1

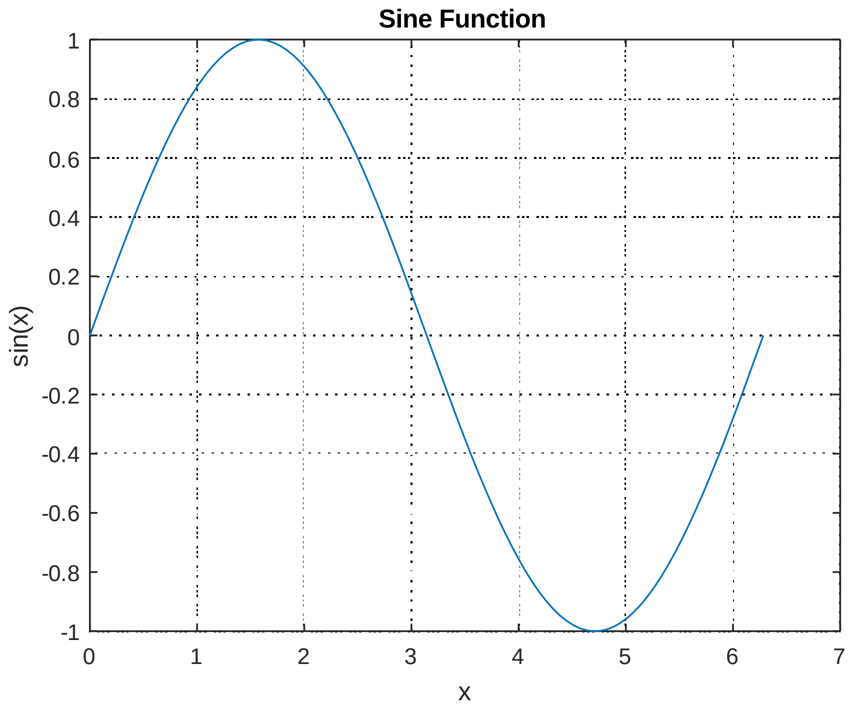
---

2 Representaion graphic in 2D and 3D

2.1 TP 01

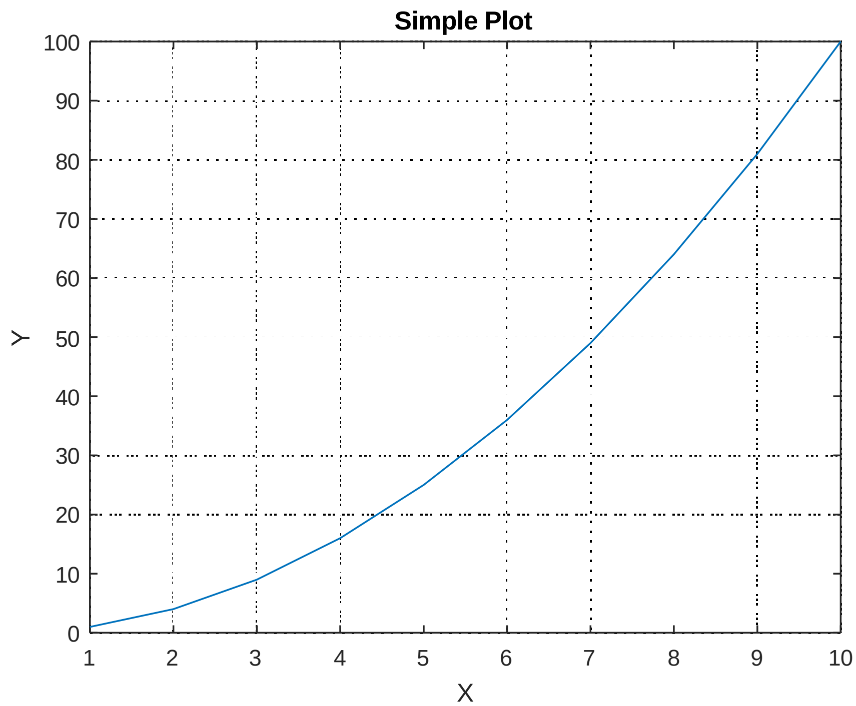
Let's use the plot command in MATLAB, a widely used programming environment for numerical computing

```
% Generate data
x = linspace(0, 2*pi, 100); % Create an array of 100 points from 0 to 2*pi
y = sin(x); % Compute sin of each point
% Plot
plot(x, y);
title('Sine Function');
xlabel('x');
ylabel('sin(x)');
grid on;
```



