Obligatorio 2

Puntajes máximos:

1a	1b	2a	2b	2c	2d	2e	3a	3b
12	12	12	8	8	12	12	10	14

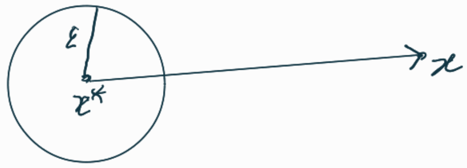
$\underbrace{12 \quad 12}_{24} \quad \underbrace{12 \quad 8 \quad 8 \quad 12 \quad 12}_{52} \quad \underbrace{10 \quad 14}_{24}$

Soluciones

1. a) Sea $f : \mathbb{R}^n \to \mathbb{R}$ una función convexa (no necesariamente diferenciable) con un mínimo local x^* estricto. Probar que x^* es también un mínimo global estricto.

Como x^* es mínimo local estricto $\Rightarrow \exists \varepsilon > 0 \,/\, f(x^*) < f(x), \ \forall x \in B^*(x^*, \varepsilon)$

Sea $x \in \mathbb{R}^n$ cualquiera, por convexidad: $f(x^* + \alpha(x - x^*)) \leq f(x^*) + \alpha(f(x) - f(x^*))$
con $x \neq x^*$ $\qquad \forall \alpha \in [0,1]$



Si tomamos $\alpha > 0$ muy pequeño (tal que $\|\alpha(x - x^*)\| = \alpha \|x - x^*\| < \varepsilon$)

tenemos que $x^* + \alpha(x - x^*) \in B^*(x^*, \varepsilon)$, luego

$f(x^*) < f(x^* + \alpha(x - x^*)) \leq f(x^*) + \alpha(f(x) - f(x^*))$

$\Rightarrow 0 < \alpha(f(x) - f(x^*)) \underset{\alpha > 0}{\Rightarrow} f(x^*) < f(x)$. Luego x^* es mínimo global.

Obs. Si hacen el argumento de arriba tomando $\alpha > 0$ pequeño sin especificar está bien igual.

b) Sea $x_0 \in \mathbb{R}^n$ y $f : \mathbb{R}^n \to \mathbb{R}$ un función diferenciable. Sea $c \in (0,1)$ y $v \in \mathbb{R}^n$ una dirección de decrecimiento de f en x_0 (es decir, $v \neq 0$ y $\frac{\partial f}{\partial v}(x_0) < 0$). Probar que existe $\epsilon > 0$ tal que para todo $\alpha \in (0, \epsilon)$ se cumple:

$$f(x_0 + v\alpha) < f(x_0) + \alpha c \cdot \frac{\partial f}{\partial v}(x_0) < f(x_0).$$

Como $c>0$ y $\frac{\partial f}{\partial v}(x_0) < 0 \Rightarrow f(x_0) + \alpha c \frac{\partial f}{\partial v}(x_0)$, $\forall \alpha > 0$ es trivial.

Recordar que $\frac{\partial f}{\partial v}(x_0) = \lim_{\alpha \to 0} \frac{f(x_0 + \alpha v) - f(x_0)}{\alpha}$

$\frac{\partial f}{\partial v}(x_0) < 0$
$c \in (0,1)$

En particular $\lim_{\alpha \to 0^+} \frac{f(x_0+\alpha v) - f(x_0)}{\alpha} = \frac{\partial f}{\partial v}(x_0) > c \cdot \frac{\partial f}{\partial v}(x_0)$

$\Rightarrow \exists \varepsilon > 0 \,/\, \forall \alpha \in (0, \varepsilon)$ se cumple $\frac{f(x_0+\alpha v) - f(x_0)}{\alpha} > c \cdot \frac{\partial f}{\partial v}(x_0)$

despejando $f(x_0+\alpha v)$ obtenemos la primera desigualdad.

2. a) Implemente los métodos "Steepest Descent" (descenso por grandiente en la dirección de máximo descenso) con paso fijo y "Conjugate Gradient" (gradiente conjugado), según especificaciones en el notebook.

b) Compare los métodos en términos de la cantidad de iteraciones necesarias y los tiempos de ejecución. ¿Los métodos son de descenso?

c) Estimar el óptimo de la función x^* como el promedio de los valores óptimos aproximados obtenidos en ambos métodos y graficar en ambos casos la función de error $e(k) = ||x_k - x^*||$ en función del números de iteraciones k.

d) Implemente nuevamente Steepest Descent pero con pasos según la regla de Armijo. Experimente con los parámetros $c, \beta \in (0,1)$ y $s > 0$. Compare con los métodos anteriores tanto en número de iteraciones como en tiempo de ejecución.

e) Variar el valor propio máximo de la matriz A y analizar como esto afecta la performance de los algoritmos.

En cada caso analice, comente y justifique. En los gráficos de la función de error en función del número de iteraciones utilice escala logarítmica[1] para el eje de las y.

Este ejercicio es para implementarlo en el Colab y completar en el reporte. Esencialmente se pide implementar dos métodos de iteración de la forma

$$x_{k+1} = x_k + \alpha_k d_k \quad \text{con } \alpha_k > 0 \text{ y } d_k \in \mathbb{R}^n \text{ una dirección}$$

de decrecimiento. Cada método da una manera diferente de elegir la dirección d_k, luego el paso α_k se puede elegir paso fijo $\alpha_k = \alpha$, realizando un análisis para elegir un buen α o utilizando algunas de las expresiones vistas en el curso (p.e. $\alpha = \frac{\kappa - 1}{\kappa + 1} = \frac{\lambda_{max} - \lambda_{min}}{\lambda_{max} + \lambda_{min}}$) o con la regla de Armijo (ver Resumen 2024 en eva).

<u>Steepest descent</u>: Tomo $d_k = -\nabla f(x_k)$

<u>Gradiente conjugado</u>: Inicio en $r_0 = b$, $x_0 = 0$, $d_0 = r_0 = b$

1. $\alpha_k = \dfrac{r_k^t d_k}{d_k^t A d_k}$

2. $x_{k+1} = x_k + \alpha_k d_k$

3. $r_{k+1} = b - A x_{k+1} = r_k - \alpha_k A d_k$

4. $\beta_{k+1} = \dfrac{-r_{k+1}^t A d_k}{d_k^t A d_k}$

5. $d_{k+1} = r_{k+1} + \beta_{k+1} d_k$

Obs: Como el producto $A d_k$ aparece muchas veces hay que precalcularlo

En la parte e se pide analizar que para $\lambda_{max} = 20$ y $\lambda_{max} = 120$ (manteniendo el mismo λ_{min}) esto afecta el número de condición $\kappa = \frac{\lambda_{max}}{\lambda_{min}} > 1$, cuanto más pequeño κ más eficiente el algoritmo.

3. a) Implemente el método de Newton según especificaciones en el notebook.

   b) Compare la performance del método de Newton con Steepest Descent para la función de Beale (definida en el notebook) en términos de iteraciones y de tiempo de ejecución. Grafique para cada método el error en función del número de iteraciones usando escala logarítmica para el eje de las y.

Este es otro ejercicio para implementar en Colab.

En el método de Newton $x_{k+1} = x_k + \alpha_k d_k$

donde $d_k = -\left(\nabla^2 f(x_k)\right)^{-1} \cdot \nabla f(x_k)$. En lugar de invertir la matriz Hessiana se pide hallar d_k resolviendo el sistema lineal $\left(\nabla^2 f(x_k)\right) \cdot d_k = -\nabla f(x_k)$ usando el método de gradiente conjugado por ejemplo (ver Resumen 2024).