A Full Potential Static Aeroelastic Solver for Preliminary Aircraft Design

Adrien Crovato

H. Silva

A. Prado

C. Breviglieri

G. Silva

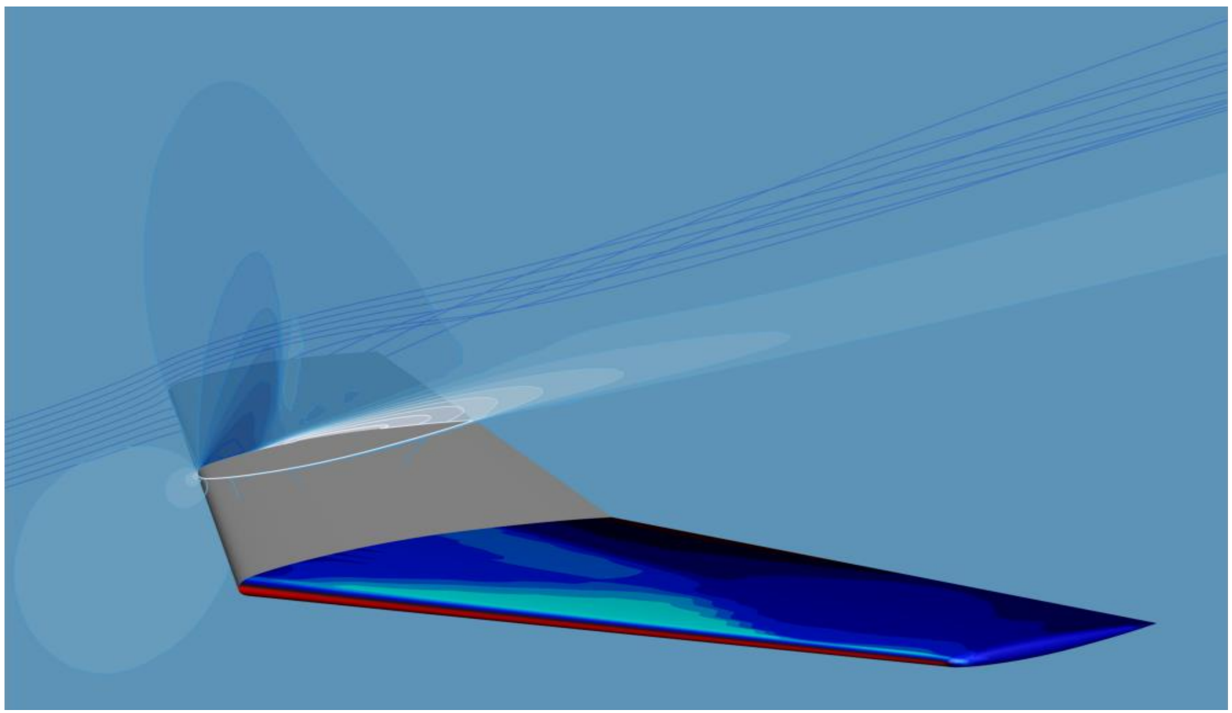
P. Cabral

H. Guner

R. Boman

V.E. Terrapon

G. Dimitriadis



EMBRAER

Savannah, June 2019

LIÈGE université

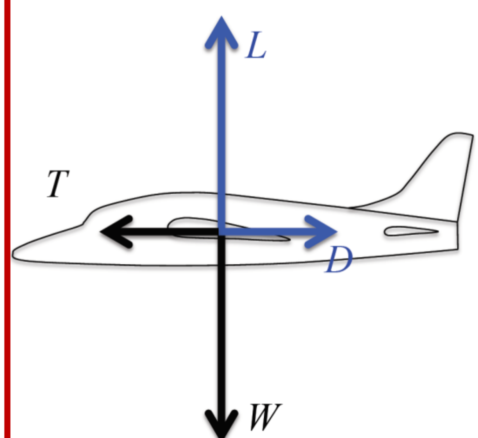
Aircraft design process

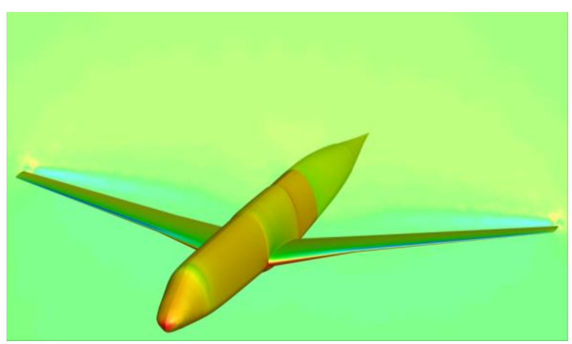
Conceptual → Preliminary → Detail

Concept (1%)
- Requirements & cost
- Aircraft configuration

Model (9%)
- Aircraft lofting
- Component optimization
- Global design

Prototype (90%)
- Manufacturing & certification
- Testing & final performance
- Flight simulators
- Local design



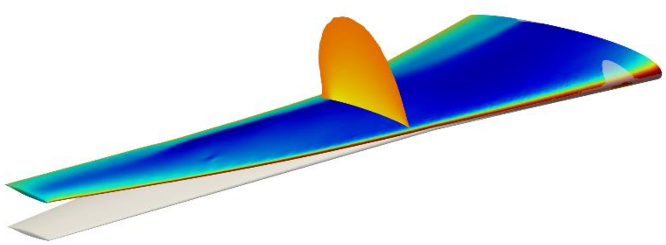


Airbus “BLADE” © T. Laurent (airliners.net)

