CX 4640 Assignment 7

Wesley Ford

October 16th, 2020

1. Chapter 10, Question 6, on page 325 (on interpolating 4 points with three different methods). Given the four data points $(-1, 1), (0, 1), (1, 2), (2, 0)$, determine the interpolating cubic polynomial

   1. using the monomial basis;
   2. using the Lagrange basis;
   3. using the Newton basis.

   Show that the three representations give the same polynomial.

   1. Using the monomial basis. We are trying to solve

   $$\begin{bmatrix} 1 & x_0 & x_0^2 & x_0^3 \\ 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

   $$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 0 \end{bmatrix}$$

   Using matlab, we can determine that $\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 7/6 \\ 1/2 \\ -2/3 \end{bmatrix}$
   so using the monomial bias our polynomial

   $$y = 1 + \frac{7}{6}x + \frac{1}{2}x^2 - \frac{2}{3}x^3$$

   2. Using the Lagrange basis:

   $$p_3(x) = y_0 * \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + y_1 * \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} + y_2 * \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}$$
   $$+ y_3 * \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}$$
   $$p_3(x) = \frac{(x)(x-1)(x-2)}{(-1)(-2)(-3)} + \frac{(x+1)(x-1)(x-2)}{(1)(-1)(-2)} + 2\frac{(x+1)(x)(x-2)}{(2)(1)(-1)}$$

   This factors to:

   $$p_3(x) = -\frac{x^3 - 3x^2 + 2x}{6} + \frac{x^3 - 2x^2 - x + 2}{2} - (x^3 - x^2 - 2x)$$
   $$p_3(x) = 1 + \frac{7}{6}x + \frac{1}{2}x^2 - \frac{2}{3}x^3$$

   3. Using the Newton basis: $p_3(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)(x - x_2)$ where c_0, c_1, c_2, c_3 are divided differences. Computing the divided differences:

i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
0	1	0	1/2	-2 / 3
1	1	1	-3/2	
2	2	-2		
3	0			

The constants $c_0, \ldots, c_3$ correspond to the following divided differences:

$$c_0 = f[x_0] = 1$$
$$c_1 = f[x_0, x_1] = 0$$
$$c_2 = f[x_0, x_1, x_2] = \frac{1}{2}$$
$$c_3 = f[x_0, x_1, x_2, x_3] = -\frac{2}{3}$$

So our Newton polynomial is:

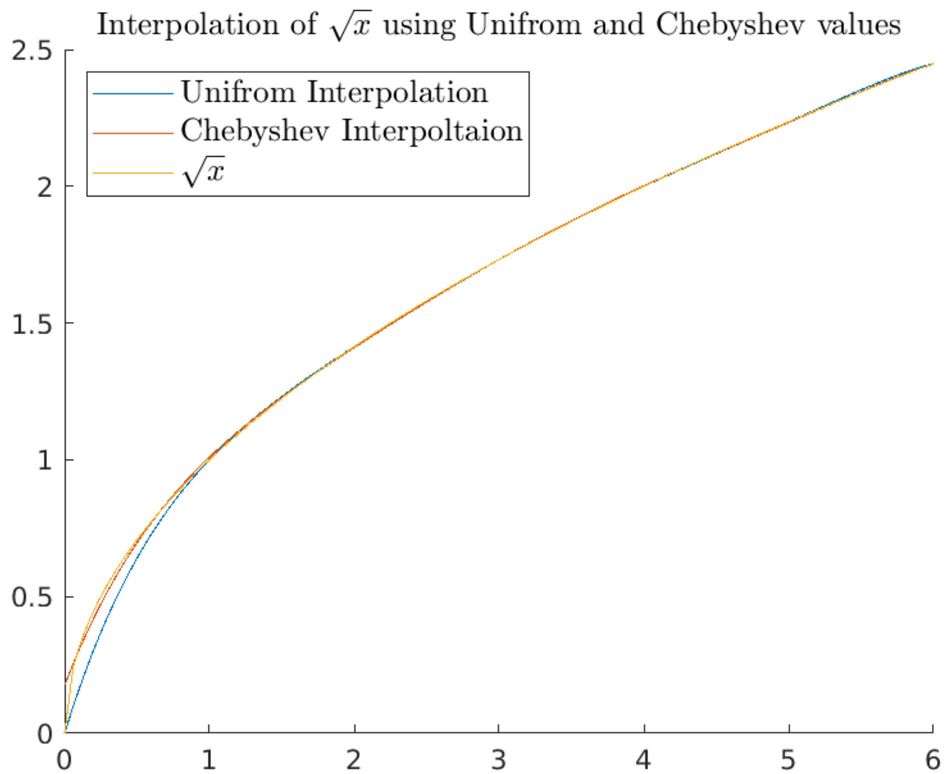
$$p_3(x) = 1 + \frac{1}{2}(x+1)(x) - \frac{2}{3}(x+1)(x)(x-1)$$
$$p_3(x) = 1 + \frac{1}{2}(x^2+x) - \frac{2}{3}(x^3-x)$$

