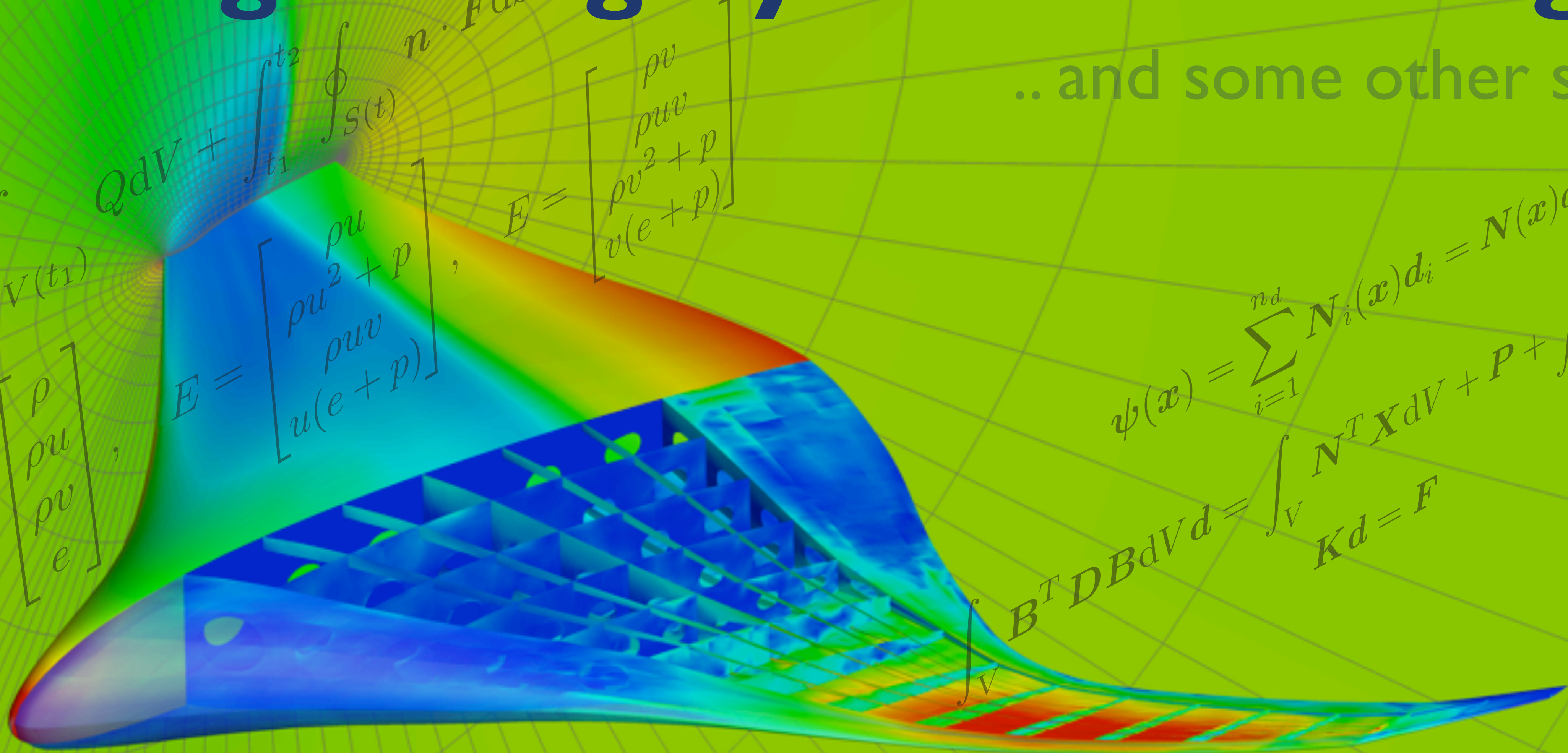
High-Fidelity Optimal Aeroelastic Tailoring of Highly Flexible Wings

.. and some other stuff



AEROSPACE ENGINEERING
UNIVERSITY of MICHIGAN

Joaquim R. R. A. Martins
Graeme Kennedy • Gaetan Kenway • John Hwang
Multidisciplinary Design Optimization Laboratory
http://mdolab.engin.umich.edu

MDO Consortium Workshop, Purdue University — July 19, 2012

What has happened since last year?

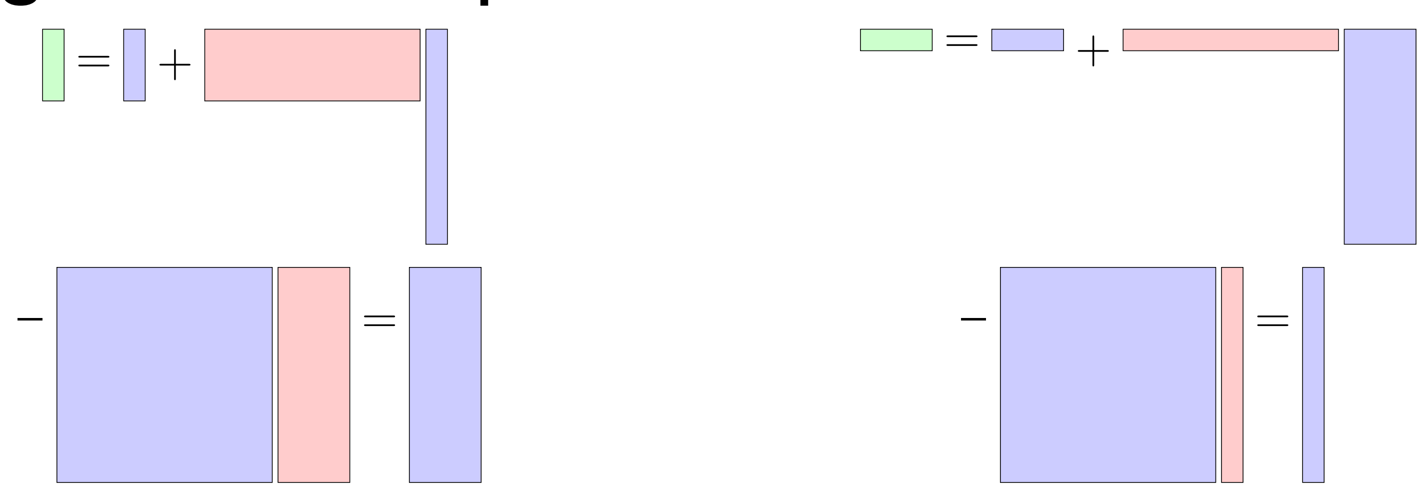
- XDSM paper has appeared:

  A. B. Lambe and J. R. R. A. Martins. "Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes". *Structural and Multidisciplinary Optimization*, 46:273–284, 2012. doi: 10.1007/s00158-012-0763-y.

- MDO survey was submitted to AIAAJ, and is now in Revision 1. Draft available at: http://mdolab.engin.umich.edu/publications
- New paper on computing derivatives for coupled systems; presented at the AIAA SDM
- New aerostructural design optimization results
- New CAD-free geometry engine in development

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} + \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}$$

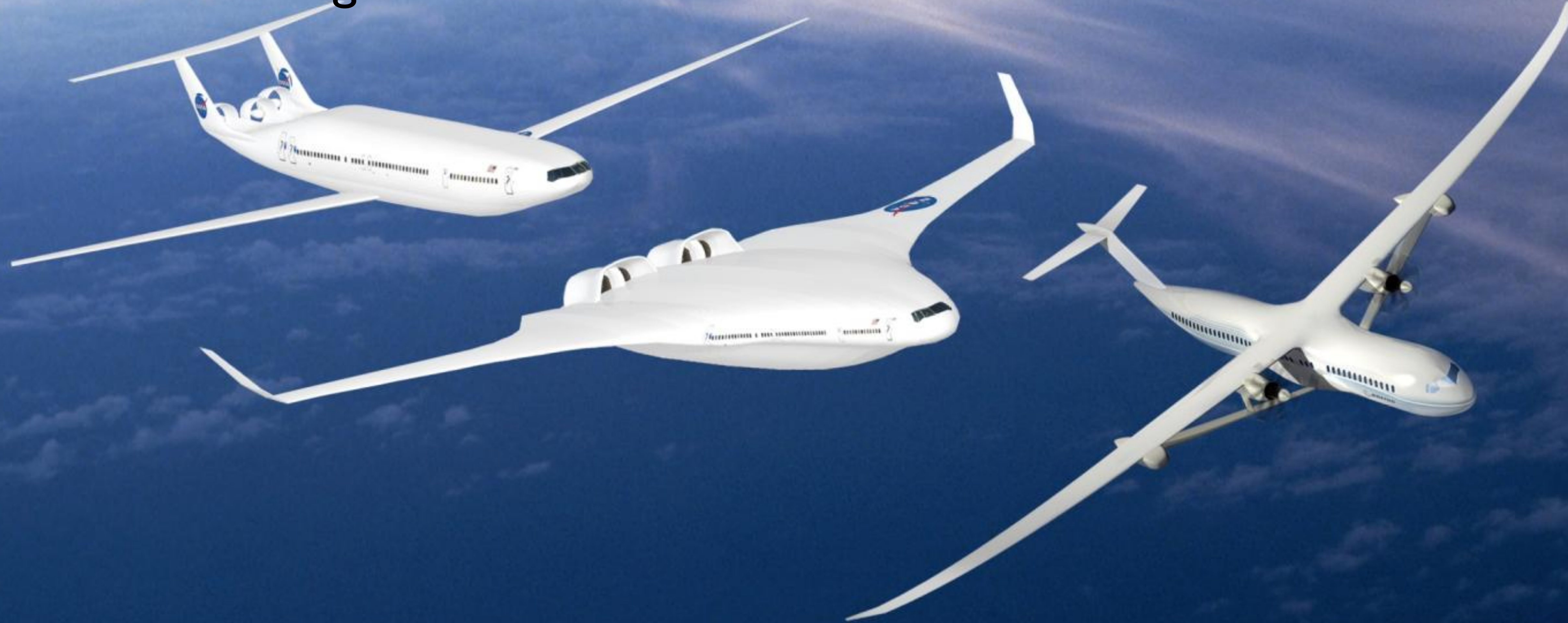
$$-\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}}\right]^T \left[\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}}\right]^T = \left[\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}}\right]^T$$



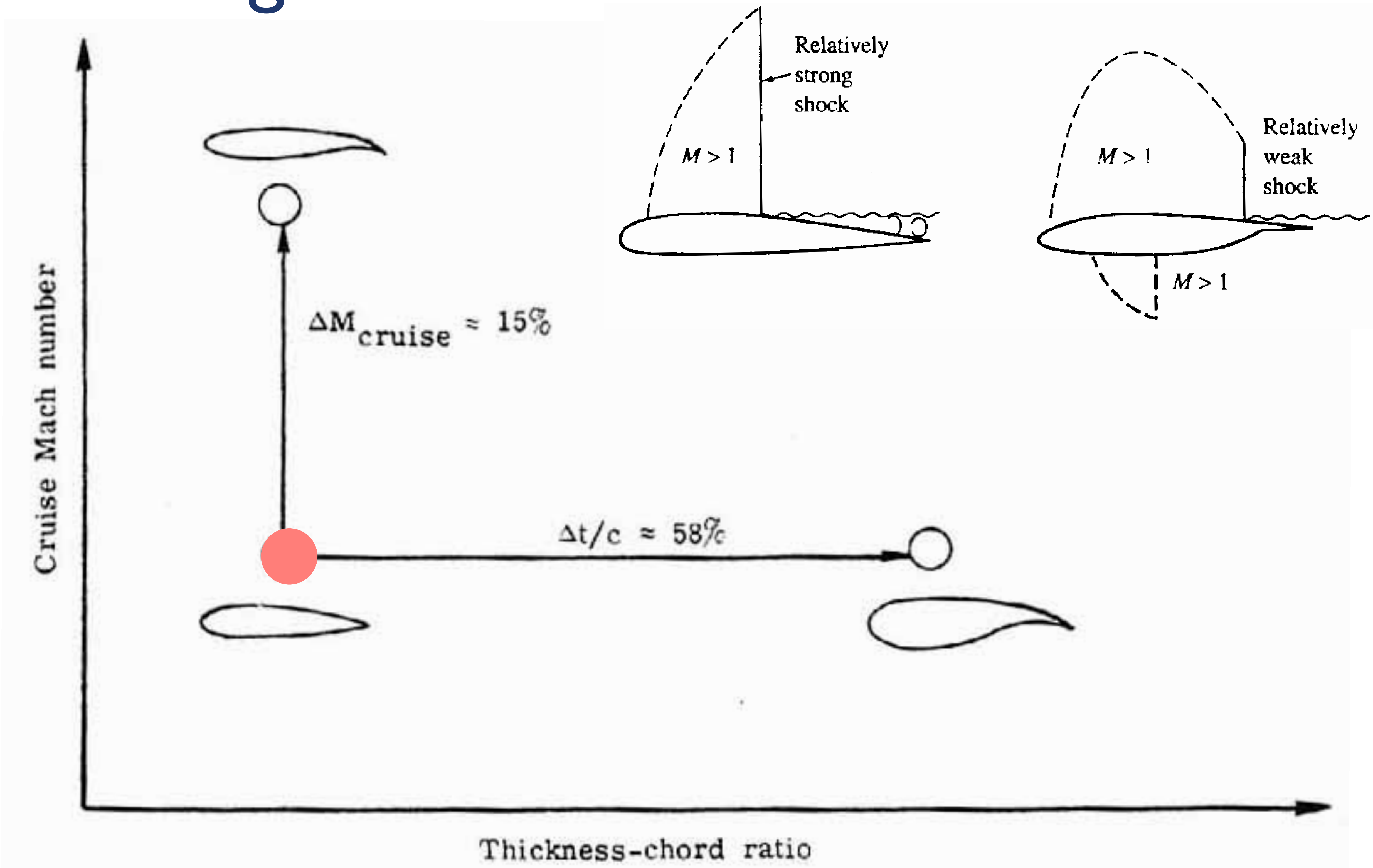
What motivates MDO?

The next generation of aircraft demands even more of the design process

- Highly-flexible high aspect ratio wings
- Unknown design space and interdisciplinary trade-offs
- High risk

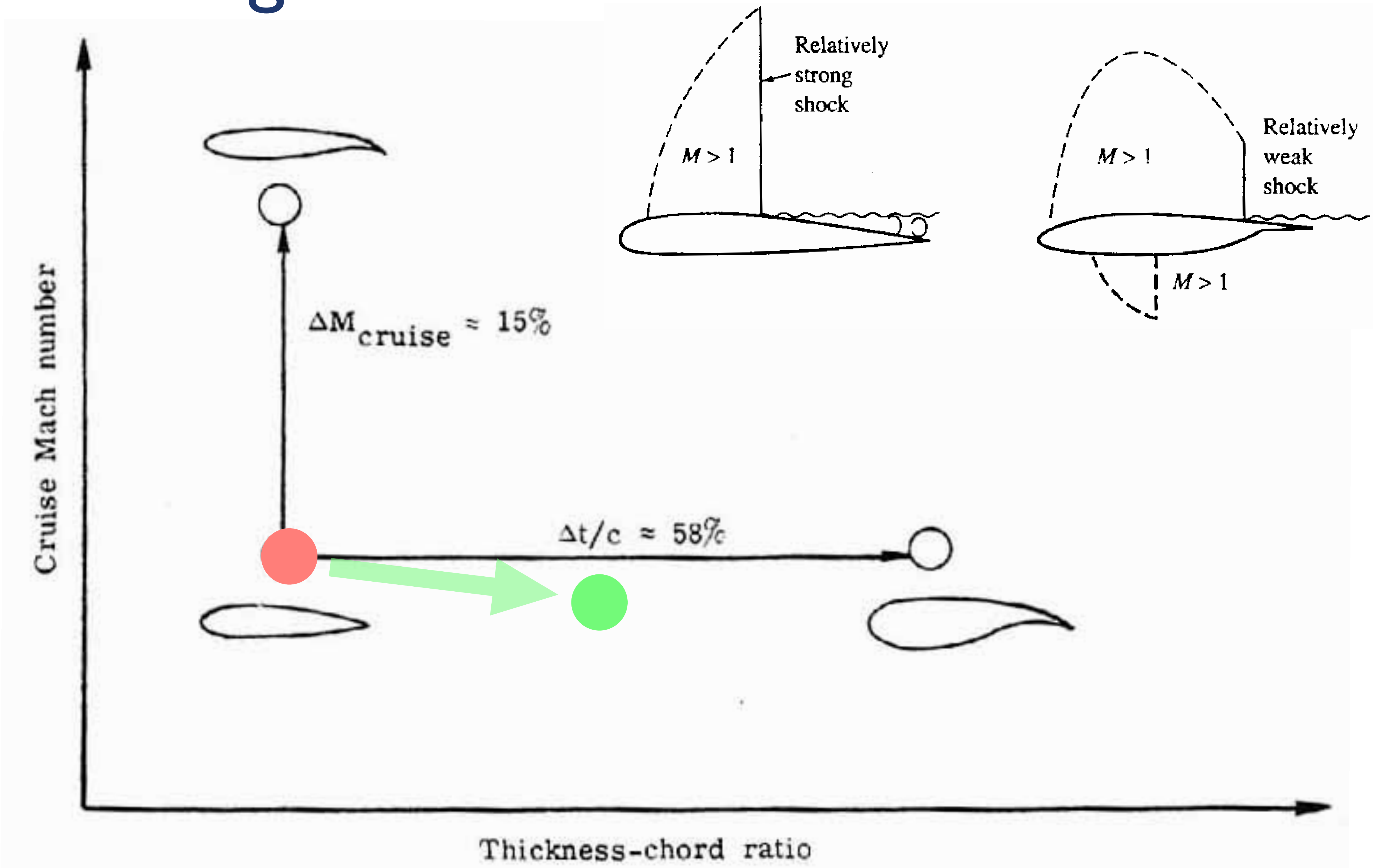


Why you should not trust an aerodynamicist (even a brilliant one) to make design decisions



[NASA]

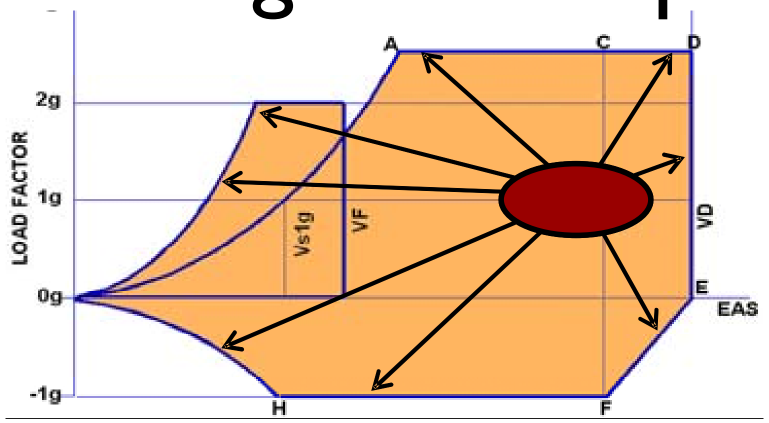
Why you should not trust an aerodynamicist (even a brilliant one) to make design decisions

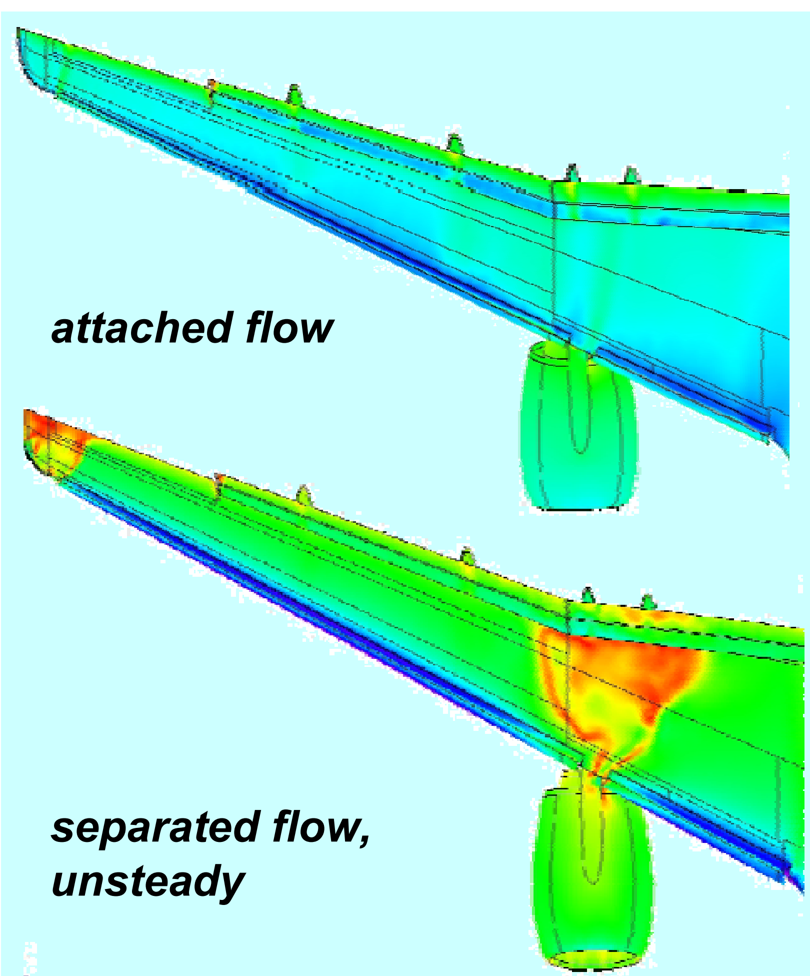


[NASA]

Next generation MDO will be computationally demanding...

Full flight envelope





[DLR]

Configurations

clean

airbrakes deployed

high lift

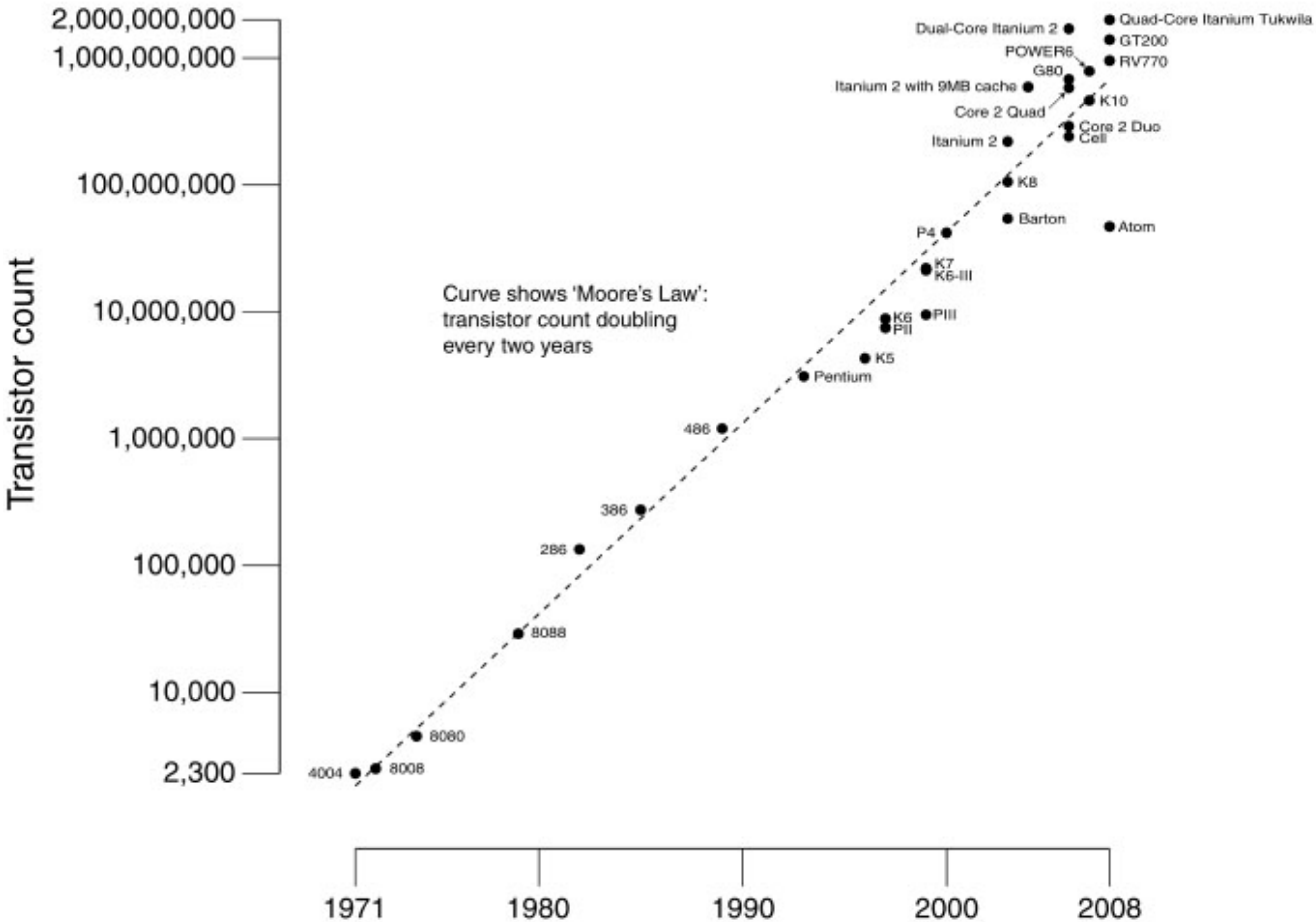
50 flight points
100 mass cases
10 configurations
5 maneuvers
20 gusts
4 control laws

20 million analyses

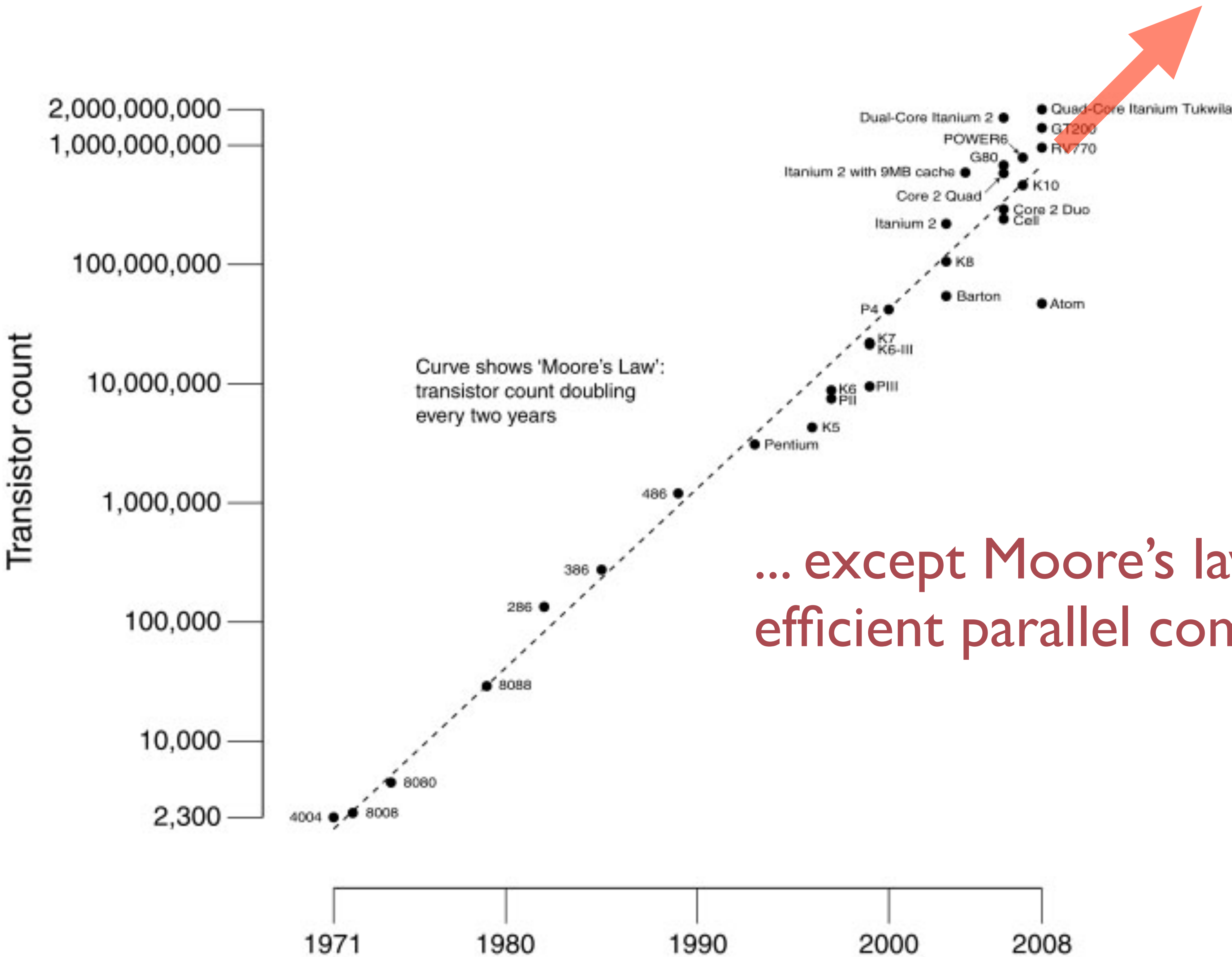
Use engineering experience from conventional designs

100,000 analyses

...but next generation computing will be much more powerful...

