Towards fully automated aerodynamic shape optimization of nonplanar wings with SU2

Luca Pustina[ii], **Rauno Cavallaro**[i], **Giovanni Bernardini**[ii], **Rocco Bombardieri**[i]

[i] Bioengineering and Aerospace Engineering Department, University Carlos III of Madrid
[ii] Department of Engineering, University of Roma Tre

uc3m | Universidad Carlos III de Madrid

ROMA TRE UNIVERSITÀ DEGLI STUDI

AGENDA

I. **Introduction: need for aerodynamic shape optimization of nonplanar configurations**

II. SU2 and Aerodynamic Shape Optimization

III. SU2 inside MDAO: NERONE

IV. Conclusions and future works

Introduction

Need for aerodynamic shape optimization of nonplanar configurations

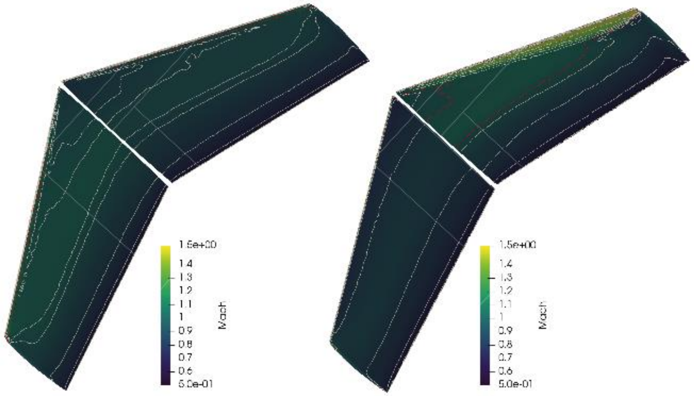
Sustainability

- Environmental challenge and increase in the demand due to the need for mobility.
- More efficient aircraft
    - Design: introduction of higher-fidelity MDO ab-initio
    - Unconventional configurations

Aerodynamic Shape Optimization

- The majority of commercial flights do operate in transonic regimes
    - properly modeling transonic flow physics is costly, and also difficult;
    - enhanced sensitiveness with respect to small and localized features of the **wing shape;**
    - to exploit the potentials of aerodynamic shape optimization (ASO), a **rich-enough design space** has to be considered;
    - adjoint method is the perfect candidate to perform such optimization sensitivities at a relatively low computational costs (cost of evaluating the gradient is almost independent of the number of design variables).

M = 0.8



AoA = 0 deg **AoA = 3.06 deg**

Introduction

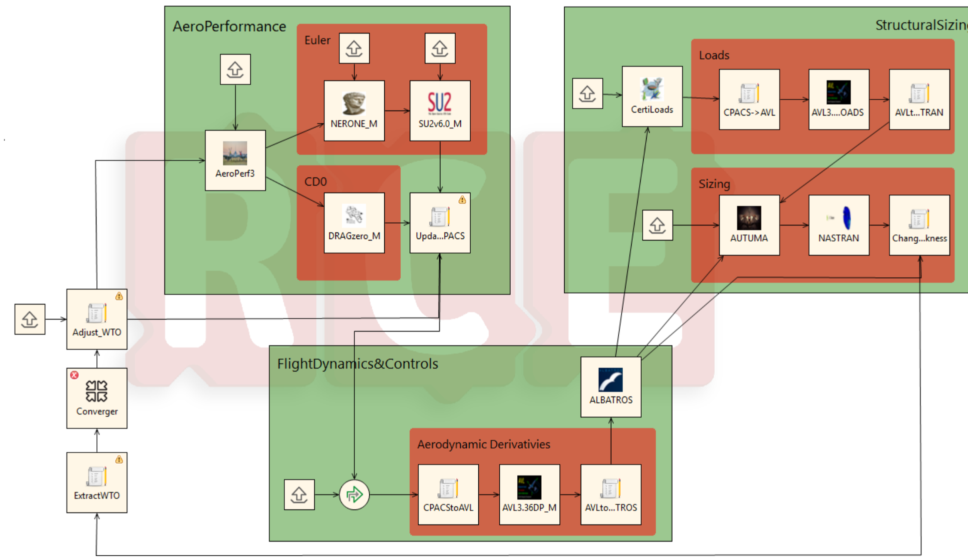
Aerodynamic Shape Optimization inside MDAO

Typically multi-level optimization problems:

- external variation of planform-level DV (sweep, dihedral, etc.
- internal local aerodynamic shape optimization (ASO)

Automatization:

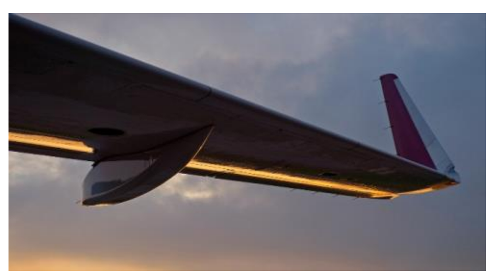
- Parametric description (wing-planform level and local-shape features)
- Robust grid generation (wing-planform level) or deformation (local shape features)
- Aerodynamic Optimization



PARSIFAL
PROJECT

Non-planar wings

- Classic winglets (to reduce drag)
- Box Wings/PrandtlPlane



A320 Sharklet (photo: Airbus)

