

# Chapter Ten

## Numerical Methods

### 10.1 Introduction

Numerical methods are used to solve mathematical problems that are difficult to solve by analytical methods. These methods are usually applied via computer programming because they involve large number of data and large number of arithmetic processes that usually replaced the few and difficult analytical methods. MATLAB contain large collection of functions and commands that facilitate the use of numerical methods.

### 10.2 Interpolation

Interpolation is the extraction of missing data from given available data. It is used with limited tabulated data so the table rows are increased by adding other rows between the existing rows. There are two types of interpolation in MATLAB, one and two dimensional. One dimensional interpolation generates data between elements of one dimensional matrix of a dependent variable (y) as they change with other conjugated matrix of independent variable (x) which is the interpolation on a two dimensional curve. Two dimensional interpolation works with a dependant variable (z) that changes with two independent variables (x) and (y) which is the interpolation over a three dimensional surface.

#### **10.2.1 One dimensional interpolation**

It is performed in MATLAB by the command (interp1) as follows:

$$y_{\text{new}} = \text{interp1}(x_{\text{old}}, y_{\text{old}}, x_{\text{new}}) \quad 10.1$$

$$y_{\text{new}} = \text{interp1}(x_{\text{old}}, y_{\text{old}}, x_{\text{new}}, 'spline') \quad 10.2$$

Where:

$x_{\text{old}}$  is the matrix of the existing (x) elements.

$y_{\text{old}}$  is the matrix of the existing (y) elements.

$x_{\text{new}}$  is the matrix of new (x) elements.

$y_{\text{new}}$  is the matrix of the new generated elements based on the old elements  $y_{\text{old}}$ .

Equation (10.1) performs linear interpolation between data while equation (10.2) performs spline interpolation where a third order degree polynomial is used to interpolate between data.

**Ex. 10.1** Write MATLAB program to expand the following table so that the values of (x) change at a step of one unit. Apply linear and spline interpolation:

x	0	2	4	6
y	0	10	16	20

**Sol.**

```
clear,clc
x=0:2:6;
y=[0 10 16 20];
xn=0:6;
ynl=interp1(x,y,xn);
y ns=interp1(x,y,xn,'spline');
R=[xn;ynl;y ns];
fprintf(' %2g %3g %6.3f \n',R)
```

x <sub>new</sub>	y <sub>new,linear</sub>	y <sub>new,spline</sub>
0	0	0.000
1	5	5.625
2	10	10.000
3	13	13.375
4	16	16.000
5	18	18.125
6	20	20.000

### 10.2.2 Two dimensional interpolation

It is performed in MATLAB by the command (interp2) as follows:

$$z_{new} = \text{interp2}(x_{old}, y_{old}, z_{old}, x_{new}, y_{new}) \quad 10.3$$

$$z_{new} = \text{interp2}(x_{old}, y_{old}, z_{old}, x_{new}, y_{new}, 'spline') \quad 10.4$$

The dependent variable (z) varies with two independent variables (x) and (y). Equation 10.3 uses linear interpolation while equation 10.4 uses spline interpolation on the third order. An example of the function (z) is the topology of a three dimensional surface.

**Note:** To convert the matrices (x) and (y) to a two dimensional coordinates for the dependent variable (z) the command (meshgrid) is used as follows:

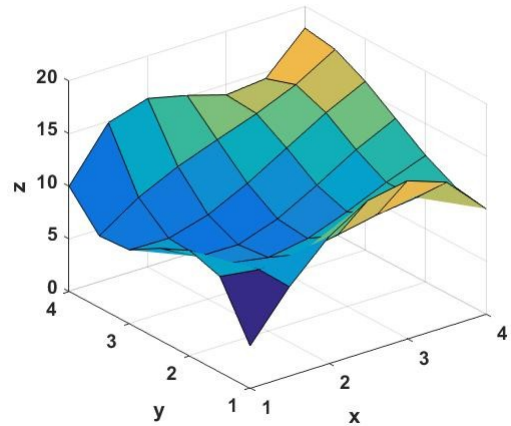
$$[x,y]=\text{meshgrid}(\text{elements of } (x), \text{elements of } (y)) \quad 10.5$$

**Note:** The command (surf) is used to draw the three dimensional surface:

$$\text{surf}(x,y,z) \quad 10.6$$

**Ex. 10.2** Write MATLAB program to expand the data in the following table so that a smooth surface of the variable (z) can be drawn over the two dimensional coordinates of (x) and (y). Expand the table to a mesh of 7×7 nodes first then expand it to 70×70 nodes and draw the expanded two surface using spline interpolation. The values of x and y in the table both change from 1 to 4 (shaded regions). The other numbers represents the values of z.