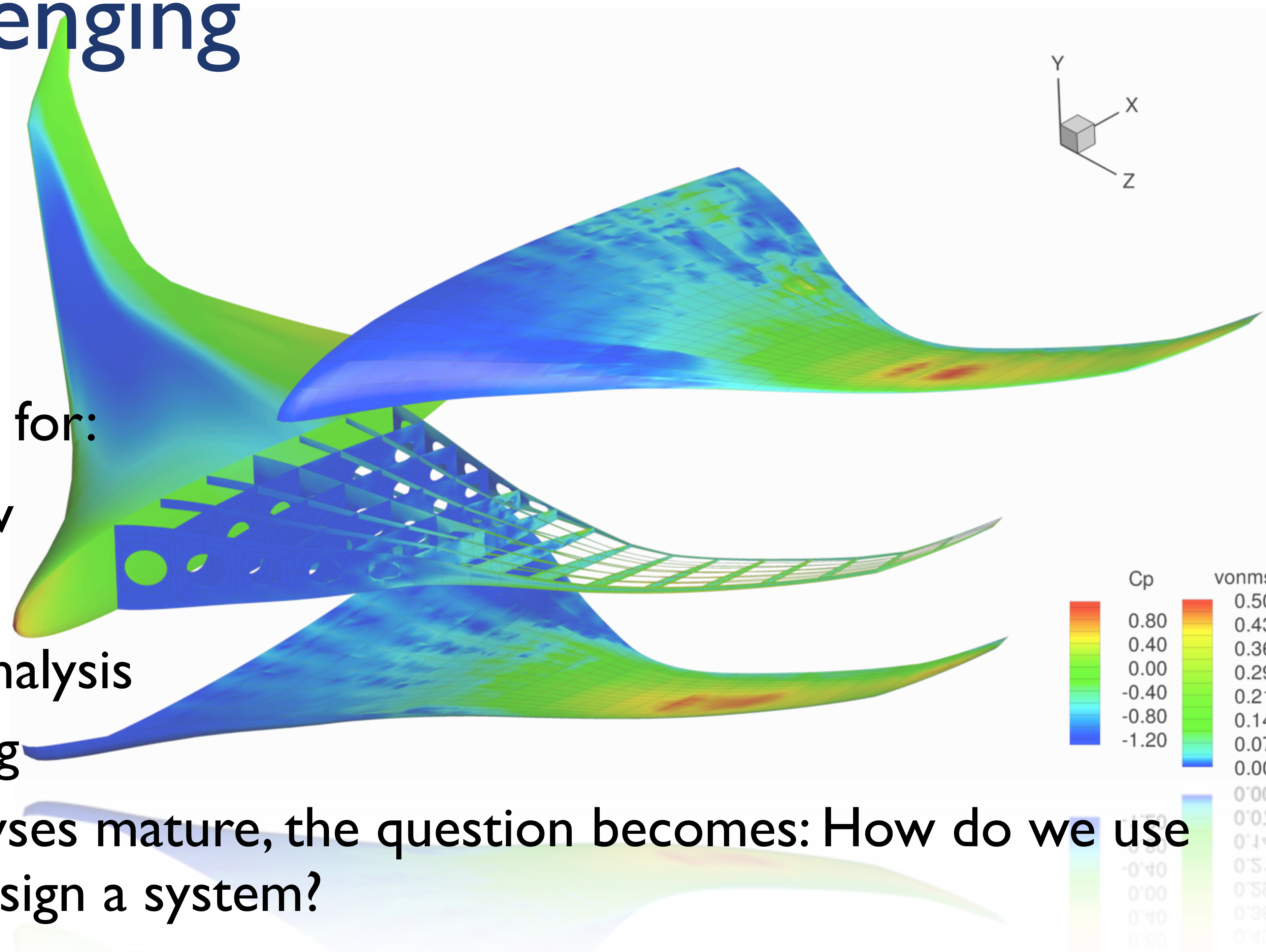


...but next generation computing will be much more powerful...



... except Moore's law requires efficient parallel computing

Why we need high-fidelity MDO, and why it is so challenging



- High-fidelity needed for:
  - Compressible flow
  - Viscous drag
  - Accurate failure analysis
  - Nonlinear coupling
- As high-fidelity analyses mature, the question becomes: How do we use these analyses to design a system?
- How do we utilize the full potential of a new technology?
- Large numbers of design variables and constraints required to take advantage of high-fidelity analyses

Some of the main challenges are:

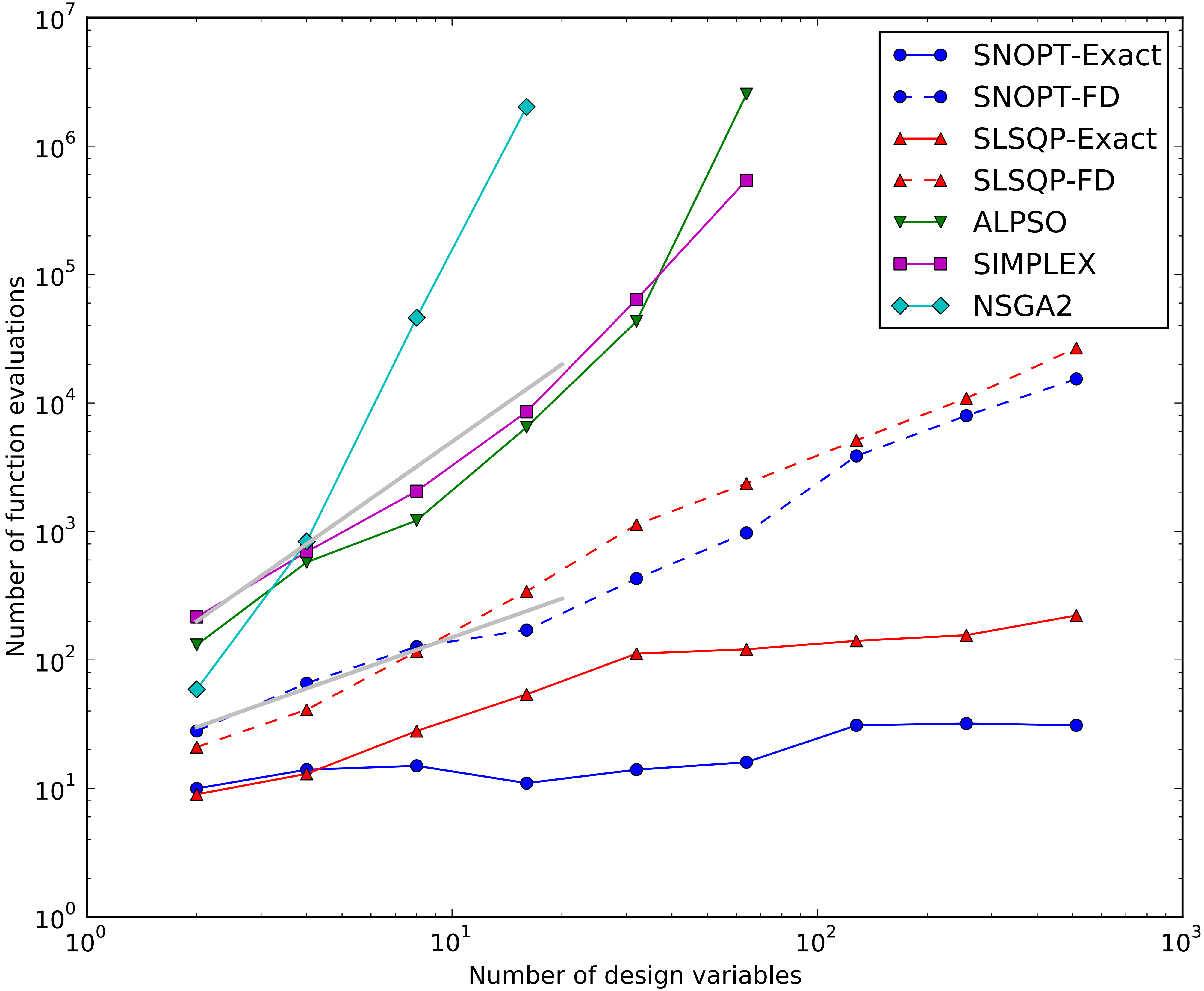
1. Multiple highly coupled systems

2. High computational cost of analysis

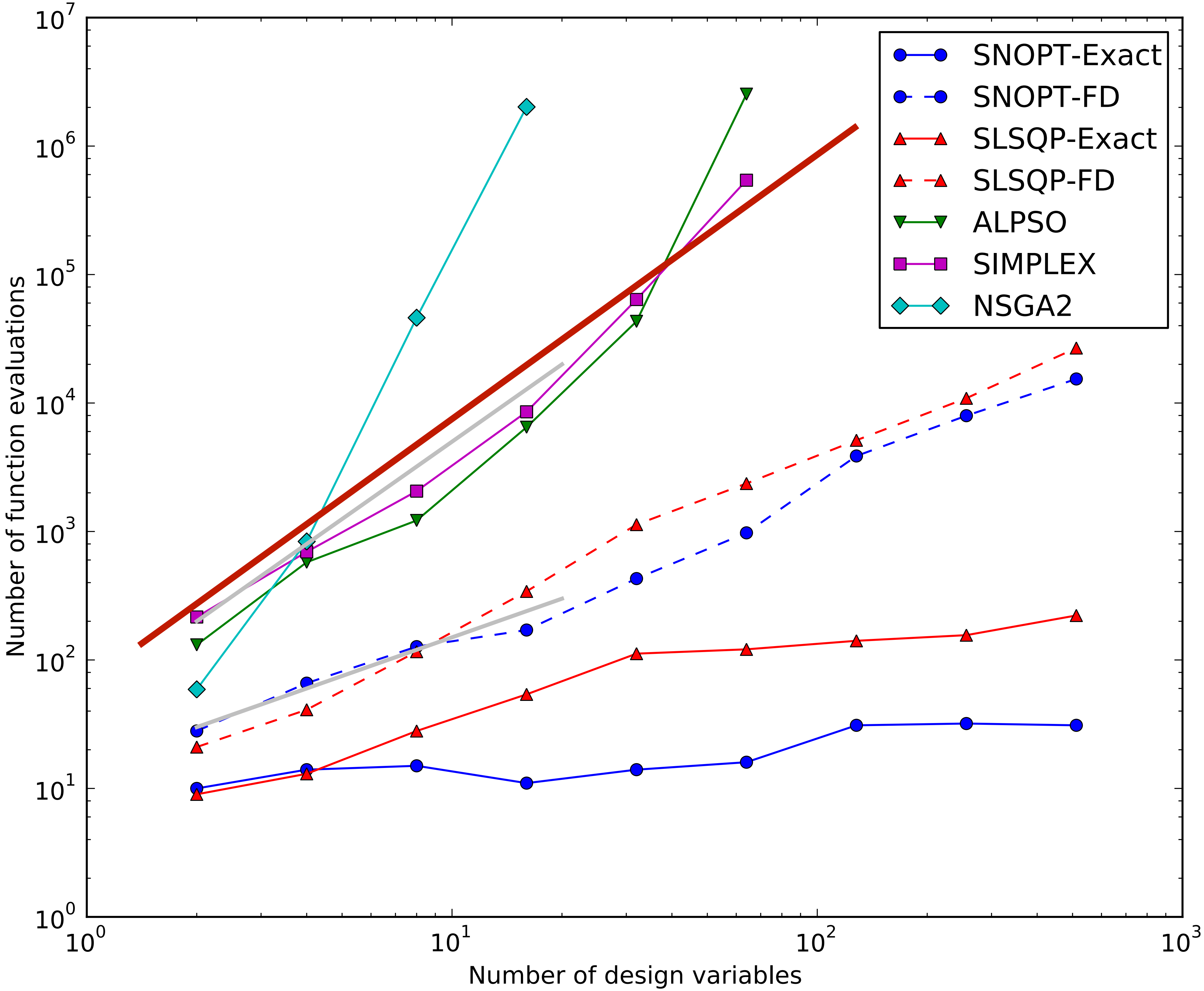
3. Large numbers of design variables and constraints

4. Relevant problem formulation

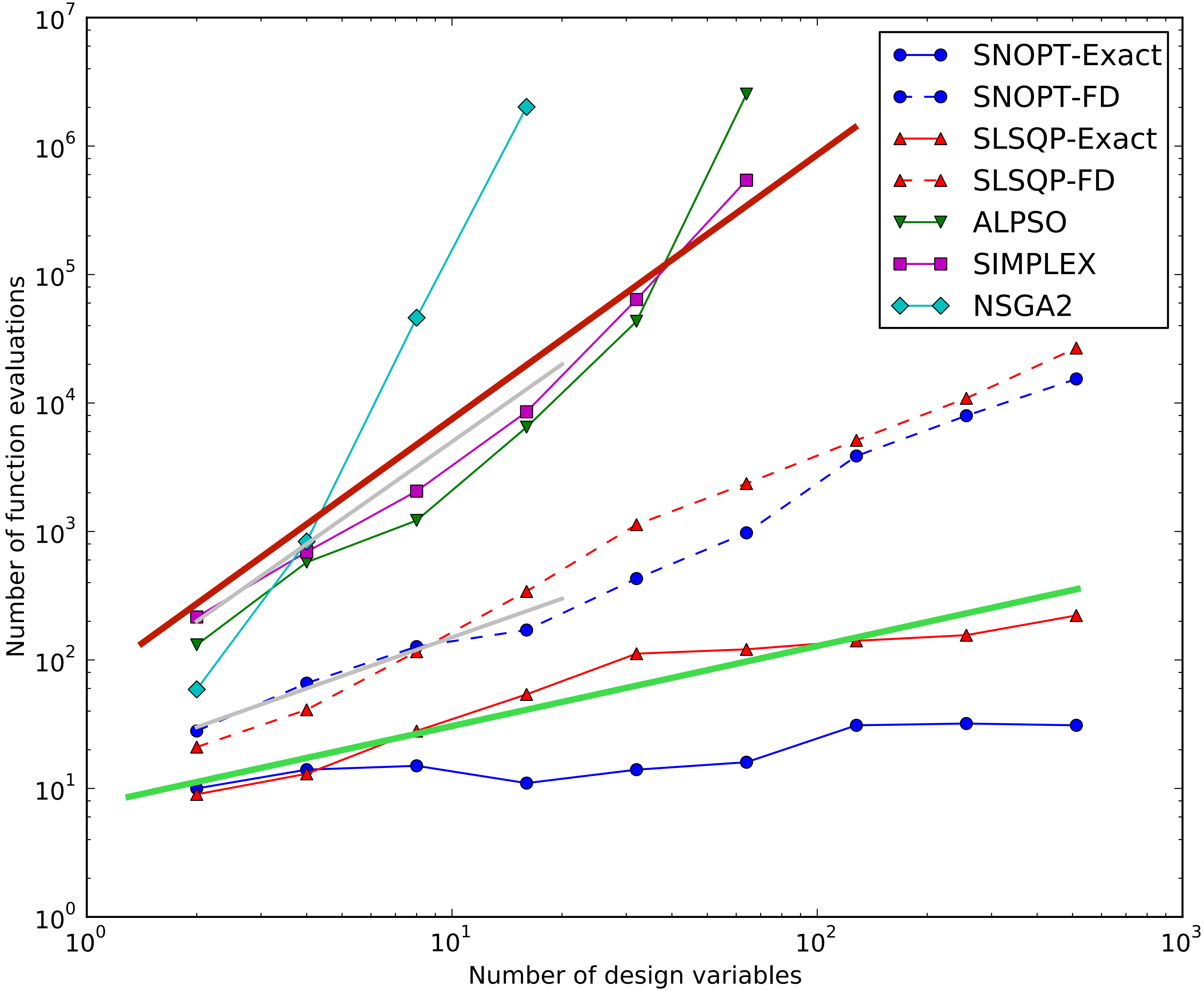
Gradient-based optimization is our only hope to explore large-dimensionality design spaces



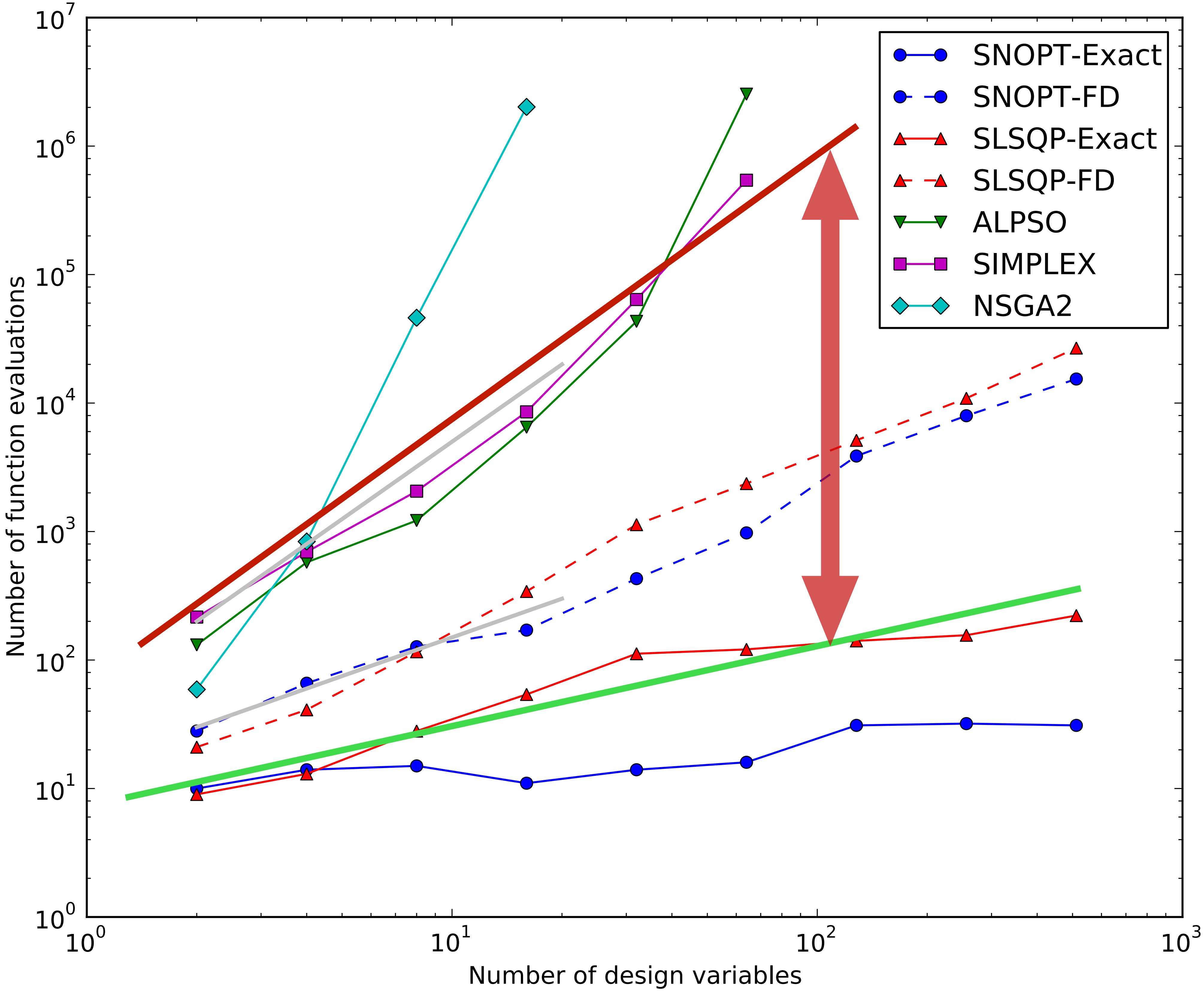
Gradient-based optimization is our only hope to explore large-dimensionality design spaces



Gradient-based optimization is our only hope to explore large-dimensionality design spaces

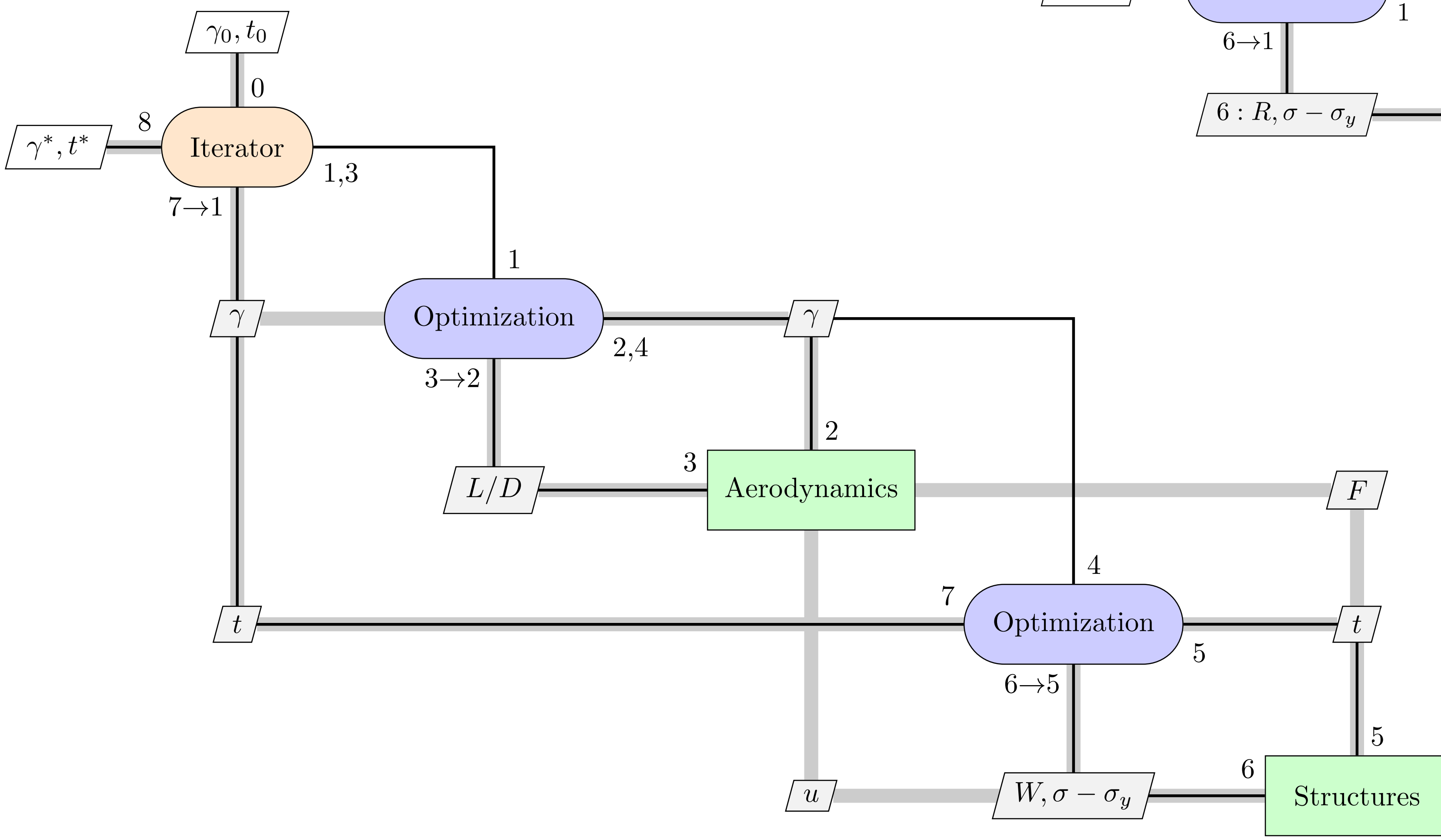


Gradient-based optimization is our only hope to explore large-dimensionality design spaces

