Instructions: From this session onwards we will start implementing algorithms for solving non-linear optimization problems. We will begin with line search today. There are three exercises. Questions marked as **[R]** must be answered in a report that must be submitted on Moodle along with all other files. All plots should be saved as pdf files and be included in the report. Please follow the same convention of naming files and directories as outlined in the previous lab.

Exercise 1: Line search without using derivatives
Suppose we are given a convex function $f : \mathbb{R} \to \mathbb{R}$, and we want to find its minimum value in a given interval $[l, u]$. Suppose also that the derivative of the function is not available to us. Let us consider the following algorithm.

$r \leftarrow 0.9$;
while *not converged* **do**
  $b \leftarrow l + r(u - l)$;
  $a \leftarrow u - r(u - l)$;
  if $f(a) < f(b)$ **then**
    $u \leftarrow b$;
  else
    $l \leftarrow a$;
  end
end

1. Implement the above algorithm in a Python function (or a language of your choice).
2. Your implementation should be independent of the function used. You may assume that the function you want to minimize is available as another Python function (evalf) that you can call from your program.
3. Test your algorithm for the function $f = 0.0729x^6 - 1.1664x^5 + 7.7760x^4 - 27.6480x^3 + 55.2960x^2 - 58.9284x + 26.2144$, $l = -10, u = 10$. Use the convergence criterion $|f(l) - f(u)| < 10^{-6}$.
4. **[R]** What is the optimal solution value?
5. **[R]** How many iterations does your routine take?
6. **[R]** We will now study how the size of interval affects the performance. Do part 1. and 2. above again taking $l = -15, u = 15$. Repeat again for $l = -20, u = 20, \ldots, l = -45, u = 45$. Plot the number of iterations against the size of the interval $u - l$ for the 8 trials.
7. **[R]** Now fix $l = -10, u = 10$, and change r. Repeat steps 1. and 2. for $r = 0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.55, 0.4, 0.3, 0.2, 0.1, 0.001$. Plot the number of iterations as a function of r.
8. **[R]** Construct a simple function (not necessarily convex) of one variable where this algorithm fails to find the correct minimum. Plot the function and explain why it fails.

Exercise 2:
Modify the above algorithm for functions of several variables. Suppose you are given a function $f : \mathbb{R}^n \to \mathbb{R}$ and two points $y \in \mathbb{R}^n$ and $w \in \mathbb{R}^n$.

1. Modify your Python function written above to find the minimum of f on the line joining the two points y, w.
2. Test your routine for $f = 3x_1^2 + 0.05x_2^4 + \frac{10}{x_3^2}$ and the two points $(0, 5, 5), (40, 0, 0.5)$.
3. **[R]** Verify your solution graphically by plotting the function along the line joining these points.

Exercise 3:
Suppose you are given a function $f : \mathbb{R}^n \to \mathbb{R}$, a starting point y and a direction d along which you want find the minimum. You only have to search along the ray $y + \lambda d, \lambda > 0$ and not the reverse direction.

1. Write a Python function to find the minimum of f along the direction d starting from a given point y. Note that we do not know the interval in which we can search.
2. [**R**] Test your routine for $f = 3x_1^2 + 0.05x_2^4 + \frac{10}{x_3^2}, y = (7, 2, 0.1), d = (-1, -0.2, 0.5)$.
3. [**R**] Clearly write your algorithm in the report.