Table of Contents

Assignment 3 Question 4

```
%Name: Emmanuel Vargas
%Date: 3/1/2021
%store all data points we were given
datx = [1960, 1970, 1980, 1990, 2000, 2010];
daty = [179323, 203302, 226542, 249633, 281422, 308746];
```

Part A

```
%create a lagrange polynomial over the interval 1950 to 2020 in
 increments
%of 0.1 years
x = [1950 : 0.1 : 2020];

xPast = [1950 : .1 : 1959.9];
xCurr = [1960 : .1 : 2010];
xFuture = [2010.1 : .1 : 2020];

[y,L] = Lagrange_poly(transpose(x), datx, daty);
```

Part B

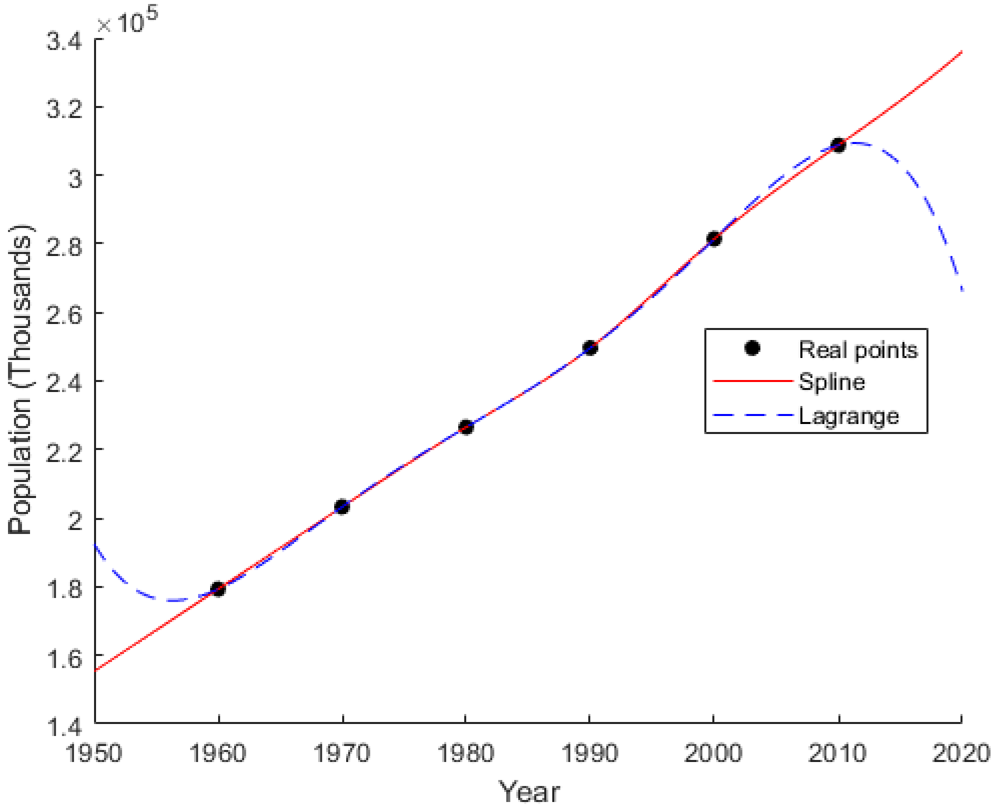
create a cubic spline interpolation from 1950 to 2020 in increments of

```
%%0.1 years
[a,b,c,d]= cubic_spline_coefs(datx, daty);

% find the length of the data points
N = length(datx);

% now create the cubic spline interpolation
spline = []; %start with the first data point to interpolate

%create spline addition for past data points
splineAddition = a(1) + (b(1) .* (xPast - datx(1))) + (c(1) .* (xPast
 - datx(1)).^2) + (d(1) .* (xPast - datx(1)).^3);
spline = [spline, splineAddition];

%create spline addition for current given data points
```

```
%add the first data point to spline
spline = [spline, daty(1)];
for j = 1 : N - 1
   % find the x values between datx[j] and datx[j + 1]
   lower = datx(j);
   upper = datx(j + 1);
   xBetween = [];
   for i = 1 : length(x)
        %grab current x
        currX = x(i);
        %if it falls between upper and lower append to xBetween
        if currX < upper && currX > lower
            xBetween = [xBetween, currX];
        end
   end

   % create the current spline term using xBetween
   splineAddition = a(j) + (b(j) .* (xBetween - datx(j))) + (c(j) .*
 (xBetween - datx(j)).^2) + (d(j) .* (xBetween - datx(j)).^3);

   % add it to the spline as well as the real data point daty[j + 1]
   spline = [spline, splineAddition, daty(j + 1)];
end

%create spline addition for future data points
splineAddition = a(5) + (b(5) .* (xFuture - datx(5))) + (c(5) .*
 (xFuture - datx(5)).^2) + (d(5) .* (xFuture - datx(5)).^3);
spline = [spline, splineAddition];

%disp(size(x))
%disp(size(spline))
```

Part C

```
%graph lagrange, spline, and the real points on the same graph
scatter(datx, daty, 'black' ,'filled');
hold on;
plot(x, spline, 'red');
plot(x, y, '--blue');
hold off;
legend('Real points', 'Spline', 'Lagrange', 'Location', 'Best');
xlabel('Year')
ylabel('Population (Thousands)')
```



Part D

```
%calculate the relative error  as a percentage for 1950 givrn the
 population was 150,697,360
pop1950 = 150697360 / 1000; % divide by 1000 since all other data in
 thousands
lagrangeErr = (abs(pop1950 - y(1)) / pop1950) * 100;
splineErr = (abs(pop1950 - spline(1)) / pop1950) * 100;

%print out the estimates for 1950 along with the relative error for
 each
fprintf('Predictions for year %d\n', x(1));
fprintf('Lagrange: %d thousand people. With relative error: %f
 percent.\n',y(1), lagrangeErr);
fprintf('Spline: %d thousand people. With relative error: %f percent.
\n', spline(1), splineErr);
fprintf('\n');
fprintf('From the errors displayed above we can see that the Spline
 interpolation was much more accurate to the true population.\n');
fprintf('\n');
fprintf('The Spline was more accurate because Splines exhbit more
 linear behavior even when not over the data points, therefore the
 Spline is more accurate when estimatiing points not in the data set.
 The lagrange on the other hand, quickly deviates from the trend after
 the data set ends.');
```

```
Predictions for year 1950
Lagrange: 192539 thousand people. With relative error: 27.765344
 percent.
Spline: 155344 thousand people. With relative error: 3.083425 percent.

From the errors displayed above we can see that the Spline
 interpolation was much more accurate to the true population.

The Spline was more accurate because Splines exhbit more linear
 behavior even when not over the data points, therefore the Spline
 is more accurate when estimatiing points not in the data set. The
 lagrange on the other hand, quickly deviates from the trend after the
 data set ends.
```

Published with MATLAB® R2020a