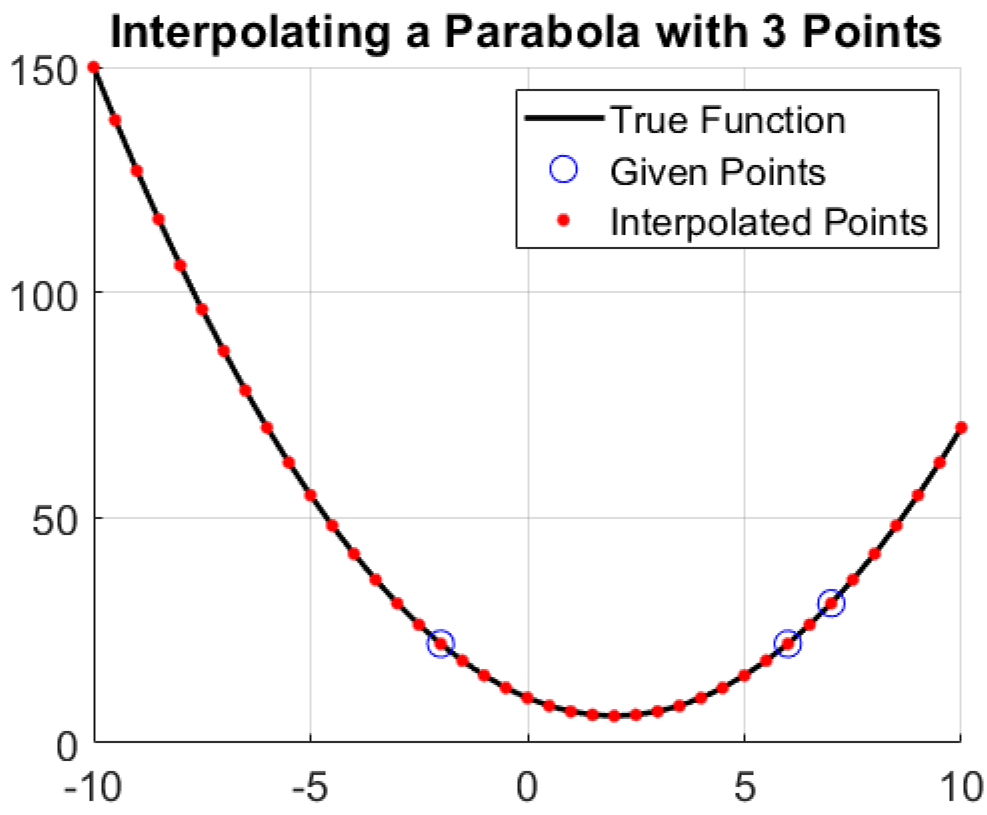
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MATH_151_Lab4
%--------------------------------------------------------------------------
% C Rocheleau, Colorado State University
% 9/23/2023
%--------------------------------------------------------------------------
% Answer key for MATH-151 Lab 4 for the Fall 2023 semester
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all; clear all; clc;
```

Task 1: Creating a Polynomial

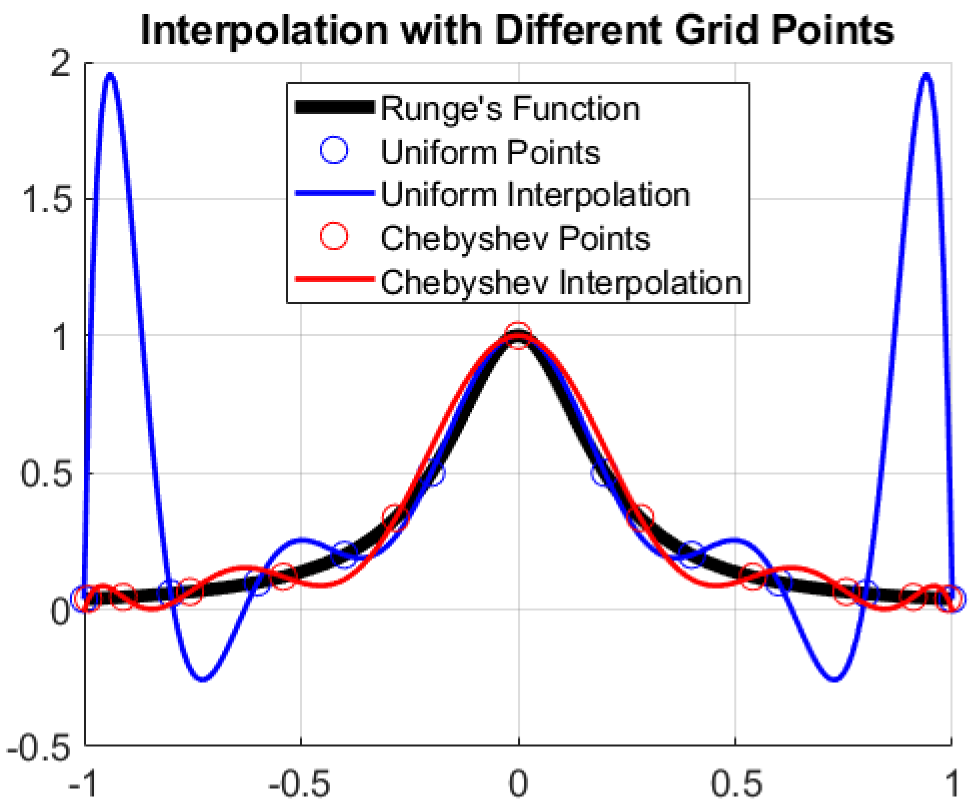
```
% (a) See lagrange_interp.m or Newton_interp.m to see interpolation functions

