Problem set 1: Newton method and gradient descent

Eva Perazzi, Gustave Besacier, Agustina María Zein

September 23, 2023

1 Exercise 1

Let us recall the Black-Scholes option price formula for a non-dividend paying call is

$$C(S,K,T,\sigma,r) = SN(d_1) - Ke^{-rT}N(d_2) \tag{1}$$

where we define the call option on a stock S, the strike of the option K, the expiration T, the risk-free rate r and

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{u^2}{2}} du \tag{2}$$

$$d_1 = \frac{\left(\log\left(\frac{S}{K}\right) + \left(r + \frac{1}{2}\sigma^2\right)T\right)}{\sigma\sqrt{T}} \qquad d_2 = d_1 - \sigma\sqrt{T} \tag{3}$$

with σ the implied volatility.
Moreover, we have

$$\frac{\partial C(S,K,T,\sigma,r)}{\partial \sigma} = S\frac{e^{-\frac{d_1^2}{2}}}{\sqrt{2\pi}}\sqrt{T} \tag{4}$$

and we know that the current value of the option is $C(100, 100, 1.5, \sigma^*, 0.04) = 10.78$.
Our goal is to determine σ^*. For that matter, let us define the function $f : \mathbb{R} \to \mathbb{R}$ such that $f(\sigma) = C(S,K,T,\sigma,r)$ when S, K, T and r are held fixed at values $S = S(0) = 100$, $K = 100$, $T = 1.5$ and $r = 0.04$, namely $f(\sigma) = C(100, 100, 1.5, \sigma, 0.04)$.
We use the Newton method to find the zeros of the function $g : \mathbb{R} \to \mathbb{R}$ defined as $g(\sigma) = f(\sigma) - f(\sigma^*) = f(\sigma) - 10.78$. Therefore our problem translates into searching for the root in the interval $[0, 1]$ of g.
The functions f and g are continuous and differentiable, and we observe:

$$f'(\sigma) = g'(\sigma) = \frac{\partial C(S,K,T,\sigma,r)}{\partial \sigma} = S\frac{e^{-\frac{d_1^2}{2}}}{\sqrt{2\pi}}\sqrt{T} \tag{5}$$

Using Matlab, we implement the Newton method

$$\sigma_{k+1} = \sigma_k - \frac{g(\sigma_k)}{g'(\sigma_k)} \tag{6}$$

using a tolerance level of 10^{-5} and a starting value of $\sigma_0 = 0.5$. As a result, we obtain an implied volatility of $\sigma^* = 0.1594$ computed in 5 iterations.

2 Exercise 2

In this exercise, we aim to find the minimum of the following function

$$f(x_1, x_2) = (x_1 - 2)^4 + (x_1 - 2x_2)^2 \tag{7}$$

To solve it, we will first use the Newton method then the gradient descent method.

2.1 Optimization using the Newton method

According to the Newton method, we have the following

$$x_{k+1} = x_k - \text{inv}(H_F(x_k))\vec{\nabla}F(x_k) \tag{8}$$

We first calculate the partial derivatives of f with respect to x_1 and x_2. We find

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 4(x_1 - 2)^3 + 2(x_1 - 2x_2) \tag{9}$$

$$\frac{\partial f(x_1, x_2)}{\partial x_2} = -4(x_1 - 2x_2) \tag{10}$$

We then compute

$$\frac{\partial^2 f(x_1, x_2)}{(\partial x_1)^2} = 12(x_1 - 2)^2 + 2 \tag{11}$$

$$\frac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} = -4 \tag{12}$$

$$\frac{\partial^2 f(x_1, x_2)}{(\partial x_2)^2} = 8 \tag{13}$$

We obtain the gradient and the Hessian matrix, which are given by

$$\vec{\nabla} f(x_1, x_2) = \begin{bmatrix} 4(x_1 - 2)^3 + 2(x_1 - 2x_2) \\ -4(x_1 - 2x_2) \end{bmatrix} \tag{14}$$

$$H_f(x_1, x_2) = \begin{bmatrix} 12(x_1 - 2)^2 + 2 & -4 \\ -4 & 8 \end{bmatrix} \tag{15}$$

By using the starting point $x_0 = (3, 3)$, and after 24 iterations, we find the local minimum $x^* = (2, 1)$. However, if we take the starting point $x_1 = (2, 2)$, the Hessian matrix at point x_1 is equal to

$$H_f(2, 2) = \begin{bmatrix} 2 & -4 \\ -4 & 8 \end{bmatrix} \tag{16}$$

and we have

$$\det(H_f(2, 2)) = 0$$

Therefore the Hessian is not invertible at that point and the equation (8) is not valid.

2.2 Optimization using the gradient descent method

By doing the calculations with the gradient descent method, we will select a few different step sizes and record the number of iterations for each of them. The gradient descent step is written

$$x_{k+1} = x_k - \alpha \vec{\nabla} f(x_k) \tag{17}$$

where $\vec{\nabla} f(x_k)$, $x_k = (x_{k,1}, x_{k,2}) \in \mathbb{R}^2$ refers to the same gradient as in part (2.1).
We choose three different step sizes, namely $\alpha_1 = 10^{-1}$, $\alpha_2 = 10^{-2}$ and $\alpha_3 = 10^{-3}$. Starting at point $x_0 = (3,3)$, we find the objectives $x^*_{0,1}$, $x^*_{0,2}$ and $x^*_{0,2}$, using respectively α_1, α_2 and α_3

$$x^*_{0,1} = (2.0068, 1.0034) \text{ computed in 33926 iterations} \tag{18}$$
$$x^*_{0,2} = (2.0068, 1.0034) \text{ computed in 339337 iterations} \tag{19}$$
$$x^*_{0,3} = (2.0068, 1.0034) \text{ computed in 3393436 iterations} \tag{20}$$

When starting at point $x_1 = (2,2)$, we find the objectives $x^*_{1,1}$, $x^*_{1,2}$ and $x^*_{1,2}$, using respectively α_1, α_2 and α_3

$$x^*_{1,1} = (2.0068, 1.0034) \text{ computed in 33918 iterations} \tag{21}$$
$$x^*_{1,2} = (2.0068, 1.0034) \text{ computed in 339243 iterations} \tag{22}$$
$$x^*_{1,3} = (2.0068, 1.0034) \text{ computed in 3392482 iterations} \tag{23}$$

We concluded that, whereas Newton method finds a local minimum of the function in less iterations, its applicability requires more restrictive conditions on continuity and smoothness of the function than the gradient descent method.