = R1 - 2*R2');
19 A(1,:) = A(1,:) + 2*A(2,:);
20 disp(A, 'A = ');
21 disp('The condition that the system have a solution
      is:');
22 disp('2*y1 - y2 + y3 = 0');
23 disp('where, y1,y2,y3 are some scalars');
24 disp('If the condition is satisfied then solutions
      are obtained by assigning a value c to x3');
25 disp('Solutions are:');
26 disp('x2 = 1/5*c + 1/5*(y2 - 2*y1)', 'x1 = -3/5*c +
      1/5*(y1 + 2*y2)');
27 //end

```

Scilab code Exa 1.10 Product of Matrices

```

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

```

```
    ');  
13 //Part b  
14 a = [1 0;-2 3;5 4;0 1];  
15 b = [0 6 1;3 8 -2];  
16 disp(a, 'a=');  
17 disp(b, 'b=');  
18 disp(a*b, 'ab = ');  
19 disp('
```

```
    ');  
20 //Part c  
21 a = [2 1;5 4];  
22 b = [1;6];  
23 disp(a, 'a=');  
24 disp(b, 'b=');  
25 disp(a*b, 'ab = ');  
26 disp('
```

```
    ');  
27 //Part d  
28 a = [-1;3];  
29 b = [2 4];  
30 disp(a, 'a=');  
31 disp(b, 'b=');  
32 disp(a*b, 'ab = ');  
33 disp('
```

```
    ');  
34 //Part e  
35 a = [2 4];  
36 b = [-1;3];  
37 disp(a, 'a=');  
38 disp(b, 'b=');  
39 disp(a*b, 'ab = ');  
40 disp('
```

```
    ');
```

```

41 //Part f
42 a = [0 1 0;0 0 0;0 0 0];
43 b = [1 -5 2;2 3 4;9 -1 3];
44 disp(a, 'a=');
45 disp(b, 'b=');
46 disp(a*b, 'ab = ');
47 disp('
');
48 //Part g
49 a = [1 -5 2;2 3 4;9 -1 3];
50 b = [0 1 0;0 0 0;0 0 0];
51 disp(a, 'a=');
52 disp(b, 'b=');
53 disp(a*b, 'ab = ');
54 //end

```

Scilab code Exa 1.14 Inverse of a matrix

```

1 //page 22
2 //Example 1.14
3 clear;
4 close;
5 clc;
6 a = [0 1;1 0];
7 disp(a, 'a = ');
8 disp(inv(a), 'inverse a = ');
9 //end

```

Scilab code Exa 1.15 Inverse of a matrix

```

1 //page 25
2 //Example 1.15

```

```

3 clear;
4 close;
5 clc;
6 a = [2 -1;1 3];
7 disp(a, 'a = ');
8 b = a; //Temporary variable to store a
9 disp('Applying row transformations');
10 disp('Interchange R1 and R2');
11 x = a(1,:);
12 a(1,:) = a(2,:);
13 a(2,:) = x;
14 disp(a, 'a = ');
15 disp('R2 = R2 - 2 * R1');
16 a(2,:) = a(2,:) - 2 * a(1,:);
17 disp(a, 'a = ');
18 disp('R2 = R2 *1/(-7)');
19 a(2,:) = (-1/7) * a(2,:);
20 disp(a, 'a = ');
21 disp('R1 = R1 - 3 * R2');
22 a(1,:) = a(1,:) - 3 * a(2,:);
23 disp(a, 'a = ');
24 disp('Since a has become an identity matrix. So, a
      is invertible');
25 disp('inverse of a = ');
26 disp(inv(b)); //a was stored in b
27 //end

```

Scilab code Exa 1.16 Inverse of a matrix

```

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

```

```

7 disp(a, 'a = ');
8 b = eye(3,3);
9 disp(b, 'b = ');
10 disp('Applying row transformations on a and b
    simultaneously, ');
11 disp('R2 = R2 - 1/2 * R1 and R3 = R3 - 1/3*R1');
12 a(2,:) = a(2,:) - 1/2 * a(1,:);
13 a(3,:) = a(3,:) - 1/3 * a(1,:);
14 b(2,:) = b(2,:) - 1/2 * b(1,:);
15 b(3,:) = b(3,:) - 1/3 * b(1,:);
16 disp(a, 'a = ');
17 disp(b, 'b = ');
18 disp('R3 = R3 - R2');
19 a(3,:) = a(3,:) - a(2,:);
20 b(3,:) = b(3,:) - b(2,:);
21 disp(a, 'a = ');
22 disp(b, 'b = ');
23 disp('R2 = R2 * 12 and R3 = R3 * 180');
24 a(2,:) = a(2,:) *12;
25 a(3,:) = a(3,:) * 180;
26 b(2,:) = b(2,:) * 12;
27 b(3,:) = b(3,:) * 180;
28 disp(a, 'a = ');
29 disp(b, 'b = ');
30 disp('R2 = R2 - R3 and R1 = R1 - 1/3*R3');
31 a(2,:) = a(2,:) - a(3,:);
32 a(1,:) = a(1,:) - 1/3 * a(3,:);
33 b(2,:) = b(2,:) - b(3,:);
34 b(1,:) = b(1,:) - 1/3 * b(3,:);
35 disp(a, 'a = ');
36 disp(b, 'b = ');
37 disp('R1 = R1 - 1/2 * R2');
38 a(1,:) = a(1,:) - 1/2 * a(2,:);
39 b(1,:) = b(1,:) - 1/2 * b(2,:);
40 disp(round(a), 'a = ');
41 disp(b, 'b = ');
42 disp('Since, a = identity matrix of order 3*3. So, b
    is inverse of a');

```

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

Chapter 2

Vector Spaces

Scilab code Exa 2.8 Vector Subspace

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

```

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

```

Scilab code Exa 2.10 Row space of matrix

```

1 //page 38
2 //Example 2.10
3 clear;
4 clc;
5 close;
6 A = [1 2 0 3 0;0 0 1 4 0;0 0 0 0 1];
7 disp(A, 'A = ');
8 disp('The subspace of  $F^5$  spanned by a1 a2 a3 (row
      vectors of A) is called row space of A. ');
9 a1 = A(1,:);
10 a2 = A(2,:);
11 a3 = A(3,:);
12 disp(a1, 'a1 = ');
13 disp(a2, 'a2 = ');
14 disp(a3, 'a3 = ');
15 disp('And, it is also the row space of B. ');
16 B = [1 2 0 3 0;0 0 1 4 0;0 0 0 0 1;-4 -8 1 -8 0];
17 disp(B, 'B = ');
18 //end

```

Scilab code Exa 2.11 Space of polynomial function

```

1 //page 39
2 //Example 2.11
3 clear;
4 clc;
5 close;
6 disp('V is the space of all polynomial functions
      over F. ');
7 disp('S contains the functions as:')
8 x = poly(0,"x");
9 n = round(rand()*10);
10 disp(n, 'n = ');
11 for i = 0 : n
12     f = x^i;
13     printf('f%d(x) = ',i);
14     disp(f);
15 end
16 disp('Then, V is the subspace spanned by set S. ');
17 //end

```

Scilab code Exa 2.12 Linear Dependency

