Exercise 5

All outputs are also found in the .sol files in the output folder.
Github: https://github.com/eschencpp/polyRoot

Exercise 1: $x^3 + 3x - 1$, on [0,1]

Bisection: Root: 0.32218535462608555 Iterations: 127 Success

Newton: Root: 0.32218535462608555 Iterations: 10000 Fail

Secant: Root: 0.3221853546260856 Iterations: 9 Fail

Hybrid: Root: 0.32218535462608555 Iterations: 10000 Fail

Exercise 2: $x^3 + 2x^2 + 10x - 20$, starting with x_0 = 2

Bisection: Root: 1.3688081078213725 Iterations: 127 Success

Newton: Root: 1.3688081078213725 Iterations: 10000 Fail

Secant: Root: 1.3688081078213725 Iterations: 8 Fail

Hybrid: Root: 1.3688081078213727 Iterations: 10000 Fail

Exercise 3: $x^3 + 2x^2 + 10x - 20$, with x_0 = 2, and x_1 = 1.

Bisection: Root: 1.3688081078213725 Iterations: 127 Success

Newton: Root: 1.3688081078213725 Iterations: 10000 Fail

Secant: Root: 1.3688081078213725 Iterations: 8 Fail

Hybrid: Root: 1.3688081078213727 Iterations: 10000 Fail

Exercise 4: $3x^3 + 5x^2 - 7$ [0,1]

Bisection: Root: 0.9451800564209666 Iterations: 53 Success

Newton: Root: 0.9451800564209666 Iterations: 5 Success

Secant: Root: 0.9451800564209666 Iterations: 7 Success

Hybrid: Root: 0.9451800564209666 Iterations: 11 Success

Analysis

For Exercises 1,2,3 all functions converged with Bisection method at 127 iterations. The Newton and Hybrid method both reached max iterations but still yielded the same root result. I assume this is because Newtons method may have trouble fully converging to a small epsilon and hybrid follows this because it ends up using Newton method to finish. The secant method converged fairly quickly, but the status ended up as Fail because (fb - fa) was 0 and that is the divisor. Dividing by 0 is not possible, so the program returned the current root. However, this root is still fairly accurate.

For Exercise 4, all methods ended in success. The Newton method converged the fastest at 5 iterations, followed by secant at 7 iterations, then hybrid at 11 iterations, and finally Bisection at 53 iterations. The resulting root was also the same to the last decimal place. The speed of these was expected, with Newton being the fastest and Bisection the slowest. The hybrid method started with bisection until epsilon of 0.1 or 10 iterations and ended up using 4 Bisection iterations followed by 7 Newton. The results being in between the Newton method and Bisection method was expected as well.