```
% ==========================================
% Undergraduate Project: Interpolation & Curve Fitting
% ==========================================

clc; clear; close all;

% Example data points (you can replace with experimental data)
x = [0 1 2 3 4 5 6 7 8 9];
y = [2.1 2.9 4.2 5.1 7.3 9.0 11.1 13.4 15.6 18.1];

%% 1. Polynomial Interpolation
xx = linspace(min(x), max(x), 200); % Fine grid
p_coeff = polyfit(x, y, length(x)-1); % Polynomial of degree n-1
y_poly_interp = polyval(p_coeff, xx);

figure;
plot(x, y, 'ro', 'MarkerSize', 8, 'LineWidth', 2); hold on;
plot(xx, y_poly_interp, 'b-', 'LineWidth', 1.5);
title('Polynomial Interpolation');
xlabel('x'); ylabel('y');
legend('Data points','Interpolated polynomial','Location','NorthWest');
grid on;

%% 2. Piecewise Cubic Spline Interpolation
y_spline = spline(x, y, xx);

figure;
plot(x, y, 'ko', 'MarkerSize', 8, 'LineWidth', 2); hold on;
plot(xx, y_spline, 'r-', 'LineWidth', 1.5);
title('Cubic Spline Interpolation');
xlabel('x'); ylabel('y');
legend('Data points','Spline interpolation','Location','NorthWest');
grid on;


%% 3. Curve Fitting (Least Squares)

% (a) Linear Fit
p1 = polyfit(x, y, 1);
y_fit1 = polyval(p1, xx);

% (b) Quadratic Fit
p2 = polyfit(x, y, 2);
y_fit2 = polyval(p2, xx);

% (c) Cubic Fit
p3 = polyfit(x, y, 3);
y_fit3 = polyval(p3, xx);

figure;
```

```
plot(x, y, 'ko', 'MarkerSize', 8, 'LineWidth', 2); hold on;
plot(xx, y_fit1, 'r-', 'LineWidth', 1.5);
plot(xx, y_fit2, 'b--', 'LineWidth', 1.5);
plot(xx, y_fit3, 'g-.', 'LineWidth', 1.5);
title('Curve Fitting using Least Squares');
xlabel('x'); ylabel('y');
legend('Data points','Linear Fit','Quadratic Fit','Cubic
Fit','Location','NorthWest');
grid on;

%% 4. Nonlinear Curve Fitting Example
% Suppose we expect an exponential model: y = a*exp(b*x)
ft = fittype('a*exp(b*x)', 'independent','x','coefficients',{'a','b'});
[exp_fit, gof] = fit(x', y', ft, 'StartPoint', [1, 0.2]);

figure;
plot(exp_fit, x, y);
title('Nonlinear Curve Fitting (Exponential)');
xlabel('x'); ylabel('y');
legend('Data','Exponential Fit');
