Scilab Manual for Numerical Techniques by Dr Javed Dhillon Electrical Engineering ABES Engineering College[1]

Solutions provided by
Dr Javed Dhillon
Electrical Engineering
ABES Engineering College

November 25, 2026

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

Contents

List of Scilab Solutions 3

1 Solution of linear equation for under determined and over determined cases 5

2 Determination of Eigen Value & Eigen Vectors 8

3 Determination of roots of a polynomial 11

4 Determination of polynomial using method for least square curve fitting 14

5 Determination of Polynomial fit, Analyzing Residuals, exponential fit & error bounds from the given data 18

6 Solution of differential equation using 4th order Runga-Kutta method 24

7 Solution of differential equation using revised Euler method 27

8 Solution of difference equation 29

9 Determination of time response of an R-L-C circuit 31

List of Experiments

Solution 1.1	Experiment 1	5
Solution 2.2	Experiment 2	8
Solution 3.3	Experiment 3	11
Solution 4.4	Experiment 4	14
Solution 5.5	Experiment 5	18
Solution 6.6	Experiment 6	24
Solution 7.7	Experiment 7	27
Solution 8.8	Experiment 8	29
Solution 9.9	Experiment 9	31

List of Figures

1.1 Experiment 1 6

2.1 Experiment 2 9

3.1 Experiment 3 12

4.1 Experiment 4 15
4.2 Experiment 4 16

5.1 Experiment 5 19
5.2 Experiment 5 19

6.1 Experiment 6 25

7.1 Experiment 7 28

8.1 Experiment 8 30

9.1 Experiment 9 32

Experiment: 1

Solution of linear equation for under determined and over determined cases

Scilab code Solution 1.1 Experiment 1

```
1 //Operating system: Windows 8
2 //SCILAB Ver: 5.5.2
3 //Expriment No. 1
4 //Objective: Solution of linear equation for under
    determined and over determined cases
5 // For example to enter a linear equation is given
    as
6 //          x + 4y = 18
7 //          2x +3y =16
8 //Enter the no of row of matrix A: 2
9 //Enter the no of column of matrix A: 2
10 //Enter the element no (1,1):1
11 //Enter the element no (1,2): 4
12 //Enter the element no (2,1): 2
13 //Enter the element no (2,2): 3
```

```
Enter the no of row of matrix A: 2
Enter the no of column of matrix A: 2
Enter the element no (1,1):
1
Enter the element no (1,2):
4
Enter the element no (2,1):
2
Enter the element no (2,2):
3
Enter the no of row of matrix B: 2
Enter the no of column of matrix B: 1
Enter the element no (1,1):
18
Enter the element no (2,1):
16

 output1 (pinv)

    2.
    4.

 output2 (\)

    2.
    4.
```

Figure 1.1: Experiment 1

```
14 //Enter the no of row of matrix B: 2
15 //Enter the no of column of matrix B: 1
16 //Enter the element no (1,1): 18
17 //Enter the element no (2,1): 16
18
19
20 clc
21 clear
22 m = input("Enter the no of row of matrix A: ")
23 n = input("Enter the no of column of matrix A: ")
24 for i=1:m
25     for j=1:n
26         mprintf("Enter the element no (%d,%d):",i,j)
27         A(i,j)=input("")
28     end
29 end
30
31 m = input("Enter the no of row of matrix B: ")
32 n = input("Enter the no of column of matrix B: ")
33 for i=1:m
34     for j=1:n
35         mprintf("Enter the element no (%d,%d):",i,j)
36         B(i,j)=input("")
37     end
38 end
39 C = pinv(A)*B
40 disp(C,"output1 (pinv)")
41 C1 = A\B
42 disp(C1,"output2 (\)")
43 //C2 = inv(A)*B
44 //disp(C2,"output3 (inv)")
```

