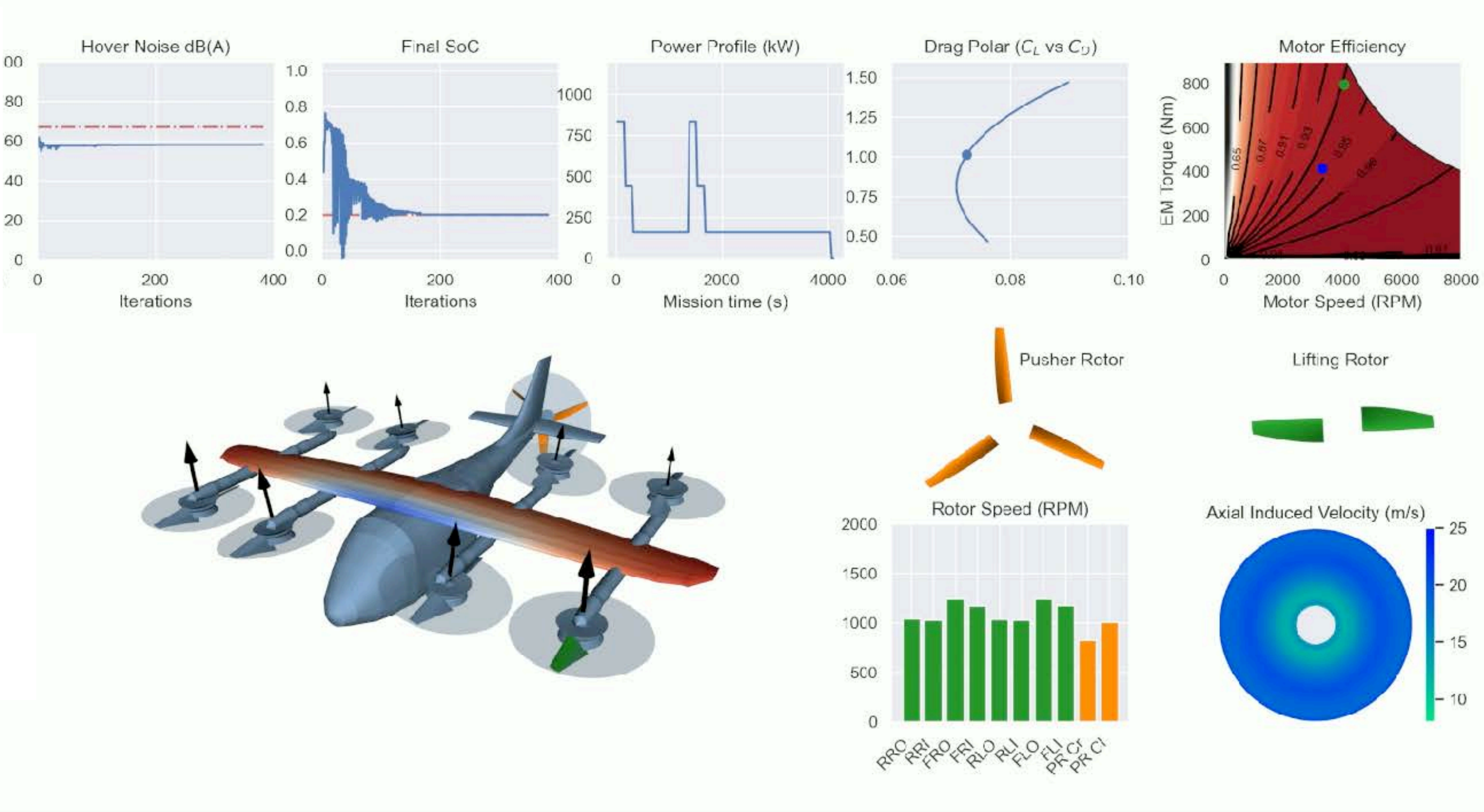
Recent developments in large-scale multidisciplinary design optimization (application to urban air mobility vehicle design)

6th Wind energy systems engineering workshop

Boulder, CO

August 31, 2022



John Hwang

Assistant Professor

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Mechanical and Aerospace Engineering

What is urban air mobility?

Advanced air mobility		
Urban air mobility		Air travel for underserved markets
Air travel in urban areas	Package delivery	

Enabled by advances in electric propulsion, batteries, autonomy, advanced manufacturing

[NASA]

Electric vertical takeoff and landing (eVTOL) aircraft

150 mph cruise

Up to 5 people

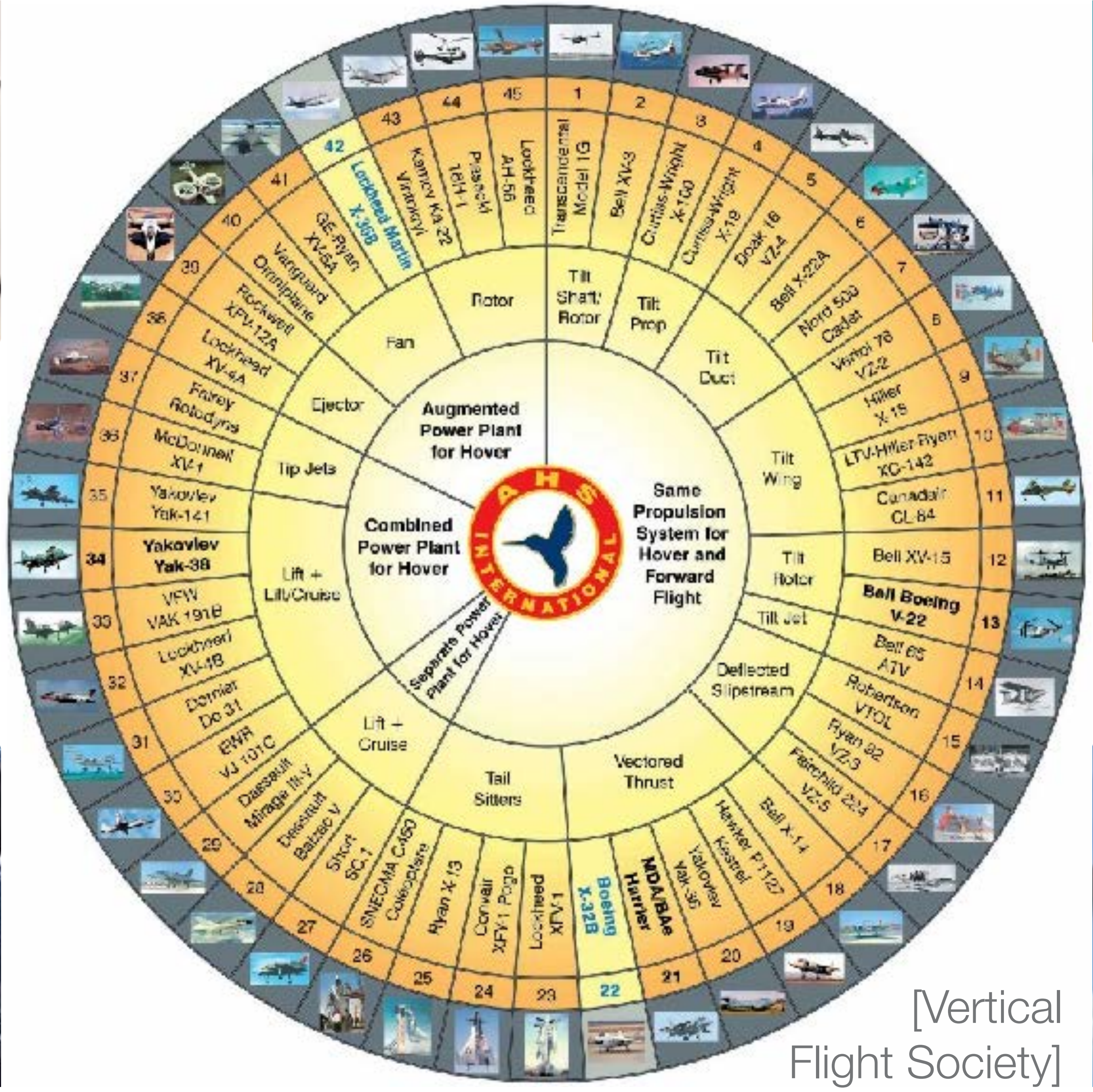
100 km range

4000~7000 lb

There are many open questions in vehicle design


Archer Aviation


Bell



[Vertical Flight Society]


Overair


Jaunt Air Mobility

Challenges:

- Diverse set of possible design configurations
- Large uncertainties on technological assumptions
- Operating parameters are also changing
- Traditional design methods that rely on existing designs are not applicable to eVTOL

Need: highly automated, physics-based design tools based on full-configuration simulation

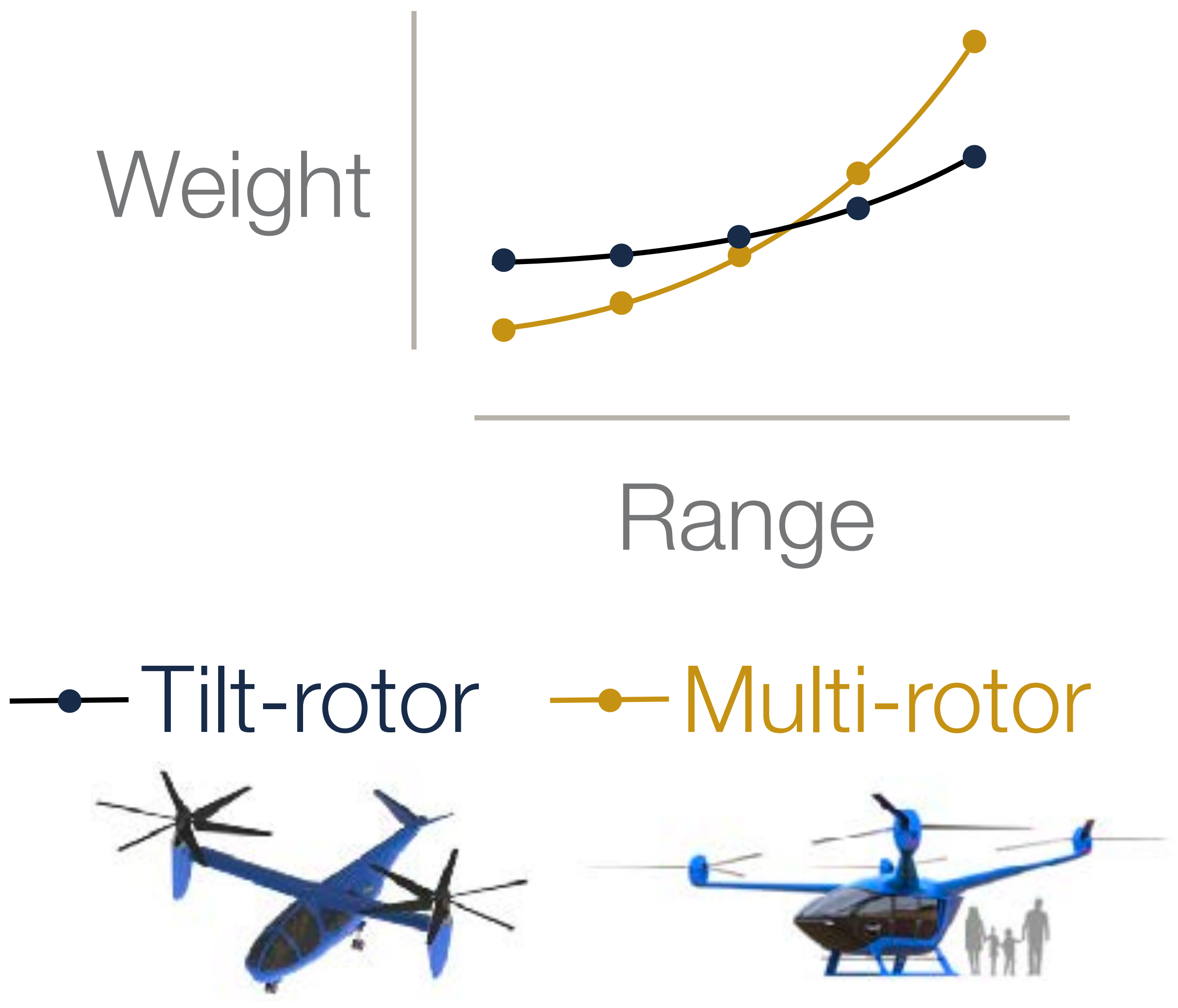
We can address this gap using
large-scale multidisciplinary design optimization (MDO)

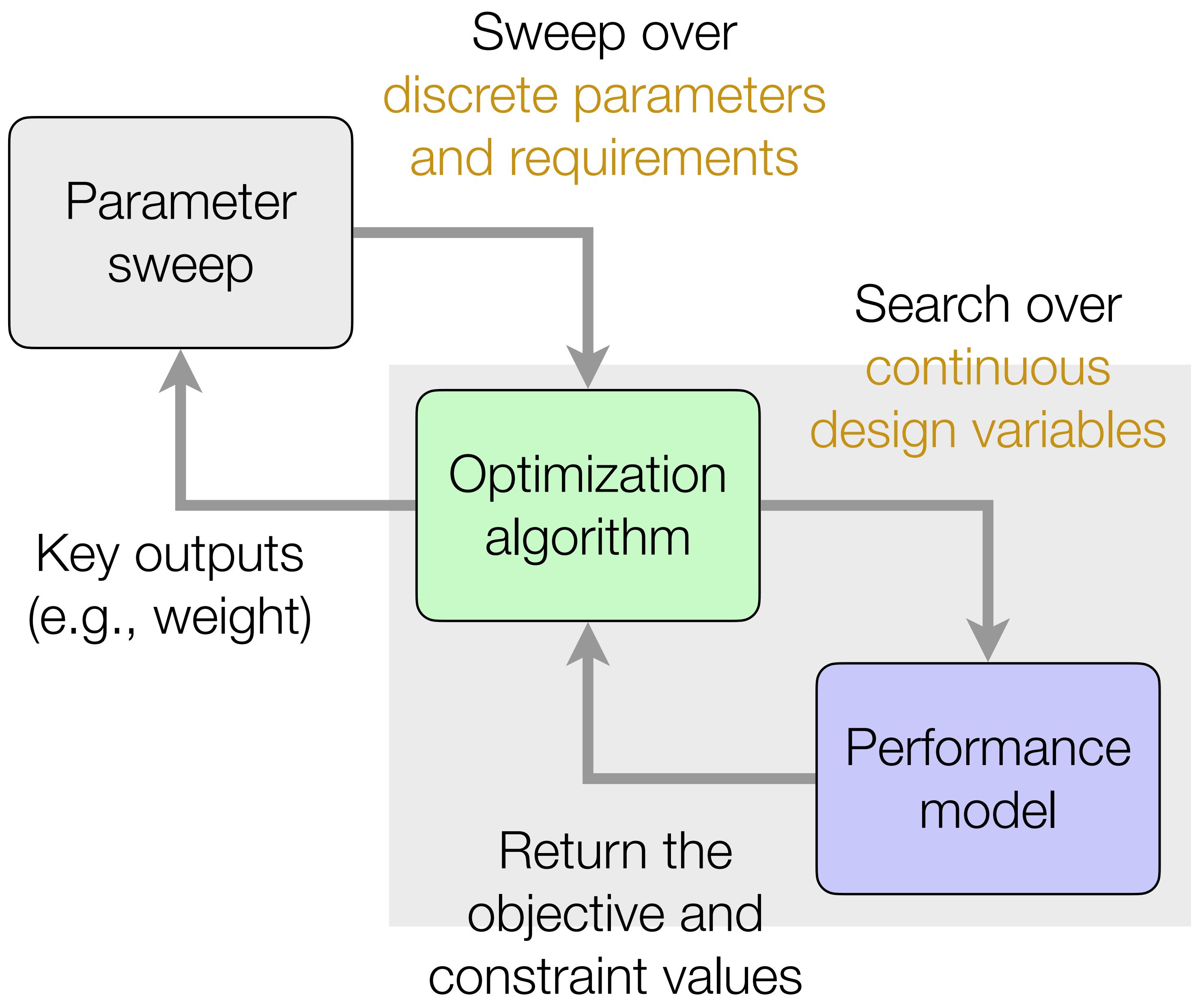
10s or more design variables

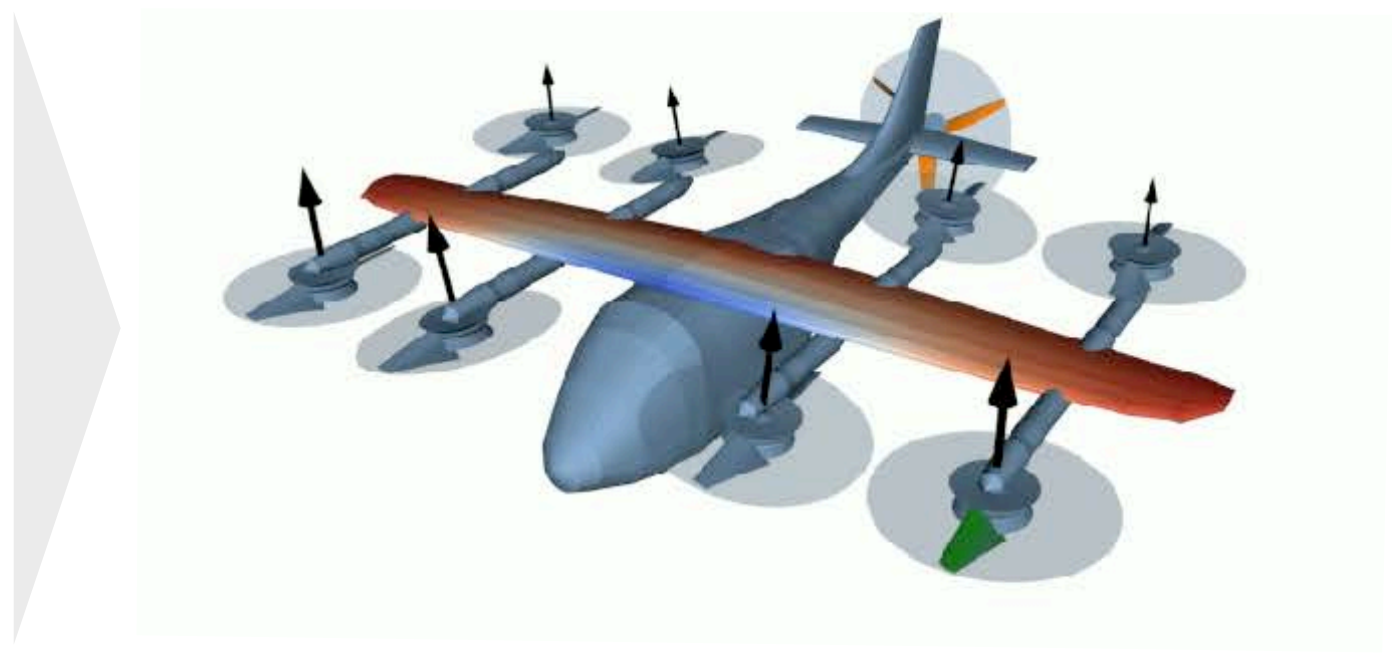
Use computational models involving multiple disciplines

Apply numerical optimization algorithms

Trade study using MDO







Outline

- Review of sensitivity analysis methods

  Novel methodology for system modeling

  Demonstration on UAM air taxi design problem

In engineering design, optimization problems are solved using either gradient-based or gradient-free optimizers

Gradient-based optimizer

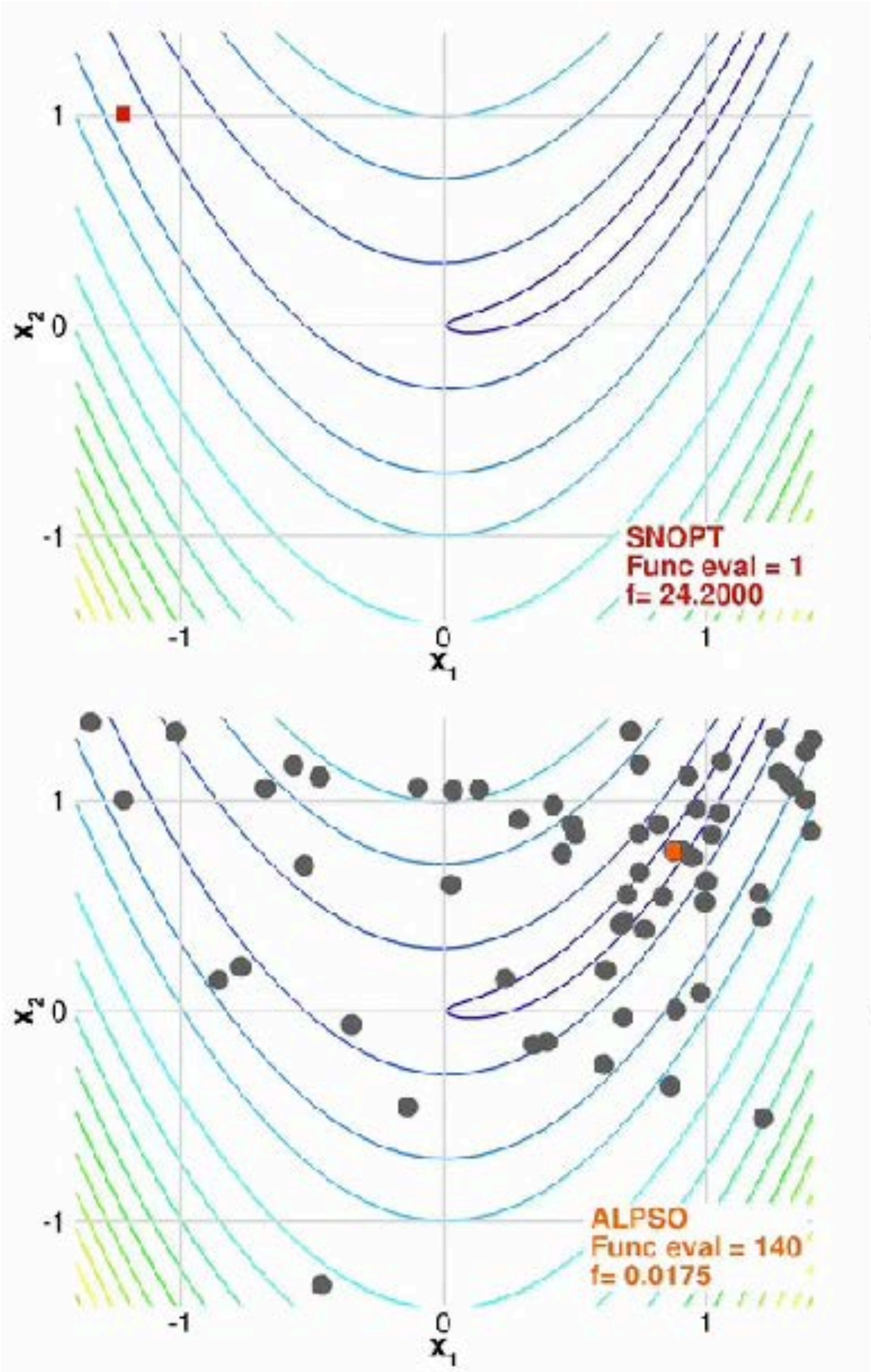
(SNOPT)

41 iterations

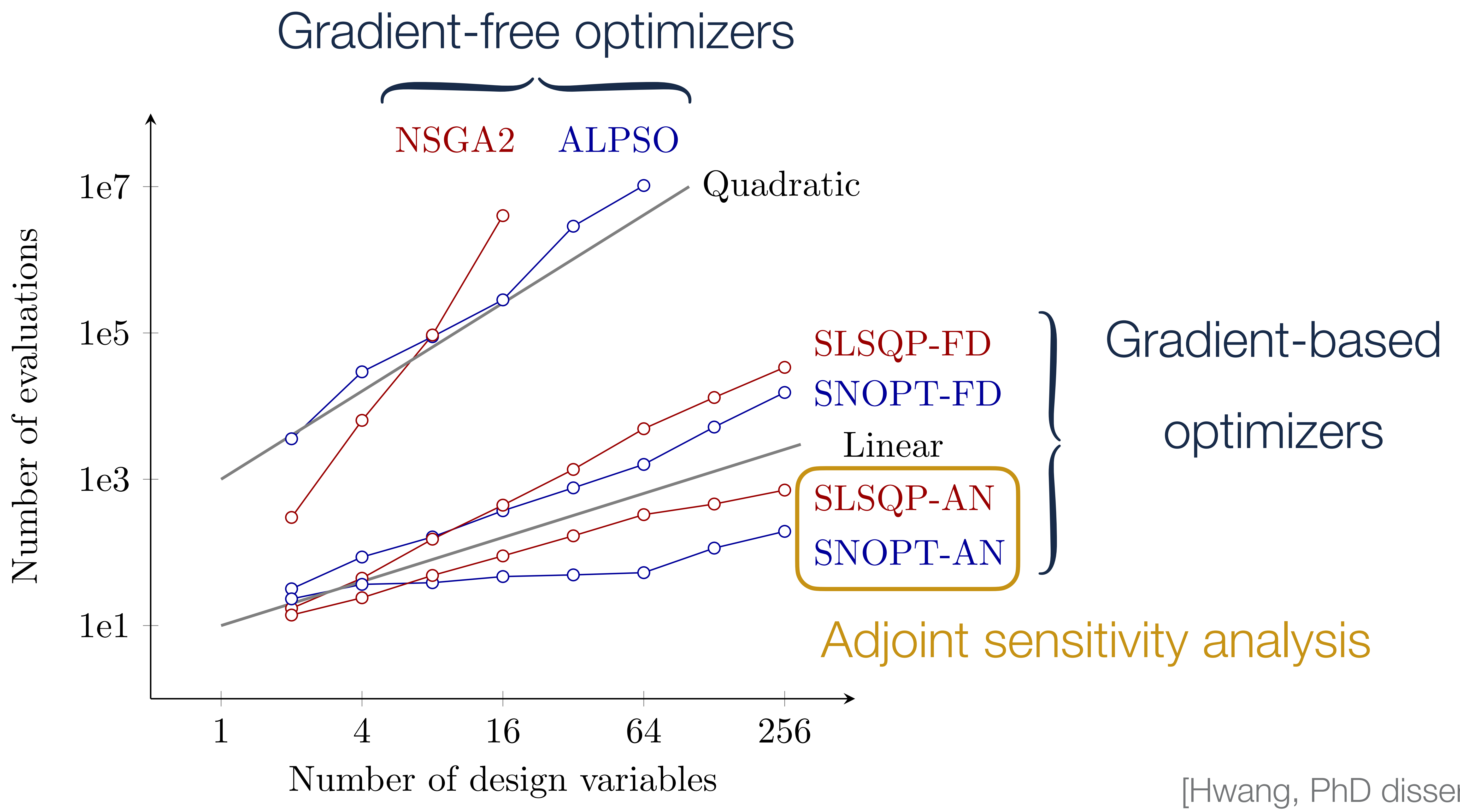
Gradient-free optimizer

(ALPSO)

1340 iterations



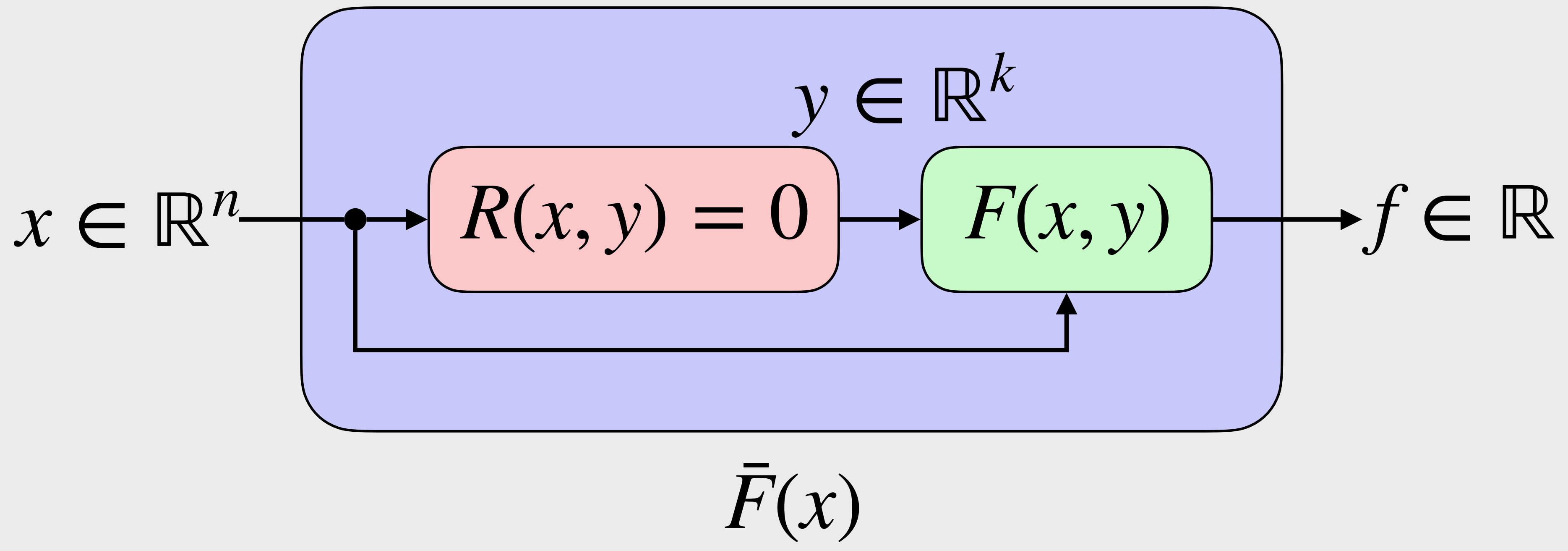
Gradient-based optimization is the only option for large-scale problems



