Name: ______________________________

Date: ______________________________

Collaborators: ____

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

Question:	1	2	3	Total
Points:	40	25	10	75
Score:				

Instructor/grader comments:

Roots of nonlinear equations

1. Apply Newton's method to find the single point of intersection of the spheres with center 1, 0, 1) and radius $\sqrt{8}$, center $(0,2,2)$ and radius $\sqrt{2}$, and center $(0,3,3)$ and radius $\sqrt{2}$.

   Hint: although a generic intersection of three spheres in in three-dimensional space is two points, it can be a single point.

   Hint: In Cartesian coordinates the equation of a sphere of radius R and the center at x_c, y_c, z_c is as following:

   $$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = R^2.$$

   (a) (10 points) Write down the system of equations that you need solve:

   (b) (10 points) Calculate the Jacobian.

(c) (10 points) Write matlab function `hw09p1spheres(x)` that accept the vector of position in three-dimensional space and returns the values of residues and the Jacobian.

(d) (10 points) Write matlab script **hw09p1.m** that uses the code we developed in class and finds the intersection point. Use the usual rules for writing scripts. In the README file (a) specify your initial approximation and the number of the iterations; (b) does the iterations converge quadratically? Explain how you came to the conclusion.

2. An algorithm for finding the roots of nonlinear equation uses three function evaluations per iteration step and claims the following relations between the errors of successive steps:

$$\epsilon_{n+1} = \alpha \epsilon_n \sqrt{\epsilon_{n-1}},$$

where α is a positive constant.

(a) (20 points) What is the convergence rate of the algorithm (per iteration)? Show your calculations in the space below.

Arrange the algorithms from the fastest to the slowest (per iteration): the "mystery" algorithm, bisection, secant, Newton's. Explain.

(b) (5 points) What is the convergence rate of the algorithm (per function evaluation)? Show your calculations in the space below. Arrange the algorithms from the fastest to the slowest (per function evaluation): the “mystery” algorithm, bisection, secant, newton’s.

Gitlab

3. (10 points) Create a gitlab project called **hw09** (name it exactly as shown). Upload **all** matlab files that are required to run your code. Create README.md file - leave it empty if appropriate. Share the project with the instructor (gitlab user name `m3510_21f_in`) and the TA (gitlab user name `m3510_21f_ta`) and grant them the **Reporter** privileges.