NUMERICAL METHODS WITH MATLAB PROGRAMMING (P)

CONTENTS

LINEAR INTERPOLATION

Question:

Find the value of y when $x = 27$ using linear interpolation using MATLAB from the following table:

x	20	30
y	6.68422	7.18192

Aim:

To write a MATLAB program for linear interpolation.

Input and Output Description:

1. **Input**

x	20	30
y	6.68422	7.18192

2. **Output**

$$y(27)$$

Procedure:

Linear interpolation formula is

$$comp\_y = y(1) + \frac{\left(given\_x\ - x(1)\right)\left(y(2) - y(1)\right)}{x(2) - x(1)}$$

Given data are,

$$x(1) = 20, x(2) = 30$$

$$y(1) = 6.68422, y(2) = 7.18192$$

$$x = 27$$

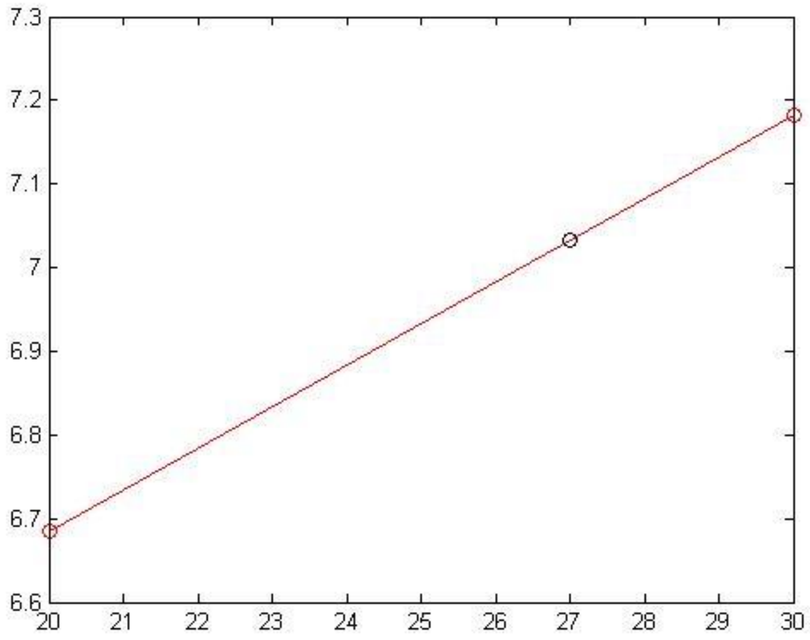
Program:

```
%Linear Interpolation
clc,clear;
%Linear interpolation
x=input('Enter a row vector x:');
y=input('Enter a row vector y:');
given_x=input('Enter the value of x to be interpolated:');
comp_y=y(1)+((given_x-x(1))/(x(2)-x(1))*(y(2)-y(1)));
fprintf('\n The value of y when x=%f is %f \n',given_x,comp_y);
%plotting the line
plot(x,y,'-or',given_x,comp_y,'ok')
```

Output:

```
Enter a row vector x:[20,30]
Enter a row vector y:[6.68422, 7.18192]
Enter the value of x to be interpolated:27

 The value of y when x=27.000000 is 7.032610
```



LINEAR REGRESSION

Question:

Write a MATLAB program to form a straight line of y on x using linear regression from the following data:

x	65	66	67	67	68	69	70	72
y	67	68	65	68	72	72	69	71

Aim:

To write a MATLAB program for implementing linear regression and to find the equation of the required regression line of y on x.

Input and Output Description:

1. **Input**

x	65	66	67	67	68	69	70	72
y	67	68	65	68	72	72	69	71

2. **Output**

The equation of regression line of y on x

Procedure:

The equation of regression line of y on x is given by

$$y - \bar{y} = r\left(\frac{\sigma_y}{\sigma_x}\right)(x - \bar{x})$$

where,

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i,$$

$$\sigma_x = std(x) = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}},$$

$$\sigma_y = std(y) = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}},$$

$$r = \frac{cov(x,y)}{\sigma_x \sigma_y}, \text{ where } cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Program:

```
%Linear Regression
%Deriving Regression line of y on x
clc,clear;
x=input('Enter a row vector x:');
y=input('Enter a row vector y:');
xbar=mean(x);
ybar=mean(y);
n=length(x);
sigma_x=std(x);
sigma_y=std(y);
cov=0;
for i=1:n
    cov=cov+(x(i)-xbar)*(y(i)-ybar);
end
cov=cov/(n-1);
r=cov/(sigma_x*sigma_y);
coef=r*(sigma_y/sigma_x);
a=coef;
b=ybar-(coef*xbar);
fprintf('\n The linear regression line of y on x is \n')
fprintf('\n y=%fx + %f\n',a,b)
```