[Hwang, PhD dissertation, 2015]

Sensitivity analysis methods—review

Model structure and nomenclature



Cost is O(n)

Finite-difference method $$\frac{df}{dx_i} \approx \frac{\bar{F}(x + he_i) - \bar{F}(x)}{h}, \quad h \in \mathbb{R}$$

Complex-step method $$\frac{df}{dx_i} \approx \frac{Im[\bar{F}(x + ihe_i)]}{h}, \quad h \in \mathbb{R}$$

Algorithmic differentiation (AD) — Cost: ~1 nonlinear solution of R(x,y)=0

$$\frac{df}{dx_i} = \sum_{j=1}^{n_t} \frac{df}{dt_j}\frac{\partial T_j}{\partial x_i} \quad \text{where } t_j = T_j(x, t_1, \ldots, t_{j-1})$$

Adjoint method — Cost: ~1 linear solution

$$\frac{df}{dx} = \frac{\partial F}{\partial x} + \psi^T \frac{\partial R}{\partial x} \quad \text{with} \quad \frac{\partial R}{\partial y}^T \psi = -\frac{\partial F}{\partial y}^T$$

Unified derivatives equation (UDE)

$$u = \begin{bmatrix} x \\ y \\ f \end{bmatrix}, \quad \bar{R}(u) = \begin{bmatrix} x - x^* \\ -R(x, y) \\ f - F(x, y) \end{bmatrix}$$

$$\frac{\partial \bar{R}}{\partial u}^T \frac{du}{dr}^T = I$$

The combination of AD and the adjoint method yields maximum efficiency and some automation

Algorithmic differentiation (AD)

$$\frac{df}{dx_i} = \sum_{j=1}^{n_t} \frac{df}{dt_j} \frac{\partial T_j}{\partial x_i} \quad \text{where } t_j = T_j(x, t_1, \ldots, t_{j-1})$$

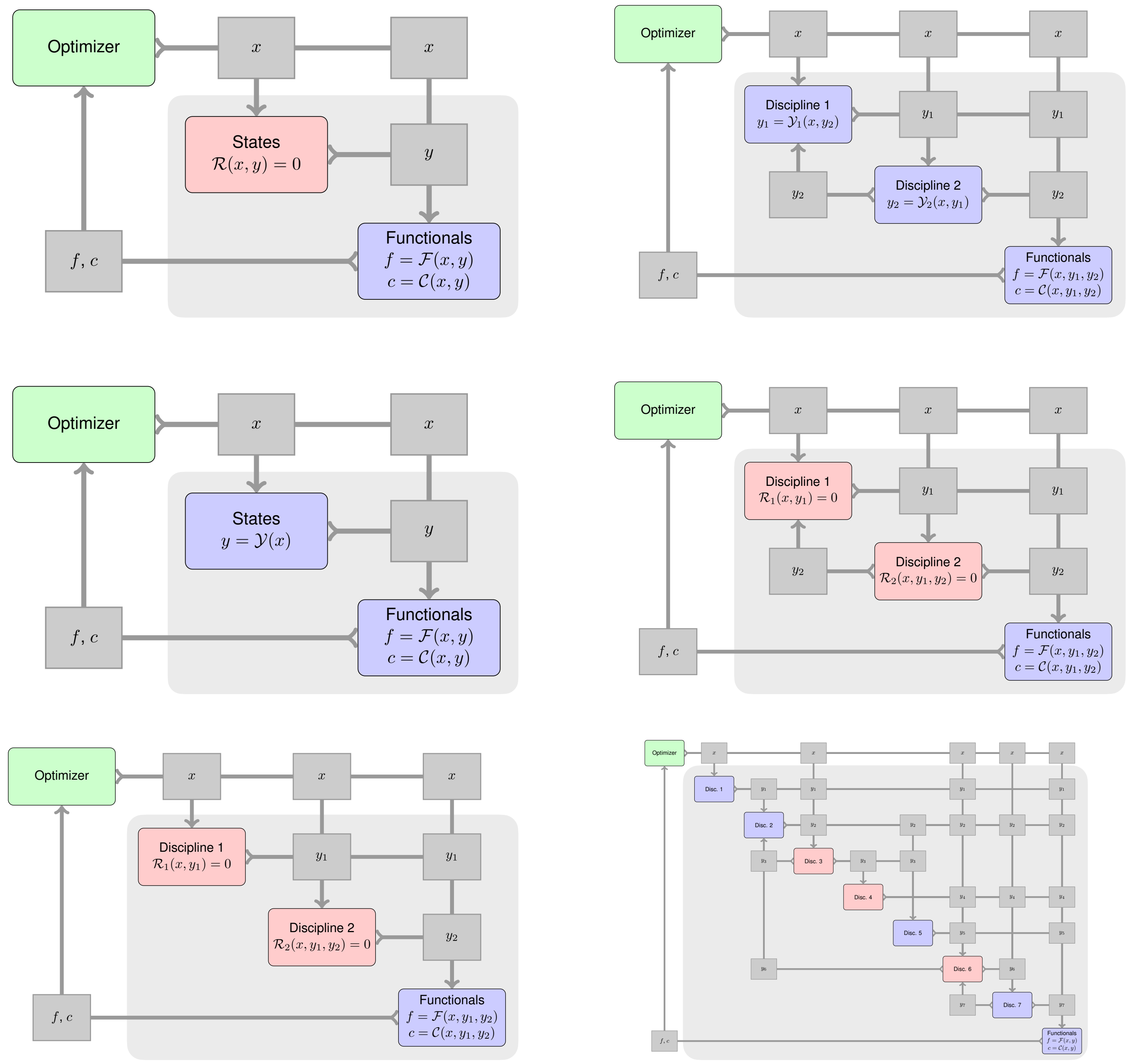
Adjoint method

$$\frac{df}{dx} = \frac{\partial F}{\partial x} + \psi^T \frac{\partial R}{\partial x} \quad \text{with} \quad \frac{\partial R}{\partial y}^T \psi = -\frac{\partial F}{\partial y}^T$$

Compute these Jacobians
of partial derivatives using AD

- Pro: Inherits advantages of both AD and the adjoint method
- Con: Still some implementation effort required when system models are assembled or reconfigured

The UDE eliminates implementation effort for sensitivity analysis when system models are assembled



System model (grey boxes) can have multiple adjoints (red boxes)

Formulate as a nonlinear system

$$R(u) = 0$$

Apply the inverse function theorem

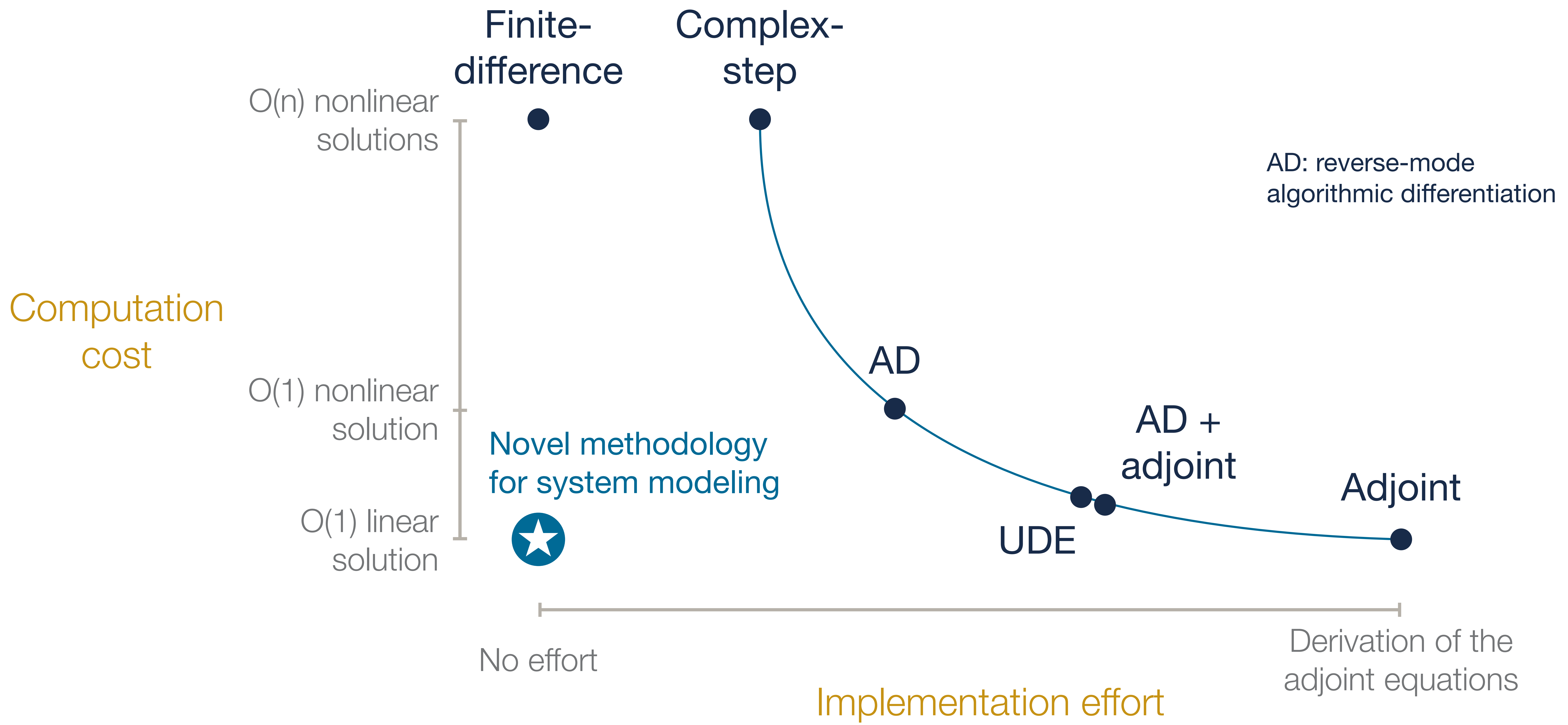
[Hwang and Martins, ACM TOMS, 2018]

UDE

$$\frac{\partial R}{\partial u} \frac{du}{dr} = I = \frac{\partial R}{\partial u}^T \frac{du}{dr}^T$$

- Con: Significant up-front implementation effort required to compute partial derivatives

Sensitivity analysis methods—Pareto front



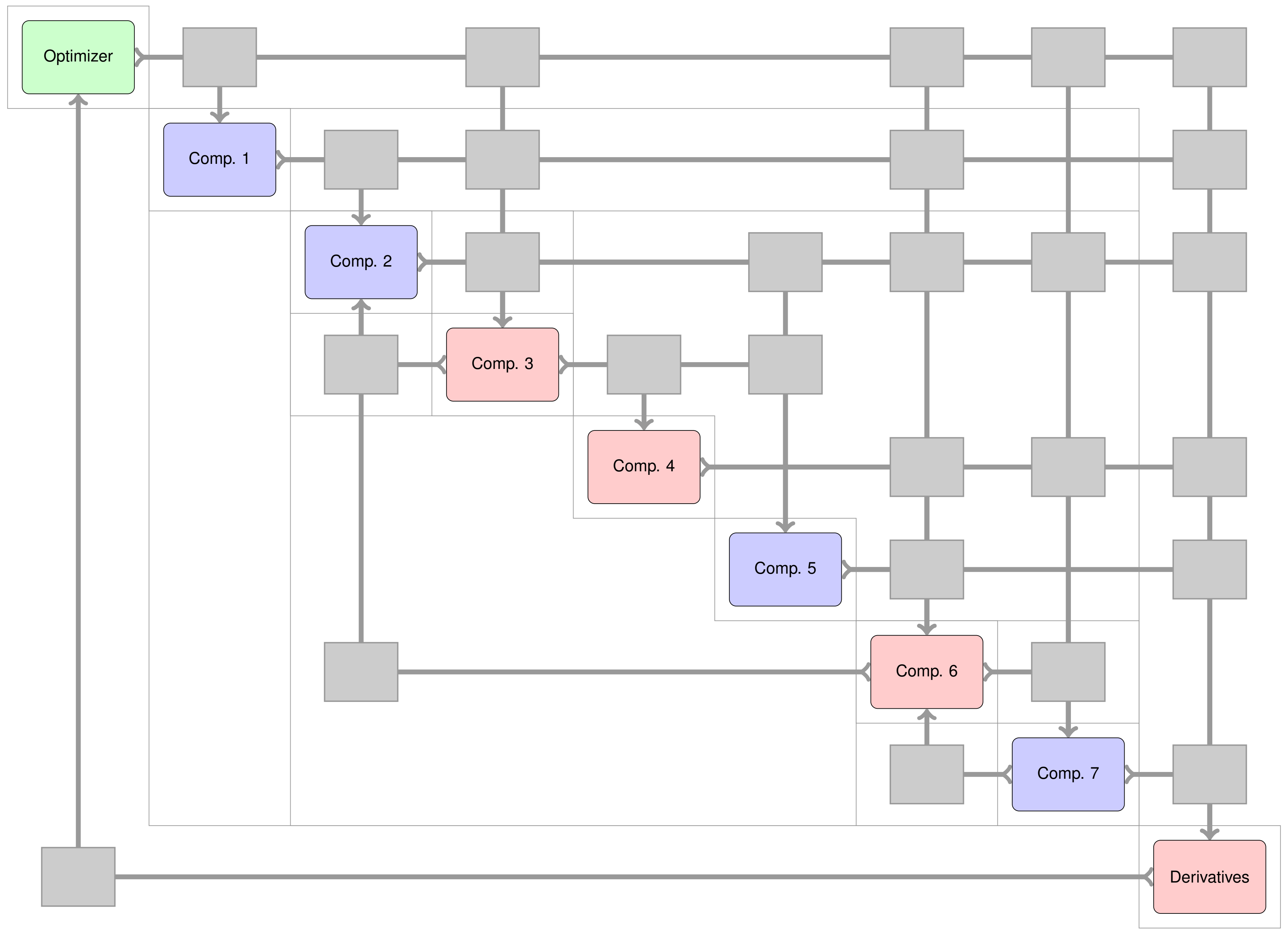
Outline

Review of sensitivity analysis methods

- Novel methodology for system modeling

Demonstration on UAM air taxi design problem

OpenMDAO (UDE) makes computing derivatives easier but the current bottleneck is computing partial derivatives

