Introduction to Computational Science & Engineering (CSE)

16.0002 / 18.0002

New and improved !
Now in person !!

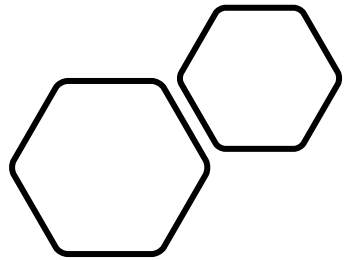
Lecture 8: Optimization II

Laurent Demanet (Math/EAPS)
Youssef Marzouk (AeroAstro)

22 November 2021

Follow along: tests_lec8.py

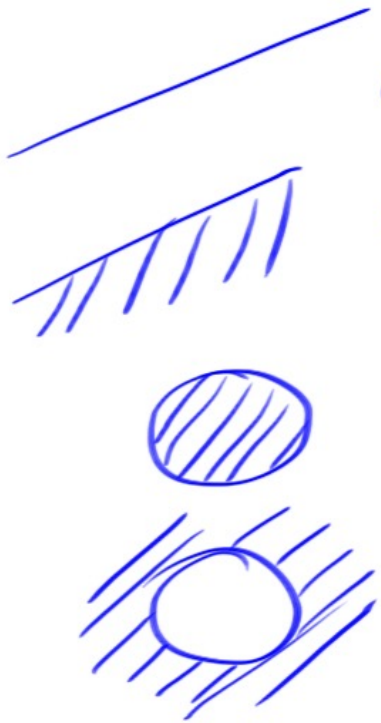
Today: Optimization
Gradient descent
+ plots in python



Unconstrained: $\min_{\underline{x}} J(\underline{x})$ ⟵

Constrained: $\min_{\underline{x} \in S} J(\underline{x})$ $\quad \underline{x} \in \mathbb{R}^2$

S (some set $\quad \underline{x} = (x_0, x_1)$



$a_0 x_0 + a_1 x_1 = b$ linear equality

$a_0 x_0 + a_1 x_1 \leq b$ — inequality

$x_0^2 + x_1^2 \leq 1$ convex constraint

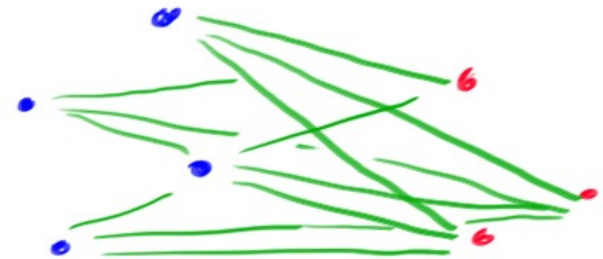
$x_0^2 + x_1^2 \geq 1$ nonconvex constraint.

1) Linear programming

ex. transportation

Supply

Demand



$i = 1,2,3,4$

S_i

Cost c_{ij}

$j = 1,2,3$

D_j

$$\min_x \sum_{ij} c_{ij} x_{ij}$$

$$\text{s.t.} \sum_j x_{ij} \leq S_i$$

$$\sum_i x_{ij} \geq D_j$$

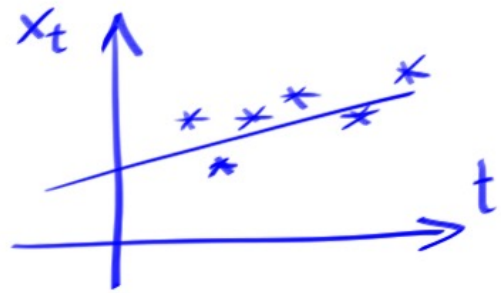
$(\sum D_j \leq \sum S_i)$

2] Quadratic programming

ex. Inverse problems

$$\min \sum_t (x_t - d_t)^2 \quad \leftarrow \text{fits data } d_i$$

$$\text{s.t. } Ax = b \quad \leftarrow \text{obeys model}$$



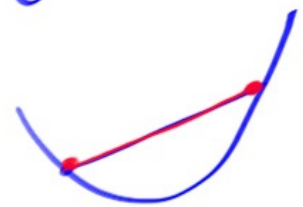
(here $\frac{d^2 x_t}{dt^2} = 0$)

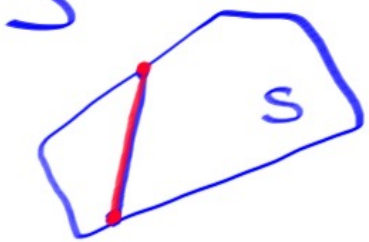
3) Convex programming

$\min J(x)$ ← convex J

s.t. $x \in S$

↑ convex S

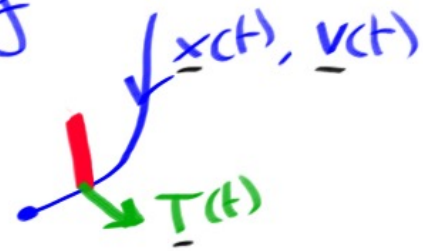




Every minimizer is global!

ex. Rocket landing

Convex in $\underline{x}, \underline{v}, \underline{T}$

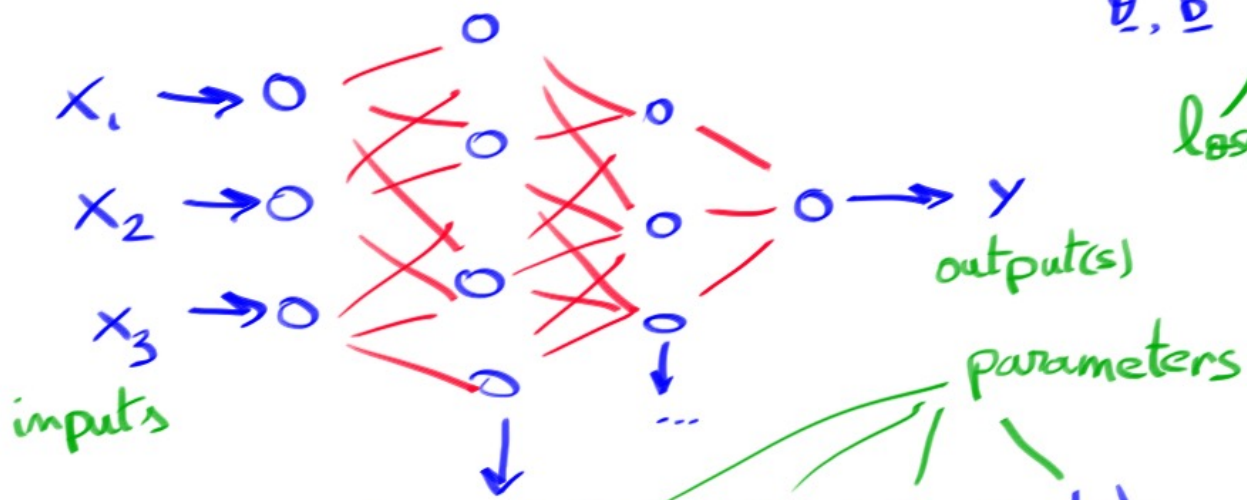


4] Nonlinear programming ("none")

ex. Neural networks

$$\min_{\underline{\theta}, \underline{b}} \; l(y(\underline{\theta}, \underline{b}) - z)$$

