For Computer Science Stream

MAC11

RAMAIAH
Institute of Technology

MATLAB

Manual for

CSE, ISE, CS(CS), CS(AIML),

AI&ML, AI&DS, BT

Department of Mathematics

Ramaiah Institute of Technology,

Bengaluru - 560054

Introduction to MATLAB

When you start MATLAB, the desktop appears in its default layout.

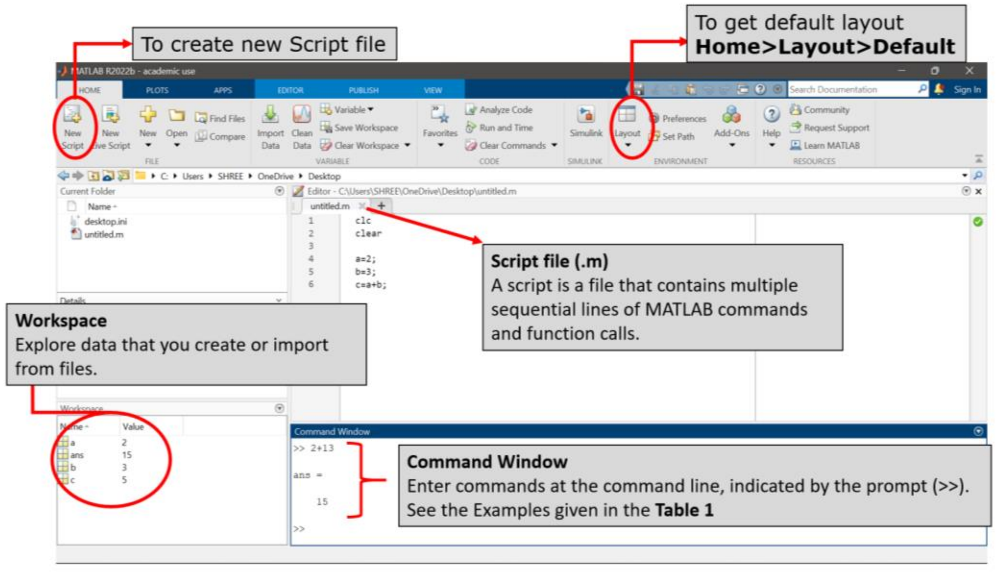


Table 1: Basic Arithmetic Operations

Sl. No.	Input	Output	Sl. No.	Input	Output
1	2+3	5	5	x^2	25
2	x=5; y=3; x+y	8	6	sqrt(y)	1.7321
3	x*y	15	7	x/0	Inf
4	x/y	1.6667	8	(x+y)/(x^2+y^2)	0.2353

Table 2: Basic Functions

Sl. No.	Input	Output	Sl. No.	Input	Output
1	x=2; y=pi;		8	nthroot(-4,3)	-1.5874
2	sin(y/4) asin(0.7071) **asin(x)=$sin^{-1}x$*	0.7071 0.7854 *Output will be displayed in terms of radians*	9	nthroot(-4,2) ***Syntax*** *nthroot(X,N)*	**Error:** Error using nthroot If X is negative, N must be an odd integer.
3	exp(x)	7.3891	10	z=sqrt(-4)	0.0000 + 2.0000i
4	log(x)	0.6931	11	imag(z)	2
5	z=log10(x)	z=0.3010	12	real(z)	0
6	format long z	z = 0.301029995663981	13	abs(z)	2
7	format short cosh(y)	11.5920	14	abs(-4)	4

Note: If you end a statement with a semicolon, MATLAB performs the computation, but suppresses the display of output in the Command Window.

Table 3: Basic Commands

<table>
<tr><th>Sl. No.</th><th>Command</th><th>Output</th></tr>
<tr><td>1</td><td>clc</td><td>Clears all the text from the Command Window, resulting in a clear screen. (Does not delete the variables created)</td></tr>
<tr><td>2</td><td>clear</td><td>Removes all variables from the current workspace, releasing them from system memory. (Does not clear the screen)</td></tr>
<tr><td>3</td><td>close all</td><td>Closes all figures whose handles are visible</td></tr>
<tr><td>4</td><td>syms</td><td>creates symbolic scalar variables
Example
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>sin(x)</td><td>Error: Unrecognized function or variable 'x'.</td></tr>
<tr><td>syms x y
sin(x)+exp(y)</td><td>sin(x)+exp(y)</td></tr>
</table></td></tr>
<tr><td>5</td><td>vpa</td><td>Variable-precision arithmetic – evaluate each element of the symbolic input x to at least d significant digits
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>a=((1 + sqrt(sym(5)))/2)</td><td>a=5^(1/2)/2 + 1/2</td></tr>
<tr><td>vpa(a)</td><td>1.6180339887498948482045868343656</td></tr>
<tr><td>vpa(a,3)</td><td>1.62</td></tr>
</table></td></tr>
<tr><td>6</td><td>simplify</td><td>Performs algebraic simplification
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>syms x
sin(x)^2+cos(x)^2</td><td>cos(x)^2 + sin(x)^2</td></tr>
<tr><td>simplify(cos(x)^2 + sin(x)^2)</td><td>1</td></tr>
</table></td></tr>
<tr><td>7</td><td>pretty</td><td>Prints expression in a plain-text format that resembles typeset mathematics.
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>syms x y
z=(y^2 + x^2)/(x*y)</td><td>z=(y^2 + x^2)/(x*y)</td></tr>
<tr><td>pretty(z)</td><td><pre>
  2   2
 y + x
----
  x y</pre></td></tr>
</table>
*For true typeset rendering, use Live Scripts instead</td></tr>
<tr><td>8</td><td>subs</td><td>subs(s, old, new) returns a copy of s, replacing all occurrences of old with new, and then evaluates s.
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>syms x y
subs(x + y,x,4)</td><td>4+y</td></tr>
<tr><td>f(x,y) = x + y;
f = subs(f,[x,y],[a,b])</td><td>b + a</td></tr>
</table></td></tr>
</table>

<table>
<tr><td>9</td><td>input</td><td>input(prompt) displays the text in prompt and waits for the user to input a value and press the Return key.
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>x=input("Enter the value of x:")</td><td>Enter the value of x: _</td></tr>
</table></td></tr>
<tr><td>10</td><td>fprintf</td><td>Formats data and displays the results on the screen
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>a=1;
b=1.45;
f=x+y;
fprintf("a=%d, b=%f, f=%s", a,b,f)</td><td>a=1, b=1.450000, f=x + y</td></tr>
</table>
To format numeric and character data use the following
%d – Integer %f – Floating-point %s – String
To format the display use the following
\t – tab space \n – next line</td></tr>
<tr><td>11</td><td>disp</td><td>Disp(x) displays the value of variable x without printing the variable name
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>a=1
disp(a)
disp("a is equal to " + a)</td><td>a=1
1
a is equal to 1</td></tr>
</table></td></tr>
<tr><td>12</td><td>diff(f,var,n)</td><td>Differentiate the given function n times with respect to variable var
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>syms f(x)
f(x)=sin(x);
diff(f)</td><td>cos(x)</td></tr>
<tr><td>syms f(x)
f(x)=sin(x);
diff(f,x)</td><td>cos(x)</td></tr>
<tr><td>syms f(x)
f(x)=sin(x);
diff(f,x,2)</td><td>-sin(x)</td></tr>
</table></td></tr>
<tr><td>13</td><td>int(f,var,a,b)</td><td>Evaluate the definite integral of the function f from a to b
<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>syms f(x)
f(x)=sin(x);
int(f)</td><td>-cos(x)</td></tr>
<tr><td>syms f(x)
f(x)=csc(x);
int(f,x)</td><td>log(tan(x/2))

*csc(x)=$\operatorname{cosec} x$</td></tr>
<tr><td>syms f(x)
f(x)=sin(x);
int(f,x,0,pi/2)</td><td>1</td></tr>
</table></td></tr>
</table>

