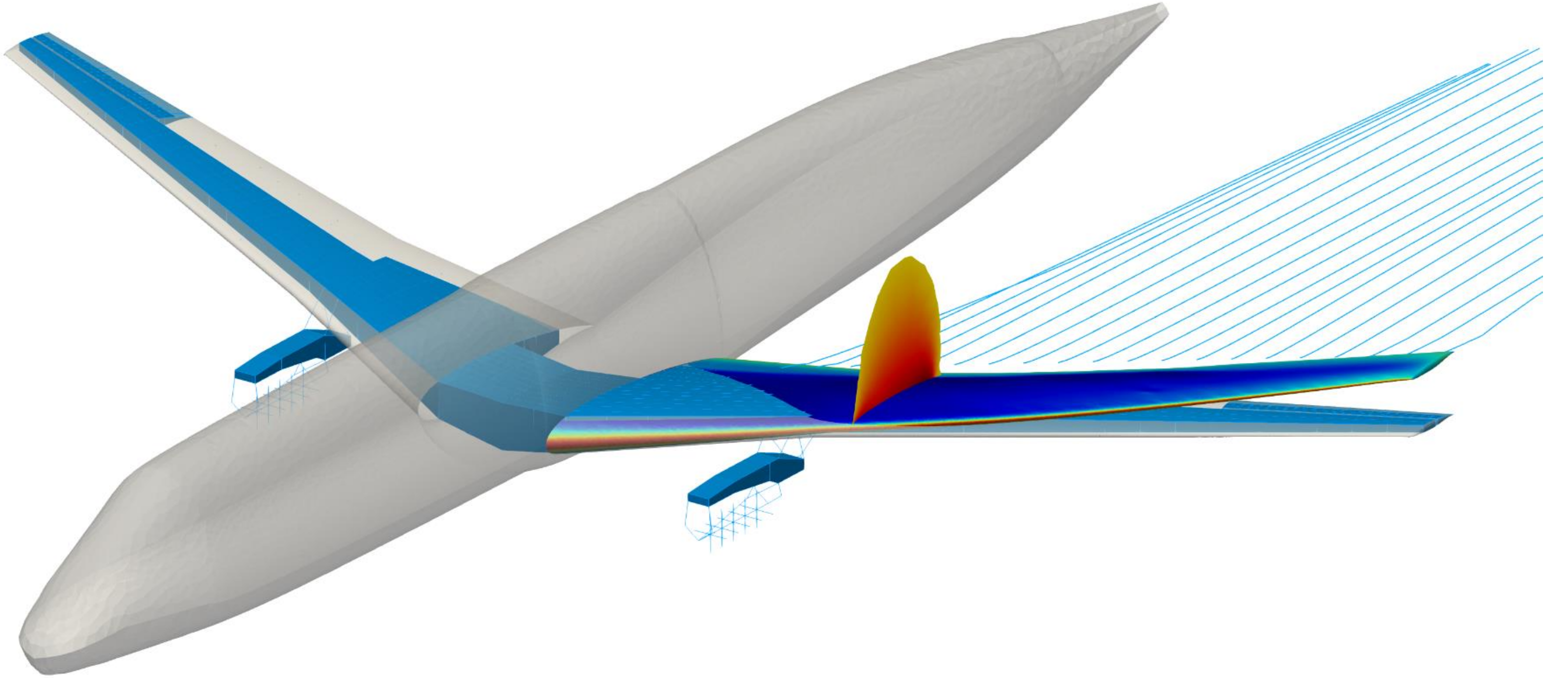
Aerostructural modeling for preliminary aircraft design

Adrien Crovato



Paris, November 2023

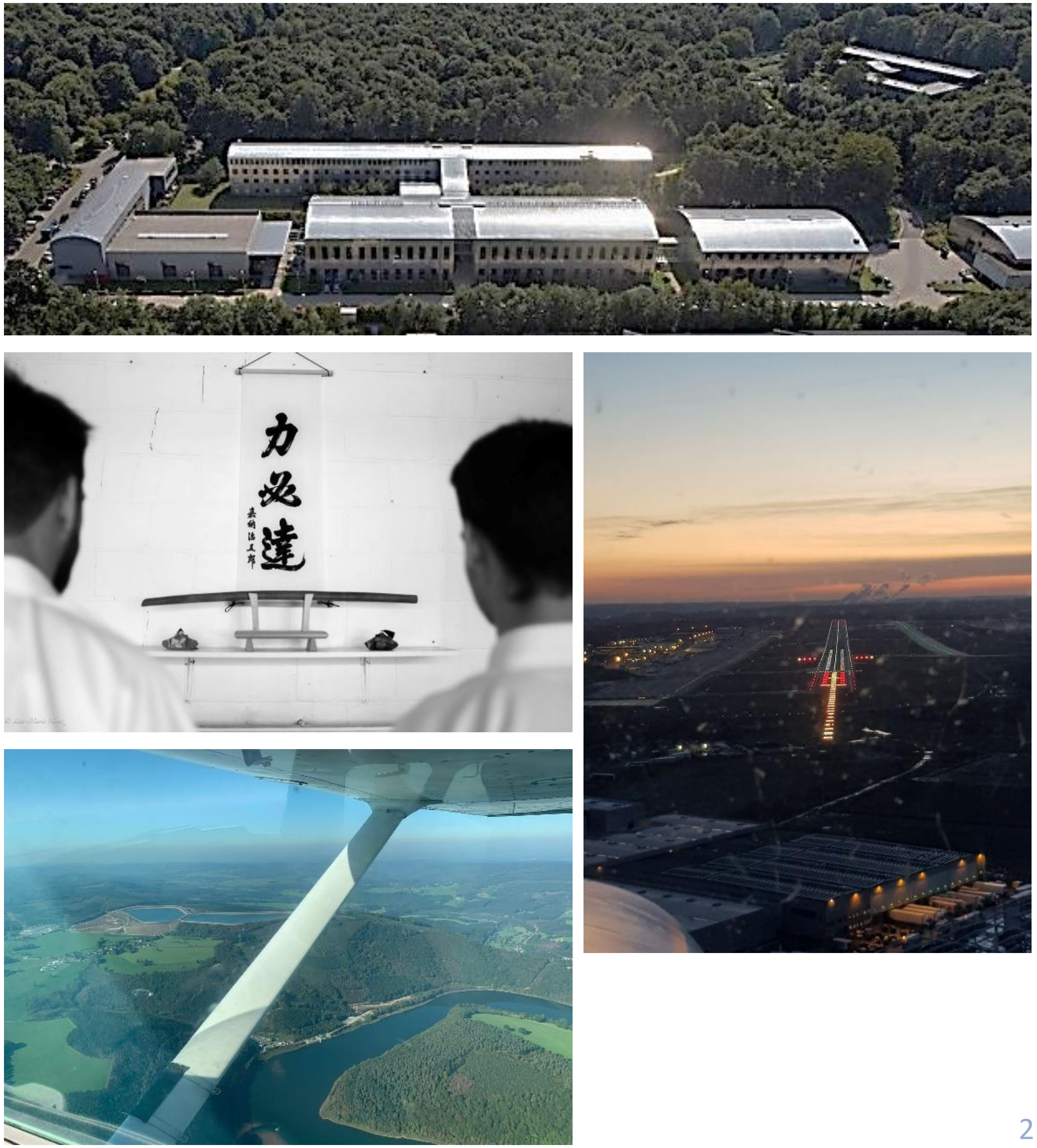
About me

Professional

- MSc in Aerospace Eng., 2015
- PhD in Aerospace Eng., 2020
- Post-doc in Aerospace Eng., since 2020
- Teaching activities, since 2015
- ULiège with Embraer

Personal

- Martial artist
- Private pilot



Reducing fuel burn

More traffic

Less emissions

More profit

Fuel consumption must be reduced!

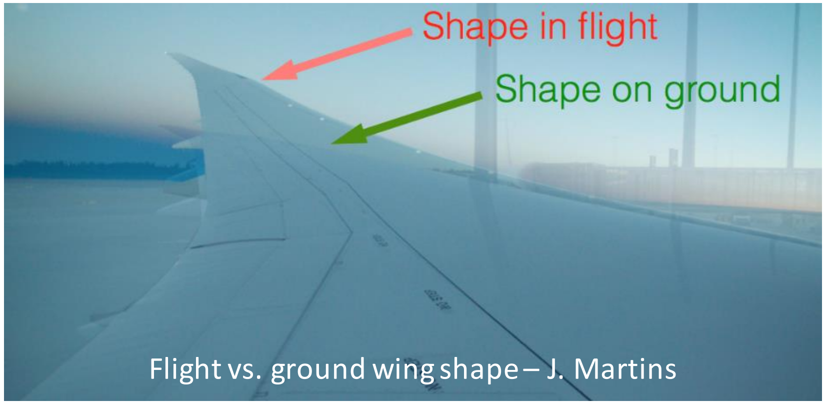
Aerodynamics

Structure/materials

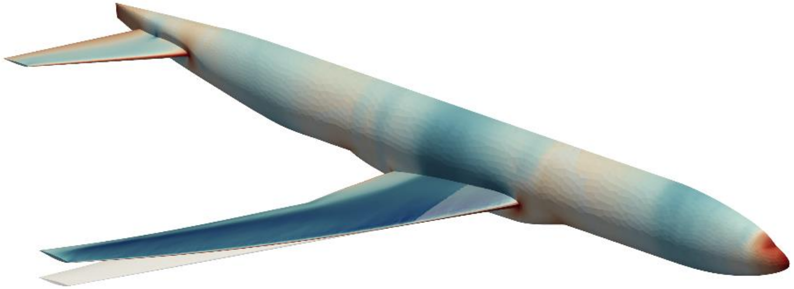
Aeroelasticity in aircraft design

Static aeroelasticity

- Divergence
- Wing shapes



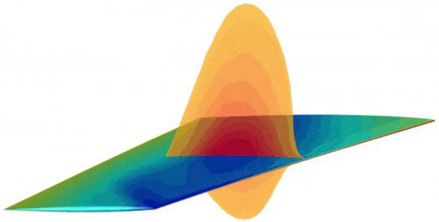
Flight vs. ground wing shape – J. Martins



Dynamic aeroelasticity

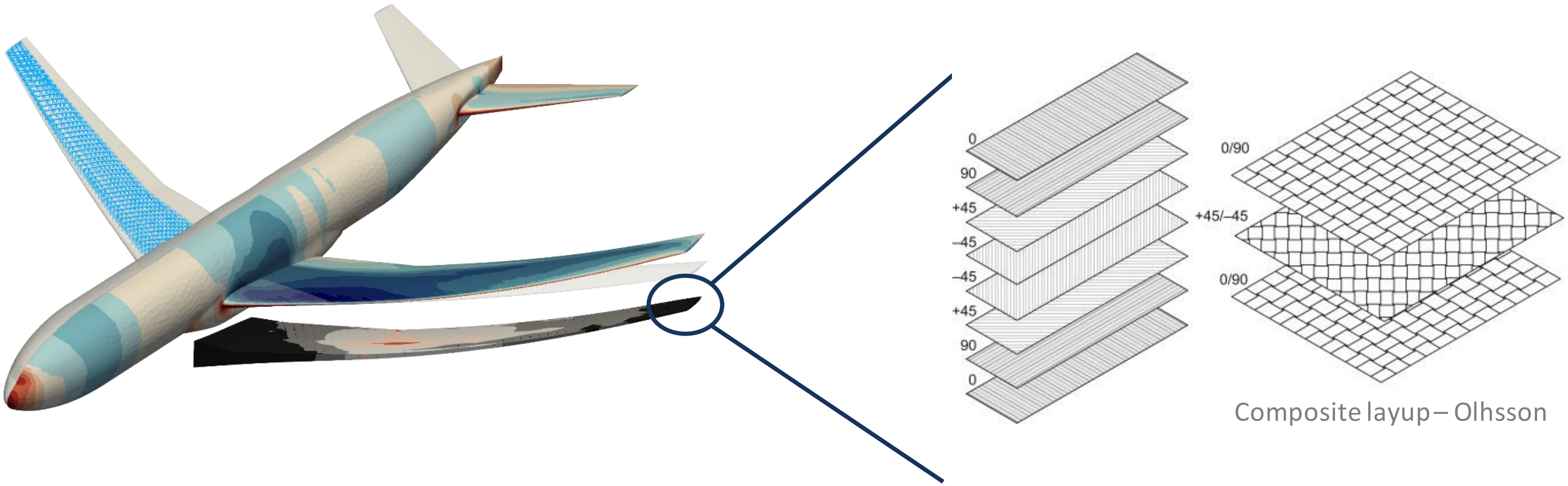
- Flutter
- Buffeting, LCO, etc.

Tail flutter on PA-38 – NASA



Flutter on AGARD wing – D. Thomas

Aerostructural optimization



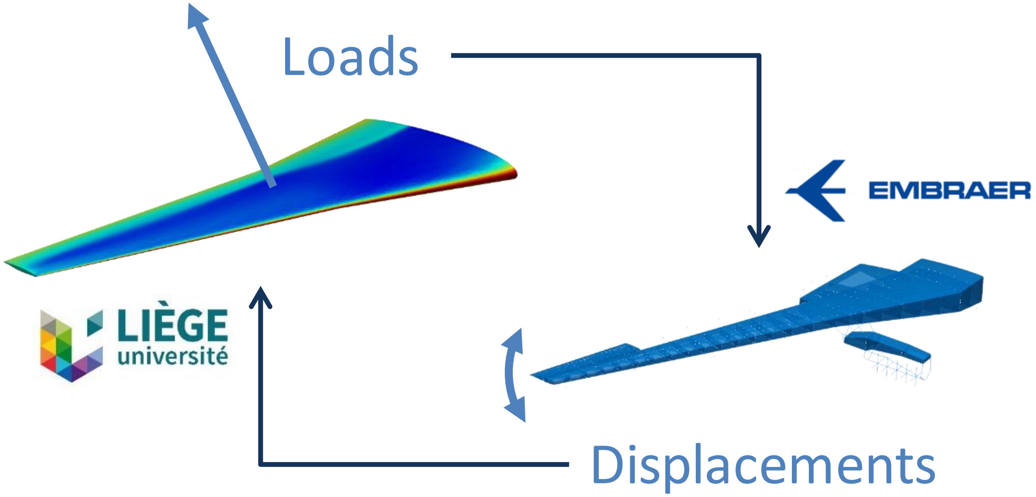
Composite layup – Olhsson

Optimize shape and laminates

- Decrease fuel burn

Such that

- No failure
- No flutter



Loads

Displacements

Aircraft design process

Conceptual

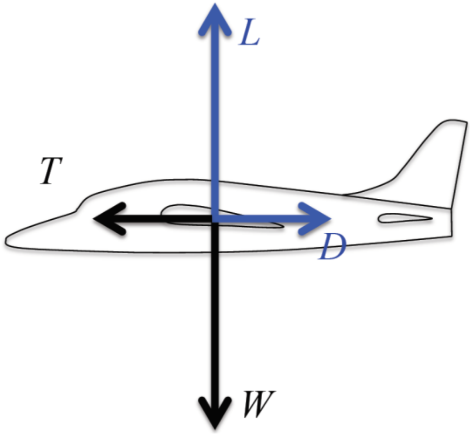
Concept (1%)
- Configuration
- Mission & cost