Experiment: 2

Determination of Eigen Value & Eigen Vectors

Scilab code Solution 2.2 Experiment 2

```
1  //Operating system: Windows 8
2  //SCILAB Ver: 5.5.2
3  //Expriment No. 2
4  //Objective: Determination of Eigen value & Eigen
       Vectors
5  //or example to enter a matrix is given as
6  //A = 1 6
7  //    4 5
8  //Enter the no of row of matrix A: 2
9  //Enter the no of column of matrix A: 2
10 //Enter the element no (1,1):1
11 //Enter the element no (1,2):6
12 //Enter the element no (2,1):4
13 //Enter the element no (2,2):5
14 clc
15 clear
16 m = input("Enter the no of row of matrix A: ")
```

```
Enter the no of row of matrix A: 2
Enter the no of column of matrix A: 2
Enter the element no (1,1):
1
Enter the element no (1,2):
4
Enter the element no (2,1):
4
Enter the element no (2,2):
5

 Eigenvalues :

  - 0.8506508    0.5257311
    0.5257311    0.8506508

 Eigenvectors :

  - 1.472136    0.
    0.          7.472136
```

Figure 2.1: Experiment 2

```
17 n = input("Enter the no of column of matrix A: ")
18 for i=1:m
19     for j=1:n
20         mprintf("Enter the element no (%d,%d):",i,j)
21         A(i,j)=input("")
22     end
23 end
24 [eVal eVect]=spec(A)
25 disp(eVal,"Eigenvalues :")
26 disp(eVect,"Eigenvectors :")
```

Experiment: 3

Determination of roots of a polynomial

Scilab code Solution 3.3 Experiment 3

```
1  //Operating system: Windows 8
2  //SCILAB Ver: 5.5.2
3  //Expriment No. 3
4  //Solver required: Stixbox: Version 2.3-1
5  //Objective: Determination of roots of a polynomial
6  // For example to find the roots of a polymomials
7  // y = x^5 + 4x^3 - 9x +5
8  //
9  //Enter the Degree of polymomial: 5
10 //Enter the coeff of x^5: 1
11 //Enter the coeff of x^4: 0
12 //Enter the coeff of x^3: 4
13 //Enter the coeff of x^2: 0
14 //Enter the coeff of x^1: -9
15 //Enter the coeff of x^0: 5
16
17 clc
```

Scilab 5.5.2 Console

```
Enter the Degree of polymomial: 5
Enter the coeff of x^5:
1
Enter the coeff of x^4:
0
Enter the coeff of x^3:
4
Enter the coeff of x^2:
0
Enter the coeff of x^1:
-9
Enter the coeff of x^0:
5

 Polymomial

                3   5
    5 - 9x + 4x + x

 Roots of polynomial

  - 0.0612282 + 2.3724442i
  - 0.0612282 - 2.3724442i
  - 1.4352725
    0.7788645 + 0.1090464i
    0.7788645 - 0.1090464i
```

Figure 3.1: Experiment 3

```
clear
m = input("Enter the Degree of polymomial: ")
j=0
for i=m+1:-1:1
    mprintf("Enter the coeff of x^%d:",i-1)
    A(1,i)=input("")
    j=j+1
    Ax(1,j)=A(1,i)
end
P = poly(A,"x","coeff")
R = roots(Ax)
disp(P,"Polymomial")
disp(R,"Roots of polynomial")
```

Experiment: 4

Determination of polynomial using method for least square curve fitting

Scilab code Solution 4.4 Experiment 4

```
1 //Operating system: Windows 8
2 //SCILAB Ver: 5.5.2
3 //Expriment No. 4
4 //Solver required: Stixbox: Version 2.3-1
5 //Objective: Determination of polynomial using
     method for least square curve fitting
6 //Try to fit the equation x^6 + 2x^5 + x^3 -x +3 in
     to the Two degree polynomial
7  //Enter the Degree of polymomial: 6
8 //Enter the coeff of x^6: 1
9 //Enter the coeff of x^5: 2
10 //Enter the coeff of x^4: 0
11 //Enter the coeff of x^3: 1
```