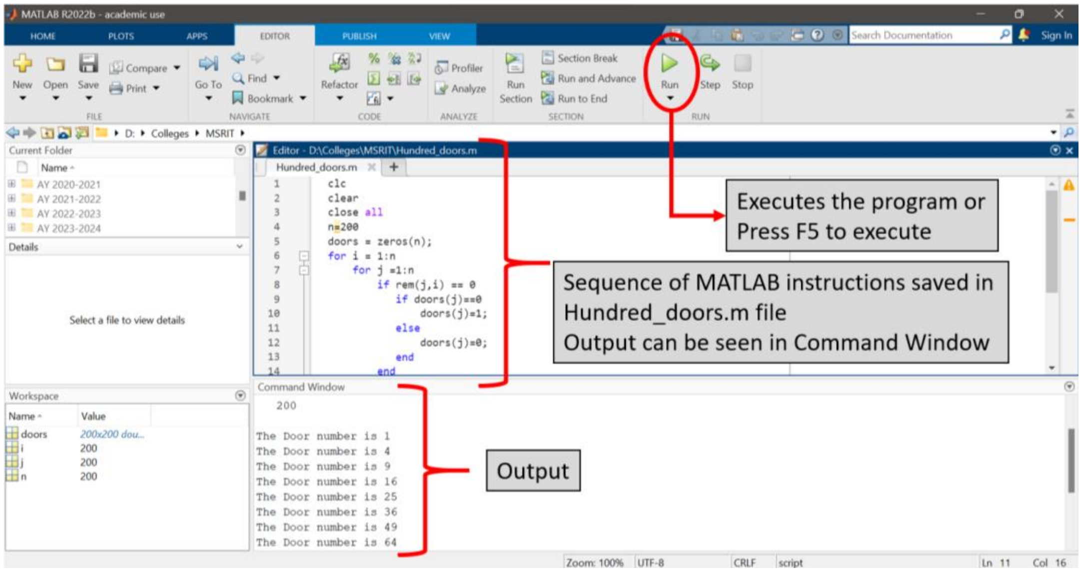
Table 4: Vectors and Matrices

<table>
<tr><th>Sl. No.</th><th>Input</th><th colspan="4">Output</th></tr>
<tr><td>1</td><td>x=[1,2,3,4,5,6]</td><td colspan="4">x = 1 2 3 4 5 6</td></tr>
<tr><td>2</td><td>x=1:1:6</td><td colspan="4">x = 1 2 3 4 5 6</td></tr>
<tr><td>3</td><td>x=1:6</td><td colspan="4">x = 1 2 3 4 5 6</td></tr>
<tr><td>4</td><td>x=linspace(1,6,5)</td><td colspan="4">x = 1.0000 2.2500 3.5000 4.7500 6.0000</td></tr>
<tr><td>5</td><td>y=x+1</td><td colspan="4">y = 2.0000 3.2500 4.5000 5.7500 7.0000</td></tr>
<tr><td>6</td><td>y=x*2</td><td colspan="4">y = 2.0000 4.5000 7.0000 9.5000 12.0000</td></tr>
<tr><td>7</td><td>a=[1,2,3,2,1]

z=a*x</td><td colspan="4">a = 2 2 2 2 2
Error: Incorrect dimensions for matrix multiplication. Check that the number of columns in the first matrix matches the number of rows in the second matrix. To operate on each element of the matrix individually, use TIMES (.*) for elementwise multiplication.
a and x are 1x5 matrices hence multiplication is not possible</td></tr>
<tr><td>8</td><td>z=a.*x</td><td colspan="4">z = 1.0000 4.5000 10.5000 9.5000 6.0000
*(**.*** represents elementwise multiplication, i.e., it will take*
*z = 1*1 2*2.25 3*3.5 2*4.75 1*6)*</td></tr>
<tr><td>9</td><td>y=x/2</td><td colspan="4">y = 0.5000 1.1250 1.7500 2.3750 3.0000</td></tr>
<tr><td>10</td><td>y=2/x</td><td colspan="4">**Error:** Matrix dimensions must agree.
(The number 2 will be interpreted as a 1x1 matrix, making regular division unfeasible. Use element-wise division instead.)</td></tr>
<tr><td>11</td><td>y=2./x</td><td colspan="4">y = 2.0000 0.8889 0.5714 0.4211 0.3333</td></tr>
<tr><td>12</td><td>y=a./x</td><td colspan="4">y = 1.0000 0.8889 0.8571 0.4211 0.1667</td></tr>
<tr><td>13</td><td>length(x)</td><td colspan="4">6</td></tr>
<tr><td>14</td><td>M=[1 2 3;3 4 5;6 7 8]</td><td>1 2 3
3 4 5
6 7 8</td><td>15</td><td>N=[1 3 5;-1 4 6;-3 4 -2]</td><td>1 3 5
-1 4 6
-3 4 -2</td></tr>
<tr><td>16</td><td>M+N</td><td>2 5 8
2 8 11
3 11 6</td><td>17</td><td>M*N</td><td>-10 23 11
-16 45 29
-25 78 56</td></tr>
<tr><td>18</td><td>b=ones(1,3)</td><td>1 1 1</td><td>19</td><td>x=zeros(3,4)</td><td>0 0 0 0
0 0 0 0
0 0 0 0</td></tr>
<tr><td>20</td><td>x*b</td><td colspan="4">Incorrect dimensions for matrix multiplication.
x is a 3x4 matrix and b is a 1x3 matrix hence multiplication is not possible</td></tr>
<tr><td>21</td><td>b*x</td><td>0 0 0 0</td><td>22</td><td>2*b.*N</td><td>2 6 10
-2 8 12
-6 8 -4</td></tr>
<tr><td>22</td><td>size(x)</td><td>3 4</td><td>23</td><td>det(N)</td><td>-52</td></tr>
<tr><td>24</td><td>inv(N)</td><td colspan="4">0.6154 -0.5000 0.0385
0.3846 -0.2500 0.2115
-0.1538 0.2500 -0.1346</td></tr>
</table>

Working with Script file (.m file)

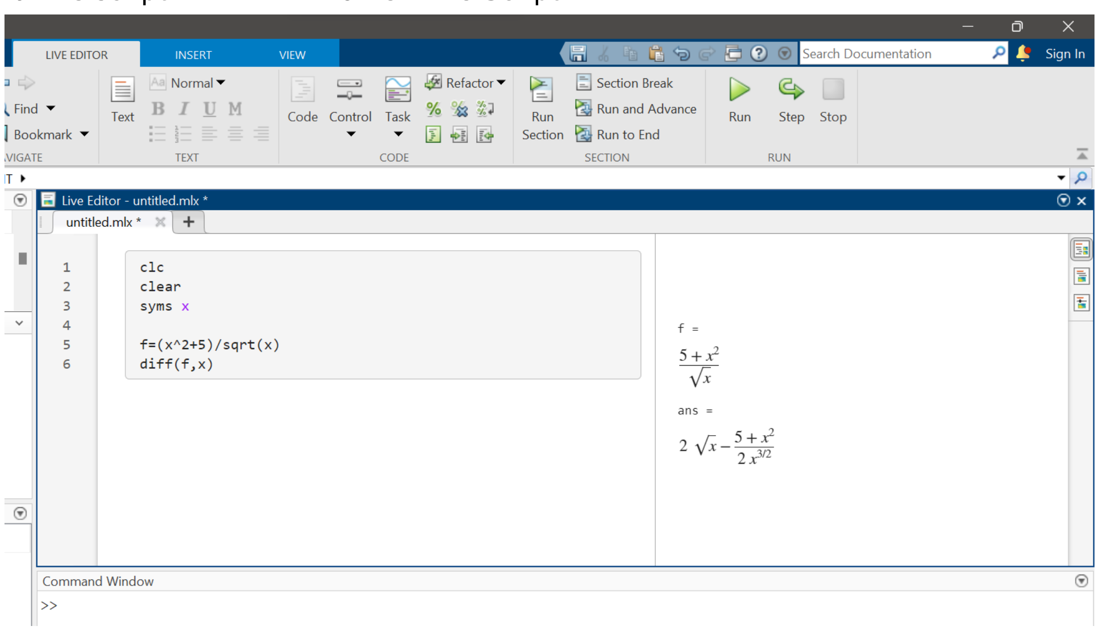
- Scripts consist of sets of MATLAB commands that are kept in basic text files.
- Script files must end with the extension '.m' (for example 'myScript.m'), and often these files are referred to as m-files.
- Essentially, m-files run a sequence of MATLAB instructions. Alternatively, they can serve as functions capable of receiving inputs, generating multiple outputs.



