Numerical Linear Algebra
Summer 2023-2024

אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

School of Electrical and
Computer Engineering

Final Exercise

Submission deadline: November 2, 2024

<u>Important! How to Not Lose Points on Style and Presentation</u>

Follow these guidelines to make it easier to grade your work favorably: (1) Use clear idiomatic English. (2) Use consistent fonts for all notations. Use distinct fonts for vectors, matrices, scalars, and non-equation text. (3) Make your plots clear and informative. Use legends, axis titles, and units. (4) Make sure the data in the figure do not cluster or lose meaning – distribute data evenly and informatively within the figures: use *log-log* or *semi-log* scales wherever wide dynamic ranges of values are of interest, especially when values approaching zero are presented. (5) Use “logspace”, rather than “linspace”, when looping over a broad dynamic ranges of values. (6) *Do not* connect discrete data points with lines – this introduces misleading visual information. (7) Instead, add trendlines and asymptotes for showing scaling. (8) When comparing plots, put them *on the same figure*. Do not leave it to the reader to visually compare figures placed side-by-side or, worse, in separate pages or with different value ranges. (9) Do not just describe your results! Explain and interpret them. (10) Lastly, do not include any code in your report!

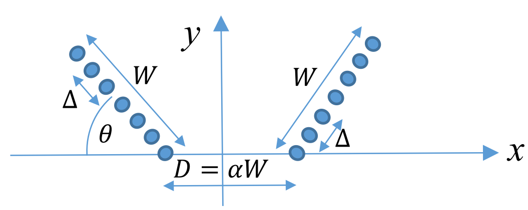
<u>Use your personal ID number to determine your tasks:</u>

- <u>Rightmost digit (for singles): odd (£), even (££). Couples do both and compare them.</u>

<u>Background</u>

This exercise studies the interaction between sets of small transmitter and receiver “antennas”.

Consider two arrays of N small “antennas” each, placed symmetrically in a 2-D space. The antennas are uniformly spaced along lines of width W tilted at an angle θ above the x axis, with a spacing D between their closest antennas ($D > 0$). The spacing between the antennas is of $\Delta = \lambda/10$ (where λ is the wavelength). We denote $\mathbf{r}_m^{\{\mathrm{r,t}\}} = (x_m, y_m)$ the location of the mth {receiver, transmitter} antenna.



Each receiver antenna is influenced by all transmitter antennas. Denoting $t \in \mathbb{C}^N$ the vector of transmitter signal weights, the vector $r \in \mathbb{C}^N$ of received signals is related to s via a matrix $A \in \mathbb{C}^{N\times N}$, such that

$$r = At.$$

Here, the interactions between pairs of receivers and transmitters are modeled as

$$A_{mn} = e^{-jk|\mathbf{r}_m^{\mathrm{r}} - \mathbf{r}_n^{\mathrm{t}}|}/|\mathbf{r}_m^{\mathrm{r}} - \mathbf{r}_n^{\mathrm{t}}|^{1/2},$$

where $k = 2\pi/\lambda$ is the wave number.

Part I – Preliminaries

(a) Implement a routine that, for given input parameters $\lambda, W, \theta, \alpha$, constructs the matrix A. Plot the matrix (imagesc of absolute values) for the choice $\lambda = 1$, $W = 4\lambda$, $\theta = \pi/2$, $D = W$. Comment on its structure. Compute its singular value decomposition (SVD) and plot the singular values. Comment on the behavior of the singular values.

(b) Increase N by repeatedly doubling W for the case $\theta = \pi/2$ and $\lambda = 1$. Study the complexity of the SVD computation, the rank for relative error truncation thresholds $\tau = 10^{-2}, 10^{-5}, 10^{-8}$, and the condition number of A as a function of N for the following cases: $D = 4\lambda$, $D = W$, $D = W^2/\lambda$. Plot your results for the complexity. Add asymptotic (not polynomial fit for the entire data) lines for the behavior. Does the behavior of the complexity match your theoretical expectations?

(c) Plot all other results computed in (b), add asymptotic lines, and interpret each of the plots separately. Relate the observed results to the behavior of the singular value plots for each case.

(d) Summarize your findings regarding the rank's dependence on the various problem parameters investigated: try to come up with an expression of the form $\mathcal{R} = C(\tau)f(W,D)$ that describes the asymptotic scaling of the rank. Do find f. Do not find $C(\tau)$ but describe how it should behave with τ.

Part II – LR Approximation

(e) Implement a routine that, for a given τ, computes a low-rank (LR) approximation of A. For the case $\lambda = 1$, $W = 128\lambda$, $\theta = 0$, $D = W$, compute a LR approximation $\tilde{A}(\tau) = UV^*$. Store $\tilde{A}(\tau)$ explicitly (like we said one should never do) and compute the error directly by subtracting $\tilde{A}(\tau)$ from A (like you did not know the theoretical error bound already). Plot the rank, the 2-norm relative error, and the time for computing it, as a function of $\tau \in 10^{[-10:-1]}$. Describe the results and explain them. Do they match your expectations?