PrandtlPlane or Box Wing

AGENDA

I. Need for aerodynamic shape optimization of nonplanar configurations

II. **SU2 and Aerodynamic Shape Optimization**

III. SU2 inside MDAO: NERONE

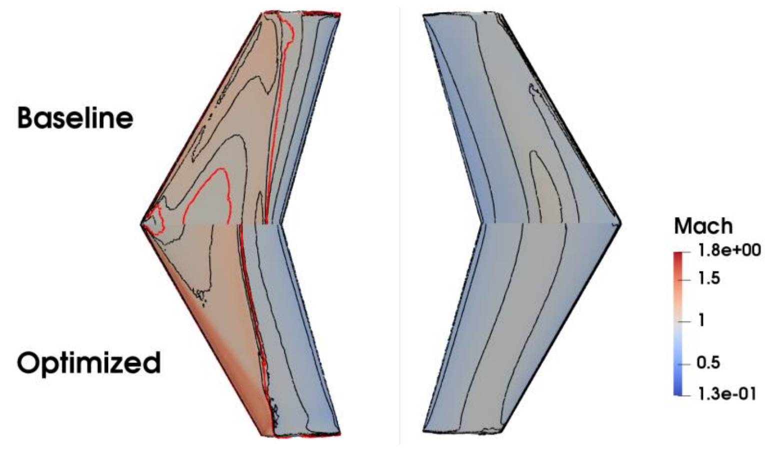
IV. Conclusions and future works

SU2 for Aerodynamic Shape Optimization

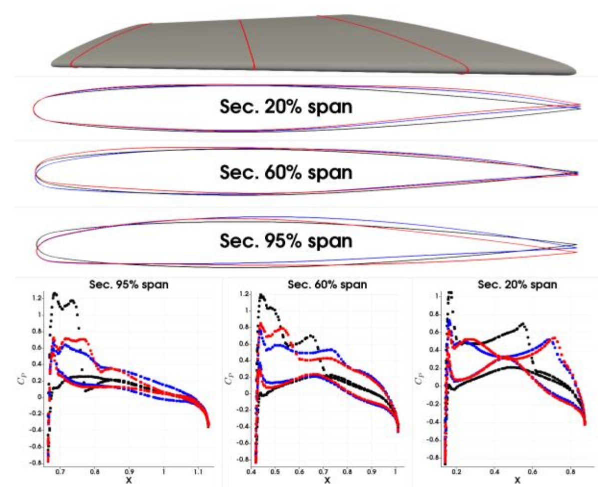
SU2 for Aerodynamic Shape Optimization

- For CFD analyses, continuous and discrete adjoints are provided to support gradient-based optimization.
- Various surface deformation parametrizations and mesh deformation algorithms available.
- High-level routine (python-written) available for aerodynamic shape optimization.

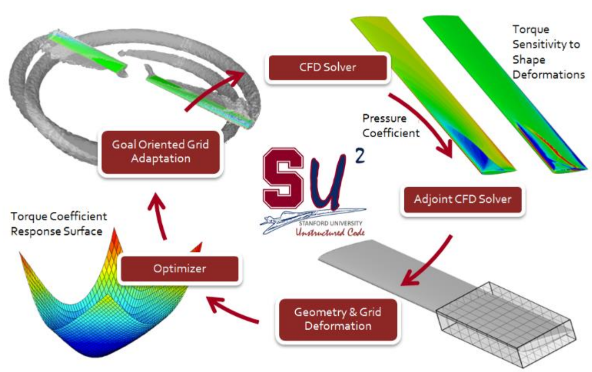
Works well for planar configuration!



Onera M6 testcase

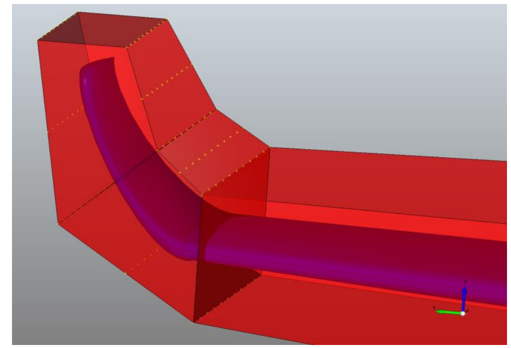


SU2
The Open-Source CFD Code



Issues when considering nonplanar wings:

- Point-inversion not robust.
- Locked position of control points on adjacent FFD boxes.

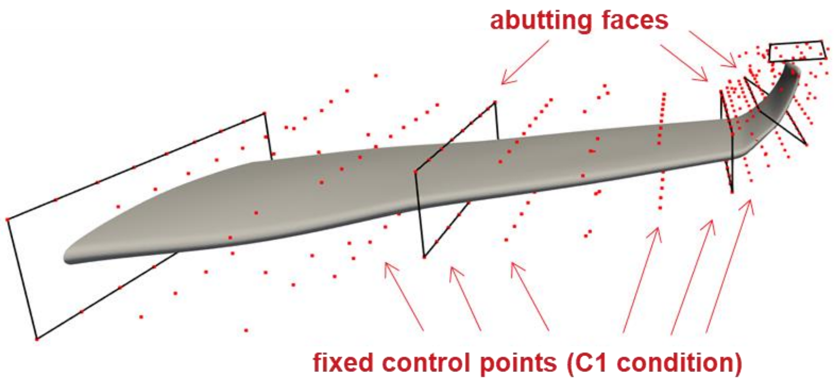


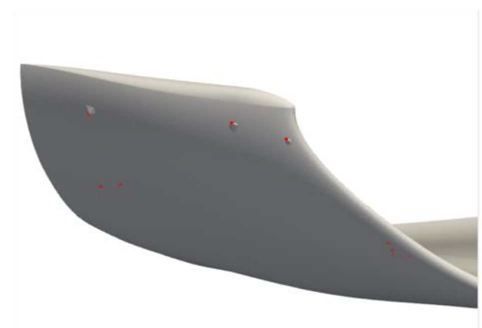
$$\boldsymbol{P} = \sum_{i=0}^{l}\sum_{j=0}^{m}\sum_{k=0}^{n} N_i(\mu)N_j(\chi)N_k(\xi)\boldsymbol{P}_{ijk}$$

surface points

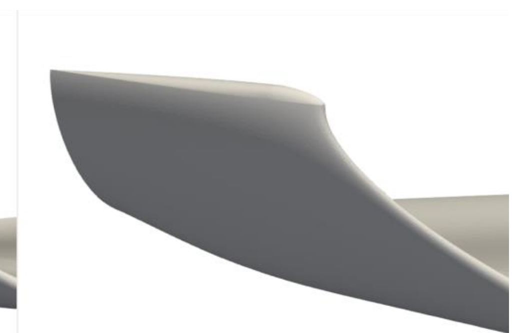
basis functions

Control Points (FFD boxes)





SU2 original code (failed point-inversion)