Working with Live Script file (.mlx file)

Live scripts and live functions are interactive documents that combine MATLAB code with formatted text, equations, and images in a single environment called the Live Editor. In addition, live scripts store and display output alongside the code that creates it.

To create a live script: MATLAB>Home>Live Script



LAB01: Plotting Curves

Plotting XY Data

Syntax

```
plot(X,Y, 'options')
```

creates a 2-D line plot of the data in Y versus the corresponding values in X, where X and Y are vectors of the same length.

Plotting function

```
fplot(f,xinterval)
```

plots expression or function f over the specified interval. Specify the interval as a two-element vector of the form [xmin xmax].

If we did not mention xinterval, fplot plots the function f over the default interval [-5 5] for x

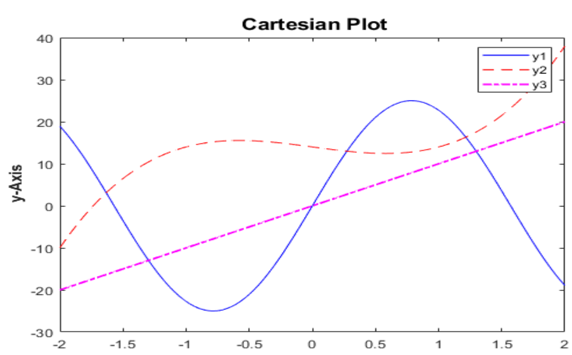
Example

Plot the functions $y1 = 50\cos x \sin x$, $y2 = 4x^3 - 4x + 14$, $y3 = 10x$ in the range $[-2,2]$

Using plot

```
clc
clear
close

x = [-2:0.01:2];
%Define the functions to plot
y1 = 50*cos(x).*sin(x);
y2 = 4*x.^3-4*x+14;
y3 = 10*x;

plot(x,y1,'b','linewidth',1)
hold on
plot(x,y2,'--r','linewidth',1)
plot(x,y3,'-.m','linewidth',1.5)
hold off
%Customization (Optional)
xlabel('\bf x-Axis'); ylabel('\bf y-Axis')
title('\fontname{Arial} Cartesian Plot','fontsize',14)

legend('y1','y2','y3')
```

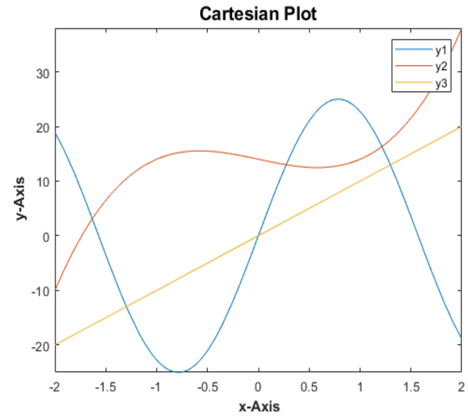
Output



Using fplot

```
clc
clear
close

syms x

y1 = 50*cos(x)*sin(x);
y2 = 4*x^3-4*x+14;
y3 = 10*x

fplot([y1,y2,y3],[-2,2])

%Customization (Optional)
xlabel('\bf x-Axis'); ylabel('\bf y-Axis')
title('Cartesian Plot','fontsize',14)

legend('y1','y2','y3')
```

Output



Plotting Polar plot

Syntax

`polarplot(theta,rho)`

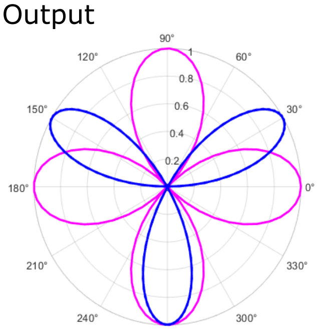
plots a line in polar coordinates, with theta indicating the angle in radians and rho indicating the radius value for each point. The inputs must be vectors of equal length or matrices of equal size.

Example

Plot the graph of the polar curves $r = \cos 2\theta\,; r = \sin 3\theta$

```
clc
clear
close all

t=linspace(0,2*pi);

r1=input("Enter the first polar curve r1=f(t): r1=");
r2=input("Enter the second polar curve r2=f(t): r2=");


polarplot(t,r1,'m',t,r2,'b','linewidth',2);
```

Output



Exercise:

Plot the graphs of the following functions:

a) $y = \sin 2x + \cos^2 x$ b) $y = x^3 + \cos 3x + tan^3 x$

Hint: Input for x^3*:* x.^3, $\tan^3 x$: (tan(x)).^3

Plot the following polar curve:

a) $r^3 = 8\cos 3\theta$ & $r^3 = 8\sin 3\theta$ b) $r = \frac{a\theta}{1+\theta}$ & $r = \frac{a}{1+\theta^2}$

Hint: Input for $\frac{a\theta}{1+\theta}$: 2*t./(1+t), $r^3 = 8\cos 3\theta \Rightarrow r = \sqrt[3]{8\cos 3\theta}$: nthroot(8*cos(3*t),3)

LAB 02:

i) Angle between radius vector and tangent vector

ii) Angle of intersection between two polar curves

i) Angle between radius vector and tangent vector

Algorithm:

- For a given curve $r = f(\theta)$, find $\phi = \cot^{-1}\left(\frac{r'}{r}\right)$, where $r' = dr/d\theta$

```
clear
clc
close
syms theta
%Get the curve
r1=input("Enter the first curve r1(theta): r1=");
%Find phi1
phi1=simplify(acot(diff(r1)/r1));
%display phi1
fprintf("\n Angle between radius vector and tangent vector to the curve %s:\n phi=
%s",r1,phi1)
```

Output

```
Enter the first curve r1(theta): r1=2*(1+cos(theta))
Angle between radius vector and tangent vector to the curve 2*cos(theta) + 2:
 phi1= -acot(sin(theta)/(cos(theta) + 1))
```

ii) Angle of intersection between two polar curves

Algorithm:

- For a given curve $r1 = f1(\theta), r2 = f2(\theta)$, find $\phi1$ and $\phi2$ using $\phi = \cot^{-1}\left(\frac{r'}{r}\right)$, where $r' = dr/d\theta$
- Solve given curves for θ value
- Get the required appropriate θ value from the solution
- Find $|\phi2 - \phi1|$ by substituting obtained θ value

```
clear
clc
close
syms theta

%Get the two curves
r1=input("Enter the first curve r1(theta): r1=");
r2=input("Enter the second curve r2(theta): r2=");
%Find phi1 and phi2

phi1=simplify(acot(diff(r1)/r1));
phi2=simplify(acot(diff(r2)/r2));

%display phi1 and phi2
fprintf("\n Angle between radius vector and tangent vector to the curve %s:\n phi1=
%s",r1,phi1)
fprintf("\n Angle between radius vector and tangent vector to the curve %s:\n phi2=
%s",r2,phi2)

%Find t(point of intersection)
```

```
fprintf('\n The values of theta at the point of intersection are: ')
S=solve(r1==r2,theta,'Real',true)

%Get the theta
tt=input("\n Choose the value of theta: ");

%Calculate the angle between the given curves
ang1= abs(vpa(subs(phi1-phi2, {theta},{tt})));
ang2=vpa(pi-ang1);
fprintf('\n Angle between given polar curves = %f or %f \n', ang1, ang2);
```

Output

```
Enter the first curve r1(theta): r1=2*(1+cos(theta))
Enter the second curve r2(theta): r2=2*(1-cos(theta))
Angle between radius vector and tangent vector to the curve 2*cos(theta) + 2:
phi1= -acot(sin(theta)/(cos(theta) + 1))
Angle between radius vector and tangent vector to the curve 2 - 2*cos(theta):
phi2= -acot(sin(theta)/(cos(theta) - 1))
The values of theta at the point of intersection are: S =
pi/2
Choose the value of theta: pi/2
Angle between given polar curves = 1.570796 or 1.570796
```

Exercise:

Find the angle of intersection between following pair of curves:

a) $\boldsymbol{r = a\cos\theta}$ and $\boldsymbol{r = b\sin\theta}$ b) $\boldsymbol{r = a(1 + sin\theta)}$ and $\boldsymbol{r = b(1 - sin\theta)}$ c) $\boldsymbol{r = a\log\theta}$ and $\boldsymbol{r = \frac{a}{\log\theta}}$

Hint: Input for $\boldsymbol{r = \frac{a}{\log\theta}}$ *: 2/((log(theta)))*

LAB 03: Partial differentiation and Jacobian

Syntax

diff(f,var,n)

computes the nth derivative of f with respect to var.

