Parag Mandal

ENG20CS0246

MATLAB Workshop

```
clc; close all; clear all;
```

1. Change the format of output in command prompt with ‘format’ command A=2*109 Get following output on command prompt:

- **a = 2.000000000000000e+19**
- **a = 2.0000e+19**
- **a = 2e+19**

```
A = 2 * 10^9;
```

```
format long;
A
```

```
A =
     2.000000000000000e+09
```

```
format short;
A
```

```
A = 2.0000e+09
```

```
format shortG;
A
```

```
A =
        2e+09
```

```
clc; close all; clear all;
```

2. Create an array and matrix:

- **A= [1 2 3 4 5 6 7]**
- **B= [1 2 3 4; 5 6 7 8; 9 10 11 12]**

```
A = 1:7
```

```
A = 1×7
     1     2     3     4     5     6     7
```

```
B = [1:4; 5:8; 9:12]
```

```
B = 3×4
     1     2     3     4
     5     6     7     8
     9    10    11    12
```

```
clc; close all; clear all;
```

3. Concatenate the arrays and matrices M= [3 4 6 2] N= [1 2 5 7; 12 15 2 5] O= [3 4 6 2; 1 2 5 7; 12 15 2 5]

```
M = [3 4 6 2];
N = [1 2 5 7; 12 15 2 5];
O = [3 4 6 2; 1 2 5 7; 12 15 2 5];
MAT = [M; N; O]
```

```
MAT = 6×4
     3     4     6     2
     1     2     5     7
    12    15     2     5
     3     4     6     2
     1     2     5     7
    12    15     2     5
```

```
clc; close all; clear all;
```

4. Multiply and add two matrices M1=[6 4 3;7 1 5; 2 1 6] M2=[2 3 6 ; 6 2 4; 3 3 1]

```
M1=[6 4 3; 7 1 5; 2 1 6];
M2=[2 3 6; 6 2 4; 3 3 1];
mat_sum = M1 + M2
```

```
mat_sum = 3×3
     8     7     9
    13     3     9
     5     4     7
```

```
mat_pro = M1*M2
```

```
mat_pro = 3×3
    45    35    55
    35    38    51
    28    26    22
```

```
clc; close all; clear all;
```

5. Create all zero matrix of size 3*4

```
Z = zeros(3,4)
```

```
Z = 3×4
```

```
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

```
clc; close all; clear all;
```

6. Generate symbolic equation ^3 + ^2 + /6

```
syms x y
f = x^3 + y^2 + x/6
```

```
f =
```

$$x^3 + \frac{x}{6} + y^2$$

7. Integrate and differentiate the above symbolic equation

```
f_int = int(f)
```

```
f_int =
```

$$\frac{x\,(3\,x^3 + x + 12\,y^2)}{12}$$

```
f_diff = diff(f)
```

```
f_diff =
```

$$3\,x^2 + \frac{1}{6}$$

```
clc; close all; clear all;
```

8. Perform round, floor and ceil operations on following numbers respectively i) 22.45632 ii) 5.4453 iii) 78.564

```
% Given numbers
num1 = 22.45632;
num2 = 5.4453;
num3 = 78.564;

% Round operation
rounded1 = round(num1)
```

```
rounded1 =
    22
```

```
rounded2 = round(num2)
```

```
rounded2 =
     5
```

```
rounded3 = round(num3)
```

```
rounded3 =
    79
```

```
% Floor operation
floored1 = floor(num1)
```

```
floored1 =
    22
```

```
floored2 = floor(num2)
```

```
floored2 =
     5
```

```
floored3 = floor(num3)
```

```
floored3 =
    78
```

```
% Ceil operation
ceiled1 = ceil(num1)
```

```
ceiled1 =
    23
```

```
ceiled2 = ceil(num2)
```

```
ceiled2 =
     6
```

```
ceiled3 = ceil(num3)
```

```
ceiled3 =
    79
```

```
clc; close all; clear all;
```

9. Find the value of () ^3 where m=22.32

```
m = 22.32;
result = nthroot(m, 3) * pi^3
```

```
result =
        87.3
```

```
clc; close all; clear all;
```

10. Find the area of the circle Area = π r2, radius=2.4