Output:

```
Enter a row vector x:[65 66 67 67 68 69 70 72]
Enter a row vector y:[67 68 65 68 72 72 69 71]

 The linear regression line of y on x is 

 y=0.666667x + 23.666667
```

STRAIGHT LINE FITTING

Question:

Write a MATLAB program to fit a straight line from the following data and hence find $y(25)$:

x	0	5	10	15	20
y	7	11	16	20	26

Aim:

To write a MATLAB program for fitting a straight line.

Input and Output Description:

1. **Input**

x	0	5	10	15	20
y	7	11	16	20	26

2. **Output**

$$y(25)$$

Procedure:

When $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ be the set of given data.

The normal equations for fitting a straight line $y = ax + b$ are,

$$a\Sigma x + nb = \Sigma y$$

$$a\Sigma x^2 + b\Sigma x = \Sigma xy$$

Solving these equations for a and b we get the straight line.

By putting $x = 25$ in $y = ax + b$ we can find $y(25)$.

Program:

```
%Curve Fitting - Straight line fitting of the form y=ax+b
clc,clear;
x=input('Enter a row vector x:');
y=input('Enter a row vector y:');
n=length(x);
sigma_x=sum(x);
sigma_y=sum(y);
sigma_xy=sum(x.*y);
sigma_x_square=sum(x.*x);
A=[sigma_x n;sigma_x_square sigma_x];
B=[sigma_y;sigma_xy];
X=A\B;
a=X(1);
b=X(2);
fprintf('\n The straight line equation is y=%fx + %f\n\n',a,b)
x0=input('Enter a value of x to find y(x):');
y0=a*x0+b;
fprintf('\n The value of y at %f is %f\n',x0,y0)
```

Output:

```
Enter a row vector x:[0:5:20]
Enter a row vector y:[7 11 16 20 26]

 The straight line equation is y=0.940000x + 6.600000

Enter a value of x to find y(x):25

 The value of y at 25.000000 is 30.100000
```

TRAPEZOIDAL RULE

Question:

Write a MATLAB program to calculate $\int_0^1 f(x)dx$ using the following table by applying trapezoidal rule:

x	0.000	0.250	0.500	0.750	1.000
y	0.79788	0.77339	0.70413	0.60227	0.48394

Aim:

To write a MATLAB program for trapezoidal rule and evaluating the given integral.

Input and Output Description:

1. **Input**

x	0.000	0.250	0.500	0.750	1.000
y	0.79788	0.77339	0.70413	0.60227	0.48394

2. **Output**

$$\int_0^1 f(x)dx$$

Procedure:

Trapezoidal rule formula is

$$\int_a^b f(x)dx = \frac{h}{2}(A + 2B)$$

where,

$$A = \text{sum of first and last ordinates}$$

$$B = \text{sum of remaining ordinates}$$

Given data:

$$h = 0.25,$$

$$a = 0,$$

$$b = 1,$$

$$n = 4.$$

Program:

```
%Trapezoidal Rule
clc,clear;
format long
x=input('Enter a row vector x:');
y=input('Enter a row vector y:');
n=length(x);
h=(x(n)-x(1))/(n-1);
A=y(1)+y(n);
B=0;
for i=2:n-1
    B=B+y(i);
end
integral=(h/2)*(A+2*B);
fprintf('\n The integral value using trapezoidal rule is \n')
disp(integral)
```

Output:

```
Enter a row vector x:[0:0.25:1]
Enter a row vector y:[0.79788 0.77339 0.70413 0.60227 0.48394]

 The integral value using trapezoidal rule is
   0.680175000000000
```

SIMPSON'S 1/3 RULE

Question:

By dividing the range into 10 equal parts, find the approximate value of $\int_0^{\pi} \sin x \, dx$ using Simpson's $1/3$ – rule in MATLAB.

Aim:

To write a MATLAB program for Simpson's 1/3 rule and evaluating the given integral.