```

1 //page 41
2 //Example 2.12
3 clear;
4 clc;
5 close;
6 a1 = [3 0 -3];
7 a2 = [-1 1 2];
8 a3 = [4 2 -2];
9 a4 = [2 1 1];
10 disp(a1, 'a1 = ');
11 disp(a2, 'a2 = ');
12 disp(a3, 'a3 = ');
13 disp(a4, 'a4 = ');
14 t = 2 * a1 + 2 * a2 - a3 + 0 * a4;

```

```

15 disp(' = 0 ',t,' Since , 2 * a1 + 2 * a2 - a3 + 0 * a4
    = ');
16 disp('a1,a2,a3,a4 are linearly independent');
17 e1 = [1 0 0];
18 e2 = [0 1 0];
19 e3 = [0 0 1];
20 disp(e1, 'Now, e1 = ');
21 disp(e2, 'e2 = ');
22 disp(e3, 'e3 = ');
23 disp('Also , e1 ,e2 ,e3 are linearly independent. ');
24 //end

```

Scilab code Exa 2.13 Standard basis of Matrix

```

1 //page 41
2 //Example 2.13
3 clear;
4 clc;
5 close;
6 disp('S is the subset of  $F^n$  consisting of n vectors
    . ');
7 n = round(rand() *10 + 1);
8 disp(n, 'n = ');
9 I = eye(n,n);
10 for i = 0 : n-1
11     e = I(i+1,:);
12     printf('e%d = ',i+1);
13     disp(e);
14 end
15 disp('x1,x2,x3...xn are the scalars in F');
16 disp('Putting a = x1*e1 + x2*e2 + x3*e3 + .... + xn*
    en ');
17 disp('So, a = (x1,x2,x3,...,xn) ');
18 disp('Therefore , e1 ,e2 .. ,en span  $F^n$  ');
19 disp('a = 0 if x1 = x2 = x3 = .. = xn = 0 ');

```

```

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

```

Scilab code Exa 2.20 Inverse of a matrix

```

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

```

Scilab code Exa 2.21 Standard basis of matrix

```

1 //page 60
2 //Example 2.21

```

```

3 clear;
4 clc;
5 close;
6 a1 = [1 2 2 1];
7 a2 = [0 2 0 1];
8 a3 = [-2 0 -4 3];
9 disp('Given row vectors are:');
10 disp(a1, 'a1 = ');
11 disp(a2, 'a2 = ');
12 disp(a3, 'a3 = ');
13 disp('The matrix A from these vectors will be:');
14 A = [a1; a2; a3];
15 disp(A, 'A = ');
16 disp('Finding Row reduced echelon matrix of A that
      is given by R');
17 disp('And applying same operations on identity
      matrix Q such that R = QA');
18 Q = eye(3,3);
19 disp(Q, 'Q = ');
20 T = A; //Temporary matrix to store A
21 disp('Applying row transformations on A and Q,we get
      ');
22 disp('R1 = R1-R2');
23 A(1,:) = A(1,:) - A(2,:);
24 Q(1,:) = Q(1,:) - Q(2,:);
25 disp(A, 'A = ');
26 disp(Q, 'Q = ');
27 disp('R3 = R3 + 2*R1');
28 A(3,:) = A(3,:) + 2*A(1,:);
29 Q(3,:) = Q(3,:) + 2*Q(1,:);
30 disp(A, 'A = ');
31 disp(Q, 'Q = ');
32 disp('R3 = R3/3');
33 A(3,:) = 1/3*A(3,:);
34 Q(3,:) = 1/3*Q(3,:);
35 disp(A, 'A = ');
36 disp(Q, 'Q = ');
37 disp('R2 = R2/2');

```

```

38 A(2,:) = 1/2*A(2,:);
39 Q(2,:) = 1/2*Q(2,:);
40 disp(A, 'A = ');
41 disp(Q, 'Q = ');
42 disp('R2 = R2 - 1/2*R3');
43 A(2,:) = A(2,:) - 1/2*A(3,:);
44 Q(2,:) = Q(2,:) - 1/2*Q(3,:);
45 disp(A, 'A = ');
46 disp(Q, 'Q = ');
47 R = A;
48 A = T;
49 disp('Row reduced echelon matrix:');
50 disp(R, 'R = ');
51 disp(Q, 'Q =');
52 //part a
53 disp(rank(R), 'rank of R = ');
54 disp('Since, Rank of R is 3, so a1, a2, a3 are
      independent');
55 //part b
56 disp('Now, basis for W can be given by row vectors
      of R i.e. p1,p2,p3');
57 disp('b is any vector in W. b = [b1 b2 b3 b4]');
58 disp('Span of vectors p1,p2,p3 consist of vector b
      with b3 = 2*b1');
59 disp('So, b = b1p1 + b2p2 + b4p3');
60 disp('And, [p1 p2 p3] = R = Q*A');
61 disp('So, b = [b1 b2 b3]* Q * A');
62 disp('hence, b = x1a1 + x2a2 + x3a3 where x1 = [b1
      b2 b4] * Q(1) and so on'); //Equation 1
63 //part c
64 disp('Now, given 3 vectors a1'' a2'' a3''');
65 c1 = [1 0 2 0];
66 c2 = [0 2 0 1];
67 c3 = [0 0 0 3];
68 disp(c1, 'a1'' = ');
69 disp(c2, 'a2'' = ');
70 disp(c3, 'a3'' = ');
71 disp('Since a1'' a2'' a3'' are all of the form (y1

```

```

    y2 y3 y4) with  $y_3 = 2*y_1$ , hence they are in W. ');
72 disp('So, they are independent. ');
73 //part d
74 c = [c1; c2; c3];
75 P = eye(3,3);
76 for i = 1:3
77     b1 = c(i,1);
78     b2 = c(i,2);
79     b4 = c(i,4);
80     x1 = [b1 b2 b4] * Q(:,1);
81     x2 = [b1 b2 b4]*Q(:,2);
82     x3 = [b1 b2 b4]*Q(:,3);
83     P(:,i) = [x1; x2; x3];
84 end
85 disp('Required matrix P such that  $X = PX$ ' is: ');
86 disp(P, 'P = ');
87 //end

```

Scilab code Exa 2.22 Standard basis of Matrix

```

1 //page 63
2 //Example 2.22
3 clear;
4 clc;
5 close;
6 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];
7 disp(A, 'A = ');
8 //part a
9 T = A; //Temporary storing A in T
10 disp('Taking an identity matrix P: ');
11 P = eye(5,5);
12 disp(P, 'P = ');
13 disp('Applying row transformations on P and A to get
      a row reduced echelon matrix R: ');

```