2.2 TP 02

```
% Define data
x = 1:10;
y = x.^2;
% Plot
plot(x, y);
title('Simple Plot');
xlabel('X');
ylabel('Y');
grid on
```

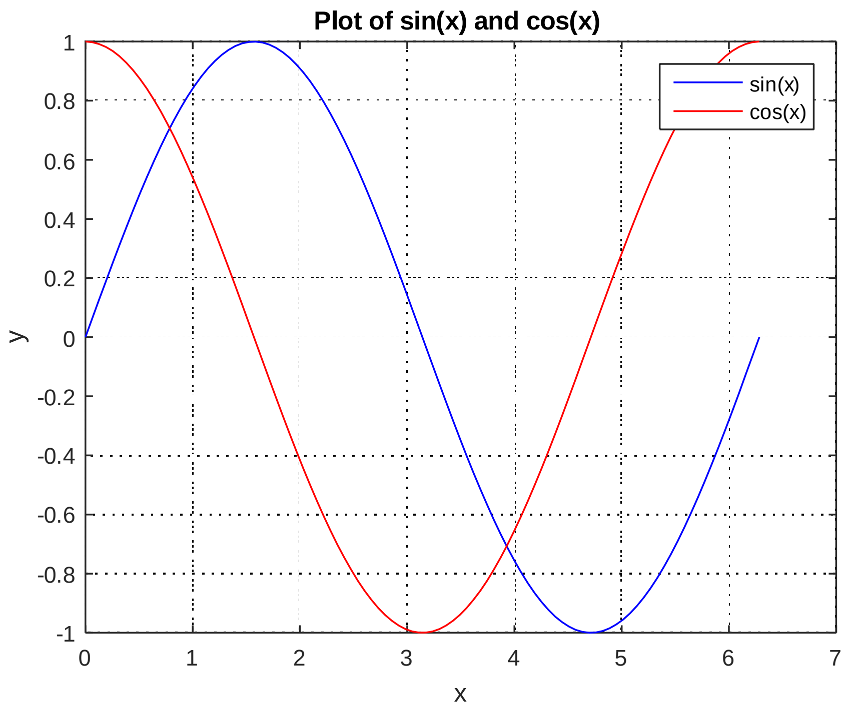


2.3 TP 03

———————————

```
% Define data
x = linspace(0, 2*pi, 100); % Array of 100 points from 0 to 2*pi
y1 = sin(x); % Compute sin of each point
y2 = cos(x); % Compute cos of each point
% Plot
plot(x, y1, 'b', x, y2, 'r');
title('Plot of sin(x) and cos(x)');
xlabel('x');
ylabel('y');
legend('sin(x)', 'cos(x)');
grid on;
```

