In this assignment, we will address two issues with the Bisection method and Newton’s method:

- Finding an interval \([a, b]\) for the Bisection method, with \(f(a)\) and \(f(b)\) having different signs.
- Combining the excellent convergence properties of Newton’s method with the guaranteed root-finding of the Bisection method.

1. Implement a MATLAB function \texttt{findbracket} of the form

   \[
   \text{function } [a,b]=\text{findbracket}\left(f,x_0\right)
   \]

   which finds an interval \([a, b]\) around \(x_0\) such that \(f(a)\) and \(f(b)\) have different signs. Use the following strategy:

   1. Start with \(a = b = x_0\), and \(dx = 0.001\)
   2. Subtract \(dx\) from \(a\), and terminate if \(f(a)f(b) < 0\)
   3. Add \(dx\) to \(b\), and terminate if \(f(a)f(b) < 0\)
   4. Multiply \(dx\) by 2 and repeat from step 2.

2. Implement a MATLAB function \texttt{newtonbisection} of the form

   \[
   \text{function } p=\text{newtonbisection}\left(f,df,a,b,tol\right)
   \]

   combining Newton’s method and the Bisection method according to the following strategy:

   1. Start with \(p = a\)
   2. Attempt a Newton step \(p = p - f(p)/f'(p)\)
   3. If \(p\) is outside of \([a, b]\), set \(p = (a + b)/2\)
   4. If \(f(p)f(b) < 0\), set \(a = p\), otherwise set \(b = p\)
   5. Terminate if \(|f(p)| < tol\)
   6. Repeat from step 2.

   Use the functions \texttt{newton} and \texttt{bisection} on the course web page as a starting point, this function will be like a combination of the two.

3. Run your function \texttt{newtonbisection} using \(f(x) = \sin x - e^{-x}\) on the interval \([1.9, 30]\):

   \[
   f=@(x) \sin(x)-\exp(-x);
   df=@(x) \cos(x)+\exp(-x);
   x=newtonbisection(f,df,1.9,30,1e-8);
   \]

   Present the result in a table showing for each iteration the method used (Newton or Bisect), \(a, b, p\), and \(f(p)\).
4. Use your combined `findbracket` and `newtonbisection` to solve for the roots of \( f(x) = \sin x - e^{-x} \) with \( x_0 = -3, -2, \ldots, 10 \):

```matlab
f=@(x) sin(x)-exp(-x);
df=@(x) cos(x)+exp(-x);
for x0=-3:10
    [a,b]=findbracket(f,x0);
    x=newtonbisection(f,df,a,b,1e-8);
    [x0,a,b,x]
end
```

Present your results in a table showing \( x_0 \), \( a \), \( b \), and \( x \).

**Reporting requirements:**

The GSIs will *not* run any submitted MATLAB codes. Prepare a report showing the requested information, which is essentially just your MATLAB functions and the computed tables. Give brief comments if things do not work as expected.