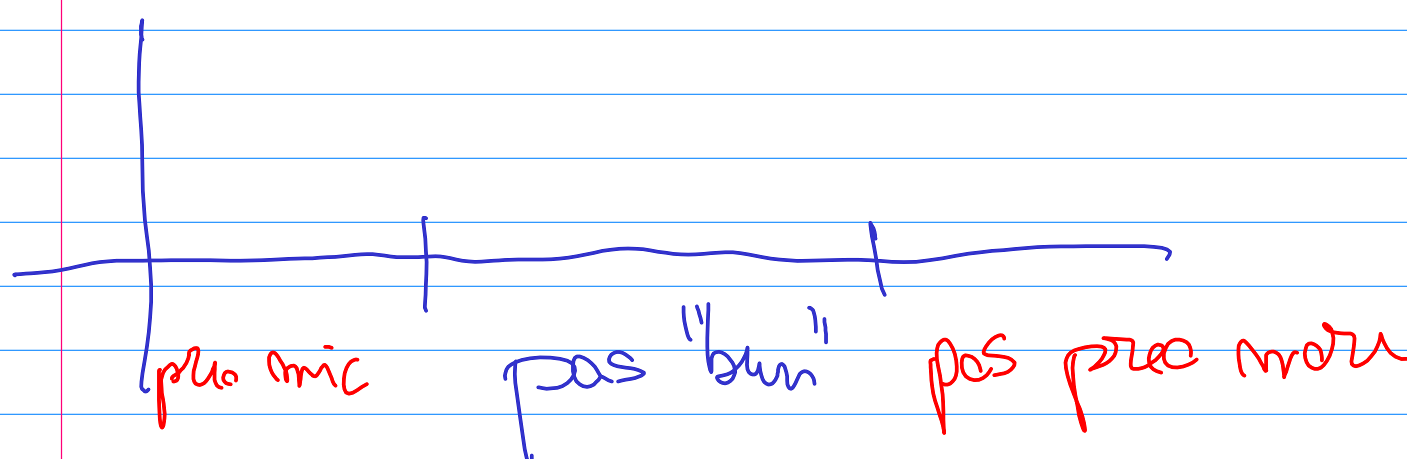
Tehnici Optimizare Dim > 1

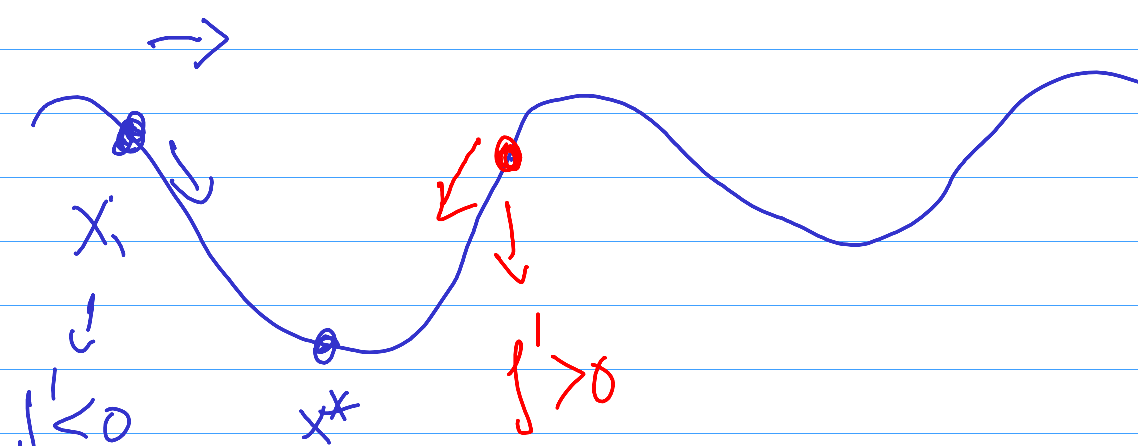
Metoda gradientului

$$x_{i+1} = x_i - \gamma f'(x_i)$$

γ ↓ pas > 0

- fix (mic)
- ales "line search"