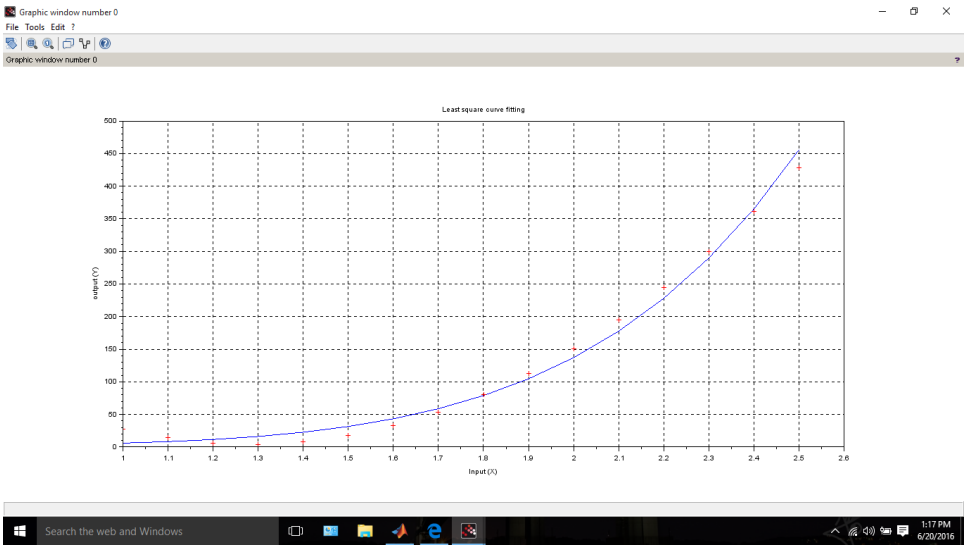


Figure 4.1: Experiment 4

```
12 //Enter the coeff of x^2: 0
13 //Enter the coeff of x^1: -1
14 //Enter the coeff of x^0: 3
15 //Enter the Degree of polymomial in which you want
      to fit: 2
16 clear
17 clc
18 m = input("Enter the Degree of polymomial: ")
19 j=0
20 for i=m+1:-1:1
21     mprintf("Enter the coeff of x^%d:",i-1)
22     A(1,i)=input("")
23     j=j+1
24     Ax(1,j)=A(1,i)
25 end
26 nx = input("Enter the Degree of polymomial in which
      you want to fit: ")
27 P = poly(A,"x","coeff")
28 x = 1:.1:2.5
29 y1 = polyval(Ax,x)
```

```
Scilab 5.5.2 Console

Enter the Degree of polymomial: 6
Enter the coeff of x^6:
1
Enter the coeff of x^5:
2
Enter the coeff of x^4:
0
Enter the coeff of x^3:
1
Enter the coeff of x^2:
0
Enter the coeff of x^1:
-1
Enter the coeff of x^0:
3
Enter the Degree of polymomial in which you want to fit: 2

 Polynomical Enter

              3    5   6
    3 - x + x + 2x + x

 Polynomical after LSQ fitting

                                        2
    479.43919 - 739.57735x + 287.61331x
```

Figure 4.2: Experiment 4

```
30 n = polyfit(x,y1,nx)
31 y2 = polyval(n,x)
32 P2 = poly(n(nx+1:-1:1),"x","coeff")
33 plot(x,y1,'-',x,y2,'+r')
34 title('Least square curve fitting')
35 xlabel('Input (X)')
36 ylabel('Output (Y)')
37 set(gca(),"grid",[1 1])
38 disp(P,"Polynomical Enter")
39 disp(P2,"Polynomical after LSQ fitting")
```

Experiment: 5

Determination of Polynomial fit, Analyzing Residuals, exponential fit & error bounds from the given data

Scilab code Solution 5.5 Experiment 5

```
1 //Operating system: Windows 8
2 ////SCILAB Ver: 5.5.2
3 //Experiment No. 5
4 //Solver required: Stixbox: Version 2.3-1
5 //Objective: Determination of Polynomial fit,
    Analyzing Residuals, exponential fit & error
    bounds form the given data
6
7 //census is the given input data of year and
    population for the experiment no 5
8 //Output provides three figure.....
```

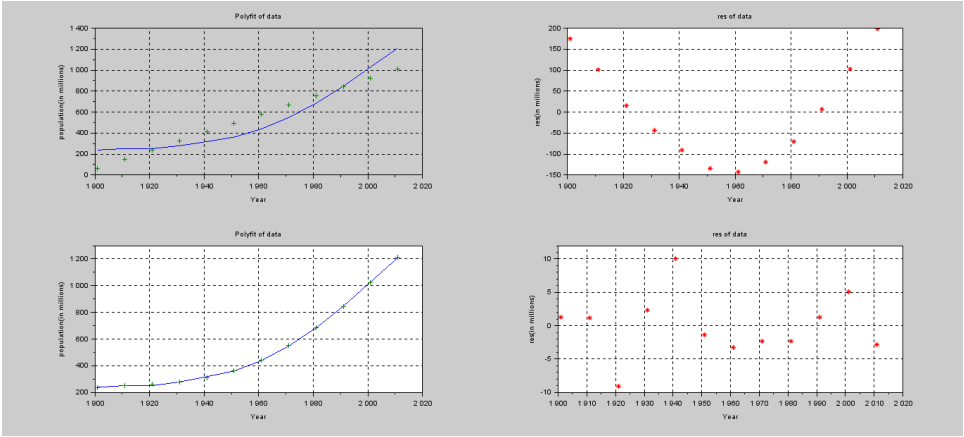


Figure 5.1: Experiment 5

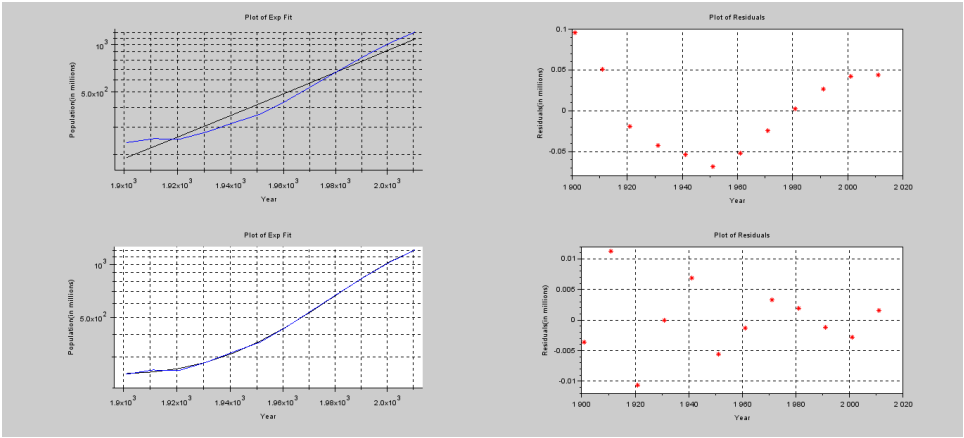


Figure 5.2: Experiment 5

```
9  //figure(1) for Polynomial fit and Analyzing
       Residuals
10 //figure (2) for exponential fit  and Analyzing
       Residuals
11 //and figure (3) for error bounds
12
13
14 clc
15 clear all
16 census=[1901 238396327;
17 1911     252093390;
18 1921     251321213;
19 1931     278977238;
20 1941     318660580;
21 1951     361088090;
22 1961     439234771;
23 1971     548159652;
24 1981     683329097;
25 1991     846421039;
26 2001     1028737436;
27 2011     1210854977]
28 year = census(:,1);
29 pop = census(:,2)./1000000;
30
31
32 //Normalisation of data
33 syear = (year-mean(year))./stdev(year);
34
35 //Fit the data in 1 deg polynomial
36 A1 = polyfit(syear,pop,1);
37 P=polyval(A1,syear);
38 figure(1)
39 subplot(2,2,1)
40 plot(year,pop,'-',year,P,'+')
41 title('Polyfit of data')
42 xlabel('Year')
43 ylabel('population(in millions)')
44 set(gca(),"grid",[1 1])
```

```
45
46 //Analyzing Residuals
47 res1 = pop-P;
48 subplot(2,2,2)
49 plot(year,res1,'r*')
50 title('res of data')
51 xlabel('Year')
52 ylabel('res(in millions)')
53 set(gca(),"grid",[1 1])
54 subplot(2,2,4)
55
56 //Fit the data in 4 deg polynomial
57 A2 = polyfit(syear,pop,4);
58 P1=polyval(A2,syear);
59 subplot(2,2,3)
60 plot(year,pop,'-',year,P1,'+')
61 title('Polyfit of data')
62 xlabel('Year')
63 ylabel('population(in millions)')
64 set(gca(),"grid",[1 1])
65
66
67 //Analyzing Residuals
68 res2 = pop-P1;
69 subplot(2,2,4)
70 plot(year,res2,'r*')
71 title('res of data')
72 xlabel('Year')
73 ylabel('res(in millions)')
74 set(gca(),"grid",[1 1]) //for grid on
75
76
77 //Exponential Fit 1 degree polynomial
78 figure(2)
79 subplot(2,2,1)
80 log1 = polyfit(syear,log10(pop),1);
81 logr1 = 10.^polyval(log1,syear);
82 plot2d1("oll",year,[logr1 pop])
```

```
83  title('Plot of Exp Fit')
84  ylabel('Population(in millions)')
85  xlabel('Year')
86  set(gca(),"grid",[1 1])
87
88  //Analyzing the residuals
89  logr1 = polyval(log1,syear);
90  res1 = log10(pop)-logr1;
91  subplot(2,2,2)
92
93  plot(year,res1,'r*')
94  title('Plot of Residuals')
95  ylabel('Residuals(in millions)')
96  xlabel('Year')
97  set(gca(),"grid",[1 1])
98
99  //Exponential Fit 4 degree polynomial
100 log1 = polyfit(syear,log10(pop),4);
101 logr1 = 10.^polyval(log1,syear);
102 subplot(2,2,3)
103 title('Plot of Exp Fit')
104 plot2d1("oll",year,[logr1 pop])
105 ylabel('Population(in millions)')
106 xlabel('Year')
107 set(gca(),"grid",[1 1])
108
109 //Analyzing the residuals
110 logr1 = polyval(log1,syear);
111 res1 = log10(pop)-logr1;
112 subplot(2,2,4)
113 plot(year,res1,'r*')
114 title('Plot of Residuals')
115 ylabel('Residuals(in millions)')
116 xlabel('Year')
117 set(gca(),"grid",[1 1])
118
119
120 //%Error Bound
```

```
121
122 figure(3)
123 [p2,s2] = polyfit(syear,pop,2);
124 [pop2,del2] = polyval(p2,syear,s2);
125 plot(year,pop,'+',year,pop2,'-',year,pop2+2*del2,'*:
      ',year,pop2-2*del2,'r:')
126 ylabel('Population(in millions)')
127 xlabel('Year')
128 set(gca(),"grid",[1 1])
```