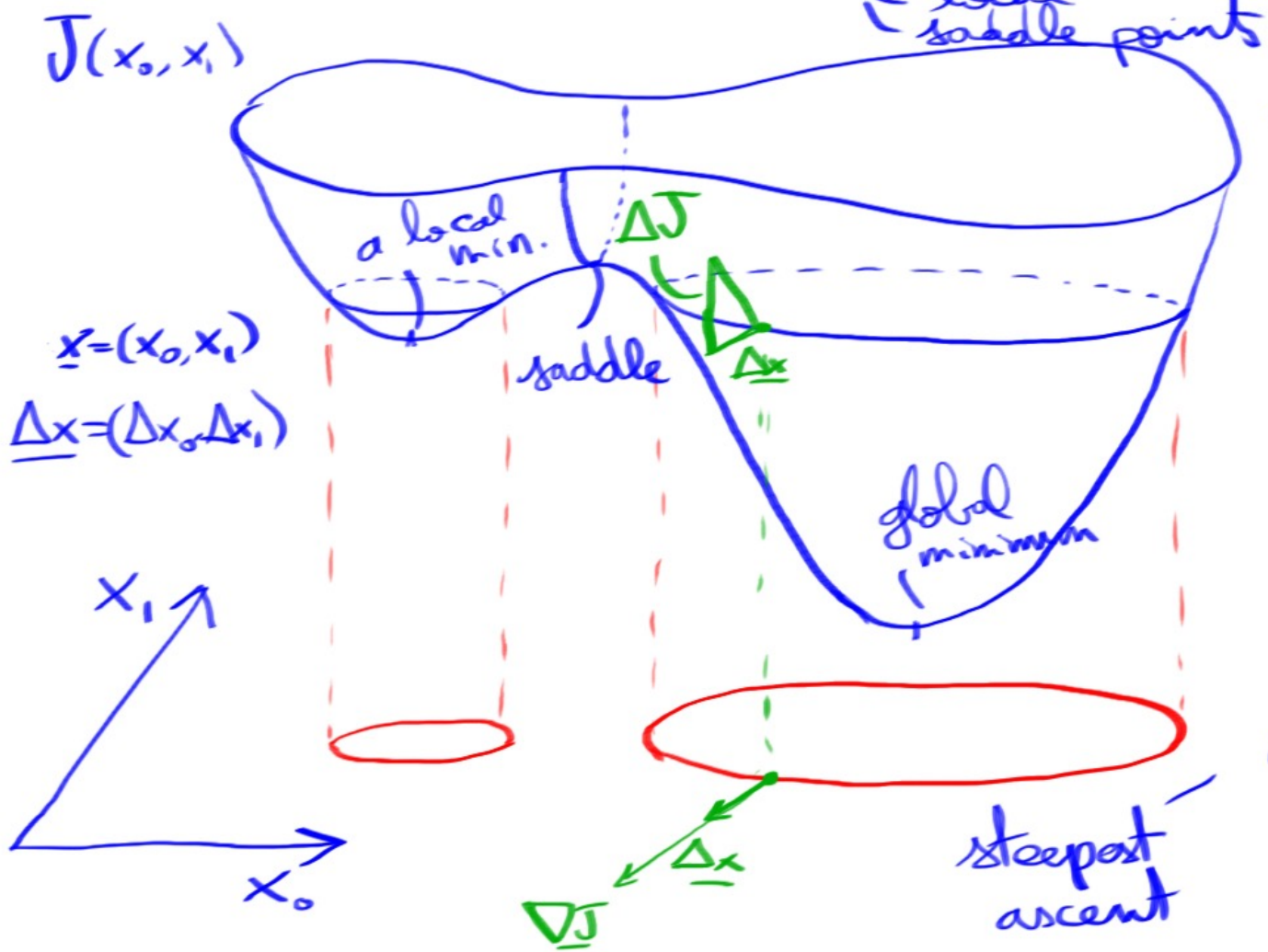
loss — labels



Gradient descent

Extremal points: $\nabla J = 0$

(global min
(local
(saddle points



$$\Delta J = \frac{\partial J}{\partial x_0}\Delta x_0 + \frac{\partial J}{\partial x_1}\Delta x_1$$

$$= \nabla \underline{J} \cdot \underline{\Delta x}$$

$$\left(\begin{bmatrix} \partial J/\partial x_0 \\ \partial J/\partial x_1 \end{bmatrix} \right.$$