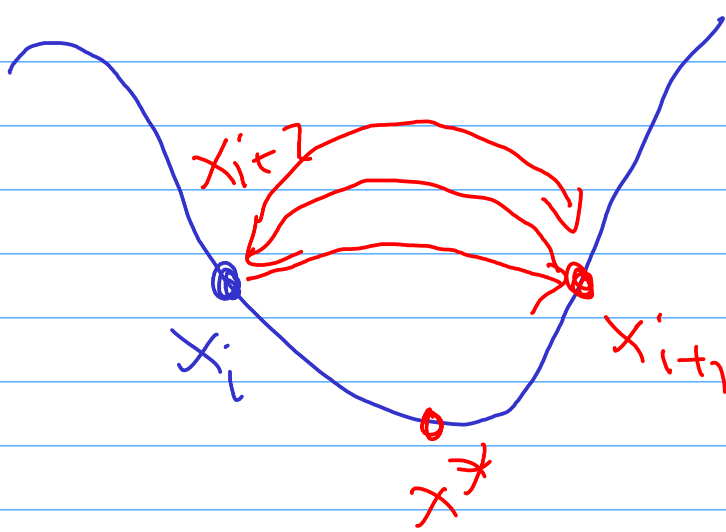


pasul din algoritmul gradientului

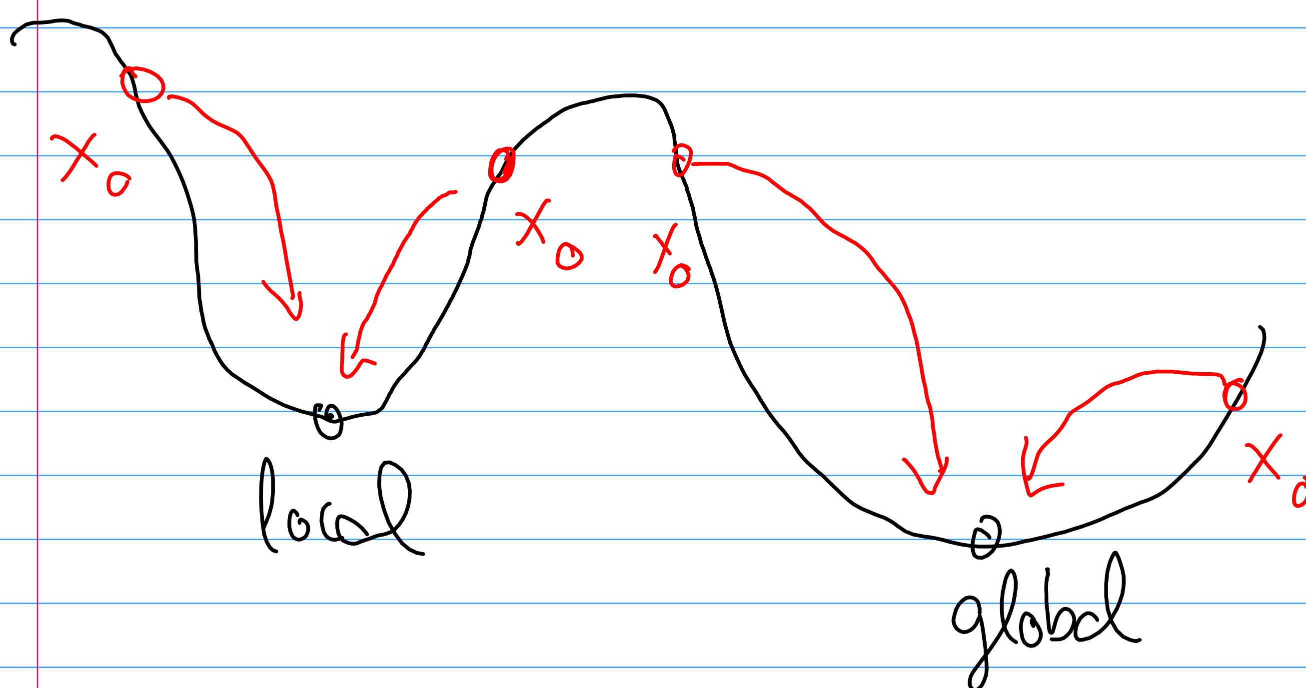
= "learning rate"

- pentru a avansa "rapid"
  pas γ mare

- dacă γ este prea mare
  atunci alg poate să
  nu convergă deloc



- Convergență: pas suficient de mic
- minim local:



Viteza de convergență

- liniară $err_{n+1} \le \rho \, err_n$, $\rho \in (0,1)$

Gradient descent: conv. liniară

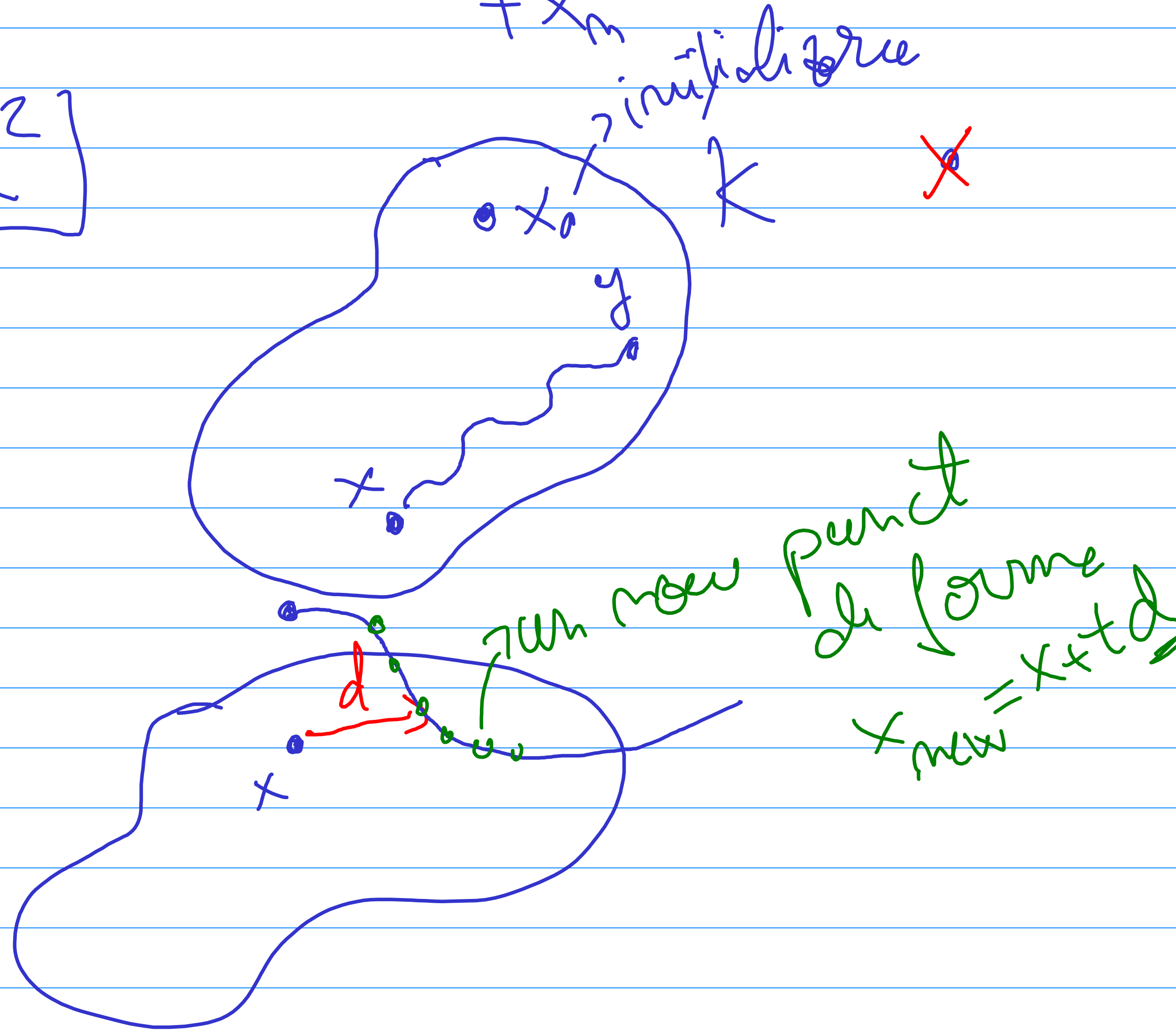
$f:\mathbb{R}^m \to \mathbb{R}$

Exemplu $\quad f(x,y) = x+y$

$$f(x,y) = x^2+y^2$$

$$f(x_1,x_2,\dots,x_m) = \sin x_1 + \cos x_2 + x_m^2$$

$\mathbb{R}^2$



Gradient:

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

$\in \mathbb{R}^2$

- $f(x,y) = x + y^2$

$$\frac{\partial f}{\partial x} = 1 + 0 = 1$$

y const

$$\frac{\partial f}{\partial y} = 0 + 2y = 2y$$

x const

$$\Rightarrow \nabla f = \begin{pmatrix} 1 \\ 2y \end{pmatrix}$$

- Calcul Simbolic sympy
- Wolfram Alpha

Misiune: deriv a 2-a

$$\frac{\partial}{\partial x}\frac{\partial f}{\partial x}, \quad \frac{\partial}{\partial y}\frac{\partial f}{\partial x}$$ → 2 deriv de ord 2

$$\frac{\partial}{\partial x}\frac{\partial f}{\partial y}, \quad \frac{\partial}{\partial y}\frac{\partial f}{\partial y}$$

$$\begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix} = D^2 f$$

Dacă $f: \mathbb{R}^n \to \mathbb{R}$, $\nabla f(x) \in \mathbb{R}^n$,

$D^2 f(x) \in \mathbb{R}^{n \times n}$ matrice de ord n

1. $f(x_1, \dots, x_n) = x_1^2 + x_2^2 + \dots + x_n^2$

$$\frac{\partial f}{\partial x_1} = 2x_1, \quad \frac{\partial f}{\partial x_2} = 2x_2, \quad \dots, \quad \frac{\partial f}{\partial x_n} = 2x_n$$

∂x_2

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_m} \end{pmatrix} = \begin{pmatrix} 2x_1 \\ 2x_2 \\ \vdots \\ 2x_m \end{pmatrix} = 2x$$

$$\nabla^2 f(x) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & \ddots \\ & & & 2 \end{pmatrix} = 2I$$

$f(x) = ax^2 + bx + c$

2. $f(x) = \frac{1}{2} x^T A x - b^T x$

prod scalar

$$= \frac{1}{2} (x_1 - x_m) \begin{pmatrix} a_{ij} \\ A \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} - b^T x$$

$$= \frac{1}{2} \sum x_i a_{ij} x_j - \sum b_i x_i$$

Prod scalar: $\langle a, b \rangle =$

$$= a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

$$\nabla f(x) = Ax - b \in \mathbb{R}^n$$

$$D^2 f(x) = A$$

$$f(x+h) = f(x) + \nabla f(x) \cdot h + (\text{mic}$$

mic

$$f(x+h) = f(x) + \underbrace{\nabla f(x)}_{b} \cdot h + \frac{1}{2} h^T \underbrace{D^2 f(x)}_{A} h$$