Aeroelasticity in aircraft design

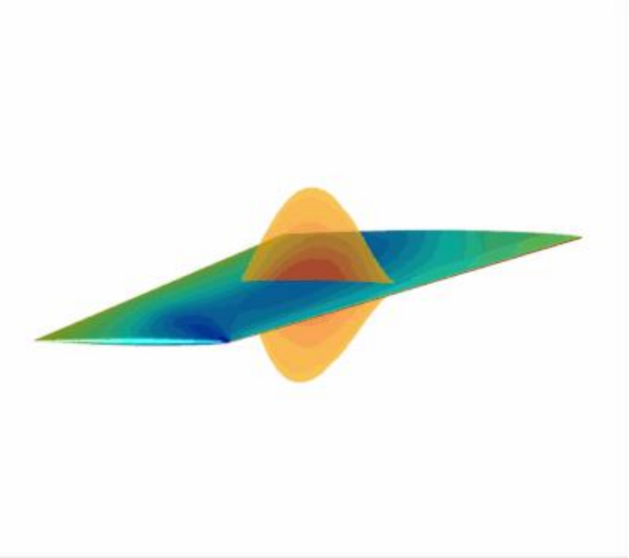
Static aeroelasticity

- Divergence speed
- Flight shape



Dynamic aeroelasiticty

- Flutter speed
- Buffeting



D. Thomas – ULiege

Enable aero-structural design and optimization

Aerodynamics for aeroelastic computations

Context

Early preliminary design

- Aerodynamic loads
- Fast linear solvers

Challenges

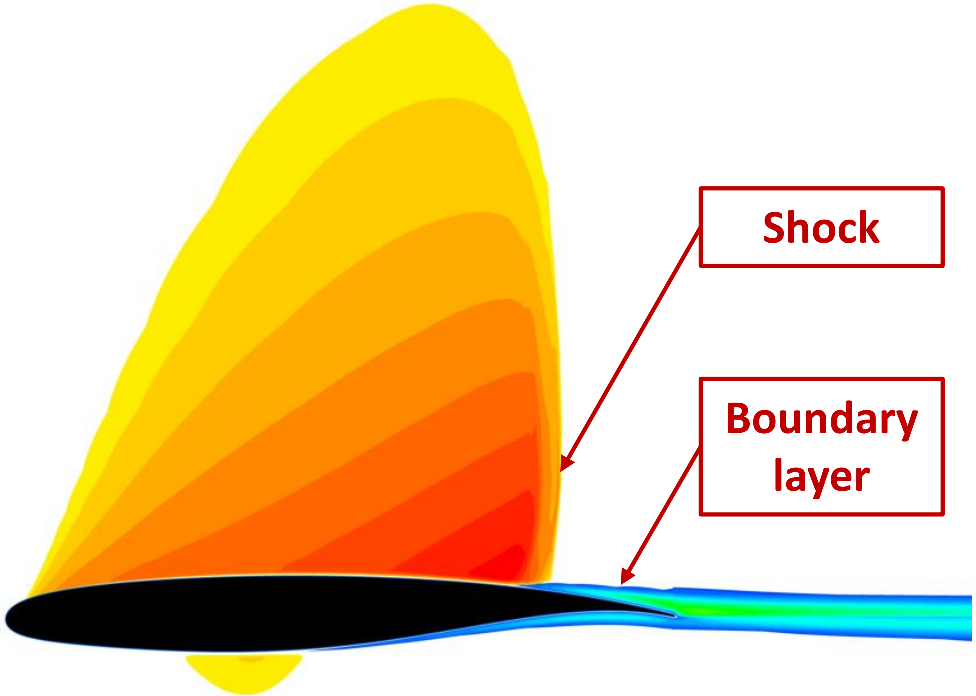
Flow nonlinearities

- Shock
- Boundary layer

Objective

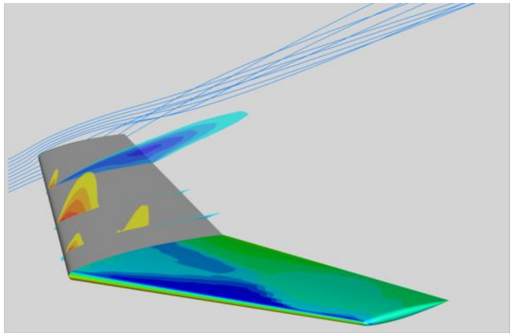
New code

- Fast
- Nonlinear
- Integrable



Research project overview

Benchmark

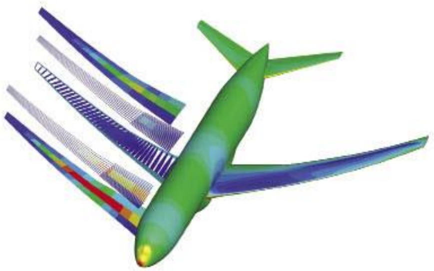


Evaluate existing models & methods that solve steady transonic flows

Development

Develop a fast aerodynamic solver for transonic loads computation based on the most efficient flow model

Integration



Implement an interface to integrate the newly developed methodology into a design framework

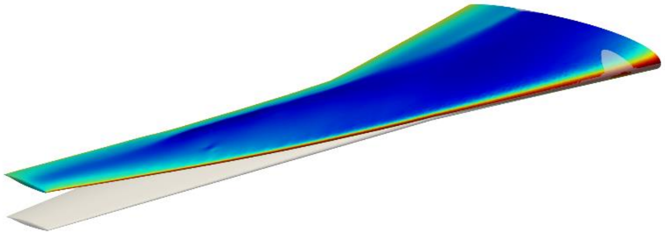
Presentation overview

Methodology

- Framework
- Flow
- CUPyDO

Results

- Solvers and benchmark
- Aerodynamic computations
- Aeroelastic computations



Framework – python wrappers

Python

```python
import flow
import gmsh

# Build mesh
msh = gmsh.meshLoader(rae.geo)

# Define problem
pbl = flow.Problem(msh)
pbl.add(flow.Neumann(…))
pbl.add(flow.Kutta(…))

# Run solver
solver = flow.Solver(pbl)
solver.run()

# …
```

SWIG

C++

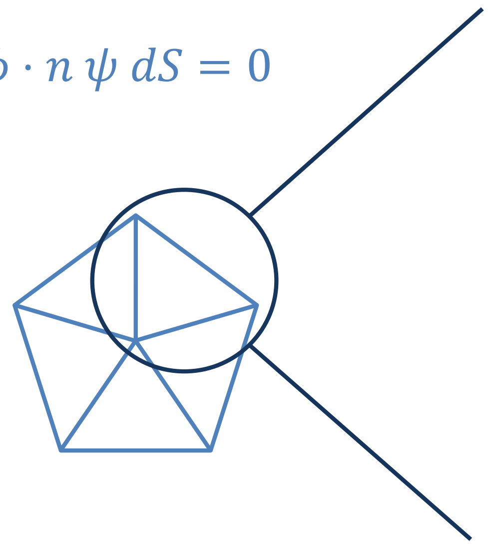
```cpp
class FLOW_API Solver : public wObject
{
public:
Solver(std::shared_ptr<Problem> _pbl);
void run();
};
```

- ✓ **CPU/memory efficient**
- ✓ **User friendly**
- ✓ **Flexible**

Flow – formulation

$$\nabla \cdot (\rho \nabla \phi) = 0$$

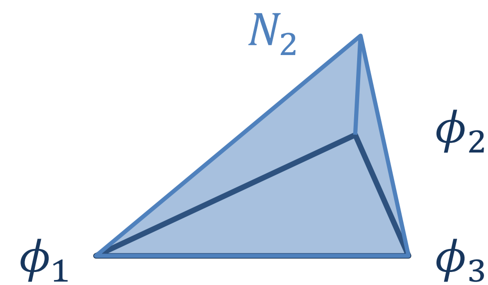
$$\iiint_\Omega \rho \nabla \phi \cdot \nabla \psi \, dV - \iint_\Gamma \rho \nabla \phi \cdot n \, \psi \, dS = 0$$