```
radius = 2.4;
area = pi*radius^2
```

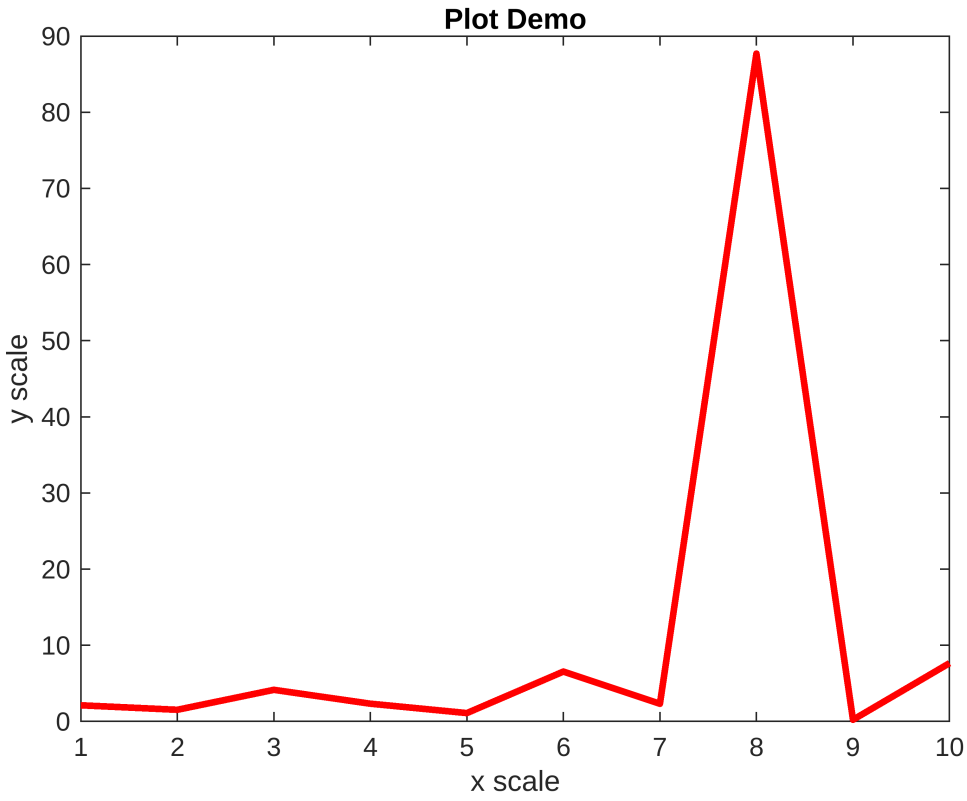
```
area =
       18.096
```

```
clc; close all; clear all;
```

11. Generate a plot with following specifications

- **X axis-> a=[1 2 3 4 5 6 7 8 9 10]**
- **Y axis-> b=[2.12 1.53 4.15 2.331 1.1 6.55 2.32 87.7 0.2212 7.6544]**
- **Line width= 2.5**
- **Line color = red**
- **Title = 'Plot demo'**
- **X axis label = ' x scale'**
- **Y axis label = 'Y scale'**

```
a = 1:10;
b = [2.12 1.53 4.15 2.331 1.1 6.55 2.32 87.7 0.2212 7.6544];
plot(a, b, '-r', 'LineWidth', 2.5)
title('Plot Demo')
xlabel('x scale')
ylabel('y scale')
```




```
clc; close all; clear all;
```

12. Write a script to display following text for 5 times in subsequent rows 'Welcome to MATLAB' using while loop & for loop.

```
% Using while loop
i = 1;
while i <= 5
    disp('Welcome to MATLAB');
    i=i+1;
end
```

```
Welcome to MATLAB
Welcome to MATLAB
Welcome to MATLAB
Welcome to MATLAB
Welcome to MATLAB
```

```
% Using for loop
for counter = 1:5
    disp('Welcome to MATLAB');
end
```

```
Welcome to MATLAB
Welcome to MATLAB
```

```
Welcome to MATLAB
Welcome to MATLAB
Welcome to MATLAB
```

```
clc; close all; clear all;
```

13. Reverse the vector elements. Example: Input x = [1,2,3,4,5,6,7,8,9] , Output y = [9,8,7,6,5,4,3,2,1]

```
x = 1:10
```

```
x = 1×10
     1     2     3     4     5     6     7     8     9    10
```

```
y = flip(x)
```

```
y = 1×10
    10     9     8     7     6     5     4     3     2     1
```

```
clc; close all; clear all;
```

14. Find the mean and median of the vector [432 354 323 312 458 344 565 675 453]

```
v = [432 354 323 312 458 344 565 675 453];
v_mean = mean(v)
```

```
v_mean =
       435.11
```

```
v_median = median(v)
```

```
v_median =
   432
```

```
clc; close all; clear all;
```

15. Solve x^4 - 5*x^2 + 6*x = 2

```
syms x
f = x^4 - 5*x^2 + 6*x - 2;
solve(f, x)
```

```
ans =
```

$$\begin{pmatrix} 1 \\ 1 \\ -\sqrt{3}-1 \\ \sqrt{3}-1 \end{pmatrix}$$

```
clc; close all; clear all;
```

16. Calculate the integral of the following function within the limits [0, inf]

f(x)= (e^(-x^2))*(log(x)^2) .