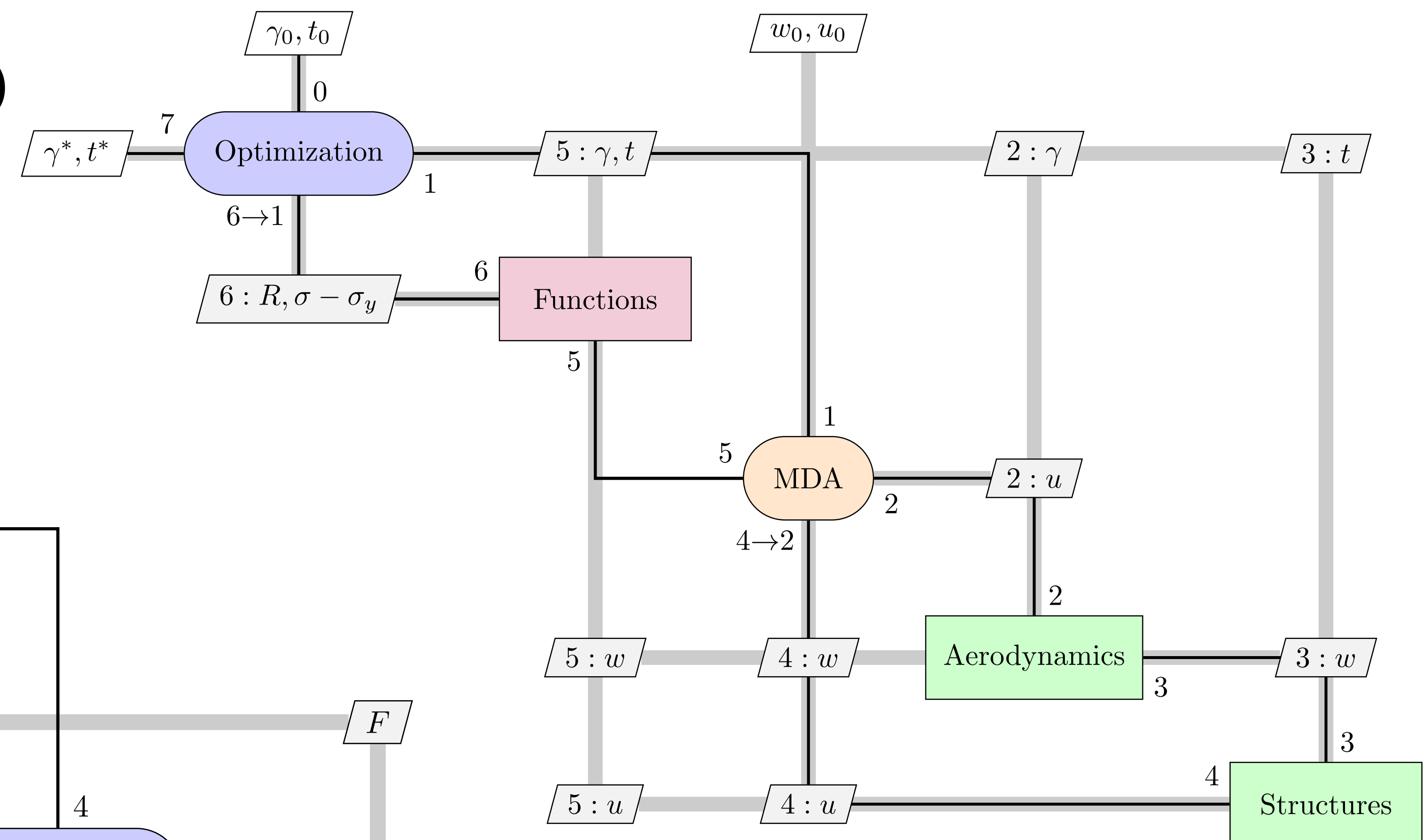


Why sequential optimization is not MDO: A wing design example

Sequential



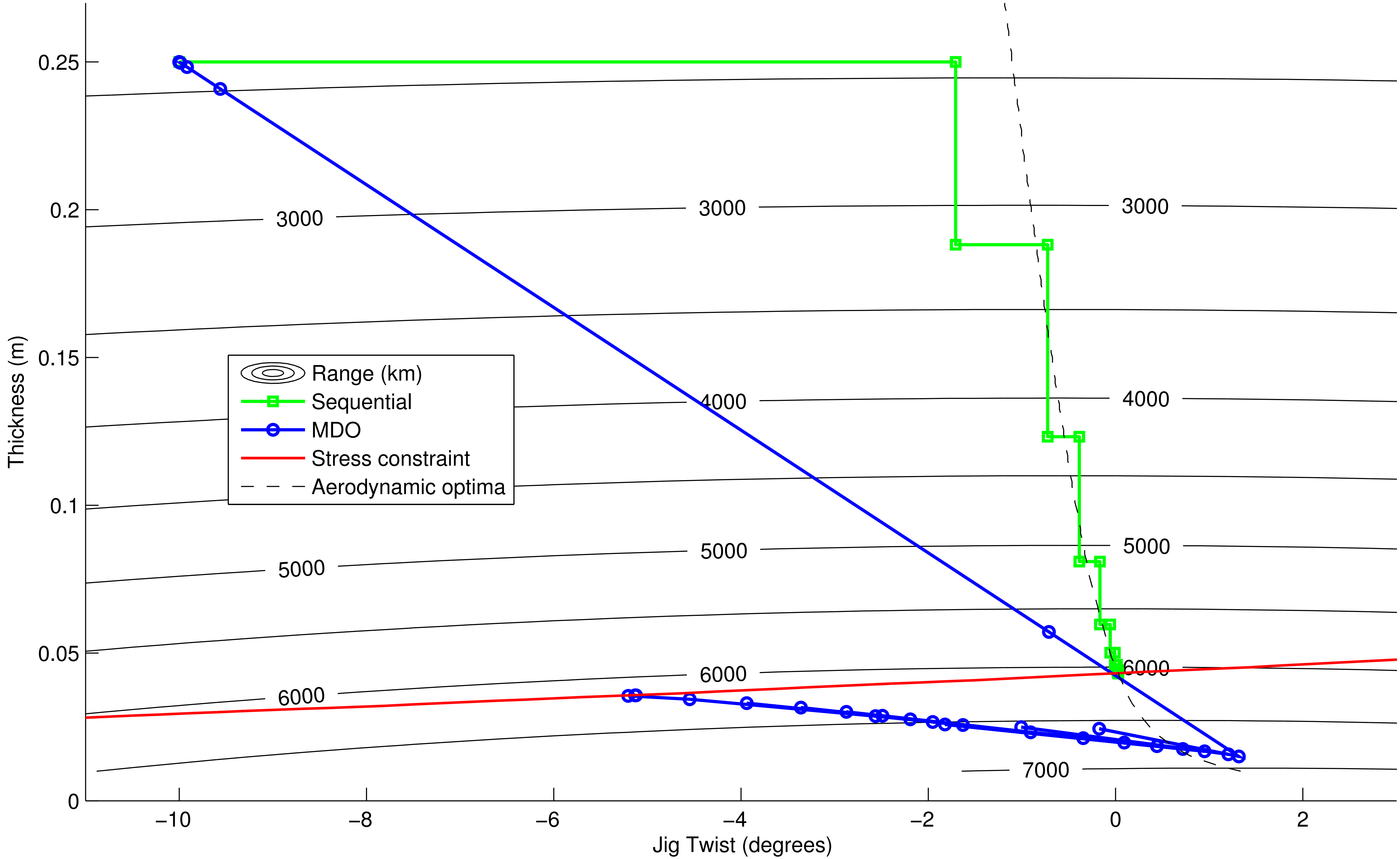
MDO



Aerodynamics: Panel code computes induced drag. Variables: wing twist and angle of attack

Structures: Beam finite-element model of the spar that computes the displacements and stresses. Variables: element thicknesses

Watch sequential optimization get stuck in a rut



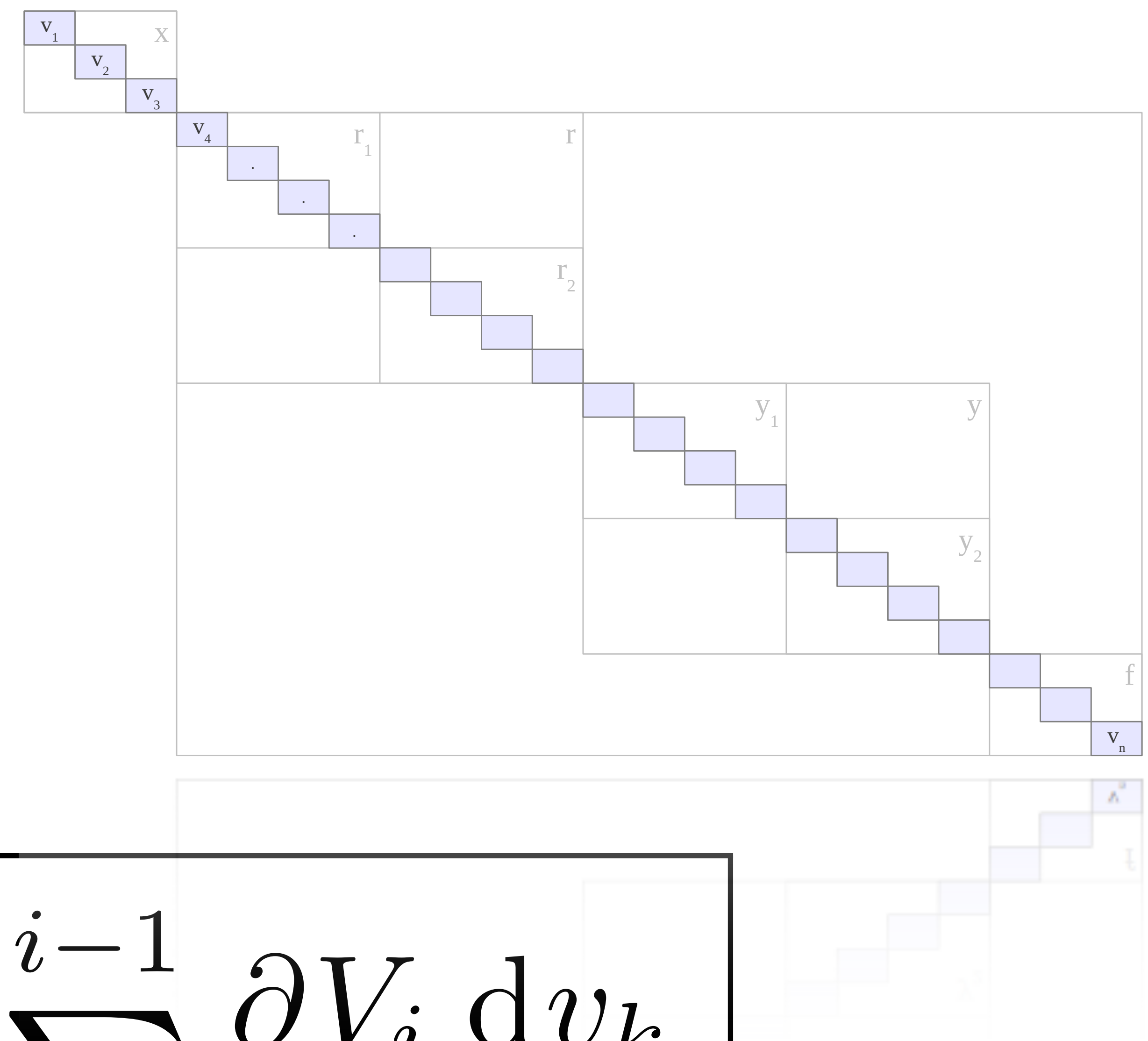
[Chittick and Martins, *Structural and Multidisciplinary Optimization*, 2008]

Computing derivatives: a short review

One Chain to Rule Them All

The total variation of a variable with respect to another is

$$\Delta v_k = \sum_{l=j}^{k-1} \frac{\partial V_k}{\partial v_l} \Delta v_l$$



We want the total derivative

$$\frac{\mathrm{d}v_i}{\mathrm{d}v_j} = \frac{\Delta v_i}{\Delta v_j}$$

This yields the chain rule

$$\boxed{\frac{\mathrm{d}v_i}{\mathrm{d}v_j} = \delta_{ij} + \sum_{k=j}^{i-1} \frac{\partial V_i}{\partial v_k} \frac{\mathrm{d}v_k}{\mathrm{d}v_j}}$$

Chain Rule in Matrix Form

Define the partial and total derivative matrices

$$\boldsymbol{D_V} = \frac{\partial V_i}{\partial v_j} = \begin{bmatrix} 0 & \cdots & & & \\ \frac{\partial V_2}{\partial v_1} & 0 & \cdots & & \\ \frac{\partial V_3}{\partial v_1} & \frac{\partial V_3}{\partial v_2} & 0 & \cdots & \\ \vdots & \vdots & \ddots & \ddots & \\ \frac{\partial V_n}{\partial v_1} & \frac{\partial V_n}{\partial v_2} & \cdots & \frac{\partial V_n}{\partial v_{n-1}} & 0 \end{bmatrix} \qquad \boldsymbol{D_v} = \frac{\mathrm{d} v_i}{\mathrm{d} v_j} = \begin{bmatrix} 1 & 0 & \cdots & & \\ \frac{\mathrm{d} v_2}{\mathrm{d} v_1} & 1 & 0 & \cdots & \\ \frac{\mathrm{d} v_3}{\mathrm{d} v_1} & \frac{\mathrm{d} v_3}{\partial v_2} & 1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ \frac{\mathrm{d} v_n}{\mathrm{d} v_1} & \frac{\mathrm{d} v_n}{\mathrm{d} v_2} & \cdots & \frac{\mathrm{d} v_n}{\mathrm{d} v_{n-1}} & 1 \end{bmatrix}$$

Use this notation to write the chain rule in matrix form

$$\frac{\mathrm{d} v_i}{\mathrm{d} v_j} = \delta_{ij} + \sum_{k=j}^{i-1} \frac{\partial V_i}{\partial v_k} \frac{\mathrm{d} v_k}{\mathrm{d} v_j} \quad \Rightarrow \quad \boldsymbol{D_v} = \boldsymbol{I} + \boldsymbol{D_V} \boldsymbol{D_v}$$

Yielding the linear system

$$\underbrace{(\boldsymbol{I} - \boldsymbol{D_V})}_{n \times n} \underbrace{\boldsymbol{D_v}}_{n \times n} = \underbrace{\boldsymbol{I}}_{n \times n}$$

The Chain Rule in Reverse

The two matrices are each other's inverses, so

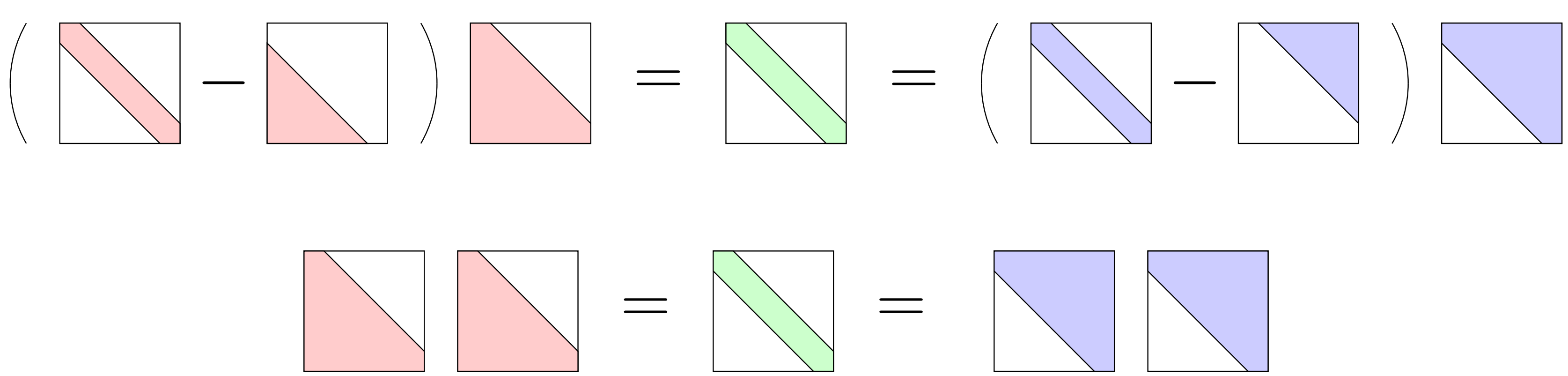
$$(\boldsymbol{I} - \boldsymbol{D_V})\,\boldsymbol{D_v} = \boldsymbol{I} \Rightarrow \quad \boldsymbol{D_v} = (\boldsymbol{I} - \boldsymbol{D_V})^{-1} \Rightarrow$$

$$\boldsymbol{D_v}^T = (\boldsymbol{I} - \boldsymbol{D_V})^{-T} \Rightarrow \quad (\boldsymbol{I} - \boldsymbol{D_V})^T\,\boldsymbol{D_v}^T = \boldsymbol{I}$$

And we get the reverse form of the chain rule

Both forward and reverse modes of the chain rule yield the identity

$$\boxed{(\boldsymbol{I} - \boldsymbol{D_V})\,\boldsymbol{D_v} = \quad \boldsymbol{I} \quad = (\boldsymbol{I} - \boldsymbol{D_V})^T\,\boldsymbol{D_v}^T}$$



Forward and Reverse Chain Rule

$$\left(\boldsymbol{I}-\boldsymbol{D}_{\boldsymbol{V}}\right)\boldsymbol{D}_{\boldsymbol{v}}=\quad\boldsymbol{I}\quad=\left(\boldsymbol{I}-\boldsymbol{D}_{\boldsymbol{V}}\right)^{T}\boldsymbol{D}_{\boldsymbol{v}}{}^{T}$$

$$\begin{bmatrix} 1 & 0 & \cdots & & \\ -\frac{\partial V_2}{\partial v_1} & 1 & 0 & \cdots & \\ -\frac{\partial V_3}{\partial v_1} & -\frac{\partial V_3}{\partial v_2} & 1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ -\frac{\partial V_n}{\partial v_1} & -\frac{\partial V_n}{\partial v_2} & \cdots & -\frac{\partial V_n}{\partial v_{n-1}} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & & \\ \frac{\mathrm{d}v_2}{\mathrm{d}v_1} & 1 & 0 & \cdots & \\ \frac{\mathrm{d}v_3}{\mathrm{d}v_1} & \frac{\mathrm{d}v_3}{\partial v_2} & 1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ \frac{\mathrm{d}v_n}{\mathrm{d}v_1} & \frac{\mathrm{d}v_n}{\mathrm{d}v_2} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}v_{n-1}} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & & \\ 0 & 1 & 0 & \cdots & \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -\frac{\partial V_2}{\partial v_1} & -\frac{\partial V_3}{\partial v_1} & \cdots & -\frac{\partial V_n}{\partial v_1} \\ 0 & 1 & -\frac{\partial V_3}{\partial v_2} & \cdots & -\frac{\partial V_n}{\partial v_2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & -\frac{\partial V_n}{\partial v_{n-1}} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\mathrm{d}v_2}{\mathrm{d}v_1} & \frac{\mathrm{d}v_3}{\mathrm{d}v_1} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}v_1} \\ 0 & 1 & \frac{\mathrm{d}v_3}{\mathrm{d}v_2} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}v_2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & \frac{\mathrm{d}v_n}{\mathrm{d}v_{n-1}} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & & \\ 0 & 1 & 0 & \cdots & \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward and Reverse Chain Rule

$$(\boldsymbol{I} - \boldsymbol{D_V})\,\boldsymbol{D_v} = \boldsymbol{I} = (\boldsymbol{I} - \boldsymbol{D_V})^T\,\boldsymbol{D_v}^T$$

$$\begin{bmatrix} 1 & 0 & \cdots & & \\ -\frac{\partial V_2}{\partial v_1} & 1 & 0 & \cdots & \\ -\frac{\partial V_3}{\partial v_1} & -\frac{\partial V_3}{\partial v_2} & 1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ -\frac{\partial V_n}{\partial v_1} & -\frac{\partial V_n}{\partial v_2} & \cdots & -\frac{\partial V_n}{\partial v_{n-1}} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & & \\ \frac{\mathrm{d}v_2}{\mathrm{d}v_1} & 1 & 0 & \cdots & \\ \frac{\mathrm{d}v_3}{\mathrm{d}v_1} & \frac{\mathrm{d}v_3}{\partial v_2} & 1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ \boxed{\frac{\mathrm{d}v_n}{\mathrm{d}v_1} \quad \frac{\mathrm{d}v_n}{\mathrm{d}v_2}} & & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}v_{n-1}} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & & \\ 0 & 1 & 0 & \cdots & \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -\frac{\partial V_2}{\partial v_1} & -\frac{\partial V_3}{\partial v_1} & \cdots & -\frac{\partial V_n}{\partial v_1} \\ 0 & 1 & -\frac{\partial V_3}{\partial v_2} & \cdots & -\frac{\partial V_n}{\partial v_2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & -\frac{\partial V_n}{\partial v_{n-1}} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\mathrm{d}v_2}{\mathrm{d}v_1} & \frac{\mathrm{d}v_3}{\mathrm{d}v_1} & \cdots & \boxed{\begin{matrix} \frac{\mathrm{d}v_n}{\mathrm{d}v_1} \\ \frac{\mathrm{d}v_n}{\mathrm{d}v_2} \end{matrix}} \\ 0 & 1 & \frac{\mathrm{d}v_3}{\mathrm{d}v_2} & \cdots & \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & \frac{\mathrm{d}v_n}{\mathrm{d}v_{n-1}} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & & \\ 0 & 1 & 0 & \cdots & \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

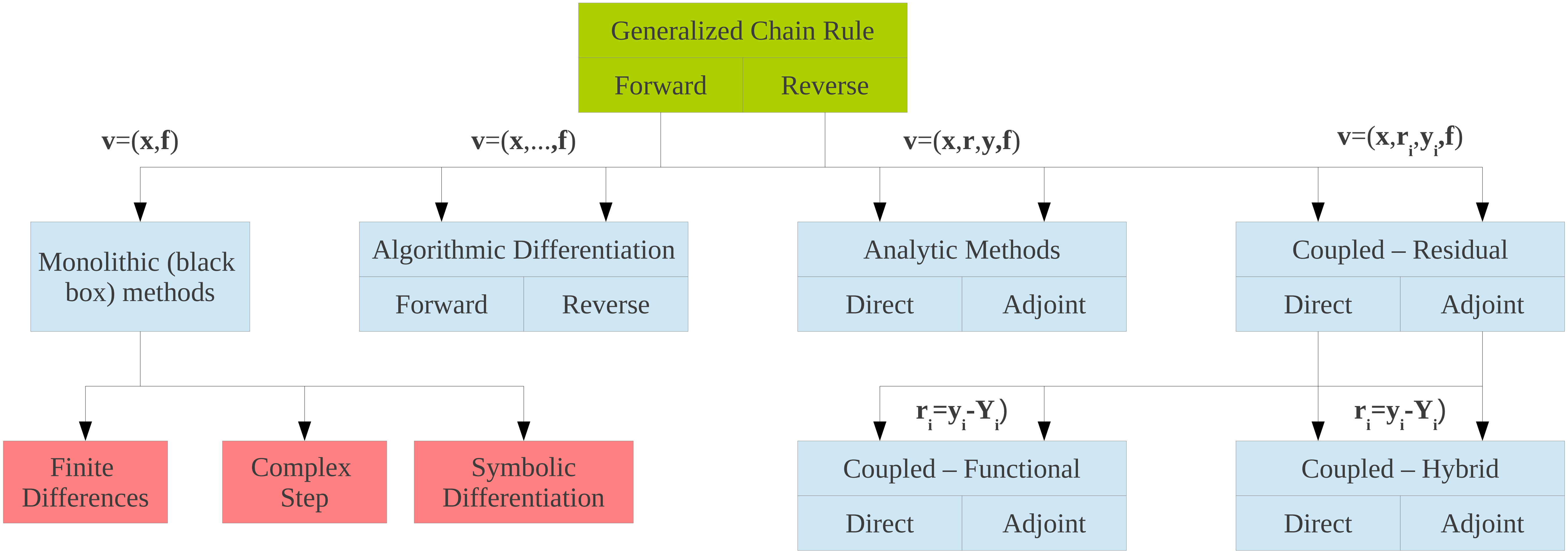
Forward and Reverse Chain Rule

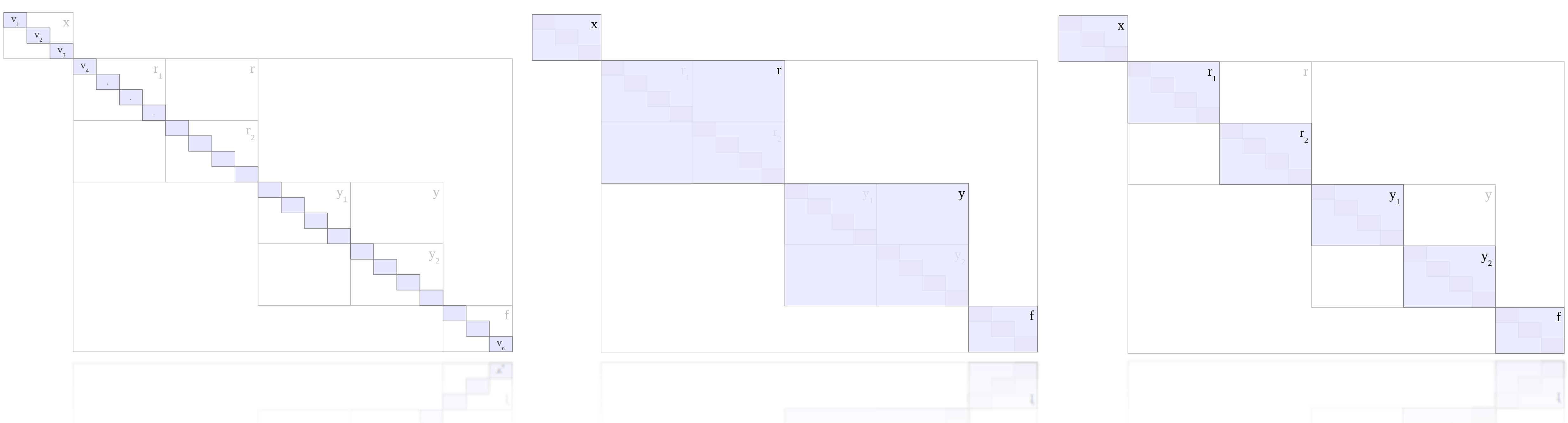
$$(\boldsymbol{I}-\boldsymbol{D_V})\,\boldsymbol{D_v} = \boldsymbol{I} = (\boldsymbol{I}-\boldsymbol{D_V})^T\,\boldsymbol{D_v}^T$$

$$\begin{bmatrix} 1 & 0 & \cdots & & \\ -\frac{\partial V_2}{\partial v_1} & 1 & 0 & \cdots & \\ -\frac{\partial V_3}{\partial v_1} & -\frac{\partial V_3}{\partial v_2} & 1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ -\frac{\partial V_n}{\partial v_1} & -\frac{\partial V_n}{\partial v_2} & \cdots & -\frac{\partial V_n}{\partial v_{n-1}} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & & \\ \frac{\mathrm{d}v_2}{\mathrm{d}v_1} & 1 & 0 & \cdots & \\ \frac{\mathrm{d}v_3}{\mathrm{d}v_1} & \frac{\mathrm{d}v_3}{\partial v_2} & 1 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ \frac{\mathrm{d}v_n}{\mathrm{d}v_1} & \frac{\mathrm{d}v_n}{\mathrm{d}v_2} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}v_{n-1}} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & & \\ 0 & 1 & 0 & \cdots & \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

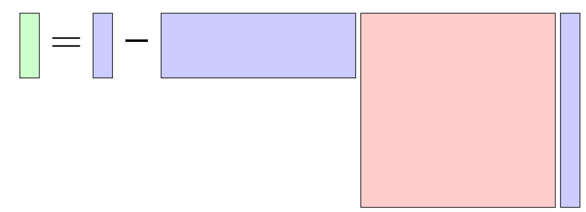
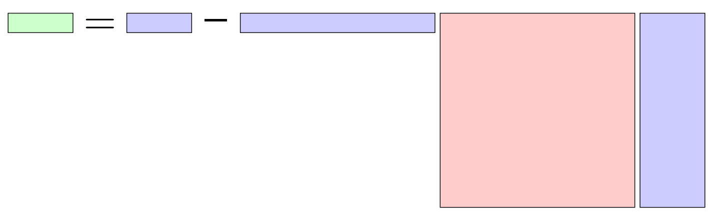
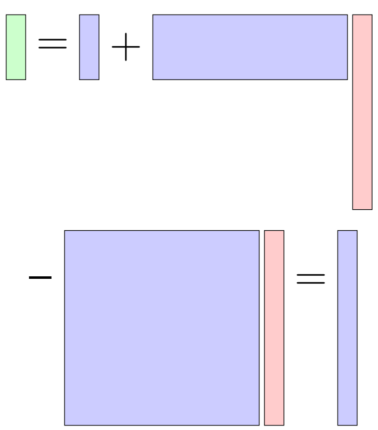
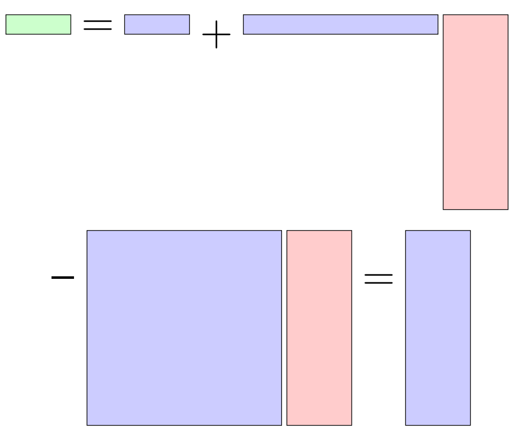
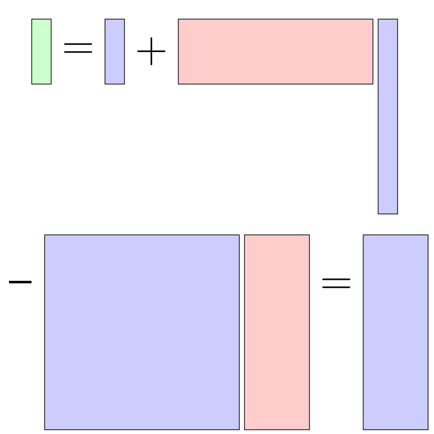
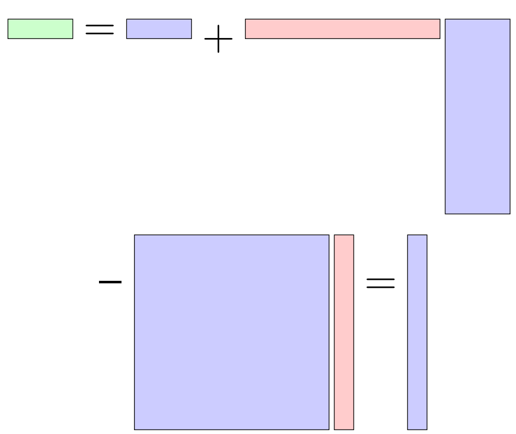
$$\begin{bmatrix} 1 & -\frac{\partial V_2}{\partial v_1} & -\frac{\partial V_3}{\partial v_1} & \cdots & -\frac{\partial V_n}{\partial v_1} \\ 0 & 1 & -\frac{\partial V_3}{\partial v_2} & \cdots & -\frac{\partial V_n}{\partial v_2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & -\frac{\partial V_n}{\partial v_{n-1}} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\mathrm{d}v_2}{\mathrm{d}v_1} & \frac{\mathrm{d}v_3}{\mathrm{d}v_1} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}v_1} \\ 0 & 1 & \frac{\mathrm{d}v_3}{\mathrm{d}v_2} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}v_2} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & \frac{\mathrm{d}v_n}{\mathrm{d}v_{n-1}} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & & \\ 0 & 1 & 0 & \cdots & \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Methods for Computing Derivatives





Direct vs. Adjoint Methods

	$n_f > n_x$	$n_x > n_f$
$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} - \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}} \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}}\right]^{-1} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}$		
Direct method	$n_f > n_x$	$n_x > n_f$
$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}} \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}}$ $-\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}} \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}$		
Adjoint method	$n_f > n_x$	$n_x > n_f$
$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} + \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}$ $-\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}}\right]^T \left[\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}}\right]^T = \left[\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}}\right]^T$		

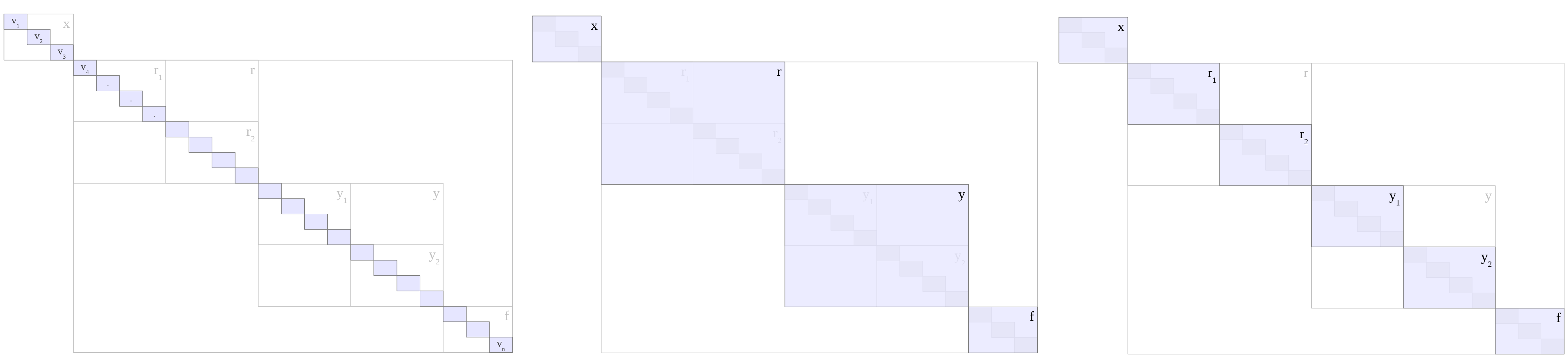
In a nutshell...