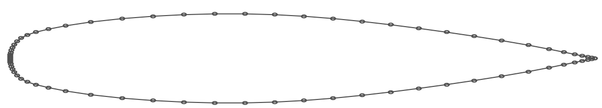


$$x = N_i\big(x(\xi,\eta), y(\xi,\eta)\big) x_i$$

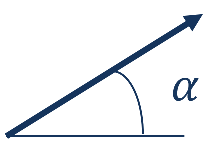
$$y = N_i\big(x(\xi,\eta), y(\xi,\eta)\big) y_i$$

$$\phi = N_i\big(x(\xi,\eta), y(\xi,\eta)\big) \phi_i$$





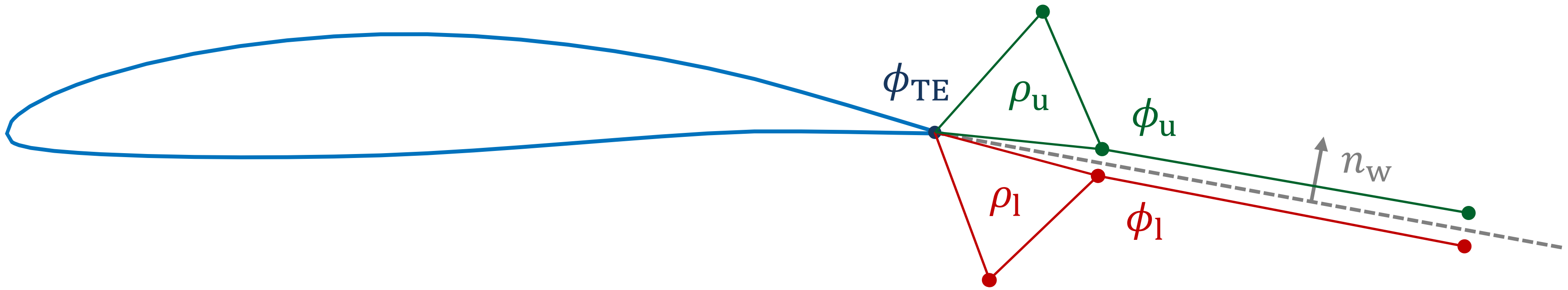
$$\nabla \phi \cdot n = 0$$



$$\nabla \phi_\infty = \{\cos\alpha, \sin\alpha\}$$

$$\nabla \phi \cdot n = \nabla \phi_\infty \cdot n$$

Flow – Kutta condition



Formulation

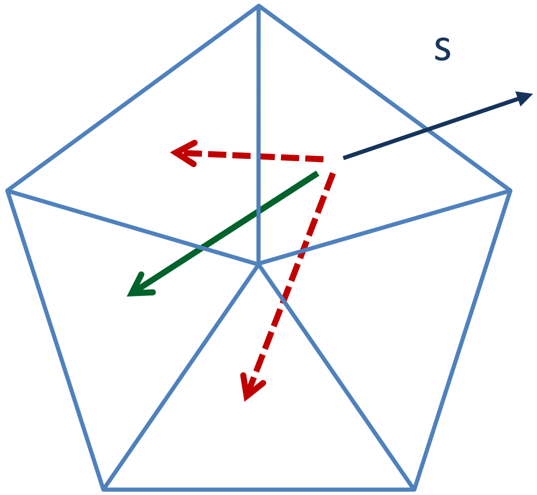
$$\rho_{\mathrm{u}} \nabla_{\mathrm{n}} \phi_{\mathrm{u}} = \rho_{\mathrm{l}} \nabla_{\mathrm{n}} \phi_{\mathrm{l}} \quad \rightarrow \iint \psi \, [[\rho \nabla \phi \cdot n]] \, dS = 0$$

$$p_{\mathrm{u}} = p_{\mathrm{l}} \quad \rightarrow \iint \left(\psi + \frac{h}{2} U_{\infty} \cdot \nabla \psi \right) [[|\nabla \phi|^2]] \, dS = 0$$

Flow – shock treatment

Density upwinding

$$\tilde{\rho} \sim \rho - \mu \frac{\partial \rho}{\partial s} \Delta s$$



Newton-Raphson procedure

$$F(\phi) = 0 \Rightarrow \frac{\partial F}{\partial \phi} \Delta \phi + F \approx 0$$

- ✓ Analytical tangent matrix
- ✓ Quadratic (3 points) line search
- ✓ Adaptive viscosity ramping

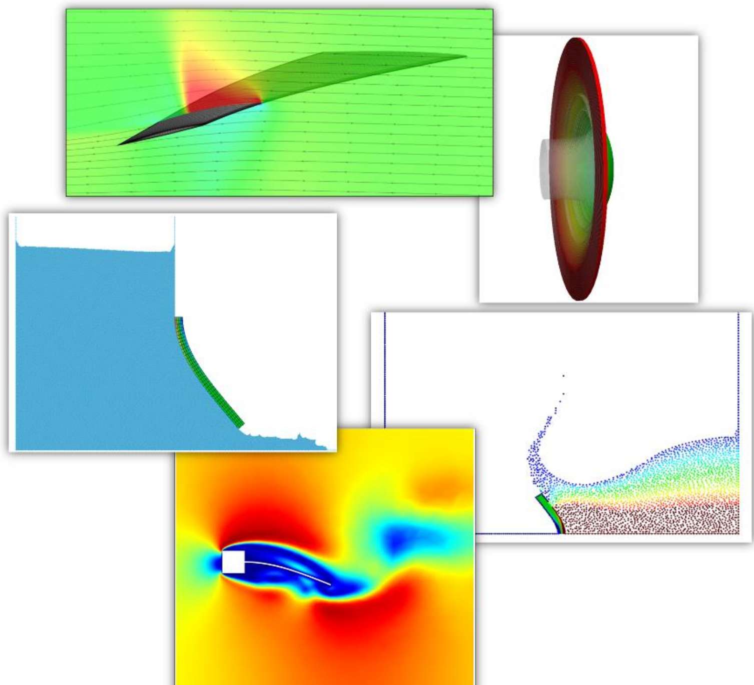
$$\mu = \boldsymbol{\mu}_{\mathbf{c}_{\downarrow}} \left(1 - \frac{\boldsymbol{M}_{\mathbf{c}_{\uparrow}}^{\mathbf{2}}}{M^2} \right)$$

CUPyDO – contributions

Romain Boman

David Thomas

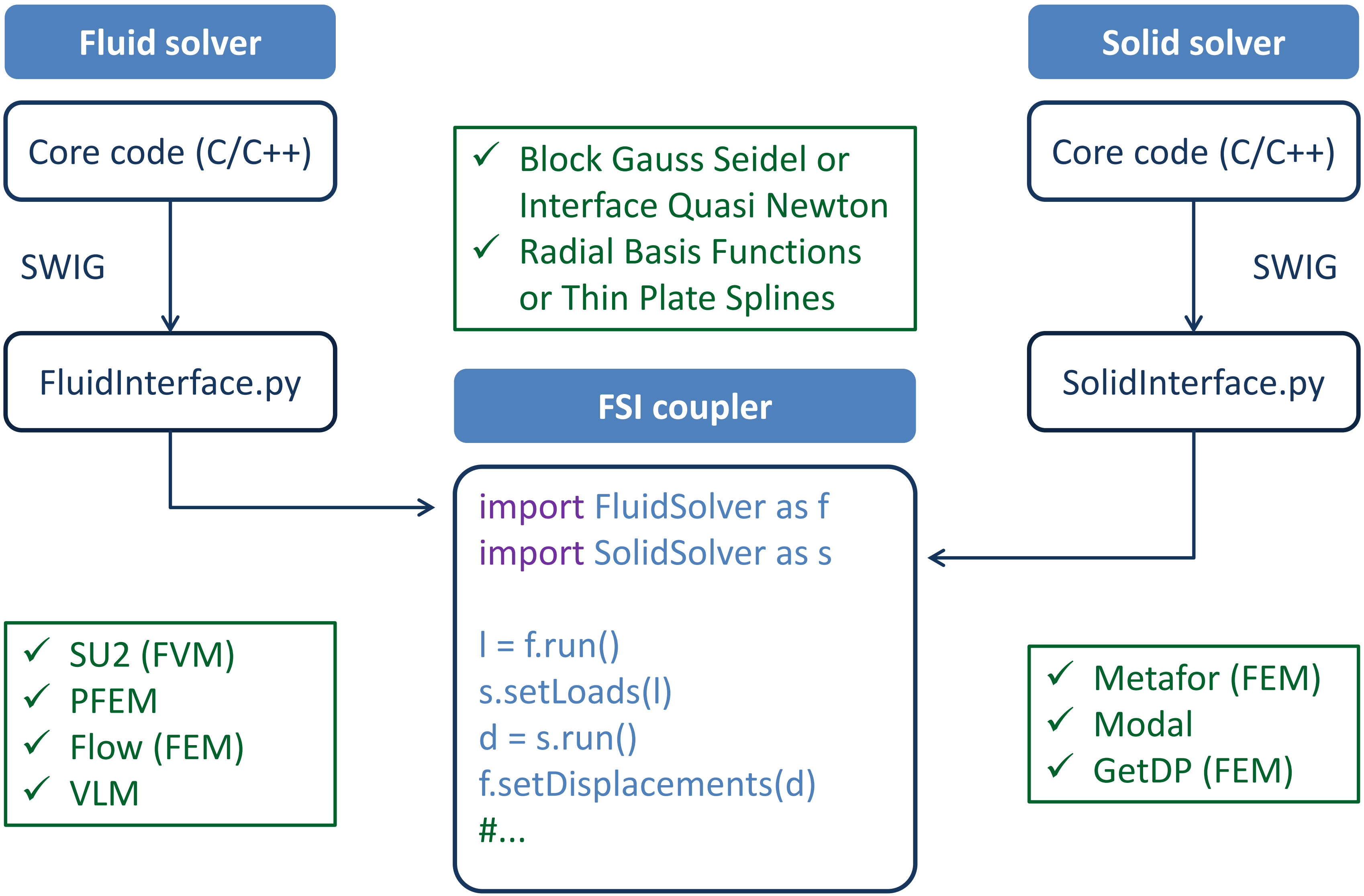
Marco L. Cerquaglia



Adrien Crovato

Mariano Sanchez M.

