APPM 4600 Lab 4

Increasing the convergence rate of root finding methods

1 Overview

In this lab, you will investigate two techniques for increasing the convergence rate of linearly convergent root finding methods such as the fixed point method. The techniques you will be playing with are post processing techniques called *extrapolation* methods. Essentially, they take combinations of the iterations of a first order method and create a new sequence of approximations which are higher order convergent. Later in the semester we will derive extrapolation techniques (Taylor will be involved) but this lab is a numerical exploration. You will get practice developing low level code and investigating order of convergence.

2 Before lab

The only thing that you need to do before this lab is to create a new fixed point iteration subroutine that returns a vector whose entries are the approximations of the fixed point at all the iterations in order. To create this subroutine, you can modify the subroutine provided in class. Recall, to create a column vector of size N in `numpy`, you can use

```
import numpy as np
x = np.zeros((N,1))
```

2.1 Review order

Recall from Homework 3 and class the definition of the order of convergence.

Definition 2.1. 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

1. Given the fixed point p and the vector $\hat{\mathbf{p}}$ of approximations made by an iteration, develop a way to numerically determine the order of convergence of the algorithm that created the approximation.

2. The fixed point of $g(x) = \left(\frac{10}{x+4}\right)^{1/2}$ is $p = 1.3652300134140976$.

   (a) With the initial guess of $p_0 = 1.5$, how many iterations does it take for the fixed point iteration to converge with an absolute tolerance of 10^{-10}?

   (b) Use the technique to developed in problem 1 to determine the order of convergence of the fixed point method. (*You know this from class but it is good to practice numerically finding it.*) Also determine the constant associated with this convergence rate.

3 Lab day: Exploring order of convergence and creating higher order approximations out of low order approximations

In this lab you will explore two methods for creating high order approximations to fixed points. The first Aitken's Δ^2 method is built from a linearly convergent sequence of approximations. The second method called Steffensen's method is a hybrid of a fixed point iteration and Aitken's method. You will determine which technique produces the highest order approximations. You can also explore which is the most cost effective for approximating fixed points.

3.1 Aitken's Δ^2 acceleration technique

We will consider the acceleration technique called **Aitken's Δ^2 method**. To use this method you start with a sequence $\{p_n\}_{n=1}^{\infty}$ that converges linearly to the value p. Then you create a new sequence of approximations with iterations defined by

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

Theorem 3.1 (2.14 from text)**.** *Suppose that $\{p_n\}_{n=1}^{\infty}$ is a sequence that converges linearly to the limit p and that*

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

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

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

3.2 Exercises

- *Derivation of the method.* We know that a sequence $\{p_n\}_{n=1}^{\infty}$ converging linearly to p means that
$$\frac{p_{n+1}-p}{p_n-p} \sim \frac{p_{n+2}-p}{p_{n+1}-p}.$$
Take this equation and solve for p. The value p gives the formula for the iterates in *Aitken's Δ^2 method.*

- Write a subroutine that takes in a sequence of approximations and returns a vector of the approximations. It should also have tolerance and max number of iterations as input.

- Apply Aitken's Δ^2 method to the sequence created by the fixed point iteration in the before lab exercise set. Determine if the convergence is in fact faster than the fixed point iteration. Can you figure out the order of convergence?

3.3 Steffenson's method

Steffenson's method is a hybrid of the fixed point iteration and Aitken's Δ^2 method. The idea is to take two mini-steps with fixed point, then use that to create the new approximation with Aitken's method.

The iteration for finding the fixed point p of $g(x)$ is as follows

- $a = p_n$
- $b = g(p_n)$
- $c = g(b)$
- $p_{n+1} = a - \frac{(b-a)^2}{c-2b+a}$

3.4 Exercises

1. Write a pseudo-code for implementing Steffenson's method. What is needed for input? What will you output?
2. From this pseudo-code create a subroutine that implements Steffenson's method.
3. Apply the method to $g(x) = \left(\frac{10}{x+4}\right)^{1/2}$ with an initial guess of $p_0 = 1.5$ and tolerance of 10^{-10}. Recall that the fixed point is $p = 1.3652300134140976$.
4. Determine the order of convergence based on the sequence of approximations. How does this compare with the two other methods?

4 Deliverables

To receive full credit for this lab, you need to commit and push your Python codes to git. You also need to upload your solutions to the lab exercises to Canvas.

5 Additional fun

You may be interested in exploring some more. Try rerunning your numerical experiments for the finding the fixed point of the following functions $g(x)$ with initial guess p_0 and the the tolerance set to 10^{-6}:

(a) $p_0 = 0.5$, $g(x) = (2 - e^x + x^2)/3$

(b) $p_0 = 2$, $g(x) = \cos(x - 1)$

(c) $p_0 = 0.5$, $g(x) = 3^{-x}$

In order to create a reference for the "exact" fixed point, run your favorite algorithm with the tolerance set to 10^{-14} and use that determine your order of approximation.