Programming Assignment 1

In this assignment we will combine Bisection and Newton's method. This allows us to have the excellent convergence properties of Newton's method while maintaining the guaranteed convergence of the Bisection method.

Complete the following steps.

1) Implement the following algorithm in whatever language you desire:

ALGORITHM: $[a, b] \leftarrow$ **FindInterval**(f, x_0)
Input : f, a function $f\colon \mathbb{R} \to \mathbb{R}$,
x_0, real number.
Output: $[a, b]$, an interval containing x_0 such that $f(a)$ and $f(b)$ have different signs.

$a \leftarrow x_0$
$b \leftarrow x_0$
$dx \leftarrow 0.001$
while $f(a)$ and $f(b)$ do not have different signs **do**
    $a \leftarrow a - dx$
    $b \leftarrow b + dx$
return $[a, b]$

2) Implement the following algorithm in whatever language you desire:

ALGORITHM: $P \leftarrow$ **NewtonBisection**$([a, b], f, \varepsilon)$
Input : $[a, b]$, an interval,
f, a function $f \in C[a, b]$ that has exactly one solution on $[a, b]$
and $f(a)$ and $f(b)$ have different signs,
ε, a real number.
Output: P, a real number such that $P \in [a, b]$ and $|f(P)| < \varepsilon$.

$n \leftarrow 1$
$p_n \leftarrow a$
$a_n \leftarrow a$
$b_n \leftarrow b$
while $|f(p_n)| \geq \varepsilon$ **do**
    $candidate \leftarrow p_n - \dfrac{f(p_n)}{f'(p_n)}$
    if $candidate \notin [a_n, b_n]$ **then**
        $p_{n+1} \leftarrow \dfrac{a_n + b_n}{2}$
    else
        $p_{n+1} \leftarrow candidate$
    if $f(p_{n+1})$ and $f(b_n)$ have different signs **then**
        $a_{n+1} \leftarrow p_{n+1}$
        $b_{n+1} \leftarrow b_n$
    else
        $b_{n+1} \leftarrow p_{n+1}$
        $a_{n+1} \leftarrow a_n$
    $n \leftarrow n + 1$
return p_n

Use the Bisection and Newton method *Mathematica* code I sent you as a starting point. This algorithm is a combination of the two. Implement two methods, one that follows the scheme above, and the other which outputs a table of values $n, p_n, a_n, b_n, f(p_n)$, and whether a Newton step or a Bisection step was taken.

3) Run your method **NewtonBisection** using $f(x) = \sin x - e^{-x}$ on the interval $[1.9, 30]$ with $\varepsilon = 0.00000001$. Present the results in a table showing $n, p_n, a_n, b_n, f(p_n)$, and whether a Newton step or a Bisection step was taken. Here is what my output looked like:

n	p_n	a_n	b_n	$f(p_n)$	Method
1	1.9	1.9	30	0.796731	--
2	6.48627	6.48627	30	0.200169	Newton
3	18.2431	6.48627	18.2431	-0.56993	Bisection
4	12.3647	6.48627	12.3647	-0.200307	Bisection
5	9.42549	6.48627	9.42549	-0.000790432	Bisection
6	9.4247	9.4247	9.42549	1.14238×10^{-10}	Newton

4) Use your combined **FindInterval** and **NewtonBisection** method to find the roots of $f(x) = \sin x - e^{-x}$ with $x_0 = -3, -2, \ldots, 10$ as inputs to **FindInterval**. Use $\varepsilon = 0.00000001$ as before for **NewtonBisection**.

```
1  f ← sin x − e^{−x}
   ε ← 0.00000001
2  for x_0 = −3 to 10 do
3  |  [a, b] ← FindInterval(f, x_0)
4  |  P ← NewtonBisection([a, b], f, ε)
5  |  print {x_0, P, [a, b]}
```

Present your results in a table which gives x_0, the approximated root, and the enclosing interval calculated by **FindInterval**. Here is what my output looked like:

x_0	p	a	b
-3	0.588533	-6.589	0.589
-2	0.588533	-4.589	0.589
-1	0.588533	-2.589	0.589
0	0.588533	-0.589	0.589
1	0.588533	0.588	1.412
2	3.09636	0.903	3.097
3	3.09636	2.903	3.097
4	3.09636	3.096	4.904
5	6.28505	3.714	6.286
6	6.28505	5.714	6.286
7	6.28505	6.285	7.715
8	9.4247	6.575	9.425
9	9.4247	8.575	9.425
10	9.4247	9.424	10.576

Be sure to turn in your source code along with the aforementioned tables.