MAC0427 – Programação Não Linear
Marcos Kazuya Yamazaki
NUSP: 7577622

Relatório EP1

Neste exercício programa foi implementado a algoritmo modificado do método de Newton que calcula a direção de descida globalmente convergente de uma função com convergência quadrática na vizinhança de x^k (ponto atual).

Primeiramente o algoritmo pede para ser digitado a função que se deseja minimizar, e ela pode conter até três variáveis, e elas terão que ser obrigatóriamente x, y e z. Após a função ser digitada, Especifique o ponto em que o algoritmo irá começar a sua interação (obs: caso a sua função não contenha todas as variáveis, atribua zero (0), para o ponto inicial naquela variável).

As constantes $\alpha, \beta, \gamma, \theta$ já estão definidas no algoritmo, mas podem ser mudadas dentro do intervalo permitido.

A partir dai, o algoritmo imprime o ponto x^k em que está, e calcula o valor da função neste ponto. O seu vetor gradiente e a matriz hessiana são calculadas em função das variáveis simbólicas x, y e z.

O gradiente: $G = \nabla f(x,y,z) = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$

E a matriz Hessiana: $\mathrm{H} = \nabla^2 f(x,y,z) = \begin{pmatrix} \partial^2 f/\partial^2 x & \partial^2 f/\partial x\partial y & \partial^2 f/\partial x\partial z \\ \partial^2 f/\partial y\partial x & \partial^2 f/\partial^2 y & \partial^2 f/\partial y\partial z \\ \partial^2 f/\partial z\partial x & \partial^2 f/\partial z\partial y & \partial^2 f/\partial^2 z \end{pmatrix}$

O vetor e a matriz g e h no algoritmo são respectivamente o gradiente e o hessiano da funçao objetiva, porém calculado no ponto x^k, enquanto G e H, são as mesmas matrizes com as veriáveis simbolicas.

Começando as iterações, calculamos o gradiente no ponto atual, caso a normal do gradiente seja igual a zero, isso quer dizer que o vetor $g_k = \nabla f(x^k) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, ou seja, estamos num ponto estacionário, que pode ser o mínimo da função. Se isso ocorrer o algoritmo para. Caso contrário, é calculado a matriz hessiana neste ponto.

Aqui queremos achar a direção viável de descida da função: $h_k * d_k = -g_k$. Porém seria muito ineficiente calcular a matriz inversa de h_k, principalmente quando a sua dimensão é grande.

Fazemos a fatoração de Cholesky, onde faz $h_k = L' * L$, onde L é uma matriz triangular inferior. Fazendo assim, seria mais fácil achar a direção resolvendo os sistemas lineares.

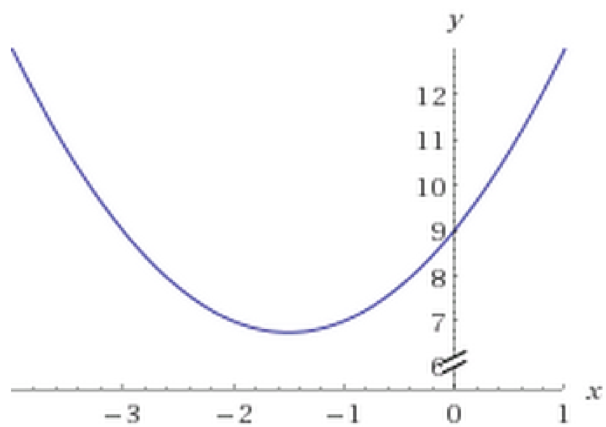
No passo seguinte, ajustamos a direção caso a $\|d_k\|$ seja menor que $\beta * \|g_k\|$.

Assim queremos achar o tamanho do passo a ser dado na direção que satisfaça a condição de Armijo que é: $Achar\ t > 0 \mid f(x_k + t * d_k) = f(x_k) + \alpha * t * g_k' * d_k$.

Assim o novo ponto será $x_{k+1} = x_k + t * d_k$.