```

14 disp('R2 = R2 - R1 and R4 = R4 - 2* R1');
15 A(2,:) = A(2,:) - A(1,:);
16 P(2,:) = P(2,:) - P(1,:);
17 A(4,:) = A(4,:) - 2 * A(1,:);
18 P(4,:) = P(4,:) - 2 * P(1,:);
19 disp(A, 'A = ');
20 disp(P, 'P = ');
21 disp('R2 = -R2 , R3 = R3 - R1 + R2 and R4 = R4 - R1
      + R2');
22 A(2,:) = -A(2,:);
23 P(2,:) = -P(2,:);
24 A(3,:) = A(3,:) - A(2,:);
25 P(3,:) = P(3,:) - P(2,:);
26 A(4,:) = A(4,:) - A(2,:);
27 P(4,:) = P(4,:) - P(2,:);
28 disp(A, 'A = ');
29 disp(P, 'P = ');
30 disp('Mutually interchanging R3, R4 and R5');
31 x = A(3,:);
32 A(3,:) = A(5,:);
33 y = A(4,:);
34 A(4,:) = x;
35 A(5,:) = y - A(3,:);
36 x = P(3,:);
37 P(3,:) = P(5,:);
38 y = P(4,:);
39 P(4,:) = x;
40 P(5,:) = y - P(3,:);
41 R = A;
42 A = T;
43 disp(R, 'Row reduced echelon matrix R = ');
44 disp(P, 'Invertible Matrix P = ');
45 disp('Invertible matrix P is not unique. There can
      be many that depends on operations used to reduce
      A');
46 disp('-----');
47 //part b
48 disp('For the basis of row space W of A, we can take

```

```

    the non-zero rows of R');
49 disp('It can be given by p1, p2, p3');
50 p1 = R(1,:);
51 p2 = R(2,:);
52 p3 = R(3,:);
53 disp(p1, 'p1 = ');
54 disp(p2, 'p2 = ');
55 disp(p3, 'p3 = ');
56 disp('_____');
57 //part c
58 disp('The row space W consists of vectors of the
    form: ');
59 disp('b = c1p1 + c2p2 + c3p3');
60 disp('i.e. b = (c1, 2*c1, c2, 3*c1+4*c2, c3) where, c1
    c2 c3 are scalars. ');
61 disp('So, if b2 = 2*b1 and b4 = 3*b1 + 4*b3 => (b1
    , b2, b3, b4, b5) = b1p1 + b3p2 + b5p3');
62 disp('then, (b1, b2, b3, b4, b5) is in W');
63 disp('_____');
64 //part d
65 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: ');
66 disp(' b1 ');
67 disp(' b2 ');
68 disp(' b3 ');
69 disp('_____');
70 //part e
71 disp('Now, to write each vector in W as a linear
    combination of rows of A: ');
72 disp('Let b = (b1, b2, b3, b4, b5) and if b is in W,
    then ');
73 disp('we know, b = (b1, 2*b1, b3, 3*b1 + 4*b3, b5) => [
    b1, b3, b5, 0, 0]*R');
74 disp('=> b = [b1, b3, b5, 0, 0] * P*A => b = [b1+b3, -
    b3, 0, 0, b5] * A');
75 disp('if b = (-5, -10, 1, -11, 20) ');
76 b1 = -5;

```

```

77 b2 = -10;
78 b3 = 1;
79 b4 = -11;
80 b5 = 20;
81 x = [b1 + b3, -b3, 0, 0, b5];
82 disp(' ', A, ' [ ', '* ', ') ', x, ' ( ', 'b = ');
83 disp('_____');
84 //part f
85 disp('The equations in system  $RX = 0$  are given by R
      * [x1 x2 x3 x4 x5]');
86 disp('i.e.,  $x1 + 2*x2 + 3*x4$ ');
87 disp('x3 + 4*x4');
88 disp('x5');
89 disp('so, V consists of all columns of the form');
90 disp(' [ ', 'X=');
91 disp(' -2*x2 - 3*x4');
92 disp(' x2');
93 disp(' -4*x4');
94 disp(' x4');
95 disp(' 0');
96 disp('where x2 and x4 are arbitrary', ']'');
97 disp('_____');
98 //part g
99 disp('Let  $x2 = 1, x4 = 0$  then the given column forms
      a basis of V');
100 x2 = 1;
101 x4 = 0;
102 disp([-2*x2-3*x4; x2; -4*x4; x4; 0]);
103 disp('Similarly, if  $x2 = 0, x4 = 1$  then the given
      column forms a basis of V');
104 x2 = 0;
105 x4 = 1;
106 disp([-2*x2-3*x4; x2; -4*x4; x4; 0]);
107 disp('_____');
108 //part h
109 disp('The equation  $AX = Y$  has solutions X if and
      only if');
110 disp('-y1 + y2 + y3 = 0');

```

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

Chapter 3

Linear Transformations

Scilab code Exa 3.6 Linear Transformation function

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

```

18 c = inv([a1;a2]') * [1;0];
19 c1 = c(1,1);
20 c2 = c(2,1);
21 disp(c1, 'c1 = ');
22 disp(c2, 'c2 = ');
23 disp('The transformation function T(1,0) will be:');
24 T = c1*Ta1 + c2*Ta2;
25 disp(T, 'T(1,0) = ');
26 //end

```

Scilab code Exa 3.12 Singular and onto linear transformation

```

1 //page 81
2 //Example 3.12
3 clc;
4 clear;
5 close;
6 x = round(rand(1,2) * 10);
7 x1 = x(1);
8 x2 = x(2);
9 T = [x1+x2 x1];
10 disp(x1, 'x1 = ');
11 disp(x2, 'x2 = ');
12 printf('T(%d,%d) = ', x1, x2);
13 disp(T);
14 disp('If, T(x1,x2) = 0, then');
15 disp('x1 = x2 = 0');
16 disp('So, T is non-singular');
17 disp('z1,z2 are two scalars in F');
18 z1 = round(rand() * 10);
19 z2 = round(rand() * 10);
20 disp(z1, 'z1 = ');
21 disp(z2, 'z2 = ');
22 x1 = z2;
23 x2 = z1 - z2;

```

```

24 disp(x1, 'So, x1 = ');
25 disp(x2, 'x2 = ');
26 disp('Hence, T is onto. ');
27 Tinv = [z2 z1-z2];
28 disp(Tinv, 'inverse(T) = ');
29 //end

```

Scilab code Exa 3.14 Standard Ordered Basis

```

1 //page 89
2 //Example 3.14
3 clc;
4 clear;
5 close;
6 disp('T is a linear operator on F^2 defined as:');
7 disp('T(x1,x2) = (x1,0)');
8 disp('B = {e1,e2} is a standard ordered basis for F
      ^2, then ');
9 x1 = 1;
10 x2 = 0;
11 Te1 = [x1 0];
12 x1 = 0;
13 x2 = 1;
14 Te2 = [x1 0];
15 disp(Te1, 'So, Te1 = T(1,0) = ');
16 disp(Te2, 'So, Te2 = T(0,1) = ');
17 disp('so, matrix T in ordered basis B is: ');
18 T = [Te1; Te2];
19 disp(T, 'T = ');
20 //end

```

Scilab code Exa 3.15 Matrix in Ordered basis

```

1 //page 89
2 //Example 3.15
3 clc;
4 clear;
5 close;
6 disp('Differentiation operator D is defined as:');
7 D = zeros(4,4);
8 x = poly(0,"x");
9 for i= 1:4
10     t= i-1;
11     f = derivat(x^t);
12     printf('(Df%d)(x) = ',i);
13     disp(f);
14     if ~(i == 1) then
15         D(i-1,i) = i-1;
16     end
17 end
18 disp('Matrix of D in ordered basis is:');
19 disp(D, '[D] = ');
20 //end

```

Scilab code Exa 3.16 Standard Ordered Basis

