M 441: Homework 2

Nathan Stouffer

Problem 1

Problem. Use Theorem 2.4 of the notes to prove

$$h - sin(h) = O(h^3) \qquad as\ h \to 0$$

Theorem 2.4 states that for functions $f(h)$ and $g(h)$ that are continuous near $h = 0$ and

$$lim_{h\to 0} \frac{f(h)}{g(h)} = L < \infty$$

then $f = O(g)$ as $h \to 0$.

Answer. To show that $h - sin(h) = O(h^3)$, we will show that $lim_{h\to 0} f(h)/g(h) = L < \infty$ for some $L \in \mathbb{R}$.

$$lim_{h\to 0} \frac{h - sin(h)}{h^3} = \frac{0 - sin(0)}{0^3} = \frac{0}{0}$$

Since the limit is $0/0$, we can apply L'Hospital's rule to say that

$$lim_{h\to 0} \frac{h - sin(h)}{h^3} = \frac{1 - cos(h)}{3h^2} = \frac{1 - cos(0)}{3 * 0^2} = \frac{1 - 1}{0} = \frac{0}{0}$$

By the same reasoning, we apply L'Hospital's rule again

$$lim_{h\to 0} \frac{h - sin(h)}{h^3} = \frac{sin(h)}{6h} = \frac{sin(0)}{6 * 0} = \frac{0}{0}$$

We then apply L'Hospital's rule a final time to say that

$$lim_{h\to 0} \frac{h - sin(h)}{h^3} = \frac{cos(h)}{6} = \frac{cos(0)}{6} = \frac{1}{6}$$

Since $lim_{h\to 0} \frac{h - sin(h)}{h^3} = 1/6 < \infty$, we say that $h - sin(h) = O(h^3)$.

Problem 2

Problem. Using Taylor series of $f(x+h)$ and $f(x-h)$ show

$$f(x+h) - 2f(x) + f(x-h) = f''(x)h^2 + O(h^4)$$

Then, solve for $f''(x)$ to get an approximation for $f''(x)$ and state the order of the truncation error.

Answer. We know the Taylor series expansion of $f(x+h)$ and $f(x-h)$ to be

$$f(x+h) = f(x) + f'(x)h + \frac{1}{2!}f''(x)h^2 + \frac{1}{3!}f^{(3)}(x)h^3 + O(h^4)$$

$$f(x-h) = f(x) - f'(x)h + \frac{1}{2!}f''(x)h^2 - \frac{1}{3!}f^{(3)}(x)h^3 + O(h^4)$$

Adding these equations together, we get

$$f(x+h) + f(x-h) = 2f(x) + 0 + f''(x)h^2 + 0 + O(h^4)$$

which is equivalent to saying that

$$f(x+h) - 2f(x) + f(x-h) = f''(x)h^2 + O(h^2)$$

which we were required to show. From here, we can rearrange the above equation to say that

