output / input

Input control - input
load

output control - ;
disp
disp with num2str

output in desired file [save
fprintf (gives more flexibility)

function file

output control

→ disp('____') → output comes as string

disp (_)

eg >> x = 4;
>> disp (x)
= 4
>> disp ('x')
= 'x'

→ disp ([ ]) (when we want print string with some no.)

eg >> x = 4;
>> disp (['the value of x = ', num2str (x)])

title (' ')

title ([' the circle of radius = ', num2str(R)])

>> a = 3; b = 4;
disp (a);
disp ('a');
disp ([' the value of a = ', num2str(a), ' & b = ', num2str(b)
])

~~Output~~

Save → ascii filename.exc variables

eg x = 0:Pi/10: 2*Pi;
y_1 = sin(x);
y_2 = Cos(x);
z = [x' y_1' y_2'];

y_1

- type data1.dat z

fopen → open the file
fprintf → Print the formating file
fclose → close the file

Scanned by CamScanner

(if we use % inside ' ' then it indicates that we give some format)

file identify

fid = fopen('filename.exc', 'w')

(variables which are we want to print)

fprintf (fid, 'output format', variable)

formate for 1 variable → % W.d(e/f/g)

Total width

% 10.3 g

if it is an integer then we use

% 10 d

fclose (fid)

\n → use for next line

(use d → for integer)

i j | k = i*j;

1 * 1 = 1

1 * 2 = 2

fprintf('%5d * %5d = %5d', i, j, k)

Q sol. of quad. Coordinates

$ax^2 + bx + c$

if $(b^2 - 4ac) > 0$

$1 + n^2 - \frac{1}{n^2} + 1$

Q sum of series $\frac{1}{n^2} + n + 1$ $n = 1$ to 1000

Q vertical motion under gravity

plot vertical displacement v/s Time

using data u & T g, u & T using MatLab

calculate s

* function file

Plotting commands 2D Plotting, 3D Plotting → Plot3 histogram
→ Single Plot → Plot
→ axis
→ x label
→ y label
→ title
→ legend
→ text
→ g text
→ grid on

→ surf, surfc
→ mesh, meshc
→ contour

→ Multiple Plot : Plot
hold on

Q write a script file for making circle

$x = r\cos\theta$, $y = r\sin\theta$

Sol

```
r = input ('enter the radius');
th = linspace (0, 2*Pi, 500);
x = r * cos(th);
y = r * sin(th);
plot (x,y)
```

* How to convert script file into function file

```
function [output variable] = filname (input variable)

end
```

• 1st line is always start with (function) eg:

```
function [x, y] = circ_mat(r)
th = linspace (0, 2*pi, 500);
x = r * sin(th);
y = r * cos(th);
plot (x,y);
end
```

how excute the function file.

```
>> circ_mat (10)
```

if we want to plot with colourfull lines, diff. line type etc, then we use,

plot (x,y, 'style option')

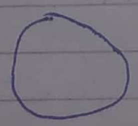
Style option are: lin style , line color , marker — * , o

lin style: ——— , - - - - , — · — ·

line color: r, w, b,

otherwise default options are selected that is blue colour, & solid lines.

eg plot (x,y,' k —')



plot (x,y,'k - - -')

where k is color
— or - - - are line type

* axis → control the limit of the axis
→ control the expected graph of the general axis

~~axis~~ axis ([xmin xmax ymin ymax])

if we not want to control xmin then we use

axis ([-inf xmax ymin ymax])

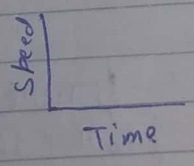
write ~~plate~~ -inf in the place of that which we don't want to control.

axis ('equal') (control the least values of x,y axis)
axis ('square') (" " size & shape)

→ xlabel ('string')
→ ylabel ('string')

string → whatever we want to show on x or y axis.

→ eg xlabel ('Time')
ylabel ('Speed')



→ tittle ('~~sinte~~ circle plot')
OR
→ tittle (['circle of radius = ' ; num2str(r)])

→ legend command is use for multiple plot

* text

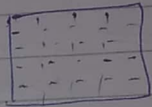
Command → text(xlocation, ylocation, 'string')

eg ~~text~~ text(0,0,'circle')

* gtext

Command → gtext('centre')

* grid on : use to on the grid



* Multiple Plot :-

for multiple plot, we have 3 options —
→ different window
→ graph are Coline
→ graph shows at diff location in 1 window

eg.

```
clear all;
th = linspace(0,2*pi,500);
x1 = sin(th);
x2 = cos(th);
x3 = sin(th).^2;
x4 = cos(th).^2;
```

Case I

```
plot(th,x1)
plot(th,x2)
plot(th,x3)
plot(th,x4)
```

By this syntax we find at last graph which is plot(th,x4)