```

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

```



```

    consisting of vectors:');
11 E1 = [1 1];
12 E2 = [2 1];
13 disp(E1, 'E1 = ');
14 disp(E2, 'E2 = ');
15 P = [E1;E2]';
16 disp(P, 'So, matrix P = ');
17 Pinv = inv(P);
18 disp(Pinv, 'P inverse = ');
19 T1 = Pinv*T*P;
20 disp(T1, 'So, matrix T in ordered basis B'' is [T]B''
    = ');
21 //end

```

Scilab code Exa 3.17 Matrix in ordered basis

```

1 //page 93
2 //Example 3.17
3 clc;
4 clear;
5 close;
6 t = poly(0, "t");
7 disp('g1 = f1 ');
8 disp('g2 = t*f1 + f2 ');
9 disp('g3 = t^2*f1 + 2*t*f2 + f3 ');
10 disp('g4 = t^3*f1 + 3*t^2*f2 + 3*t*f3 + f4 ');
11 P = [1 t t^2 t^3;0 1 2*t 3*t^2;0 0 1 3*t;0 0 0 1];
12 disp(P, 'P = ');
13 disp(inv(P), 'inverse P = ');
14 disp('Matrix of differentiation operator D in
    ordered basis B is:'); //As found in example 15
15 D = [0 1 0 0;0 0 2 0;0 0 0 3;0 0 0 0];
16 disp(D, 'D = ');
17 disp('Matrix of D in ordered basis B'' is:');
18 disp(inv(P)*D*P, 'inverse(P) * D * P = ');

```

19 //end

Scilab code Exa 3.19 Trace of a matrix

```
1 //page 98
2 //Example 3.19
3 clc;
4 clear;
5 close;
6 function [tr] = trace_matrix(M,n)
7     for i = 1 : n
8         tr = tr + M(i,i);
9     end
10 endfunction
11 n = round(rand() * 10 + 2);
12 disp(n, 'n = ');
13 A = round(rand(n,n) * 10);
14 disp(A, 'A = ');
15 tr = 0;
16 disp('Trace of A:');
17 tr1 = trace_matrix(A,n);
18 disp(tr1, 'tr(A) = ');
19 disp('-----');
20 c = round(rand() * 10 + 2);
21 disp(c, 'c = ');
22 B = round(rand(n,n) * 10);
23 disp(B, 'B = ');
24 disp('Trace of B:');
25 tr2 = trace_matrix(B,n);
26 disp(tr2, 'tr(B) = ');
27 disp(c*tr1+tr2, 'tr(cA + B) = ');
28 //end
```

Scilab code Exa 3.23 Linear functional on vector space

```
1 //page 103
2 //Example 3.23
3 clc;
4 clear;
5 close;
6 disp('Matrix represented by given linear functionals
      on R^4:');
7 A = [1 2 2 1;0 2 0 1;-2 0 -4 3];
8 disp(A, 'A = ');
9 T = A; //Temporary matrix to store A
10 disp('To find Row reduced echelon matrix of A given
      by R: ')
11 disp('Applying row transformations on A,we get');
12 disp('R1 = R1-R2');
13 A(1,:) = A(1,:) - A(2,:);
14 disp(A, 'A = ');
15 disp('R3 = R3 + 2*R1');
16 A(3,:) = A(3,:) + 2*A(1,:);
17 disp(A, 'A = ');
18 disp('R3 = R3/3');
19 A(3,:) = 1/3*A(3,:);
20 disp(A, 'A = ');
21 disp('R2 = R2/2');
22 A(2,:) = 1/2*A(2,:);
23 disp(A, 'A = ');
24 disp('R2 = R2 - 1/2*R3');
25 A(2,:) = A(2,:) - 1/2*A(3,:);
26 disp(A, 'A = ');
27 R = A;
28 A = T;
29 disp('Row reduced echelon matrix of A is:');
30 disp(R, 'R = ');
31 disp('Therefore, linear functionals g1,g2,g3 span the
      same subspace of (R^4)* as f1,f2,f3 are given by
      : ');
32 disp('g1(x1,x2,x3,x4) = x1 + 2*x3');
```

```

33 disp('g1(x1,x2,x3,x4) = x2');
34 disp('g1(x1,x2,x3,x4) = x4');
35 disp('The subspace consists of the vectors with');
36 disp('x1 = -2*x3');
37 disp('x2 = x4 = 0');
38 //end

```

Scilab code Exa 3.24 Linear functional on vector space

```

1 //page 104
2 //Example 3.24
3 clc;
4 clear;
5 close;
6 disp('W be the subspace of R^5 spanned by vectors:')
7 ;
8 a1 = [2 -2 3 4 -1];
9 a2 = [-1 1 2 5 2];
10 a3 = [0 0 -1 -2 3];
11 a4 = [1 -1 2 3 0];
12 disp(a1, 'a1 = ');
13 disp(a2, 'a2 = ');
14 disp(a3, 'a3 = ');
15 disp(a4, 'a4 = ');
16 disp('Matrix A by the row vectors a1,a2,a3,a4 will
17 be:');
18 A = [a1;a2;a3;a4];
19 disp(A, 'A = ');
20 disp('After Applying row transformations, we get the
21 row reduced echelon matrix R of A;');
22 T = A; //Temporary matrix to store
23 A
24 //R1 = R1 - R4 and R2 = R2 + R4
25 A(1,:) = A(1,:) - A(4,:);
26 A(2,:) = A(2,:) + A(4,:);

```

```

23 //R2 = R2/2
24 A(2,:) = 1/2 * A(2,:);
25 //R3 = R3 + R2 and R4 = R4 - R1
26 A(3,:) = A(3,:) + A(2,:);
27 A(4,:) = A(4,:) - A(1,:);
28 //R3 = R3 - R4
29 A(3,:) = A(3,:) - A(4,:);
30 //R3 = R3/3
31 A(3,:) = 1/3 * A(3,:);
32 //R2 = R2 - R3
33 A(2,:) = A(2,:) - A(3,:);
34 //R2 = R2/2 and R4 = R4 - R2 - R3
35 A(2,:) = 1/2 * A(2,:);
36 A(4,:) = A(4,:) - A(2,:) - A(3,:);
37 //R1 = R1 - R2 + R3
38 A(1,:) = A(1,:) - A(2,:) + A(3,:);
39 R = A;
40 A = T;
41 disp(R, 'R = ');
42 disp('Then we obtain all the linear functionals f by
      assigning arbitrary values to c2 and c4');
43 disp('Let c2 = a, c4 = b then c1 = a+b, c3 = -2b, c5
      = 0. ');
44 disp('So, W0 consists all linear functionals f of
      the form');
45 disp('f(x1,x2,x3,x4,x5) = (a+b)x1 + ax2 -2bx3 + bx4'
      );
46 disp('Dimension of W0 = 2 and basis {f1,f2} can be
      found by first taking a = 1, b = 0. Then a = 0,b
      = 1');
47 //end

```

Chapter 4

Polynomials

Scilab code Exa 4.3 Algebra of linear operators

```
1 //page 121
2 //Example 4.3
3 clc;
4 clear;
5 close;
6 disp('C is the field of complex numbers');
7 x = poly(0,"x");
8 f = x^2 + 2;
9 disp(f, 'f = ');
10 //part a
11 disp('if a = C and z belongs to C, then f(z) = z^2 +
      2');
12 disp(horner(f,2), 'f(2) = ');
13 disp(horner(f, (1+%i)/(1-%i)), 'f(1+%i/1-%i) = ');
14 disp('-----');
15 //part b
16 disp('If a is the algebra of all 2*2 matrices over C
      and');
17 B = [1 0;-1 2];
18 disp(B, 'B = ');
19 disp(2*eye(2,2) + B^2, 'then , f(B) = ');
```