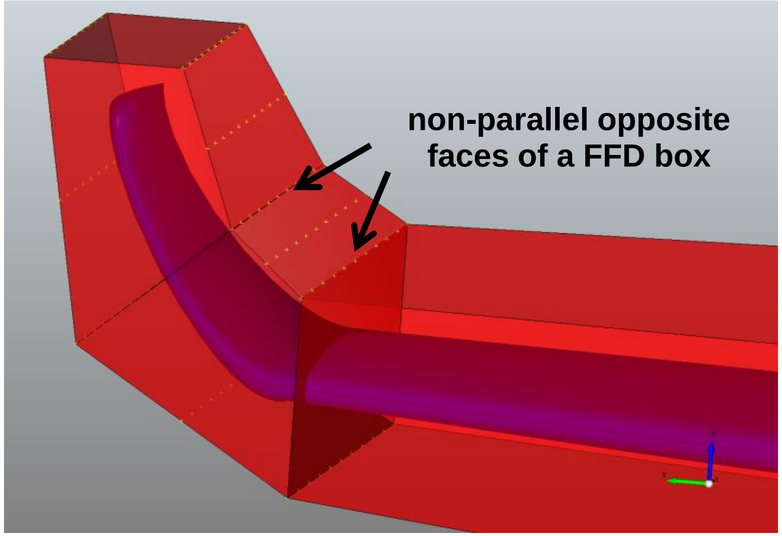
modified SU2 point-inversion

SU2 for Aerodynamic Shape Optimization

Functionalities added to SU2 core

Robust point-inversion to allow non-parallel opposite faces of a FFD box.

- use of genetic algorithm for point-inversion (avoids local minima)



$$\boldsymbol{P} = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} N_i(\mu) N_j(\chi) N_k(\xi) \boldsymbol{P}_{ijk}$$

surface points

basis functions

Control Points (FFD boxes)

Changes in "*grid_movement_structure.cpp*" file, "*CsurfaceMovement::SetParametricCoord*" class member function

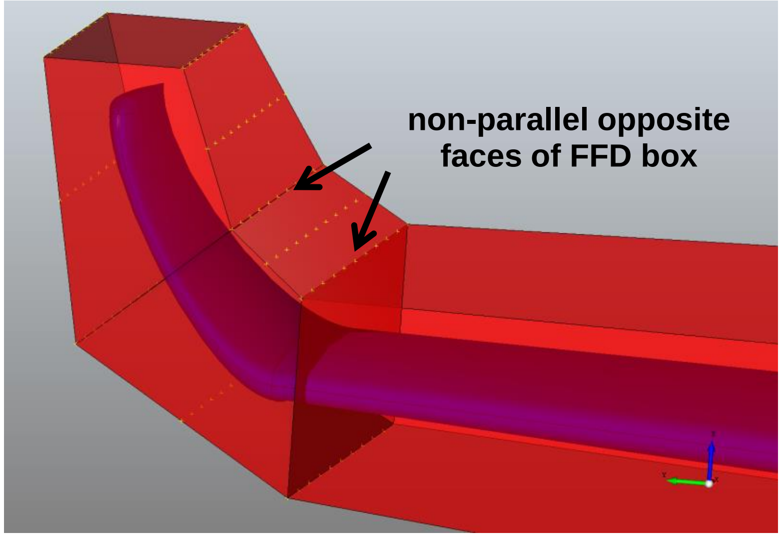
- if the SU2 original point inversion algorithm fails (parametric coordinates not bounded in [0,1]) a new genetic algorithm (based on https://www.iitk.ac.in/kangal/codes.shtml) is called

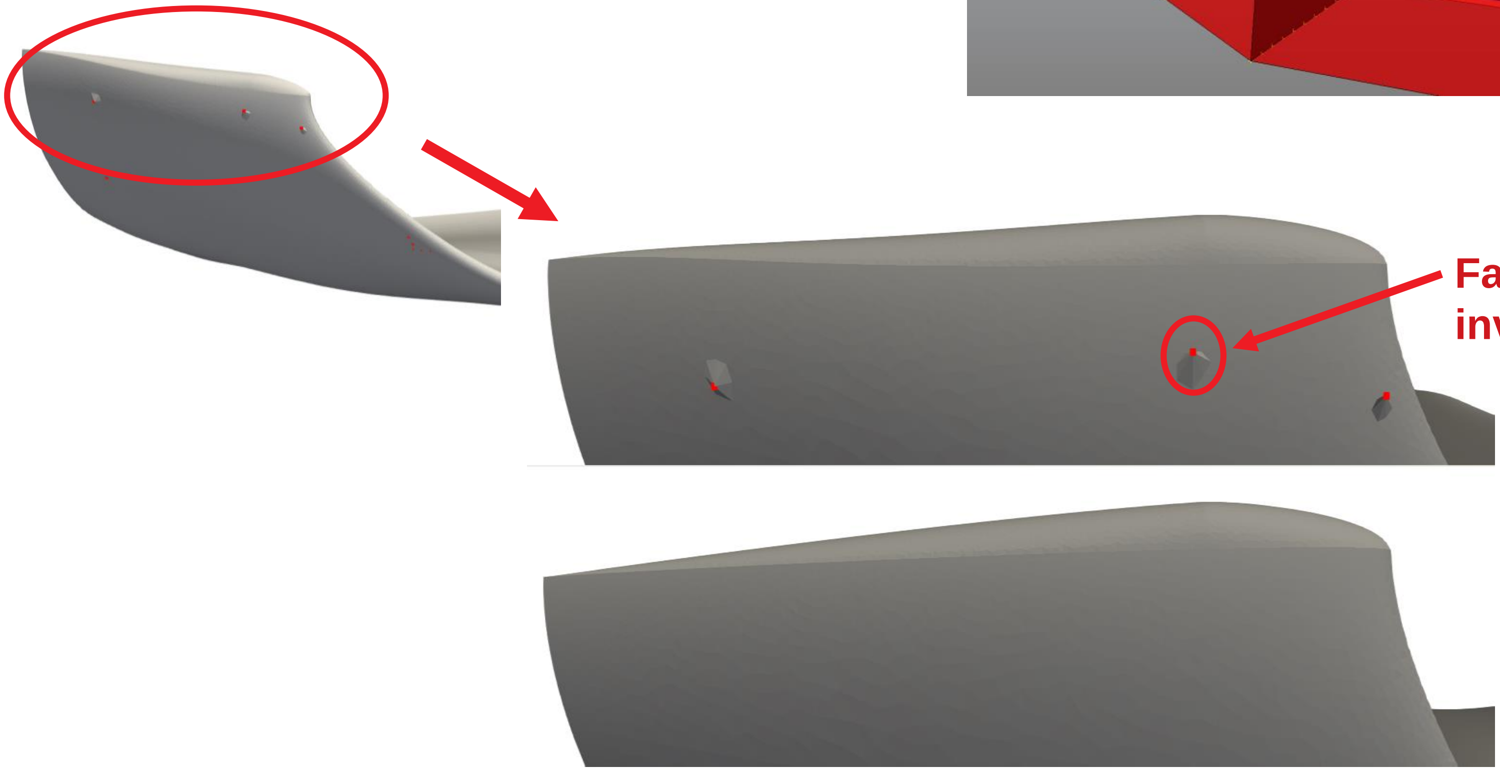
SU2 for Aerodynamic Shape Optimization

Functionalities added to SU2 core

Robust point-inversion to allow non-parallel opposite faces of a FFD box.

