

Assignment #1

Due Friday, 6 September 2024, 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 (lightly in Section 1.7), Appendix B.4, and Sections 2.1–2.4.

DO THE FOLLOWING EXERCISES from page 47 of the textbook:

- Exercise 2.1
- Exercise 2.3

The problems below are based on the *5 example optimization problems* hand-out, also accessible from the Daily Log tab:

bueler.github.io/opt/assets/worksheets/F24/5exs.pdf

Please do *not* use optimization-type black boxes on these problems. For example, do not use the MATLAB/OCTAVE commands `fzero`, `fsolve`, `fminsearch`, or `fminbnd`. If you use a black box on Homework, or on your Project, then I can/will require that you add an appendix explaining *in detail* how the black box works, before giving any credit for your solution.

PLEASE DO THE FOLLOWING PROBLEMS.

Problem P1. Solve `calc` by implementing a strategy for one-variable optimization on a closed, bounded interval. Your solution strategy should *not* be based on human interaction with a figure window, e.g. repeated mouse-zooming into a figure,¹ but there are many correct strategies. Choose one, implement it, and understand it. Briefly describe it in well-written english. Implement the strategy as a MATLAB/OCTAVE, or other, code. Use elementary programming structures such as variables, arrays, `for` loops, and `if` conditionals, but not any iterative black boxes. Your strategy will necessarily be iterative, but argue that your answer has at least 5-digit accuracy. Visualize the function $f(x)$ and the solution. Discuss any issues about the general performance/success of your strategy, emphasizing how it might fail on problems of this type.²

¹Why is mouse-zooming not a valid strategy? It is because higher-dimensional problems are unvisualizable by humans. Programs *must* run autonomously to be useful.

²Almost every numerical procedure can be made to fail by careful input or problem design. Professionals will know how to break what they build.

Problem P2. Solve `fit`.

Please follow the same rules as above: Describe a strategy (algorithm) for solving this and similar problems. Implement it using elementary programming, and demonstrate 5 digit accuracy on this problem. Plot the solution curve on the same graph as the data. Discuss the success and performance of your strategy.

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

Problem P3. Solve `salmon`.

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

Is this problem discrete? Is it fair to interpret it as continuous? Comment.

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

name	discrete	constrained	linear	quadratic	dimension
<code>calc</code>					
<code>fit</code>					
<code>salmon</code>					
<code>tsp</code>					
<code>glacier</code>					

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? Next note that an optimization problem is called linear or quadratic if *both* the objective function and the constraints have that property. Finally, note that the class of quadratic functions has linear functions as a subset.