Lab5Simons.m

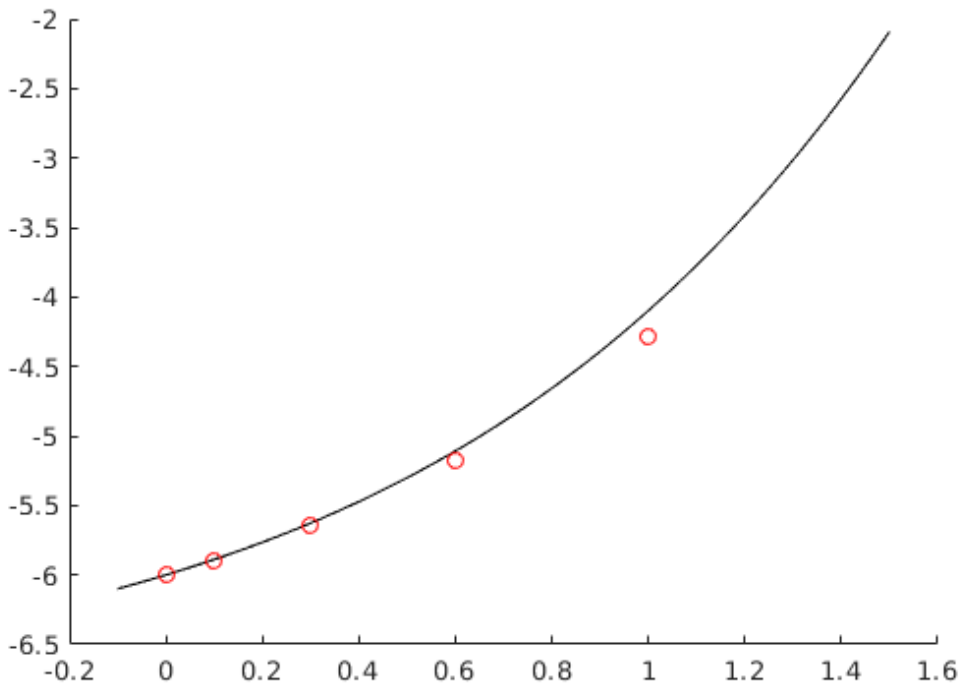
Table of Contents

Madilyn Simons

Problem 1a

Use divided differences to construct the interpolating polynomial

```
% number of given data points
i = 5;

% given x values
xi = [0.0 0.1 0.3 0.6 1.0];

% f(x) for all x
yi = [-6.00000 -5.89483 -5.65014 -5.17788 -4.28172];

% interpolating polynomial coefficients
coeffs = divided_difference(xi, yi, i);

% parse coeffs
a = coeffs(5); b = coeffs(4); c = coeffs(3); d = coeffs(2); e =
 coeffs(1);

% create a linespace and the interpolating polynomial
x = -0.1:0.01:1.5;
polynomial = a*x.^4 + b*x.^3 + c*x.^2 + d*x + e;

% plot the results
figure();
hold on;
plot(x, polynomial, 'k');
scatter(xi, yi, 'r');
hold off
```



Problem 1b

Use the constructing interpolating polynomial to estimate f(0.9)

```
% coefficients
a = -6.0000; b = 1.0517; c = 0.5725; d = 0.2150; e = 0.0630;

% interpolating polynomial
f = @(x) a*x.^4 + b*x.^3 + c*x.^2 + d*x + e;

% print f(0.9)
fprintf('f(0.9) = %.4f\n', f(0.9));
```

