Math 104A: Homework 3

Raghav Thirumulu, Perm 3499720
rrajuthirumulu@umail.ucsb.edu

July 19, 2018

1. (a) We know that

$$P_1(x) = \frac{(x - x_1)}{(x_0 - x_1}f(x_0) + \frac{(x - x_1)}{(x_1 - x_0}f(x_1)$$

We can calculate the error of $P_1(x)$ through the following equation:

$$\begin{aligned} |f(x) - P_1(x)| &= |\frac{(x - x_0)(x - x_1)}{2}f''(x)| \\ &= |\frac{(x - x_0)(x - x_1)}{2}||f''(x)| \end{aligned}$$

The maximum value of $(x - x_0)(x - x_1)$ is at $x = \frac{x_0+x_1}{2}$. So

$$|(x - x_0)(x - x_1)| \leq \frac{(x_1 - x_0)^2}{4}$$

Also $|f''(x)| \leq M_2$. So we can now write

$$\begin{aligned} |f(x) - P_1(x)| &\leq \frac{1}{2}\frac{(x_1 - x_0)^2}{4}M_2 \\ &\leq \frac{1}{8}(x_1 - x_0)^2 M_2 \end{aligned}$$

as desired

(b) Given, $f(x) = \sin x$ where $0 \leq x, x_0, x_1 \leq \pi/2$ we can find the second derivative $f''(x) = -\sin x$.

From part (a):

$$\begin{aligned} \|f - P_1\|_\infty &\leq \frac{1}{8}(x_1 - x_0)^2 M_2 \\ \frac{1}{8}(x_1 - x_0)^2|-\sin x_0| &\leq \frac{1}{8}(x_1 - x_0)^2|-\sin x_1| \\ 0 \leq \|f - P_1\|_\infty &\leq \frac{1}{8}x(\pi/2)^2 \\ 0 \leq \|f - P_1\|_\infty &\leq \pi^2/32 \end{aligned}$$

Therefore, the maximum error can be $\pi^2/32$. The actual error can be found through the following calculation:

$$\begin{aligned} Error &= \frac{1}{8}(\pi/2)^2(\sin\frac{\pi}{4}) \\ &= \frac{\pi^2}{32\sqrt{2}} \end{aligned}$$

This error fits the bounds that we calculated.

2. (a) Let

$$w_n(x) = (x - x_0)(x - x_1)...(x - x_{n-1})$$

We can express the polynomial $P_n(x)$ by

$$P_n(x) = P_{n-1}(x) + c_n w_n(x) : c_n \in \mathbb{R}$$

Now let $x = x_i$ for some $i \in 0, 1, ..., n-1$. Therefore

$$w_n(x_i) = 0$$

and

$$P_n(x_i) = P_{n-1}(x_i) = f(x_i)$$

We can obtain c_n from the fact that $P_n(x_n) = f(x_n)$. So

$$f(x_n) = P_{n-1}(x_n) + c_n w_n(x_n)$$
$$c_n = \frac{f(x_n) - P_{n-1}(x_n)}{w_n(x_n)}$$

We also know that $P_0(x) = f(x_0)$. So we obtain

$$P_n(x) = \sum_{k=0}^{n} c_k w_k(x)$$

From the formula for Langrange's interpolating polynomial, we say

$$f[x_0, x_1, ..., x_k] = \frac{f[x_0, x_1, ..., x_k] - f[x_0, x_1, ..., x_{k-1}]}{x_k - x_0}$$

We can now deduce that

$$P_{n+1}(x) = P_n(x) + f[x_0, x_1, ..., x_{n+1}]w_{n+1}(x)$$

and

$$w_{n+1}(x) = (x - x_n)w_n(x)$$

The formula for approximating P_n using Lagrange's form is

$$P_n(x) = \sum_{j=0}^{n} f(x_j) \frac{x - x_k}{\prod_{k=0}^{n}{}_{k \neq j}(x_j - x_k)}$$