CUPyDO – architecture



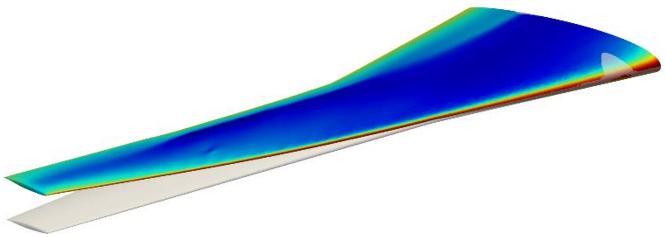
Presentation overview

Methodology

- Framework
- Flow
- CUPyDO

Results

- Solvers and benchmark
- Aerodynamic computations
- Aeroelastic computations

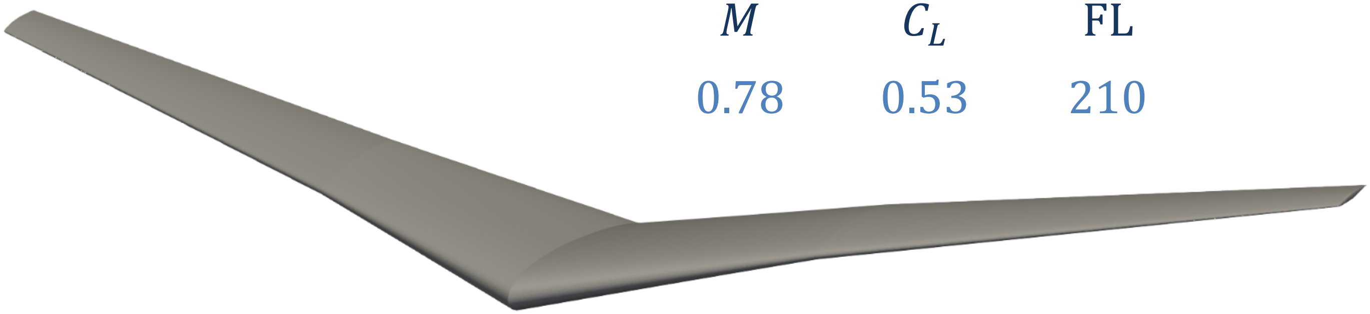


Solvers and benchmark case

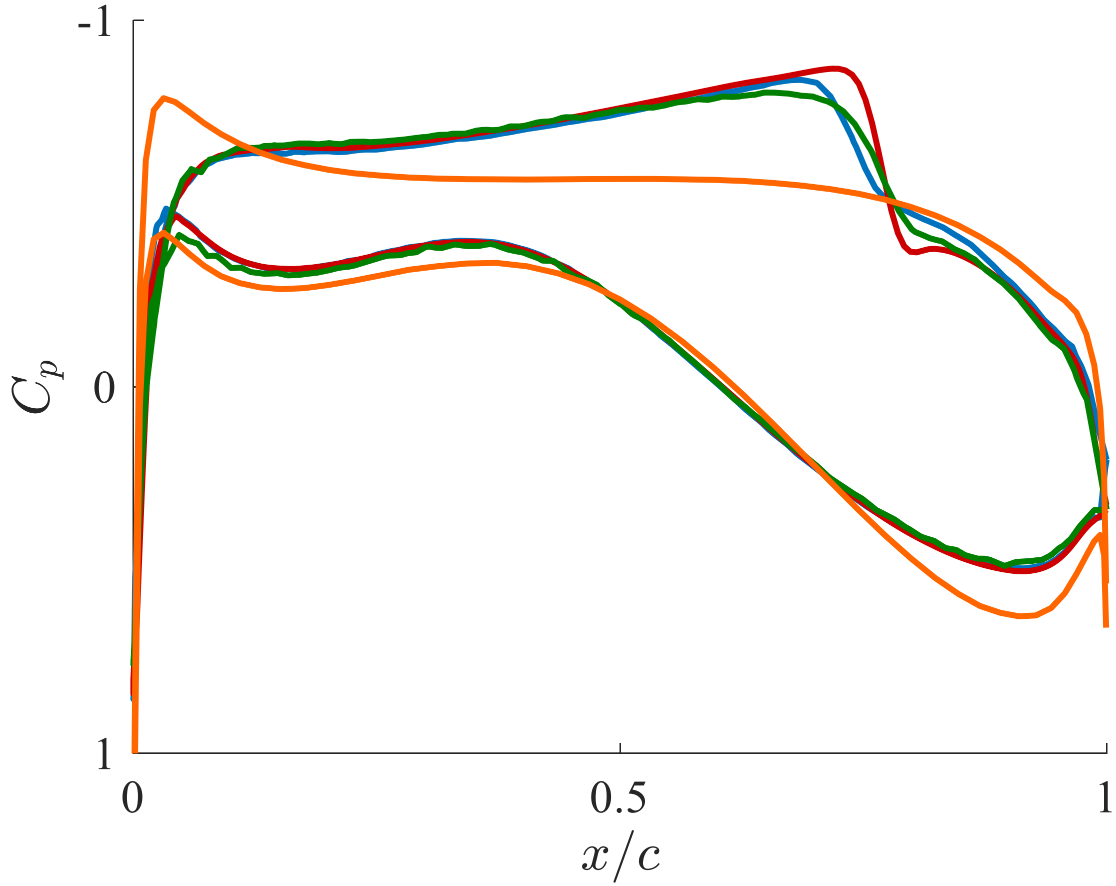
SU2	Euler	Finite Volume
Tranair	Full Potential	Finite Element
Flow	Full Potential	Finite Element
Panair/NASTRAN	Linear Potential	Boundary Element

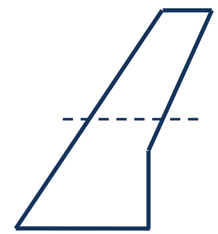
Embraer Benchmark Wing

M	C_L	FL
0.78	0.53	210



Pressure distributions





$$\frac{y}{b} = 0.406\ (\bar{\bar{c}})$$

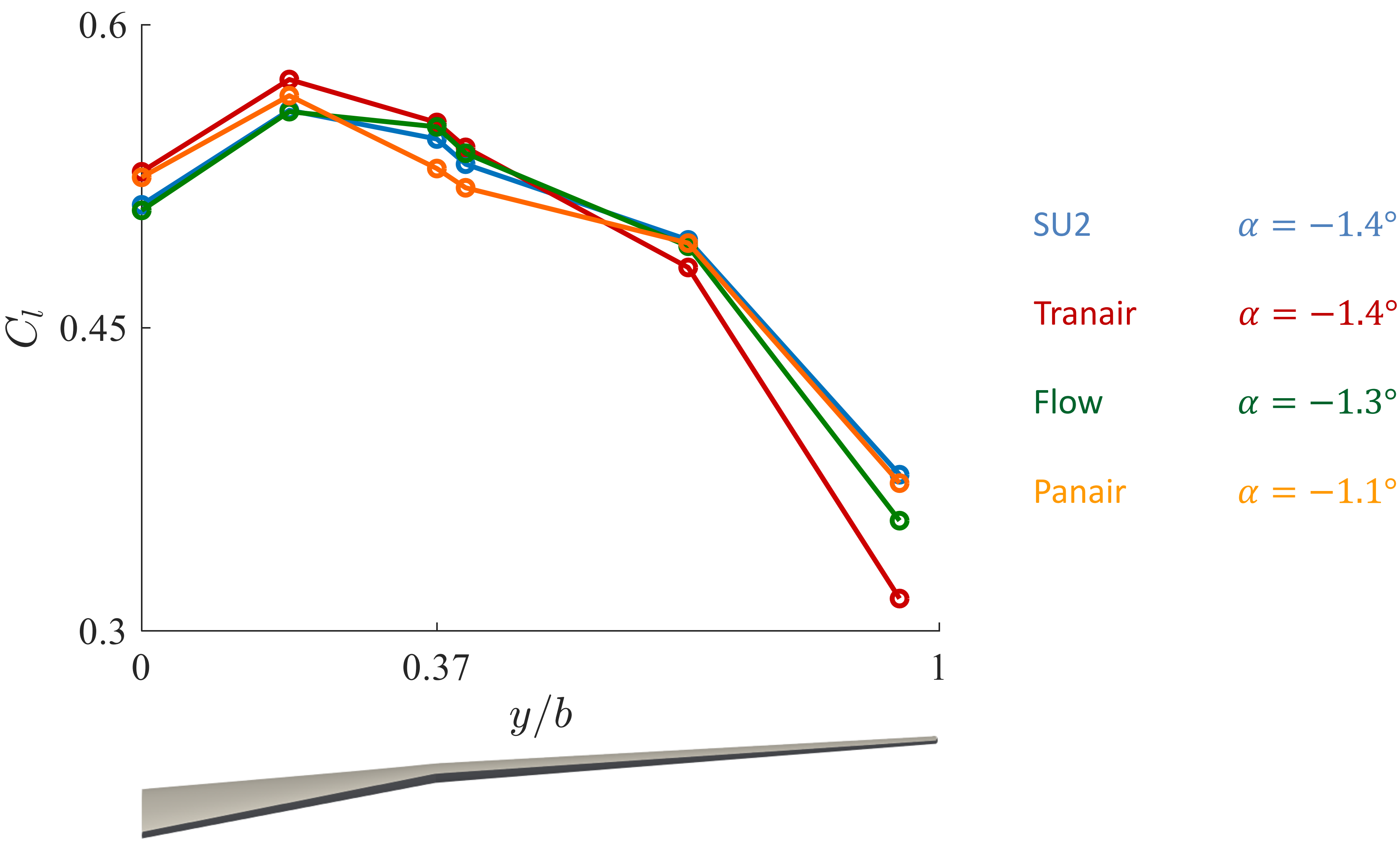
SU2 6×9000 [s]

Tranair 1×500 [s]

