

Name: \_\_\_\_\_

Date: \_\_\_\_\_

Collaborators: \_\_\_\_\_

(Collaborators submit their individually written assignments together)

Question:	1	2	3	Total
Points:	45	20	10	75
Score:				

**Instructor/grader comments:**

1. One of the most attractive properties of the Newton's method is its quadratic convergence. This property is however lost when attempting to use the method to find a root of multiplicity  $m$ ,  $m > 1$  — the convergence turns to the linear one.

There are several ways to “recover” the quadratic convergence property. For example, given a function  $f(x)$  with a root of **known** multiplicity  $m$ , we can instead use Newton's method on the function  $g(x) = [f(x)]^{\frac{1}{m}}$  which has the same root but with multiplicity one.

Another variation of Newton's method can be used even when the multiplicity of the root is not known. If a function  $f(x)$  has a root of multiplicity  $m$  at a point  $x_0$ , its derivative  $f'(x)$  has a root of multiplicity  $m - 1$  at that point. Thus the function

$$g(x) = \frac{f(x)}{f'(x)}$$

has all simple roots at all of roots of  $f(x)$ . Thus we can apply Newton's method to the function  $g(x)$  and expect the quadratic convergence.

- (a) (15 points) Starting from the formula for Newton's iterations for a function with simple roots,

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)},$$

where  $g(x)$  is a function defined above, derive the formula for “modified” Newton's iterations for the function  $f(x)$ . Write your derivation in the space below.

- (b) (10 points) Write a matlab function, `newtm(F, Fprime, Fdprime, x0, tol)`, that implements your algorithm and returns the values of the root and the number of function evaluations that was used to find the root. (Count together the number of evaluations of the function and its derivatives.) Provide a help string for your function.
- (c) (15 points) Write a matlab script (call it e.g. **hw07p1.m**) that tests your algorithm using the following function:

$$f(x) = \sin^4(x).$$

Find the expressions for  $f'(x)$  and  $f''(x)$ . Use the tolerance  $\text{tol} = 2 * \pi * \text{eps}$  and the initial approximation for the root  $x_0 = 3$ .

Compare your results with the results (both for the value of the root and the number of function evaluations) of “unmodified” Newton’s algorithm. Use the code for the Newton’s algorithm that we developed in class.

- (d) (5 points) Describe your results in the project’s README.md file.
2. (20 points) Find closest to zero root of the following nonlinear equation:

$$e^{-x} = \sin(x).$$

Compare the results and the performance (in terms of the number of function evaluations) of the bisection, the Newton’s, and the secant algorithms. Use the code that we developed in class. Use  $\text{tol} = \pi * \text{eps}$ . Determine (and describe your reasoning in the README.md file) a suitable initial approximation (for the Newton’s method) and a suitable initial interval (for bisection and secant methods). Describe your results in the README.md file. Place your code to the file named **hw07p2.m**

Hint: sketch the graphs of both sides of the equation to see where the roots of the equation are located.

### Gitlab

3. (10 points) Create a gitlab project called **hw07** (name it exactly as shown). Upload **all** matlab files that are required to run your code. Share the project with the instructor and the TA and grant them **Reporter** privileges.