∇J, θ, $\underline{\Delta x}$

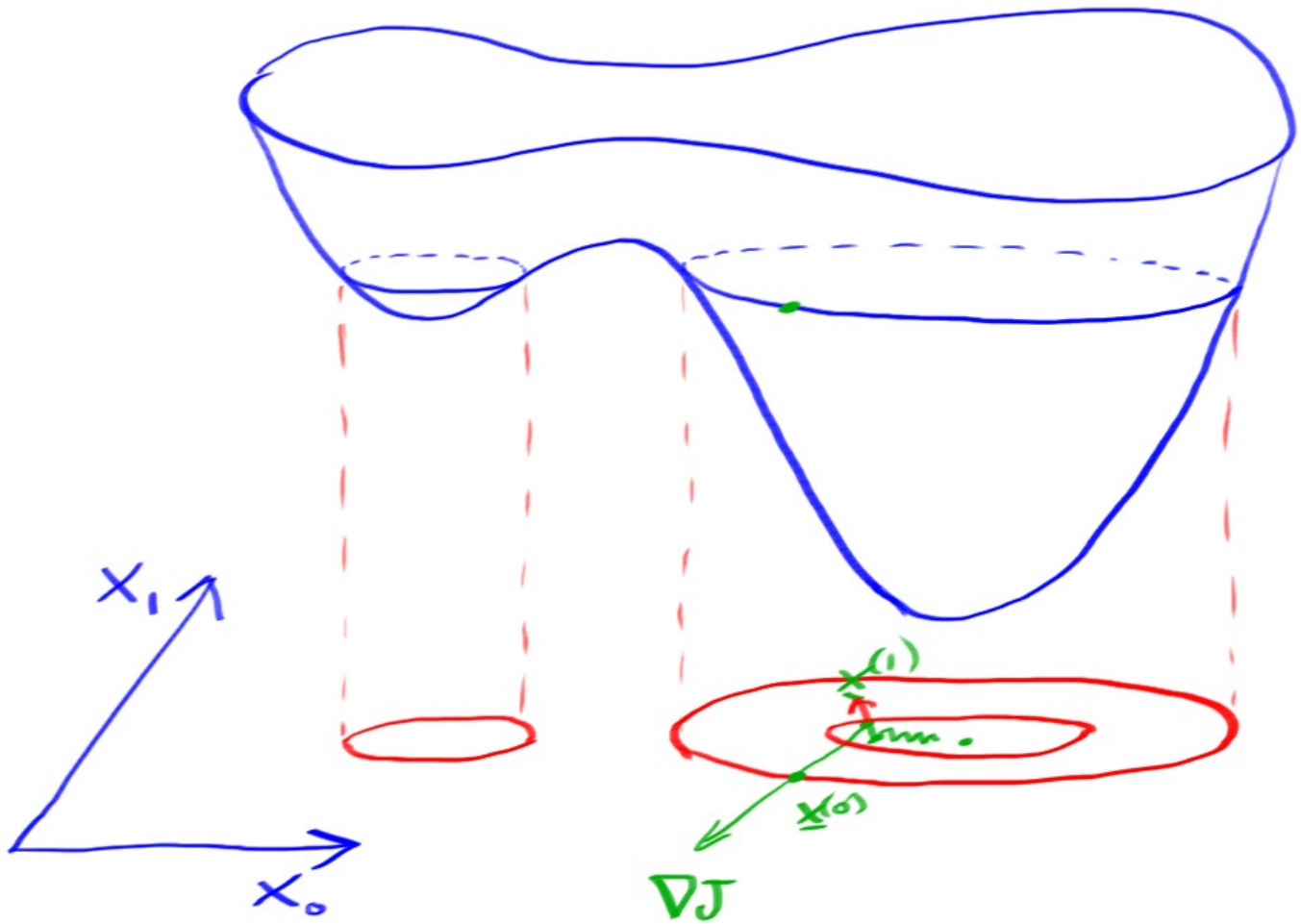
$$= \|\nabla \underline{J}\| \, \|\underline{\Delta x}\| \cos\theta$$

$\|\underline{\Delta x}\| = 1$, $\cos\theta = 1$

$$\underline{\Delta x} = \frac{\nabla \underline{J}}{\|\nabla \underline{J}\|}$$

$\theta = 0$

Gradient descent.



$$\underline{x}^{(0)} = (x_0, x_1)$$

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \alpha \nabla J(\underline{x}^{(k)})$$

(α: step length, $\alpha > 0$)

"line search"

Newton's method (optimization!)

Replace with

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - \alpha \nabla \underline{J}(\underline{x}^{(k)})$$

$$\underline{x}^{(k+1)} = \underline{x}^{(k)} - H^{-1} \nabla \underline{J}(\underline{x}^{(k)})$$

$$\nabla \underline{J} = \begin{bmatrix} \partial J/\partial x_0 \\ \partial J/\partial x_1 \end{bmatrix}$$

solution u to

$$H u = \nabla \underline{J}(\underline{x}^{(k)})$$

HESSIAN:

$$\begin{bmatrix} \partial^2 J/\partial x_0^2 & \partial^2 J/\partial x_0 \partial x_1 \\ \partial^2 J/\partial x_0 \partial x_1 & \partial^2 J/\partial x_1^2 \end{bmatrix}$$