Lab6Simons.m

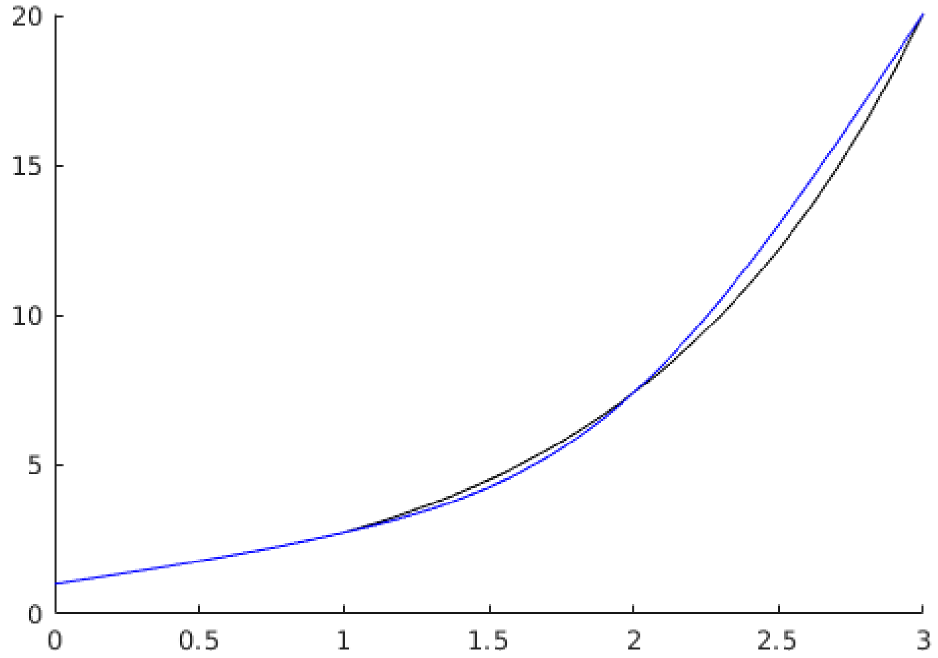
Table of Contents

```
Madilyn Simons
```

Problem 1

```
        Approximate a f(x) = e^x by forming a natural cubic spline
```

```
% number of data points used to form spline
n = 4;

% x-values of data points
x = [0 1 2 3];

% y-values of data points
a = [1 exp(1) exp(2) exp(3)];

% form the natural cubic spline
[a, b, c, d] = natural_cubic_spline(n, x, a);

% the actual value of f(x) = e^x from 1 to 3
x0 = 1:0.1:3;
true_ex = exp(x_axis);

% S(x) from 0 to 1
x1 = 0:0.1:1;
y1 = polynomial(x1, a(1), b(1), c(1), d(1), x(1));

% S(x) from 1 to 2
x2 = 1:0.1:2;
y2 = polynomial(x2, a(2), b(2), c(2), d(2), x(2));

% S(x) from 2 to 3
x3 = 2:0.1:3;
y3 = polynomial(x3, a(3), b(3), c(3), d(3), x(3));

% plot f(x) and S(x)
figure();
hold on
plot(x0, true_ex, 'black');
plot(x1, y1, 'blue');
plot(x2, y2, 'blue');
plot(x3, y3, 'blue');
hold off
```

