Spin-Off del Politecnico di Torino

OPTIMAD engineering

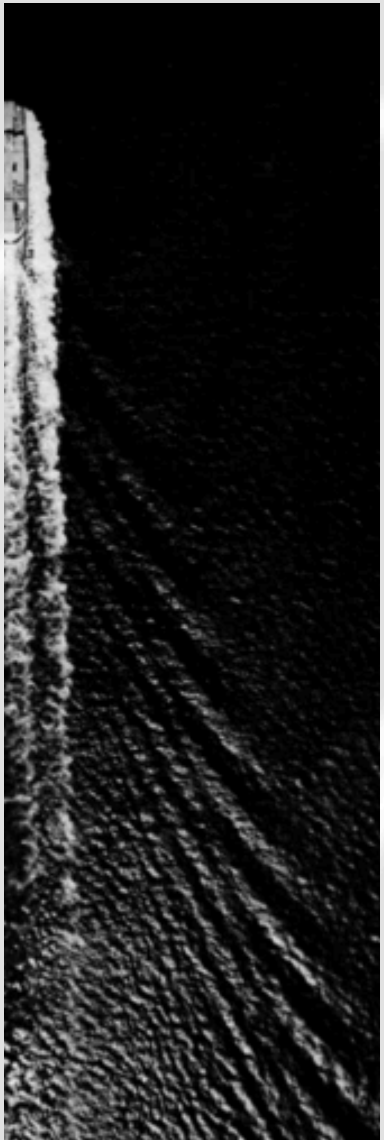
Design Optimization and Design Exploration using an Open Source Framework on HPC facilities

Joel GUERRERO Wolf Dynamics, Genova

Haysam TELIB OPTIMAD engineering, Turin

HPC methods for engineering, @CINECA, June 2015

Outline



1. Introduction to DE/DO; the role of HPC
2. Geometry parametrization
3. *Data Analytics **
4. *Open Source Framework **
5. *Conclusions **

*** presented by JG,** wolf dynamics

Spin-Off del Politecnico di Torino

Aerodynamic shape optimization is done only if strategic

Spin-Off del Politecnico di Torino

2 major burdens to aerodynamic shape optimization

1. **Difficult**
   - you have to know/explore a lot (physics, uncertainty, parameterization)
   - automatize everything (geometry creation, pre /processing/ post)
   - *especially critical if at advanced design due to project constraints and time*

2. **Expensive**
   - computing resources sized for analysis
   - licenses CAD, CFD…
   - specialized technical staff