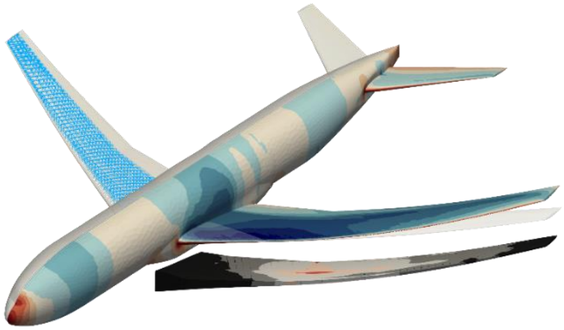
Preliminary

Model (9%)
- Global design
- Optimization
- Performance

Detail

Prototype (90%)
- Local design
- Manufacturing
- Testing
- Certification






ERJ-190-300-STD © C. Hines (airliners.net)

Appropriate models must be chosen

Outline

Modeling

- Optimization formulation
- Aerodynamic models
- Numerical methods

Static aeroelasticity

- Framework
- DART code
- Applications

Dynamic aeroelasticity

- Framework
- SDPM and NIPK codes
- Applications

Optimization formulation

Gradient-based approach

$$d_x F(u;\ x) \to 0$$

s.t. $$\begin{aligned} R(u;x) &= 0 \\ C(u;x) &= 0 \end{aligned}$$

$$\begin{aligned} d_x F(u;x) &\to 0 \\ R(u;x) &= 0 \end{aligned}$$

“perturbation”

$$\begin{cases} R\big(u(x+\delta x)\big) = 0 \\ d_x F = \Delta\left\{\dfrac{F\big(u(x+\delta x)\big)}{\delta x}\right\} \end{cases}$$

“chain rule”

$$\begin{cases} R\big(u(x)\big) = 0 \\ d_x F = \partial_x F - \partial_u F \partial_u R^{-1} \partial_x R \end{cases}$$

Methods based on perturbation

Finite differences

$$\begin{cases} R\big(u(x)\big) = 0 \\ R\big(u^{+}(x+\delta x)\big) = 0 \\ d_x F = \dfrac{F(u^{+}) - F(u)}{\delta x} + O(\delta x) \end{cases}$$

Complex step

$$\begin{cases} R\big(u(x)\big) = 0 \\ R\big(u^{+}(x+i\delta x)\big) = 0 \\ d_x F = \mathrm{Im}\left\{\dfrac{F(u^{+})}{\delta x}\right\} + O(\delta x^{2}) \end{cases}$$

Cost

Solve equations: $n_x \times n_s \times t_s$

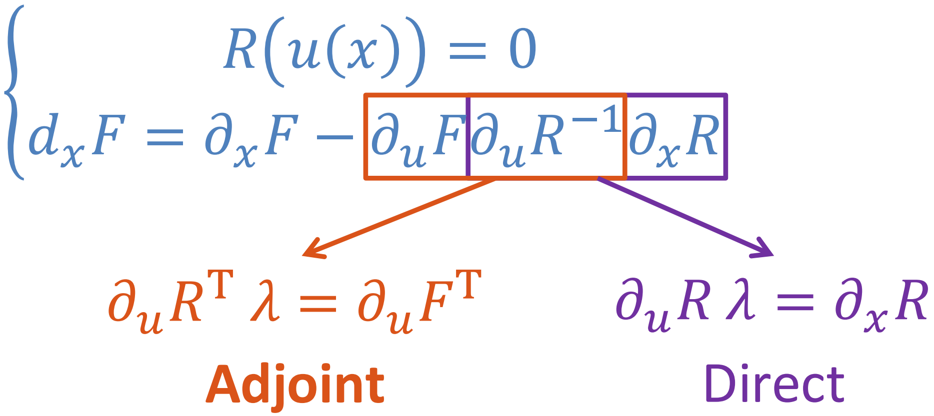
Evaluate gradients: $n_x \times n_f \times t_f$

Total: $n_x \times \left(n_s \times t_s + n_f \times t_f\right)$

n_x: n.o. design variables
n_s: n.o. nonlinear iterations
n_f: n.o. functionals
t_s: time to solve linear equations
t_f: time to compute functional

Methods based on chain rule

Direct and adjoint



$$\begin{cases} R(u(x)) = 0 \\ d_x F = \partial_x F - \partial_u F \, \partial_u R^{-1} \, \partial_x R \end{cases}$$

$$\partial_u R^{\mathrm{T}} \lambda = \partial_u F^{\mathrm{T}}$$

Adjoint

$$\partial_u R \, \lambda = \partial_x R$$

Direct

Cost (adjoint)

Solve adjoint: $n_f \times t_s$

Evaluate gradients: $(n_u + n_x) \times (n_f \times t_f + t_r)$

Total: $\left((n_u + n_x) \times (n_f \times t_f + t_r) + n_f \times t_s\right)$

n_x: n.o. design variables
n_u: n.o. variables
n_s: n.o. nonlinear iterations
n_f: n.o. functionals
t_s: time to solve linear equations
t_f: time to compute functional
t_f: time to compute residuals

▶ **Nearly independent on number of design variables**

Computation of the gradients

Hand differentiation

✓ Most effective

× Difficult, sometimes not feasible

Finite differences

✓ Very easy

× Inaccurate

Complex step

✓ Accurate

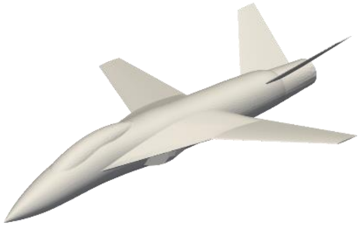
× Complex arithmetic

Automatic differentiation

✓ Straightforward

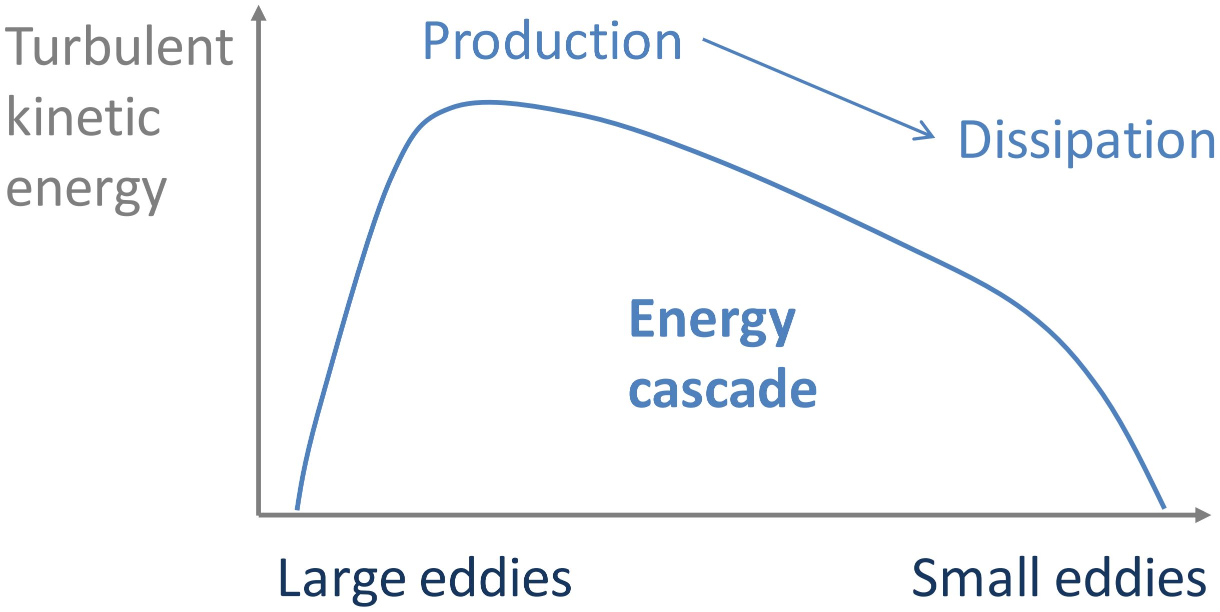
× Increased memory usage

High-fidelity aerodynamic modeling



$Re \sim 10^7$

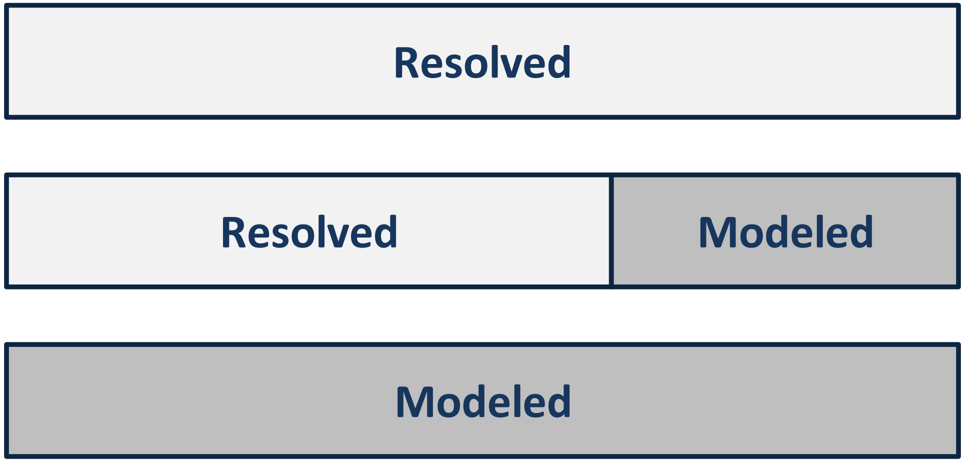
Flow is **turbulent**



Direct **N**umerical **S**imulation

Large **E**ddy **S**imulation

Reynolds-**A**veraged **N**avier-**S**tokes



Aerodynamic models for aircraft design

RANS equations
- Subsonic
- Supersonic
- Transonic
- Viscous

Euler equations
- Subsonic
- Supersonic
- Transonic
- **Inviscid**

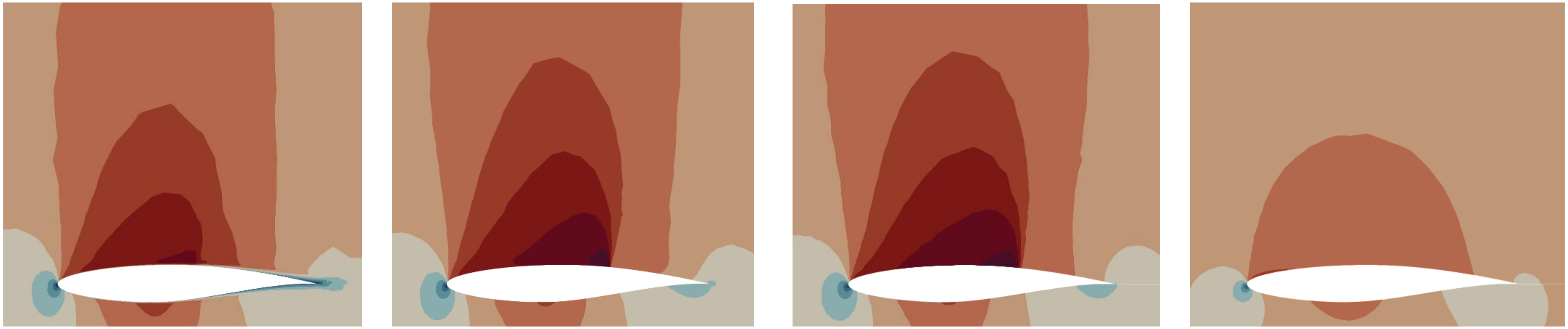
Full potential equation
- Subsonic
- Supersonic
- **∼Transonic**
- Inviscid

Linear potential equation
- **∼Subsonic**
- **∼Supersonic**
- ~~**Transonic**~~
- Inviscid