Case II

```
figure;
plot(th,x1);
figure;
plot(th,x2);
figure;
plot(th,x3);
figure;
plot(th,x4);
figure;
plot(th,x1,th,x2,th,x3
```

By this syntax, all graph are show in 1 window at diff location of all fig show in one window

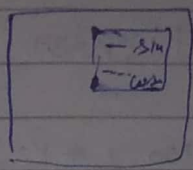
Case III plot (x1, y1, 'r--', x2, y2, x3, y3 ---)

legend

plot (th, x1, th, x2, th, x3, th, x4)
legend ('sinx', 'cosx', 'sinsquare', 'cossquare') / use to identify the multiple graph in same window in case of multiple plots)

Hold on

figure;
plot (th, x1)
hold on
plot (th, x2)
plot (th, x3)
plot (th, x4)
hold off



(All Plot after hold on commands are Plot on 1st Plot which are written before hold on command)

Ques By making of function like the and hold on command & Plot circle whose radius is 1 upto 100.

Sol

```
function [x, y] = cir_mat (r)
 th = linspace (0, 2*pi, 500)
 x = r * sin(th);
 y = r * sin(th);
 figure;
 Plot (x, y)
 hold on
 for r = 1:100,
  hold on
  Circ_mat (r)
hold off end
```

Pause command eg → for r = 1:100
  Circ_mat (r)
  Pause (1)
 end

After make 1st circle command is stop for 1sec then make new circle again stop 1sec and so on

Con. +C → Terminate the command

* call function file to plot servaral plot

```
function [x,y] = fun_file(r)
% clear all
%   r= input ('enter radius: ');
%. r =10 ;
theta = 0: pi / 100 : 2 * pi;
x = r * cos ( theta):
y = r * sin ( theta);
% plot (x,y)
% axis ('equal')
end
```

Q write a programme to make circle whose radius is 2, 3, 4, 5

```
for R=2:1:5
    [Z1 , Z2] = fun_file(R)
    plot (Z1, Z2)
    hold on
    axis ('equal')
end
```

Gussian value $\varphi = A_0 e^{-(r-r_0)^2/w_0^2}$

```
% gaussion function with variable centre ' r°, A° & w°)
function[gauss] = gauss_cen (r, r0, A0, w0);
% r=-10: 0.1: 10 ;
```

(divide fig. widnow into n no. of rows and columns)

* **Subplot (m, n, P)**
  row column , value of P = 1(m x n)

fig	2 fig
3	4

```
th = 0: Pi /20 : 2 * Pi ;
SIN = sin (th);
COS = cos (th);
SINS = sin (th).^2 ;
COSS = cos (th).^2;
% figure;
subplot (2,2,1)
Plot (th, SINS)
% figure'
Subplot (2, 2, 2)
```

~~Plot (th, SINS)~~
~~Subplot (2,2,3)~~
~~Plot (th, COSS)~~
~~SU~~

```
Plot (th, COS)
% figure
subplot (2,2,3)
Plot (th, sins)
% figure
Subplot (2,2,4)
Plot (th ,coss)
```

plotyy → use to ~~show~~ plot different scale

eg plot (x1 ,y1, x2 ,y2)

```
x = 0:100;
y1 = x;
y2 = x.^2;
figure;
Plot (x,y1, x,y2)
```

commands

3 D Plot , → meshc (gives mesh command as well as contourcomm

→ mesh

→ surf

→ surfc (gives surf command as well as contour)

→ contour

figure

% subplot (2,1,1)

contour (x, y, z, 100)

% contour (x, y, z, [0.36 0.64]) (calculating contour on z plane on used values)

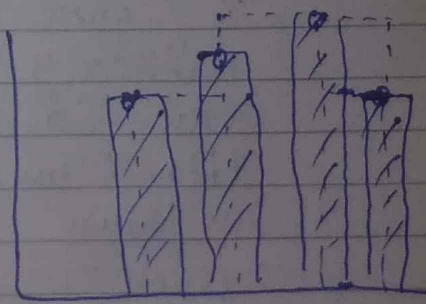
% hold on

% contour (x, y, z, 0.32, 'w...')

% hold off

---

stem	mean
stairs	median
bar	std (standard division)
barh	min
polar	max
fplot	sort
ezplot	diff
pie	sum



→ stem (x,y)

bar (make bar)

barh (make bar horizontally)

→ polar (theta, r)

→ fplot ('function', [ ])

→ ezplot ('function', [ ])

Scanned by CamScanner

x = [1 2 3 4 5]
pie(x) x/sum(x) (Calculate % of each element contribut

th = 0: pi /100 : 2*pi ;
r = sin (th).^2 ;
Polar (th, r, 'b')

Polar 8

% function Plot
figure;
ezplot ('sin x./x', [-10 10] ;
figure;

function Plot

clc
% x = [1 2 3 4 5];
% xp = x
A = [6 0 -2; 7 4 -1; 8 3 0; 9 2 1; 10 2 2]
% mean
% mean(A)
% mean(A, 1)
% mean(A, 2) .
% meadian (x)
% median (A)
% std (A)
% std (A, 0, 1)
% std (A, 1, 1)