Example1

Find all the first order and second order partial derivatives of the function $U = e^{\frac{x}{y}}$. Also, find $\frac{\partial U}{\partial x}$ at x=2, $\frac{\partial U}{\partial x}$ at y=2 and $\frac{\partial U}{\partial y}$ at x=2 & y=3.

```
clear
clc
syms x y

%Define U
U(x,y)=exp(x/y);
%First derivatives
Ux=diff(U,x);
Uy=diff(U,y);
%Second derivatives
Uxx=simplify(diff(Ux,x));
Uyy=simplify(diff(Uy,y));
Uxy=simplify(diff(Ux,y));
Uyx=simplify(diff(Uy,x));

fprintf('The first order partial derivatives are \n Ux = %s \n Uy = %s \n',Ux, Uy);
fprintf('\n The second order partial derivatives are \n Uxx = %s \n Uyy = %s\n Uxy = %s \n Uyx
= %s \n',Uxx,Uyy,Uxy,Uyx);

U1=Ux(2,y);
U2=Ux(x,2);
U3=Uy(2,3);

fprintf('\n Ux(2,y)=%s; \t Ux(x,2)=%s; \t Uy(2,3)=%f \n',U1,U2,U3)
```

Output

```
The first order partial derivatives are
Ux = exp(x/y)/y
Uy = -(x*exp(x/y))/y^2

The second order partial derivatives are
Uxx = exp(x/y)/y^2
Uyy = (x*exp(x/y)*(x + 2*y))/y^4
Uxy = -(exp(x/y)*(x + y))/y^3
Uyx = -(exp(x/y)*(x + y))/y^3

Ux(2,y)=exp(2/y)/y;  Ux(x,2)=exp(x/2)/2;       Uy(2,3)=-0.432830
```

Example 2

If $z = \frac{1}{\sqrt{y^2 - 2xy + 1}}$ then prove that $x\frac{\partial z}{\partial x} - y\frac{\partial z}{\partial y} = y^2 z^3$.

```
clear all
clc
syms z x y
z=(1-2*x*y+y^2)^(-1/2);
zx=diff(z,x);
zy=diff(z,y);
LHS=simplify(x*zx-y*zy)
RHS=simplify(y^2*z^3)
if(LHS==RHS)
    fprintf('\n The given condition is satisfied\n');

else
    fprintf('\n The given condition is not satisfied')
end
```

Output

```
LHS =
y^2/(- 2*x*y + y^2 + 1)^(3/2)
RHS =
y^2/(- 2*x*y + y^2 + 1)^(3/2)
The given condition is satisfied
```

Jacobian

```
jacobian([u1,u2,…],[x1,x2,…])
```

computes the Jacobian of a vector function which is a matrix of the partial derivatives of that function

- To find the determinant find the determinant of the resultant Jacobian matrix.

Example 1

If $u = x^2 - 2y$ and $v = x + y$ then find the Jacobian $\frac{\partial(u,v)}{\partial(x,y)}$. Also, find the value of Jacobian at (1, 0).

```
clear all
clc
syms x y u v
u(x,y)=input('Enter u(x,y):');
v(x,y)=input('Enter v(x,y):');
J(x,y)=simplify(det(jacobian([u,v],[x,y])));
fprintf('\n J((u,v)/(x,y))= %s \n', J);

p=input('\n Enter the point in the form [a,b] : ');
val=J(p(1),p(2));
fprintf('\n J((u,v)/(x,y)) at (%d,%d) is %d \n', p(1),p(2), val)
```

Output

```
Enter u(x,y):x^2-2*y
Enter v(x,y):x+y
J((u,v)/(x,y))= 2*x + 2
Enter the point in the form [a,b] : [1,0]
J((u,v)/(x,y)) at (1,0) is 4
```

Example 2

If $u = x^2 - y^2$ and $v = 2xy$ where $x = r\cos t,\ y = r\sin t$ then find the Jacobian $\frac{\partial(u,v)}{\partial(x,y)}$. Also, show that $\frac{\partial(u,v)}{\partial(r,t)} = 4r^3$ and hence find the value of Jacobian $\frac{\partial(u,v)}{\partial(r,t)}$ at (3,1).

```
clear all
clc
syms x y u v r t
u(x,y)=input('Enter u(x,y):');
v(x,y)=input('Enter v(x,y):');
J1(x,y)=det(jacobian([u(x,y),v(x,y)],[x,y]));
fprintf('\n J((u,v)/(x,y))= %s \n', J1);

x=input('\n Enter x(r,t):');
y=input('Enter y(r,t):');
J2(r,t)=simplify(det(jacobian([u(x,y),v(x,y)],[r,t])));
fprintf('\n J((u,v)/(r,t))=%s \n', J2)

p=input('\n Enter the point in the form [a,b] : ');
val=J2(p(1),p(2));
fprintf('\n J((u,v)/(r,t)) at (%d,%d) is %d \n', p(1),p(2), val)
```

Output

```
Enter u(x,y):x^2-y^2
Enter v(x,y):2*x*y

J((u,v)/(x,y))= 4*x^2 + 4*y^2

Enter x(r,t):r*cos(t)
Enter y(r,t):r*sin(t)

J((u,v)/(r,t))=4*r^3

Enter the point in the form [a,b] : [3,1]
J((u,v)/(r,t)) at (3,1) is 108
```

Example 3

If $u = x\sin y\cos z$, $v = x\sin y\sin z$ and $w = x\cos y$ then find the Jacobian $\dfrac{\partial(u,v,w)}{\partial(x,y,z)}$.

```
clear all
clc
syms x y z u v w
u(x,y,z)=input('Enter u(x,y,z):');
v(x,y,z)=input('Enter v(x,y,z):');
w(x,y,z)=input('Enter w(x,y,z):');
J(x,y,z)=simplify(det(jacobian([u,v,w],[x,y,z])));
fprintf('\n J((u,v,w)/(x,y,z))= %s \n', J);
```

Output

```
Enter u(x,y,z):x*sin(y)*cos(z)
Enter v(x,y,z):x*sin(y)*sin(z)
Enter w(x,y,z):x*cos(y)
J((u,v,w)/(x,y,z))= x^2*sin(y)
```

Exercise:

a) If $u = \frac{y}{z} + \frac{z}{x}$ **the show that** $x\frac{\partial u}{\partial x} + y\frac{\partial u}{\partial y} + z\frac{\partial u}{\partial z} = 0$

b) If $f = \frac{1}{x^2} + \frac{1}{xy} + \frac{\log x - \log y}{x^2 + y^2}$ **then prove that** $x\frac{\partial f}{\partial x} + y\frac{\partial f}{\partial y} + 2f = 0$**.**

c) If $u = \frac{x^3 + y^3}{\sqrt{x + y}}$ **then prove that** $x\frac{\partial u}{\partial x} + y\frac{\partial u}{\partial y} = \frac{5}{2}u$ **.**

d) If $u = x + 3y^2 - z^2, v = 4x^2 yz, w = 2z^2 - xy$ **then evaluate** $\frac{\partial(u,v,w)}{\partial(x,y,z)}$ **at (1,-1,0).**

e) If $x = e^v \sec u$ **and** $y = e^v \tan u$ **then find the Jacobian** $\frac{\partial(x,y)}{\partial(u,v)}$

f) If $u = x^2 - 2y^2$ **and** $v = 2x^2 - y^2$ **where** $x = r\cos\theta, y = r\sin\theta$ **then show that** $\frac{\partial(u,v)}{\partial(r,\theta)} = 6r^3 \sin 2\theta.$

g) If $u = \sqrt{yz}\, v = \sqrt{zx}, w = \sqrt{xy}$ **and** $x = r\cos\varphi \sin\theta, y = r\sin\varphi \sin\theta, z = r\cos\theta$ **then find** $\frac{\partial(u,v,w)}{\partial(r,\varphi,\theta)}.$

LAB 04: Vector Calculus

Gradient, Divergence, Curl, Laplacian

Syntax	
`g = gradient(f,v)`	returns the gradient vector of symbolic scalar field f with respect to vector v in Cartesian coordinates.
`d = divergence(V,X)`	returns the divergence of symbolic vector field V with respect to vector X in Cartesian coordinates. Vectors V and X must have the same length.
`c = curl(V,X)`	returns the curl of symbolic vector field V with respect to vector X in three-dimensional Cartesian coordinates. Both the vector field V and the vector X must be vectors with three components.
`l = laplacian(f,v)`	returns the Laplacian of the symbolic field f with respect to the vector v in Cartesian coordinates. If f is an array, then the function computes the Laplacian for each element of f and returns the output l that is the same size as f.