(f) The follow algorithms for fast relative error estimation were proposed:

<u>(£)</u>

1. Set $l := 1$, $n := 5$, $\tau_\varepsilon = 10^{-1}$
2. At random, draw two sets of indices $\boldsymbol{i}_{\text{row}} \in \mathbb{Z}^n$ and $\boldsymbol{i}_{\text{col}} \in \mathbb{Z}^n$ such that each vector contains a set of non-repeating indices in the range $[1, N]$.
3. Extract the inteesection submatrix $A_l := A(\boldsymbol{i}_{\text{row}}, \boldsymbol{i}_{\text{col}})$ from A.
4. Use $U(\boldsymbol{i}_{\text{row}}, :)$ and $V(\boldsymbol{i}_{\text{col}}, :)$ to compute the submatrix $\tilde{A}_l := \tilde{A}(\boldsymbol{i}_{\text{row}}, \boldsymbol{i}_{\text{col}})$.
5. Compute the 2-norms $\|A_l - \tilde{A}_l\|_2$ and $\|A_l\|_2$ and $\varepsilon_l = \|A_l - \tilde{A}_l\|_2 / \|A_l\|_2$.
6. If $l \neq 1$, compute $e_l = (\varepsilon_l - \varepsilon_{l-1})/\varepsilon_l$. If $e_l < \tau_\varepsilon$, terminate and return ε_l.
7. Update $l := l + 1$, $n := 2n$, and go to Step 2.

<u>(££)</u>

1. Set $l := 1$, $n := 5$, $\tau_\varepsilon = 10^{-1}$
2. At random, draw two sets of indices $\boldsymbol{i}_{\text{row}} \in \mathbb{Z}^n$ and $\boldsymbol{i}_{\text{col}} \in \mathbb{Z}^n$ such that each vector contains a set of non-repeating indices in the range $[1, N]$.
3. Build a vector $\mathbf{a}_l$ such that $a_j = A(i_{\text{row}}(j), i_{\text{col}}(j))$.
4. Compute a vector $\tilde{\mathbf{a}}_l$ with $\tilde{a}_j = U(i_{\text{row}}(j), :)\big(V(i_{\text{col}}(j), :)\big)^*$
5. Compute the 2-norms $\|\mathbf{a}_l - \tilde{\mathbf{a}}_l\|_2$ and $\|\mathbf{a}_l\|_2$ and $\varepsilon_l = \|\mathbf{a}_l - \tilde{\mathbf{a}}_l\|_2 / \|\mathbf{a}_l\|_2$.
6. If $l \neq 1$, compute $e_l = (\varepsilon_l - \varepsilon_{l-1})/\varepsilon_l$. If $e_l < \tau_\varepsilon$, terminate and return ε_l.
7. Update $l := l + 1$, $n := 2n$, and go to Step 2.

Explain the operation of the algorithm and the meaning of ε_l, e_l and τ_ε.

(g) Implement the algorithm in (f) and run it for the parameters in (e). Compare ε_l at the output, for the various τ values, to the relative errors in (e). Compare the computation times as a function of τ to those in (e). How does n depends on τ? How does the computation time depend on it?

Couples: compare the two algorithms: in particular, discuss the difference in computation time required for achieving the error estimate as a function of τ.

Part III – Fast LR Approximation

(h) A fast method for computing the LR approximation is proposed:

Alg. Randomized LR approximation

1. Set $\mathcal{B}_0$ to some value. Set $\widehat{U} = [\,], \widehat{B} = [\,]$. Compute $\|A\|_F$. $l = 1$
2. Construct a matrix $G \in \mathbb{C}^{N \times \mathcal{B}_0}$ of complex Gaussian i.i.d. random values.
3. Compute $M = AG$.
4. Compute the SVD $U\Sigma V^* = M$ using the SVD.
5. $B = U^*(:,1:\mathcal{B}_0)A$
6. Update $\widehat{U}: [\widehat{U} \quad U(:,1:\mathcal{B}_0)]$, $\widehat{B}: = [\widehat{B} \quad B]$;
7. If $\|B\|_\mathrm{F} \leq \tau\|A\|_\mathrm{F}$, terminate and return the factors $\widehat{U}$ and $\widehat{B}$ (that form $\widehat{U}\widehat{B} \approx A$) and $\mathcal{R}_l = l\mathcal{B}_0$.
8. Update $A \coloneqq A - U(:,1:\mathcal{B}_0)B$, $l: = l + 1$
9. Go to step 2.

Explain the operation of the algorithm. Discuss the meaning of the parameter $\mathcal{B}_0$.

(i) Set $\mathcal{B}_0 = 5$ and run the algorithm for the parameters in (e). That is, replace the SVD-based LR approximation with the fast algorithm. Present the computation time as a function of τ. For each τ, also plot the actual relative error of the LR approximation. Is it exactly τ or is it lower/higher? Discuss the results.

(j) Replace Step 7 of the proposed algorithm with the pertinent algorithm(s) in (f) as an assessment of the LR approximation relative error. Compare the computation times and LR approximation errors to those of the implementation in (i).

Good luck!