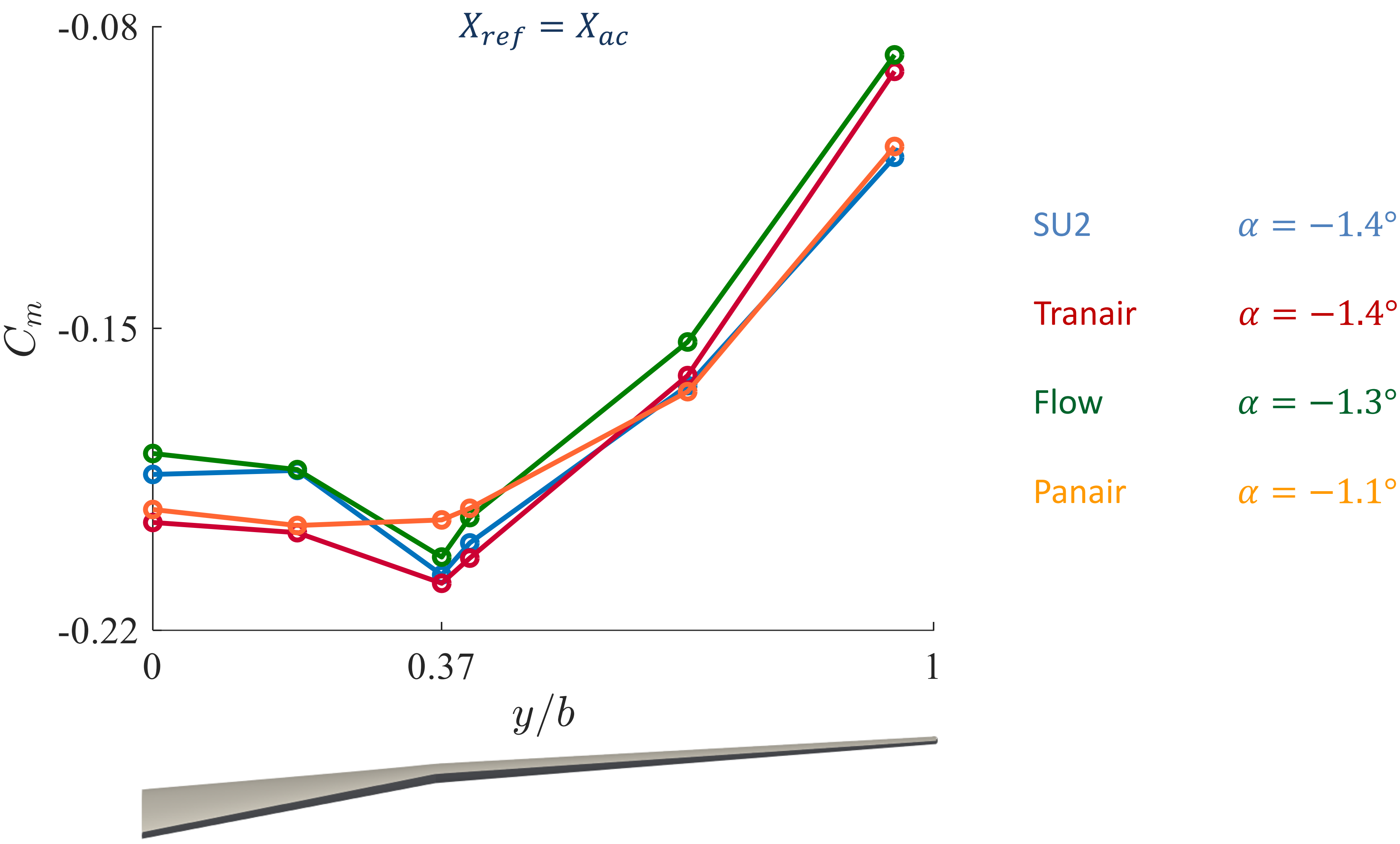
Flow 1×1500 [s]

Panair 1×10 [s]

