Nonlinear Solvers

MATH-151: Mathematical Algorithms in Matlab

October 9, 2023

Why Solve Nonlinear Equations?

- We are going to focus now on finding x for a function such that $f(x) = 0$
- Lines are great, we know how to solve for where a line crosses zero quite easily, but many things in real life are nonlinear so we need to develop techniques for finding zeros for those equations as well

Why Solve Nonlinear Equations?

- We are going to focus now on finding x for a function such that $f(x) = 0$
- Lines are great, we know how to solve for where a line crosses zero quite easily, but many things in real life are nonlinear so we need to develop techniques for finding zeros for those equations as well
- Why do we want to find zeros in the first place?

Why Solve Nonlinear Equations?

- We are going to focus now on finding x for a function such that $f(x) = 0$
- Lines are great, we know how to solve for where a line crosses zero quite easily, but many things in real life are nonlinear so we need to develop techniques for finding zeros for those equations as well
- Why do we want to find zeros in the first place?
  - If we know how to find where $f(x) = 0$, we know how to find where it is any other c since we can instead solve $f(x) - c = g(x) = 0$
  - If we want to find where two curves $f(x)$ and $g(x)$ intersect, we can do $f(x) - g(x) = h(x) = 0$
  - Most engineering problems are interested in optimization. We remember from Calc I, if we want to find a minimizer or maxmizer, we start by finding a critical point where $f'(x) = 0$

Why Solve Nonlinear Equations?

- We are going to focus now on finding x for a function such that $f(x) = 0$
- Lines are great, we know how to solve for where a line crosses zero quite easily, but many things in real life are nonlinear so we need to develop techniques for finding zeros for those equations as well
- Why do we want to find zeros in the first place?
  - If we know how to find where $f(x) = 0$, we know how to find where it is any other c since we can instead solve $f(x) - c = g(x) = 0$
  - If we want to find where two curves $f(x)$ and $g(x)$ intersect, we can do $f(x) - g(x) = h(x) = 0$
  - Most engineering problems are interested in optimization. We remember from Calc I, if we want to find a minimizer or maxmizer, we start by finding a critical point where $f'(x) = 0$
- Basically any time we want to know where anything occurs, we either run into a line or a case to solve for the root of a nonlinear equation!
  - We usually like to know when/where things occur

Bisection Method

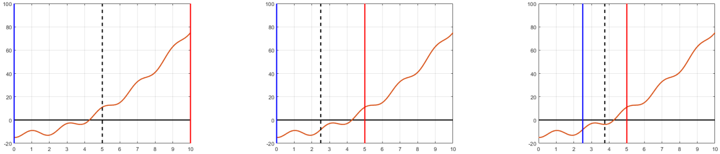
- We've already seen one of these a few weeks ago! The Bisection method!

Bisection Method

- We've already seen one of these a few weeks ago! The Bisection method!
- Suppose we know $f(a)$ is negative and $f(b)$ is positive. If f is a continuous function, it must pass through 0 between a and b.

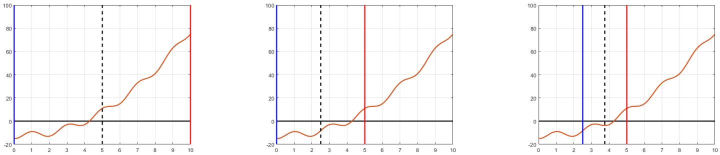
Bisection Method

- We've already seen one of these a few weeks ago! The Bisection method!
- Suppose we know $f(a)$ is negative and $f(b)$ is positive. If f is a continuous function, it must pass through 0 between a and b.
- So we cut it in half and try $f(m)$, where $m = \frac{a+b}{2}$
  - If it has the same sign as $f(a)$, m becomes our new a
  - If it has the same sign as $f(b)$, m becomes our new b



Bisection Method

- We've already seen one of these a few weeks ago! The Bisection method!
- Suppose we know $f(a)$ is negative and $f(b)$ is positive. If f is a continuous function, it must pass through 0 between a and b.
- So we cut it in half and try $f(m)$, where $m = \frac{a+b}{2}$
  - If it has the same sign as $f(a)$, m becomes our new a
  - If it has the same sign as $f(b)$, m becomes our new b



- This gives us a new, smaller possible range, so we do Bisection Method on that range
- We continue until $f(m)$ is sufficiently small.

Fixed Point Equations

- How do we tell the computer to know when it's found a zero? For that we introduce the concept of a **fixed point**.

