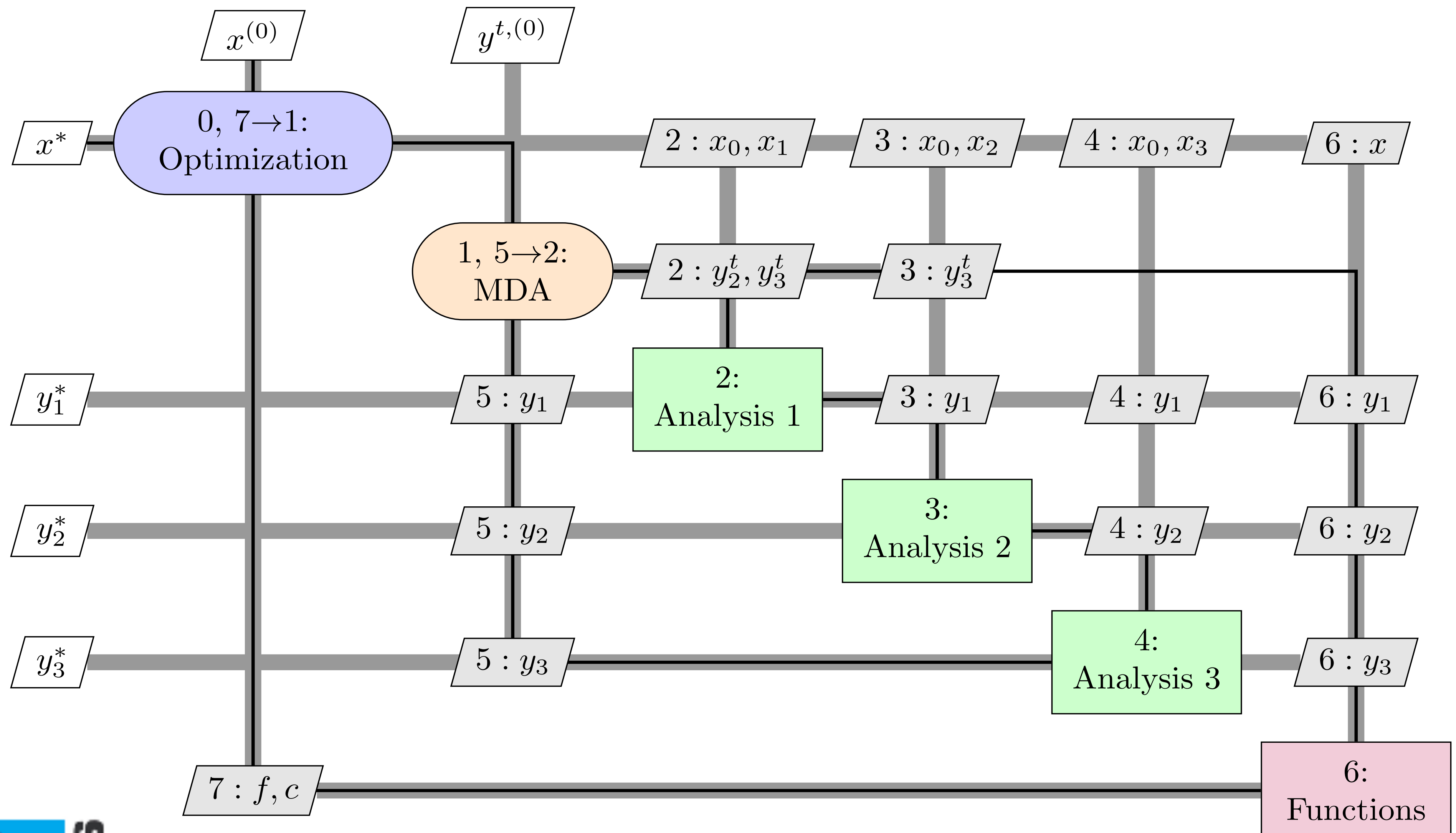
A Very Short Course on Multidisciplinary Design Optimization



Joaquim R. R. A. Martins • John T. Hwang
Multidisciplinary Design Optimization Laboratory
http://mdolab.engin.umich.edu

MARIE CURIE ACTIONS

isae Institut Supérieur de l'Aéronautique et de l'Espace SUPAERO

Vitae

Bio

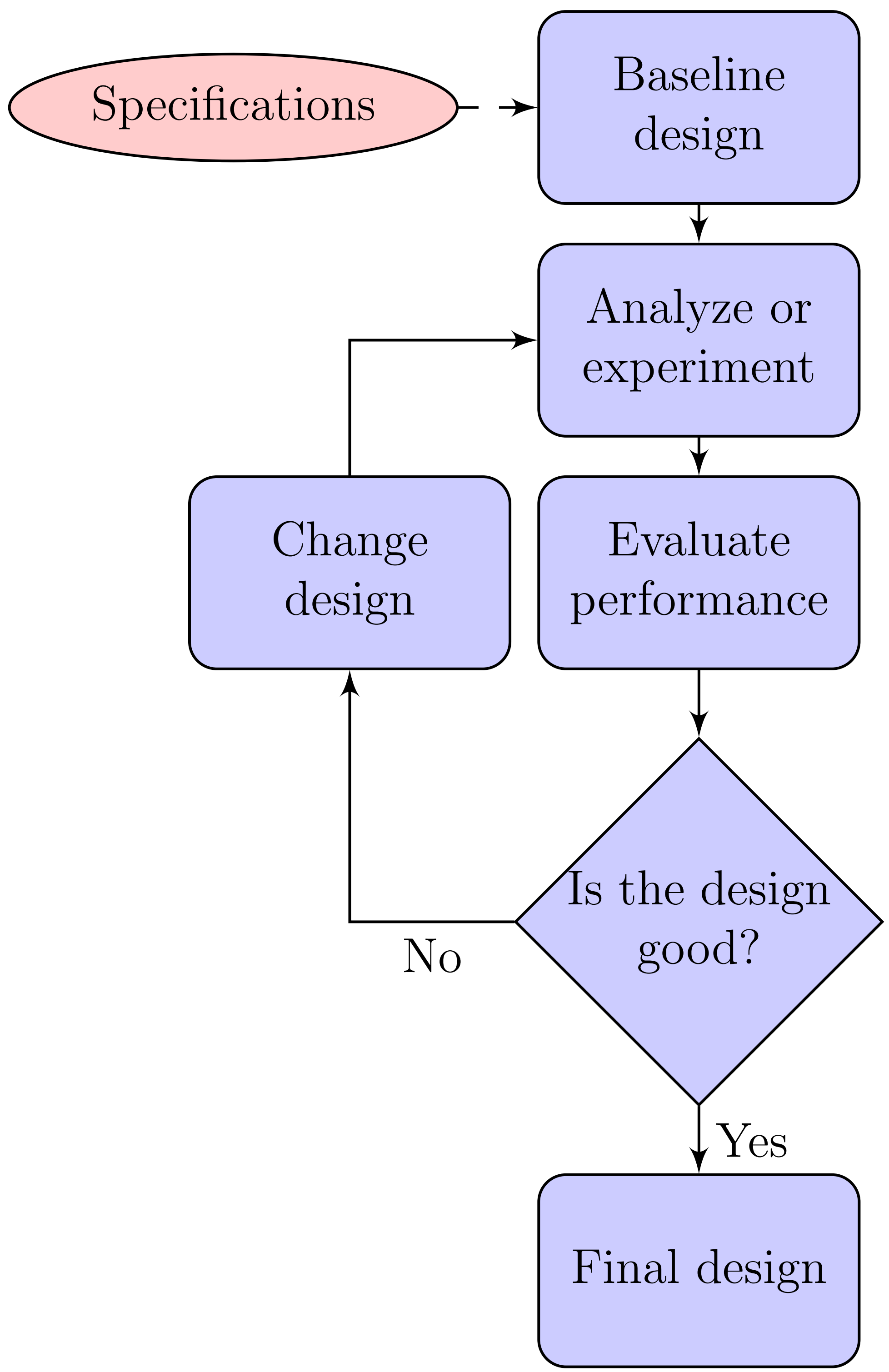
- 1991–1995: M.Eng. in Aeronautics, Imperial College, London, UK
- 1996–2002: M.Sc. and Ph.D. in Aeronautics and Astronautics, Stanford University
- 2002–2008: Assistant Professor, University of Toronto, Institute for Aerospace Studies
- 2008–2009: Associate Professor, University of Toronto, Institute for Aerospace Studies
- 2009–2015 : Associate Professor, University of Michigan, Dept. of Aerospace Engineering
- 2015– : Professor, University of Michigan, Dept. of Aerospace Engineering
- 2015–2016: Professeur Invité, ISAE–SUPAERO

Highlights

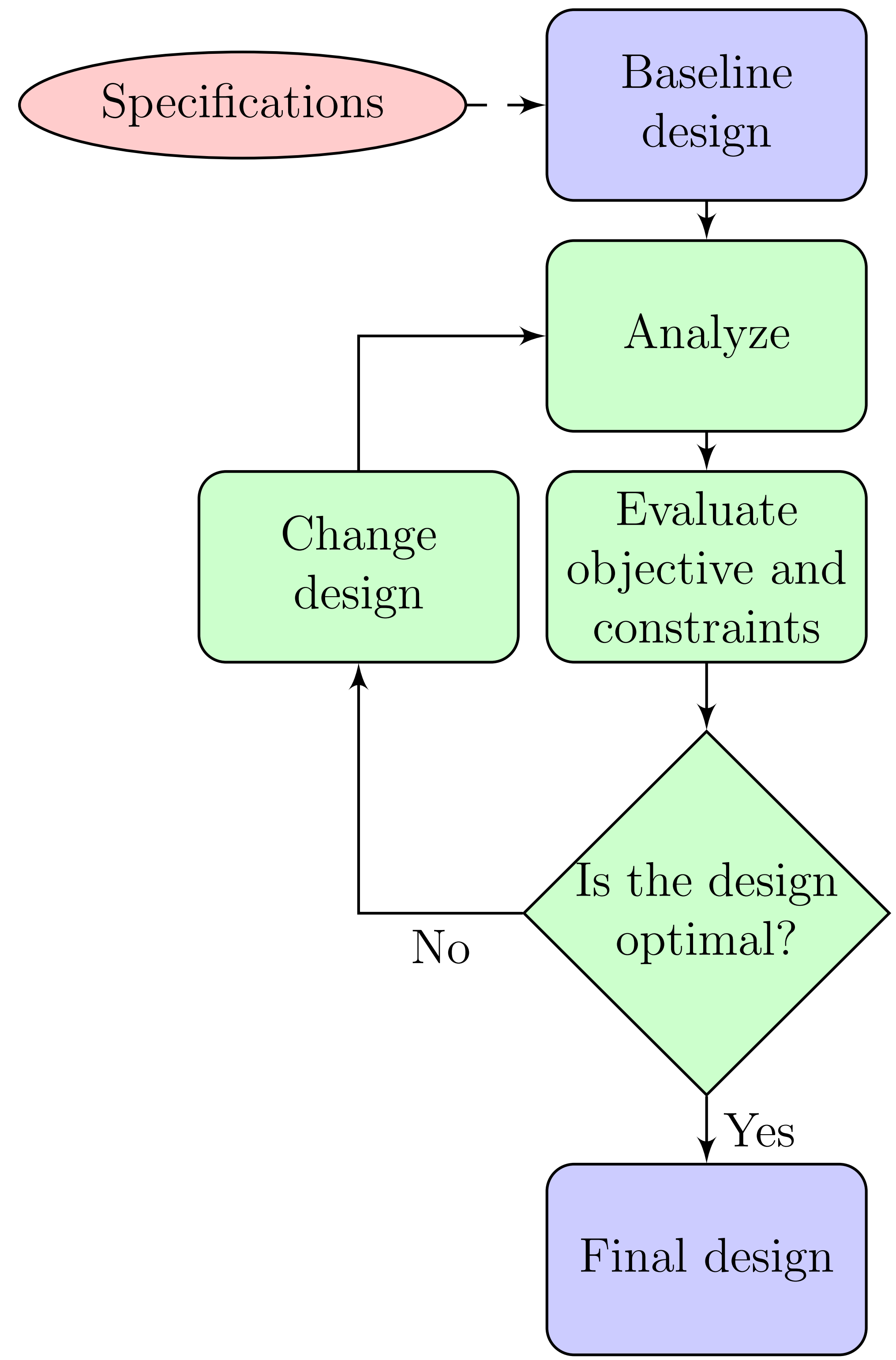
- 4 best papers at the *AIAA Multidisciplinary Analysis and Optimization Conference* (2002, 2006, 2012, 2014)
- Canada Research Chair in Multidisciplinary Optimization (2002–2009)
- Keynote speaker at the *International Forum on Aeroelasticity and Structural Dynamics* (Stockholm, 2007)
- Keynote speaker at the *Aircraft Structural Design Conference* (London, 2010)
- Associate editor for *Optimization and Engineering, Structural and Multidisciplinary Optimization*
- Marie Curie Fellow (2015–2016)

Design Optimization

What is Multidisciplinary Design Optimization – MDO?

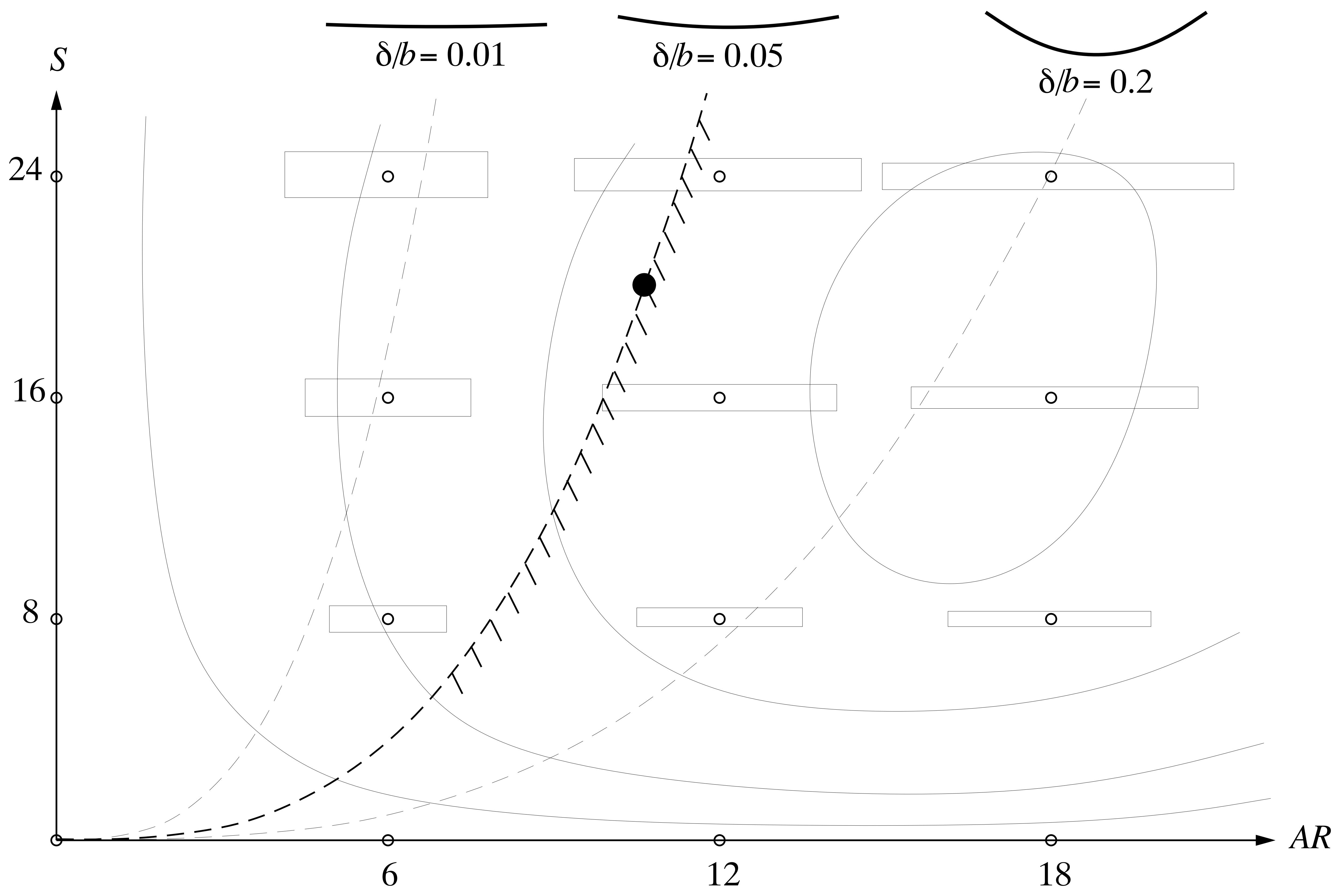


Conventional design



Optimal design

Design Variables, Objective Function, and Constraints



[Drela, 2006]

Contents 1

Multidisciplinary Design Optimization

Introduction 1

- In the last few decades, numerical models that predict the performance of engineering systems have been developed, and many of these models are now mature areas of research. For example . . .
- Once engineers can predict the effect that changes in the design have on the performance of a system, the next logical question is what changes in the design produced optimal performance. The application of the numerical optimization techniques described in the preceding chapters address this question.
- Single-discipline optimization is in some cases quite mature, but the design and optimization of systems that involve more than one discipline is still in its infancy.
- When systems are composed of multiple systems, additional issues arise in both the analysis and design optimization.
- MDO researchers think industry will not adopt MDO more widely because they do not realize their utility.

Introduction 2

- Industry think that researchers are not presenting anything new, since industry has already been doing multidisciplinary design.
- There is some truth to each of these perspectives ...
- Real-world aerospace design problem may involve thousands of variables and hundreds of analyses and engineers, and it is often difficult to apply the numerical optimization techniques and solve the mathematically correct optimization problems.
- The kinds of problems in industry are often of much larger scale, involve much uncertainty, and include human decisions in the loop, making them difficult to solve with traditional numerical optimization techniques.
- On the other hand, a better understanding of MDO by engineers in industry is now contributing a more widespread use in practical design.

Why MDO?

- Parametric trade studies are subject to the "curse of dimensionality".
- Iterated procedures for which convergence is not guaranteed.
- Sequential optimization that does not lead to the true optimum of the system

Introduction 3

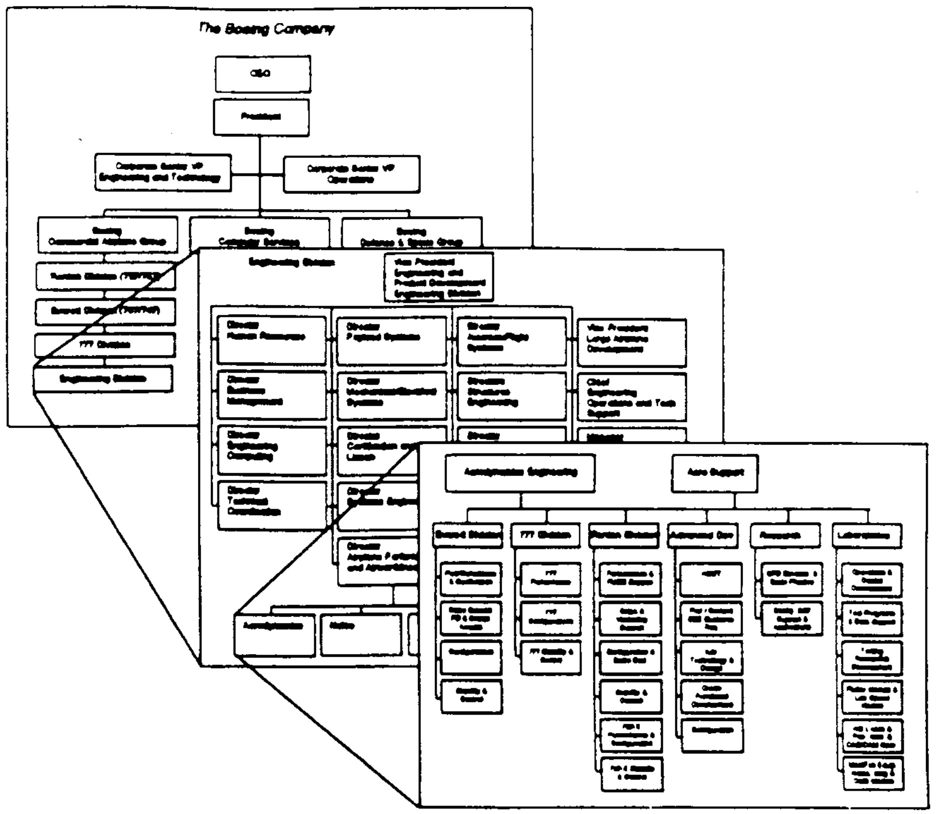
Objectives of MDO:

- Avoid difficulties associated with sequential design or partial optimization.
- Provide more efficient and robust convergence than by simple iteration.
- Aid in the management of the design process.

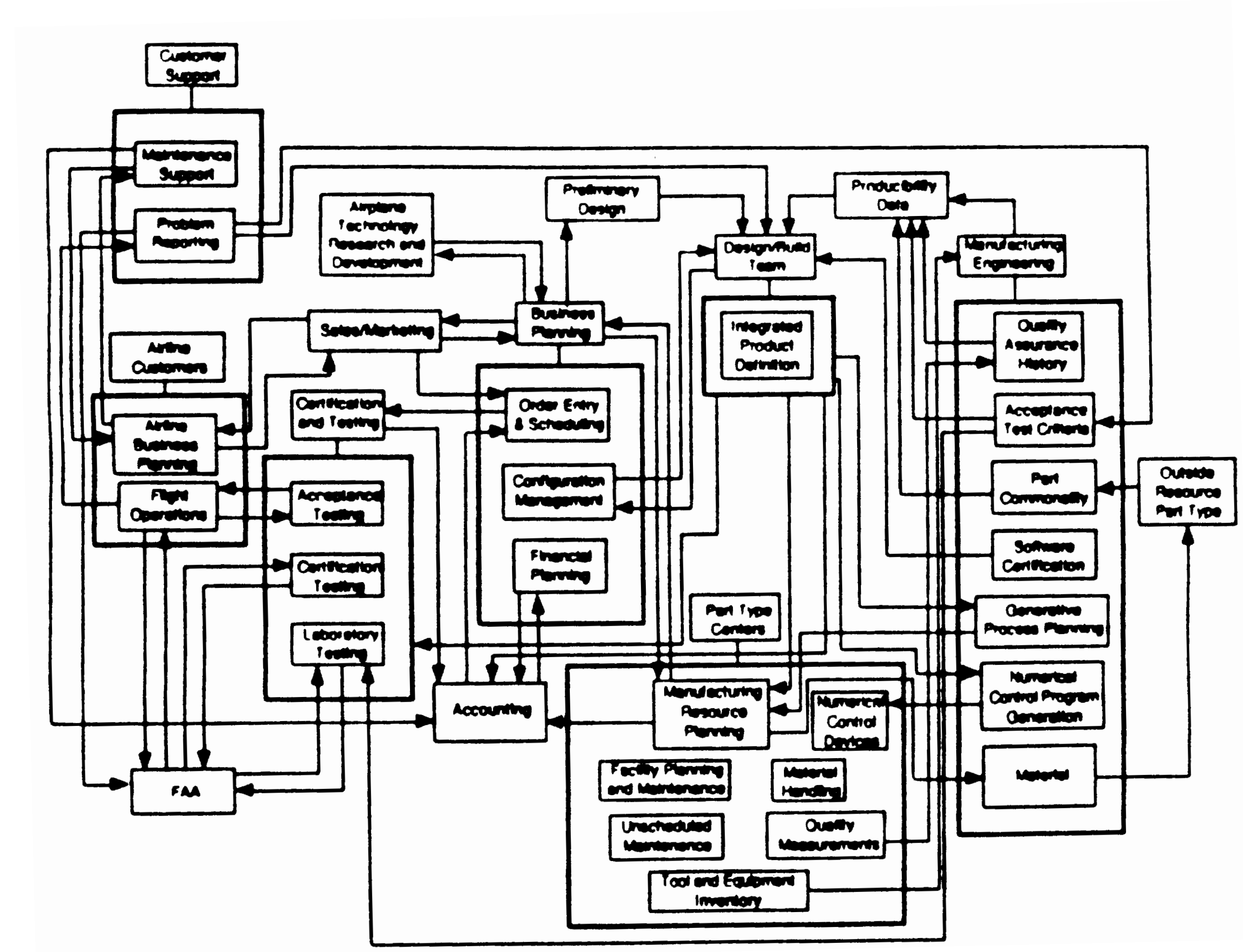
Difficulties of MDO:

- Communication and translation
- Time
- Scheduling and planning
- Implementation

Typical Aircraft Company Organization



Personnel hierarchy



Design process

MDO Architectures

- MDO focuses on the development of strategies that use numerical analyses and optimization techniques to enable the automation of the design process of a multidisciplinary system.
- The big challenge: make such a strategy scalable and practical.
- An MDO architecture is a particular strategy for organizing the analysis software, optimization software, and optimization subproblem statements to achieve an optimal design.
- Other terms are used: "method", "methodology", "problem formulation", "strategy", "procedure" and "algorithm".

Nomenclature and Mathematical Notation 1

Symbol	Definition
x	Vector of design variables
y^t	Vector of coupling variable targets (inputs to a discipline analysis)
y	Vector of coupling variable responses (outputs from a discipline analysis)
$\bar{y}$	Vector of state variables (variables used inside only one discipline analysis
f	Objective function
c	Vector of design constraints
c^c	Vector of consistency constraints
$\mathcal{R}$	Governing equations of a discipline analysis in residual form
N	Number of disciplines
$n_{()}$	Length of given variable vector
$m_{()}$	Length of given constraint vector
$()_0$	Functions or variables that are shared by more than one discipline
$()_i$	Functions or variables that apply only to discipline i
$()^*$	Functions or variables at their optimal value
$\tilde{()}$	Approximation of a given function or vector of functions
$\hat{()}$	Duplicates of certain variable sets distributed to other disciplines

Nomenclature and Mathematical Notation 2

- In MDO, we make the distinction between:
  - Local design variables x_i — directly affect only one discipline
  - Shared design variables x_0 — directly affect more than one discipline.
- Full vector of design variables $x = \left[x_0^T, x_1^T, \ldots, x_N^T\right]^T$
- A discipline analysis solves a system of equations that computes the state variables. Examples?
- In many formulations, independent copies of the coupling variables must be made to allow discipline analyses to run independently and in parallel.
- These copies are also known as target variables, which we denote by a superscript t.
- To preserve consistency between the coupling variable inputs and outputs at the optimal solution, we define consistency constraints

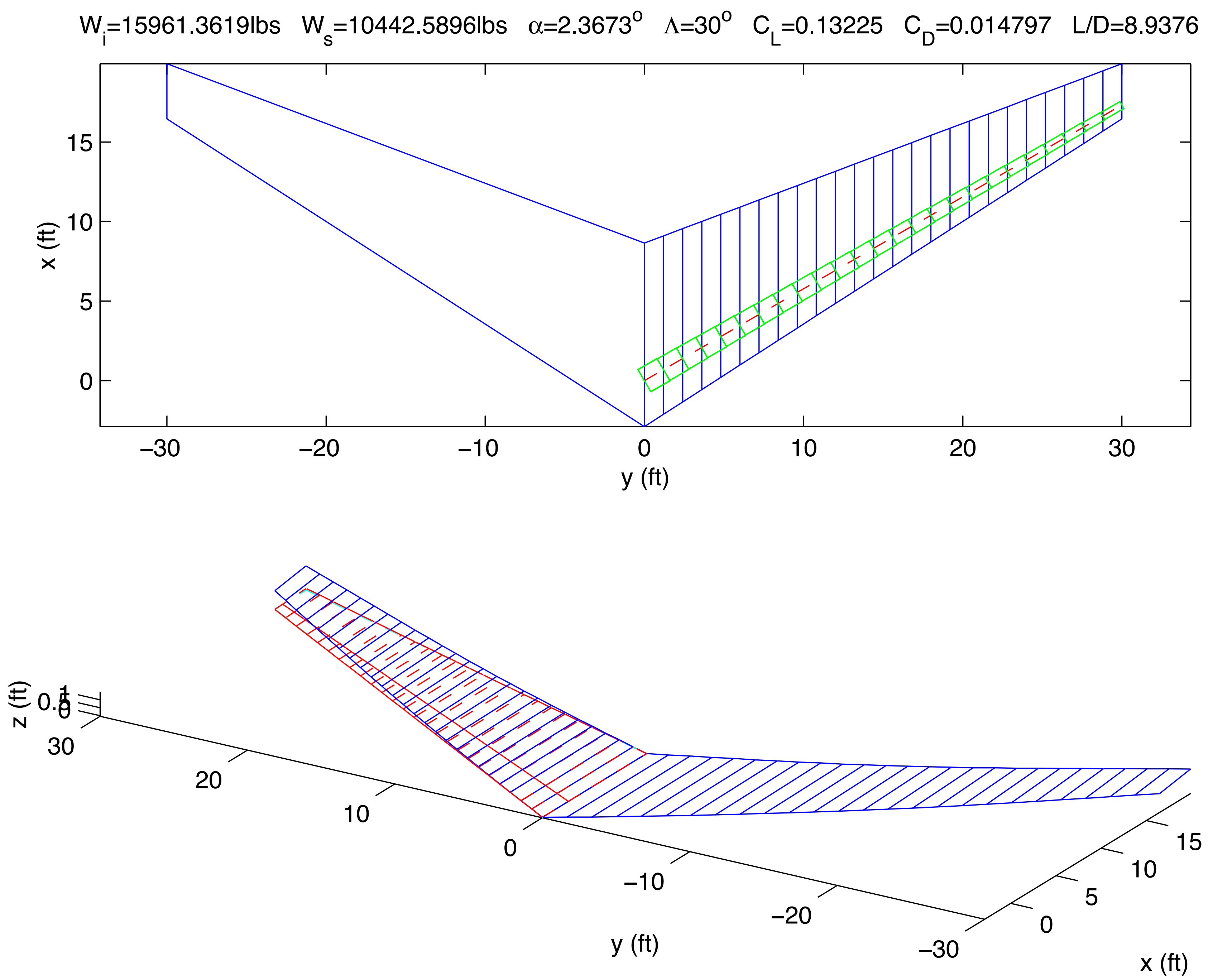
$$c_i^c = y_i^t - y_i$$

which we add to the optimization problem formulation.

Example: Aerostructural Problem Definition 1

- Common example used throughout this chapter to illustrate the notation and MDO architectures.
- Suppose we want to design the wing of a business jet using low-fidelity analysis tools.
- Model the aerodynamics using a panel method
- Model the structure as a single beam using finite elements

Example: Aerostructural Problem Definition 2



- Aerodynamic inputs: angle-of-attack (α), wing twist distribution (γ_i)
- Aerodynamic outputs: lift (L) and the induced drag (D).

Example: Aerostructural Problem Definition 3

- Structural inputs: thicknesses of the beam (t_i)
- Structural output: beam weight, which is added to a fixed weight to obtain the total weight (W), and the maximum stresses in each finite-element (σ_i).
- In this example, we want to maximize the range of the aircraft, as given by the Breguet range equation,

$$f = \text{Range} = \frac{V}{c}\frac{L}{D}\ln\left(\frac{W_i}{W_f}\right).$$

- The multidisciplinary analysis consists in the simultaneous solution of the following equations:

$$\begin{aligned}
\mathcal{R}_1 = 0 \Rightarrow &\quad A\Gamma - v(u, \alpha) = 0 \\
\mathcal{R}_2 = 0 \Rightarrow &\quad Ku - F(\Gamma) = 0 \\
\mathcal{R}_3 = 0 \Rightarrow &\quad L(\Gamma) - W = 0
\end{aligned}$$

Example: Aerostructural Problem Definition 4

- The complete state vector is

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} \Gamma \\ u \\ \alpha \end{bmatrix}.$$

- The angle of attack is considered a state variable here, and helps satisfy $L = W$.
- The design variables are the the wing sweep (Λ), structural thicknesses (t) and twist distribution (γ).

$$x_0 = \Lambda$$
$$x = \begin{bmatrix} t \\ \gamma \end{bmatrix},$$

- Sweep is a shared variable because changing the sweep has a direct effect on both the aerodynamic influence matrix and the stiffness matrix.

Example: Aerostructural Problem Definition 5

- The other two sets of design variables are local to the structures and aerodynamics, respectively.
- In later examples, we will see the options we have to optimize the wing in this example.

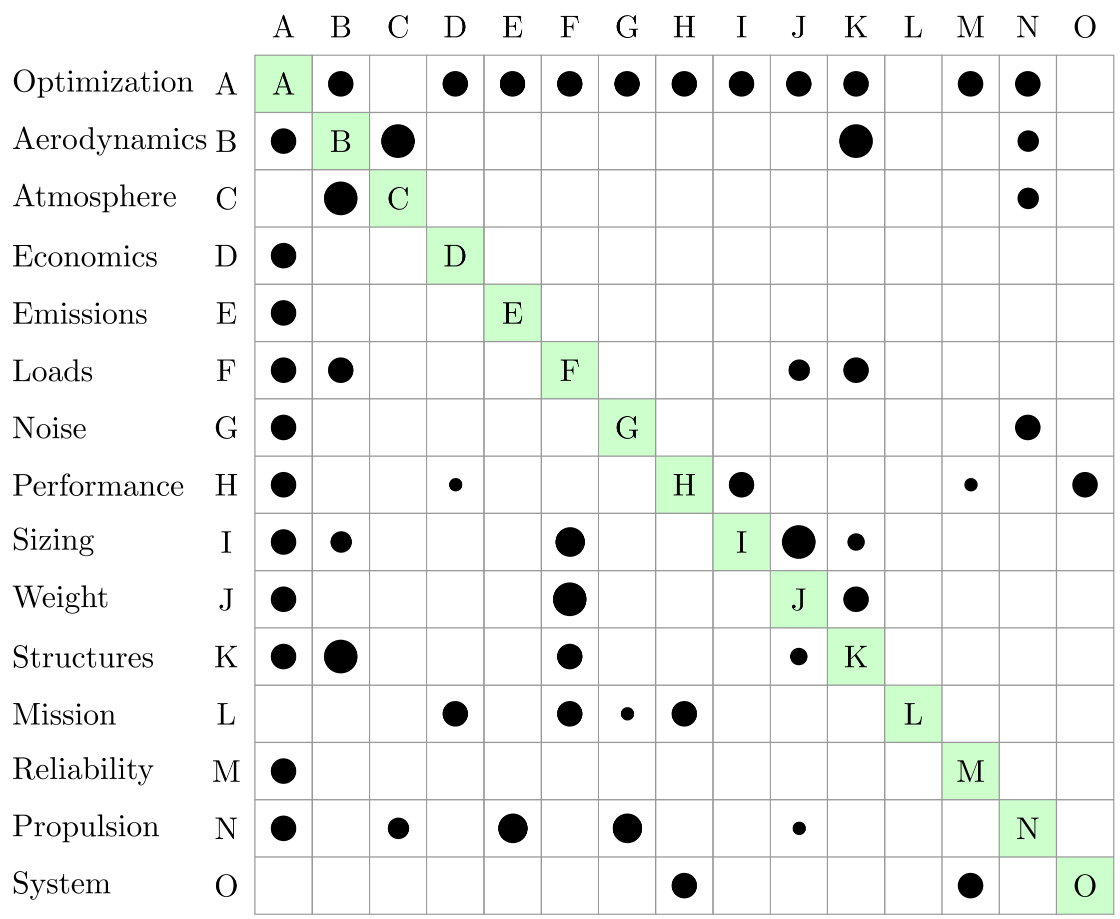
Multidisciplinary Analysis 1

- To find the coupled state of a multidisciplinary system we need to perform a multidisciplinary analysis — MDA.
- This is often done by repeating each disciplinary analysis until $y_i^t = y_i^r$ for all is.

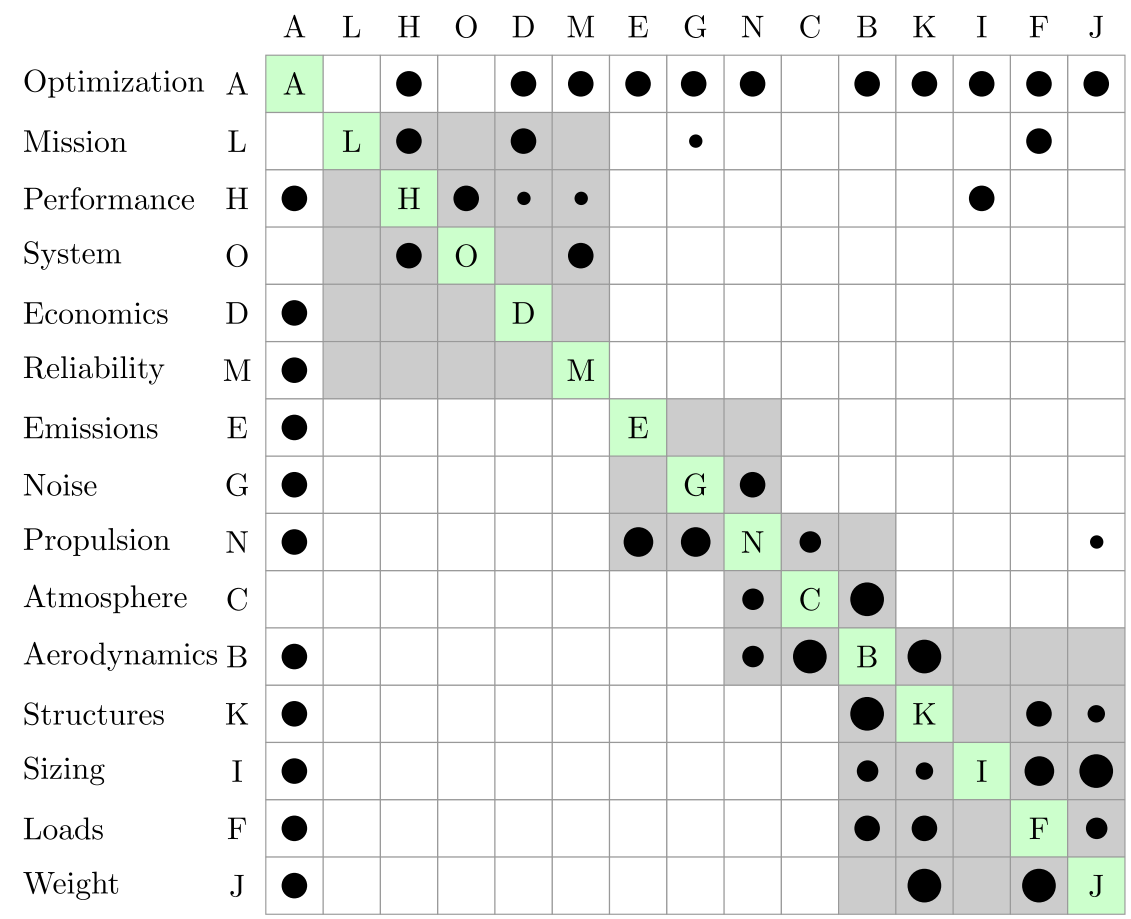
Input: Design variables x
Output: Coupling variables, y
0: Initiate MDA iteration loop
repeat
  1: Evaluate Analysis 1 and update $y_1(y_2, y_3)$
  2: Evaluate Analysis 2 and update $y_2(y_1, y_3)$
  3: Evaluate Analysis 3 and update $y_3(y_1, y_2)$
until $4 \rightarrow 1$: MDA has converged

Multidisciplinary Analysis 2

- The design structure matrix (DSM) was originally developed to visualize the interconnections between the various components of a system.



Original ordering



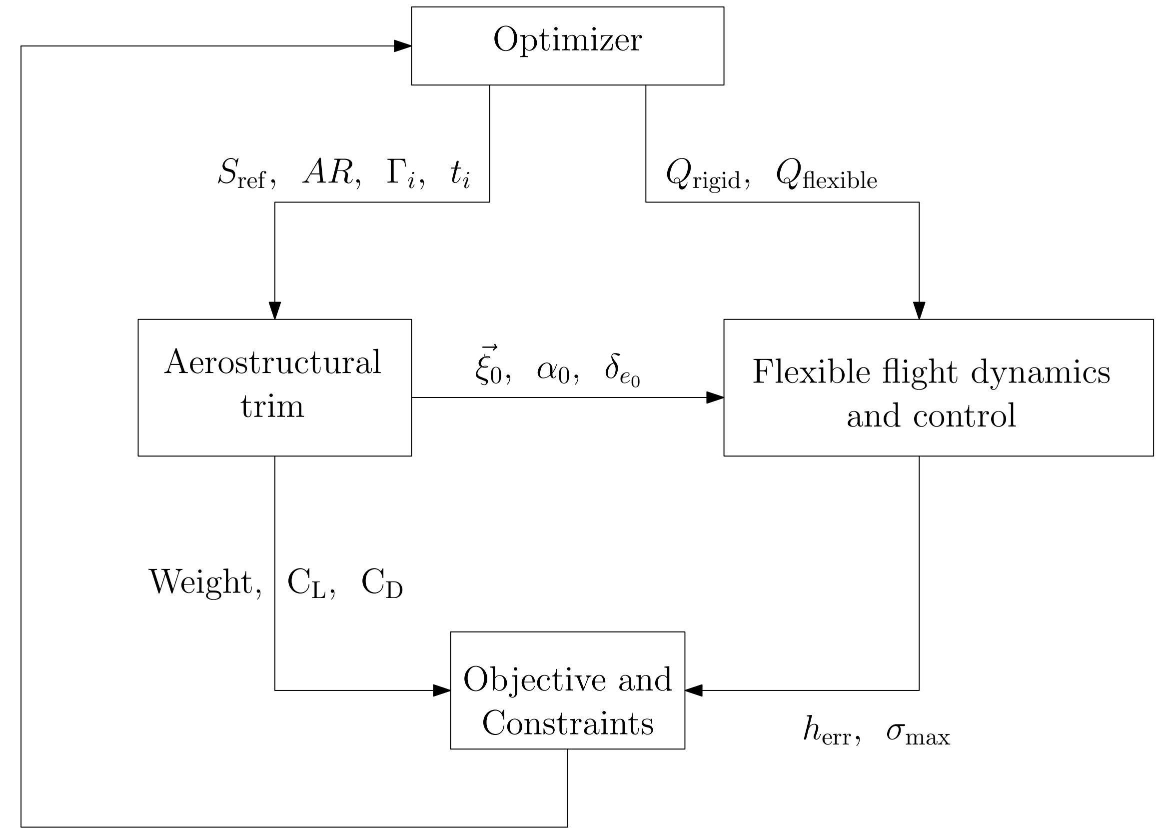
Improved ordering

- Fixed-point iteration, such as the Gauss–Seidel algorithm above converge slowly and sometimes do not converge at all.
- One way to improve the disciplines, is to reorder the sequence and possibly do some inner loops for more coupled clusters.

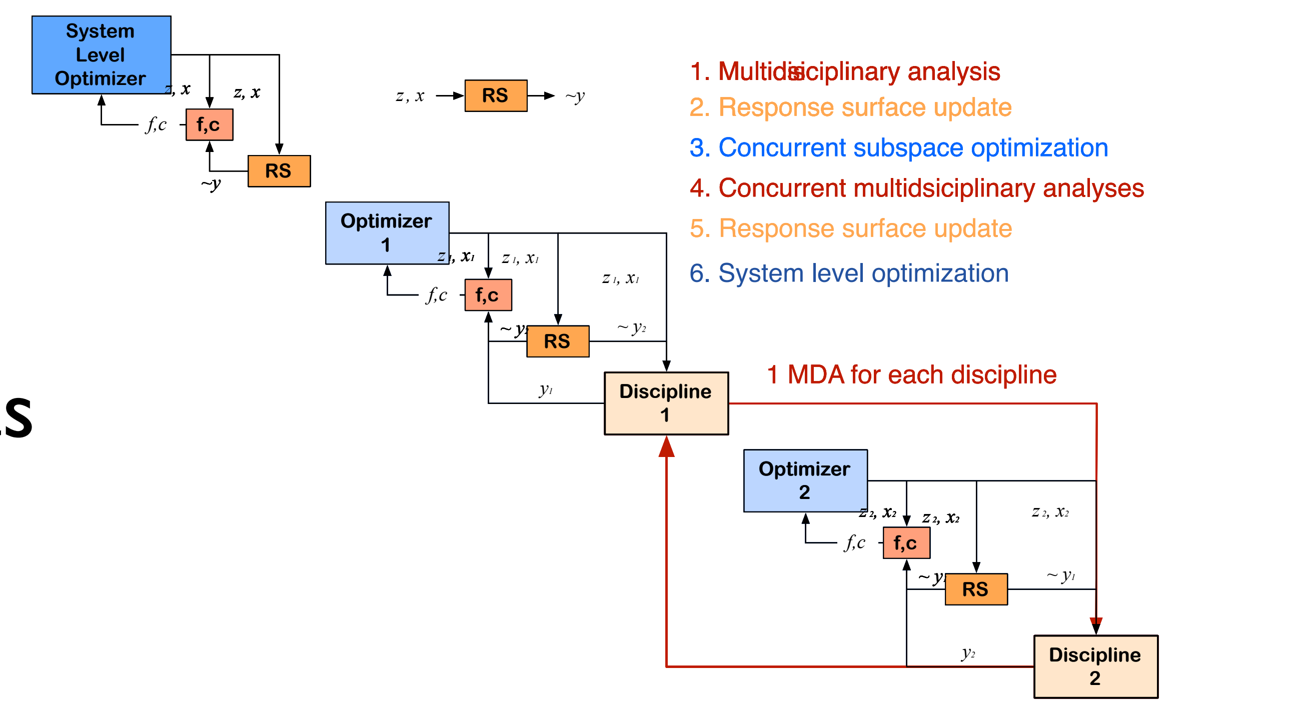
Extended Design Structure Matrix
A Unified Description of MDO Architectures

Motivation

- No comprehensive description of MDO architectures in a unified notation
- Often not enough detail when a given MDO architecture is described
- Flow diagrams are not standardized
- Flow diagrams do not provide as much information as they could — lack of information density



[Haghighat, Liu and Martins, 2009]



[Martins and Lambe, *Consortium Workshop*, 2009]

Notation and Problem Statement

$$
\begin{array}{rl}
\text{minimize} & f_0(x, y) \\
\text{with respect to} & x, y^t, \bar{y}, y \\
\text{subject to} & c_0(x, y) \geq 0 \\
& c_i(x_0, x_i, y_i) \geq 0 \\
& c_i^c = y_i^t - y_i = 0 \\
& \mathcal{R}_i\left(x_0, x_i, y_{j\neq i}^t, \bar{y}_i, y_i\right) = 0
\end{array}
$$

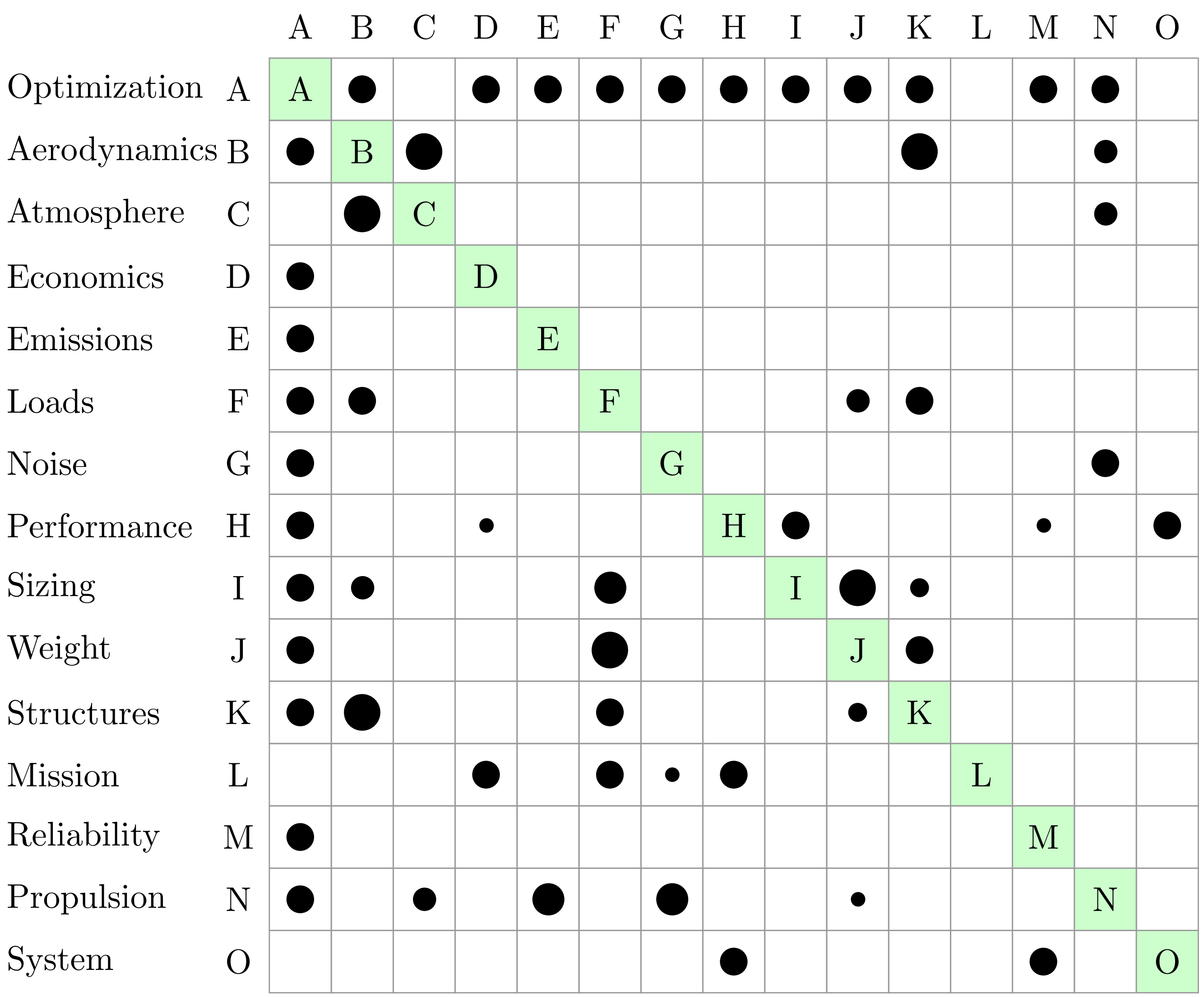
x	Design var.
y^t	Coupling inputs
y	Coupling outputs
$\bar{y}$	State var.
f	Objective
c	Design constraints
c^c	Consistency cons.
$\mathcal{R}$	Governing equations
$()_0$	Shared data
$()_i$	Discipline i data

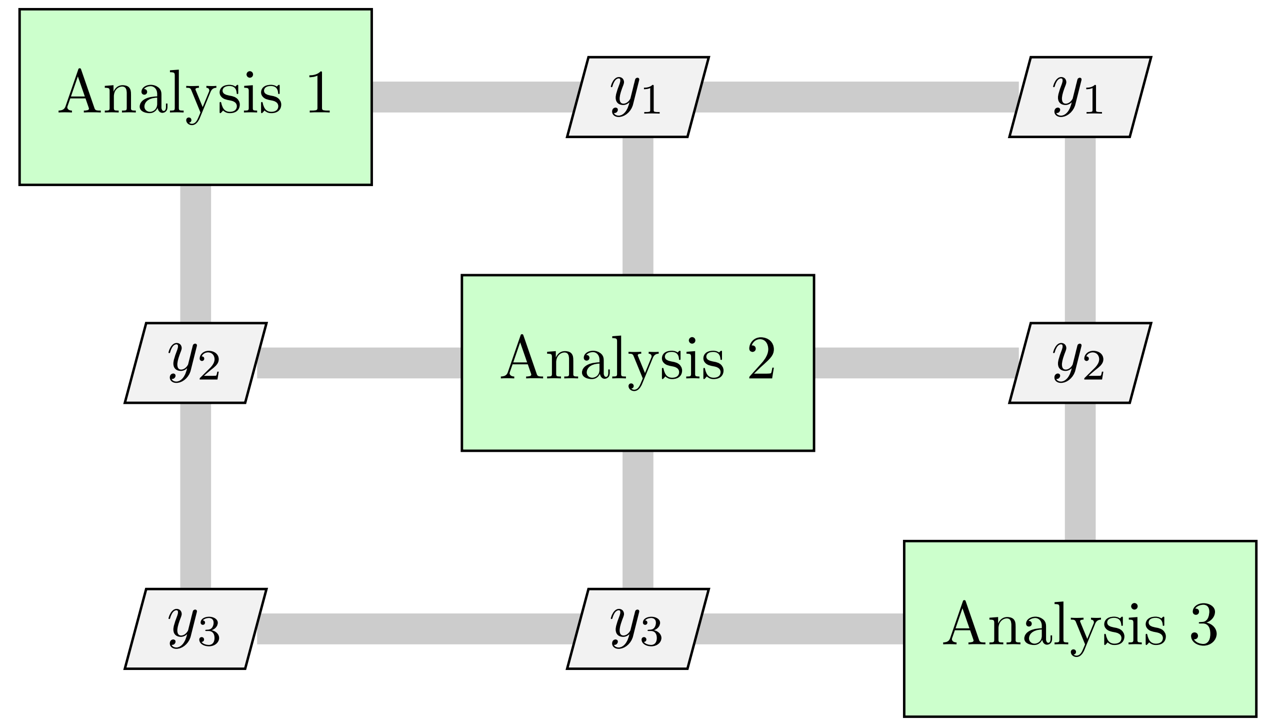
Convention: $x = [x_0^T, x_1^T, ..., x_N^T]^T$ and $y = [y_1^T, ..., y_N^T]^T$

Convention: c_i, c_i^c, and $\mathcal{R}_i$ exist for $i = 1, \ldots, N$

All architectures solve an equivalent reformulation of this problem

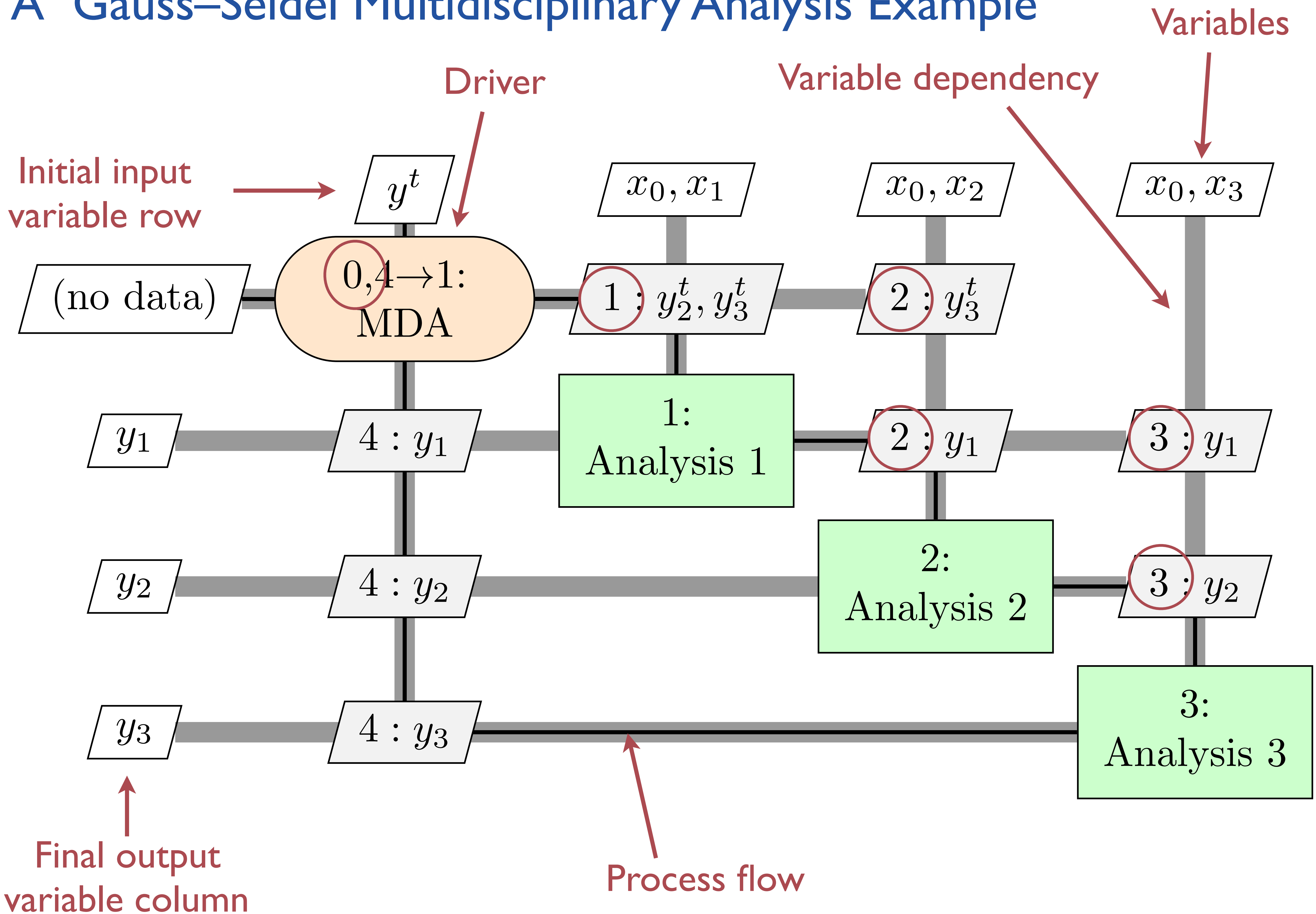
The N^2 Diagram and Design Structure Matrix



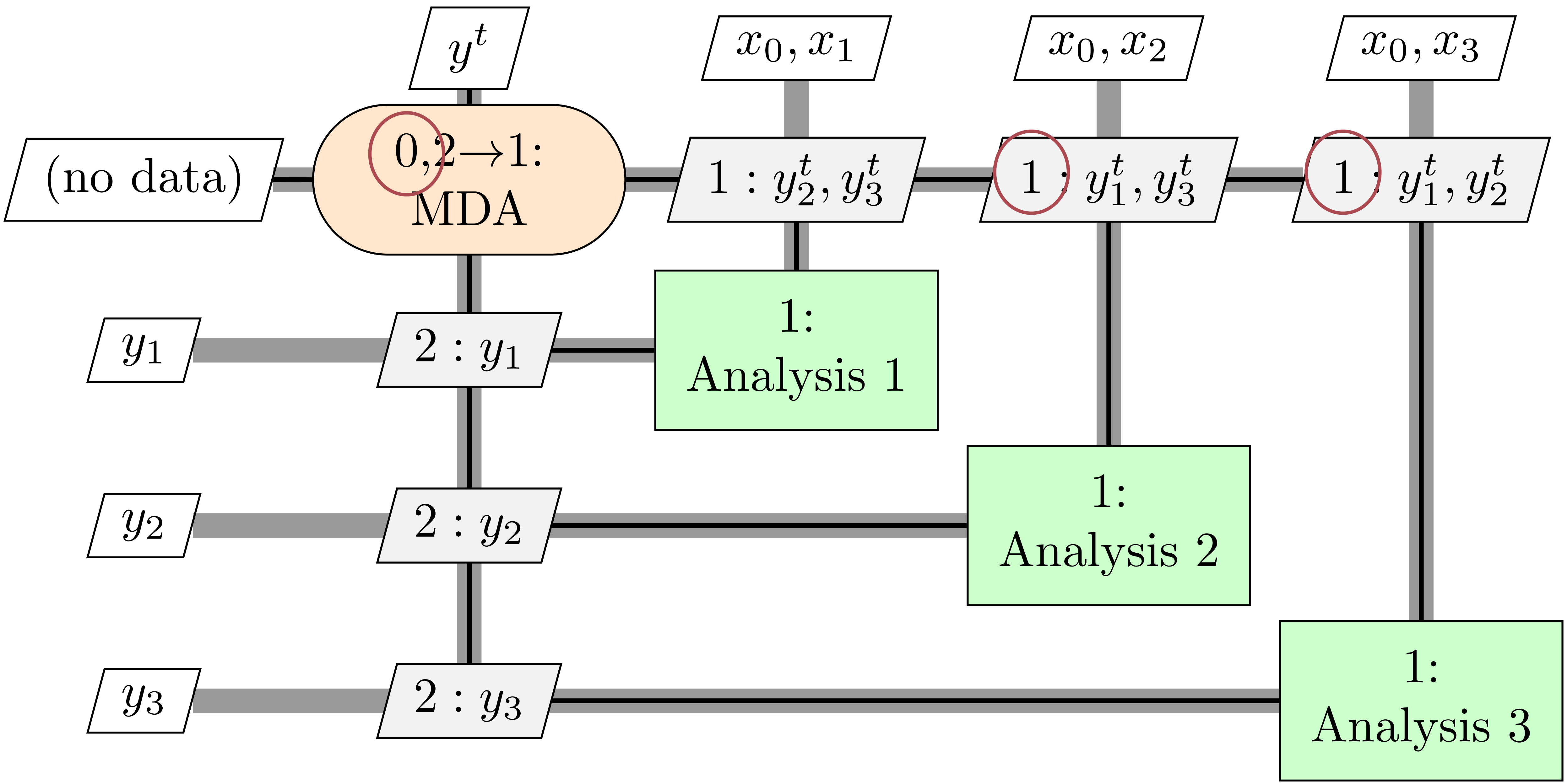


- Components on main diagonal, coupling data on off-diagonal nodes
- Component inputs in same column, component outputs in same row
- External inputs and outputs may also be included

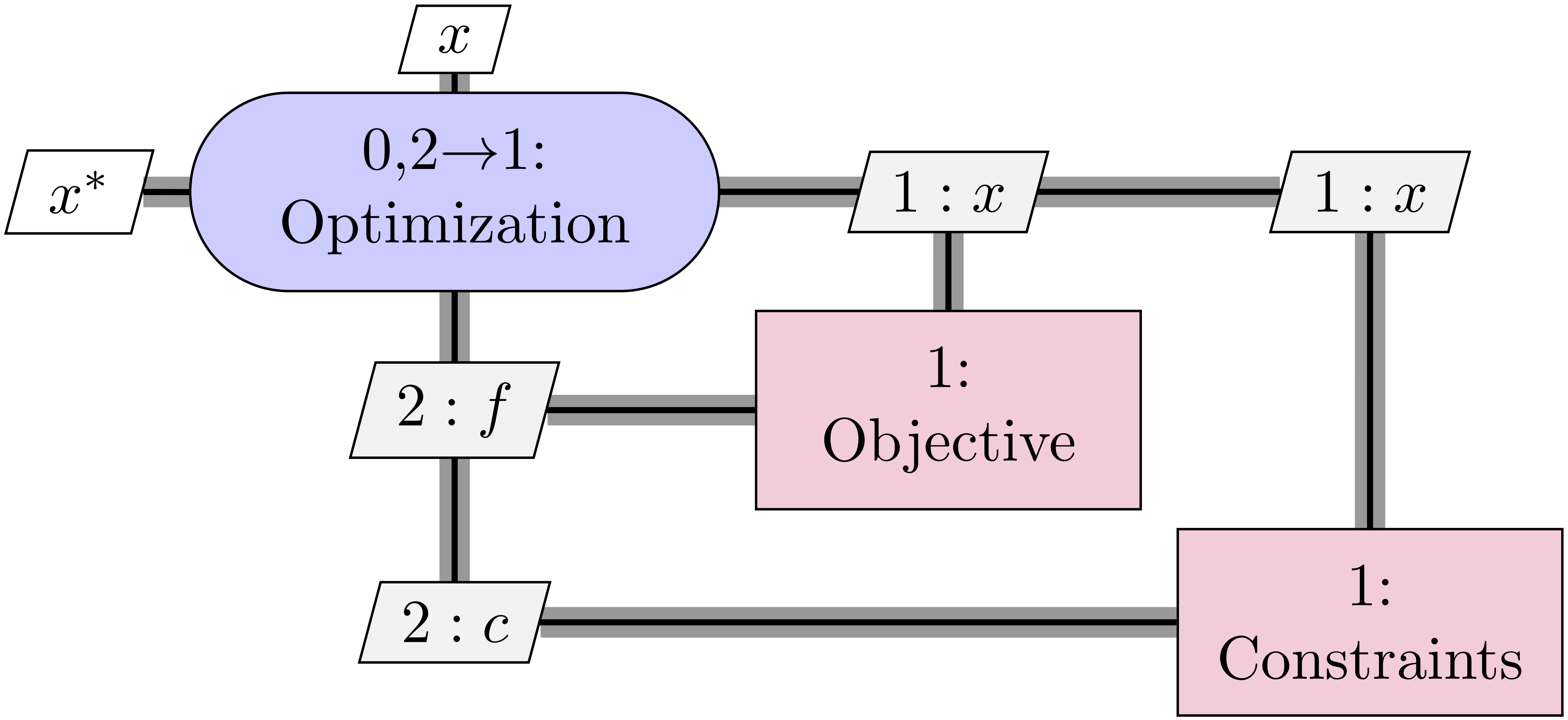
Extending the DSM syntax:
A Gauss–Seidel Multidisciplinary Analysis Example



A Jacobi Multidisciplinary Analysis Example



An Optimization Problem



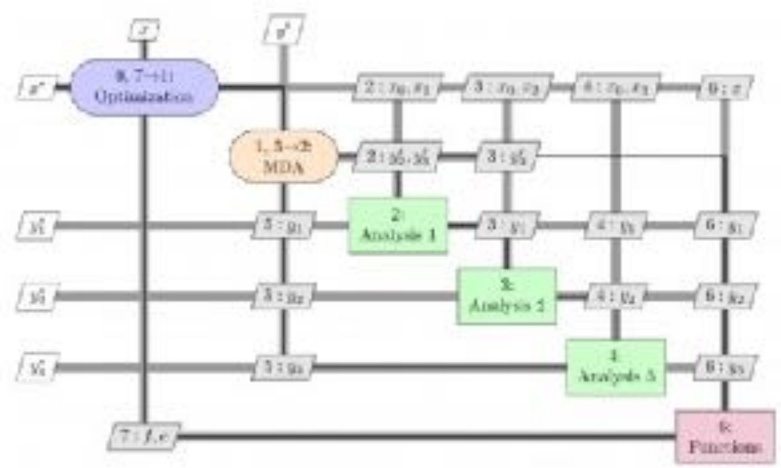
- Follow sequence of numbers and thin black lines
- When number or index is repeated, procedures can be parallelized
- Close the loops

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 subject 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

Sequential Optimization vs. MDO

Example: Aerostructural Optimization — Sequential Design vs. MDO 1

- One commonly used approach to design is to perform a sequential "optimization" approach, which consists in optimizing each discipline in sequence:
  1. For example, we could start by optimizing the aerodynamics,

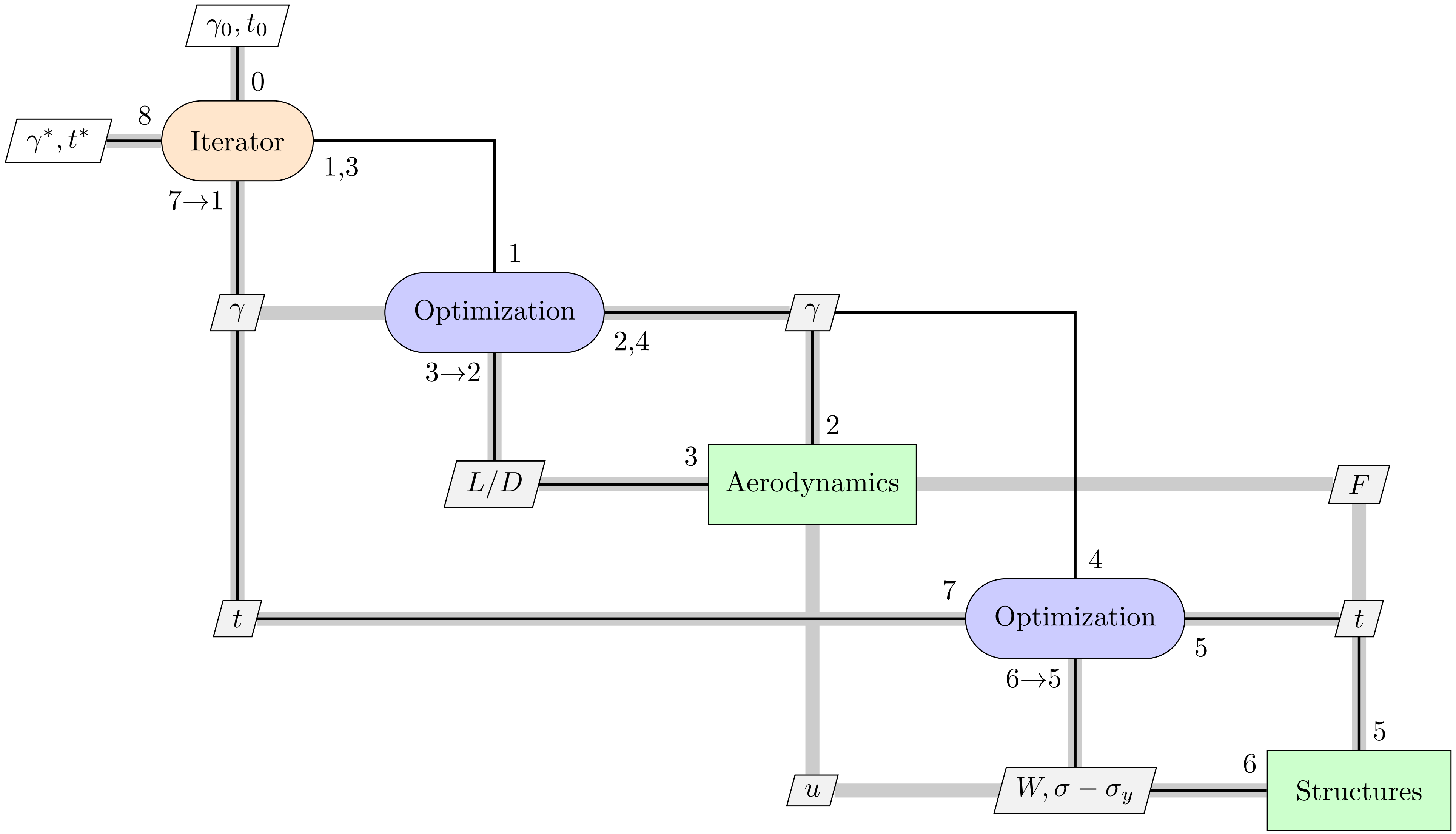
$$\begin{aligned} \text{minimize} \quad & D(\alpha, \gamma_i) \\ \text{w.r.t.} \quad & \alpha, \gamma_i \\ \text{s.t.} \quad & L(\alpha, \gamma_i) = W \end{aligned}$$

  2. Once the aerodynamic optimization has converged, the twist distribution and the forces are fixed
  3. Then we optimize the structure by minimizing weight subject to stress constraints at the maneuver condition, i.e.,

$$\begin{aligned} \text{minimize} \quad & W(t_i) \\ \text{w.r.t.} \quad & t_i \\ \text{s.t.} \quad & \sigma_j(t_i) \leq \sigma_{\text{yield}} \end{aligned}$$

Example: Aerostructural Optimization — Sequential Design vs. MDO 2

4. Repeat until this sequence has converged.

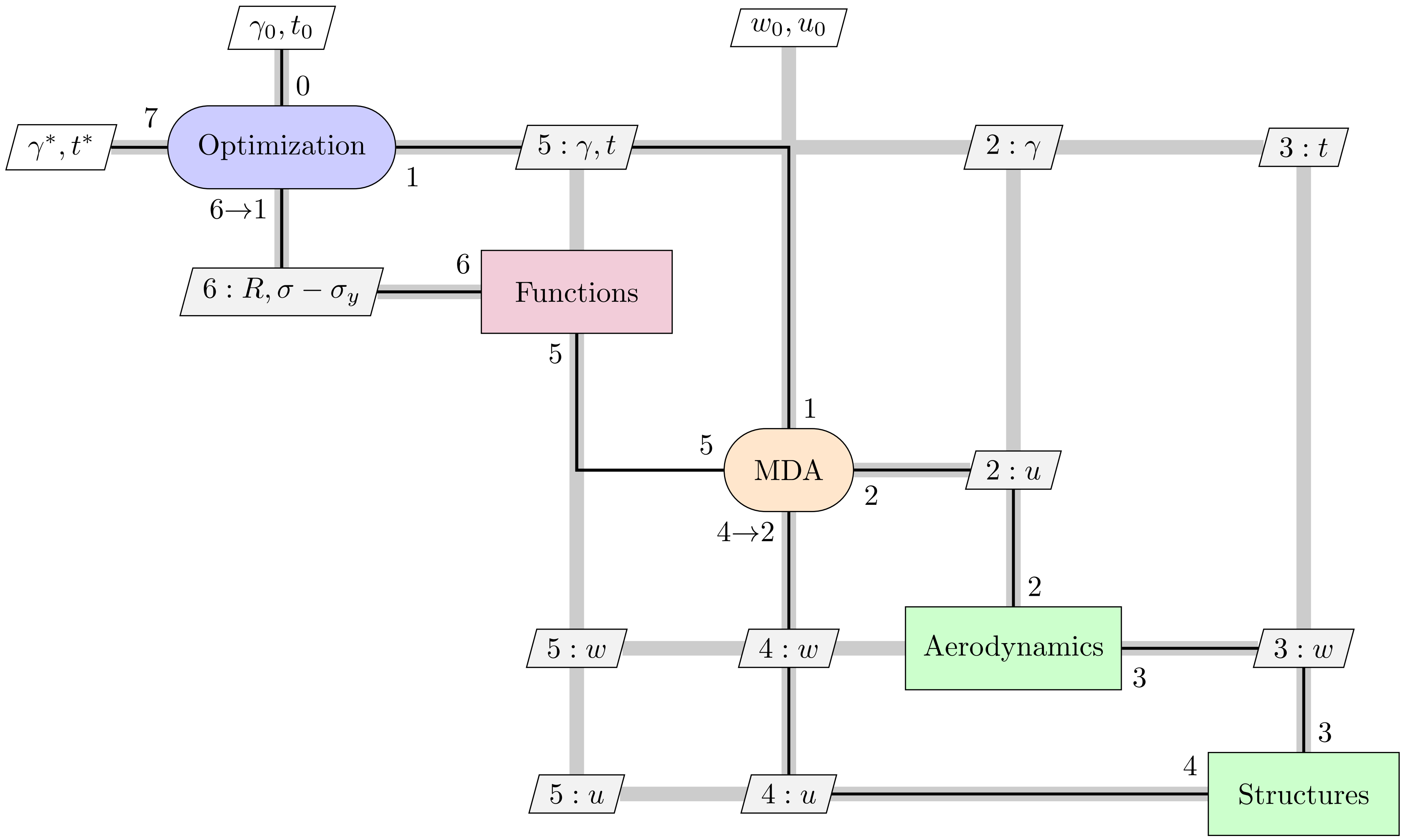


Example: Aerostructural Optimization — Sequential Design vs. MDO 3

- The MDO procedure differs from the sequential approach in that it considers all variables simultaneously

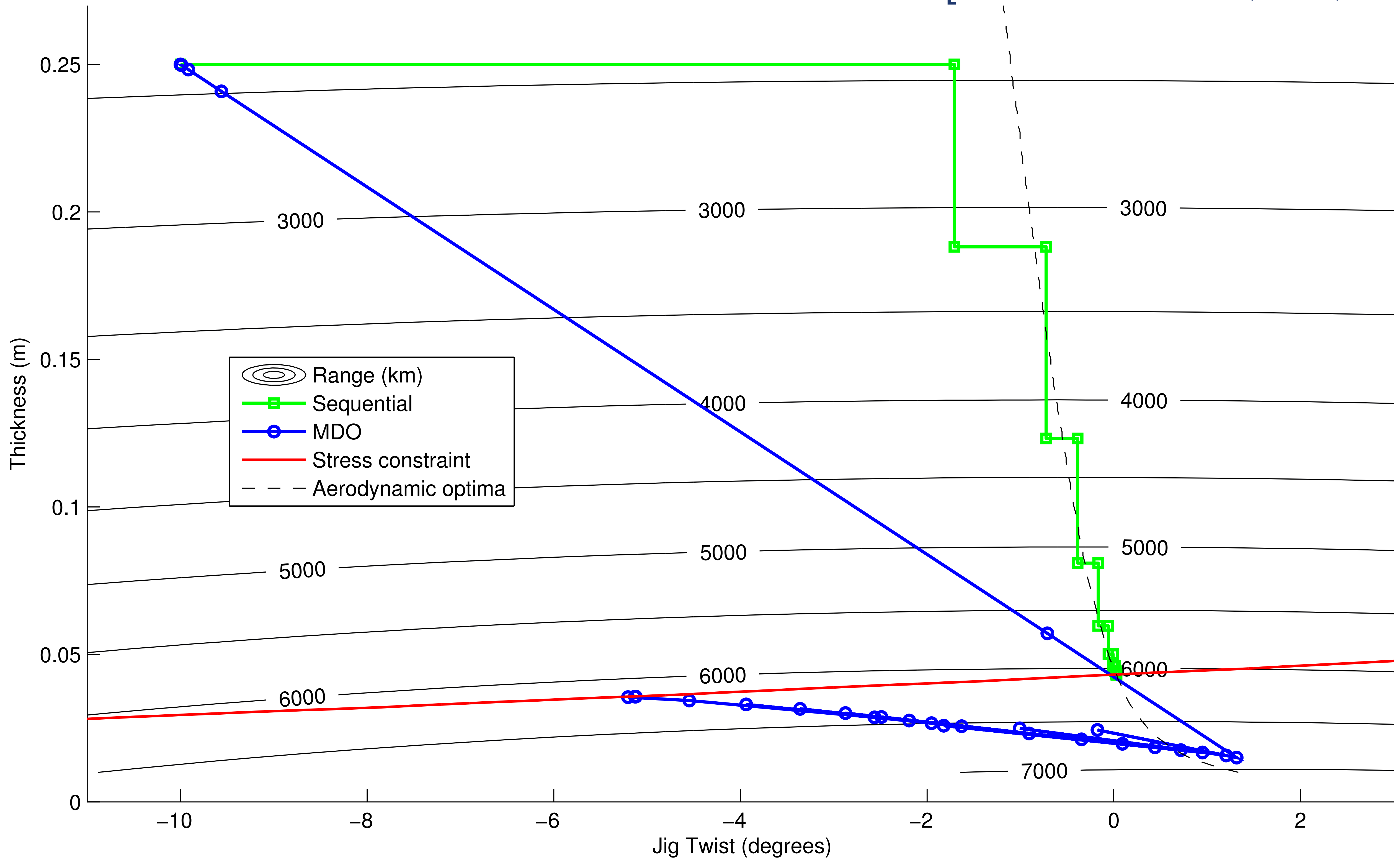
$$
\begin{aligned}
\text{minimize} \quad & \text{Range}\left(\alpha, \gamma_i, t_i\right) \\
\text{w.r.t.} \quad & \alpha, \gamma_i, t_i \\
\text{s.t.} \quad & \sigma_{\text{yield}} - \sigma_j\left(t_i\right) \geq 0 \\
& L\left(\alpha, \gamma_i\right) - W = 0
\end{aligned}
$$

Example: Aerostructural Optimization — Sequential Design vs. MDO 4



Example: Aerostructural Optimization — Sequential Design vs. MDO 5

[Chittick and Martins, SMO, 2008]



Monolithic MDO Architectures

Monolithic Architectures

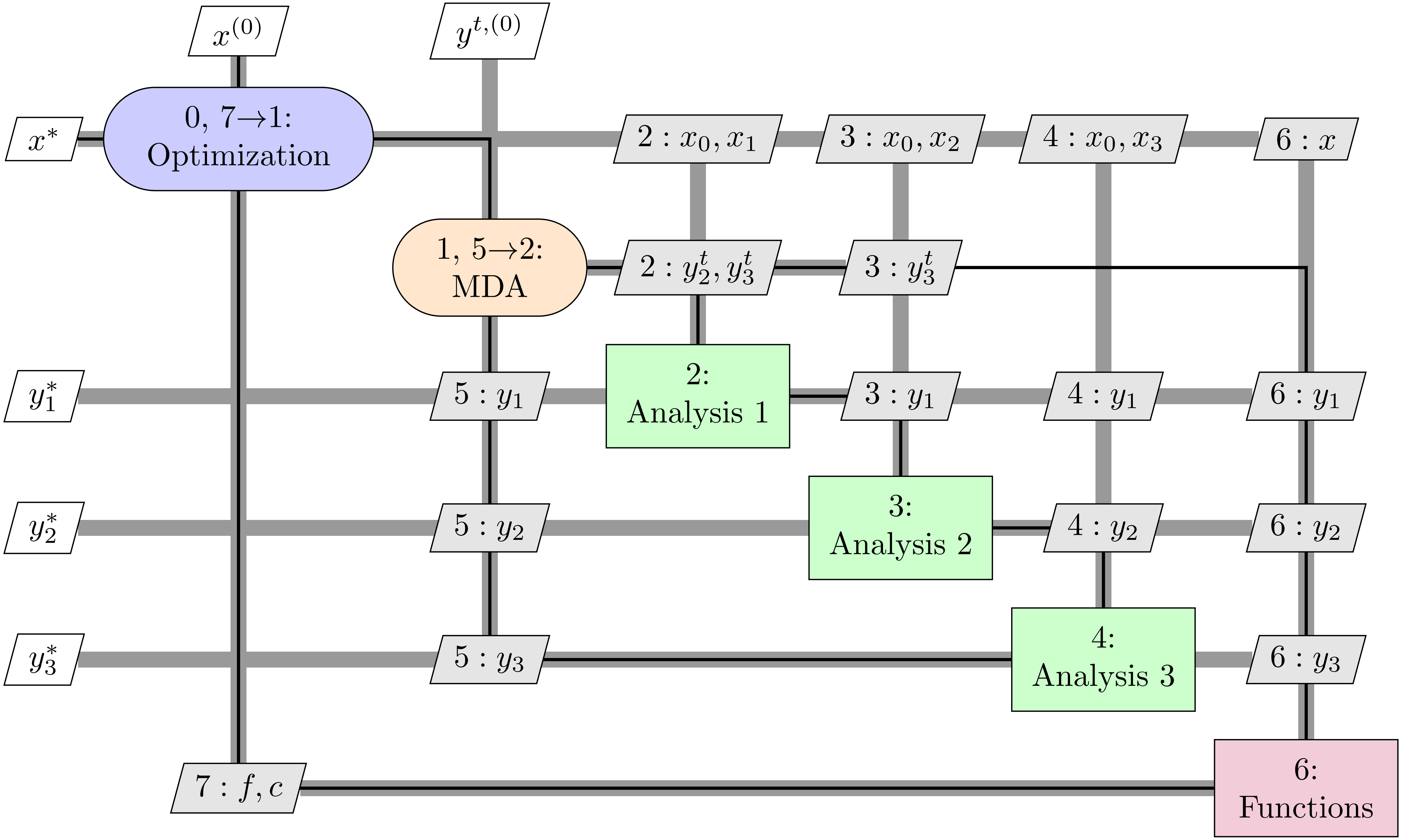
- Monolithic architectures solve the MDO problem by casting it as single optimization problem.
- Distributed architectures, on the other hand, decompose the overall problem into smaller ones.
- Monolithic architectures include:
  - Multidisciplinary Feasible — MDF
  - Individual Discipline Feasible — IDF
  - Simultaneous Analysis and Design — SAND
  - All-At-Once — AAO

Multidisciplinary Feasible (MDF) 1

- The MDF architecture is the most intuitive for engineers.
- The optimization problem formulation is identical to the single discipline case, except the disciplinary analysis is replace by an MDA

$$
\begin{aligned}
\text{minimize} \quad & f_0\left(x, y\left(x, y\right)\right) \\
\text{with respect to} \quad & x \\
\text{subject to} \quad & c_0\left(x, y\left(x, y\right)\right) \geq 0 \\
& c_i\left(x_0, x_i, y_i\left(x_0, x_i, y_{j\neq i}\right)\right) \geq 0 \quad \text{for} \quad i = 1, \ldots, N.
\end{aligned}
$$

Multidisciplinary Feasible (MDF) 2



Multidisciplinary Feasible (MDF) 3

- Advantages:
  - Optimization problem is as small as it can be for a monolithic architecture
  - Always returns a system design that satisfies the consistency constraints, even if the optimization process is terminated early — good from the practical engineering point of view
- Disadvantages:
  - Intermediate results do not necessarily satisfy the optimization constraints
  - Developing the MDA procedure might be time consuming, if not already in place
  - Gradients of the coupled system more challenging to compute (more in later section)

Example: Aerostructural Optimization with MDF

$$
\begin{aligned}
\text{minimize} \quad & -R \\
\text{w.r.t.} \quad & \Lambda, \gamma, t \\
\text{s.t.} \quad & \sigma_{\text{yield}} - \sigma_i(u) \geq 0
\end{aligned}
$$

where the aerostructural analysis is as before:

$$
\begin{aligned}
A\Gamma - v(u, \alpha) &= 0 \\
K(t, \Lambda)u - F(\Gamma) &= 0 \\
L(\Gamma) - W(t) &= 0
\end{aligned}
$$

Individual Discipline Feasible (IDF) 1

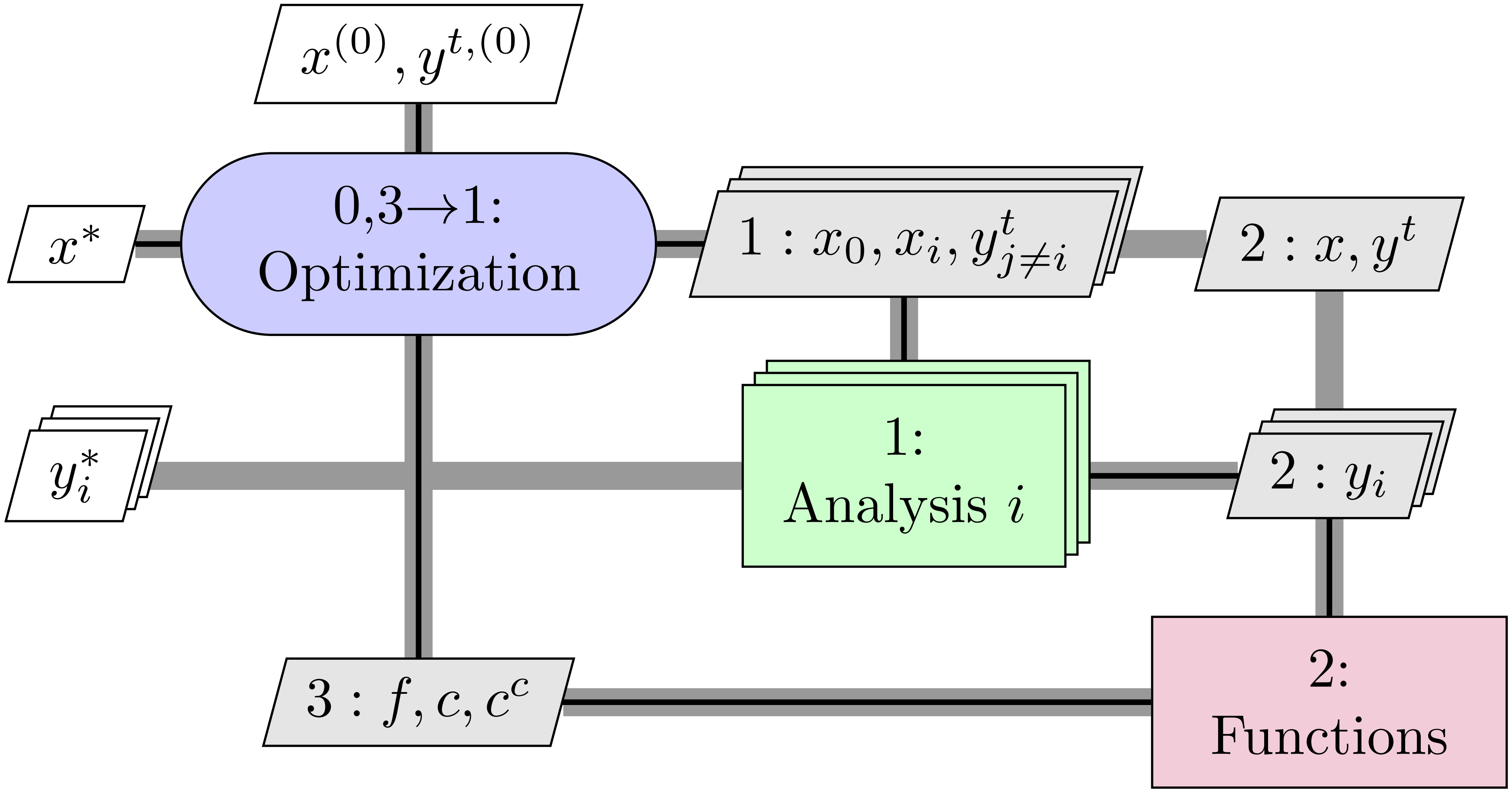
The IDF architecture decouples the MDA, adding consistency constraints, and giving the optimizer control of the coupling variables.

$$
\begin{aligned}
\text{minimize} \quad & f_0\left(x, y\left(x, y^t\right)\right) \\
\text{with respect to} \quad & x, y^t \\
\text{subject to} \quad & c_0\left(x, y\left(x, y^t\right)\right) \geq 0 \\
& c_i\left(x_0, x_i, y_i\left(x_0, x_i, y^t_{j\neq i}\right)\right) \geq 0 \quad \text{for} \quad i = 1, \ldots, N \\
& c^c_i = y^t_i - y_i\left(x_0, x_i, y^t_{j\neq i}\right) = 0 \quad \text{for} \quad i = 1, \ldots, N.
\end{aligned}
$$

- Advantages:
  - Optimizer typically converges the multidisciplinary feasibility better than fixed-point MDA iterations
- Disadvantages:
  - Problem is potentially much larger than MDF, depending on the number of coupling variables
  - Gradient computation can be costly

Individual Discipline Feasible (IDF) 2

- The large problem size can be mitigated to some extent by careful selection of the disciplinary variable partitions or aggregation of the coupling variables to reduce information transfer between disciplines.



Example: Aerostructural Optimization Using IDF

$$
\begin{aligned}
\text{minimize} \quad & -R \\
\text{w.r.t.} \quad & \Lambda, \gamma, t, \Gamma^t, \alpha^t, u^t \\
\text{s.t.} \quad & \sigma_{\text{yield}} - \sigma_i \geq 0 \\
& \Gamma^t - \Gamma = 0 \\
& \alpha^t - \alpha = 0 \\
& u^t - u = 0
\end{aligned}
$$

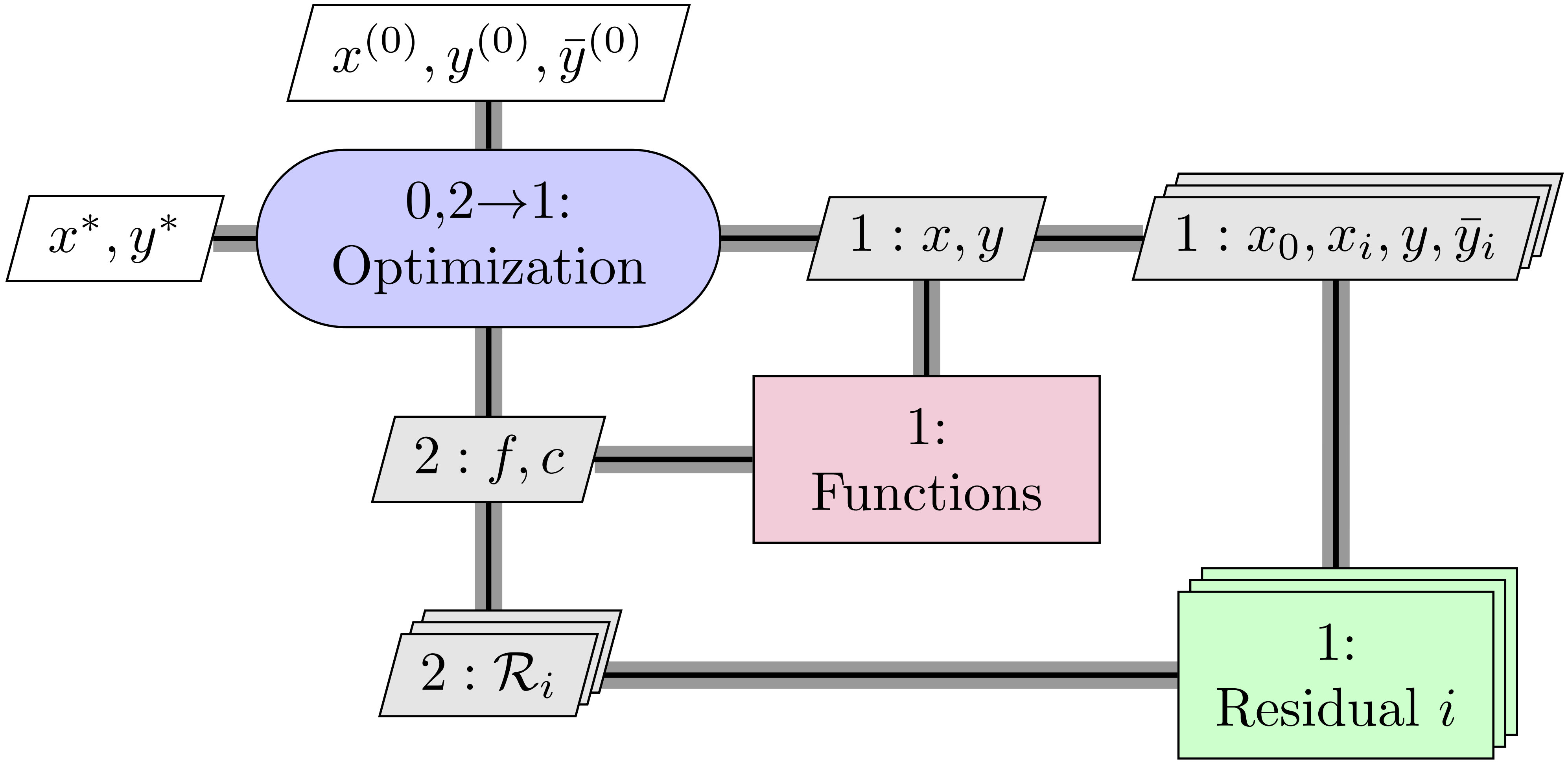
Simultaneous Analysis and Design (SAND) 1

- SAND makes no distinction between disciplines, and can also be applied to single discipline problems.
- The governing equations are constraints at the optimizer level.

$$
\begin{aligned}
\text{minimize} \quad & f_0(x, y) \\
\text{with respect to} \quad & x, y, \bar{y} \\
\text{subject to} \quad & c_0(x, y) \geq 0 \\
& c_i(x_0, x_i, y_i) \geq 0 \qquad \text{for} \quad i = 1, \ldots, N \\
& \mathcal{R}_i(x_0, x_i, y, \bar{y}_i) = 0 \quad \text{for} \quad i = 1, \ldots, N.
\end{aligned}
$$

- Advantages:
  - If implemented well, can be the most efficient architecture
- Disadvantages:
  - Intermediate results do not even satisfy the governing equations
  - Difficult or impossible to implement for “black-box” components

Simultaneous Analysis and Design (SAND) 2



Aerostructural Optimization Using SAND 1

$$
\begin{aligned}
\text{minimize} \quad & -R \\
\text{w.r.t.} \quad & \Lambda, \gamma, t, \Gamma, \alpha, u \\
\text{s.t.} \quad & \sigma_{\text{yield}} - \sigma_i(u) \geq 0 \\
& A\Gamma = v(u, \alpha) \\
& K(t)u = f(\Gamma) \\
& L(\Gamma) - W(t) = 0
\end{aligned}
$$

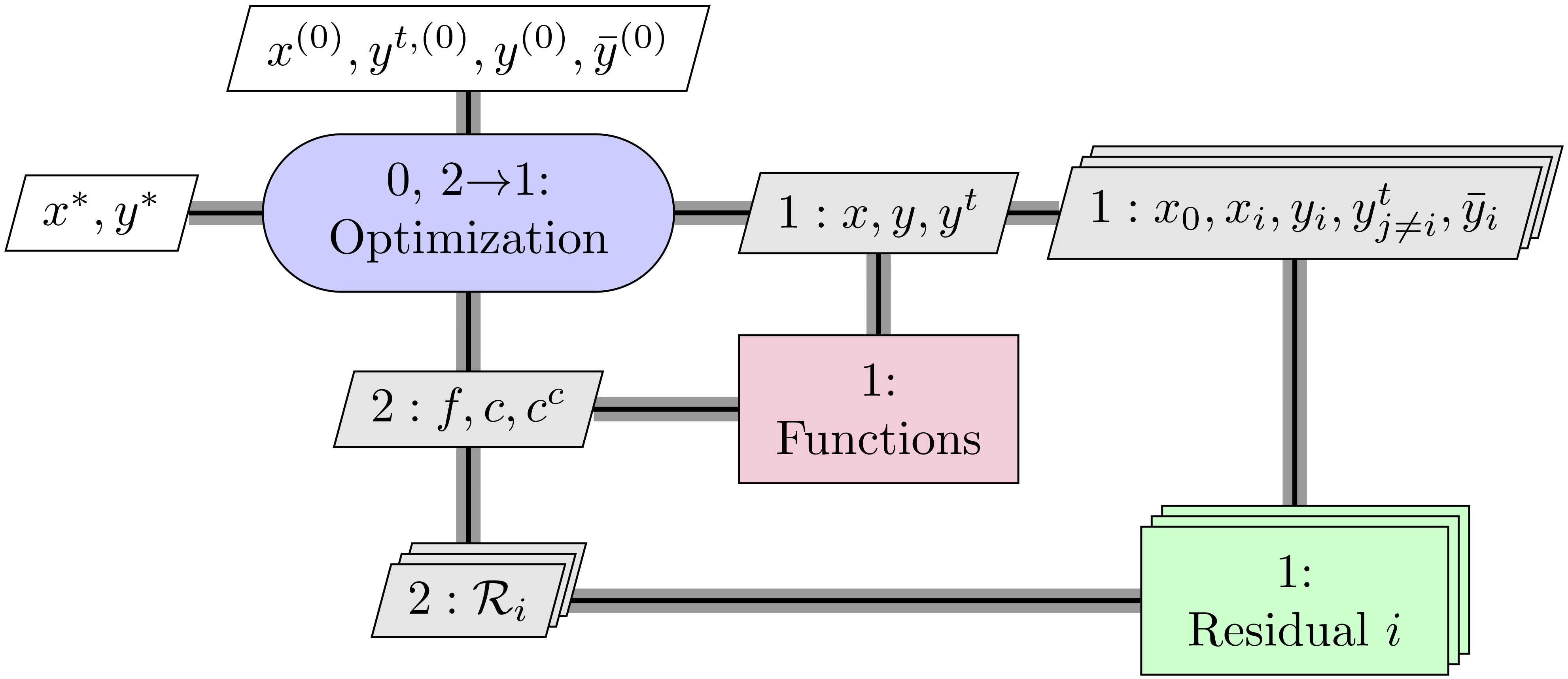
The All-at-Once (AAO) Problem Statement 1

- AAO is not strictly an architecture, as it is not practical to solve a problem of this form: the consistency constraints are linear and can be eliminated, leading to SAND.
- Some inconsistency in the name, in the literature
- We present AAO for completeness, and to relate this to the other monolithic architectures.

$$
\begin{aligned}
\text{minimize} \quad & f_0(x, y) + \sum_{i=1}^{N} f_i(x_0, x_i, y_i) \\
\text{with respect to} \quad & x, y^t, y, \bar{y} \\
\text{subject to} \quad & c_0(x, y) \geq 0 \\
& c_i(x_0, x_i, y_i) \geq 0 && \text{for} \quad i = 1, \ldots, N \\
& c_i^c = y_i^t - y_i = 0 && \text{for} \quad i = 1, \ldots, N \\
& \mathcal{R}_i\left(x_0, x_i, y_{j\neq i}^t, \bar{y}_i, y_i\right) = 0 && \text{for} \quad i = 1, \ldots, N.
\end{aligned}
$$

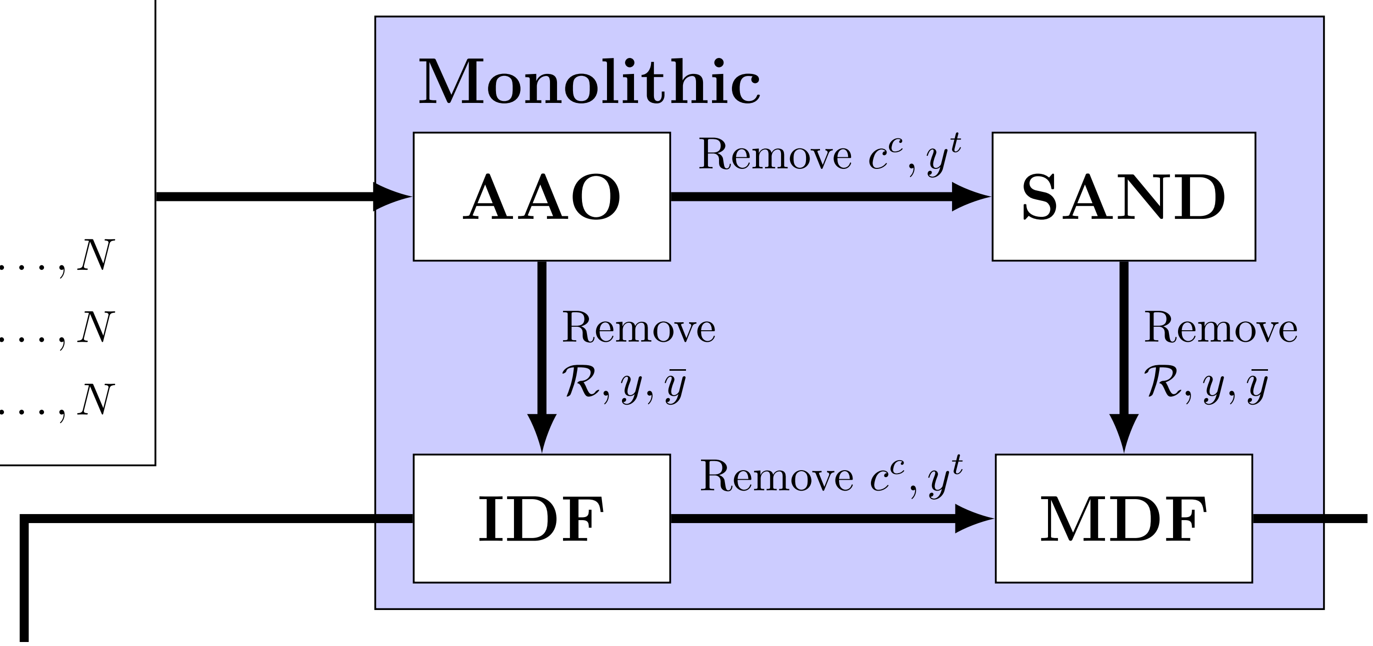
The All-at-Once (AAO) Problem Statement 2

- As we can see, it includes all the constraints that other monolithic architectures eliminated.



The All-at-Once (AAO) Problem Statement 3

$$
\begin{aligned}
\text{minimize} \quad & f_0\left(x, y\right) + \sum_{i=1}^{N} f_i\left(x_0, x_i, y_i\right) & & \\
\text{with respect to} \quad & x, y^t, y, \bar{y} & & \\
\text{subject to} \quad & c_0\left(x, y\right) \geq 0 & & \\
& c_i\left(x_0, x_i, y_i\right) \geq 0 & \text{for} \quad & i = 1, \ldots, N \\
& c_i^c = y_i^t - y_i = 0 & \text{for} \quad & i = 1, \ldots, N \\
& \mathcal{R}_i\left(x_0, x_i, y_{j\neq i}^t, \bar{y}_i, y_i\right) = 0 & \text{for} \quad & i = 1, \ldots, N
\end{aligned}
$$



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

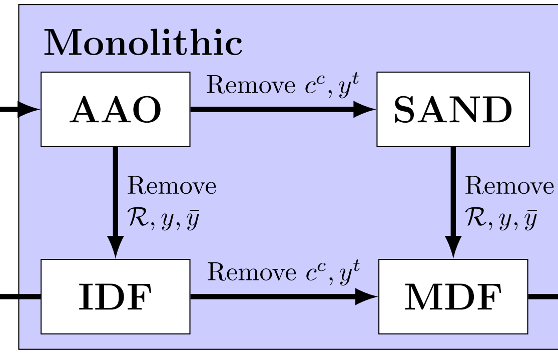
Distributed MDO Architectures

Distributed Architectures

- Monolithic MDO architectures solve a single optimization problem
- Distributed MDO architectures decompose the original problem into multiple optimization problems
- Some problems have a special structure and can be efficiently decomposed, but that is usually not the case
- In reality, the primary motivation for decomposing the MDO problem comes from the structure of the engineering design environment
- Typical industrial practice involves breaking up the design of a large system and distributing aspects of that design to specific engineering groups.
- These groups may be geographically distributed and may only communicate infrequently.
- In addition, these groups typically like to retain control of their own design procedures and make use of in-house expertise

Classification of MDO Architectures

$$
\begin{aligned}
\text{minimize} \quad & f_0(x,y) + \sum_{i=1}^{N} f_i(x_0, x_i, y_i) \\
\text{with respect to} \quad & x, y^t, y, \bar{y} \\
\text{subject to} \quad & c_0(x,y) \geq 0 \\
& c_i(x_0, x_i, y_i) \geq 0 \qquad \text{for} \quad i = 1, \ldots, N \\
& c_i^c = y_i^t - y_i = 0 \qquad \text{for} \quad i = 1, \ldots, N \\
& \mathcal{R}_i\left(x_0, x_i, y_{j\neq i}^t, \bar{y}_i, y_i\right) = 0 \qquad \text{for} \quad i = 1, \ldots, N
\end{aligned}
$$



Distributed IDF

Multilevel

CO: Copies of the shared variables are created for each discipline, together with corresponding consistency constraints. Discipline subproblems minimize difference between the copies of shared and local variables subject to local constraints. System subproblem minimizes objective subject to shared constraints subject to consistency constraints.

BLISS-2000: Discipline subproblems minimize the objective with respect to local variables subject to local constraints. A surrogate model of the local optima with respect to the shared variables is maintained. Then, system subproblem minimizes objective with respect to shared design and coupling variables subject to shared design and consistency constraints, considering the disciplinary preferences.

QSD: Each discipline is assigned a "budget" for a local objective and the discipline problems maximize the margin in their local constraints and the budgeted objective. System subproblem minimizes a shared objective and the budgets of each discipline subject to shared design constraints and positivity of the margin in each discipline.

Penalty

ATC: Copies of the shared variables are used in discipline subproblems together with the corresponding consistency constraints. These consistency constraints are relaxed using a penalty function. System and discipline subproblems solve their respective relaxed problem independently. Penalty weights are increased until the desired consistency is achieved.

IPD/EPD: Applicable to MDO problems with no shared objectives or constraints. Like ATC, copies of shared variables are used for every discipline subproblem and the consistency constraints are relaxed with a penalty function. Unlike ATC, the simple structure of the disciplinary subproblems is exploited to compute post-optimality sensitivities to guide the system subproblem.

ECO: As in CO, copies of the shared design variables are used. Disciplinary subproblems minimize quadratic approximations of the objective subject to local constraints and linear models of nonlocal constraints. Shared variables are determined by the system subproblem, which minimizes the total violation of all consistency constraints.

Distributed MDF

CSSO: In system subproblem, disciplinary analyses are replaced by surrogate models. Discipline subproblems are solved using surrogates for the other disciplines, and the solutions from these discipline subproblems are used to update the surrogate models.

BLISS: Coupled derivatives of the multidisciplinary analysis are used to construct linear subproblems for each discipline with respect to local design variables. Post-optimality derivatives from the solutions of these subproblems are computed to form the system linear subproblem, which is solved with respect to shared design variables.

MDOIS: Applicable to MDO problems with no shared objectives, constraints, or design variables. Discipline subproblems are solved independently assuming fixed coupling variables, and then a multidisciplinary analysis is performed to update the coupling.

ASO: System subproblem is like that of MDF, but some disciplines solve a discipline optimization subproblem within the multidisciplinary analysis with respect to local variables subject to local constraints. Coupled post-optimality derivatives from the discipline subproblems are computed to guide the system subproblem.

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

Concurrent Subspace Optimization (CSSO) 1

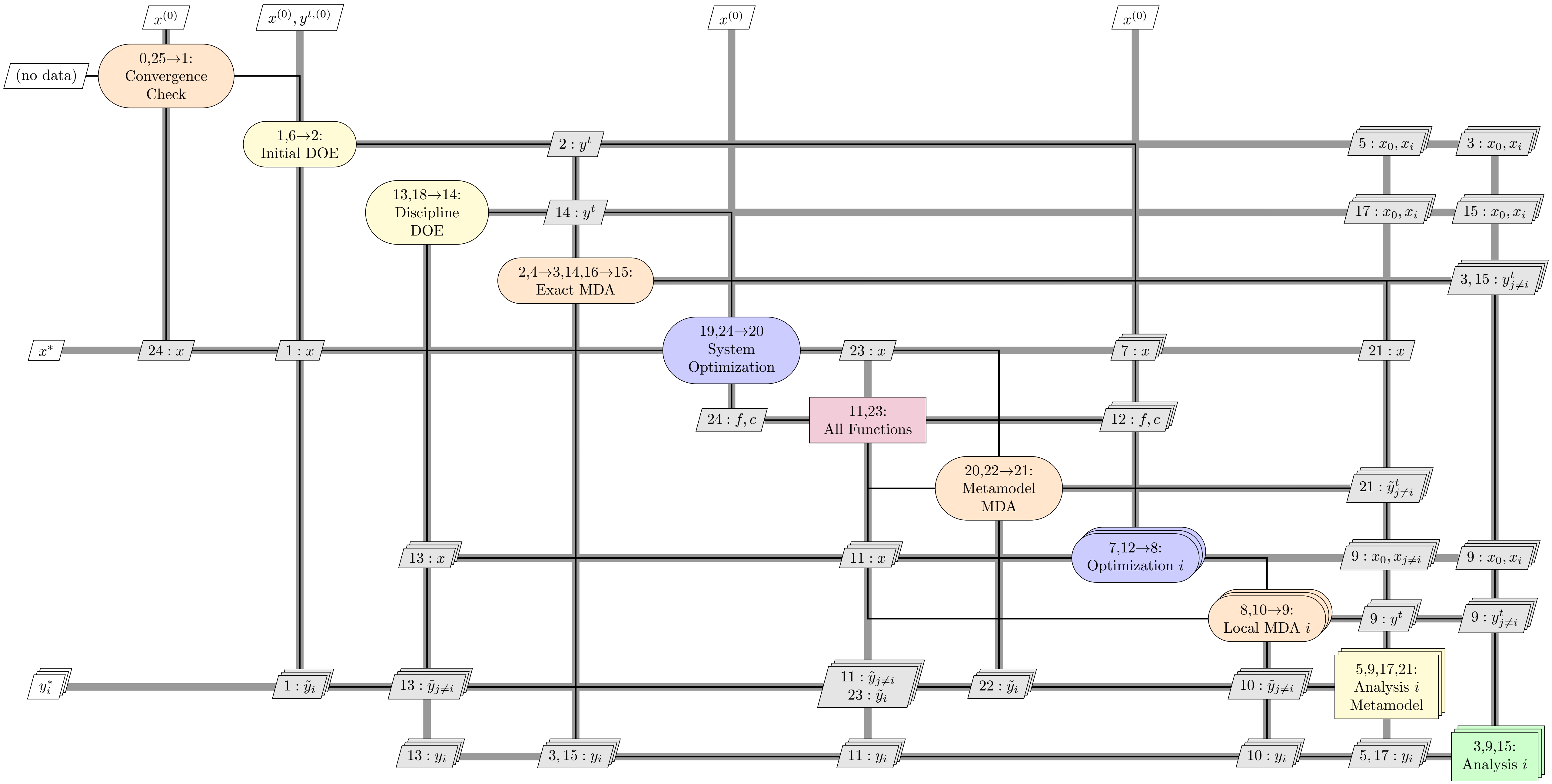
The CSSO system subproblem is given by

$$
\begin{aligned}
\text{minimize} \quad & f_0\left(x, \tilde{y}\left(x, \tilde{y}\right)\right) \\
\text{with respect to} \quad & x \\
\text{subject to} \quad & c_0\left(x, \tilde{y}\left(x, \tilde{y}\right)\right) \geq 0 \\
& c_i\left(x_0, x_i, \tilde{y}_i\left(x_0, x_i, \tilde{y}_{j\neq i}\right)\right) \geq 0 \text{ for} \quad i = 1, \ldots, N
\end{aligned}
$$

and the discipline i subproblem is given by

$$
\begin{aligned}
\text{minimize} \quad & f_0\left(x, y_i\left(x_i, \tilde{y}_{j\neq i}\right), \tilde{y}_{j\neq i}\right) \\
\text{with respect to} \quad & x_0, x_i \\
\text{subject to} \quad & c_0\left(x, \tilde{y}\left(x, \tilde{y}\right)\right) \geq 0 \\
& c_i\left(x_0, x_i, y_i\left(x_0, x_i, \tilde{y}_{j\neq i}\right)\right) \geq 0 \\
& c_j\left(x_0, \tilde{y}_j\left(x_0, \tilde{y}\right)\right) \geq 0 \quad \text{for} \quad j = 1, \ldots, N \; j \neq i.
\end{aligned}
$$

Concurrent Subspace Optimization (CSSO) 2



CSSO Algorithm

Input: Initial design variables x
Output: Optimal variables x^*, objective function f^*, and constraint values c^*

0: Initiate main CSSO iteration
repeat
  1: Initiate a design of experiments (DOE) to generate design points
  for Each DOE point **do**
    2: Initiate an MDA that uses exact disciplinary information
    repeat
      3: Evaluate discipline analyses
      4: Update coupling variables y
    until $4 \to 3$: MDA has converged
    5: Update the disciplinary surrogate models with the latest design
  end for $6 \to 2$
  7: Initiate independent disciplinary optimizations (in parallel)
  for Each discipline i **do**
    repeat
      8: Initiate an MDA with exact coupling variables for discipline i and approximate coupling variables for the other disciplines
      repeat
        9: Evaluate discipline i outputs y_i, and surrogate models for the other disciplines, $\tilde{y}_{j\neq i}$
      until $10 \to 9$: MDA has converged
      11: Compute objective f_0 and constraint functions c using current data
    until $12 \to 8$: Disciplinary optimization i has converged
  end for
  13: Initiate a DOE that uses the subproblem solutions as sample points
  for Each subproblem solution i **do**
    14: Initiate an MDA that uses exact disciplinary information
    repeat
      15: Evaluate discipline analyses.
    until $16 \to 15$ MDA has converged
    17: Update the disciplinary surrogate models with the newest design
  end for $18 \to 14$
  19: Initiate system-level optimization
  repeat
    20: Initiate an MDA that uses only surrogate model information
    repeat
      21: Evaluate disciplinary surrogate models
    until $22 \to 21$: MDA has converged
    23: Compute objective f_0, and constraint function values c
  until $24 \to 20$: System level problem has converged
until $25 \to 1$: CSSO has converged

Collaborative Optimization (CO) 1

The CO_2 system subproblem is given by:

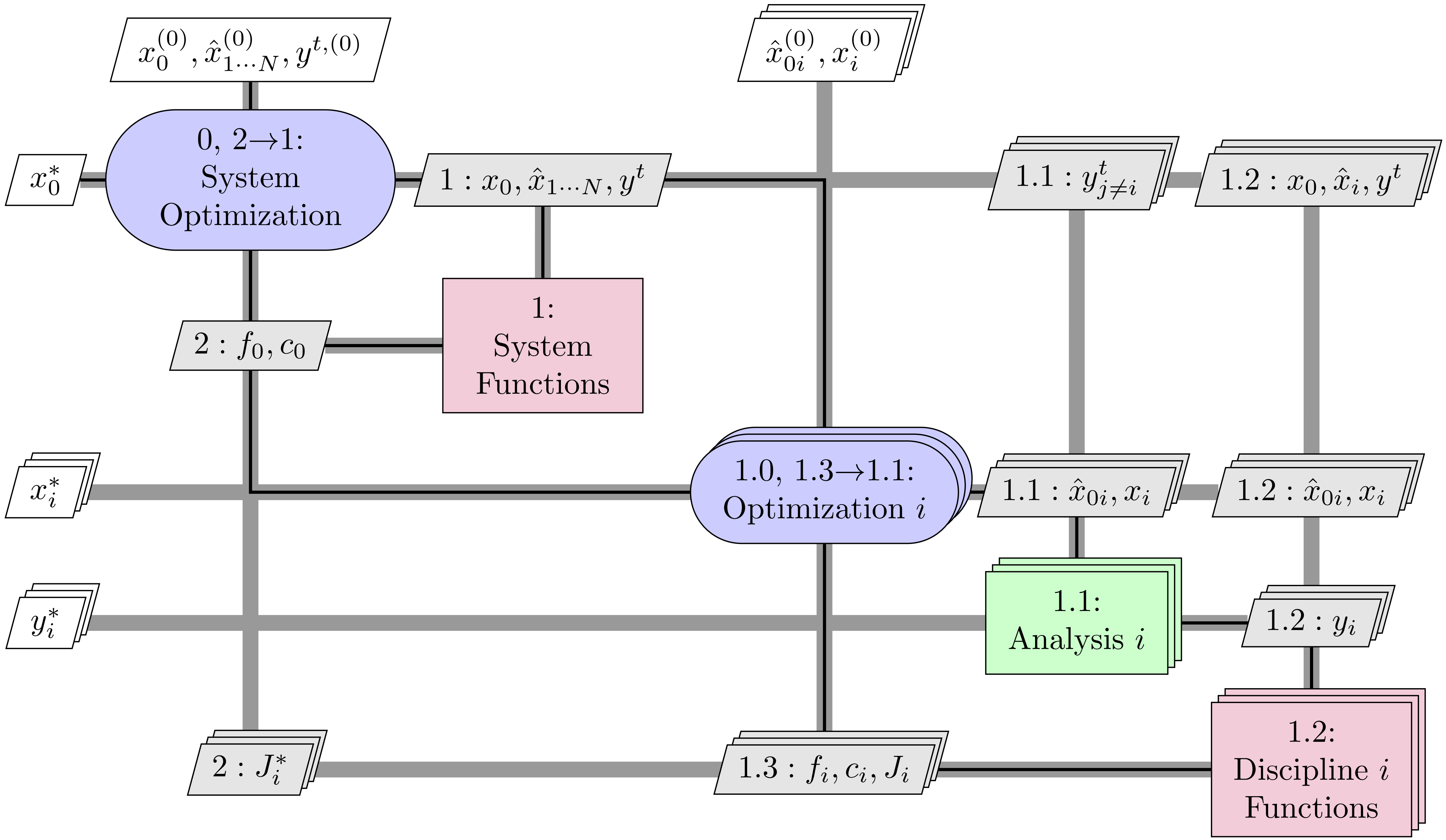
$$
\begin{aligned}
\text{minimize} \quad & f_0\left(x_0, \hat{x}_1, \ldots, \hat{x}_N, y^t\right) \\
\text{with respect to} \quad & x_0, \hat{x}_1, \ldots, \hat{x}_N, y^t \\
\text{subject to} \quad & c_0\left(x_0, \hat{x}_1, \ldots, \hat{x}_N, y^t\right) \geq 0 \\
& J_i^* = ||\hat{x}_{0i} - x_0||_2^2 + ||\hat{x}_i - x_i||_2^2 + \\
& \qquad ||y_i^t - y_i\left(\hat{x}_{0i}, x_i, y_{j \neq i}^t\right)||_2^2 = 0 \quad \text{for} \quad i = 1, \ldots, N
\end{aligned}
$$

where $\hat{x}_{0i}$ are duplicates of the global design variables passed to (and manipulated by) discipline i and $\hat{x}_i$ are duplicates of the local design variables passed to the system subproblem.

The discipline i subproblem in both CO_1 and CO_2 is

$$
\begin{aligned}
\text{minimize} \quad & J_i\left(\hat{x}_{0i}, x_i, y_i\left(\hat{x}_{0i}, x_i, y_{j \neq i}^t\right)\right) \\
\text{with respect to} \quad & \hat{x}_{0i}, x_i \\
\text{subject to} \quad & c_i\left(\hat{x}_{0i}, x_i, y_i\left(\hat{x}_{0i}, x_i, y_{j \neq i}^t\right)\right) \geq 0.
\end{aligned}
$$

Collaborative Optimization (CO) 2



CO Algorithm 1

Input: Initial design variables x
Output: Optimal variables x^*, objective function f^*, and constraint values c^*

0: Initiate system optimization iteration
repeat
    1: Compute system subproblem objectives and constraints
    for Each discipline i (in parallel) **do**
        1.0: Initiate disciplinary subproblem optimization
        repeat
            1.1: Evaluate disciplinary analysis
            1.2: Compute disciplinary subproblem objective and constraints
            1.3: Compute new disciplinary subproblem design point and J_i
        until $1.3 \to 1.1$: Optimization i has converged
    end for
    2: Compute a new system subproblem design point
until $2 \to 1$: System optimization has converged

Aerostructural Optimization Using CO 1

System-level problem:

$$\begin{aligned} \text{minimize} \quad & -R \\ \text{w.r.t.} \quad & \Lambda^t, \Gamma^t, \alpha^t, u^t, W^t \\ \text{s.t.} \quad & J_1^* \leq 10^{-6} \\ & J_2^* \leq 10^{-6} \end{aligned}$$

Aerodynamics subproblem:

$$\begin{aligned} \text{minimize} \quad & J_1 = \left(1 - \frac{\Lambda}{\Lambda^t}\right)^2 + \sum \left(1 - \frac{\Gamma_i}{\Gamma_i^t}\right)^2 + \left(1 - \frac{\alpha}{\alpha^t}\right)^2 + \left(1 - \frac{W}{W^t}\right)^2 \\ \text{w.r.t.} \quad & \Lambda, \gamma, \alpha \\ \text{s.t.} \quad & L - W = 0 \end{aligned}$$

Aerostructural Optimization Using CO 2

Structures subproblem:

$$
\begin{aligned}
\text{minimize} \quad & J_2 = \left(1 - \frac{\Lambda}{\Lambda^t}\right)^2 + \sum \left(1 - \frac{u_i}{u_i^t}\right)^2 \\
\text{w.r.t.} \quad & \Lambda, t \\
\text{s.t.} \quad & \sigma_{\text{yield}} - \sigma_i \geq 0
\end{aligned}
$$

Bilevel Integrated System Synthesis (BLISS) 1

The system level subproblem is formulated as

$$
\begin{aligned}
\text{minimize} \quad & (f_0^*)_0 + \left(\frac{\mathrm{d}f_0^*}{\mathrm{d}x_0}\right)\Delta x_0 \\
\text{with respect to} \quad & \Delta x_0 \\
\text{subject to} \quad & (c_0^*)_0 + \left(\frac{\mathrm{d}c_0^*}{\mathrm{d}x_0}\right)\Delta x_0 \geq 0 \\
& (c_i^*)_0 + \left(\frac{\mathrm{d}c_i^*}{\mathrm{d}x_0}\right)\Delta x_0 \geq 0 \quad \text{for} \quad i = 1, \ldots, N \\
& \Delta x_{0L} \leq \Delta x_0 \leq \Delta x_{0U}.
\end{aligned}
$$

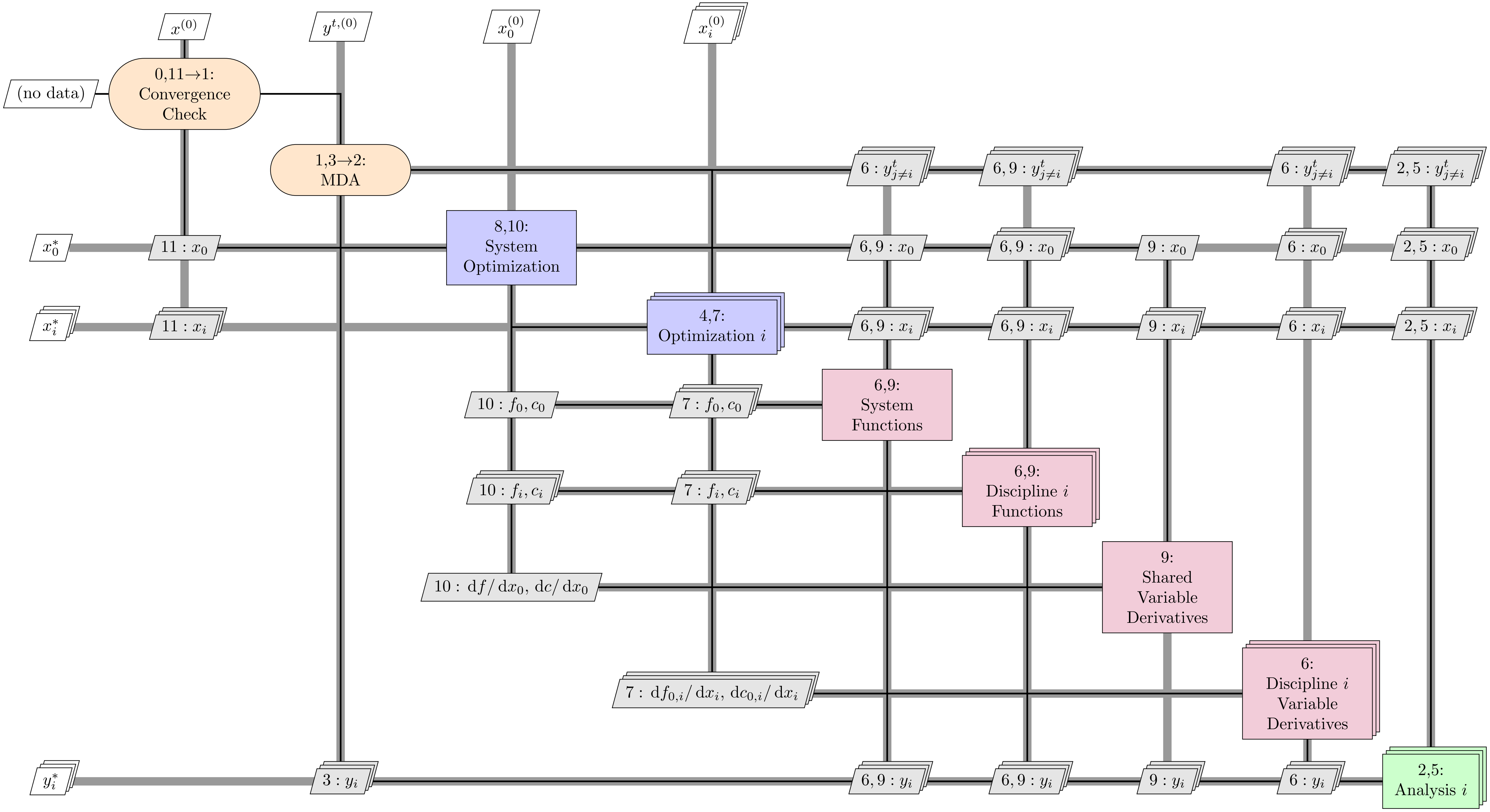
Bilevel Integrated System Synthesis (BLISS) 2

The discipline i subproblem is given by

$$
\begin{aligned}
\text{minimize} \quad & (f_0)_0 + \left(\frac{\mathrm{d}f_0}{\mathrm{d}x_i}\right)\Delta x_i \\
\text{with respect to} \quad & \Delta x_i \\
\text{subject to} \quad & (c_0)_0 + \left(\frac{\mathrm{d}c_0}{\mathrm{d}x_i}\right)\Delta x_i \geq 0 \\
& (c_i)_0 + \left(\frac{\mathrm{d}c_i}{\mathrm{d}x_i}\right)\Delta x_i \geq 0 \\
& \Delta x_{iL} \leq \Delta x_i \leq \Delta x_{iU}.
\end{aligned}
$$

Note the extra set of constraints in both system and discipline subproblems denoting the design variables bounds.

Bilevel Integrated System Synthesis (BLISS) 3



BLISS Algorithm

Input: Initial design variables x
Output: Optimal variables x^*, objective function f^*, and constraint values c^*

0: Initiate system optimization
repeat
  1: Initiate MDA
  repeat
    2: Evaluate discipline analyses
    3: Update coupling variables
  until $3 \to 2$: MDA has converged
  4: Initiate parallel discipline optimizations
  for Each discipline i **do**
    5: Evaluate discipline analysis
    6: Compute objective and constraint function values and derivatives with respect to local design variables
    7: Compute the optimal solutions for the disciplinary subproblem
  end for
  8: Initiate system optimization
  9: Compute objective and constraint function values and derivatives with respect to shared design variables using post-optimality analysis
  10: Compute optimal solution to system subproblem
until $11 \to 1$: System optimization has converged

Analytical Target Cascading (ATC) 1

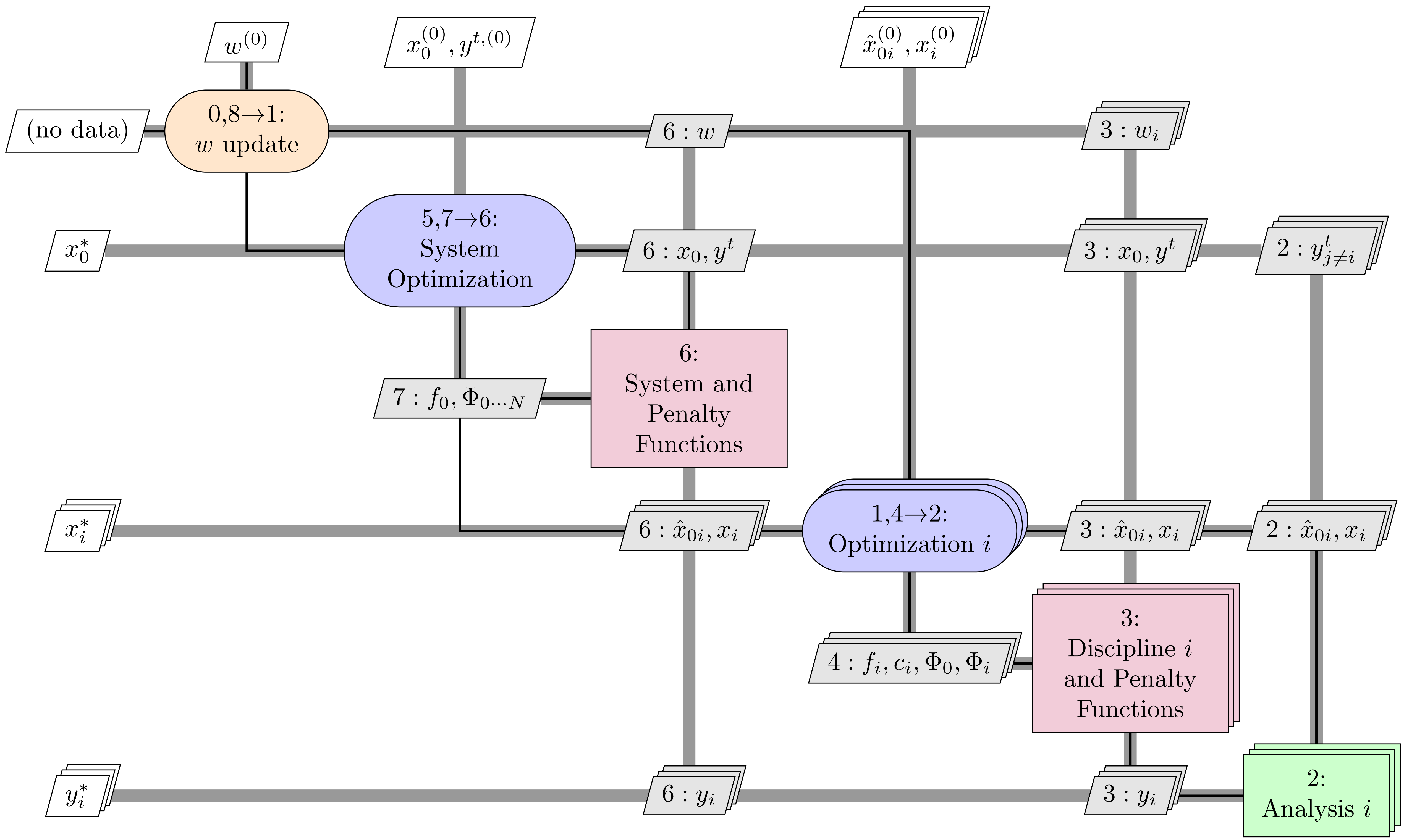
The ATC system subproblem is given by

$$
\begin{aligned}
\text{minimize} \quad & f_0\left(x, y^t\right) + \sum_{i=1}^{N} \Phi_i\left(\hat{x}_{0i} - x_0, y_i^t - y_i\left(x_0, x_i, y^t\right)\right) + \\
& \Phi_0\left(c_0\left(x, y^t\right)\right) \\
\text{with respect to} \quad & x_0, y^t,
\end{aligned}
$$

where Φ_0 is a penalty relaxation of the global design constraints and Φ_i is a penalty relaxation of the discipline i consistency constraints. The i^{th} discipline subproblem is:

$$
\begin{aligned}
\text{minimize} \quad & f_0\left(\hat{x}_{0i}, x_i, y_i\left(\hat{x}_{0i}, x_i, y_{j\neq i}^t\right), y_{j\neq i}^t\right) + f_i\left(\hat{x}_{0i}, x_i, y_i\left(\hat{x}_{0i}, x_i, y_{j\neq i}^t\right)\right) + \\
& \Phi_i\left(y_i^t - y_i\left(\hat{x}_{0i}, x_i, y_{j\neq i}^t\right), \hat{x}_{0i} - x_0\right) + \\
& \Phi_0\left(c_0\left(\hat{x}_{0i}, x_i, y_i\left(\hat{x}_{0i}, x_i, y_{j\neq i}^t\right), y_{j\neq i}^t\right)\right) \\
\text{with respect to} \quad & \hat{x}_{0i}, x_i \\
\text{subject to} \quad & c_i\left(\hat{x}_{0i}, x_i, y_i\left(\hat{x}_{0i}, x_i, y_{j\neq i}^t\right)\right) \geq 0.
\end{aligned}
$$

Analytical Target Cascading (ATC) 2



ATC Algorithm

Input: Initial design variables x
Output: Optimal variables x^*, objective function f^*, and constraint values c^*

0: Initiate main ATC iteration
repeat
  for Each discipline i **do**
    1: Initiate discipline optimizer
    repeat
      2: Evaluate disciplinary analysis
      3: Compute discipline objective and constraint functions and penalty function values
      4: Update discipline design variables
    until $4 \to 2$: Discipline optimization has converged
  end for
  5: Initiate system optimizer
  repeat
    6: Compute system objective, constraints, and all penalty functions
    7: Update system design variables and coupling targets.
  until $7 \to 6$: System optimization has converged
  8: Update penalty weights
until $8 \to 1$: Penalty weights are large enough

Asymmetric Subspace Optimization (ASO) 1

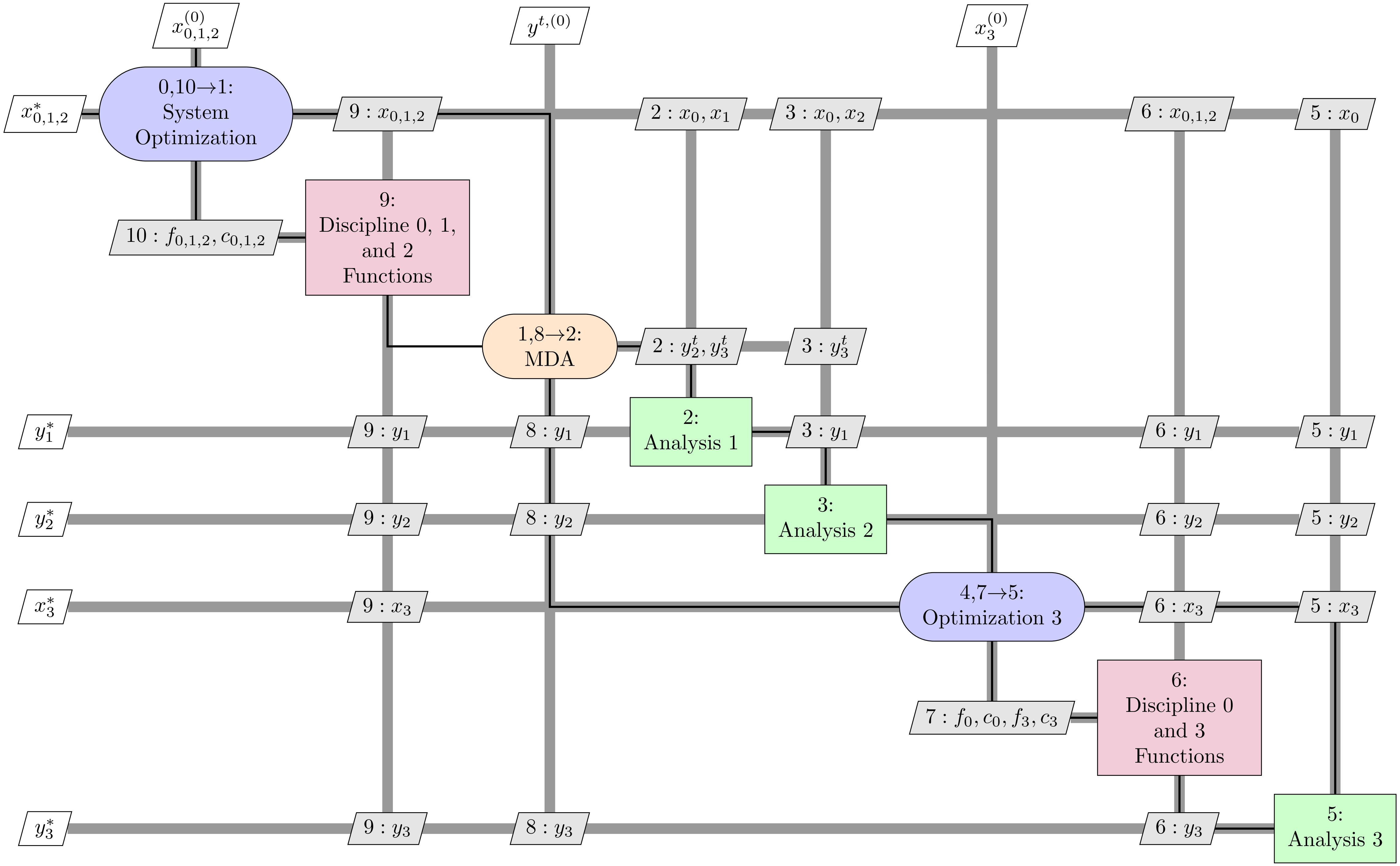
The system subproblem in ASO is

$$\begin{aligned}
\text{minimize} \quad & f_0\left(x, y\left(x, y\right)\right) + \sum_k f_k\left(x_0, x_k, y_k\left(x_0, x_k, y_{j\neq k}\right)\right) \\
\text{with respect to} \quad & x_0, x_k \\
\text{subject to} \quad & c_0\left(x, y\left(x, y\right)\right) \geq 0 \\
& c_k\left(x_0, x_k, y_k\left(x_0, x_k, y_{j\neq k}\right)\right) \geq 0 \qquad \text{for all } k,
\end{aligned}$$

where subscript k denotes disciplinary information that remains outside of the MDA. The disciplinary problem for discipline i, which is resolved inside the MDA, is

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

Asymmetric Subspace Optimization (ASO) 2

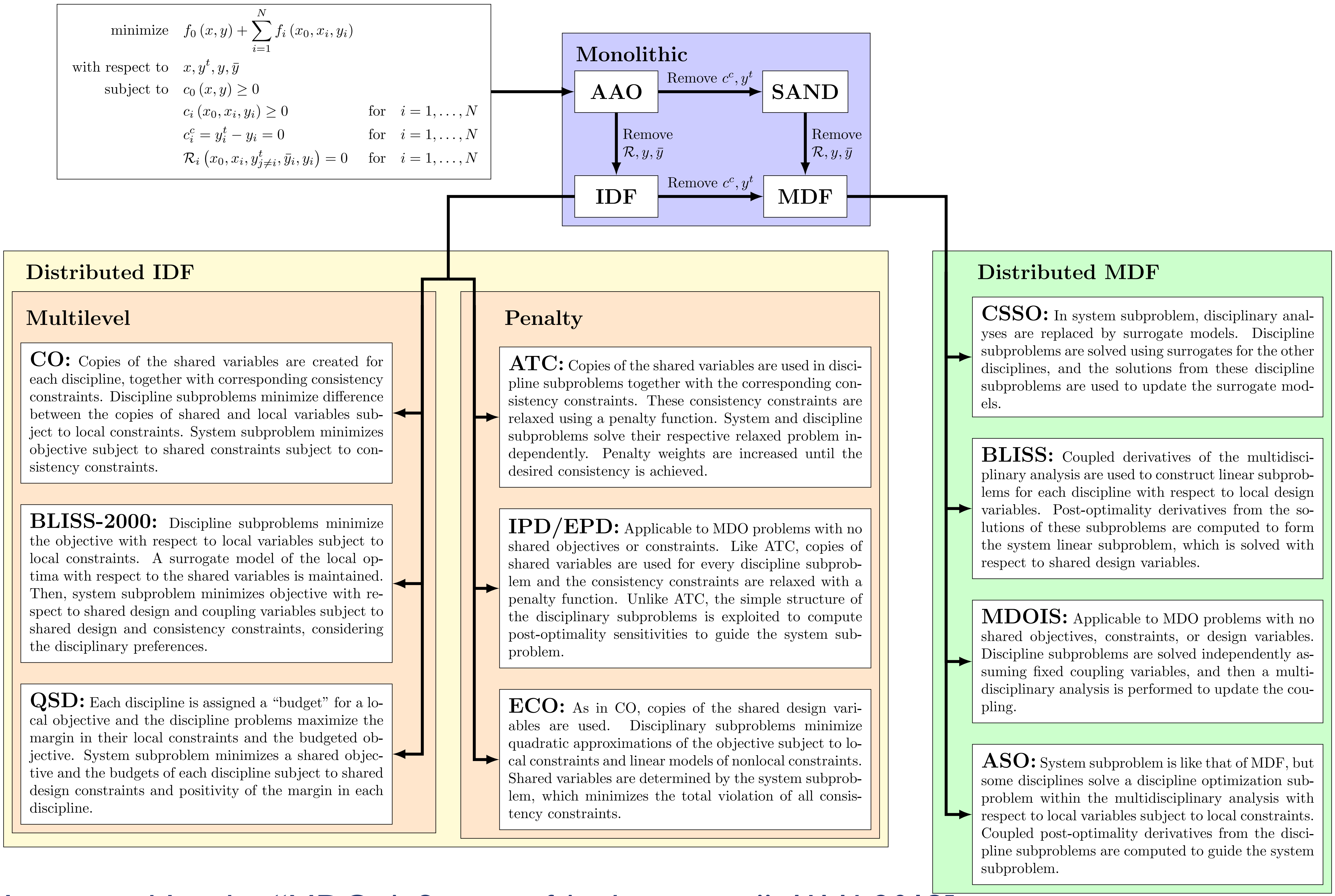


ASO Algorithm

Input: Initial design variables x
Output: Optimal variables x^*, objective function f^*, and constraint values c^*

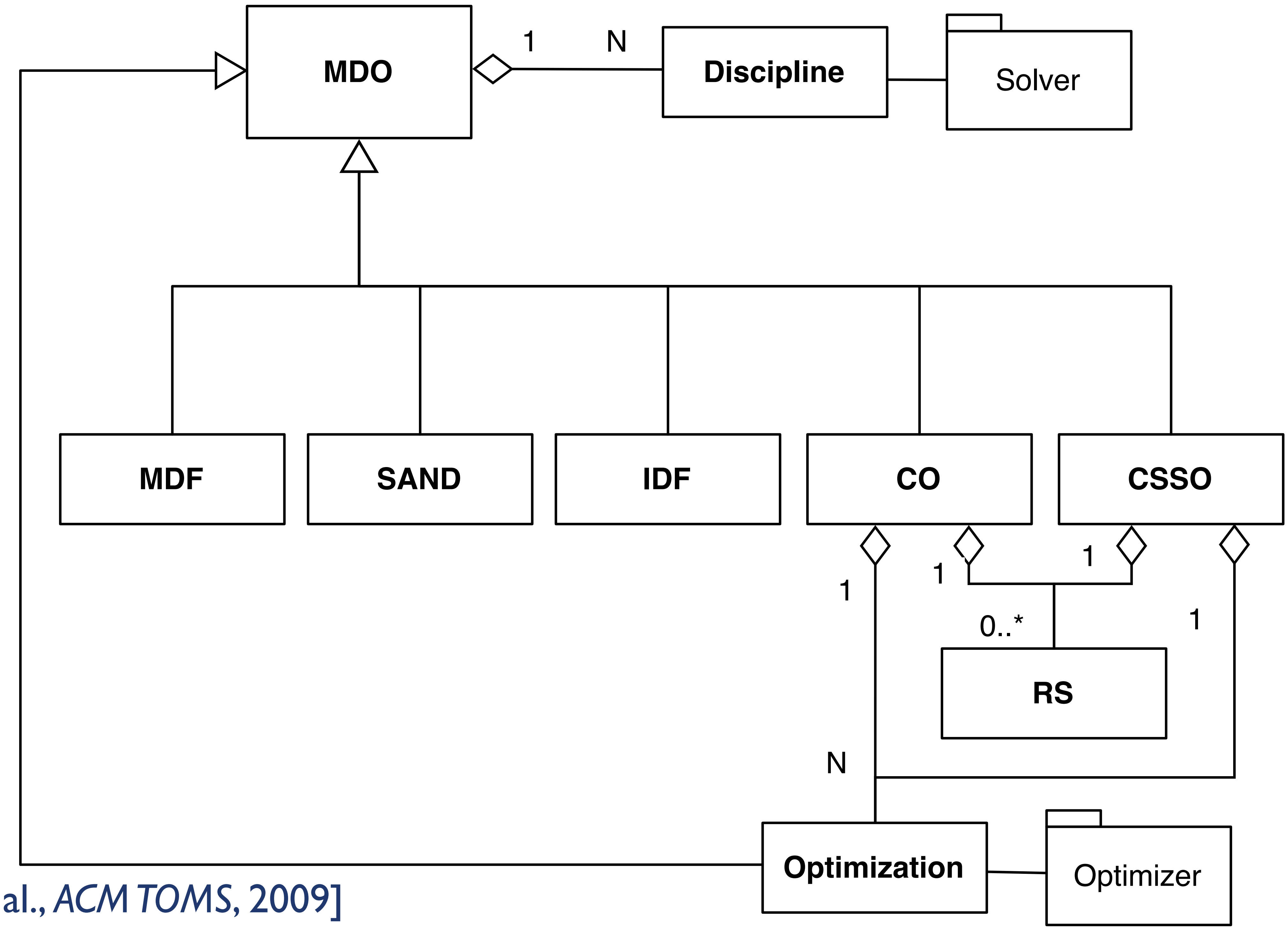
0: Initiate system optimization
repeat
  1: Initiate MDA
  repeat
    2: Evaluate Analysis 1
    3: Evaluate Analysis 2
    4: Initiate optimization of Discipline 3
    repeat
      5: Evaluate Analysis 3
      6: Compute discipline 3 objectives and constraints
      7: Update local design variables
    until $7 \rightarrow 5$: Discipline 3 optimization has converged
    8: Update coupling variables
  until $8 \rightarrow 2$ MDA has converged
  9: Compute objective and constraint function values for all disciplines 1 and 2
  10: Update design variables
until $10 \rightarrow 1$: System optimization has converged

Classification of MDO Architectures



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

Example: A Framework for Automatic Implementation of MDO 2



[Martins et al., *ACM TOMS*, 2009]

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

Computing derivatives: review and unification

[Martins and Hwang, *AIAAJ*, 2013]

What's in a name?

- **Sensitivity analysis**: Includes much more than derivatives of functions and numerical models
- **Sensitivity derivative**: Somewhat redundant?
- **Design sensitivities**: Acceptable term
- **Derivative**: Matches the scope of this talk most closely
- **Gradient/Jacobian**: This vector/matrix of derivatives is what we need for optimization

Scope

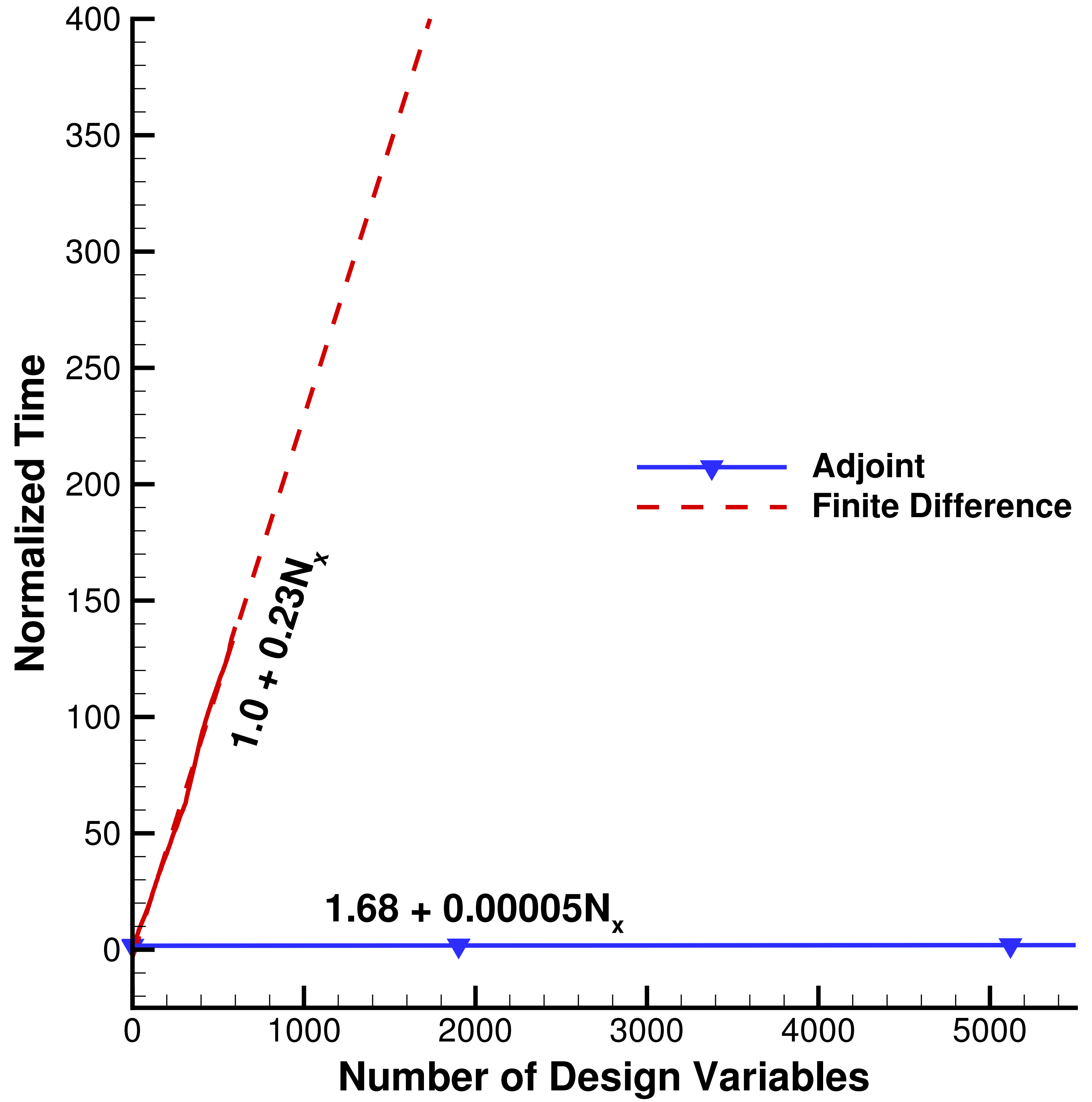
- First order derivatives
- Deterministic numerical models

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \underbrace{\begin{bmatrix} \frac{\mathrm{d}f_1}{\mathrm{d}x_1} & \cdots & \frac{\mathrm{d}f_1}{\mathrm{d}x_{n_x}} \\ \vdots & \ddots & \vdots \\ \frac{\mathrm{d}f_{n_f}}{\mathrm{d}x_1} & \cdots & \frac{\mathrm{d}f_{n_f}}{\mathrm{d}x_{n_x}} \end{bmatrix}}_{n_f \times n_x}$$

Applications of Derivatives

- Numerical optimization
  - For gradient-based optimization, need the gradient of the objective and constraints to iterate and satisfy the KKT optimality conditions
  - Only viable option for problems with large numbers of design variables
- Construction of linear approximations
- Gradient-enhanced surrogate models
- Newton-type methods
- Functional analysis
- Parameter estimation
- Aircraft stability derivatives

Computational Cost vs. Number of Variables

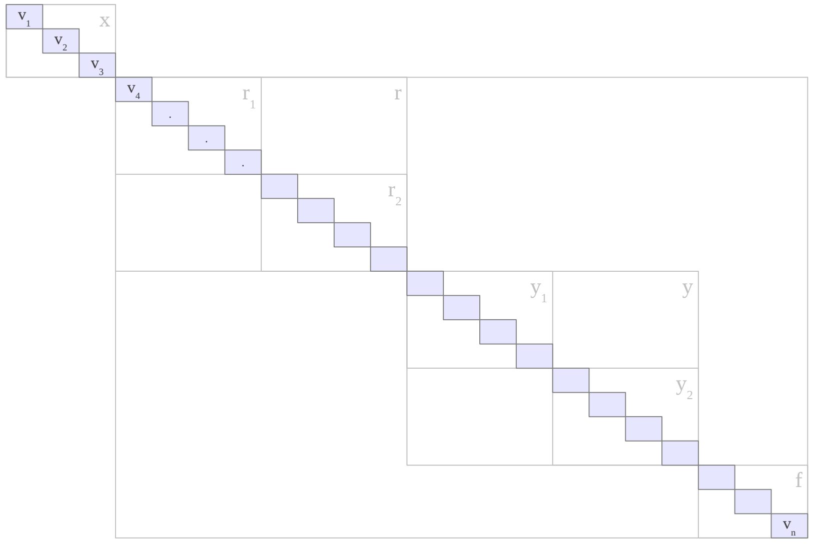


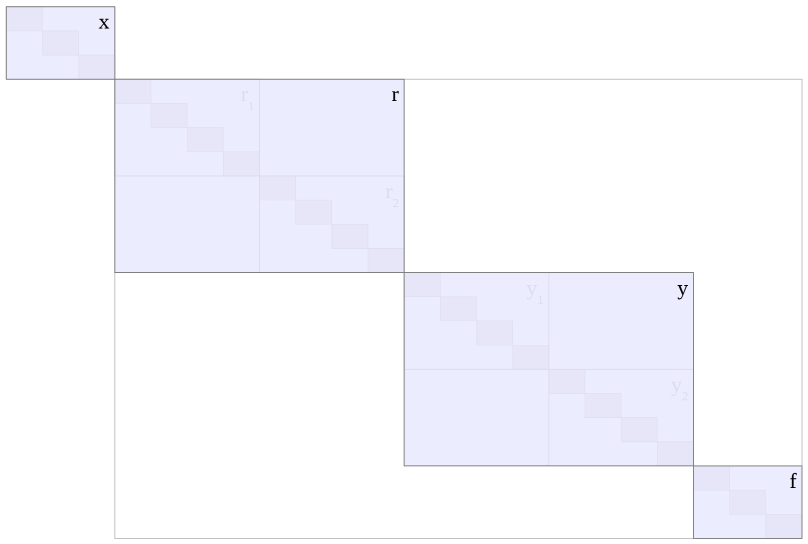
[Kenway, Kennedy and Martins, *AIAA Journal*, 2013]

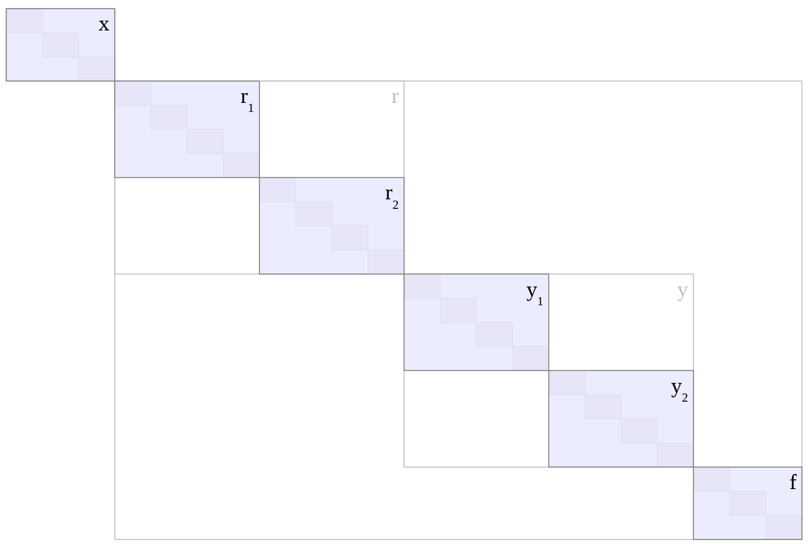
Objectives

- Review all methods for computing derivatives of multidisciplinary systems
- Unify the theory behind these methods
- Create opportunities for new insights

$$\boxed{\frac{\partial \mathcal{C}}{\partial \boldsymbol{v}} \frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}} = \boldsymbol{I} = \frac{\partial \mathcal{C}}{\partial \boldsymbol{v}}^T \frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}^T .}$$

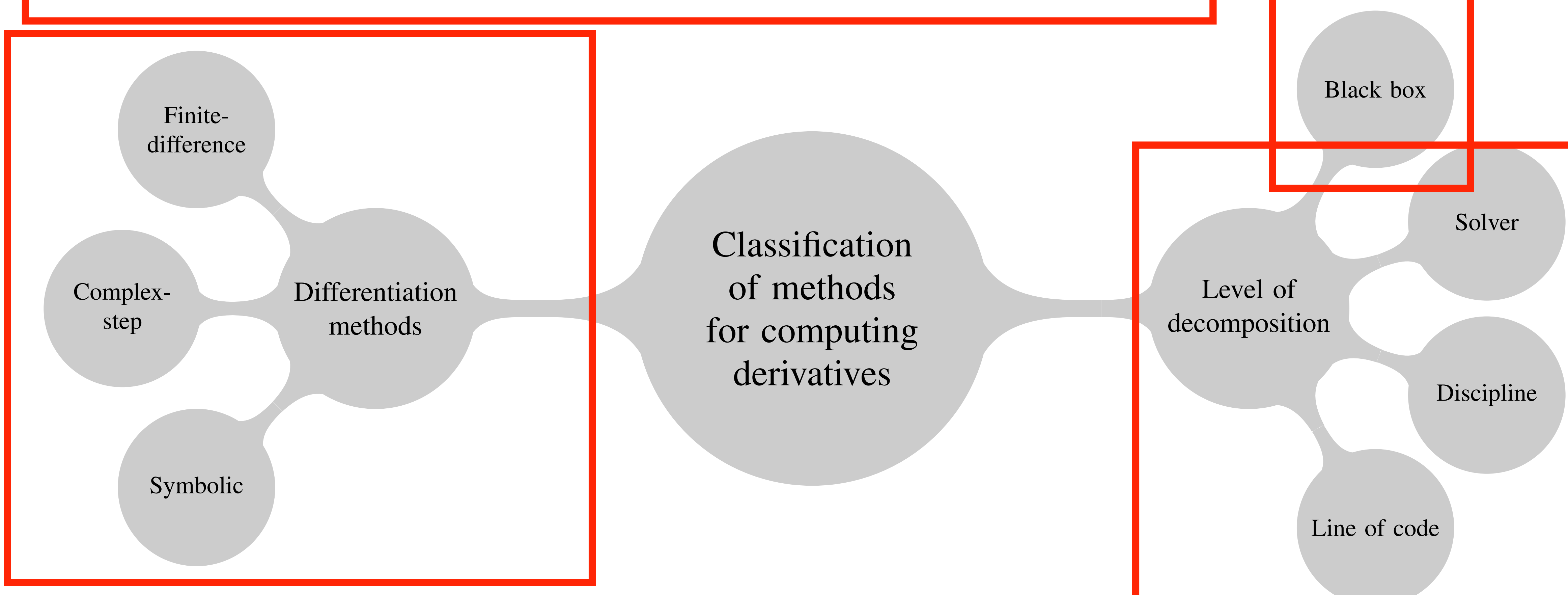






Methods for Computing Derivatives

- Finite differences
- Complex step
- Symbolic differentiation
- Automatic differentiation: forward and reverse
- Analytic methods: direct and adjoint
- Coupled derivatives of multidisciplinary systems



Black Box Methods

Finite Differences

$$\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{e}_j h) = \boldsymbol{f}(\boldsymbol{x}) + h\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}x_j} + \frac{h^2}{2}\frac{\mathrm{d}^2\boldsymbol{f}}{\mathrm{d}x_j^2} + \ldots \quad \Rightarrow$$

$$\boxed{\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}x_j} = \frac{\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{e}_j h) - \boldsymbol{f}(\boldsymbol{x})}{h} + \mathcal{O}(h)}$$

- Want to decrease truncation error by decreasing the step, but...
- Subtractive cancellation becomes worse as step decreases
- Require n_x evaluations of $\boldsymbol{f}$

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \begin{bmatrix} \boxed{\begin{matrix} \frac{\mathrm{d}f_1}{\mathrm{d}x_1} \\ \vdots \\ \frac{\mathrm{d}f_{n_f}}{\mathrm{d}x_1} \end{matrix}} & \begin{matrix} \cdots \\ \ddots \\ \cdots \end{matrix} & \begin{matrix} \frac{\mathrm{d}f_1}{\mathrm{d}x_{n_x}} \\ \vdots \\ \frac{\mathrm{d}f_{n_f}}{\mathrm{d}x_{n_x}} \end{matrix} \end{bmatrix}$$

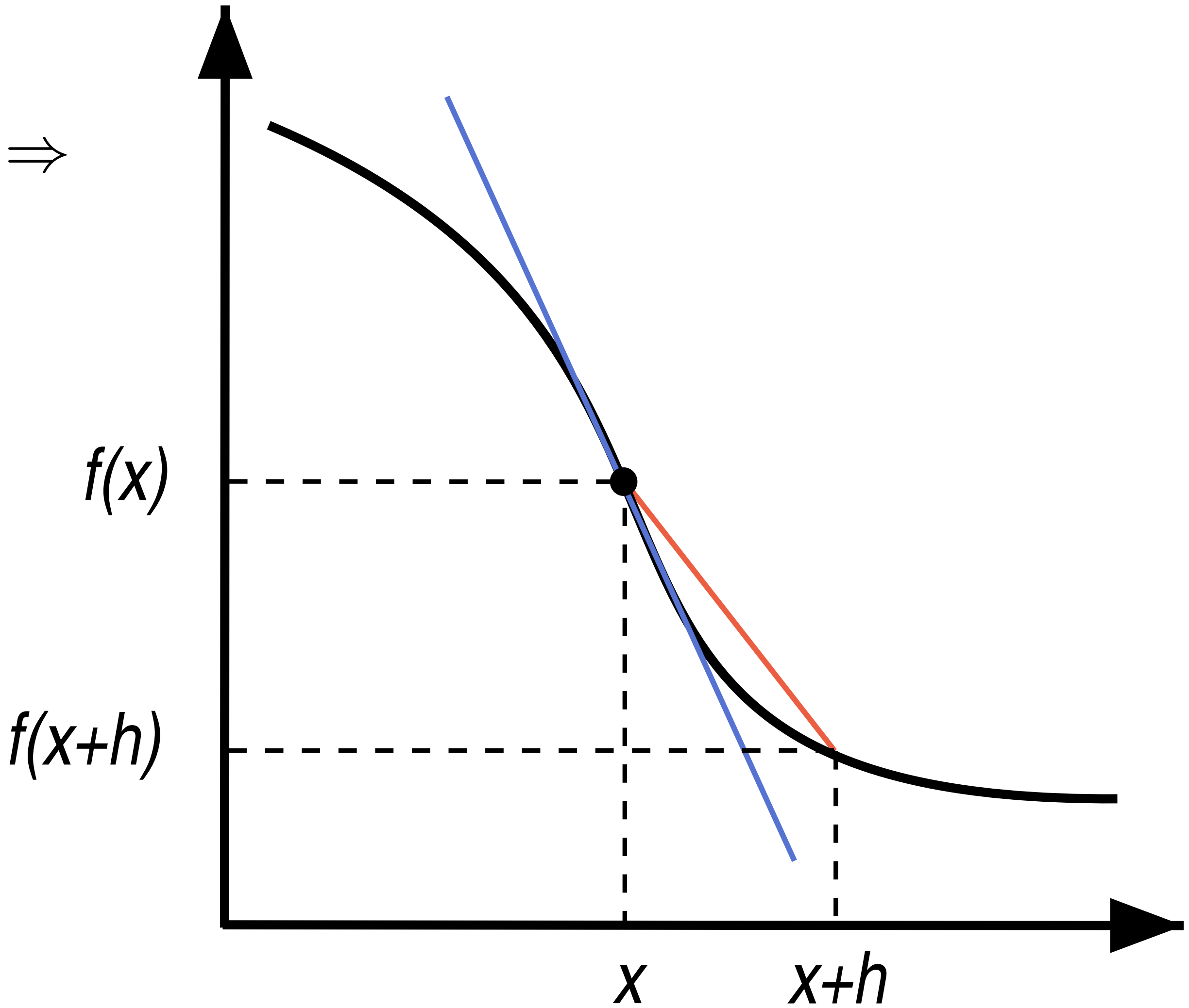
$n_f \times n_x$

Finite Differences

$f(x+h)$	$+1.234567890123431$
$f(x)$	$+1.234567890123456$
Δ f	-0.000000000000025

$$\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{e}_j h) = \boldsymbol{f}(\boldsymbol{x}) + h\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}x_j} + \frac{h^2}{2}\frac{\mathrm{d}^2\boldsymbol{f}}{\mathrm{d}x_j^2} + \ldots \quad \Rightarrow$$

$$\boxed{\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}x_j} = \frac{\boldsymbol{f}(\boldsymbol{x}+\boldsymbol{e}_j h) - \boldsymbol{f}(\boldsymbol{x})}{h} + \mathcal{O}(h)}$$



Finite difference approximation

Complex Step

$$\boldsymbol{f}(\boldsymbol{x} + ih\boldsymbol{e}_j) = \boldsymbol{f}(\boldsymbol{x}) + ih\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}x_j} - \frac{h^2}{2}\frac{\mathrm{d}^2\boldsymbol{f}}{\mathrm{d}x_j^2} - \frac{ih^3}{6}\frac{\mathrm{d}^3\boldsymbol{f}}{\mathrm{d}x_j^3} + \ldots$$

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}x_j} = \frac{\mathrm{Im}\left[\boldsymbol{f}(\boldsymbol{x} + ih\boldsymbol{e}_j)\right]}{h} + \mathcal{O}(h^2)$$

- No subtractive cancellation
- Precision of derivative matches that of $\boldsymbol{f}$
- Flexible implementation

[Martins, et al., *ACM TOMS*, 2003]

Complex Step: Another Derivation

- Consider a function, $f = u + iv$, of the complex variable, $z = x + iy$. If f is analytic the Cauchy–Riemann equations apply, i.e.,

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}$$

$$\frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x}.$$

- We can use the definition of a derivative in the right hand side of the first Cauchy–Riemann to get

$$\frac{\partial u}{\partial x} = \lim_{h \to 0} \frac{v(x + i(y + h)) - v(x + iy)}{h}$$

where h is a small real number.

[Martins, et al., *ACM TOMS*, 2003]

Complex Step: Another Derivation

- Since the functions are real functions of a real variable, $y = 0$, $u(x) = f(x)$ and $v(x) = 0$ and we can write,

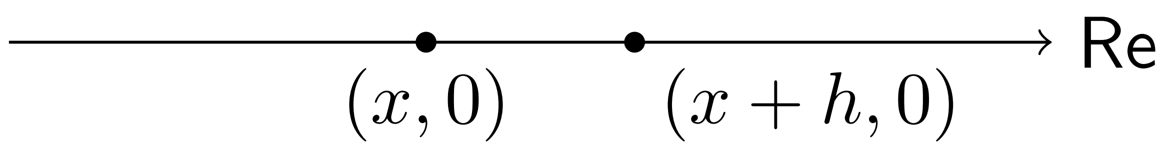
$$\frac{\partial f}{\partial x} = \lim_{h \to 0} \frac{\operatorname{Im}\left[f\left(x + ih\right)\right]}{h}.$$

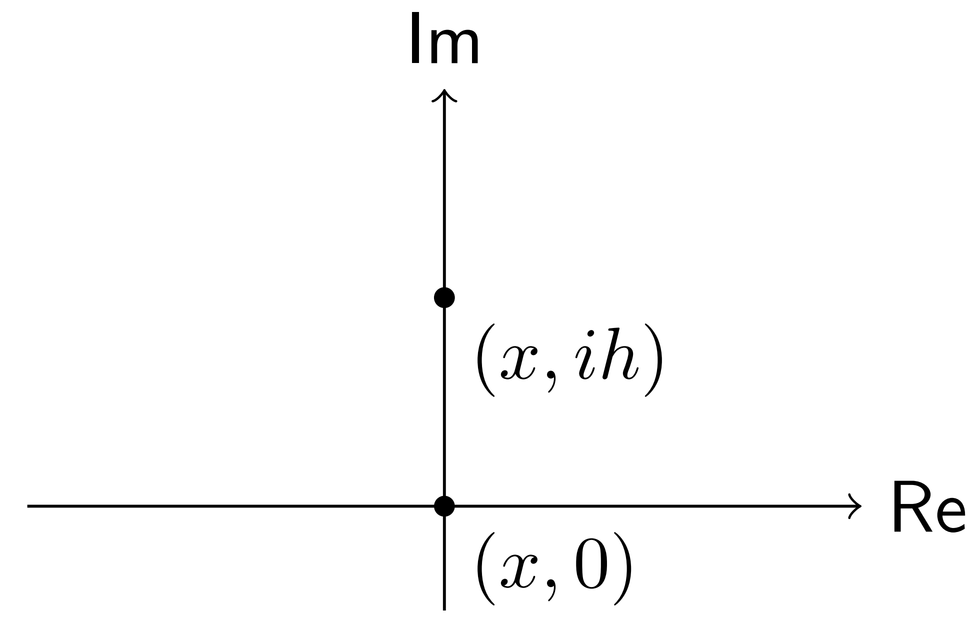
- For a small discrete h, this can be approximated by,

$$\frac{\partial f}{\partial x} \approx \frac{\operatorname{Im}\left[f\left(x + ih\right)\right]}{h}.$$

[Martins, et al., *ACM TOMS*, 2003]

