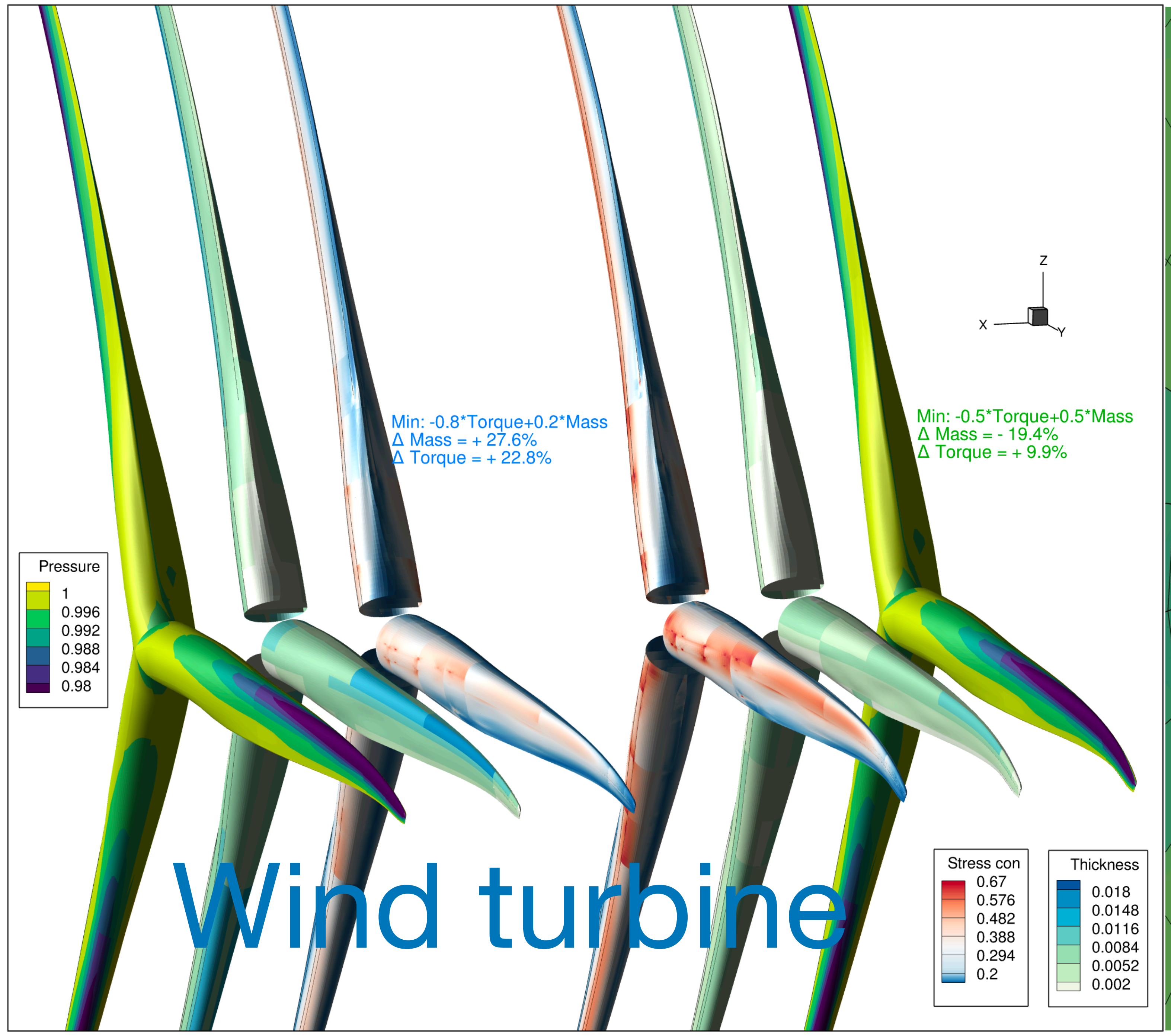
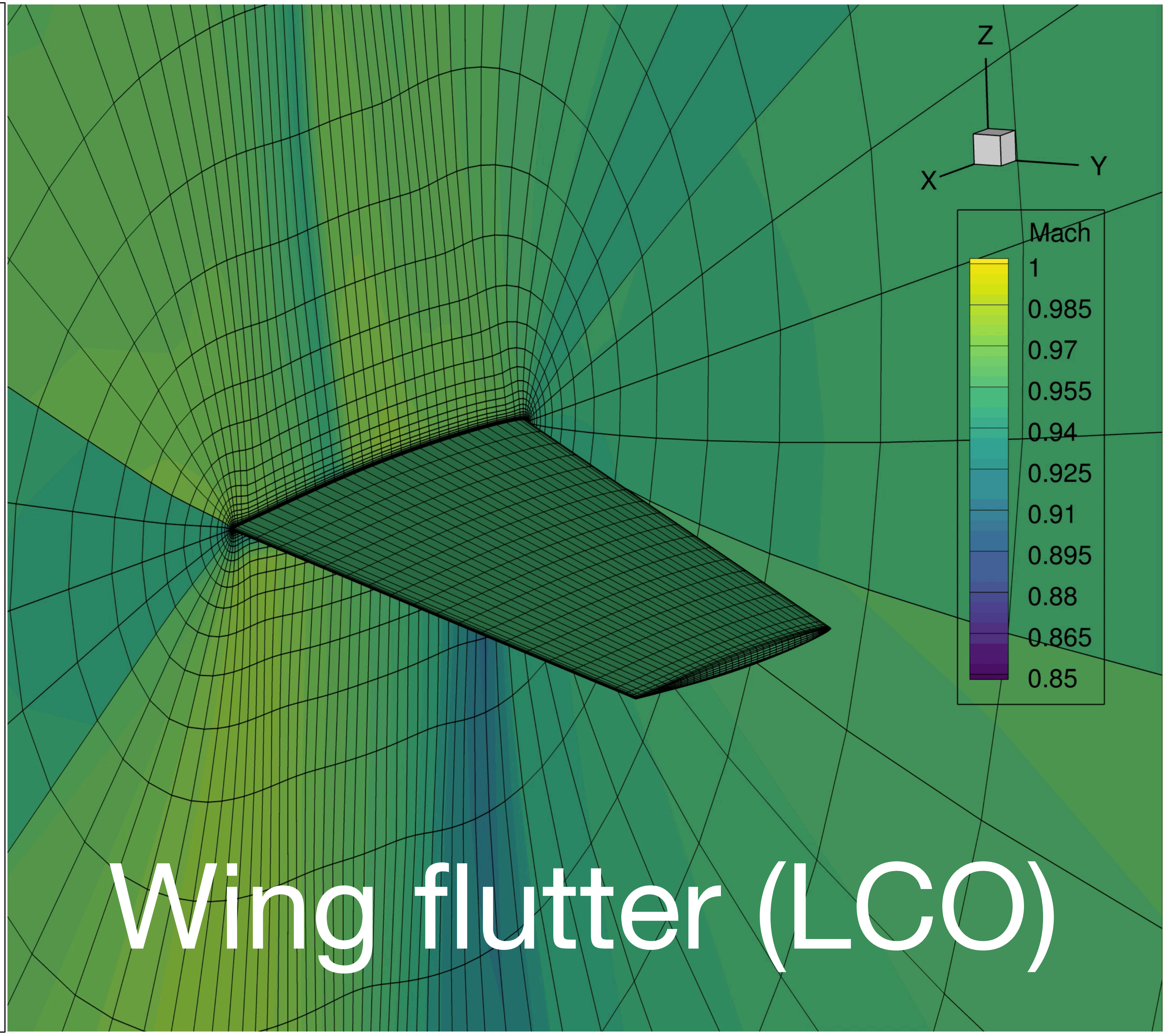
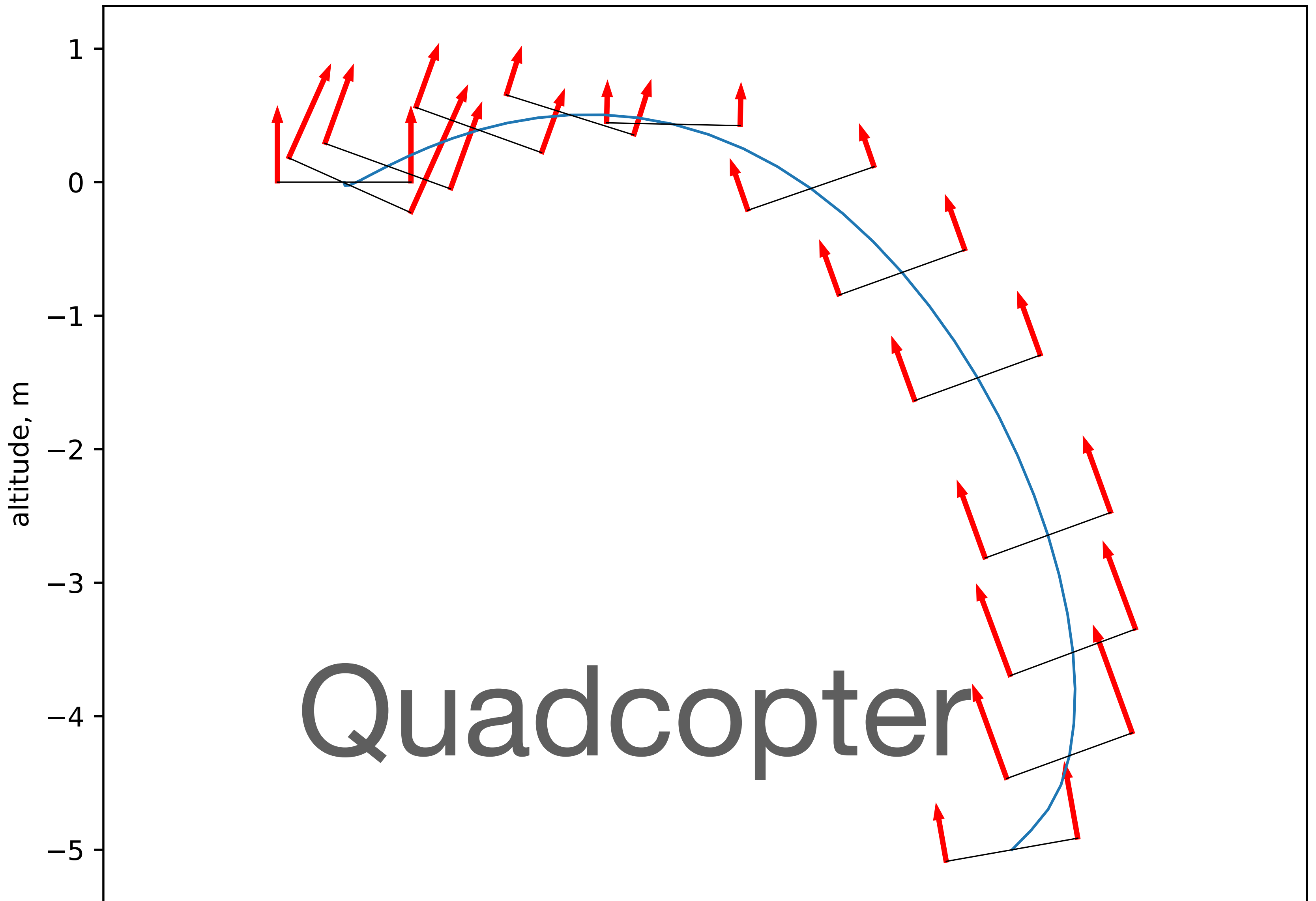
Multidisciplinary Design Optimization (MDO): with dynamical system & control and machine learning

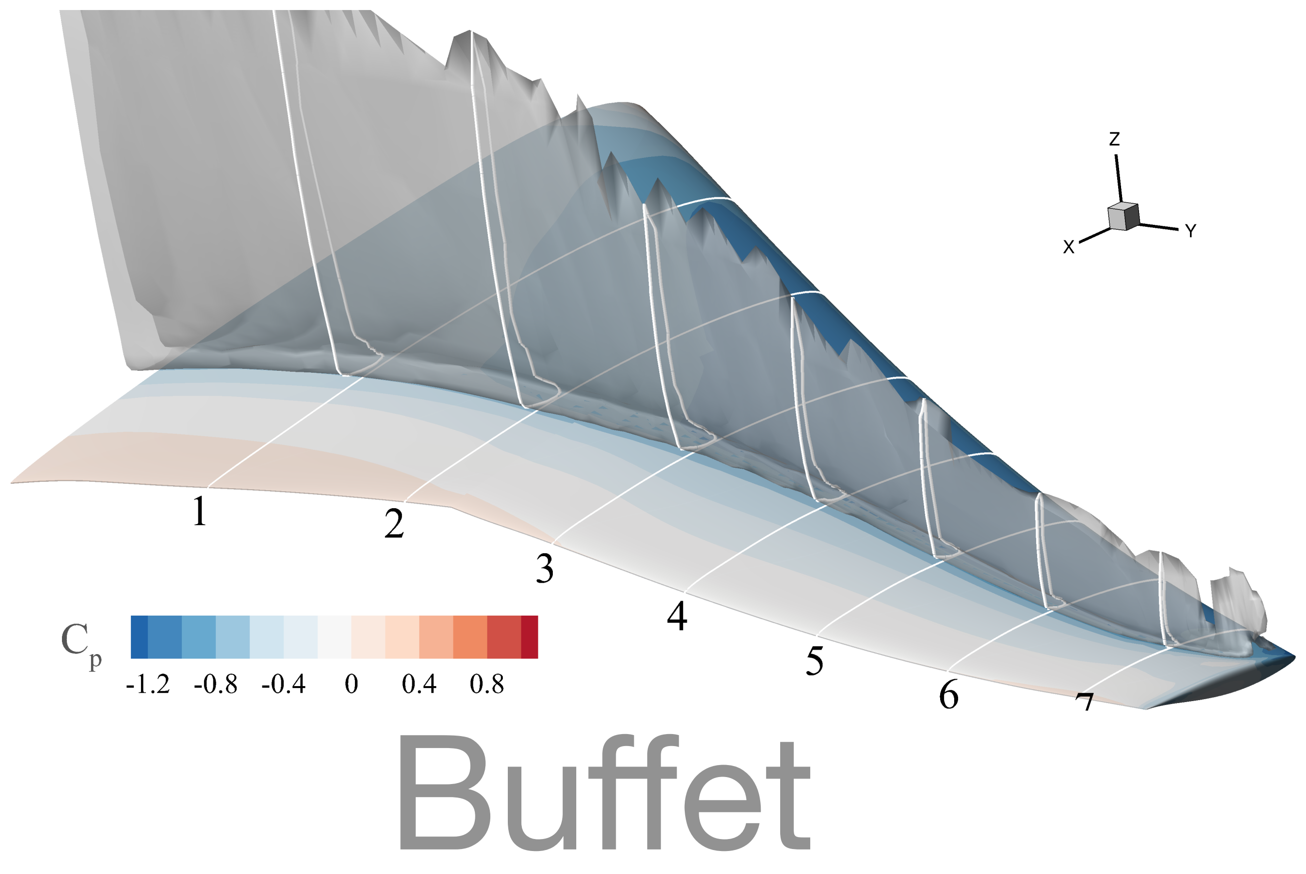




Sicheng He

with contribution from Marco Mangano, Shugo Kaneko, and Jichao Li





PostDoc Associate

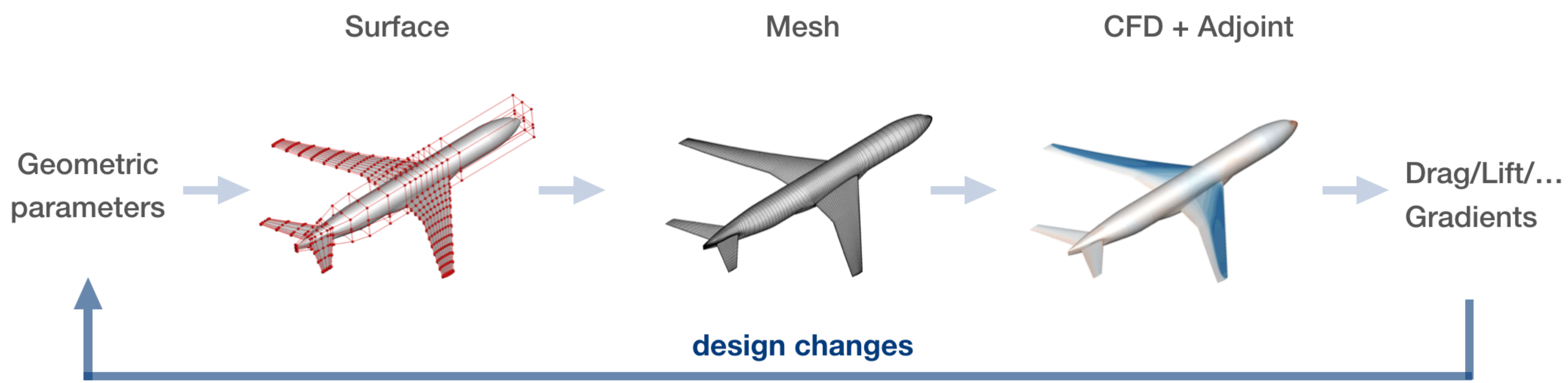
LAE seminar

03/02/2022

MIT LABORATORY FOR AVIATION AND THE ENVIRONMENT

MIT

Challenges in MDO call for innovative solutions.



