

Project 2: Self-Evaluation

Niño Aloysius V. De Mesa

Grouped with: Joshnare Ong and Peter Estacio

ENGG 27.01 – M

October 14, 2025

Documentation:

```
== 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]: 3
You chose: f(x) = e^(-x) - x
Enter a (left endpoint): -5
Enter b (right endpoint): 4
Enable step-by-step TRACE output? (y/n): y

Solving with Brent's method...
(secant method) 3.8e+00, f(3.77232e+00) = -3.74932e+00
(inverse quadratic interpolation) 1.862042891859517e-01, f(1.86204e-01) = 6.43900e-01
(bisection method) 1.979263728856475e+00, f(1.97926e+00) = -1.84109e+00
(bisection method) 1.082734009021213e+00, f(1.08273e+00) = -7.44066e-01
(bisection method) 6.344691491035825e-01, f(6.34469e-01) = -1.04252e-01
(secant method) 5.7e-01, f(5.72005e-01) = -7.61240e-03
(inverse quadratic interpolation) 5.67e-01, f(5.67142e-01) = 1.70014e-06
(bisection method) 5.70e-01, f(5.69574e-01) = -3.80702e-03
(bisection method) 5.68e-01, f(5.68358e-01) = -1.90308e-03
(bisection method) 5.678e-01, f(5.67750e-01) = -9.50795e-04
(bisection method) 5.674e-01, f(5.67446e-01) = -4.74574e-04
(bisection method) 5.673e-01, f(5.67294e-01) = -2.36443e-04
(bisection method) 5.6722e-01, f(5.67218e-01) = -1.17373e-04
(bisection method) 5.6718e-01, f(5.67180e-01) = -5.78369e-05
(bisection method) 5.6716e-01, f(5.67161e-01) = -2.80685e-05
(bisection method) 5.67152e-01, f(5.67152e-01) = -1.31842e-05
(bisection method) 5.67147e-01, f(5.67147e-01) = -5.74204e-06
(bisection method) 5.67145e-01, f(5.67145e-01) = -2.02095e-06
(bisection method) 5.67143e-01, f(5.67143e-01) = -1.60406e-07
(secant method) 5.671433e-01, f(5.67143e-01) = -3.16414e-14

== Result ==
Converged: yes
Root      : 5.671432904098040e-01
f(root)   : -3.164136e-14
Iterations: 21
Would you like to choose again? (y/n): |
```

Code Locator:

Code items	File name	Line numbers
Routine/Function implementing the hybrid root-finding approach	brent_root.cpp	48 - 198
Routine for the open method used	brent_root.cpp	Inverse Quadratic Interpolation: 107 - 115 Secant: 116 - 125
Routine for the bracketing method used	brent_root.cpp	133 - 152
Code used to decide between open and bracketing methods	brent_root.cpp	126 - 131

Project Evaluation - Hybrid Root-Finding Approach		
Item	Points	Rubrics
Early work	8/8	(all or nothing) 8: at most 10% of the code in the final implementation differs from that in early work submission
required documentation and screenshots of at least one successful hybrid computation	50/50	0 - no documentation or screenshots or documentation or screenshots inconsistent with actual program run 50 - documentation and screenshots have all required details
trace details	10/10	0 - instructions not followed 10 - all details are correctly followed
modularity and generality of the hybrid root-finding routine	16/16	6 - routine can easily be used to find roots of other functions, with different parameters, and settings
C++ function implementation	8/8	8 - routine can easily be used

of the hybrid root-finding routine		to find roots of other functions, with different parameters, and settings, and was encoded as a C++ function with appropriate parameters
code locator table	4/4	4 - all details in code locator table are accurate
self-evaluation	4/4	0 - no self-evaluation 2 - self-evaluation score differs from project score by more than 10 points 4 - self-evaluation accurate (or evaluating this item leads to an error)
total	100/100	