- Algorithmic differentiation (forward and reverse) and analytic methods (direct and reverse) can be derived from:

$$(I - D_V)\, D_v = \quad I \quad = (I - D_V)^T\, {D_v}^T$$

- It is all about defining the variables involved to the right level of decomposition

- More details in the paper

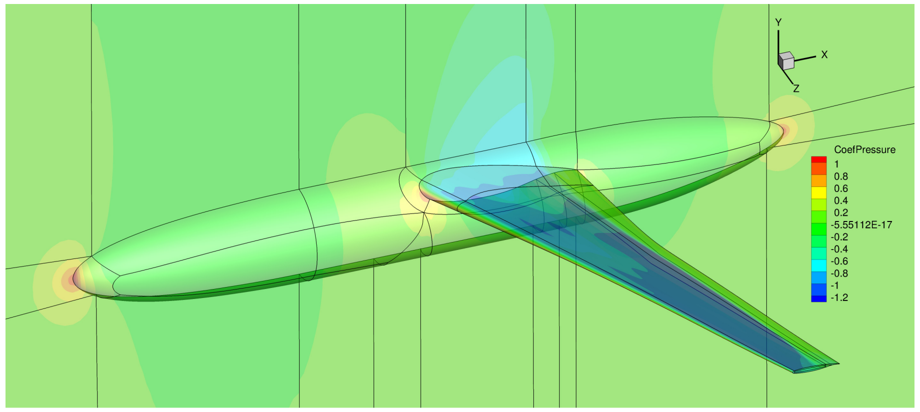


[Martins and Hwang, *AIAA SDM*, 2012]

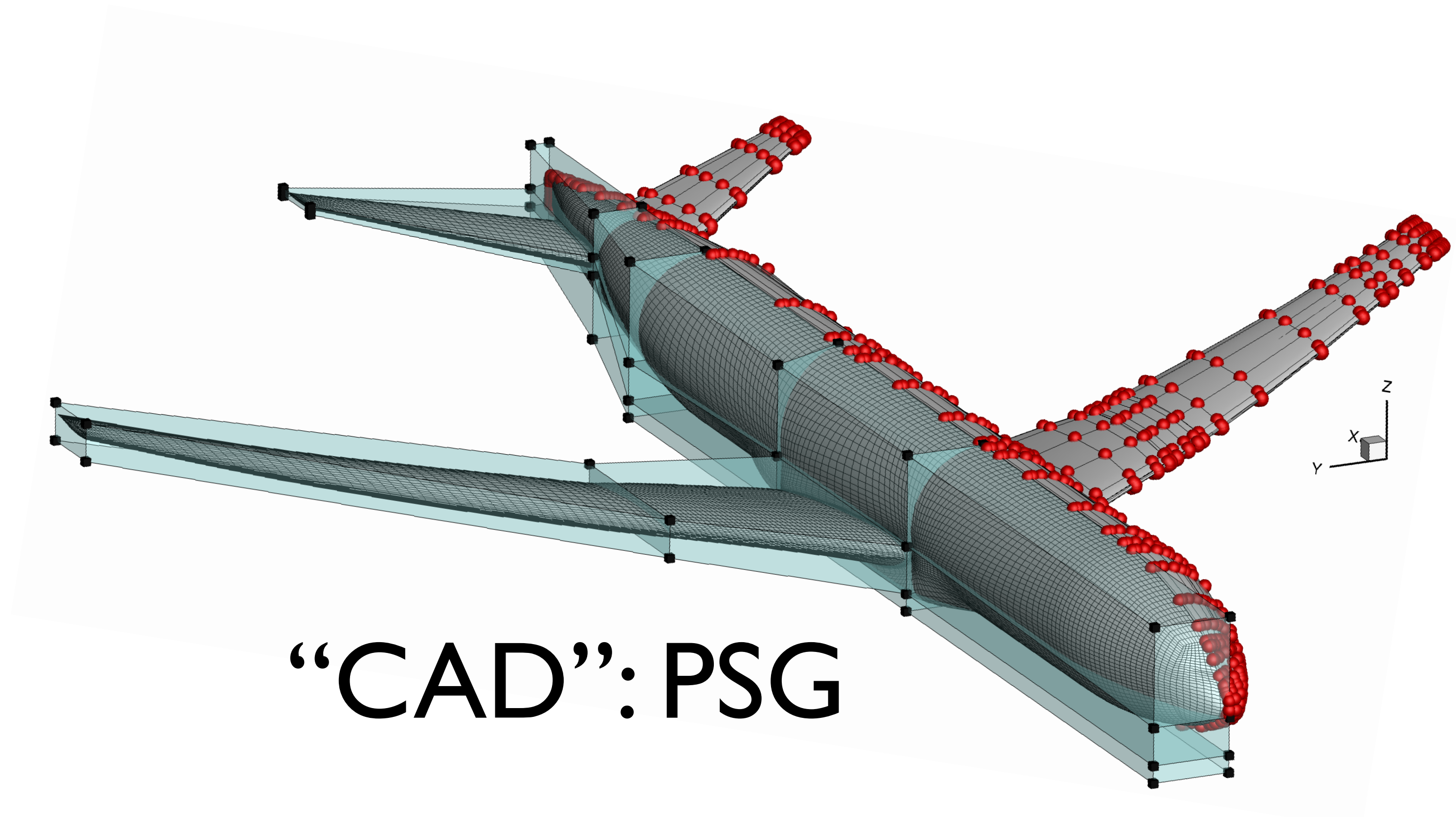
fantastic

What tools do we have for high-fidelity aerostructural analysis and optimization?

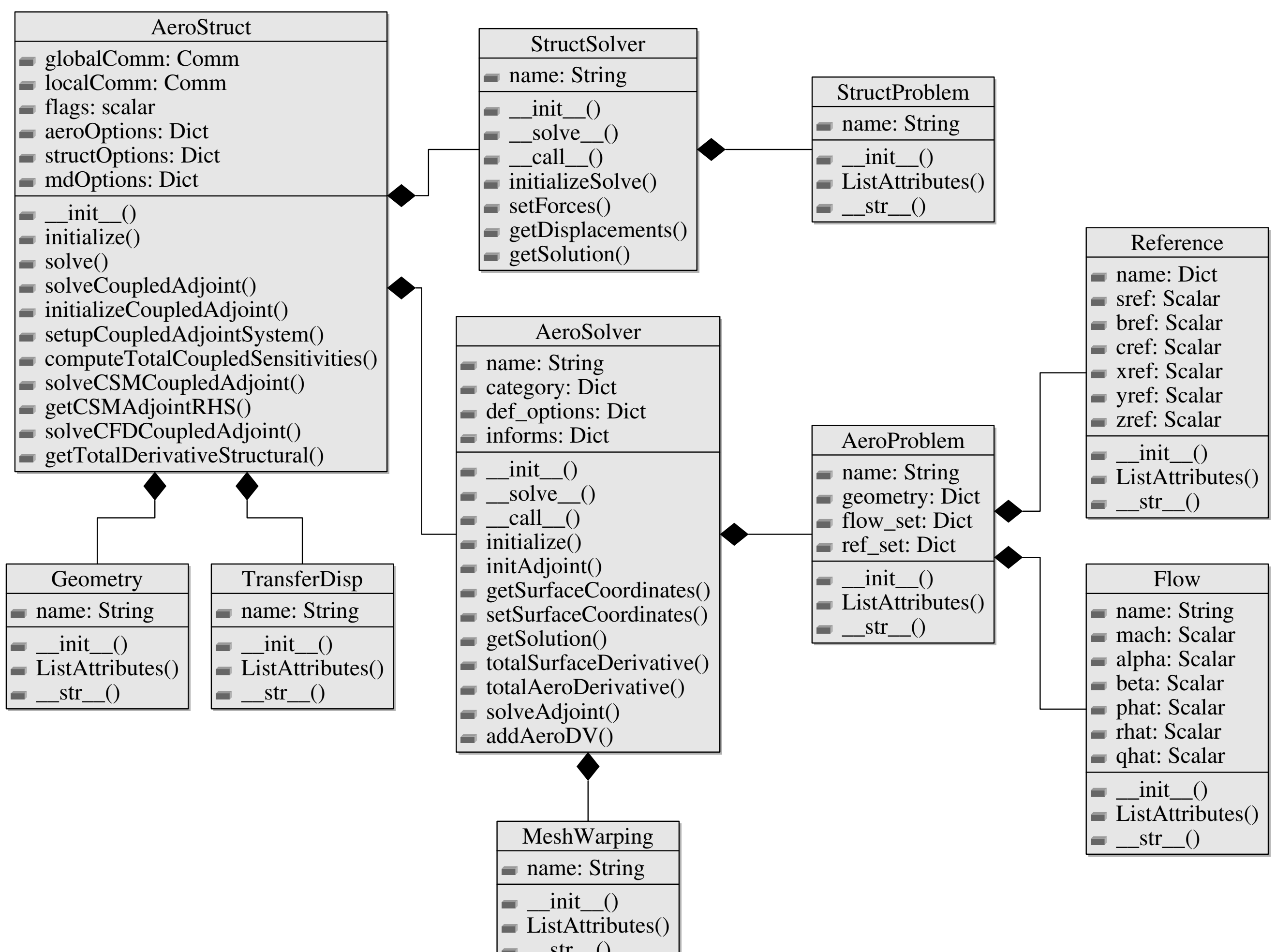
MDO for Aircraft Configurations with High-fidelity (MACH)



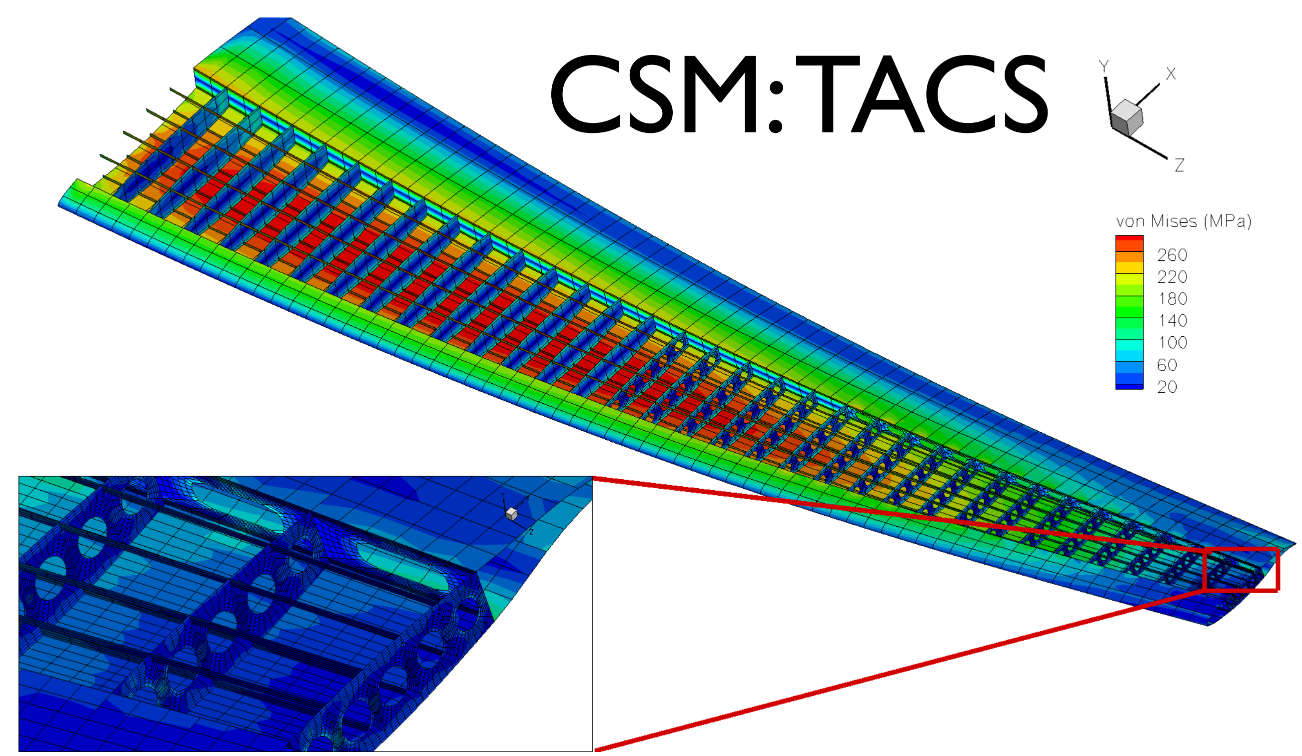
CFD: SUmb



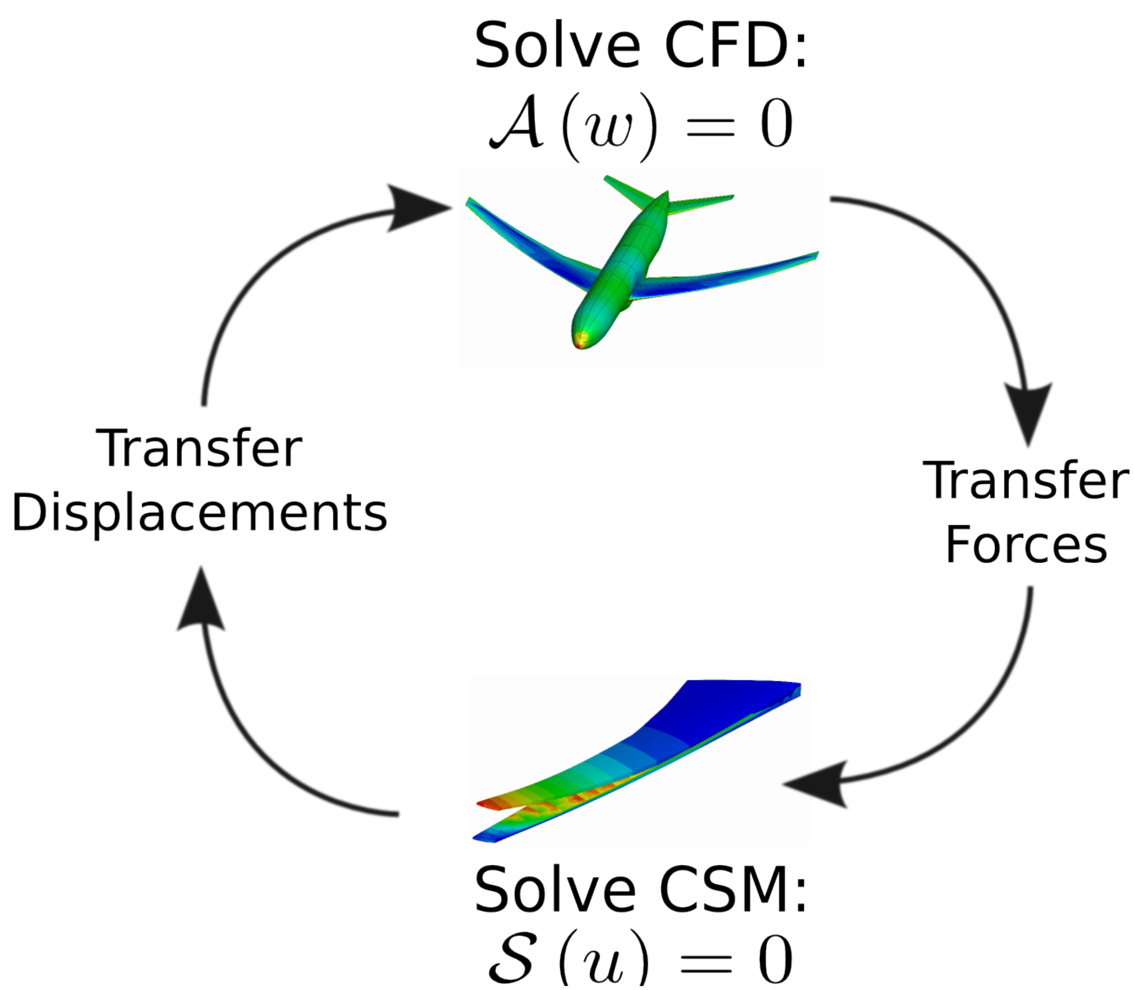
“CAD”: PSG



CSM: TACS



Fully coupled aerostructural analysis



$\mathcal{A}$: Aerodynamic residuals
w: Aerodynamic states
$\mathcal{S}$: Structural residuals
u: Structural states

Two available methods:

- A nonlinear block Gauss–Seidel method with Aitken acceleration
- A coupled Newton–Krylov method

$$\begin{bmatrix} \frac{\partial \mathcal{A}}{\partial w} & \frac{\partial \mathcal{A}}{\partial u} \\ \frac{\partial \mathcal{S}}{\partial w} & \frac{\partial \mathcal{S}}{\partial u} \end{bmatrix} \begin{bmatrix} \Delta w \\ \Delta u \end{bmatrix} = - \begin{bmatrix} \mathcal{A}(w) \\ \mathcal{S}(u) \end{bmatrix}$$

The coupled adjoint is the reason we require the source code for each component

Adjoint equations for the aerostructural system

$$\begin{bmatrix} \frac{\partial \boldsymbol{A}}{\partial \boldsymbol{w}} & \frac{\partial \boldsymbol{A}}{\partial \boldsymbol{u}} \\ \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{w}} & \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{u}} \end{bmatrix}^T \begin{bmatrix} \psi \\ \phi \end{bmatrix} = - \begin{bmatrix} \frac{\partial f}{\partial \boldsymbol{w}} \\ \frac{\partial f}{\partial \boldsymbol{u}} \end{bmatrix}$$

Total derivatives

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\partial f}{\partial \boldsymbol{x}} + \psi \frac{\partial \boldsymbol{A}}{\partial x} + \phi \frac{\partial \boldsymbol{S}}{\partial x}$$

[Martins et al., *Optimization and Engineering*, 2005]

The coupled adjoint is the reason we require the source code for each component

Adjoint equations for the aerostructural system

$$\begin{bmatrix} \frac{\partial \boldsymbol{A}}{\partial \boldsymbol{w}} & \boxed{\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{u}}} \\ \boxed{\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{w}}} & \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{u}} \end{bmatrix}^T \begin{bmatrix} \psi \\ \phi \end{bmatrix} = - \begin{bmatrix} \frac{\partial f}{\partial \boldsymbol{w}} \\ \frac{\partial f}{\partial \boldsymbol{u}} \end{bmatrix}$$

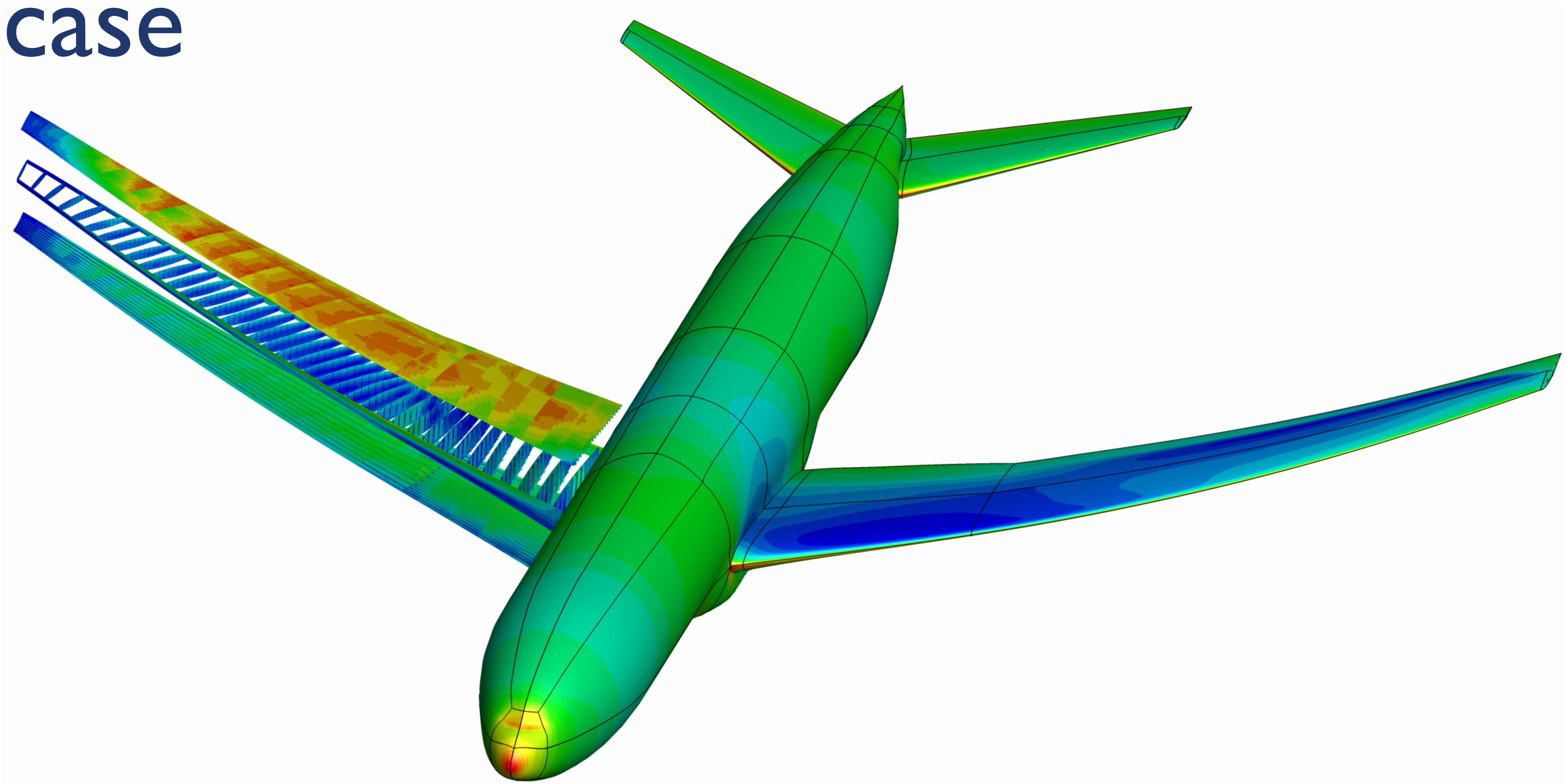
Total derivatives

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\partial f}{\partial \boldsymbol{x}} + \psi \frac{\partial \boldsymbol{A}}{\partial x} + \phi \frac{\partial \boldsymbol{S}}{\partial x}$$

[Martins et al., *Optimization and Engineering*, 2005]

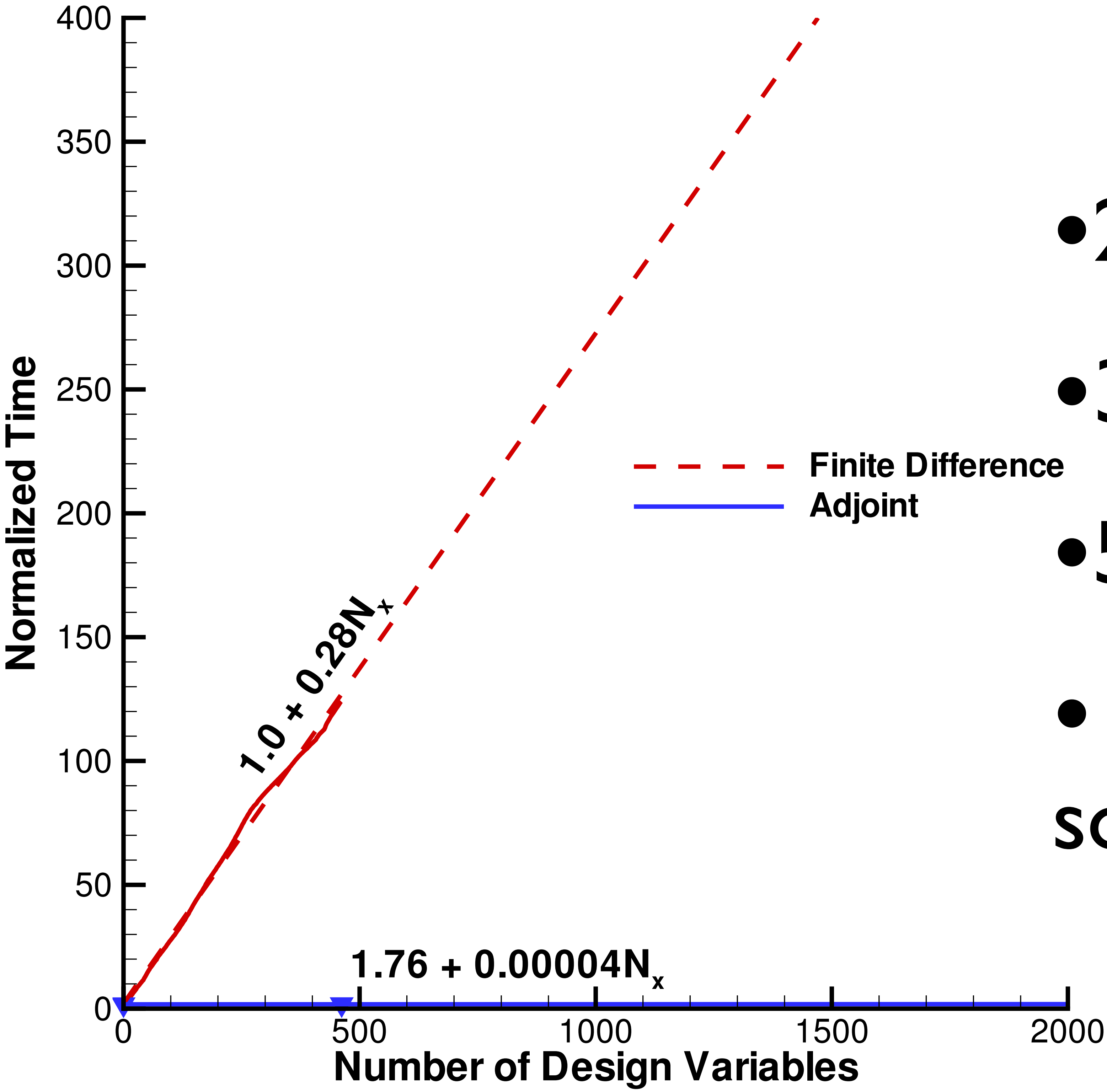
Let’s optimize a wing!