Design optimization problem:

minimize	$f(x)$	objective
with respect to	x	design variables
subject to	$c(x) \leq 0$	constraints

- Complex dynamical system phenomena beyond equilibrium points.
- Sequential plant and control design resulting in suboptimal design.
- Expensive CFD simulations.

- MDO with general dynamical systems
- Control co-design (CCD)
- Data-driven MDO

Content

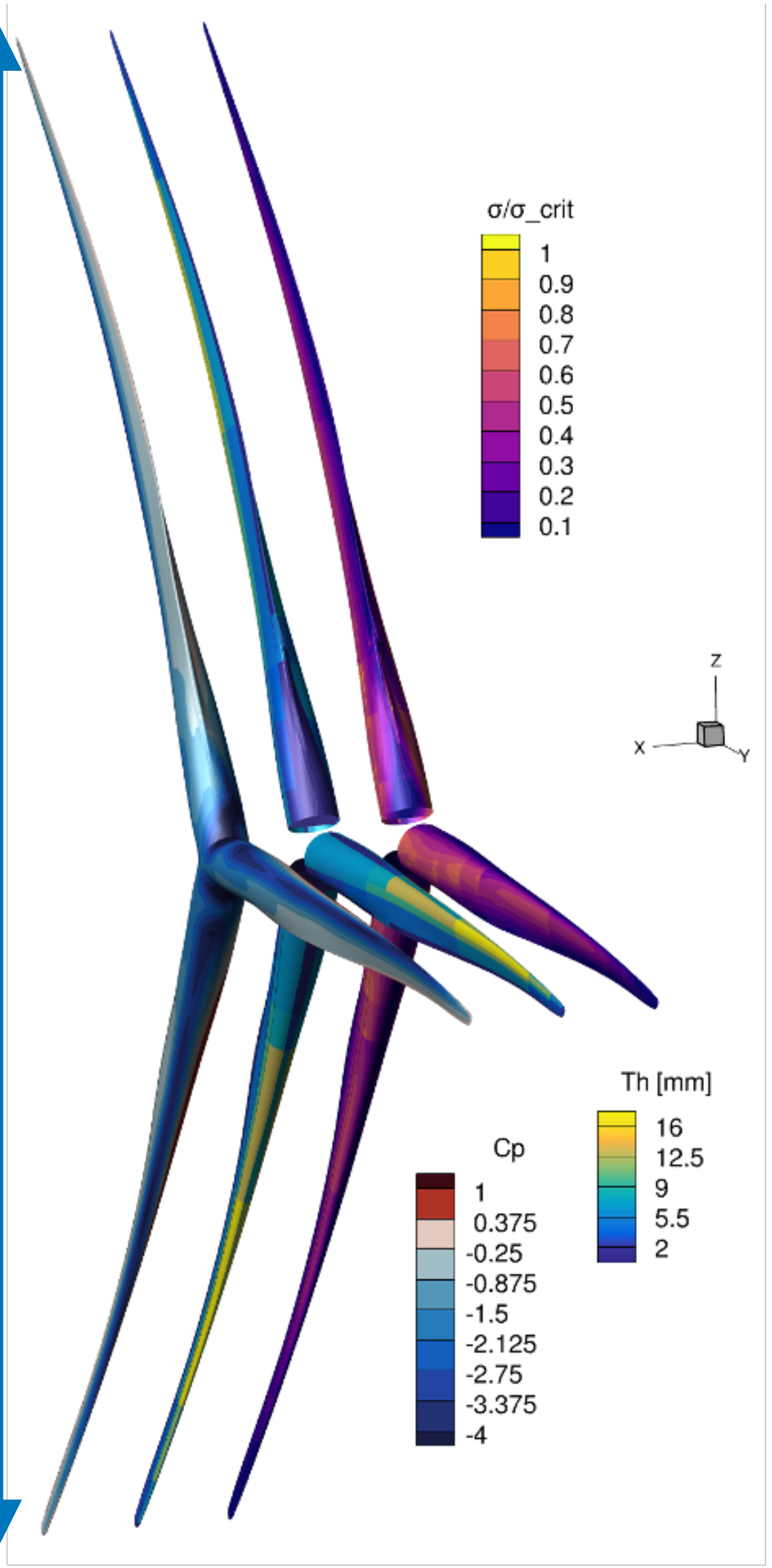
1. **MDO with general dynamical systems**
   - Incorporating more physics in design.
   - Develop efficient adjoint methods (one dynamical system-one adjoint).

2. **Control co-design (CCD)**
   - Exploiting control & plant variables interaction.
   - Efficient derivative computation.

3. **Data-driven MDO**
   - Real-time transonic airfoil shape optimization.
   - Speed up expensive simulation.

Design lighter but more powerful wind turbine blades.

180 m

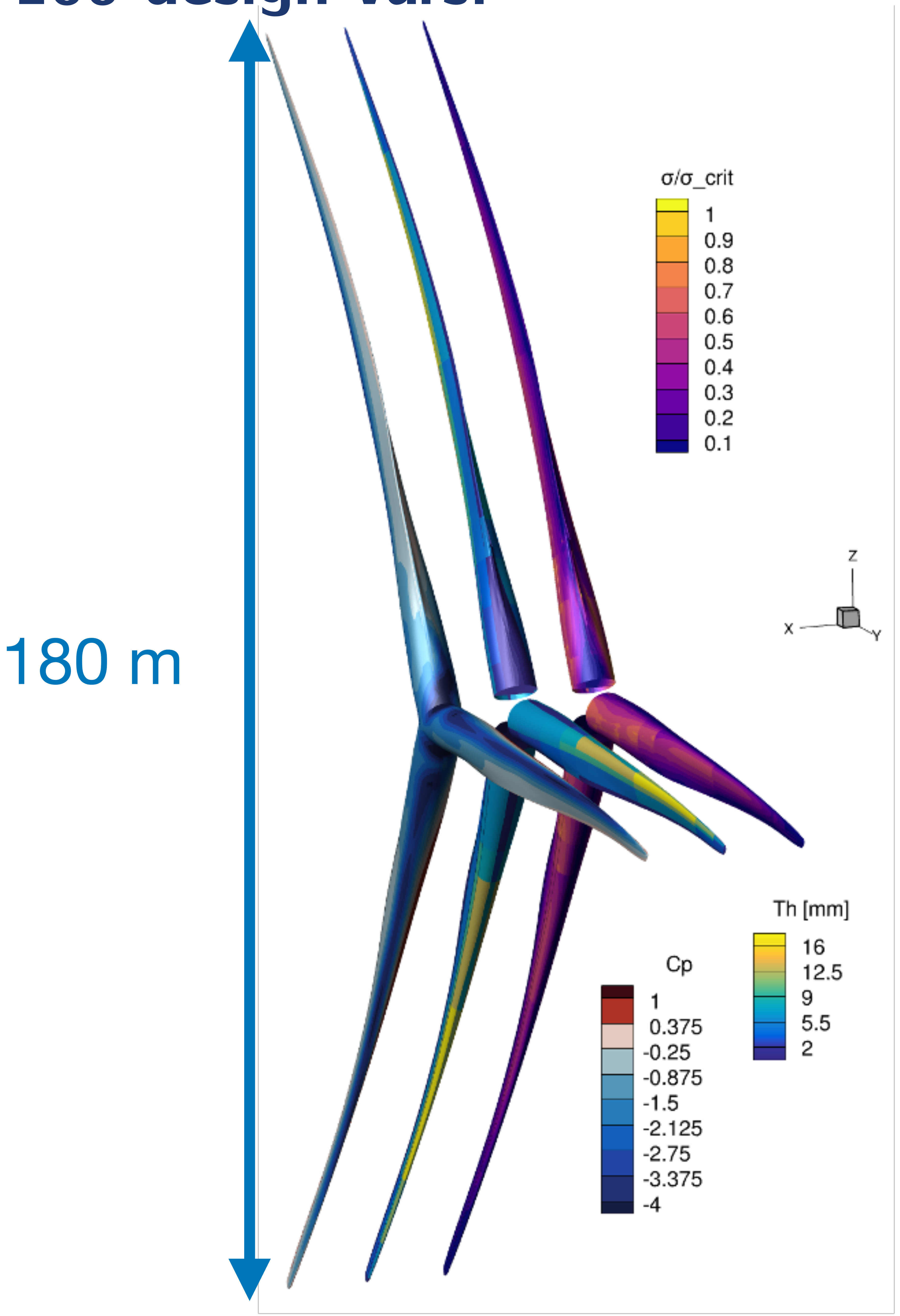


3x 787

ATLANTIS project

arpa·e

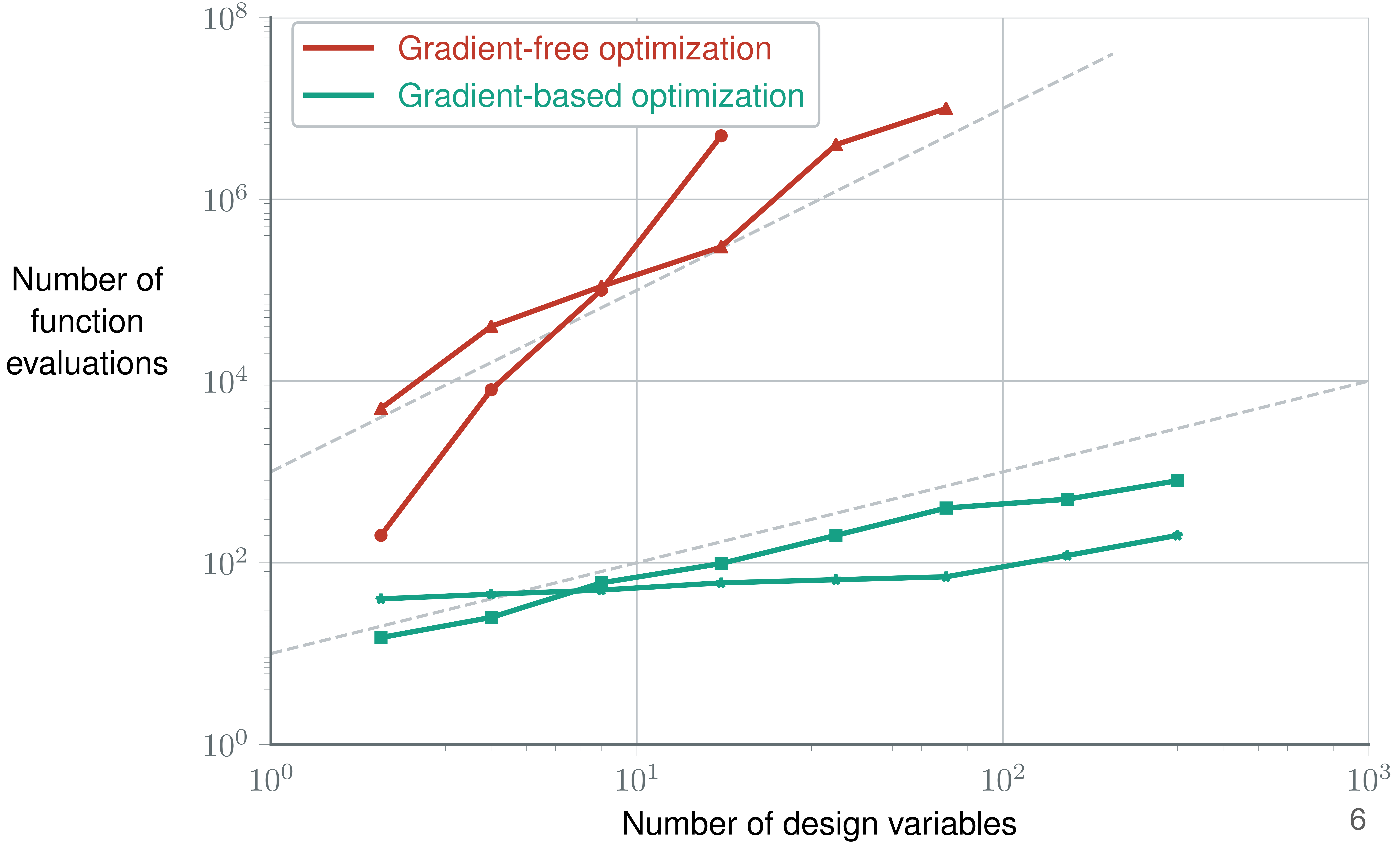
We optimize the wind turbine with about millions of state vars and 100 design vars.



- 1.7 million CFD cells.
- 0.2 million structural cells.
- 100 design variables.

Aerostructural Optimization			
	Name	Symbol	Qty
Objectives	Torque	Q	1
	Mass	M	1
Design Variables	Panel thickness	x_{st}	100+
	Twist	x_{θ}	7
	Chord	x_{ch}	7
	Thickness	x_{tk}	7
	Airfoil shape	x_{sh}	+70
	...	...	...
Constraints	Max Stress	$\mathrm{KS}_{\sigma_{\max}} \leqslant 1$	3
	Thrust	$F_x \leqslant F_{x_{\mathrm{ref}}}$	1
	Tip Displacement	$\mathrm{KS}_{\mathrm{disp}} \leqslant 1$	1
	Torque†	$Q_x \geqslant Q_{x_{\mathrm{ref}}}$	1
	Adjacency constraints		318
	Buckling	$\mathrm{KS}_{\mathrm{buck}} \leqslant 1$	2

We use the gradient-based optimization method (SNOPT) where the sensitivity is computed using the ADjoint method.



[Lyu et al, *2014*]

Sensitivity is computed using the ADjoint method

(almost) $\mathcal{O}(1)$ computational cost with respect to number of design variables.

$$
\begin{aligned}
\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x}} &= \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}},\\
\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}\mathbf{x}} &= \frac{\partial \mathbf{r}}{\partial \mathbf{x}} + \frac{\partial \mathbf{r}}{\partial \mathbf{u}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\mathbf{x}} = 0.
\end{aligned}
$$

$$\Downarrow$$

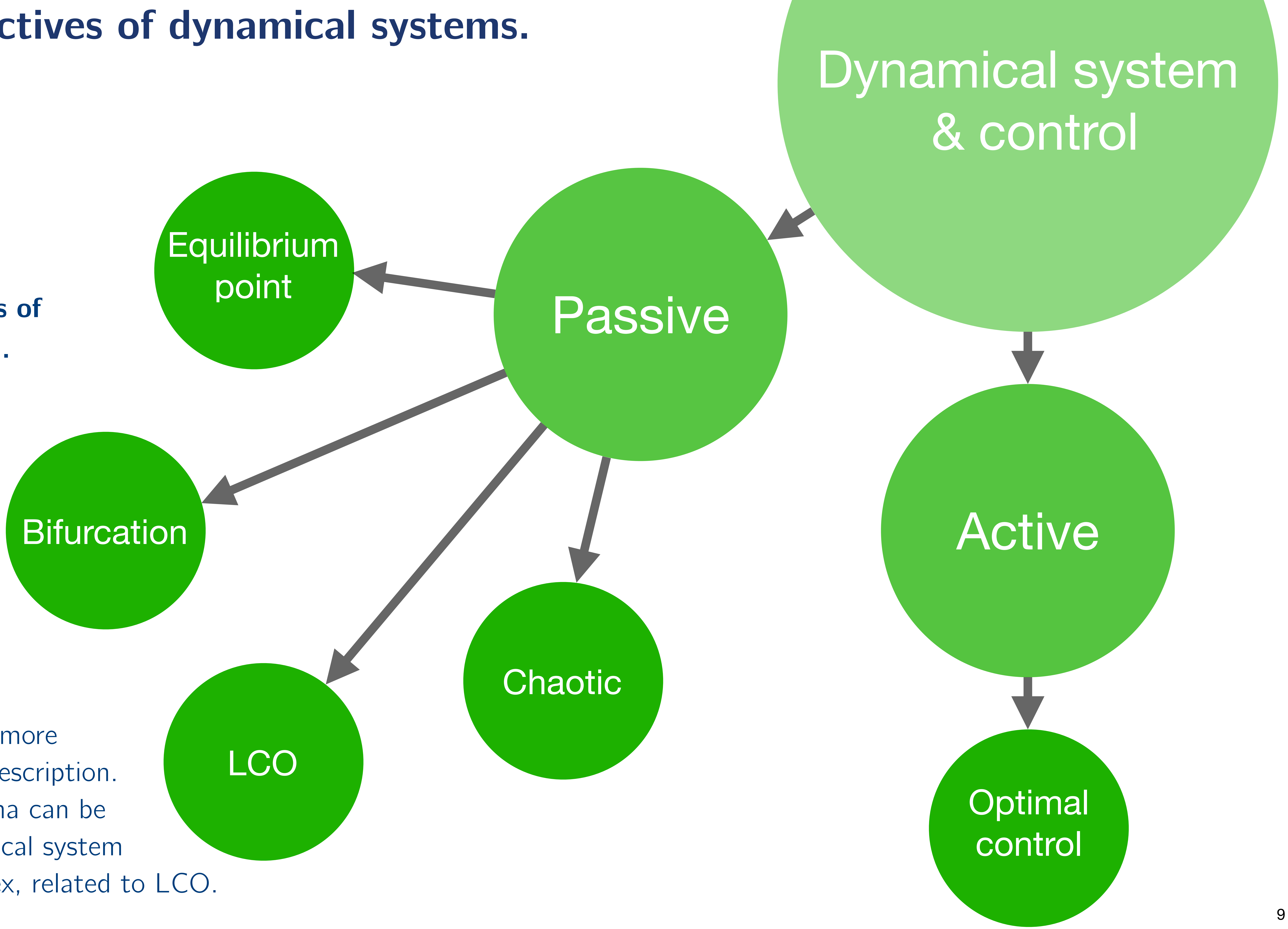
$$
\underset{(n_f \times n_x)}{\frac{\mathrm{d}f}{\mathrm{d}x}} = \underset{(n_f \times n_x)}{\frac{\partial f}{\partial x}} - \underbrace{\underset{(n_f \times n_u)}{\frac{\partial f}{\partial u}} \overbrace{\underset{(n_u \times n_u)}{\frac{\partial r}{\partial u}^{-1}} \underset{(n_u \times n_x)}{\frac{\partial r}{\partial x}}}^{\phi \quad (n_u \times n_x)}}_{\psi^{\intercal} \quad (n_f \times n_u)}
$$

We conduct the *first* aerostructural optimization of offshore wind turbines.