- use of genetic algorithm for point-inversion (avoids local minima)





Failed point-inversion

SU2 original code, random deformation.

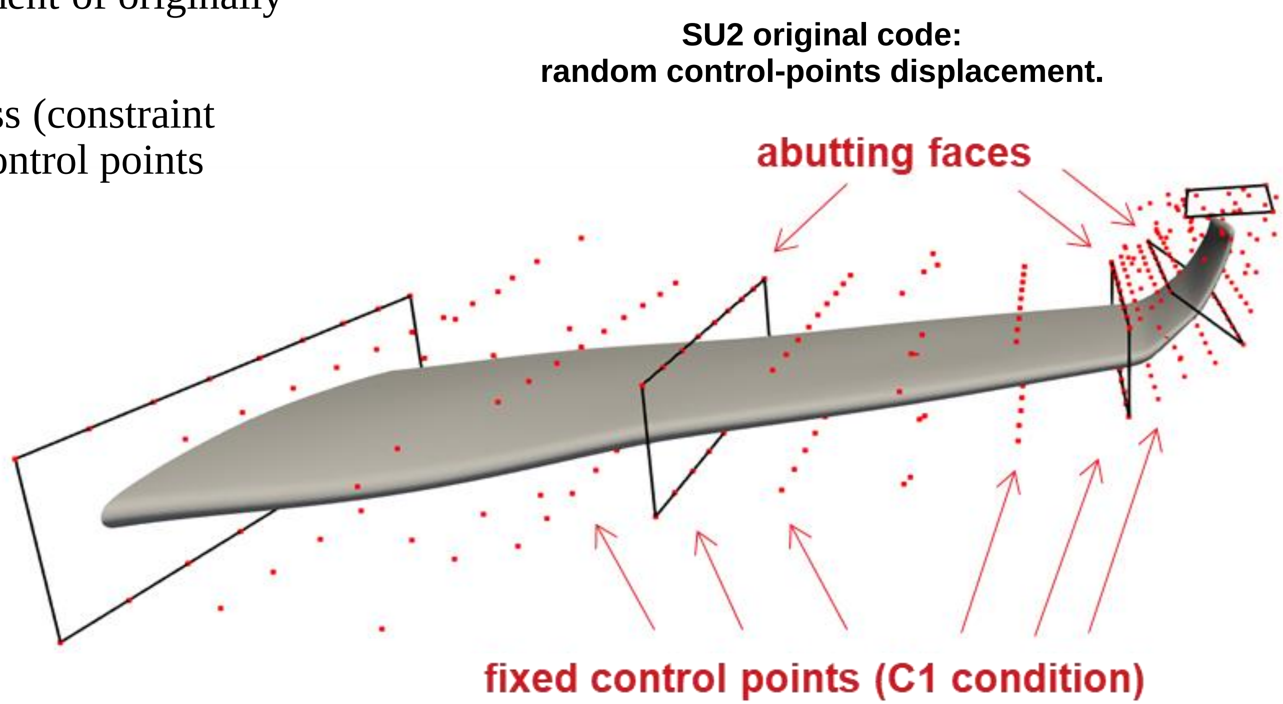
modified SU2, random deformation.

SU2 for Aerodynamic Shape Optimization

Functionalities added to SU2 core

New conditions on adjacent FFD boxes control points:

- Allows for displacement of originally fixed control points.
- Maintains smoothness (constraint equations between control points displacements).



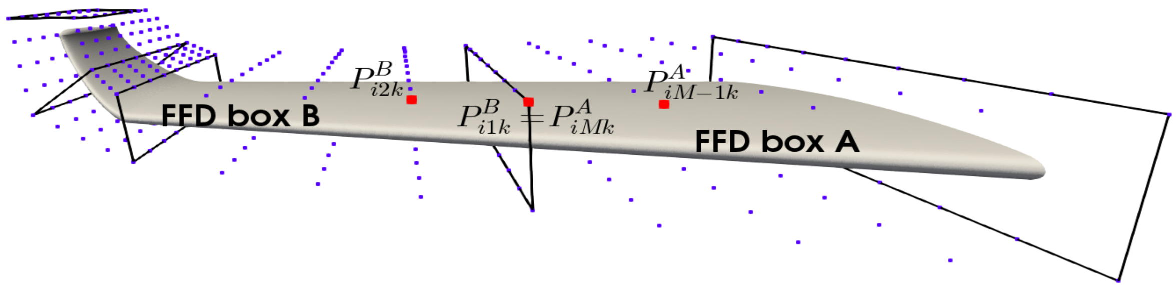
Have been modified:

- python SU2-optimizer interface "*scipy_tools.py*"
- input file and parser "*config.py*"

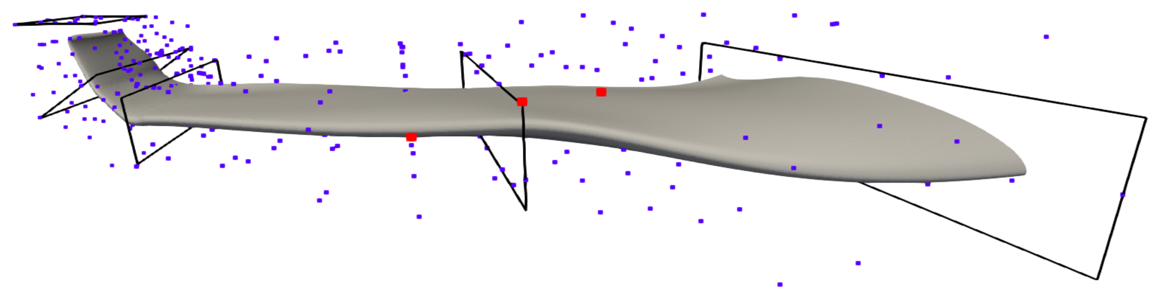
SU2 for Aerodynamic Shape Optimization

Functionalities added to SU2 core

New conditions on adjacent FFD boxes control points.