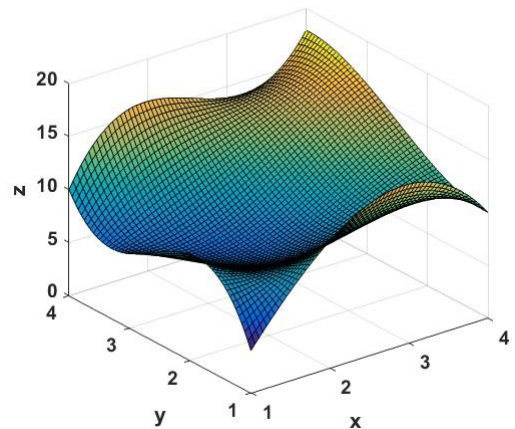
		X			
		1	2	3	4
Y	1	4	12	15	10
	2	10	7	9	12
	3	7	8	12	16
	4	10	16	14	18



The mesh of 7×7 nodes

**Sol.**

```
clear,clc
i=0;j=0;
x=1:4;
y=1:4;
z=[4 12 15 10;10 7 9 12;...
   7 8 12 16;10 16 14 18];
for xn=1:0.5:4;
    i=i+1;
    for yn=1:0.5:4;
        j=j+1;
        zn(j,i)=interp2(x,y,z,xn,yn,'spline');
    end
    j=0;
end
[xp,yp]=meshgrid(1:0.5:4,1:0.5:4);
surf(xp,yp,zn)
grid on
xlabel('x');ylabel('y');zlabel('z')
```



The mesh of 70×70 nodes

### 10.3 Polynomial data fitting

The tabulated data can be represented in terms of polynomial equations that best fit the data. The polynomial takes the form:

$$y = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n \quad 10.7$$

Where the values of the coefficients  $a_s$  depends on the given data.

MATLAB performs polynomial fitting through the command (polyfit) as follows:

$$z = \text{polyfit} ( x , y , \text{order} ) \quad 10.8$$

where z represents the matrix of the coefficients  $a_s$  starting from the coefficient of the highest order downward. The number of the matrix z is therefore equals the order of the polynomial plus one. The one dimensional matrices x and y are the independent and dependent variables respectively and (order) is a number equals to the polynomial order. To facilitate the process of writing the polynomial equation based on its coefficients  $z_s$  the command (polyval) can be used to automatically give the values of the dependent variable (y) as evaluated from the polynomial equation containing the coefficients  $z_s$  as follows:

$$y_{\text{fit}} = \text{polyval} ( z , x ) \quad 10.9$$

Where (z) is the output matrix of the command (polyfit) and (x) is the independent variable.

**Ex. 10.3** Write MATLAB program to evaluate the velocity of a car after 3 seconds from its move according to the following table. Use linear and polynomial interpolation and compare the results with third order fitting equation. Write down the third order equation and compare the result with that generated by the command (polyval).

Time (s)	0	2	4	6	8
Velocity (m/s)	0	5	15	40	100

**Sol.**

```
clear,clc
xn=input('Enter the time = ');
x=[0 , 2 , 4 , 6 , 8];
y=[0 , 5 , 15 , 40 , 100];
yn_linear=interp1(x,y,xn);
yn_spline=interp1(x,y,xn,'spline');
zs=polyfit(x,y,3);
y_fit_1=zs(1)*xn^3+zs(2)*xn^2+zs(3)*xn+zs(4);
y_fit_2=polyval(zs,xn);
disp('    x_new    y_linear    y_spline    y_fit_1    y_fit_2 ')
disp('-----')
disp([xn,yn_linear,yn_spline,y_fit_1,y_fit_2])
```

