CSE 6643 Homework 2

Schäfer, Spring 2023

Deadline: Feb. 3 Friday, 8:00 am

- There are 2 sections in grade scope: Homework 2 and Homework 2 Programming. Submit your answers as a PDF file to Homework 2 (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 2 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 Less positive [12.5 pts]

Consider a symmetric positive definite matrix $\boldsymbol{A}$ separated into subblocks according to

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{1,1} & \boldsymbol{A}_{1,2} \\ \boldsymbol{A}_{2,1} & \boldsymbol{A}_{2,2} \end{pmatrix}.$$

Show that the matrix $\boldsymbol{A}_{2,2} - \boldsymbol{A}_{2,1}\boldsymbol{A}_{1,1}^{-1}\boldsymbol{A}_{1,2}$ is symmetric and positive definite.

2 Just one [12.5 pts]

Consider a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times m}$, with (unpivoted) LU-factorization given by $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$. Propose an algorithm for computing the (i, j)-th entry of $\boldsymbol{A}^{-1}$ using only $\mathcal{O}\Big((m+1-j)^2 + (m+1-i)^2\Big)$ floating point operations.

3 SVD [25 pts]

In class, we have reminded ourselves of the SVD of square matrices. Use the recommended literature from the class syllabus to catch up on the SVD applied to non-square matrices.

We denote as $\sigma_{\min}$ and $\sigma_{\max}$ the smallest and largest singular value of a matrix and by λ_i its i-th eigenvalue. We consider general matrices $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times l}$. Show that the following results hold

(a) [5 pts]

$$\sigma_{\max}(\boldsymbol{A})\|\boldsymbol{x}\| \geq \|\boldsymbol{A}\boldsymbol{x}\|_2, \forall \boldsymbol{x} \in \mathbb{R}^n$$

(b) [5 pts]

$$m \geq n \Rightarrow \|\boldsymbol{A}\boldsymbol{x}\|_2 \geq \sigma_{\min}(\boldsymbol{A})\|\boldsymbol{x}\|_2, \forall \boldsymbol{x} \in \mathbb{R}^n$$

(c) [5 pts]

$$m = n \Rightarrow \sigma_{\min}(\boldsymbol{A}) \leq |\lambda_i(\boldsymbol{A})| \leq \sigma_{\max}(\boldsymbol{A}), \forall i$$

(d) [5 pts]

$$\sigma_{\max}(\boldsymbol{A}) = 1/\sigma_{\min}\big(\boldsymbol{A}^{-1}\big), \text{ if } \boldsymbol{A}^{-1} \text{ exists}$$

(e) [5 pts]

Find the most general conditions on m, n, and l, under which $\sigma_{\min}(\boldsymbol{A}\boldsymbol{B}) \geq \sigma_{\min}(\boldsymbol{A})\sigma_{\min}(\boldsymbol{B})$.

4 Inverses [25 pts]

We consider $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ and assume that $\boldsymbol{A}^{-1}$ exists. You may use without proof that if $\boldsymbol{A}$ is symmetric, its largest and smallest eigenvalues are characterized as

$$\lambda_{\max}((A)) = \sup_{\boldsymbol{x}\in\mathbb{R}^m\setminus\{\boldsymbol{0}\}} \frac{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\|\boldsymbol{x}\|_2^2}$$

and

$$\lambda_{\min}((A)) = \inf_{\boldsymbol{x}\in\mathbb{R}^m\setminus\{\boldsymbol{0}\}} \frac{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\|\boldsymbol{x}\|_2^2}$$

(a) [5 pts]

Show that

$$\big\|\boldsymbol{A}^{-1}\big\|_2^{-1} = \inf_{\boldsymbol{x}\in\mathbb{R}^m\setminus\{\boldsymbol{0}\}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} \tag{1}$$

(b) [5 pts]

Show that

$$\big\|\boldsymbol{A}^{-1}\big\|_2^{-1} \geq \inf_{\boldsymbol{x}\in\mathbb{R}^m\setminus\{\boldsymbol{0}\}} \frac{|\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}|}{\|\boldsymbol{x}\|_2^2} \tag{2}$$

(c) [5 pts]

Show that for symmetric matrices $\boldsymbol{M}, \boldsymbol{N} \in \mathbb{R}^{m\times m}$,

$$\lambda_{\min}(\boldsymbol{M}+\boldsymbol{N}) \geq \lambda_{\min}(\boldsymbol{M}) + \lambda_{\min}(\boldsymbol{N}) \tag{3}$$

(d) [5 pts]

In the same setting as (c), show that

$$\lambda_{\max}(\boldsymbol{M}+\boldsymbol{N}) \leq \lambda_{\max}(\boldsymbol{M}) + \lambda_{\max}(\boldsymbol{N}) \tag{4}$$

(e) [5 pts]

For $\boldsymbol{A}$ symmetric positive definite, show that

$$\|\boldsymbol{A}^{-1}\|_2^{-1} = \inf_{\boldsymbol{x}\in\mathbb{R}^m\setminus\{\boldsymbol{0}\}} \frac{\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}}{\|\boldsymbol{x}\|_2^2}$$

5 Oblivion [25 pts]

(a) [5 pts]

Have a look at the function `add_to_A_B_times_C!` in `HW2_your_code.jl`. This function adds the product of the input variables `B` and `C` to the input variable `A`. It is written such as to use the optimal loop order, and employs multiple optimizations by means of the `@turbo` macro.

Go to section (a) of the file `HW2_driver.jl` to see how this function performs when compared to the built-in `mul!` function. If necessary, adapt the size of the matrix to your system. Hand in the file `performance_per_size.pdf` produced by the code with your homework.

(b)locked [5 pts]

Now implement a blocked/tiled variant of `add_to_A_B_times_C!` that takes an integer `bks` as a fourth input. This method should perform the matrix multiplication by calling the original `add_to_A_B_times_C!` on blocks of size (roughly) `bks` times `bks`. Again, your code should be correct and avoid memory allocations as checked by section (b) of `HW2_driver.jl`.

(c) [5 pts]

Use section `HW2_driver.jl` to try the performance of the blocked algorithm for different matrix sizes and block sizes. Describe your results, and for the most interesting set of parameters, hand in the resulting plot with your homework. Describe and explain the results.

(d) oblivious [5 pts]

We can sometimes obtain better performance by using a hierarchical algorithm. Complete the skeleton for `oblivious_add_to_A_B_times_C!` to obtain an algorithm that equally divides each of its input matrices into four parts and then recurses on the resulting eight subproblems, until one of the problems reaches a size below `bks`. Again, make sure that your algorithm does not allocate memory and is correct using part (d) `HW2_driver.jl`.

(e) [5 pts]

Use part (e) of `HW2_driver.jl` to benchmark your new algorithm for different values of `bks`. Are you able to obtain an improvement over the blocked algorithm? Hand in the figure produced by part (e) of `HW2_driver.jl` for what you consider the most interesting set of parameters. Algorithms of this type are called "cache-oblivious." Explain why this is the case and what could be the theoretical advantages of this particular cache-oblivious algorithm.