CW Metoda gradientu

$F\colon \mathbb{R}^n \to \mathbb{R}$

Problem minimalizacji $F(x_1, \ldots, x_n) \to$ minimum (lokalne)

Algorytm metody gradientu
Ustalić stałe $c > 0$ i $\varepsilon > 0$.
Startować od
(i-a) dowolnie wybranego
albo
(i-b) losowanego
warunku początkowego $x^{\text{old}} = (x_1^{\text{old}}, \ldots, x_n^{\text{old}})$.
(W przypadku (ii), losować $x_i^{\text{old}} \in [-N, N]$ $(1 \le i \le n)$ dla z góry ustalonego N.)
(ii)
Niech

$$x_i^{\text{new}} = x_i^{\text{old}} - c\,\frac{\partial F}{\partial x_i}(x_1^{\text{old}}, \ldots, x_n^{\text{old}}) \quad (1 \le i \le n).$$

(iii)
while

$$\max_{1\le i\le n} |x_i^{\text{new}} - x_i^{\text{old}}| > \varepsilon$$

do

$$x_i^{\text{old}} = x_i^{\text{new}} \quad (1 \le i \le n)$$
$$x_i^{\text{new}} = x_i^{\text{old}} - c\,\frac{\partial F}{\partial x_i}(x_1^{\text{old}}, \ldots, x_n^{\text{old}}) \quad (1 \le i \le n).$$

(iv)
print x_i^{new} $(1 \le i \le n)$, $F(x_1^{\text{new}}, \ldots, x_n^{\text{new}})$

Zadanie
Implementować powyżej podany algorytm metody gradientu. Za pomocą tego algorytmu znaleźć **lokalne** oraz **globalne** minimum następujących funkcji oraz punkty, które osiągają takie minimum.
Próbować i porównać (i-a) i (i-b).
Próbować i porównać różne parametry c, ε (oraz N).
np. $c = 0.01, \varepsilon = 0.00001 \sim 0.00000001$
(1) $F_1(x_1, x_2, x_3) = 2x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 - 2x_2x_3 - 2x_1 + 3 \quad (n = 3)$
(2) $F_2(x_1, x_2) = 3x_1^4 + 4x_1^3 - 12x_1^2 + 12x_2^2 - 24x_2 \quad (n = 2)$

Uwaga Metoda gradientu trafi tylko minimum <u>lokalne</u>. Aby znaleźć minimum globalne, trzeba próbować różne warunki początkowe. Zob. (i-a) i (i-b) powyższego algorytmu.

Uwaga Natępujące normy są równoważne w sensie $\|x\|_p \to 0 \Leftrightarrow \|x\|_q \to 0$ (o ile wektory są skończenie wymiarowe tzn. $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$).
(i) $\|x\|_\infty = \max_{1\le i\le n} |x_i|$ (indeks $p = \infty$)
(ii) $\|x\|_1 = \sum_{i=1}^n |x_i|$
(iii) $\|x\|_2 = (\sum_{i=1}^n |x_i|^2)^{\frac{1}{2}}$ (norma eukledesowa)

(iv) Ogólnie $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}} \quad p \in [1, \infty)$

Wskazówki do pisania programu (Propozycja)

$c \rightsquigarrow c$, $\varepsilon \rightsquigarrow epsilon$ lub e

$x_i \rightsquigarrow x[i]$, $x_i^{\text{old}} \rightsquigarrow x\_old[i]$, $x_i^{\text{new}} \rightsquigarrow x\_new[i]$

$F(x_1, \ldots, x_n) \rightsquigarrow F$, $\;\mathbf{def} F$

$\frac{\partial F}{\partial x_i}(x_1, \ldots, x_n) \rightsquigarrow Df\_x[i] \quad \mathbf{def} DF\_x[i][x]$, $x = x[i]$ (Python?)

$x_i^{\text{new}} = x_i^{\text{old}} - c\,\frac{\partial F}{\partial x_i}(x_1^{\text{old}}, \ldots, x_n^{\text{old}}) \rightsquigarrow x\_new[i] = x\_old[i] - c * DF\_x[i]$

Generowanie losowanego $x \in [-N, N] \rightsquigarrow x = (\frac{rand()}{Rand\_Max} - 0.5) * 2 * N$