→ Inviscid

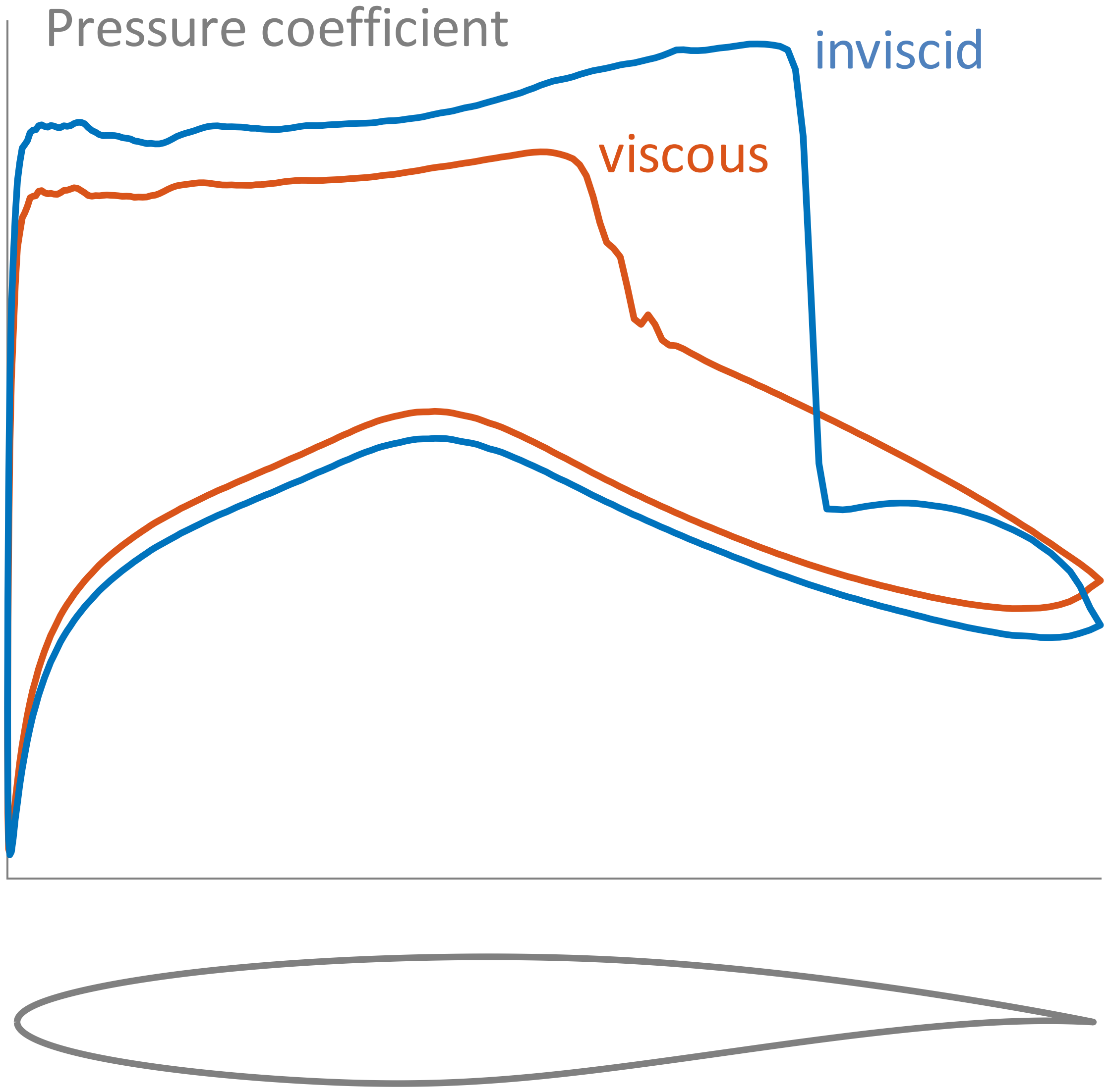
→ Isentropic

→ Linear



Mach number

Shock and boundary layer interaction

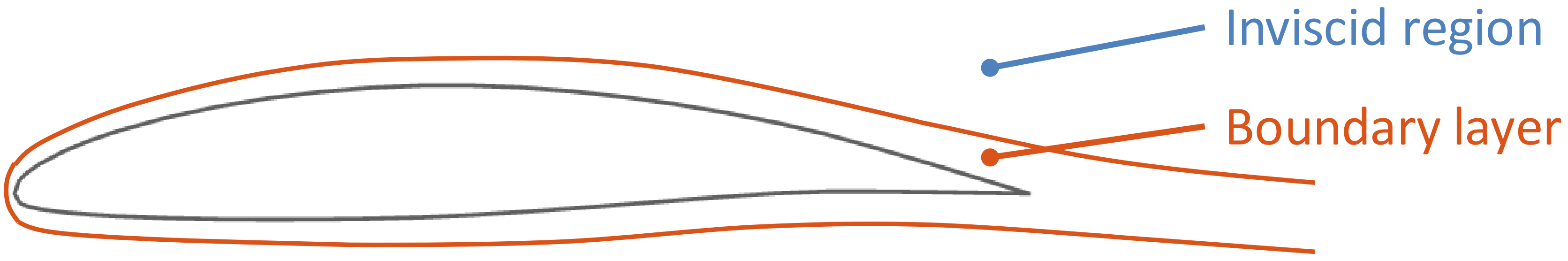


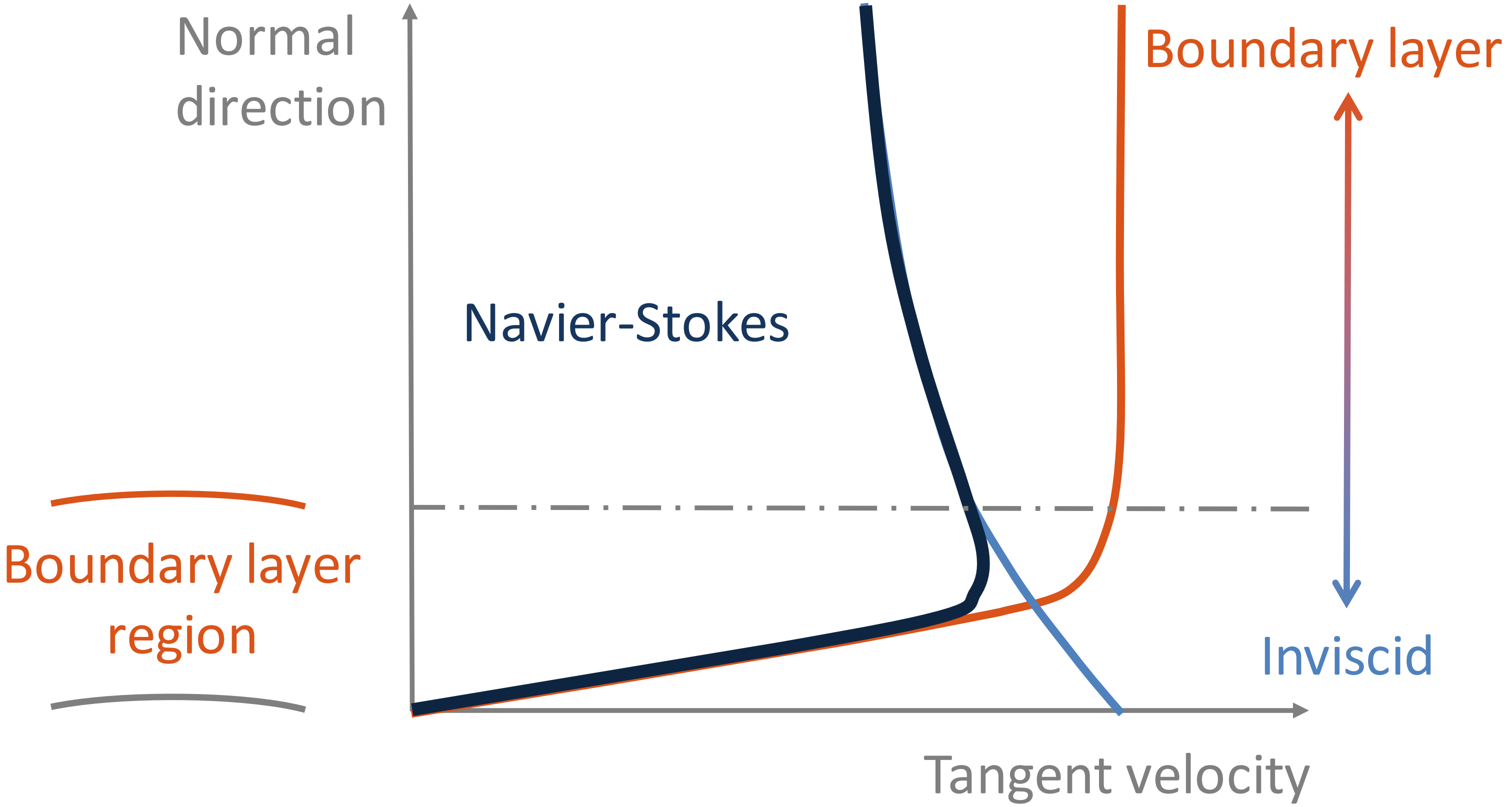
No friction in inviscid flow

- stronger shock ← higher total pressure gradient
- aft location ← later compression

Boundary layer must be taken into account for **accurate predictions**

Viscous-inviscid interaction





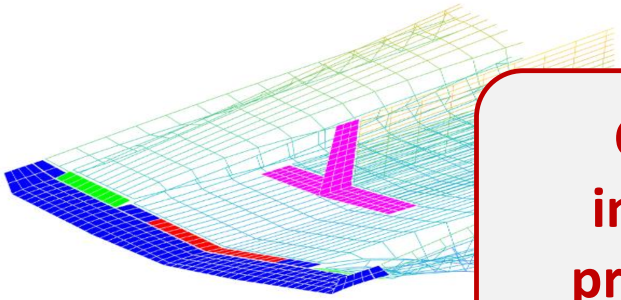
Numerical methods

Boundary element method

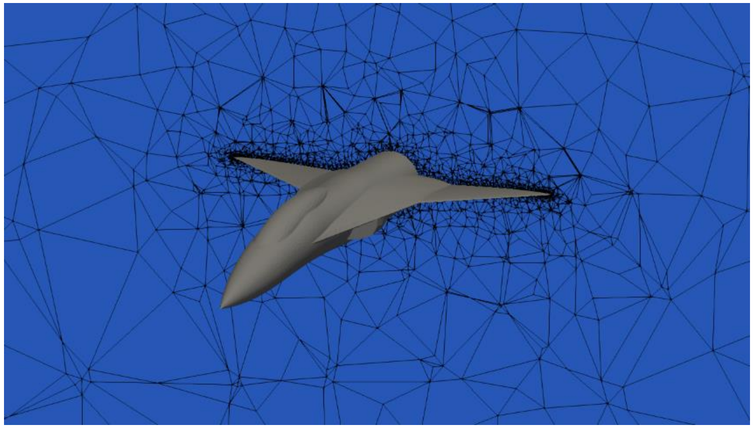
- Only boundary is discretized
- Linear equations only
- Panel/lattice/particle methods

Field method

- Whole field is discretized
- Linear and nonlinear equations
- Finite volume/element methods



Current industrial practice for aeroelastic computations

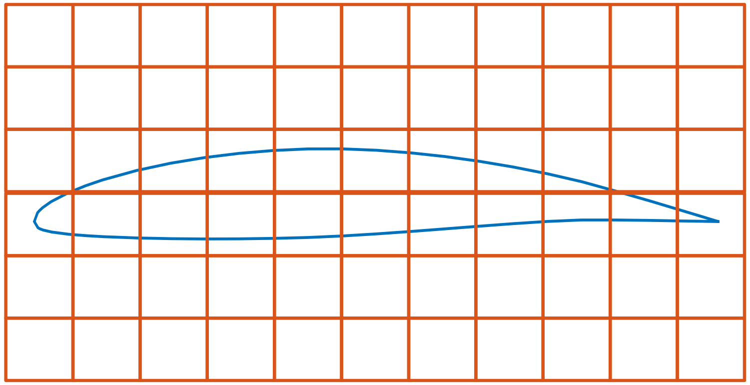


Field panel method

Combination

Boundary element method
- Linear part
- On the wing surface

Field method
- Nonlinear part
- In the field



Advantages
- Extension to panel method
- Simple grid generation

Disadvantages
- High memory requirement
- Disagreement in literature

Outline

Modeling

- Optimization formulation
- Aerodynamic models
- Numerical methods

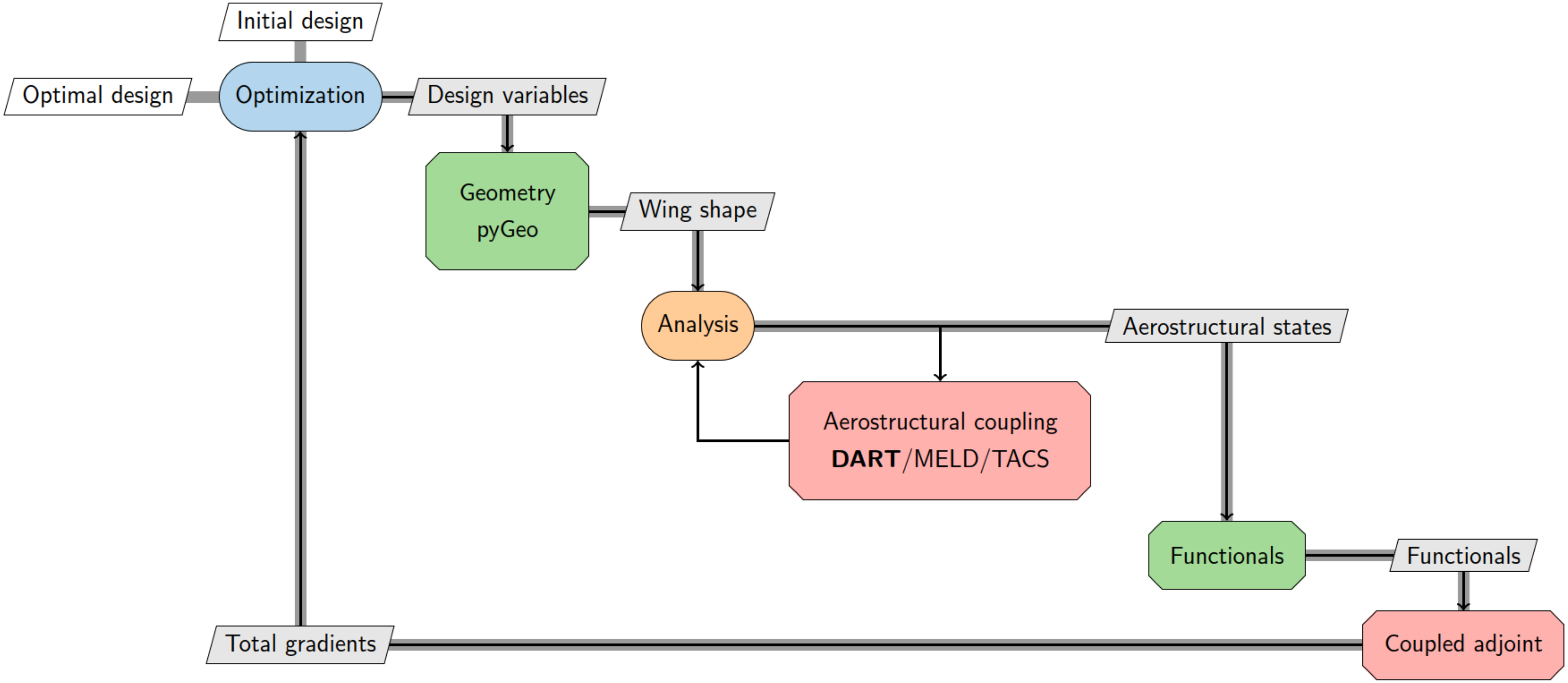
Static aeroelasticity

- Framework
- DART code
- Applications

Dynamic aeroelasticity

- Framework
- SDPM and NIPK codes
- Applications

Optimization framework



Initial design

Optimal design

Optimization

Design variables

Geometry
pyGeo

Wing shape

Analysis

Aerostructural states

Aerostructural coupling
DART/MELD/TACS

Functionals

Functionals

Coupled adjoint

Total gradients

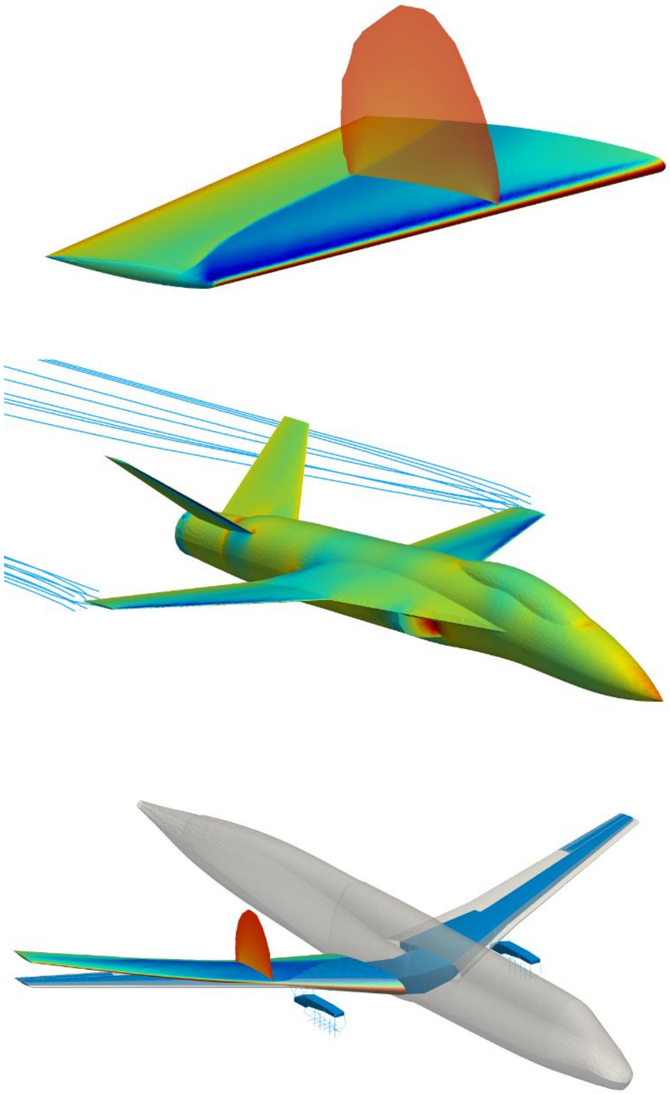
openMDAO

https://openmdao.org

MPhys

https://github.com/openmdao/mphys

DART



Discrete Adjoint for Rapid Transonic Flows

- Steady full potential formulation
- Finite element discretization
- Unstructured tetrahedral grid
- Analytical discrete adjoint
- Mesh morphing
- Viscous-inviscid interaction
- C++ with Python API