Experiment: 6

Solution of differential equation using 4th order Runga-Kutta method

Scilab code Solution 6.6 Experiment 6

```
1  //Operating system: Windows 8
2  ////SCILAB Ver: 5.5.2
3  //Expriment No. 6
4  //Objective: Solution of differntial equation using
       4th order R-K method
5
6  // To find the solution of differntial eq  dy/dx or
       y1
7  // Define the equation y1 in function "dequation" (x
       -y)/(x+y)
8
9  clc
10 clear
11 function y1 = dequation(x,y)
12     y1 = (x-y)/(x+y)
```

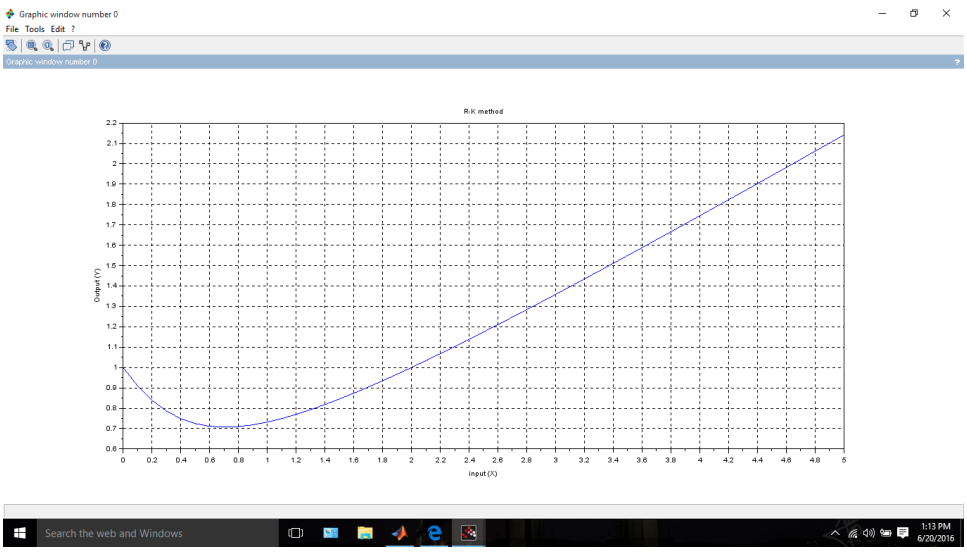


Figure 6.1: Experiment 6

```
endfunction

x(1)=0;
y(1)=1;
h=0.1;
a=0;
b=5;
n = round((b-a)/h)
for i=1:n

xn = x(i)
yn= y(i)
k1 = h*dequation(xn,yn)
k2 = h*dequation(xn +h/2,yn +k1/2)
k3 = h*dequation(xn +h/2, yn +k2/2 )
k4 = h*dequation(xn + h, yn +k3)
y(i+1) = yn + (1/6)*(k1 + 2*k2 + 2*k3 + k4)
x(i+1)=xn+h;
end