grid on;

% Display fitted parameters and goodness of fit
disp(exp_fit);
```

```
     General model:
     exp_fit(x) = a*exp(b*x)
     Coefficients (with 95% confidence bounds):
       a =       3.045  (2.456, 3.635)
       b =      0.2033  (0.1774, 0.2292)
```

```
disp(gof);
```

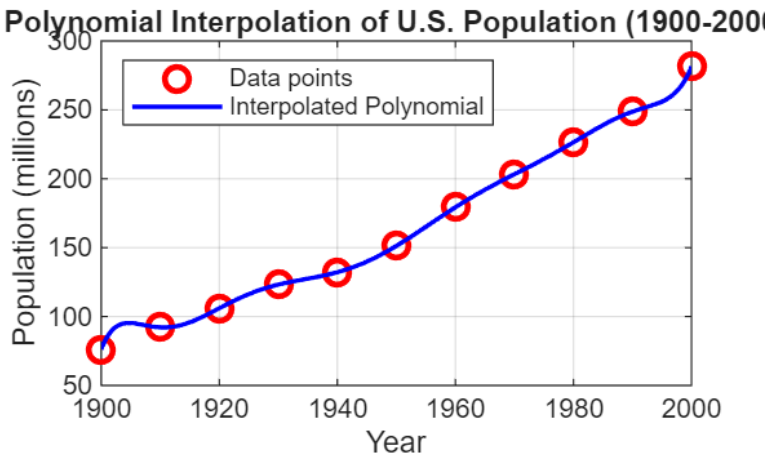
```
           sse: 4.4919
       rsquare: 0.9837
           dfe: 8
     adjrsquare: 0.9817
          rmse: 0.7493
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%% Part Two %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ================================================
% Interpolation & Curve Fitting with Real Data
% U.S. Census Population Data (1900 - 2000)
% ================================================

clc; clear; close all;

% U.S. Census population data (in millions)
% Source: U.S. Census Bureau (decennial census)
```

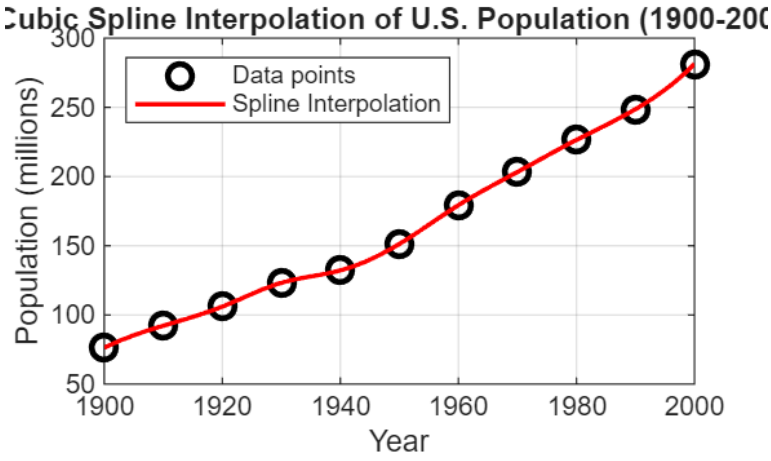
```
years = 1900:10:2000;
population = [76.2, 92.2, 106.0, 123.2, 132.2, ...
              151.3, 179.3, 203.3, 226.5, 248.7, 281.4];

% Normalize years to avoid numerical instability
x = years - 1900; % start from 0
y = population;

xx = linspace(min(x), max(x), 500);

%% 1. Polynomial Interpolation (degree n-1)
p_coeff = polyfit(x, y, length(x)-1);
```

```
Warning: Polynomial is badly conditioned. Add points with distinct X values, reduce the degree of the
polynomial, or try centering and scaling as described in HELP POLYFIT.
```

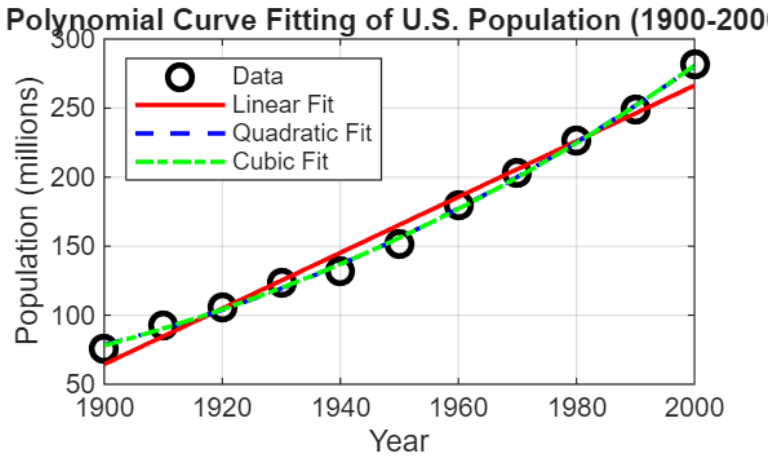
```
y_poly_interp = polyval(p_coeff, xx);

figure;
plot(years, y, 'ro', 'MarkerSize', 8, 'LineWidth', 2); hold on;
plot(xx+1900, y_poly_interp, 'b-', 'LineWidth', 1.5);
title('Polynomial Interpolation of U.S. Population (1900-2000)');
xlabel('Year'); ylabel('Population (millions)');
legend('Data points','Interpolated Polynomial','Location','NorthWest');
grid on;
```




```
%% 2. Cubic Spline Interpolation
y_spline = spline(x, y, xx);

figure;
plot(years, y, 'ko', 'MarkerSize', 8, 'LineWidth', 2); hold on;
plot(xx+1900, y_spline, 'r-', 'LineWidth', 1.5);
