Math 450

Programming Assignment 1

Due: Monday October 8th

Let $f(x) = x^5 - 5x^4 + 8x^3 - 5x^2 + 11x - 7$ and let $p_0 = 0$.
When needed, you can take the interval of interest to be $[0, 1]$.

1. Write a program that carries out the *Bisection Method* on $f(x)$. You do not need to make your program take arbitrary input (i.e. you can tailor it to this specific $f(x)$). Your program should take as input the appropriate number of initial guesses, an interval $[a, b]$ and a desired error tolerance TOL. It should output a root r that is within the desired tolerance and it should output N, the number of iterations it took to get to within the error tolerance. Make it clear what procedure you are using to compute the error of your approximations.

2. Write a program that carries out *Newton's Method* on $f(x)$. You do not need to make your program take arbitrary input (i.e. you can tailor it to this specific $f(x)$). Your program should take as input the appropriate number of initial guesses, an interval $[a, b]$ and a desired error tolerance TOL. It should output a root r that is within the desired tolerance and it should output N, the number of iterations it took to get to within the error tolerance. Make it clear what procedure you are using to compute the error of your approximations.

3. Write a program that carries out *Aitken's Method* on $f(x)$. You do not need to make your program take arbitrary input (i.e. you can tailor it to this specific $f(x)$). Your program should take as input the appropriate number of initial guesses, an interval $[a, b]$ and a desired error tolerance TOL. It should output a root r that is within the desired tolerance and it should output N, the number of iterations it took to get to within the error tolerance. Make it clear what procedure you are using to compute the error of your approximations.

4. Write a program that carries out *Steffensen's Method* on $f(x)$. You do not need to make your program take arbitrary input (i.e. you can tailor it to this specific $f(x)$). Your program should take as input the appropriate number of initial guesses, an interval $[a, b]$ and a desired error tolerance TOL. It should output a root r that is within the desired tolerance and it should output N, the number of iterations it took to get to within the error tolerance. Make it clear what procedure you are using to compute the error of your approximations.

5. Write a program that carries out *Müller's Method* on $f(x)$. You do not need to make your program take arbitrary input (i.e. you can tailor it to this specific $f(x)$). Your program should take as input the appropriate number of initial guesses, an interval $[a, b]$ and a desired error tolerance TOL. It should output a root r that is within the desired tolerance and it should output N, the number of iterations it took to get to within the error tolerance. Make it clear what procedure you are using to compute your initial points p_1 and p_2, and what procedure you are using to compute the error of your approximations.

6. Use each of the programs you wrote to find an approximation of a root of $f(x)$ as given above on the interval $[0, 1]$ (with $p_0 = 0$) to within an accuracy of 10^{-5}.

7. Based on the results of your algorithms, comment on the relative effectiveness of these methods of approximating roots. Which appears to be the fastest? Which appears to be the slowest?