CSE 6643 Homework 1

Schäfer, Spring 2023

Deadline: Jan. 19 Thursday, 8:00 am

- There are 2 sections in grade scope: Homework 1 and Homework 1 Programming. Submit your answers as a PDF file to Homework 1 (report the results that you obtain using programming by using plots, tables, and a description of your implementation like you would when writing a paper.) and also submit your code in a zip file to Homework 1 Programming.

- Programming questions are posted in Julia. You are allowed to use basic library functions like sorting, plotting, matrix-vector products etc, but nothing that renders the problem itself trivial. Please use your common sense and ask the instructors if you are unsure. You should never add additional packages to the environment.

- Late homework incurs a penalty of 20% for every 24 hours that it is late. Thus, right after the deadline, it will only be worth 80% credit, and after four days, it will not be worth any credit.

- We recommend the use of LaTeX for typing up your solutions. No credit will be given to unreadable handwriting.

- List explicitly with whom in the class you discussed which problem, if any. Cite all external resources that you were using to complete the homework. For details, consult the collaboration policy in the class syllabus on canvas.

1 Basics [25 pts]

(a) [5 pts]

Suppose that v_1, v_2, v_3, v_4 is a basis of the vector space $V \subset \mathbb{R}^n$. Prove that the list

$$v_1 + v_2, v_2 + v_3, v_3 + v_4, v_4 \tag{1}$$

is also a basis of V.

(b) [10 pts]

For U a subspace of the vector space $V \subset \mathbb{R}^n$ with $\dim(U) = \dim(V)$. Prove that $U = V$.

(c) [10 pts]

Show that the subspaces of $\mathbb{R}^3$ are precisely $\{0\}$, $\mathbb{R}^3$, all lines in $\mathbb{R}^3$ through the origin, and all planes in $\mathbb{R}^3$ through the origin.

2 Norm Equivalencies [25 pts]

In a finite-dimensional space, all norms are equivalent. In this problem, you will be asked to verify this theorem for some special norms. Prove the following inequalities.

Let $x \in \mathbb{R}^n$ be an n-dimensional vector. Let $A \in \mathbb{R}^{m \times n}$ be an $m \times n$ matrix. Then:

(a) [10 pts]

$$\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty.$$

(b) [7.5 pts]

$$\|A\|_\infty \leq \sqrt{n}\|A\|_2.$$

(c) [7.5 pts]

$$\|A\|_2 \leq \sqrt{m}\|A\|_\infty.$$

3 Perturbing [25 pts]

For $u, v \in \mathbb{K}^m$, the matrix $A := I + uv^*$ is called a *rank-one* perturbation of the identity.

(a) [15 pts]

Show that if A is nonsingular, then its inverse has the form $A^{-1} = I + \alpha uv^*$ for some scalar α, and give an expression for α.

(b) [5 pts]

For what u and v is A nonsingular?

(c) [5 pts]

If A is singular, what is $\text{null}(A)$?

4 Julia [25 pts]

This class will use the Julia programming language. This first programming assignment is not very involved, and primarily serves to introduce you to Julia. The homework contains a folder `HW1_CODE` with four files `HW1_your_code.jl`, `HW1_driver.jl`, `Manifest.toml`, `Project.toml`. **Only the first file should be modified!** `HW1_driver.jl` is used to execute your code, while `Manifest.toml` and `Project.toml` tell Julia which versions of packages to use, to ensure reproducibility. You should not install additional packages to solve the homework.

(a) Installing Julia [5 pts]

Install Julia version 1.8 on your computer, and make yourself familiar with its basic functionality.

(b) Matrix-vector-multiplication [5 pts]

Complete the function `u_is_A_times_v!(u, A, v)` in `HW1_your_code.jl` that overwrites the input vector `u` with the product of the input matrix `A` and the input vector `v`.

(c) Matrix-matrix-multiplication [5 pts]

Complete the function `A_is_B_times_C!(A, B, C)` in `HW1_your_code.jl` that overwrites the input matrix `A` with the product of the input matrices `B` and `C`.

(d) Testing [5 pts]

From the directory `HW1_CODE`, run the command

```
julia --project=. HW1 driver.jl
```

to test your code. Here, the `--project=.` tells Julia to use `Manifest.toml` and `Project.toml` determine which version (if any) of packages to use. Make sure that your code passes all `@assert` statements, which test your functions against Julia's built-in functions.

(e) Optimization [5 pts]

Make sure that your code does not allocate any memory, as evidenced by the `@btime` calls in `HW1_driver.jl` returning (`0 allocations: 0 bytes`). This is important for performance reasons since allocating memory may be orders of magnitudes slower than floating-point arithmetic. Now try reordering the for-loops in your implementations from parts (a) and (b) and observe the resulting timings provided by `@btime`. Which order leads to the best and worst performance? What are the corresponding wall-clock times as measured by `@btime`?