Assignment 5 - Polynomial and Spline Interpolation, Least Squares Approximation **LE, Nhat Hung**

COMP 350 - Numerical Computing McGill ID: 260793376

Prof. Chang Xiao-Wen Date: November 12, 2018

Fall 2018 Due date: November 13, 2018

1. (6 points) Find the Vandermonde form, the Lagrange form, and the Newton form of the interpolating polynomial for these data

x	-2	0	1	2
y	2	4	2	2

Vandermonde form:

$$\left[\begin{array}{cccc|c} 1 & -2 & 4 & -8 & 2 \\ 1 & 0 & 0 & 0 & 4 \\ 1 & 1 & 1 & 1 & 2 \\ 1 & 2 & 4 & 8 & 2 \end{array}\right] \sim \left[\begin{array}{cccc|c} 1 & -2 & 4 & -8 & 2 \\ & 2 & -4 & 8 & 2 \\ & 3 & -3 & 9 & 0 \\ & 4 & 0 & 16 & 0 \end{array}\right]$$

$$\sim \left[\begin{array}{cccc|c} 1 & -2 & 4 & -8 & 2 \\ & 2 & -4 & 8 & 2 \\ & & 3 & -3 & -3 \\ & & 8 & 0 & -4 \end{array}\right]$$

$$\sim \left[\begin{array}{cccc|c} 1 & -2 & 4 & -8 & 2 \\ & 2 & -4 & 8 & 2 \\ & & 3 & -3 & -3 \\ & & & 8 & 4 \end{array}\right]$$

$$p(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3$$

$$c_3 = \frac{1}{2}$$
$$c_2 = \frac{-3-(-3)(\frac{1}{2})}{3} = -\frac{1}{2}$$
$$c_1 = \frac{2-(-4)(-\frac{1}{2})-(8)(\frac{1}{2})}{2} = -2$$
$$c_0 = 2-4+2+4 = 4$$

Therefore,

$$p(x) = 4 - 2x - \frac{1}{2}x^2 + \frac{1}{2}x^3$$

Lagrange form:

$$p(x) = l_0(x)y_0 + l_1(x)y_1 + l_2(x)y_2 + l_3(x)y_3$$

$$\begin{aligned}
l_0(x)y_0 &= \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}y_0 = -\frac{1}{6}x + \frac{1}{4}x^2 - \frac{1}{12}x^3 \\
l_1(x)y_1 &= \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}y_1 = 4 - 4x - x^2 + x^3 \\
l_2(x)y_2 &= \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}y_2 = \frac{8}{3}x - \frac{2}{3}x^3 \\
l_3(x)y_3 &= \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}y_3 = -\frac{1}{2}x + \frac{1}{4}x^2 + \frac{1}{4}x^3
\end{aligned}$$

Therefore,

$$p(x) = 4 - 2x - \frac{1}{2}x^2 + \frac{1}{2}x^3$$

Newton form:

$$p(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2)$$

x	y			
-2	2			
0	4	1		
1	2	0	-1	
2	2	0	-1/2	1/2

Therefore,

$$\begin{aligned}
a_0 &= 2 \\
a_1 &= 1 \\
a_2 &= -1 \\
a_3 &= \frac{1}{2}
\end{aligned}$$

and,

$$p(x) = 2 + (x+2) - (x+2)x + \frac{1}{2}(x+2)(x)(x-1) = 4 - 2x - \frac{1}{2}x^2 + \frac{1}{2}x^3$$

2. (Programming by MATLAB) The US census data from 1900 to 2000 are as follows (numbers are in million):

x	1900	1910	1920	1930	1940	1950	1960
y	75.995	91.972	105.711	123.203	131.669	150.697	179.323
	1970	1980	1990	2000			
	203.212	226.505	249.633	281.422			

(a) (8 points) Spline interpolation

i. Find the natural cubic spline function S to interpolate the data.

ii. Find the population estimate for 1985 by the spline function

iii. Find the population estimate for 2010 by the spline function.