```
syms x
f = (exp(-x^2))*(log(x)^2);
int(f, 0, inf)
```

```
ans =
```

$$\frac{\sqrt{\pi}\left((\text{eulergamma}+\log(4))^2+\frac{\pi^2}{2}\right)}{8}$$

```
clc; close all; clear all;
```

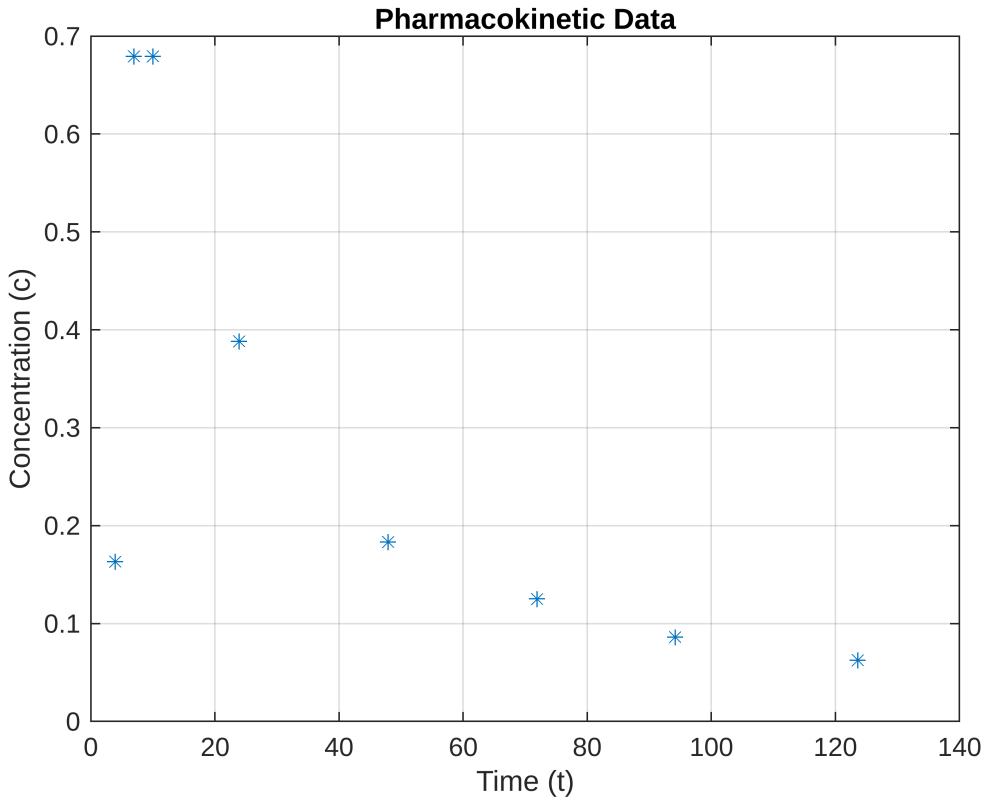
17. Consider the Pharmacokinetic Data

- **t = [3.92, 6.9, 9.99, 23.90, 47.87, 71.91, 94.15, 123.6]**
- **c = [0.163, 0.679, 0.679, 0.388, 0.183, 0.125, 0.086, 0.0624]**

Plot the data point use multistart optimization to fit a curve S

```
% Given Data
t = [ 3.92, 6.9, 9.99, 23.90, 47.87, 71.91, 94.15, 123.6];
c = [0.163, 0.679, 0.679, 0.388, 0.183, 0.125, 0.086, 0.0624];
```

```
figure
plot(t, c, '*')
xlabel('Time (t)');
ylabel('Concentration (c)');
title('Pharmacokinetic Data');
grid on
hold on
```




```
% Model [Got from Internet]
model = @(b, t) b(1)*exp(-b(4)*t) + b(2)*exp(-b(5)*t) + b(3)*exp(-b(6)*t);
```