[Mangano, **SH**, et al., *Scitech 2022*]

MDO from perspectives of dynamical systems.

The underlying constraints of an MDO problem can be...



- Fundamental math and is more general than the physics description.
- Multiple physics phenomena can be categorized as one dynamical system e.g., flutter, Karman vortex, related to LCO.

The challenge / opportunity for MDO with dynamical systems ...

1. New engineering applications.

   e.g., wind energy, automobile, robotics, etc.

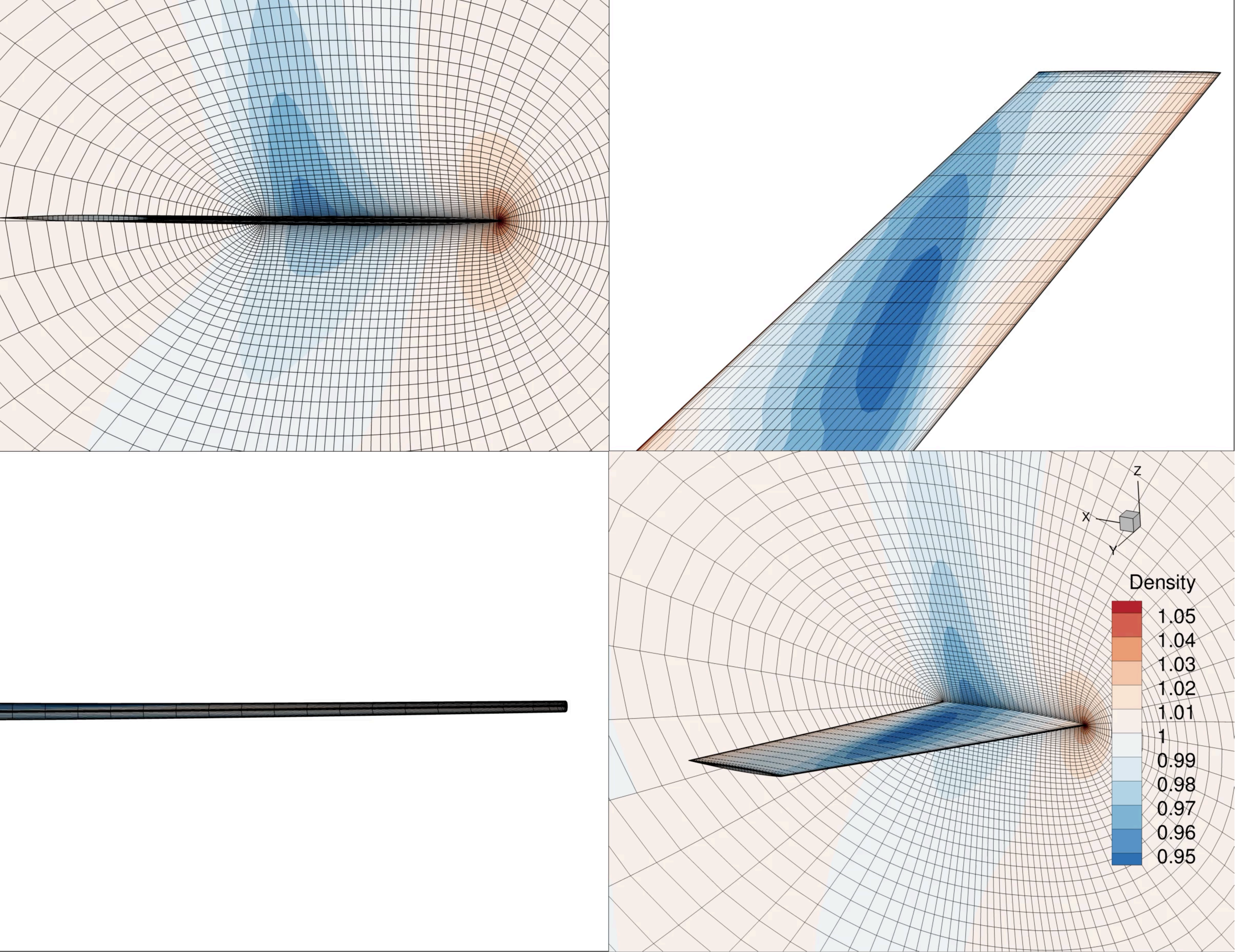
2. Incorporate more and higher fidelity physics.

   e.g., laminar-turbulent transition (bifurcation/chaos), flutter (LCO).

3. Larger scale problems (efficient prime/adjoint solvers).

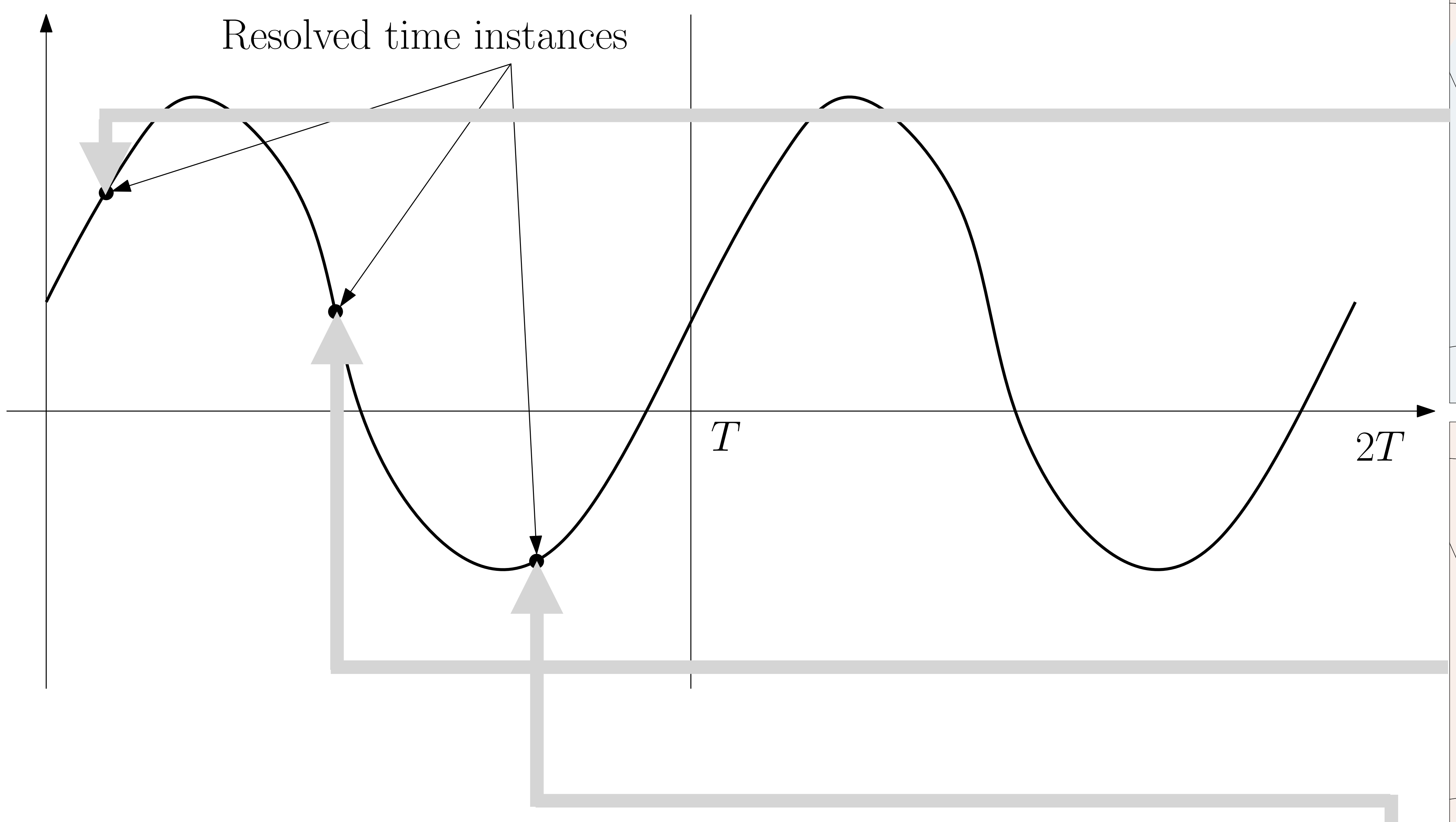
Aircraft tail flutter

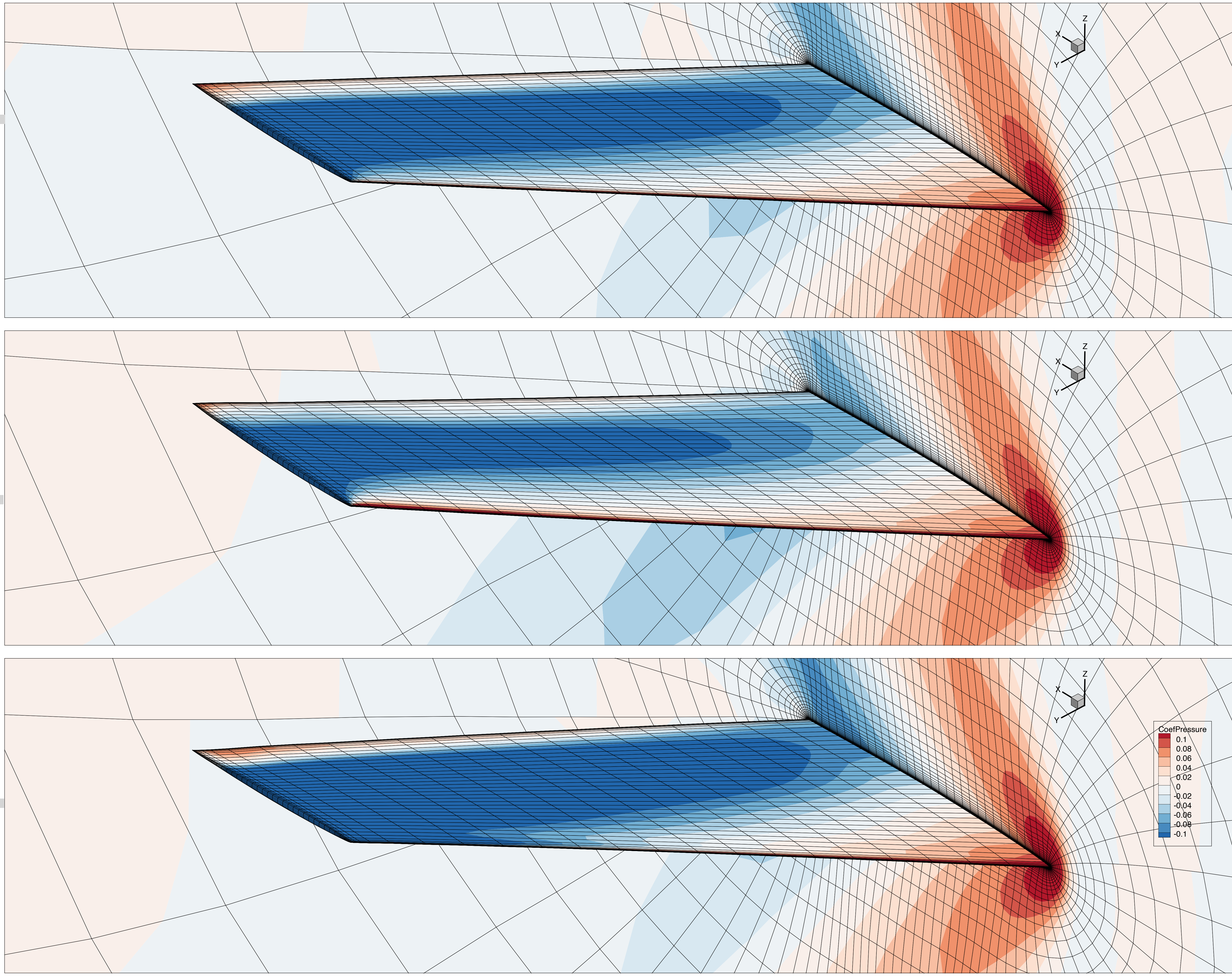
Numerical simulation of flutter.



- Time spectral method CFD / CSD.
- Time spectral LCO equation is solved.

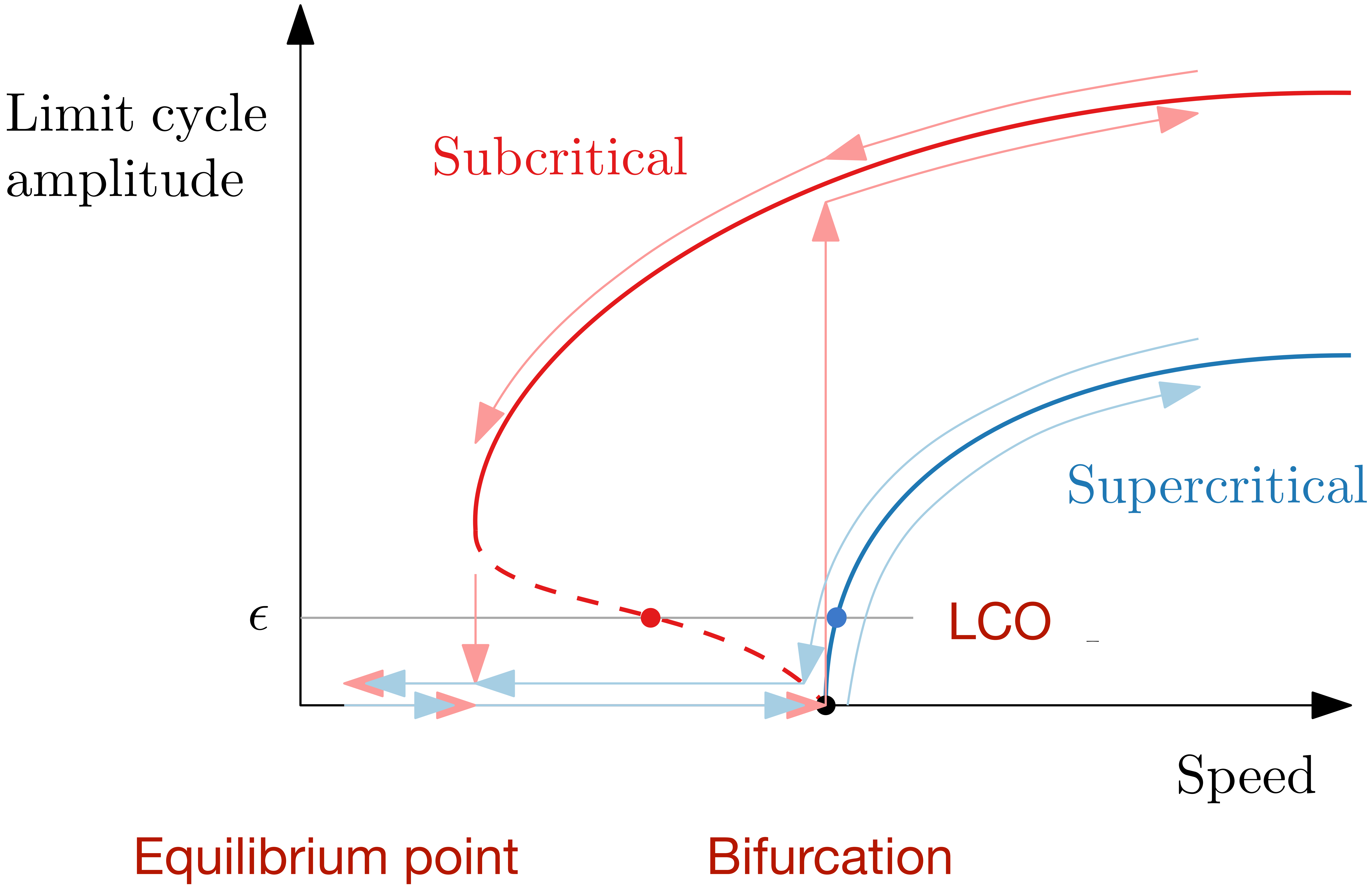
Time spectral method is an efficient way to model time-periodic systems.