Input and Output Description:

1. **Input**

$$h = \frac{\pi}{10}, a = 0, \;\; b = \pi, \qquad f(x) = \sin(x)$$

2. **Output**

$$\int_0^{\pi} \sin x \, dx$$

Procedure:

Simpson's 1/3 rule formula is

$$\int_a^b f(x)dx = \frac{h}{3}(A + 4B + 2C)$$

where,

A =sum of first and last ordinates, B =sum of even ordinates & C =sum of remaining ordinates

Given data:

$$h = \frac{\pi}{10}, a = 0, \;\; b = \pi,$$

In the range $(0, \pi)$ the $(x, sinx)$ values are tabulated below

$\boldsymbol{x}$	0	$\frac{\pi}{10}$	$\frac{2\pi}{10}$	$\frac{3\pi}{10}$	$\frac{4\pi}{10}$	$\frac{5\pi}{10}$	$\frac{6\pi}{10}$	$\frac{7\pi}{10}$	$\frac{8\pi}{10}$	$\frac{9\pi}{10}$	π
$\boldsymbol{sinx}$	0	0.3090	0.5878	0.8090	0.9511	1	0.9511	0.8090	0.5878	0.3090	0

Program:

```
%Simpson's one third Rule
clc,clear;
format long
x=input('Enter a row vector x:');
y=sin(x);
n=length(x);
h=(x(n)-x(1))/(n-1);
A=y(1)+y(n);
B=0;
for i=3:2:n-1
    B=B+y(i);
end
C=0;
for i=2:2:n-1
    C=C+y(i);
end
integral=(h/3)*(A+4*B+2*C);
fprintf('\n The integral value using Simpson’s 1/3 rule is \n')
disp(integral)
```

Output:

```
Enter a row vector x:[0:pi/10:pi]


 The integral value using Simpson’s 1/3 rule is
   1.966937557703905
```

NEWTON RAPHSON METHOD

Question:

Write a MATLAB program to implement Newton Raphson method and to find a positive root of the equation $f(x) = 3x - \cos x - 1$ correct to four decimal places.

Aim:

To write a MATLAB program for implementing Newton Raphson method.

Input and Output Description:

1. **Input**

$$f(x) = 3x - \cos x - 1$$

2. **Output**

A positive root of the equation $f(x)$

Procedure:

Let $f(x)$ be the given function. i.e., $f(x) = 3x - cosx - 1$.

$$f(0) = -2\ (-ve)$$

$$f(1) = 1.459\ (+ve)$$

Let $x_0 = 0.5$ be the first approximation of the root of the successive approximation of the root is obtained from the formula

$$x_{n+1} = x_n - \left(\frac{f(x_n)}{f'(x_n)}\right), for\ n = 0,1,2, \dots$$

Here $f'(x) = 3 + sinx$.

We write three programs one for $f(x)$, one for $f'(x)$ and other for the main program (Newton Raphson Method).

Program:

f.m

```
function value=f(x)
value=3*x-cos(x)-1;
```

fdash.m

```
function value=fdash(x)
value=3+sin(x);
```

nrmethod.m

```
%Newton Raphson Method
clc,clear;
x0=0.5;
error=1;
n=0;
fprintf('\n n \t \t x \t \t \t \t f(x)\n');
fprintf(' %d \t %10.7f \t %10.7f \n',n,x0,f(x0));
while(error>0.00001)
    x1=x0-f(x0)/fdash(x0);
    fprintf(' %d \t %10.7f \t %10.7f \n',n+1,x1,f(x1));
    error=abs(x0-x1);
    x0=x1;
    n=n+1;
end
```

Output:

```
 n          x              f(x)

 0      0.5000000     -0.3775826

 1      0.6085186      0.0050602

 2      0.6071019      0.0000008

 3      0.6071016      0.0000000
```

GAUSS ELIMINATION METHOD

Question:

Write a MATLAB program for solving the following system of simultaneous equations using Gauss elimination method:

$$2x + y + 4z = 12; 8x - 3y + 2z = 20; 4x + 11y - z = 33$$

Aim:

To write a MATLAB program for solving a system of simultaneous equation by using Gauss Elimination method.

Input and Output Description:

1. **Input**

$$Ax = B$$

$$\begin{pmatrix} 2 & 1 & 4 \\ 8 & -3 & 2 \\ 4 & 11 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 12 \\ 20 \\ 33 \end{pmatrix}$$

2. **Output**

$$x, y \text{ and } z$$

Procedure:

Gauss Elimination method consists of the following two steps:

Step1: Upper Triangularization

Making elements below the main diagonal of the augmented matrix to be zero.

Step2: Backward Substitution

Obtaining values of the variables in the order $x_n, x_{n-1}, x_{n-2}, \dots, x_1$.

