1 Direction of the steepest change

1.1 Gradient operator

if we consider a multi-variable function $f : \mathbb{R}^n \leftarrow \mathbb{R}$ and a vector $x \in \mathbb{R}^n$, then the gradient operator can be computed as follows:

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{d\mathbf{x}}{dx_1} \\ \frac{d\mathbf{x}}{dx_2} \\ .. \\ .. \\ \frac{d\mathbf{x}}{dx_n} \end{bmatrix}$$

1.2 directional Derivative

given a point / vector x_0 and a vector v, we define the directional derivative as

$$D_v \cdot \nabla f(\mathbf{x})|_{x=x_0} = \nabla f(\mathbf{x})|_{x=x_0} \cdot v$$

The resulting value represents the change introduced by going in the direction of vector v starting from point x_0.

1.3 Direction of the steepest change

let's consider **unit vectors**, as our goal in the direction and the direction is not affected by vector's scale. we have $D_v \cdot \nabla f(\mathbf{x})|_{x=x_0} = |\nabla f(\mathbf{x})| \cdot |v| \cdot \cos\theta$ where θ is the angle between the two directions. This dot product is maximized for $\theta = 0$. Thus, the direction giving the maximum increase is the gradient's direction. Therefore, it is the direction of the steepest change.

2 Gradient Descent: the steepest descent

The general formula for Newton's method is as follows:

$$x_{k+1} = x_k + \alpha \cdot p_k$$

where α is the step size and p_k is the search direction. Using the conclusions made in the previous section, we consider $p_k = \nabla f(x_k)$

2.1 step size

The step size affects the algorithm's outcome, as a small one might result in the slow convergence, while a step too large might lead to **divergence**, totally missing the minimum point.

2.2 Good Steps

The main criteria to avoid divergence is to have $f(x_{k+1}) \leq f(x_k)$: to make sure that the function is actually descending. More formally, this criteria is known as **Armijo rule**.
The steepest rule can be found at each step by solving the following optimization problem:

$$\alpha_k = arg_\alpha min(f(x_k - \alpha \cdot \triangledown f(x_k))$$