HW 2 - 151A

ASHER CHRISTIAN 006-150-286

1. Exercise 1

Use the Theorem on Existence and Uniqueness of Fixed Points from the class notes to show that there exists a unique fixed point for the function $g(x) = \pi + .5\sin(\frac{x}{2})$ *on the interval* $[0, 2\pi]$

Firstly $g(x) \geq 0$ since $sin(\frac{x}{2})$ strictly positive on $0 \leq x \leq 2\pi$ additionally $|.5sin(\frac{x}{2}| \leq 0.5 < \frac{\pi}{2}$ for all x. So by the existence portion of the existence uniqueness theorem $0 \leq g(x) \leq 2\pi$ for all $x \in [0, 2\pi]$ so there exists p such that $g(p) = p$. Additionally $|g'(x)| \leq 0.5$ for all x so the fixed point is unique by the uniqueness theorem.

2. Exercise 2

We perform fixed point iterations on two functions in this problem

$$g_1(x) = (3 + x - 2x^2)^{\frac{1}{4}}, \quad g_2(x) = (\frac{x + 3 - x^4}{2})^{\frac{1}{2}}.$$

Using fixed point iteration $p_{n+1} = g(p_n)$ *with* $p_0 = 1$

1. *show that the fixed points of both* g_1 *and* g_2 *are roots of* $f(x) = x^4 + 2x^2 - x - 3$
   For $g_1(x)$ a fixed point implies

$$\begin{aligned} x &= (3 + x - 2x^2)^{\frac{1}{4}} \\ x^4 &= 3 + x - 2x^2 \\ x^4 + 2x^2 - x - 3 &= 0 \end{aligned}$$

   Thus x is a zero of $f(x)$ For $g_2(x)$ similarly

$$\begin{aligned} x &= (\frac{x + 3 - x^4}{2})^{\frac{1}{2}} \\ x^2 &= \frac{x + 3 - x^4}{2} \\ 2x^2 &= x + 3 - x^4 \\ x^4 + 2x^2 - x - 3 &= 0 \end{aligned}$$

   again a root of $f(x)$
2. *Perform 4 iterations on both of the functions. Report the result of each iteration with 3 digits after decimal point (No rounding)*

Date: 23.01.25.

For g_1

$$\begin{aligned}
p_0 &= 1\\
p_1 &= g_1(p_0) = 1.189\\
p_2 &= g_1(p_1) = 1.080\\
p_3 &= g_1(p_2) = 1.150\\
p_4 &= g_1(p_3) = 1.109
\end{aligned}$$

For g_2

$$\begin{aligned}
p_0 &= 1\\
p_1 &= g_2(p_0) = 1.225\\
p_2 &= g_2(p_1) = 1.046\\
p_3 &= g_2(p_2) = 1.166\\
p_4 &= g_2(p_3) = 1.096
\end{aligned}$$

3. *Which function provides a better approx*
   g_1 provides a better approximation of absolute error 0.016 vs 0.028 for g_2

3. Exercise 3

1. *Let $f(x) = -x^3 - cos(x)$ and $p_0 = -1$. Use Netwon's method to find p_2 what about $p_0 = 0$?* Note that

$$f'(x) = -3x^2 + sin(x).$$

For the first case

$$\begin{aligned}
p_0 &= -1\\
p_1 &= p_0 - \frac{f(p_0)}{f'(p_0)} = -0.880\\
p_2 &= p_1 - \frac{f(p_1)}{f'(p_1)} = -0.866
\end{aligned}$$

In the second case $f'(0) = 0$ so computing the second iteration of newton's method is impossible.

2. *Same function, $p_0 = -1, p_1 = 0$ use secant method to find p_2*

$$p_2 = p_1 - \frac{f(p_1)(p_1 - p_0)}{f(p_1) - f(p_0)} = 0 - \frac{-1(0-1)}{-1 - f(-1)} = -0.675.$$

4. Exercise 4

given $f(x)$ and an iterate p_n find the root of $L(x)$ where L is the tangent line of $f(x)$ at the point $x = p_n$

1. *Write down the equation that $L(x)$ satisfies.*
2. *From your answer find the root*

A line is uniquely determined by a point and a slope The point is $(p_n, f(p_n))$ and the slope is $f'(p_n)$ The equation in point slope form is

$$L(x) = (x - p_n)f'(p_n) + f(p_n).$$

Solving for the zero we get

$$0 = (x - p_n)f'(p_n) + f(p_n) \to -f(p_n) = (x - p_n)f'(p_n).$$

$$-\frac{f(p_n)}{f'(p_n)} = x - p_n \to x = p_n - \frac{f(p_n)}{f'(p_n)}.$$

This is the exact formula from Newtons method

5. Exercise 5

Let $p \in [a, b]$ be the root of $f \in C^1([a, b])$. Assuming that for some $p_0 \in [a, b]$ we have $f'(p) \neq f'(p_0)$ And $|f'(p_0) - f'(p)| < |f'(p_0)|$. Consider an iteration scheme similar to Newton's Method:

$$p_{n+1} = p_n - \frac{f(p_n)}{f'(p_0)}, \quad n \geq 0.$$

Assume that $\lim_{n\to\infty} p_n = p$ *show that the convergence is linear*
Consider the taylor expansion around p