The coefficient of each term solves to

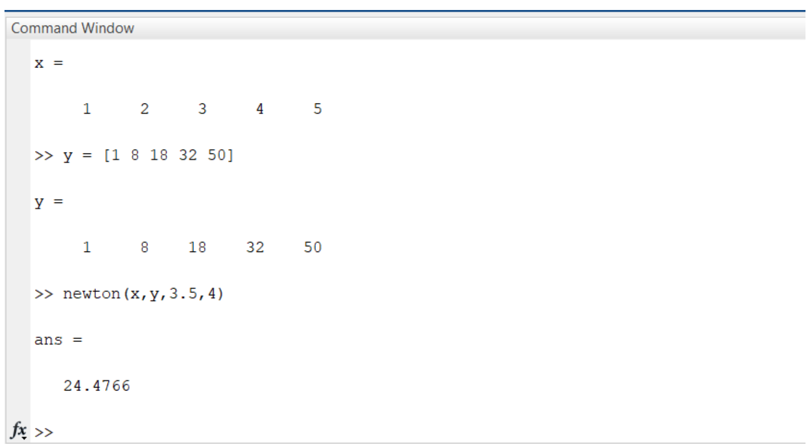
$$f[x_0, x_1, ..., x_n] = \sum_{j=0}^{n} \frac{f(x_j)}{\prod_{k=0}^{n}{}_{k \neq j}(x_j - x_k)}$$

as desired.

(b) We can see from the procedure above that P_k is unaffected by the ordering of the various x values (multiplication is commutative). Therefore, we can conclude that the divivded difference is symmetric in its arguments.

3. (a)

```
function T = newton(x,y,xbar,n)
% Computer code for evaluating the Newton polynomial based on a set of
% given points and degree
% Input:  x     --- Sample x points for evaluating polynomial
%         y     --- Sample y points for evaluating polynomial
%         xbar  --- Point we want to evaluate at
%         n     --- degree of Newton polynomial
%
% Author: Raghav Thirumulu, Perm 3499720
% Date:   07/10/2018

% Create array of n+1 entries for computing values of c
c=zeros(1,n+1);
T=0;

% Iterate through, computing divided difference for each value
% We use n+1 because matrix values begin with 1 in MATLAB
for i=1:(n+1)
    f = 1;
    for j=1:(n+1)
        % We do not want a denominator of 0
        if j~=i
            f = f * (xbar-x(j))/(x(i)-x(j));
        end
    end
    % Sum each term to find the approximation
    T = T + y(i)*f;
end
end
```