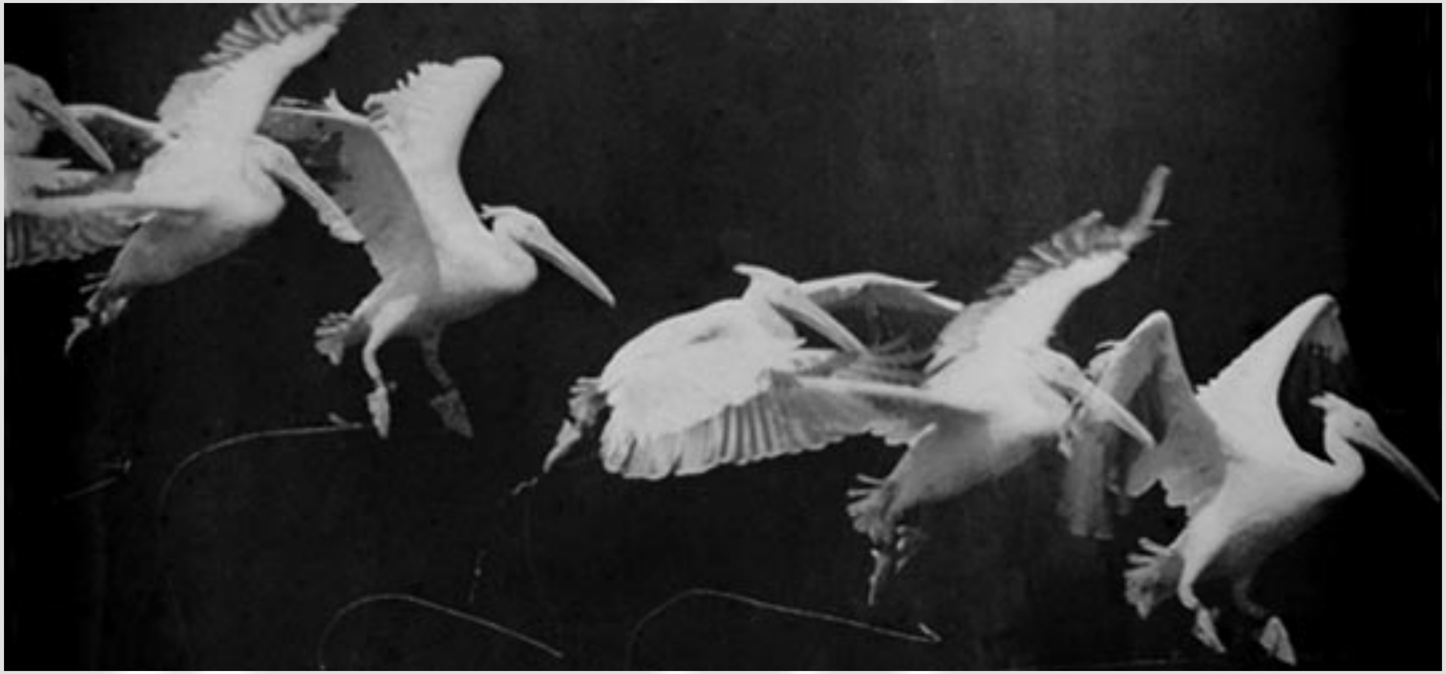
(on demand) HPC

FFD parameterization

OS CAE software

Data Analytics for DE

Spin-Off del Politecnico di Torino



Free-Form Deformation using Level-Sets

Free-Form deformation vs CAD-in-the-loop

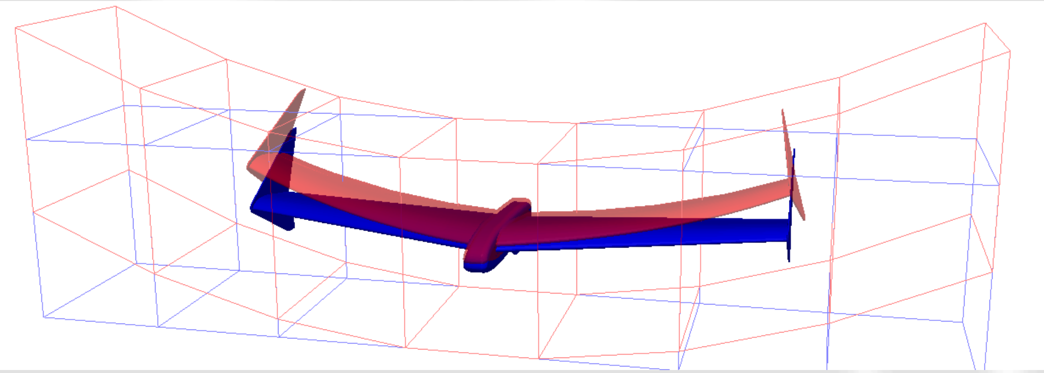
- CAD software often not suited for HPC (licenses and platform)
- ad-hoc CAD needs to be created (which parameters?) -> critical if at advanced design phase

- CAD needs to be re-created from FFD surface mesh
- FFD has difficulty to impose manufacturing constraints

Surface Constraints via Level-Set information

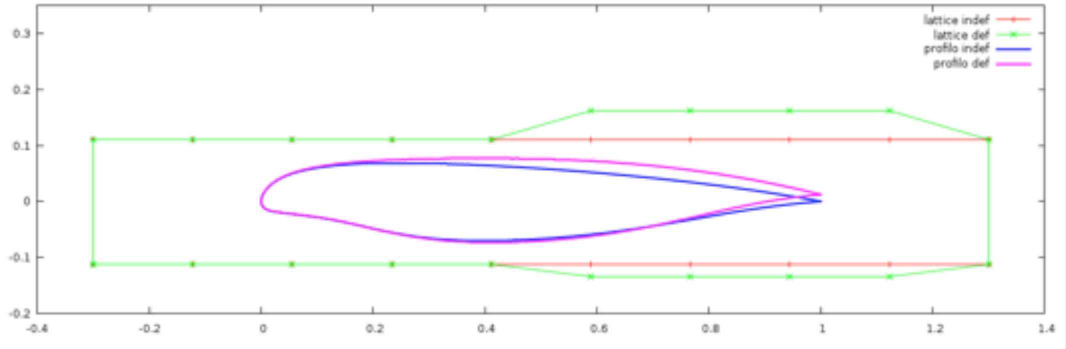
Free-Form Deformation applies a displacement vector $N_i = S_i + D(S_i)$;

- excellent for global deformations



Difficult to impose localized deformations

- often addressed by constraining the CP or ad-hoc shape functions
- ineffective if boundaries do not correspond to the deformation kernel



Surface Constraints via Level-Set information

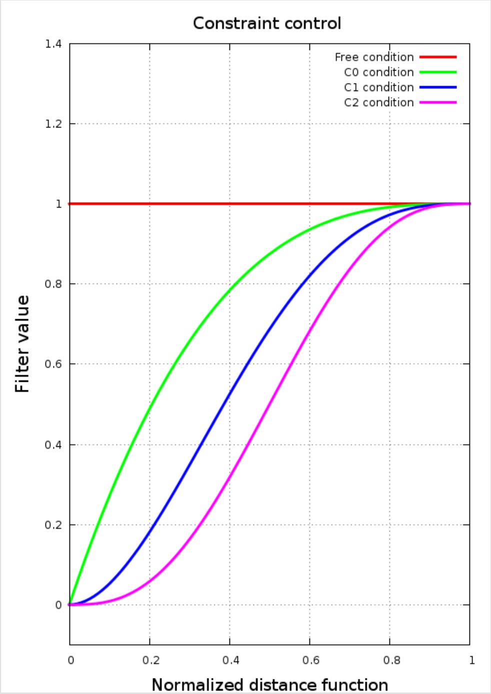
Our approach introduces a weight function