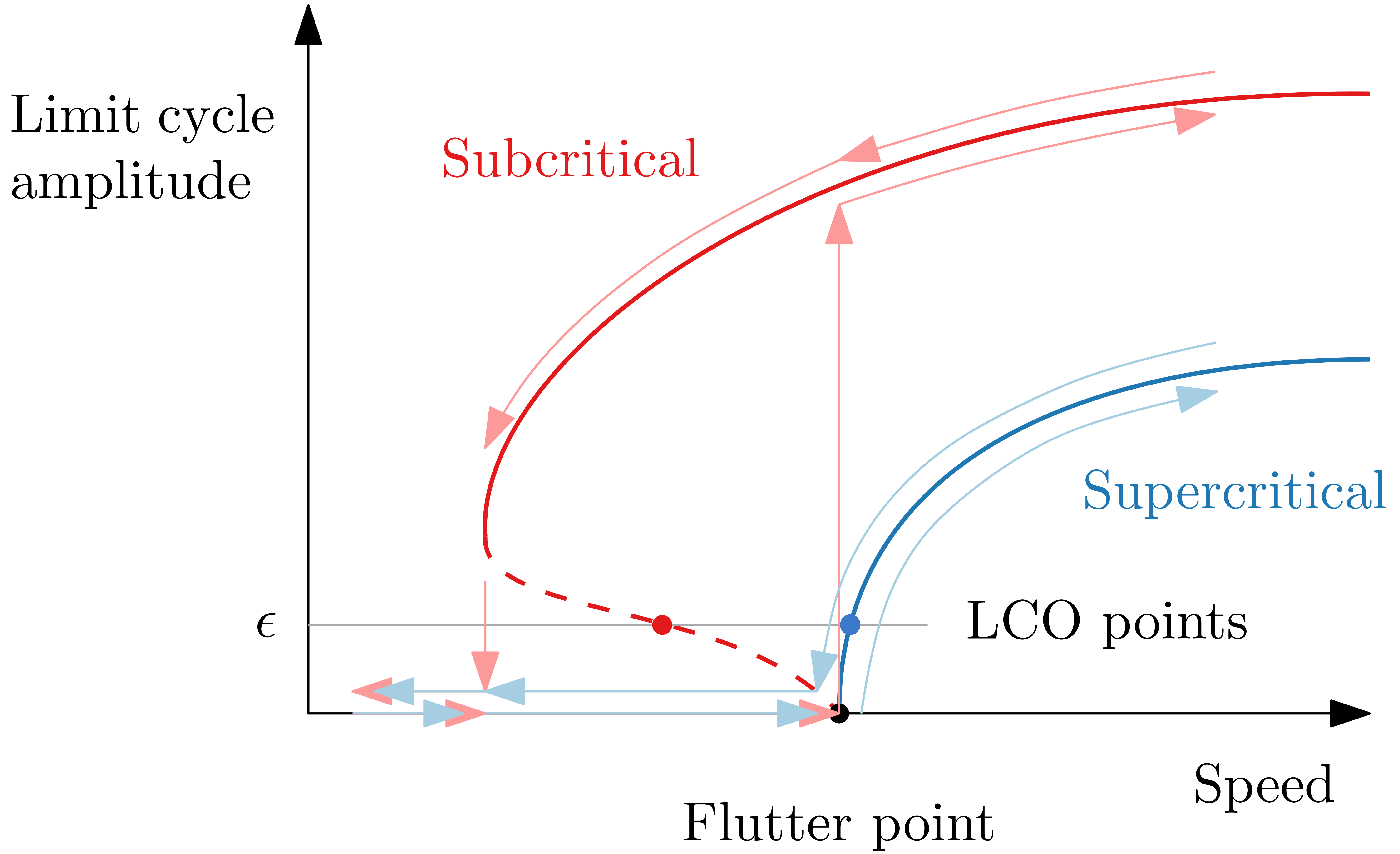
- Leverage DFT.
- Capture k freq components with 2k + 1 instances.
- 10x faster than time accurate method because transient response is avoided.

Wing can demonstrate complex dynamical system behavior.



- Different inflow speeds will induce different dynamical system behaviors.
- Bifurcation and LCO can be stable or unstable.

LCO is challenging to simulate because the speed and frequency are not known *a priori*, but we can solve them by imposing a motion.



Equations:

$$\mathcal{R} = \begin{bmatrix} \mathcal{R}_{\mathrm{mag}} \\ \mathcal{R}_{\mathrm{phi}} \\ \mathcal{S}_{\mathrm{TS}} \\ \mathcal{A}_{\mathrm{TS}} \end{bmatrix}$$

Motion magnitude
Motion phase
Struct residual
Aero residual

Variables:

$$q = \begin{bmatrix} V_f \\ \omega \\ \bar{\eta}^n \\ \zeta^n \end{bmatrix}$$

LCO speed
LCO frequency
Struct displacement
Aero state var

[Thomas, et al. *AIAAJ 2002*]
[**SH**, et al. *AIAAJ 2020*]
[**SH**, et al. *Aviation 2019*]

We use Jacobian-free Newton—Krylov method from the PETSc + PETSc4py packages to solve the aeroelastic equation

Solve time-spectral aeroelastic equation with Newton method implemented using PETSc

$$J\Delta q = -r(q^{(k)})$$
$$q^{(k+1)} = q^{(k)} + \theta \Delta q'$$

Directly interface with the whole system

$$R = \begin{bmatrix} \mathbf{R}_m \\ \mathbf{R}_p \\ \mathbf{S}_{\mathrm{TS}} \\ \mathbf{A}_{\mathrm{TS}} \end{bmatrix}, q = \begin{bmatrix} V_f \\ \omega \\ \eta^n \\ \zeta^n \end{bmatrix},$$

- Krylov subspace solver
- Matrix-vector product approximated by FD

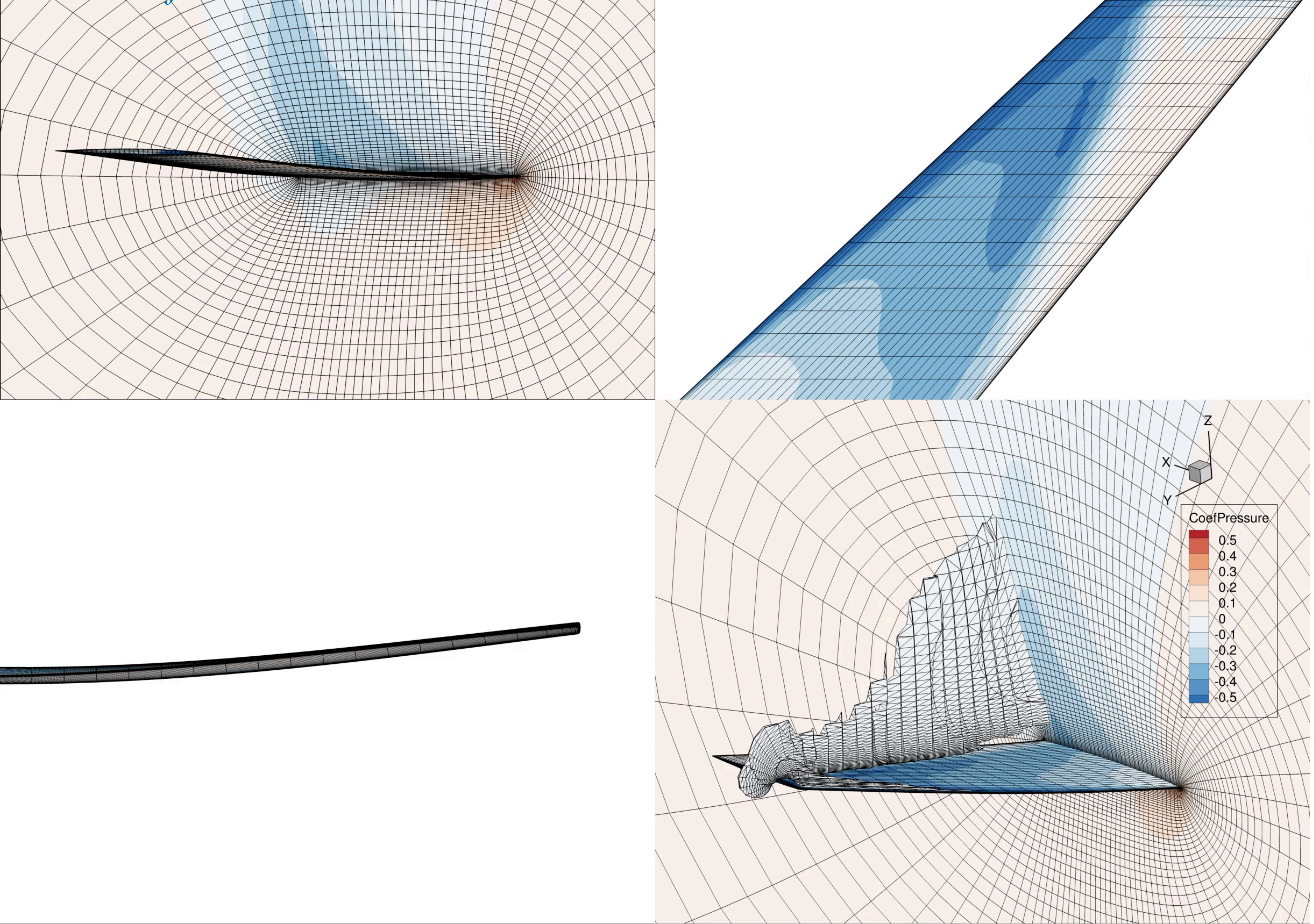
$$Jv \approx \frac{R(q + \epsilon v) - R(q)}{\epsilon}$$

θ is a line search parameter. It is used to give it better global convergence.

We maximize the LCO speed by changing the aerodynamic shape.

	Function/variable	Description	Quantity
maximize	V_f	LCO speed index	
w.r.t	y	FFD control points y coordinates	50
	AOA	angle of attack	1
s.t.	$-0.02\ \text{m} \leq y \leq 0.02\ \text{m}$	bounds on FFD control points y coordinates	50
	$0^\circ \leq \text{AOA} \leq 10^\circ$	bounds AOA y coordinates	1
	$\overline{C}_L = 0.3$	lift constraint	1
	$t \geq 0.75 t_0$	thickness constraint	25
	$V_0 \leq V \leq 10 V_0$	volume constraint	1
	$y_{\text{upper}} = -y_{\text{lower}}$	symmetric leading/trailing edge constraints	10

We get the *first* 3D gradient-based LCO optimization results with time spectral method where V_f increased by 118%.



Function	Baseline	Optimized
V_f	2.78040×10^{-1}	6.04802×10^{-1}
$\overline{C}_l$	3.15069×10^{-1}	3.00165×10^{-1}
V	V_0	$1.00000V_0$

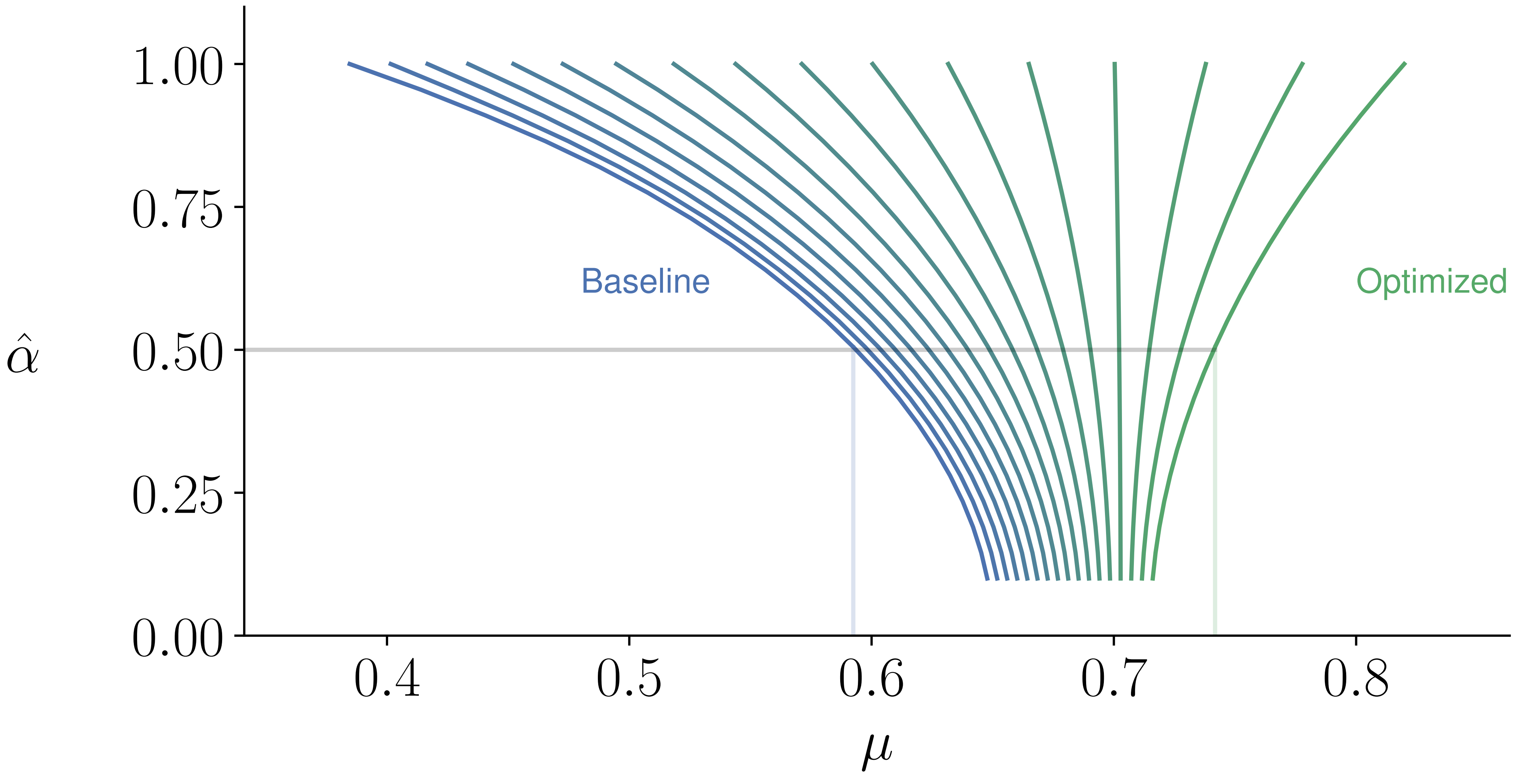
[**SH**, et al., *Aviation 2022*]

Load is shifted inboard passive load alleviation.

Other dynamical systems ...