```
% Using Multistart Optimization
% Use multistart optimization to fit the curve
problem = createOptimProblem('lsqcurvefit', ...
                             'objective', model, ...
                             'xdata', t, ...
                             'ydata', c, ...
                             'x0', ones(1,6), ...
                             'lb', [-10 -10 -10 0 0 0], ...
                             'ub', [10 10 10 0.5 0.5 0.5])
```

```
problem = struct with fields:
    objective: @(b,t)b(1)*exp(-b(4)*t)+b(2)*exp(-b(5)*t)+b(3)*exp(-b(6)*t)
           x0: [1 1 1 1 1 1]
        xdata: [3.92 6.9 9.99 23.9 47.87 71.91 94.15 123.6]
        ydata: [0.163 0.679 0.679 0.388 0.183 0.125 0.086 0.0624]
           lb: [-10 -10 -10 0 0 0]
           ub: [10 10 10 0.5 0.5 0.5]
        Aineq: []
        bineq: []
          Aeq: []
          beq: []
      nonlcon: []
       solver: 'lsqcurvefit'
      options: [1×1 optim.options.Lsqcurvefit]
```

```
b = lsqcurvefit(problem)
```

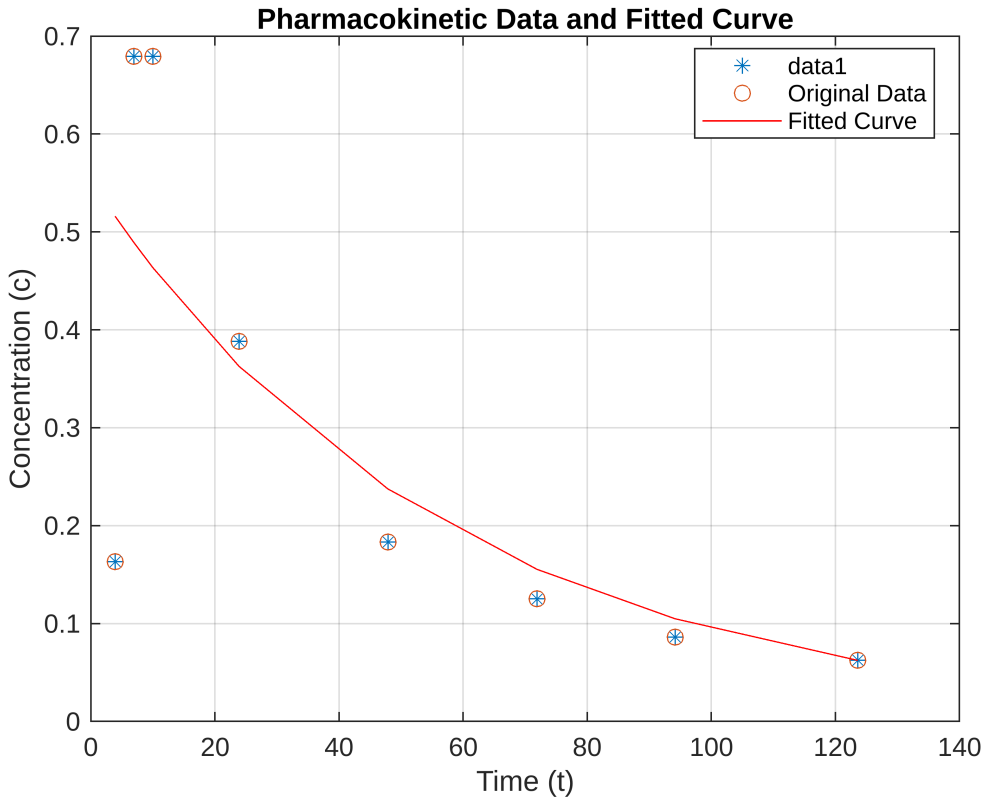
```
Local minimum possible.

lsqcurvefit stopped because the final change in the sum of squares relative to
its initial value is less than the value of the function tolerance.