Chose the CRM geometry as a first case



- Common Research Model (CRM) from DPW4
- 2 million cells in CFD mesh
- Includes a structural model with 300 thousand DOFs

The coupled adjoint is the key for correct and efficient gradients



- 2M CFD cells
- 300k CSM DOFs
- 56 processors
- 1 aerostructural solution = 5.5 min

[Kenway, Kennedy and Martins, *AIAA Journal*, 2012 (forthcoming)]

The baseline aircraft is similar to a 777-200ER

Design and Maneuver Conditions

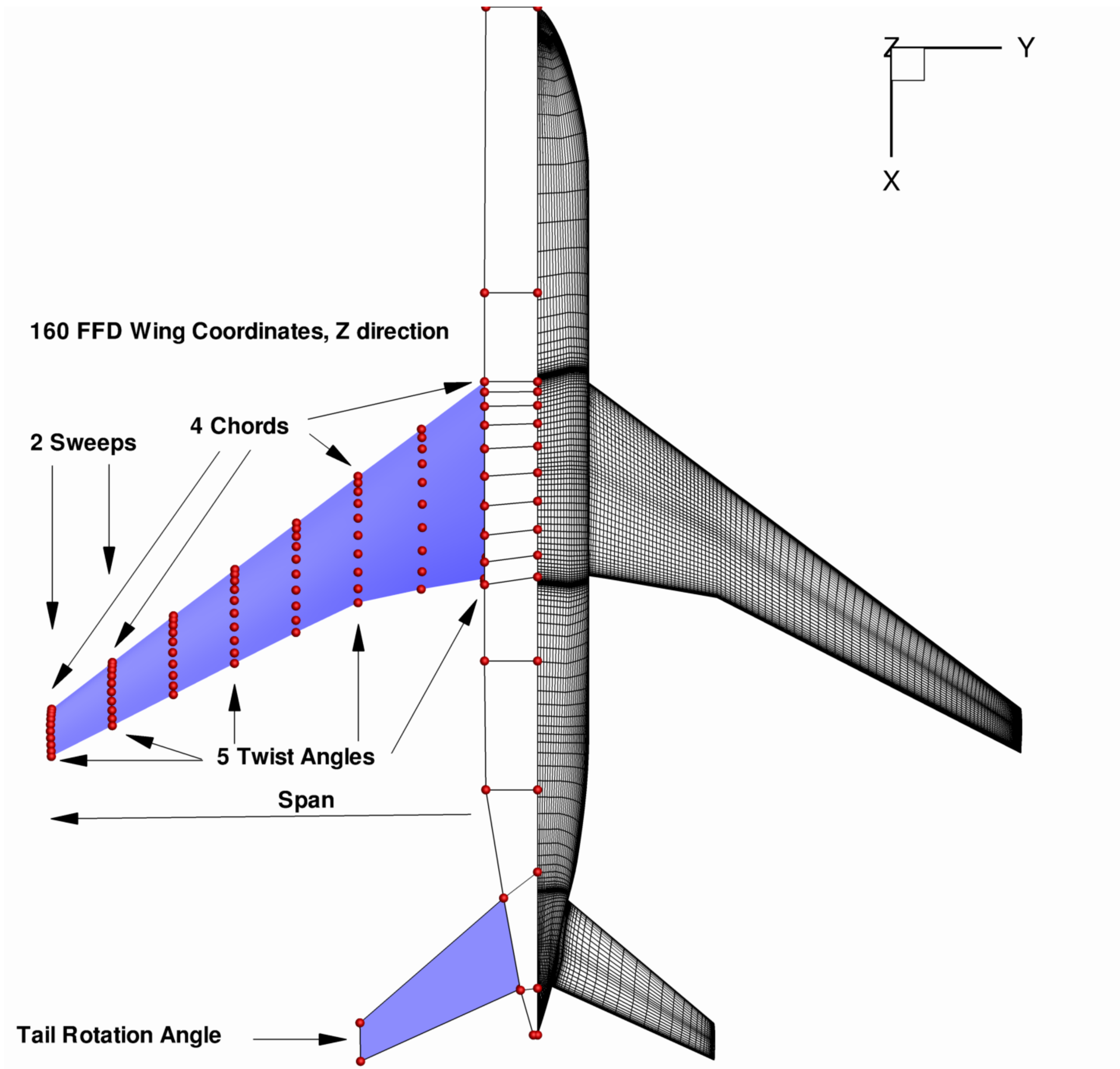
- Multi-point optimization considered a necessity in transonic flow with sufficient design freedom

Group	Identifier	Mach	Altitude, (ft)	Load Factor
Cruise	1	0.85	35 000	1.0
	2	0.84	35 000	1.0
	3	0.86	35 000	1.0
	4	0.85	34 000	1.0
	5	0.85	36 000	1.0
Maneuver	1	0.86	20 000	2.5
	2	0.85	32 000	1.3
Stability	1	0.85	35 000	1.0

- Static margin estimate requires an additional flow analysis to estimate derivatives C_{M_α} and C_{L_α}

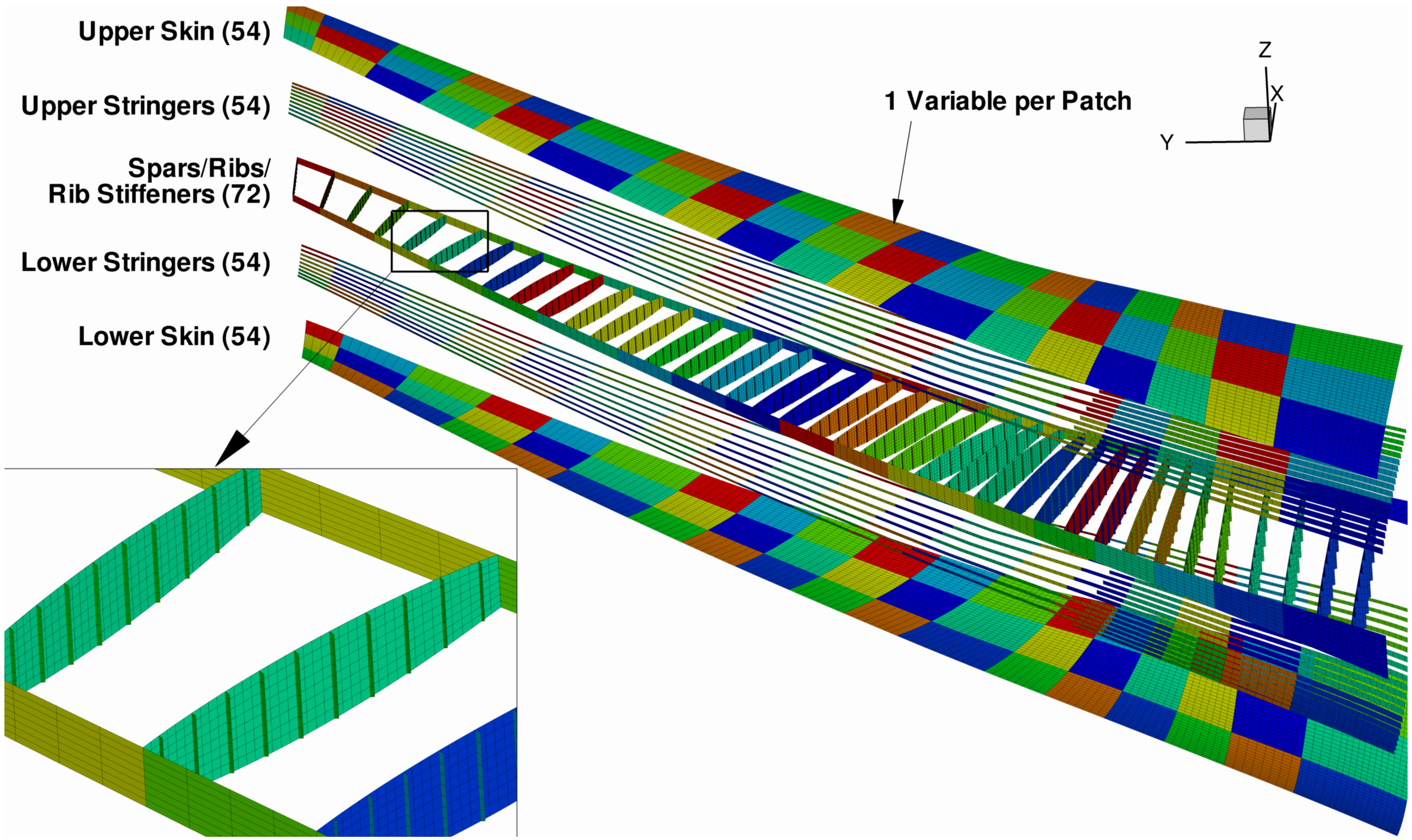
$$K_n = -\frac{C_{M_\alpha}}{C_{L_\alpha}}.$$

“Aerodynamic” shape variables also affect the structure directly



- 12 global geometric design variables
- 160 local shape design variables
- 2.1 million cell CFD mesh
- 1 angle of attack and 1 tail rotation angle for each operating condition

Structural sizing patchwork



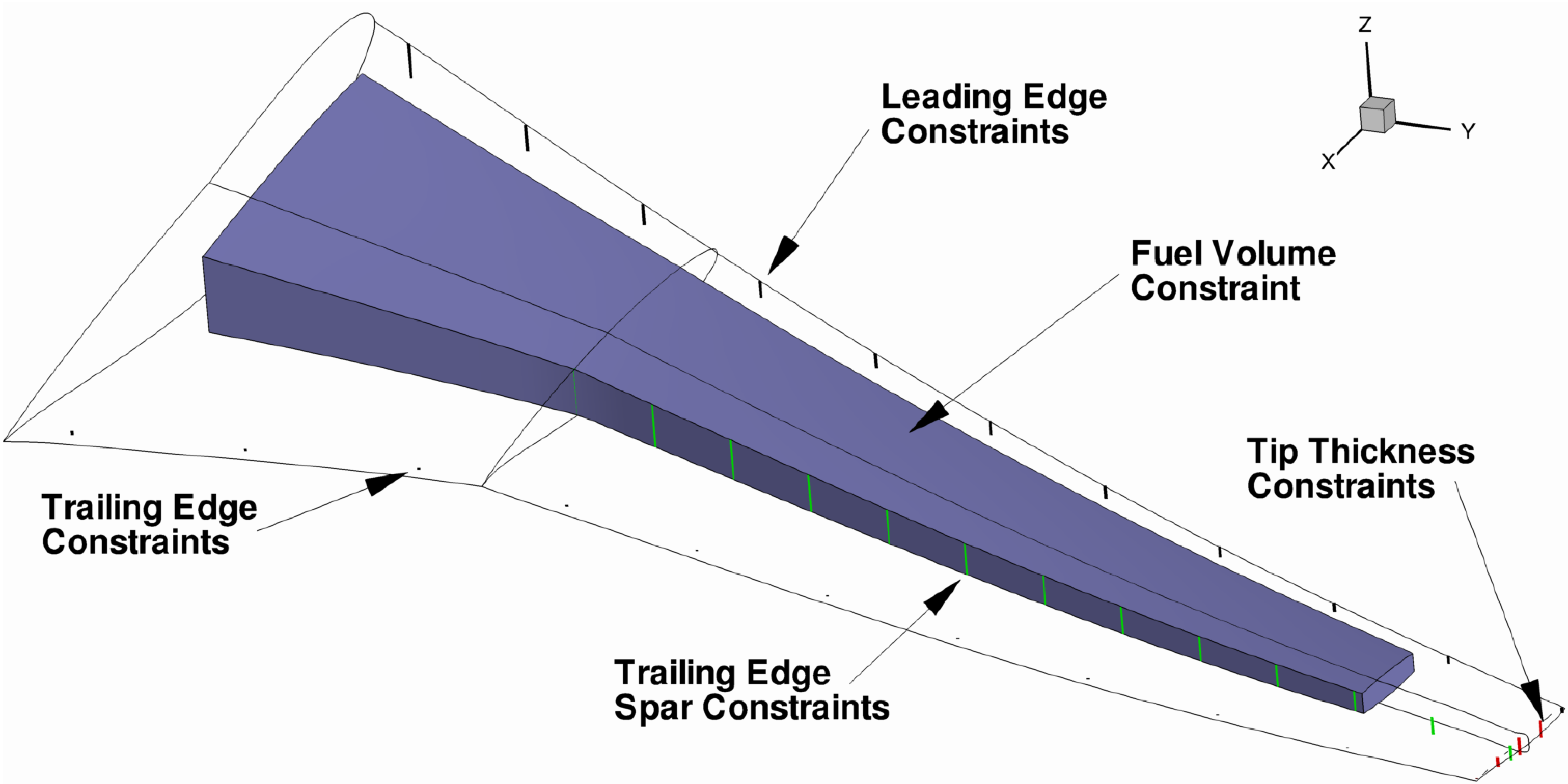
- 288 thickness design variables
- 300 000 structural degrees of freedom
- 476 total design variables

Need these constraints to make it realistic (and probably more)

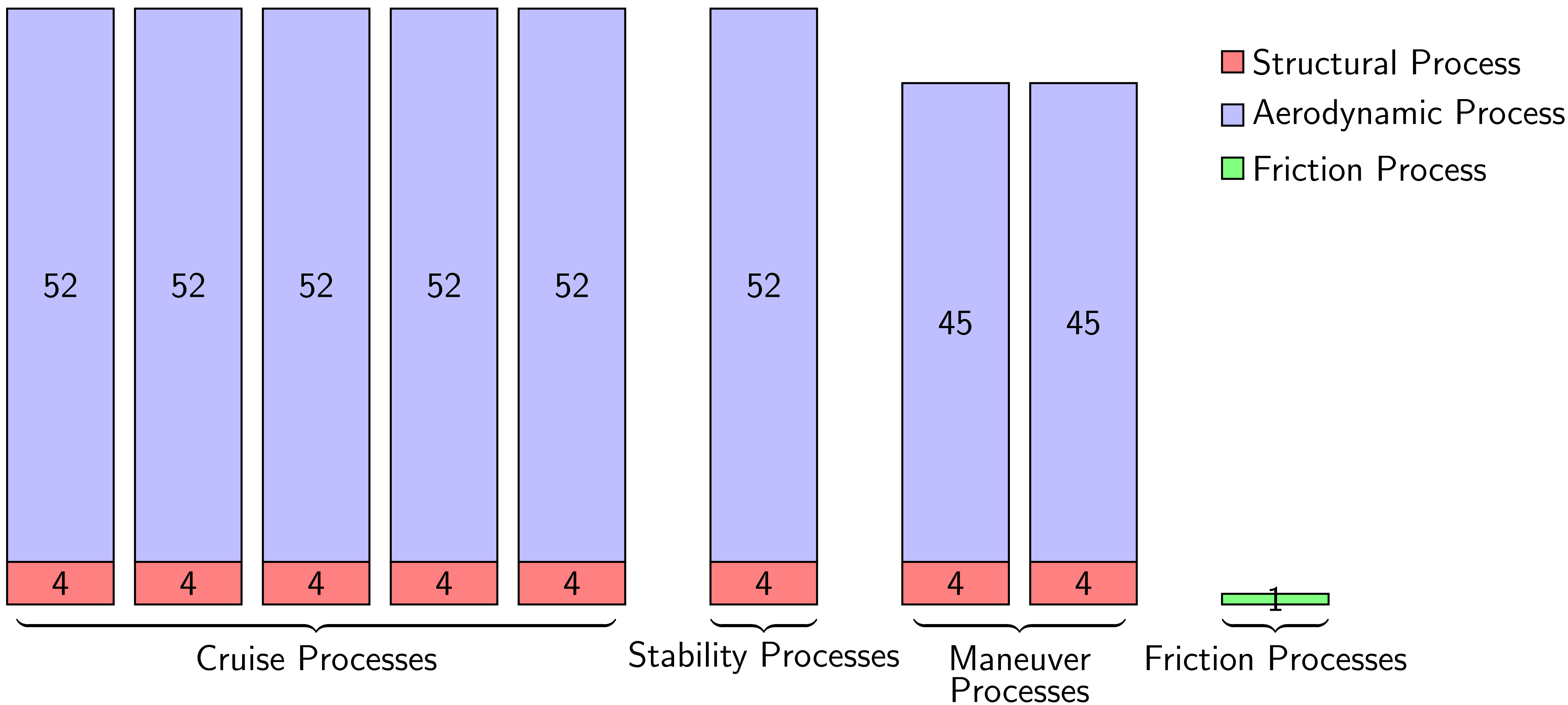
- A variety of geometric constraints are required to produce physically realistic designs
- Lift and moment constraints at each cruise and maneuver condition
- Three Kreisselmeier–Steinhauser (KS) yield stress constraint aggregation functions each maneuver condition

Geometric/target constraints				Aerodynamic constraints				Structural constraints			
Description		Quantity		Description		Quantity		Description		Quantity	
$t_{\text{LE}}/t_{\text{LE}_{\text{init}}}$	$\geq$	1.0	11	$(L-W)_{\text{cruise}}$	$=$	0.0	5	2.5 g Lower skin: KS	$\leq$	1.0	1
$t_{\text{TE}}/t_{\text{TE}_{\text{init}}}$	$\geq$	1.0	11	$C_{my_{\text{cruise}}}$	$=$	0.0	5	2.5 g Upper skin: KS	$\leq$	1.0	1
A/A_{init}	$\geq$	1.0	1	$(L-W)_{\text{Manvr.}}$	$=$	0.0	2	2.5 g Rib/spars: KS	$\leq$	1.0	1
V/V_{init}	$\geq$	1.0	1	$C_{my_{\text{Manvr.}}}$	$=$	0.0	2	1.3 g Lower skin: KS	$\leq$	0.42	1
$t_{\text{TE Spar}}$	$\geq$	0.20	5	Cruise K_n	$\geq$	0.15	1	1.3 g Upper skin: KS	$\leq$	1.0	1
$t_{\text{tip}}/t_{\text{tip}_{\text{init}}}$	$\geq$	0.5	5					1.3 g Rib/spars: KS	$\leq$	1.0	1
MAC-MAC*	$=$	0.0	1								
$X_{\text{CG}} - X^*_{\text{CG}}$	$=$	0.0	1								
Total			36	Total			15	Total		6	
								Grand total		57	

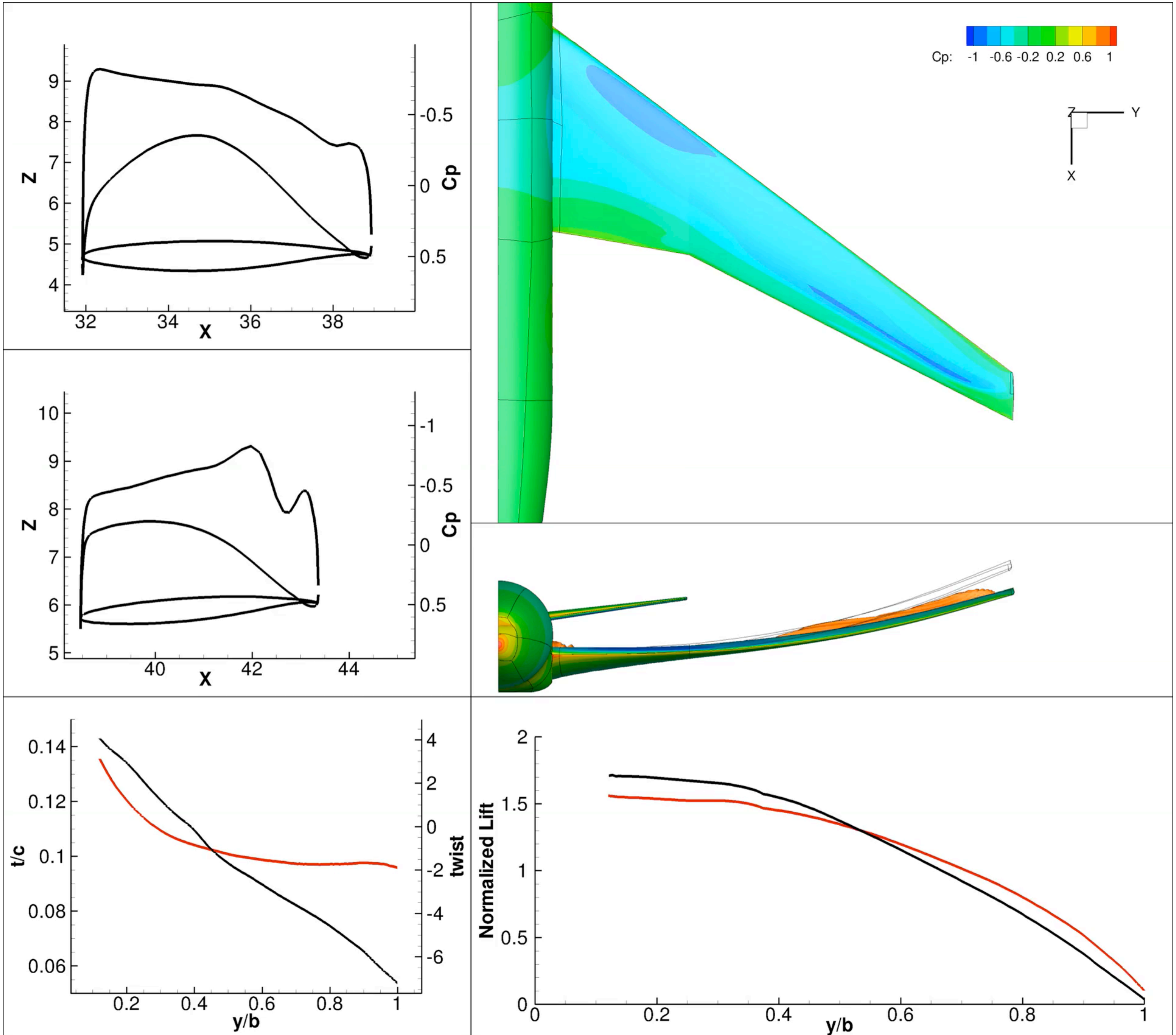
Don’t forget the fuel!



Parallelize, and then parallelize some more



Total: 435 processors

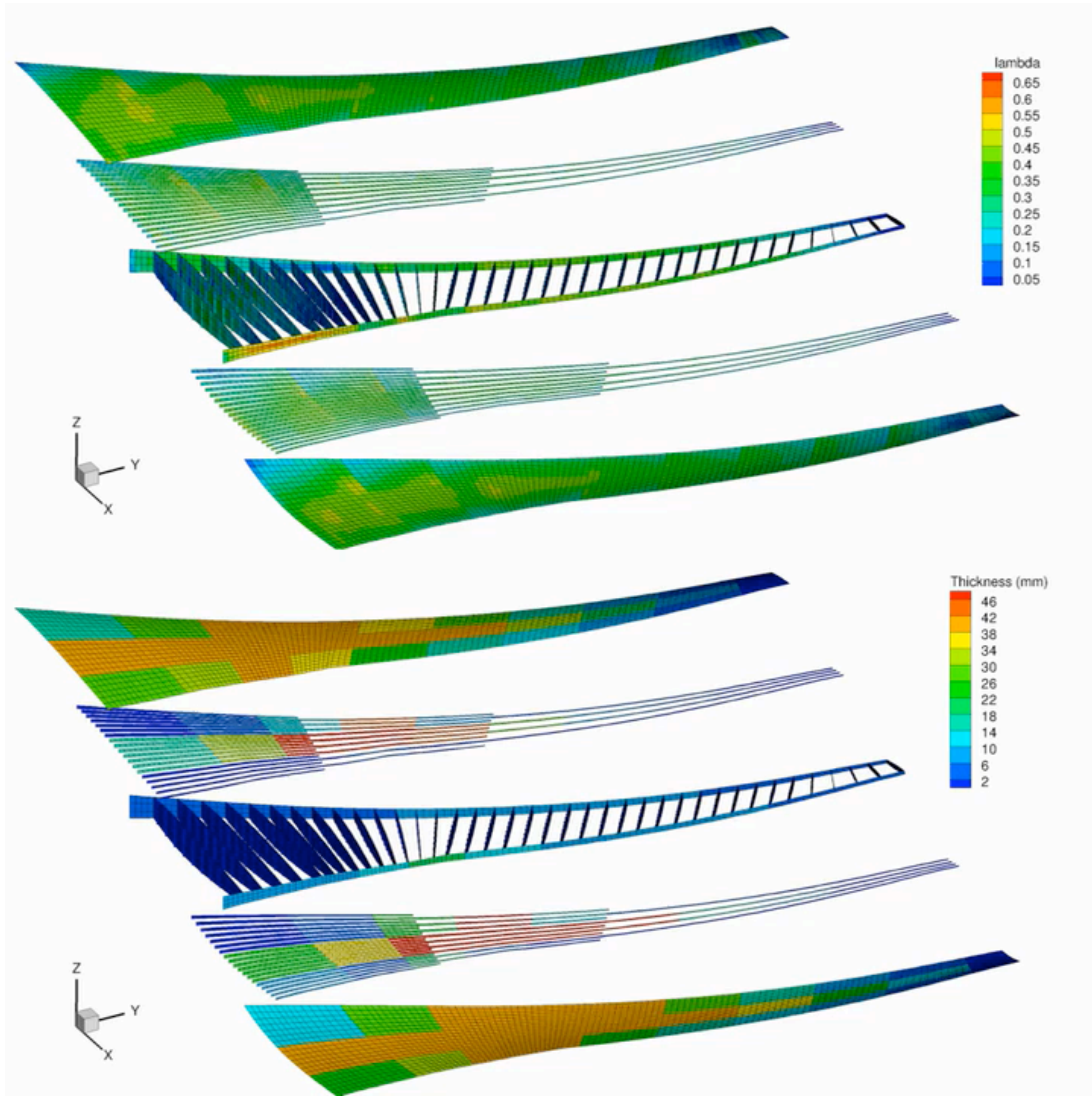


Z
X
Cp
32 34 36 38
4 5 6 7 8 9
-0.5 0 0.5
10
40 42 44
-1
Cp: -1 -0.6 -0.2 0.2 0.6 1
Z Y
X
t/c
twist
y/b
0.06 0.08 0.1 0.12 0.14
0.2 0.4 0.6 0.8 1
-6 -4 -2 0 2 4
Normalized Lift
0 0.5 1 1.5 2

Click here to see the video

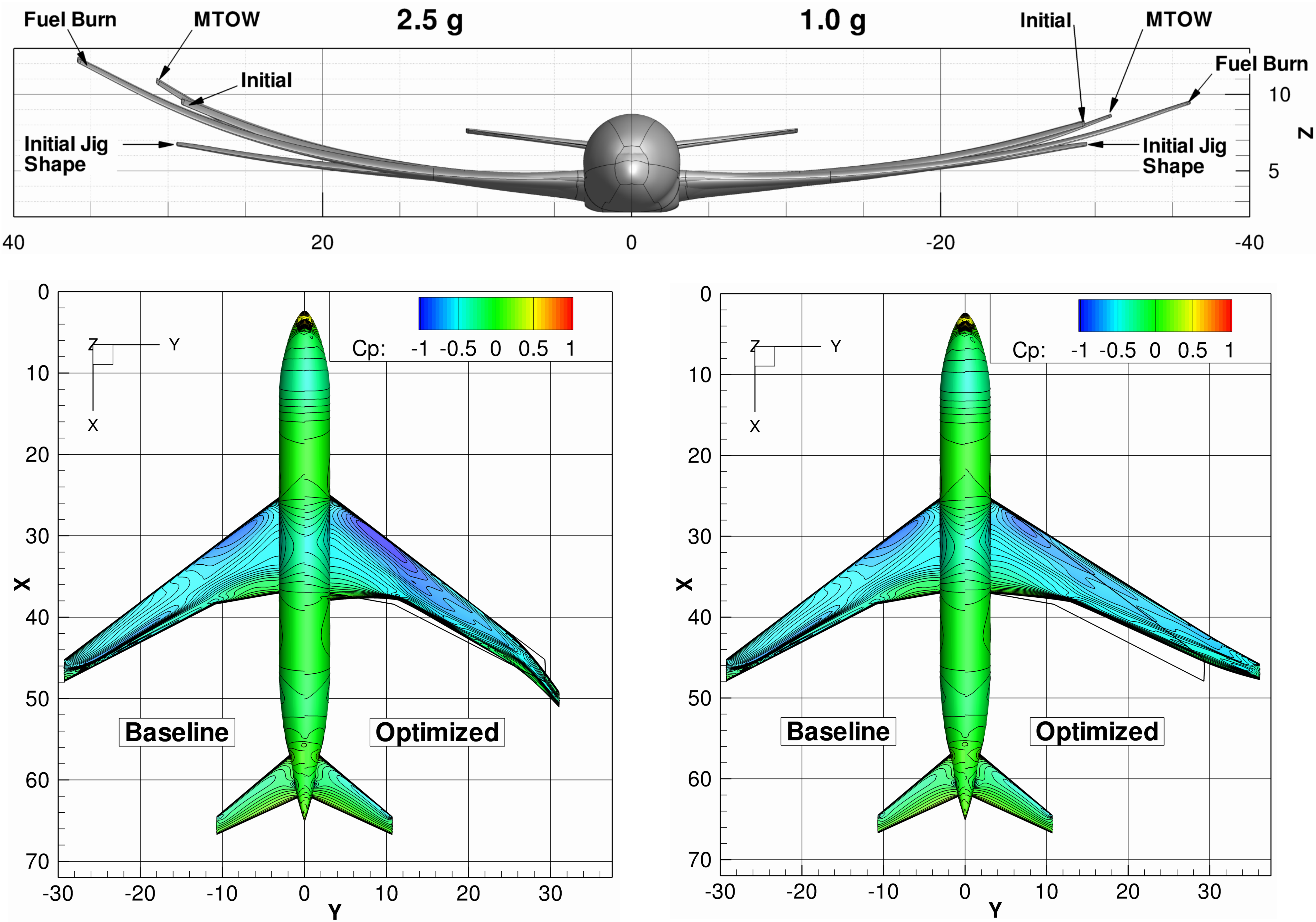
Let's see what happened when we minimized the TOGW...