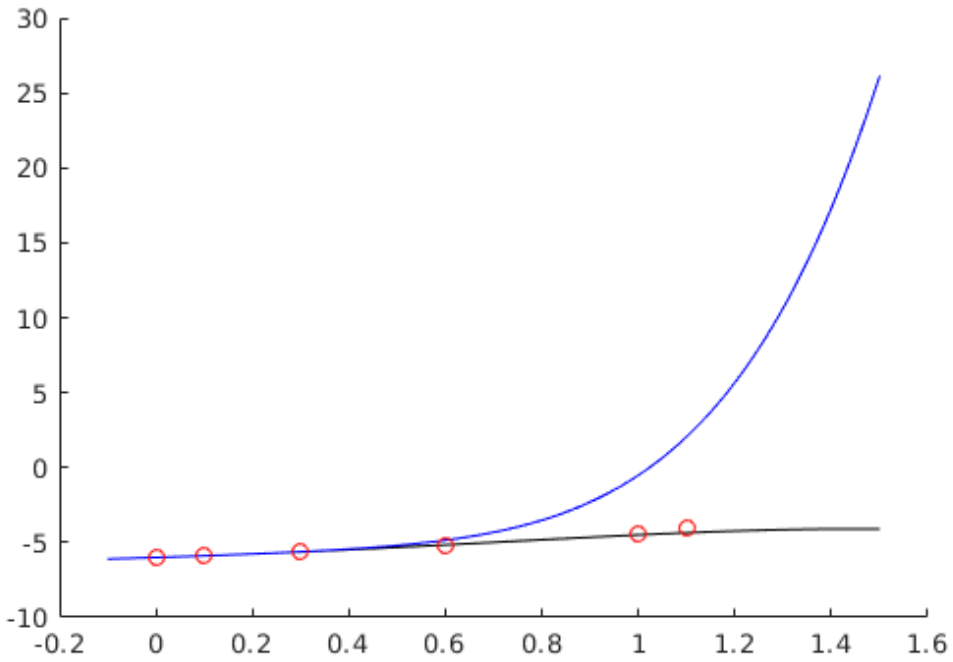
```
f(0.9) = -2.4497
```

Problem 1c

Add another data point to the given set of data to construct another interpolating polynomial. Compare these two interpolating polynomials.

```
% original data set
i0 = 5;
x0 = [0.0 0.1 0.3 0.6 1.0];
y0 = [-6.00000 -5.89483 -5.65014 -5.17788 -4.38172];

% new data set
i1 = 6;
```

```
x1 = [0.0 0.1 0.3 0.6 1.0 1.1];
y1 = [-6.00000 -5.89483 -5.65014 -5.17788 -4.38172 -3.99583];

% coefficients for our two interpolating polynomials
coeffs0 = divided_difference(x0, y0, i0);
coeffs1 = divided_difference(x1, y1, i1);

% coeffs0 parsed
a0 = coeffs0(5); b0 = coeffs0(4); c0 = coeffs0(3);
d0 = coeffs0(2); e0 = coeffs0(1);

% coeffs1 parsed
a1 = coeffs1(6); b1 = coeffs1(5); c1 = coeffs1(4);
d1 = coeffs1(3); e1 = coeffs1(2); f1 = coeffs1(1);

% the new and old polynomials
x = -0.1:0.01:1.5;
polynomial0 = a0*x.^4 + b0*x.^3 + c0*x.^2 + d0*x + e0;
polynomial1 = a1*x.^5 + b1*x.^4 + c1*x.^3 + d1*x.^2 + e1*x + f1;

% plot the results
figure();
hold on
plot(x, polynomial0, 'k');
plot(x, polynomial1, 'b');
scatter(x1, y1, 'r');
hold off
```



Problem 2

Use divided differences to construct an interpolating polynomial that can be used to estimate how far Secretariat ran at a given time

```
% number of data points
i = 5;

% time in minutes
time = [0.0 25.2 49.2 96.4 119.4];

% distance in miles
distance = [0.0 0.25 0.50 1.00 1.25];

% interpolating polynomial coefficients
coeffs = divided_difference(distance, time, i);

% parse coeffs
a = coeffs(5); b = coeffs(4); c = coeffs(3); d = coeffs(2); e =
 coeffs(1);

% interpolating polynomial
f = @(x) a*x.^4 + b*x.^3 + c*x.^2 + d*x + e;

% estimated time and actual time
estimated_time = f(0.75);
actual_time = 73.0;

% percent error of estimated time
error = abs(actual_time - estimated_time) ./ actual_time;
error = error * 100;

% convert estimated time from minutes to hours and minutes
hours = 0;

while estimated_time >= 60
    estimated_time = estimated_time - 60;
    hours = hours + 1;
end

minutes = estimated_time;

% print the results
fprintf('Estimated time:\t%d:%.1f\n', hours, minutes);
fprintf('Actual time:\t1:13.0\n');
fprintf('The estimated value is 1.8 seconds less than the actual value
\n');
fprintf('The estimated value is %.2f%% off\n', error);
```

```
Estimated time: 1:11.2
Actual time: 1:13.0
The estimated value is 1.8 seconds less than the actual value
The estimated value is 2.48% off
```

Problem 3

Find the degree of the polynomial constructed using divided differences and a given set of data

```
% given data x and f(x)
x = [0 1 2 3 4 5 6 7];
y = [0 -2 -8 0 64 250 648 1372];

% calculate coefficients for the interpolating polynomial
coeffs = divided_difference(x, y, 8);

% display coefficients
disp(coeffs);

% print the results
fprintf('The polynomial is x^4 + 3x^3 - 2x^2 - 2x\n');
fprintf('so the degree of the polynomial is 4\n');
```

```
     0    -2    -2     3     1     0     0     0

The polynomial is x^4 + 3x^3 - 2x^2 - 2x
so the degree of the polynomial is 4
```

Published with MATLAB® R2018b