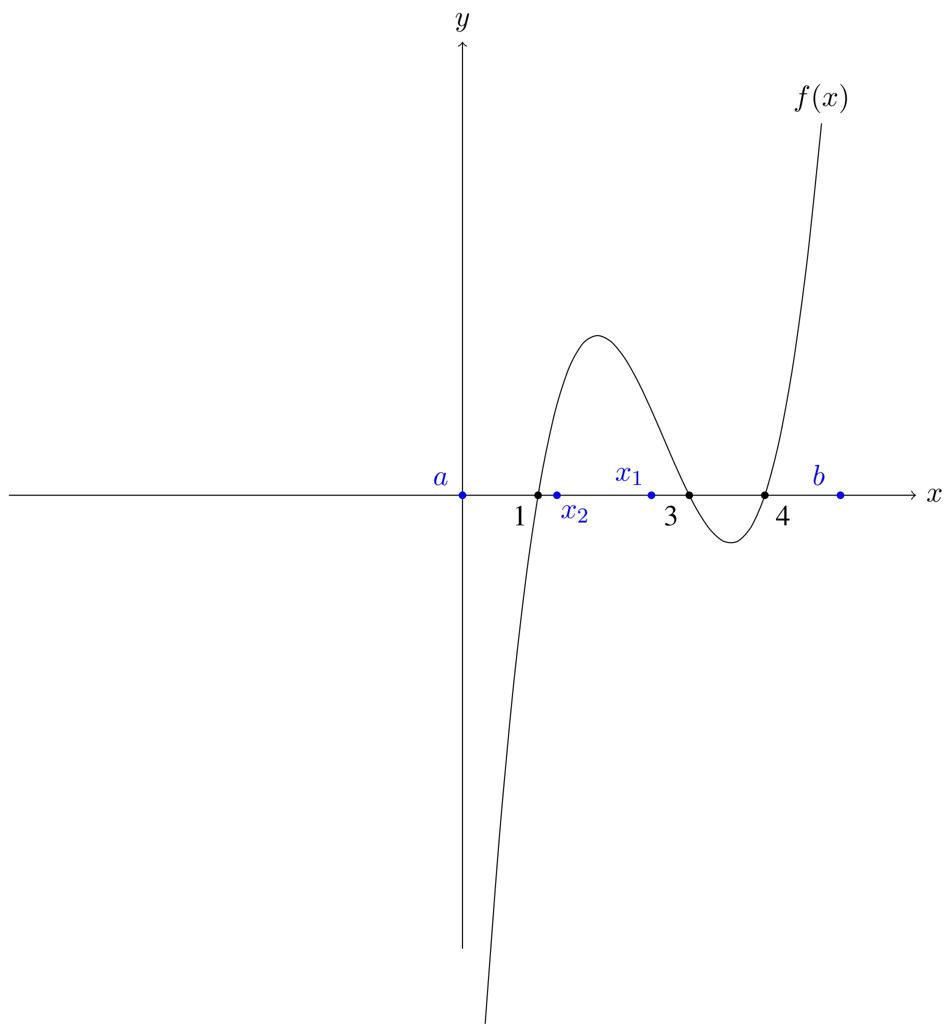
$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \frac{O(h^4)}{h^2} = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2)$$

Thus we can approximate $f''(x) \approx \dfrac{f(x+h) - 2f(x) + f(x-h)}{h^2}$ with $O(h^2)$ error.

Problem 3

Problem. Let $f(x) = (x-1)(x-3)(x-4)$. What root of $f(x)$ does the Bisection Method converge to on the interval $[a, b] = [0, 5]$. Sketch $f(x)$ and label the first two midpoint values x_1, x_2.

Answer. Based on the graph below, we can say that the Bisection Method converges to $x^* = 1$ on the interval $[a, b] = [0, 5]$. We now display the graph of $f(x) = (x-1)(x-3)(x-4)$ and label the first two midpoints x_1 and x_2.



Problem 4

Problem. For the following three problems use modified versions of the posted $Newton.m$, $f.m$, and $df.m$ matlab files to find an approximation of a root of $f(x)$ using the indicated starting guess x_1. For each case, print output in the three columns: $n \;\; x_n \;\; E_n \quad (1 \leq n \leq 10)$. Lastly, state if the convergence rate is linear, superlinear, or quadratic.

a) Newton's Method: $x_1 = 4, f(x) = (x-1)^2 - 2$

b) Newton's Method: $x_1 = 4, f(x) = (x-1)^2$

c) Accelerated Newton's Method: $x_1 = 4, f(x) = (x-1)^2$

Answer.

a) We know $x^* = \sqrt{2} + 1$ to be a simple root of $f(x) = (x-1)^2 - 2$. Therefore, Newton's Method will have quadratic convergence. Here is the output from my Matlab code.