$N_i = S_i + w[\, LS(S_i|\mathbf{\Gamma})\,]\, D(S_i)$

with

Γ boundary non/deformable

$LS(S_i|\Gamma)$ represents a **topological information of S_i wrt to Γ** and must satisify **$LS(\Gamma|\Gamma) = 0$**



with $w(0) = 1$ for G^{-1} condition

with $w(0) = 0$ for G^0 condition

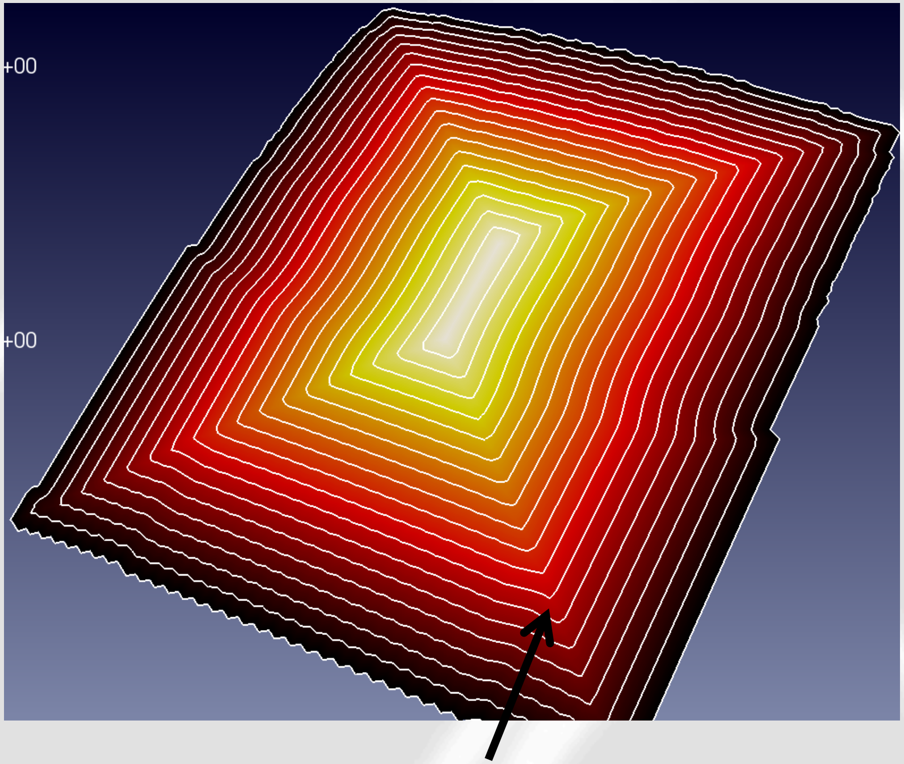
with $w(0) = 0\,,\; w'(0) = 0$ for G^1 condition

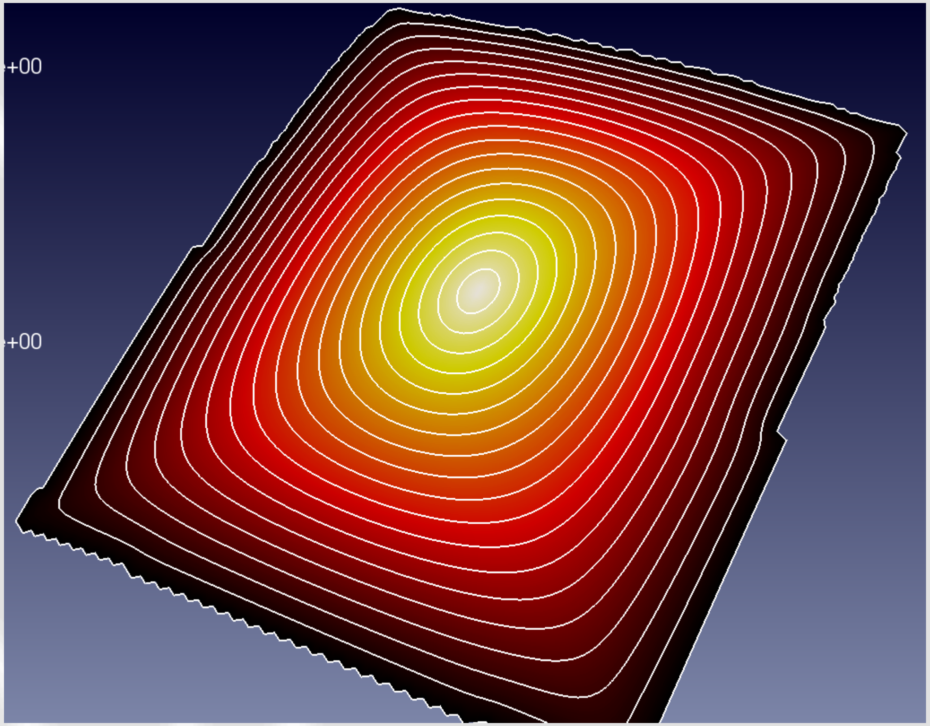
with $w(0) = 0\,,\; w'(0) = 0\,,\; w''(0)=0$ for G^2 condition