<stopping criteria details>
b = 1×6
      0.18436      0.18436      0.18428     0.017656     0.017656     0.017656
```

```
fitted_curve = model(b, t);
```

```
plot(t, c, 'o', 'DisplayName', 'Original Data');
hold on;
plot(t, fitted_curve, 'r', 'DisplayName', 'Fitted Curve');
xlabel('Time (t)');
ylabel('Concentration (c)');
title('Pharmacokinetic Data and Fitted Curve');
legend('Location', 'best');
grid on;
hold off;
```




```
ms = MultiStart
```

```
ms =
```

```
  MultiStart with properties:

         UseParallel: 0
             Display: 'final'
   FunctionTolerance: 1e-06
             MaxTime: Inf
           OutputFcn: []
             PlotFcn: []
    StartPointsToRun: 'all'
          XTolerance: 1e-06
```

```
[b, fval, exitflag, output, solutions] = run(ms, problem, 50)
```

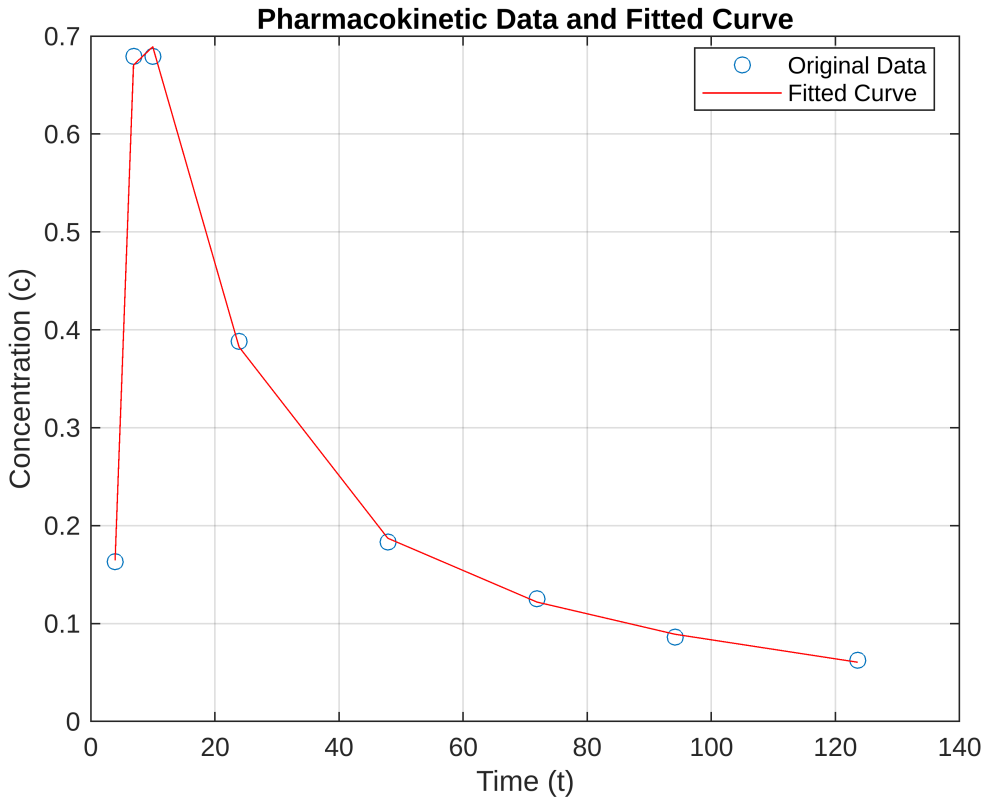
```
MultiStart completed the runs from all start points.

All 50 local solver runs converged with a positive local solver exit flag.
b = 1×6
      -6.1063      0.29688      0.99784          0.5     0.012886     0.075473
fval =
   0.00024653
exitflag =
     1
output = struct with fields:
                funcCount: 12187
         localSolverTotal: 50
       localSolverSuccess: 50
    localSolverIncomplete: 0
    localSolverNoSolution: 0
                  message: 'MultiStart completed the runs from all start points.⊠⊠All 50 local solver run
solutions = 1×48 GlobalOptimSolution
                                                                                                    ...
```

	1	2	3	4
1	1×1 GlobalOptimSolution	1×1 GlobalOptimSolution	1×1 GlobalOptimSolution	1×1 GlobalOptimSolution

```
fitted_curve = model(b, t);
```

```
plot(t, c, 'o', 'DisplayName', 'Original Data');
hold on;
plot(t, fitted_curve, 'r', 'DisplayName', 'Fitted Curve');
xlabel('Time (t)');
ylabel('Concentration (c)');
title('Pharmacokinetic Data and Fitted Curve');
legend('Location', 'best');
grid on;
hold off;
```


Pharmacokinetic Data and Fitted Curve
Original Data
Fitted Curve
Concentration (c)
Time (t)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
20
40
60
80
100
120
140