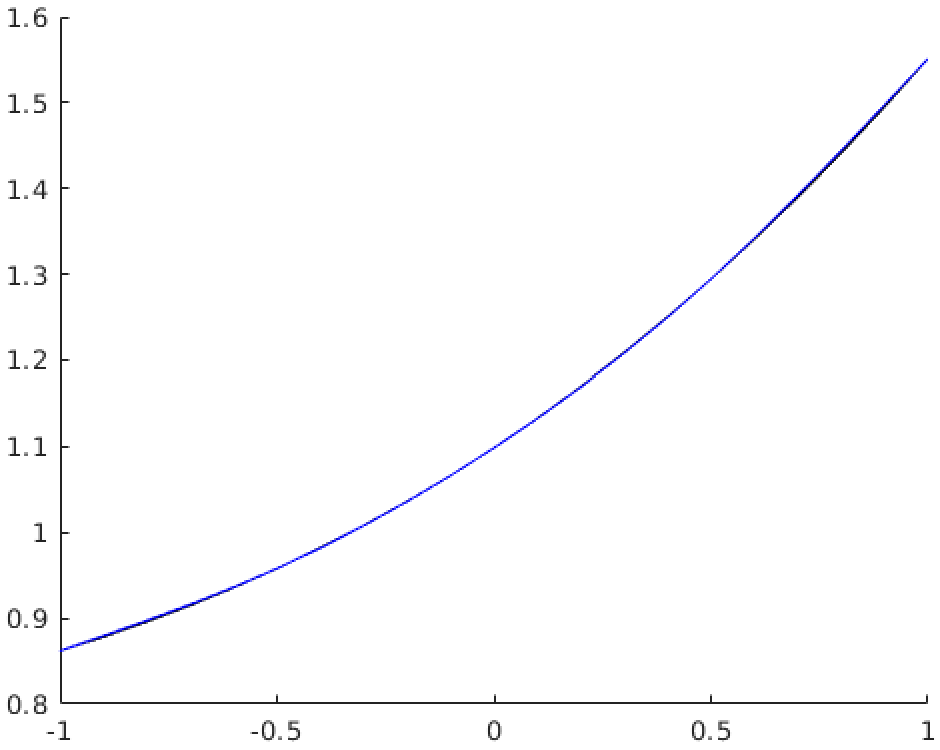


Problem 2

```
        Approximate f(x) = ln(e^x + 2) by forming a natural cubic spline
```

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

% x-values of data
x = [-1.0 -0.5 0.0 0.5 1.0];

% y-values of data
a = [0.8619948 0.9580201 1.0986123 1.2943767 1.5514447];

% form the natural cubic spline
[a, b, c, d] = natural_cubic_spline(n, x, a);

% the actual value of f(x)
x0 = -1:0.1:1;
true_fx = log(exp(x0) + 2);

% S(x) from -1 to -0.5
x1 = -1.0:0.1:-0.5;
y1 = polynomial(x1, a(1), b(1), c(1), d(1), x(1));

% S(x) from -0.5 to 0
x2 = -0.5:0.1:0.0;
y2 = polynomial(x2, a(2), b(2), c(2), d(2), x(2));

% S(x) from 0 to 0.5
```

```
x3 = 0.0:0.1:0.5;
y3 = polynomial(x3, a(3), b(3), c(3), d(3), x(3));

% S(x) from 0.5 to 1.0
x4 = 0.5:0.1:1.0;
y4 = polynomial(x4, a(4), b(4), c(4), d(4), x(4));

