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

June 2, 2025

[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 Experiments

List of Figures

Experiment: 1

Solution of linear equation for under determined and over determined cases

Scilab code Solution 1.1 Experiment 1

```
//Operating system: Windows 8
//SCILAB Ver: 5.5.2
//Expriment No. 1
//Objective: Solution of linear equation for under
   determined and over determined cases
// For example to enter a linear equation is given
   as
//          x + 4y = 18
//          2x +3y =16
//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 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

```
//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


clc
clear
m = input("Enter the no of row of matrix A: ")
n = input("Enter the no of column of matrix A: ")
for i=1:m
    for j=1:n
        mprintf("Enter the element no (%d,%d):",i,j)
        A(i,j)=input("")
    end
end

m = input("Enter the no of row of matrix B: ")
n = input("Enter the no of column of matrix B: ")
for i=1:m
    for j=1:n
        mprintf("Enter the element no (%d,%d):",i,j)
        B(i,j)=input("")
    end
end
C = pinv(A)*B
disp(C,"output1 (pinv)")
C1 = A\B
disp(C1,"output2 (\)")
//C2 = inv(A)*B
//disp(C2,"output3 (inv)")
```

Experiment: 2

Determination of Eigen Value & Eigen Vectors

Scilab code Solution 2.2 Experiment 2

```
//Operating system: Windows 8
//SCILAB Ver: 5.5.2
//Expriment No. 2
//Objective: Determination of Eigen value & Eigen
    Vectors
//or example to enter a matrix is given as
//A = 1 6
//    4 5
//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):6
//Enter the element no (2,1):4
//Enter the element no (2,2):5
clc
clear
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

```
n = input("Enter the no of column of matrix A: ")
for i=1:m
    for j=1:n
        mprintf("Enter the element no (%d,%d):",i,j)
        A(i,j)=input("")
    end
end
[eVal eVect]=spec(A)
disp(eVal,"Eigenvalues :")
disp(eVect,"Eigenvectors :")
```

Experiment: 3

Determination of roots of a polynomial

Scilab code Solution 3.3 Experiment 3

```
//Operating system: Windows 8
//SCILAB Ver: 5.5.2
//Expriment No. 3
//Solver required: Stixbox: Version 2.3-1
//Objective: Determination of roots of a polynomial
// For example to find the roots of a polymomials
// y = x^5 + 4x^3 - 9x +5
//
//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

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

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

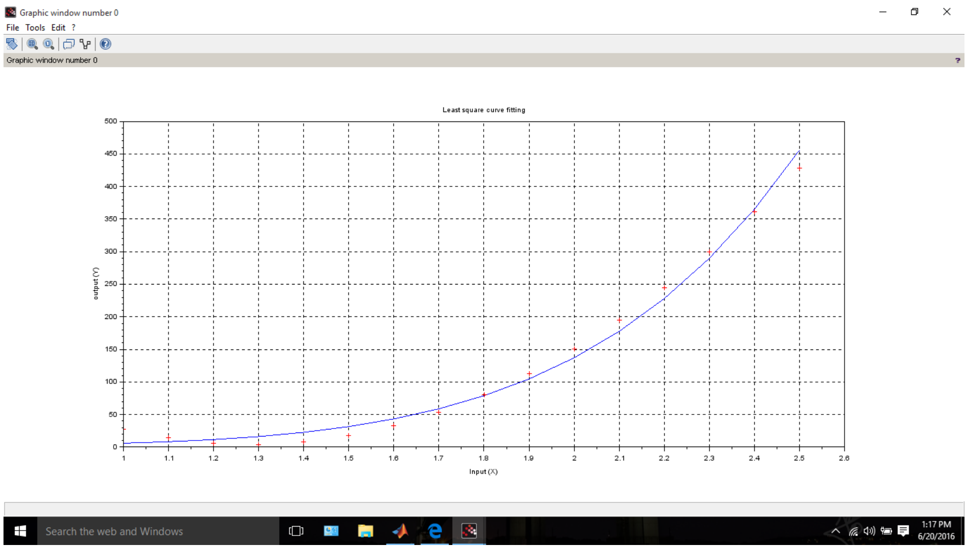

Figure 4.1: Experiment 4

```
//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
clear
clc
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
nx = input("Enter the Degree of polymomial in which
    you want to fit: ")
P = poly(A,"x","coeff")
x = 1:.1:2.5
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

```
n = polyfit(x,y1,nx)
y2 = polyval(n,x)
P2 = poly(n(nx+1:-1:1),"x","coeff")
plot(x,y1,'--',x,y2,'+r')
title('Least square curve fitting')
xlabel('Input (X)')
ylabel('Output (Y)')
set(gca(),"grid",[1 1])
disp(P,"Polynomical Enter")
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

