Lab 08: MATLAB Interpolation Routines and Their Derivatives

Melissa Butler

October 11, 2021

1 Figures

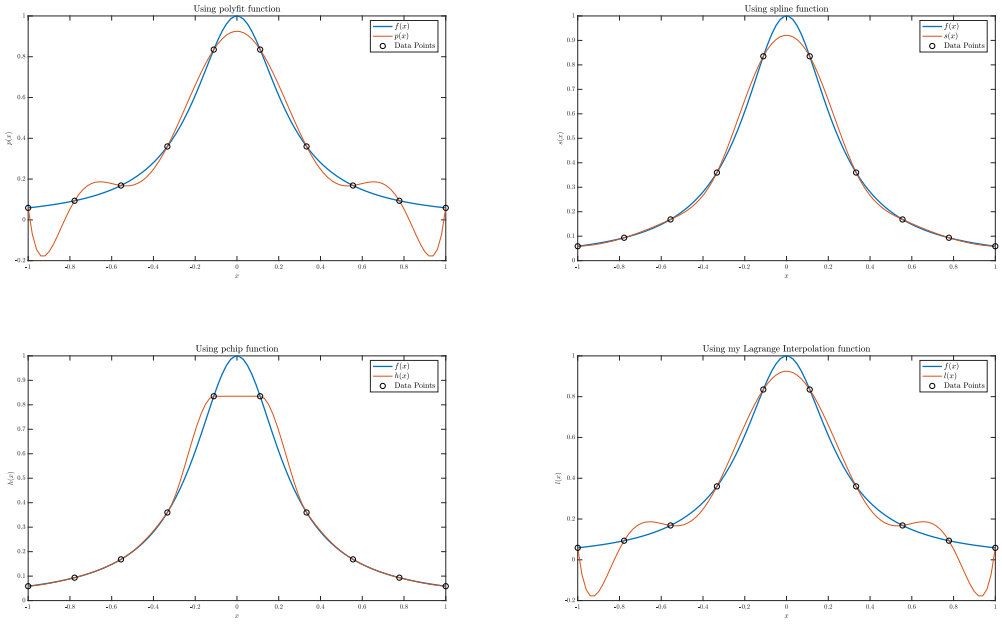

Figure 1: Polynomial Interpolation using different routines

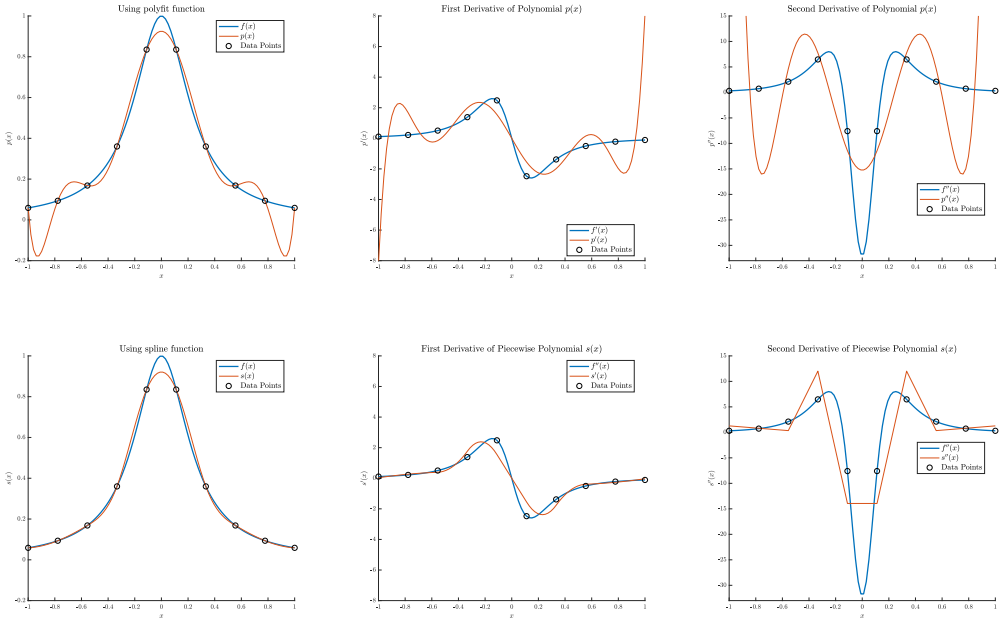

Figure 2: Derivatives of Interpolation Polynomials

2 Script

2.1 Script file: lab_08_script.m

```
% Math 3341, Fall 2021
% Lab 08: MATLAB Interpolation Routines and Their Derivatives
% Author: Melissa Butler
% Date: 10/11/2021

clc; clear; close all;

% Change default text interpreter to LaTeX
set(groot, 'defaultTextInterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex')


%% Homework 5, Problem 1 function
f = @(x) (1 + 16 * x.^2).^(-1);
f1 = @(x) -(1 + 16 * x.^2).^(-2) * 32 .* x;                                  % f'(x)
f2 = @(x) 2 * (1 + 16 * x.^2).^(-3) .* (32 * x).^2 -(1 + 16 * x.^2).^(-2) * 32; % f''(x)

x = linspace(-1, 1, 100);
y = f(x);
y1 = f1(x);      % yvals for f'(x)
y2 = f2(x);      % yvals for f''(x)
n = 9;           % degree of polynomial desired
```

```
N = n+1;          % number of nodes needed

% Generate set of equispaced nodes
xdata = linspace(-1, 1, N);
ydata = f(xdata);
ydata1 = f1(xdata);
ydata2 = f2(xdata);

%% 1 Polynomial Interpolation Routines

% 1(a) Using polyfit and polyval
p_coeff = polyfit(xdata, ydata, n);
p_yvals = polyval(p_coeff, x);

% 1(b) Using spline
sp = spline(xdata, ydata, x);

% 1(c) Using pchip
pc = pchip(xdata, ydata, x);

% 1(d) Using your own Lagrange interpoltation polynomial function
lp = lagrange(xdata, ydata, x);

%% 2 Derivatives of interpolation polynomials

% 2(a) find 1st derivative using polyder
pd1_coeff = polyder(p_coeff);
pd1_yvals = polyval(pd1_coeff, x);

% 2(b) find 2nd derivative using polyder
pd2_coeff = polyder(pd1_coeff);
pd2_yvals = polyval(pd2_coeff, x);

% 2(c) Derivatives of spline

% stores structure of spline
cs_struct = spline(xdata, ydata);
% store the coefficients of the derivative
b = cs_struct.coefs(:,3);
c = cs_struct.coefs(:,2);
d = cs_struct.coefs(:,1);

% find 1st derivative of spline
sp_d1 = cubic_spline_der(b, c, d, xdata, x, 1);

% find 2nd derivative of spline
sp_d2 = cubic_spline_der(b, c, d, xdata, x, 2);

%% 3 Plot interpolation polynomials
figure(1);
% set figure window to full screen
set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);

% plot polyfit interpolation polynomial
```

```
78  subplot(2,2,1);
79  plot(x, y,'LineWidth',2)                        % plot original function
80  hold on