Lift distributions



Moment distributions

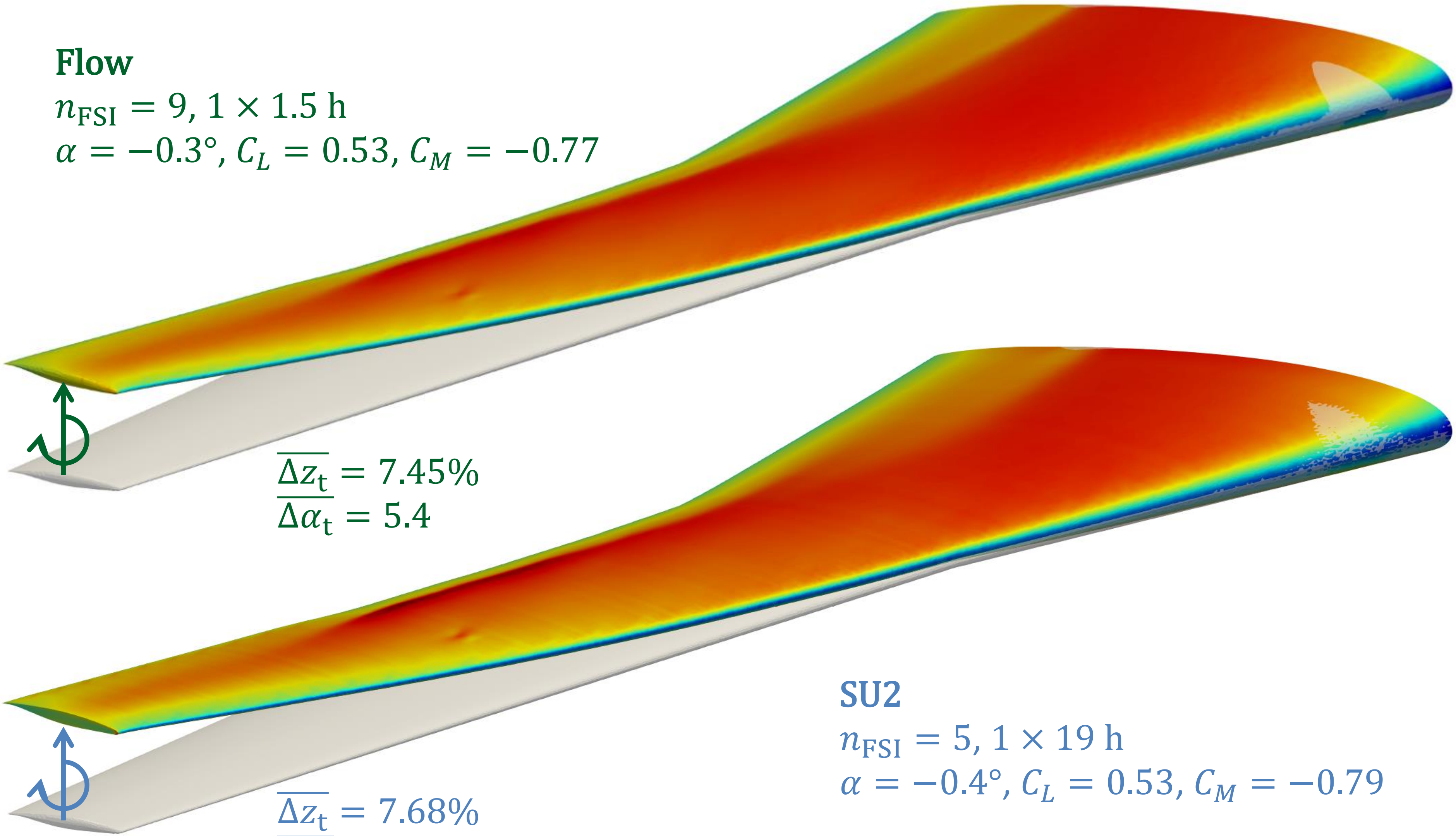


Deformed wing shape

Flow

$n_{\mathrm{FSI}} = 9,\ 1 \times 1.5\ \mathrm{h}$

$\alpha = -0.3°,\ C_L = 0.53,\ C_M = -0.77$



$\overline{\Delta z_{\mathrm{t}}} = 7.45\%$

$\overline{\Delta \alpha_{\mathrm{t}}} = 5.4$

SU2

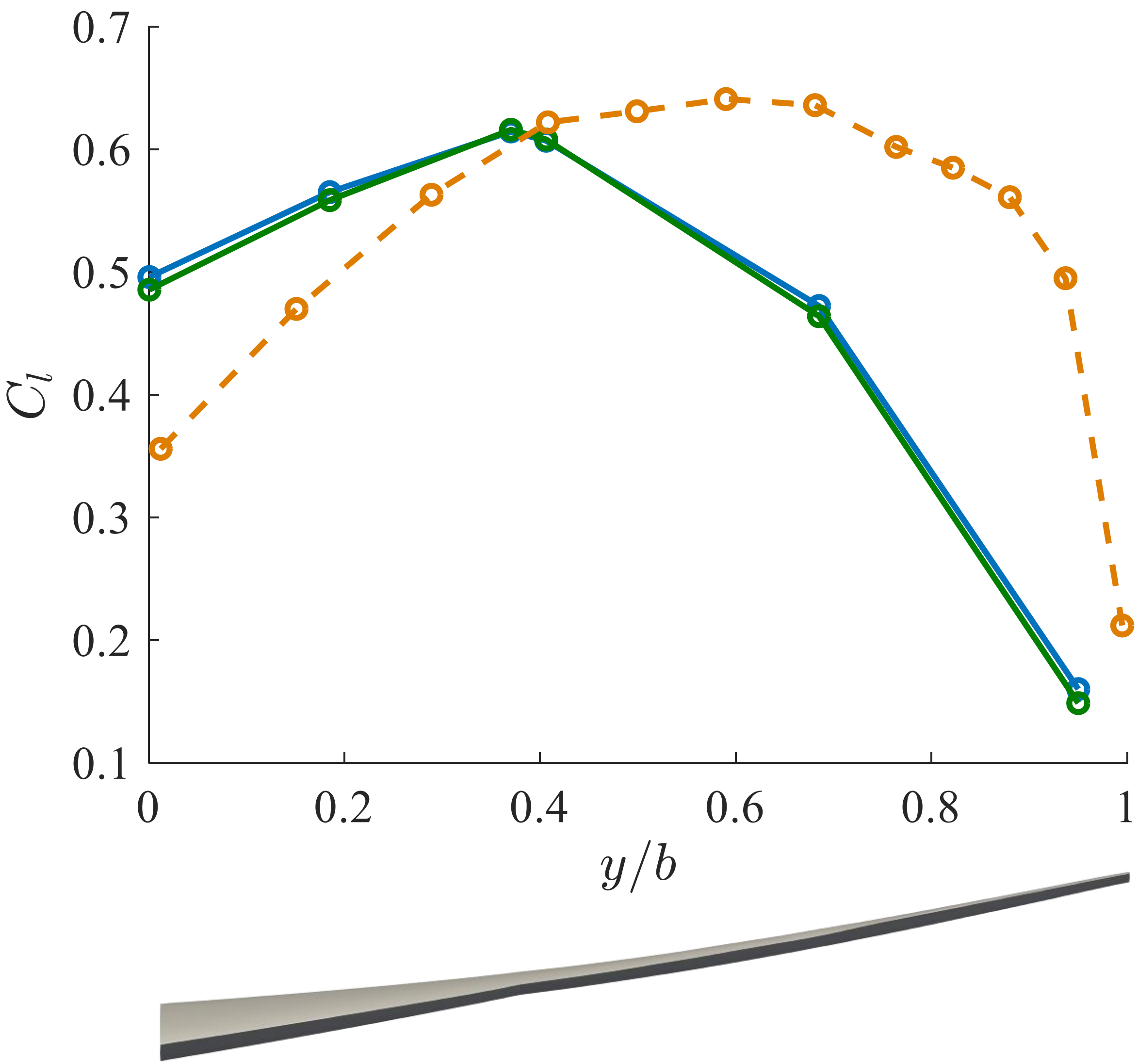
$n_{\mathrm{FSI}} = 5,\ 1 \times 19\ \mathrm{h}$