$$\frac{1}{2} h^T A h + b^T h$$

1D $\quad ax^2 \geq 0 \quad$ dacă $a \geq 0$

nD $\quad x^T A x \geq 0 \iff \boxed{A \geq 0}$

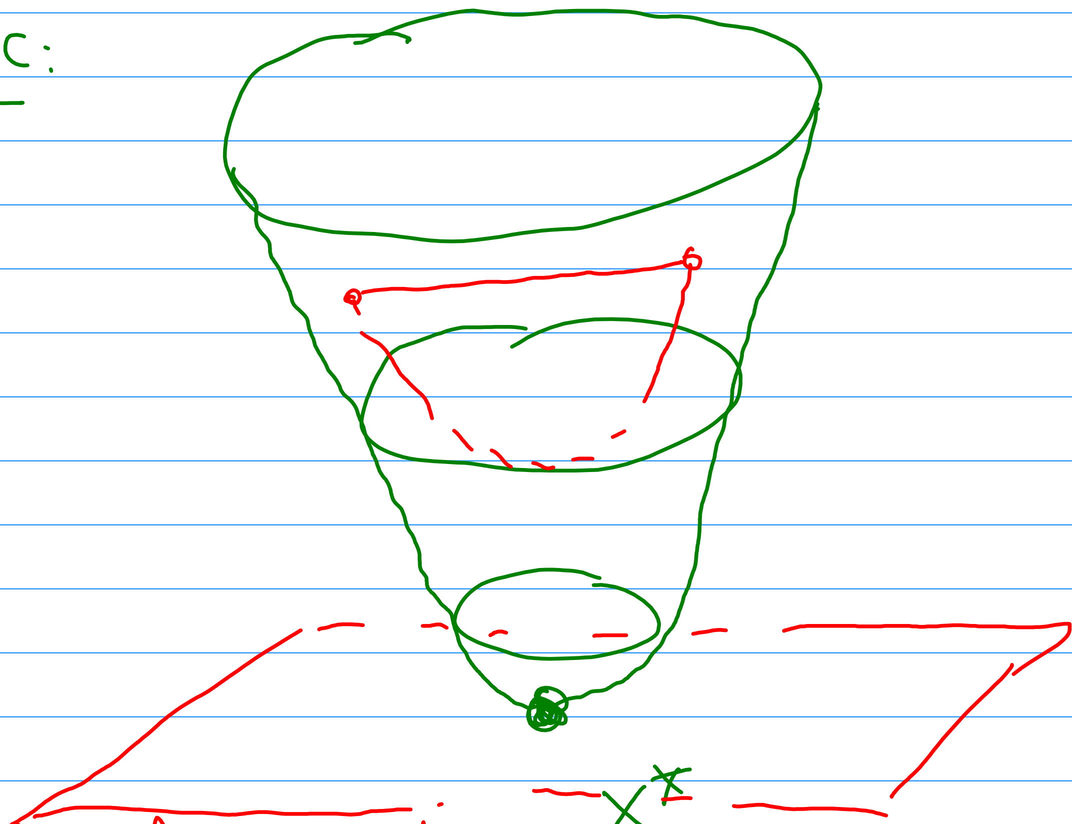
$\forall x$

Dacă x^* este soluție pt.

$\min\limits_{x \in \mathbb{R}^n} f(x)$ atunci $\underline{\nabla f(x^*) = 0}$

<u>ord 1</u>

<u>Grafic:</u>



planul tangent este orizontal

$\Leftrightarrow \nabla f(x^*) = 0$

x^* min local $\rightarrow$ $\nabla^2 f(x^*) \succeq 0$

Gradient descent

Taylor:

$$f(x+\underline{h}) = f(x) + \underbrace{\overset{\in\mathbb{R}^n}{\nabla f(x)} \cdot \overset{\in\mathbb{R}^n}{h}}_{\text{produs scalar}} \overset{?}{<} f(x)$$

Descent direction:

$$\underline{\underline{d \cdot \nabla f(x)}} < 0$$

$$f(x+d) \cong f(x) + \underbrace{\nabla f(x) \cdot d}_{<0} < f(x)$$

Simplu:

$$d = -\nabla f(x)$$

$$(x \cdot (-x) = -x^2 < 0)$$

$$d \cdot \nabla f(x) = -\nabla f(x) \cdot \nabla f(x)$$

$$= -|\nabla f(x)|^2 < 0$$

$$x \cdot x = x_1 \cdot x_1 + \ldots + x_n \cdot x_n$$
$$= x_1^2 + x_2^2 + \ldots + x_n^2$$

Alg optimizare (gradient)

- $x_{i+1} = x_i - t \cdot \nabla f(x_i)$

pas > 0

- trebuie să alegem un pas ?!

line search:

pas fixat $t = t_0$

Exemplu funcția patratică

$$f(x) = \frac{1}{2} x^T A x - b \cdot x$$

A sym, pozitivă, (v.p. > 0)

$$0 < \lambda_1(A) \leq \lambda_2(A) \ldots \leq \lambda_n(A)$$

Numpy: $x \cdot y = \sum_{i=1}^{n} x_i y_i$

Opțiunea 1: np.dot(x, y)

Opțiunea 2: $S = 0$

for $i = 1, \dots, n$

$S = S + x[i] \cdot y[i]$

Opt 2 corectă dar este $\boxed{\text{MULT}}$ mai lentă ca și opt 1

$x^T A x = x \cdot Ax$

np.dot(x, A@x)

$J(x_1, x_2) \rightarrow$ def $J(x)$: | nu depinde de dimensiune

...

def $J(x, y)$:] depinde de dimensiunea problemei

...

$$\nabla J(x) = Ax - b$$

$$A@x - b$$

Gradient descent -> pas optimal