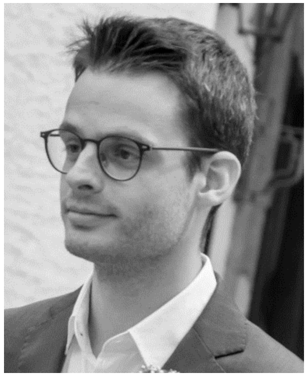
Performance (712Ke – 4.3GB @ 3.4GHz)

- Solution – 100 s
- Morphing – 25 s
- Gradient – 45 s

https://gitlab.uliege.be/am-dept/dartflo

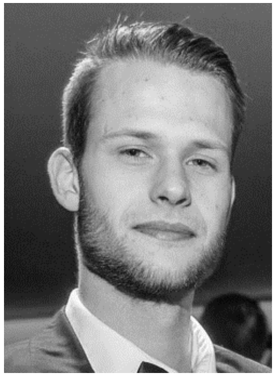
Acknowledgements

Lead developer



Adrien Crovato

Current developers



Paul Dechamps

Romain Boman

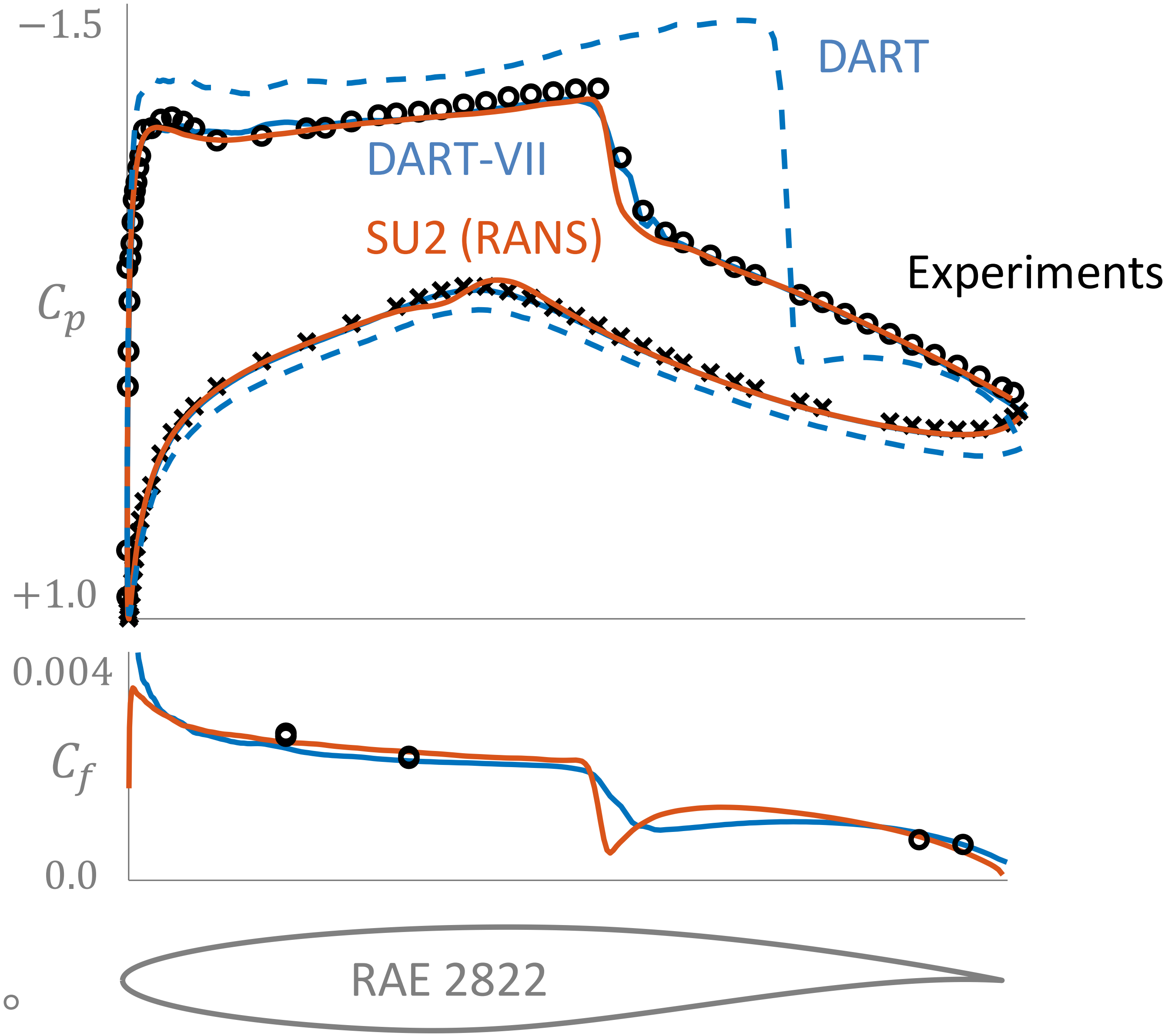
Former collaborators

Amaury Bilocq

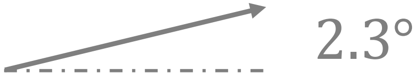
Guillaume Brian

Guillem Batlle i Capa

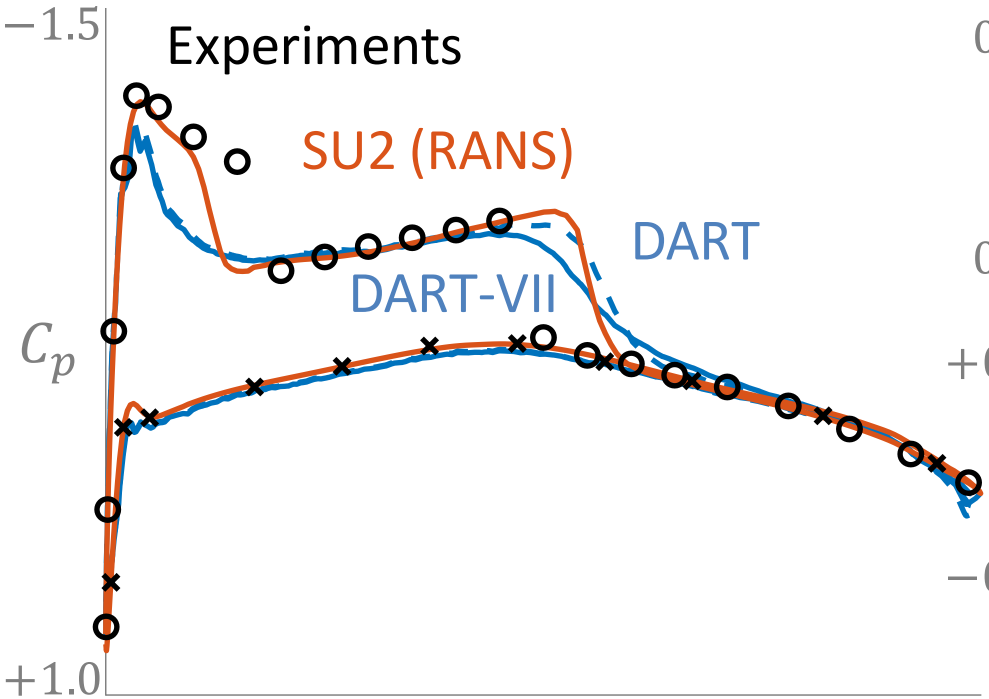
Two–dimensional viscous analysis

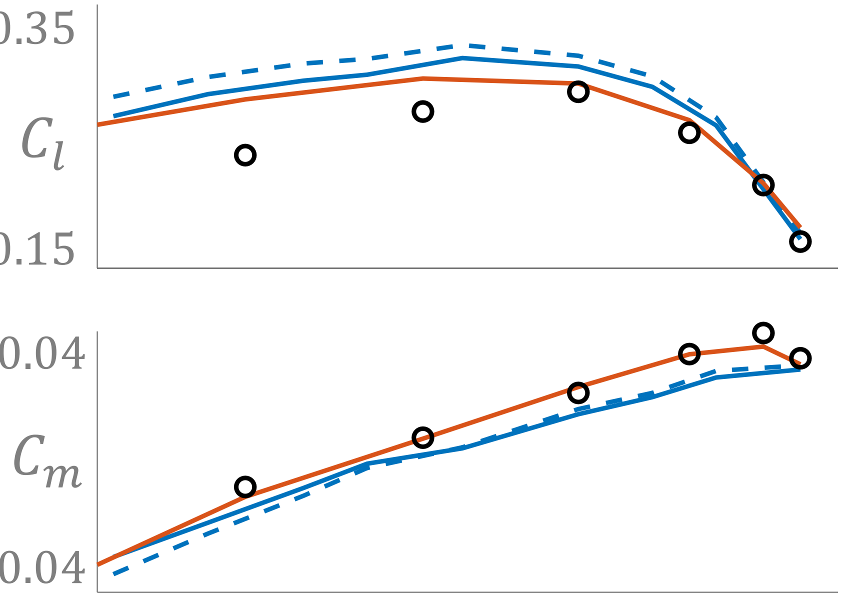


$M_\infty = 0.73$
$Re = 6.5\text{ M}$

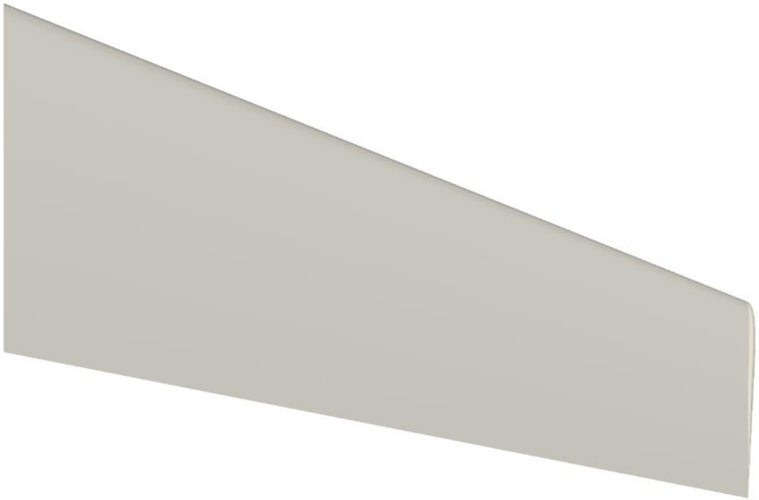


Three–dimensional viscous analysis