Given data:

$$AX = B$$

$$\begin{pmatrix} 2 & 1 & 4 \\ 8 & -3 & 2 \\ 4 & 11 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 12 \\ 20 \\ 33 \end{pmatrix}$$

Augmented Matrix is

$$\begin{pmatrix} 2 & 1 & 4 & 12 \\ 8 & -3 & 2 & 20 \\ 4 & 11 & -1 & 33 \end{pmatrix}$$

Program:

```
%Gauss Elimination Method
clc,clear
A=input('Enter the coefficient matrix: ');
if det(A)==0
    fprintf('\n Determinant cannot be zero\n');
end
if det(A)~=0
    B=input('Enter the constant matrix:');
    A(:,4)=B;
    [m,n]=size(A);
    for i=1:m-1
        for j=2:m
            if A(i,i)==0
                temp=A(i,:);
                A(i,:)=A(j,:);
                A(j,:)=temp;
            end
        end
    end
    for j=1:m-1
        for i=j+1:m
            A(i,:)=A(i,:)-(A(i,j)/A(j,j))*A(j,:);
        end
    end
    X=zeros(m,1);
    for k=m:-1:1
        c=0;
        for l=2:m
            c=c+A(k,l)*X(l);
        end
        X(k)=(A(k,n)-c)/A(k,k);
```

```
    end
    fprintf('\n Solution by Gauss elimination method: ');
    for i =1:m
        fprintf('\n x(%d) = %f \n',i,X(i));
    end
end
```

Output:

```
Enter the coefficient matrix: [2 1 4; 8 -3 2; 4 11 -1]
Enter the constant matrix:[12; 20; 33]

 Solution by Gauss elimination method: 

 x(1) = 3.000000 

 x(2) = 2.000000 

 x(3) = 1.000000 
```

GAUSS SEIDEL ITERATIVE METHOD

Question:

Write a MATLAB program to implement Gauss-Seidel iterative method for solving n equations in n variables.

Sample data: $27x + 6y - z = 85; 6x + 15y + 2z = 72; x + y + 54z = 110.$

Aim:

To write a MATLAB program for implement Gauss Seidel iterative method.

Input and Output Description:

1. **Input**

$$Ax = B$$

$$\begin{pmatrix} 27 & 6 & -1 \\ 6 & 15 & 2 \\ 1 & 2 & 54 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 85 \\ 72 \\ 110 \end{pmatrix}$$

2. **Output**

x, y and z

Procedure:

Let the given system of equations be

$$a_1 x + b_1 y + c_1 z = d_1$$

$$a_2 x + b_2 y + c_2 z = d_2$$

$$a_3 x + b_3 y + c_3 z = d_3$$

First, we rearrange the equation for pivoting. Then we compute x, y, z values respectively from the first, second, third equation as given below.Let the initial values are

$$x^{(0)} = 0, y^{(0)} = 0, z^{(0)} = 0$$

Then

$$x^{(1)} = \frac{1}{a_1}\left(d_1 - b_1 y^{(0)} - c_1 z^{(0)}\right)$$

$$y^{(1)} = \frac{1}{b_2}\left(d_2 - a_2 x^{(1)} - c_2 z^{(0)}\right)$$

$$z^{(1)} = \frac{1}{c_3}\left(d_3 - a_3 x^{(1)} - b_3 y^{(1)}\right)$$

The successive values are calculated similarly.

Program:

```
%Gauss Seidal Iterative Method
clc,clear;
format long;
A=input('Enter the Matrix A:');
B=input('Enter the Matrix B:');
n=input('Enter the number of iteration:');
A(:,4)=B;
fprintf('\n The Augmented Matrix is: \n');
disp(A);
x(1)=0;
y(1)=0;
z(1)=0;
for i=1:n
    x(i+1)=(A(1,4)-A(1,2)*y(i)-A(1,3)*z(i))/A(1,1);
    y(i+1)=(A(2,4)-A(2,1)*x(i+1)-A(2,3)*z(i))/A(2,2);
    z(i+1)=(A(3,4)-A(3,1)*x(i+1)-A(3,2)*y(i+1))/A(3,3);
end
for k=1:n
    fprintf('\n Iteration: %d \n x= %f, \n y= %f and \n z= %f
\n',k,x(k+1),y(k+1),z(k+1))
end
```

Input:

```
Enter the Matrix A:[27 6 -1;6 15 2; 1 1 54]
Enter the Matrix B:[85; 72; 110]
Enter the number of iteration:6
```

Output:

```
 The Augmented Matrix is:
    27     6    -1    85
     6    15     2    72
     1     1    54   110
```

```
Iteration: 1
x= 3.148148,
y= 3.540741 and
z= 1.913169

Iteration: 2
x= 2.432175,
y= 3.572041 and
z= 1.925848

Iteration: 3
x= 2.425689,
y= 3.572945 and
z= 1.925951

Iteration: 4
x= 2.425492,
y= 3.573010 and
z= 1.925954

Iteration: 5
x= 2.425478,
y= 3.573015 and
z= 1.925954

Iteration: 6
x= 2.425476,
y= 3.573016 and
z= 1.925954
```

RUNGE KUTTA FOURTH ORDER METHOD

Question:

Write a MATLAB program to implement 4^{th} order Runge-Kutta method and find an approximate value of y when $x = 0.2$ given that $y' = x + y; y(0) = 1$.

