MÉTODOS NUMÉRICOS.

TAREA 14. PSEUDOCÓDIGO PARA OPTIMIZAR FUNCIONES MEDIANTE EL MÉTODO DEL GRADIENTE, UTILIZANDO DIFERENCIAS FINITAS.

,

Adolfo Hernández Ramírez (427560)
Correo: a.hernandezramirez3@ugto.mx.

Licenciatura Ingeniera Química Sustentable. Universidad de Guanajuato. División de Ciencias e Ingenierías. Campus León. Loma del Bosque 103, Lomas del Campestre. León, Gto, México.

Pseudocódigo.

0- INICIO.
1- Definir la función a minimizar:

   F(x,y) = 2xy + 2x –x^2 – 2y^2.

2- Definir las derivadas parciales de x y y por diferencias finitas.
   2.1- Establecer un paso pequeño: h = 1e-6.
   2.2- Definir la derivada parcial de x como una función que retorna:

   Return [f(x+h,y) - f(x-h,y)]/(2*h)

   2.3. Definir de la misma manera la derivada parcial de y.

   Return [f(x,y+h) - f(x,y-h)]/(2*h)

3- Definir la función g(h) = f (x_i+1,y_i+1).
   3.1- Definimos la x_i+1 y y_i+1:

   x_siguiente = x_actual + dfx(x_actual, y_actual) * h
   y_siguiente = y_actual + dfy(x_actual, y_actual) * h

   3.2- Retornar la función evaluada en los nuevos puntos de x y y.

   Return f(x_siguiente, y_siguiente)

4- Definir la derivada de g mediante la aproximación por diferencias finitas centradas.
   4.1- Declaramos una delta = 1e-6.
   4.2- Retornamos el valor de la derivada.

   Return [g(h+delta, x_actual, y_actual) - g(h-delta, x_actual, y_actual)] / (2*delta)

5- Declaramos una funcion para encontrar_h_optimo.
   5.1-Declaro h_bajo = -1, h_alto 0 1, max_iter_biseccion = 50, tol_biseccion = 1e-8.
   5.2- Defino que las derivadas con ambos valores h son:

   deriv_bajo = derivada_g(h_bajo, x_actual, y_actual)
   deriv_alto = derivada_g(h_alto, x_actual, y_actual)

   5.3- SI deriv_bajo * deriv_alto > 0 ENTONCES return 0.1.
   FIN SI.

5.4. PARA i = 1 HASTA max_iter_biseccion HACER

h_medio = (h_bajo + h_alto) / 2
deriv_medio = derivada_g(h_medio, x_actual, y_actual)

5.4.1- SI |deriv_medio| < tol_biseccion ENTONCES break.
FIN SI.

5.4.2- SI deriv_bajo * deriv_medio < 0 ENTONCES

h_alto = h_medio
deriv_alto = deriv_medio

SINO

h_bajo = h_medio
deriv_bajo = deriv_medio

FIN SI.

5.4.3- FIN PARA.

5.5- Retornar h_medio.

6- Declarar una función llamada metodo_gradiente.
6.1- Declarar los valores actuales de x y y como:

x_actual = x_inicial
y_actual = y_inicial

6.2- Imprimir “Metodo del Gradiente” y “Punto inicial”.
6.3- PARA iteracion = 1 HASTA max_iteraciones HACER.

6.3.1- Calcular el gradiente.

derivada_x = dfx(x_actual, y_actual)
derivada_y = dfy(x_actual, y_actual)

6.3.2- Encontrar el tamaño de paso óptimo.

h_optimo = encontrar_h_optimo(x_actual, y_actual).

6.3.3-Actualizar la posición.

x_siguiente = x_actual + derivada_x * h_optimo
y_siguiente = y_actual + derivada_y * h_optimo

6.3.4 - Mostrar los resultados para cada iteración.

IMPRIMIR "Iteración:", iteracion
IMPRIMIR "Punto actual:", x_actual, y_actual
IMPRIMIR "Gradiente:", derivada_x, derivada_y
IMPRIMIR "h óptimo:", h_optimo
IMPRIMIR "Nuevo punto:", x_siguiente, y_siguiente

6.3.5 - Actualizar para la siguiente iteración.

x_actual = x_siguiente
y_actual = y_siguiente

6.3.6- FIN PARA:

6.4- IMPRIMIR "Resultado final:", x_actual, y_actual.

7- Dentro del Int main.
7.1- Declarar las variables y parámetros del programa.

x_inicial = -1.0
y_inicial = 1.0
max_iter = 20
tolerancia = 1e-6

7.2- Llamar a la función metodo_gradiente.

8- FIN.