Name: _____

Date: _____

Section: _____

Collaborators: _____

(Collaborators submit their individually written assignments together)

Question:	1	2	3	4	5	Total
Points:	10	20	30	10	5	75
Score:						

Instructor/grader comments:

1. When solving a particular nonlinear equation using Newton's method, a student found that the absolute error of the third iteration was $\epsilon_3 = 10^{-2}$ and the absolute error of the next iteration was $\epsilon_4 = 2 \times 10^{-4}$. Assume that the theoretical convergence rate of the method has been already achieved by the third iteration.

   (a) (5 points) What was the absolute error of the fifth iteration?

   (b) (5 points) How many iterations (in total) would require to achieve the absolute error of the solution smaller than 10^{-14}?

   Present your calculations and explain your reasoning in the space below.

2. (20 points) 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.

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.

Hint: Your formula will include $f(x)$, $f'(x)$, and $f''(x)$.

Matlab

3. (a) (15 points) Write two matlab functions, `conditionone` and `conditioninf` that accept a square matrix as a parameter and calculate condition number of this matrix using one-norm and infinity-norm. Use the code for the functions `hw05p5normone`, `hw05p5norminf`, and `hw03p2inverse` that you wrote for the previous assignments. Place the functions in their own files. Provide the help texts.

   (b) (15 points) Write a matlab script (call it **hw06p3.m**) that compares the results produced by your functions with the result produced matlab's own `cond` function. Use a test set of 5 random matrices of size $n = 500$. Use `for` loop, do not repeat the same code five time! Place the commands `clear`, and `format compact` at the top of your script. Include the help commands for your functions in your script. Describe your conclusions in the project's README file.

4. (10 points) The purpose of the exercise is to motivate students to present the results of calculations in low entropy form.

   Write a script, `hw06p4.m` that initializes the matrix P that you obtained in Problem 3 of homework 5, exactly the way you wrote it as your answer. Calculate the 2-norm of the difference between left and right hand sides of the equation in Problem 3 of homework assignment 5. Place the commands `clear`, and `format compact` at the top of your script.

   To get credit for this problem you must write the matrix exactly as requested and get the norm not larger than $10eps$.

Gitlab

5. (5 points) Create a gitlab project called **hw06** (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.