```

20 disp('_____');
21 //part c
22 disp('If a is algebra of all linear operators on C^3
      ');
23 disp('And T is element of a as:');
24 disp('T(c1,c2,c3) = (i*2^1/2*c1,c2,i*2^1/2*c3)');
25 disp('Then, f(T)(c1,c2,c3) = (0,3*c2,0)');
26 disp('_____');
27 //part d
28 disp('If a is the algebra of all polynomials over C'
      ');
29 g = x^4 + 3*i;
30 disp(g, 'And, g = ');
31 disp(horner(f,g), 'Then f(g) = ');
32 //end

```

Scilab code Exa 4.7 Ideal of a polynomial

```

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

17 //end

Scilab code Exa 4.8 G C D of polynomials

```
1 //page 133
2 //Example 4.8
3 clc;
4 clear;
5 close;
6 x = poly(0,"x");
7 //part a
8 p1 = x + 2;
9 p2 = x^2 + 8*x + 16;
10 disp(p1,'p1 = ');
11 disp(p2,'p2 = ');
12 disp('M = (x+2)F[x] + (x^2 + 8x + 16)F[x]');
13 disp('We assert , M = F[x]');
14 disp('M contains:');
15 t = p2 - x*p1;
16 disp(t);
17 disp('And hence M contains:');
18 disp(t - 6*p1);
19 disp('Thus the scalar polynomial 1 belongs to M as
    well all its multiples');
20 disp('So, gcd(p1,p2) = 1');
21 disp('-----
    ');
22 //part b
23 p1 = (x - 2)^2*(x+%i);
24 p2 = (x-2)*(x^2 + 1);
25 disp(p1,'p1 = ');
26 disp(p2,'p2 = ');
27 disp('M = (x - 2)^2*(x+%i)F[x] + (x-2)*(x^2 + 1)');
28 disp('The ideal M contains p1 - p2 i.e.,');
29 disp(p1 - p2);
```



```

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

```

Scilab code Exa 4.9 Ideal of a Polynomial

```

1 //page 133
2 //Example 4.9
3 clc;
4 clear;
5 close;
6 disp('M is the ideal in F[x] generated by:');
7 disp('(x-1)*(x+2)^2');
8 disp('(x+2)^2*(x+3)');
9 disp('(x-3)', 'and');
10 x = poly(0, "x");
11 p1 = (x-1)*(x+2)^2;
12 p2 = (x+2)^2*(x-3);
13 p3 = (x-3);
14 disp('M = (x-1)*(x+2)^2 F[x] + (x+2)^2*(x-3) + (x-3)
      ');
15 disp('Then M contains:');
16 t = 1/2*(x+2)^2*((x-1) - (x-3));
17 disp(t);
18 disp('i.e., M contains (x+2)^2');
19 disp('and since, (x+2)^2 = (x-3)(x-7) - 17');
20 disp('So M contains the scalar polynomial 1. ');
21 disp('So, M = F[x] and given polynomials are
      relatively prime. ');
22 //end

```

Scilab code Exa 4.10 Reducible Polynomial

```
1 //page 135
2 //Example 4.10
3 clc;
4 clear;
5 close;
6 x = poly(0,"x");
7 P = x^2 + 1;
8 disp(P, 'P = ');
9 disp('P is reducible over complex numbers as: ');
10 disp('=',P);
11 disp('(x-i)(x+i)');
12 disp('Whereas, P is irreducible over real numbers as
    ..');
13 disp('=',P);
14 disp('(ax + b)(a''x + b''')');
15 disp('For, a,a'',b,b'' to be in R,');
16 disp('aa'' = 1');
17 disp('ab'' + ba'' = 0');
18 disp('bb'' = 1');
19 disp('=> a^2 + b^2 = 0');
20 disp('=> a = b = 0');
21 //end
```

Chapter 5

Determinants

Scilab code Exa 5.3 Two linear function

```
1 //page 143
2 //Example 5.3
3 clc;
4 clear;
5 close;
6 A = round(rand(2,2) *10 );
7 disp(A, 'A = ');
8 D1 = A(1,1)*A(2,2);
9 D2 = - A(1,2)*A(2,1);
10 disp(D1, 'D1(A) = ');
11 disp(D2, 'D2(A) = ');
12 disp(D1 + D2, 'D(A) = D1(A) + D2(A) = ');
13 disp('That is, D is a 2-linear function.');
```

Scilab code Exa 5.4 Alternating 3 Linear Functions

```
1 //page 145
```

```

2 //Example 5.4
3 clc;
4 clear;
5 close;
6 x = poly(0,"x");
7 A = [x 0 -x^2;0 1 0;1 0 x^3];
8 disp(A,'A = ');
9 disp('e1,e2,e3 are the rows of 3*3 identity matrix,
      then');
10 T = eye(3,3);
11 e1 = T(1,:);
12 e2 = T(2,:);
13 e3 = T(3,:);
14 disp(e1,'e1 = ');
15 disp(e2,'e2 = ');
16 disp(e3,'e3 = ');
17 disp('D(A) = D(x*e1 - x^2*e3, e2, e1 + x^3*e3)');
18 disp('Since, D is linear as a function of each row,');
19 disp('D(A) = x*D(e1,e2,e1 + x^3*e3) - x^2*D(e3,e2,e1
      + x^3*e3)');
20 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)');
21 disp('As D is alternating, So');
22 disp('D(A) = (x^4 + x^2)*D(e1,e2,e3)');
23 //end

```

Scilab code Exa 5.5 Determinant of a matrix

```

1 //page 147
2 //Example 5.5
3 clc;
4 clear;
5 close;
6 function [E1 , E2 , E3] = determinant(A)

```

```

7     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)]);
8     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)]);
9     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)]);
10    endfunction
11
12    //part a
13    x = poly(0,"x");
14    A = [x-1 x^2 x^3;0 x-2 1;0 0 x-3];
15    disp(A,'A = ');
16    [E1, E2, E3] = determinant(A);
17    disp(E1,'E1(A) = ');
18    disp(E2,'E2(A) = ');
19    disp(E3,'E3(A) = ');
20    disp('-----');
21    //part b
22    A = [0 1 0;0 0 1;1 0 0];
23    disp(A,'A = ');
24    [E1, E2, E3] = determinant(A);
25    disp(E1,'E1(A) = ');
26    disp(E2,'E2(A) = ');
27    disp(E3,'E3(A) = ');
28    //end

```

Scilab code Exa 5.6 Determinant of a matrix

```

1 //page 158
2 //Example 5.6
3 clc;
4 clear;