SU2 modified code: random control-points displacement.



- Continuity requirements on control points up- down-stream the shared FFD interfaces to guarantee a certain level of smoothness

$$\delta \boldsymbol{P}^B_{i1k} = \delta \boldsymbol{P}^A_{iMk} = \frac{\delta \boldsymbol{P}^A_{iM-1k} + \delta \boldsymbol{P}^B_{i2k}}{2}$$

- Dependent design variables (control points) eliminated from the independent DVs.

$$J = J(\delta \boldsymbol{P}^A_{ijk},\ \delta \boldsymbol{P}^B_{rst},\ \delta \boldsymbol{P}^A_{iMk}(\delta \boldsymbol{P}^A_{iM-1k},\ \delta \boldsymbol{P}^B_{i2k}),\ \delta \boldsymbol{P}^B_{i1k}(\delta \boldsymbol{P}^A_{iM-1k}, \delta \boldsymbol{P}^B_{i2k})\ )$$

- Gradient projection adapted to reflect the constraint.

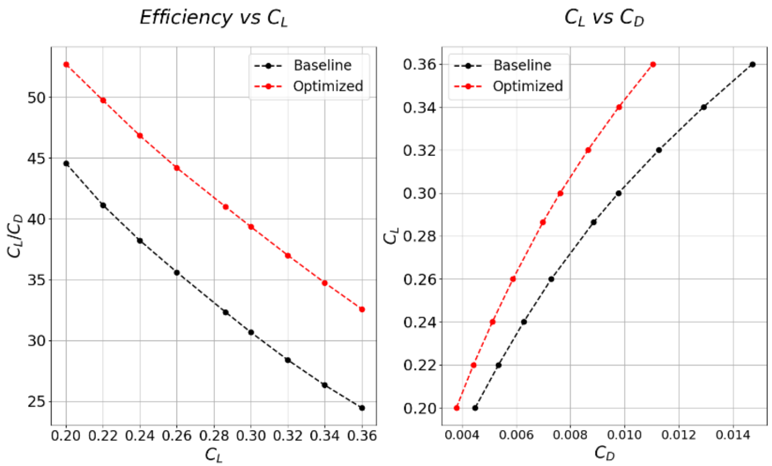
$$\frac{dJ}{d\delta \boldsymbol{P}^A_{iM-1k}} = \frac{\partial J}{\partial \delta \boldsymbol{P}^A_{iM-1k}} + \frac{\partial J}{\partial \delta \boldsymbol{P}^A_{iMk}} \frac{\partial \delta \boldsymbol{P}^A_{iMk}}{\partial \delta \boldsymbol{P}^A_{iM-1k}} + \frac{\partial J}{\partial \delta \boldsymbol{P}^B_{i1k}} \frac{\partial \delta \boldsymbol{P}^B_{i1k}}{\partial \delta \boldsymbol{P}^A_{iM-1k}}$$

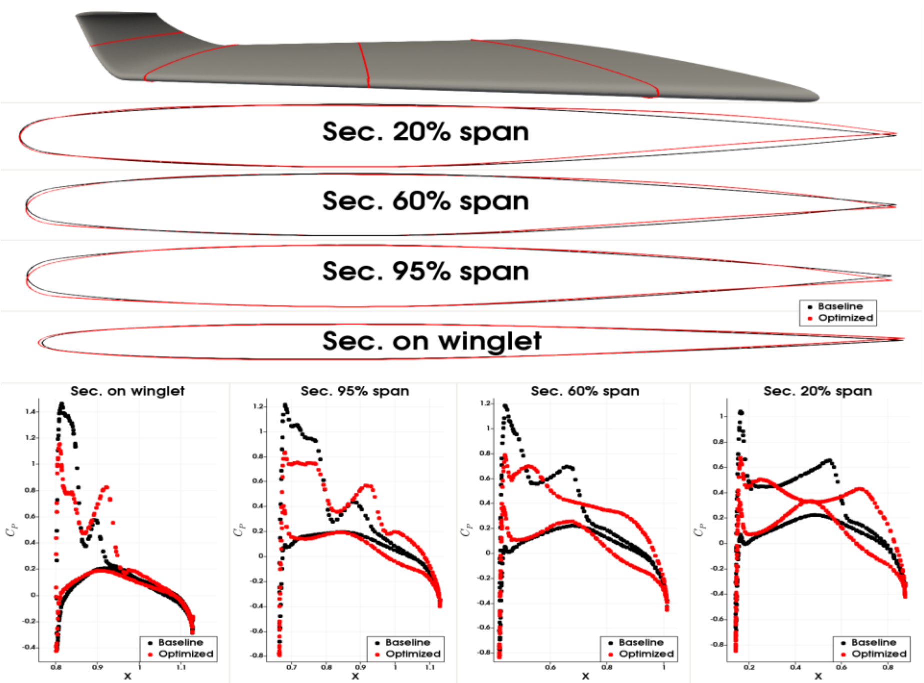
SU2 for Aerodynamic Shape Optimization

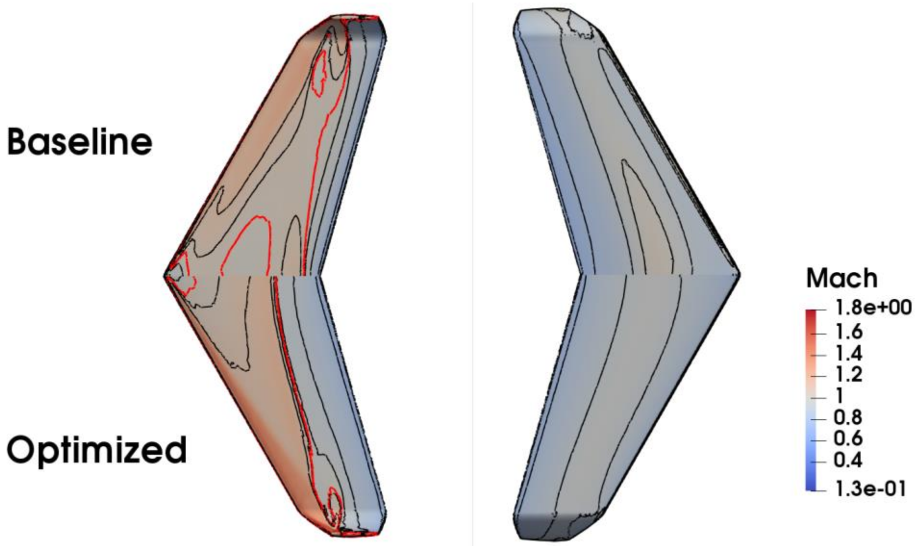
Performance of modified SU2

Synthetic testcase, based on ONERA M6 with added winglets:

- Euler-based
  - multi-point optimization (several C_L, fixed M=0.84) ;
  - t/c and C_M (pitching moment) constraints.