n	x_n	E_n
1	4.000000000000000	1.585786437626905
2	2.833333333333333	0.419119770960238
3	2.462121212121212	0.047907649748117
4	2.414998429894803	0.000784867521708
5	2.414213780047198	0.000000217674103
6	2.414213562373112	0.000000000000017
7	2.414213562373095	0
9	2.414213562373095	0
10	2.414213562373095	0
8	2.414213562373095	0

b) $x^* = 1$ is the only real root of $f(x) = (x-1)^2$ but $x^* = 1$ is not a simple root. So Newton's Method will have linear convergence. Here is the otuput from my Matlab code.

n	x_n	E_n
1	4.000000000000000	3.000000000000000
2	2.500000000000000	1.500000000000000
3	1.750000000000000	0.750000000000000
4	1.375000000000000	0.375000000000000
5	1.187500000000000	0.187500000000000
6	1.093750000000000	0.093750000000000
7	1.046875000000000	0.046875000000000
8	1.023437500000000	0.023437500000000
9	1.011718750000000	0.011718750000000
10	1.005859375000000	0.005859375000000

c) As before, $x^* = 1$ is the only real root of $f(x) = (x-1)^2$ but it is not simple. We will use accelerated Newton's Method with $\lambda = 2$ to obtain quadtratic convergence (since $x^* = 1$ has degree 2). Here is the output from my Matlab code.

n	x_n	E_n
1	4	3
2	1	0
3	NaN	NaN
4	NaN	NaN
5	NaN	NaN
6	NaN	NaN
7	NaN	NaN
8	NaN	NaN
9	NaN	NaN
10	NaN	NaN

Problem 5

Problem. Steffensen method for solving $f(x) = 0$ is defined by:

$$x_{n+1} = x_n - \frac{f(x_n)}{g(x_n)} = x_n - F(x_n)$$

where

$$g(x) = \frac{f(x + f(x)) - f(x)}{f(x)}$$

For simple roots $\bar{x}$ where $f(\bar{x}) \neq 0$, one can show the method has quadratic convergence. Write a Matlab file Steffensen.m whose output is

$$n \qquad x_n \qquad E_n$$

for $1 \leq n \leq 10$ for the case $f(x) = x^2 - 4$ and $x_0 = 1.5$. Include your code for Steffensen.m and output.

Answer. The following table contains the output for my Steffensen.m file with $f(x) = x^2 - 4$ and $x_0 = 1.5$

n	x_n	E_n
1	1.500000000000000	0.500000000000000
2	2.900000000000000	0.900000000000000
3	2.468070519098922	0.468070519098922
4	2.170472770092460	0.170472770092460
5	2.029743067317519	0.029743067317519
6	2.001064655955713	0.001064655955713
7	2.000001414907090	0.000001414907090
8	2.000000000002502	0.000000000002502
9	2.000000000000000	0
10	NaN	NaN

I changed up notation a bit in my Matlab file because I kept every file in the same directory as the Newton.m, f.m, and df.m files. So I changed $f(x)$ in this problem to $g(x)$ and $g(x)$ to $h(x)$. My Matlab code is split between this page and the next.

```
%
%  A "for loop" for Steffensen's method for f(x)=0
%
%   x_{n+1} = x_n + g(x_n)/h(x_n) where h(x) = (g(x + g(x)) - g(x))/g(x)
%
%  1) g.m defines the numerator function
%  2) h.m defines the denominator function
%  3) xexact is the exact root
%  4) m is the number of iterates
%
%  The output matrix is X with n-th row X(n,:)
%  The columns of X are
%
%          X(n,:) = [ n , x(n) , E_n ]
%
%  where E_n is the absolute error.
```

```
%
clear x
clear X
format long
xexact=2;
x(1)=1.5;
X(1,:)=[1,x(1),abs(x(1)-xexact)];
m=9;
for n=1:m
    x(n+1)=x(n)-g(x(n))./h(x(n));
    X(n+1,:)=[n+1,x(n+1),abs(x(n+1)-xexact)];
end;
X
```