% plot f(x) and S(x)
figure();
hold on
plot(x0, true_fx, 'black');
plot(x1, y1, 'blue');
plot(x2, y2, 'blue');
plot(x3, y3, 'blue');
plot(x4, y4, 'blue');
hold off
```



Problem 3

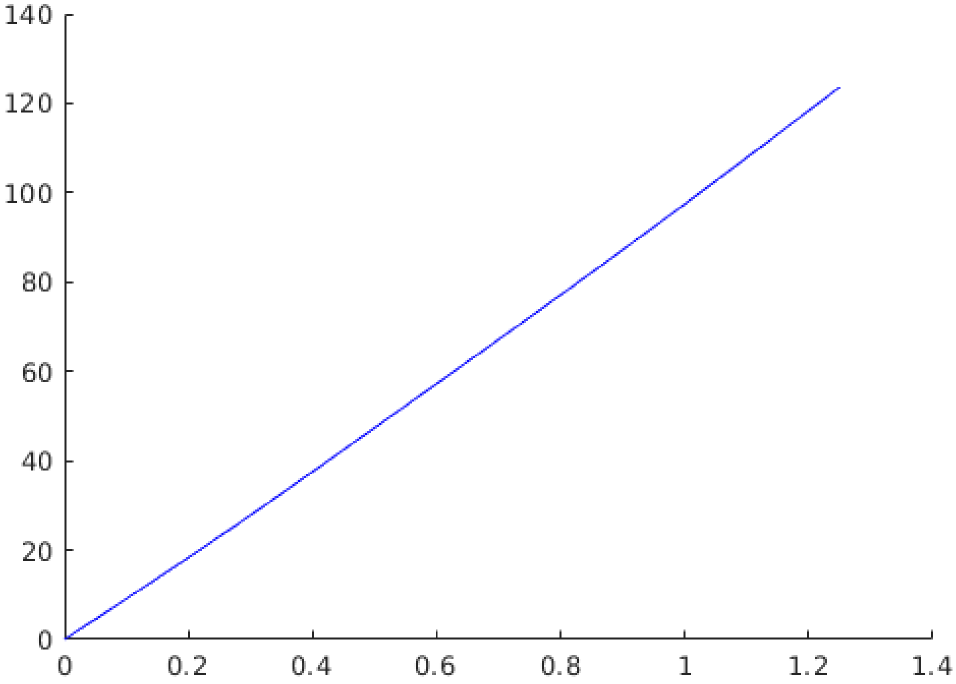
Approximate California Chromes speed throughout the 2014 Kentucky Derby
by forming a natural cubic spline

```
% number of times recorded
n = 5;

% how many miles California Chrome had run at each time
```

```
x = [0.0 0.25 0.5 1.0 1.25];

% how many seconds at each mile marker
a = [0.0 23.04 47.37 97.45 123.66];

% form a natural cubic spline
[a, b, c, d] = natural_cubic_spline(n, x, a);

% time over distance from 0.0 to 0.25 miles
x1 = 0.0:0.01:0.25;
y1 = polynomial(x1, a(1), b(1), c(1), d(1), x(1));

% time over distance from 0.25 to 0.5 miles
x2 = 0.25:0.01:0.5;
y2 = polynomial(x2, a(2), b(2), c(2), d(2), x(2));

% time over distance from 0.5 to 1.0 miles
x3 = 0.5:0.01:1.0;
y3 = polynomial(x3, a(3), b(3), c(3), d(3), x(3));

% time over distance from 1.0 to 1.25 miles
x4 = 1.0:0.01:1.25;
y4 = polynomial(x4, a(4), b(4), c(4), d(4), x(4));

% plot California Chrome's estimated time (s) over distance (mi)
figure();
hold on
plot(x1, y1, 'blue');
plot(x2, y2, 'blue');
plot(x3, y3, 'blue');
plot(x4, y4, 'blue');
hold off

% estimated amount of time passed at the three-quarter-mile mark
% convert seconds to minutes and seconds
time = y3(26);
minutes = 1;
seconds = time - 60;

% print the estimate
fprintf('According to the natural cubic spline, we estimate that\n');
fprintf('California Chromes time at the three-quarter-mile pole was
\n');
fprintf('%d minutes and %.2f seconds.\n\n', minutes, seconds);

fprintf('California Chromes actual time was 1 minute and 11.8 seconds.
\n\n');

fprintf('Our estimate was off by %.2f seconds.\n\n', seconds - 11.8);

% approximate California Chromes speed by finding delta distance over
% delta time at the start  and end of the race
% and then convert this to miles per hour
diff_start = (diff(x1) ./ diff(y1)) * 3600;
```

```
diff_end   = (diff(x4) ./ diff(y4)) * 3600;

% print the results
fprintf('Estimated starting speed:\t %.2f miles per hour\n',
 diff_start(1));
fprintf('Estimated ending speed:\t\t %.2f miles per hour\n',
 diff_end(25));
```

```
According to the natural cubic spline, we estimate that
California Chromes time at the three-quarter-mile pole was
1 minutes and 12.12 seconds.

California Chromes actual time was 1 minute and 11.8 seconds.

Our estimate was off by 0.32 seconds.

Estimated starting speed:  39.62 miles per hour
Estimated ending speed:   34.08 miles per hour
```



Published with MATLAB® R2018b