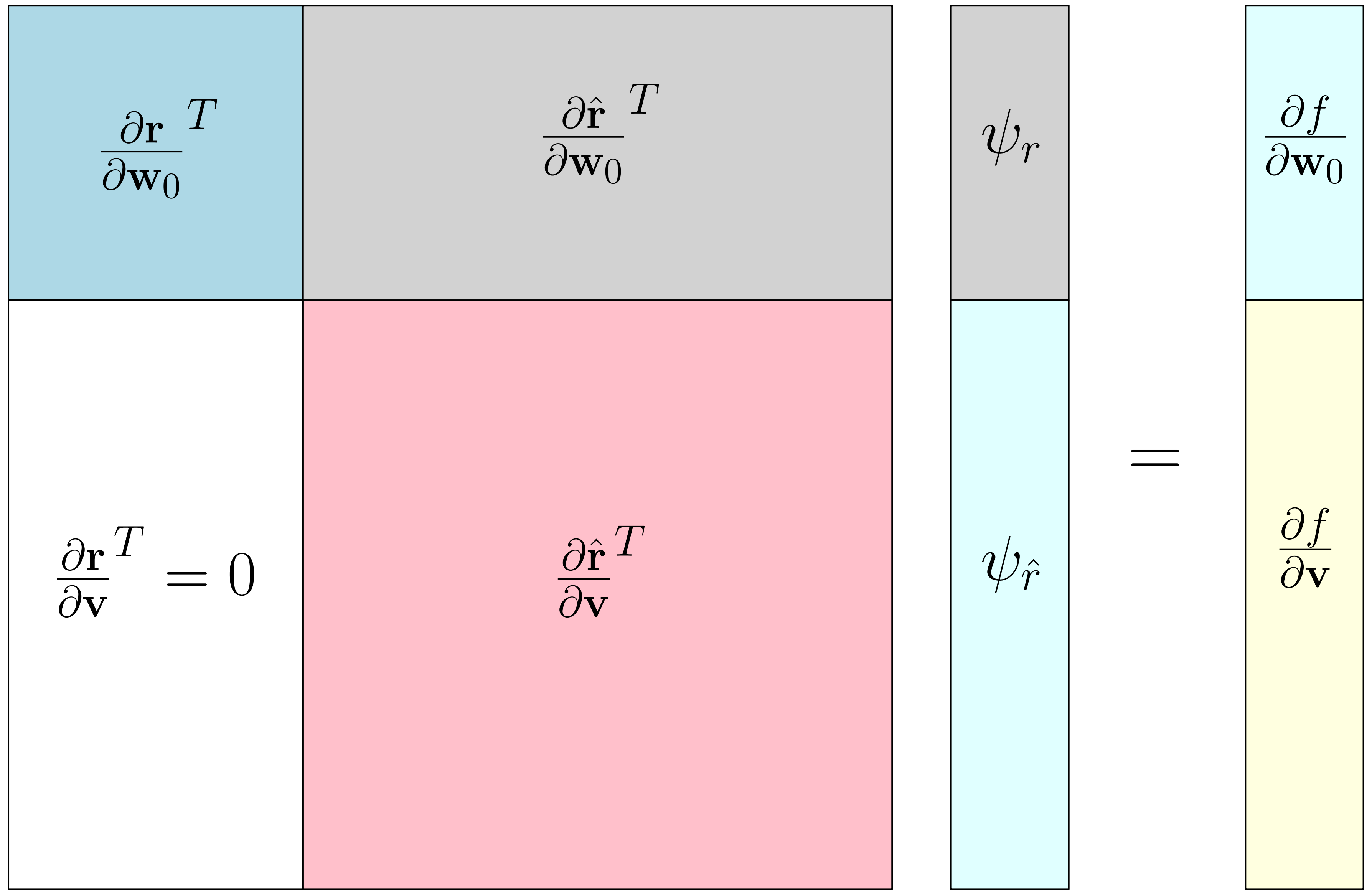
- Stabilizing unstable LCO (subcritical).
- Time spectral method adjoint.
- Fitted curve.

[**SH**, et al. *NODY 2022*]



- Suppressing bifurcation.
- Efficient eigenvalue derivative.
- Block backsubstitution adjoint solver.

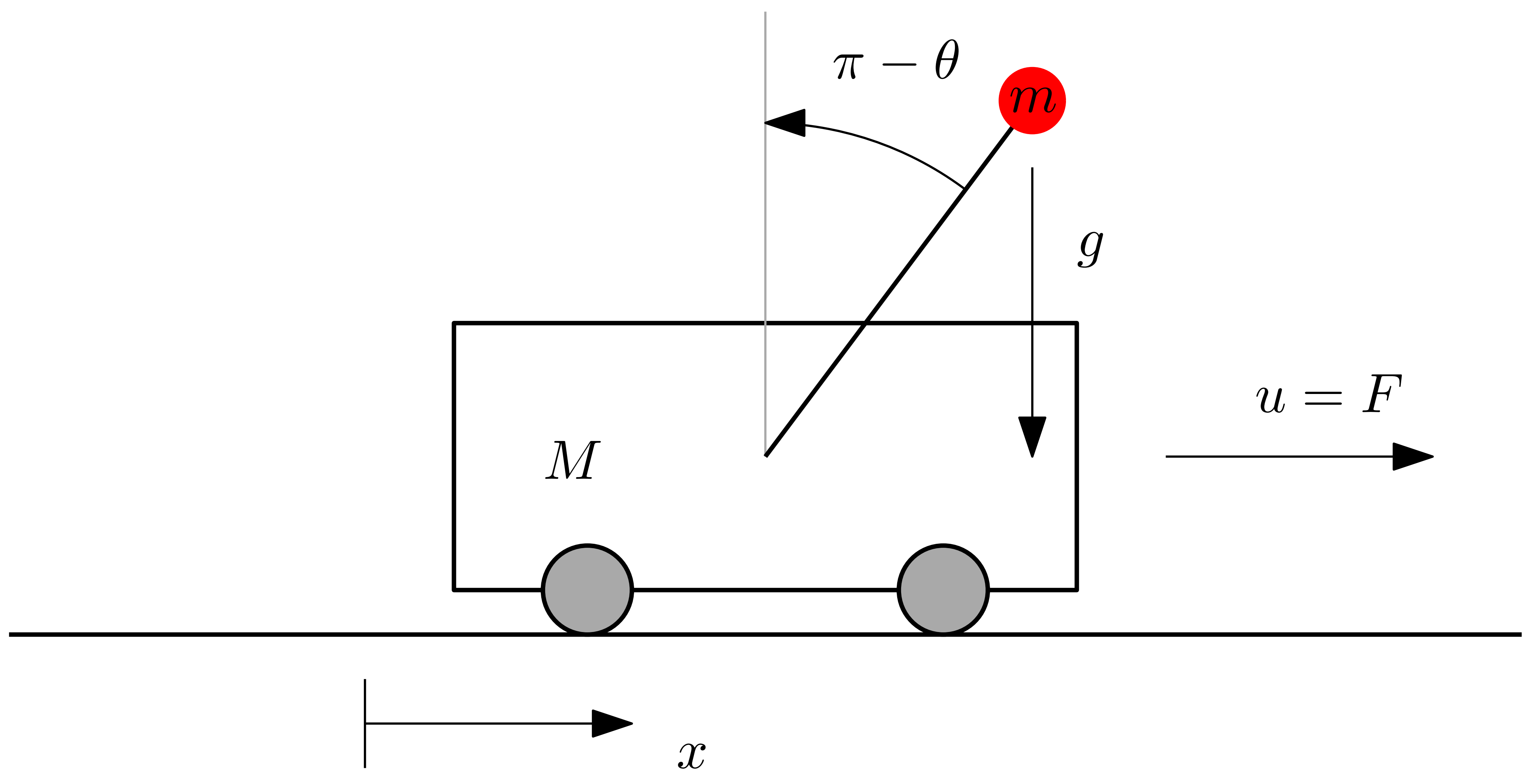
[**SH**, et al. *MSSP 2023*]



Content

1. **MDO with general dynamical systems**
    - Incorporating more physics in design.
    - Develop efficient adjoint methods (one dynamical system-one adjoint).
2. **Control co-design (CCD)**
    - Exploiting control & plant variables interaction.
    - Efficient derivative computation.
3. **Data-driven MDO**
    - Real-time transonic airfoil shape optimization.
    - Speed up expensive simulation.

Cartpole problem as an example...



$$\dot{\mathbf{x}} = \mathbf{r}_{\text{nl}}(\mathbf{x}, \mathbf{u}, \mathbf{d})$$

$\mathbf{r}_{\text{nl}}$: dynamics

$\mathbf{x}$: state variable

$\mathbf{u}$: control variable

$\mathbf{d}$: plant variable

1. Optimal control:

   Under fixed plant variable, $\mathbf{d}$, find a control variable, $\mathbf{u}$, to minimize a control cost, f

   $$f^*(\mathbf{d}) = \min_{\mathbf{u}} f(\mathbf{u}, \mathbf{d})$$

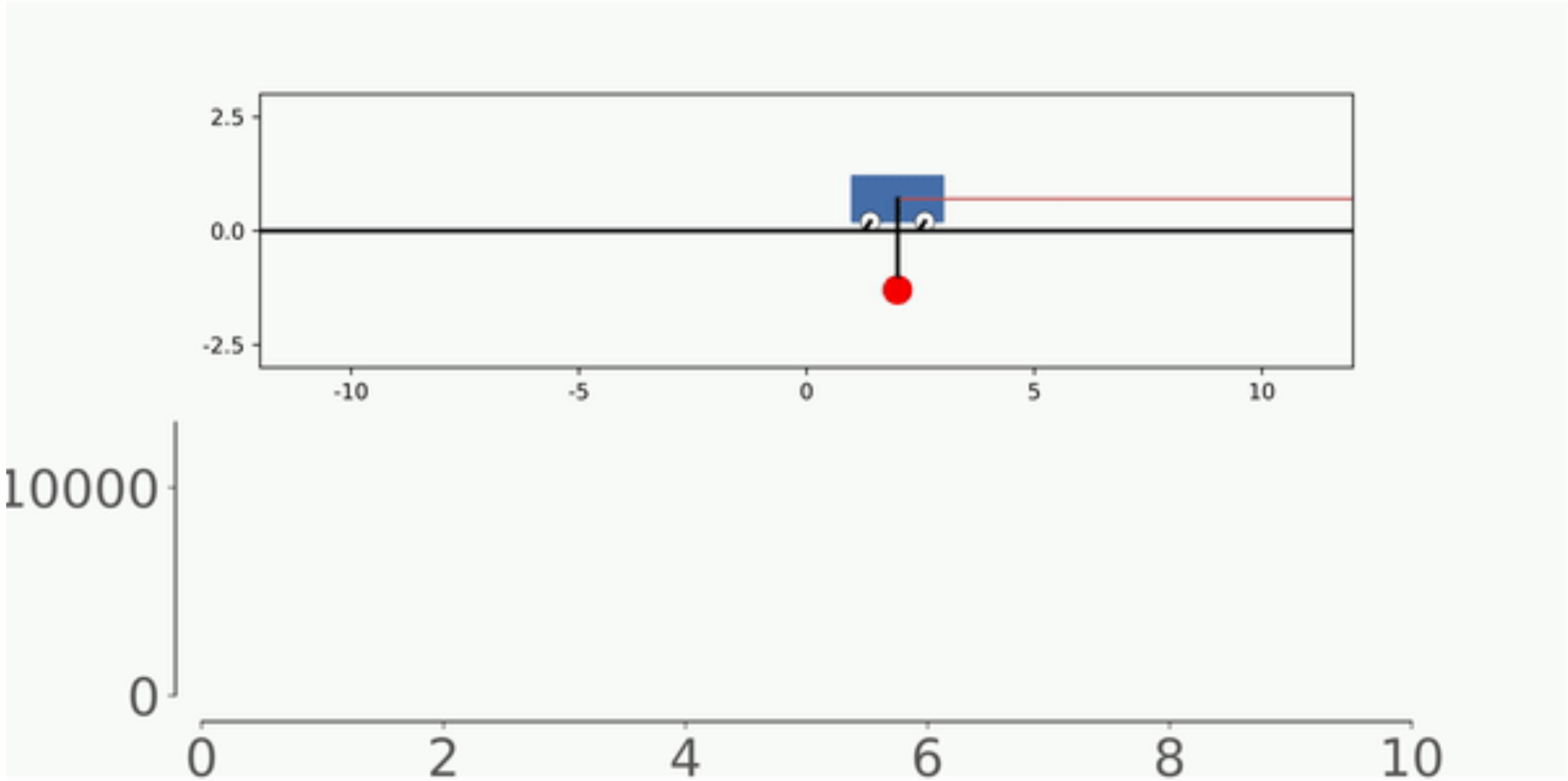
2. CCD:

   Find the plant variable, $\mathbf{d}$, to minimize the optimal control cost, f^*

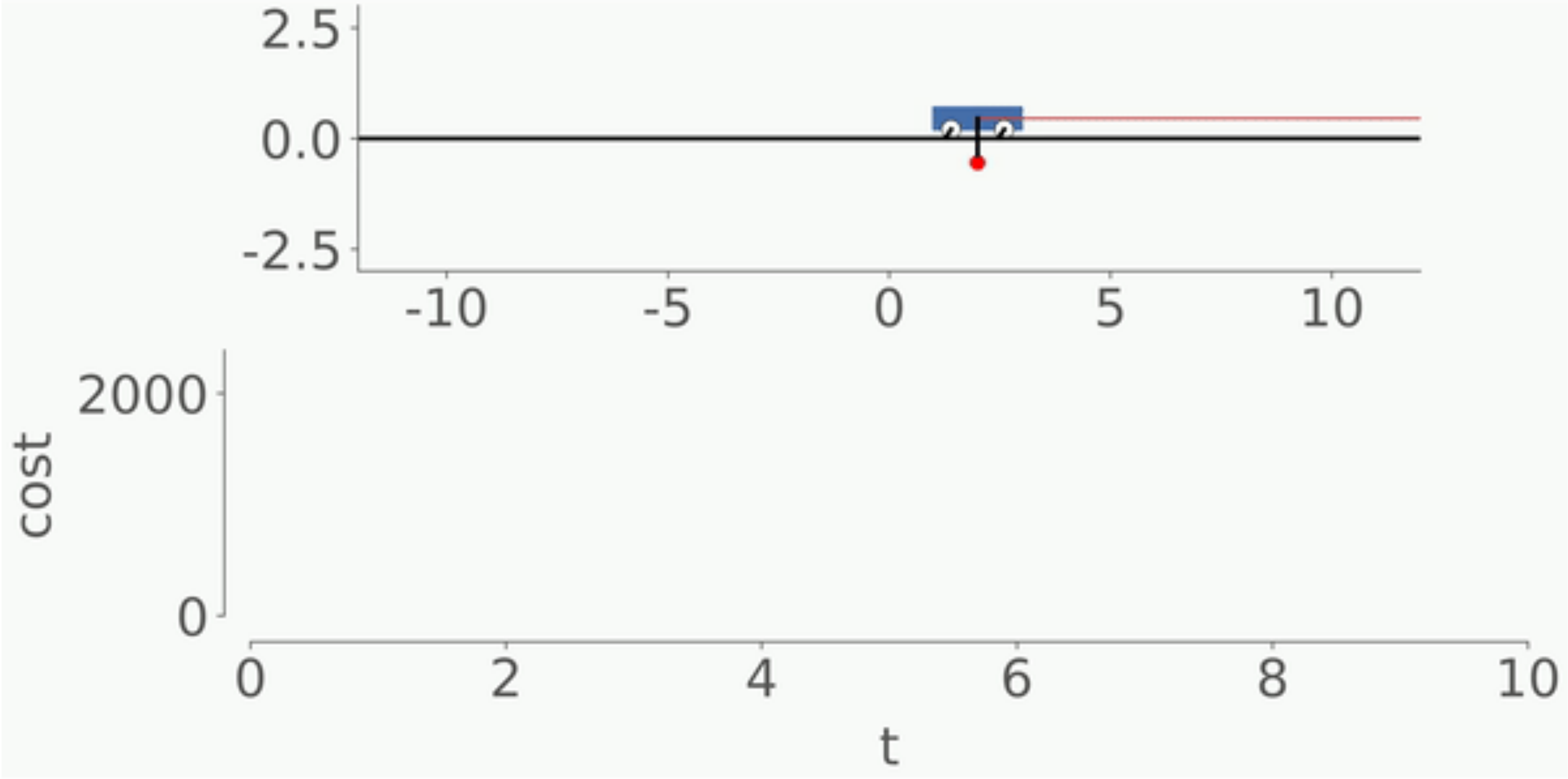
   $$f^{**} = \min_{\mathbf{d}} f^*(\mathbf{d})$$

The optimizer wants to have less mass for both carts and short bar...

Baseline



Optimized

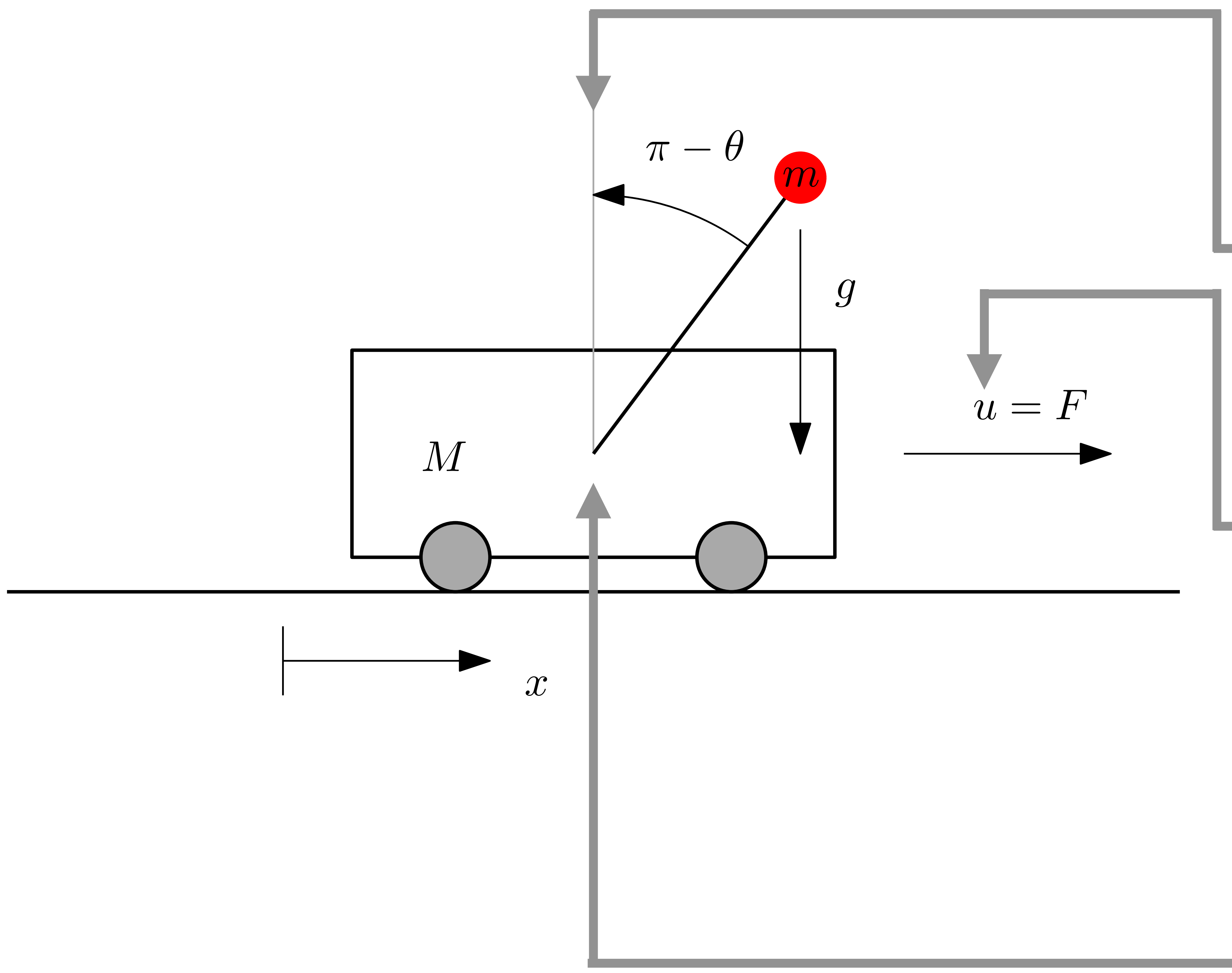


The challenge for CCD ...