$\alpha = -0.4°,\ C_L = 0.53,\ C_M = -0.79$

$\overline{\Delta z_{\mathrm{t}}} = 7.68\%$

$\overline{\Delta \alpha_{\mathrm{t}}} = 6.0$

New lift distributions

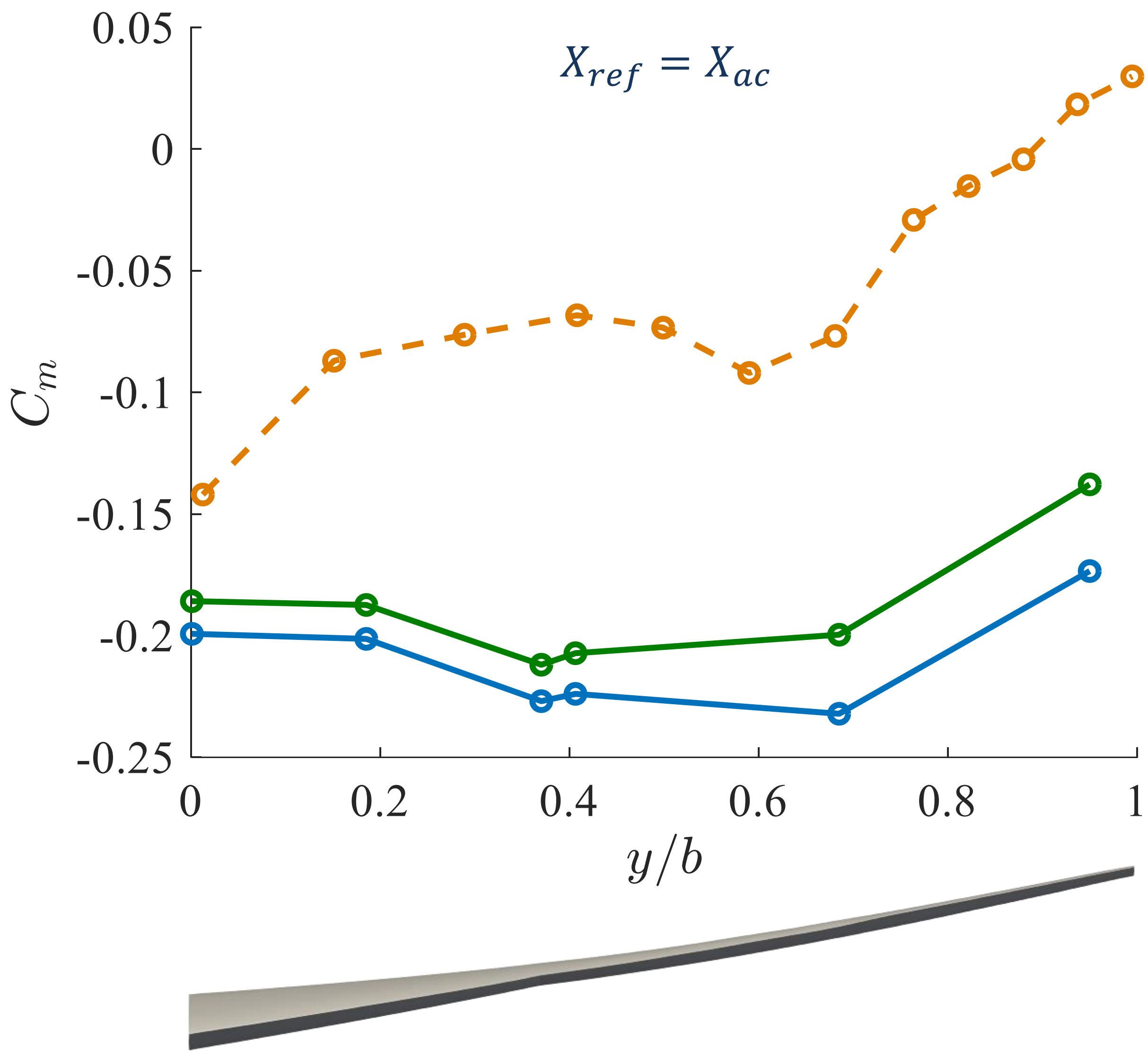


SU2 $\alpha = -0.4°$

Flow $\alpha = -0.3°$

NASTRAN $\alpha = 5.6°$

New moment distributions



SU2 $\alpha = -0.4°$

Flow $\alpha = -0.3°$

NASTRAN $\alpha = 5.6°$

Conclusion and perspectives

Summary

- Development of **Flow** and **CUPyDO**
- **Full Potential** equation offers a **good tradeoff** between accuracy and cost compared to Euler or Linear Potential equations

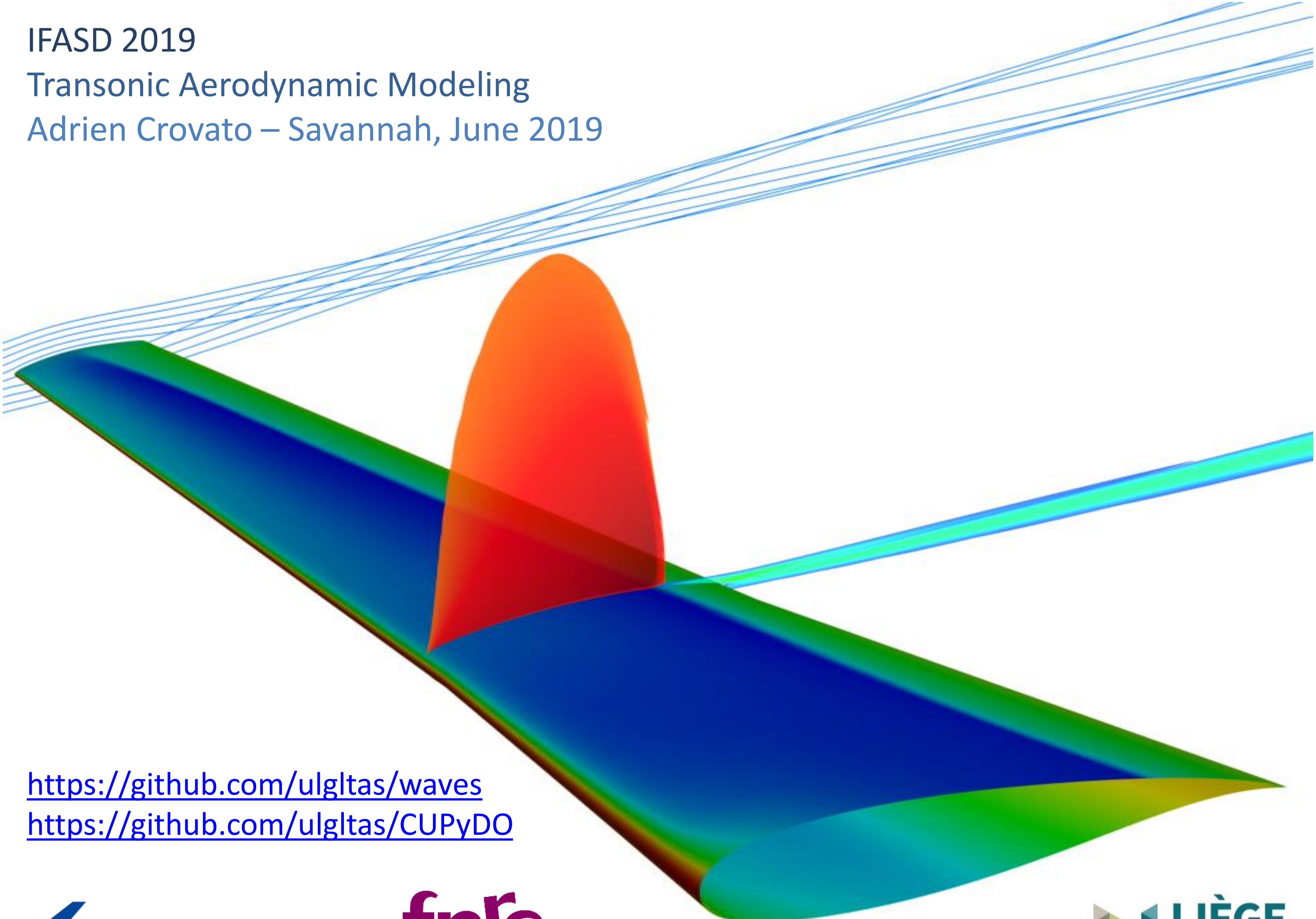
Next steps

- Optimize Flow (**Quasi Newton** and **line search methods**, other **inner solvers**, **Intel** compilers, …)
- Enhance Flow (**adaptive gridding, unsteady** and **viscous coupling** capabilities)
- Investigate **camber** and **transonic correction** methods for NASTRAN
- Investigate **multi-fidelity FSI** computations

IFASD 2019

Transonic Aerodynamic Modeling

Adrien Crovato – Savannah, June 2019



https://github.com/ulgltas/waves
https://github.com/ulgltas/CUPyDO

EMBRAER

fnrs
LA LIBERTÉ DE CHERCHER

LIÈGE université