Project 4: Iterative Methods

Hieu Bui

October 20, 2022

1 Introduction

We want to approximate what certain inputs will give the desired outputs. For example, finding the root of a function is a well known problem, where the root is our desired output. Although numerical approximation is a more brute force way of getting to the answers, we have computers to carry out such time consuming task. Approximation or numerical methods provide a fast and easy way to get close to the answers relative to the classical analytical way, especially for nonlinear and complex equations. In this project, we explore different approximation methods to find roots of any given functions. Furthermore, we could also compute the residuals of a method to gauge accuracy and efficiency.

2 Methods

2.1 Jacobi

For a system of equations, we derive expressions for each unknowns. Once we have the collection of such expressions, we begin to approximate them. First, we arbitrarily guess an initial value. We use plug in the initial value as input for all the expressions we derived earlier. After we obtain numerical values from all the expressions, we go back and substitute the new values back into the original expressions for a new set of values. We carry on the procedure until the results converge. For instance, we have 2 set of original equations:

$$ax_1 + bx_2^2 = 0$$

$$cx_1 - ax_2^2 = 0$$

We can solve for each x to get 2 expressions:

$$x_1 = \frac{-bx_2^2}{a}$$
$$x_2 = \sqrt{\frac{cx_1}{a}}$$

From here, we can begin to say that x_1 and x_2 are equal to 1. We would plug in 1 for the 2 expressions above to get new values of x_1 and x_2. The process continue until the results converge or diverge.

2.2 Gauss-Seidel

Gauss-Seidel process is nearly identical to the Jacobi method; however, we do not finish computing old values for all the expressions. Instead, we compute value for one expression and immediately use that value for the next expression. From the example above, we would have $x_1 = \frac{-b}{a}$ and $x_2 = \sqrt{\frac{c}{a}}$ if we use Jacobi and plug in 1 for 'all' the expressions. However, we would have $x_1 = \frac{-b}{a}$ and $x_2 = \sqrt{\frac{-cb}{a^2}}$ because we use the new value of x_1 for x_2 expression. This way, the values are updated much quicker, decreasing the number of iterations we have to do for the same final result.

2.3 SOR-Successive Over Relaxation

The Successive Over-Relaxation method add a weighted component that penalize any new approximation for having bigger error. Furthermore, it incorporates the Gauss-Seidel method of constantly using new iterated values for the next one, decreasing the amount of iterations significantly.

2.4 Fixed Point Iteration

In this project, we use fixed-point iteration to determine the roots of given functions. The root equation has a unique quality of the output being equivalent to the input: $f(x) = 0$, where x = 0. We can say that $f(x)$ is fixed. Now, let's write a new expression that contain $f(x)$ while maintaining the out-equal-input quality: $g(x) = x - f(x)$. Since $f(x) = 0$, it means that $g(x) = x$, thus, the equivalence quality still stand. Now we have 2 different expressions with a 'fixed' value, we can iterate to find the intersection of the 2 expressions, which is the root we try to find. One of the famous equation, in which the output is equal to the input, we can use as reference is y=x. The intersection between y=x and the equations of the root from the given function will allow us to determine the input where the root lie.

2.5 Newton Method

The Newton's Method basically say that if we follow the change of the function or the derivative of the function, we will get close to the root (if there exists a root). The derivation based on a 2 point slope form because the first order derivative we use to approximate is a linear line.

$$\begin{aligned} y - f(x) &= \frac{df(x)}{dx}(x - x_o) \\ \text{y} &= \text{0 for the root} \\ \frac{-f(x)}{\dot{f}(x)} &= x - x_o \\ x &= x_o - \frac{f(x)}{\dot{f}(x)} \end{aligned}$$

We iterate until x converge or diverge.

2.6 Secant Method

Geometrically speaking, the Secant Method works on the same principle as the Newton's Method. However, the Secant Method use the 2 points close to each other for the linear line rather than the exact tangent line.

2.7 GPS System

The GPS triangulation system determines the position of the receiver (our phones) by approximating the intersection of distance between the satellites and the receiver. Since we live in a 3 dimensional world, our position can be represented with spherical coordinates. Therefore, the minimum amount of satellites needed for an exact position is 3. However, due to signal noise, General Relativity, etc, we have a slight deviation in time between all the devices. In such case, we need a fourth satellite to account for the deviation. With four satellite, we can represent the positioning apparatus with system of four equations that depend on x,y, and z. Since we have a system of equations and their respective values; we can use Newton's Method to approximate for the unknowns, which are x, y, and z. The unknowns are our position in space.

3 Programming

3.1 Jacobi, Gauss-Seidel, and SOR

Inputs: function, initial guess, number of iteration Outputs: time taken to solve, iterations, residual values, correction values - Perform the iterative process with the input function and initial guess with respect to the method chosen - Log the residual and correction value after each iteration. - Continue to iterate until residual value is under the set threshold

3.2 Fixed-Point Iteration

Inputs: equation to solve, number of iteration, initial guess Outputs: root 1 and root 2 - Compute the chosen function - Set the function result to new input value (new x value) - Repeat until the input number of iteration is satisfied - Log the root values - Plot

3.3 Newton's Method and Secant's Method

Inputs: equation to solve, number of iteration, initial guess Outputs: root 1 and root 2 - Derive the derivative expression for a chosen function - Compute the chosen equation with initial guess - Compute the derivative from the derived expression - Compute new x based on results of original equation and derivative value - Iterate until number of iteration is satisfied - Log the root values - Plot

4 Results

Ranking from the fastest computational prowess(time, corrections, and residuals), we have SOR, Gauss-Seidel, and Jacobi respectively. On the other hand, the plot for the Newton's Method is simpler with only the curve of the original function compared fixed-point iteration method.

5 Discussion

We could use the method from problem 1 to test and modified the input parameters for later problems. This way, we would be able to achieve the most efficient guess as well as catching any divergence early on, preventing wasted time and energy. Overall, the approximation methods in this project works best when the initial guess is close to the desired output (root). Moreover, convergence is not guaranteed with these methods of approximation. Lastly, we could improve speed of an approximation method through the use of weighted variable and constant updated values.