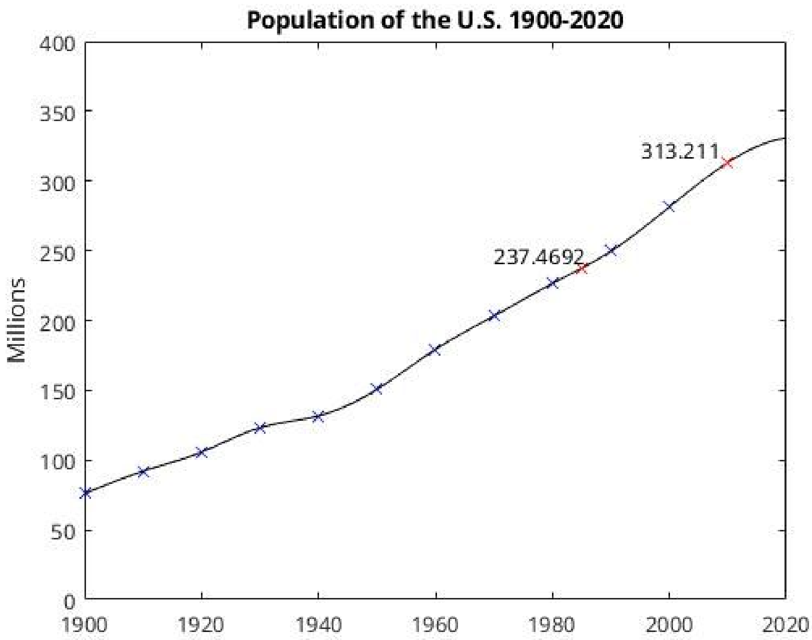

Figure 1: Natural cubic spline interpolation of the US census

ass5.m:

```
% Here is the US Census data from 1900 to 2000.
t = (1900:10:2000)';
y = [75.9950 91.9720 105.7110 123.2030 131.6690 150.6970 ...
     179.3230 203.2120 226.5050 249.6330 281.422]';

z = splineCubicZ(t,y,11);
S = @(x) splineCubic(x,t,y,z,11);

% Plot the given 11 points
plot(t,arrayfun(S,t),'bx');
axis([1900 2020 0 400]);
title('Population of the U.S. 1900-2020');
ylabel('Millions');
```

```
hold on;

% Plot the spline interpolation
x = 1900:1:2020;
plot(x,arrayfun(S,x),'k-');

% Mark the population in 1985
plot(1985,S(1985),'rx');
text(1970,S(1985)+10,num2str(S(1985)));

% Mark the population in 2010
plot(2010,S(2010),'rx');
text(1995,S(2010)+10,num2str(S(2010)));
hold off;
```

splineCubic.m:

```
function S = splineCubic(x,t,y,z,n)

for i = 1:n-1
    if x - t(i+1) <= 0
        break
    end
end

h = t(i+1) - t(i);
B = -h*z(i+1)/6 - h*z(i)/3 + (y(i+1) - y(i))/h;
D = (z(i+1) - z(i)) / (6*h);
S = y(i) + (x-t(i))*(B + (x-t(i))*(z(i)/2 + (x-t(i)) * D));
```

splineCubicZ.m:

```
function Z = splineCubicZ(t,y,n)

for i = 1:n-1
    h(i) = t(i+1) - t(i);
    b(i) = (y(i+1) - y(i)) / h(i);
end

% Forward elimination
u(2) = 2*(h(1)+h(2));
v(2) = 6*(b(2)-b(1));
for i = 3:n-1
    mult = h(i-1) / u(i-1);
    u(i) = 2*(h(i-1) + h(i)) - mult*h(i-1);
```

```
    v(i) = 6*(b(i) - b(i-1)) - mult*v(i-1);
end

% Back substitution
z(n) = 0;
for i = n-1:-1:1
    z(i) = (v(i) - h(i)*z(i+1)) / u(i);
end
z(1) = 0;

Z = z;
```

(b) (6 points) LS approximation

i. Find the straight line which best fits the data in the least-squares sense.

ii. Find the population estimate for 1985 by this straight line.

iii. Find the population estimate for 2010 by this straight line.

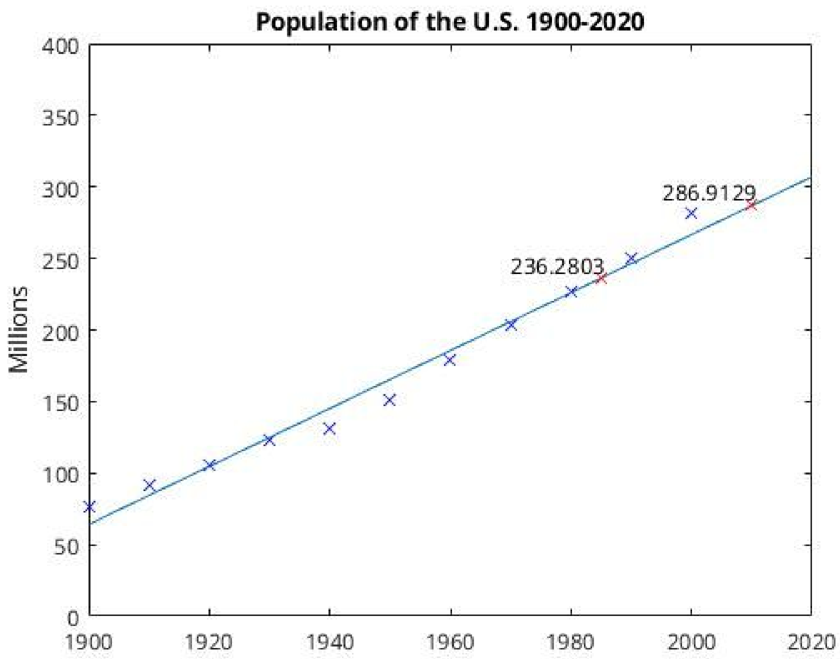

Figure 2: Straight line least squares approximation of the US census

ass5.m:

```
...
LS = @(x) leastSquaresStraightLine(x,t,y,11);

% Plot the least squares approximation
plot(x,LS(x));
```

```
% Mark the population in 1985
plot(1985,LS(1985),'rx');
text(1970,LS(1985)+10,num2str(LS(1985)));

% Mark the population in 2010
plot(2010,LS(2010),'rx');
text(1995,LS(2010)+10,num2str(LS(2010)));
```

leastSquaresStraightLine.m:

```
function y = leastSquaresStraightLine(x,X,Y,m)

a = (m*sum(X.*Y) - sum(X)*sum(Y)) / (m*sum(X.*X) - sum(X)*sum(X));
b = (sum(X.*X)*sum(Y) - sum(X)*sum(X.*Y)) / (m*sum(X.*X) - sum(X)*sum(X));

y = a*x + b;
```