$M_\infty = 0.94$
$Re = 11.7$ M



Three–dimensional aeroelastic analysis

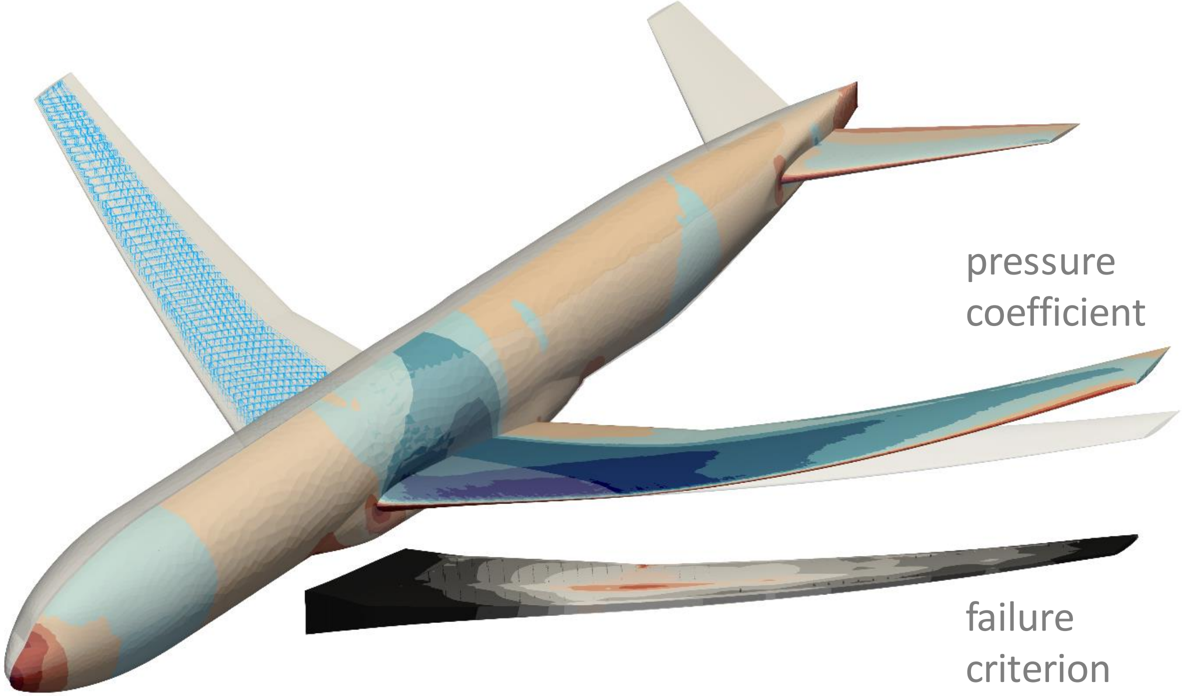
NASA CRM

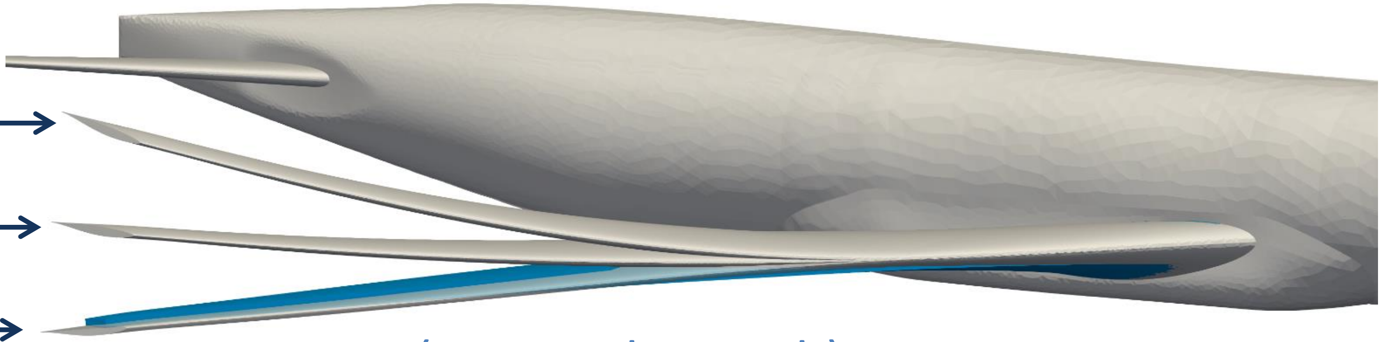
Cruise

$$\mathrm{M}_{\infty} = 0.85 \; - \; \mathrm{FL}\ 370$$
$$n = 1.0\ (C_L = 0.5)$$

Maneuver

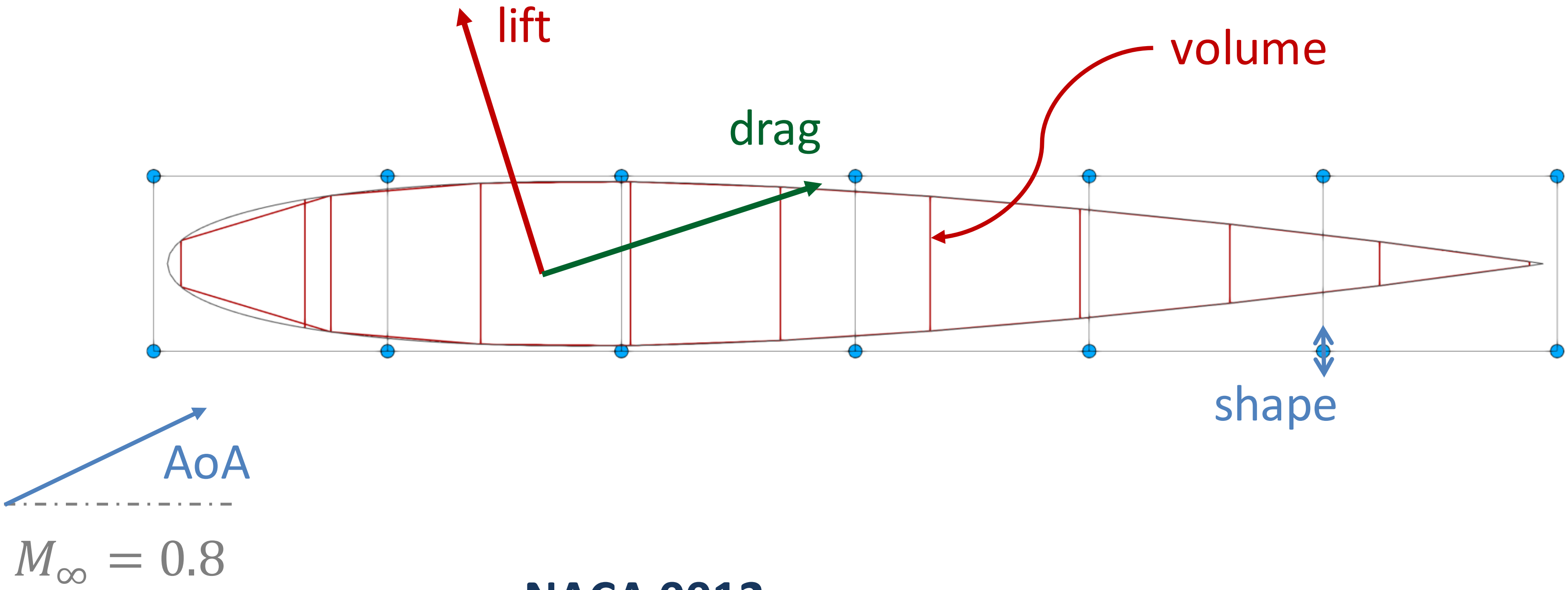
$$\mathrm{M}_{\infty} = 0.85 \; - \; \mathrm{FL}\ 200$$
$$n = 2.5$$





uCRM-9 (MDO Lab UMich)

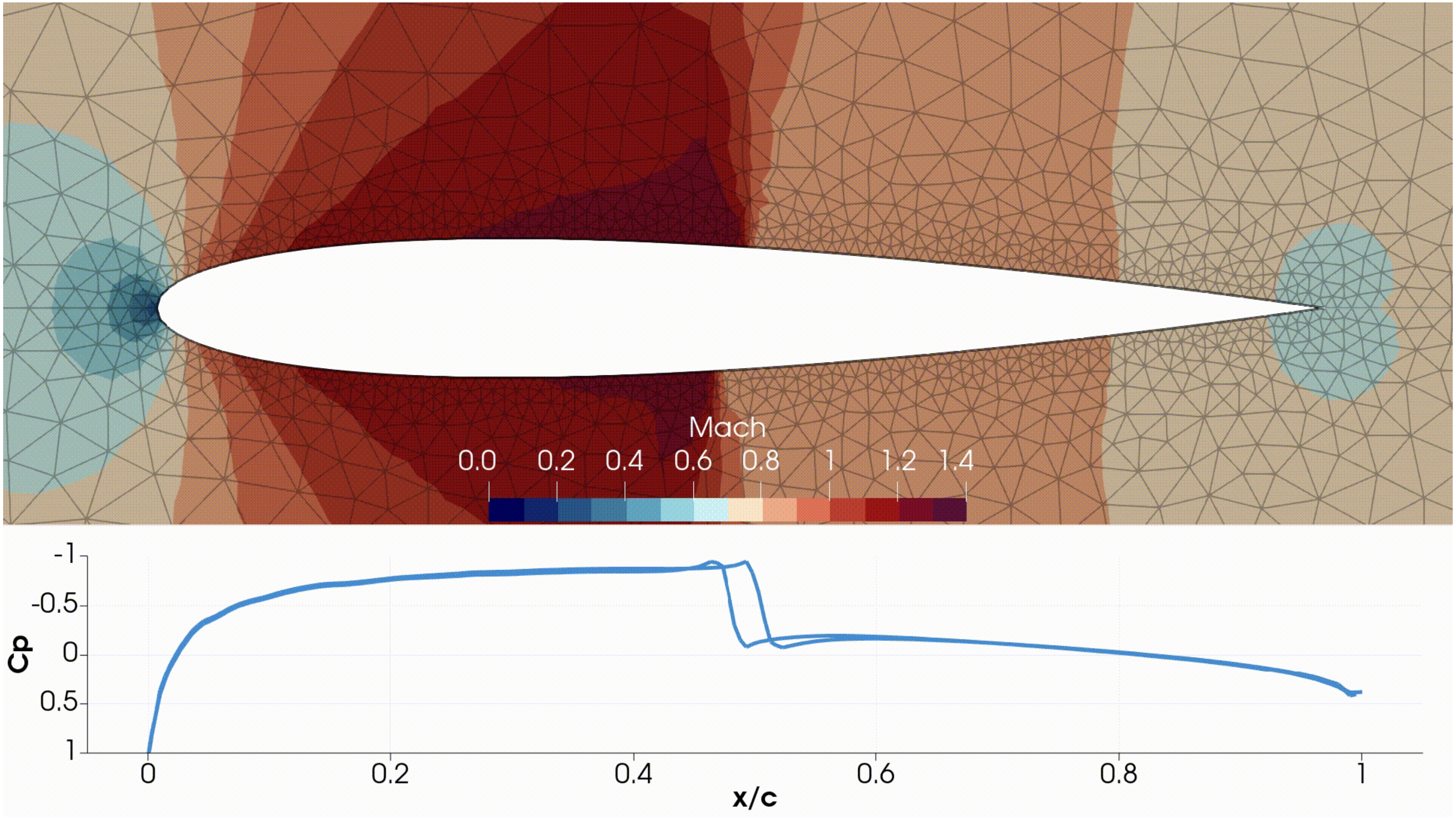
Two–dimensional shape optimization



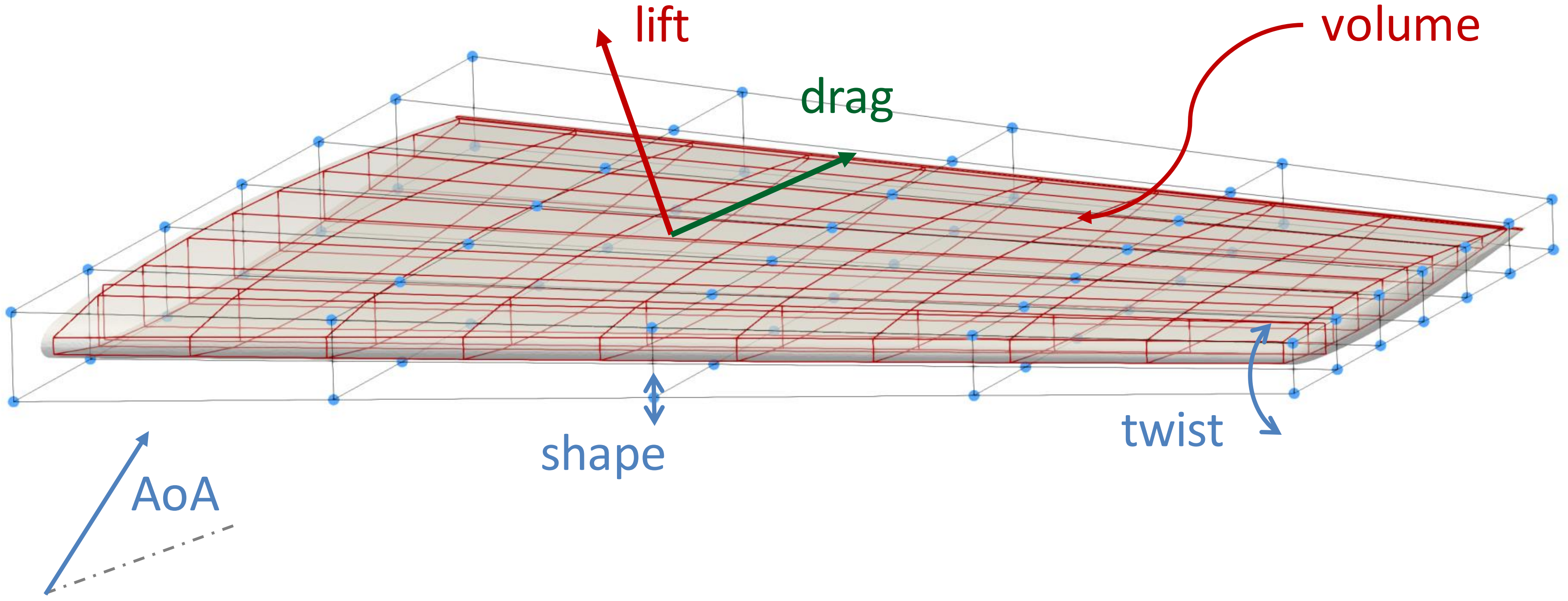
NACA 0012

$$\begin{array}{ll} \text{min} & \text{drag} \\ \text{w.r.t.} & \text{AoA, shape} \\ \text{s.t.} & \text{lift} \\ & \text{internal volume} \end{array}$$