The weight kernels are chosen in order to **minimize the risk of inflection** (change of curvature) but cannot be guaranteed

$LS(S_i|\Gamma)$ must be **C^p if G^p continuity** is required

Toplogical information: Eikonal equation vs heat kernel

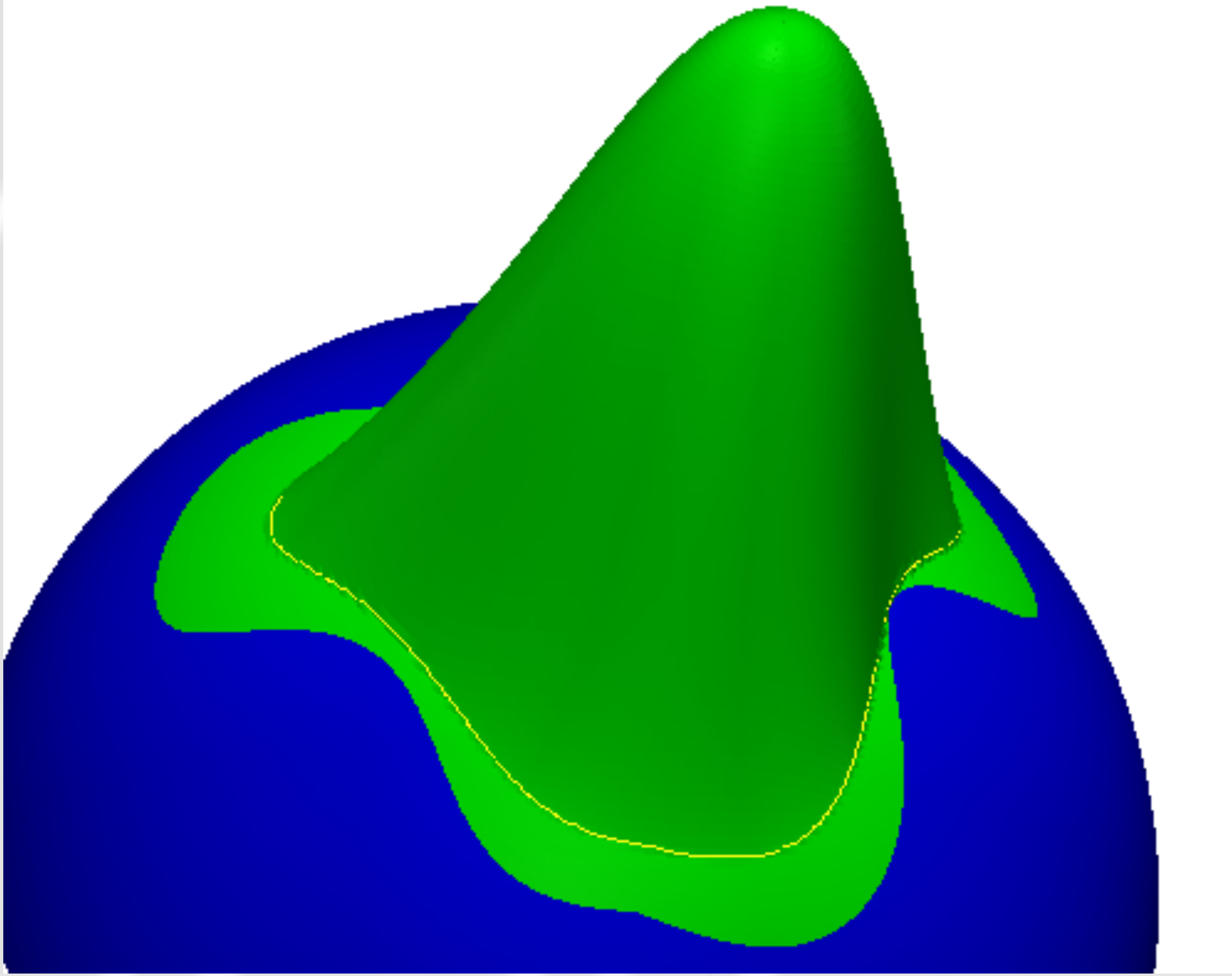