```

```
33 plot(x,y,'-')
34 title('R-K method')
35 xlabel('input (X)')
36 ylabel('Output (Y)')
37 set(gca(),"grid",[1 1])
```

Experiment: 7

Solution of differential equation using revised Euler method

Scilab code Solution 7.7 Experiment 7

```
1 //Operating system: Windows 8
2 ////SCILAB Ver: 5.5.2
3 //Expriment No. 7
4 //Objective: Solution of differntial equation using
    revised Euler method
5
6 // To find the solution of differntial eq  dy/dx or
    y1
7 // Define the equation y1 in function "dequation" (x
    -y)/(x+y)
8 clc
9 clear
10 function y1 = dequation(x,y)
11     y1 = (x-y)/(x+y)
12 endfunction
13
14 x(1)=0;
```

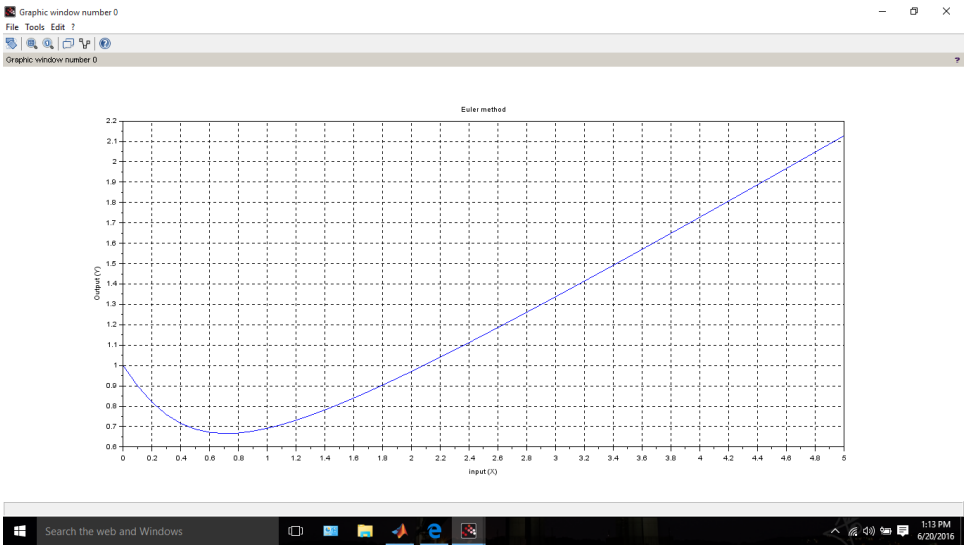


Figure 7.1: Experiment 7

```
y(1)=1;
h=0.1;
a=0;
b=5;
n = round((b-a)/h)
for i=1:n
    y(i+1)=y(i)+h*dequation(x(i),y(i));
    x(i+1)=x(i)+h;
end

plot(x,y,'-')
title('Euler method')
xlabel('input (X)')
ylabel('Output (Y)')
set(gca(),"grid",[1 1])
```

Experiment: 8

Solution of difference equation

Scilab code Solution 8.8 Experiment 8

```
1  //Operating system: Windows 8
2  ////SCILAB Ver: 5.5.2
3  //Experiment No. 8
4  //Objective: Solution of difference equation
5
6  //In this programme differnece equation has been
       solved by using filter command
7  // x in the input vector, a and b discribed the
       filter and y is the filtered data
8  //
9  clear all
10 x = [1  2  3  4  5];
11 a = 1;
12 b =[1/4  1/4  1/4  1/4  1/4];
13 t =1:length(x);
14 y = filter(b,a,x);
15 plot(t,x,'-',t,y,'-')
16 title('Solution of difference equation')
17 xlabel('input (X)')
```

