Name: ______________________________

Date: ______________________________

Collaborators: ____

(If applicable, collaborators submit their individually written assignments together)

Question:	1	2	3	4	Total
Points:	30	25	30	5	90
Score:					

Instructor/grader comments:

1. Two ladders crisscross an alley of unknown width W. Each ladder reaches from the base of one wall to some point on the opposite wall. The ladders cross at a height H above the pavement. Find W given that the lengths of the ladders are $L_1 = 20$ and $L_2 = 30$ and that $H = 8$. (Lengths are in some arbitrary units.)

   Hint: the equation that determines the width of the alley is as following:
   $$f(W, L_1, L_2, H) = 0,$$
   where
   $$f(W, L_1, L_2, H) = \frac{1}{H} - \frac{1}{\sqrt{L_1^2 - W^2}} - \frac{1}{\sqrt{L_2^2 - W^2}}.$$
   For the reference,
   $$\frac{df}{dW} = -\frac{W}{\sqrt{(L_1^2 - W^2)^3}} - \frac{W}{\sqrt{(L_2^2 - W^2)^3}}$$

   (a) (5 points) Write a matlab function, `hw07p1alley(w, l1, l2, h)`, that returns the values of $f(W, L_1, L_2, H)$. Place it in its own matlab file. Provide a help string for your function.

   (b) (5 points) Write a matlab function, `hw07p1alleyd(w, l1, l2, h)`, that returns the values of df/dW. Place it in its own matlab file. Provide a help string for your function.

   (c) (20 points) Write a matlab script (call it e.g. **hw07p1.m**) that (a) initializes the values of the parameters L_1, L_2, and H; (b) prints the help texts for `hw07p1alley` and `hw07p1alleyd` (c) defines an anonymous functions of a single variable, `ff(w)`, that itself uses `hw07p1alley(w, l1, l2, h)` and `ffd(w)` uses `hw07p1alleyd(w, l1, l2, h)` ; (c) solves the equation $ff(w) = 0$ using the Newton's method.

   Use the code that we developed in class. Place the commands `clear`, and `format compact` at the top of your script.

   Specify your initial approximation for W and your final answer in your README.md file.

2. Find the height, H, reached by $V_0 = 1.1\ \mathrm{m}^3$ of water stored in a spherical tank of radius $R = 1.0$ m.

   The volume of the spherical segment of the "depth" H, $H \le R$ is
   $$V(R, H) = \pi H^2 \left(R - \frac{H}{3}\right),$$

where R is the radius of the sphere.

(a) (5 points) Write a matlab function of a single variable, `sphsegment(h)`, that returns the values of $V(R,H)-V_0$. Include the values of the parameters V_0 and R into the code of your function.

(b) (5 points) Write a matlab function of a single variable, `sphsegmentd(h)`, that returns the derivative of $V(R,H)-V_0$ with respect to H. Include the values of the parameters V_0 and R into the code of your function.

(c) (15 points) Write a matlab script (call it e.g. **hw07p2.m**) that solves the equation using the Newton's method with the height of the water in the tank.

Modified Newton's method for nonlinear equation with a multiple root

3. 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.

   The iteration steps are as following:

$$x_{i+1}=x_i-\frac{f(x_i)f'(x_i)}{f'(x_i)^2-f(x_i)f''(x_i)}.$$

   (a) (10 points) Write a matlab function, `hw07p2newtm(F, Fprime, Fdprime, x0, itermax)`, that implements the algorithm and returns the vector of the root approximations. Provide a help string for your function. Place the function in its own file.

   (b) (15 points) Write a matlab script (call it e.g. **hw07p2.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 initial approximation for the root `x1 = 3`.

Count and the number of function evaluations that was used to find the root - add together the number of evaluations of the function and its derivatives.

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.

(c) (5 points) Describe your results in the project's README.md file.

Gitlab

4. (5 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 TAs and grant them **Reporter** privileges.