```

```

5 close;
6 disp('Given Matrix:');
7 A = [1 -1 2 3; 2 2 0 2; 4 1 -1 -1;1 2 3 0];
8 disp(A, 'A = ');
9 disp('After , Subtracting multiples of row 1 from rows
      2 3 4');
10 disp('R2 = R2 - 2*R1');
11 A(2,:) = A(2,:) - 2 * A(1,:);
12 disp('R3 = R3 - 4*R1');
13 A(3,:) = A(3,:) - 4 * A(1,:);
14 disp('R4 = R4 - R1');
15 A(4,:) = A(4,:) - A(1,:);
16 disp(A, 'A = ');
17 T = A;                                //Temporary matrix to store
      A
18 disp('We obtain the same determinant as before. ');
19 disp('Now, applying some more row transformations as
      : ');
20 disp('R3 = R3 - 5/4 * R2');
21 T(3,:) = T(3,:) - 5/4 * T(2,:);
22 disp('R4 = R4 - 3/4 * R2');
23 T(4,:) = T(4,:) - 3/4 * T(2,:);
24 B = T;
25 disp('We get B as: ');
26 disp(B, 'B = ');
27 disp('Now, determinant of A and B will be same');
28 disp(det(B), 'det A = det B = ');
29 //end

```

Scilab code Exa 5.7 Inverse of a matrix

```

1 //page 160
2 //Example 5.7
3 clc;
4 clear;

```

```

5 close;
6 x = poly(0,"x");
7 A = [x^2+x x+1;x-1 1];
8 B = [x^2-1 x+2;x^2-2*x+3 x];
9 disp(A,'A = ');
10 disp(B,'B = ');
11 disp(det(A),'det A = ');
12 disp(det(B),'det B = ');
13 disp('Thus, A is not invertible over K whereas B is
      invertible');
14 disp(inv(A)*det(A),'adj A = ');
15 disp(inv(B)*det(B),'adj B = ');
16 disp('(adj A)A = (x+1)I');
17 disp('(adj B)B = -6I');
18 disp(inv(B),'B inverse = ');
19 //end

```

Scilab code Exa 5.8 Inverse of a matrix

```

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

```

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

Chapter 6

Elementary Canonical Forms

Scilab code Exa 6.1 Characteristic Polynomial of a matrix

```
1 //page 184
2 //Example 6.1
3 clc;
4 clear;
5 close;
6 disp('Standard ordered matrix for Linear operator T
      on R^2 is:');
7 A = [0 -1;1 0];
8 disp(A, 'A = ');
9 disp('The characteristic polynomial for T or A is:')
10 x = poly(0, "x");
11 p = det(x*eye(2,2)-A);
12 disp(p);
13 disp('Since this polynomial has no real roots, T has
      no characteristic values.');
```

Scilab code Exa 6.2 Characteristic Polynomial of a matrix

```

1 //page 184
2 //Example 6.2
3 clc;
4 clear;
5 close;
6 A = [3 1 -1; 2 2 -1; 2 2 0];
7 disp(A, 'A = ');
8 disp('Characteristic polynomial for A is:');
9 p = poly(A, 'x');
10 disp(p);
11 disp('or ');
12 disp('(x-1)(x-2)^2');
13 r = roots(p);
14 [m,n] = size(A);
15 disp('The characteristic values of A are:');
16 disp(round(r));
17 B = A-eye(m,n);
18 disp(B, 'Now, A-I = ');
19 disp(rank(B), 'rank of A - I= ');
20 disp('So, nullity of T-I = 1');
21 a1 = [1 0 2];
22 disp(a1, 'The vector that spans the null space of T-I
    = ');
23 B = A-2*eye(m,n);
24 disp(B, 'Now, A-2I = ');
25 disp(rank(B), 'rank of A - 2I= ');
26 disp('T*alpha = 2*alpha if alpha is a scalar
    multiple of a2');
27 a2 = [1 1 2];
28 disp(a2, 'a2 = ');
29 //end

```

Scilab code Exa 6.3 Characteristic Polynomial of a matrix

```

1 //page 187

```

```

2 //Example 6.3
3 clc;
4 clear;
5 close;
6 disp('Standard ordered matrix for Linear operator T
      on  $R^3$  is:');
7 A = [5 -6 -6; -1 4 2; 3 -6 -4];
8 disp(A, 'A = ');
9 disp('xI - A = ');
10 B = eye(3,3);
11 x = poly(0,"x");
12 P = x*B - A;
13 disp(P);
14 disp('Applying row and column transformations:');
15 disp('C2 = C2 - C3');
16 P(:,2) = P(:,2) - P(:,3);
17 disp('=>');
18 disp(P);
19 disp('Taking (x-2) common from C2');
20 c = x-2;
21 P(:,2) = P(:,2) / (x-2);
22 disp('=>');
23 disp(' * ', c);
24 disp(P);
25 disp('R3 = R3 + R2');
26 P(3,:) = P(3,:) + P(2,:);
27 disp('=>');
28 disp(' * ', c);
29 disp(P);
30 P = [P(1,1) P(1,3); P(3,1) P(3,3)];
31 disp('=>');
32 disp(' * ', c);
33 disp(P);
34 disp('=>');
35 disp(' * ', c);
36 disp(det(P));
37 disp('This is the characteristic polynomial');
38 disp(A-B, 'Now, A - I = ');

```

```

39 disp(A-2*B, 'And, A- 2I = ');
40 disp(rank(A-B), 'rank(A-I) = ');
41 disp(rank(A-2*B), 'rank(A-2I) = ');
42 disp('W1,W2 be the spaces of characteristic vectors
    associated with values 1,2');
43 disp('So by theorem 2, T is diagonalizable');
44 a1 = [3 -1 3];
45 a2 = [2 1 0];
46 a3 = [2 0 1];
47 disp(a1, 'Null space of (T- I) i.e basis of W1 is
    spanned by a1 = ');
48 disp('Null space of (T- 2I) i.e. basis of W2 is
    spanned by vectors x1,x2,x3 such that x1 = 2x1 +
    2x3');
49 disp('One example are;');
50 disp(a2, 'a2 = ');
51 disp(a3, 'a3 = ');
52 disp('The diagonal matrix is:');
53 D = [1 0 0 ;0 2 0;0 0 2];
54 disp(D, 'D = ');
55 disp('The standard basis matrix is denoted as:');
56 P = [a1;a2;a3]';
57 disp(P, 'P = ');
58 disp(A*P, 'AP = ');
59 disp(P*D, 'PD = ');
60 disp('That is , AP = PD');
61 disp('=> inverse(P)*A*P = D');
62 //end

```

Scilab code Exa 6.4 Diagonalizable Operator

```

1 //page 193
2 //Example 6.4
3 clc;
4 clear;

```

```

5 close;
6 x = poly(0,"x");
7 A = [5 -6 -6; -1 4 2; 3 -6 -4]; //Matrix given
    in Example 3
8 disp(A,'A = ');
9 f = (x-1)*(x-2)^2;
10 disp('Characteristic polynomial of A is:');
11 disp('f = (x-1)(x-2)^2');
12 disp(f,'i.e., f = ');
13 p = (x-1)*(x-2);
14 disp((A-eye(3,3))*(A-2 * eye(3,3)),'(A-I)(A-2I) = ');
    ;
15 disp('Since, (A-I)(A-2I) = 0. So, Minimal polynomial
    for above is:');
16 disp(p,'p = ');
17 disp('-----');
18 A = [3 1 -1; 2 2 -1;2 2 0]; //Matrix given in
    Example 2
19 disp(A,'A = ');
20 f = (x-1)*(x-2)^2;
21 disp('Characteristic polynomial of A is:');
22 disp('f = (x-1)(x-2)^2');
23 disp(f,'i.e., f = ');
24 disp((A-eye(3,3))*(A-2 * eye(3,3)),'(A-I)(A-2I) = ');
    ;
25 disp('Since, (A-I)(A-2I) is not equal to 0. T is not
    diagonalizable. So, Minimal polynomial cannot be
    p. ');
26 disp('-----');
27 A = [0 -1;1 0];
28 disp(A,'A = ');
29 f = x^2 + 1;
30 disp('Characteristic polynomial of A is:');
31 disp(f,'f = ');
32 disp(A^2 + eye(2,2),'A^2 + I = ');
33 disp('Since, A^2 + I = 0, so minimal polynomial is')
    ;
34 p = x^2 + 1;

