LAB # 04

02/04/2025

---

1 Introduction

This lab focuses on two different techniques for increasing the convergence rate of linearly convergent root finding methods. The techniques explored in lab are *extrapolation* methods. Through this lab, I got practice developing low-level code and investigating order of convergence.

2 Pre-Lab

In order to prepare for this lab, a new fixed point iteration sub-routine was developed. This version of the FPI routine returns a vector whose entries are the approximations of the fixed point at each iteration. This new sub-routine can be found in the GitHub under the **Lab 04.py** file.

2.1 Review Order

From Homework 3 and class, we are given the following definition outlining order of convergence.

Definition 1. *Suppose $p_n{}_{n=0}^{\infty}$ is a sequence that converges to p with $p_n \neq p$ for all n. If there exists positive constants λ and α such that*

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^\alpha} < \lambda$$

then $p_n{}_{n=1}^{\infty}$ converges to p with an order α and asymptotic error constant λ. if $\alpha = 1$ and $\lambda 1$ then the sequence converges linearly. If $\alpha = 2$, the sequence is quadratically convergent.

2.2 Exercises

2.2.1

For this question, we are given the fixed point p and the vector $\vec{\mathbf{p}}$ of approximations made by an iteration and asked to numerically determine the order of convergence of the algorithm that created the algorithm. To numerically solve for α, we can reconfigure the given definition to obtain the following.

$$\alpha \approx \frac{\log|\frac{p_{n+1}-p_n}{p_n-p_{n-1}}|}{\log|\frac{p_n-p_{n-1}}{p_{n-1}-p_{n-2}}|}$$

2.2.2

For this question, we are given a function $g(x)$ and its fixed point as seen below.

$$g(x) = (\frac{10}{x+4})^{1/2}, \text{ where } p = 1.3652300134140976$$

For Part (a), we are asked to find the number of iterations it takes to converge to an absolute tolerance of 10^{-10} if $p_0 = 1.5$. It was found that it took 12 iterations to converge to an absolute tolerance of 10^{-10}.

For Part (b), we are asked to find the order of convergence for the given function and parameters. The developed code produced an alpha value of $\alpha = 1.000000023880272$.

3 Lab Day: Order of Convergence and Low to High Order Approximations

During lab, we investigated and explored **Aitken's Δ^2 Technique** which is built from a linearly convergent sequence of approximations and **Steffenson's Method** which is a hybrid of a fixed point iteration and Aitken's method.

3.1 Aitken's Δ^2 Acceleration Technique

Aitken's method requires one to begin with a sequence $\{p_n\}_{n=1}^{\infty}$ that converges linearly to a value p. This sequence is then used to create a new sequence given by the following.

$$\hat{p}_n = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}$$

Theorem 1. *Supppose that $\{p_n\}_{n=1}^{\infty}$ is a sequence that converges linearly to the limit p and that*

$$\lim_{n\to\infty} \frac{p_{n+1} - p_n}{p_n - p} < 1$$

then the Aitken's sequence $\hat{p}_{n=1}^{\infty}$ converges to p faster than $p_{n=1}^{\infty}$ in the little-o sense. i.e

$$\lim_{n\to\infty} \frac{\hat{p}_n - p}{p_n - p} = 0$$

3.2 Exercises: Aitken's Technique

3.2.1

section The first question involves the derivation of the aforementioned equation for $\hat{p}$ explained above. We start with the given equation that describes a sequence $\{p_n\}_{n=1}^{\infty}$ converging linearly to p:

$$\frac{p_{n+1} - p}{p_n - p} \approx \frac{p_{n+2} - p}{p_{n+1} - p}.$$

Since our goal is to solve for p, we will begin with rewriting the equation:

$$(p_{n+1} - p)(p_{n+1} - p) = (p_n - p)(p_{n+2} - p).$$

$$(p_{n+1} - p)^2 = (p_n - p)(p_{n+2} - p).$$

Now we will define Δp_n and $\Delta^2 p_n$

$$\Delta p_n = p_{n+1} - p_n, \quad \Delta^2 p_n = p_{n+2} - 2p_{n+1} + p_n.$$

$$\hat{p} \approx p_n - \frac{(\Delta p_n)^2}{\Delta^2 p_n}.$$

$$\hat{p}_n = p_n - \frac{(p_{n+1} - p_n)^2}{p_{n+2} - 2p_{n+1} + p_n}.$$

This formula accelerates the convergence of a linearly convergent sequence by producing a refined estimate $\hat{p_n}$ using three consecutive iterates.

3.2.2

This question asks to write a sub-routine that takes a sequence of approximations, maximum number of iterations, and a tolerance level. The code for this question was completed in part. I have yet to implement the maximum number of iterations and tolerance into this subroutine. This code can be found in the Github repository.

3.2.3

I was not able to reach this portion of the lab.

3.3 Steffenson's Method

3.4 Exercises: Steffenson's Method

I was not able to reach this portion of the lab.

4 Deliverables

The code, *LaTeX* rendering, as well as the *LaTeX* code has been submitted to both canvas as well as the GitHub repository for grading.

5 Additional Fun

I was not able to reach this portion of the lab.