1. Large scale problems (efficient solvers).
2. Large scale optimization problems (efficient derivative computation method).
3. Advanced control strategy.

   e.g., model predictive control (MPC) etc.

Three modules for the CCD problem: equilibrium point, optimal control, and closed loop dynamics.



Equilibrium point

$$\mathbf{r}_{\mathrm{nl}}(\mathbf{x}, \mathbf{u}, \mathbf{d}) = \mathbf{0}$$

Optimal control with LQR (ARE)

$$\mathbf{u} = \mathbf{W}\mathbf{x}$$

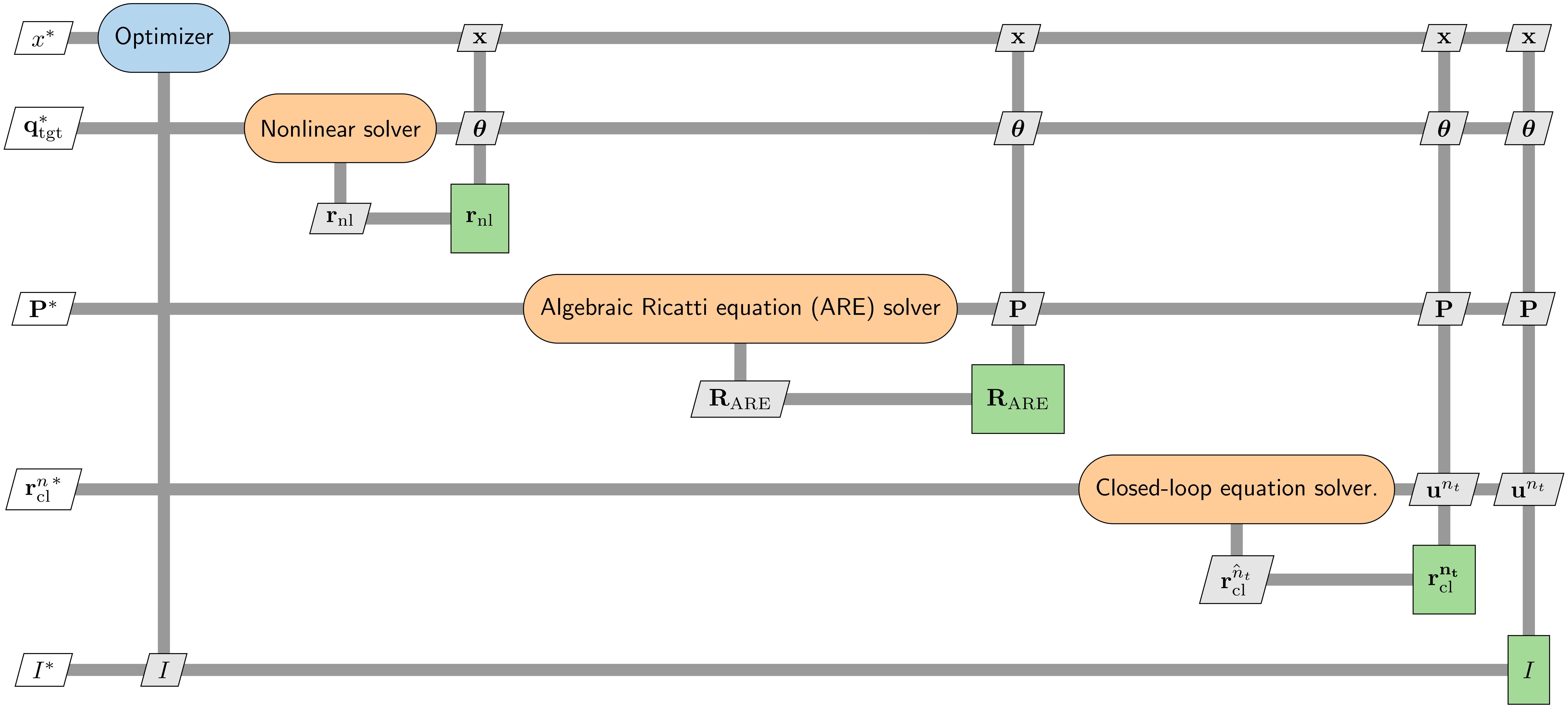
$$\mathbf{J}_{\mathrm{tgt}}^{\intercal}\mathbf{P} + \mathbf{P}\mathbf{J}_{\mathrm{tgt}} - \mathbf{P}\mathbf{G}_{\mathrm{tgt}}\mathbf{S}^{-1}\mathbf{G}_{\mathrm{tgt}}^{\intercal}\mathbf{P} + \mathbf{Q} = \mathbf{O}$$

$$\mathbf{W} = -\mathbf{S}^{-1}\mathbf{G}_{\mathrm{tgt}}^{\intercal}\mathbf{P}$$

Closed loop dynamics

$$\dot{\mathbf{x}} = \mathbf{r}_{\mathrm{nl}}(\mathbf{x}, \mathbf{W}\mathbf{x}, \mathbf{d})$$

Nested control co-design (CCD) with efficient derivative computation.



Solving the coupled adjoint equation efficiently leveraging feed-forward structure

Analysis

Equilibrium point

$$\mathbf{r}_{\mathrm{nl}}(\mathbf{x}_{\mathrm{tgt}}, \mathbf{u}_{\mathrm{tgt}}, \mathbf{d}) = \mathbf{0}.$$

Optimal control, Riccati equation

$$\mathbf{R}_{\mathrm{ARE}}(\mathbf{P}; \mathbf{J}_{\mathrm{tgt}}, \mathbf{G}_{\mathrm{tgt}}, \mathbf{Q}, \mathbf{S}) = \mathbf{J}_{\mathrm{tgt}}^{\intercal}\mathbf{P} + \mathbf{P}\mathbf{J}_{\mathrm{tgt}} - \mathbf{P}\mathbf{G}_{\mathrm{tgt}}\mathbf{S}^{-1}\mathbf{G}_{\mathrm{tgt}}^{\intercal}\mathbf{P} + \mathbf{Q} = \mathbf{O}$$

Forward closed loop system

$$\mathbf{r}_{\mathrm{cl}}^{n_t} = \begin{bmatrix} \mathbf{I} & \mathbf{O} & \cdots & \mathbf{O} & \mathbf{O} \\ -\mathbf{I} & \mathbf{I} & \cdots & \mathbf{O} & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{I} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \cdots & -\mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}^{(1)} \\ \delta\mathbf{x}^{(2)} \\ \vdots \\ \delta\mathbf{x}^{(n-1)} \\ \delta\mathbf{x}^{(n)} \end{bmatrix} + \begin{bmatrix} -\mathbf{I}\delta\mathbf{x}^{(0)} - \mathbf{r}_{\mathrm{nl}}(\mathbf{x}_{\mathrm{tgt}} + \delta\mathbf{x}^{(0)}, \mathbf{u}_{\mathrm{tgt}} + \mathbf{W}\delta\mathbf{x}^{(0)}, \mathbf{d})\Delta t \\ -\mathbf{r}_{\mathrm{nl}}(\mathbf{x}_{\mathrm{tgt}} + \delta\mathbf{x}^{(1)}, \mathbf{u}_{\mathrm{tgt}} + \mathbf{W}\delta\mathbf{x}^{(1)}, \mathbf{d})\Delta t \\ \vdots \\ -\mathbf{r}_{\mathrm{nl}}(\mathbf{x}_{\mathrm{tgt}} + \delta\mathbf{x}^{(n-2)}, \mathbf{u}_{\mathrm{tgt}} + \mathbf{W}\delta\mathbf{x}^{(n-2)}, \mathbf{d})\Delta t \\ -\mathbf{r}_{\mathrm{nl}}(\mathbf{x}_{\mathrm{tgt}} + \delta\mathbf{x}^{(n-1)}, \mathbf{u}_{\mathrm{tgt}} + \mathbf{W}\delta\mathbf{x}^{(n-1)}, \mathbf{d})\Delta t \end{bmatrix} = \mathbf{0}.$$

Adjoint

Equilibrium point adjoint

$$\frac{\partial \check{\mathbf{r}}_{\mathrm{nl}}}{\partial \boldsymbol{\theta}}^{\intercal} \boldsymbol{\psi}_{\mathrm{nl}} = -\left(\frac{\partial I}{\partial \boldsymbol{\theta}} + \boldsymbol{\psi}_{\mathrm{cl}}^{\intercal} \frac{\partial \mathbf{r}_{\mathrm{cl}}^{n_t}}{\partial \boldsymbol{\theta}} + \frac{\partial \mathrm{Tr}\left(\boldsymbol{\Psi}_{\mathrm{ARE}}^{\intercal}\mathbf{R}_{\mathrm{ARE}}\right)}{\partial \boldsymbol{\theta}} \right).$$

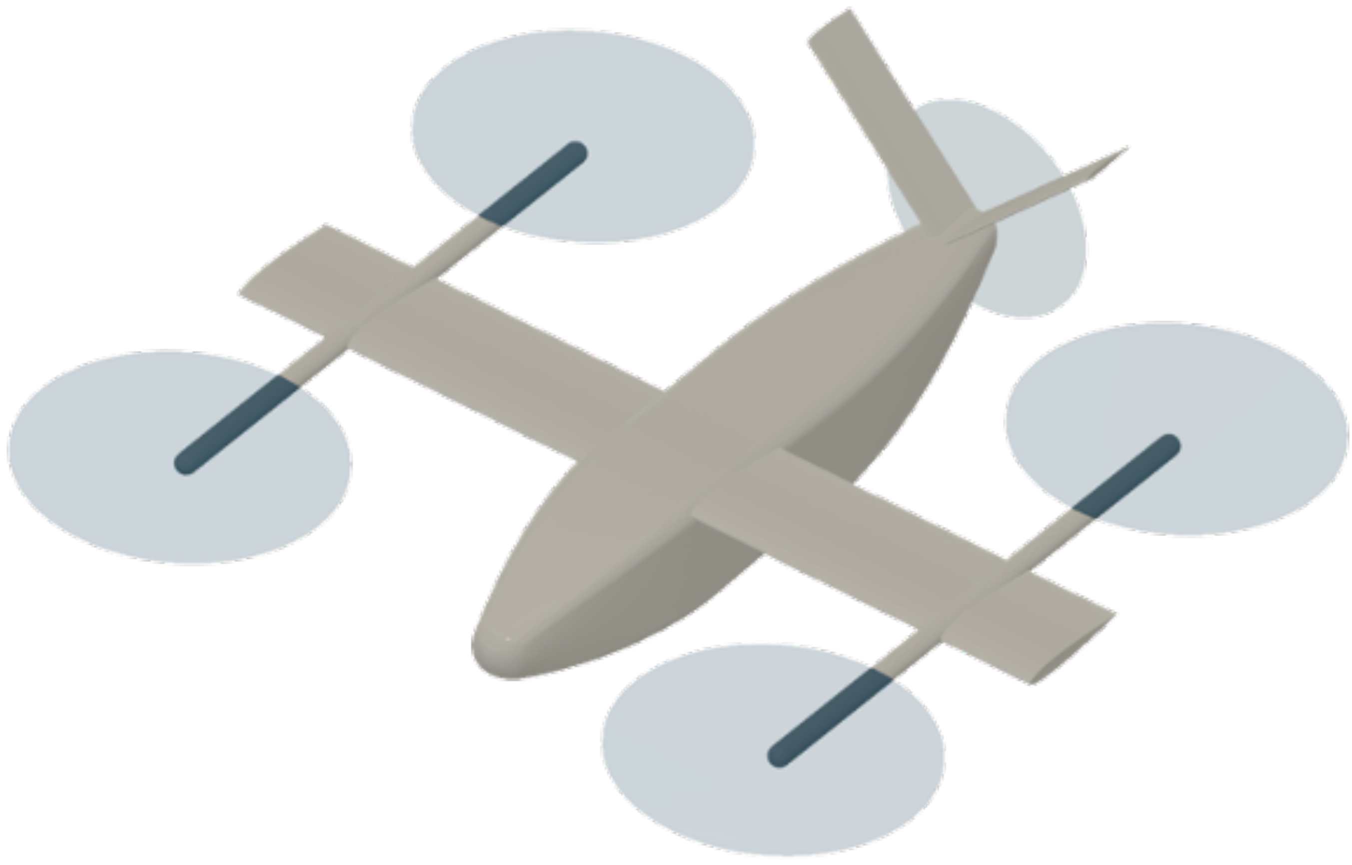
Optimal control adjoint, Lyapunov equation

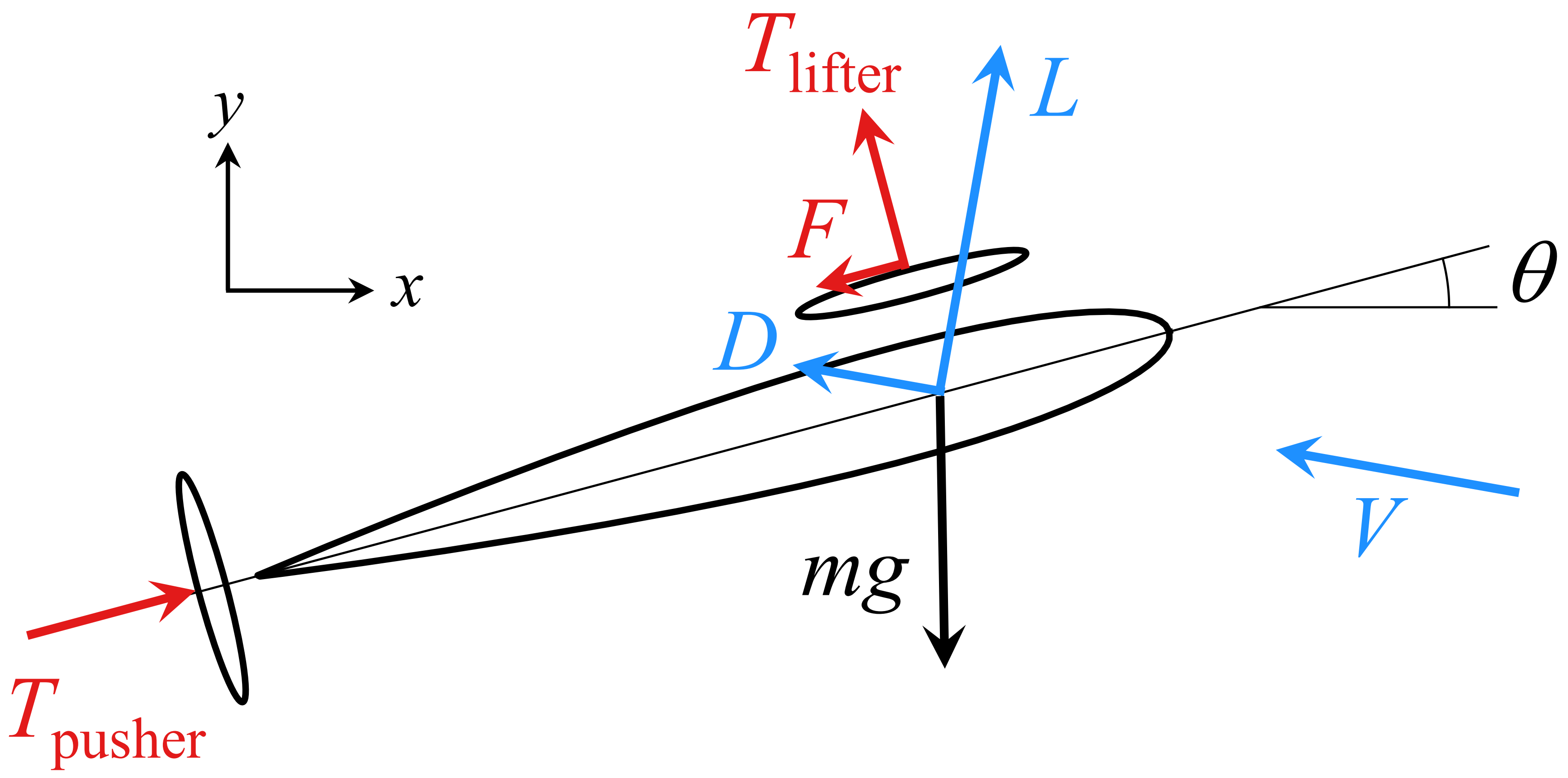
$$\tilde{\mathbf{J}}\boldsymbol{\Psi}_{\mathrm{ARE}} + \boldsymbol{\Psi}_{\mathrm{ARE}}\tilde{\mathbf{J}}^{\intercal} + \frac{1}{2}\left(\left(\frac{\partial I}{\partial \mathbf{P}} + \frac{\partial \mathbf{r}_{\mathrm{cl}}^{n_t}}{\partial \mathbf{P}}^{\intercal} \boldsymbol{\psi}_{\mathrm{cl}}^{n_t} \right) + \left(\frac{\partial I}{\partial \mathbf{P}} + \frac{\partial \mathbf{r}_{\mathrm{cl}}^{n_t}}{\partial \mathbf{P}}^{\intercal} \boldsymbol{\psi}_{\mathrm{cl}}^{n_t} \right)^{\intercal} \right) = 0,$$