Two–dimensional shape optimization



Three–dimensional shape optimization



$M_\infty = 0.83$

ONERA M6

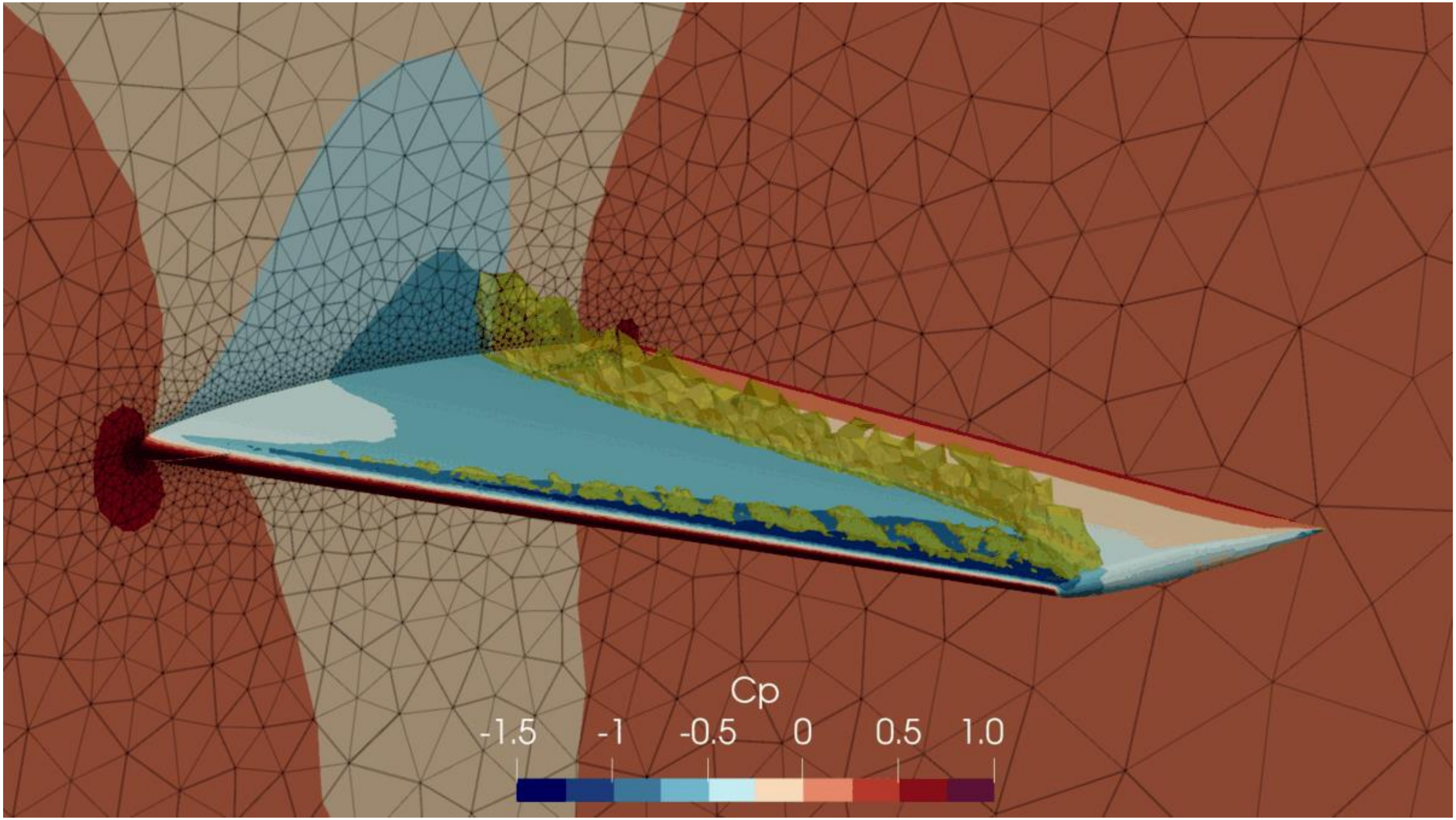
$\text{min} \quad \text{drag}$

$\text{w.r.t.} \quad \text{AoA, shape, twist}$

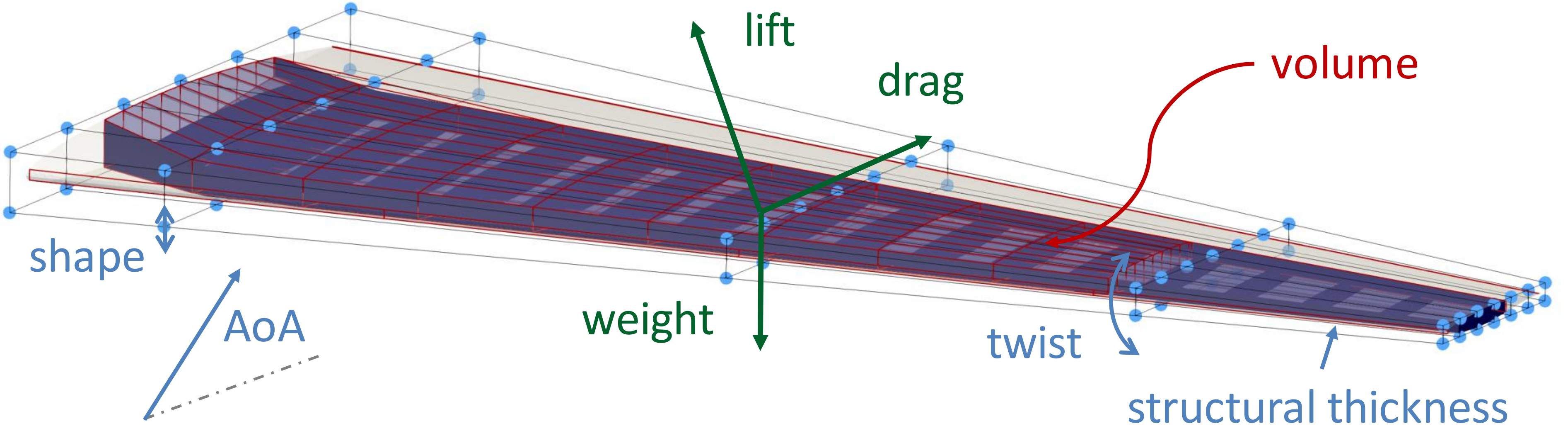
$\text{s.t.} \quad \text{lift}$

$\quad \text{internal volume}$

Three–dimensional shape optimization



Three–dimensional aeroelastic optimization



RAE

Cruise

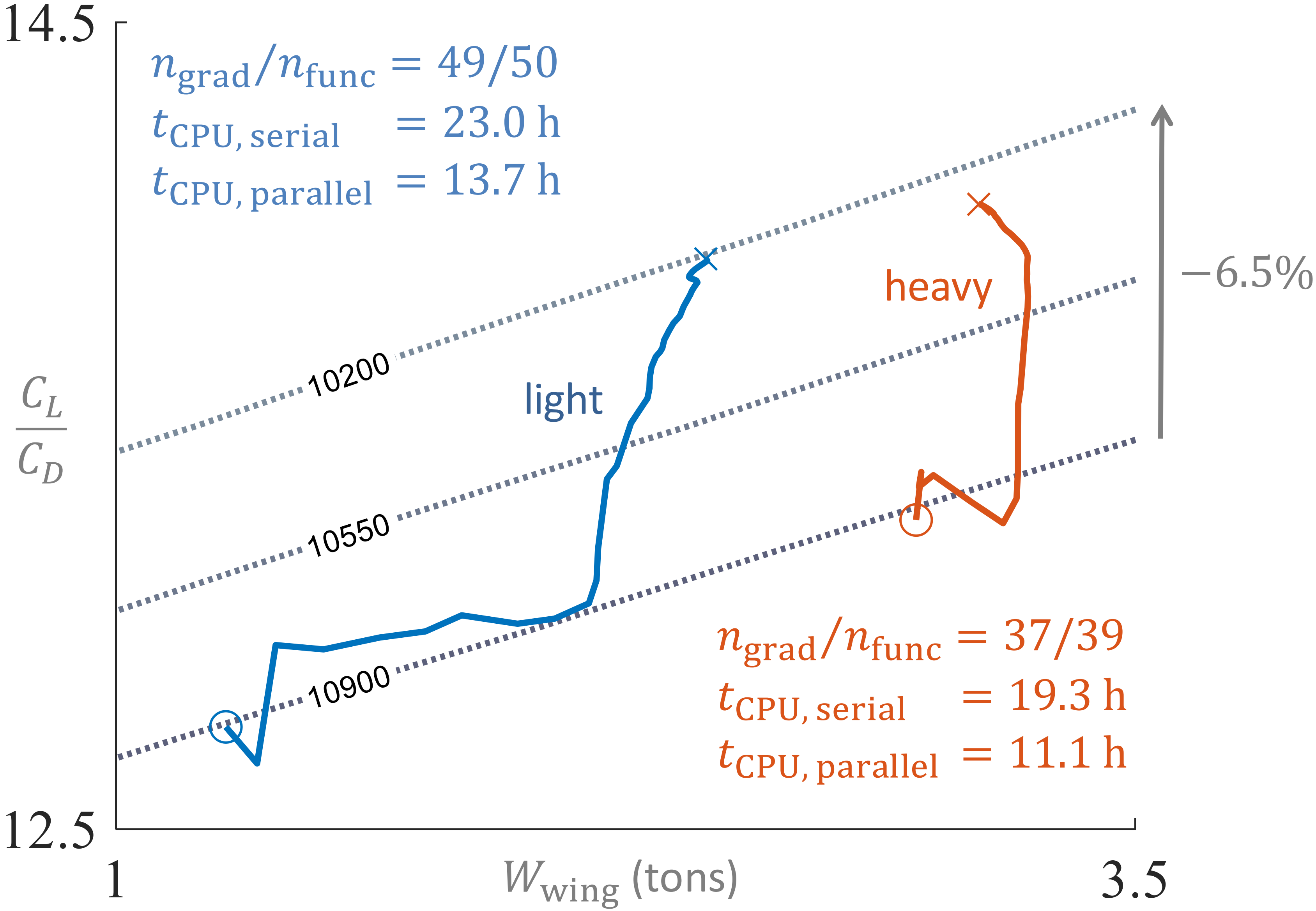
$$M_\infty = 0.82 - \text{FL } 350$$

Maneuver

$$M_\infty = 0.78 - \text{FL } 200$$

$$\begin{aligned} \min \quad & \text{fuel} = \text{Breguet}(\text{lift}, \text{drag}, \text{weight}) \\ \text{w.r.t.} \quad & \text{AoA}, \text{shape}, \text{twist}, \text{structural thickness} \\ \text{s.t.} \quad & \text{load factor} \\ & \text{internal volume} \\ & \text{structural adjacency} \\ & \text{structural failure} \end{aligned}$$

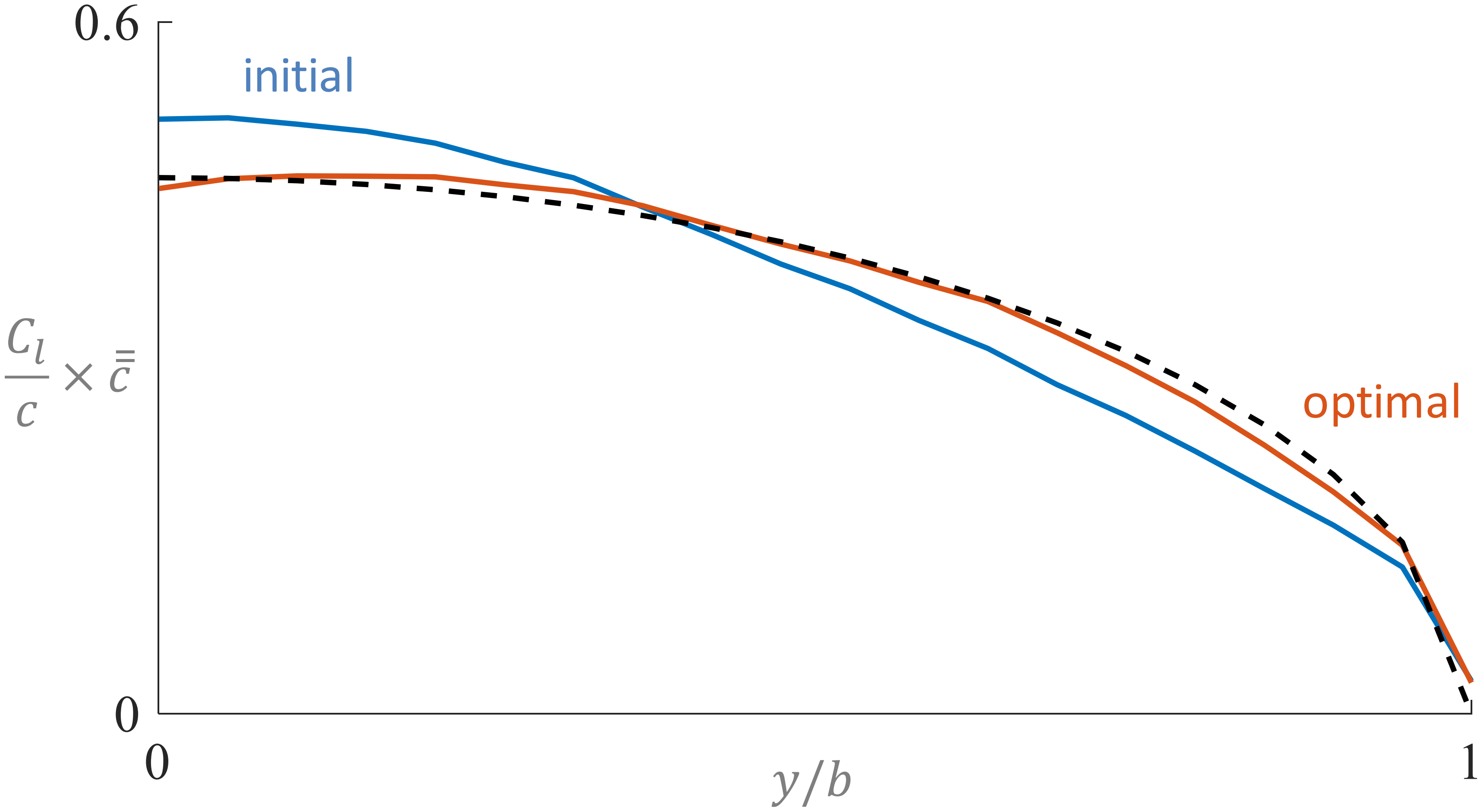
Fuel burn

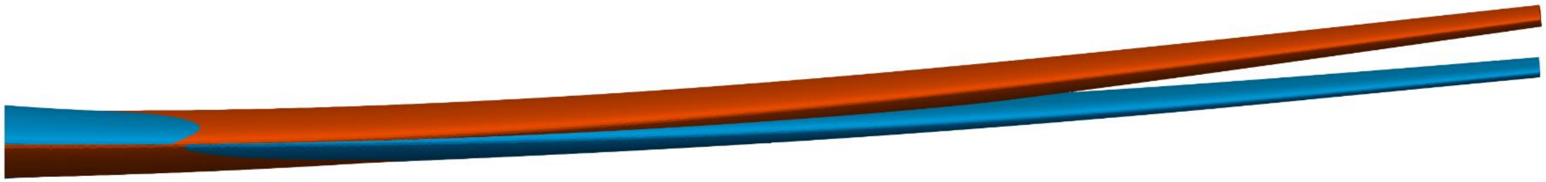


$n_{\text{grad}}/n_{\text{func}} = 49/50$
$t_{\text{CPU, serial}} = 23.0\ \text{h}$
$t_{\text{CPU, parallel}} = 13.7\ \text{h}$

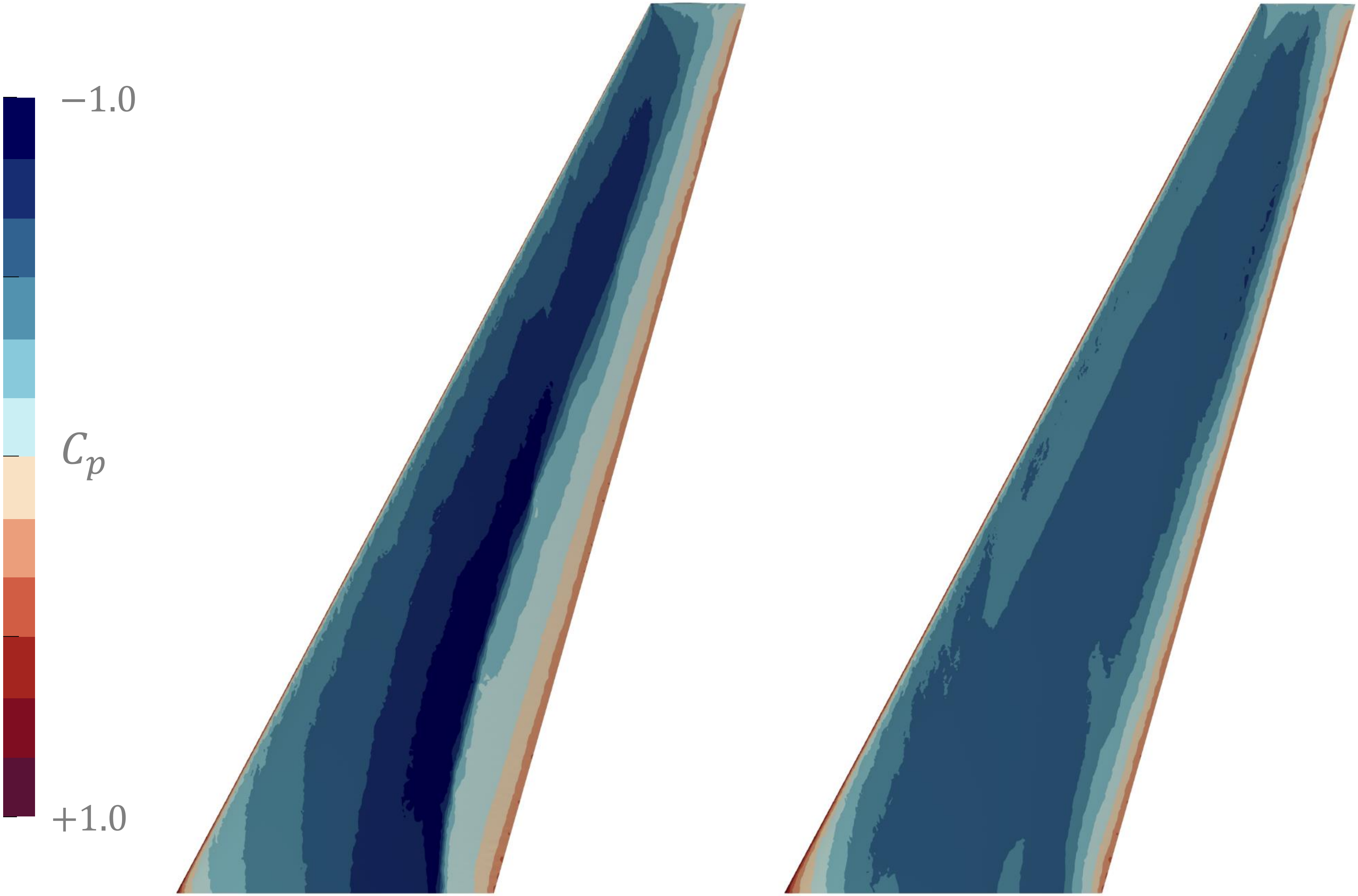
$n_{\text{grad}}/n_{\text{func}} = 37/39$
$t_{\text{CPU, serial}} = 19.3\ \text{h}$
$t_{\text{CPU, parallel}} = 11.1\ \text{h}$