Aim:

To write a MATLAB program for implementing Runge-Kutta fourth order method.

Input and Output Description:

1. **Input**

$$y' = f(x, y) = x + y;$$
$$y(0) = 1, \text{ i.e., } x_0 = -0,\ y_0 = 1$$
$$x_n = 0.2$$
$$h = 0.1$$

2. **Output**

$$y(0.2)$$

Procedure:

Given a differential equation of the form

$$y' = f(x, y), y(x_0) = y_0$$

The RungeKutta fourth order formula for finding a solution is given below,

$$k_1 = hf(x_0, y_0)$$
$$k_2 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_1}{2}\right)$$
$$k_3 = hf\left(x_0 + \frac{h}{2}, y_0 + \frac{k_2}{2}\right)$$
$$k_4 = hf(x_0 + h, y_0 + k_3)$$
$$\Delta y = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
$$x_1 = x_0 + h, y_1 = y(x) = y_0 + \Delta y$$

We write two programs, one for $f(x, y)$ and the other for finding $y(0.2)$ using Runge-Kutta fourth order method.

Program:

f.m

```
function value=f(x,y)

value=x+y;
```

rkmethod.m

```
%RungeKutta fourth order method

clc,clear;

x0=input('Enter the value of x0: ');

y0=input('Enter the value of y0: ');

xn=input(' Enter the value of xn: ');

h=input(Enter the value of step size: ');

fprintf('\n It.No \t\t X \t\t Y \n');

n=1;

while(x0<xn)

    k1=h*f(x0,y0);

    k2=h*f(x0+h/2,y0+k1/2);

    k3=h*f(x0+h/2,y0+k2/2);

    k4=h*f(x0+h,y0+k3);

    delta_y=(k1+2*k2+2*k3+k4)/6;

    x0=x0+h;

    y0=y0+delta_y;

    fprintf('\n%d \t %f \t %f\n',n,x0,y0);

    n=n+1;

end
```

Input:

```
Enter the value of x0: 0
Enter the value of y0: 1
Enter the value of step size: 0.1
Enter the value of xn: 0.2
```

Output:

```
     X           Y

  0.100000    1.110342

  0.200000    1.242805
```

LAGRANGE'S INTERPOLATION METHOD

Question:

Write a MATLAB program to implement Lagrange's interpolation formula and compute $f(27)$ from the following table:

x	14	17	31	35
y	68.7	64.0	44.0	39.1

Aim:

To write a MATLAB program for implementing Lagrange's interpolation method.

Input and Output Description:

1. **Input**

x	14	17	31	35
y	68.7	64.0	44.0	39.1

2. **Output**

$$f(27)$$

Procedure:

Given a set of $n+1$ tabulated values $(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. The Lagrange's interpolation formula to find $y = f(x)$ at a given x is given below,

$$y = f(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)\dots(x_0-x_n)}y_0 + \frac{(x-x_0)(x-x_2)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)\dots(x_1-x_n)}y_1 + \cdots + \frac{(x-x_0)(x-x_1)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)\dots(x_n-x_{n-1})}y_n$$

Program:

```
%Lagrange Interpolation
clc,clear;
format long
X=input('Enter the array of x: ');
Y=input('Enter the array of y: ');
given_x=input('Enter the value of x to be interpolated: ');
n=length(X);
sum=0;
for i=1:n
    factor(i)=1;
    for j=1:n;
        if j~=i
            factor(i)=factor(i)*(given_x-X(j))/(X(i)-X(j));
        end
    end
    factor(i)=factor(i)*Y(i);
    sum=sum+factor(i);
end
fprintf('\n The value of y when x=%f is %f\n',given_x,sum);
```

Input:

```
Enter the array of x: [14 17 31 35]
Enter the array of y=f(x): [68.7 64 44 39.1]
Enter the value of x to be interpolated: 27
```

Output:

```
 The value of f(27) is
  49.310457516339852
```