81  plot(x, p_yvals,'-','LineWidth',1)              % plot interp. polynomial
82  plot(xdata, ydata,'ko','MarkerSize',8)          % plot nodes
83  title('Using polyfit function','FontSize',14)
84  xlabel('$x$'); ylabel('$p(x)$');
85  legend({'$f(x)$','$p(x)$','Data Points'},'FontSize',12,'Location','best')
86
87  % plots spline interpolation polynomial
88  subplot(2,2,2);
89  plot(x, y,'LineWidth',2)                        % plot original function
90  hold on
91  plot(x, sp,'-','LineWidth',1)                   % plot spline interp. polynomial
92  plot(xdata, ydata,'ko','MarkerSize',8)          % plot nodes
93  title('Using spline function','FontSize',14)
94  xlabel('$x$'), ylabel('$s(x)$')
95  legend({'$f(x)$','$s(x)$','Data Points'},'FontSize',12,'Location','best')
96
97  % plots pchip interpolation polynomial
98  subplot(2,2,3);
99  plot(x, y,'LineWidth',2)                        % plot original function
100 hold on
101 plot(x, pc,'-','LineWidth',1)                   % plot spline interp. polynomial
102 plot(xdata, ydata,'ko','MarkerSize',8)          % plot nodes
103 title('Using pchip function','FontSize',14)
104 xlabel('$x$'), ylabel('$h(x)$')
105 legend({'$f(x)$','$h(x)$','Data Points'},'FontSize',12,'Location','best')
106
107 % plots your lagrange interpolation polynomial
108 subplot(2,2,4);
109 plot(x, y,'LineWidth',2)                        % plot original function
110 hold on
111 plot(x, lp,'-','LineWidth',1)                   % plot interp. polynomial
112 plot(xdata, ydata,'ko','MarkerSize',8)          % plot nodes
113 title('Using my Lagrange Interpolation function','FontSize',14)
114 xlabel('$x$'), ylabel('$l(x)$')
115 legend({'$f(x)$','$l(x)$','Data Points'},'FontSize',12,'Location','best')
116
117 %% 4 Plot derivatives
118 figure(2);
119 % set figure window to full screen
120 set(gcf, 'Units', 'Normalized', 'OuterPosition', [0 0 1 1]);
121
122 % plot polyfit interpolation polynomial
123 subplot(2,3,1); hold on;
124 plot(x, y,'LineWidth',2)                        % plot original function
125 plot(x, p_yvals,'-','LineWidth',1)              % plot interp. polynomial
126 plot(xdata, ydata,'ko','MarkerSize',8)          % plot nodes
127 axis([-1 1 -0.2 1])
128 title('Using polyfit function','FontSize',14)
129 xlabel('$x$'), ylabel('$p(x)$')
130 legend({'$f(x)$','$p(x)$','Data Points'},'FontSize',12,'Location','best')
131
```

```
% plots 1st derivative of polyfit interpolation polynomial
subplot(2,3,2); hold on;
plot(x, y1,'LineWidth',2)                       % plot original function