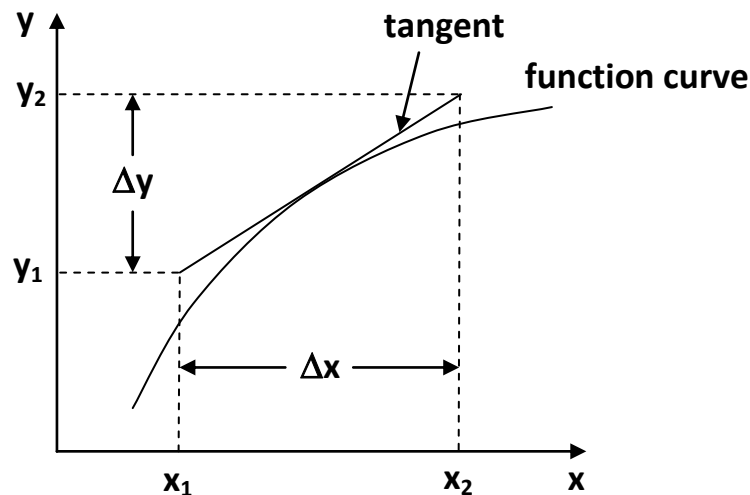
Run:  
Enter the time = 5

x_new	y_linear	y_spline	y_fit_1	y_fit_2
5.0000	27.5000	24.5313	24.1875	24.1875

## 10.4 Numerical differentiation

The derivative of function is the rate of change of that function. So, it represents the rate at which the dependent variable ( $y$ ) changes relative to the independent variable ( $x$ ). This definition can be approximated as a finite difference of ( $y$ ) or ( $\Delta y$ ) divided by the corresponding change in ( $x$ ) or ( $\Delta x$ ):

$$\frac{dy}{dx} \cong \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \quad 10.10$$



The differences ( $\Delta x$ ) and ( $\Delta y$ ) can be evaluated in MATLAB using the command (`diff`) which calculates the differences between one dimensional matrix elements.

**Ex. 10.4** Write MATLAB program to evaluate the derivative of the function ( $x^3$ ) with respect to ( $y$ ) numerically. Take the values of ( $x$ ) to change from 0 to 10 at a step of one unit. Compare the numerical results with the analytical derivative of the function ( $x^3$ ).

<b>Sol.</b>	0.5000	1.0000	0.7500
<code>clear,clc</code>	1.5000	7.0000	6.7500
<code>x=0:10;</code>	2.5000	19.0000	18.7500
<code>y=x.^3;</code>	3.5000	37.0000	36.7500
<code>dx=diff(x);</code>	4.5000	61.0000	60.7500
<code>dy=diff(y);</code>	5.5000	91.0000	90.7500
<code>der=dy./dx;</code>	6.5000	127.0000	126.7500
<code>x1=0.5:9.5;</code>	7.5000	169.0000	168.7500
<code>der2=3*x1.^2;</code>	8.5000	217.0000	216.7500
<code>disp([x1',der',der2'])</code>	9.5000	271.0000	270.7500

**Ex. 10.5** Repeat the previous example using the following function:

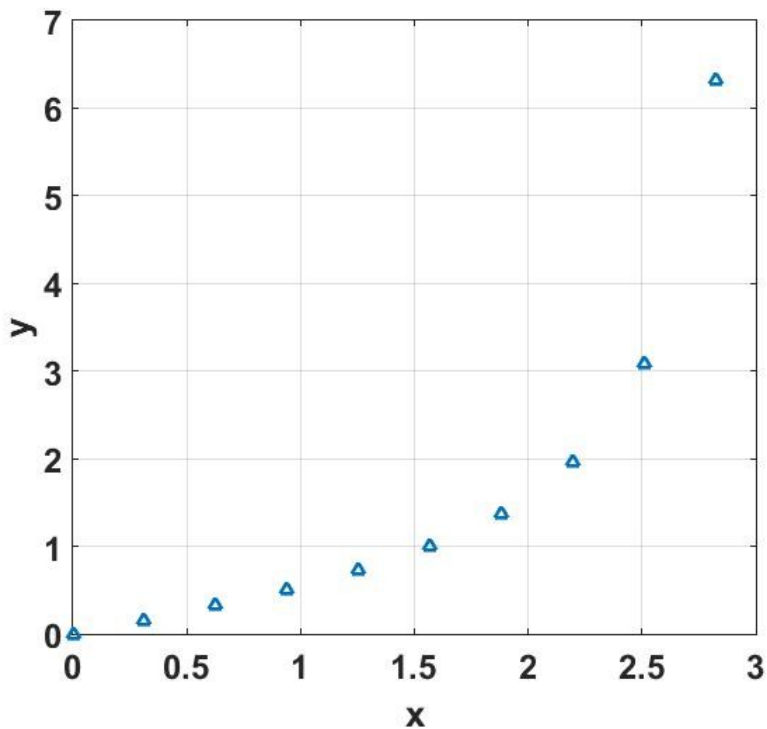
$$y = \frac{\sin(x)}{1 + \cos(x)}$$

The domain of the (x) values is between 0 to  $\pi$  at a step of  $\pi/10$ .

