Lab 1 – Optimization

Use MATLAB to:

- Visualize objective functions and constraints in 2D/3D.
- Determine minimum and maximum points of constrained problems.
- Find and classify stationary points (local min, local max, inflection/saddle points).

Part A – Graphical optimization (Arora, Problems 3.11, 3.12, 3.13)

For each problem below:

- Use MATLAB or Python to create suitable plots (e.g. surface and/or contour plots) that illustrate the objective function and constraint(s).
- Identify and report the minimum and maximum points of the objective function subject to the given constraint(s).

1. Objective function:
   $f(x, y) = 2x^2 + y^2 - 2xy - 3x - 2y$
   Constraints:
   $y - x \leq 0$
   $x^2 + y^2 - 1 = 0$

2. Objective function:
   $f(x, y) = 4x^2 + 3y^2 - 5xy - 8x$
   Constraint:
   $x + y = 4$

3. Objective function:
   $f(x, y) = 9x^2 + 13y^2 + 18xy - 4$
   Constraint:
   $x^2 + y^2 + 2x = 16$

In your report, briefly explain how the plots help you locate the minimum/maximum points.

Part B – Stationary points and classification (Arora, Problems 4.22, 4.23)

For each function below:

- Compute all stationary points (where the gradient is zero) analytically with MATLAB or Python.
- Use suitable tests (e.g. Hessian) to classify each stationary point as local minimum, local maximum, or inflection/saddle point.

1. $f(x_1, x_2) = 3x_1^2 + 2x_1x_2 + 2x_2^2 + 7$
2. $f(x_1, x_2) = x_1^2 + 4x_1x_2 + x_2^2 + 3$

Recommended reading (Arora, *Introduction to Optimum Design*, 3rd ed.)

- Chapter 3, Section 3.3 – Use of MATLAB for graphical optimization
  - Useful for Problems 3.11, 3.12, 3.13.
- Chapter 4, Section 4.1 – Local and global minimum and maximum
  - Useful for Problems 4.22 and 4.23.
- Chapter 4, Sections 4.3–4.4 – Optimality conditions
- Chapter 7 – Optimum design with MATLAB

Helpful functions and libraries

Depending on whether you choose to perform the lab using MATLAB or Python, the following functions and libraries may be useful. The two lists below provide equivalent tools for each environment.

MATLAB

- syms
- meshgrid
- contour
- plot
- diff
- solve
- gradient
- hessian

Python (NumPy / SymPy / Matplotlib)

- sympy.symbols
- numpy.meshgrid
- matplotlib.pyplot.contour
- matplotlib.pyplot.plot
- sympy.diff
- sympy.solve
- sympy.gradient
- sympy.hessian