At the same time, under the skin, the structural sizing processors did their job



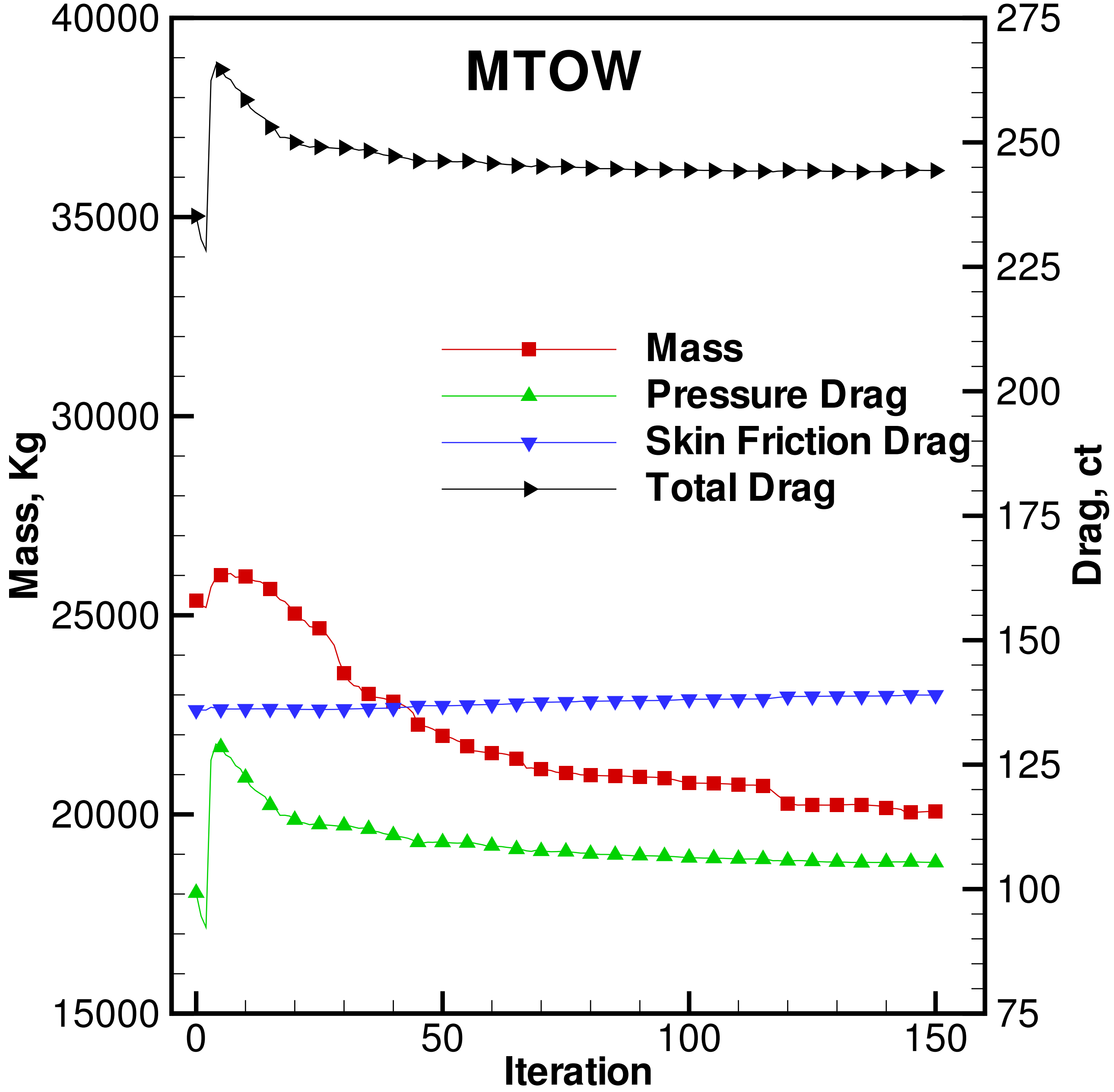
Let's compare this result with a fuel burn minimization...

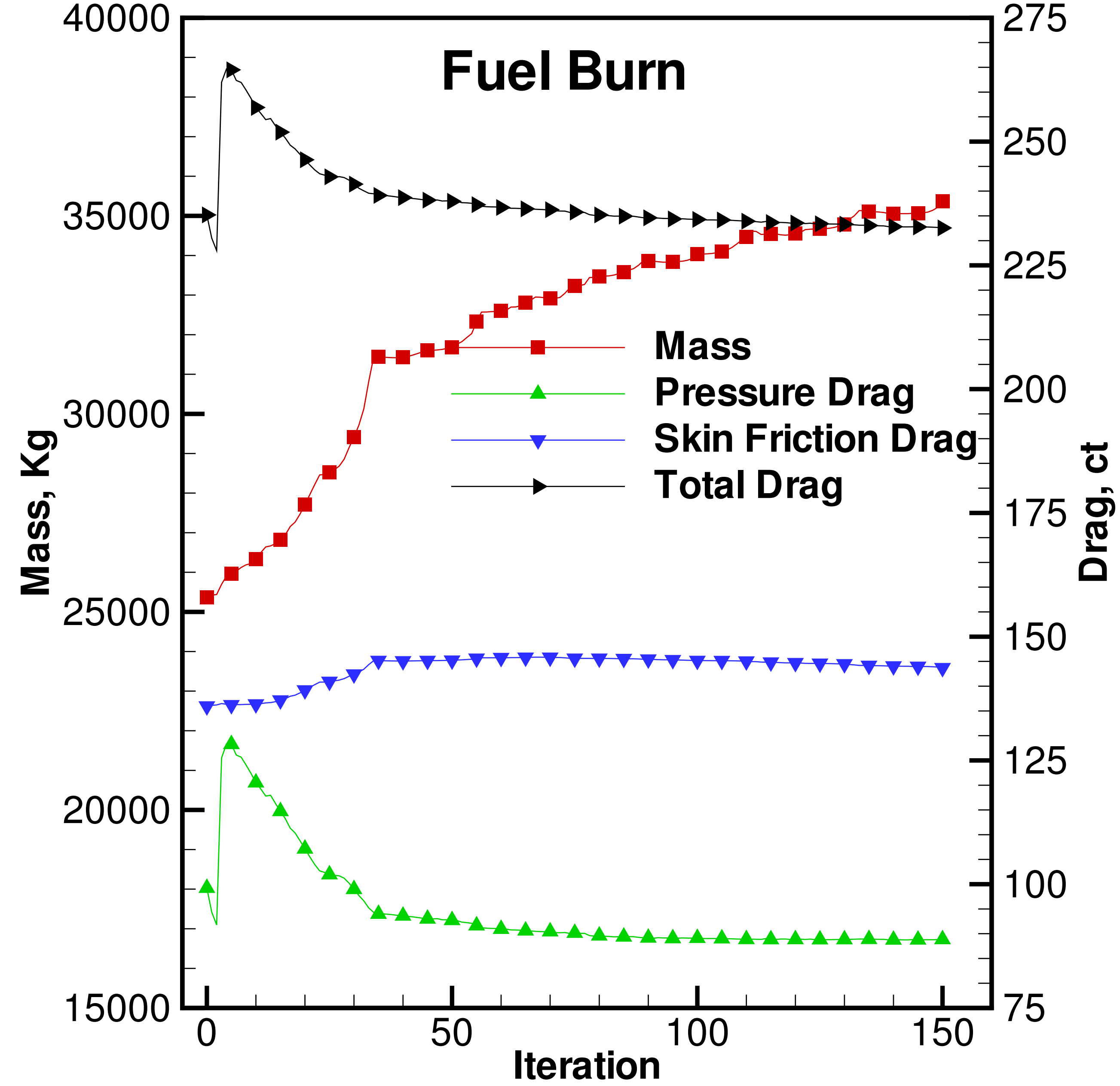
...with custom visualization!



[Kenway, Kennedy and Martins, *AIAA SDM* 2012]

The tale of two objective functions



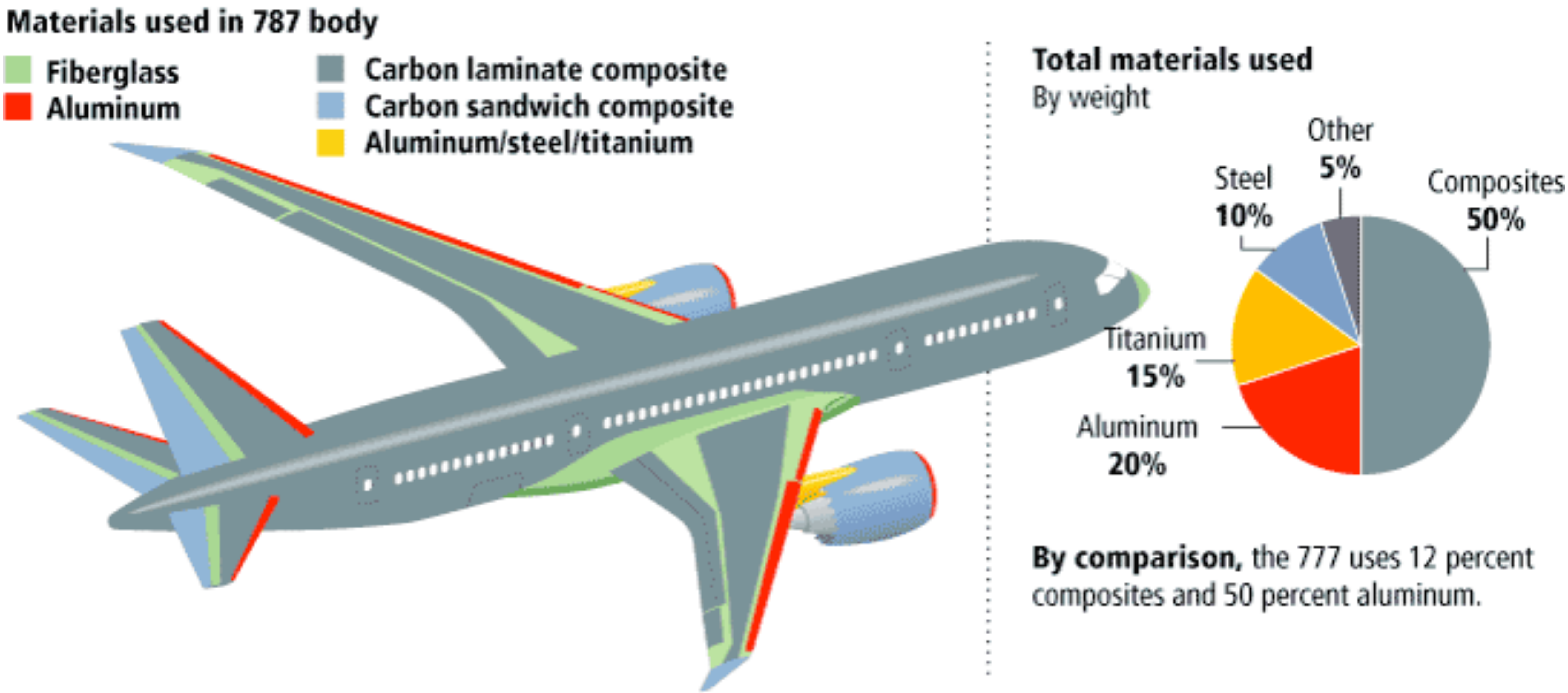


[Kenway, Kennedy and Martins, *AIAA SDM*, 2012]

Composites

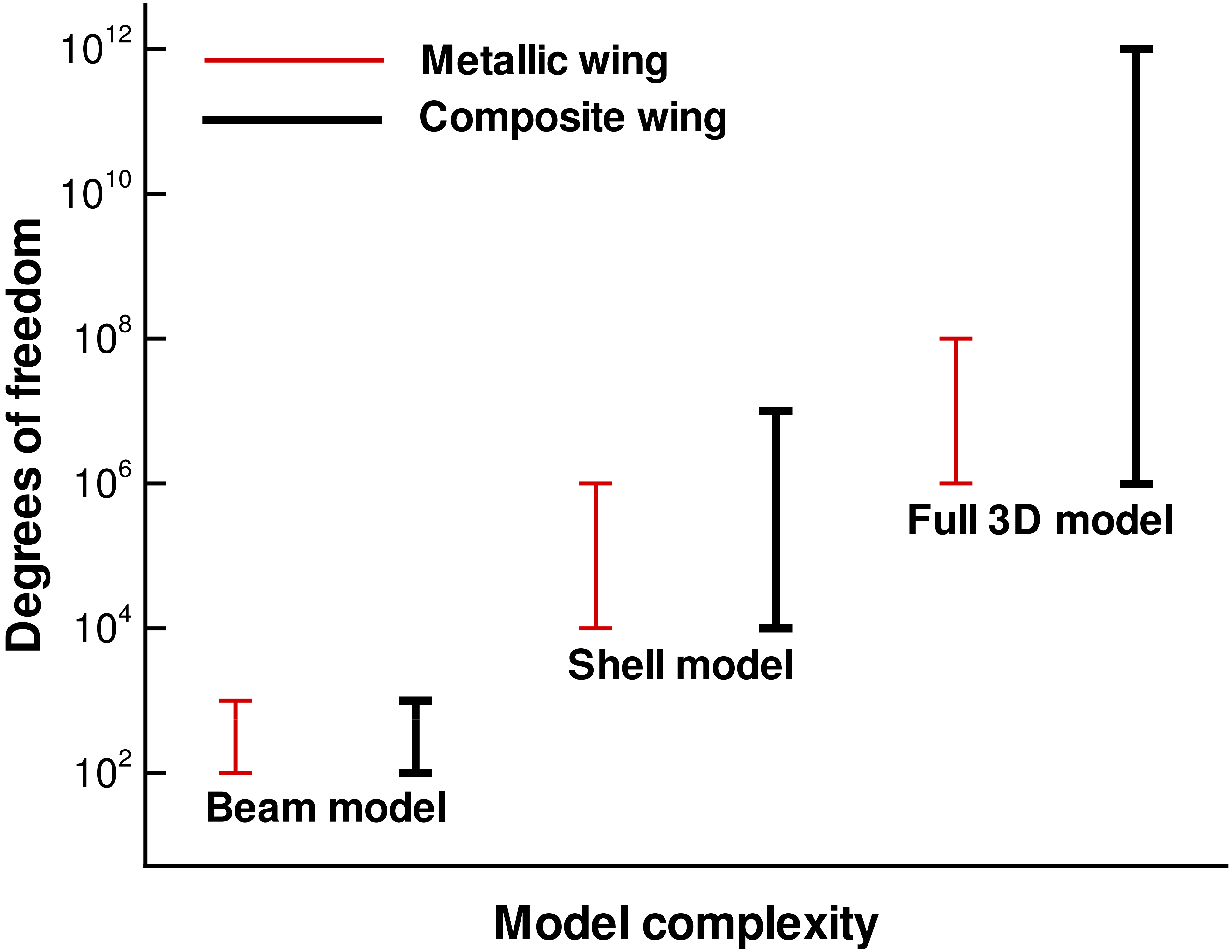
I have just one word for you:

It's taken decades, but composites finally made it to commercial airplanes

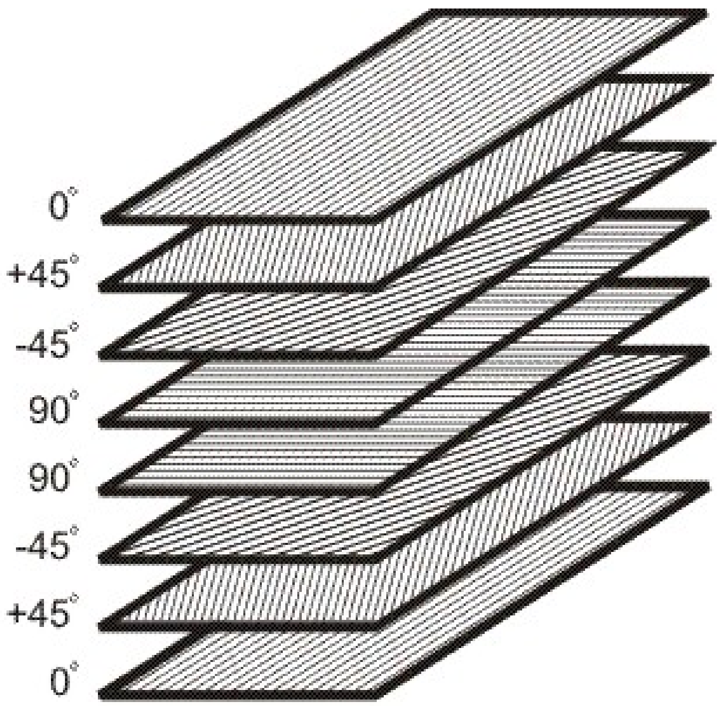


[Flight International]

Step aside CFD; meet the new CPU hog



How to tackle 10^{75} possible lamination sequences



Lamination sequence design:

- Determine a sequence of lamination angles $\{\theta_1, \theta_2, \ldots, \theta_n\}$ to optimize structural performance

Issues:

- Available ply angles may be limited to a discrete set of values, $\Theta = \{-45°, 0°, 45°, 90°\}$
- Parametrization should handle design for strength, buckling and stiffness
- Constrain lamination sequence: matrix cracking

Common approaches:

- Genetic algorithms
- Discrete material optimization (DMO) – a SIMP-type method

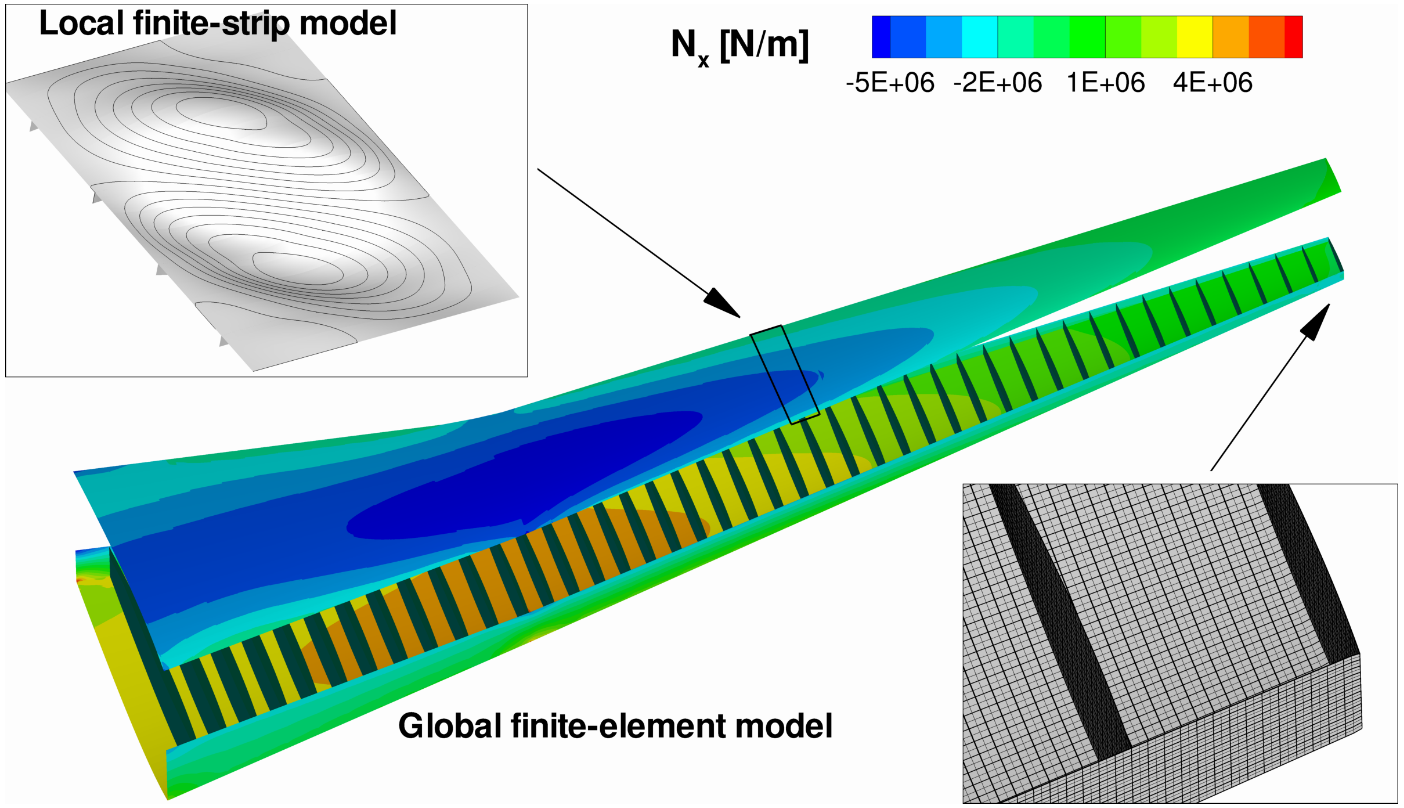
Our proposed approach:

- Use continuous design variable weights for a discrete set of angles
- Use gradient-based optimization so we can handle large problems

Also developed a global-local approach for wing box optimization

1. Initial sizing: mass-minimization using structural thicknesses, stiffener geometry and lamination parameters
2. Layup design using the proposed parametrization technique: maximize load factor with fixed stiffener geometry

- Approximate planform of a 777-200 wing
- Two maneuver conditions: 2.5g and -1g loads
- Global-local approach: local panel model with discrete stiffeners, global model with smeared stiffeners



- Global model contains 67 584, 3rd order MITC9 shell elements, with just over 1.6 million degrees of freedom
- 64 processors: function evaluation: 30s, gradient using adjoint: 45s

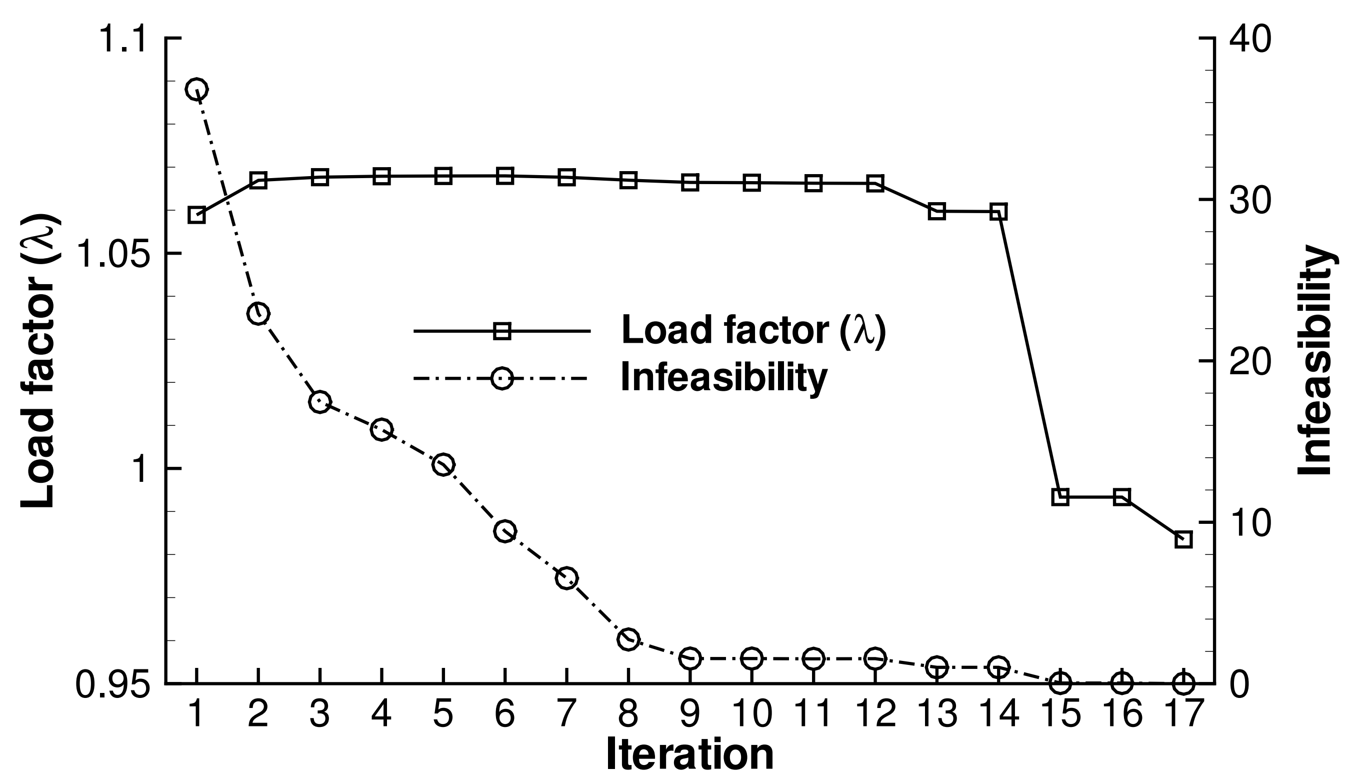
40 hours later... we have an optimum

Lamination sequence:

- Use a thick, guide laminate: changes in thickness accomplished by removing outer-most layers
- Group plies into 0°_2, $\pm 45^\circ$ and 90°_2
- No more than four contiguous 0° or 90° plies