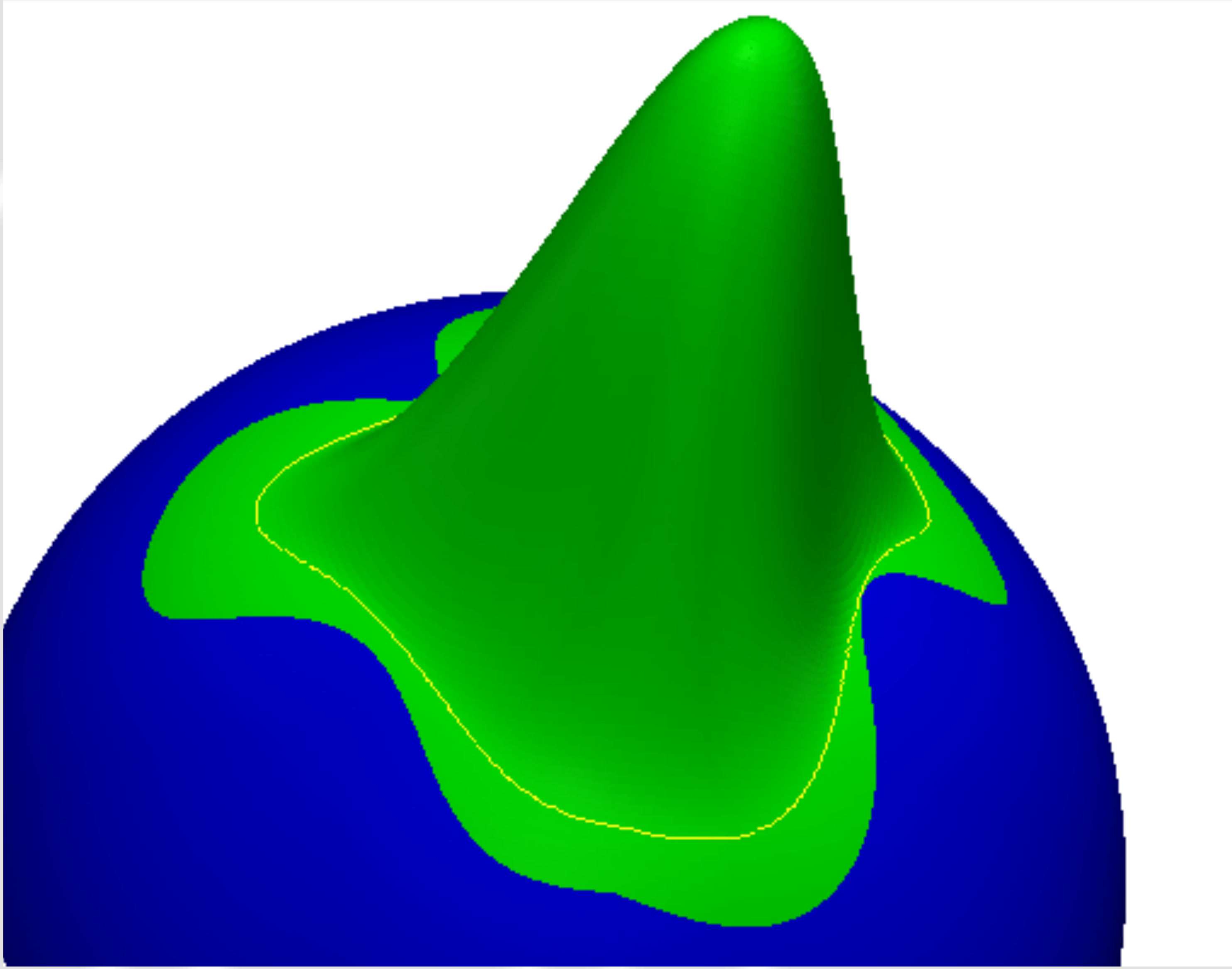
- distance function wrt to boundary
- C^0 function
- ca 0.01s for 100K triangles on Intel Xeon

- as close as possible to geodesic LS to keep topology
- constraint on C^2 continuity
- ca 3s for 100K triangles on Intel Xeon