Example 1

Find gradient of a given scalar function $f = 2yzsin(x) + 3xsin(z)\cos(y)$ at (1,2,3)

```
clear
clc
syms x y z
f(x,y,z) =input("Enter the function f(x,y,z):");
gradf=(simplify(gradient(f,[x y z])));
fprintf("grad(f)=%s\n",gradf)
p=input('\n Enter the point in the form [a,b,c] : ');
val=vpa(gradf(p(1),p(2),p(3)));
fprintf('\n grad(f) at (%d,%d,%d) is (%.4f)i + (%.4f)j + (%.4f)k
\n',p(1),p(2),p(3),val(1),val(2),val(3))
```

Output

```
Enter the function f(x,y,z):2*y*z*sin(x) + 3*x*sin(z)*cos(y)
grad(f)=[3*cos(y)*sin(z) + 2*y*z*cos(x); 2*z*sin(x) - 3*x*sin(y)*sin(z); 2*y*sin(x) +
3*x*cos(y)*cos(z)]
Enter the point in the form [a,b,c] : [1,2,3]
grad(f) at (1,2,3) is (6.3074)i + (4.6639)j + (4.6018)k
```

Example 2

Find the divergence of the given vector field $f = xyi + 2xy^2 + 3xz^3k$ at (3,4,5).

```
clear all
clc
syms x y z
F =input('Enter the components of vector F in [x y z] form : ');
div(x,y,z)=divergence(F,[x y z]);
fprintf('\n div(F) = %s \n',div);
p=input('\n Enter the point in the form [a,b,c] : ');
val=div(p(1),p(2),p(3));
fprintf('\n div(F) at (%d,%d,%d) is %d \n',p(1),p(2),p(3),val);
```

Output

```
Enter the components of vector F in [x y z] form : [x*y 2*x*y^2 3*x*z^3]
 div(F) = y + 4*x*y + 9*x*z^2
 Enter the point in the form [a,b,c] : [3,4,5]
 div(F) at (3,4,5) is 727
```

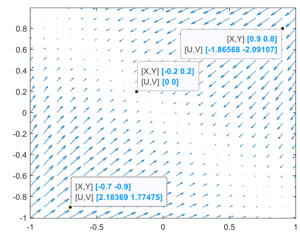
Example 3

Find the gradient of the scalar function $f = -(\sin(x) + \sin(y))^2$ and interpret geometrically

```
clear all
clc
syms x y
f = input("Enter the function f(x,y):");
gradf = gradient(f,[x y])
[X1, Y1] = meshgrid(-1:.1:1,-1:.1:1);
G1 = subs(gradf(1),[x y],{X1,Y1});
G2 = subs(gradf(2),[x y],{X1,Y1});
quiver(X1,Y1,G1,G2)
```

Output

```
Enter the function f(x,y):-(sin(x) +
sin(y))^2

gradf =

-2*cos(x)*(sin(x) + sin(y))
-2*cos(y)*(sin(x) + sin(y))
```



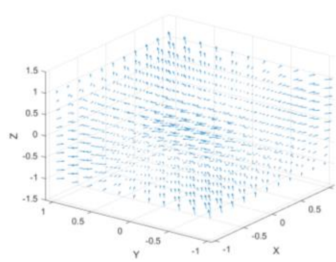
Example 4

Find the gradient of the scalar function $f = xy^2 + yz^2 + zx^2$ and interpret geometrically.

```
clear all
clc
syms x y z
f = input("Enter the function f(x,y,z):");
gradf = gradient(f,[x y z])
[X1, Y1, Z1] = meshgrid(-1:.2:1,-1:.2:1,-1:.2:1);
G1 = subs(gradf(1),[x y z],{X1,Y1,Z1});
G2 = subs(gradf(2),[x y z],{X1,Y1,Z1});
G3 = subs(gradf(3),[x y z],{X1,Y1,Z1});
quiver3(X1,Y1,Z1,G1,G2,G3)
xlabel('X')
ylabel('Y')
zlabel('Z')
```

Output

```
Enter the function
f(x,y,z):x*y^2+y*z^2+z*x^2

gradf =

2*x*z + y^2
2*x*y + z^2
x^2 + 2*y*z
```



Example 5

Find the Laplacian of the scalar function $x^3 + y^2 - logz$

```
clear
clc
syms x y z
f(x,y,z) =input("Enter the scalar function f(x,y,z):");
L=(simplify(laplacian(f,[x y z])));
fprintf("Laplacian of f(x,y,z) is %s \n",L)
```

Output

```
Enter the scalar function f(x,y,z):x^3 + y^2 - log(z)
Laplacian of f(x,y,z) is 6*x + 1/z^2 + 2
```

Example 6

To find directional derivative of function $xy^2 + yz^2 + zx^2$in the direction of the vector $i + j + k$ at (1,2,3)

```
clear all
clc
syms x y z
f= input("Enter the function f(x,y,z):");
gradf = gradient(f, [x y z])
n=input('\nEnter the components of directional vector as [x y z]:');
m = n/norm(n);
D(x,y,z) = simplify(dot(gradf',m));
fprintf('\nDirectional derivative is %s\n', D)
p=input('\nEnter the point as [a,b,c]:');
val=vpa(D(p(1),p(2),p(3)));
fprintf('Directional derivative at (%d,%d,%d) is %f\n', p(1),p(2),p(3),val)
```

Output

```
Enter the function f(x,y,z):x^2*z + x*y^2 + y*z^2
gradf =
2*x*z + y^2
2*x*y + z^2
x^2 + 2*y*z
Enter the components of directional vector as [x y z]:[1 1 1]
Directional derivative is (3^(1/2)*(x + y + z)^2)/3
Enter the point as [a,b,c]:[1,2,3]
Directional derivative at (1,2,3) is 20.784610
```

Exercise:

1. Find gradient of a given scalar function $\boldsymbol{f = xyz + 3ysin(z)cos(y)}$ at the point (1,2,3).
2. Find the divergence of the given vector field $\boldsymbol{f = xi + 2y^2 + 3z^3k}$.
3. Find div$\vec{\boldsymbol{F}}$ at the point (2,3,4) where $\vec{\boldsymbol{F}} = (\boldsymbol{y^2 + z^2 - x^2})\hat{\boldsymbol{\imath}} + (\boldsymbol{z^2 + x^2 - y^2})\hat{\boldsymbol{\jmath}} + (\boldsymbol{x^2 + y^2 - z^2})\hat{\boldsymbol{k}}$.
4. Find the gradient of the scalar function $\boldsymbol{f = x^2 - y^2}$ and interpret geometrically.
5. Find the Laplacian of the scalar function $\boldsymbol{x^2 - y^2 + 4z}$.
6. To find directional derivative of function $\boldsymbol{xcosy + xyz^2 + yzcosx}$in the direction of the vector $\boldsymbol{xy\hat{\imath} + xyz\hat{\jmath} + yz\hat{k}}$.
7. Find the directional derivative of $\phi = \boldsymbol{x^2 - 2xy + z^3}$ at the point (1,–2,–1) along the vector $\boldsymbol{2\hat{\imath} - 4\hat{\jmath} + 4\hat{k}}$.

LAB 05: Double Integrals

Plotting the region

```
clear
clc
% Region of integration
xL=input("Enter lower limit of x")
xU=input("Enter Upper limit of x")
x = linspace(xL, xU);                    % x limits
yL=input("Enter lower limit of y")
yU=input("Enter Upper limit of y")
Li = yU >= yL ;                         % Logical Vector
figure
plot(x,yU,x,yL)
hold on
patch([x(Li) fliplr(x(Li))], [yU(Li) fliplr(yL(Li))], 'b')
hold off
grid
```

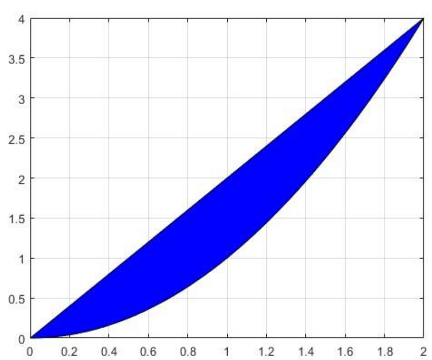
Example:

With the program to mark the region of integration in the following double integrals:

1) $\int_{0}^{2}\int_{x^2}^{2x} f(x,y)\,dy\,dx$

Output

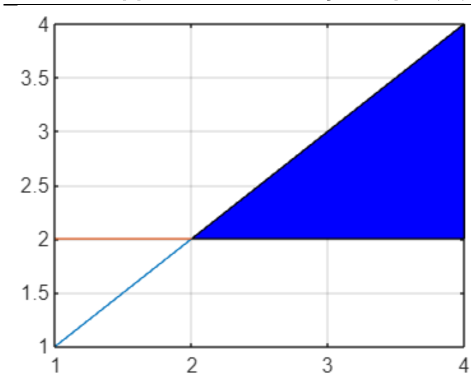
```
Enter lower limit of x: 0
Enter Upper limit of x: 2
Enter lower limit of y: x.^2
Enter Upper limit of y: 2.*x
```



2) $\int_{1}^{4}\int_{2}^{\sqrt{x}} f(x,y)\,dy\,dx$

Output

```
Enter lower limit of x: 1
Enter Upper limit of x: 4
Enter lower limit of y: 2+0.*x
Enter Upper limit of y: sqrt(x)
```



NOTE: i) If 'y' lower limit is constant, give input as yL.*x.

ii) For the polar plot patch won't work (instead of that we need to use polyfill but it is not user-friendly).

Exercise:

a) $\int_{0}^{2}\int_{2-x}^{\sqrt{4-x^2}} f(x,y)\,dy\,dx$ **b)** $\int_{0}^{\pi}\int_{0}^{\sin x} f(x,y)\,dy\,dx$ **c)** $\int_{0}^{4}\int_{x^2/4}^{2\sqrt{x}} f(x,y)\,dy\,dx$ **d)** $\int_{0}^{2}\int_{y/4}^{3-y} f(x,y)\,dy\,dx$

Double Integral

Syntax	Remark	Syntax	Remark
`integral2(f,xL,xU,yL,yU)`	f,xL,xU,yL, yU must be function handlers, i.e., use @(x,y), @(x), @(y)	`int(int(f,y,yL,yU),x,xL,xU)`	f,xL,xU,yL,yU are need not be function handlers, i.e., no need to use @(x,y), @(x), @(y)

```
clear
clc
syms x y
f = matlabFunction(input("Enter the integrand: "));
disp('f(x,y) :');
disp(f);

xL=input("Enter lower limit of x: ");
xU=input("Enter Upper limit of x: ");
yL=input("Enter lower limit of y: ");
yU=input("Enter Upper limit of y: ");

d = integral2(f,xL,xU,yL,yU);
disp("Double Integral of f(x,y) :");
disp(d);
```

Write a program to evaluate double integral.

a) $\int_1^2 \int_2^3 (x-1/y)^2 dxdy$

b) $\int_0^1 \int_x^{\sqrt{x}} xy\ dydx$

Output:

```
Enter the integrand: (x-1/y)^2
f(x,y) :
    @(x,y) (x-1./y).^2
Enter lower limit of x: 1
Enter Upper limit of x: 2
Enter lower limit of y: 2
Enter Upper limit of y: 3
Double Integral of f(x,y) : 1.2836
```

Output:

```
 Enter the integrand: x*y
f(x,y) :
    @(x,y)x.*y
Enter lower limit of x: 0
Enter Upper limit of x: 1
Enter lower limit of y: @(x) x
Enter Upper limit of y: @(x) sqrt(x)
Double Integral of f(x,y) :
    0.0417
```

Exercise:

a) $\int_0^a \int_0^{\sqrt{a-x^2}} \sqrt{a^2-x^2-y^2}\, dy\, dx$ (ans:0.5236)

b) $\int_0^{a/\sqrt{3}} \int_0^{\sqrt{x^2+a^2}} \frac{x}{x^2+y^2+a^2}\, dy\, dx$ (ans:0.1215)

c) $\int_\pi^{2\pi} \int_0^\pi \boldsymbol{y\, sinx + x\, cosy\, dxdy}$ (ans:9.8696)

d) $\int_\pi^{2\pi} \int_0^x \boldsymbol{y\, sinx + x\, cosy\, dxdy}$ (ans:32.0988)

NOTE: `matlabFunction` **which converts the variable expression into to function handler, if the integrand is a constant function then don't use this converter. Just use**

```
f = input("Enter the integrand: ");
```

LAB 06: Triple Integrals

Triple Integral

```
clear
clc
syms x y z
f = matlabFunction(input("Enter the integrand f(x,y,z) : "));
disp('f(x,y,z) :');
disp(f);

xL=input("Enter lower limit of x: ");
xU=input("Enter Upper limit of x: ");
yL=input("Enter lower limit of y: ");
yU=input("Enter Upper limit of y: ");
zL=input("Enter lower limit of z: ");
zU=input("Enter Upper limit of z: ");

d = integral3(f,xL,xU,yL,yU,zL,zU);
disp("Triple Integral of f(x,y,z) :");
disp(d);
```

Write a program to evaluate Triple integral.

a) $\int_0^1\int_0^2\int_1^2 \left(xyz^2\right) dx\,dy\,dz$

b) $\int_0^1\int_0^x\int_0^{x+y} e^{x+y+z}\,dz\,dy\,dx$

OUTPUT:

```
 Enter the integrand f(x,y,z): x*y*z^2
f(x,y,z) :
    @(x,y,z)x.*y.*z.^2
Enter lower limit of x: 0
Enter Upper limit of x: 1
Enter lower limit of y: 0
Enter Upper limit of y: 2
Enter lower limit of z: 1
Enter Upper limit of z: 2
Triple Integral of f(x,y,z) : 2.3333
```

OUTPUT:

```
Enter the integrand f(x,y,z): exp(x+y+z)
f(x,y,z) :
    @(x,y,z)exp(x+y+z)
Enter lower limit of x: 0
Enter Upper limit of x: 1
Enter lower limit of y: 0
Enter Upper limit of y: @(x) x
Enter lower limit of z: 0
Enter Upper limit of z: @(x,y) x+y
Triple Integral of f(x,y,z) : 3.6263
```

Exercise:

a) $\int_0^1\int_0^1\int_0^y xyz\,dz\,dx\,dy$

Ans: 0.0625

b) $\int_1^e\int_1^{\log y}\int_1^{e^x} \log z\,dz\,dx\,dy$

Ans: 2.7593

c) $\int_0^1\int_0^{\sqrt{1-x^2}}\int_0^{\sqrt{1-x^2-y^2}} xyz\,dx\,dy\,dz$

Ans: 0.0208

NOTE: Use @(x,y) if the limit is a function of two variable, @(x) for function of one variable

LAB 07: Line Integrals

Line integral

- Because the line integral needs to be computed repeatedly, it is advisable to encapsulate it within a function.
  To create a function
  - The function should begin as follows:
    `function [y1, ..., yN] = fun_name(x1, ..., xM)`
  - This declares a function named `fun_name` that accepts **inputs** `x1, ..., xM` and returns outputs `y1, ..., yN`. The function should end with **end**.
  - Function definition should be at the end of the main program.
  - To use the function, call it from the main main program as many times as needed.