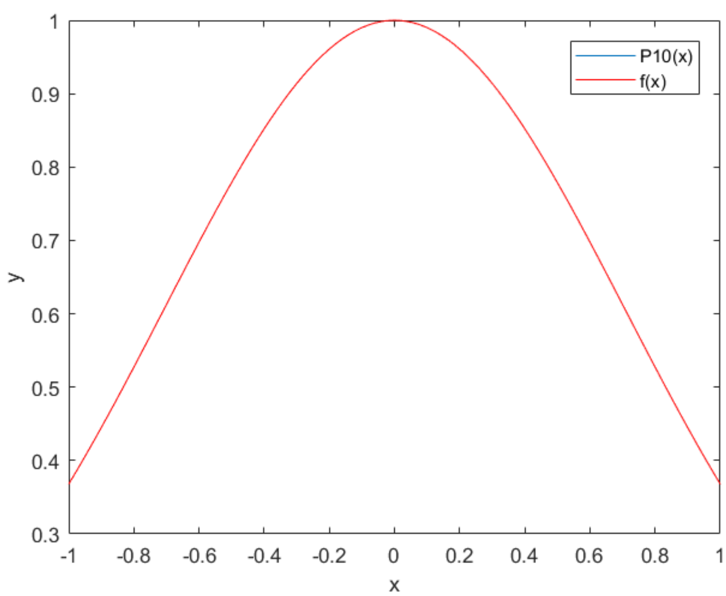


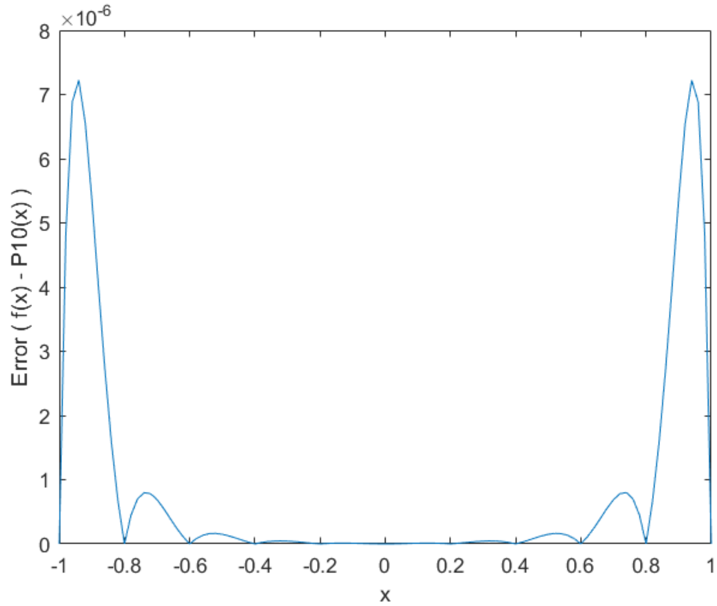
```
Command Window
x =

     1     2     3     4     5

>> y = [1 8 18 32 50]

y =

     1     8    18    32    50

>> newton(x,y,3.5,4)

ans =

   24.4766

fx >>
```

(b)

```
function T = plot_newton_error()
% Computer code for plotting the error of our Newton interpolation
% using equidistributed nodes.
% Input:  None
% Output: Plot of f(x), P10(x), and error
%
% Author: Raghav Thirumulu, Perm 3499720
% Date:   07/11/2018

% Create row vectors for storing values of x and f(x) for evaluating
% P10(x)
x=zeros(1,11);
y=zeros(1,11);

% Use the given function to iterate and store these points
for j=1:11
    x(j)=-1+(j-1)*(2/10);
    y(j)=exp((-1)*((x(j))^2));
end

% Create row vectors because we want to solve P10(x) for a 101 different x
% values using the same equidistributed node equation
```

```
xbar=zeros(1,101);
f=zeros(1,101);
err=zeros(1,101);
for i=1:101
    xbar(i)=-1+(i-1)*(2/100);
    % Evaluate Newton polynomial at each of the 101 different x points and
    % solve for the error at each point using the given equation
    T(i) = newton(x,y,xbar(i),10);
    f(i)=exp((-1)*(xbar(i)^2));
    err(i) = abs(T(i)-f(i));
end

figure(1);
plot(xbar,T); hold on
plot(xbar,f,'r');
xlabel('x');
ylabel('y');
legend('P10(x)','f(x)');
hold off

figure(2);
plot(xbar,err);
xlabel('x');
ylabel('Error ( f(x) - P10(x) )');
end
```





4. Given: $f(0)=0$, $f'(0)=0$, $f(1)=2$, $f'(1)=3$. Let

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$
$$f'(x) = a_1 + 2a_2 x + 3a_3 x^2$$

Plugging in the given conditions into the equations above gives us

$$a_0 = 0$$
$$a_1 = 0$$
$$a_0 + a_1 + a_2 + a_3 = 2$$
$$a_1 + 2a_2 + 3a_3 = 3$$

Through elimination and substitution, we solve for a_2 and a_3.

$$a_2 = -3$$
$$a_3 = 5$$

Plugging back values of a into the equation gives us the interpolating polynomial:

$$f(x) = 5x^3 - 3x^2$$