About your project

Goal. The goal of the Math 661 project is for you to focus on topics of particular interest, and become more familiar with certain optimization problems and algorithms than is possible with the brief coverage typical of the rest of the course.

Expectations. Your project may be application-driven (*choose the problem(s) first*) or algorithm-driven (*choose the algorithm(s) first*). However, all projects must include both a specific optimization problem and a specific optimization algorithm. You will implement at least one algorithm, i.e. by a code you write in MATLAB/PYTHON/JULIA/etc., and then apply your code to at least one example problem.

Both numerical computation and mathematical analysis are required. For the latter you will analyze the problem(s) and algorithm(s) using theory from the textbook[1] and/or from other references. Once your code is running you should provide some empirical (numerical experimentation) evidence regarding the error and/or performance of your algorithm(s).

Analysis is important as it shows you have absorbed ideas from the course, and because it distinquishes between algorithms. Numerical evidence shows that you understood the algorithm well enough to implement it correctly.

The problem(s) you choose must be in the following form:

$$\min_{x\in\mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{array}{ll} g_i(x) = 0, & i \in \mathcal{E}, \\ g_i(x) \ge 0, & i \in \mathcal{I}, \end{array} \tag{1}$$

(Of course you may replace $\min$ with $\max$.) If your project is algorithm-driven then you must identify which such problems are solved by your algorithms(s).

Form (1) describes a very large class of problems. Your problem(s) must be finite-dimensional, but they may arise from an infinite-dimensional source. The problem(s) must be well-enough understood to allow you to both precisely identify the objective function $f(x)$ and to precisely identify a feasible set S defined by finitely-many equality and inequality constraints $g_i(x)$. It is o.k. if there are no constraints, with $\mathcal{E}$ and $\mathcal{I}$ as empty sets, but I may provide feedback on your Part I proposal which suggests you consider a constrained form of your problem.

Due dates. There are two due dates for the project:

Part I = project proposal: **Part I is due Friday 11 November at the start of class.**

There are no format requirements for this part, but it must be **two pages or less**.

The proposal should precisely say what problem(s) or algorithm(s) you will address. If application-driven it should explain *briefly* where the optimization problem(s) came from. In any case it should briefly motivate your proposed choice(s) of algorithm(s). Several quality references are *expected*; online references are o.k. though many informal online documents are of low quality. Please make specific references

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

to our textbook when that is appropriate. Finally, your proposal should talk though what the complete project will contain, to the degree possible.

Spending at least a few hours on thinking and research at this stage can be very effective, but I suggest that you spend at most 6 hours on Part I.

Part II = actual project: **The completed project is due Monday 12 December at 5pm.**

It should have the format as shown below on page 4. Please use the indicated section headings! The total length **must be 20 pages or less**; I will not accept longer projects. The total time spent on the whole project should be at most 25 hours.

The format expectations can be met by using the LaTeX template posted online at bueler.github.io/opt/projects.html, but you do not have to use LaTeX.

Choosing a topic. One of my jobs will be to help you choose problem(s) and algorithm(s) so that your project has reasonable difficulty. The bigger the scope the easier it is to get lost in the application, the algorithmic details, or difficulties with programming/debugging/analysis. Your Part I proposal allows me to give good feedback on the topic, perhaps a gentle nudge in the direction of a nice variation in topic, or a different analysis to consider, or that you bite off less, and etc.

You may **not** choose a topic which will be adequately covered in lecture. For example, the basic simplex method is not a good topic. However, continuing the example, implementations of the simplex method which respect sparsity, a topic we do not cover, would be a great choice (Chapter 7). Similarly, the classical Newton method is not a good subject, but comparing quasi-Newton methods beyond BFGS, or line search methods beyond backtracking, or trust region methods, would all be good choices (Chapters 11 and 12). Finally, there are many constrained optimization algorithms we will not get to, especially Chapters 8, 10, and 16.

Here are three approaches to choosing a topic if you don't already have one:

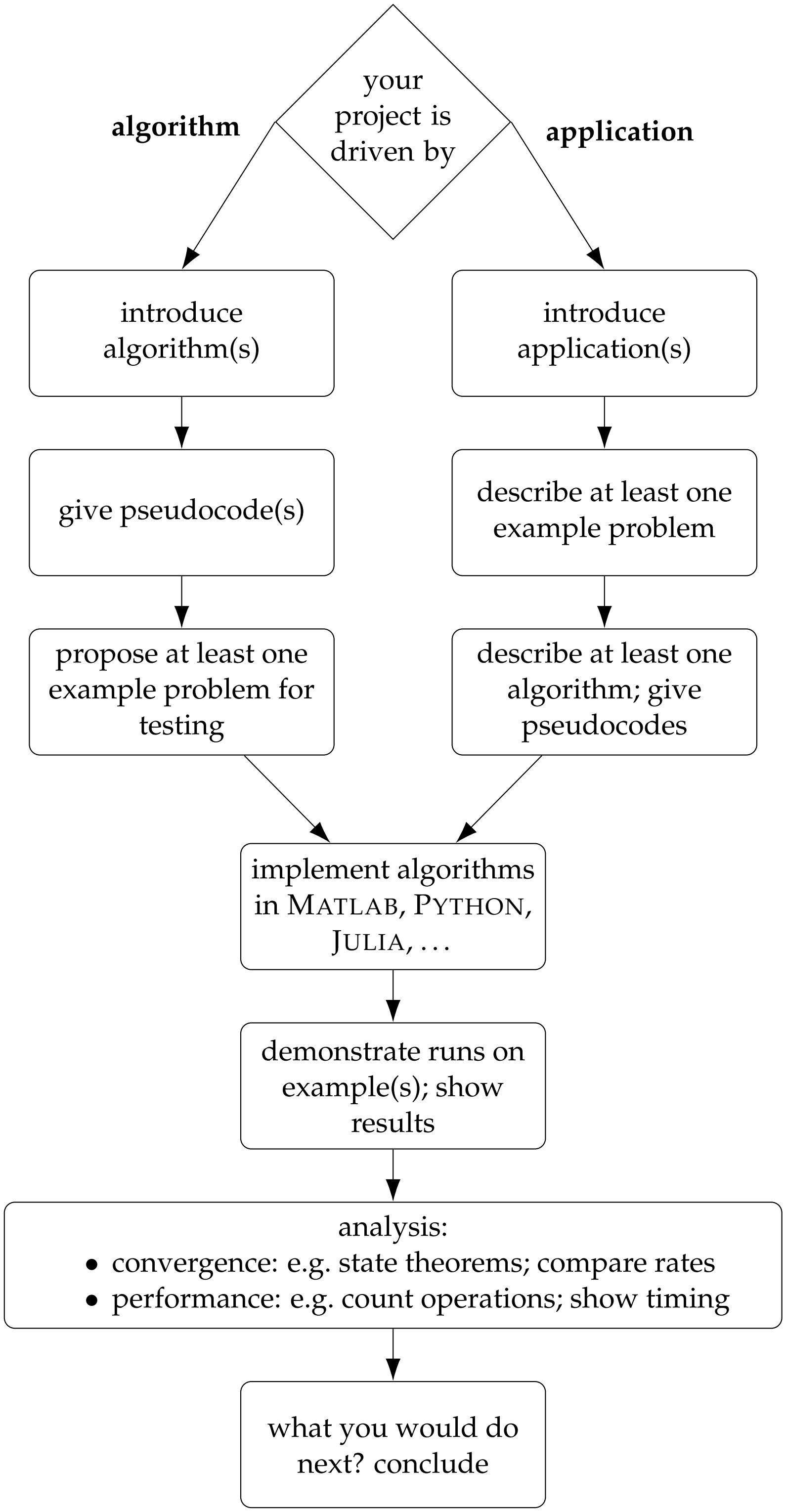
Approach 1: Inspiration from the Wikipedia page on mathematical optimization. See the "Major subfields," "Computational ... techniques", and "Applications" sections.