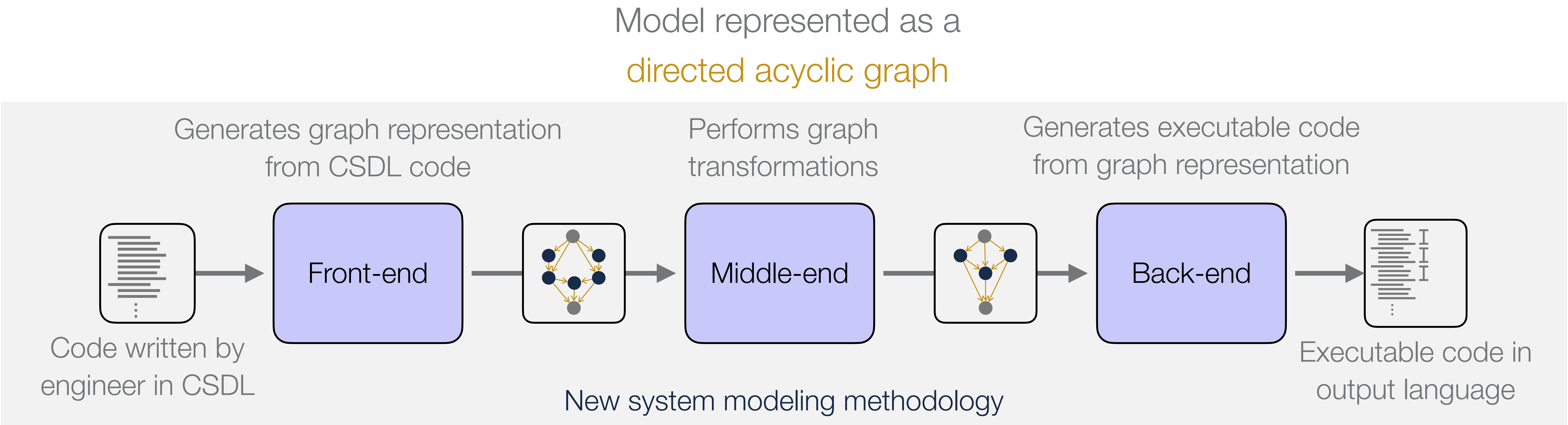


Step (1): Each component of the model provides its partial derivatives

$$\frac{\partial R}{\partial u}^T \frac{du}{dr}^T = I \quad (*)$$

Step (2): The modeling framework solves Eq. (*) to compute the total derivatives

We resolved this via a new methodology for system modeling that fully automates adjoint-based sensitivity analysis



CSDL is a new algebraic modeling language

Automatic code generation paradigm used in AD and PDE solution frameworks (FEniCS)

CSDL enables this new methodology for system modeling

The computational system design language (CSDL) is an algebraic modeling language for large-scale MDO.

Characteristics:

- An embedded domain-specific language (a subset of Python; intended for system modeling)
- Designed to be expressive (easy to use as CSDL code looks like ordinary Python code)
- Large standard library of operations (see right)
- Extensive support for tensor algebra to encourage vectorization (to minimize the number of operations)
- It enables a computational graph to be constructed that fully describes the model (at the level of fundamental operations)

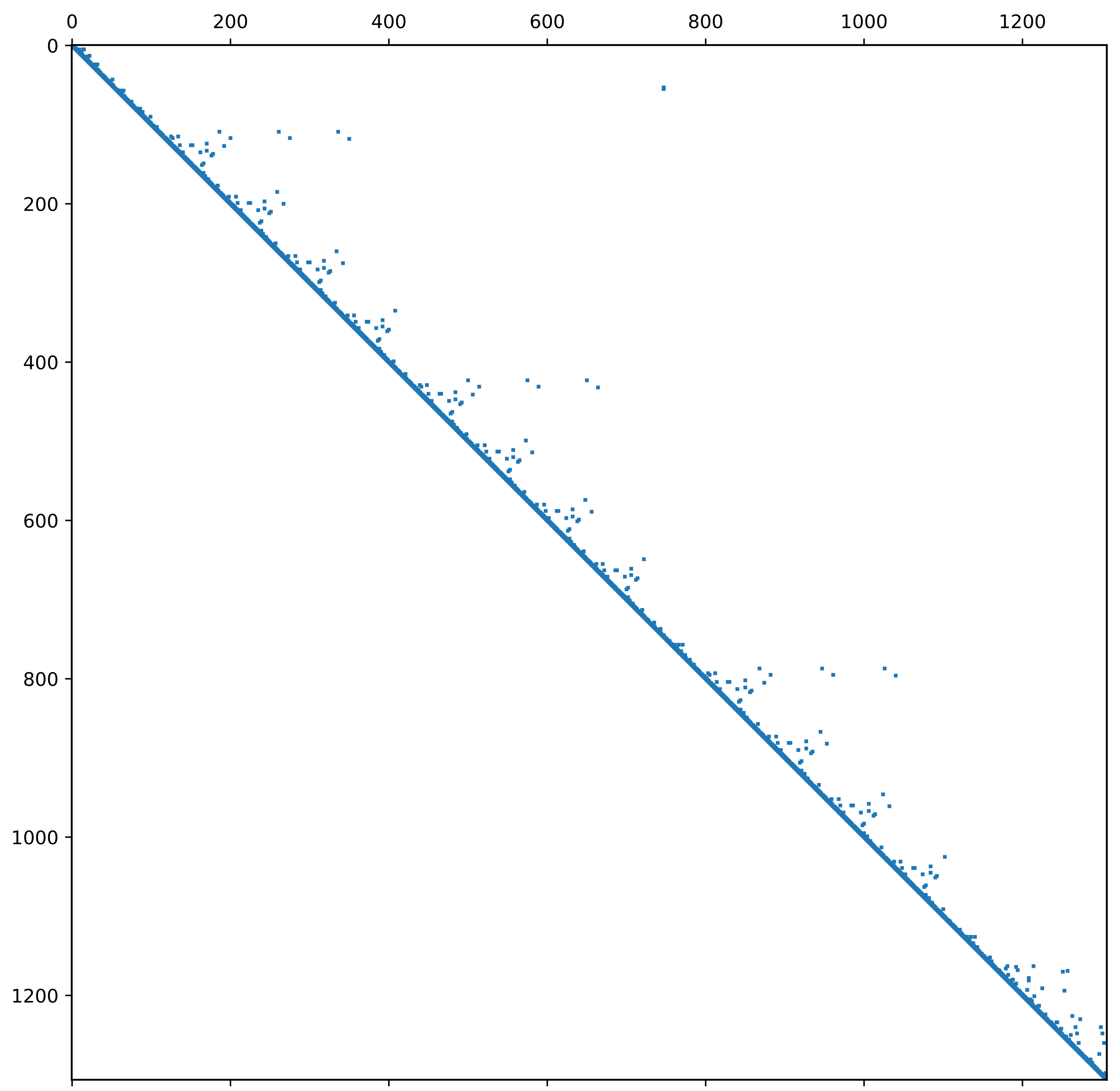

CSDL

Tutorial
CSDL by Example
Introduction
Basic Examples
Standard Library
average
cross
dot
einsum_new
einsum_old
expand
inner
matmat
matvec
max
min
outer
pnorm

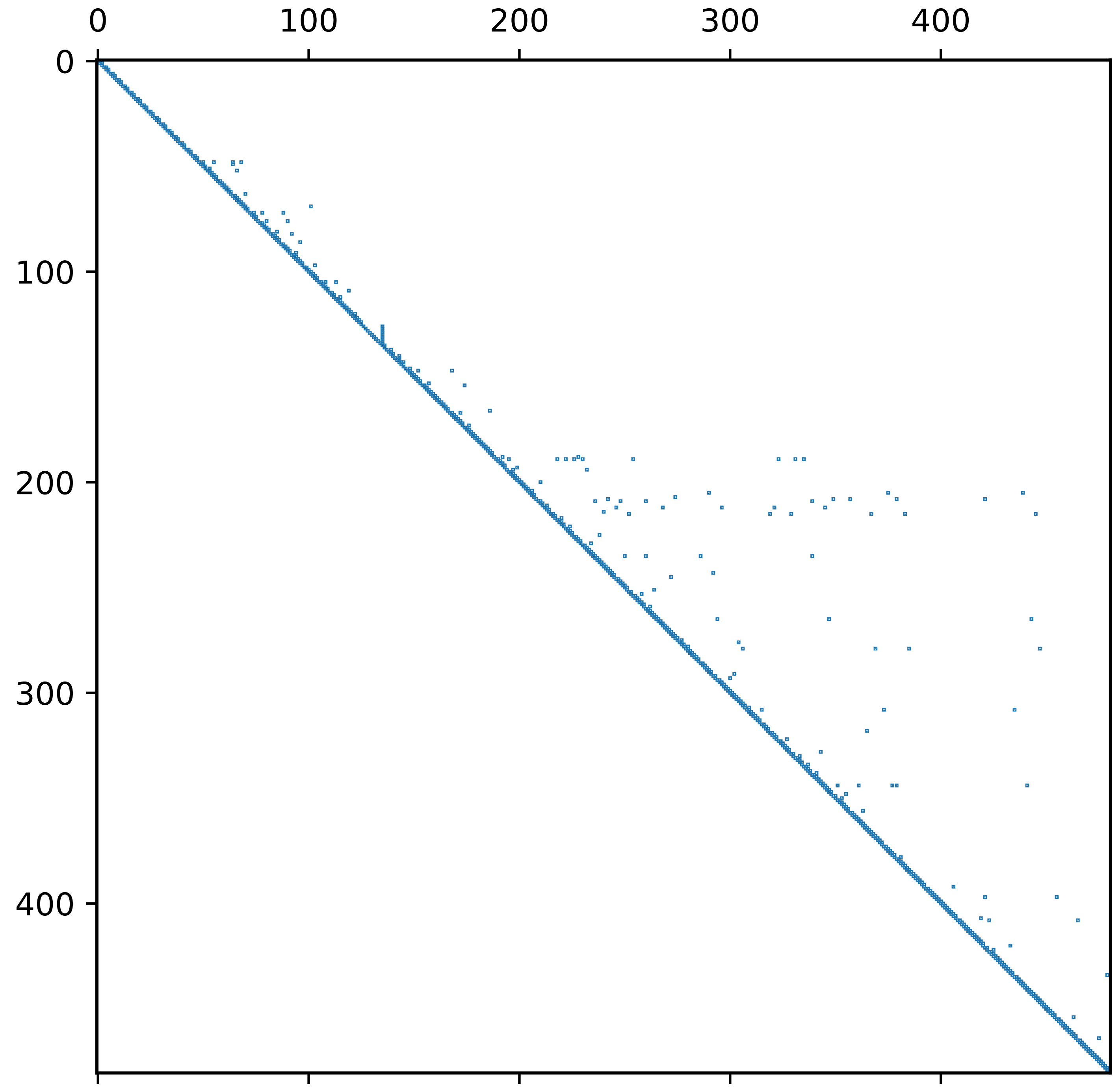
```python
from csdl_om import Simulator
import numpy as np
import csdl
from csdl import Model


class ExampleInteger(Model):

    def define(self):
        a = self.declare_variable('a', val=0)
        b = self.declare_variable('b', val=1)
        c = self.declare_variable('c', val=2)
        d = self.declare_variable('d', val=7.4)
        e = self.declare_variable('e', val=np.pi)
        f = self.declare_variable('f', val=9)
        g = e + f
        x = self.create_output('x', shape=(7, ))
        x[0] = a
        x[1] = b
        x[2] = c
        x[3] = d
        x[4] = e
        x[5] = f
        x[6] = g

        # Get value from indices
        self.register_output('x0', x[0])
        self.register_output('x6', x[6])
        self.register_output('x_2', x[-2])
```