% (b) Let's use our function to calculate points on our interpolating
% polynomial

% Define our given poitns (-2.22), (6,22), (7,31)
x_given = [-2,  6,  7];
y_given = [22, 22, 31];

% Define our X grid for and calculate Y
X = -10:0.5:10;
% Y = lagrange_interp(x_given, y_given, X);
Y = Newton_interp(x_given, y_given, X);

% (c) Plot and compare to x^2 -4x + 10
figure(); hold on; grid on;
h_true  = plot(X, X.^2 - 4*X + 10, 'k-','linewidth',2); % Plot truth
h_given = plot(x_given, y_given, 'bo','markersize',10); % Plot given points
h_int   = plot(X, Y, 'r.','markersize',15);             % Plot interpolation

set(gca,'fontsize',16);
legend([h_true, h_given, h_int],{'True Function','Given Points','Interpolated
 Points'})
title('Interpolating a Parabola with 3 Points');
```



Task 2: Runge's Function

```
% (a) Calculate and plot the true Runge's function
Runge_x = -1:0.01:1;
Runge_y = 1./(1 + 25*Runge_x.^2);

figure(); hold on; grid on;
h_runge = plot(Runge_x, Runge_y, 'k-','linewidth',5); % Plot truth

% (b) Calcuate and plot interpolation with Uniform grid
unif_x  = linspace(-1, 1, 11);
unif_y_given  = 1./(1 + 25*unif_x.^2);
unif_y_interp = lagrange_interp( unif_x, unif_y_given, Runge_x);

h_unif_pts = plot(unif_x, unif_y_given, 'bo','markersize',10); % Plot given
 points
h_unif_int = plot(Runge_x, unif_y_interp, 'b-','linewidth',2); % Plot
 interpolation

% (c) Calcuate and plot interpolation with Chebyshev points
K = 11; k = 1:K;
cheb_x  = cos(pi* (2*k - 1)./(2*K) );
cheb_y_given  = 1./(1 + 25*cheb_x.^2);
cheb_y_interp = lagrange_interp( cheb_x, cheb_y_given, Runge_x);
```

```
h_cheb_pts = plot(cheb_x, cheb_y_given, 'ro','markersize',10); % Plot given
 points
h_cheb_int = plot(Runge_x, cheb_y_interp, 'r-', 'linewidth',2); % Plot
 interpolation

% Make it look nice
set(gca,'Fontsize',14);
legend([h_runge, h_unif_pts, h_unif_int, h_cheb_pts, h_cheb_int], ...
    {'Runge''s Function', 'Uniform Points','Uniform Interpolation','Chebyshev
 Points','Chebyshev Interpolation'},...
    'location','north');
title('Interpolation with Different Grid Points');
```



Published with MATLAB® R2022b