Backward closed loop system

$$\begin{bmatrix} \mathbf{I} & -\mathbf{I} + \left(-\mathbf{J}^{(1)\intercal} - \mathbf{W}^{\intercal}\mathbf{G}^{(1)\intercal}\right)\Delta t & \cdots & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{I} & \cdots & \mathbf{O} & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{I} & -\mathbf{I} + \left(-\mathbf{J}^{(n-1)\intercal} - \mathbf{W}^{\intercal}\mathbf{G}^{(n-1)\intercal}\right)\Delta t \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}_{\mathrm{cl}}^{(1)} \\ \boldsymbol{\psi}_{\mathrm{cl}}^{(2)} \\ \vdots \\ \boldsymbol{\psi}_{\mathrm{cl}}^{(n-1)} \\ \boldsymbol{\psi}_{\mathrm{cl}}^{(n)} \end{bmatrix} = \begin{bmatrix} -\frac{\partial I}{\partial \delta\mathbf{x}^{(1)}} \\ -\frac{\partial I}{\partial \delta\mathbf{x}^{(2)}} \\ \vdots \\ -\frac{\partial I}{\partial \delta\mathbf{x}^{(n-1)}} \\ -\frac{\partial I}{\partial \delta\mathbf{x}^{(n)}} \end{bmatrix}.$$

Rotors analysis: blade element momentum (BEM) theory





Rotor RPM
Inflow speed
Inflow direction

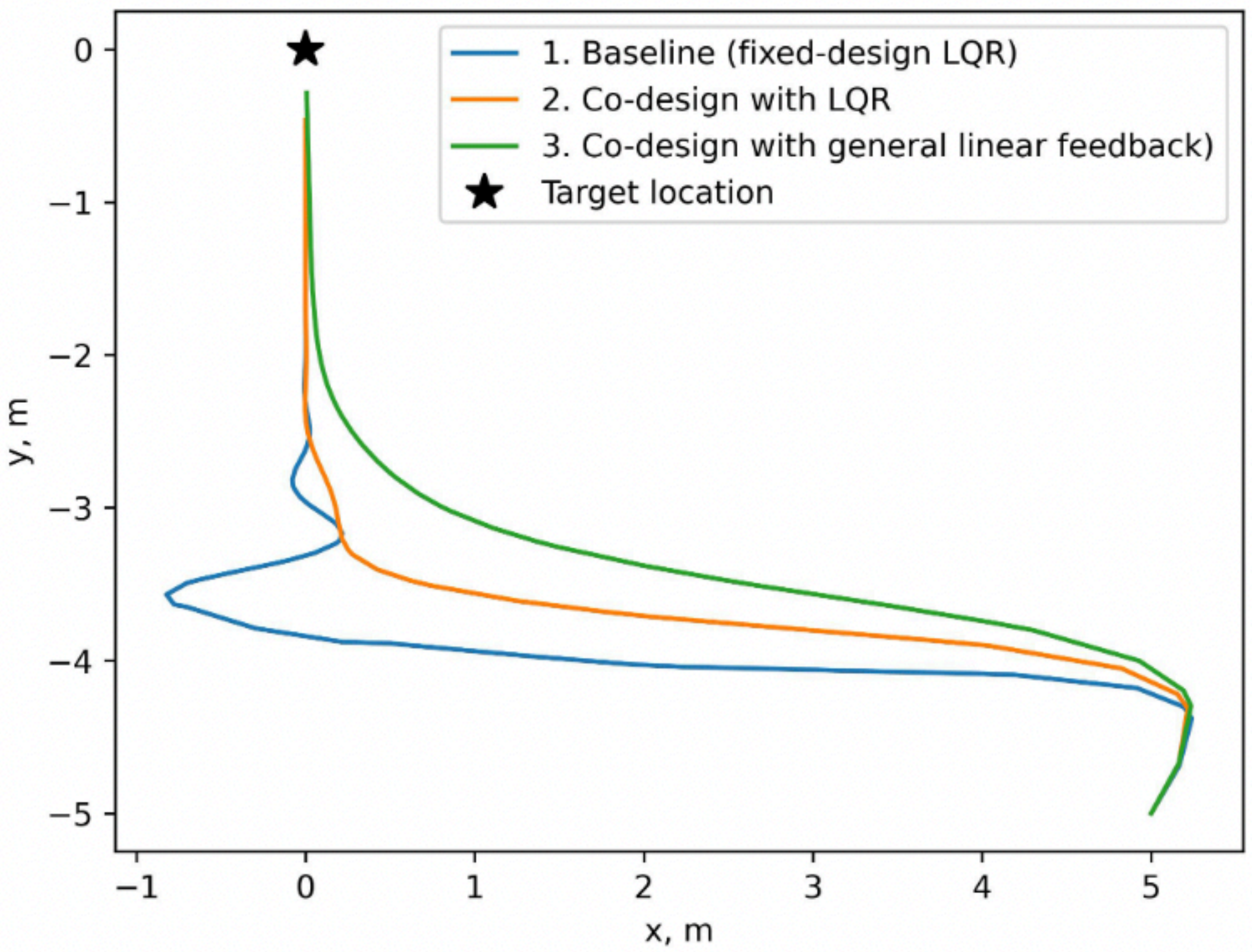
↓

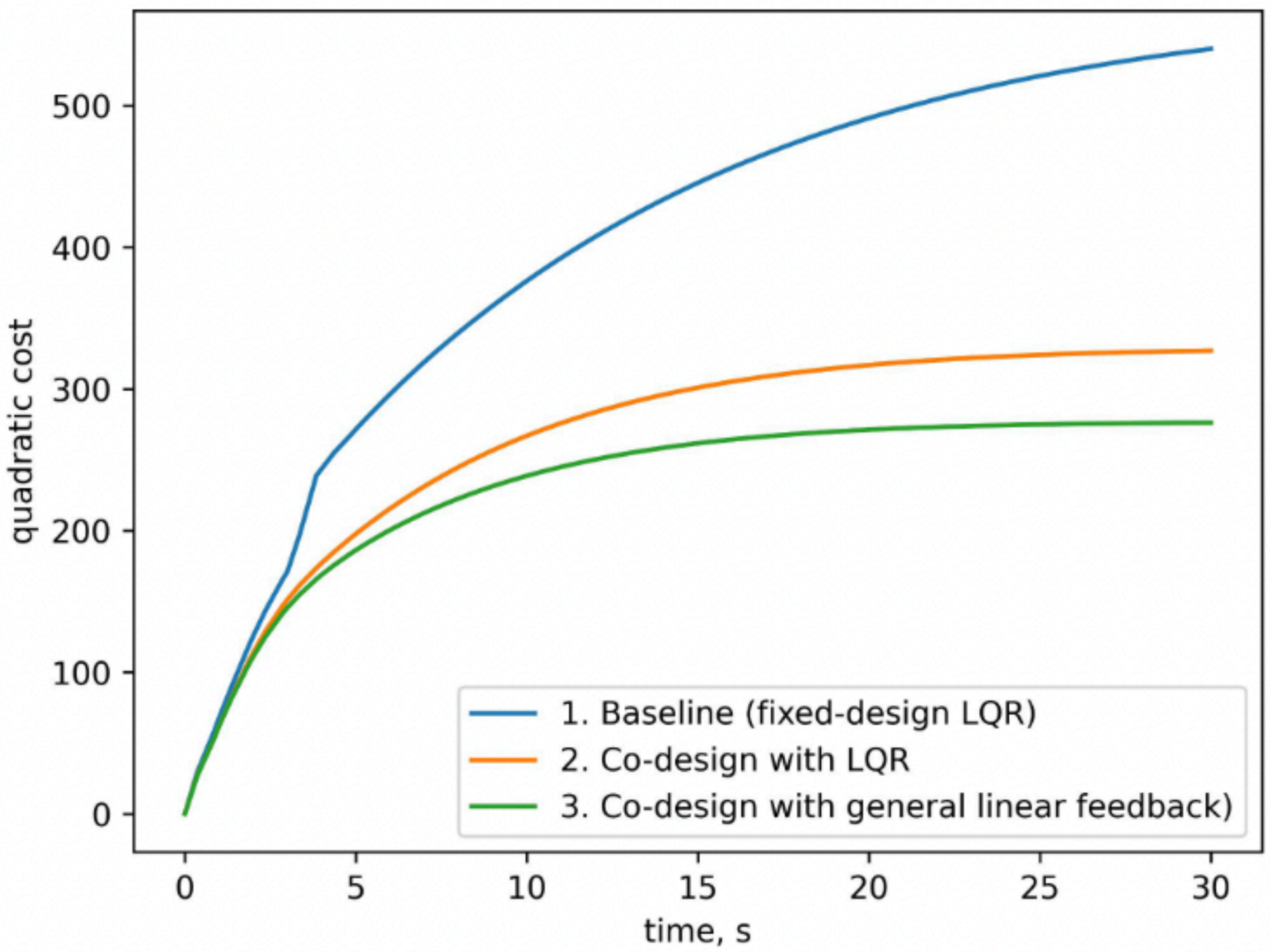
BEM Analysis
CCBlade [1]

↓

Thrust
Power

Control co-design reduces the cost by about 50%.





Content

1. **MDO with general dynamical systems**
   - Incorporating more physics in design.
   - Develop efficient adjoint methods (one dynamical system-one adjoint).

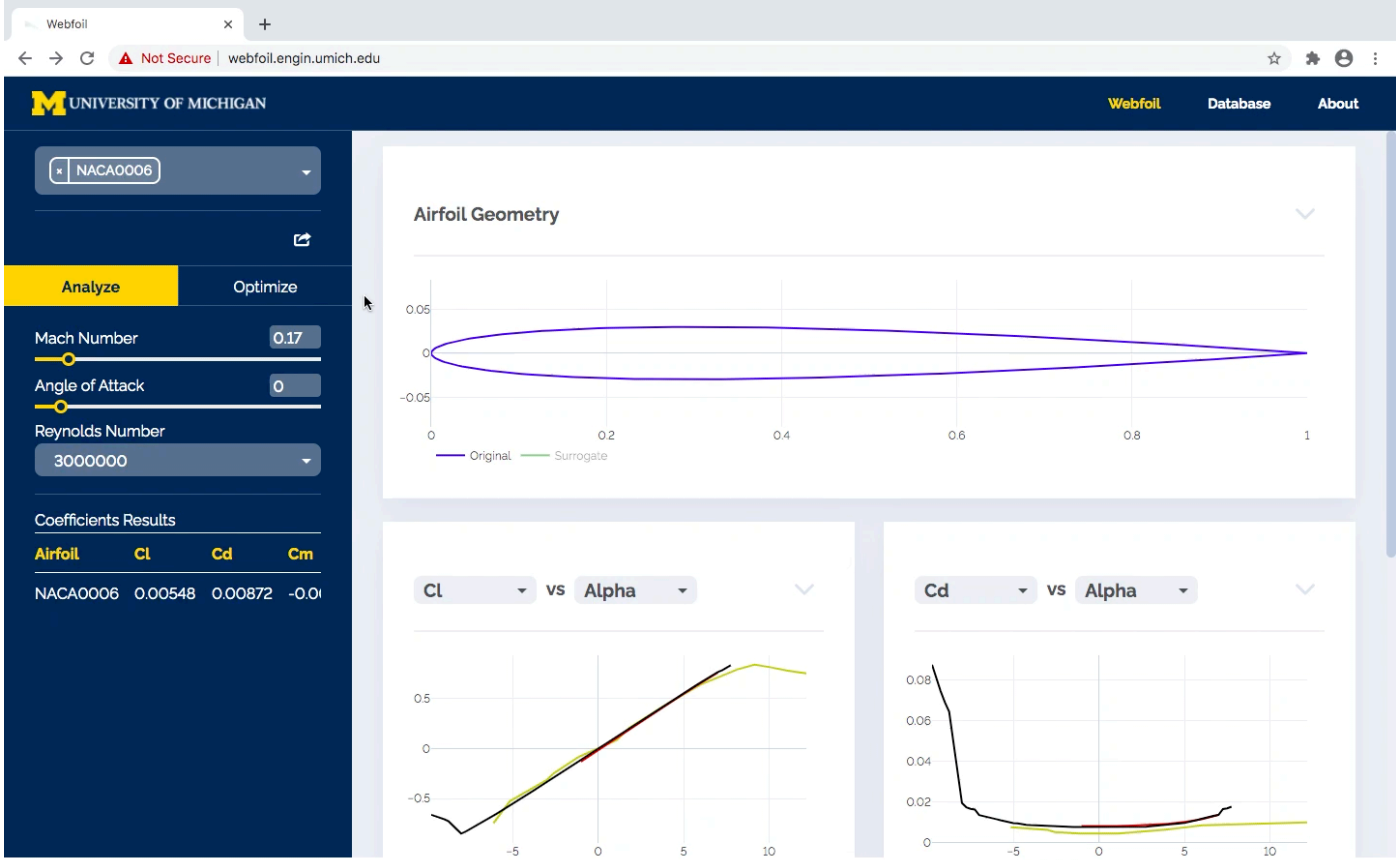
---

2. **Control co-design (CCD)**
   - Exploiting control & plant variables interaction.
   - Efficient derivative computation.

---

3. **Data-driven MDO**
   - Real-time transonic airfoil shape optimization.
   - Speed up expensive simulation.

Online data-driven airfoil shape optimization tool Webfoil.



- Ultra fast real-time airfoil shape optimization for subsonic and transonic regimes.
- Highly accurate (<1% relative error for C_l, C_d, C_m prediction).
- Deep neural network trained with function data and additional gradient data (Sobolev neural network).

webfoil.engin.umich.edu

The challenge for data-driven optimization ...

1. The traditional surrogate methods, e.g., Kriging method (Gaussian process regression) has difficulty to use large data set (>10 k) and overfitting issue.
2. Directly relating the design variables to outputs generalize poorly (missing physics / different parametrization).

Sobolev artificial neural network (SANN) takes the gradient into account in loss function.

