

Homework 2

Numerical Analysis Spring 2023

Instructions:

- Due 02/16 at 11:59pm on Gradescope.
- Write the names of anyone you work with on the top of your assignment. If you worked alone, write that you worked alone.
- Show your work.
- Include all code you use as copyable monospaced text in the PDF (i.e. not as a screenshot).
- Do not put the solutions to multiple problems on the same page.
- Tag your responses on gradescope. Each page should have a *single* problem tag. Improperly tagged responses will not receive credit.

Problem 1. Consider the problem of evaluating the function $f : \mathbb{R} \rightarrow \mathbb{R}$ on some domain $D \subset \mathbb{R}$. We can formalize our definition of condition number of f at an input x as

$$\kappa(f, x) = \lim_{\tilde{x} \rightarrow x} \frac{|f(\tilde{x}) - f(x)|}{|\tilde{x} - x|}.$$

- (a) Suppose $f(x)$ is differentiable at x . How does $\kappa(f, x)$ relate to $f'(x)$?
- (b) Let $f(x) = \sqrt{x}$. What is $\kappa(f, 4)$? Show $\kappa(f, 0) = \infty$.
- (c) Suppose $g(x) = 1000f(x)$. How do $\kappa(f, x)$ and $\kappa(g, x)$ relate?
- (d) Suppose $g(x) = f(x) + 1000$. How do $\kappa(f, x)$ and $\kappa(g, x)$ relate?
- (e) Suppose $g(x) = f(x - 1000)$. How do $\kappa(f, x)$ and $\kappa(g, x + 1000)$ relate?

Problem 2. We can also define a “relative” condition number

$$\kappa_{\text{rel}}(f, x) = \lim_{\tilde{x} \rightarrow x} \frac{|f(\tilde{x}) - f(x)|/|f(x)|}{|\tilde{x} - x|/|x|}.$$

- (a) Suppose $g(x) = 1000f(x)$. How do $\kappa_{\text{rel}}(f, x)$ and $\kappa_{\text{rel}}(g, x)$ relate?
- (b) Suppose $g(x) = f(x) + 1000$. How do $\kappa_{\text{rel}}(f, x)$ and $\kappa_{\text{rel}}(g, x)$ relate?
- (c) Suppose $g(x) = f(x - 1000)$. How do $\kappa_{\text{rel}}(f, x)$ and $\kappa_{\text{rel}}(g, x + 1000)$ relate?
- (d) Think about the implications of your responses to these problems, and then describe a real-life situation in which it makes more sense to use the relative condition number than the regular condition number.

Problem 3. Describe whether the following problems/tasks are well-conditioned or not. Justify each of your responses.

- (a) Problem/task: You are given a vector $[x_1, x_2]$ and must compute the solution $[z_1, z_2]$ to the linear system of equations

$$2.3z_1 - 1.01z_2 = x_1, \quad 2.31z_1 - 1.00z_2 = x_2.$$

Example inputs/outputs:

input x	solution $f(x)$
$[1, 2]$	$[30.816, 69.184]$
$[1, 1]$	$[0.30211, -0.302115]$

- (b) Problem/task: You are given function $h : [-1, 1] \rightarrow \mathbb{R}$ and must return $\int_{-1}^1 g(s)ds$. Here you should interpret $\tilde{h} \approx h$ to mean $\tilde{h}(s) \approx h(s)$ for all $s \in [-1, 1]$.

Example inputs/outputs:

input x	solution $f(x)$
$h(s) = 1$	2
$h(s) = s^2$	2/3
$h(s) = \sin(s)$	0





- (c) Problem/task: You're designing a self-driving car. As you approach an intersection, you are given a color image of the stoplight. You must determine whether it is green or red; i.e. if the car should continue through the intersection or stop.

Example inputs/outputs:

input x	solution $f(x)$
	green
	red
	green
	red

- (d) Problem/task: You're designing a self-driving car. As you approach an intersection, you are given a greyscale image of the stoplight. You must determine whether it is green or red; i.e. if the car should continue through the intersection or stop.

Example inputs/outputs:

input x	solution $f(x)$
	green
	red
	green
	red

Images from: Brooklyn 4K - Night Drive and Jazz New York City Drive

Problem 4. Consider the problem of adding n numbers. That is $f(x_1, \dots, x_n) = x_1 + x_2 + \dots + x_n$.

There are many ways we could implement an algorithm for this problem. Perhaps the simplest is

```
import numpy as np

def add_list(x,n):
    s = np.zeros(1,dtype=x.dtype)
    for i in range(n):
        s = s + x[i]
    return s
```

Consider $x = [10^8, 1, 1, \dots, 1]$ for $n = 34412$. We can set this problem up as

```
n = 34412
x = np.ones(n,dtype=np.float32)
x[0] = 1*10**8

add_list(x,n)
```

Note here we are using 32 bit floating point arithmetic throughout.

- (a) What is the correct answer to this problem, and what does the algorithm output?
- (b) Suppose instead we implement an algorithm

```
def add_list_sorted(x,n):
    z = np.sort(x)
    s = np.array(1,dtype=x.dtype)
    for i in range(n):
        s = s + z[i]
    return s
```

What is the output of `add_list_sorted(x,n)`?

- (c) Explain why the two algorithms have different outputs.

Problem 5. In class (and the notes) we saw that our implementation of a triangular solver was faster than numpy's `np.linalg.solve`.

Scipy has a solver `sp.linalg.solve_triangular` for triangular systems which uses LAPACK's triangular solver.

Add this solver to the runtime comparison of `np.linalg.solve` and our triangular solver in Chapter 7 of the notes (code is online in Google Drive).

Include the new figure showing the runtime scaling of all three algorithms. How does scipy's triangular solver compare?

Remember to include your code as well as the figure.

Note also that the timings don't work that well in colab, so it is preferable to run locally if possible.