Adjacency matrices of graphs of CSDL models

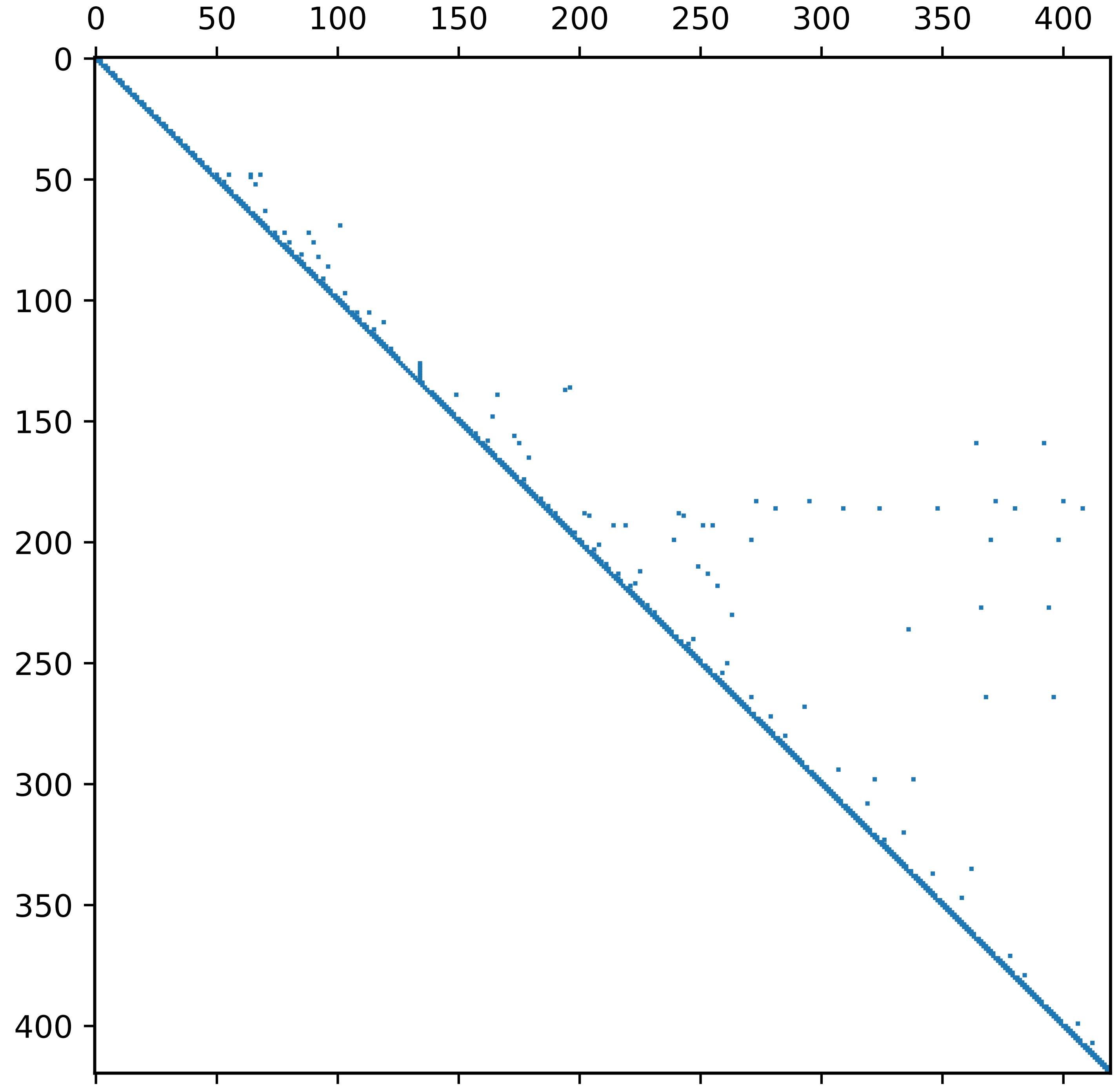
Vortex lattice method



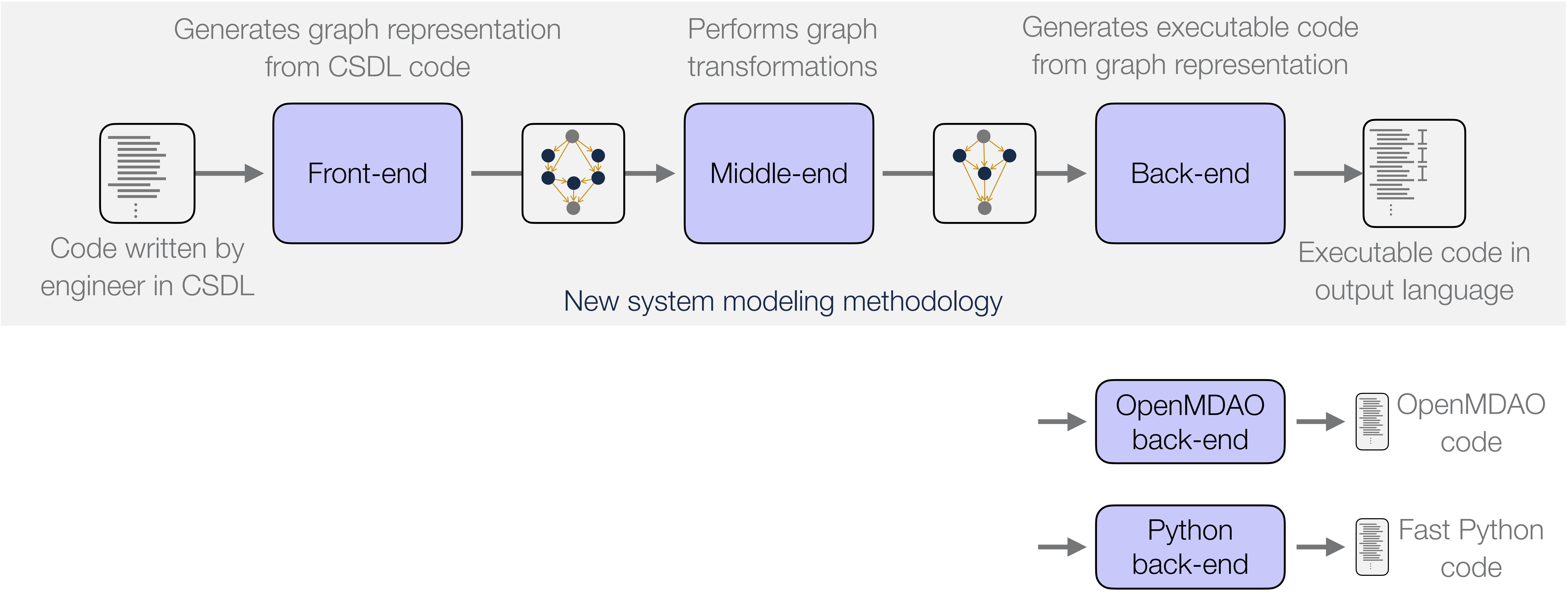
Blade element momentum method



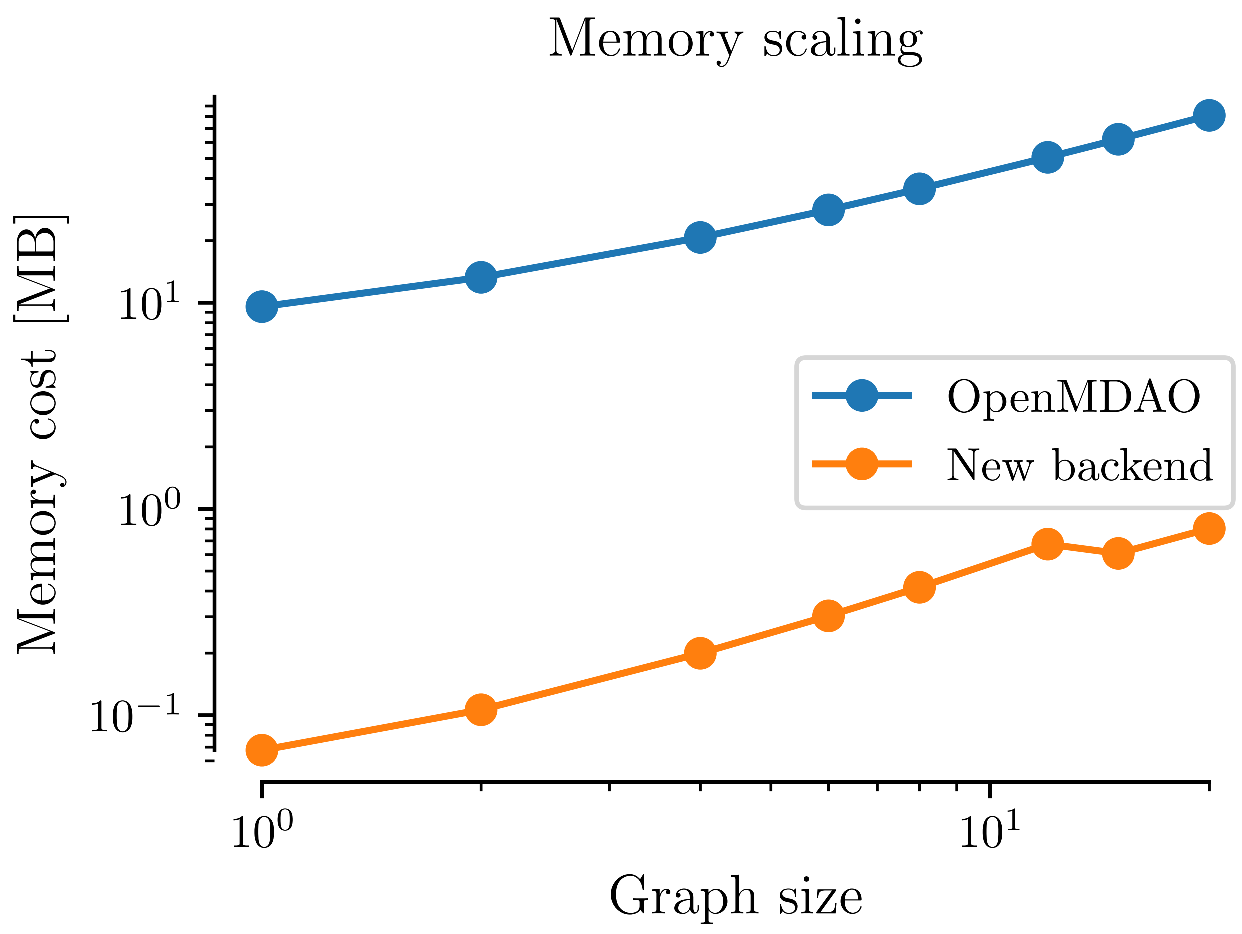
Pitt—Peters method

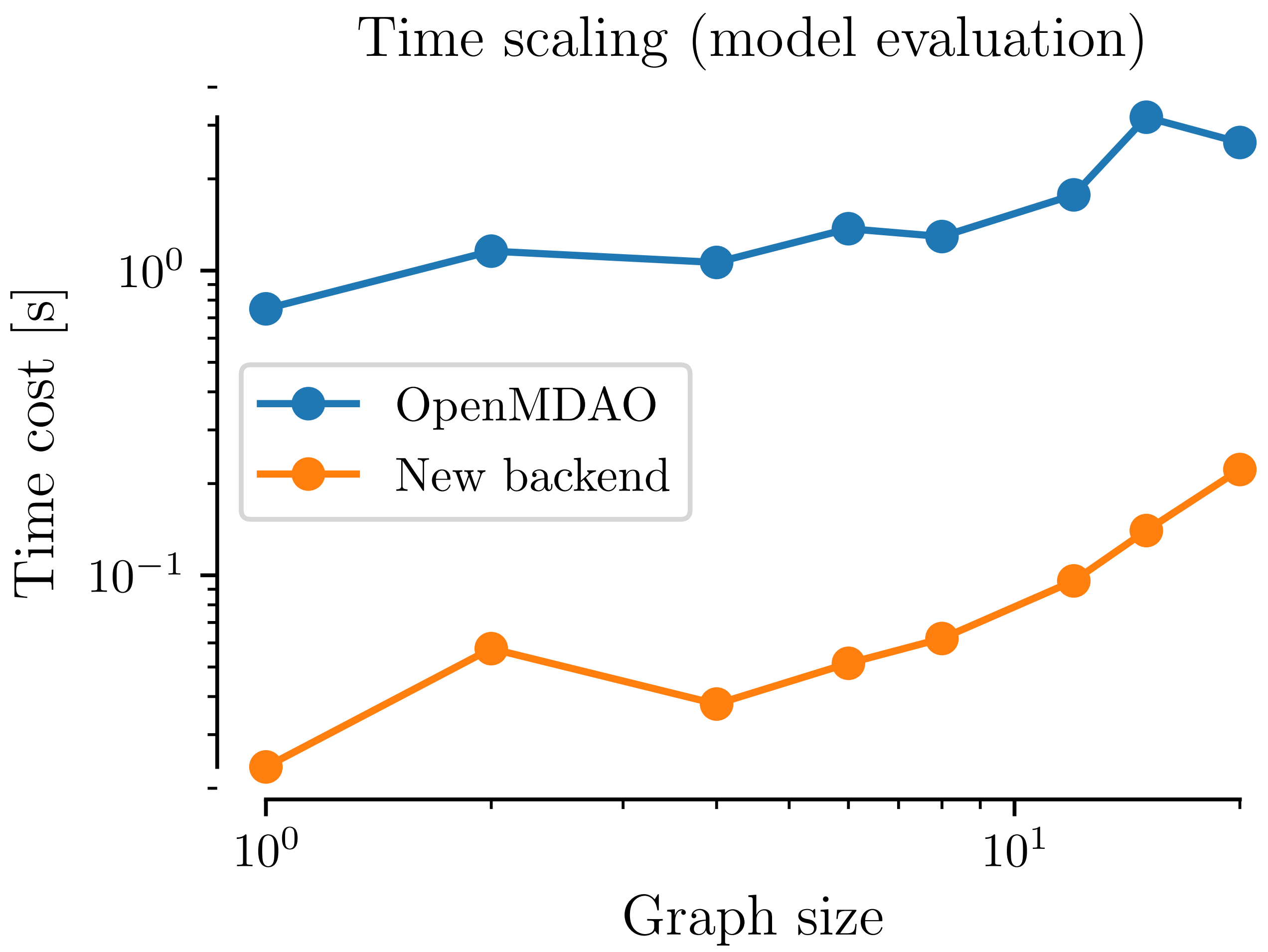


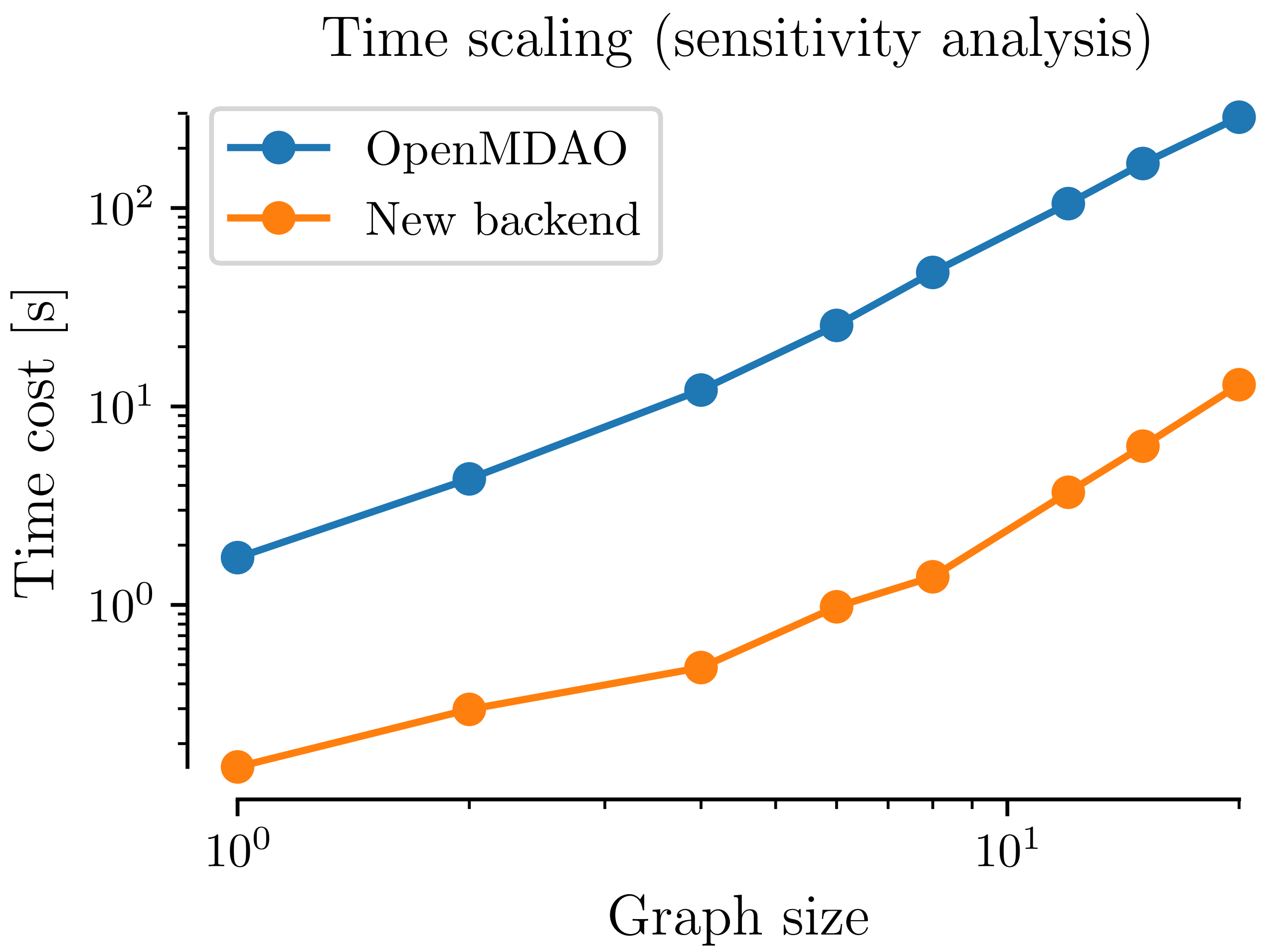
Our first back-end that generated OpenMDAO code was slow; our second back-end generated optimized Python code