Loss function

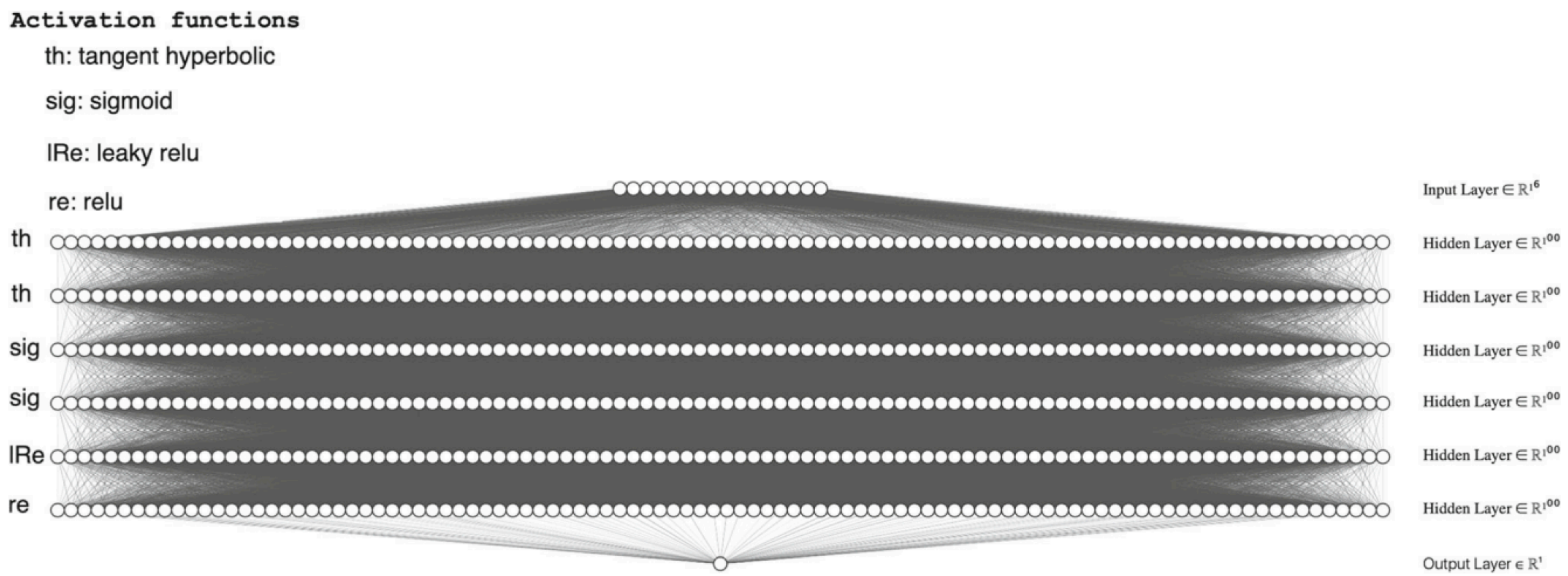
$$\min_{\theta} \bar{y}_{\text{loss}}(\mathbf{X}, \mathbf{y}, df|\theta),$$

Function value penalty

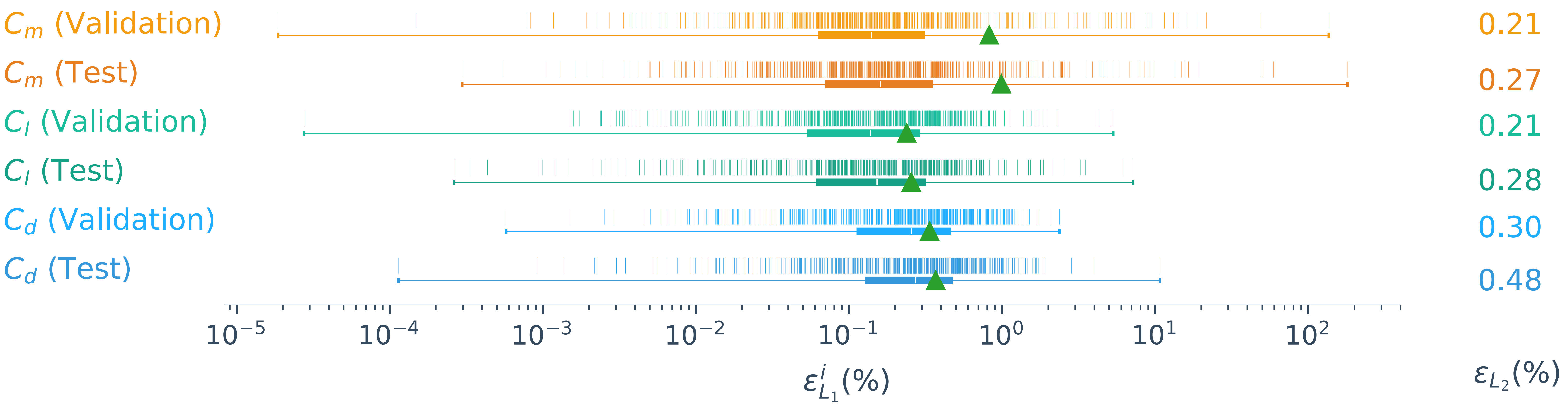
$$\bar{y}_{\text{loss}}(\mathbf{X}, \mathbf{y}, \mathbf{df}|\theta) := \sum_{i=1}^{n_t} l\left(m\left(\mathbf{x}^{(i)}|\theta\right), y^{(i)}\right) + \lambda_k \sum_{j=1}^{d} l_j \left(\frac{\partial m}{\partial x_j}\left(\mathbf{x}^{(i)}|\theta\right), \frac{\partial f}{\partial x_j}\left(\mathbf{x}^{(i)}\right)\right),$$

Gradient penalty

Neural network structure (Tensorflow/Keras)

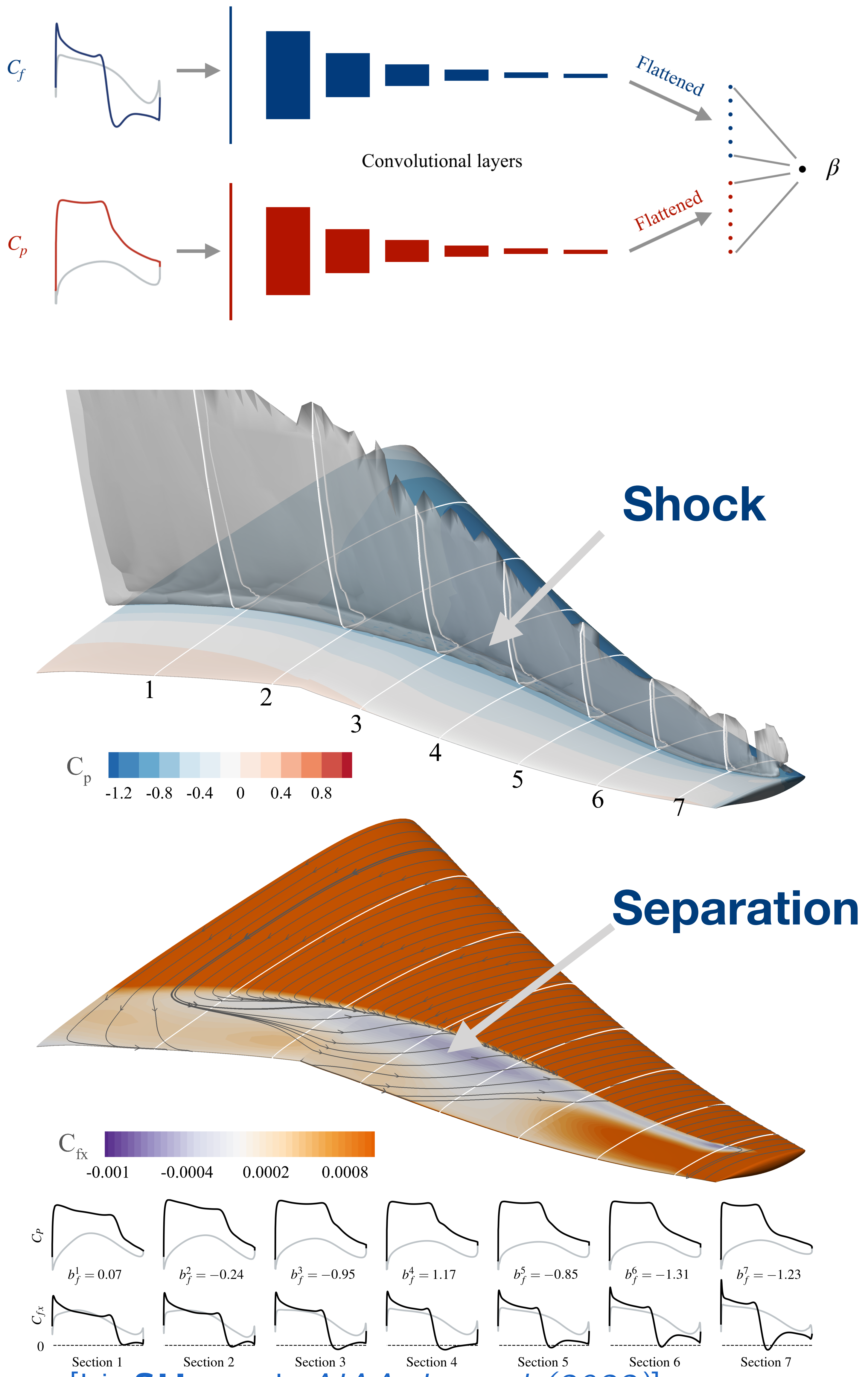


The prediction error is less than 1% in transonic regimes.



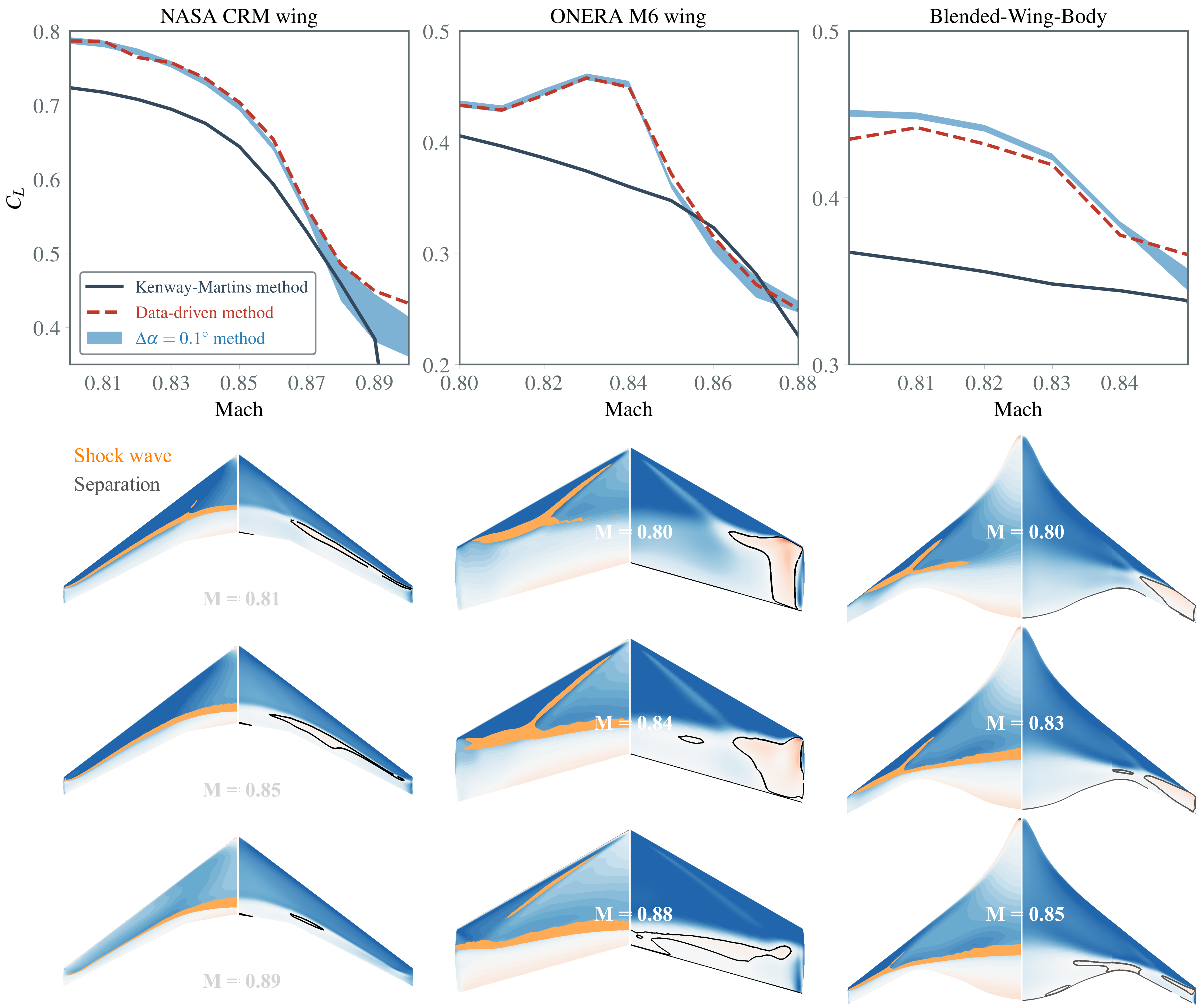
[Bouhlel, **SH**, Martins, *SAMO 2020*]

To make machine learning model generalize better, we need to teach it some physics ...

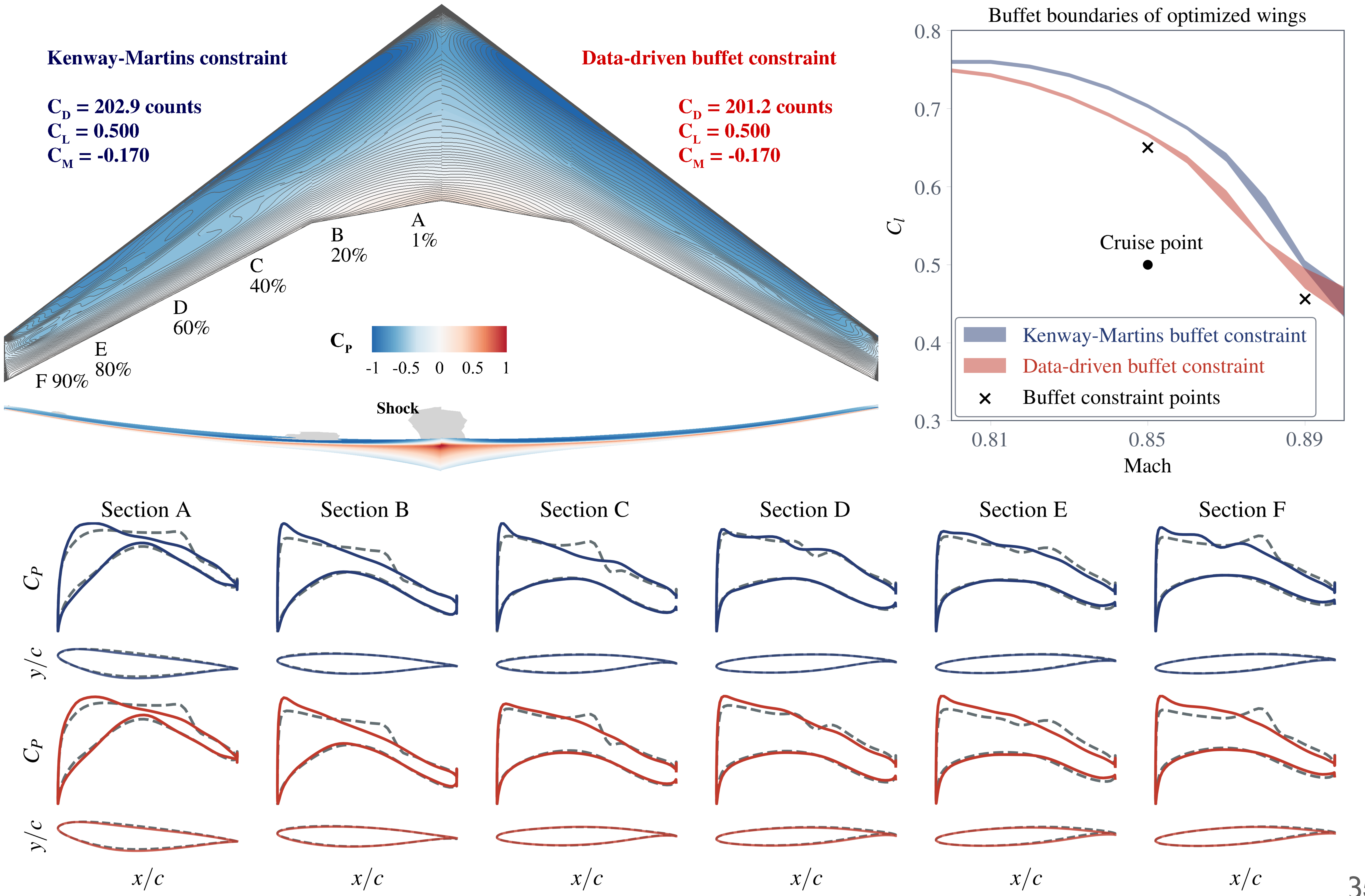


[Li, **SH**, et al. *AIAA Journal (2022)*]

- Data-driven model trained in a physics-based manner using 2-D simulation data
- Extendable to any 3-D wings



The data-driven model finds a better optimal solution at a lower cost.



[Li, **SH**, et al. *AIAA Journal (2022)*]

Summary

1. **MDO with general dynamical systems**
   - Dynamical system theory enables us to develop more general numerical methods that solves different physical problems.
   - Developed multiple efficient derivative computation methods for LCO, bifurcation, etc.

2. **Control co-design (CCD)**
   - Found better design by exploiting plant and control variables interaction.
   - Developed efficient coupled adjoint solver leveraging feedforward structure of the problem.

3. **Data-driven MDO**
   - Realized real-time airfoil shape optimization using deep neural network.
   - Developed machine learning model that generalizes better by embedding physics in the model.

Let’s come up with better designs using MDO, dynamical system & control, and machine learning tools!