Further simplifying this expression we get that

$$p_3(x) = 1 + \frac{7}{6}x + \frac{1}{2}x^2 - \frac{2}{3}x^3$$

2. Construct a polynomial of degree 6 that interpolates the 7 data points $(i, \sqrt{i}), i = 0, \ldots, 6$. This polynomials is an approximation to the square root function in the interval $[0, 6]$. Now approximate the square root function using 7 Chebyshev points in the same interval. Plot the two polynomials and the exact square root function on the same graph. Please label the plots on your graph.

   The Chebyshev points used in this interpolation are $5.924, 5.345, 4.301, 3, 1.698, 0.654, 0.075$, which are generated using the formula $3(\cos(\frac{2i+1}{2n+2}\pi) + 1)$, $i = 0, \ldots, 6$, $n = 6$.

   We can see that although both interpolation's approximate $\sqrt{x}$ well in the interval $[1, 6]$, using the Chebyshev points for the interpolation increases the accuracy of the approximation in the interval $[0, 1]$.

   Code for problem 2:

```
%Creates two 6 degree polynomail interpolations of sqrt(x) on [0,6], one
%using uniform points on that interval, one using chebyshev points
clear;
clf;
hold on
x = 0:6;
y = sqrt(x);
P1 = polyfit(x,y,6);
%Compute chebyshev points for [-1,.1], then add
%1 and scale by 3 to get nodes for the interval [0,6]

chebyX = cos((2*x + 1) / (2*6+2) * pi);
chebyX = 3 * (chebyX+1);
chebyY = sqrt(chebyX);
P2 = polyfit(chebyX, chebyY, 6);

%Plot both polynomails and sqrt(x) on the interval [0,6]
```

```
x= linspace(0,6);
plot(x, polyval(P1,x));
plot(x, polyval(P2, x));
plot(x, sqrt(x));
l = legend("Unifrom Interpolation", "Chebyshev Interpoltaion", "$\sqrt{x}$");
legend('Location', 'northwest');
t = title('Interpolation of $\sqrt{x}$ using Unifrom and Chebyshev values');
set(l, 'Interpreter', 'latex', 'fontsize',12);
set(t, 'Interpreter', 'latex', 'fontsize',12);
```

3. Consider the three points $(1, 0.1), (2, 0.9), (3, 2)$.

   (a) Find the two *natural* cubic splines that interpolate this data of the form

$$\begin{aligned} p_1(x) &= \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3, \qquad 1 \le x \le 2 \\ p_2(x) &= \beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3, \qquad 2 \le x \le 3 \end{aligned}$$

   A natural spline has the second derivative equal to 0 at the endpoints. We get the following system of equations:

$$\begin{aligned} &p_1(1) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 0.1 \\ &p_1(2) = \alpha_1 + 2\alpha_2 + 4\alpha_3 + 8\alpha_4 = 0.9 \\ &p_1''(1) = 2\alpha_3 + 6\alpha_4 = 0 \\ &p_2(2) = \beta_1 + 2\beta_2 + 4\beta_3 + 8\beta_4 = 0.9 \\ &p_2(3) = \beta_1 + 3\beta_2 + 9\beta_3 + 27\beta_4 = 2 \\ &p_2''(3) = 2\beta_3 + 18\beta_4 = 0 \\ &p_1'(2) = p_2'(2) \\ &p_1''(2) = p_2''(2) \end{aligned}$$

   or

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 0 & 1 & 4 & 12 \\ 0 & 0 & 2 & 6 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.9 \\ z \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ 0 & 1 & 4 & 12 \\ 0 & 0 & 2 & 18 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} 0.9 \\ 2 \\ z \\ 0 \end{bmatrix}$$

   where z is the derivative of both functions at $x = 2$. Using Matlab to help solve this system of equations, we get that

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} -0.7 \\ 0.95 \\ -0.225 \\ 0.075 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -0.85 \\ 0.675 \\ -0.075 \end{bmatrix}$$

   So we get our cubic spline polynomials to be:

$$\begin{aligned} p_1(x) &= -.07 + 0.95x - 0.225x^2 + 0.075x^3 \\ p_2(x) &= 0.5 - 0.85x + 0.675x^2 - 0.075x^3 \end{aligned}$$