**Sol.**

```
clear,clc
x=0:pi/10:pi;
y=sin(x)/(1+cos(x));
dx=diff(x);
dy=diff(y);
der=dy./dx;
x1=pi/20:pi/10:pi-pi/20;
der2=1./(1+cos(x1));
disp([x1',der',der2'])
plot(x,y,'^','linewidth',2)
grid on;xlabel('x');ylabel('y')
```

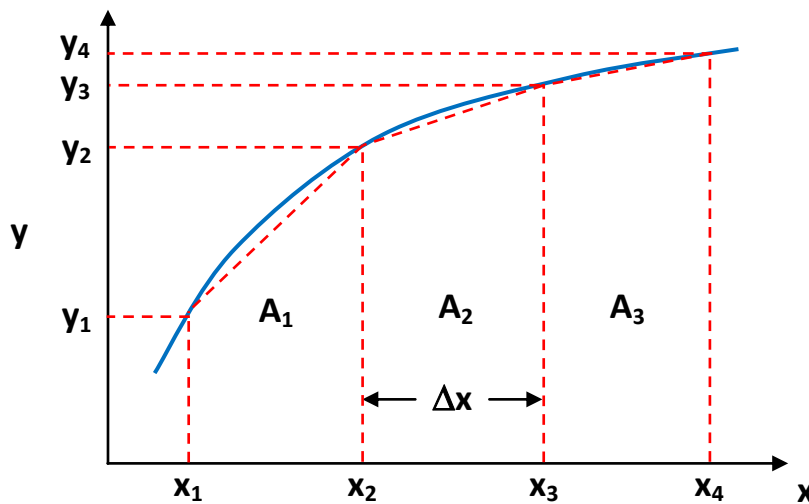
0.1571	0.5042	0.5031
0.4712	0.5301	0.5288
0.7854	0.5876	0.5858
1.0996	0.6908	0.6878
1.4137	0.8704	0.8647
1.7279	1.1981	1.1854
2.0420	1.8660	1.8315
2.3562	3.5494	3.4142
2.6704	10.3007	9.1749
2.9845	Inf	81.2238



## 10.5 Numerical integration

It is the numerical procedure to find the definite integral or the value of the integration of a function between two limits in the domain ( $x$ ). This value equals the area between the function curve and the  $x$  axis and bounded from right and left by two vertical lines extending from ( $x$ ) limits ( $x_1$ ) and ( $x_2$ ). The domain of integration is divided into small stripes and their areas are summed to find the value of integration. This method is called (**Trapezoidal Rule**). It is performed as follows:

$$A = \sum_{n=1}^N A_n = A_1 + A_2 + \dots + A_N = \Delta x \left[ \frac{y_1}{2} + y_2 + y_3 + \dots + y_{N-1} + \frac{y_N}{2} \right] \quad 10.11$$



Note: The integral of a function can also be evaluated in MATLAB via the command (integral) which incorporates (**Quadrature Method**). It takes the following form:

$$A = \text{integral} (\text{Function} , \text{min} , \text{max} )$$

Where:  $A$  : the value of integral or area under the curve

Function: is the function handle

min: is the lower limit of the integral (minimum value of ( $x$ ))

max: is the upper limit of the integral (maximum value of ( $x$ ))

**Ex. 10.6** Write MATLAB program to evaluate the following integral using two methods: the trapezoidal rule and the quadrature method:

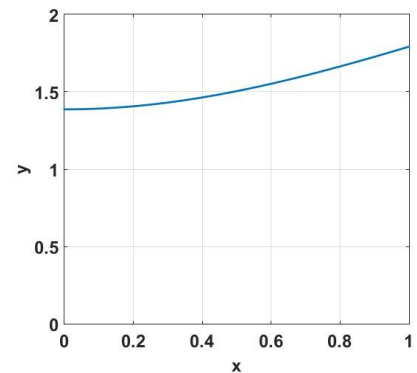
$$A = \int_0^1 \ln(2x^2 + 4) dx$$

**Sol.**

```
clear,clc
min1=0;max1=1;
step1=.01;
x=min1:step1:max1;
y=log(2*x.^2+4);
A1=sum(y)*step1;
f1=@(x)log(2*x.^2+4);
A2=integral(f1,min1,max1);
A3=log(6)-2+2*sqrt(2)*atan(1/sqrt(2));
disp([A1,A2,A3])
plot(x,y);grid on
xlabel('x');ylabel('y')
```

**run:**

1.5485      1.5326      1.5326



**Ex. 10.7** Repeat the previous example using the following integral:

$$A = \int_{\pi/6}^{\pi/2} \frac{\cot(x)}{1 + \sin^2(x)} dx$$

**Sol.**

```
clear,clc
min2=pi/6;max2=pi/2;
step2=pi/100;
x1=min2:step2:max2;
y1=cot(x1)./(1+(sin(x1)).^2);
A11=sum(y1)*step2;
f2=@(x1)cot(x1)./(1+(sin(x1)).^2);
A22=integral(f2,min2,max2);
A33=.5*log(5/2);
disp([A11,A22,A33])
plot(x1,y1);grid on
xlabel('x1');ylabel('y1')
```

**Run:**

0.4803      0.4581      0.4581