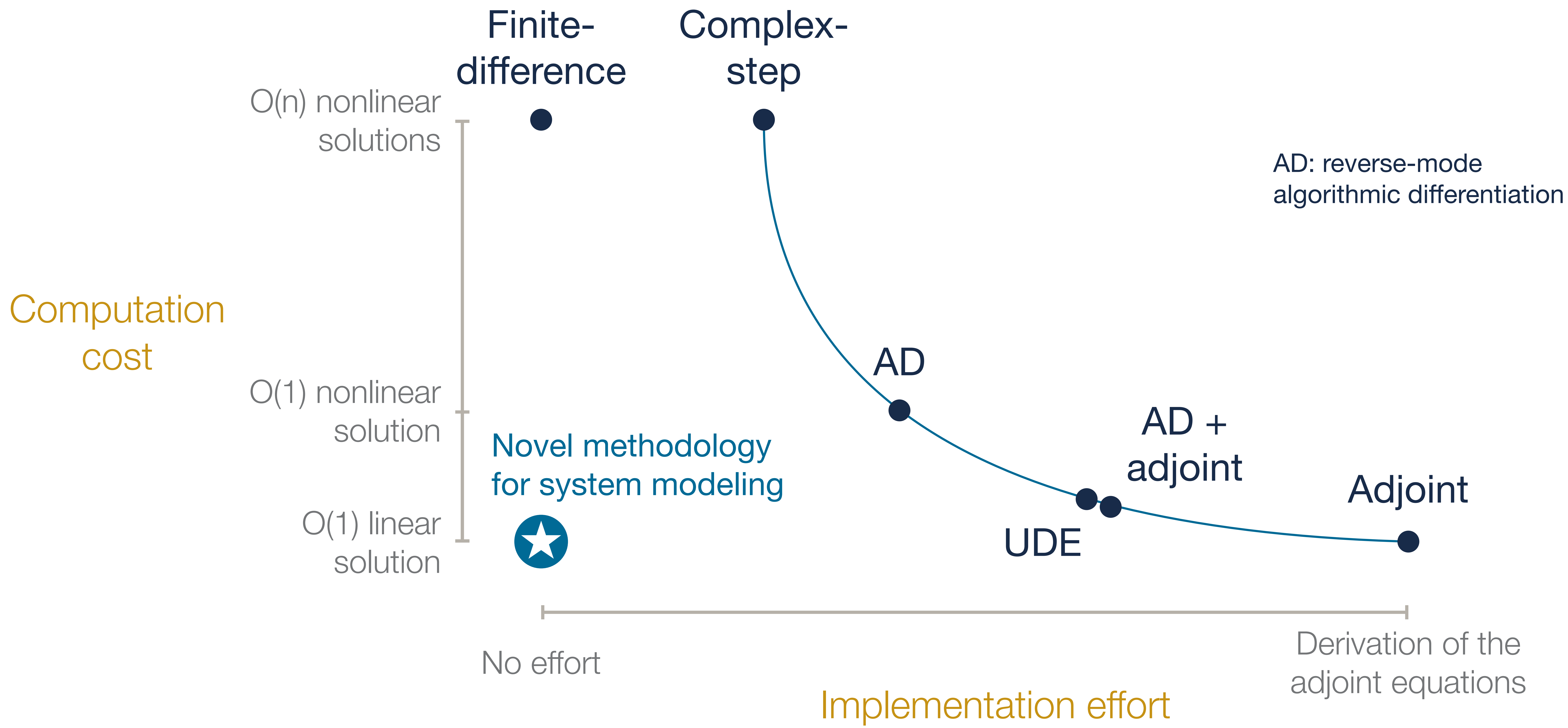
The second back-end reduces time & memory by >10x compared to the OpenMDAO back-end







Sensitivity analysis methods—Pareto front



Outline

Review of sensitivity analysis methods

Novel methodology for system modeling

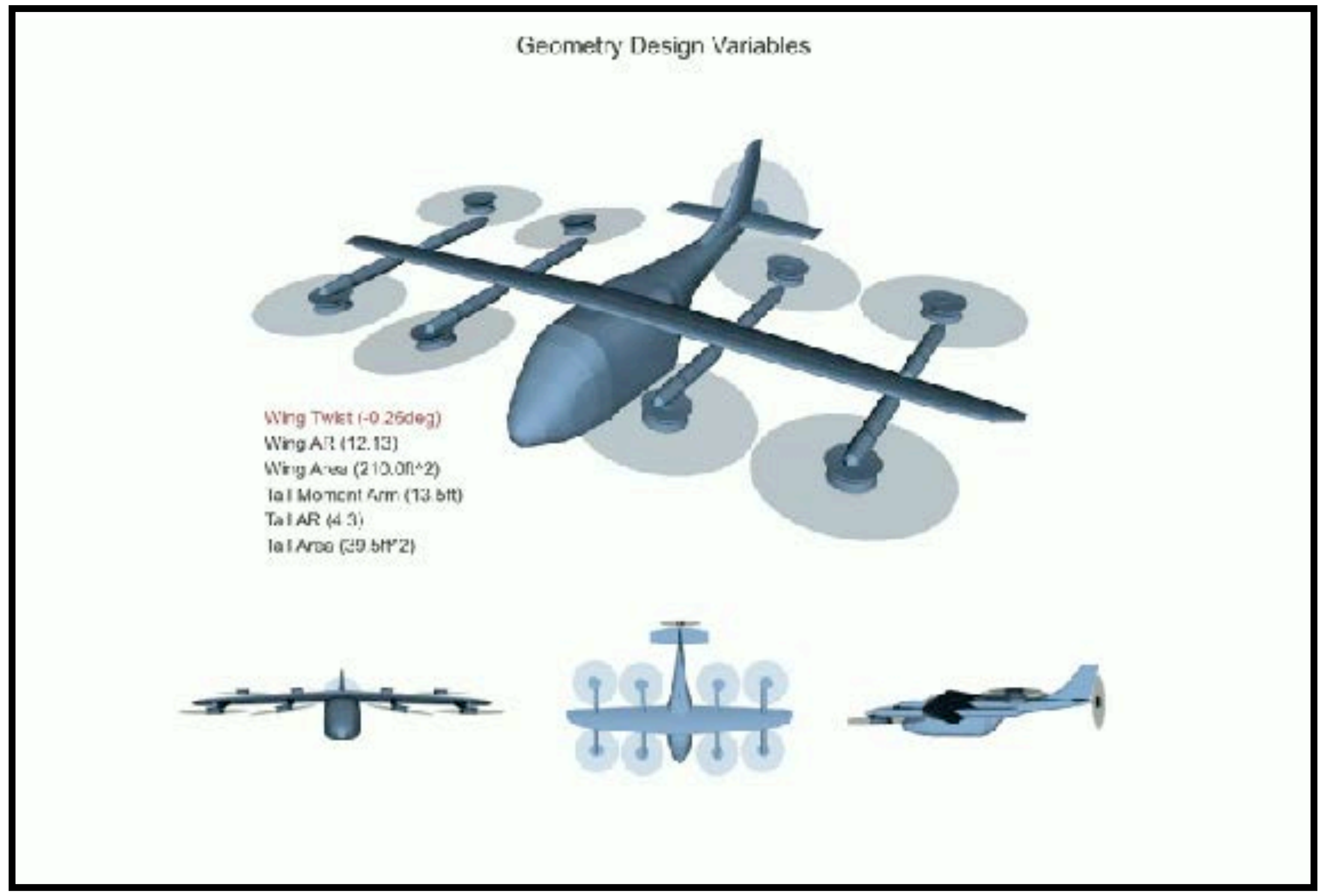
- Demonstration on UAM air taxi design problem

UCSD-led NASA University Leadership Initiative (ULI) project

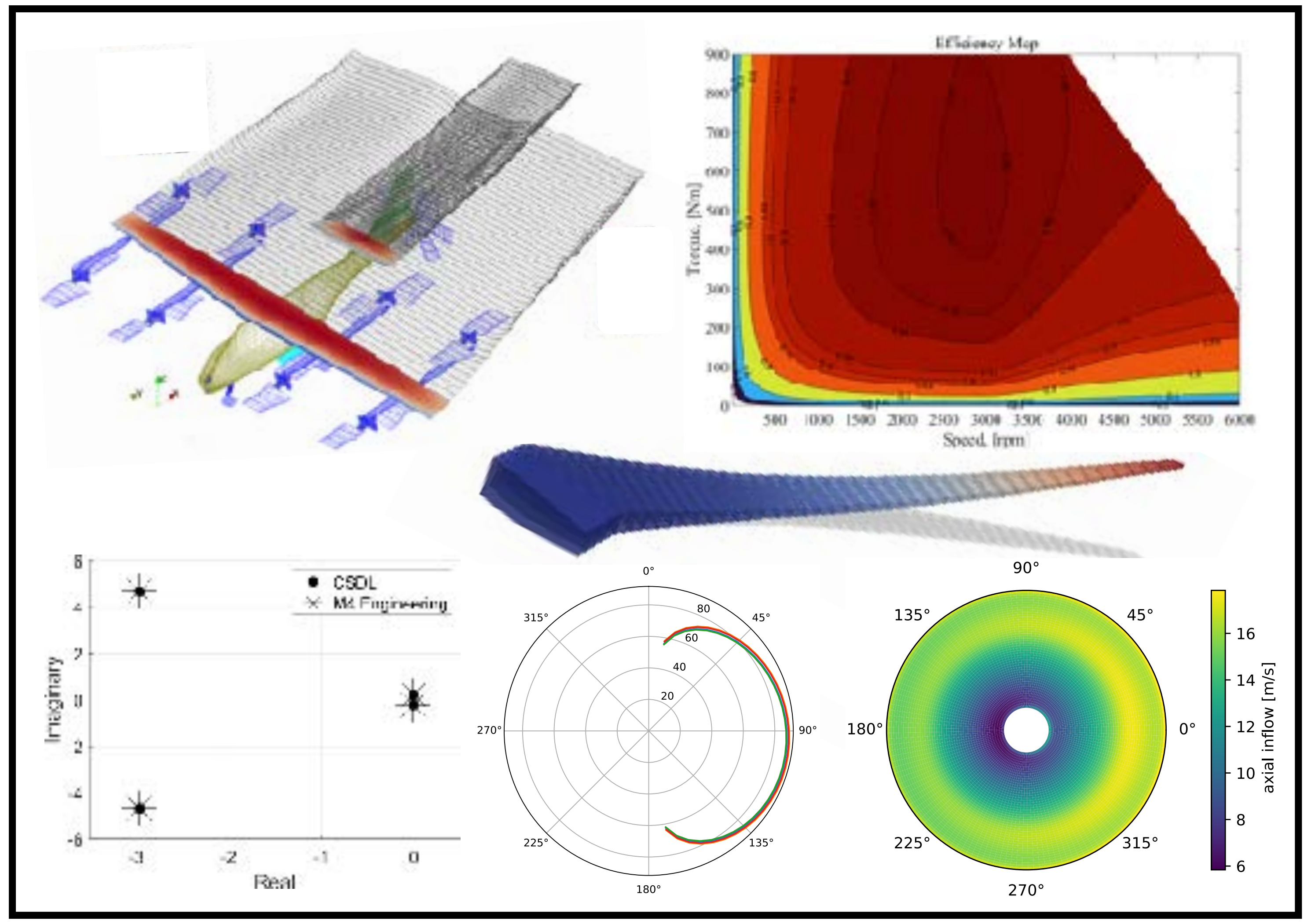
- Three-year project (2021-2024)
- 10 investigators, ~30 students
- Peer review board (industry, government)
- Investigates low-/mid-/high-fidelity large-scale MDO

NASA ULI (Y1): we developed an aircraft design tool called CADDEE within the new system modeling methodology

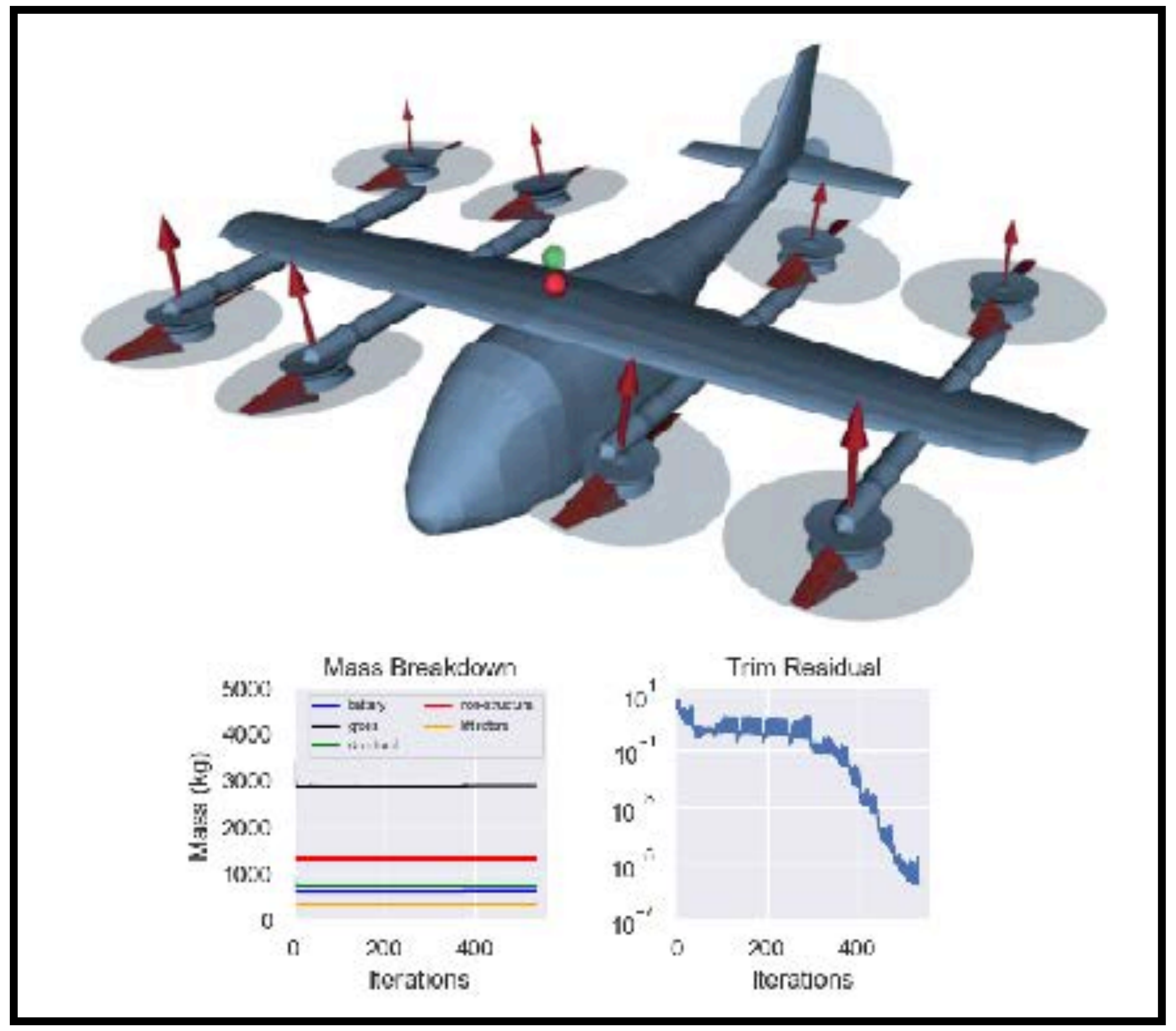
Comprehensive Aircraft high-Dimensional DEsign Environment