   (b) Now find the two *natural* cubic splines using the from

$$\begin{aligned} q_1(x) &= \alpha_1 + \alpha_2(x-1) + \alpha_3(x-1)^2 + \alpha_4(x-1)^3, \qquad 1 \le x \le 2 \\ q_2(x) &= \beta_1 + \beta_2(x-2) + \beta_3(x-2)^2 + \beta_4(x-2)^3, \qquad 2 \le x \le 3 \end{aligned}$$

We start with a similar set of equations as before

$$\begin{aligned}
&1)\ q_1(1) = \alpha_1 = 0.1\\
&2)\ q_1(2) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 0.9\\
&3)\ q_1''(1) = 2\alpha_3 = 0\\
&4)\ q_2(2) = \beta_1 = 0.9\\
&5)\ q_2(3) = \beta_1 + \beta_2 + \beta_3 + \beta_4 = 2\\
&6)\ q_2''(3) = 2\beta_3 + 6\beta_4 = 0\\
&7)\ q_1'(2) = q_2'(2)\\
&8)\ q_1''(2) = q_2''(2)
\end{aligned}$$

Already we know that $\alpha_1 = 0.1$, $\alpha_3 = 0$, and $\beta_1 = 0.9$ from equations 1,3 and 4. We then get

$$\begin{aligned}
&2)\ q_1(2) = \alpha_2 + \alpha_4 = 0.8\\
&5)\ q_2(3) = \beta_2 + \beta_3 + \beta_4 = 1.1\\
&6)\ \beta_3 = -3\beta_4\\
&7)\ \alpha_2 + 3\alpha_4 = \beta_2\\
&8)\ 3\alpha_4 = \beta_3
\end{aligned}$$

From equation 6 and 8, we see that $\alpha_4 = -\beta_4$. So equation 5 becomes

$$5)\ \alpha_2 + 5\alpha_4 = 1.1$$

. Using this with equation 2, we see that $\alpha_4 = 0.075$, and so $\alpha_2 = 0.725$. From this, we can find that $\beta_3 = 0.225$ and $\beta_4 = -0.075$. Finally, $\beta_2 = 0.95$. With these coefficients, our polynomials q_1, q_2 become

$$\begin{aligned}
q_1(x) &= 0.1 + 0.725(x-1) + 0.075(x-1)^3\\
q_2(x) &= 0.9 + 0.95(x-2) + 0.225(x-2)^2 - 0.075(x-2)^3
\end{aligned}$$

(c) Check that your answers in part (a) and part (b) give you the same polynomials.

$$\begin{aligned}
q_1(x) &= 0.1 + 0.725(x-1) + 0.075(x-1)^3\\
q_1(x) &= 0.1 + 0.725(x-1) + 0.075(x^3 - 3x^2 + 3x - 1)\\
q_1(x) &= 0.1 + 0.725x - 0.725 + 0.075x^3 - 0.225x^2 + 0.225x - 0.075\\
q_1(x) &= -0.7 + 0.95x - 0.225x^2 + 0.075x^3\\
q_1(x) &= p_1(x)
\end{aligned}$$

$$\begin{aligned}
q_2(x) &= 0.9 + 0.95(x-2) + 0.225(x-2)^2 - 0.075(x-2)^3\\
q_2(x) &= 0.9 + 0.95(x-2) + 0.225(x^2 - 4x + 4) - 0.075(x^3 - 6x^2 + 12x - 8)\\
q_2(x) &= 0.5 - 0.85x + 0.675x^2 - 0.075x^3\\
q_2(x) &= p_2(x)
\end{aligned}$$

W2hen expanded, both $q_1(x)$ and $q_2(x)$ evaluate to $p_1(x)$ and $p_2(x)$, respectively.

(d) Which of the above two forms do you find easier for hand-calculation?

The second form was much easier to calculate because it vastly simplified some of our equations. The expressions $(x-1)$ and $(x-2)$ evaluated to 0 for $x = 1$ and $x = 2$, which were critical points for our splines. Often times, we got equations of one or two variables instead of equations of 3 or 4 like we did in the first form. A good example of this was that we are able to identify three coefficients in the second from ($\alpha_1, \alpha_3, \beta_1$) without having to do any algebra. The same can not be said for the first form.

```
%Using Matlab's symbolic math tools to help solve the
%set of equations in part A

syms x;
P1 = [1 1 1 1; 1 2 4 8; 0 1 4 12;0 0 2 6];
Y1 = [0.1; 0.9 ; x; 0];
P2 = [1 2 4 8; 1 3 9 27; 0 1 4 12; 0 0 2 18];
Y2 = [0.9; 2; x; 0];

% Solve for each variable in terms of x
a = P1 \ Y1
b = P2 \ Y2
%Sovle for x using the fact that the second derivative at 2 for both
%polynomails must be equal
x = solve(2*a(3) + 12 * a(4) == 2 * b(3) + 12 * b(4))
a_real = subs(a)
b_real = subs(b)
```