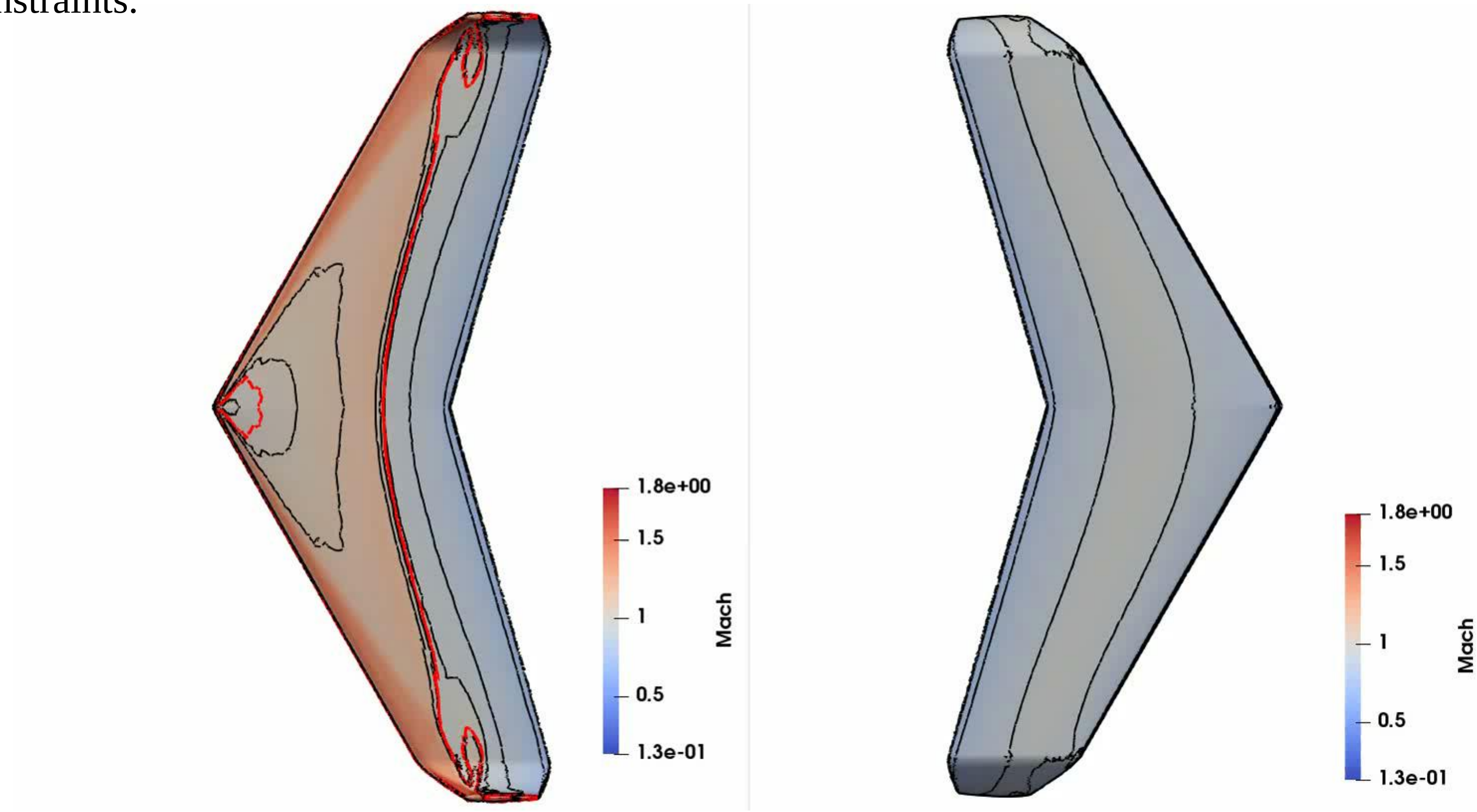


SU2 for Aerodynamic Shape Optimization

Performance of modified SU2

Synthetic testcase, based on ONERA M6 with added winglets:

- Euler-based
  - multi-point optimization (several C_L, fixed M=0.84) ;
  - t/c and C_M (pitching moment) constraints.