plot(x, pd1_yvals,'-','LineWidth',1)            % plot interp. polynomial
plot(xdata, ydata1,'ko','MarkerSize',8)         % plot nodes
axis([-1 1 -8 8])
title('First Derivative of Polynomial $p(x)$','FontSize',14)
xlabel('$x$'), ylabel('$p''(x)$')
legend({'$f''(x)$','$p''(x)$','Data Points'},'FontSize',12,'Location','best')

% plots 2nd derivative of polyfit interpolation polynomial
subplot(2,3,3); hold on;
plot(x, y2,'LineWidth',2)                       % plot original function
plot(x, pd2_yvals,'-','LineWidth',1)            % plot interp. polynomial
plot(xdata, ydata2,'ko','MarkerSize',8)         % plot nodes
axis([-1 1 -33 15])
title('Second Derivative of Polynomial $p(x)$','FontSize',14)
xlabel('$x$'), ylabel('$p''''(x)$')
legend({'$f''''(x)$','$p''''(x)$','Data Points'},'FontSize',12,'Location','best')

% plots spline interpolation polynomial
subplot(2,3,4); hold on;
plot(x, y,'LineWidth',2)                        % plot original function
plot(x, sp,'-','LineWidth',1)                   % plot spline interp. polynomial
plot(xdata, ydata,'ko','MarkerSize',8)          % plot nodes
axis([-1 1 -0.2 1])
title('Using spline function','FontSize',14)
xlabel('$x$'), ylabel('$s(x)$')
legend({'$f(x)$','$s(x)$','Data Points'},'FontSize',12,'Location','best')

% plots 1st derivative of spline interpolation polynomial
subplot(2,3,5); hold on;
plot(x, y1,'LineWidth',2)                       % plot original function
plot(x, sp_d1,'-','LineWidth',1)                % plot spline interp. polynomial
plot(xdata, ydata1,'ko','MarkerSize',8)         % plot nodes
axis([-1 1 -8 8])
title('First Derivative of Piecewise Polynomial $s(x)$','FontSize',14)
xlabel('$x$'), ylabel('$s''(x)$')
legend({'$f''''(x)$','$s''(x)$','Data Points'},'FontSize',12,'Location','best')

% plots 2nd derivative of spline interpolation polynomial
subplot(2,3,6); hold on;
plot(x, y2,'LineWidth',2)                       % plot original function
plot(x, sp_d2,'-','LineWidth',1)                % plot spline interp. polynomial
plot(xdata, ydata2,'ko','MarkerSize',8)         % plot nodes
axis([-1 1 -33 15])
title('Second Derivative of Piecewise Polynomial $s(x)$','FontSize',14)
xlabel('$x$'), ylabel('$s''''(x)$')
legend({'$f''''(x)$','$s''''(x)$','Data Points'},'FontSize',12,'Location','best')

% save plots
for i = 1:2
    fig = figure(i);
    fig.PaperPositionMode = 'auto';
```

```
    pos = fig.PaperPosition;
    fig.PaperSize = [pos(3) pos(4)];
    filename = sprintf('lab_08_figure_0%d.pdf', i);
    print(fig, '-dpdf', filename)
end
```