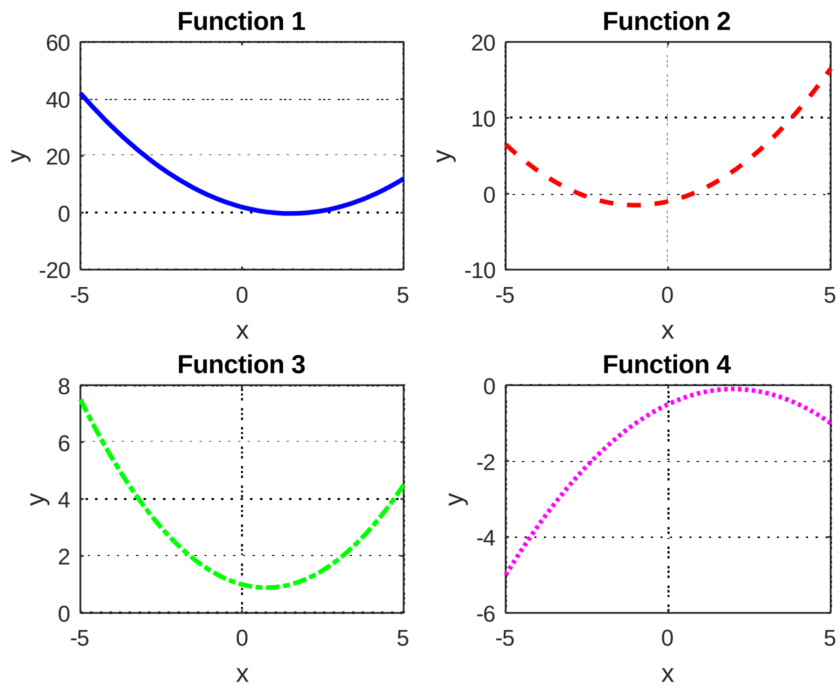
——————————-



2.4 TP 04

% Define data
x = linspace(-5, 5, 100); % Array of 100 points from -5 to 5
% Define polynomial coefficients for four functions
coefficients1 = [1, -3, 2]; % Coefficients for the polynomial 1 - 3x + 2x^2
coefficients2 = [0.5, 1, -1]; % Coefficients for the polynomial 0.5 + x - x^2
coefficients3 = [0.2, -0.3, 1]; % Coefficients for the polynomial 0.2 - 0.3x + x^2
coefficients4 = [-0.1, 0.4, -0.5]; % Coefficients for the polynomial -0.1 + 0.4x - 0.5x^2
% Compute y-values for each polynomial
y1 = polyval(coefficients1, x);
y2 = polyval(coefficients2, x);
y3 = polyval(coefficients3, x);
y4 = polyval(coefficients4, x);
% Create subplots
figure;

```
% Subplot 1
subplot(2, 2, 1);
plot(x, y1, '-b', 'LineWidth', 2);
title('Function 1');
xlabel('x');
ylabel('y');
% Subplot 2
subplot(2, 2, 2);
plot(x, y2, '–r', 'LineWidth', 2);
title('Function 2');
xlabel('x');
ylabel('y');
% Subplot 3
subplot(2, 2, 3);
plot(x, y3, '-.g', 'LineWidth', 2);
title('Function 3');
xlabel('x');
ylabel('y');
% Subplot 4
subplot(2, 2, 4);
plot(x, y4, ':m', 'LineWidth', 2);
title('Function 4');
xlabel('x');
ylabel('y');
```



3 The constraints graphically

3.1 TP 01

To visualize the constraints graphically, we can plot the feasible region defined by the constraints and then overlay the contour plot of the objective function to see how it behaves within that region. Let's illustrate this in MATLAB:

```
% Define the constraint function
g = @(x, y) x + y - 2;
% Define the range of x and y
x = linspace(-1, 2, 100);
y = linspace(-1, 2, 100);
% Create a grid of points
[X, Y] = meshgrid(x, y);
% Evaluate the constraint function at each point
Z = g(X, Y);
% Plot the contour of the constraint function
contour(X, Y, Z, [0 0], 'k', 'LineWidth', 2);
hold on;
```

```
% Plot the feasible region (shaded area)
fill([0 1 1 0], [1 0 2 2], 'r', 'FaceAlpha', 0.2);
hold on;
% Plot the contour of the objective function
[X, Y] = meshgrid(-1:0.1:2, -1:0.1:2);
Z = X.^2 + Y.^2+1;
contour(X, Y, Z, 'LineWidth', 1);
colorbar;
% Add labels and title
xlabel('x');
ylabel('y');
title('Feasible Region and Contour Plot of Objective Function');
% Add legend
legend('Constraint: x + y = 2', 'Feasible Region', 'Contour of Objective
Function');
```

This script will create a plot showing the feasible region defined by the constraint

$x + y = 2$ (represented by a black line) and overlay the contour plot of the objective function

$f(x, y) = x^2 + y^2$

The feasible region will be shaded in red.