AGENDA

I. Need for aerodynamic shape optimization of nonplanar configurations

II. SU2 and Aerodynamic Shape Optimization

III. **SU2 inside MDAO: NERONE**

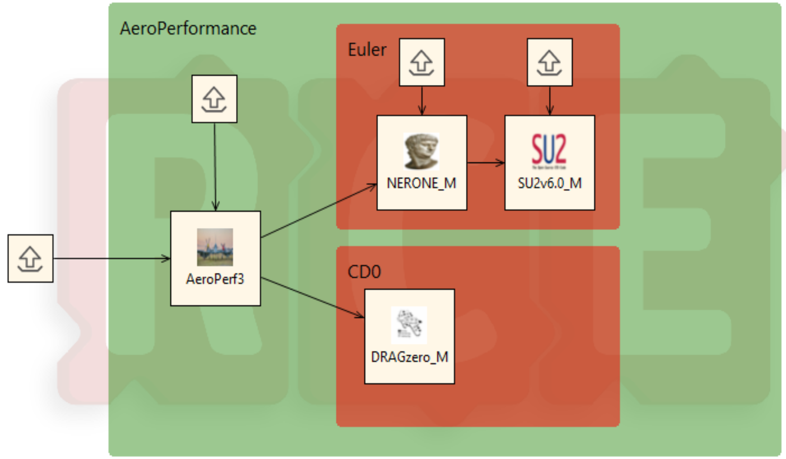
IV. Conclusions and future works

Aerodynamic Shape Optimization: NERONE

NERONE

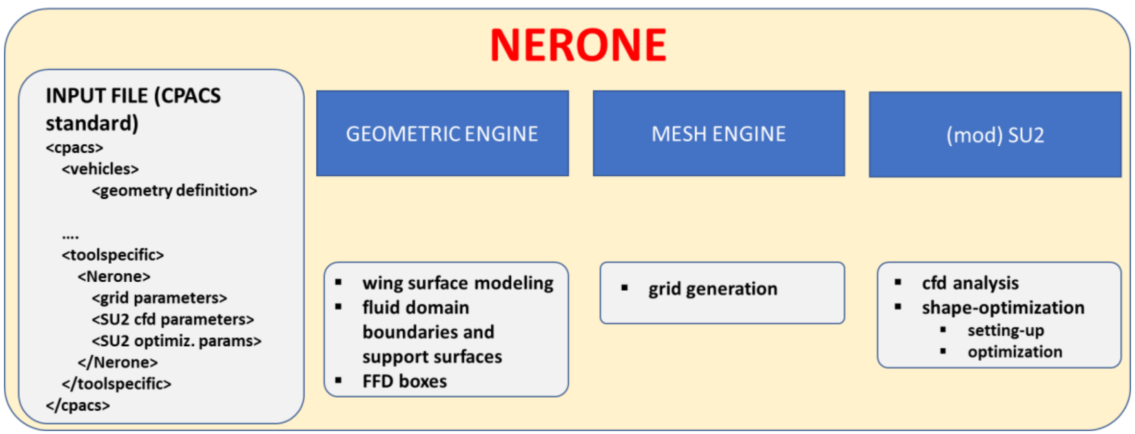
(opeNsource mEsh cReation fOr aerodyNamic Evaluation)

Tool for the aerodynamic shape optimization of **nonplanar** configurations based on open-source software.



Highlights

- Fully automatic.
- Fairing surfaces controlled in an intuitive and robust way.
- User-guided meshing strategy.
- Well-suited to be used within an MDAO workflow.
- *Features added to **SU2** code to handle nonplanar configurations.*

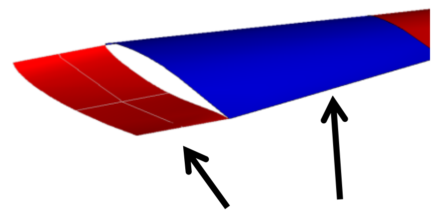


NERONE: Geometric Engine

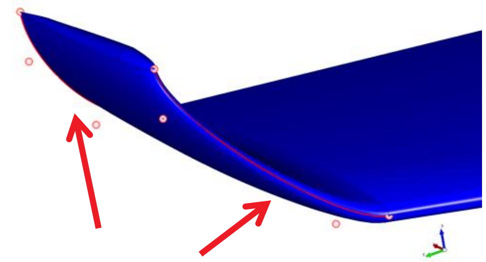
Geometric engine

Based on the **OpenCascade** library.

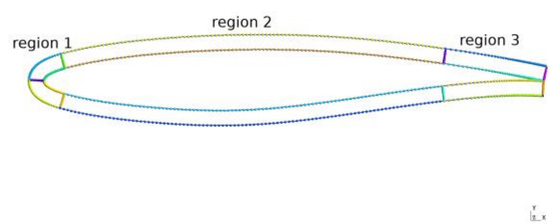
- Converts the **CPACS** (common language) parametric description into a mathematical continuous surface.
- Creates regions on the surface to guide the meshing process.
- Creates external boundaries of the domain, and intermediate surfaces to support meshing process.
- Creates, in case an optimization is carried out, the Free Form Deformation (FFD) boxes.

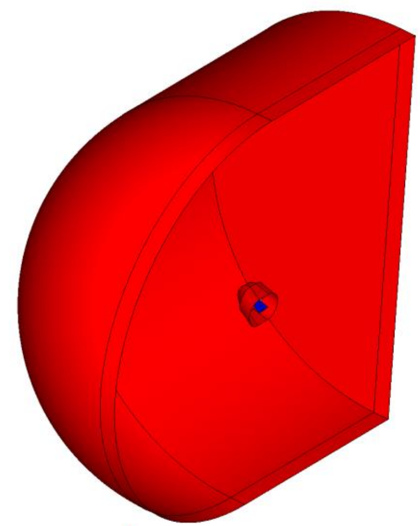


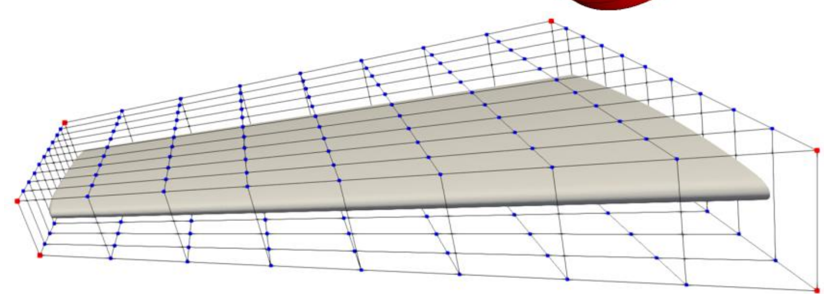
Coons' surfaces



Bezier support curves

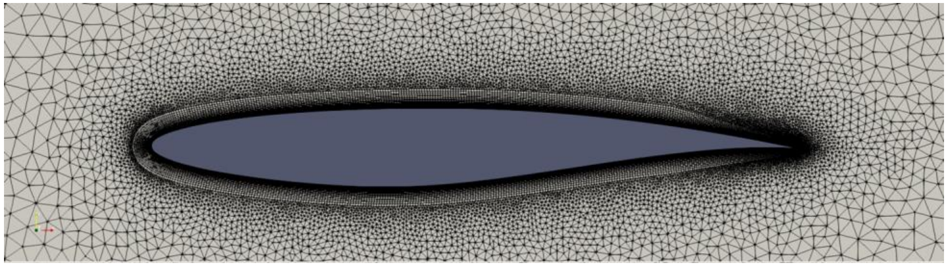






Freeform-deformation (FFD) boxes

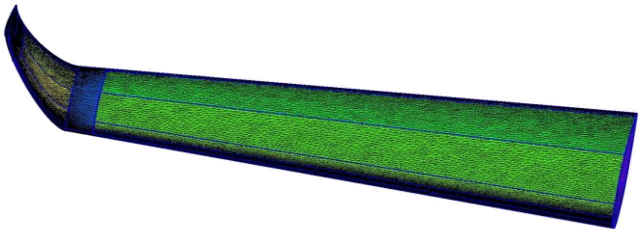
NERONE: Grid Generation module

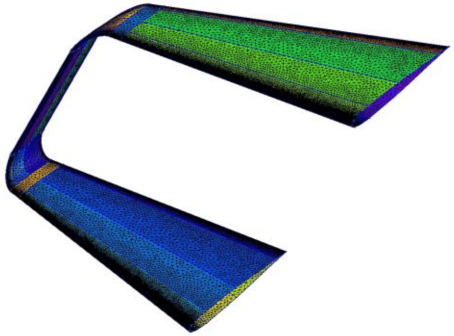


Meshing Module

Based on **GMSH**.