title('Cubic Spline Interpolation of U.S. Population (1900-2000)');
xlabel('Year'); ylabel('Population (millions)');
legend('Data points','Spline Interpolation','Location','NorthWest');
grid on;
```

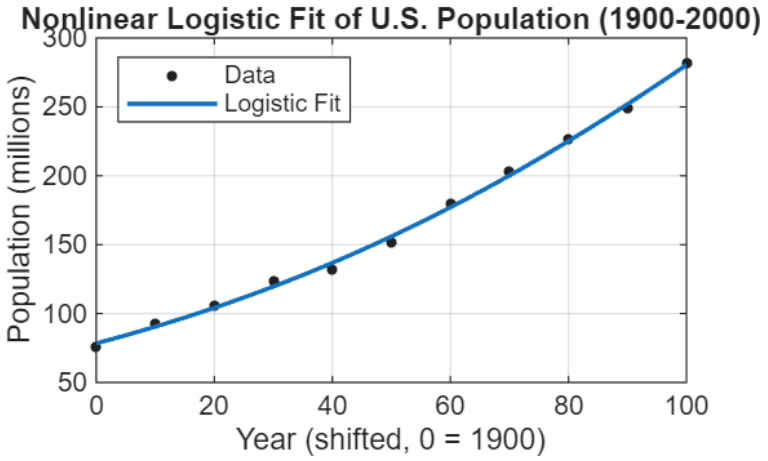



```
%% 3. Polynomial Curve Fitting (Regression)
% Fit lower-degree polynomials to avoid overfitting
p1 = polyfit(x, y, 1);  % Linear
p2 = polyfit(x, y, 2);  % Quadratic
p3 = polyfit(x, y, 3);  % Cubic

y_fit1 = polyval(p1, xx);
y_fit2 = polyval(p2, xx);
y_fit3 = polyval(p3, xx);

figure;
plot(years, y, 'ko', 'MarkerSize', 8, 'LineWidth', 2); hold on;
plot(xx+1900, y_fit1, 'r-', 'LineWidth', 1.5);
plot(xx+1900, y_fit2, 'b--', 'LineWidth', 1.5);
plot(xx+1900, y_fit3, 'g-.', 'LineWidth', 1.5);
title('Polynomial Curve Fitting of U.S. Population (1900-2000)');
xlabel('Year'); ylabel('Population (millions)');
legend('Data','Linear Fit','Quadratic Fit','Cubic Fit','Location','NorthWest');
grid on;
```




```
%% 4. Nonlinear Curve Fitting (Logistic Model)
```

```
% Logistic model: P(t) = K / (1 + A*exp(-B*t))
ft = fittype('K ./ (1 + A*exp(-B*x))', ...
             'independent','x', ...
             'coefficients',{'K','A','B'});

[logistic_fit, gof] = fit(x', y', ft, ...
    'StartPoint', [400, 10, 0.03]); % initial guesses

figure;
plot(logistic_fit, x, y);
title('Nonlinear Logistic Fit of U.S. Population (1900-2000)');
xlabel('Year (shifted, 0 = 1900)'); ylabel('Population (millions)');
legend('Data','Logistic Fit','Location','NorthWest');
grid on;
```




```
disp('Logistic Fit Parameters:');
```

```
Logistic Fit Parameters:
```

```
disp(logistic_fit);
```

```
     General model:
     logistic_fit(x) = K ./ (1 + A*exp(-B*x))
     Coefficients (with 95% confidence bounds):
       K =       798.8  (280, 1318)
       A =        9.19  (3.008, 15.37)
       B =     0.01604  (0.01291, 0.01916)
```

```
disp('Goodness of Fit:');
```

```
Goodness of Fit:
```

```
disp(gof);
```

```
           sse: 95.5261
       rsquare: 0.9979
           dfe: 8
    adjrsquare: 0.9974
          rmse: 3.4555
```