# Laboratorio No. 3
## Gradient Descent Methods

Considere el problema de optimización no restringida:

$$\min_{x \in \mathbb{R}^n} \; f(x), \tag{1}$$

en donde $f : \mathbb{R}^n \to \mathbb{R}$ es una función continua y diferenciable. Como se discutió en clase, la idea central detrás del algoritmo *Gradient Descent* (GD) es intentar resolver el problema (1) mediante la evaluación del valor de la función en la dirección en donde se encuentra el mínimo. Una pregunta natural en este momento es: ¿cuál es esta dirección? Sabemos que el vector gradiente (i.e. $\nabla f(x)$) nos provee la dirección de máximo crecimiento de la función $f$. Por tanto, $-\nabla f(x)$ nos devuelve la dirección de menor crecimiento de la función. Esta observación da origen al método GD:

$$\boxed{x_{k+1} = x_k - \alpha_k \nabla f(x_k),}$$

en donde $\alpha_k$ es el llamado "step size" o mejor conocido como "learning rate" en el ámbito de machine learning. En este laboratorio nos enfocaremos en dos puntos importantes:

1. Implementar el algoritmo de GD presentado en clase.
2. Investigar la *convergencia* de este algoritmo para distintas formas de elegir $\alpha_k$.

**Instrucciones:** implemente el algoritmo GD y utilícelo para resolver cada uno de los problemas presentados a continuación.

# Problema 1

Aplique el método GD para minimizar la *función cuadrática*:

$$f(x) = \frac{1}{2} x^T Q x + c^T x.$$

Detenga la ejecución del algoritmo cuando $\|\nabla f(x_k)\| < \epsilon$ o bien cuando el número de iteraciones exceda un $N$ dado. Utilice los valores de $Q$, $c$, $\epsilon$, $N$ y el punto inicial $x_0$ listados a continuación. Para la elección de $\alpha_k$ aplique:

- *Step size exacto:* esto es: $\alpha_k \triangleq \arg\min_{\alpha \geq 0} \; f(x_k - \alpha_k \nabla f(x_k))$ (ver Ejercicio 1).
- *Step size constante:* esto es: $\alpha_k = \alpha$ para todo $k \geq 0$. Pruebe con $\alpha = 0.001, 0.01, 0.1, 0.5, 1$. ¿Qué sucede con el algoritmo para las distintas elecciones de $\alpha_k$?
- *Step size variable:* utilice la sucesión $\alpha_k = \dfrac{1}{k}$ para todo $k > 0$.

Para cada caso, su output debe ser mostrado en una *tabla* con las cuatro columnas siguientes:

a) el número $k$ de la iteración,

b) $x_k$,

c) la dirección $p_k$, y

d) $\|\nabla f(x_k)\|$.

Finalmente, realice una *gráfica* de $\|\nabla f(x_k)\|$ versus $k$, en donde se observe el comportamiento del algoritmo para cada una de las formas de elegir el step size $\alpha_k$. ¿Qué observa? ¿Con cuál elección se obtiene el mejor comportamiento?

Pruebe su programa con los siguientes parámetros:

1. $Q = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad x_0 = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}, \quad \epsilon = 10^{-6}, \quad N = 30.$

2. $Q = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad x_0 = \begin{bmatrix} -1 \\ 2 \\ -3 \end{bmatrix}, \quad \epsilon = 10^{-6}, \quad N = 30.$

## Problema 2

Considere la función $f : \mathbb{R}^2 \to \mathbb{R}_{\ni}$

$$f(x_1, x_2) = 100\,(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Esta función es conocida como *Rosenbrock's Function* y es utilizada como benchmark en la evaluación de algoritmos. Algunos autores le llaman *banana function* debido a la forma de sus curvas de nivel (ver el ejercicio 3 de la Hoja de Trabajo No. 6).

Aplique el método GD para resolver el problema de optimización: $\min\limits_{(x_1,x_2)\in\mathbb{R}^2} f(x_1, x_2)$. Utilice $x^0 = (0, 0)^T$ y un step-size fijo de $\alpha_k = 0.05$ para todo $k$. Detenga la ejecución del algoritmo cuando $\|\nabla f(x_k)\| < 10^{-8}$ o bien cuando el número de iteraciones exceda 1000. Su output debe ser mostrado en una tabla con las cuatro columnas siguientes:

*a)* el número $k$ de la iteración,

*b)* $x_k$,

*c)* la dirección $p_k$, y

*d)* $\|\nabla f(x_k)\|$.

Finalmente, varíe el punto inicial $x_0$, ¿qué observa?