Lift distribution – cruise





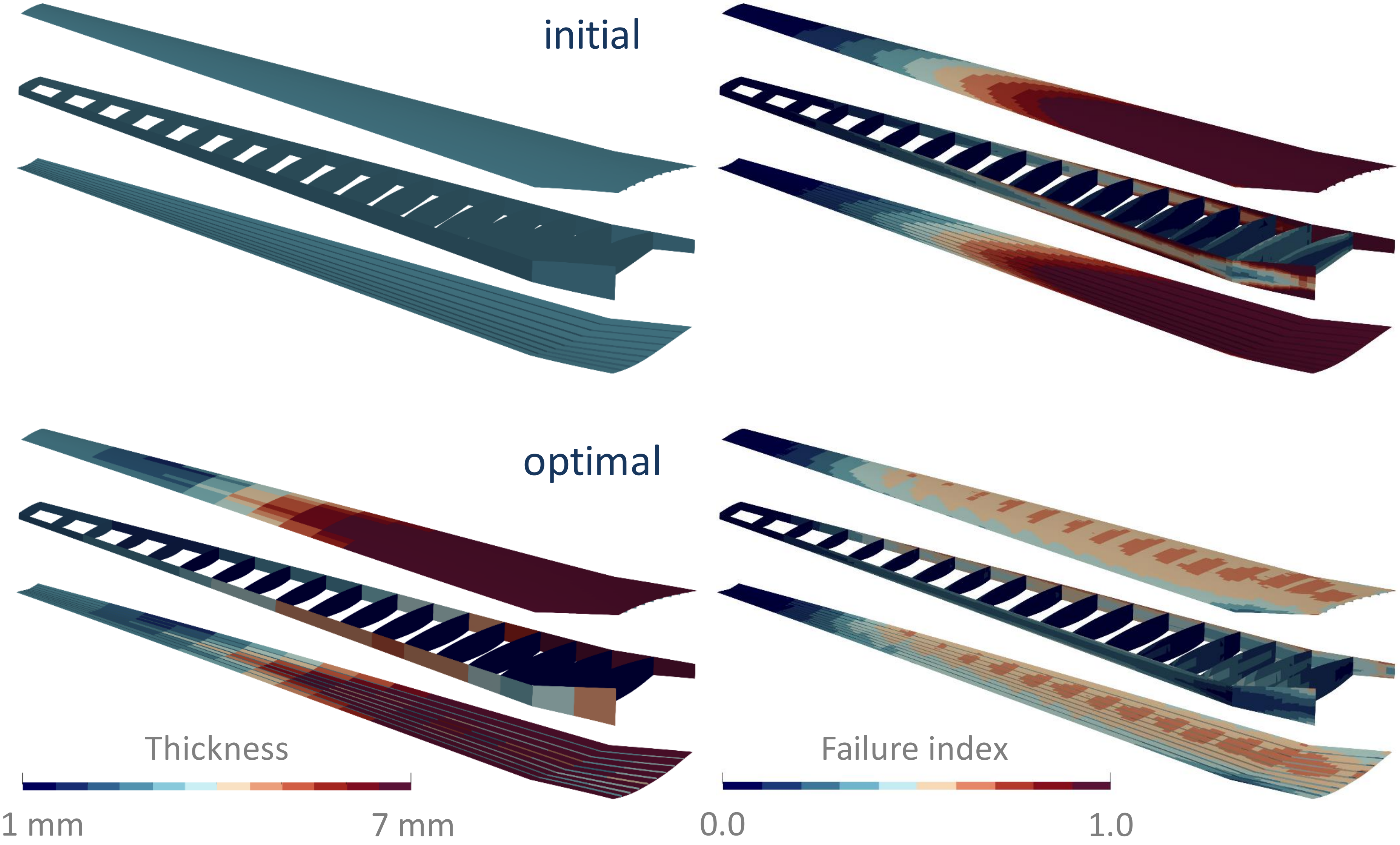
Pressure coefficient – cruise



initial

optimal

Thickness and failure index – maneuver



Outline

Modeling

- Optimization formulation
- Aerodynamic models
- Numerical methods

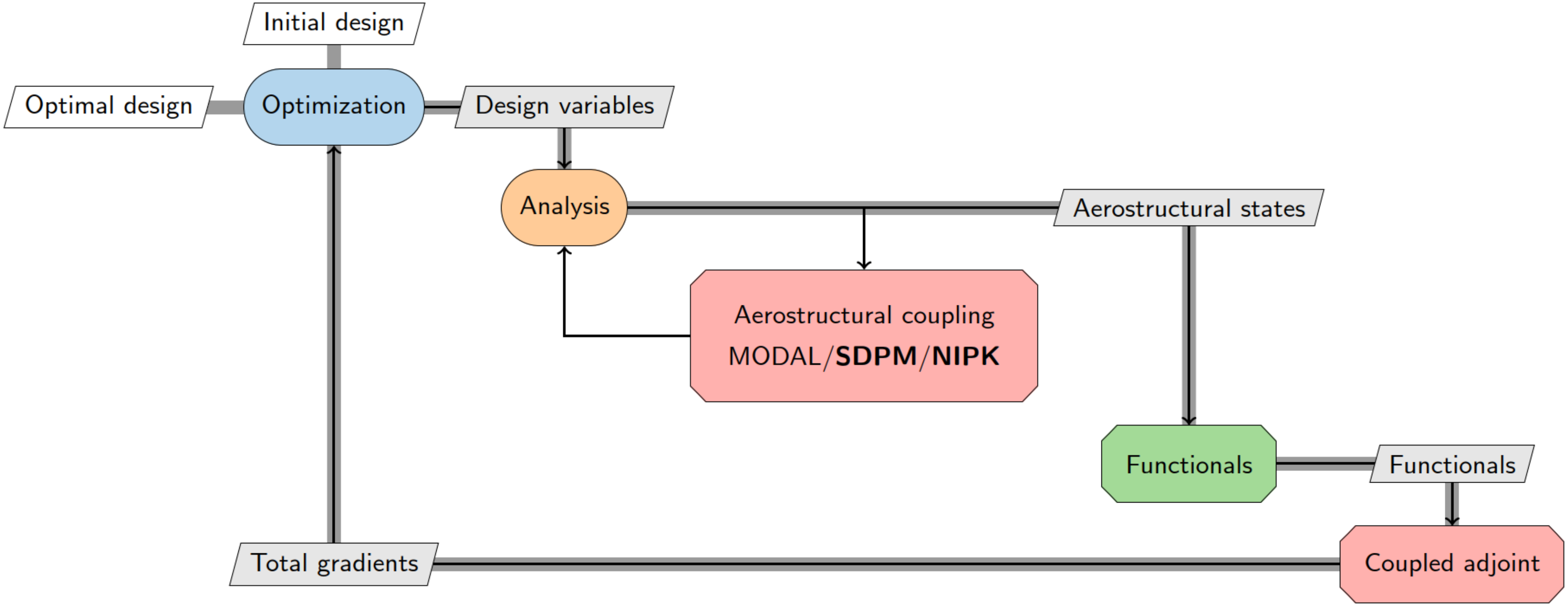
Static aeroelasticity

- Framework
- DART code
- Applications

Dynamic aeroelasticity

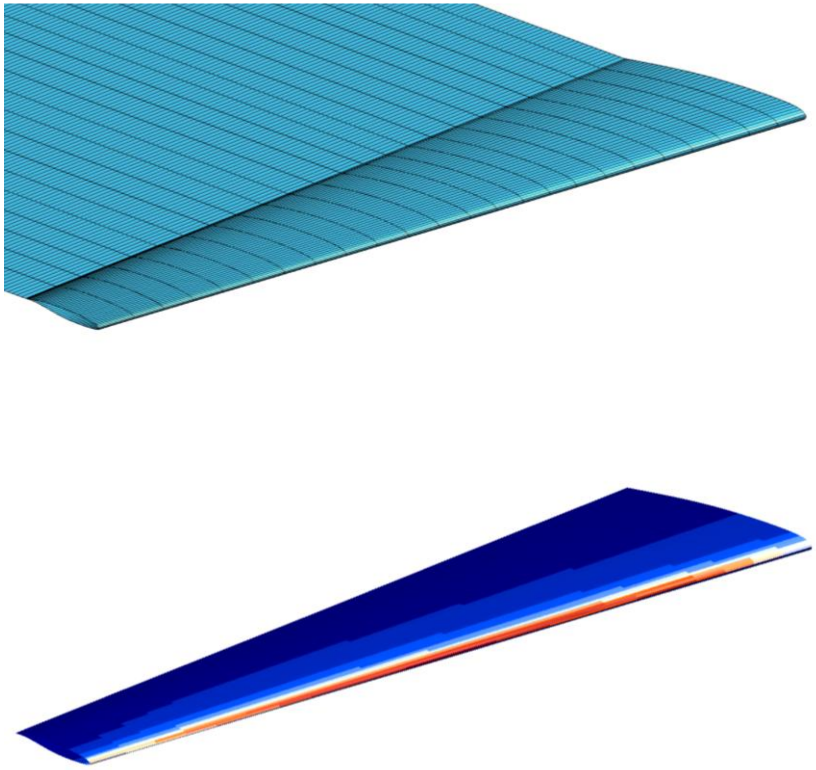
- Framework
- SDPM and NIPK codes
- Applications

Optimization framework



https://openmdao.org

SDPM



Source and Doublet Panel Method

- Unsteady potential formulation
- Panel discretization
- Unstructured quadrangular grid
- Reverse automatic differentiation
- C++ with Python API

Performance (2Ke – 56GB @ 4.2GHz)

- Solution – 134 s
- Gradient – 40 s

https://gitlab.uliege.be/am-dept/sdpm

Flutter solution

Flutter equation

$$\left(\frac{u_\infty^2}{l_{\text{ref}}^2}p^2 M + K - \frac{1}{2}\rho_\infty u_\infty^2 Q(k)\right)q = 0$$

$$p = gk + ik$$

Frequency matching (p-k)

1. Guess $k = \omega_N \frac{l_{\text{ref}}}{u_\infty}$
2. Compute $Q(k)$
3. Solve eigenvalue problem for p
4. Compute $k = \Im(p)$
5. Repeat 2-4 until k has converged