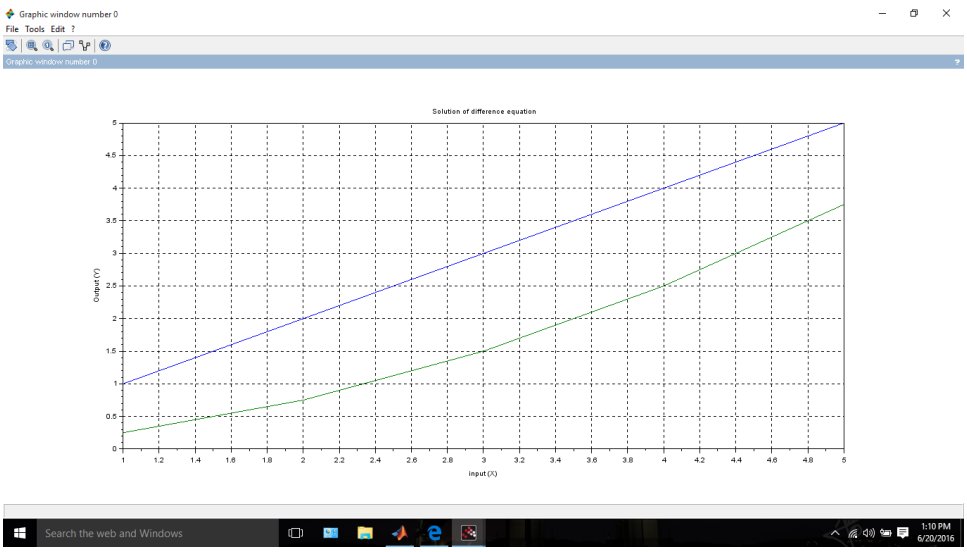


Figure 8.1: Experiment 8

```
18 ylabel('Output (Y)')
19 set(gca(),"grid",[1 1])
```

Experiment: 9

Determination of time response of an R-L-C circuit

Scilab code Solution 9.9 Experiment 9

```
1 //Operating system: Windows 8
2 ////SCILAB Ver: 5.5.2
3 //Experiment No. 9
4 //Objective: Determination of time response of an R-
    L-C circuit
5 clc
6 clear all
7 path = get_absolute_file_path('Experiment9.sce')
8 chdir(path)
9 xcos('Experiment9.xcos')
```

This code can be downloaded from the website wwww.scilab.in

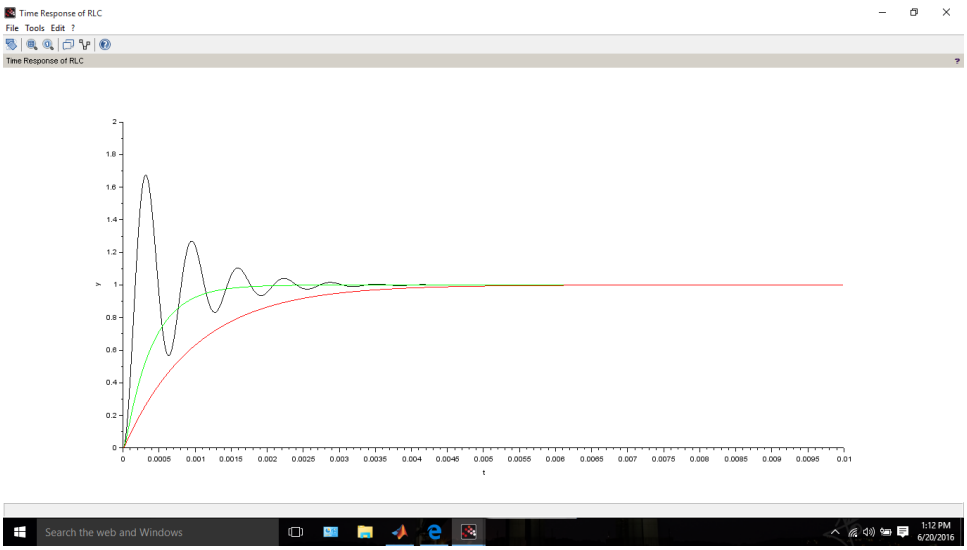


Figure 9.1: Experiment 9