Alguns exemplos de saida:

Digite a funcao que deseja minimizar
em função de x, y e z: x^2 + 3*x + 9



f =

x^2 + 3*x + 9

Digite os valores do primeiro ponto
x: 102312
y: 0
z: 0
Ponto de partida [102312.000, 0.000, 0.000]`
O valor da função neste ponto é
10468052289.0000

------------------- Iteração 1 -------------------

Gradiente da função: [204627.000, 0.000, 0.000]`
Passo e direção: 0.100*[-1023135.00, 0.00, 0.00]`

Novo ponto calculado: [-1.500, 0.000, 0.000]`
O valor da função neste ponto é: 6.750000000

------------------- Iteração 2 -------------------

Gradiente da função: [0.000, 0.000, 0.000]`

Ponto de mímino local: [-1.500, 0.000, 0.000]`
Com valor da função: 6.7500000000

Digite a funcao que deseja minimizar
em função de x, y e z: -x^2 + 3*x + 9

f =

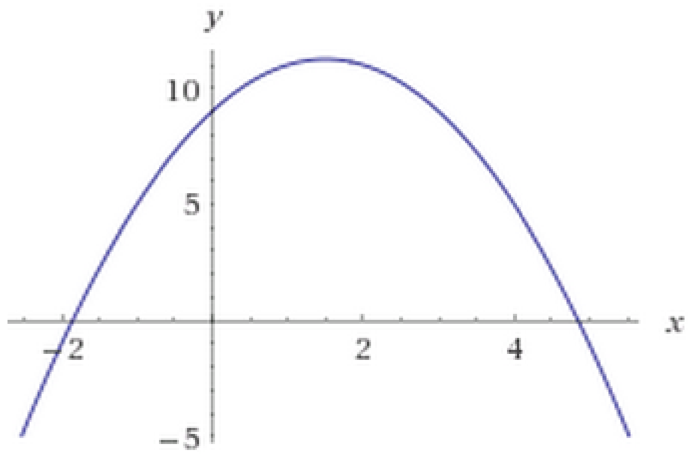
- x^2 + 3*x + 9

Digite os valores do primeiro ponto
x: 1
y: 0
z: 0
Ponto de partida [1.000, 0.000, 0.000]`
O valor da função neste ponto é 11.0000

------------------- Iteração 1 -------------------

Gradiente da função: [1.000, 0.000, 0.000]`

A função é ilimitada
Com o valor da função objetiva –infinito



Digite a funcao que deseja minimizar
em função de x, y e z: x^2 + 3*x*y + 5*y^2 + 2*x + 3*y^2

f =

x^2 + 3*x*y + 2*x + 8*y^2

Digite os valores do primeiro ponto
x: 1
y: 1
z: 0
Ponto de partida [1.000, 1.000, 0.000]`
O valor da função neste ponto é 14.0000

------------------- Iteração 1 -------------------

Gradiente da função: [7.000, 19.000, 0.000]`
Passo e direção: 0.012*[-33.15, -95.66, 0.00]`

Novo ponto calculado: [0.595, -0.170, 0.000]`
O valor da função neste ponto é: 1.470548286

[...]

------------------- Iteração 9 -------------------

Gradiente da função: [0.001, 0.004, 0.000]`
Passo e direção: 0.012*[-0.01, -0.02, 0.00]`

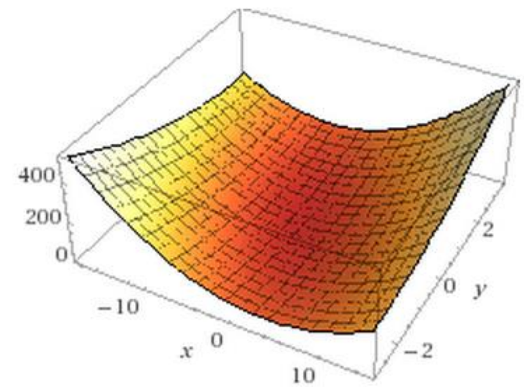
Novo ponto calculado: [-1.391, 0.261, 0.000]`
O valor da função neste ponto é: -1.391304280

------------------- Iteração 10 -------------------

Gradiente da função: [0.000, -0.000, 0.000]`

Ponto de mímino local: [-1.391, 0.261, 0.000]`
Com valor da função: -1.3913042795

3D plot:



Contour plot: