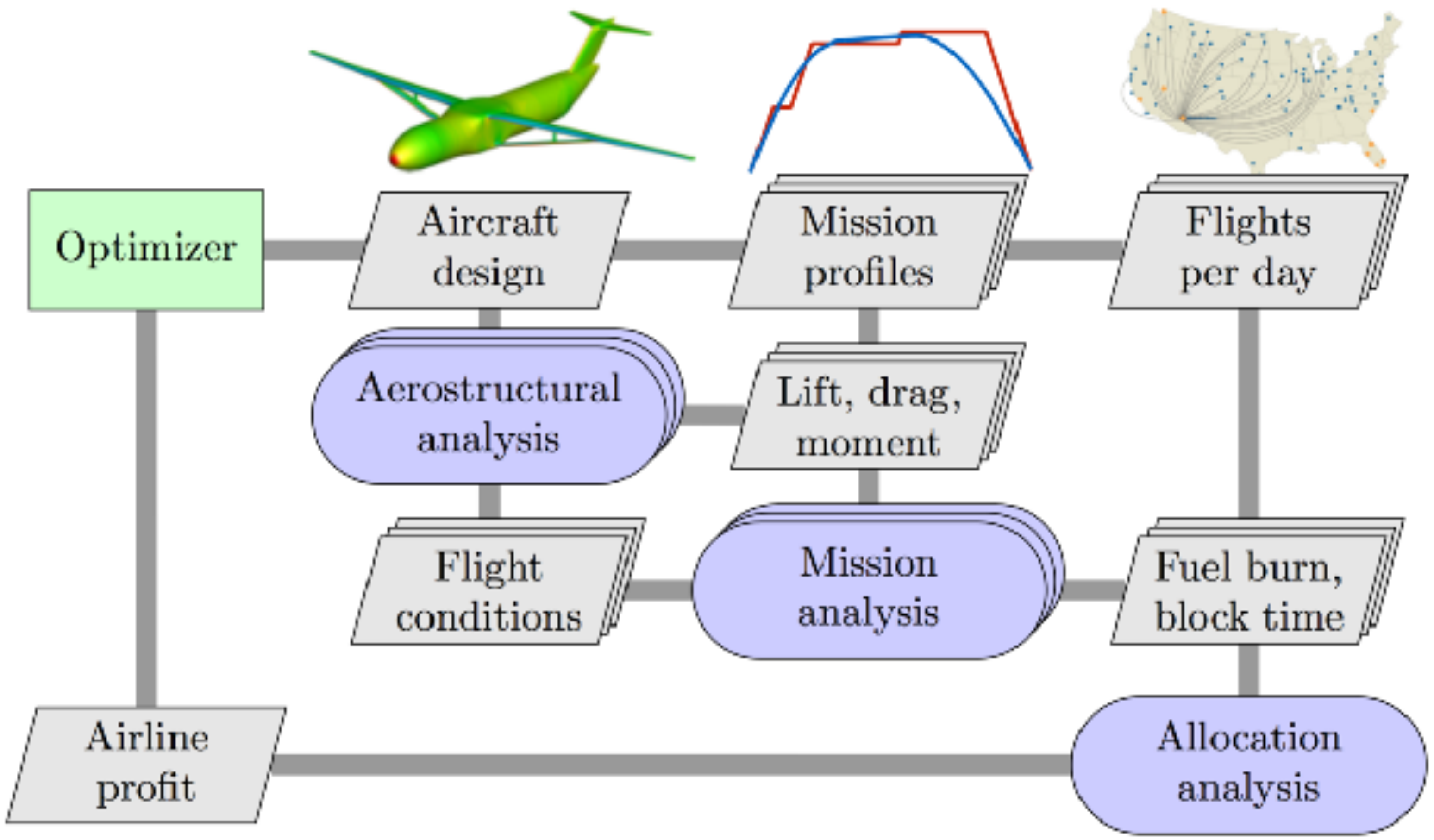
Multidisciplinary Design Optimization of Aircraft Configurations

Part 1: A modular coupled adjoint approach



Joaquim R. R. A. Martins

with contributions from John T. Hwang
and Andrew Lambe

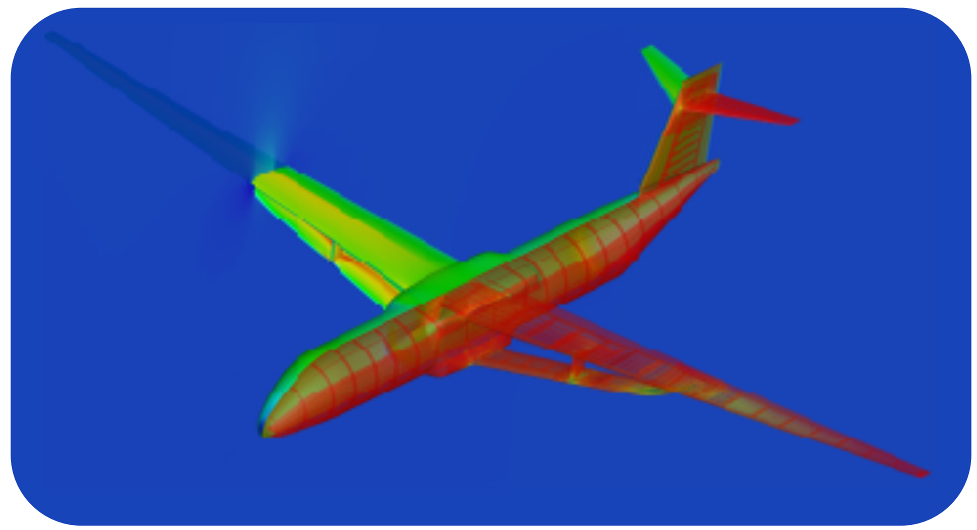
MARIE CURIE ACTIONS

The von Karman Institute for Fluid Dynamics

VKI Lecture Series • Brussels, Belgium • May 23, 2016

Once numerical simulations are developed, they can be used for design optimization

wing span
airfoil shapes
structural sizing



fuel burn

structural stresses

design changes

Design optimization problem:

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

Aircraft are performance critical and hence numerical optimization is especially valuable

JAL B787 climbing after takeoff from SAN • © 2013 J.R.R.A. Martins

Complex systems require the consideration of multiple disciplines, hence MDO was born

Multidisciplinary Design Optimization of Aircraft Configurations

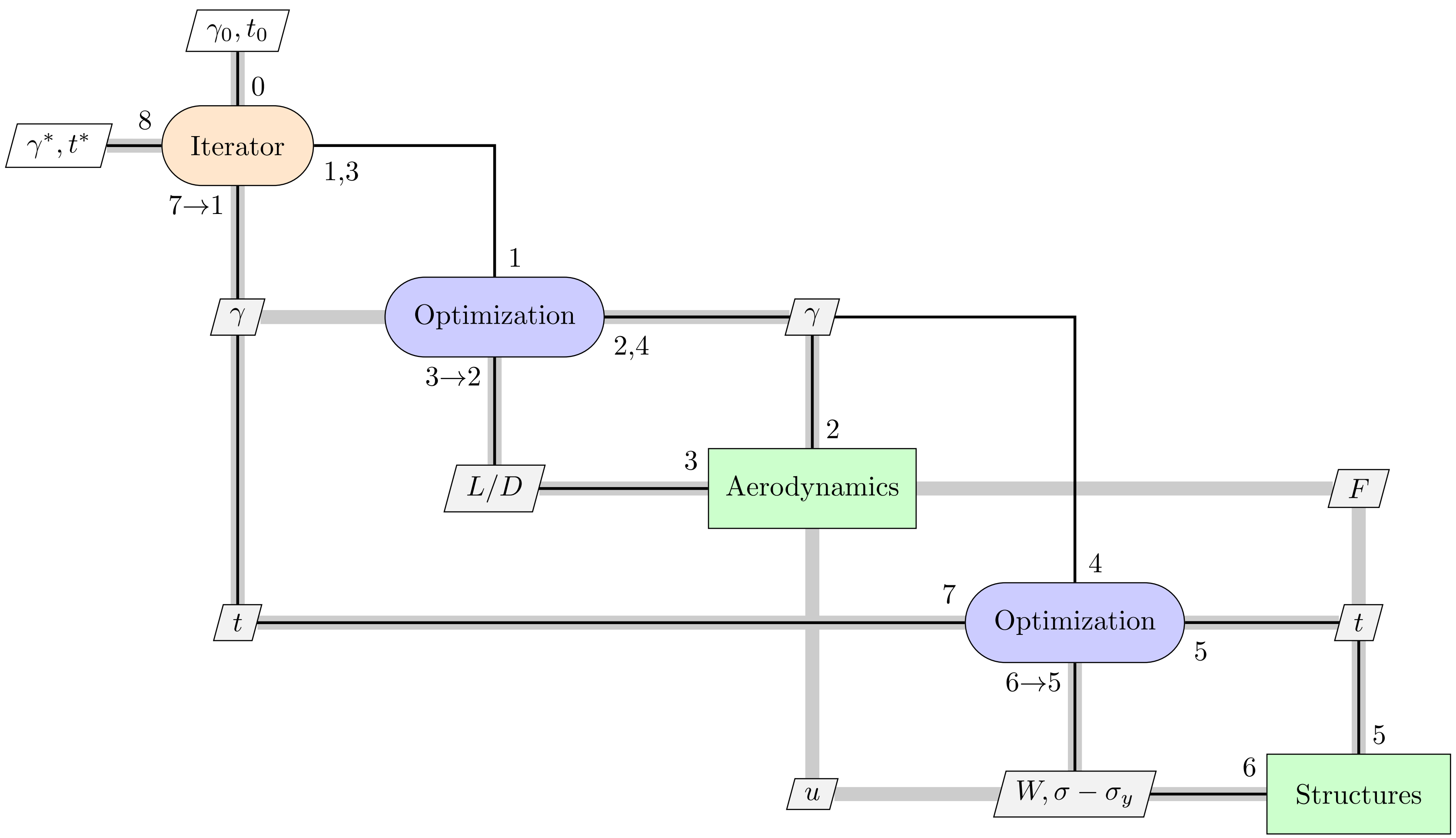
Part 1: A modular coupled adjoint approach

- A review of MDO architectures
- A new MDO architecture
- Applications

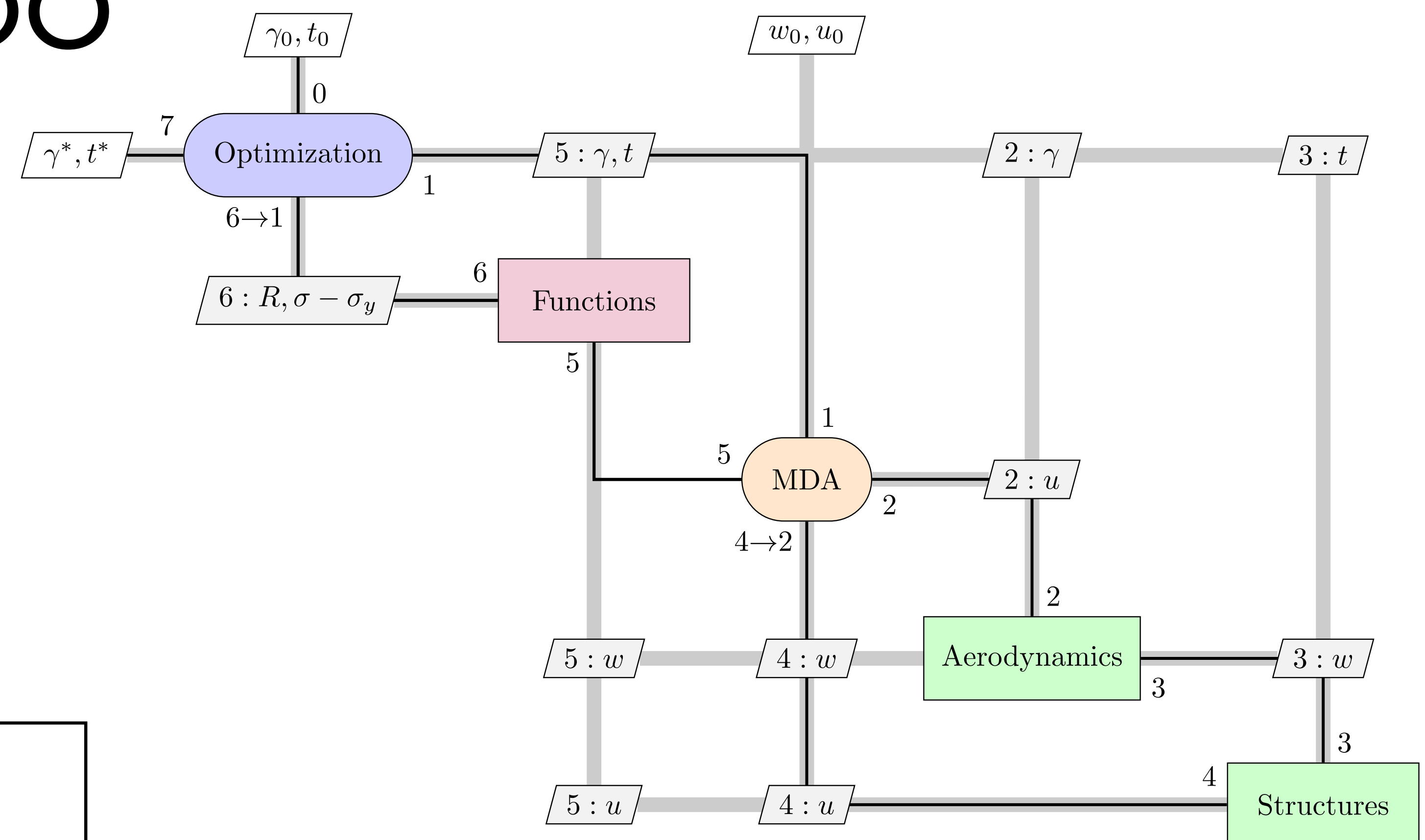
A review of MDO architectures

Sequential optimization is equivalent to coordinate descent

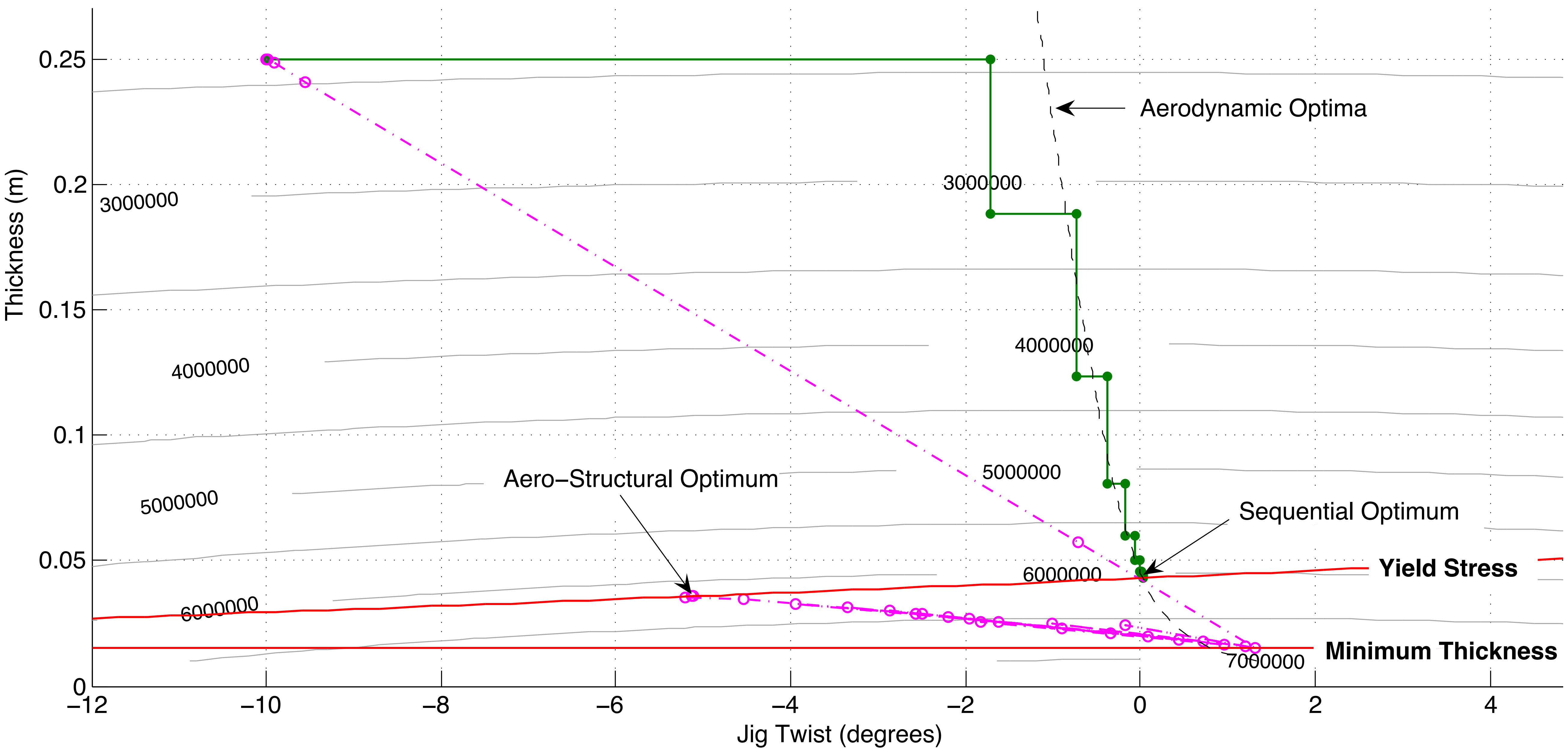
Sequential optimization



MDO

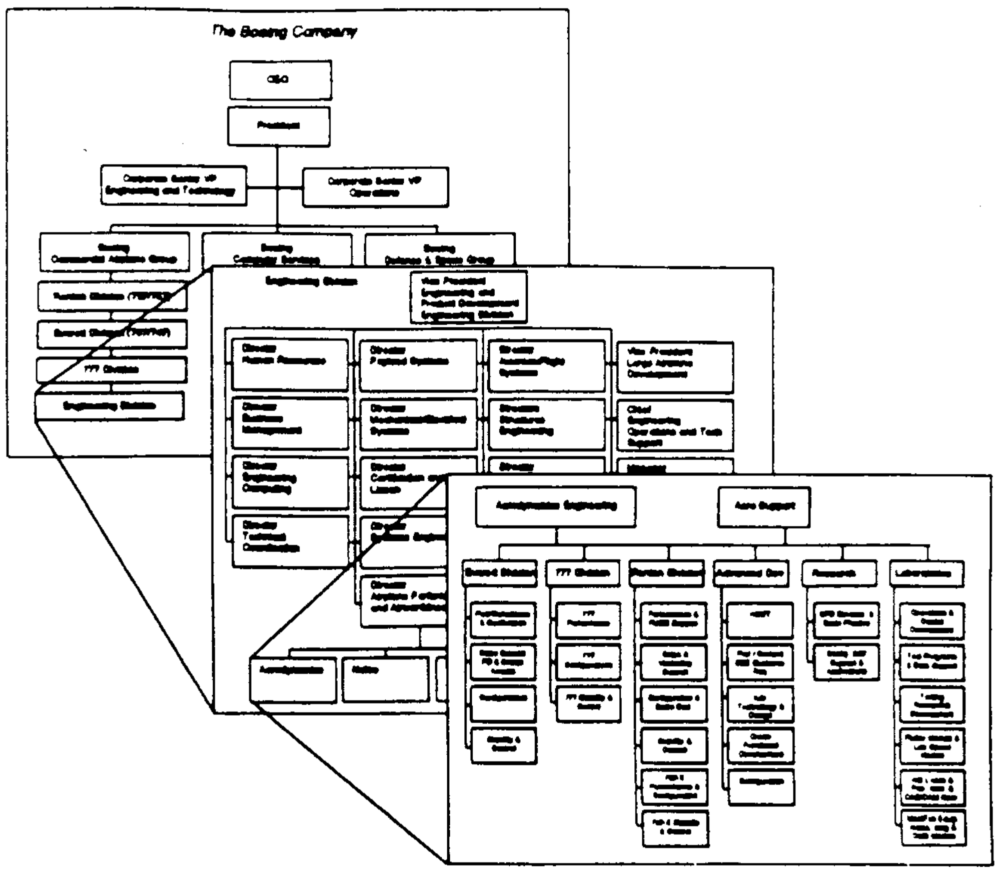


Sequential optimization fails to find the multidisciplinary optimum

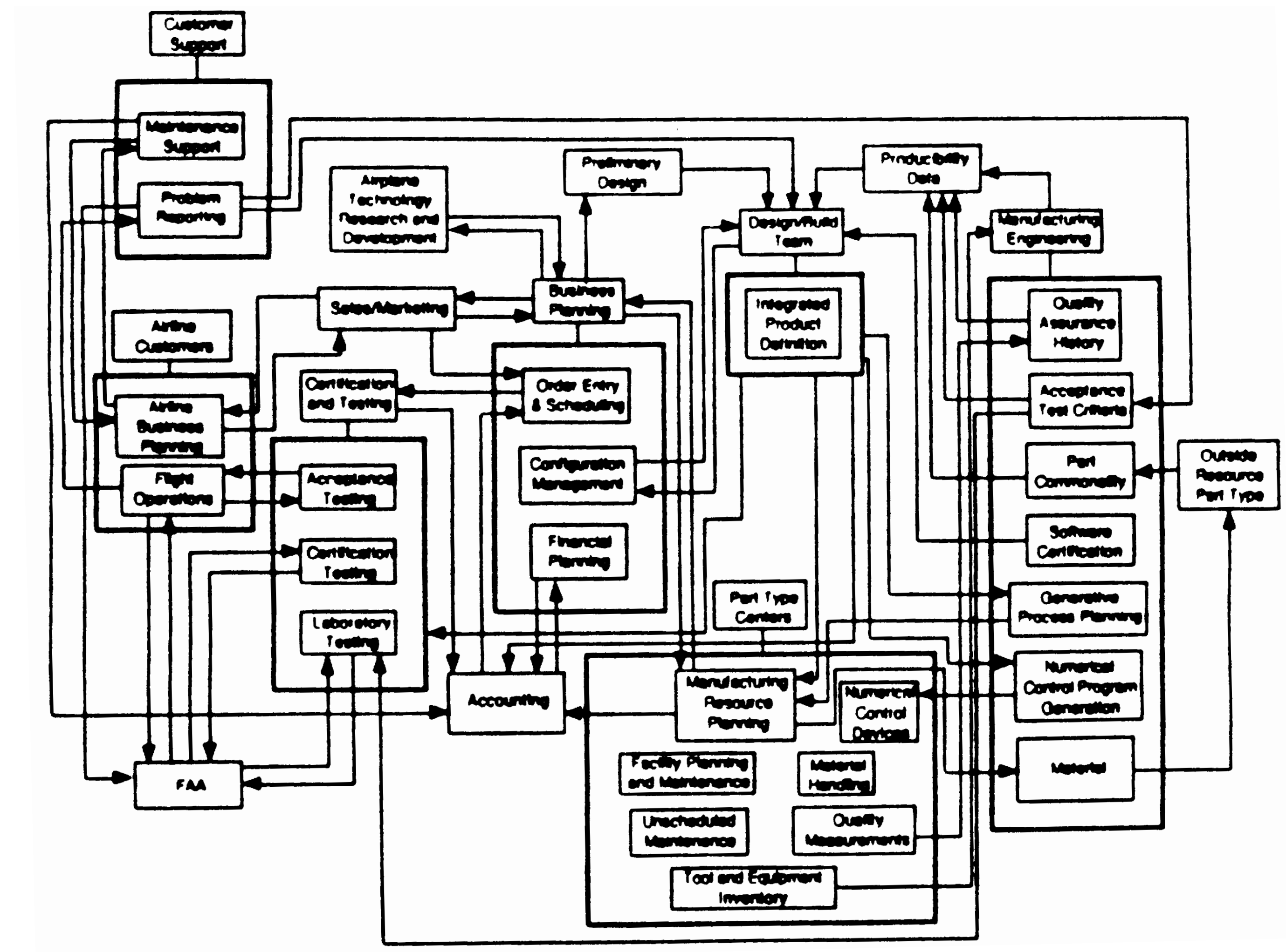


[Chittick and Martins, *Struct. Multidiscip. O.*, 2008]

The design process in industry is not tailored towards MDO

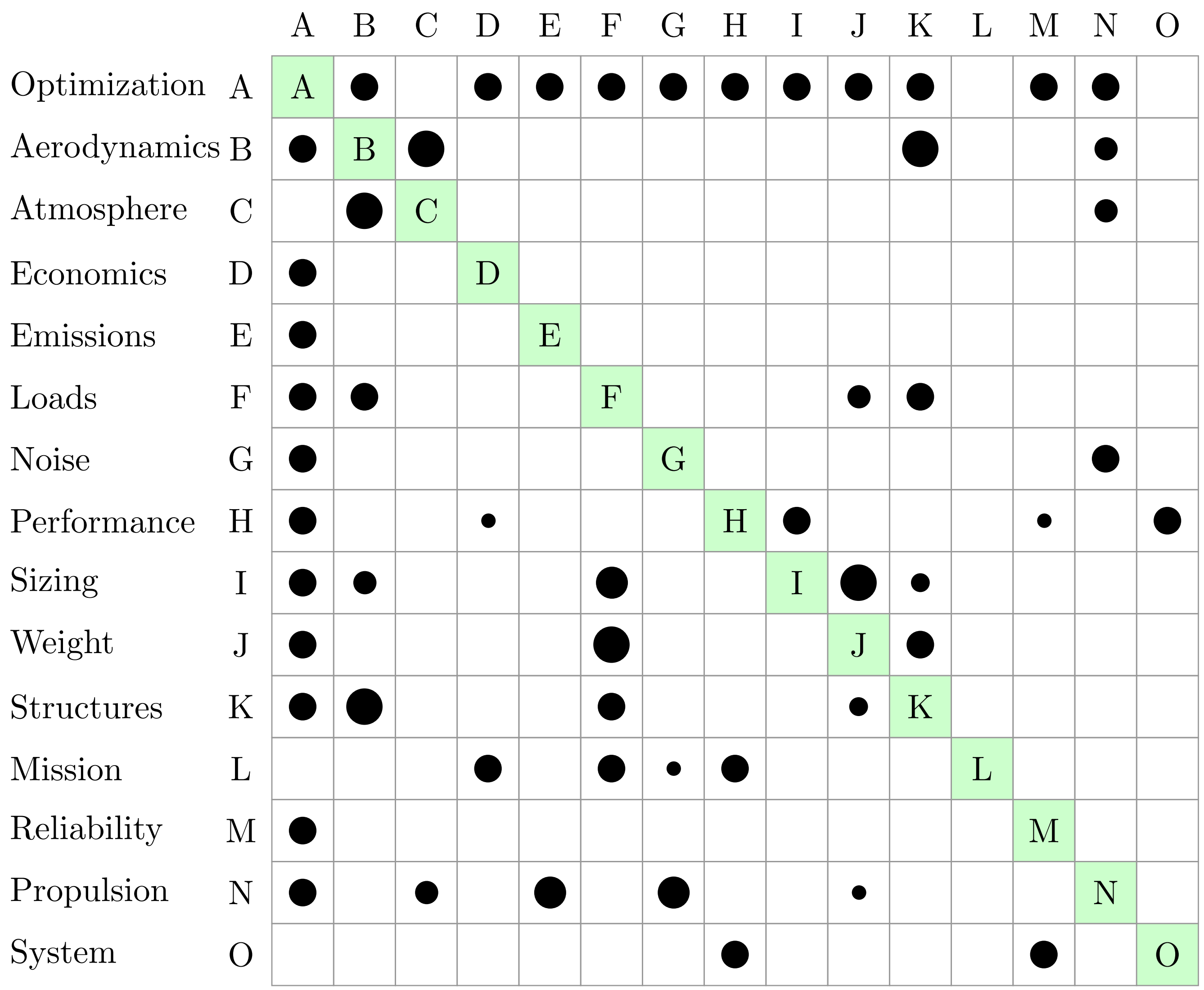


Personnel hierarchy



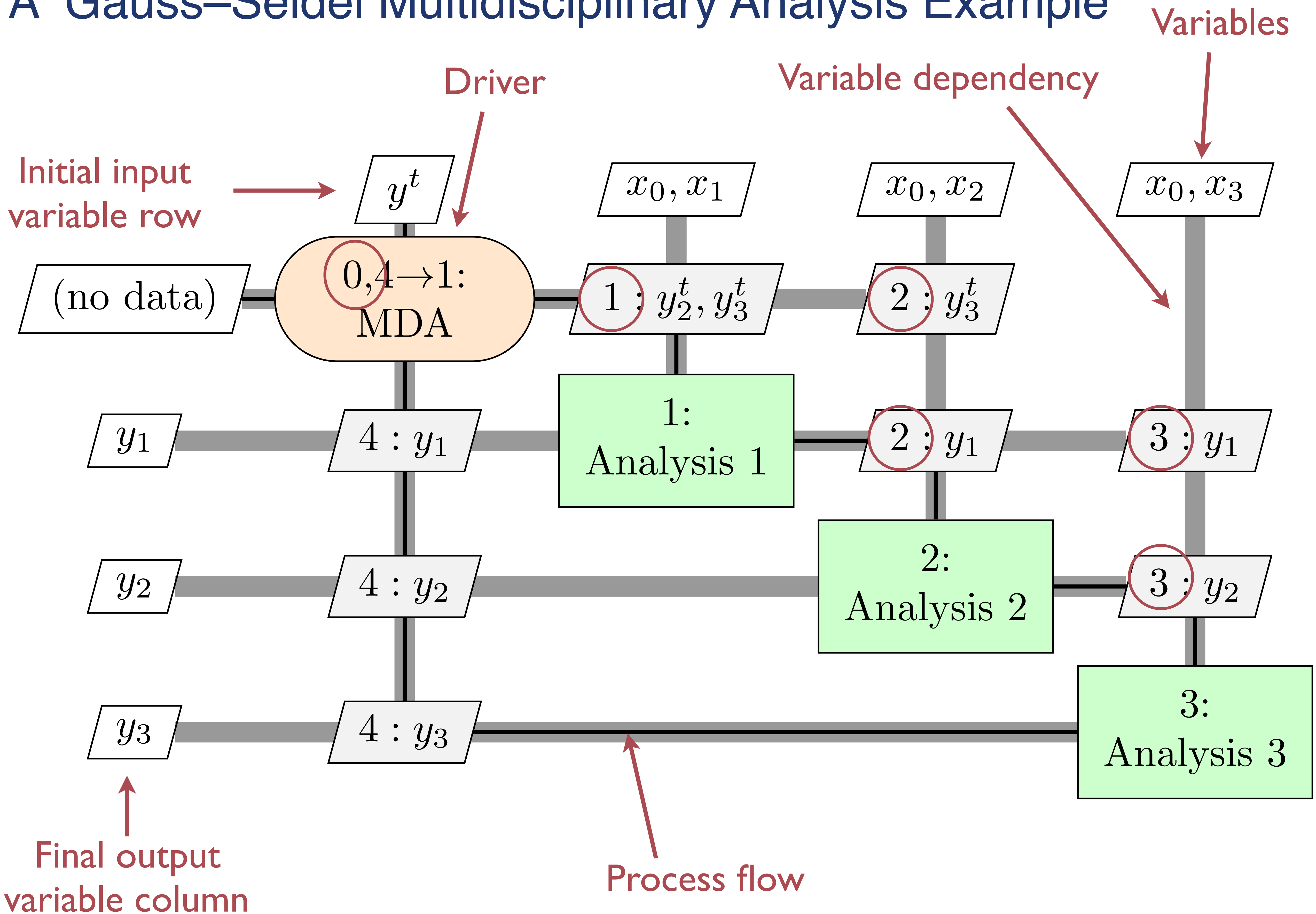
Design process

The N2 Diagram and Design Structure Matrix (DSM)



- Components on main diagonal, coupling data on off-diagonal nodes
- Component inputs in same column, component outputs in same row
- External inputs and outputs may also be included

Extending the DSM syntax:
A Gauss–Seidel Multidisciplinary Analysis Example

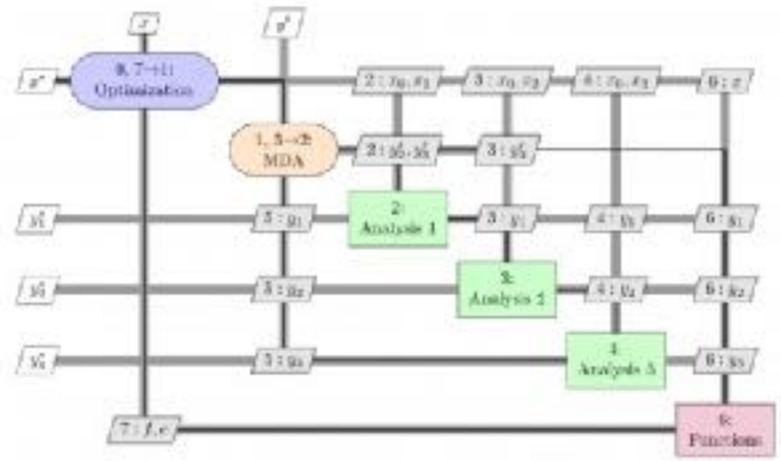


XDSM

http://mdolab.engin.umich.edu/content/xdsm-overview

XDSM Overview

View Edit Track Access control



The XDSM (eXtended Design Structure Matrix) is a tool used to visualize MDO processes. It is an extension of the classical Design Structure Matrix commonly used in systems engineering to describe the interfaces among components of a complex system. In a computational MDO context, the complex system is the MDO architecture, the components of the system are pieces of software (disciplinary analyses, optimization algorithms, surrogate models, etc.) used by the architecture, and the interfaces between components are the data exchanged by this software. Because the architecture also contains an algorithm defining the order in which the software is run, a numbering system and lines depicting the process are introduced in the diagram. In this way, we are able to capture all of the data and process flow of an architecture in a single diagram.

The full details of how to construct and interpret XDSMs are the subect of the paper cited in the footnote.[1]

For those interested in constructing XDSMs for their own work, see the attached files. We draw our diagrams using the TikZ package in LaTeX. The files contain the specific block and line styles, TikZ library imports, and formats that are common to all of our diagrams. We have also included some example diagrams and a how-to guide for the LaTeX files. Comments and suggestions are welcome.

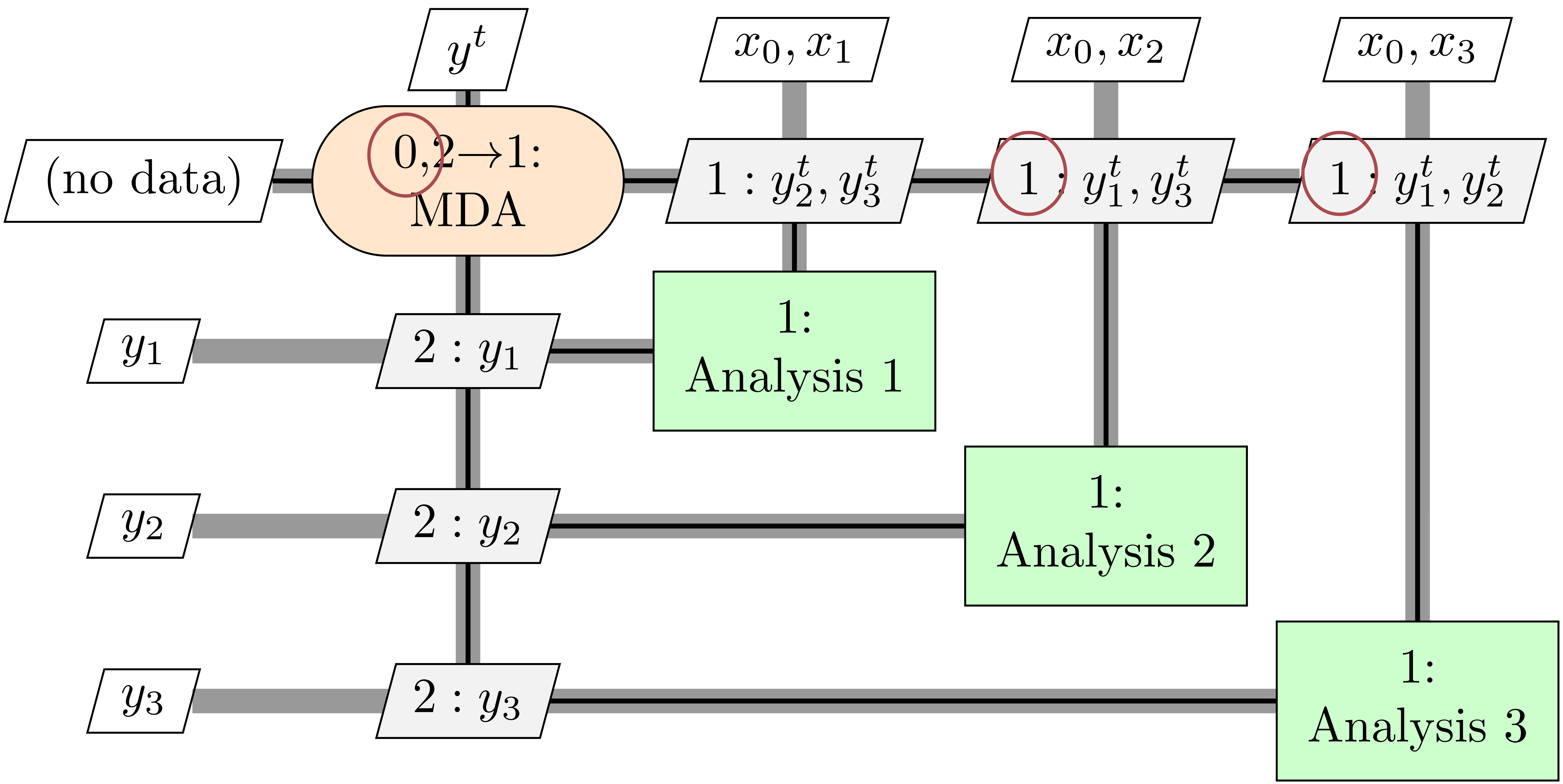
A Python script for automatically generating XDSM tex sources has been added. This script contains a class to which components and dependencies can be added, and this class automatically writes a tex file that draws the diagonal and off-diagonal blocks, as well as data flow lines. Details can be found in the Python script.

---

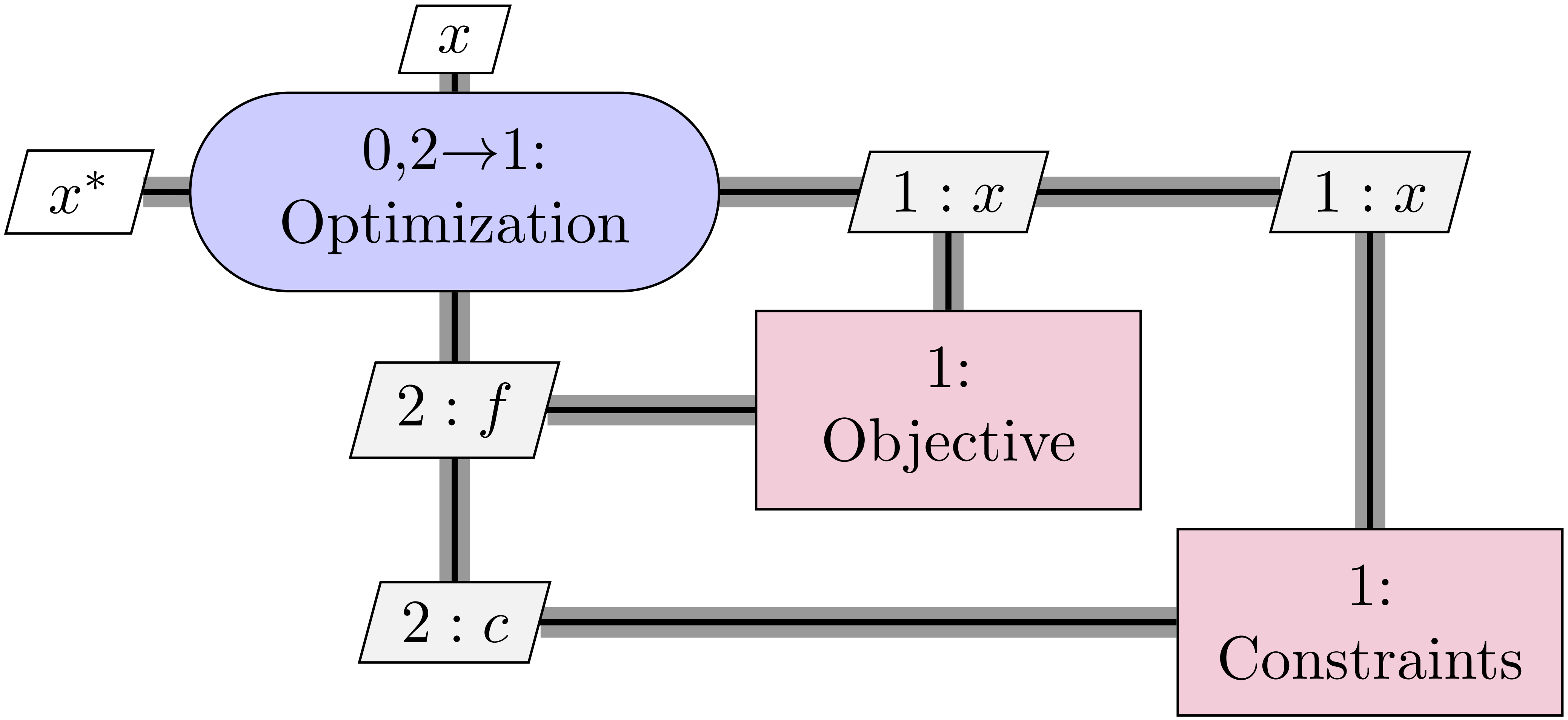
1. A. B. Lambe and J. R. R. A. Martins, "**Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes**", *Structural and Multidisciplinary Optimization*, vol. 46, no. 2, p. 273-284, 2012.

Attachment	Size
diagram_border.tex	437 bytes
diagram_styles.tex	4.81 KB
XDSM_how_to.txt	5.16 KB
CO.tex	3.07 KB
CO.pdf	59.36 KB
MDF.tex	3.04 KB
MDF.pdf	57.76 KB
XDSM.py.txt	4.83 KB

A Jacobi Multidisciplinary Analysis Example



An Optimization Problem



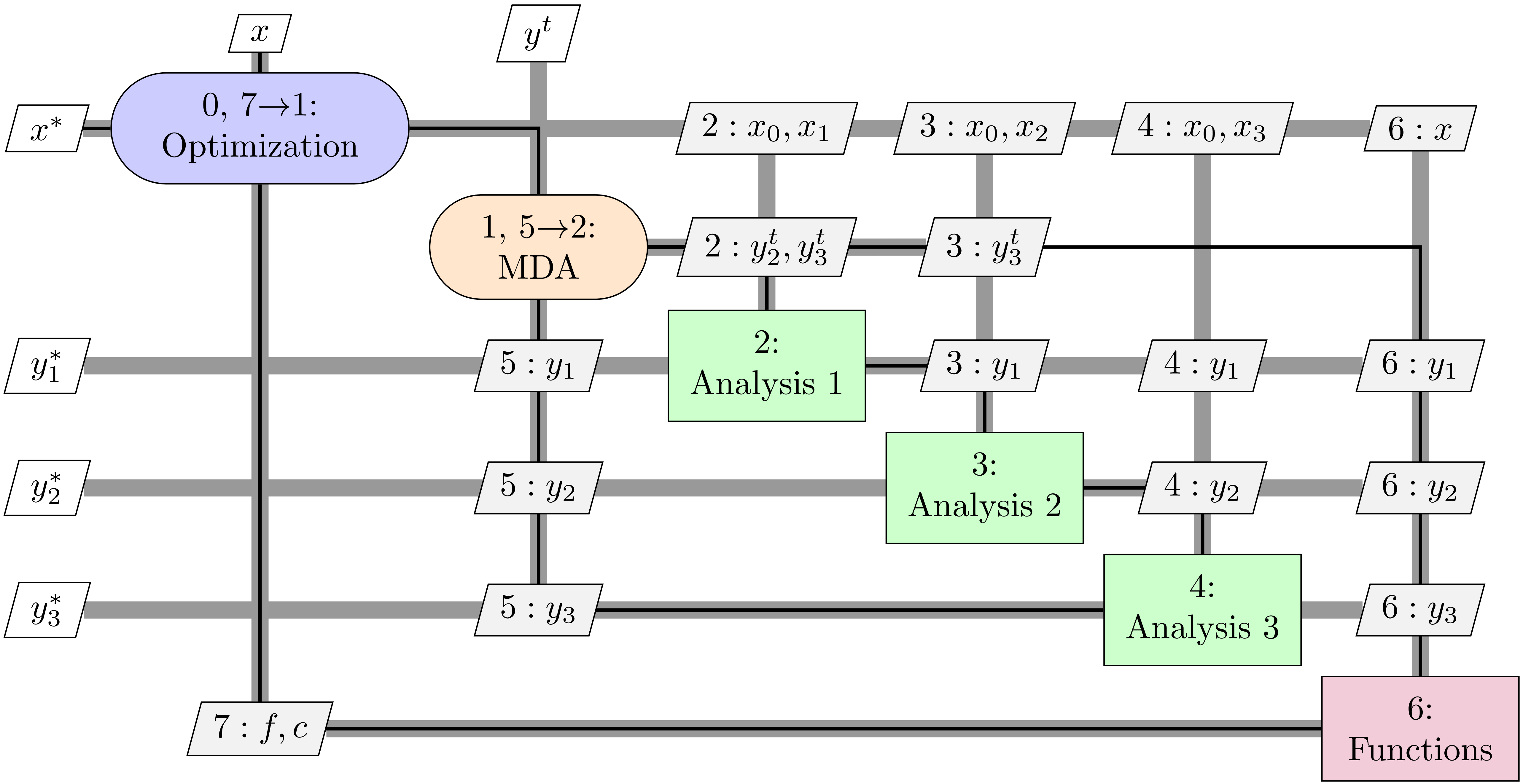
- Follow sequence of numbers and thin black lines
- When number or index is repeated, procedures can be parallelized
- Close the loops

The Multidisciplinary Feasible (MDF) Architecture

Problem Formulation

$$
\begin{aligned}
\text{minimize} \quad & f_0\left(x, y\left(x\right)\right) \\
\text{with respect to} \quad & x \\
\text{subject to} \quad & c_0\left(x, y\left(x\right)\right) \geq 0 \\
& c_i\left(x_0, x_i, y_i\left(x_0, x_i, y_{j \neq i}\right)\right) \geq 0
\end{aligned}
$$

The Multidisciplinary Feasible (MDF) Architecture

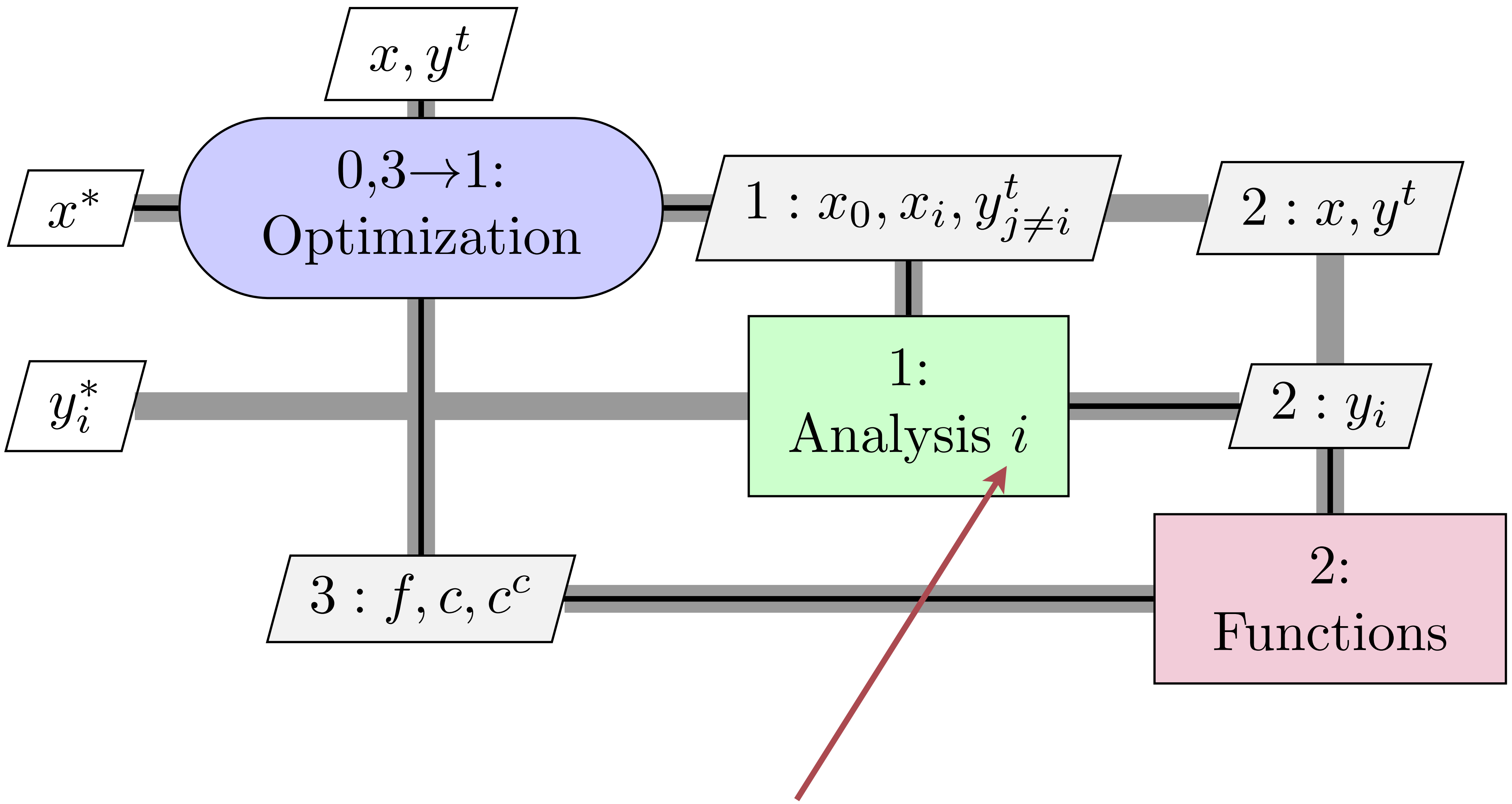


Individual Discipline Feasible (IDF)

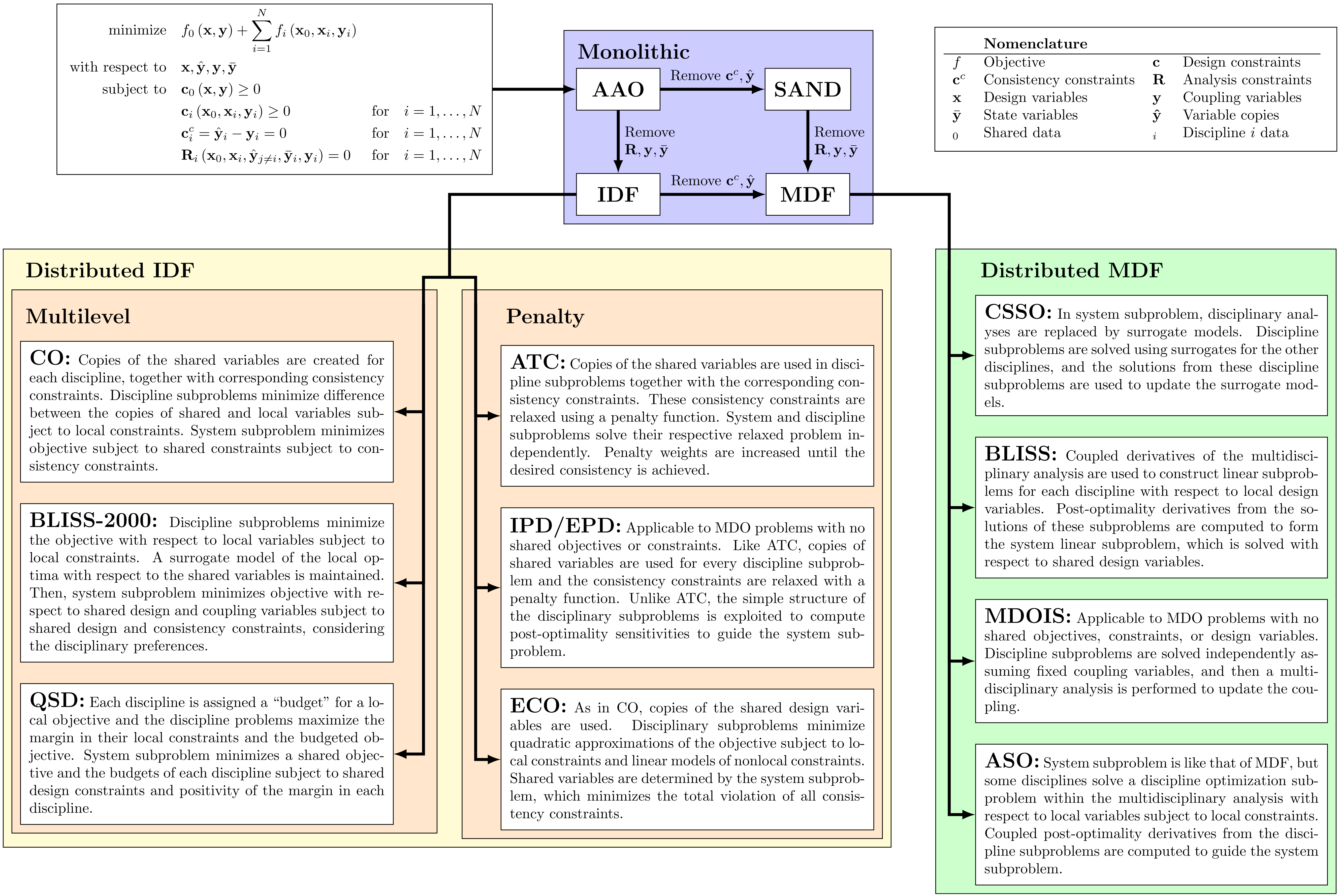
Problem Formulation

$$
\begin{aligned}
\text{minimize} \quad & f_0\left(x, y\left(x, y^t\right)\right) \\
\text{with respect to} \quad & x, y^t \\
\text{subject to} \quad & c_0\left(x, y\left(x, y^t\right)\right) \geq 0 \\
& c_i\left(x_0, x_i, y_i\left(x_0, x_i, y^t_{j \neq i}\right)\right) \geq 0 \\
& y^t_i - y_i\left(x_0, x_i, y^t_{j \neq i}\right) = 0
\end{aligned}
$$

Individual Discipline Feasible (IDF)

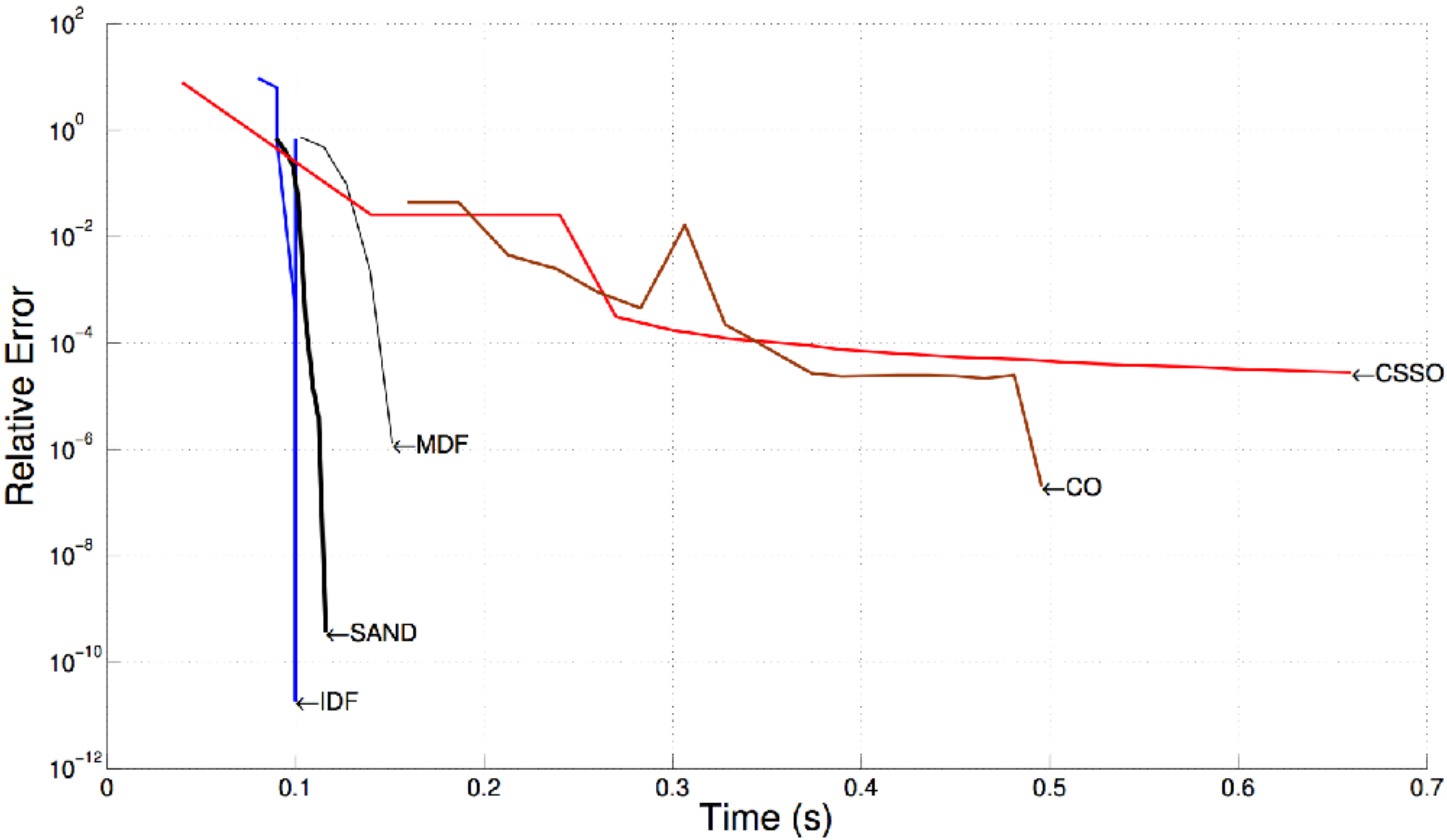


Index indicates multiple analysis
that can be run in parallel



[Martins and Lambe, "MDO: A Survey of Architectures", *AIAAJ*, 2013]

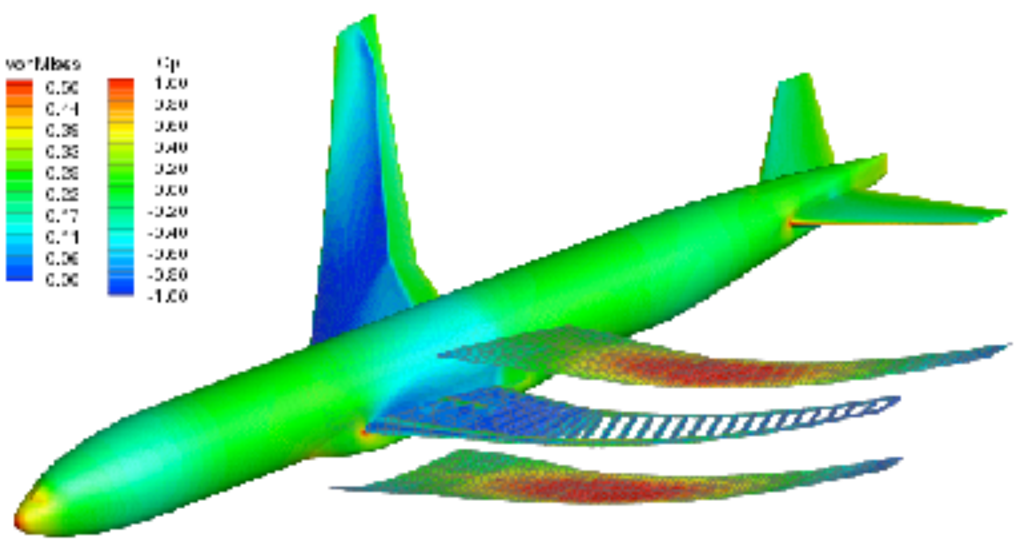
Benchmarking MDO Architectures

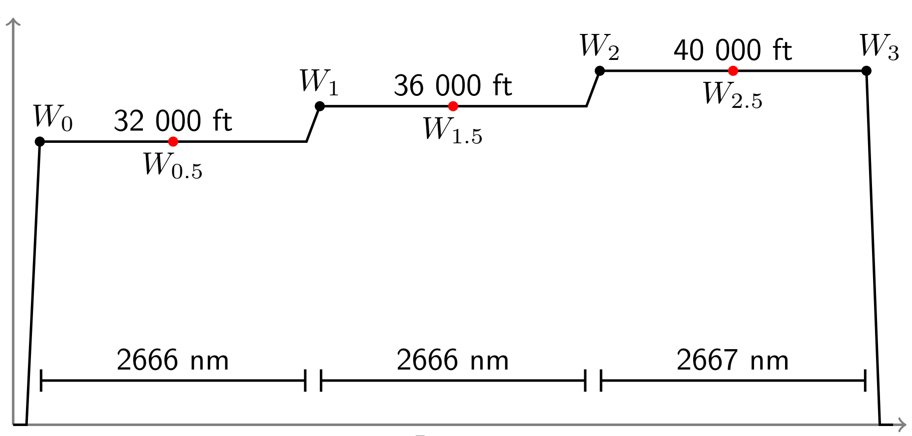


[Tedford and Martins, *Optimization and Engineering*, 2010]

MAUD—Modular Analysis and Unified Derivatives

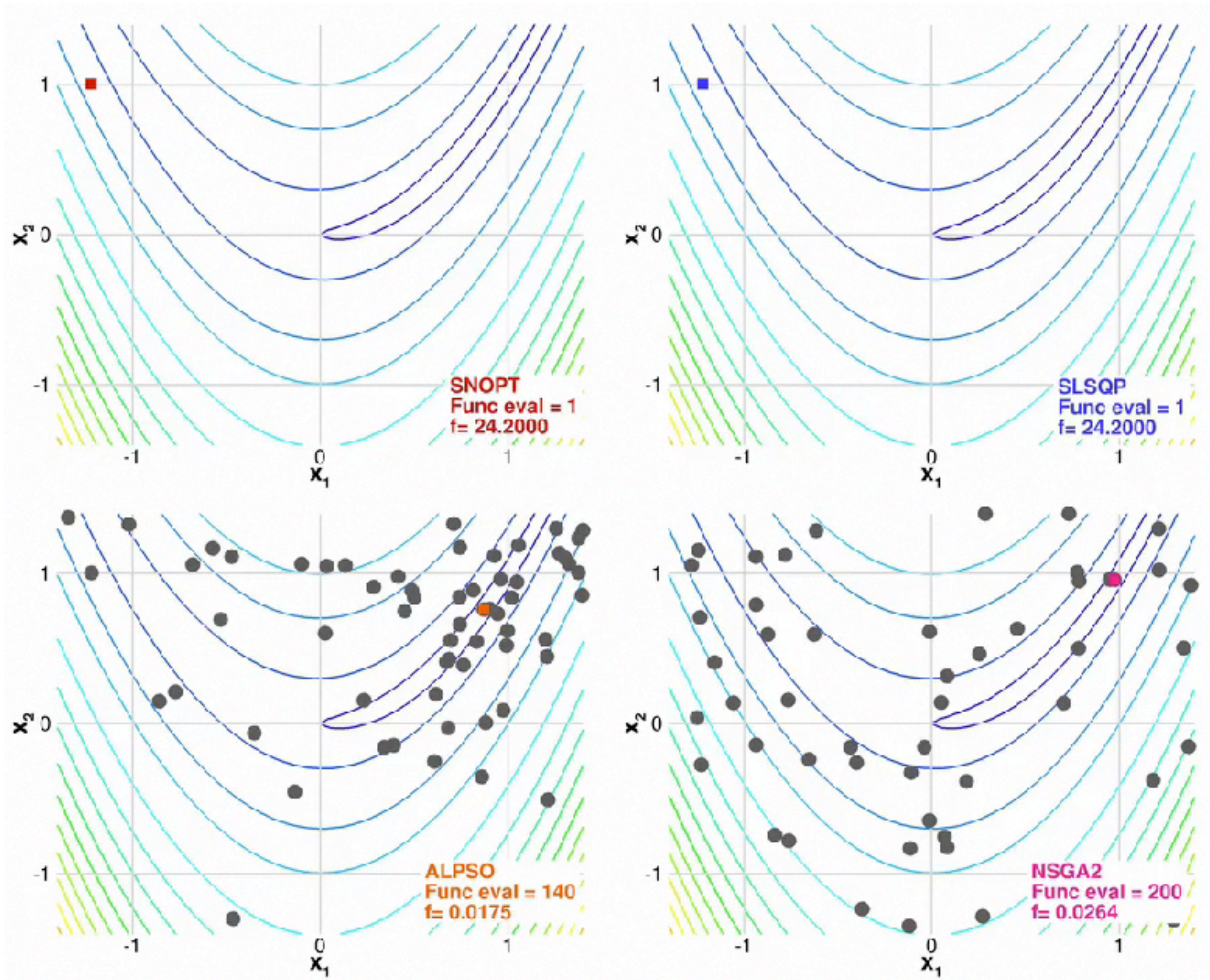
3 major challenges



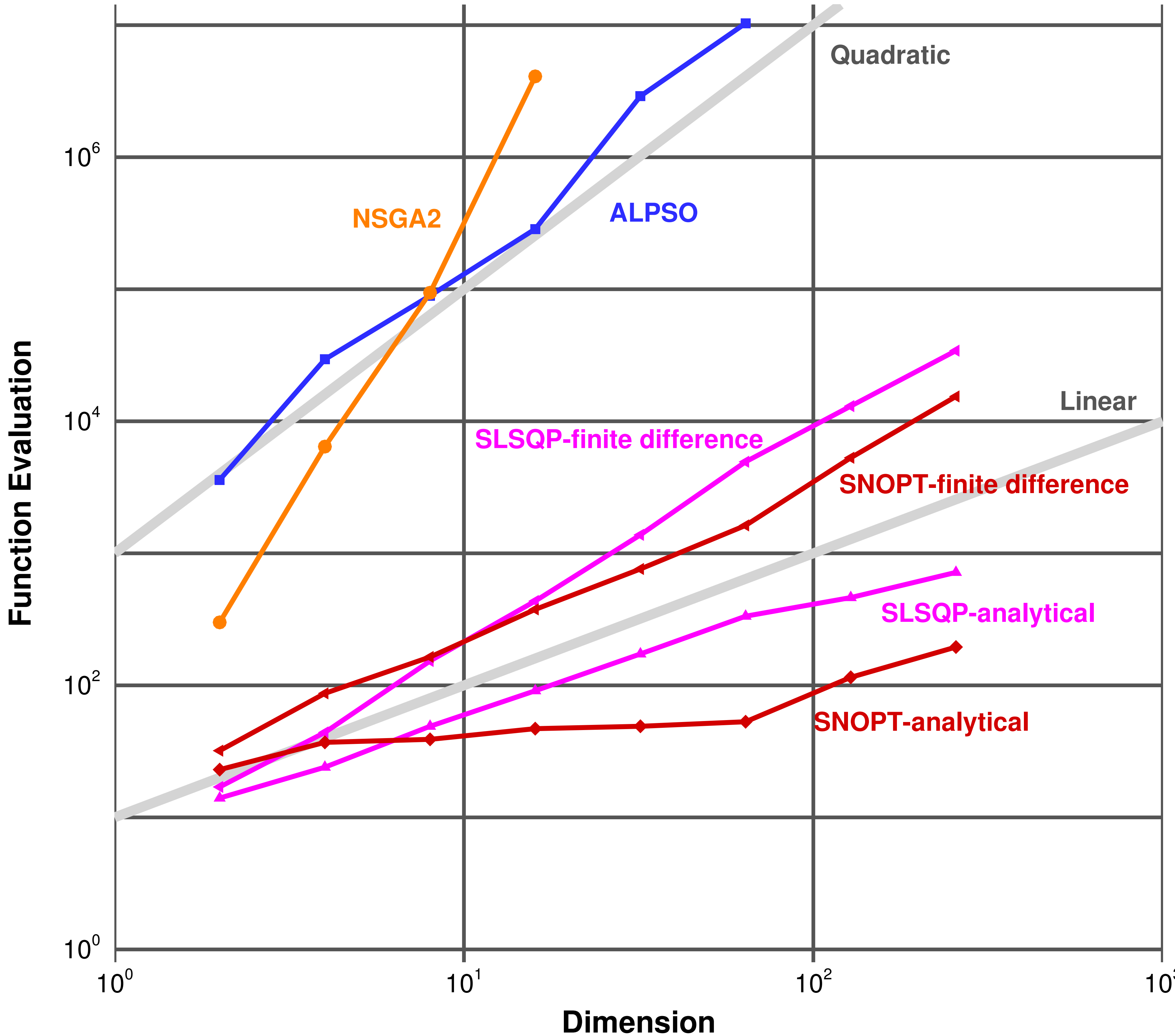


1. Computational costly to evaluate objective and constraints
2. Multiple highly coupled systems
3. Large numbers of design variables, design points and constraints

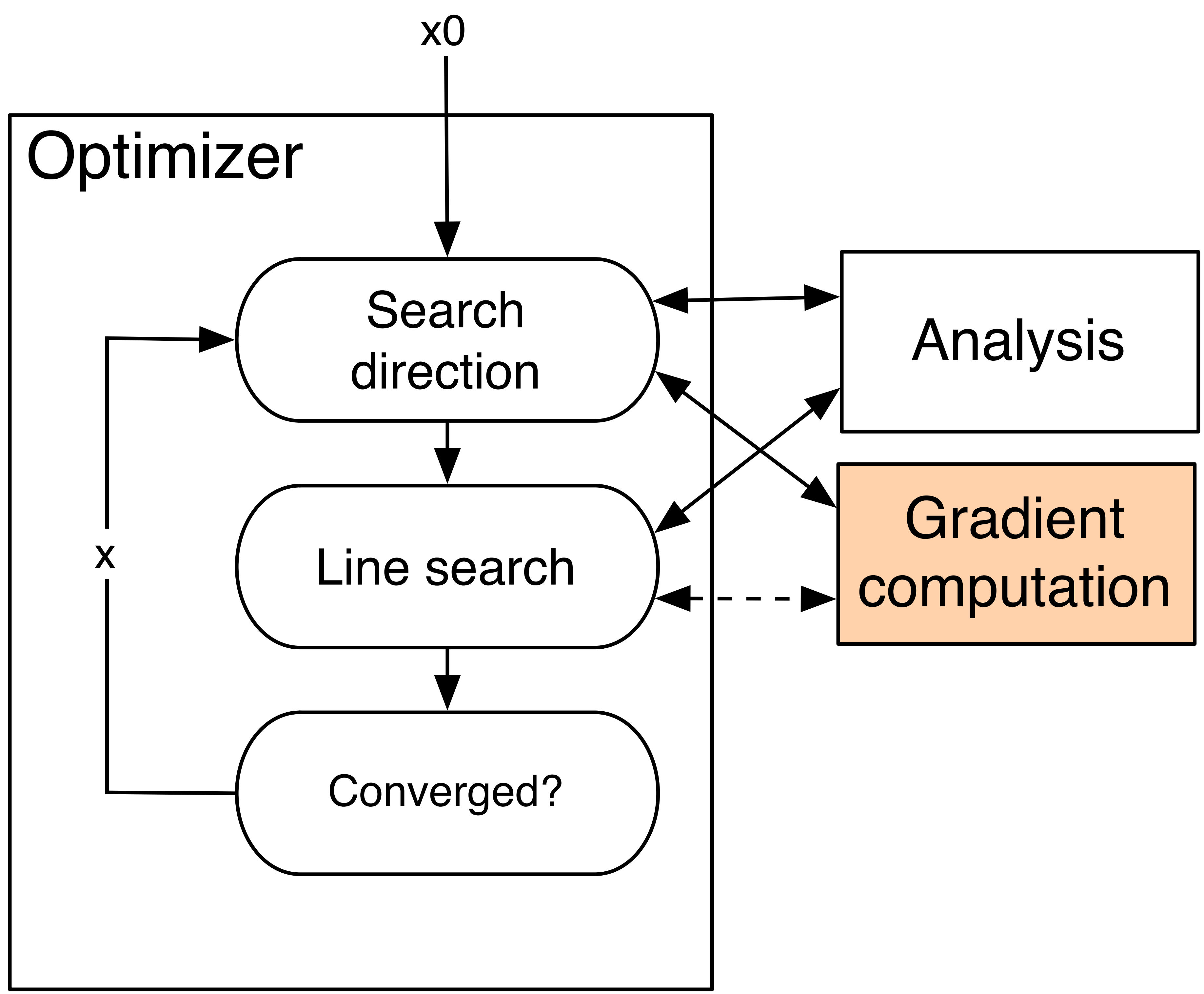
Gradient-based methods take a more direct path to the optimum



Gradient-based optimization is the only hope for large numbers of design variables



Gradient-based optimization works best with accurate and efficient gradient computations



Gradient-based optimization requires gradient of objective and Jacobian of constraints

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x, y(x)) \\ \text{s.t.} \quad & h(x, y(x)) = 0 \\ & g(x, y(x)) \leq 0 \end{aligned}$$

x: design variables
y: state variables, determined implicitly by solving $R(x, y(x)) = 0$

Need $\mathrm{d}f / \mathrm{d}x$ (and also $\mathrm{d}h / \mathrm{d}x$, $\mathrm{d}g / \mathrm{d}x$)

Methods for computing derivatives

Monolithic ***Black boxes input and outputs***	Finite-differences	$\frac{\mathrm{d}f}{\mathrm{d}x_j} = \frac{f(x_j + h) - f(x)}{h} + \mathcal{O}(h)$
	Complex-step	$\frac{\mathrm{d}f}{\mathrm{d}x_j} = \frac{\mathrm{Im}\left[f(x_j + ih)\right]}{h} + \mathcal{O}(h^2)$
Analytic ***Governing eqns state variables***	Direct	$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\partial f}{\partial x} - \underbrace{\frac{\partial f}{\partial y} \overbrace{\left[\frac{\partial R}{\partial y}\right]^{-1} \frac{\partial R}{\partial x}}^{-\,\mathrm{d}y/\,\mathrm{d}x}}_{\psi}$
	Adjoint	
Algorithmic differentiation ***Lines of code code variables***	Forward	$\begin{bmatrix} 1 & 0 & \dots & 0 \\ -\frac{\partial T_2}{\partial t_1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ -\frac{\partial T_n}{\partial t_1} & \dots & -\frac{\partial T_n}{\partial t_{n-1}} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \dots & 0 \\ \frac{\mathrm{d}t_2}{\mathrm{d}t_1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \frac{\mathrm{d}t_n}{\mathrm{d}t_1} & \dots & \frac{\mathrm{d}t_n}{\mathrm{d}t_{n-1}} & 1 \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} 1 & -\frac{\partial T_2}{\partial t_1} & \dots & -\frac{\partial T_n}{\partial t_1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\frac{\partial T_n}{\partial t_{n-1}} \\ 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\mathrm{d}t_2}{\mathrm{d}t_1} & \dots & \frac{\mathrm{d}t_n}{\mathrm{d}t_1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \frac{\mathrm{d}t_n}{\mathrm{d}t_{n-1}} \\ 0 & \dots & 0 & 1 \end{bmatrix}$
	Reverse	

[Martins and Hwang, AIAA Journal, 2013] [Martins et al., ACM TOMS, 2003]

Analytic methods evaluate derivatives by linearizing the governing equations

Need $\mathrm{d}f/\,\mathrm{d}x$ (and also $\mathrm{d}h/\,\mathrm{d}x$, $\mathrm{d}g/\,\mathrm{d}x$), $\quad f(x, y(x))$

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x}$$

Derivative of the governing equations: $R(x, y(x)) = 0$

$$\frac{\mathrm{d}R}{\mathrm{d}x} = \frac{\partial R}{\partial x} + \frac{\partial R}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x} = 0 \quad \Rightarrow \quad \frac{\partial R}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x} = -\frac{\partial R}{\partial x}$$

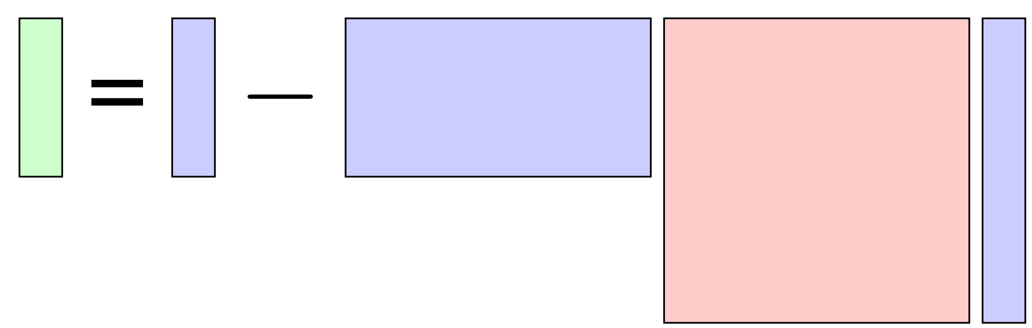
Substitute result into the derivative equation

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\partial f}{\partial x} - \underbrace{\frac{\partial f}{\partial y}\left[\frac{\partial R}{\partial y}\right]^{-1}}_{\psi}\kern-2.5em\overbrace{\phantom{\left[\frac{\partial R}{\partial y}\right]^{-1}}\frac{\partial R}{\partial x}}^{-\,\mathrm{d}y/\,\mathrm{d}x}$$

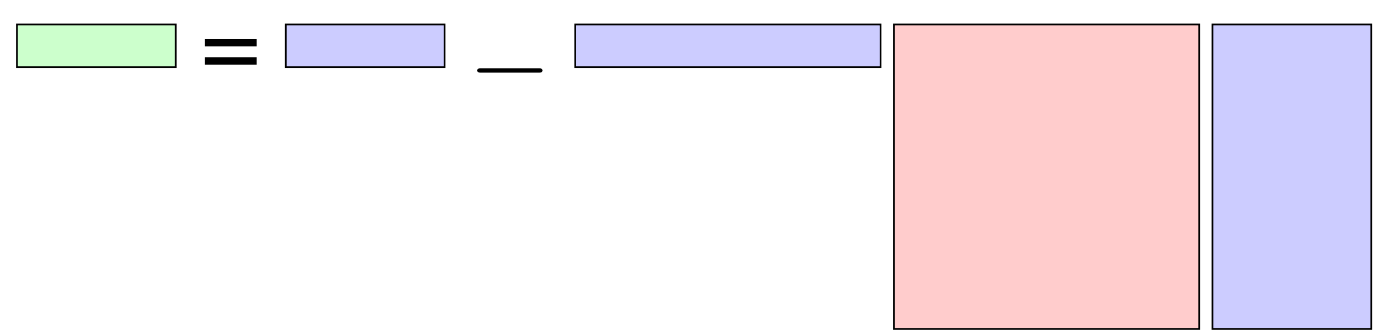
Cost of adjoint evaluation is independent of the number of design variables

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\partial f}{\partial x} - \underbrace{\frac{\partial f}{\partial y}\left[\frac{\partial R}{\partial y}\right]^{-1}}_{\psi}\frac{\partial R}{\partial x}$$

$n_x < n_f$



$n_x > n_f$



Coupled solution of aerodynamics and structures, and the corresponding coupled adjoint

Solve the coupled governing equations

$$R(x, y) = \begin{bmatrix} R_A(x, y_A, y_S) \\ R_S(x, y_A, y_S) \end{bmatrix} = 0$$

form and solve the adjoint equations

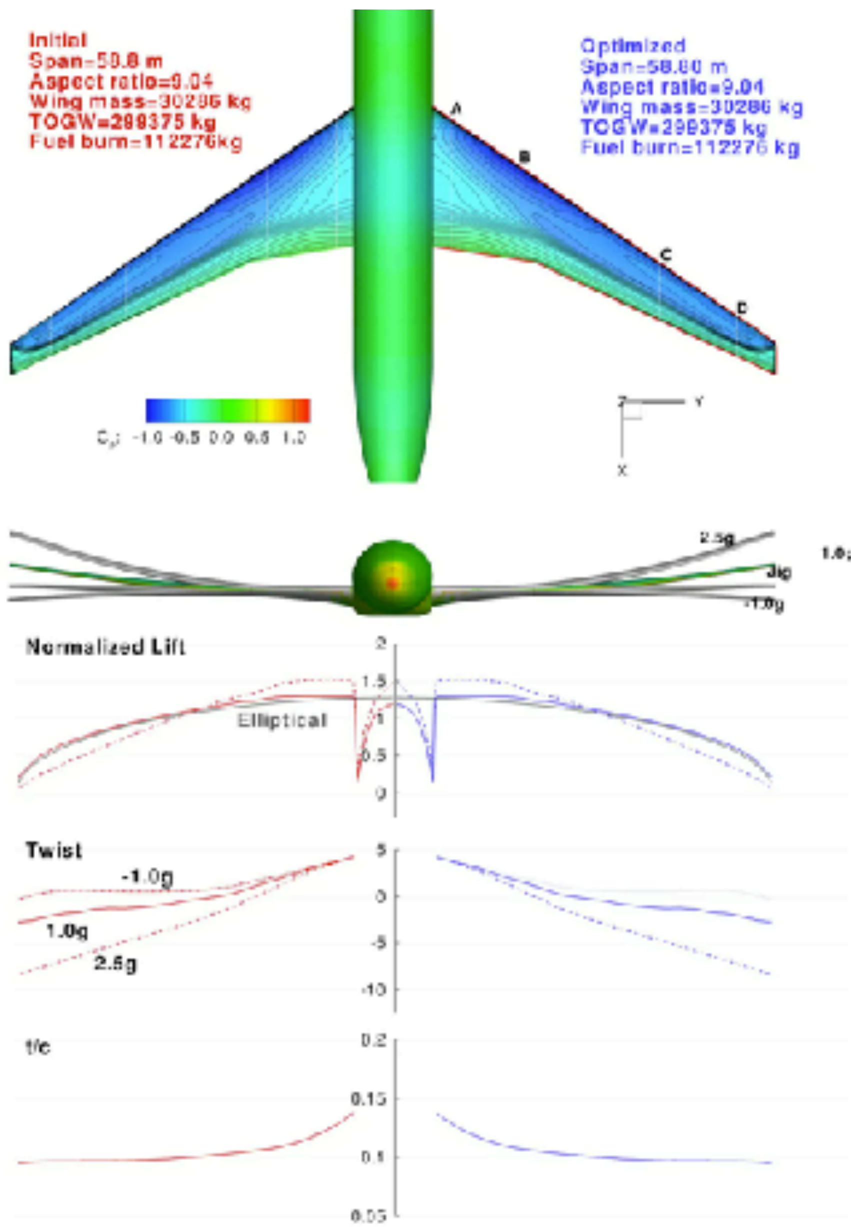
$$\frac{\partial R}{\partial y}^T \psi = \frac{\partial f}{\partial y}^T \quad \text{where} \quad \frac{\partial R}{\partial y}^T = \begin{bmatrix} \frac{\partial R_A}{\partial y_A}^T & \frac{\partial R_S}{\partial y_A}^T \\ \frac{\partial R_A}{\partial y_S}^T & \frac{\partial R_S}{\partial y_S}^T \end{bmatrix}$$

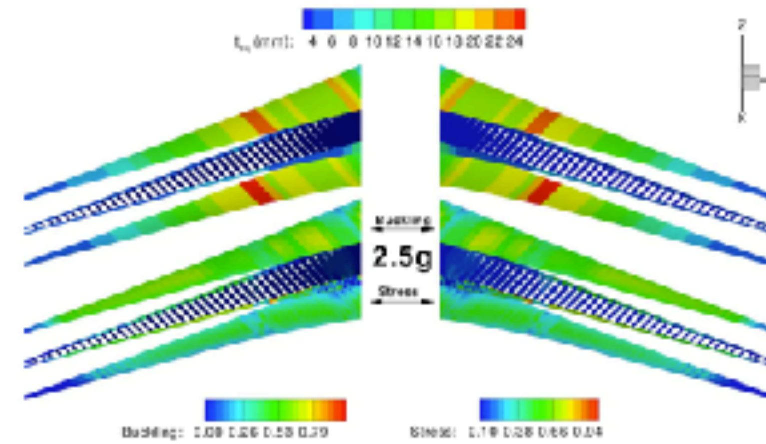
and compute the gradient

$$\frac{\mathrm{d}f}{\mathrm{d}x} = \frac{\partial f}{\partial x} - \psi^T \frac{\partial R}{\partial x}$$

The coupled adjoint approach has shown promising results for a 2-discipline high-fidelity problem

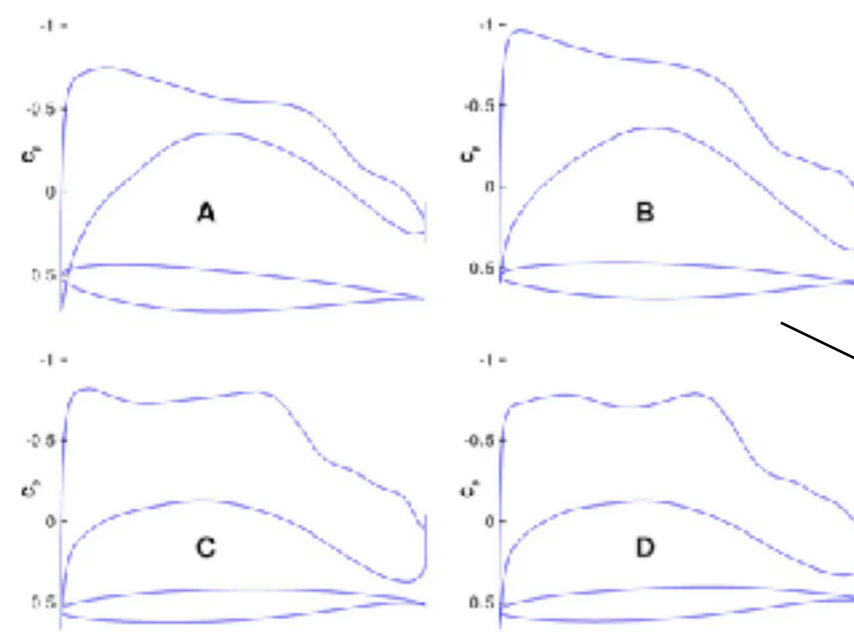
Aerodynamics
(pressure)





Structures
(stresses)

737 structural design variables

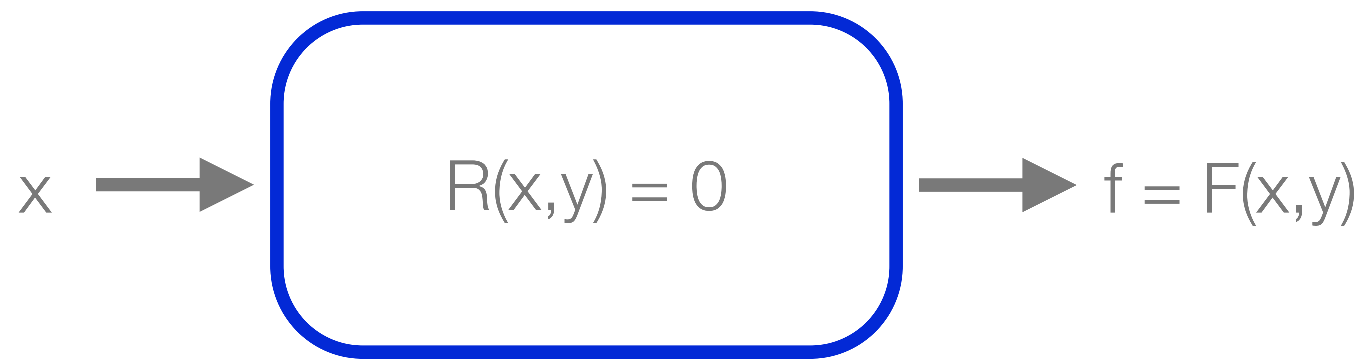


Fuel burn minimization

203 shape design variables

[Kenway, Kennedy and Martins, *AIAA Aviation* 2014]

The adjoint method computes the gradient at the cost of about 1 evaluation

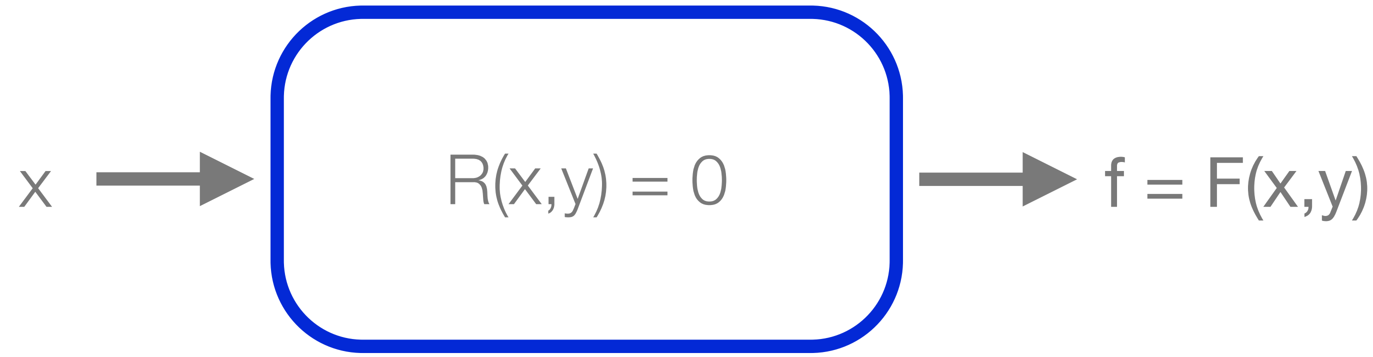


$$\frac{df}{dx} = \frac{\partial F}{\partial x} - \underbrace{\overbrace{\frac{\partial F}{\partial y}}^{b}\,\overbrace{\frac{\partial R}{\partial y}^{-1}}^{A^{-1}}}_{z}\,\frac{\partial R}{\partial x}$$

Cost: solution of a linear system Az=b

Optimization cost
= O(100) iterations x O(1) evaluation = O(100) hrs

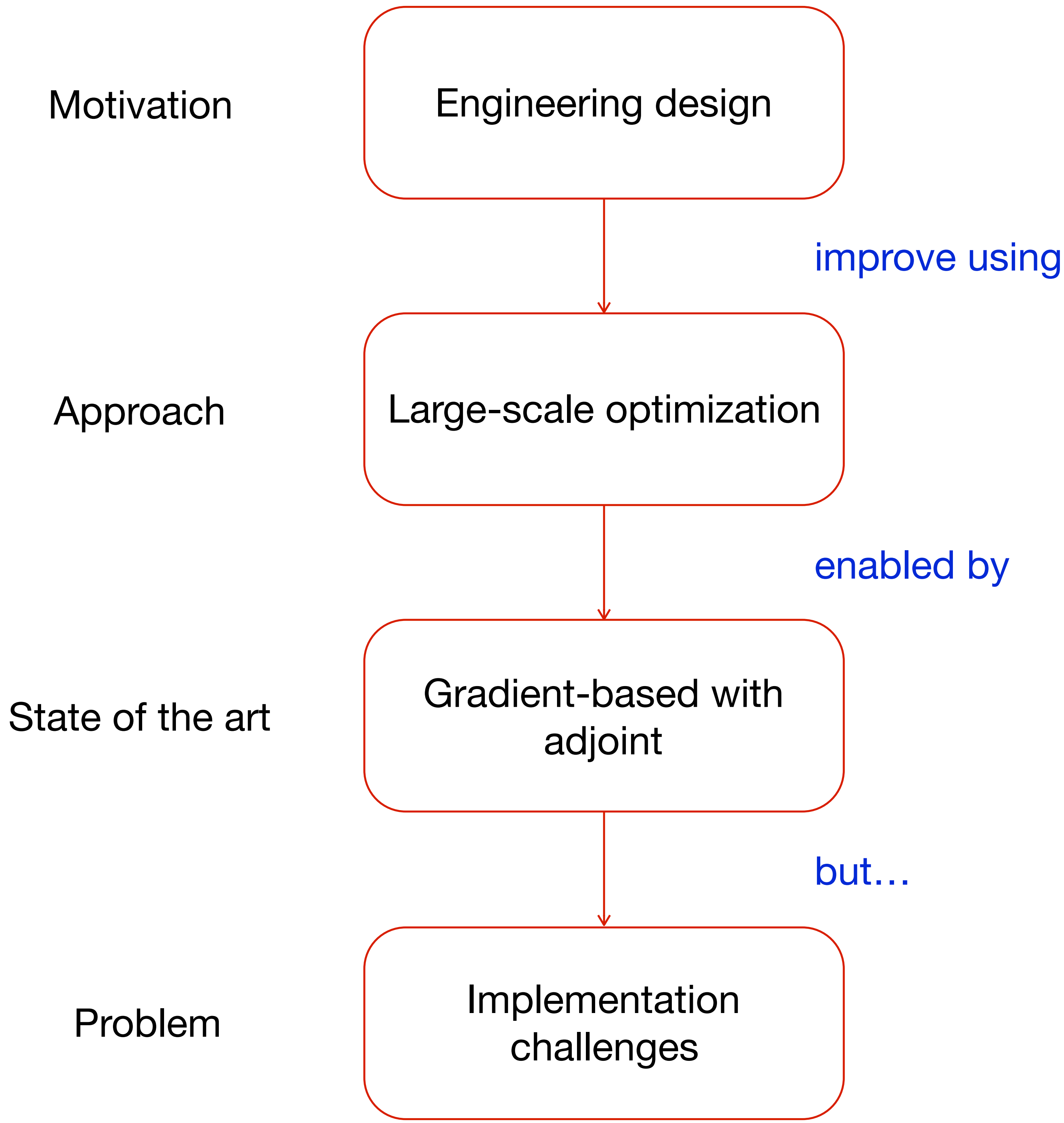
Gradient-based methods enable large-scale optimization but they are difficult to implement



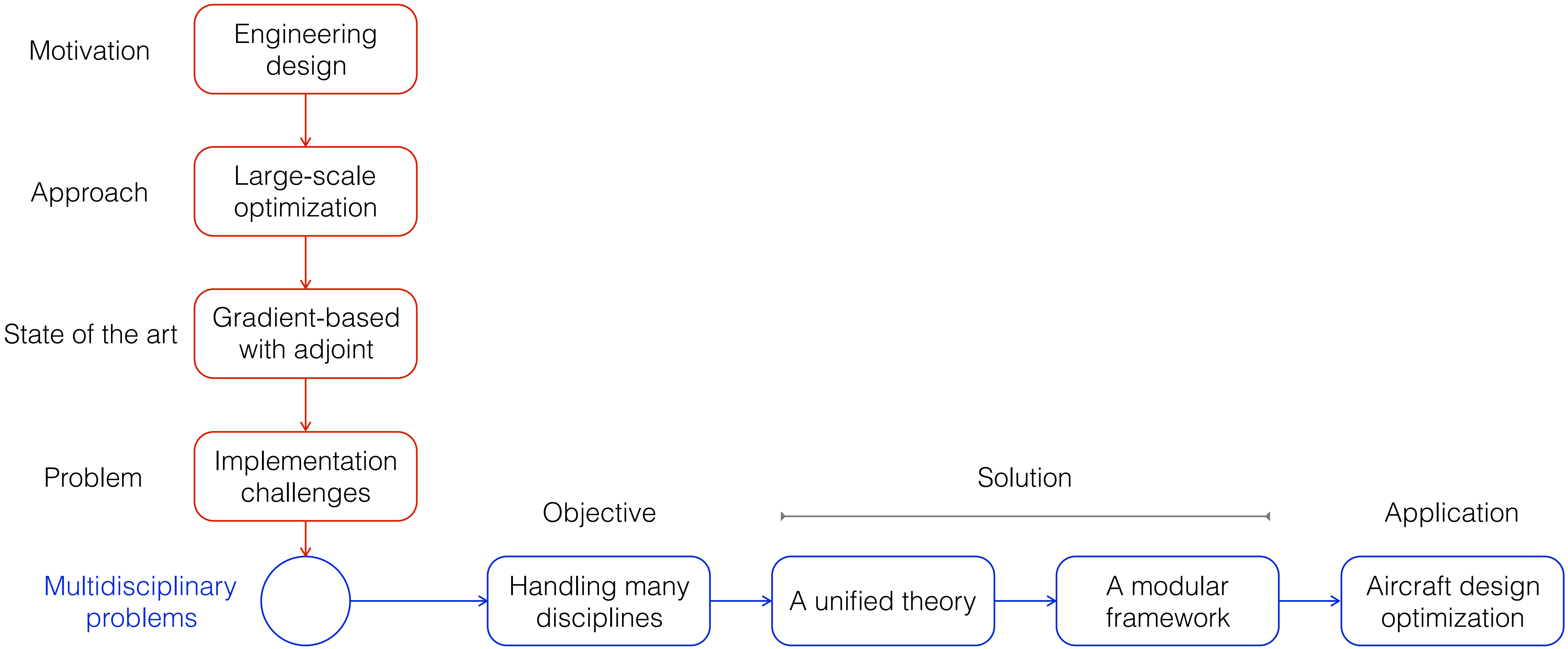
Disadvantages:

- The adjoint method requires modification of the computational model
- The entire model must be differentiable

Our objective: Facilitate the coupling of disciplines and the coupled adjoint method implementation



Outline


Motivation
Engineering design
Approach
Large-scale optimization
State of the art
Gradient-based with adjoint
Problem
Implementation challenges
Multidisciplinary problems
Objective
Handling many disciplines
Solution
A unified theory
A modular framework
Application
Aircraft design optimization

For optimization to be a useful design tool, setting up should be as streamlined as possible

Run optimization

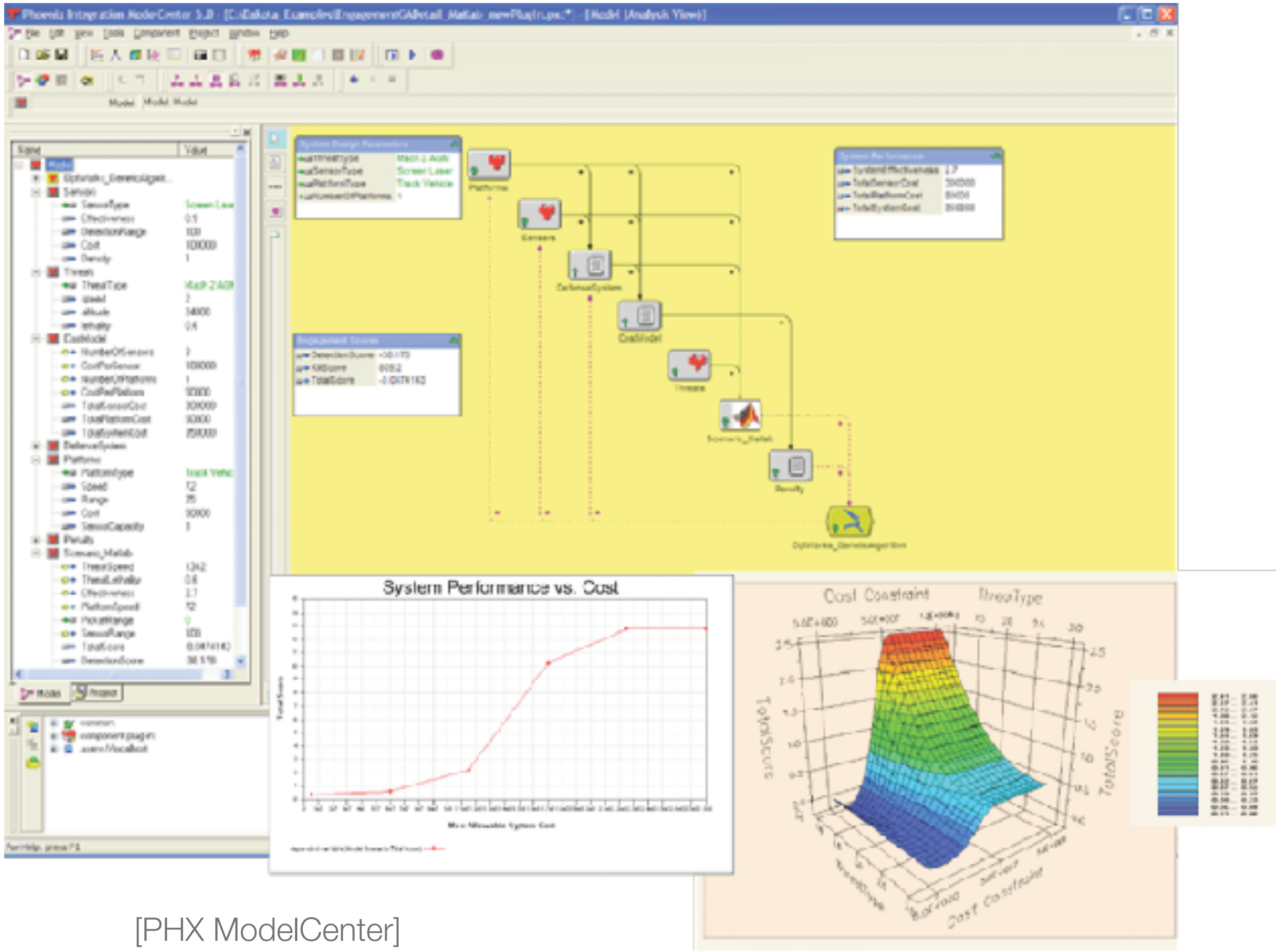
Interpret results

Set up a new optimization problem

- Change the objective function
- Add/remove design variables from the inputs
- Add/remove constraints from the outputs
- Add/remove disciplines and models

We need a software framework
to enable a modular approach

However, existing software frameworks are not designed to support the coupled adjoint method



[PHX ModelCenter]

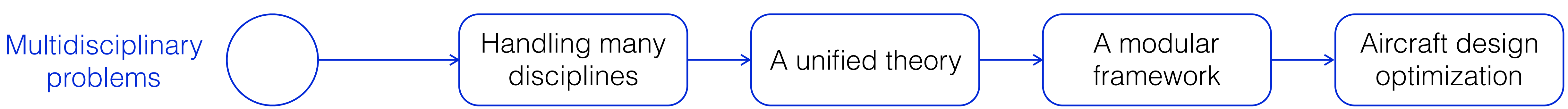
Our approach is to formulate computational models as a system of equations

Any computational models can be decomposed into

Input variables	$x = (x_1, \ldots, x_m)^T$
State variables	$y = (y_1, \ldots, y_p)^T$
Output variables	$f = (f_1, \ldots, f_q)^T$

The objective is to express it as a system of algebraic equations,

$$R(u) = 0 \quad \text{where} \quad u = \begin{pmatrix} x \\ y \\ f \end{pmatrix}$$



Example: aerodynamic and structural analysis coupled together

Question: how do we compute df/db in the following problem?

Variable	Type		
Wing span	input	$b:$	$b = b^*$
Pressures	state	$p:$	$A(b, p, d) = 0$
Displacements	state	$d:$	$d = S(b, p)$
Lift	output	$f:$	$f = F(b, p)$

Formulated as a system of algebraic equations:

$$u = \begin{bmatrix} b \\ p \\ d \\ f \end{bmatrix} \qquad R(u) = \begin{bmatrix} R_b(b, p, d, f) \\ R_p(b, p, d, f) \\ R_d(b, p, d, f) \\ R_f(b, p, d, f) \end{bmatrix} = \begin{bmatrix} b - b^* \\ A(b, p, d) \\ d - S(b, p) \\ f - F(b, p) \end{bmatrix}$$

We derived the unifying chain rule, which generalizes all differentiation methods

$$\boxed{\frac{\partial R}{\partial u}\frac{du}{dr} = \mathcal{I} = \left[\frac{\partial R}{\partial u}\right]^T \left[\frac{du}{dr}\right]^T}$$

Other methods
Algorithmic diff.
Coupled adjoint

$$\frac{df}{dx} = \frac{\partial F}{\partial x}$$

finite differences

$$\frac{df}{dx} = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial y}\frac{dy}{dx}$$

chain rule

$$\frac{df}{dx} = \frac{\partial F}{\partial x} - \frac{\partial F}{\partial y}\frac{\partial R}{\partial y}^{-1}\frac{\partial R}{\partial x}$$

adjoint method

$$\begin{bmatrix} -\frac{\partial R_1}{\partial y_1} & -\frac{\partial Y_2}{\partial y_1} \\ -\frac{\partial R_1}{\partial y_2} & 1 \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d} f_1}{\mathrm{d} r_1} & \frac{\mathrm{d} f_2}{\mathrm{d} r_1} \\ \frac{\mathrm{d} f_1}{\mathrm{d} y_2} & \frac{\mathrm{d} f_2}{\mathrm{d} y_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial F_1}{\partial y_1} & \frac{\partial F_2}{\partial y_1} \\ \frac{\partial F_1}{\partial y_2} & \frac{\partial F_2}{\partial y_2} \end{bmatrix}$$

chain rule/adjoint hybrid

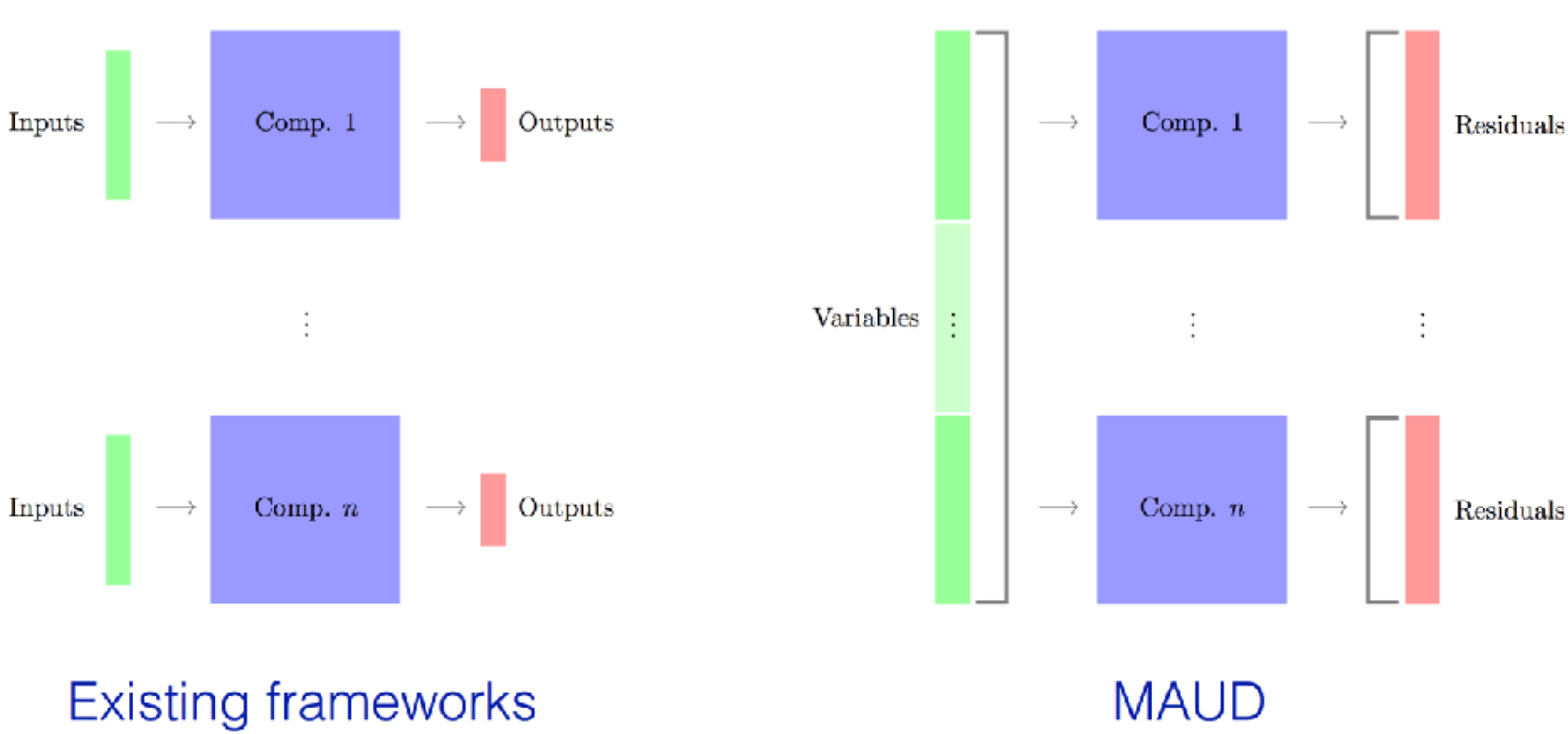
[Hwang and Martins, *AIAAJ*, 2013]

This equation inspired the development of the Modular Analysis and Unified Derivatives (MAUD) framework

The MAUD framework was developed to facilitate two tasks:

1. Solution of a computational model with multiple components
2. Efficient computation of the derivatives of the coupled system of computational models

The MAUD architecture works by viewing the model as one large nonlinear system

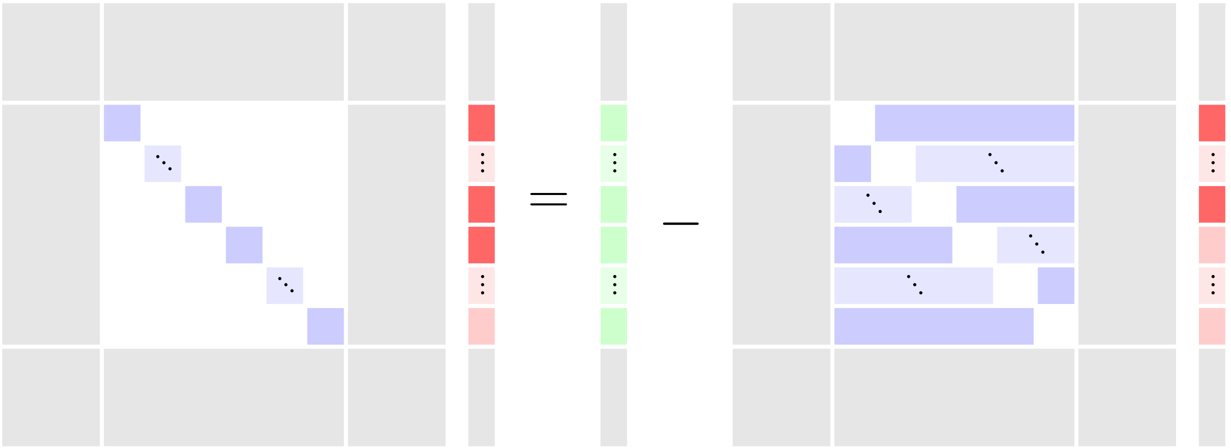


The MAUD framework core solves 4 types of sub-problems

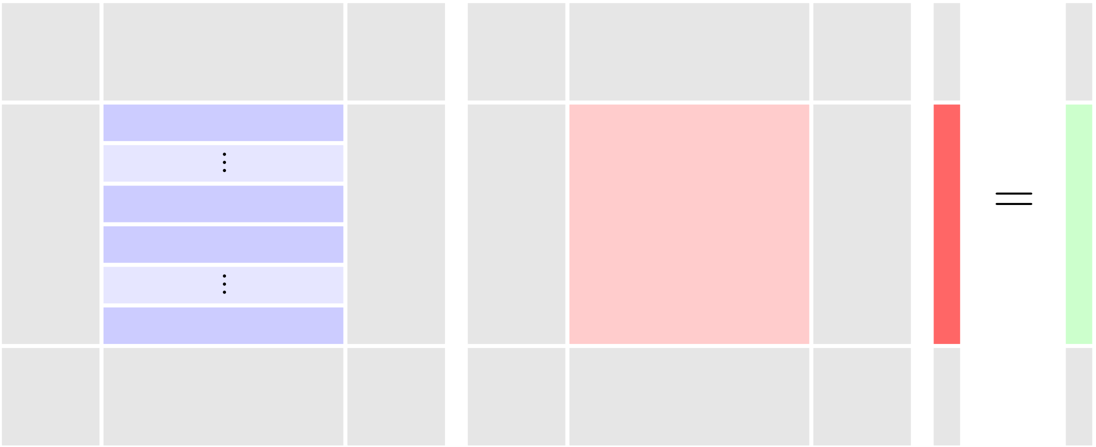
1. Nonlinear system $$R(u) = 0$$
2. Newton system $$\frac{\partial R}{\partial u} \Delta u = -r$$
3. Derivatives in forward mode $$\frac{\partial R}{\partial u} \frac{\mathrm{d}u}{\mathrm{d}r} = \mathcal{I}$$
4. Derivatives in reverse mode $$\frac{\partial R}{\partial u}^T \frac{\mathrm{d}u}{\mathrm{d}r}^T = \mathcal{I}$$

MAUD uses linear and nonlinear solvers that take advantage of the problem structure

Block Gauss-Seidel



Preconditioned
Krylov subspace methods



[Hwang and Martins, 2015 (to be submitted)]

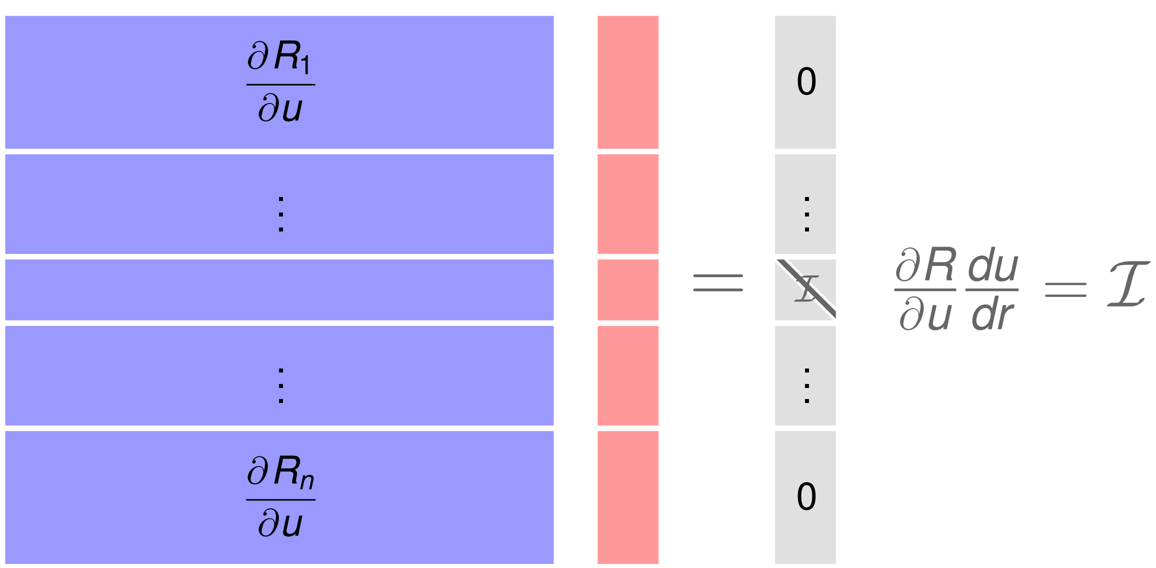
MAUD requires user to define a few basic functions for each component

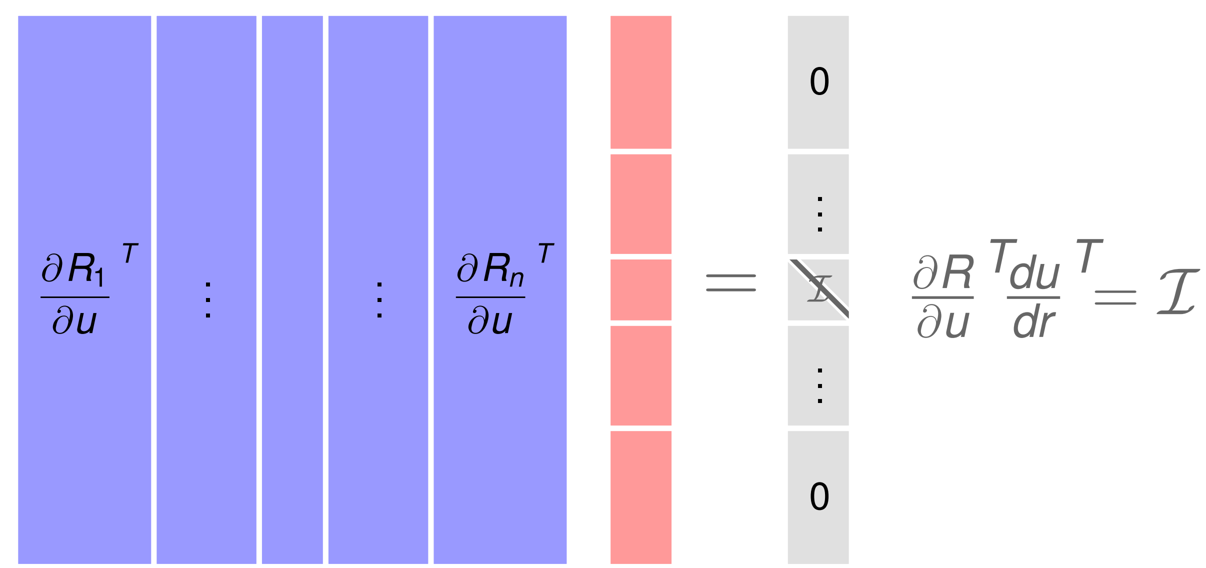
		System classes		
		System	*CompoundSystem*	*ElementarySystem*
Virtual methods	*apply_nonlinear*	—	Recursive	User-implemented
	apply_linear		Recursive	User-implemented or FD*
	solve_nonlinear	Newton with line search	Nonlinear block Gauss–Seidel/Jacobi	Optional
	solve_linear	Krylov-type with preconditioning	Linear block Gauss–Seidel/Jacobi	Optional

*FD: finite-difference approximation of the Jacobian.

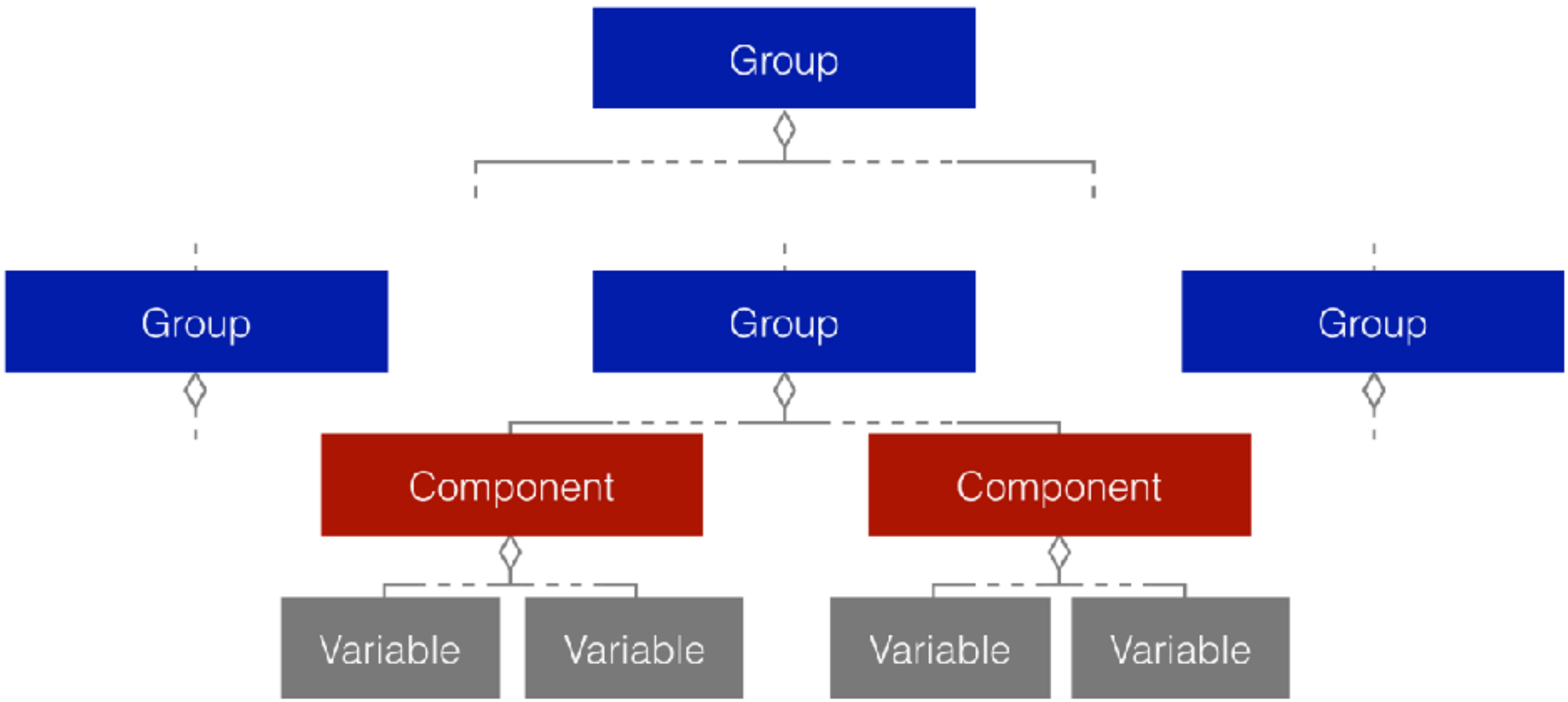
The user provides the computation of partial derivatives, and MAUD computes the coupled derivatives automatically

$$\boxed{\frac{\partial R}{\partial u}\frac{du}{dr} = \mathcal{I} = \left[\frac{\partial R}{\partial u}\right]^T \left[\frac{du}{dr}\right]^T}$$

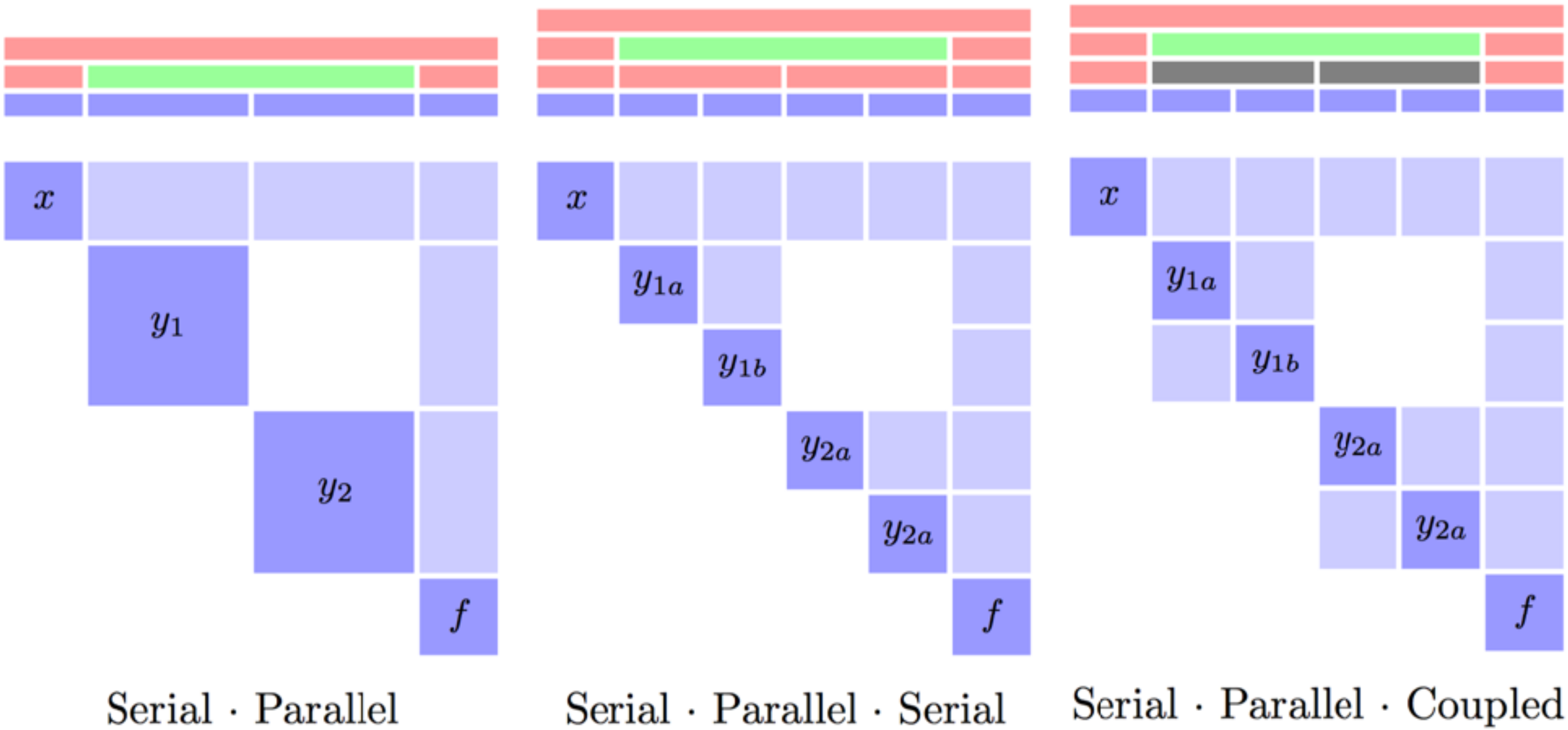




MAUD uses a multiple component hierarchical representation to handle multi-physics systems



This hierarchical component representation enables better parallelization and sequencing



MAUD now provides the core algorithms in the OpenMDAO framework

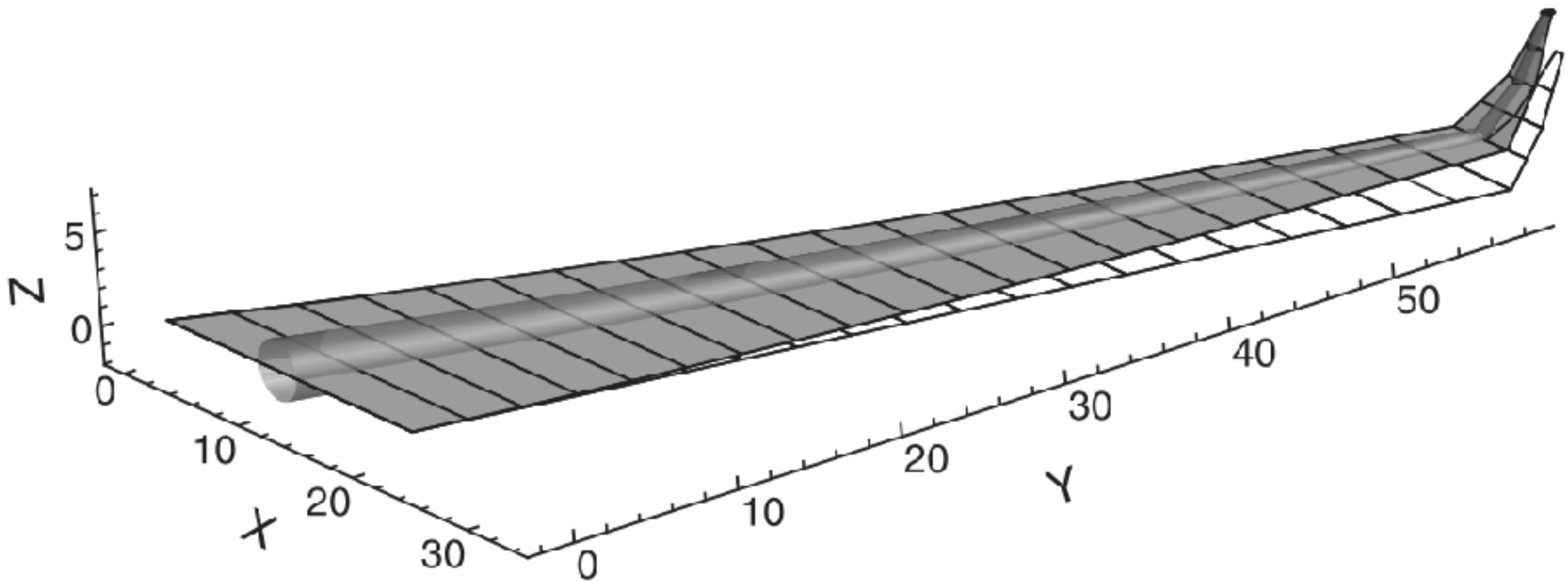
- Originally developed by team at NASA Glenn
- Python-based, open source framework for coupling multiple models and optimization
- Facilitates collaboration between industry, academia, and government
- Provides a common platform for the development of new multidisciplinary analysis and design methods

Multidisciplinary Design Optimization of Aircraft Configurations

Part 1: A modular coupled adjoint approach

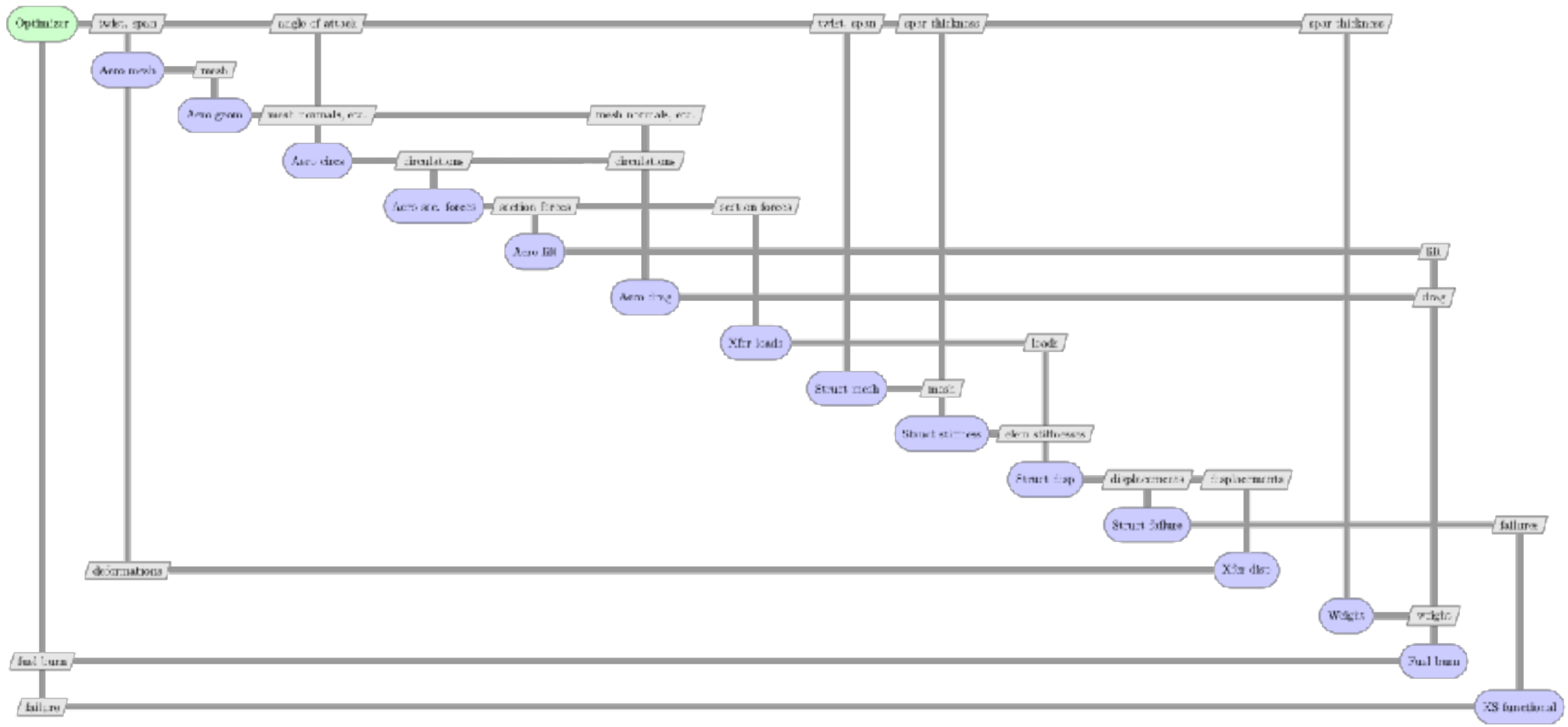
Applications of MAUD

A simple application of MAUD:
Low-fidelity wing aerostructural analysis and design



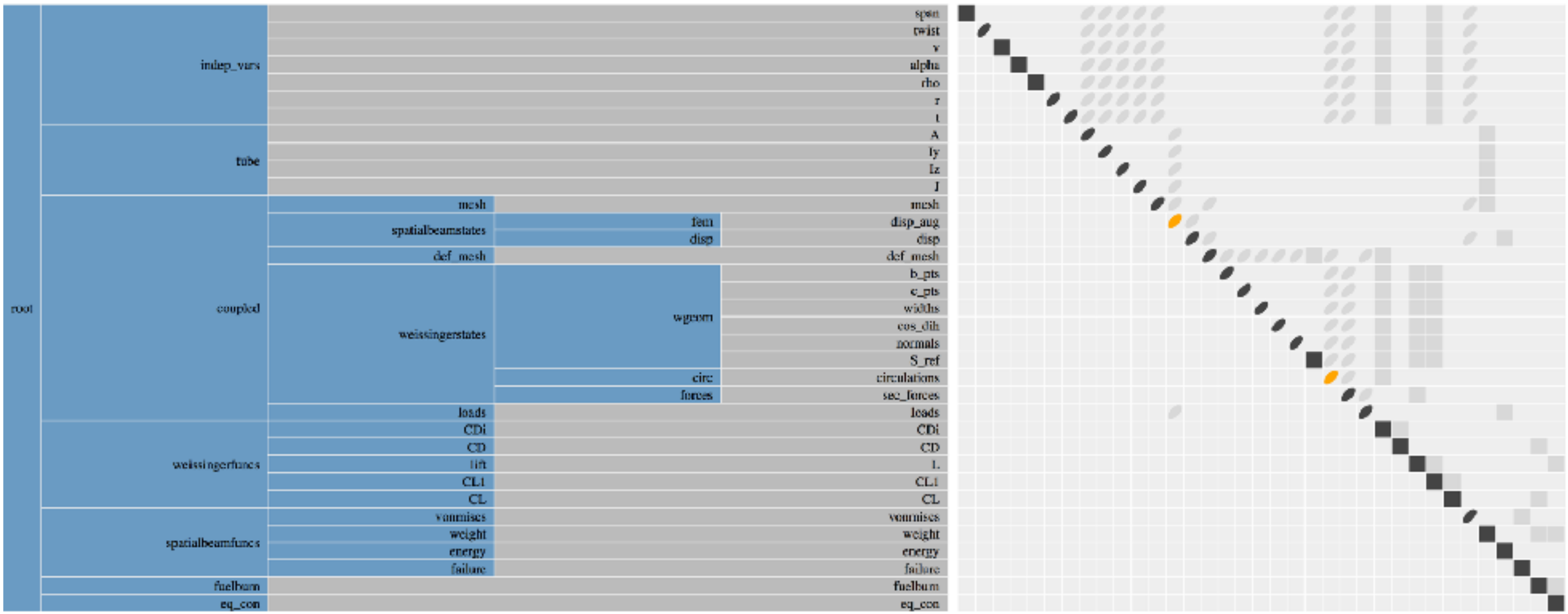
- Lifting line aerodynamic analysis
- Spatial beam element structural analysis
- Coupled solution yields wing flying shape

Breaking down the problem into multiple components is beneficial in MAUD



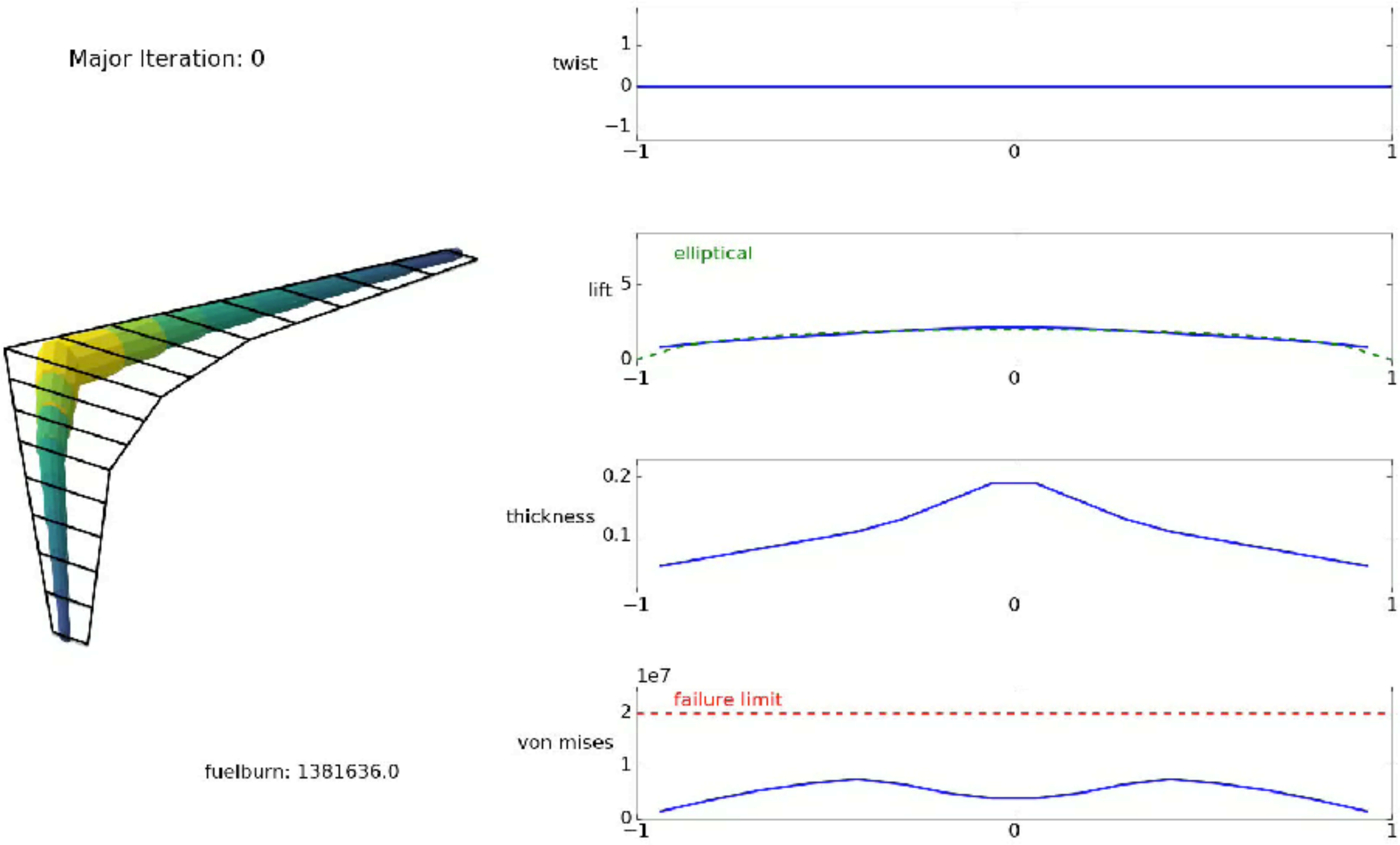
- Easier to debug
- Component derivatives are easier
- More flexibility in algorithms and parallelization

Problem structure can be visualized using an interactive design structure matrix diagram

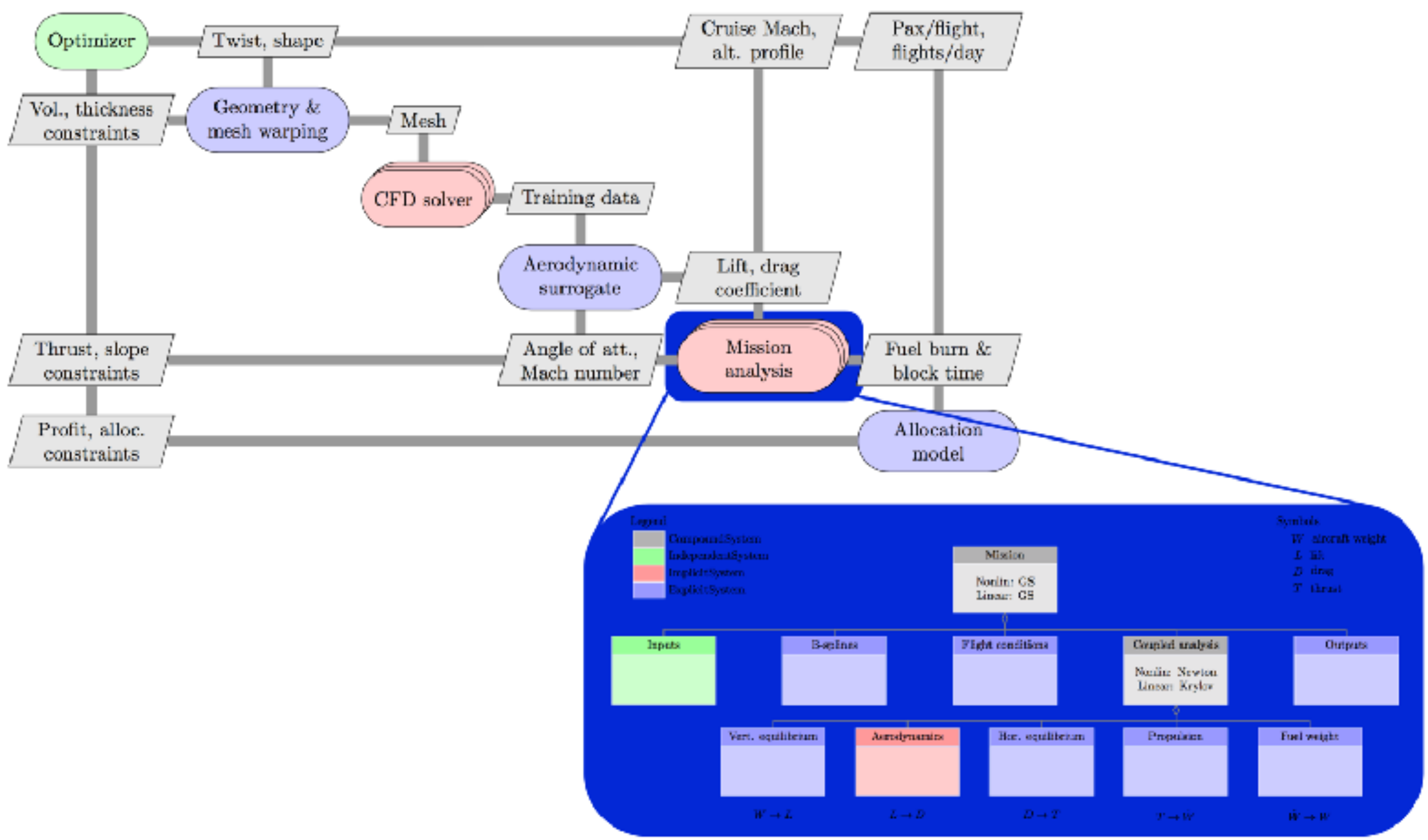


- Hierarchy of the problem is clearly visible
- The structure is shown as a direct structure matrix
- Couplings are shown when hovering on components
- This facilitates the exploitation of problem structure

The results can be visualized in real time



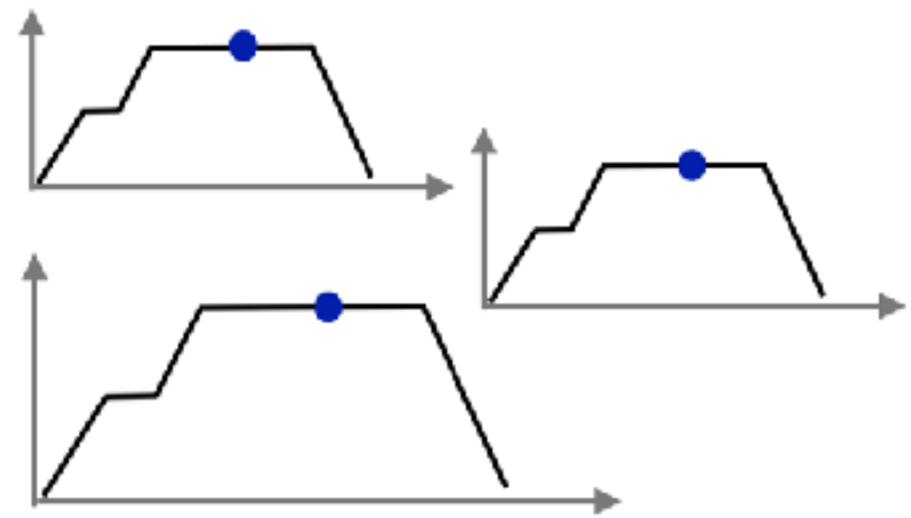
A more complex application of MAUD: Aircraft design with allocation, and mission optimization



[Hwang and Martins, *AIAA* 2016-1662]

The current aircraft design optimization approach is multipoint fuel burn minimization

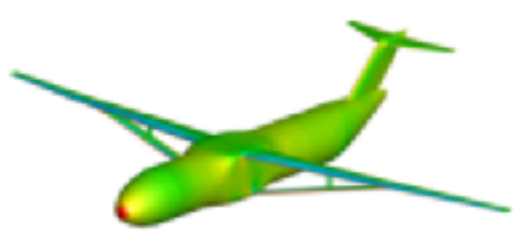
Select a single design mission range, R



Select points $(M_1, C_{L,1}) \ldots (M_n, C_{L,n})$ and their weights $w_1 \ldots w_n$

minimize $w_1\, fb(R, M_1, C_{L,1}) + \ldots + w_n\, fb(R, M_n, C_{L,n})$

with respect to the aircraft design



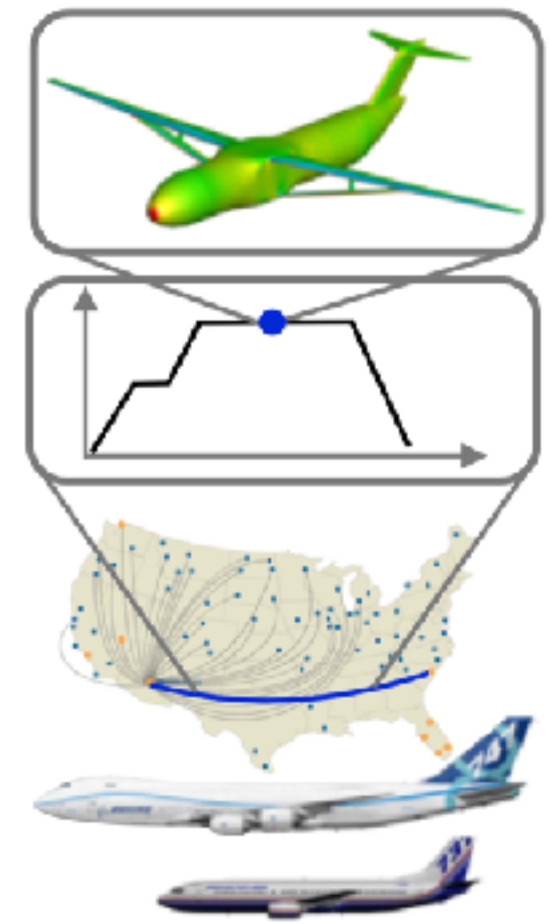
In reality, the allocation, mission profiles, and design should be optimized simultaneously

maximize airline profit

with respect to design: shape, structure, engine

mission: Mach number, altitude profile

airline allocation: flights per day



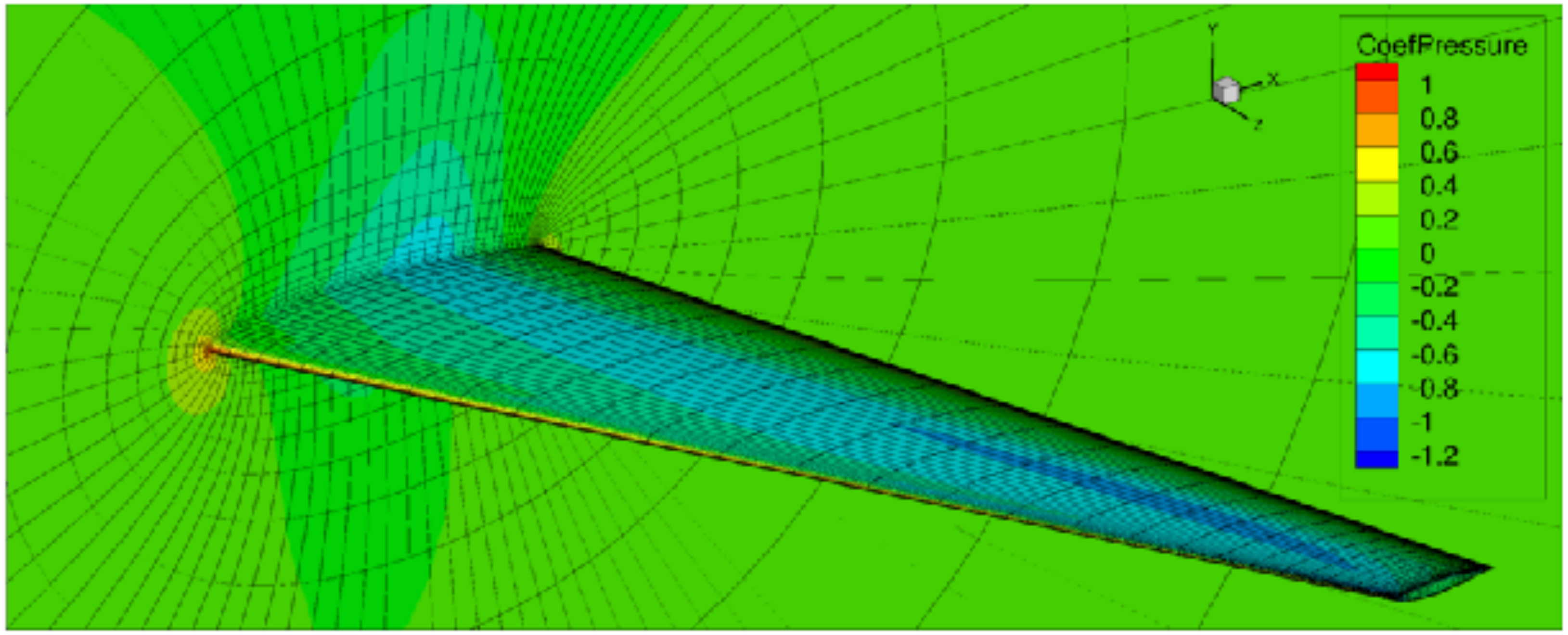
Why?

- Aircraft often flown below their range
- Climb/descent significant on short flights
- Model morphing and continuous descent
- Determine optimal cruise Mach numbers

We perform the allocation-mission-design optimization of an aircraft in a 128-route network

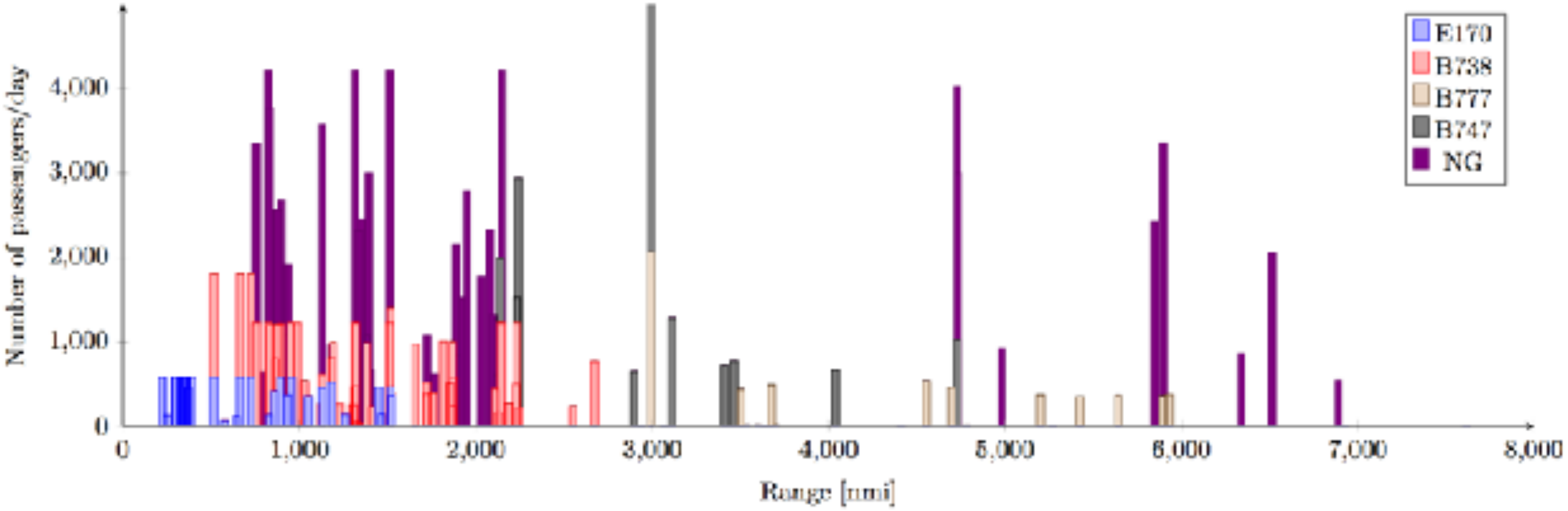
Linearly tapered wing with Euler analysis

twist and shape design variables

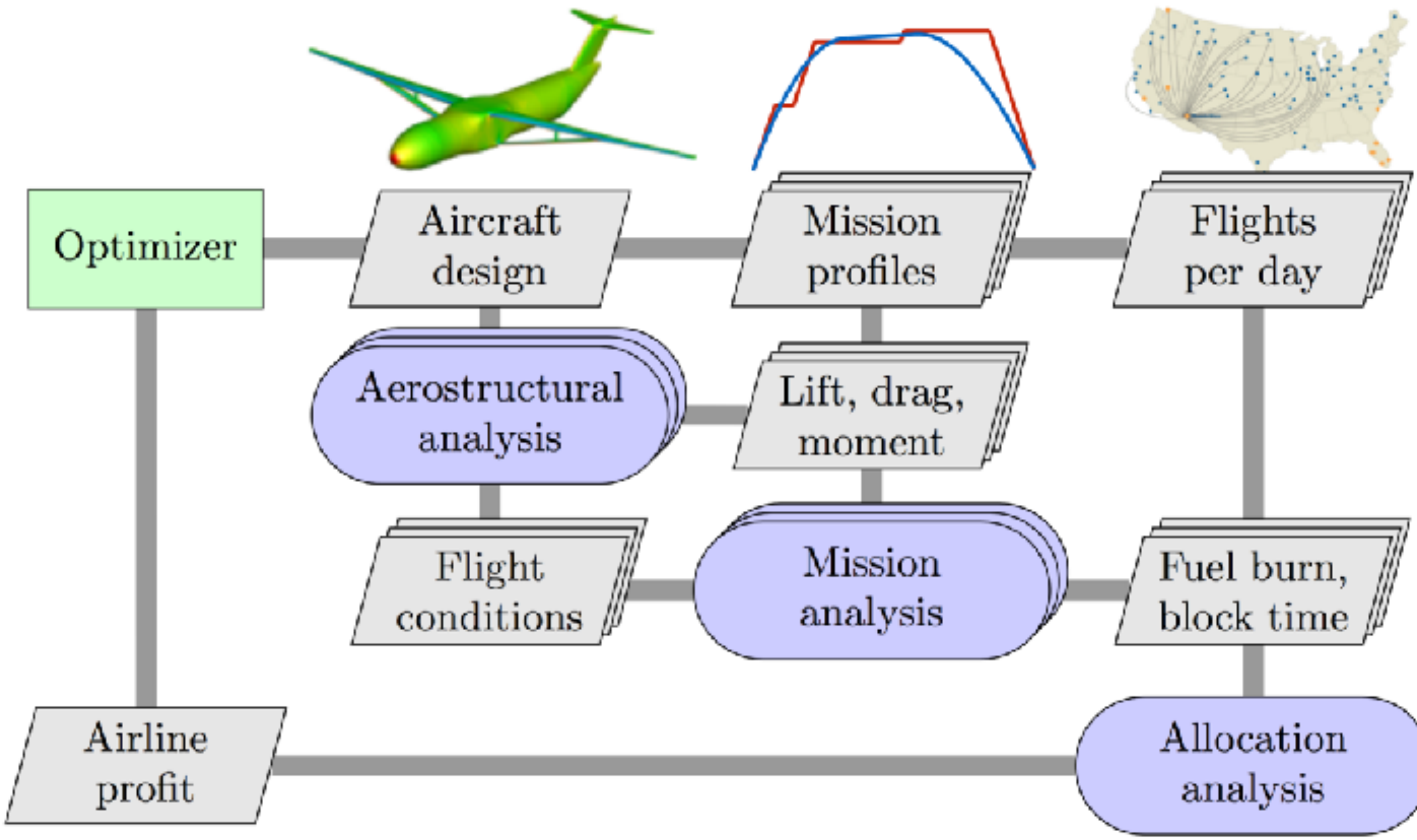


128 route network

altitude profiles and optimal cruise M design variables



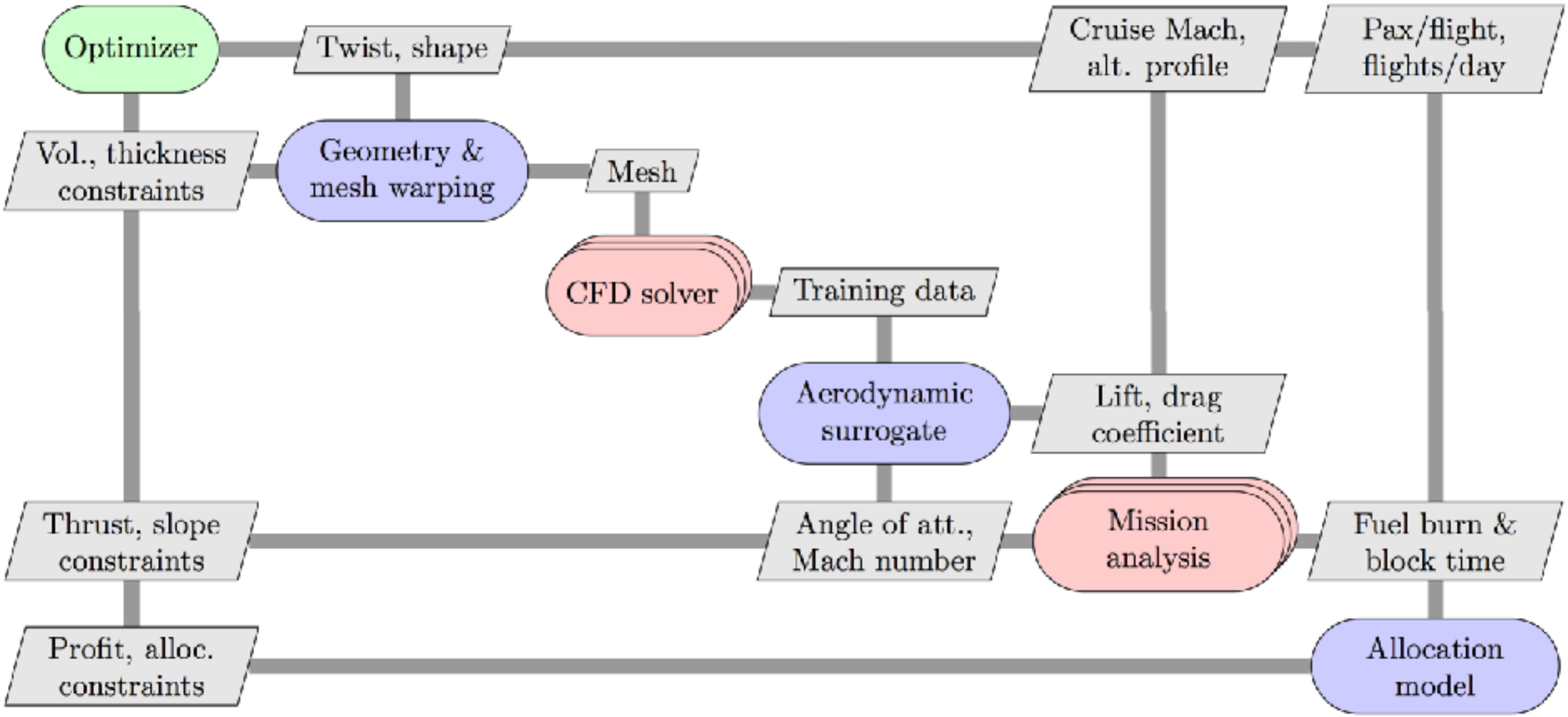
Potential issue: Need to compute the performance of aircraft in a 128-route network



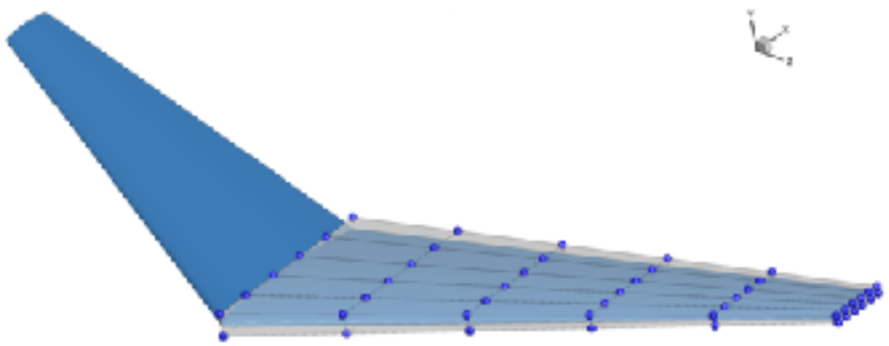
But this requires…

128 missions	x	O(100) mission points	x	O(10) mission iterations	x	O(100) optimization iterations	=	$O(10^7)$ CFD evaluations

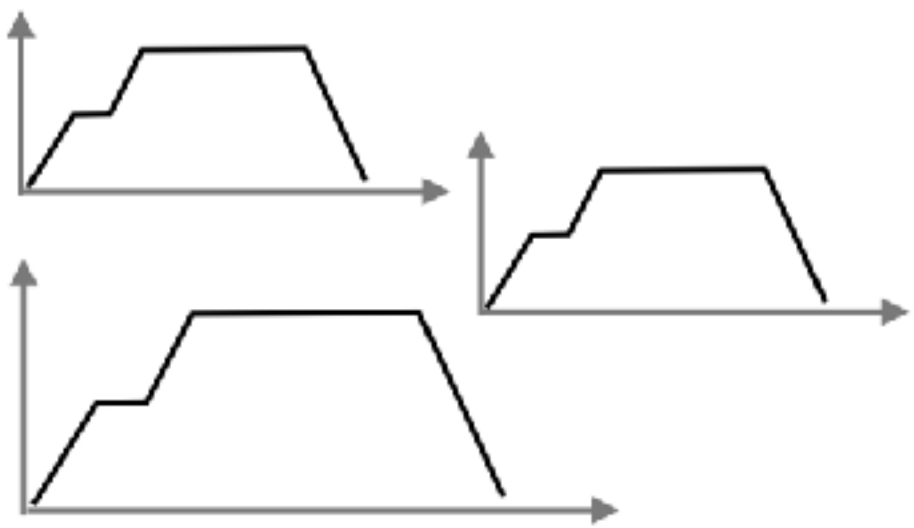
The solution is to replace the CFD with a dynamically re-trained surrogate model



The allocation-mission-design optimization involves O(1000) design variables



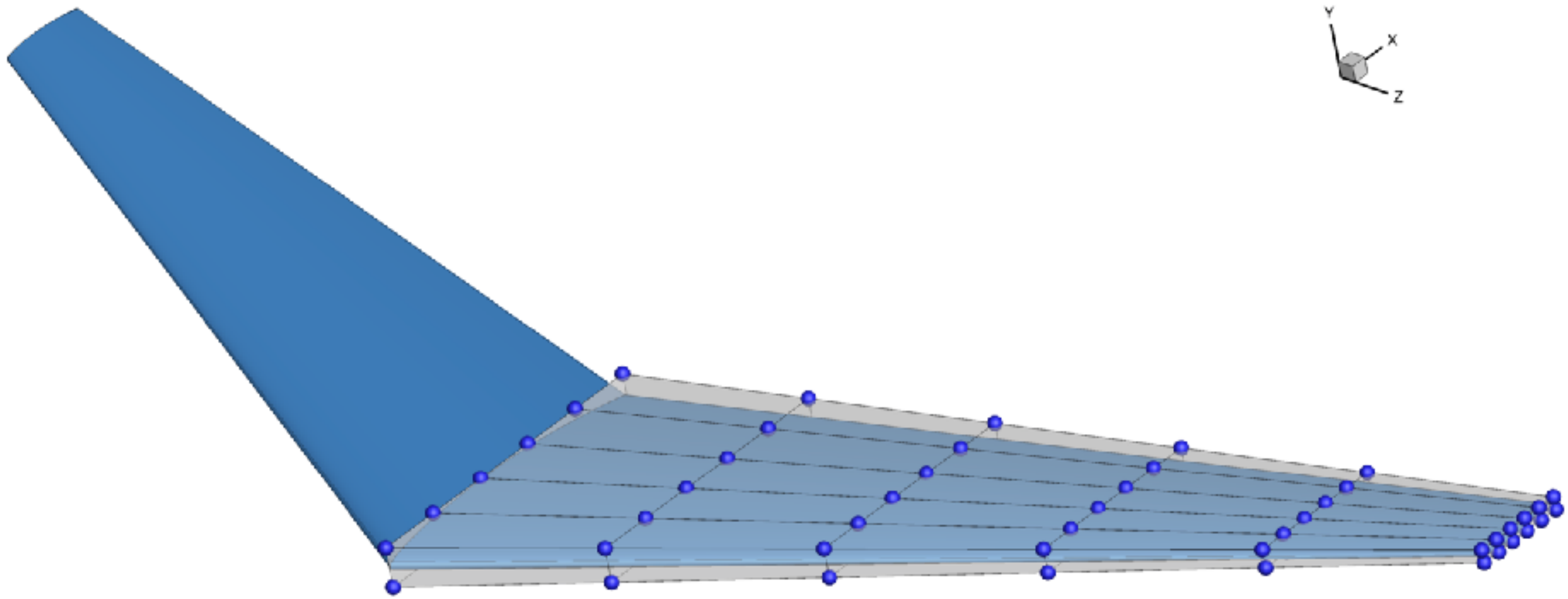
O(100)
Aircraft geometric variables



O(1000)
Mission profile variables

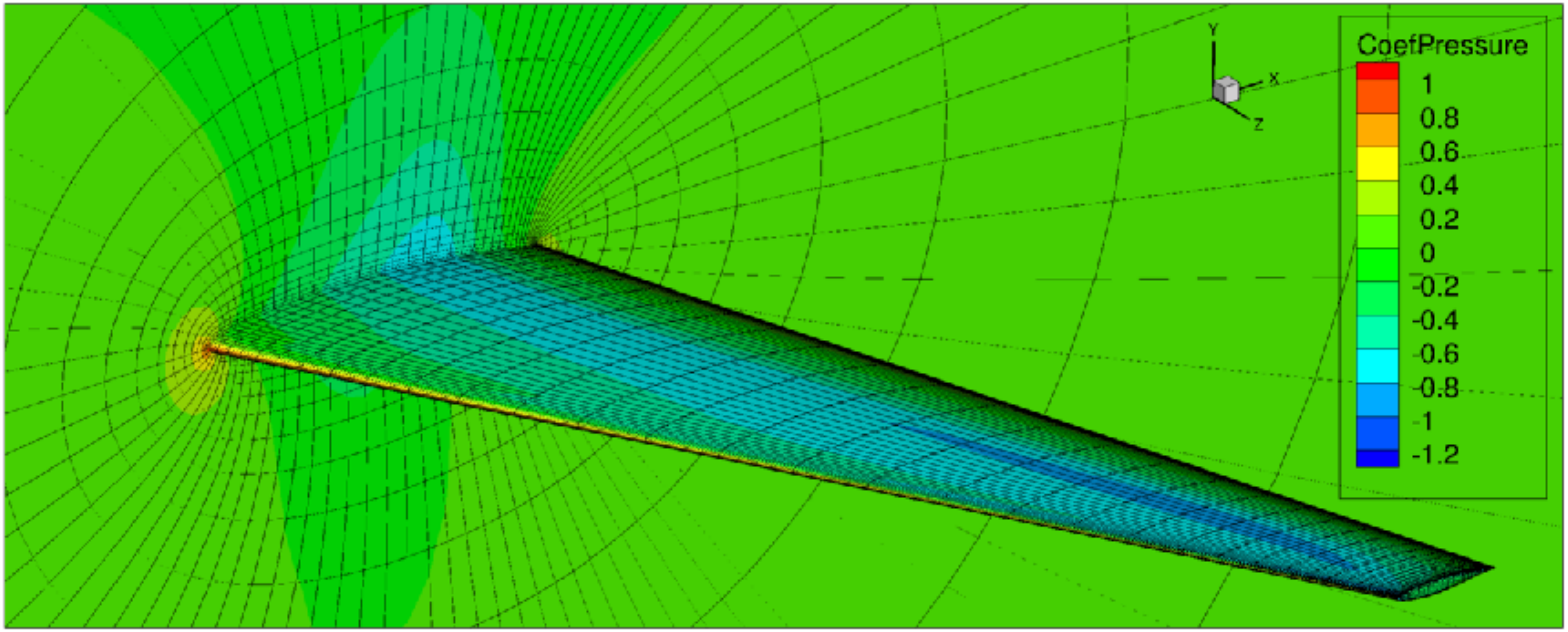
O(100)
Airline allocation variables

We parametrize a B717-based wing with shape and twist variables



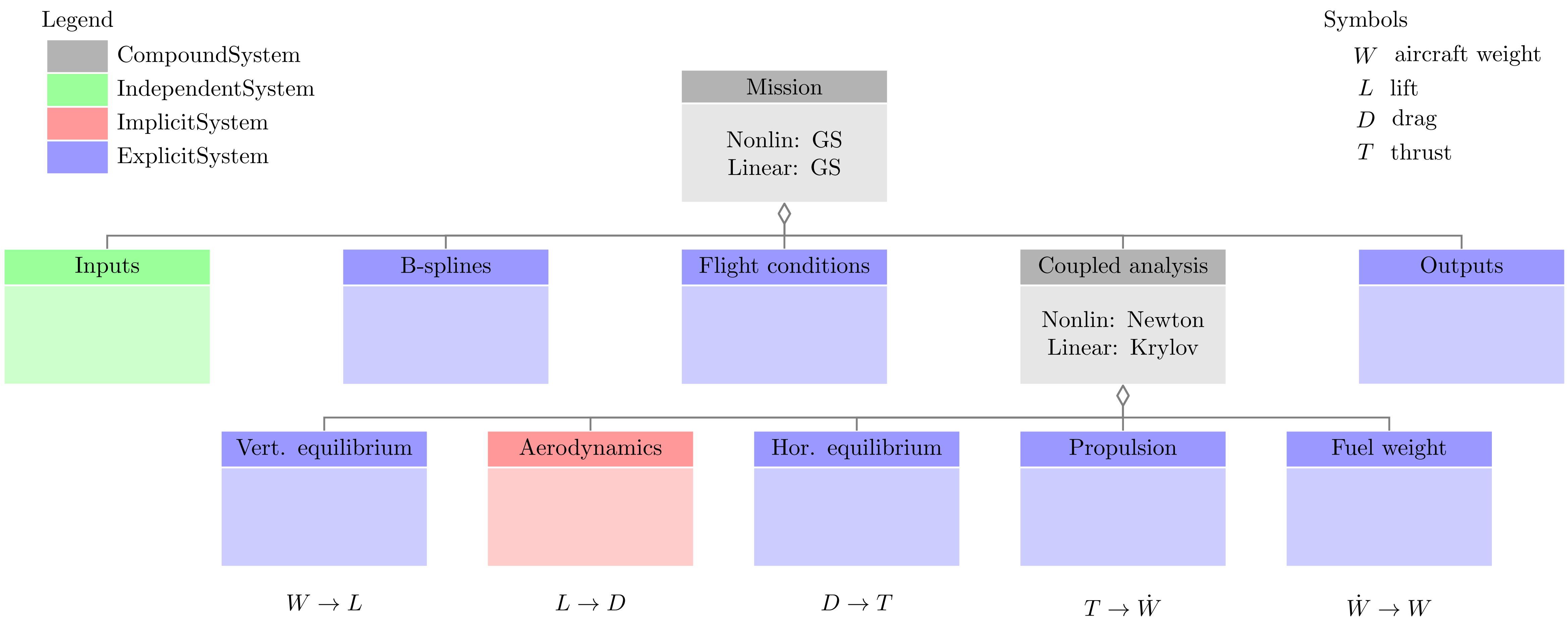
Free-form deformation

We use CFD to solve the Euler equations



SUMad flow solver

We developed a unique mission analysis tool within the parallel framework



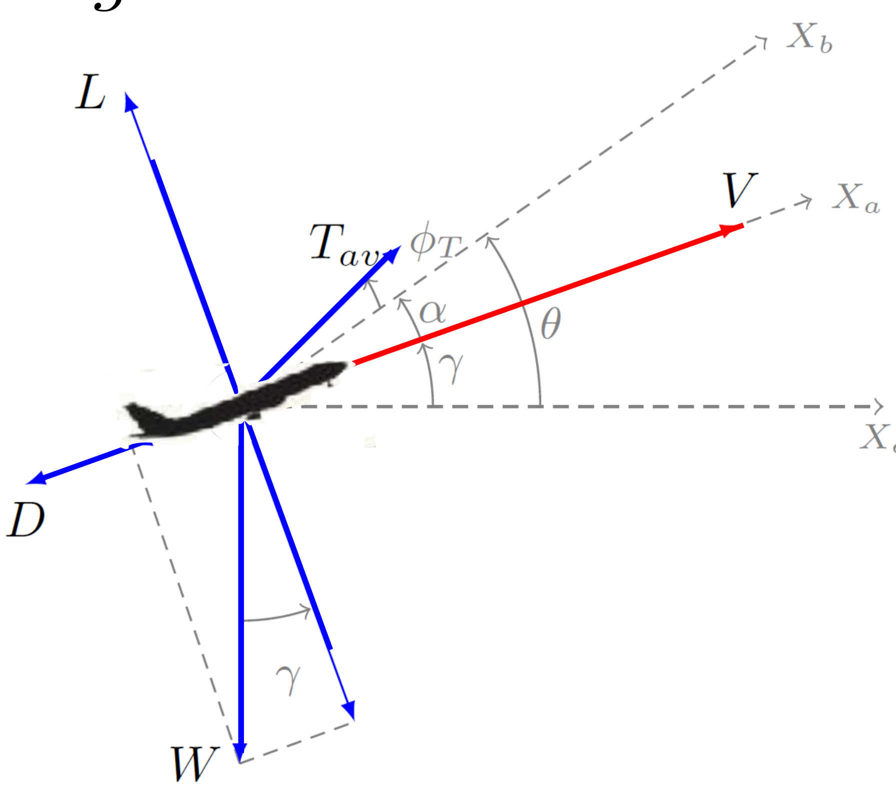
The framework automatically computes derivatives using the adjoint method

The mission analysis solves the flight equilibrium equations

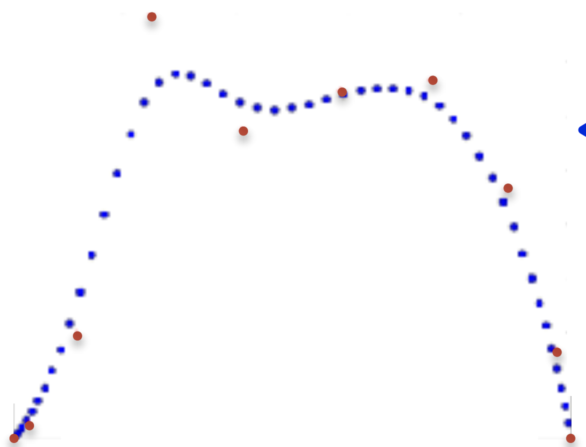
$$L + W\cos\gamma - T\sin\alpha + \frac{W}{g}v^2\cos\gamma\frac{\mathrm{d}\gamma}{\mathrm{d}x} = 0$$

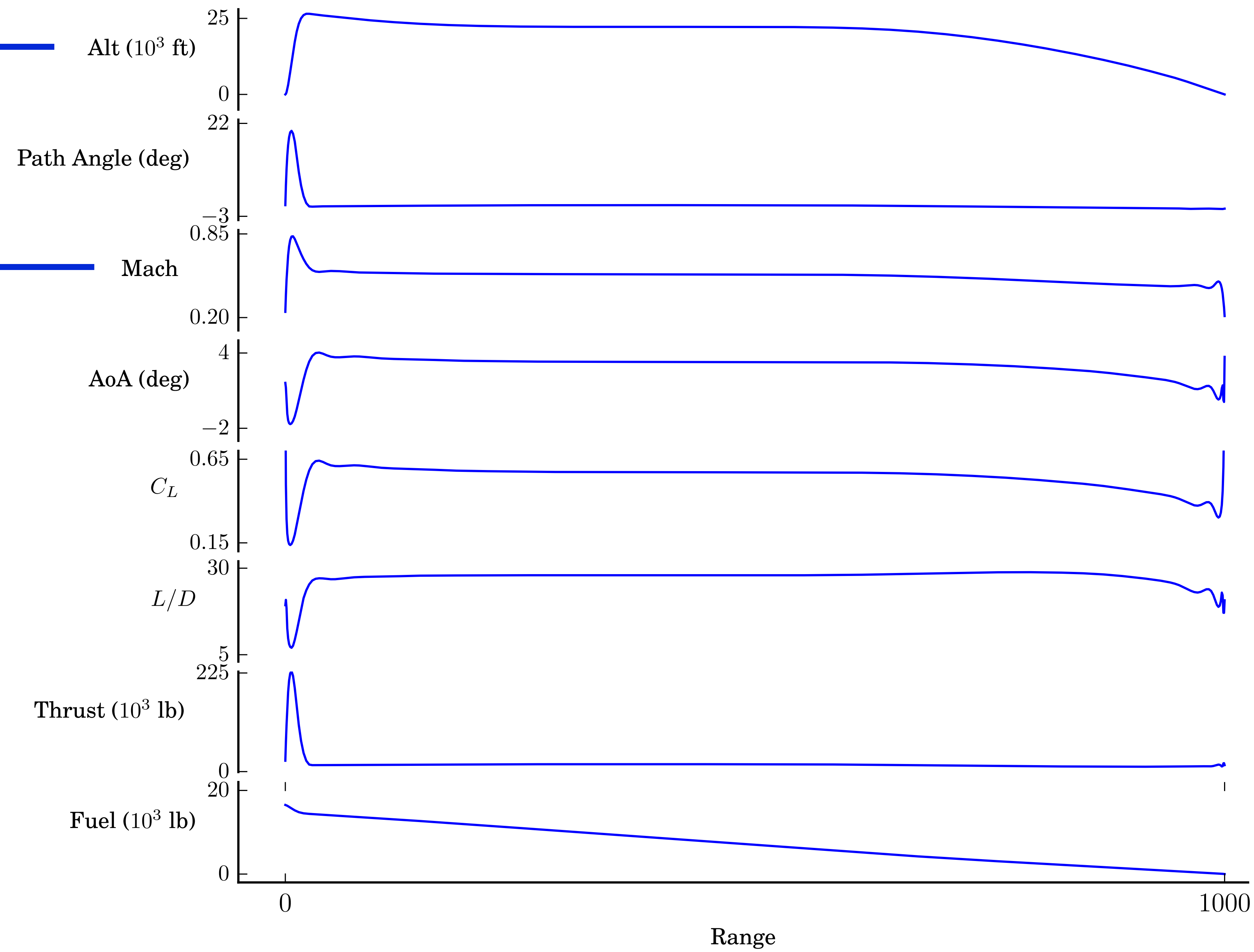
$$T\cos\alpha + D + W\sin\gamma + \frac{W}{g}v\cos\gamma\frac{\mathrm{d}v}{\mathrm{d}x} = 0$$

$$\frac{\mathrm{d}W_f}{\mathrm{d}x} = \frac{\mathrm{SFC}\,\frac{1}{2}\rho v^2 S C_T}{v\cos\gamma}$$



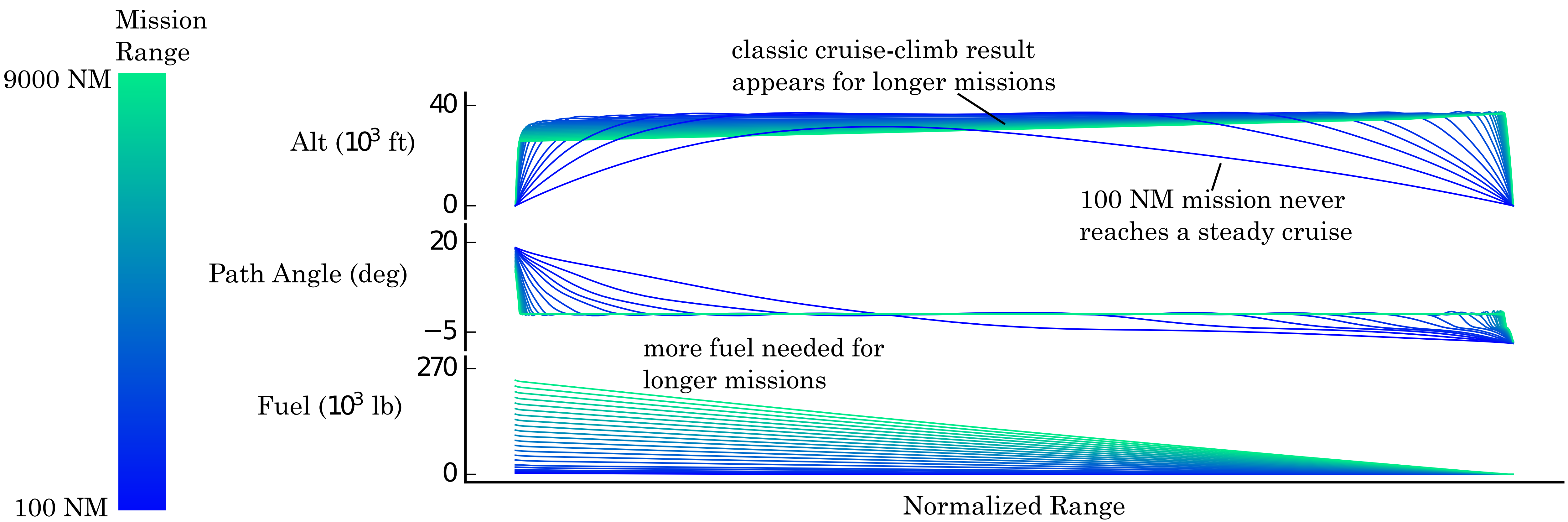
The altitude and Mach profiles can be optimized using a B-spline parametrization





The remaining variables are computed by solving a system of equations

Multiple trajectories can be optimized quickly

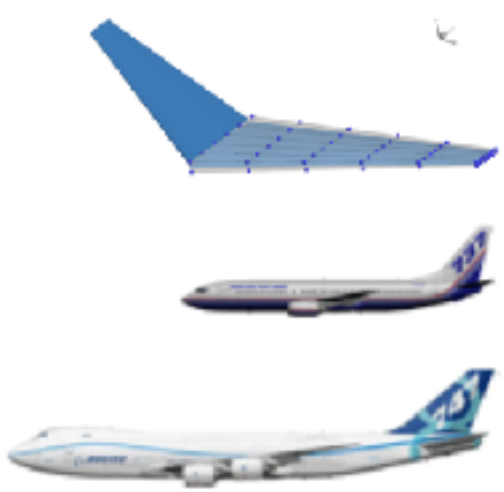


[Kao, Hwang , Martins, Gray, and Moore, *AIAA 2015-0136*]

Allocation problem:
Maximize airline profit for a given fleet and network

This is a mixed-integer optimization problem

[Hwang et al., AIAA 2015-0900]



	Rt 1	Rt 2	Rt 3
NG			
B737			
B747			

Design variables:
1. passengers per flight
2. flights per day

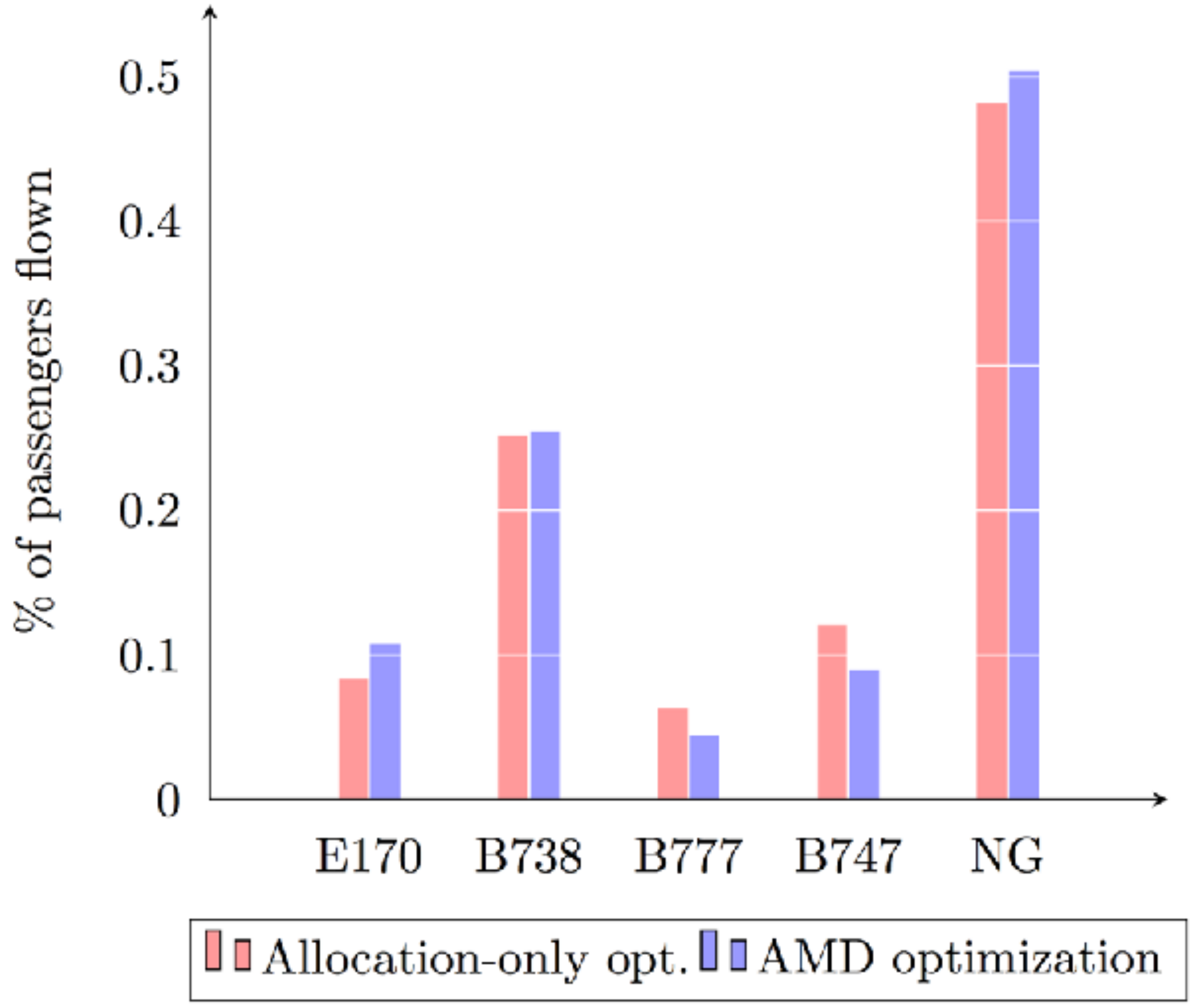
Aircraft availability constraints

Passenger demand constraints

The optimization problem contains over 6000 design variables

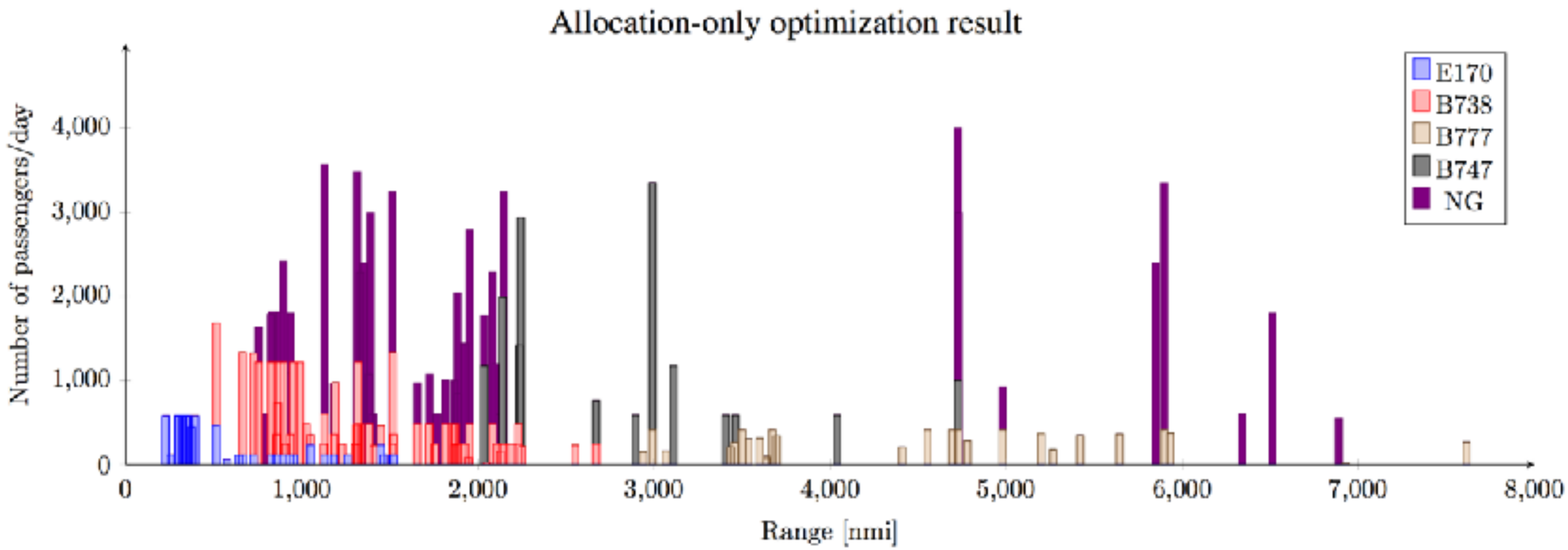
	Variable	Quantity
maximize	profit	
with respect to	twist	6
	shape	72
	cruise Mach number for each route (between 0.6 and 0.82)	1×128
	altitude control points for each route	4575
	passengers per flight for each aircraft type and route	5×128
	flights per day for each aircraft type and route	5×128
	Total number of design variables	6061
subject to	wing volume constraint	1
	wing thickness constraints	100
	idle thrust KS constraint for each route	128
	max thrust KS constraint for each route	128
	linear climb angle bounds for each mission	22875
	demand constraint for each route	128
	total flight time constraint for each aircraft	5
	Total number of constraints	23365

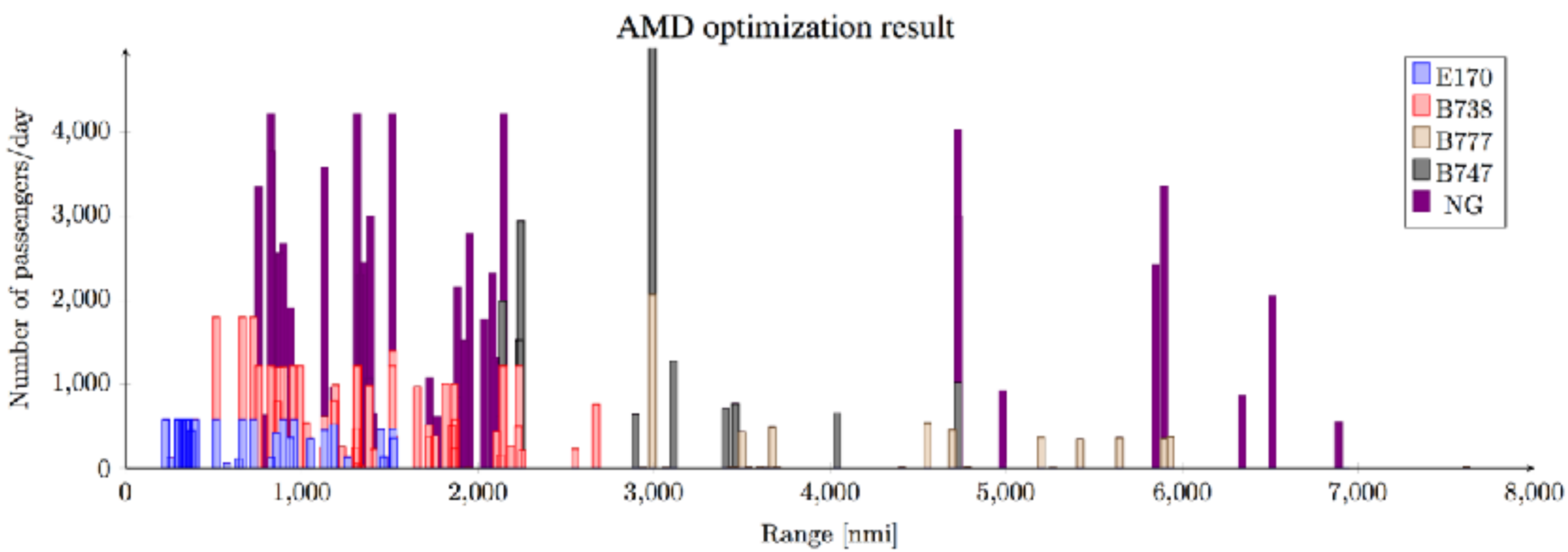
Allocation-mission-design optimization yields a 27% increase in airline profit



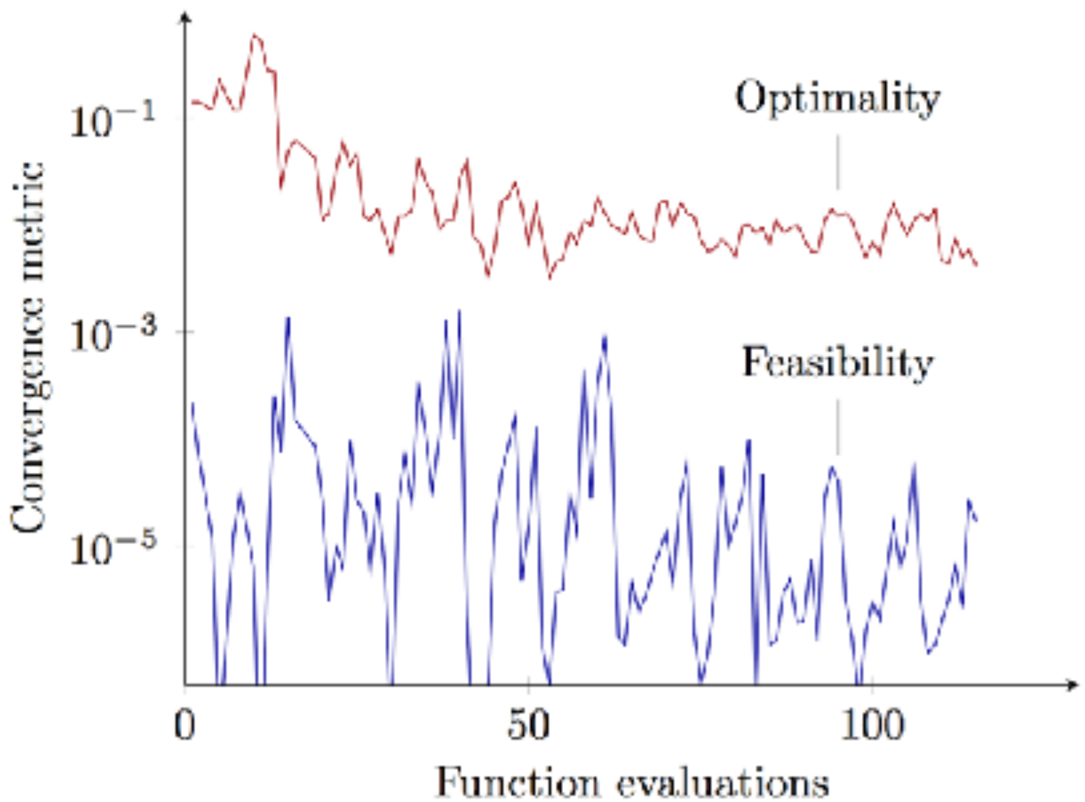
$23.4 million $29.8 million

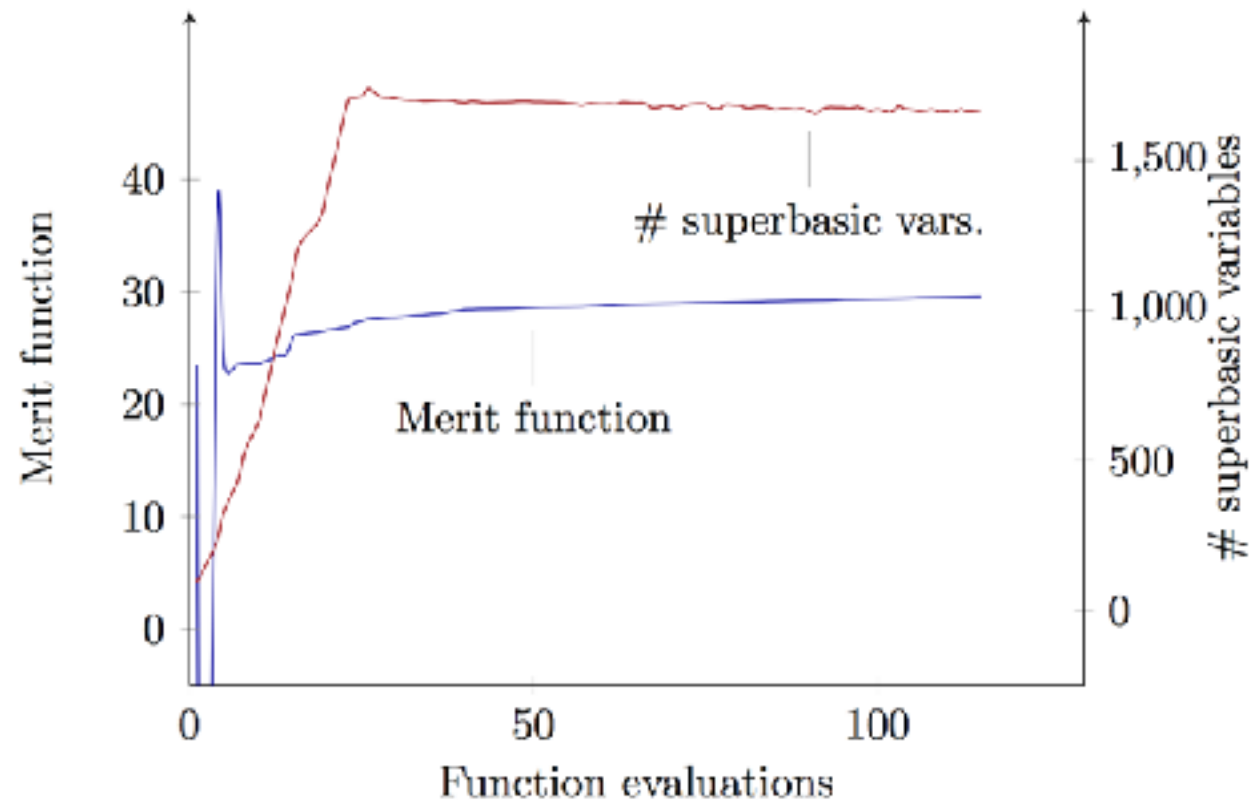
For the AMD optimization, the next-generation aircraft is flown more on the short range routes



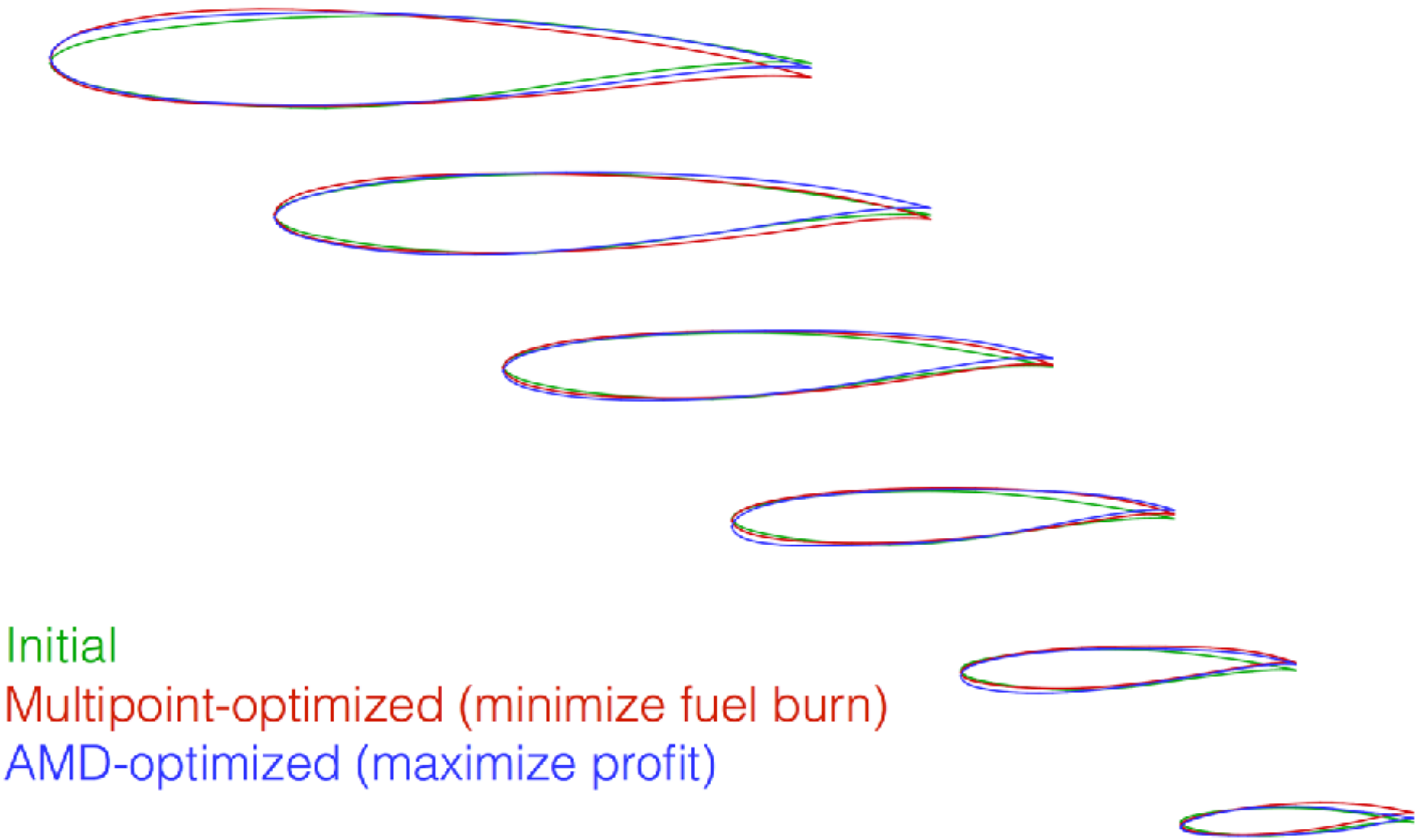


Despite the 6000 design variables, the optimizer uses only ~100 iterations

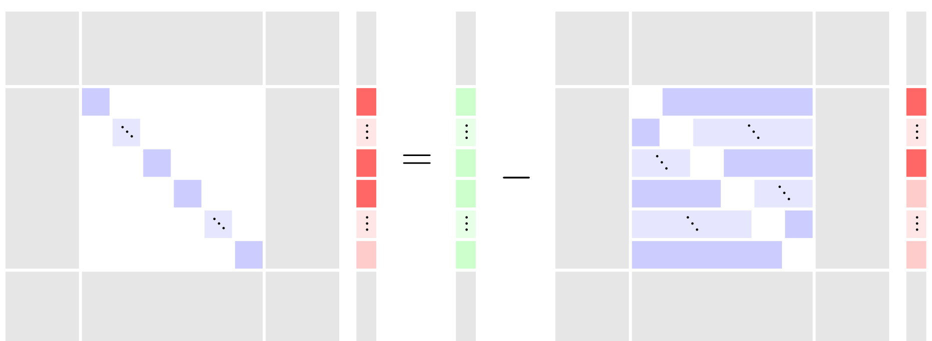


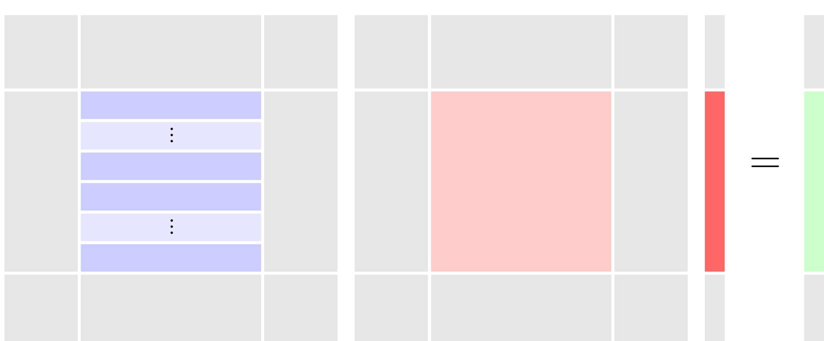


The AMD-optimized wing shape is different from the multipoint result



Summary





- Introduced MDO architectures
- Developed MAUD, a novel algorithmic framework for coupled analysis and gradient computation
- MAUD demonstrated in large-scale MDO problems, including high-fidelity models
- Implemented MAUD architecture in OpenMDAO

Go forth and optimize!

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

AEROSPACE ENGINEERING
UNIVERSITY of MICHIGAN

NASA

MDOlab Newsletter—Fall 2014

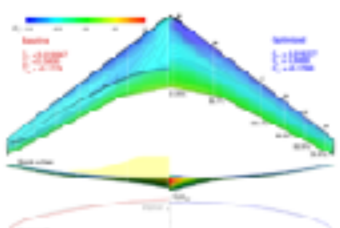
Dear Friend,
Welcome to the MDOlab newsletter, an update on research and open source software that we send a few times a year. You are receiving this because I think you are interested in numerical optimization, MDO, engineering design, or aircraft design. If this is not the case, feel free to unsubscribe. If you know someone who might like to subscribe, please forward them this newsletter. Best regards,
-Joaquim Martins

Subscribe

Like Tweet Pin it +1 Share

Latest publications

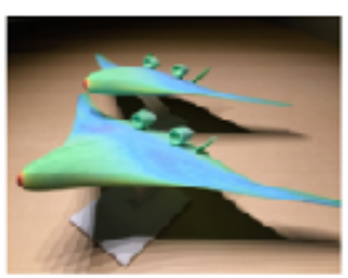
Wing aerodynamic shape optimization benchmark



The AIAA Aerodynamic Design Optimization Discussion Group developed a series of benchmark cases. In this paper, we solve the RANS-based wing optimization problem, try to find multiple local minima, and solve a number of related wing design optimization problems. The initial and optimized geometries and meshes are provided here.

[Paper] [Preprint] [Optimization movie]

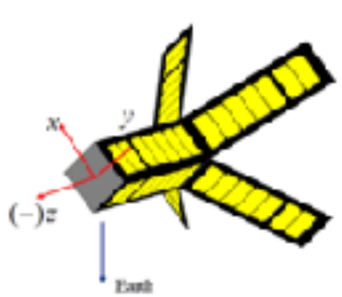
Aerodynamic design optimization of a blended-wing body aircraft



This builds on our previous work on stability-constrained flying wing optimization. A series of RANS-based aerodynamic design optimization studies shows the tradeoffs between drag, trim, and stability for the NASA/Boeing BWB. The photo on the left shows 3D-printed models with pressure colormaps.

[Paper] [Preprint]

Satellite multidisciplinary design optimization benchmark



In collaboration with NASA and the Michigan Exploration Lab, we developed a new large-scale benchmark MDO problem, and solved a problem with 25,000 design variables and 2.2 million state variables by optimizing the data downloaded from a CubeSat subject to operational and physical constraints. This problem is now a plugin in the OpenMDAO open source project.

[Paper] [Preprint]

MARIE CURIE ACTIONS

AEROSPACE ENGINEERING
UNIVERSITY of MICHIGAN

NASA

Download our publications and subscribe to our newsletter at:

http://mdolab.engin.umich.edu

Selected publications

1. J. R. R. A. Martins and J. T. Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. AIAA Journal, 51(11):2582–2599, November 2013. doi: 10.2514/1.J052184.
2. J. R. R. A. Martins and A. B. Lambe. Multidisciplinary design optimization: A survey of architectures. AIAA Journal, 51(9):2049–2075, September 2013. doi:10.2514/1.J051895.
3. J. T. Hwang, D. Y. Lee, J. W. Cutler, and J. R. R. A. Martins. Large-scale multidisciplinary optimization of a small satellite's design and operation. Journal of Spacecraft and Rockets, 51(5):1648–1663, September 2014. doi: 10.2514/1.A32751.
4. G. K. W. Kenway and J. R. R. A. Martins. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. Journal of Aircraft, 51(1):144–160, January 2014. doi:10.2514/1.C032150.
5. G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins. Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and derivative computations. AIAA Journal, 52(5):935–951, May 2014. doi: 10.2514/1.J052255.
6. R. E. Perez, P. W. Jansen, and J. R. R. A. Martins. pyOpt: a Python-based object-oriented framework for nonlinear constrained optimization. Structural and Multidisciplinary Optimization, 45(1):101–118, January 2012. doi:10.1007/s00158-011-0666-3.
7. J. T. Hwang, S. Roy, J. Y. Kao, J. R. R. A. Martins, and W. A. Crossley. Simultaneous aircraft allocation and mission optimization using a modular adjoint approach. In Proceedings of the 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Kissimmee, FL, Jan. 2015. AIAA 2015-0900.
8. J. Y. Kao, J. T. Hwang, J. R. R. A. Martins, J. S. Gray, and K. T. Moore. A modular adjoint approach to aircraft mission analysis and optimization. In Proceedings of the AIAA Science and Technology Forum and Exposition (SciTech), Kissimmee, FL, January 2015. AIAA 2015-0136.