```
//Operating system: Windows 8
////SCILAB Ver: 5.5.2
//Experiment No. 5
//Solver required: Stixbox: Version 2.3-1
//Objective: Determination of Polynomial fit ,
   Analyzing Residuals , exponential fit & error
   bounds form the given data

//census is the given input data of year and
   population for the experiment no 5
//Output provides three figure .....
```

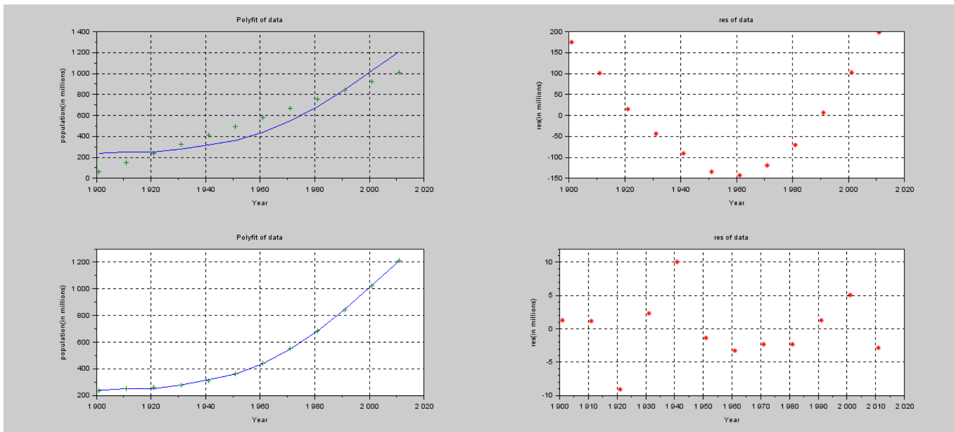

Figure 5.1: Experiment 5

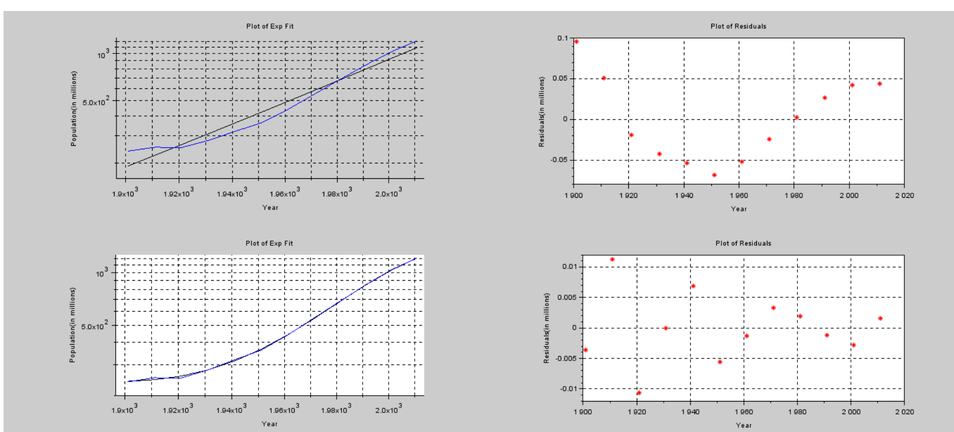

Figure 5.2: Experiment 5

```
//figure(1) for Polynomial fit and Analyzing
   Residuals
//figure (2) for exponential fit  and Analyzing
   Residuals
//and figure (3) for error bounds


clc
clear all
census=[1901 238396327;
1911    252093390;
1921    251321213;
1931    278977238;
1941    318660580;
1951    361088090;
1961    439234771;
1971    548159652;
1981    683329097;
1991    846421039;
2001    1028737436;
2011    1210854977]
year = census(:,1);
pop = census(:,2)./1000000;


//Normalisation of data
syear = (year-mean(year))./stdev(year);

//Fit the data in 1 deg polynomial
A1 = polyfit(syear,pop,1);
P=polyval(A1,syear);
figure(1)
subplot(2,2,1)
plot(year,pop,'-',year,P,'+')
title('Polyfit of data')
xlabel('Year')
ylabel('population(in millions)')
set(gca(),"grid",[1 1])
```

```
//Analyzing Residuals
res1 = pop-P;
subplot(2,2,2)
plot(year,res1,'r*')
title('res of data')
xlabel('Year')
ylabel('res(in millions)')
set(gca(),"grid",[1 1])
subplot(2,2,4)