Geometry parametrization with kinematic relations

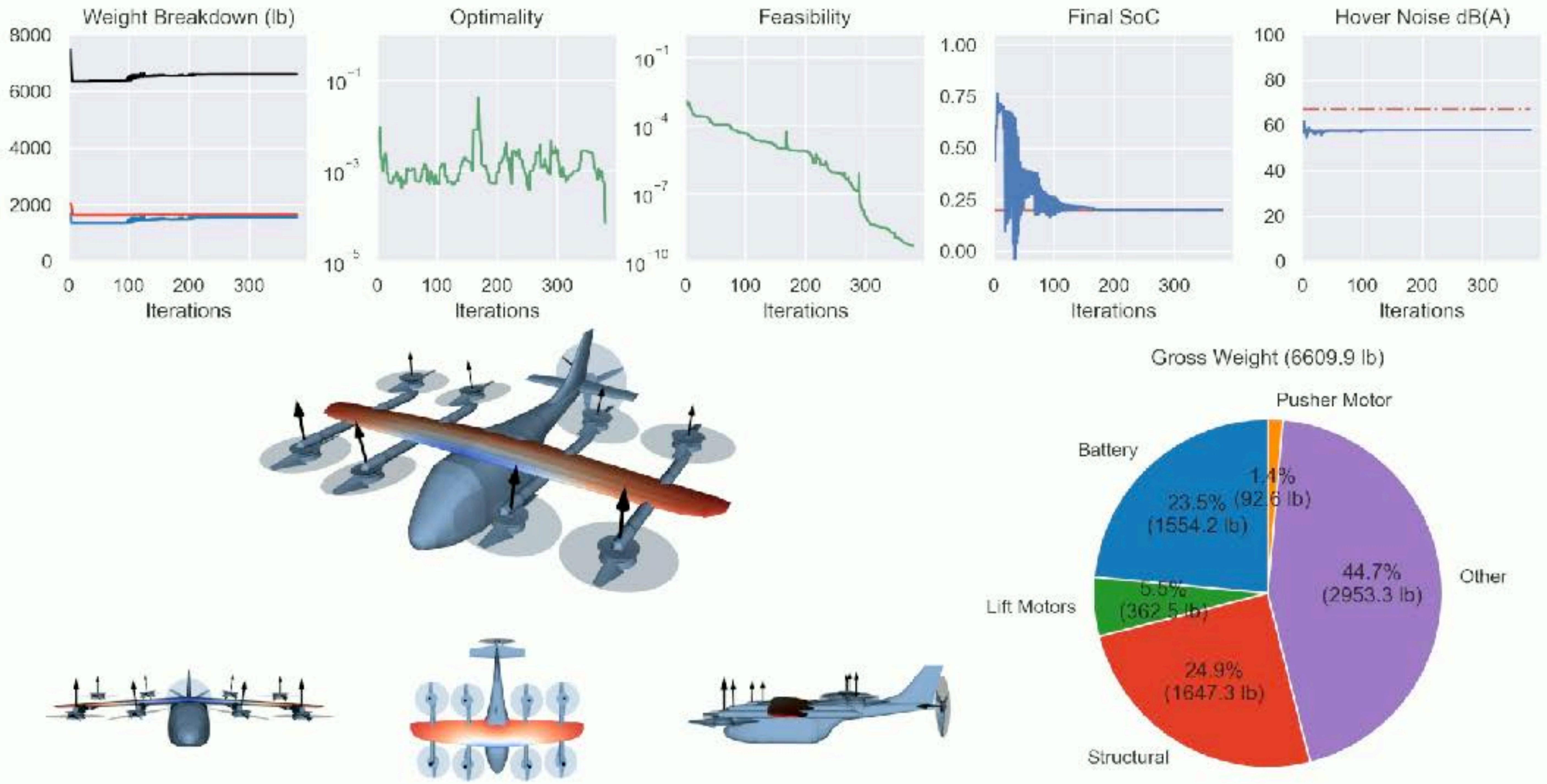


Modular interfaces to physics-based analyses

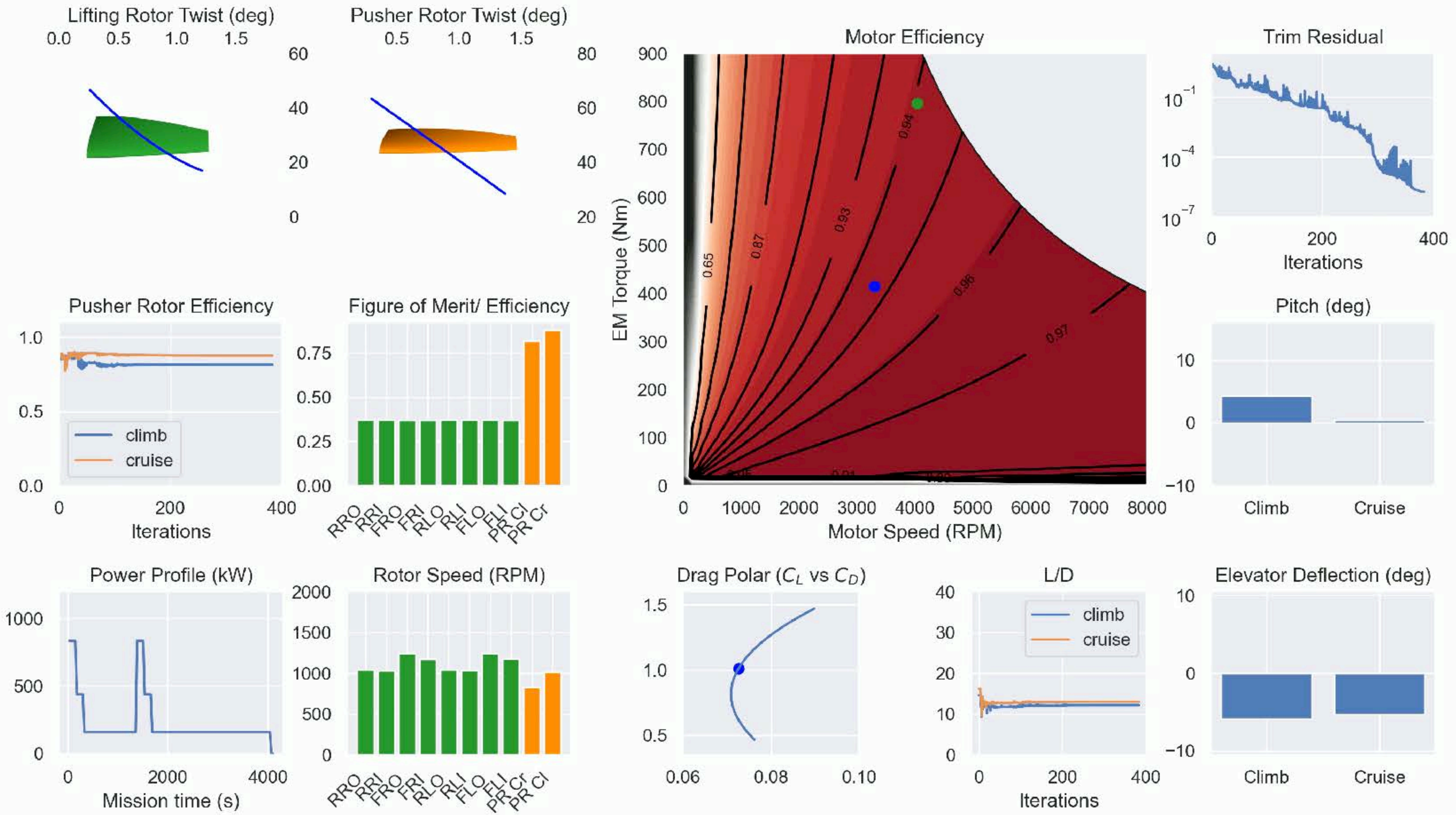


Full-mission simulation with trim-state and transient segments

NASA ULI (Y1): we demonstrated full-configuration, large-scale MDO of an air taxi with only ~30 min runtime



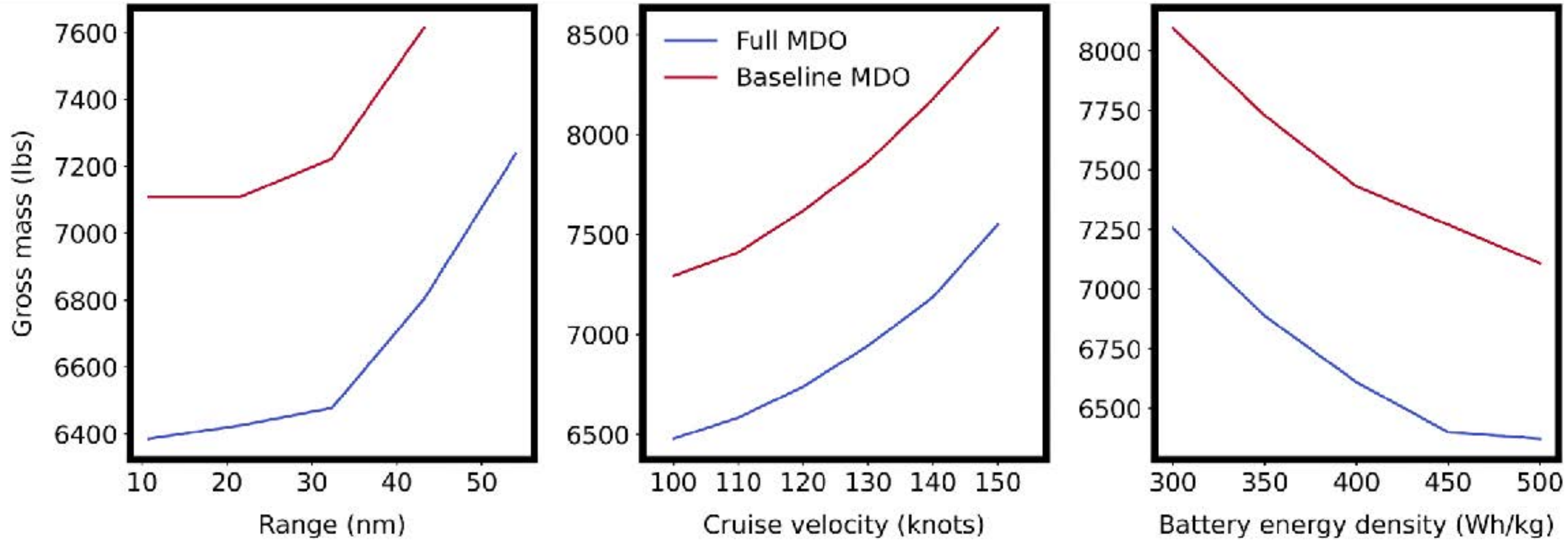
Objective	Gross weight			
Design variables	Rotor radii	9	Battery location	1
	Blade twist	36	Battery mass	1
	Blade chord	10	Motor length	9
	Rotor location	2	Motor diameter	9
	Wing area	1	Lift rotor speed	8
	Wing AR	1	Propeller speed	2
	Wing twist	5	Angle of attack	2
	Horizontal tail area	1	Tail trim angle	2
	Horizontal tail AR	1	Cruise altitude	1
	Horizontal tail location	1		
	Total design variables:	102		
Constraints	Trim residual norm	1	Sound pr. level	1
	Final state of charge	1	Stall speed	1
	Rotor tip clearance	4	Max. motor torque	2
	Motor left-right symm.	4	Rotor lateral symm.	4
	Final climb altitude	1		
	Total constraints:	19		



Lifting Rotor Twist (deg)
Pusher Rotor Twist (deg)
Motor Efficiency
Trim Residual
Iterations
Pusher Rotor Efficiency
climb
cruise
Iterations
Figure of Merit/ Efficiency
RRO RRI FRO FRI RLO RLI FLO FLI PR Cl PR Cr
EM Torque (Nm)
Motor Speed (RPM)
Pitch (deg)
Climb
Cruise
Power Profile (kW)
Mission time (s)
Rotor Speed (RPM)
RRO RRI FRO FRI RLO RLI FLO FLI PR Cr PR Cl
Drag Polar (C_L vs C_D)
L/D
climb
cruise
Iterations
Elevator Deflection (deg)
Climb
Cruise

NASA ULI (Y1): we showed that we can perform parameter sweeps using the large-scale MDO algorithm

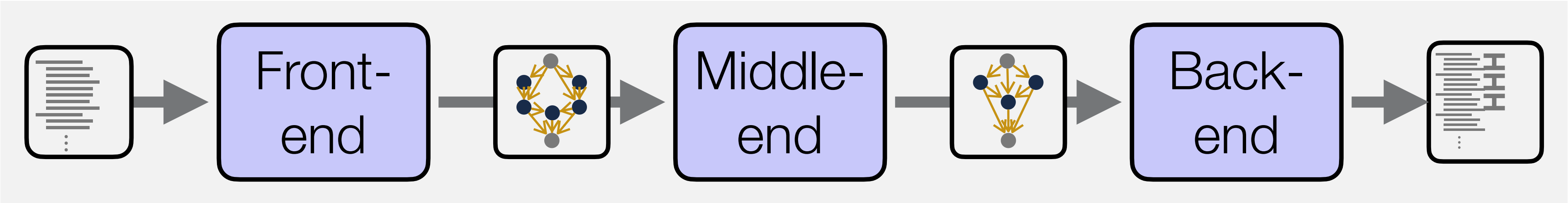
Design variable	Ct.	Baseline MDO	Full MDO
Rotor radii	9		●
Blade twist	36		●
Blade chord	10		●
Rotor location	2		●
Wing area	1		●
Wing AR	1		●
Wing twist	5		●
Horizontal tail area	1		●
Horizontal tail AR	1		●
Horizontal tail location	1		●
Battery location	1		●
Battery mass	1	●	●
Motor length	9	●	●
Motor diameter	9	●	●
Lift rotor speed	8	●	●
Propeller speed	2	●	●
Angle of attack	2	●	●
Tail trim angle	2	●	●
Cruise altitude	1	●	●
Total design variables:	102		



1. Gross mass is reduced ~10% with full MDO (this is the benefit of large-scale MDO)
2. The large-scale MDO algorithm is fast and robust enough for parameter sweeps to be completed in a few hours (enables engineer to gain insights about trade studies)

Summary

We developed a fully automated method for adjoint-based sensitivity analysis using a three-stage compiler.