```

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

Scilab code Exa 6.5 Characteristic Polynomial of matrix

```
1 //page 197
2 //Example 6.5
3 clc;
4 clear;
5 close;
6 A = [0 1 0 1;1 0 1 0;0 1 0 1;1 0 1 0];
7 disp(A, 'A = ');
8 disp('Computing powers on A: ');
9 disp(A*A, 'A^2 = ');
10 disp(A*A*A, 'A^3 = ');
11 deff('[p] = p(x)', 'p = x^3 - 4*x');
12 disp('if p = x^3 - 4x, then');
13 disp(p(A), 'p(A) = ');
14 x = poly(0, 'x');
15 f = x^3 - 4*x;
16 disp(f, 'Minimal polynomial for A is: ');
17 disp(roots(f), 'Characteristic values for A are:');
18 disp(rank(A), 'Rank(A) = ');
19 disp(round(poly(A, 'x')), 'So, from theorem 2,
    characteristic polynomial for A is:');
20 //end
```

Scilab code Exa 6.12 Symmetric and skew symmetric matrix

```
1 //page 210
2 //Example 6.12
3 clc;
4 clear;
```

```

5 close;
6 A = round(rand(3,3) * 10);
7 disp(A, 'A = ');
8 disp('A transpose is:');
9 disp(A', 'A'' = ');
10 if A' == A then
11     disp('Since , A'' = A, A is a symmetric matrix.')
12     ;
13 else
14     disp('Since , A'' is not equal to A, A is not a
15         symmetric matrix. ');
16 end
17 if A' == -A then
18     disp('Since , A'' = -A, A is a skew-symmetric
19         matrix. ');
20 else
21     disp('Since , A'' is not equal to -A, A is not a
22         skew-symmetric matrix. ');
23 end
24 A1 = 1/2*(A + A');
25 A2 = 1/2*(A - A');
26 disp('A can be expressed as sum of A1 and A2');
27 disp('i.e., A = A1 + A2');
28 disp(A1, 'A1 = ');
29 disp(A2, 'A2 = ');
30 disp(A1 + A2, 'A1 + A2 = ');
31 //end

```

Chapter 7

The Rational and Jordan Forms

Scilab code Exa 7.3 Linear operator annihilator

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



```

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

```

Scilab code Exa 7.6 Characteristic and minimal polynomial of matrix

```

1 //page 247
2 //Example 7.6
3 clc;
4 clear;
5 close;
6 disp('A = ');
7 disp('2  0  0');
8 disp('a  2  0');
9 disp('b  c  -1');
10 a = 1;
11 b = 0;
12 c = 0;
13 A = [2 0 0;a 2 0;b c -1];
14 disp(A, 'A = ');
15 disp('Characteristic polynomial for A is:');
16 disp(poly(A, "x"), 'p = ');
17 disp('In this case, minimal polynomial is same as
      characteristic polynomial. ');
18 disp('-----');

```

```

19 a = 0;
20 b = 0;
21 c = 0;
22 A = [2 0 0;a 2 0;b c -1];
23 disp(A, 'A = ');
24 disp('Characteristic polynomial for A is:');
25 disp(poly(A,"x"), 'p = ');
26 disp('In this case, minimal polynomial is:');
27 disp('(x-2)(x+1)');
28 disp('or');
29 x = poly(0,"x");
30 s = (x-2)*(x+1);
31 disp(s);
32 disp('(A-2I)(A+I) = ');
33 disp('0 0 0');
34 disp('3a 0 0');
35 disp('ac 0 0');
36 disp('if a = 0, A is similar to diagonal matrix.')
37 //end

```

Scilab code Exa 7.7 Characteristic and minimal polynomial of matrix

```

1 //page 247
2 //Example 7.7
3 clc;
4 clear;
5 close;
6 disp('A = ');
7 disp('2 0 0 0');
8 disp('1 2 0 0');
9 disp('0 0 2 0');
10 disp('0 0 a 2');
11 disp('Considering a = 1');
12 A = [2 0 0 0;1 2 0 0;0 0 2 0;0 0 1 2];
13 p = poly(A,"x");

```

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

Chapter 8

Inner Product Spaces

Scilab code Exa 8.1 Standard Inner Product

```
1 //page 271
2 //Example 8.1
3 clc;
4 clear;
5 close;
6 n = round(rand() * 10 + 2);
7 a = round(rand(1,n) * 10)
8 b = round(rand(1,n) * 10)
9 disp(n, 'n = ');
10 disp(a, 'a = ');
11 disp(b, 'b = ');
12 disp(a*b', 'Then, (a|b) = ');
13 //end
```

Scilab code Exa 8.2 Standard Inner Product

```
1 //page 271
2 //Example 8.2
```

```

3  clc;
4  clear;
5  close;
6  a = round(rand(1,2) * 10)
7  b = round(rand(1,2) * 10)
8  disp(a, 'a = ');
9  disp(b, 'b = ');
10 x1 = a(1);
11 x2 = a(2);
12 y1 = b(1);
13 y2 = b(2);
14 t = x1*y1 - x2*y1 - x1*y2 + 4*x2*y2;
15 disp(t, 'Then, a|b = ');
16 //end

```

Scilab code Exa 8.9 Standard Inner Product

```

1  //page 278
2  //Example 8.9
3  clc;
4  clear;
5  close;
6  a = round(rand(1,2) * 10);
7  x = a(1);
8  y = a(2);
9  b = [-y x];
10 disp(a, '(x,y) = ');
11 disp(b, '(-y,x) = ');
12 disp('Inner product of these vectors is:');
13 t = -x*y + y*x;
14 disp(t, '(x,y)|(-y,x) = ');
15 disp('So, these are orthogonal. ');
16 disp('-----');
17 disp('If inner product is defined as:');
18 disp('(x1,x2)|(y1,y2) = x1y1- x2y1 - x1y2 + 4x2y2');

```

```

19 disp('Then, (x,y)|(-y,x) = -x*y+y^2-x^2+4*x*y = 0 if
    ,');
20 disp('y = 1/2(-3 + sqrt(13))*x');
21 disp('or');
22 disp('y = 1/2(-3 - sqrt(13))*x');
23 disp('Hence,');
24 if y == 1/2*(-3 + sqrt(13))*x | y == 1/2*(-3 - sqrt
    (13))*x then
25 disp(a);
26 disp('is orthogonal to');
27 disp(b);
28 else
29 disp(a);
30 disp('is not orthogonal to');
31 disp(b);
32 end
33 //end

```

Scilab code Exa 8.12 Orthogonal Vectors

```

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