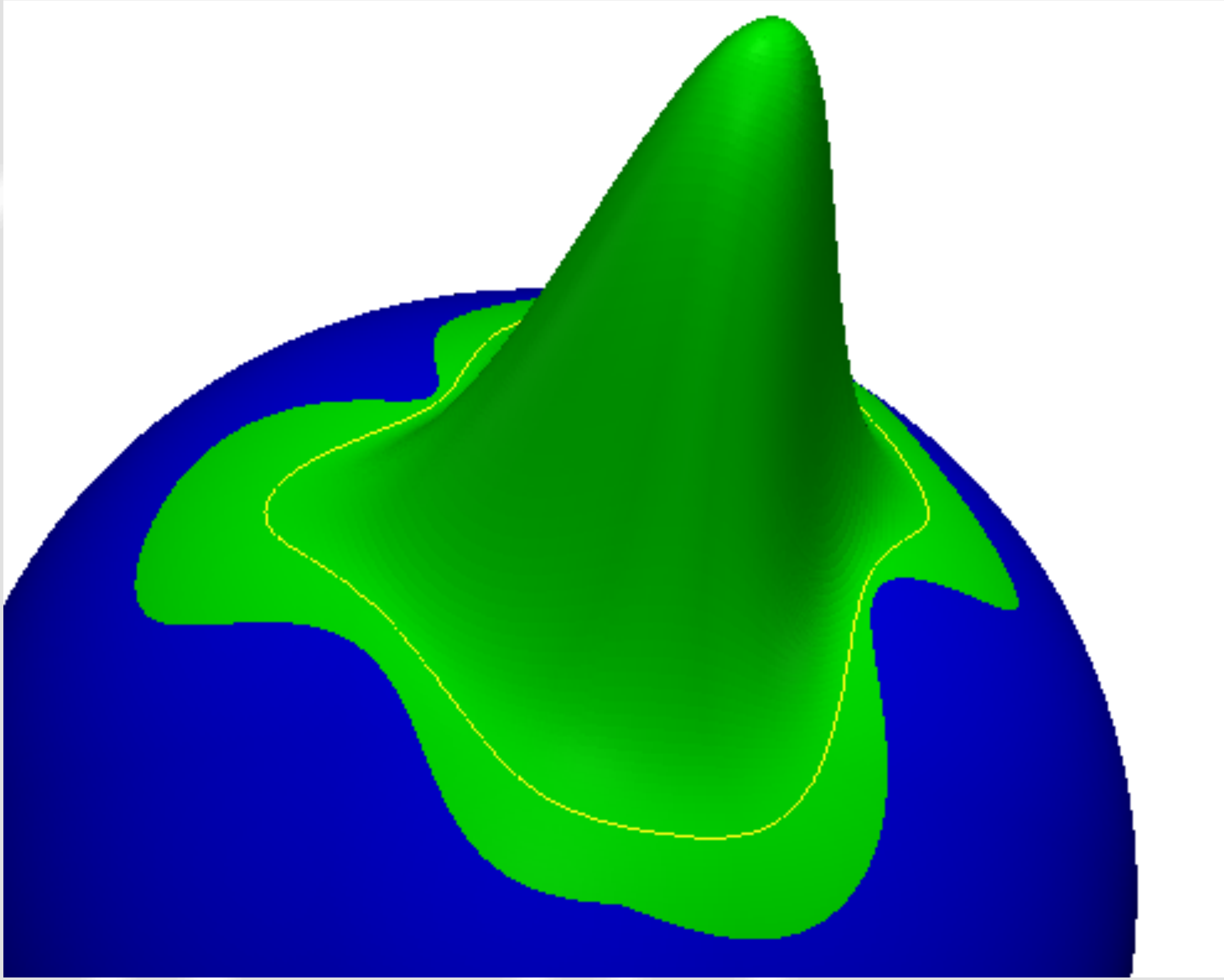
Deformation using C^0 constraint



Deformation using C^1 constraint



Deformation using C^2 constraint



Control of penetration

Distance to a given surface should be maintained

- bounds on CP non-intuitive and often not efficient

User should indicate only intuitive information

- surface (open or closed)
- distance to be maintained

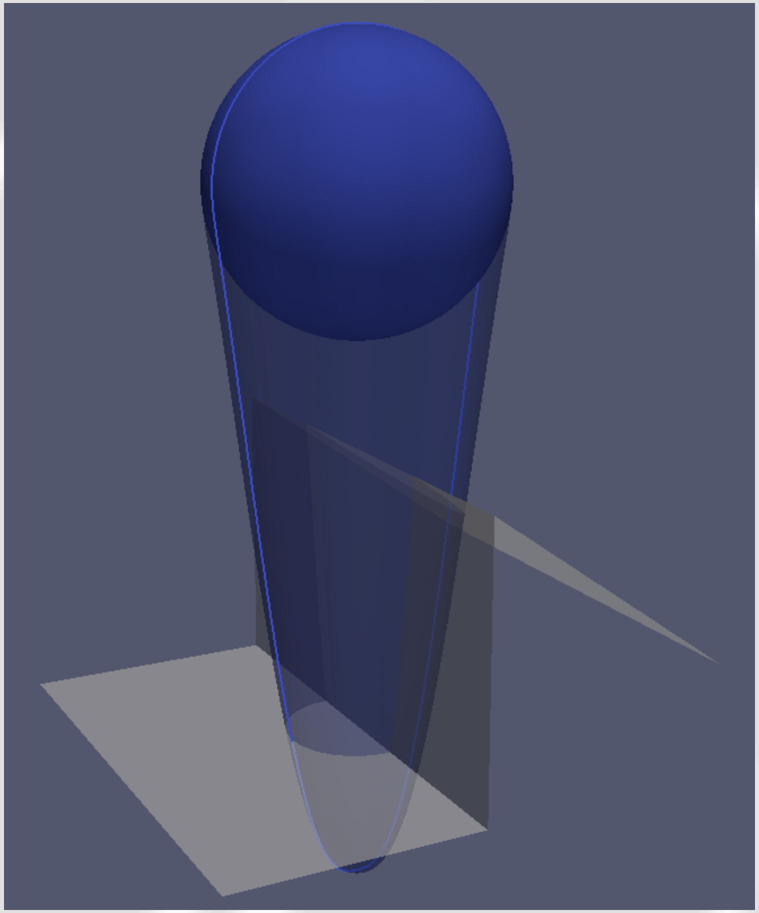
Combined control algorithm

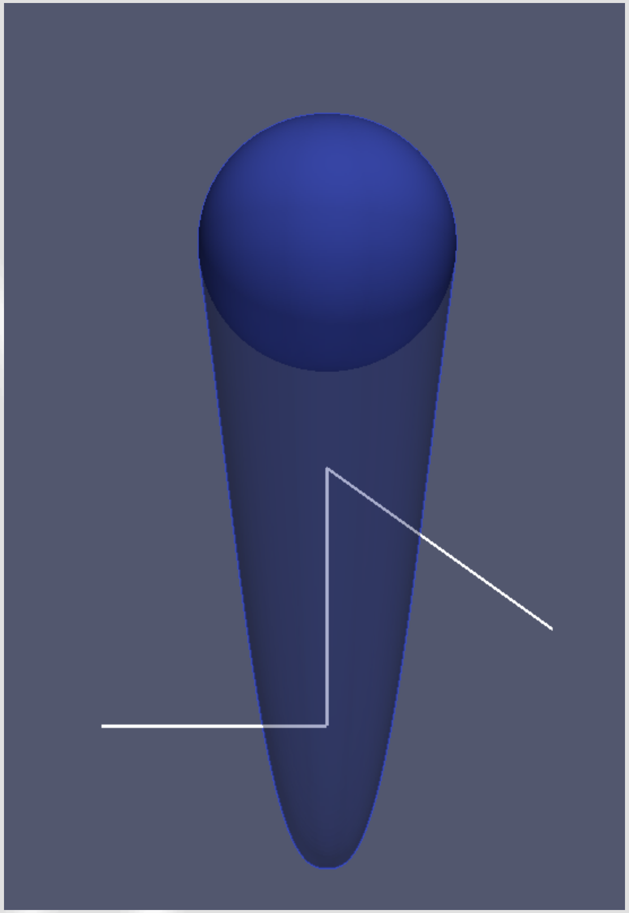
- Ray-tracing
- Level Set