Main program

Call function1

End of the main program

%Define the function1

function f=function1(x,y)

{

}

end

```
%%%%%%%%%%%%%%%% Main Program %%%%%%%%%%%%%%%%%%
clc
clear
syms x y
LInt=0;
%Get the number of sub path in the curve C
n=input("How many subcurves are ther in the curve C? :")
%Calculate the line integral along each path
for i=1:n
    line(i)=LineIntegral(x,y);
    LInt =vpa(LInt +line(i))
end
%%%%%%%%%%%%%%%% Main Program ends %%%%%%%%%%%%%%%%%
%Define function for Line integral
function line = LineIntegral(x,y)
option=input("1: y=f(x)\n2: x=g(y)\n")

switch option
    case 1
        y=input("Enter the function f(x): \n")
        r=[x y];
        F=input("Enter the vector function F(x,y)=[F1 F2]: F(x,y)")
        dr=diff(r,x);
        integrand1=dot(F,dr);
        xL=input("Enter the lower limit of x:");
        xU=input("Enter the Upper limit of x:");
        line=int(integrand1,x,xL,xU)
    case 2
        x=input("Enter the function g(y): \n")
        r=[x y];
        F=input("Enter the vector function F(x,y)=[F1 F2]: F(x,y)")
        dr=diff(r,y);
        integrand1=dot(F,dr);
        yL=input("Enter the lower limit of y:");
        yU=input("Enter the Upper limit of y:");
        line=int(integrand1,y,yL,yU)
    otherwise
        fprintf("Give the proper input")
end

end
%%%%%%%%%%%%%%%% Line Integral definition ends %%%%%%%%%%%%%%%%%
```

Example 1

Evaluate $\int_c F \cdot dr$, where c is the curve in plane $y = 2x^2$ from $(0,0)$ to $(1,2)$ and $F = 3xy\,\hat{\imath} - y^2\hat{\jmath}$.

```
How many subcurves are ther in the curve C? :1
n =      1
1: y=f(x)
2: x=g(y)
1
option =      1
Enter the function f(x):
2*x^2
y =
2*x^2
Enter the vector function F(x,y)=[F1 F2]: F(x,y)=[3*x*y -y^2]
F =
[6*x^3, -4*x^4]
Enter the lower limit of x:0
Enter the Upper limit of x:1
line =
-7/6
LInt =
-1.1666666666666666666666666666667
```

Line integral using parametric form

```
clear
clc
syms x y z t
x=input("Enter x(t):"); % If x ,y, z is a function of 't'
y=input("Enter y(t):");
z=input("Enter z(t):");

r=[x y z];
F=input("Enter the vector function F(x,y,z):")
dr=diff(r,t);
integrand1=dot(F,dr);
tl=input("Enter the lower limit of t:");
tu=input("Enter the Upper limit of t:");
line=int(integrand1,t,tl,tu)
```

Example 1

Evaluate $\int_c F \cdot dr$, where c is the curve given by $x = 2t^2, y = t, z = t^3$ from $(0,0,0)$ to $(2,1,1)$ and $F = (2y+3)\hat{\imath} + xz\hat{\jmath} + (yz - x)\hat{k}$.

Output

```
Enter x(t):2*t^2
Enter y(t):t
Enter z(t):t^3
Enter the vector function F(x,y,z):[2*y+3 x*z yz-x]
Enter the vector function F(x,y,z):[2*y+3 x*z y*z-x]
F =
[3 + 2*t, 2*t^5, - 2*t^2 + t^4]
Enter the Lower limit of t:0
Enter the Upper limit of t:1
line =
288/35
```

Exercise:

1. If $\vec{F} = xy\,\hat{i} + yz\,\hat{j} + zx\,\hat{k}$ evaluate $\int_c \vec{F}.d\vec{r}$ where c is the curve represented by $x = t, y = t^2, z = t^3, -1 \le t \le 1$.
2. Find the circulation of $\vec{F}$ round the curve c, where $\vec{F} = (x-y)\,\hat{i} + (x+y)\,\hat{j}$ and c is the circle $x^2 + y^2 = 4,\ z = 0$.
3. Find the total work done by a force $\vec{F} = 2xy\,\hat{i} - 4z\,\hat{j} + 5x\,\hat{k}$ along the curve $x = t^2,\ y = 2t\ +1,\ z = t^3$ from $t = 0$ to $t = 1$.
4. Evaluate $\int_c \vec{F}.d\vec{r}$, where $\vec{F} = \cos y\,\hat{i} - x\sin y\,\hat{j}$ and c is the curve $y = \sqrt{1-x^2}$ in xy-plane from (1, 0) to (0, 1).

LAB 08: Green's Theorem

Greens theorem

$M(x,y), N(x,y)$ be continuous in a region R of xy plane bounded by a closed curve c then

$$\oint_c Mdx + Ndy = \int\int_R \frac{\partial N}{\partial x} - \frac{\partial M}{\partial y}\, dxdy$$

Algorithm

- Create a function for LHS: Evaluate $\oint_c Mdx + Ndy$ along the curve c using Line integral
- Create a function for RHS: Evaluate $\int\int_R \frac{\partial N}{\partial x} - \frac{\partial M}{\partial y}\, dxdy$ using multiple integrals
- Check whether LHS=RHS or not

```
%%%%%%%%%%%%%%%% Main Program %%%%%%%%%%%%%%%%%%
clc
clear

syms x y

LHS=0;
%Get the number of sub path in the curve C
n=input("How many sub curves are ther in the curve C? :")
%Calculate the line integral along each path (LHS)
for i=1:n
    line(i)=LineIntegral(x,y);
    LHS=vpa(LHS+line(i))
end
%Calculate the double integral (RHS)
RHS=vpa(MultiIntegral(x,y))

%%%%%%%%%%%%%%%% Main Program ends %%%%%%%%%%%%%%%%%%
%Define function for Line integral
function line = LineIntegral(x,y)
option=input("1: y=f(x)\n2: x=g(y)\n")

switch option
    case 1
        y=input("Enter the function f(x): \n")
        r=[x y];
        F=input("Enter the vector function F(x,y)=[F1 F2]: F(x,y)=")
        dr=diff(r,x);
        integrand1=dot(F,dr);
        xL=input("Enter the lower limit of x:");
        xU=input("Enter the Upper limit of x:");
        line=int(integrand1,x,xL,xU)
    case 2
        x=input("Enter the function g(y): \n")
        r=[x y];
        F=input("Enter the vector function F(x,y)=[F1 F2]: F(x,y)=")
        dr=diff(r,y);
        integrand1=dot(F,dr);
        yL=input("Enter the lower limit of y:");
        yU=input("Enter the Upper limit of y:");
        line=int(integrand1,y,yL,yU)
    otherwise
        fprintf("Give the proper input")
```

```
end

end

%%%%%%%%%%%%%%%% Line Integral definition ends %%%%%%%%%%%%%%%%%%
%Define function for Double integral
function d = MultiIntegral(x,y)

M= input("Enter the M(x,y): ");
N= input("Enter the N(x,y): ");
f=diff(N,x)-diff(M,y);

disp('f(x,y) :');
disp(f);

xL=input("Enter lower limit of x: ");
xU=input("Enter Upper limit of x: ");
yL=input("Enter lower limit of y: ");
yU=input("Enter Upper limit of y: ");

d = int(int(f,y,yL,yU),x,xL,xU);
end
%%%%%%%%%%%%%%%% Double Integral definition ends %%%%%%%%%%%%%%%%%%
```

Example 1

Verify Green's theorem for $\int_c (y-\sin x)dx+\cos y\,dy$ where c is the plane triangle enclosed by the line $y=0,\ x=\dfrac{\pi}{2},\ y=\dfrac{2x}{\pi}$.

