Table of Contents

Assignment 3

```
%Name: Jim Nguyen
%Date: 3/01/2021
```

Question 4

```
%Show all code and work. I suggest checking out the video on
 Catcourses for using the 'Publish' feature
%in Matlab to ensure everything is included. Consider the census
 population of the United States which is
%taken every 10 years. The following table displays the data from 1960
 to 2010:
```

Question 4 part a

```
%use lagrane poly fxn to create an interpolating polynomial over the
%interval 1950 to 2020 in increments of 0.1 years

    datx = [1960,1970,1980,1990,2000,2010];
    daty = [179323,203302,226542,249633,281422,308746];
    x = [1950:.1:2020]';

    N = length(datx);
    L = ones(length(x),N); % specify the initialization also F
    y = zeros(length(x),1); % specify how y should be initialized

    %creation of Lagrange polynomial
    for k = 1:N
       for i = 1:N
          if i == k
              L = L;
          else
              L(:,k) = L(:,k).*(x - datx(i))/(datx(k) - datx(i));
          end
       end
    end

    %creation of the interpolant
    for k = 1:N
```

```
        y = y + daty(k) * L(:,k);
    end
```

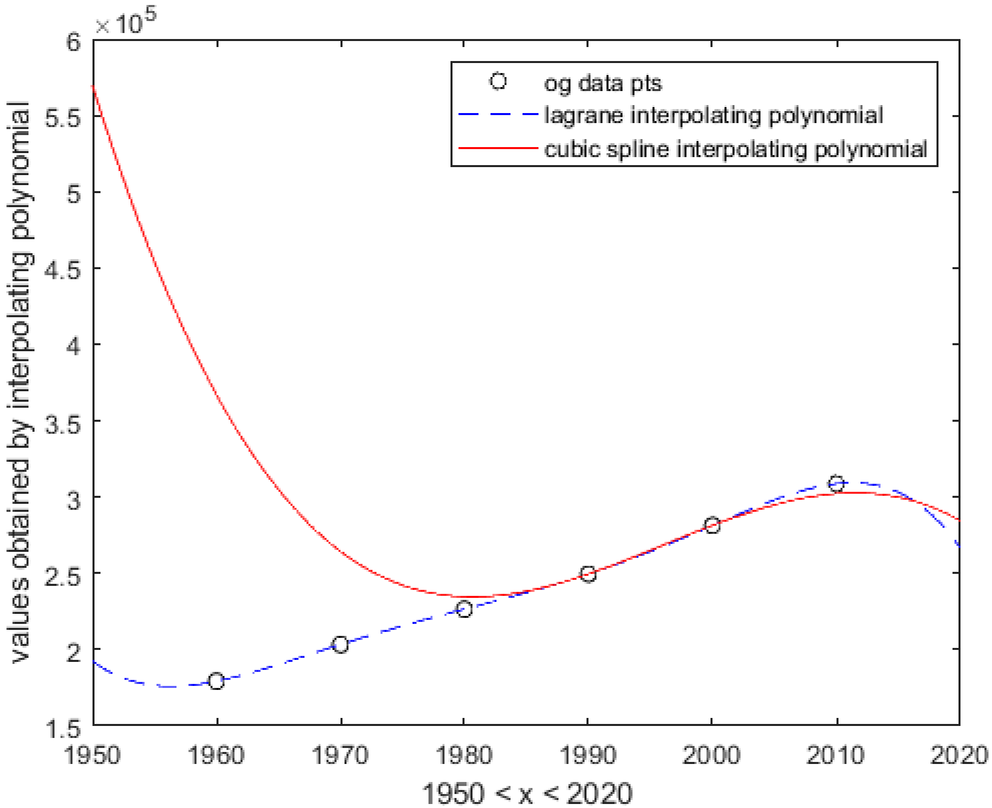
Question 4 part b

```
%calc cubic spline interpolation over interval 1950 to 2020 in
 increments
%of 0.1 years. For years 1950-1960, use interpolating cubic fxn calced
%from 1960-1970 to extrapolate years 1950-1960. for years 2010-2020,
 use
%interpolating cubic fxn from 2000-2010 t extrapolate years 2010-2020

% obtain coefficients for cubic spline
    [a,b,c,d]= cubic_spline_coefs(datx,daty);

    % now create the cubic spline interpolation
    spline=zeros(1,length(x));

    for i = 1:length(x)
        if x(i) >= 0 && x(i) < 0.5
            spline(i) = a(1) + b(1) * (x(i) - datx(1)) + c(1) * (x(i)
 - datx(1))^2 + d(1) * (x(i) - datx(1))^3;
        end
        if x(i) >= 0.5 && x(i) < 2
            spline(i) = a(2) + b(2) * (x(i) - datx(2)) + c(2) * (x(i)
 - datx(2))^2 + d(2) * (x(i) - datx(2))^3;
        end
        if x(i) >= 2 && x(i) < 4
            spline(i) = a(3) + b(3) * (x(i) - datx(3)) + c(3) * (x(i)
 - datx(3))^2 + d(3) * (x(i) - datx(3))^3;
        end
        if x(i) >= 4
            spline(i) = a(4) + b(4) * (x(i) - datx(4)) + c(4) * (x(i)
 - datx(4))^2 + d(4) * (x(i) - datx(4))^3;
        end
    end
```

Question 4 part c

```
%on 1 figure, plot true data pts in black circles, lagrane
 interpolating
%polynomial in blue dashed line, cubic spline interpolating polynomial
 in
%red solid line. Make sure to label axes and include legend

    plot(datx,daty,'ko'); %og data pts
    hold on
    plot(x,y,'--blue');   %lagrane interpolating polynomial
    hold on
    plot(x,spline,'red'); %cubic spline interpolating polynomial
    legend('og data pts', 'lagrane interpolating polynomial','cubic
 spline interpolating polynomial');
    xlabel('1950 < x < 2020');
```

```
ylabel('values obtained by interpolating polynomial');
hold off
```



Question 4 part d

```
%predictions for population in 1950 for 2 interpolants? calc relative
 error
%and comment on which is more accurate to the true 1950 population,
 which
%is 150,697,360. Why do you think one method is more accurate than the
%other?

    %lagrane interpolating polynomial error at 1950 is
    lerr = abs(y(1) - daty(1));

    %cubic spline interpolating polynomial error at 1950 is
    cerr = abs(spline(1) - daty(1));

    %based from these two errors, I think the lagrane interpolating
    %polynomial is more accurate to the true 1950 population because
 it has
    %a much lower error compared to the cublic spline interpolating
    %polynomial. I think its more accurate because we are checking at
 the
    %endpoints of our data. Also cublic spline from what I can
 understand
```

```
   %is used to connect exist points to one other and what we were
trying
   %to find is something that was not given to us, so I think thats
where
   %most of the error for cublic spline comes from.
```

Published with MATLAB® R2020a