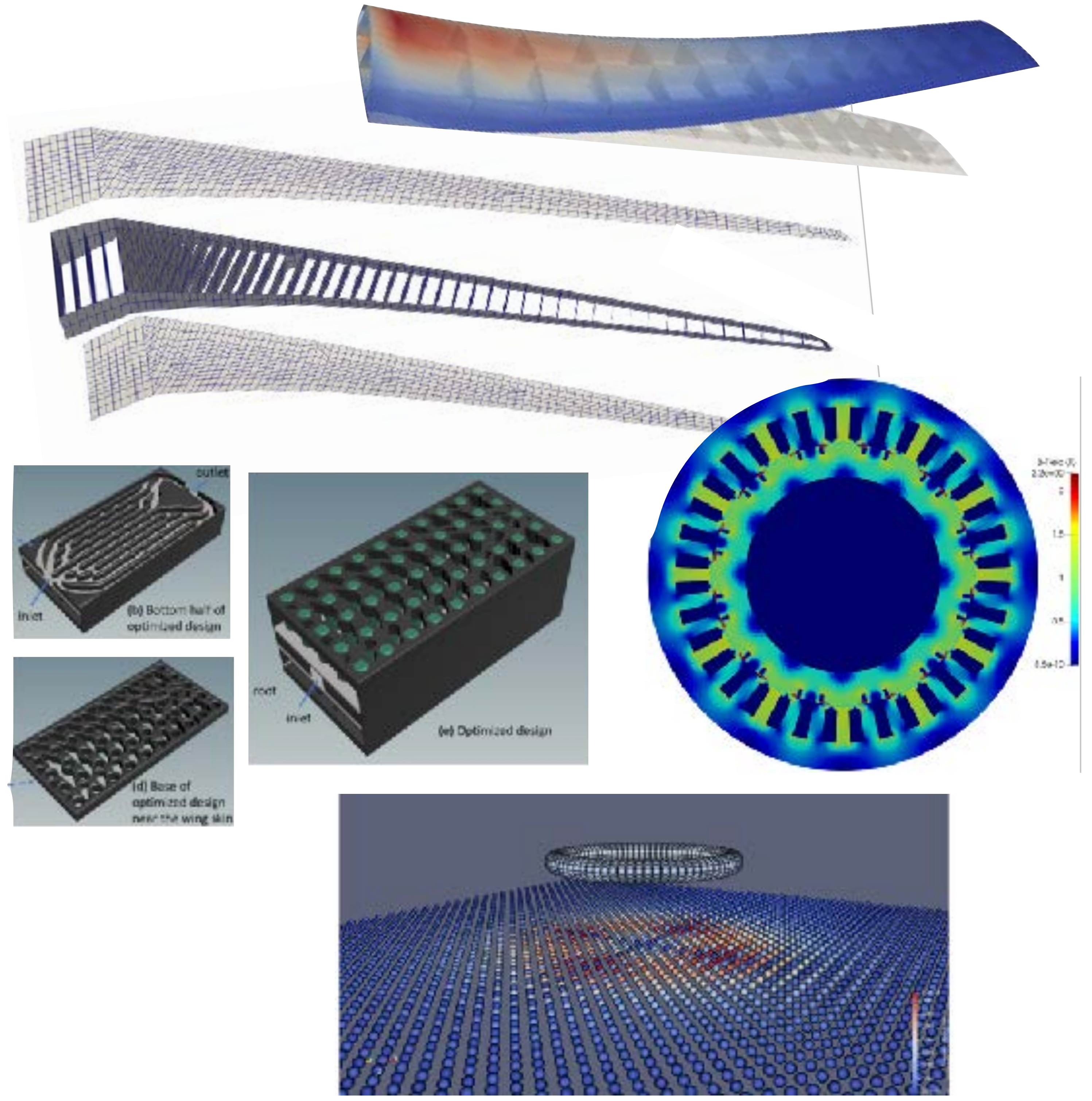
Discipline	Analysis	Timeline	Verification	Validation
Aerodynamics	VLM (lifting surface)	TC1	AS1	-
	UVLM (lifting surface)	TC2/3	-	PS
	BEM, PP (rotors)	TC1	AS1	AS2, SPEC
	VPM with boundaries (all)	TC2/3	SELF	PE
Acoustics	Tonal	TC1	AS1	PE
	Tonal (unsteady freq-domain)	TC2	-	SPEC
	Broadband	TC1	PS	PE
	Broadband (new empirical)	TC2/3	-	PE
Structures	Regression on M4 structures studio data (weights)	TC1	SELF	AS2, SPEC
	Reissner-Mindlin	TC2/3	AS1	-
	IMGA	TC2/3	AS1	AS2, TBD
	ShellMesh	TC2/3	AS2	-
Stability & Control	S&C analysis	TC1	AS1	SPEC
	Controller design & closed-loop analysis	TC2/3	-	PS, PE
Motors	Low-fidelity sizing & performance models	TC1	AS2	AS2, SPEC
	FEniCS EM model	TC2/3	AS1/2	SPEC
Batteries	ECM	TC1	AS1	EXP
	Pack sizing	TC1	-	SPEC
	Thermal model	TC2	TBD	EXP
	Pack topology optimization	TC3	TBD	TBD
Coupled & system-level	CADDEE	TC1	-	AS2
	Aero/structures/acoustics	TC2/3	-	AS2
	TBD	TC1/2/3	-	SPEC

This automation enabled, in one year, the development of:

- CADDEE, an aircraft design framework (WEIS)
- Set of low-fidelity aircraft models (WISDEM) with V&V
- A full-configuration air taxi large-scale MDO algorithm

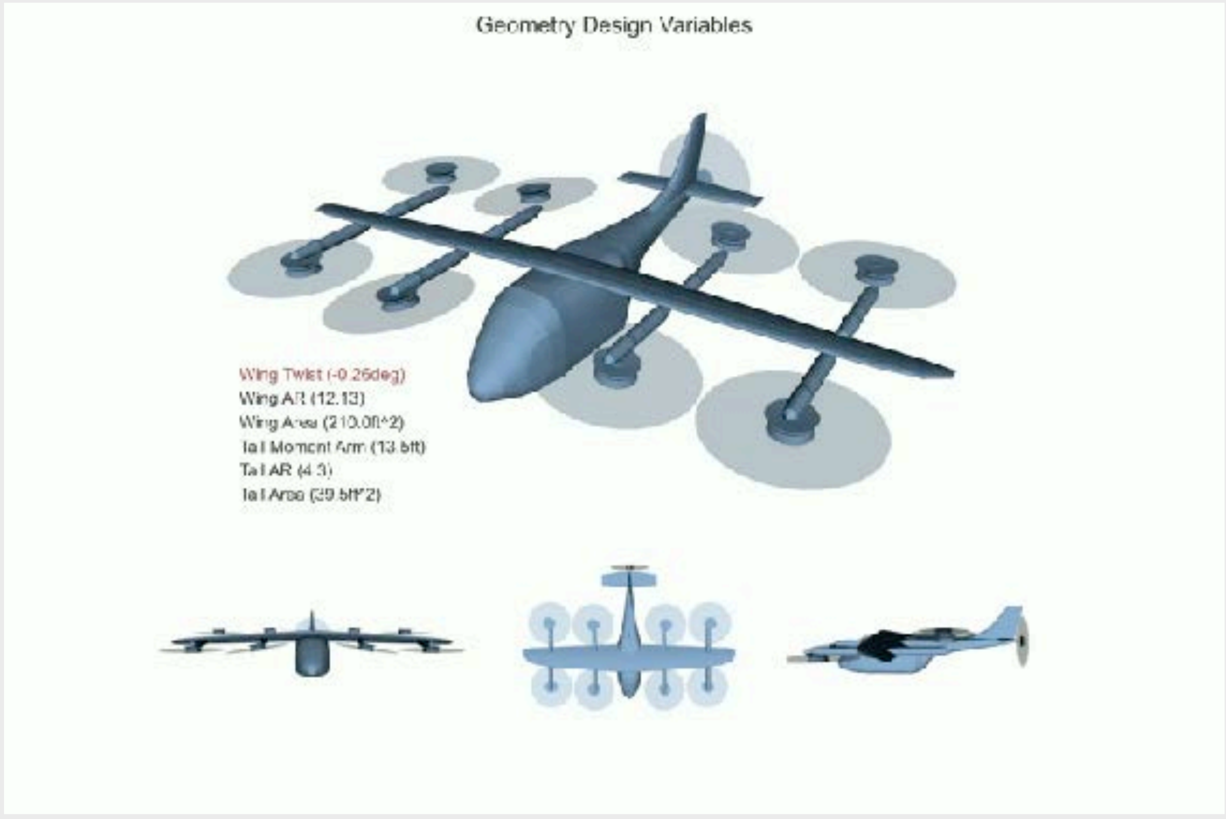
In year 2: we will add mid-fidelity models (OpenFAST)

Our ongoing work builds on this new methodology



Mid/high-fidelity MDO of air taxi

Years 2 and 3 of NASA ULI

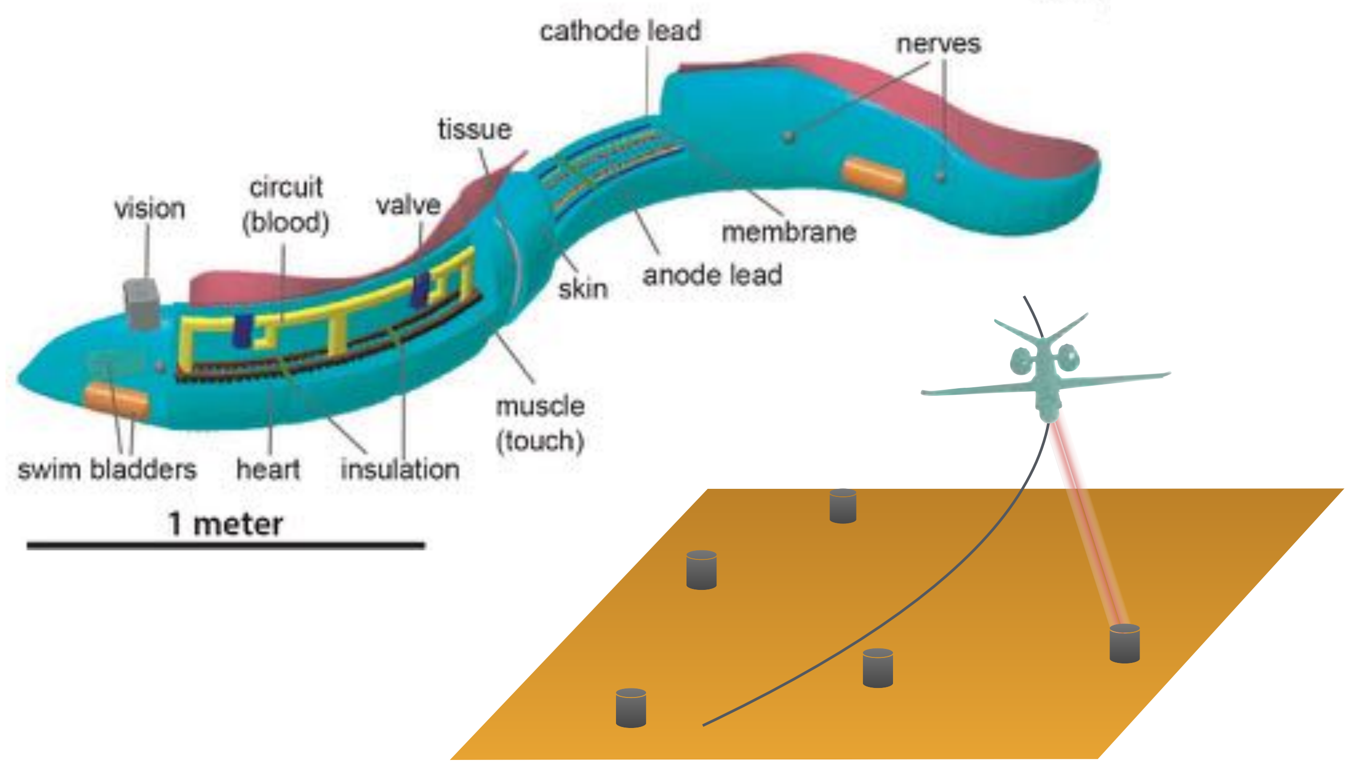


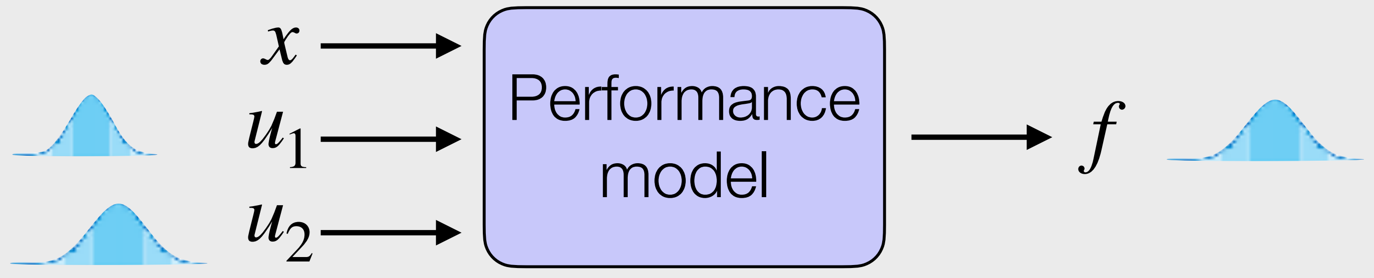
Further development of CADDEE

Explore applications beyond aircraft (wind turbines?)

New applications of large-scale MDO

CSDL/CADDEE will be used for robotic fish (ONR), laser-powered UAVs (DARPA)





Uncertainty propagation using CSDL

Preliminary results show potential for 10~100x speed up using CSDL graph

Thank you!

These slides include contributions from many people, some of whom are acknowledged below.

Large-scale design optimization (LSDO) lab students: Andrew Fletcher, Victor Gandarillas, Alexander Ivanov, Anugrah Jo Joshy, Nicholas Orndorff, Marius Ruh, Darshan Sarojini, Luca Scotzniovsky, Mark Sperry, Bingran Wang, Jiayao Yan.

ULI collaborators: Isaac Asher, Jeff Chambers, David Kamensky, Alicia Kim, Seongkyu Lee, Shirley Meng, Chris Mi, Andrew Ning, Tyler Winter

http://lsdo.eng.ucsd.edu • http://uli.eng.ucsd.edu

We are grateful for financial support from the following organizations:

NASA NSF HYUNDAI MOTOR GROUP DARPA Office of Naval Research ONR Science & Technology UC San Diego JACOBS SCHOOL OF ENGINEERING Mechanical and Aerospace Engineering