//Fit the data in 4 deg polynomial
A2 = polyfit(syear,pop,4);
P1=polyval(A2,syear);
subplot(2,2,3)
plot(year,pop,'-',year,P1,'+')
title('Polyfit of data')
xlabel('Year')
ylabel('population(in millions)')
set(gca(),"grid",[1 1])


//Analyzing Residuals
res2 = pop-P1;
subplot(2,2,4)
plot(year,res2,'r*')
title('res of data')
xlabel('Year')
ylabel('res(in millions)')
set(gca(),"grid",[1 1]) //for grid on


//Exponential Fit 1 degree polynomial
figure(2)
subplot(2,2,1)
log1 = polyfit(syear,log10(pop),1);
logr1 = 10.^polyval(log1,syear);
plot2d1("oll",year,[logr1 pop])
```

```
title('Plot of Exp Fit')
ylabel('Population(in millions)')
xlabel('Year')
set(gca(),"grid",[1 1])

//Analyzing the residuals
logr1 = polyval(log1,syear);
res1 = log10(pop)-logr1;
subplot(2,2,2)

plot(year,res1,'r*')
title('Plot of Residuals')
ylabel('Residuals(in millions)')
xlabel('Year')
set(gca(),"grid",[1 1])

//Exponential Fit 4 degree polynomial
log1 = polyfit(syear,log10(pop),4);
logr1 = 10.^polyval(log1,syear);
subplot(2,2,3)
title('Plot of Exp Fit')
plot2d1("oll",year,[logr1 pop])
ylabel('Population(in millions)')
xlabel('Year')
set(gca(),"grid",[1 1])

//Analyzing the residuals
logr1 = polyval(log1,syear);
res1 = log10(pop)-logr1;
subplot(2,2,4)
plot(year,res1,'r*')
title('Plot of Residuals')
ylabel('Residuals(in millions)')
xlabel('Year')
set(gca(),"grid",[1 1])


//%Error Bound
```

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

Experiment: 6

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

Scilab code Solution 6.6 Experiment 6

```
//Operating system: Windows 8
////SCILAB Ver: 5.5.2
//Expriment No. 6
//Objective: Solution of differntial equation using
   4th order R-K method

// To find the solution of differntial eq  dy/dx or
   y1
// Define the equation y1 in function "dequation" (x
   -y)/(x+y)

clc
clear
function y1 = dequation(x,y)
    y1 = (x-y)/(x+y)
```

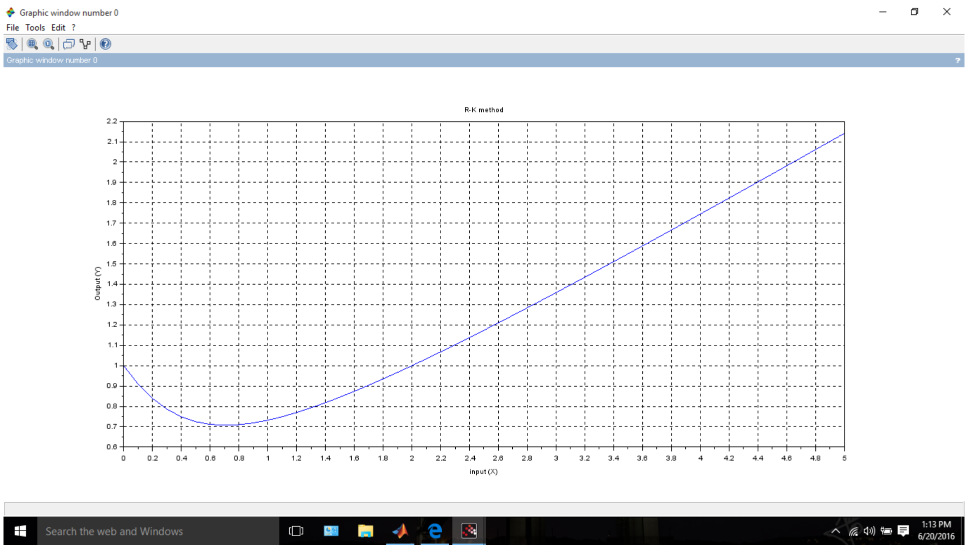

Figure 6.1: Experiment 6

```
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

```

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

Experiment: 7

Solution of differential equation using revised Euler method

Scilab code Solution 7.7 Experiment 7

```
//Operating system: Windows 8
////SCILAB Ver: 5.5.2
//Expriment No. 7
//Objective: Solution of differntial equation using
   revised Euler method

