PROJECT 5 – BISECTION AND NEWTON'S METHOD

Due by 11pm Wednesday, October 14, 2020:

In a Mathematica notebook title the project, and create sections called "Bisection Method," "Newton's Method," "Numerical Newton's Method," "Building a Function Call," and "Problems with Newton's Method."

(1) **Bisection Method:** Define the function $F(x) = x^2 \cos(x) + x + 3$ and plot it over the region $-6 \leq x \leq 6$. Let $a = -4$ and $b = 2$. Evaluate $F(a)$ and $F(b)$ and show that they have different signs. Write a `For` loop which implements the Bisection Method with 10 iterations. Print your final estimate of c where $F(c) = 0$. It may be helpful to use the `Max` and `Min` functions.

(2) **Newton's Method:** Implement Newton's Method (only 5 iterates) on the same function F and using $x = -3$ as a starting point. Print your estimate of c where $F(c) = 0$.

(3) **Numerical Newton's Method:** Copy, paste, and edit your code for Newton's method by replacing the reference to $F'(x)$ with the approximation $F'(x) \approx \frac{F(x+h)-F(x)}{h}$ for $h = 0.001$. How does this version of the algorithm compare to the version using the actual derivative of F?

(4) **Building a Function Call:** Using the `Module` command, write a Mathematica function named `bisectionMethod` which takes four arguments: f, a, b, and n. The function should apply the Bisection Method to the function f with initial end points a and b and iterate the process n times. The function `bisectionMethod` should return a list of pairs: `{{min,max},{f(min), f(max)}}` where `min` and `max` form the interval in which the zero of f must lie.

(5) **Problems with Newton's Method:** Sometimes Newtons Method works quite well. For example, apply your Numerical Newton's Method to the function $H(x) = \arctan(x)$ with the initial point $x = 0.5$ and stepsize $h = 0.001$ and 10 iterates. Edit your code so at each iteration it prints the pair `{x,H(x)}`. Note however, that Newton's Method can be sensitive to starting with a decent approximation for the zero of the given function. For example, copy-paste your code and run it again with a single change: Let the initial condition be $x = 2.0$. What happens? Why do you think this is happening? It may help to plot the arctan function and think about how Newton's Method works. Finally, define the Mathematica function:

```
G[x_]:=Sign[x]*(Abs[x])^(1/3).
```

This is one way to define the function $G(x) = x^{\frac{1}{3}}$ in Mathematica. Implement your Numerical Newton's Method on G using $x = 1.0$ as the intial conditions. What happens? Why does this happen? It may be useful to plot G.