Programación: el método del descenso en la dirección del antigradiente

Objetivos. Programar el método iterativo de descenso en la dirección del antigradiente, con la búsqueda exacta sobre cada recta.

Requisitos. Solución de sistemas simétricas positivas definidas con métodos iterativos, con búsqueda exacta sobre cada recta.

1. Del sistema de ecuaciones a la minimización de una función (repaso). Sea A una matriz real simétrica estrictamente positiva definida de orden n, y sea $b \in \mathbb{R}^n$. Ya sabemos que el problema $Ax = b$ es equivalente al problema de minimización de la función

$$f(x) = \frac{1}{2}x^\top Ax - b^\top x. \tag{1}$$

2. El mínimo de una forma cuadrática sobre una recta (repaso). Sea A una matriz real simétrica estrictamente positiva definida de orden n, y sea $b \in \mathbb{R}^n$. Dado un vector $x \in \mathbb{R}^n$ y una *dirección* $p \in \mathbb{R}^n \setminus \{0\}$, consideremos la función

$$g(\alpha) = \frac{1}{2}(x + \alpha p)^\top A(x + \alpha p) - (x + \alpha p)^\top b.$$

Escriba g como un polinomio de grado 2 en la variable α, calcule g' y encuentre el punto $\alpha_{\min}$ donde f alcanza su valor mínimo. Se recomienda usar la notación $r = b - Ax$.

3. El cambio del residuo al cambiar el vector (repaso). Sean A, b, x, p, r los mismos que en el ejercicio anterior. Definimos el vector nuevo y como

$$y = x + \alpha p.$$

Exprese el residuo nuevo $b - Ay$ a través del residuo previo:

$$b - Ay = \underbrace{b - Ax}_{r} + ???.$$

4. El gradiente y el antigradiente de la función f. En el método que estudiamos, el vector p en cada paso se define como el vector de *antigradiente* de la función f:

$$p \coloneqq -(\operatorname{grad} f)(x).$$

Calcule el gradiente $(\operatorname{grad} f)(x)$ y su opuesto $-(\operatorname{grad} f)(x)$ de la función (1).

5. Algoritmo (método de descenso en el sentido del antigradiente).

```
function [x, s] = gradientdescent(A, b, tol, smax),
   n = length(b);
   x = zeros(n, 1);
   r = b; s = 0;
   while (s < smax) && (norm(r) >= tol),
      p = ???;             # p = - (grad f)(x)
      q = A * p;
      alpha = ???;
      x = ???; r = ???;
      s = s + 1;
   end
end
```

6. Prueba en el plano con un dibujo. El siguiente código consiste de dos funciones y se debe guardar en un archivo `plotgradientdescent.m`

```
function [] = plotgradientdescent(),
   A = [3 -2; -2 2];
   sol = [1; -6]; # exact solution
   b = A * sol;
   n = length(b);
   xsequence = zeros(n, 5);
   for s = 1 : 5,
      xsequence(:, s) = gradientdescent(A, b, 1.0E-8, s);
   end
   disp(xsequence);
   r = max(abs(xsequence(:, 1) - sol));
   plotdomain = [sol(1) - r, sol(1) + r, sol(2) - r, sol(2) + r];
   plot(xsequence(1, :), xsequence(2, :), '-w', 'linewidth', 3);
   hold on;
   ezcontourf(@(var1, var2) f(var1, var2, A, b, sol), plotdomain, 100);
end

function result = f(var1, var2, A, b, sol),
   [size1, size2] = size(var1);
   result = zeros(size1, size2);
   for j = 1 : size1,
      for k = 1 : size2,
         x = [var1(j, k); var2(j, k)];
         result(j, k) = sqrt(0.5 * (x - sol)' * A * (x - sol));
      end
   end
end
```