$$f(p_n) = f(p) + f'(p)(p_n - p) + o(p) = f'(p)(p_n - p) + o(p).$$

so our term

$$|p_{n+1} - p| = p_n - p - \frac{f'(p)(p_n - p) + o(p_n)}{f'(p_0)}.$$

$$= (p_n - p)(1 - \frac{f'(p)}{f'(p_0)}) + o(p_n).$$

Let

$$C = (1 - \frac{f'(p)}{f'(p_0)}).$$

clearly

$$\frac{1}{2} < \frac{f'(p)}{f'(p_0)} < 2.$$

so

$$|C| < 1.$$

and so

$$\frac{|p_{n+1} - p|}{|p_n - p|} \approx C < 1.$$

Proving linear convergence

6. Exercise 6

$$f(x) = x + cos(x) \;\; f(p) = 0 \;\; p \approx -0.7390851332.$$

1. *let $p_0 = -5$ and $p_1 = 5$. Compute the approx solution using (1) Newton's method. (2) the Secant method, and (3) the modified Newton's method defined in problem [5]. For (1) and (3) just use p_0 = -5. Stop when $|f(x)| < 10^{-10}$ or $|f(x)| > 10^5$ or # of iterations = 100 Report the following:*
   - *The number of iterations needed by each method to achieve the desired tolerance or the result of divergence.*
   - *The final approx solution p_n with 10 digits after decimal (No rounding between iterations). Which method converges faster?*
2. *Repeat the experiment using $p_0 = -0.9$ and $p_1 = 5$. Explain your result*

```
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;


double EPSILON = pow(10,-10);
double DELTA = pow(10, 5);
double MAX_ITERATIONS = 100;

double newtons_method(double p0, double (*f)(double),
    double (*fp)(double));
double secant_method(double p0, double p1, double(*f)(
    double));
double mod_newton_method(double p0, double (*f)(double),
    double fp0);
double f(double x);
double fp(double x);

int main(void){
    double s0 = -5;
    double s1 = 5;
    double e0 = -0.9;
    double e1 = 5;
    double fs0 = fp(s0);
    double fe0 = fp(e0);
    cout << setprecision(16);
    newtons_method(s0,f,fp);
    secant_method(s0,s1,f);
    mod_newton_method(s0,f,fs0);
    cout << " --- Now with Modified starting --- " << endl;
    newtons_method(e0,f,fp);
    secant_method(e0,e1,f);
    mod_newton_method(e0,f,fe0);
}

double f(double x){
    return x + cos(x);
}

double fp(double x){
    return 1 - sin(x);
}

double newtons_method(double p0, double (*f)(double),
    double (*fp)(double))
{
    double p = p0;
    int iterations = 0;
    while(abs(f(p)) >= EPSILON && abs(f(p)) <= DELTA &&
        iterations < 100){
        p = p - (f(p)/fp(p));
```

```
        iterations += 1;
    }
    cout << "Newton's Method starting at p0 = " << p0 << ",
         iterations: " << iterations;
    cout << " p estimate: " << p<< endl << endl;
    return p;
}
double secant_method(double p0, double p1, double(*f)(
    double)){
    double savedp0 = p0;
    double savedp1 = p1;
    double placeholder;
    int iterations = 0;
    while(abs(f(p1)) >= EPSILON && abs(f(p1)) <= DELTA &&
        iterations < 100){
        placeholder = p1;
        p1 = p1 - ((f(p1)*(p1-p0))/(f(p1)-f(p0)));
        p0 = placeholder;
        iterations += 1;
    }
    cout << "Secant Method starting at p0 = " << savedp0 <<
         ", p1 = " << savedp1;
    cout << ", iterations: " << iterations;
    cout << " p estimate: " << p1 << endl << endl;
    return p1;

}
double mod_newton_method(double p0, double (*f)(double),
    double fp0){
    double p = p0;
    int iterations = 0;
    while(abs(f(p)) >= EPSILON && abs(f(p)) <= DELTA &&
        iterations < 100){
        p = p - (f(p)/fp0);
        iterations += 1;
    }
    cout << "Modified Newton's Method starting at p0 = " <<
         p0 << ", iterations: " << iterations;
    cout << " f'(p0) = " << fp0;
    cout << " p estimate: " << p << endl << endl;
    return p;
}
```

OUTPUT: Newton's Method starting at p0 = -5, iterations: 21 p estimate: -0.7390851332151607

Secant Method starting at p0 = -5, p1 = 5, iterations: 7 p estimate: -0.7390851332151339

Modified Newton's Method starting at p0 = -5, iterations: 4 f'(p0) = 0.04107572533686155
p estimate: -1383975.954119681

— Now with Modified starting —
Newton's Method starting at p0 = -0.9, iterations: 3 p estimate: -0.7390851332208545

Secant Method starting at p0 = -0.9, p1 = 5, iterations: 6 p estimate: -0.7390851332143527

Modified Newton's Method starting at p0 = -0.9, iterations: 8 f'(p0) = 1.783326909627483 p estimate: -0.7390851332309241

Modified Newton's Method on the first pair of starting points did not converge. In the second starting pair, every method converged and the methods that converged in both cases converged much faster with the second starting coordinates. Rate of convergence was very slow in the first part which implies that the starting points were too far to employ the theorem that implies quadratic rate of convergence.