The design problem:

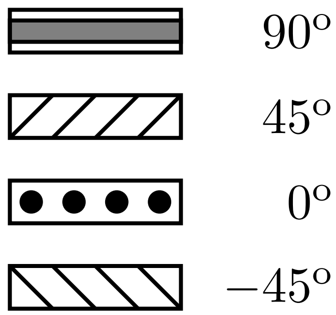
- 157 plies, 472 design variables
- 30 KS failure and 30 KS buckling constraints: 15 for each load case
- 298 contiguity constraints
- There are $3^{157} \approx 8 \times 10^{74}$ possible sequences



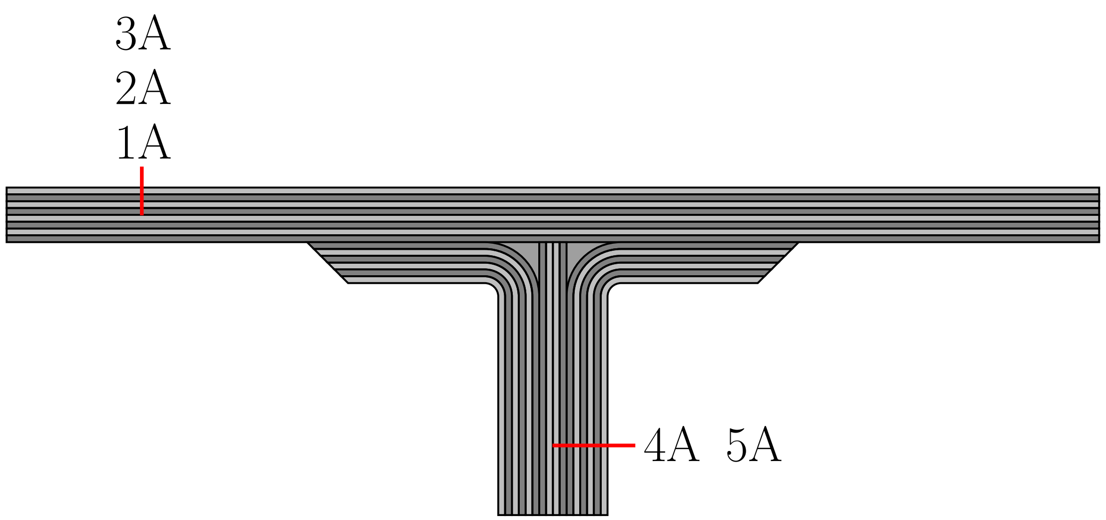
- Optimization time: 39 hours 53 minutes on 64 processors
- 3230 function evaluations, 1101 gradient evaluations

How these results stack up

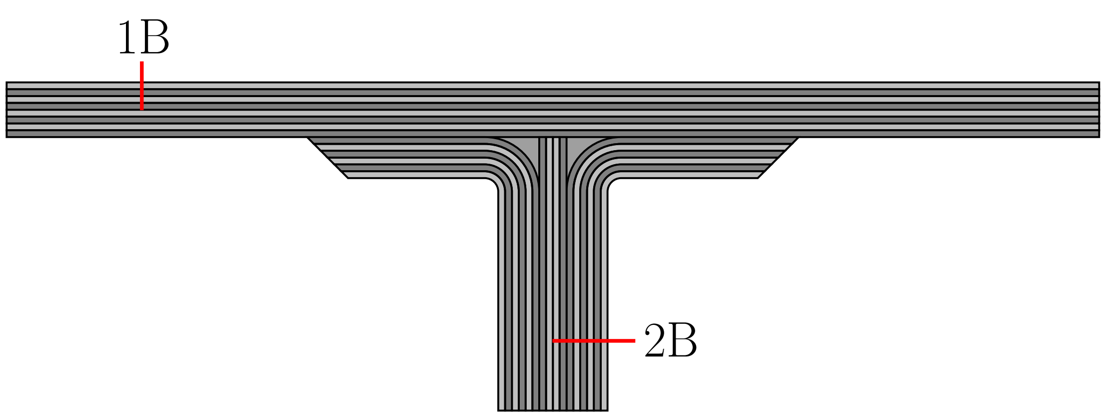
- Symmetric laminates: sequence from the middle to outer layer

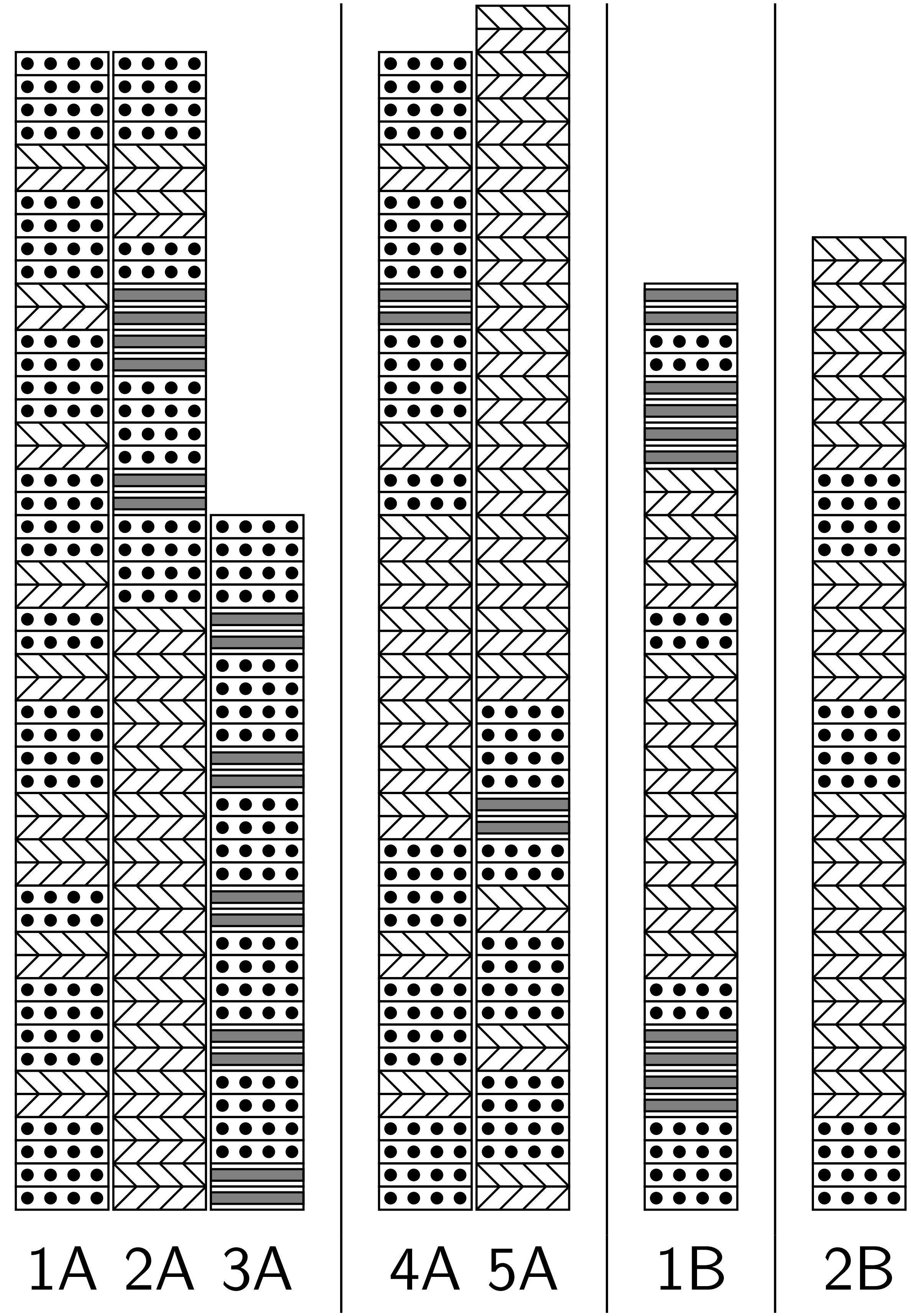


Top skin layup



Bottom skin layup





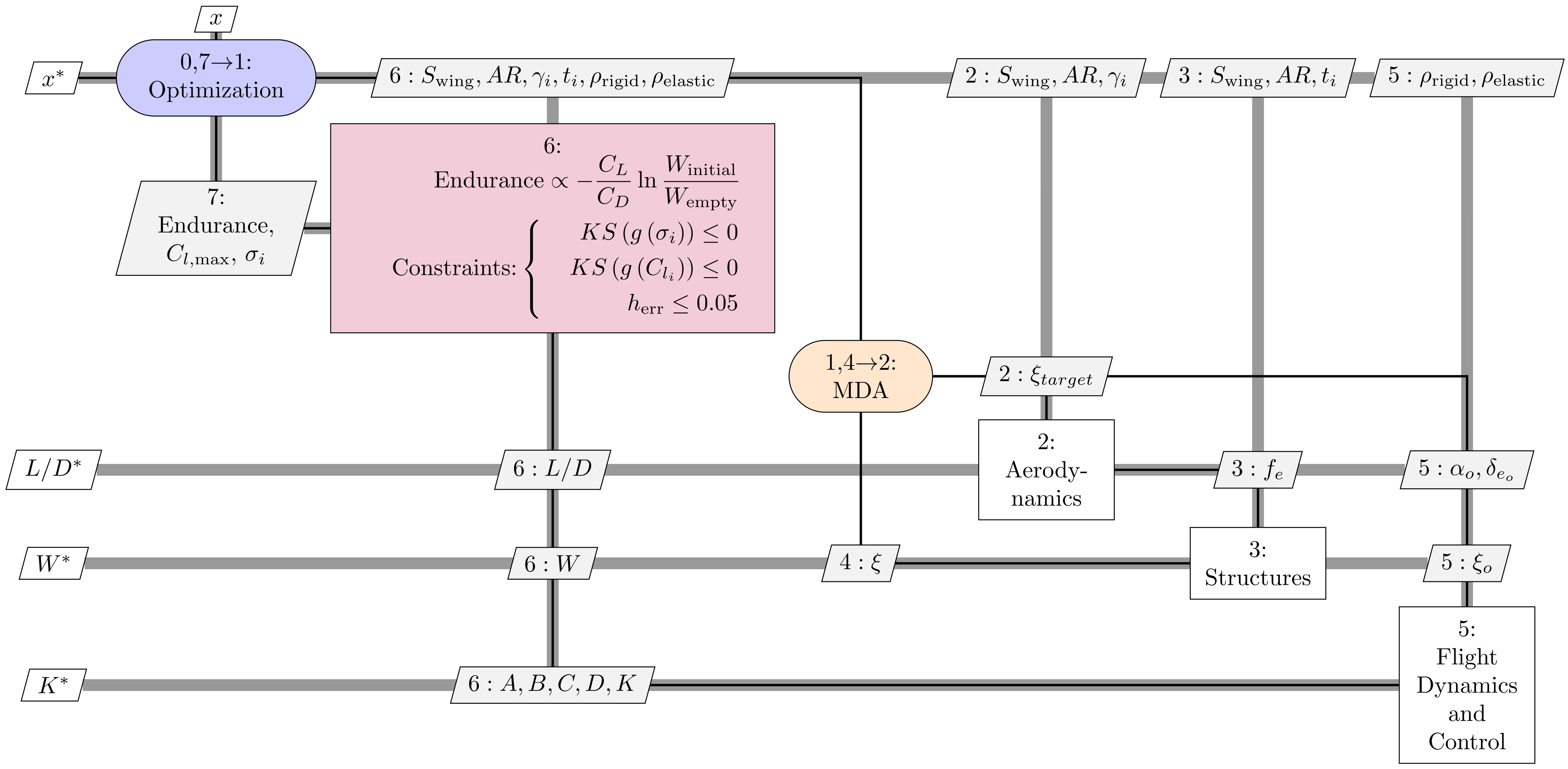
[Kennedy and Martins, *AIAA SDM*, 2012]

Why can't we just all work together?

Aerodynamic shape + Structural sizing + Control gains =

Aeroservoelastic Optimization

This aeroservoelastic optimization considers maneuver and gust loads

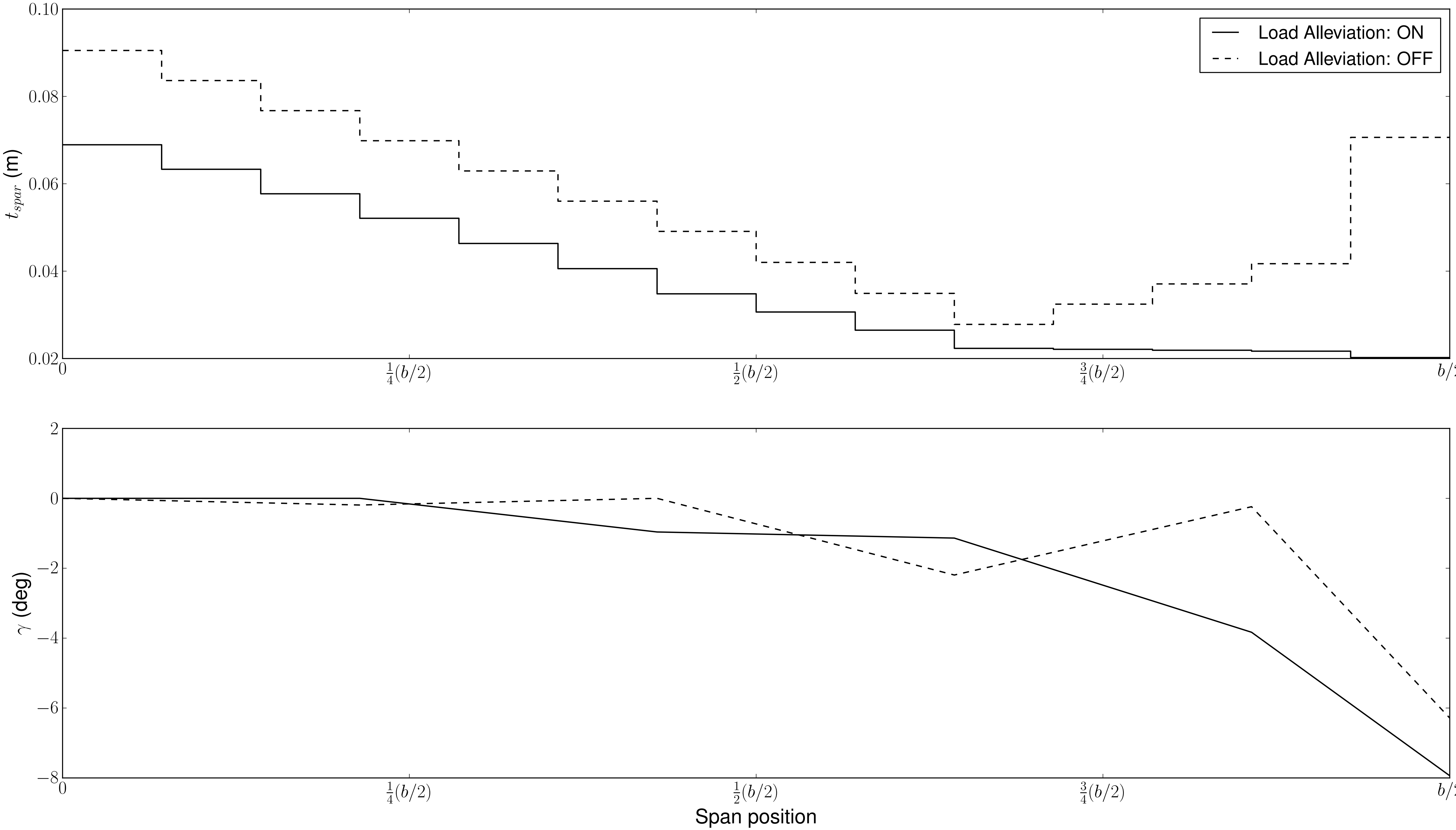


Aeroservoelastic optimum was significantly better than the aerostructural one...

Optimization results with and without load alleviation system.

Load alleviation	Off	On	
S_{ref} (m^2)	219.18	191.47	14.5% smaller
AR	13.98	14.03	
L/D	34.29	34.37	
q_{elastic}	1499.95	1499.88	
q_{rigid}	90.63	75.71	
Wing mass (kg)	13,378	7,817	41.5% lighter
Endurance factor	31.90	38.83	21.7% higher

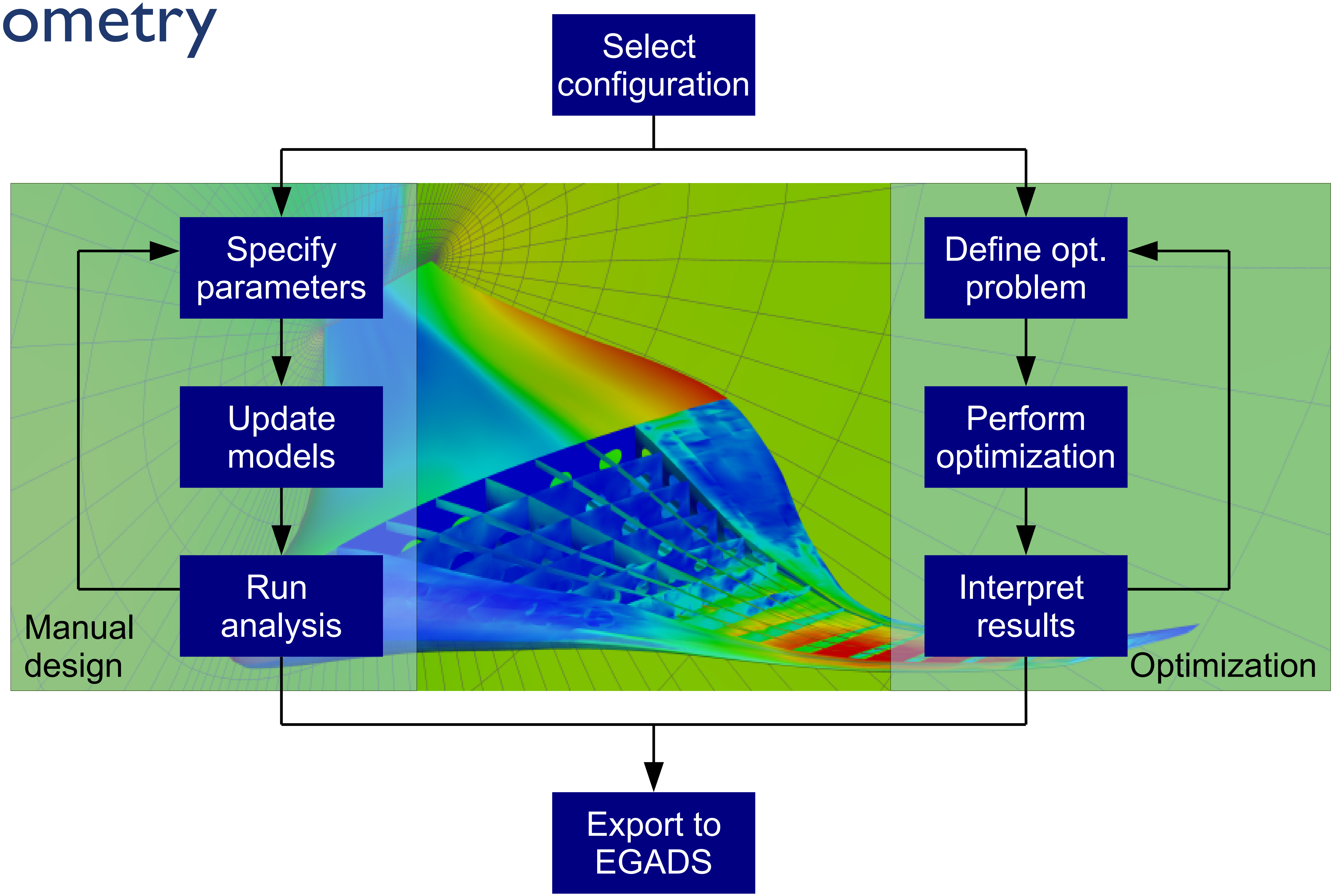
... but the aerostructural optimization found it's own way to alleviate loads



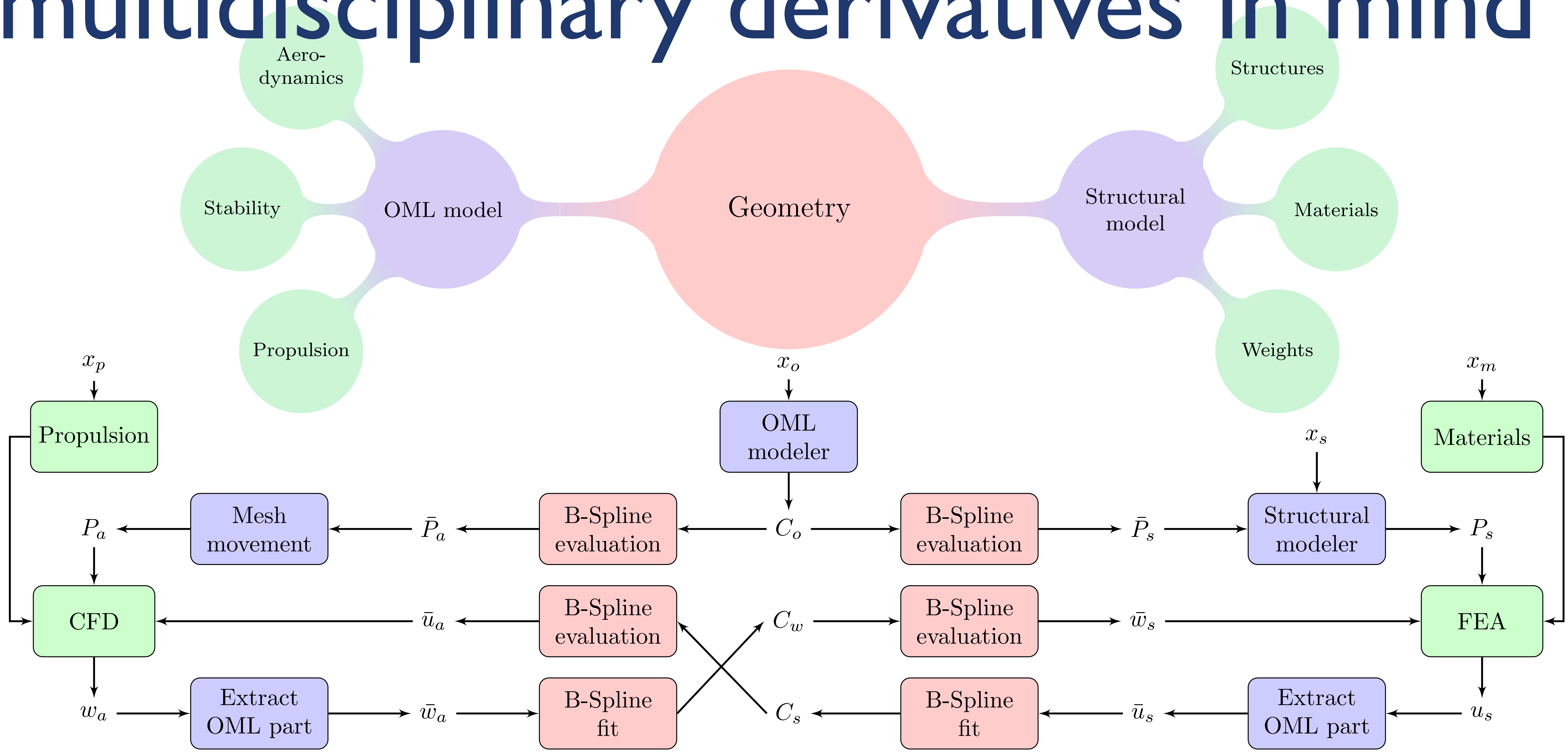
[Haghighat and Martins, *Journal of Aircraft*, 2012]

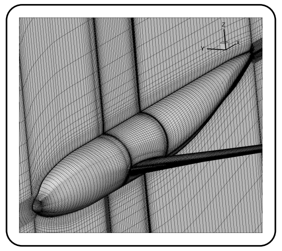
CAD-free
(but CAD-friendly)
Geometry

GeoMACH is a NASA-funded open-source project to handle parametric aircraft geometry

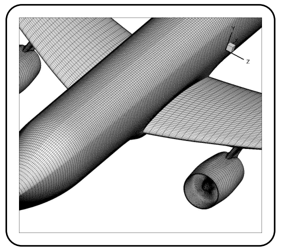


OML modeler developed with multidisciplinary derivatives in mind

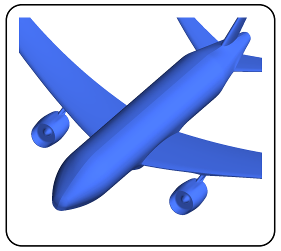




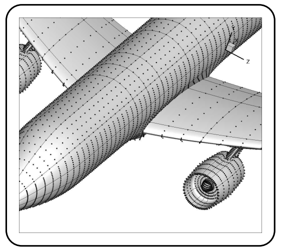
CFD mesh



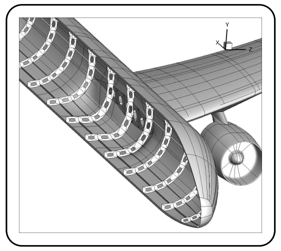
CFD surf. mesh



Continuous OML

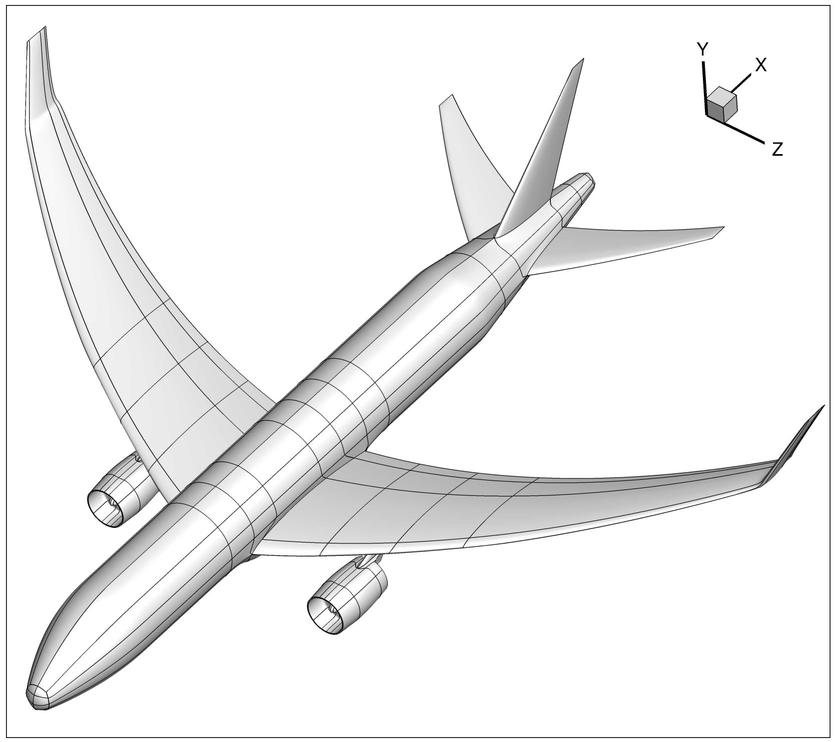


FEA surf. mesh

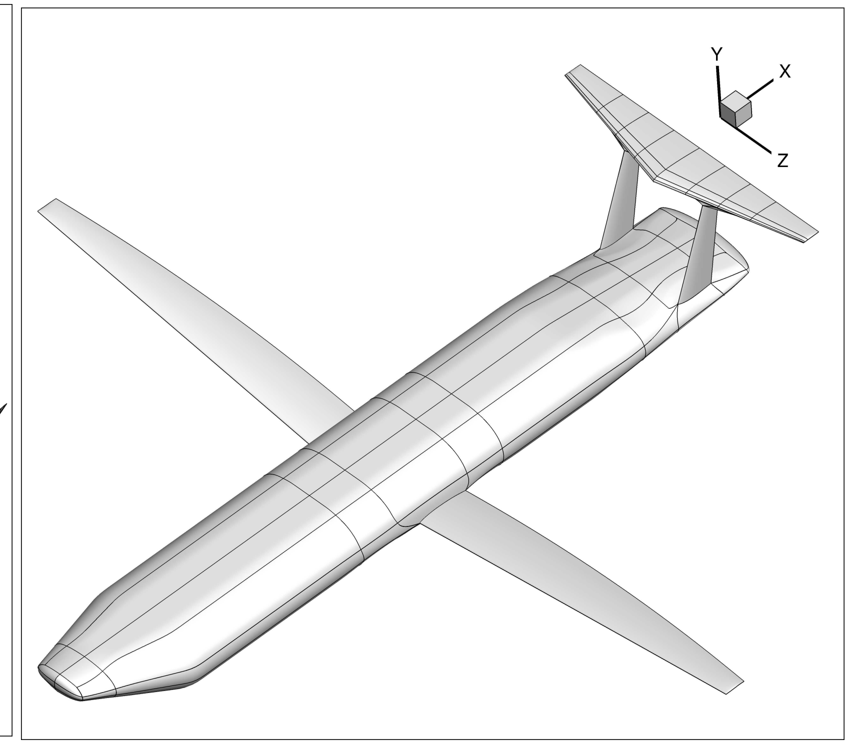


FEA mesh

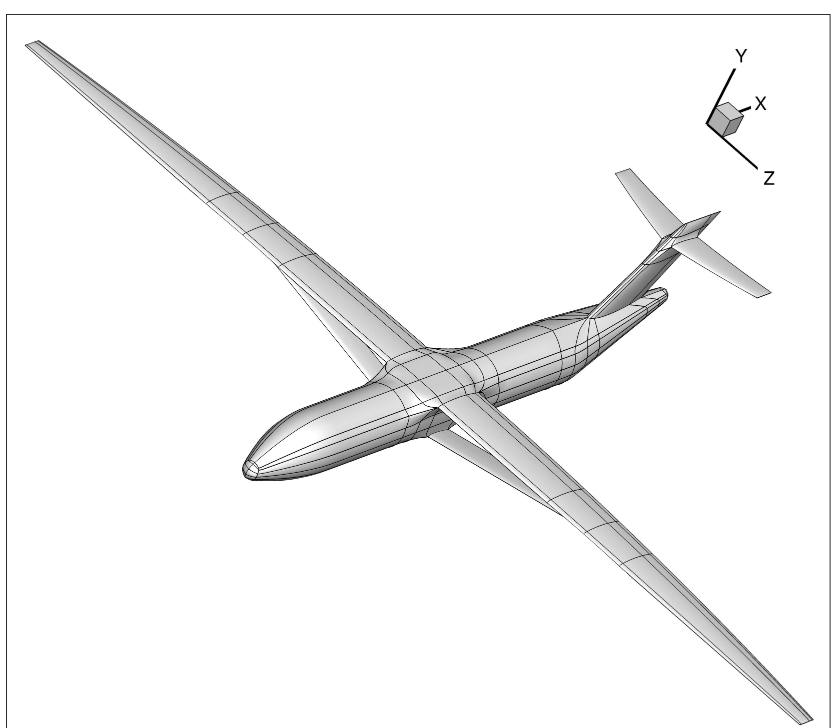
Parametric aircraft configurations can be created with 10–20 lines of code



