# Math 105AL – Lab 3

Haoyu Zhang, 25327328

## 1 Objective

In this project, I will do the experiment with three root-finding algorithms, which are Bisection, Fixed-Point Iteration, and Newton's Method, by implementing the "my_newtons.m" function. Then I can solve f(x) = x – 0.8 – 0.2sinx on [0, pi/2]. Plotting abs(f(p_n)) per iteration for each method, and finally study Newton's convergence on f(x) = x^2 -10cosx from several initial guesses.

## 2 Procedure

1. Implement the "my_newtons.m" function uses p_n+1 = p_n – f(p_n)/f'(p_n) with stopping criterion abs(p_n+1 – p_n) < tolerance.
2. Bisection stops when abs(f(c_n) < tol or (b-a)/2 <= tol.
3. Fixed-Point Iteration uses g(x) = 0.8 +0.2sinx

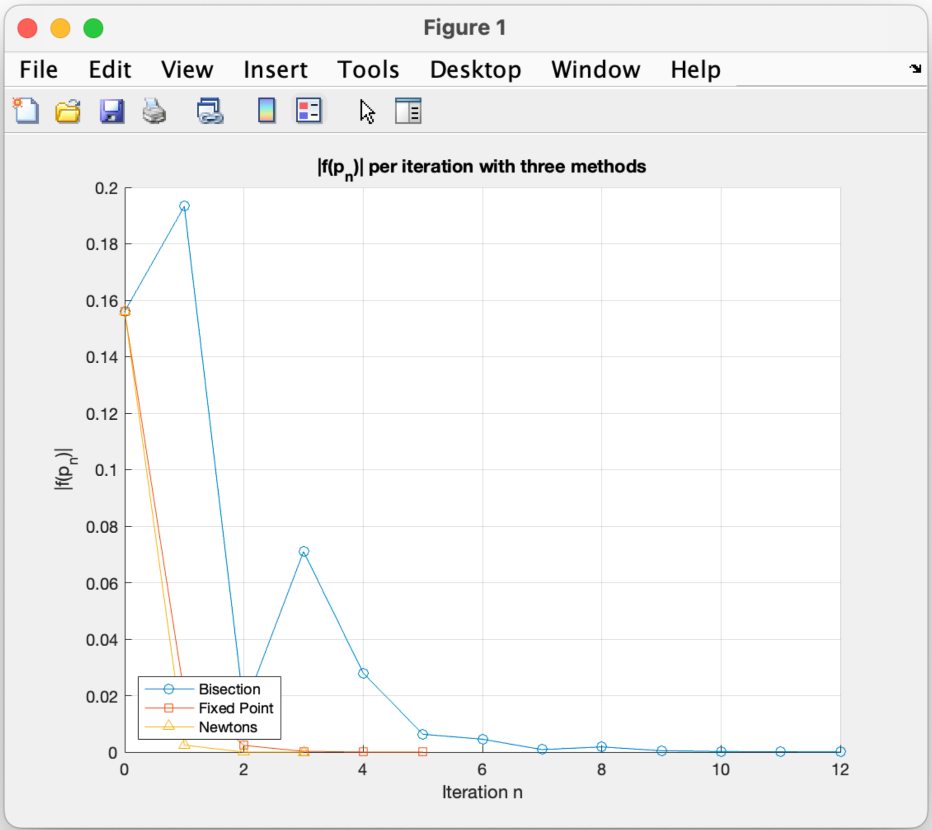
## 3 Results

For question 1, the "my-newtons.m" function correctly converges when f'(p_n) is not equal to 0, and the iterate is in the basin of attraction. It uses abs(p_n+1 – p_n) as the stopping test.

About the question 2, I used my previous codes from the lab 2. I can see that all three methods converge to the same root (p^* approximately equals to 0.9643339). Here are the data results from three different methods.

```
root_newton = my_newtons(f, df, p0, tol, 50);
fprintf('(c) Newton: root ≈ %.7f,  f(root)=%.2e\n', root_newton, f(root_newton));
Converged in 13 iterations.
(a) Bisection: root ≈ 0.9642987,  f(root)=-3.12e-05,  iters=13
Converged in 5 iterations.
(b) Fixed point: root ≈ 0.9643299,  residual g(p)-p=3.49e-06,  iters=5
Converged after 3 iterations.
(c) Newton: root ≈ 0.9643339,  f(root)=4.26e-14
```

Here is the Plotted figure for Q3:




In the Q3, the iterations are to converge. (a). The bisection method has approximately 13-14 iterations, the fixed-point one has 5-7 iterations, and Newton's one has 3-4 iterations from p_0 = pi/4; (b). Bisection method: abs(f(c_n)) < tol or interval half-width <= tol. The Fixed-Point & Newton's Methods: abs(p_n+1 – p_n) < tol. (c). Newton's one is the fastest, and the fixed – point method is linear with factor approximately equaled to 0.2 (g'). However, bisection method is linear with factor of 1/2 and it is the slowest one.

For Q4, Here is the result with multiple different p_0 values.

```
p0=-26  ->  root ≈ -1.379364594   iters=12   |f(root)|=3.95e-11
p0=-25  ->  root ≈ 1.379364594   iters= 7   |f(root)|=2.85e-11
p0=-24  ->  root ≈ 1.379364594   iters=11   |f(root)|=1.92e-12
p0= 24  ->  root ≈ -1.379364594   iters=11   |f(root)|=1.92e-12
p0= 25  ->  root ≈ -1.379364594   iters= 7   |f(root)|=2.85e-11
p0= 26  ->  root ≈ 1.379364594   iters=12   |f(root)|=3.95e-11
```

In part (b), it is not true that Newton's method always converges to the root closest to the initial value. I think the results above give the counterexamples, like p_0 = 24, 25. They both converge to the negative root.

## 4 Conclusion

For x - 0.8 - 0.2sinx, all methods agree on p* approximately equal to 0..9643339. Newton's Method requires the fewest steps, the Fixed-Point is faster due to a small constant, and the bisection one is the slowest one.

For x^2 - 10cosx, Newton's Method converges quickly from all tested p_0 but not to the nearest root in general.

## 5 Challenges and Bugs

1. I forget to open the previous codes, "bisection method" and "fixed point method" functions from Lab 2 zip file so that MATLAB gives me the warning that unrecognized functions and variable.
2. Plot the abs(f(p_n)) for fixed-point instead of the abs(g(p_n) – p_n). So all three curves are comparable.