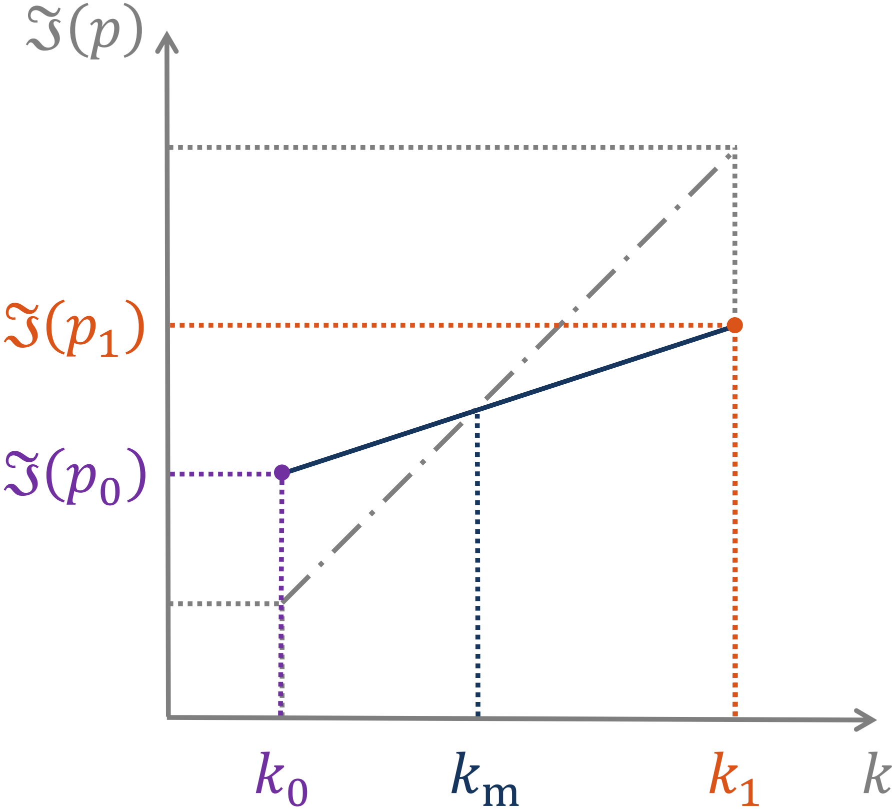
▶ **Computation of Q is costly**

▶ **Interpolate Q or k**

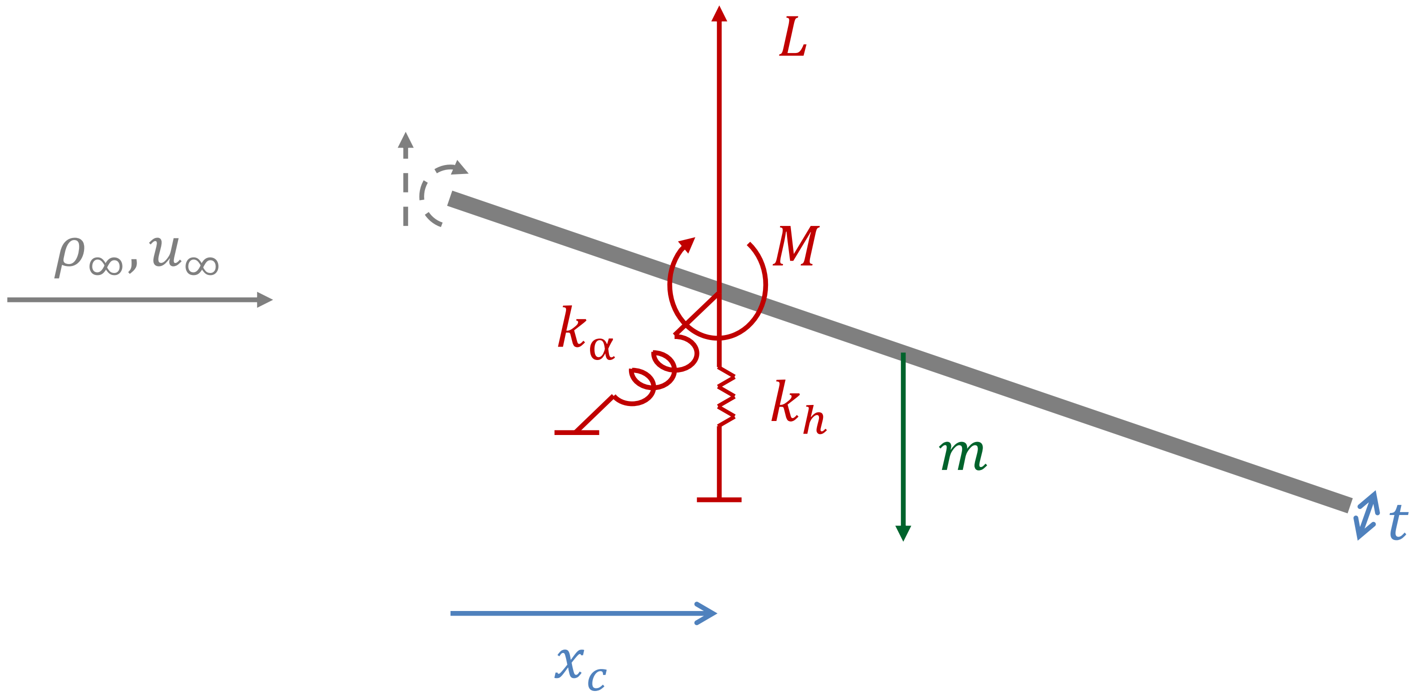
Non-iterative p-k method

Algorithm

1. Compute $Q_i(k_i)$ from a set of k_i
2. Solve eigenvalue problem for p_i
3. Interpolate k_m such that $\Im(p_\mathrm{m}) - k_\mathrm{m} = 0$



Flutter-constrained optimization



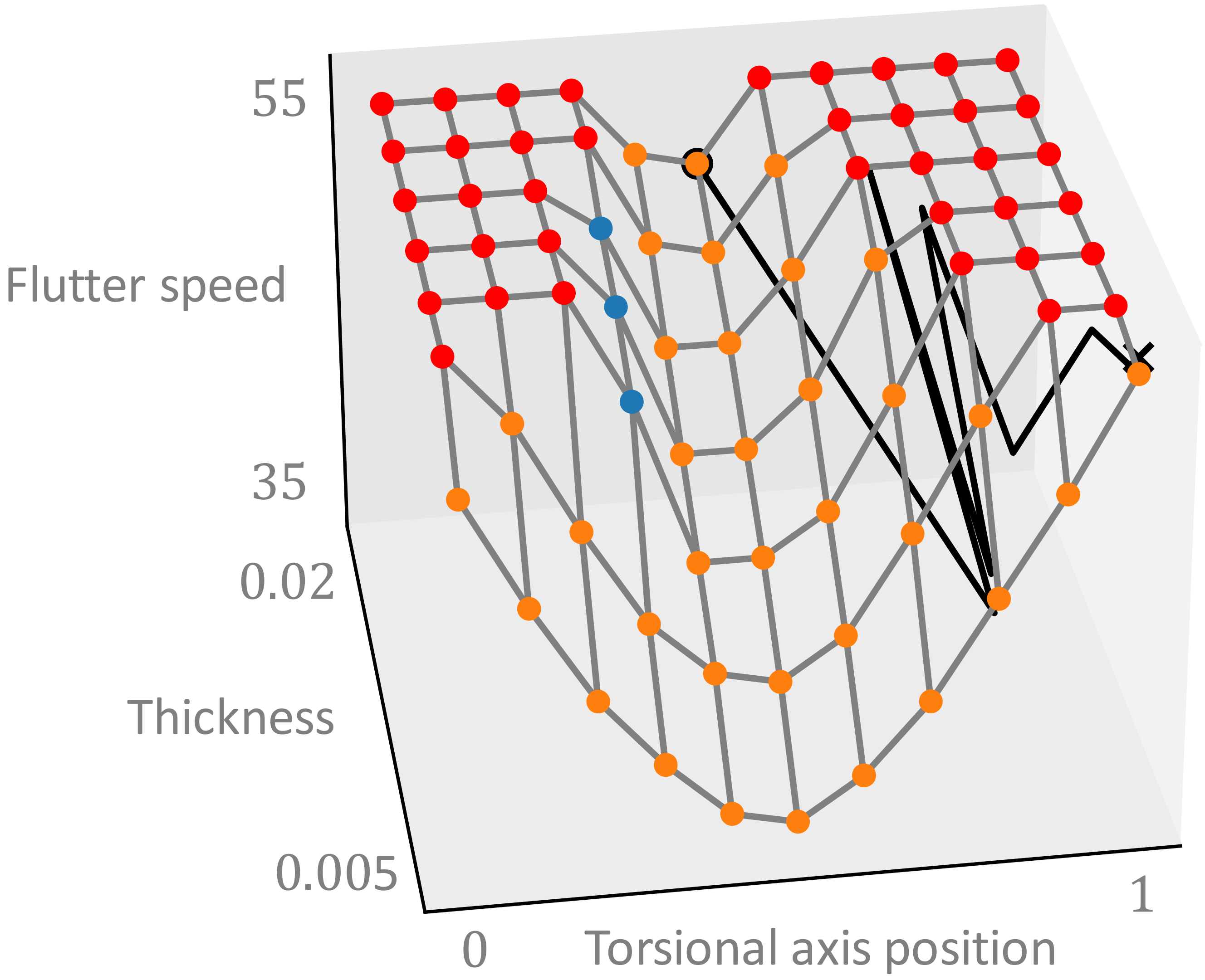
Pitch-plunge flat plate

$\min$ mass

w.r.t. torsion center position, thickness

s.t. flutter

Optimization path in parameter space



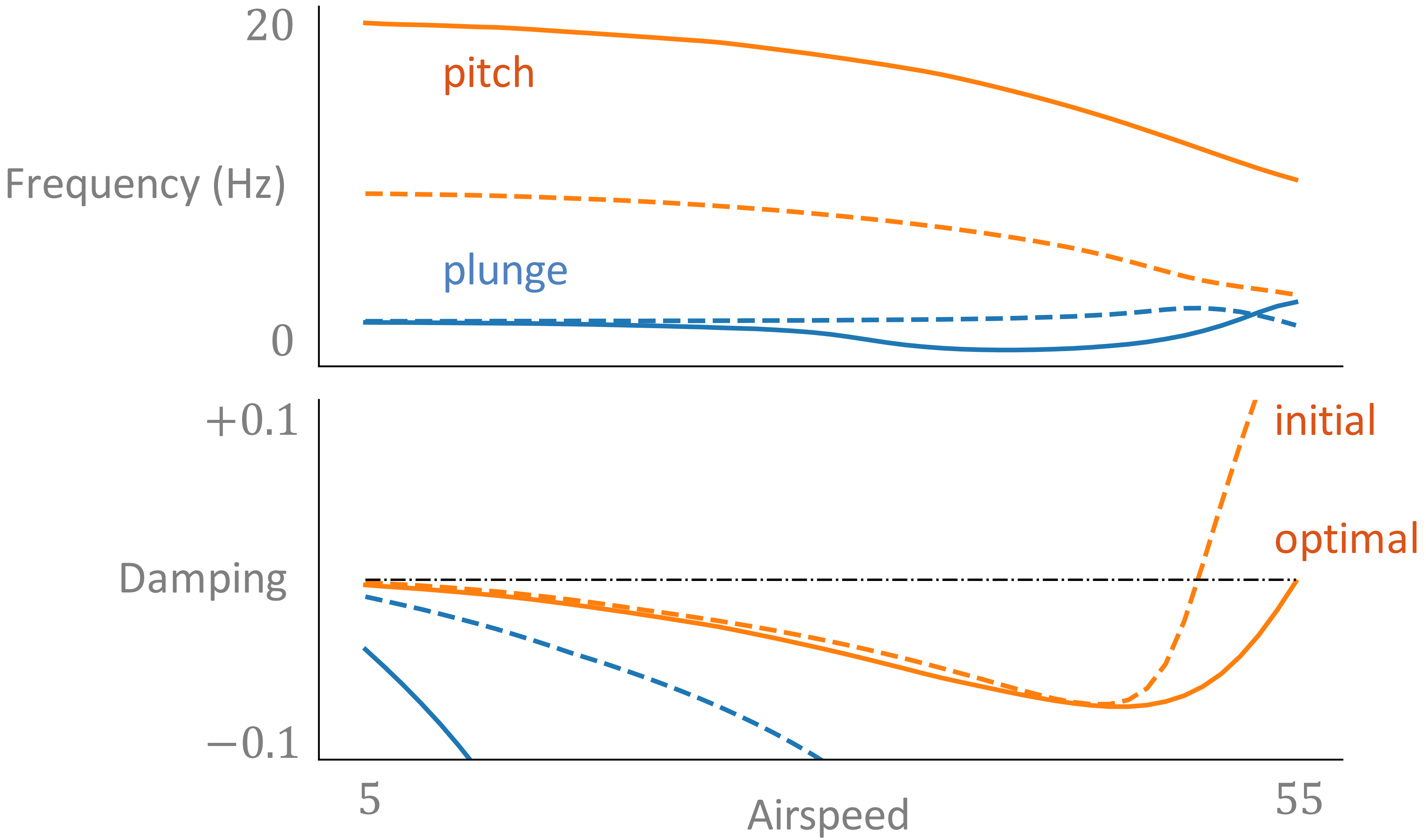
No flutter

Pitch flutter

Plunge flutter

Optimization path

Frequency-damping plots



Conclusion

Main points

- **Aerostructural optimization** is performed in **preliminary aircraft design**; choosing the **appropriate numerical models** and **methods** is of paramount importance
- **Developed DART** and **interfaced** with **OpenMDAO**; **relevant** results for **static aerostructural** calculations can be obtained within **a day**
- **Implemented NIPK** and **interfaced** with **OpenMDAO**; can effectively **suppress flutter** for **dynamic aerostructural** calculations

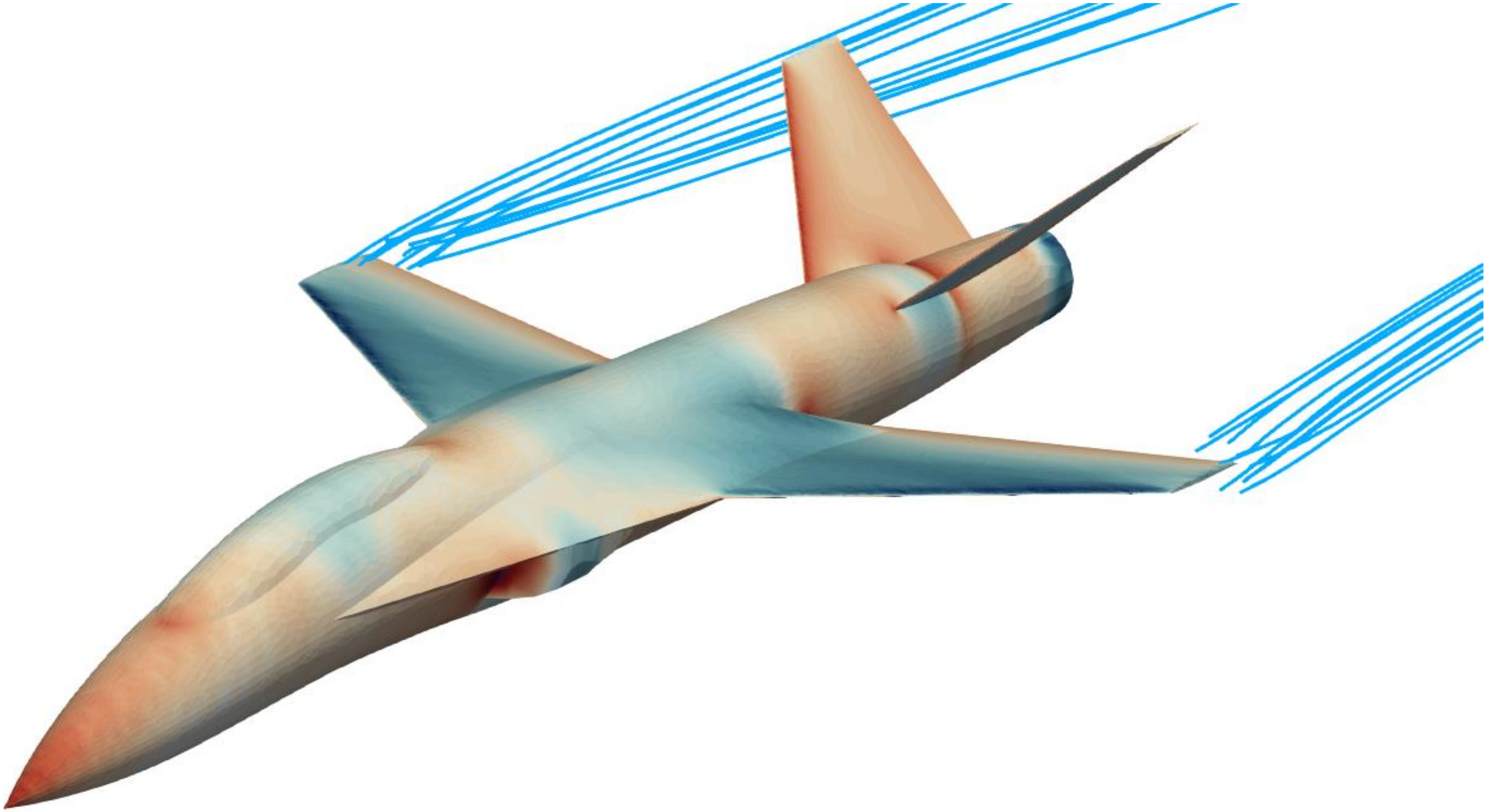
Next steps

- **Integrate viscous-inviscid interaction** in static optimization
- **Integrate SDPM** in dynamic optimization
- **Use full aircraft** configuration and realistic **composite** structure

CNAM seminar

Aerostructural modeling for preliminary aircraft design

Adrien Crovato – Paris, November 2023



https://acrovato.github.io