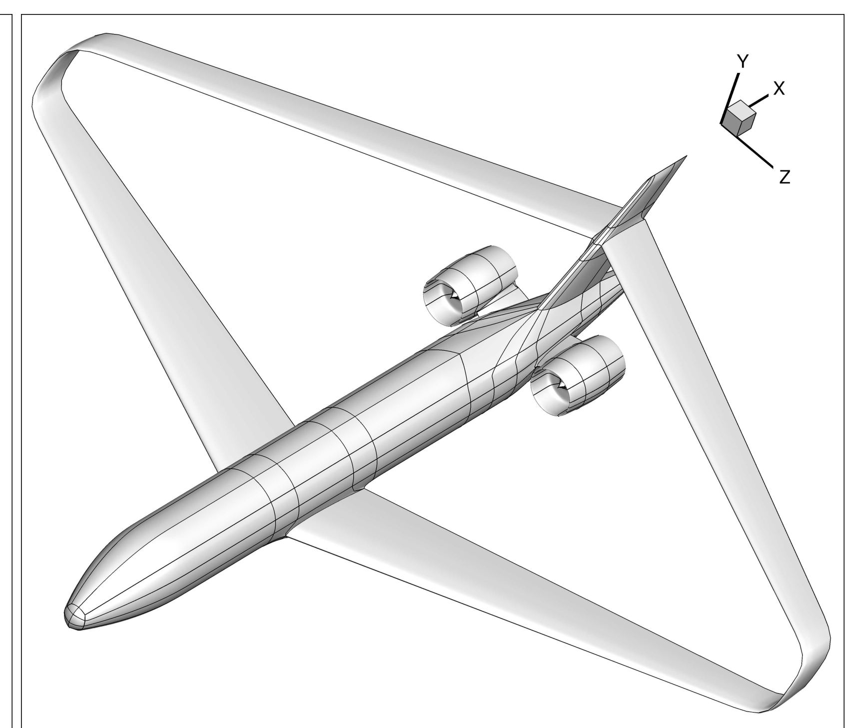
Conventional with winglets



D8

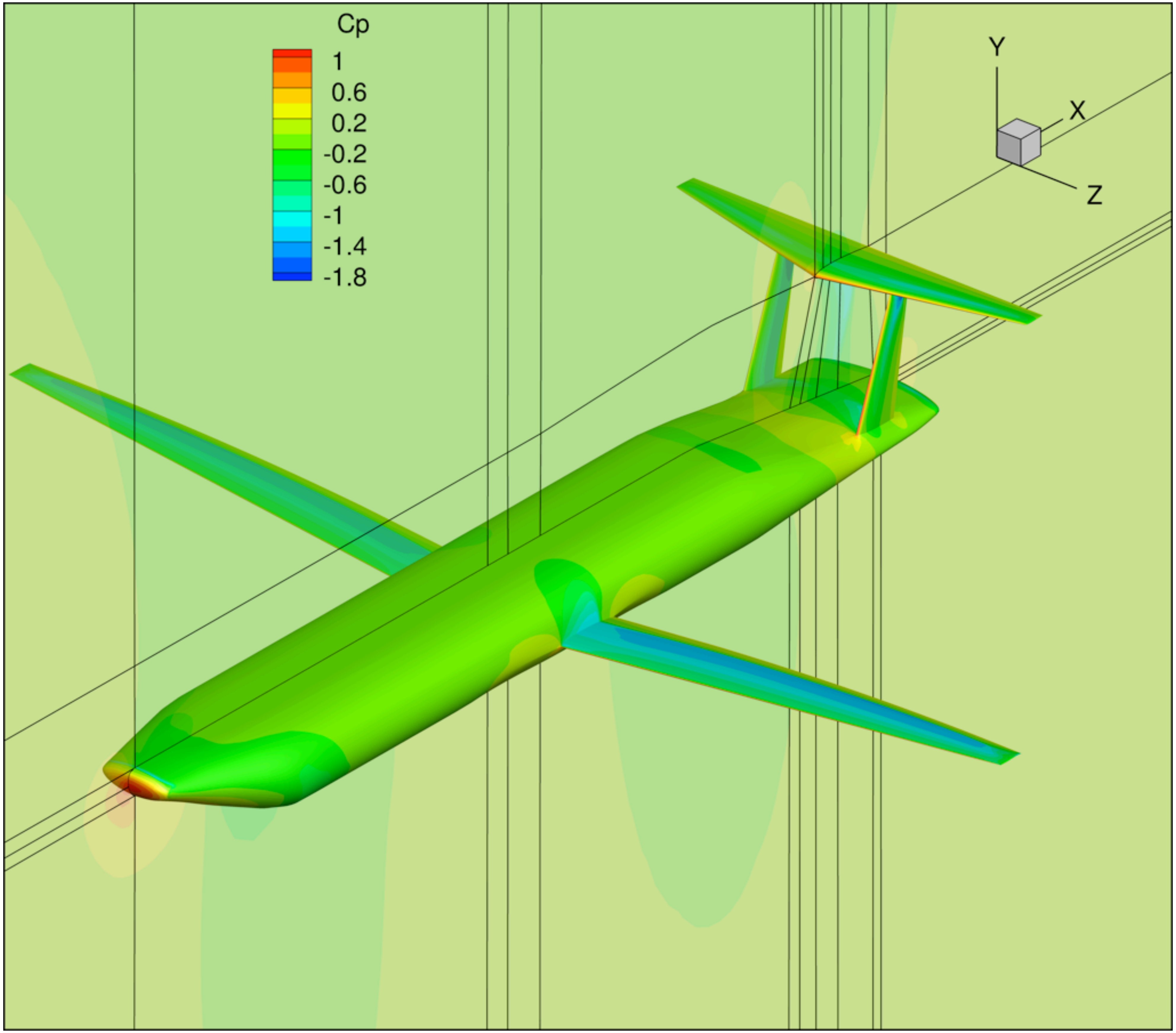


Strut-braced wing

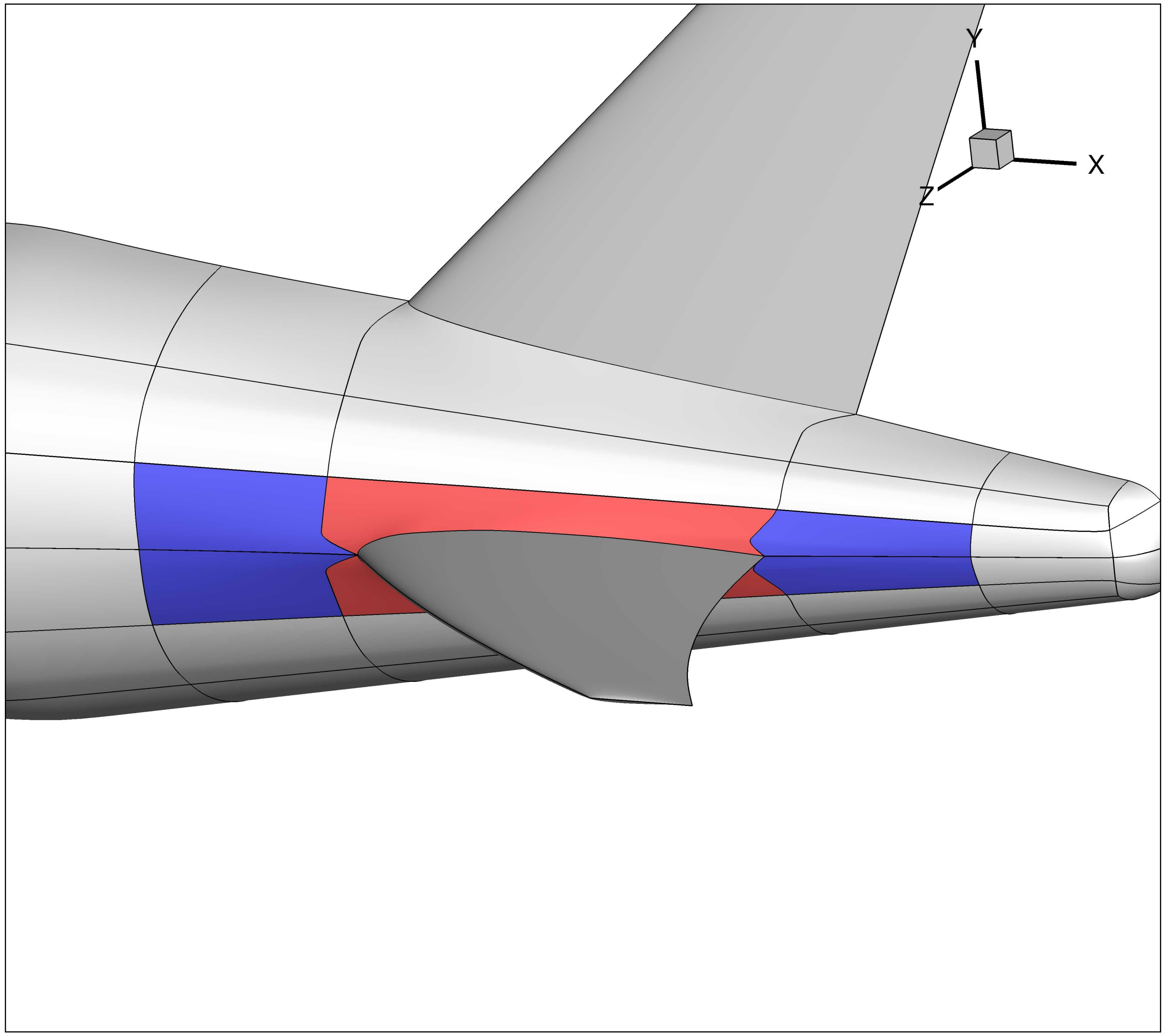


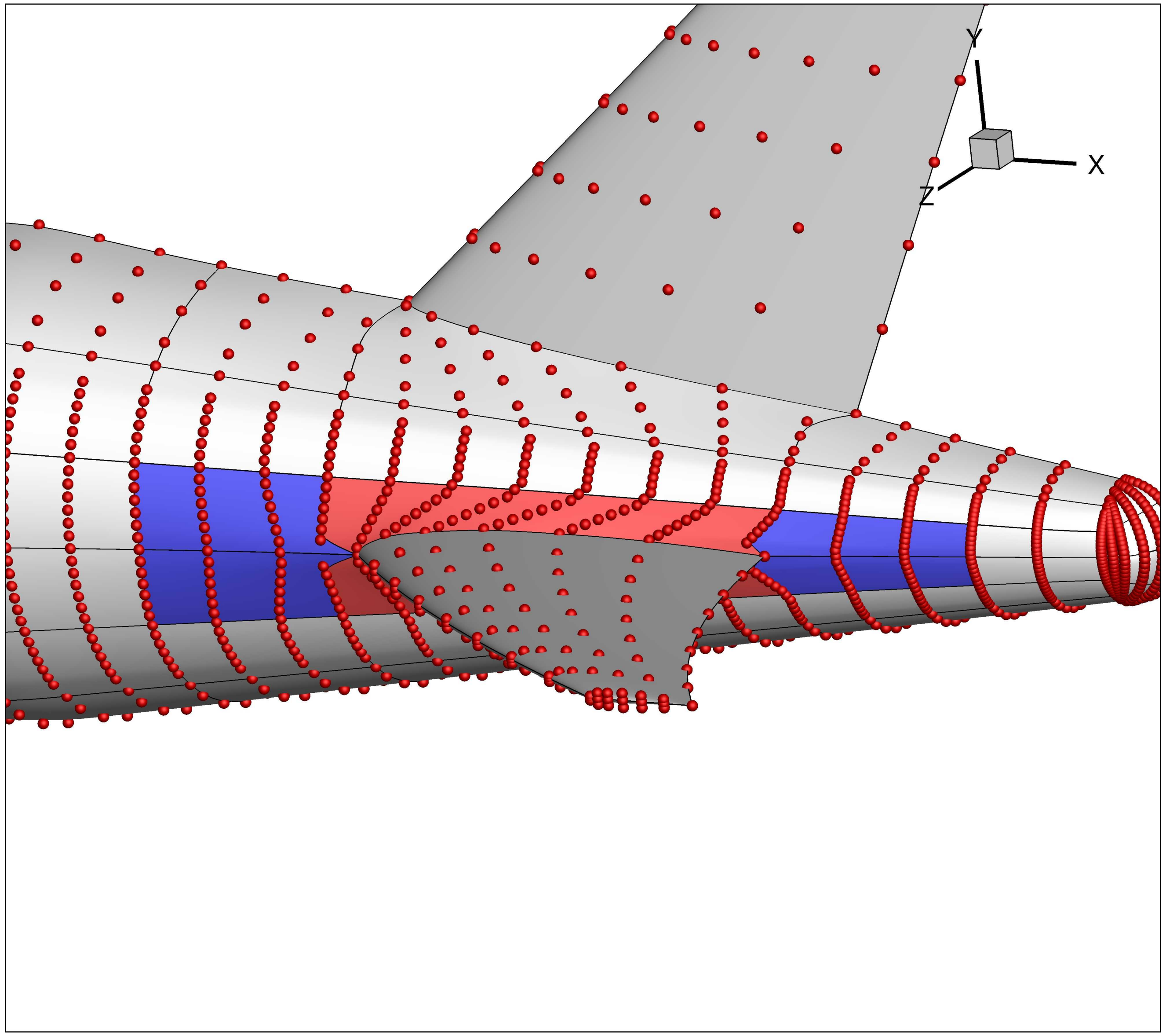
Joined wing

CFD grid for each configuration

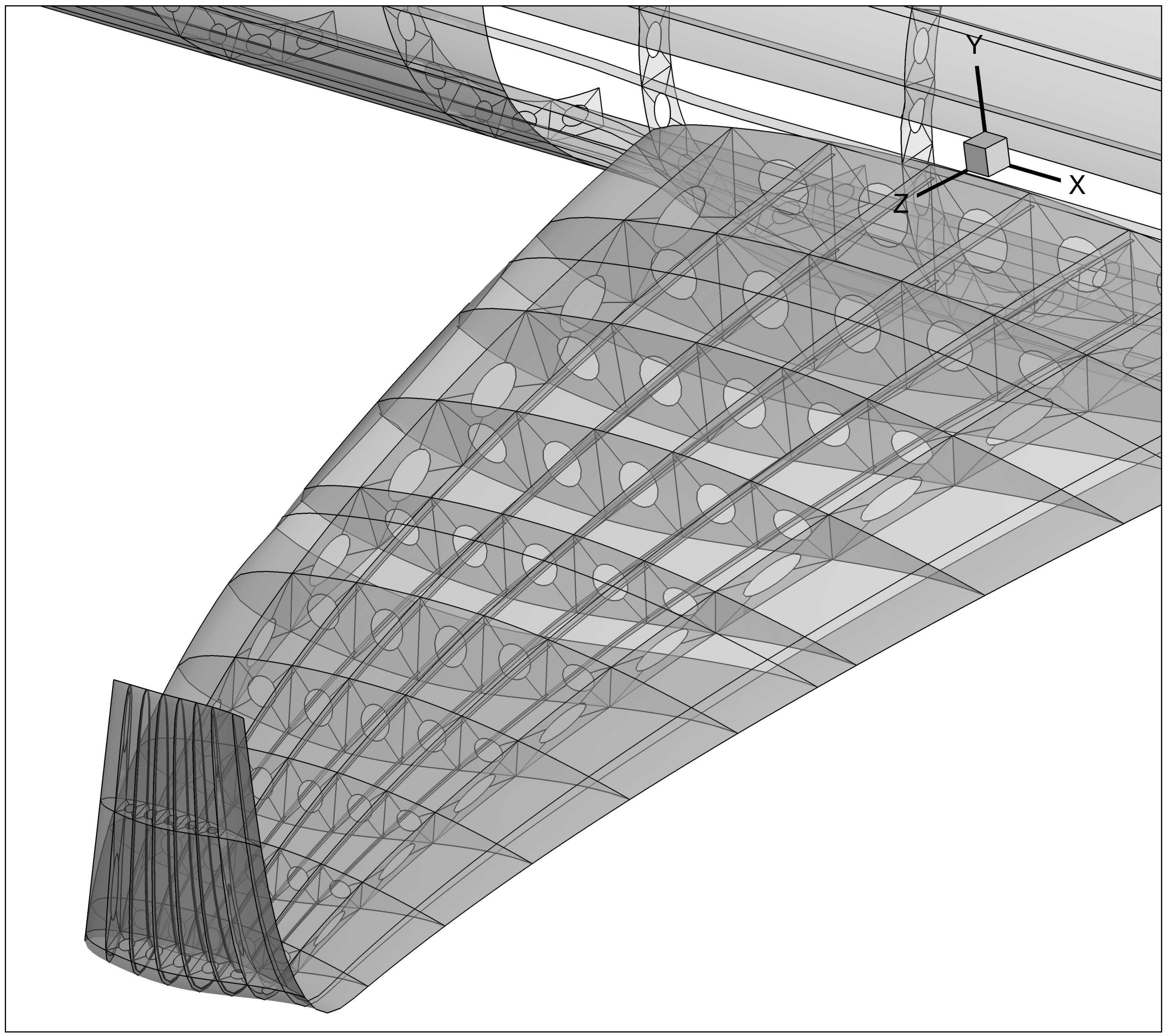


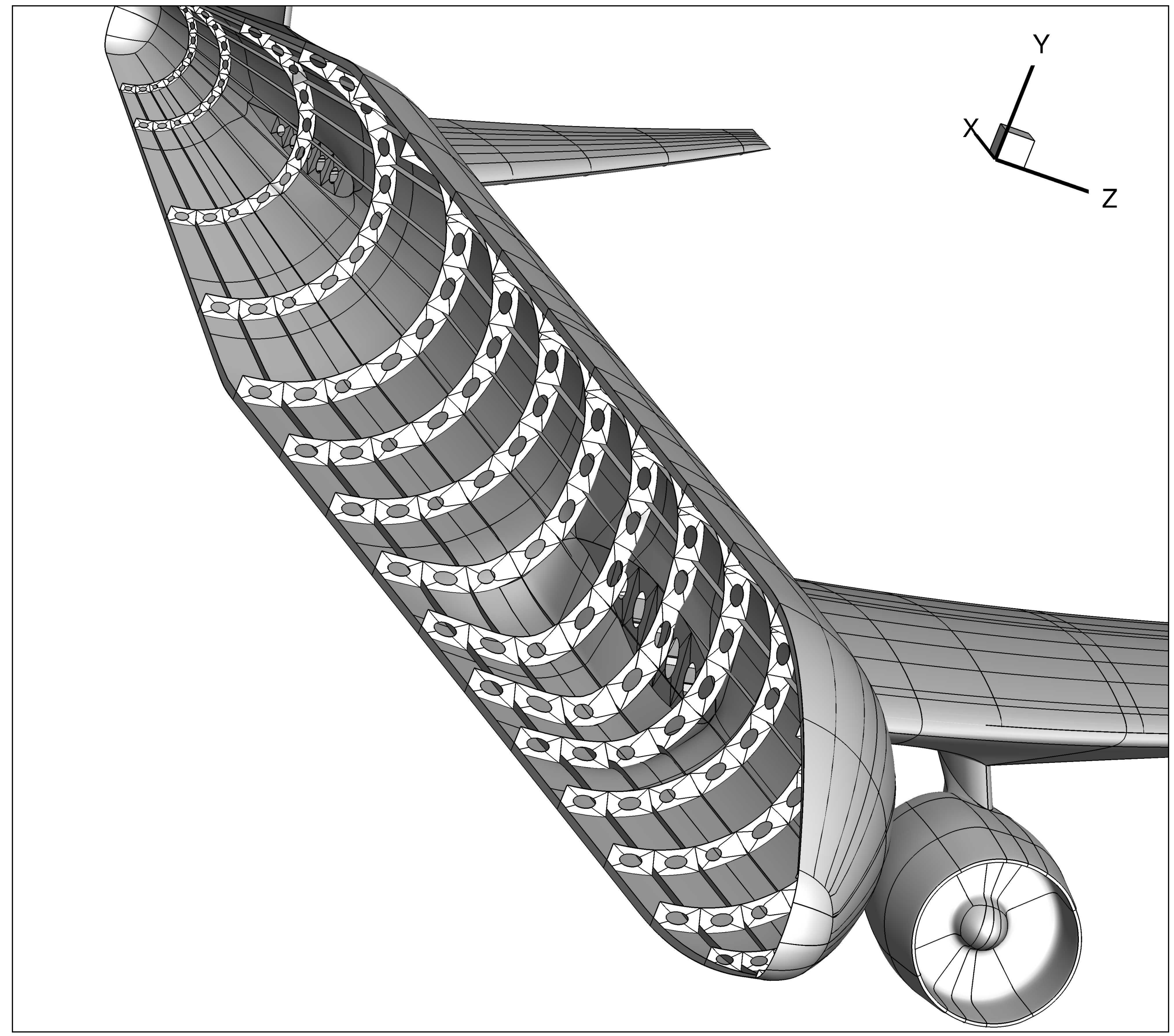
The devil is in the junctions





In addition to the OML, we also generate the internal structure



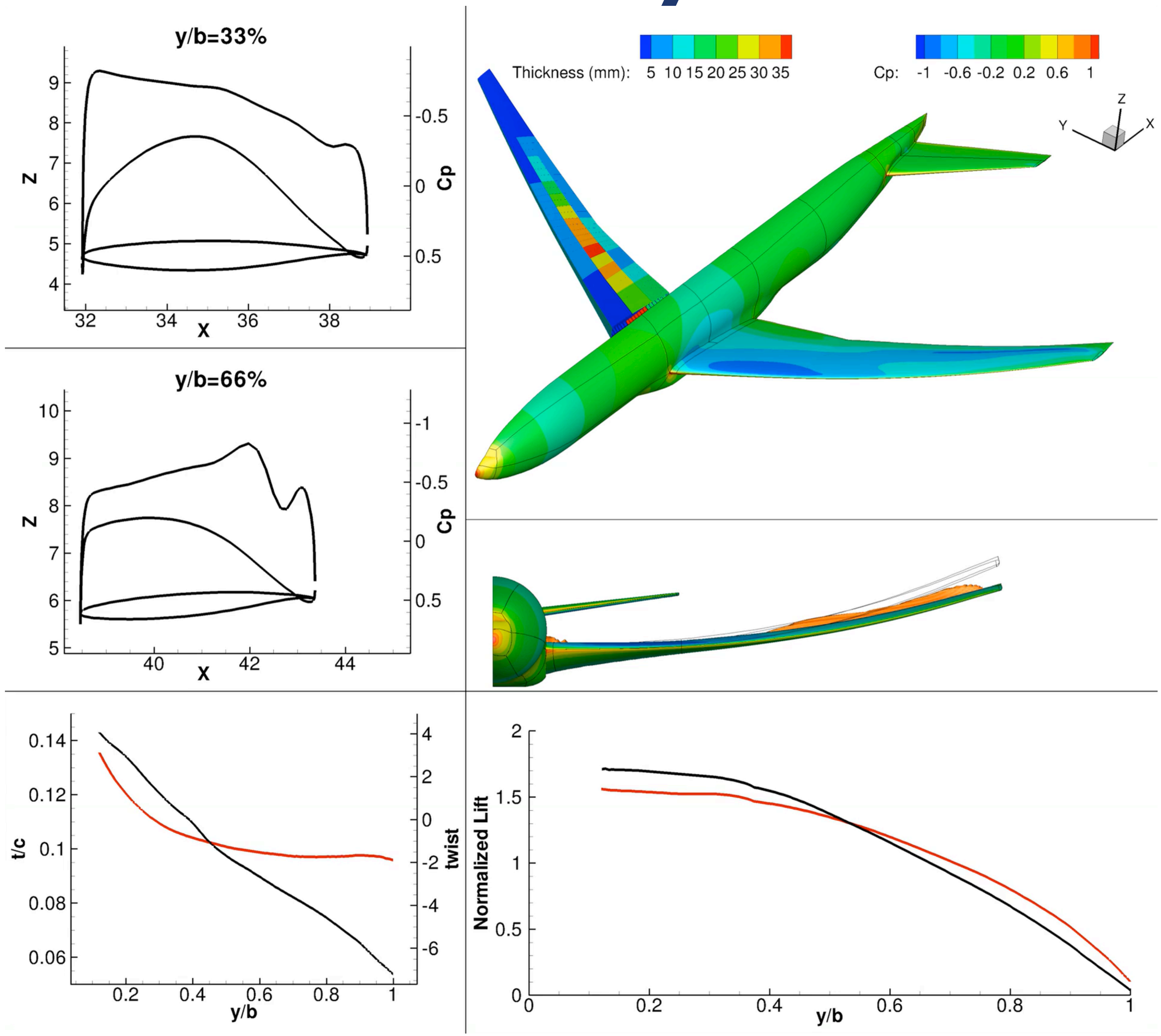


Morphing video

[Hwang and Martins, AIAA MA&O, 2012]

Thanks to my ~~minions~~ heroes

Thank you!



http://mdolab.engin.umich.edu/publications