Peter Julius M. Estacio
Grouped with Nino de Mesa and Osh Ong
ENGG 27.01
September 19, 2025

Progress Report 1 for Project 2: Hybrid Root-Finding Approach

The program is able to initially implement Brent's method to find roots of a given function through the Secant Method and Reverse Quadratic Interpolation. To guarantee covergence, it also makes use of the Bisection method to find the root. All these are implemented via the `brent_root_find()` function within the `RootResult` struct. The program also is able to provide a step-by-step trace of the function, allowing the user to see what method is used, how the program iterates, and the final result. The function value was also able to be displayed in scientific notation, up to 6 decimal places. Lastly, work has been done to create a user interface for the program, allowing only valid and safe inputs. Further work includes checking the accuracy of the program's results.

A sample output of the program is shown below:

```
=== Hybrid Root Finder (Brent) ===
  1) f(x) = x^3 - x - 2
  2) f(x) = cos(x) - x
  3) f(x) = e^(-x) - x
  4) f(x) = x*sin(x) - 1
  5) f(x) = (x-1)*(x-1)*(x-1)
  0) Exit
Choose a function [0-5]: 2
You chose: f(x) = cos(x) - x
Enter a (left endpoint): -7
Enter b (right endpoint): 5
Enable step-by-step TRACE output? (y/n): y

Solving with Brent's method...
(secant method) 4.615104872215952e-01, f(4.615105e-01) = 4.338704e-01
(bisection) 2.730755243610798e+00, f(2.730755e+00) = -3.647542e+00

=== Result ===
Converged: yes
Root     : 4.615104872215952e-01
f(root)  : 4.338704e-01
Iterations: 2
Would you like to choose again? (y/n): n
[1] + Done                       "/usr/bin/gdb" --interpreter=mi --tty=${DbgTerm} 0<"/tmp/Microsoft-MIEngine-In-bw5zxz44.y1h" 1>"
/tmp/Microsoft-MIEngine-Out-3tsu1qaj.lhb"
```