```

```

15 a2 = b2 - ((b2*b1')')/25*b1);
16 disp(a2, 'a2 = ');
17 a3 = b3 - ((b3*b1')')/25*b1) - ((b3*a2')')/25*a2);
18 disp(a3, 'a3 = ');
19 disp('{a1, a2, a3} are mutually orthogonal and hence
      forms orthogonal basis for R^3');
20 disp('Any arbitrary vector {x1, x2, x3} in R^3 can be
      expressed as:');
21 disp('y = {x1, x2, x3} = (3*x1 + 4*x3)/25*a1 + (-4*x1
      + 3*x3)/25*a2 + x2/9*a3');
22 x1 = 1;
23 x2 = 2;
24 x3 = 3;
25 y = (3*x1 + 4*x3)/25*a1 + (-4*x1 + 3*x3)/25*a2 + x2
      /9*a3;
26 disp(x1, 'x1 = ');
27 disp(x2, 'x2 = ');
28 disp(x3, 'x3 = ');
29 disp(y, 'y = ');
30 disp('i.e. y = [x1 x2 x3], according to above
      equation. ');
31 disp('Hence, we get the orthonormal basis as:');
32 disp(' ', ' ', 1/5*a1);
33 disp(' ', ' ', 1/5*a2);
34 disp(1/9*a3);
35 //end

```

Scilab code Exa 8.13 Orthogonal Vectors

```

1 //page 283
2 //Example 8.13
3 clc;
4 clear;
5 close;
6 A = rand(2,2);

```

```

7 A(1,:) = A(1,:) + 1; //so b1 is not equal to zero
8 a = A(1,1);
9 b = A(1,2);
10 c = A(2,1);
11 d = A(2,2);
12 b1 = A(1,:);
13 b2 = A(2,:);
14 disp(A, 'A = ');
15 disp(b1, 'b1 = ');
16 disp(b2, 'b2 = ');
17 disp('Applying the orthogonalization process to b1,
      b2: ');
18 a1 = [a b];
19 a2 = (det(A)/(a^2 + b^2))*[-b' a'];
20 disp(a1, 'a1 = ');
21 disp(a2, 'a2 = ');
22 disp('a2 is not equal to zero if and only if b1 and
      b2 are linearly independent. ');
23 disp('That is, if determinant of A is non-zero. ');
24 //end

```

Scilab code Exa 8.14 Orthogonal Projection

```

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

```



```

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

```

Scilab code Exa 8.15 Orthogonal sets

```

1 //page 288
2 //Example 8.15
3 clc;
4 clear;
5 close;
6 //part c
7 disp('f = (sqrt(2)*cos(2*pi*t) + sqrt(2)*sin(4*pi*t)
      )^2');
8 disp('Integration (f dt) in limits 0 to 1 = ');
9 x0 = 0;
10 x1 = 1;
11 X = integrate('(sqrt(2)*cos(2*pi*t) + sqrt(2)*sin
      (4*pi*t))^2','t',x0,x1);
12 disp(X);
13 //end

```

Scilab code Exa 8.17 Inner product space and orthogonal projection

```

1 //page 294
2 //Example 8.17
3 //Equation given in example 14 is used.
4 clc;

```

```

5 clear;
6 close;
7 function [m] = transform(x,y,z)
8     x1 = 3*x;
9     x2 = 12*y;
10    x3 = -z;
11    m = [x1 x2 x3];
12 endfunction
13
14 disp('Matrix of projection E in orthonormal basis is
      : ');
15 t1 = transform(3,3,3);
16 t2 = transform(12,12,12);
17 t3 = transform(-1,-1,-1);
18 A = [t1; t2; t3];
19 disp(A, 'A = 1/154 * ');
20 A1 = (conj(A))';
21 disp(A1, 'A* = ');
22 disp('Since , E = E* and A = A*, then A is also the
      matrix of E* ');
23 a1 = [154 0 0];
24 a2 = [145 -36 3];
25 a3 = [-36 10 12];
26 disp(a1, 'a1 = ');
27 disp(a2, 'a2 = ');
28 disp(a3, 'a3 = ');
29 disp('{a1, a2, a3} is the basis. ');
30 Ea1 = [9 36 -3];
31 Ea2 = [0 0 0];
32 Ea3 = [0 0 0];
33 disp(Ea1, 'Ea1 = ');
34 disp(Ea2, 'Ea2 = ');
35 disp(Ea3, 'Ea3 = ');
36 B = [-1 0 0; -1 0 0; 0 0 0];
37 disp('Matrix B of E in the basis is: ');
38 disp(B, 'B = ');
39 B1 = (conj(B))';
40 disp(B1, 'B* = ');

```

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

Scilab code Exa 8.28 Unitary matrix

```
1 //page 307
2 //Example 8.28
3 clc;
4 clear;
5 close;
6 disp('x1 and x2 are two real nos. i.e.,  $x_1^2 + x_2^2 = 1$ ');
7 x1 = rand();
8 x2 = sqrt(1 - x1^2);
9 disp(x1, 'x1 = ');
10 disp(x2, 'x2 = ');
11 B = [x1 x2 0; 0 1 0; 0 0 1];
12 disp(B, 'B = ');
13 disp('Applying Gram-Schmidt process to B:');
14 a1 = [x1 x2 0];
15 a2 = [0 1 0] - x2 * [x1 x2 0];
16 a3 = [0 0 1];
17 disp(a1, 'a1 = ');
18 disp(a2, 'a2 = ');
19 disp(a3, 'a3 = ');
20 U = [a1; a2/x1; a3];
21 disp(U, 'U = ');
22 M = [1 0 0; -x2/x1 1/x1 0; 0 0 1];
23 disp(M, 'M = ');
24 disp(inv(M) * U, 'inverse(M) * U = ');
25 disp('So, B = inverse(M) * U');
26 //end
```

Chapter 10

Bilinear Forms

Scilab code Exa 10.4 Bilinear Form of vectors

```
1 //page 363
2 //Example 10.4
3 clc;
4 clear;
5 close;
6 disp('a = [x1 x2]');
7 disp('b = [y1 y2]');
8 disp('f(a,b) = x1*y1 + x1*y2 + x2*y1 + x2*y2');
9 disp('so, f(a,b) = ');
10 disp(' [x1 x2] * | 1 1 | * |y1|');
11 disp(' | 1 1 | |y2|');
12 disp('So the matrix of f in standard order basis B =
    {e1, e2} is:');
13 fb = [1 1; 1 1];
14 disp(fb, '[f]B = ');
15 P = [1 1; -1 1];
16 disp(P, 'P = ');
17 disp('Thus, [f]B'' = P''*[f]B*P');
18 fb1 = P' * fb * P;
19 disp(fb1, '[f]B'' = ');
20 //end
```

Scilab code Exa 10.5 Bilinear Form of vectors

```
1 //page 365
2 //Example 10.5
3 clc;
4 clear;
5 close;
6 n = round(rand() * 10 + 2);
7 a = round(rand(1,n) * 10);
8 b = round(rand(1,n) * 10);
9 disp(n, 'n = ');
10 disp(a, 'a = ');
11 disp(b, 'b = ');
12 f = a * b';
13 disp(f, 'f(a,b) = ');
14 disp('f is non-degenerate bilinear form on R^n.');
```