Two different types of rescaling algorithms available

- choice depends on case considered

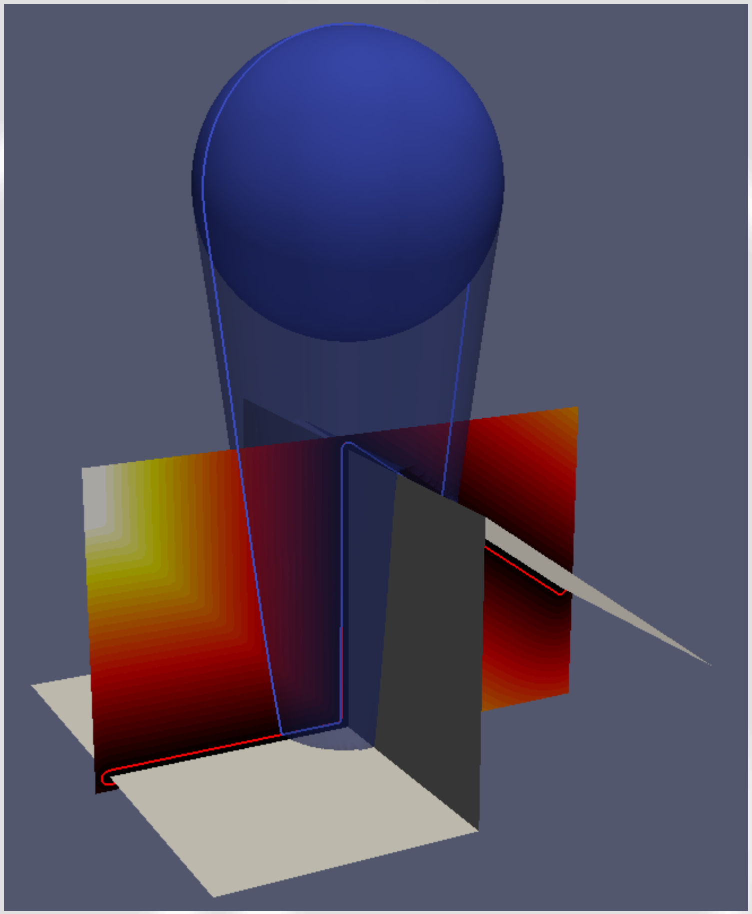
Control of penetration – Controll off

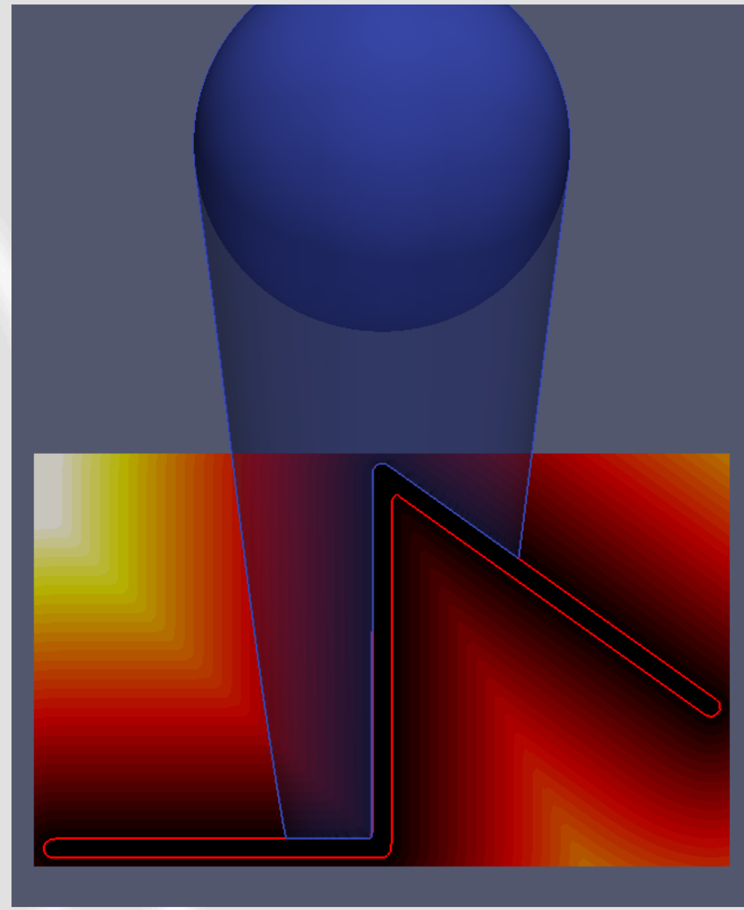




Spin-Off del Politecnico di Torino

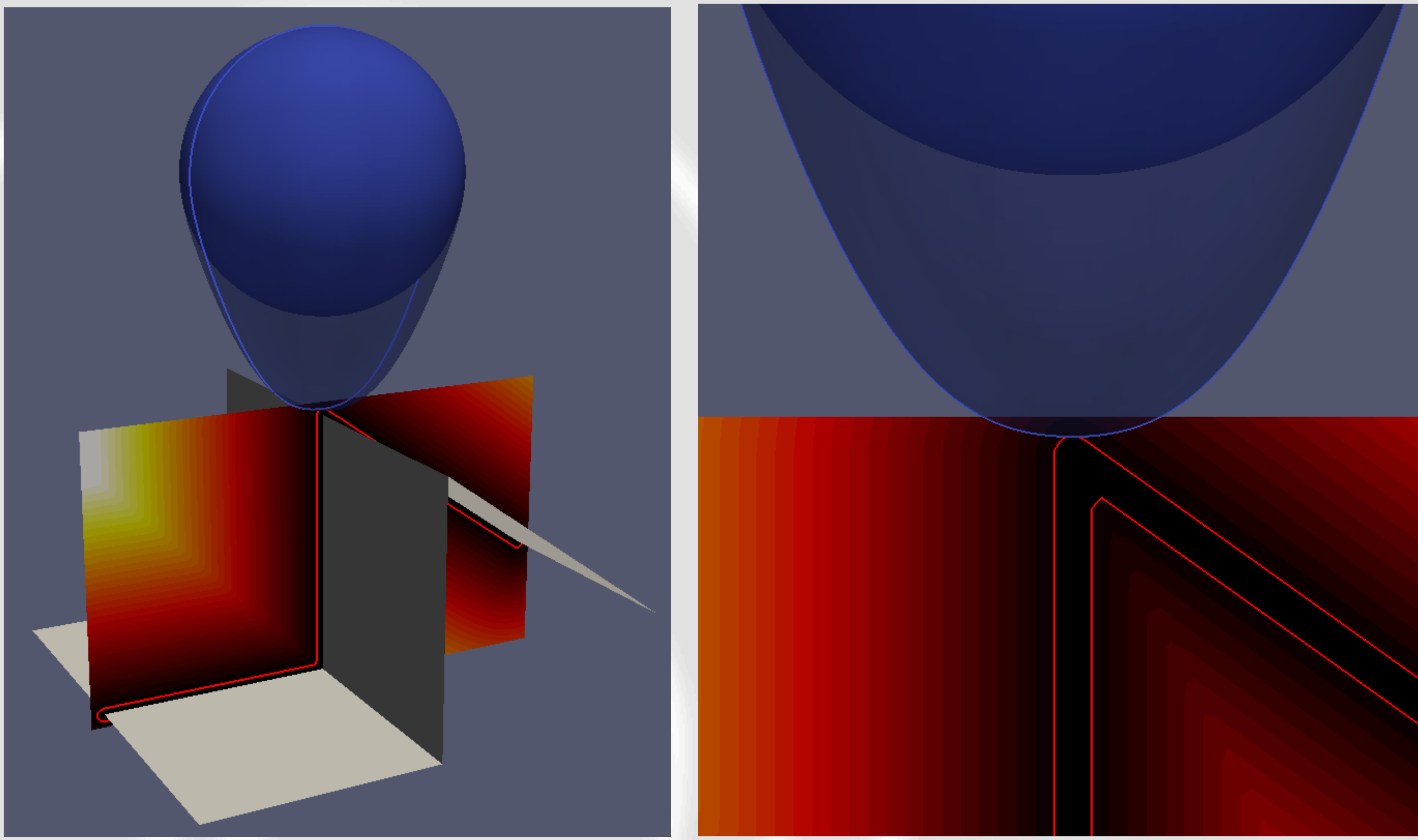
Control of penetration – Local rescaling





distance wrt tangential projection

Control of penetration – Global rescaling



identification of critical point

CAMILO – description of work flow

1. install **GUI on workstation** and put **executable on cluster**
2. import **geometry and surfaces** to be used as constraints in GUI
3. play with GUI in order **to impose parameterization and constraints**
4. export **control and load file** and copy file on cluster
5. **let Dakota change load file** and call executable with **control and load file as argument** -> modified geometry file (e.g. stl)
6. *… pass geometry to pre-processing …*

end of my part…..

→>>> Joel G

→>>> happy to discuss any question at the end of JG's part

Spin-Off del Politecnico di Torino