An interface imports the geometries, and associates parameter specified in a dedicated section of the input file (CPACS) to the appropriate regions on the wing surface and domain boundaries.



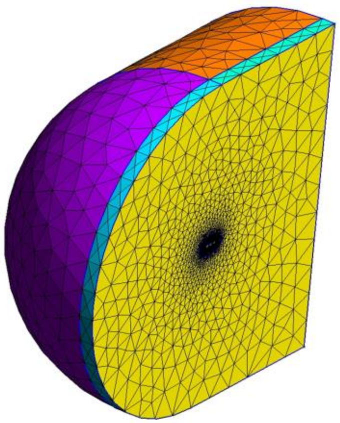


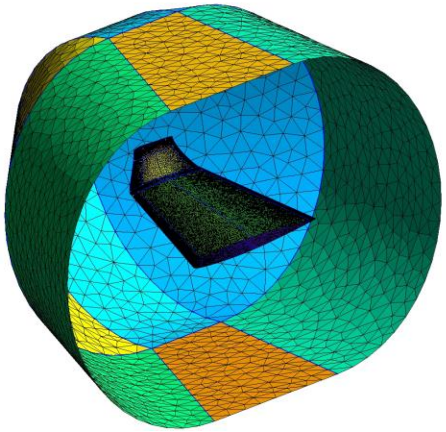
Current fully working

- 2D Euler Mesh
- 2D Rans Mesh
- 3D Euler Mesh

Under development

- 3D RANS mesh
  - Issues with GMSH algorithms if a quality mesh is sought (requirement for SU2 code, and for good results in general).





AGENDA

I. Need for aerodynamic shape optimization of nonplanar configurations

II. SU2 and Aerodynamic Shape Optimization

III. SU2 inside MDAO: NERONE

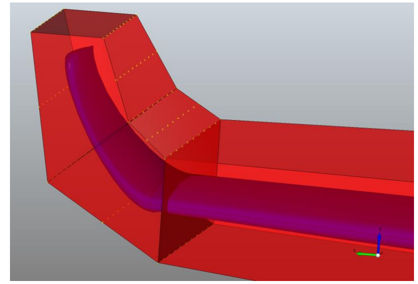
IV. Conclusions and future works

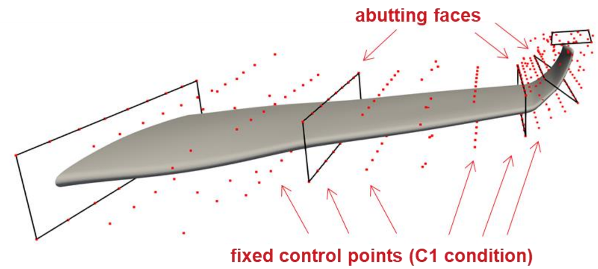
Conclusions and Future Works

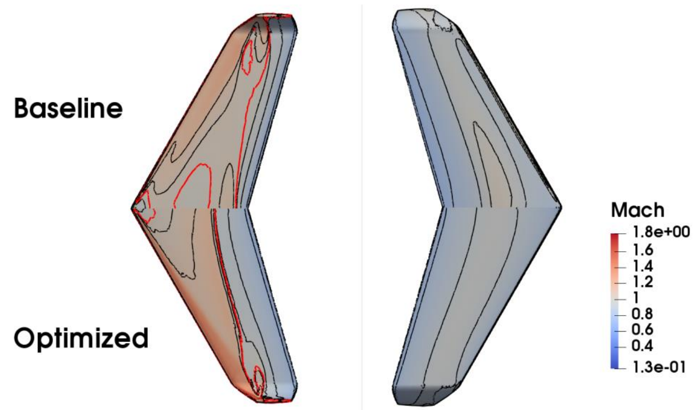
Conclusions

- Modification on SU2 code have been carried out in order to perform aerodynamic shape optimization of nonplanar wings.

- SU2 has been employed inside NERONE, for automatic shape optimization inside an MDAO.



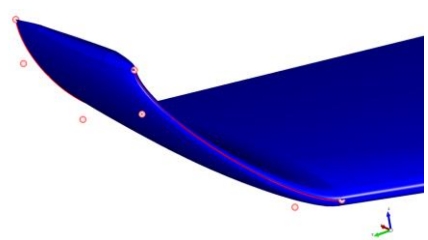


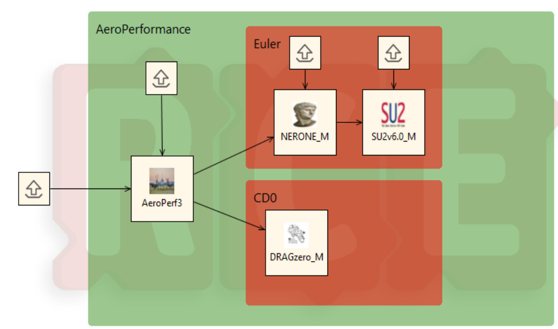


Future Works

- SU2 level:
    - Code “polishing” (comments, comply with SU2-code standard.
    - Add generality.

- NERONE level
    - 3D meshing for RANS





4th Annual SU2 Developers Meeting - May 8-10, 2019

Towards fully automated aerodynamic shape optimization of nonplanar wings with SU2

Luca Pustina[ii], **Rauno Cavallaro**[i], **Giovanni Bernardini**[ii], **Rocco Bombardieri**[i]

[i] Bioengineering and Aerospace Engineering Department, University Carlos III of Madrid
[ii] Department of Engineering, University of Roma Tre

uc3m | Universidad Carlos III de Madrid

ROMA TRE UNIVERSITÀ DEGLI STUDI