// To find the solution of differntial eq  dy/dx or
   y1
// Define the equation y1 in function "dequation" (x
   -y)/(x+y)
clc
clear
function y1 = dequation(x,y)
    y1 = (x-y)/(x+y)
endfunction

x(1)=0;
```

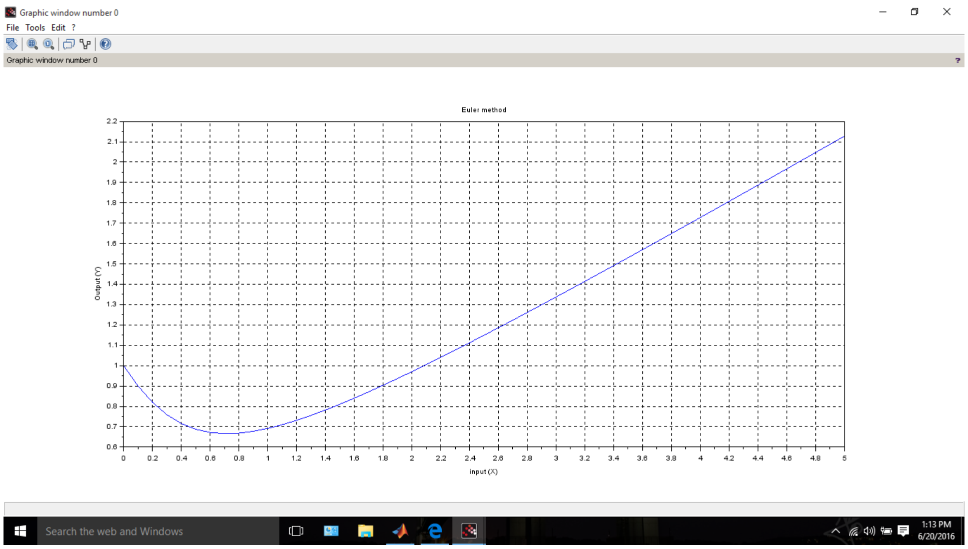

Figure 7.1: Experiment 7

```
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

```
//Operating system: Windows 8
////SCILAB Ver: 5.5.2
//Experiment No. 8
//Objective: Solution of difference equation

//In this programme differnece equation has been solved by using filter command
// x in the input vector, a and b discribed the filter and y is the filtered data
//
clear all
x = [1 2 3 4 5];
a = 1;
b =[1/4 1/4 1/4 1/4 1/4];
t =1:length(x);
y = filter(b,a,x);
plot(t,x,'-',t,y,'-')
title('Solution of difference equation')
xlabel('input (X)')
```

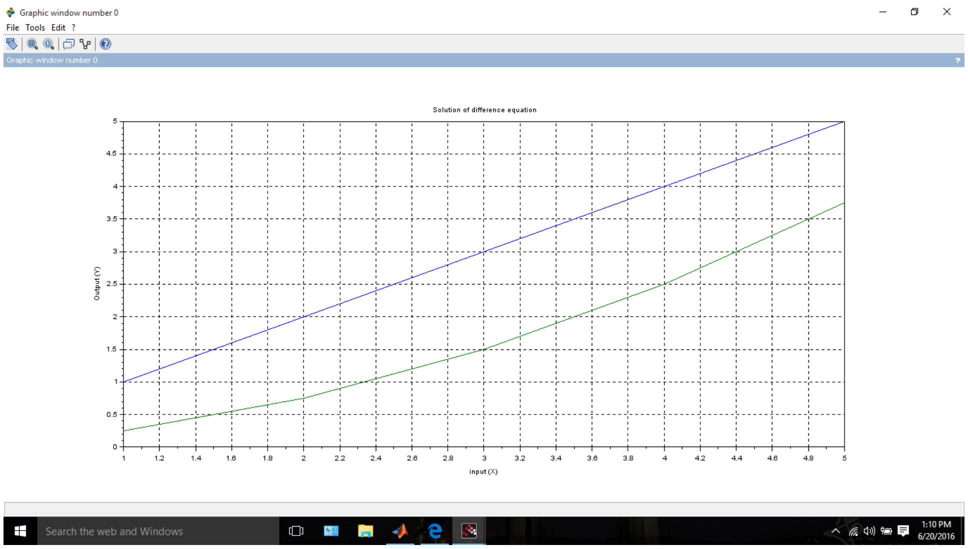

Figure 8.1: Experiment 8

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

Experiment: 9

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

Scilab code Solution 9.9 Experiment 9

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

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

Figure 9.1: Experiment 9