OPTIMIZACIÓN II
Práctica 1

Francisco J. Aragón Artacho
Dpto. Matemáticas, Universidad de Alicante

2-3 de marzo de 2022

En esta práctica vamos a considerar la función

$$f(x) = 9x_1^2 - 2x_1x_2 + x_2^2 - 4x_1^3 + \frac{1}{2}x_1^4.$$

El origen es un óptimo global de f que verifica la condición suficiente de optimalidad de segundo orden. Por el teorema de Taylor, f será similar a su aproximación de segundo orden cerca del punto 0_2, es decir, para todo x próximo a 0_2,

$$f(x) \approx f(0_2) + \nabla f(0_2)^T x + \frac{1}{2}x^T\nabla^2 f(0_2)x = \frac{1}{2}x^T\nabla^2 f(0_2)x,$$

Como $\nabla^2 f(0_2)$ es definida positiva, hemos visto que el método del gradiente aplicado a la función cuadrática $\frac{1}{2}x^T\nabla^2 f(0_2)x$ verifica

$$\frac{\|x_{k+1}\|}{\|x_k\|} \leq \text{máx}\{|1-\alpha_k m|, |1-\alpha_k M|\},$$

donde m y M son los autovalores más pequeño y más grande de $\nabla^2 f(0_2)$, respectivamente. Además, el valor de α_k que minimiza esta cota es

$$\alpha^* := \frac{2}{m+M}.$$

Crear un programa en Python llamado `Pr1.py`. Se pide:

1. Calcular (con lápiz y papel) el gradiente de f, su matriz hessiana, y los puntos críticos de f. Anotar en un comentario dentro del programa los puntos críticos. Definir dentro del programa la función f, una función g con su gradiente, y una función H que devuelva su matriz hessiana, para cualquier vector x. Usar un `array` del paquete `numpy` tanto para el gradiente como para la matriz hessiana. Verificar con Python que el origen satisface la condición suficiente de optimalidad de segundo orden. Para tener acceso a las funciones del álgebra lineal, se deben cargar usando el comando `from scipy import linalg`.

2. El programa debe dibujar en 3D la función f en el intervalo $[-1, 5] \times [-3, 8]$. Para ello, cargar `from mpl_toolkits import mplot3d` además de importar las funciones de `matplotlib` con el comando `import matplotlib.pyplot as plt`. Se debe crear una figura con `plt.figure()`, unos ejes tridimensionales con `plt.axes(projection='3d')` y dibujar la gráfica con el *método* `plot_surface` de los ejes que has creado.

3. El programa debe crear otra figura y dibujar 50 curvas de nivel de f en el mismo intervalo mediante el comando `plt.contour`.

4. El programa debe calcular 40 iteraciones del método del gradiente con tamaño de paso constante $\alpha_k = \alpha^*$ y punto inicial $x_0 = (0.4, -1.9)^T$, mostrando en pantalla solo $f(x_{40})$.

5. El programa debe calcular 5 iteraciones del método de Newton puro con el mismo punto semilla del apartado anterior, mostrando solo el valor $f(x_5)$. Comparar con el valor obtenido en el apartado anterior.

6. El programa debe realizar 40 iteraciones del método de descenso más rápido con búsqueda lineal exacta. Para hallar el tamaño de paso, usaremos la función `minimize` del paquete `scipy.optimize` con semilla 1. Comparar el resultado con el de los apartados 4 y 5.

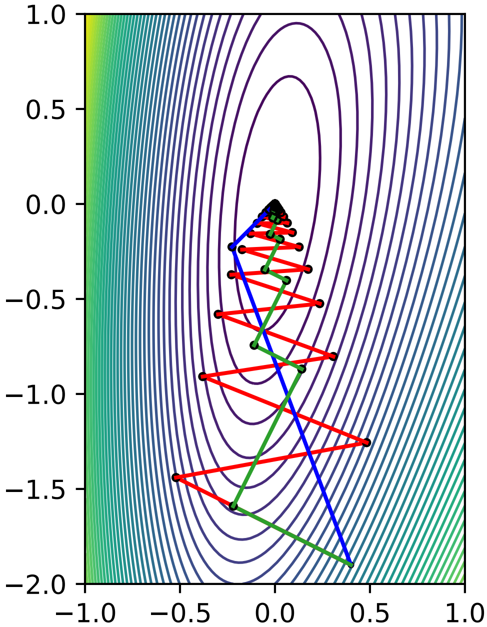

Figura 1: Mostramos 40 iteraciones del método del gradiente con tamaño de paso constante α^* (en rojo) y con búsqueda lineal exacta (en verde). Mostramos en azul 5 iteraciones del método de Newton puro.

7. Usando el comando `plt.plot`, el programa debe dibujar en una nueva figura las dos funciones $\phi_d(\alpha) := f(x_0 + \alpha p_d)$ y $\phi_N(\alpha) := f(x_0 + \alpha p_N)$ en el intervalo $[0, 2]$, donde $x_0 = (0.4, -1.9)^T$,

$$p_d := -\frac{\nabla f(x_0)}{\|\nabla f(x_0)\|} \quad \text{y} \quad p_N := -\frac{(\nabla^2 f(x_0))^{-1}\nabla f(x_0)}{\|(\nabla^2 f(x_0))^{-1}\nabla f(x_0)\|},$$

   correspondientes a la direcciones de norma unidad del método de descenso más rápido y del método de Newton. Para distinguir ambas funciones, dibujar ϕ_d en azul y ϕ_N en rojo. Añadir la leyenda al gráfico con el comando `plt.legend`, denotando las funciones en la leyenda mediante “descenso” y “Newton”. Añadir un comentario dentro del script respondiendo a las siguientes preguntas (basar la respuesta en la figura obtenida):

   a) ¿Cuál de las dos direcciones p_d y p_N sería mejor para una búsqueda lineal exacta?

   b) ¿Cuál sería mejor para tamaños de paso muy pequeños?

8. (OPCIONAL) Crea una gráfica igual a la mostrada en la Figura 1. Los segmentos verdes consecutivos parecen ortogonales... ¿lo son?