Output

```
How many subcurves are ther in the curve C? :3
n =
     3
1: y=f(x)
2: x=g(y)
1
option =
     1
Enter the function f(x):
0
y =
     0
Enter the vector function F(x,y)=[F1 F2]: F(x,y)=[y-sin(x) cos(y)]
F =
[-sin(x), 1]
Enter the lower limit of x:0
Enter the Upper limit of x:pi/2
line =
-1

LHS =
-1.0

1: y=f(x)
2: x=g(y)
2
option =
```

```
     2
Enter the function g(y):
pi/2
x =
    1.5708
Enter the vector function F(x,y)=[F1 F2]: F(x,y)=[y-sin(x) cos(y)]
F =
[- 1 + y, cos(y)]
Enter the lower limit of y:0
Enter the Upper limit of y:1
line =
sin(1)

LHS =
-0.1585290151921034933474976783697

1: y=f(x)
2: x=g(y)
1
option =
     1
Enter the function f(x):
2*x/pi
y =
(2*x)/pi
Enter the vector function F(x,y)=[F1 F2]: F(x,y)=[y-sin(x) cos(y)]
F =
 [(2*x)/pi - sin(x), cos((2*x)/pi)]
Enter the lower limit of x:pi/2
Enter the Upper limit of x:0
line =
1 - sin(1) - pi/4

LHS =
-0.78539816339744830961566084581988

Enter the M(x,y): y-sin(x)
Enter the N(x,y): cos(y)
f(x,y) :
-1

Enter lower limit of x: 0
Enter Upper limit of x: pi/2
Enter lower limit of y: 0
Enter Upper limit of y: 2*x/pi

RHS =

-0.78539816339744830961566084581988
```

Exercise:

1. Verify Green's theorem for $\int_c (3x^2 - 8y^2)dx + (4y - 6xy)dy$ where c is the plane triangle enclosed by the line $x = 0, y = 0, x + y = 1.$

2. . Verify Green's theorem for $\int_c (x^2 - y^2)dx + 2xydy$ where c is the rectangle bounded by rectangle $y = 0, x = 0, y = b$ and $x = a.$

LAB 09: Modular Arithmetic- Diophantine Equation

Remainder function – (mod)

mod function is used to find the remainder when one number is divided by another. If R is the remainder obtained when X is divided by Y then R=mod(X,Y)

Syntax of the mod function:

```
R = mod (X, Y)
```

Example 1: 'mod' function for scalar inputs

Code	Output
`R = mod (15, 6)`	R = 3

Solving linear Diophantine equation

The simplest linear Diophantine equation takes the form $ax + by = c$, where a, b and c are given integers. The solutions are described by the following theorem:

This Diophantine equation has a solution (where x and y are integers) if and only if c is a multiple of the greatest common divisor of a and b. Moreover, if (x, y) is a solution, then the other solutions have the form $(x + kv, y - ku)$, where k is an arbitrary integer, and u and v are the quotients of a and b (respectively) by the greatest common divisor of a and b.

Let $ax + by = 0, x, y \in Z$ be a homogeneous linear Diophantine equation. If $gcd(a, b) = d$, then the complete family of solutions to the above equation is $x = \frac{b}{d}k$, and $y = -\frac{a}{d}k$, $k \in Z$.

To find the particular solution of the Diophantine equation:

```
clc
clear;
syms x y t
assume([x y], 'integer')        % assume 'x' and 'y' are integers
eqn = 50*x +20*y == 300 ;      % declare the equation
disp(eqn)
c = coeffs(lhs(eqn) ,[y,x]); %c(1)->Coefficient of x, c(2)->Coefficient of y
r=rem(rhs(eqn),gcd(c(1),c(2)));
if r~=0
    disp('No Solution')
else
    fprintf('%d is multiple of gcd(%d,%d)=%d, hence it has 
solution\n',rhs(eqn),c(1),c(2),gcd(c(1),c(2)))
    if rhs(eqn)~=0
        [xSol, ySol] = solve(eqn,[x y]);  % solve for ;x' and 'y'
        xComp=xSol+t*(c(2)/gcd(c(1),c(2)));
        yComp=ySol+t*(-c(1)/gcd(c(1),c(2)));
    else
        xSol=c(2)/gcd(c(1),c(2));
        ySol=-c(1)/gcd(c(1),c(2));
        xComp=xSol*t;
        yComp=ySol*t;
```

```
    end
    fprintf(' Solution is : x=%d, y=%d\n',xSol,ySol)

    fprintf('General Solution [x,y]:\n')
    fprintf('x=')
    disp(xComp)
    fprintf('y=')
    disp(yComp)
end
```

Output

```
50*x + 20*y == 300
300 is multiple of gcd(50,20)=10, hence it has solution
Solution is : x=-30, y=90
General Solution [x,y]:
x=2*t - 30
y=90 - 5*t
```

Exercise:

1. Find the particular solution of the following linear Diophantine Equations:

a) $6x+9y=0$ b) $2x+4y=21$ c) $20x+16y=500$

2. Find all the solutions (x, y) to the Diophantine equation $11x + 13y = 369$ for which x and y are both positive.

LAB 10: Euclid's Algorithm & Linear Congruences

GCD of two numbers m, n using Euclid's algorithm

```
clc
clear;
prompt1 = "Enter the first number: ";
m = input(prompt1); %Get the first number
prompt2 = "Enter the second number: ";
n = input(prompt2); %Get the first number
ans=euclid(m,n); %Call the recursive function euclid
if ans==-1
   fprintf('Enter the numeric values\n')
elseif ans==-2
   fprintf('Enter the integral values\n')
elseif ans==-3
   fprintf('Enter the positive values\n')
else
   fprintf('%d\n',ans)
end

function [res] = euclid(m,n)
    % Are m and n the right types?
    if ~isnumeric(m) || ~isnumeric(n)
        res = -1;
        return
    end
    % Are m and n integer-like?
    if m ~= int64(m) || n ~= int64(n)
        res = -2;
        return;
    end
    % Are m and n greater than zero?
    if m <= 0 || n <= 0
        res = -3;
        return
    end
    % Swap m and n if m less than n
    % to allow the algorithm to function properly
    if m < n
        tmp = m;
        m = n;
        n = tmp;
    end
    % Result of modulus is zero so we have found the gcd
    if mod(m,n) == 0
        fprintf('GCD(%d,%d)=>',m,n);
        res = n;
    else
        fprintf('GCD(%d,%d)=>',m,n);
        res = euclid(n,mod(m,n));  % Euclid's algorithm
    end

end
```

Output

```
Enter the first number: 18
Enter the second number: 480
GCD(480,18)=>GCD(18,12)=>GCD(12,6)=>6
```

Exercise

Find GCD of the following: $1)\ (24,30),\quad 2)\ (128,96)$

Solution of a linear congruence $ax \equiv c(mod\ m)$

```
clc
clear;
prompt1 = "To solve Linear Congruence ax =c(mod m)\n Enter a: ";
a = input(prompt1);
prompt2 = "Enter c: ";
c = input(prompt2);
prompt3 = "Enter m: ";
m = input(prompt3);


[g,u0,v0] = EuclidMatrix(m,a);
    if ( mod(c,g) ) ~= 0
        disp('No solutions.')
        solutions = [];
        return
    end

    disp([ num2str(m) 'x + ' num2str(a) 'y = ' num2str(g)  ])
    disp(['Number of Solutions: (' num2str(u0) ')' num2str(m)...
      ' + (' num2str(v0) ')' num2str(a) ' = ' num2str(g) ]);

    u = u0 -  (a / g);
    v = v0 +  (m / g);
    x = v * (c / g);
    y = u * (c / g);
    S = @(k) x + k * (m/g);
    solutions = mod(S(0:g-1),m);
    fprintf('solution=%f\n',solutions)

    % Euclidean Algorithm, Beazout's Coefficients are stored in matrix
    function [g,u,v] = EuclidMatrix(a,b)
        M = [ 1 0; 0 1 ];
        n = 0;
        while (b ~= 0)
            q = floor(a/b);
            M = M * [ q 1; 1 0];
            t = a;
            a = b;
            b = t - q * b;
            n = n + 1;
        end
        g = a;
        u = ( -1 )^n * M(2,2);
        v = (-1)^(n+1) * M(1,2);
        disp([ num2str(u) 'x + ' num2str(v) 'y = ' num2str(g)  ])
    end
```

Output

```
To solve Linear Congruence ax =c(mod m)
 Enter a: 15
Enter c: 6
Enter m: 7
-2x + 1y = 1
7x + 15y = 1
Number of Solutions: (-2)7 + (1)15 = 1
solution=6.000000
```

Exercise:

Solve the following congruences

1) $2x \equiv 51(mod\ 8)$ **2)** $4x \equiv 26(mod\ 7)$ **3)** $25x \equiv 15(mod\ 29)$

4) $9x \equiv 42(mod\ 6)$ **5)** $6x \equiv 15(mod\ 21)$