Labwork 4: Applied Linear and Convex Optimization

Instructor: Parikshit Pareek

General Instructions:

- You may use standard optimization modeling languages such as `CVXPY` (in Python) or `JuMP` (in Julia) along with a solver like `ECOS`, `SCS`, `OSQP`, or `Gurobi`.
- Include comments in your code explaining the mathematical formulation.
- For every plot, ensure axes are labeled properly and include legends where multiple curves are shown.

Part 1: Curve Fitting & Distance Metrics (Warm-up)

Objective: Understand overdetermined systems and how different norms (L_2, L_1, L_∞) affect the optimal solution.

Task 1.1: Data Loading: Load the provided Book.csv file. Assume the first column represents x values and the second column represents y values. Create a scatter plot of the data.

Task 1.2: Standard Least Squares (L_2-norm): We want to fit a polynomial of degree $k = 3$ of the form $f(x) = u_1x^3 + u_2x^2 + u_3x + u_4$.

- Construct the Vandermonde-like matrix A such that $Au \approx y$.
- Solve for u using the closed-form Normal Equations $u = (A^TA)^{-1}A^Ty$. Plot the resulting polynomial against the scatter plot.

Task 1.3: Alternative Norms (L_1 and L_∞): Minimizing the sum of squared errors (L_2) is just one measure of distance. Now, formulate optimization models to minimize the L_1-norm and L_∞-norm of the residuals.

- *Instruction:* Do *not* use built-in norm minimization functions. Formulate them manually using auxiliary variables (e.g., for L_∞, minimize t subject to $-t \leq Au - y \leq t$).
- Solve using CVXPY / JuMP and plot the L_1, L_2, and L_∞ polynomials on the same figure. Add a legend to distinguish them.

Part 2: Time-Series Modeling (Let Get the Ball Rolling)

Objective: Formulate a Moving Average (Autoregressive) model by constructing appropriate matrix structures.

Task 2.1: The Model: Consider the y data from `Book1.csv` as a sequential time series (ignoring x). We want to predict y_t using the past k observations:

$$y_t \approx w_1 y_{t-1} + w_2 y_{t-2} + \cdots + w_k y_{t-k}$$

Task 2.2: Formulate the Matrix: Write a script to generate the matrix A and vector b for a given window size k. (*Note: Your usable time points will start from index* $k+1$ *to* T).

Task 2.3: Least Squares Fit: Solve for the weights w using standard least squares for $k = 3$. Plot the true output y against your predicted output $\hat{y} = Aw$. Compute the norm of the error.

Task 2.4: Window Size Trade-off:

- Loop over window sizes from $k = 1$ up to $k = 40$.
- For each k, compute the L_2 error of the moving average model.
- Plot the "Error as a function of window size". What do you observe about the relationship between degrees of freedom and the residual error?