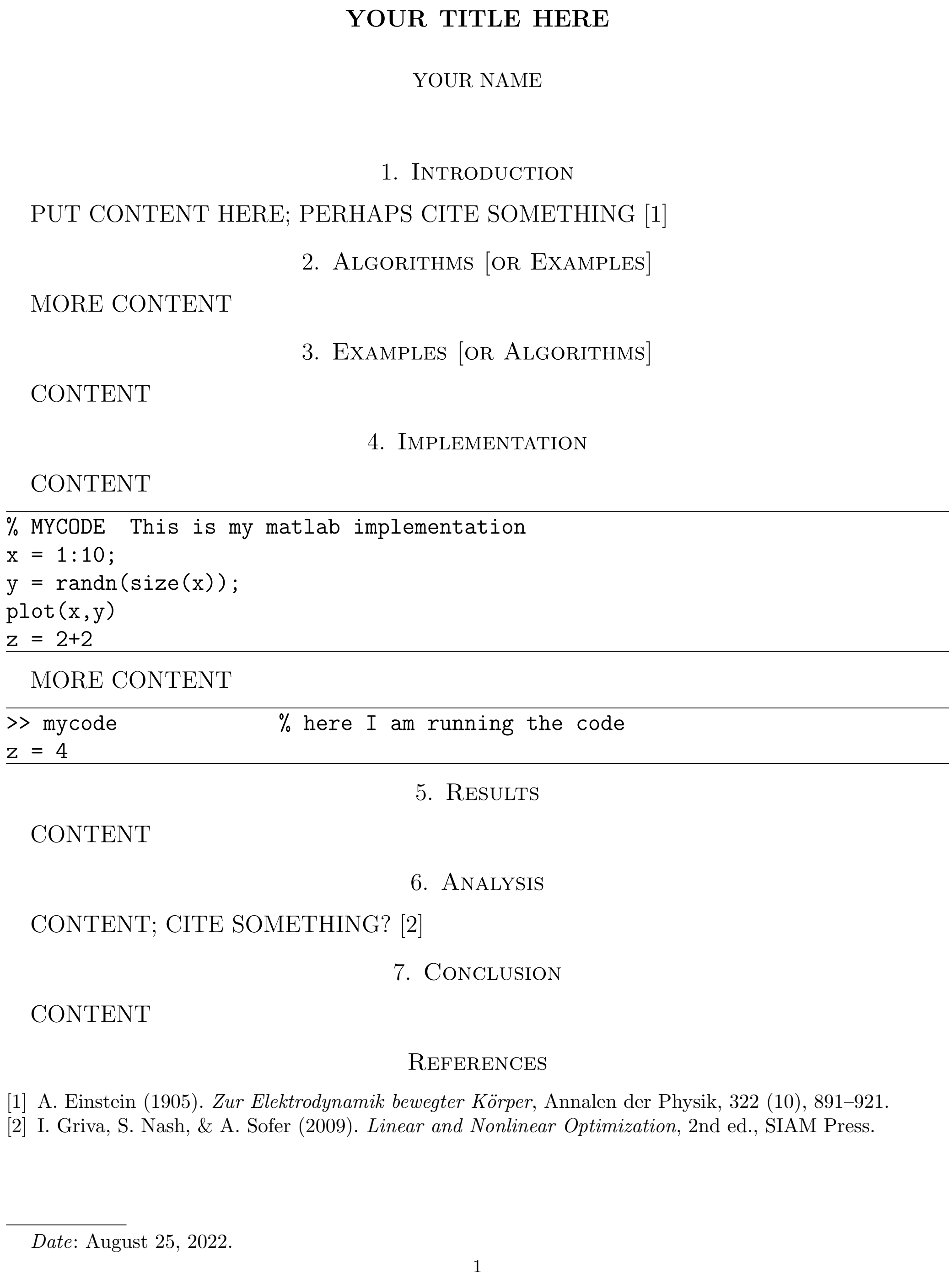
en.wikipedia.org/wiki/Mathematical_optimization

Approach 2: Investigate skipped material from the textbook. Consider section(s) that you find interesting and which we did not cover. However, don't just choose a section at random; the above Wikipedia page is better for starting from scratch.

Approach 3: A topic related to your thesis (if you have one). Please talk to your thesis advisor. It is reasonable to ask "are there optimization problems related to my expected thesis"? Often these are parameter fitting, inverse modeling, or optimal design problems. On the other hand there may be significant algorithms which arise in your field of interest. There may be a paper to read about optimization in your field. Please **do not** cover territory comparable to your whole thesis; instead extract a little part and do it carefully. You should also talk to me, but it may take a while for me to understand the context of your problem.

Structure of the project. Here is a rough flow-chart. It aligns well with the section headings on the next page.


your project is driven by
algorithm
application
introduce algorithm(s)
give pseudocode(s)
propose at least one example problem for testing
introduce application(s)
describe at least one example problem
describe at least one algorithm; give pseudocodes
implement algorithms in MATLAB, PYTHON, JULIA, . . .
demonstrate runs on example(s); show results
analysis:
• convergence: e.g. state theorems; compare rates
• performance: e.g. count operations; show timing
what you would do next? conclude



YOUR TITLE HERE

YOUR NAME

1. Introduction

PUT CONTENT HERE; PERHAPS CITE SOMETHING [1]

2. Algorithms [or Examples]

MORE CONTENT

3. Examples [or Algorithms]

CONTENT

4. Implementation

CONTENT

```
% MYCODE  This is my matlab implementation
x = 1:10;
y = randn(size(x));
plot(x,y)
z = 2+2
```

MORE CONTENT

```
>> mycode             % here I am running the code
z = 4
```

5. Results

CONTENT

6. Analysis

CONTENT; CITE SOMETHING? [2]

7. Conclusion

CONTENT

References

[1] A. Einstein (1905). *Zur Elektrodynamik bewegter Körper*, Annalen der Physik, 322 (10), 891–921.
[2] I. Griva, S. Nash, & A. Sofer (2009). *Linear and Nonlinear Optimization*, 2nd ed., SIAM Press.

Date: August 25, 2022.