

parcial2_optimizacion

May 16, 2024

1 Curso de Optimización I

1.1 CIMAT; DEMAT, UG

1.2 Parcial 2

Descripción:	Fechas
Fecha de publicación del documento:	Mayo 16, 2024
Hora de inicio:	11:00
Hora límite de entrega:	12:30

1.2.1 Indicaciones

Lea con cuidado los ejercicios.

Puede usar las notas de clase y las tareas hechas para resolver el examen.

Al final, entregue el notebook con sus respuestas, junto con los códigos que hagan falta para reproducir los resultados. Si es más de un archivo, genere un archivo ZIP que contenga el notebook y los scripts adicionales.

A partir del notebook genere un archivo PDF con las respuestas y envíelo por separado antes de la hora límite.

1.3 Ejercicio 1. (6 puntos)

Considere el problema

$$\min f(\mathbf{x}) \quad \text{sujeto a} \quad c_1(\mathbf{x}) = 0.$$

Encontrar la solución usando una penalización cuadrática (clase 29). Para esto construimos la función

$$Q(\mathbf{x}; \mu) = f(\mathbf{x}) + \frac{\mu}{2}(c_1(\mathbf{x}))^2$$

1. Programar la función $Q(x; \mu)$ y su gradiente

$$\nabla Q(\mathbf{x}; \mu) = \nabla f(\mathbf{x}) + \mu c_1(\mathbf{x}) \nabla c_1(\mathbf{x}).$$

2. Programar el método de penalización cuadrática usando el método BFGS modificado:

a) Dar la función $f(\mathbf{x})$, $c_1(\mathbf{x})$, la función $Q(\mathbf{x}; \mu)$, su gradiente $\nabla Q(\mathbf{x}; \mu)$, un punto inicial \mathbf{x}_0 , μ_0 , una tolerancia $\tau > 0$, el número máximo de iteraciones N , y los parámetros que se necesiten para usar el método BFGS modificado.

b) Para $k = 0, 1, \dots, N$ repetir los siguientes pasos:

b1) Definir $\tau_k = \left(1 + \frac{10N}{10k+1}\right) \tau$

b2) Calcular el punto \mathbf{x}_{k+1} como el minimizador de $Q(\mathbf{x}; \mu_k)$ con el método BFGS modificado usando como punto inicial a \mathbf{x}_k y la tolerancia τ_k .

b3) Imprimir el punto \mathbf{x}_{k+1} , $f(\mathbf{x}_{k+1})$, $Q(\mathbf{x}; \mu_k)$, el número de iteraciones realizó el algoritmo BFGS y el valor $c_1(\mathbf{x}_{k+1})$.

b4) Si $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \tau$, terminar devolviendo \mathbf{x}_{k+1}

b5) En caso contrario, hacer $\mu_{k+1} = 2\mu_k$ y volver al paso (b1)

3. Probar el algoritmo tomando como f a la función de Beale, $c_1(\mathbf{x}) = x_1^2 + x_2^2 - 4$, $\mu_0 = 0.5$, $N = 1000$ y $\tau = \epsilon_m^{1/3}$.

Use los puntos iniciales $\mathbf{x}_0 = (0, 2)$ y $\mathbf{x}_0 = (0, -2)$.

4. Para verificar el resultado obtenido haga lo siguiente:

- Genere una partición $\theta_0 < \theta_1 < \dots < \theta_m$ del intervalo $[0, 2\pi]$ con $m = 1000$
- Evalúe la función de Beale en los puntos $(2 \cos \theta_i, 2 \sin \theta_i)$ para $i = 0, 1, \dots, m$. e imprima el punto en donde la función tuvo el menor valor y el valor de la función en ese punto.

Nota: Si no tiene implementado el método BFGS modificado, puede elegir otro método de optimización, pero se aplica una penalización de 0.5 puntos.

1.3.1 Solución:

[]:

—

1.4 Ejercicio 2. (4 puntos)

Programar el método de Newton para resolver el sistema de ecuaciones no lineales (Algoritmo 1 de la Clase 24):

$$\begin{aligned} 2x_0 + x_1 &= 5 - 2x_2^2 \\ x_1^3 + 4x_2 &= 4 \\ x_0x_1 + x_2 &= \exp(x_2) \end{aligned}$$

1. Programar la función $\mathbf{F}(\mathbf{x})$ correspondiente a este sistema de ecuaciones y su Jacobiana $\mathbf{J}(\mathbf{x})$
2. Programe el algoritmo del método de Newton. Use como condición de paro que el ciclo termine cuando $\|\mathbf{F}(\mathbf{x}_k)\| < \tau$, para una tolerancia τ dada. Haga que el algoritmo devuelva el punto \mathbf{x}_k , el número de iteraciones k , el valor $\|\mathbf{F}(\mathbf{x}_k)\|$ y una variable indicadora *bres* que es 1 si se cumplió el criterio de paro o 0 si terminó por iteraciones.
3. Para probar el algoritmo y tratar de encontrar varias raíces, haga un ciclo para hacer 20 iteraciones y en cada iteración haga lo siguiente:
 - Dé el punto inicial \mathbf{x}_0 como un punto aleatorio generado con `numpy.random.randn(3)`
 - Ejecute el método de Newton usando \mathbf{x}_0 , la tolerancia $\tau = \sqrt{\epsilon_m}$ y un máximo de iteraciones $N = 100$.
 - Imprima el punto \mathbf{x}_k que devuelve el algoritmo, la cantidad de iteraciones realizadas, el valor de $\|\mathbf{F}(\mathbf{x}_k)\|$ y la variable indicadora *bres*.

1.4.1 Solución:

[]:

—

[]: