# Labwork 5: 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: Underdetermined Systems & Regularization (Run Forex Run)

**Objective:** Understand the geometry of least squares, the pseudo-inverse, and Tikhonov ($L_2$) regularization.

**Task 3.1: Create an Underdetermined System:** Generate synthetic data by picking $m = 5$ points $x$ evenly spaced in the interval $[-1, 1]$, and compute their corresponding $y$ values using a simple function like $y = \sin(\pi x)$. Attempt to fit a polynomial of degree $k = 8$ (which requires finding $n = 9$ coefficients).

- Construct the $m \times n$ Vandermonde matrix $A$.
- Since $m < n$, the linear system $Au = y$ is underdetermined and has an infinite number of solutions (an affine subspace).

**Task 3.2: Minimum Norm Solution:**

- Formulate the equality-constrained optimization problem: $\min \|u\|_2^2$ subject to $Au = y$.
- Solve this using CVXPY/JuMP.
- Compute the analytical solution using the pseudo-inverse: $u = A^T(AA^T)^{-1}y$ and verify it matches your optimizer's output.

**Task 3.3: Regularization Trade-off:**

- Formulate the unconstrained regularized objective: $\min \|Au - y\|_2^2 + \lambda\|u\|_2^2$.
- Solve this for $\lambda \in \{10, 1, 0.1, 0.001\}$.
- **Analytical Verification:** Verify the class theorem experimentally. Show that as $\lambda \to 0$, the solution of the regularized problem approaches the minimum norm solution found in Task 3.2.

## Part 4: Trajectory Optimization & Pareto Curves (Assemble your Avengers!)

**Objective:** Formulate a multi-objective optimal control problem (Hovercraft model) and analyze optimal trade-offs via a Pareto curve.

**Task 4.1: Kinematic Setup:** Consider a 2D hovercraft. Let $x_t \in \mathbb{R}^2$ be the position and $v_t \in \mathbb{R}^2$ be the velocity. The control input is thrust $u_t \in \mathbb{R}^2$.

- Dynamics: $x_{t+1} = x_t + v_t$ and $v_{t+1} = v_t + u_t$.
- Initial state: $x_1 = [0, 0]^T, v_1 = [0, 0]^T$.

**Task 4.2: Target Waypoints:** The hovercraft must pass as close as possible to the following waypoints at specific times $T_i$:

- $T_1 = 20 \rightarrow$ Waypoint $1 = [4, 3]^T$
- $T_2 = 50 \rightarrow$ Waypoint $2 = [6, 0]^T$
- $T_3 = 60 \rightarrow$ Waypoint $3 = [1, 1]^T$

**Task 4.3: Multi-Objective Formulation:** Define two conflicting objectives:

- $J_1 = \sum_t \|u_t\|_2^2$ (Minimize Fuel / Control Effort)
- $J_2 = \sum_i \|x_{T_i} - \text{Waypoint}_i\|_2^2$ (Minimize Waypoint Mismatch)

Combine them into a single objective using a trade-off parameter $\lambda$: $\min J_1 + \lambda J_2$.

**Task 4.4: Trajectory Generation:** Write a function `solve_hovercraft(λ)` that sets up the variables and constraints in your solver, minimizes the combined objective, and returns $(J_1, J_2, x_{\text{trajectory}})$.

- Test your function with $\lambda = 0.01$ and plot the resulting 2D spatial trajectory ($x_t^{(1)}$ vs $x_t^{(2)}$). Mark the waypoints clearly with red dots.

**Task 4.5: Pareto Trade-off Curve:**

- Generate a logarithmically spaced array of 30 $\lambda$ values ranging from $10^{-5}$ to $10^1$.
- Run `solve_hovercraft(λ)` for each value and store the resulting $J_1$ and $J_2$.
- Plot $J_2$ (Waypoint Mismatch) on the Y-axis vs $J_1$ (Fuel Cost) on the X-axis.
- **Reflection:** Identify the convex shape of the curve. Mark a region on the plot that represents "Feasible but strictly suboptimal" points, as discussed in the lecture notes. Explain why a designer would strictly pick a point on the line rather than above it.