

Neste projeto de otimização convexa eu quero que voce compare 3 quaisquer algoritmos das familias abaixo. Note que os algoritmos ja estao implementados no *scipy.optimize* ou na função *optim* do R.

- alguma tecnica que precisa do gradiente da função explicitamente (como o conjugado gradiente)
- alguma técnica de quasi-Newton que NAO precisa do gradiente explicito (como o L-BFGS)
- alguma técnica que NAO usa o gradiente (como Nelder-Mead)

Para as funções em duas dimensões f_1 e f_2

$$\begin{aligned}f_1(x, y) &= x^2 + 3 * y^2 \\f_2(x, y) &= x^2 + 100(y - x^2)^2\end{aligned}$$

e começando do ponto (1,1) faça

- plote num grafico x,y a sequencia de solucoes dos algoritmos das tres familias acima. Para o Nelder-Mead assuma que a solução numa interação é o centro geometrico do simplex
- imprima o número de interações até encontrar a solução (assuma uma tolerancia de 10^{-6})
- imprima o número de chamadas da função f (e da função que retorna o gradiente de f para o algoritmo da primeira familia).

Para a função:

$$f3(x_1 \dots x_{100}) = \sum_{i=1}^{99} 2(x_{i+1} - x_i^2)^2 + x_i^2$$

e partindo do ponto (1,1,... 1) faça

- imprima o número de interações até encontrar a solução (assuma uma tolerancia de 10^{-6})
- imprima o número de chamadas da função f e do gradiente de f