Assignment #1

Due Wednesday, 5 September 2018, at the start of class

Make sure you have a copy of the textbook:

> Griva, Nash, and Sofer, *Linear and Nonlinear Optimization*, 2nd ed., SIAM Press 2009.

Please read chapter 1 except section 1.7. Read sections 2.1 through 2.4.

DO THE FOLLOWING EXERCISES from page 47 of the textbook:

- Exercise 2.1
- Exercise 2.3
- Exercise 2.4
- Exercise 2.5
- Exercise 2.7

DO THE FOLLOWING PROBLEMS which are based on the notes *Example optimization problems* which were handed out in class:

Problem P1. Solve `calcone`.

Specifically, describe, in brief and well-written english, a strategy (algorithm) for solving this and similar one-variable optimization problems on bounded, closed intervals. Your strategy will necessarily be iterative, and it will not get the exact answer, but otherwise you can solve any way you want. Discuss any issues about the general performance/success of your strategy, emphasizing how it might fail on other problems of this type. (Note that every numerical procedure can be made to fail by careful input (i.e. problem) design. Professionals know how to break what they build.)

Use MATLAB[1] to visualize the function. However, your strategy should *not* be based on human interaction with a figure window. (Why? Because higher-dimensional problems are un-visualizable by humans. Programs must run autonomously to be useful.)

Implement your strategy as a MATLAB[2] code using elementary programming structures such as variables, arrays, `for` loops, `if` conditionals, and such. Do *not* use black boxes, such as the MATLAB commands `fzero`, `fsolve`, `fminsearch`, or `fminbnd`, for this or any other exercises.

Demonstrate at least 6-digit accuracy for the solution to this particular problem.

[1] You may use other languages such as PYTHON or JULIA, but I will only provide examples and solutions in MATLAB/OCTAVE.

[2] Ditto.

Problem P2. Solve `fit`.

Follow essentially the same rules as above: Describe a strategy (algorithm) for solving this and similar problems. Discuss any issues about the general performance/success of your strategy. Implement your strategy as a MATLAB code using elementary programming. Demonstrate 6 digit accuracy. Plot the solution curve on the same graph as the data.

Please *avoid* copying formulas from books or online. *Avoid* recipes you do not understand. Though problems like `fit` are standard in the statistics and linear algebra courses, I want you to start from scratch and understand what you are doing.

Problem P3. Solve `salmon`.

In fact this problem is embarassingly simple to solve, so start by writing a few clear sentences justifying the solution. Then visualize, in 3D and probably with pencil and paper, the set of feasible solutions; mark and label the solution as well. Also use a straightforward substitution to eliminate the equality constraint, and then re-visualize the feasible set and solution in 2D.

Is this problem discrete? Can you reinterpret it as continuous? Comment.

Problem P4. Complete the following classification table for the example problems:

name	discrete	constrained	linear	quadratic	dimension
`calcone`					
`fit`					
`salmon`					
`tsp`					
`glacier`					

Directions. Except for the last column, use a check (✓) if the property is true, leave blank if it is not, or write "NA" for not applicable. In the last column give an integer for the dimension, or ∞, or "NA".

Regarding the "linear" and "quadratic" columns, first check the form of the objective function; is it linear or quadratic or neither? Check linear or quadratic if both the objective function and the constraints have that property.