% 1 : Column wise
% 2 : Row wise

mean(A) by default
mean(A, 1) column
mean(A, 2) Row

C, R
(1, 2)

std (A, flag = 0, 1, dim) flag = definition
If std (A, 1) by flag
by default division is Column wise

(division done by two method flag is use to show the method (0, 1))

% max (A)
% max (A, [ ], 1)
% max (A, [ ], 2)

Scanned by CamScanner

optional

max (A, [ ], 1/2)

OR

[value index] = max (A, [ ], 1/2)

min (A, [ ], 1/2)

eg : A = [1 2 3; 4 5 6; 7 8 9];

max (A, [ ], 1)

max (A, [ ], 2)

max (A, 1) — Relace some elements with 1 of in a matrix which are less than 1

max (A, 2) — less than 2

min (A, 1) relace elements with 1 which are more than 1

min (A, 2) " " " " " " 2

[d, index] = max (A, [ ], 1)

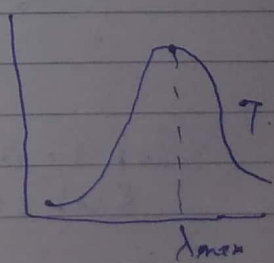
show max value — show Position of max value

Q $\lambda_{max} T = b$

λ = Range

μ :

u = formula



[x, y] = max (u, [ ], 1)

Rotation

eg

eg

* sum (A, 1/2) (sum Rows or column wise)

* cumsum (A, 1/2)

eg. A = [1 2 3; 4 5 6; 7 8 9]

= 1 2 3
4 5 6
7 8 9

cumsum (A, 1)

= 1 2 3
5 7 9
12 15 18

sum (A, 1)
= 12 15 18

* prod (A, 1/2)
* comprod (A, 1/2)

* sort (A, 1/2) (sort min. to max value)

eg sort (A, 1)
= 1 2 3
4 6 7
2 8 5

A = 1 2 3
7 6 4
5 8 2

How much times we want slp

* slop 1 = diff (A, 1, 1)
slop 1 = diff (A, 1, 1)

y = diff (x, n) ; diff (x, n, dim)

eg = 1 4 1
1 -1 1
1 -1 1
1 0 1

A = 6 0 -2
7 4 -1
8 3 0
9 2 1
10 2 2

Curve fitting

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$$

for n^{th}, a' coeff $= n+1$

$= [a_n \quad a_{n-1} \quad a_{n-2} \quad \dots \quad a_1 \quad a_0]$

Highter order of x

$$x^5 - 4x^3 + 3x^2 + 2x + 1 = 0$$

$[1 \quad 0 \quad -4 \quad 3 \quad 2 \quad 1]$

order → Higher to lower

Two steps — Polyfit
— Polyval

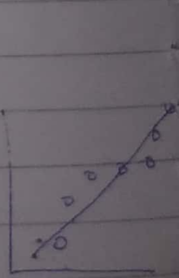
order of polynomial

P. Syntax Polyfit → a = polyfit (x, y, order)

y_i = polyval (a, x_i) (calculate y value corresponding to x_i)

eg:

```
x = [1 2 3 4 5 6 7 8];
y = [1 2 8.5 4.5 5 5.7 6.8];
a = Polyfit (x, y, 1)        No. of data
xi = linspace(1, 8, 100);
yi = polyval (a, xi);
Plot (x, y, 'o', xi, yi);
legend ('exp data Point', 'fitted curve')
```



of cylinder
Pressure is changes with time

$$\boxed{P_t = P_0 e^{-t/\tau}}$$

$$\underbrace{\log P_t}_{Pbar} = \log P_0 - \underbrace{t}_{t = tbar}/\tau$$

```
t = [        ];
P = [        ];
Pbar = P log P0;
tbar = t;
a = polyfit(tbar, pbar, 1);
tau = -1/(a(1))
P0 = exp(a(2));
```

~~Pbar = log P~~

$Pbar = \log P_0 - \frac{1}{\tau} tbar$

$a(1) = -\frac{1}{\tau}$

$a(2) = \log P_0$

$\therefore P_0 = e(a(2))$

$\tau = -1/(a(1));$

```
t = [0 0.5 1.0 5.0 10 20];
p = [760 625 528 85 14 0.16];
pbar = log(P);
tbar = t;
a = polyfit(tbar, pbar, 1);
tau = -1/a(1);   disp(['coeff: P0 = ' num2str(P0), 'and tau', num2str(tau)]);
P0 = exp(a(2));
tnew = linespace(0, 20, 100);
pnew = P0.*exp(-tnew/tau);
plot(t, P, 'o', tnew
```