Fixed Point Equations

- How do we tell the computer to know when it's found a zero? For that we introduce the concept of a **fixed point**.
- If $g(x)$ is a function that tells us our next guess for the zero, a fixed point satisfies the following

$$x_n = g(x_n) = x_n + f(x_n)h(x_n)$$

Fixed Point Equations

- How do we tell the computer to know when it's found a zero? For that we introduce the concept of a **fixed point**.
- If $g(x)$ is a function that tells us our next guess for the zero, a fixed point satisfies the following

$$x_n = g(x_n) = x_n + f(x_n)h(x_n)$$

- When this occurs, our guess for the next point is the same as our current guess
  - This will occur when $f(x_n) = 0$

Fixed Point Equations

- How do we tell the computer to know when it's found a zero? For that we introduce the concept of a **fixed point**.
- If $g(x)$ is a function that tells us our next guess for the zero, a fixed point satisfies the following

$$x_n = g(x_n) = x_n + f(x_n)h(x_n)$$

- When this occurs, our guess for the next point is the same as our current guess
  - This will occur when $f(x_n) = 0$
- That's what we are looking for! We found an x_n such that $f(x_n) = 0$.
  - Remember, we will almost never get exactly 0 so we repeat our process until $f(x_n)$ or $x_n - x_{n-1}$ become small enough
  - Or perhaps, we repeat *while* $f(x_n)$ is too big

Fixed Point Equations

- How do we tell the computer to know when it's found a zero? For that we introduce the concept of a **fixed point**.
- If $g(x)$ is a function that tells us our next guess for the zero, a fixed point satisfies the following

$$x_n = g(x_n) = x_n + f(x_n)h(x_n)$$

- When this occurs, our guess for the next point is the same as our current guess
  - This will occur when $f(x_n) = 0$
- That's what we are looking for! We found an x_n such that $f(x_n) = 0$.
  - Remember, we will almost never get exactly 0 so we repeat our process until $f(x_n)$ or $x_n - x_{n-1}$ become small enough
  - Or perhaps, we repeat *while* $f(x_n)$ is too big
- Great! So what can we do to set these up? What would this $g(x_n)$ look like?

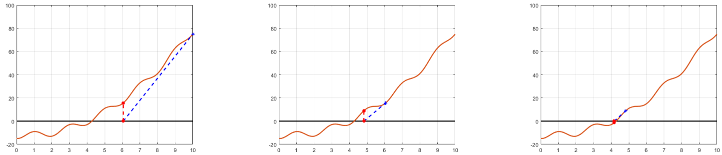
Newton's Method

- This is a widely used method that is usually much faster than bisection method!

Newton's Method

- This is a widely used method that is usually much faster than bisection method!
- We pretend our function is the tangent line at our current guess, so that means our next guess for x is where the tangent line crosses 0

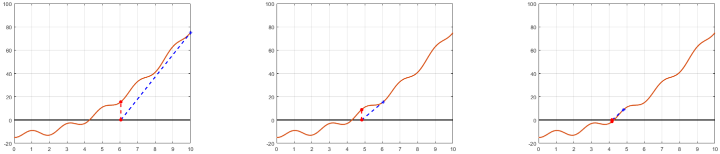
$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



Newton's Method

- This is a widely used method that is usually much faster than bisection method!
- We pretend our function is the tangent line at our current guess, so that means our next guess for x is where the tangent line crosses 0

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



- We are not only using the values of our function, but also information about what the curve looks like. This will help us get closer to our 0 faster!
  - This does have some risk though, we could encounter a point where $f'(x_n) = 0$...

Secant Method

- Sometimes we don't have a derivative function, or it would be very difficult to implement

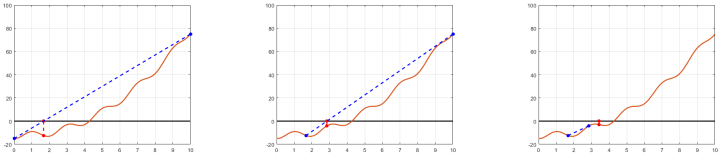
Secant Method

- Sometimes we don't have a derivative function, or it would be very difficult to implement
- In those cases, we can estimate our derivative using finite difference method with our last 2 points!
    - But this means we need to make 2 initial guesses to start

Secant Method

- Sometimes we don't have a derivative function, or it would be very difficult to implement
- In those cases, we can estimate our derivative using finite difference method with our last 2 points!
    - But this means we need to make 2 initial guesses to start
- Doing this we have a slightly different method called **Secant Method**

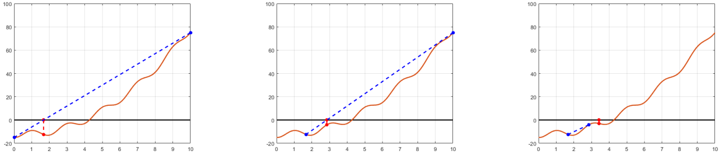
$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$



Secant Method

- Sometimes we don't have a derivative function, or it would be very difficult to implement
- In those cases, we can estimate our derivative using finite difference method with our last 2 points!
  - But this means we need to make 2 initial guesses to start
- Doing this we have a slightly different method called **Secant Method**

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$



- The code I use in my research is actually using Secant method because our gradients are too complicated to find analytically!