Complex Step: Another Derivation





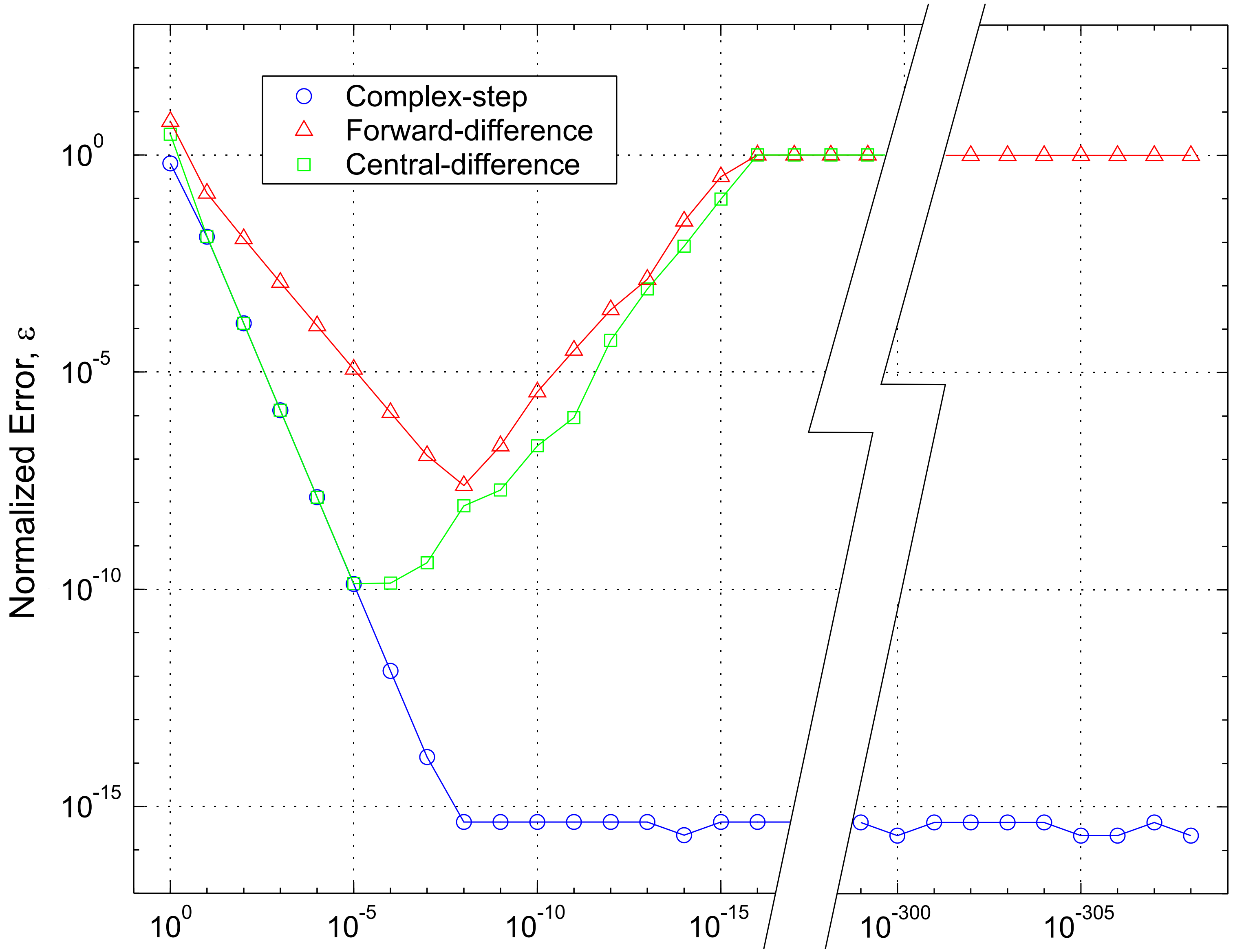
$$\frac{\partial F}{\partial x} \approx \frac{F(x+h) - F(x)}{h}$$

$$\frac{\partial F}{\partial x} \approx \frac{\mathrm{Im}[F(x+ih)] - \mathrm{Im}[F(x)]}{\mathrm{Im}[ih]}$$

$$\Rightarrow \frac{\partial F}{\partial x} \approx \frac{\mathrm{Im}[F(x+ih)]}{h}$$

[Martins, et al., *ACM TOMS*, 2003]

Example: The Complex-Step Method Applied to a Simple Function 2



Relative error of the derivative vs. decreasing step size

[Martins, et al., *ACM TOMS*, 2003]

Thinking Inside the Box

Algorithm, Variables, and Functions

Consider the sequence of all the variables and functions in an algorithm

$$\underbrace{v_i}_{\text{variable}} = \underbrace{V_i}_{\text{function}}(v_1, v_2, \ldots, v_{i-1}), \quad i = 1, \ldots, n$$

Assume a given variable depends only on previous one: all loops must be unrolled

The partial derivative of any of these functions wrt to any other variable is

$$\frac{\partial V_i}{\partial v_j} = \frac{V_i(v_1, \ldots, v_{j-1}, v_j + h, v_{j+1}, \ldots, v_{i-1}) - V_i(\cdot)}{h}$$

Residual Functions and State Variables

Computational model can be represented as a set of residuals of the governing equations:

$$\boldsymbol{r} = \boldsymbol{R}(\boldsymbol{x}, \boldsymbol{Y}(\boldsymbol{x})) = 0$$

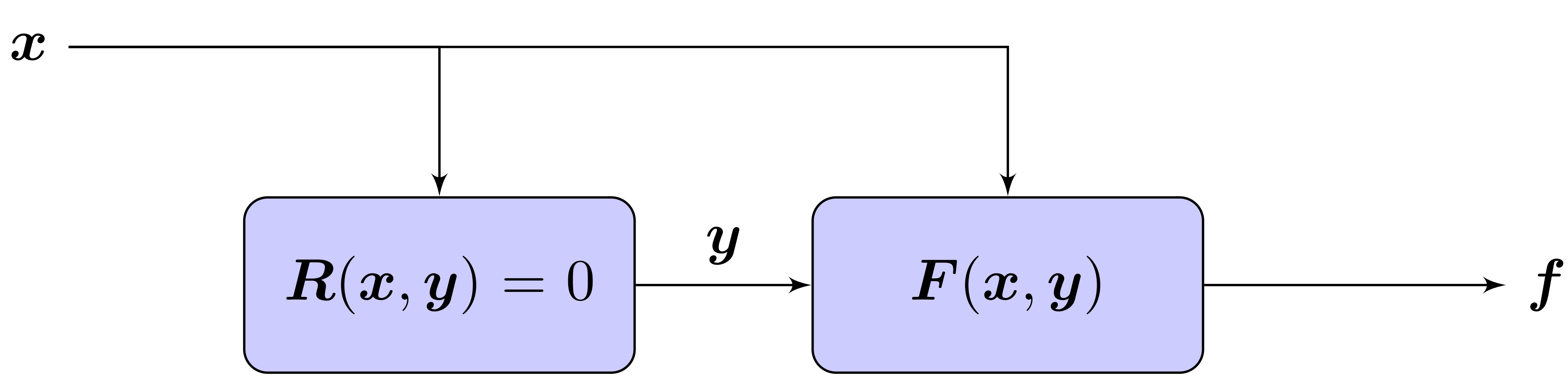
residuals

independent variables

state variables

The function of interest is:

$$\boldsymbol{f} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{Y}(\boldsymbol{x})).$$



$\boldsymbol{x} \in \mathbb{R}^{n_x}$

$\boldsymbol{y} \in \mathbb{R}^{n_y}$

$\boldsymbol{r} \in \mathbb{R}^{n_y}$

$\boldsymbol{f} \in \mathbb{R}^{n_f}$

One Chain to Rule Them All

Consider a set of variables $\boldsymbol{v} = [v_1, \ldots, v_n]^T$

and a set of functions $\boldsymbol{C} = [C_1(\boldsymbol{v}), \ldots, C_n(\boldsymbol{v})]^T.$

where the variables are uniquely defined by constraints $C_i(\boldsymbol{v}) = 0,$

Linearizing these functions:

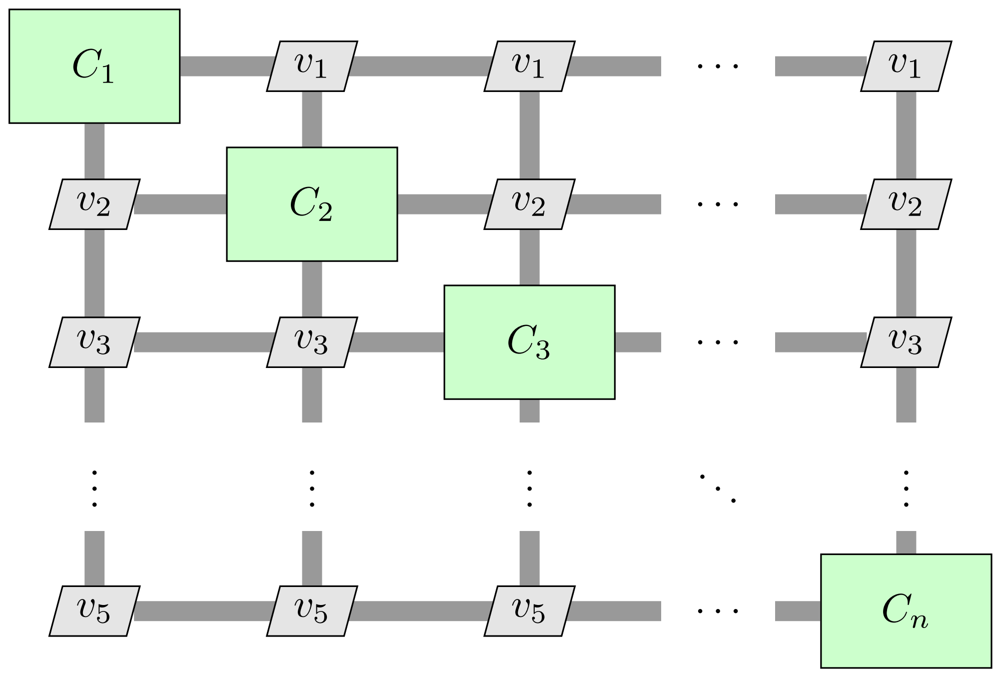
$$\Delta\bar{\boldsymbol{c}} = \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}} \Delta\boldsymbol{v},$$

After some manipulations, this yields the chain rule

$$\boxed{\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}} \frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}} = \boldsymbol{I} = \frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}^T \frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}^T .}$$

Chain Rule in Matrix Form

Variables and Constraints



$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{v}) = \begin{bmatrix} C_1(v_1, \ldots, v_n) \\ C_2(v_1, \ldots, v_n) \\ \vdots \\ C_n(v_1, \ldots, v_n) \end{bmatrix}$$

Derivation

$$\left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right] = \boldsymbol{I} = \left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]^T\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]^T$$

$$\begin{bmatrix} \frac{\partial C_1}{\partial v_1} & \cdots & \frac{\partial C_1}{\partial v_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial C_n}{\partial v_1} & \cdots & \frac{\partial C_n}{\partial v_n} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}v_1}{\mathrm{d}c_1} & \cdots & \frac{\mathrm{d}v_1}{\mathrm{d}c_n} \\ \vdots & \ddots & \vdots \\ \frac{\mathrm{d}v_n}{\mathrm{d}c_1} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}c_n} \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} \frac{\partial C_1}{\partial v_1} & \cdots & \frac{\partial C_n}{\partial v_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial C_1}{\partial v_n} & \cdots & \frac{\partial C_n}{\partial v_n} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}v_1}{\mathrm{d}c_1} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}c_1} \\ \vdots & \ddots & \vdots \\ \frac{\mathrm{d}v_1}{\mathrm{d}c_n} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}c_n} \end{bmatrix}$$

Forward form

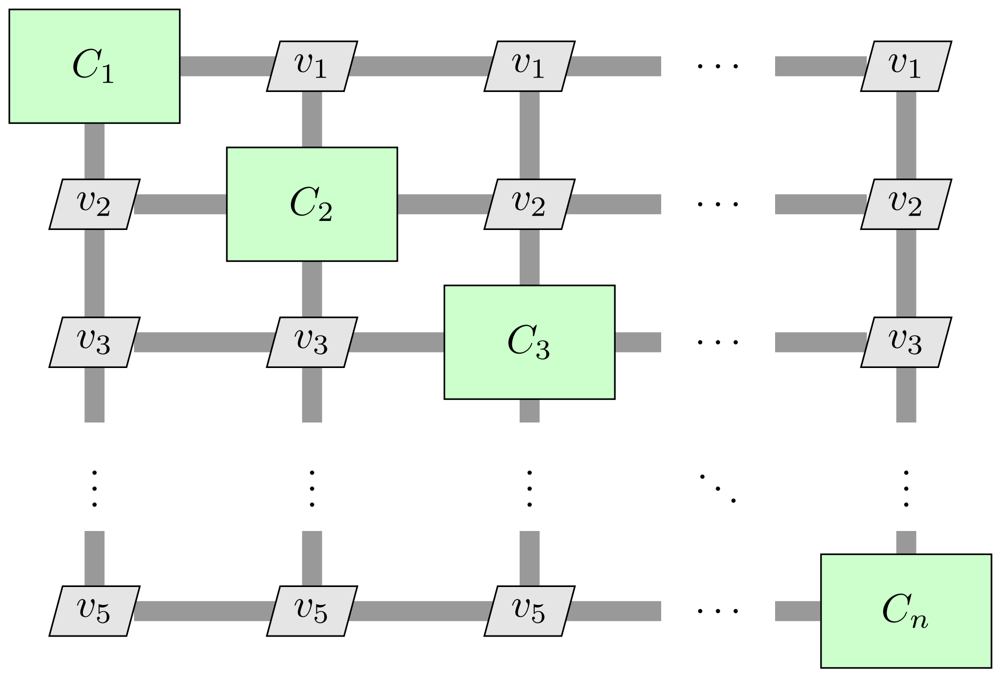
$$\sum_{k=1}^{n} \frac{\partial C_i}{\partial v_k}\frac{\mathrm{d}v_k}{\mathrm{d}c_j} = \delta_{ij}$$

Reverse form

$$\sum_{k=1}^{n} \frac{\mathrm{d}v_i}{\mathrm{d}c_k}\frac{\partial C_k}{\partial v_j} = \delta_{ij}$$

Chain Rule in Matrix Form

Variables and Constraints



$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{v}) = \begin{bmatrix} C_1(v_1, \ldots, v_n) \\ C_2(v_1, \ldots, v_n) \\ \vdots \\ C_n(v_1, \ldots, v_n) \end{bmatrix}$$

Derivation

$$\left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right] = \boldsymbol{I} = \left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]^T \left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]^T$$

$$\begin{bmatrix} \frac{\partial C_1}{\partial v_1} & \cdots & \frac{\partial C_1}{\partial v_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial C_n}{\partial v_1} & \cdots & \frac{\partial C_n}{\partial v_n} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}v_1}{\mathrm{d}c_1} & \cdots & \frac{\mathrm{d}v_1}{\mathrm{d}c_n} \\ \vdots & \ddots & \vdots \\ \frac{\mathrm{d}v_n}{\mathrm{d}c_1} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}c_n} \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} \frac{\partial C_1}{\partial v_1} & \cdots & \frac{\partial C_n}{\partial v_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial C_1}{\partial v_n} & \cdots & \frac{\partial C_n}{\partial v_n} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}v_1}{\mathrm{d}c_1} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}c_1} \\ \vdots & \ddots & \vdots \\ \frac{\mathrm{d}v_1}{\mathrm{d}c_n} & \cdots & \frac{\mathrm{d}v_n}{\mathrm{d}c_n} \end{bmatrix}$$

Forward form

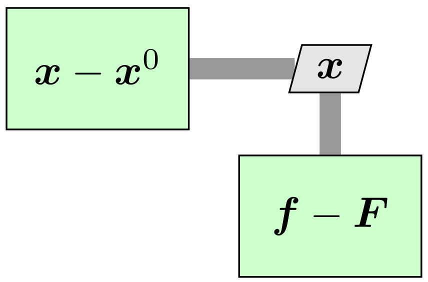
$$\sum_{k=1}^{n} \frac{\partial C_i}{\partial v_k} \frac{\mathrm{d}v_k}{\mathrm{d}c_j} = \delta_{ij}$$

Reverse form

$$\sum_{k=1}^{n} \frac{\mathrm{d}v_i}{\mathrm{d}c_k} \frac{\partial C_k}{\partial v_j} = \delta_{ij}$$

Monolithic Differentiation

Variables and Constraints



$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{f} \end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{v}) = \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}^0 \\ \boldsymbol{f} - \boldsymbol{F}(\boldsymbol{x}) \end{bmatrix}$$

Derivation

$$\left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right] \left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right] = \boldsymbol{I} = \left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]^T \left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]^T$$

$$\begin{bmatrix} \dfrac{\partial(\boldsymbol{x} - \boldsymbol{x}^0)}{\partial \boldsymbol{x}} & \dfrac{\partial(\boldsymbol{x} - \boldsymbol{x}^0)}{\partial \boldsymbol{f}} \\ \dfrac{\partial(\boldsymbol{f} - \boldsymbol{F})}{\partial \boldsymbol{x}} & \dfrac{\partial(\boldsymbol{f} - \boldsymbol{F})}{\partial \boldsymbol{f}} \end{bmatrix} \begin{bmatrix} \dfrac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{x}} & \dfrac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{f}} \\ \dfrac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} & \dfrac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{f}} \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} \dfrac{\partial(\boldsymbol{x} - \boldsymbol{x}^0)}{\partial \boldsymbol{x}}^T & \dfrac{\partial(\boldsymbol{f} - \boldsymbol{F})}{\partial \boldsymbol{x}}^T \\ \dfrac{\partial(\boldsymbol{x} - \boldsymbol{x}^0)}{\partial \boldsymbol{f}}^T & \dfrac{\partial(\boldsymbol{f} - \boldsymbol{F})}{\partial \boldsymbol{f}}^T \end{bmatrix} \begin{bmatrix} \dfrac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{x}}^T & \dfrac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}^T \\ \dfrac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{f}}^T & \dfrac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{f}}^T \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\dfrac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \dfrac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} & \boldsymbol{I} \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} \boldsymbol{I} & -\dfrac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}^T \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \dfrac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}^T \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

Monolithic differentiation (from forward form)

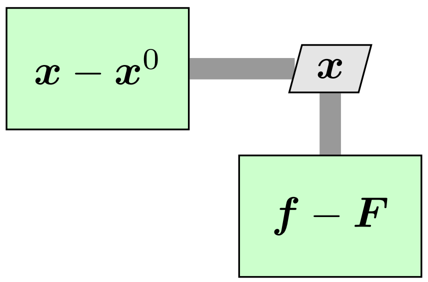
$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}$$

Monolithic differentiation (from reverse form)

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

Monolithic Differentiation

Variables and Constraints



$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{f} \end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{v}) = \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}^0 \\ \boldsymbol{f} - \boldsymbol{F}(\boldsymbol{x}) \end{bmatrix}$$

Derivation

$$\left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right] \left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right] = \boldsymbol{I} = \left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]^T \left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]^T$$

$$\begin{bmatrix} \dfrac{\partial(\boldsymbol{x} - \boldsymbol{x}^0)}{\partial \boldsymbol{x}} & \dfrac{\partial(\boldsymbol{x} - \boldsymbol{x}^0)}{\partial \boldsymbol{f}} \\ \dfrac{\partial(\boldsymbol{f} - \boldsymbol{F})}{\partial \boldsymbol{x}} & \dfrac{\partial(\boldsymbol{f} - \boldsymbol{F})}{\partial \boldsymbol{f}} \end{bmatrix} \begin{bmatrix} \dfrac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{x}} & \dfrac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{f}} \\ \dfrac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} & \dfrac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{f}} \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} \dfrac{\partial(\boldsymbol{x} - \boldsymbol{x}^0)^T}{\partial \boldsymbol{x}} & \dfrac{\partial(\boldsymbol{f} - \boldsymbol{F})^T}{\partial \boldsymbol{x}} \\ \dfrac{\partial(\boldsymbol{x} - \boldsymbol{x}^0)^T}{\partial \boldsymbol{f}} & \dfrac{\partial(\boldsymbol{f} - \boldsymbol{F})^T}{\partial \boldsymbol{f}} \end{bmatrix} \begin{bmatrix} \dfrac{\mathrm{d}\boldsymbol{x}^T}{\mathrm{d}\boldsymbol{x}} & \dfrac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{x}} \\ \dfrac{\mathrm{d}\boldsymbol{x}^T}{\mathrm{d}\boldsymbol{f}} & \dfrac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{f}} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\dfrac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \dfrac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} & \boldsymbol{I} \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} \boldsymbol{I} & -\dfrac{\partial \boldsymbol{F}^T}{\partial \boldsymbol{x}} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \dfrac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{x}} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

Monolithic differentiation (from forward form)

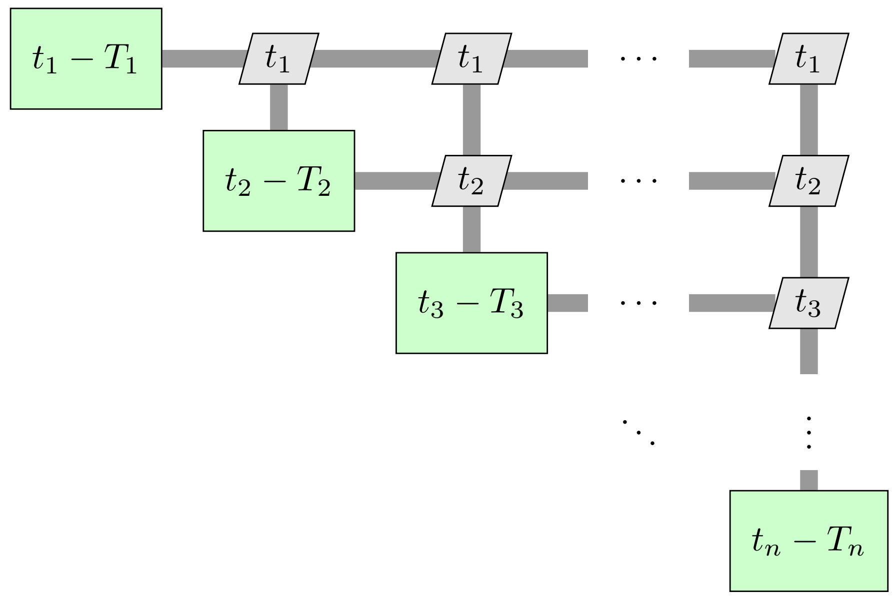
$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}$$

Monolithic differentiation (from reverse form)

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

Algorithmic Differentiation

Variables and Constraints



$$\boldsymbol{v} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{v}) = \begin{bmatrix} t_1 - T_1() \\ t_2 - T_2(t_1) \\ \vdots \\ t_n - T_n(t_1, \ldots, t_{n-1}) \end{bmatrix}$$

Derivation

$$\left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right] = \boldsymbol{I} = \left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]^T \left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]^T$$

$$\begin{bmatrix} 1 & 0 & \ldots & 0 \\ -\dfrac{\partial T_2}{\partial t_1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ -\dfrac{\partial T_n}{\partial t_1} & \ldots & -\dfrac{\partial T_n}{\partial t_{n-1}} & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & \ldots & 0 \\ \dfrac{\mathrm{d}t_2}{\mathrm{d}t_1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \dfrac{\mathrm{d}t_n}{\mathrm{d}t_1} & \ldots & \dfrac{\mathrm{d}t_n}{\mathrm{d}t_{n-1}} & 1 \end{bmatrix}
= \boldsymbol{I} =
\begin{bmatrix} 1 & -\dfrac{\partial T_2}{\partial t_1} & \ldots & -\dfrac{\partial T_n}{\partial t_1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\dfrac{\partial T_n}{\partial t_{n-1}} \\ 0 & \ldots & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & \dfrac{\mathrm{d}t_2}{\mathrm{d}t_1} & \ldots & \dfrac{\mathrm{d}t_n}{\mathrm{d}t_1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & 1 & \dfrac{\mathrm{d}t_n}{\mathrm{d}t_{n-1}} \\ 0 & \ldots & 0 & 1 \end{bmatrix}$$

Forward mode AD

$$\frac{\mathrm{d}t_i}{\mathrm{d}t_j} = \delta_{ij} + \sum_{k=j}^{i-1} \frac{\partial T_i}{\partial t_k} \frac{\mathrm{d}t_k}{\mathrm{d}t_j}$$

Reverse mode AD

$$\frac{\mathrm{d}t_i}{\mathrm{d}t_j} = \delta_{ij} + \sum_{k=j+1}^{i} \frac{\mathrm{d}t_i}{\mathrm{d}t_k} \frac{\partial T_k}{\partial t_j}$$

AD Example

$$x_1$$
$$x_2$$
$$y_1$$
$$y_2$$
$$R_1(x_1, x_2, y_1, y_2) = x_1 y_1 + 2y_2 - \sin x_1$$
$$R_2(x_1, x_2, y_1, y_2) = -y_1 + x_2^2 y_2$$
$$F_1(x_1, x_2, y_1, y_2) = y_1$$
$$F_2(x_1, x_2, y_1, y_2) = y_2 \sin x_1$$

```
FUNCTION F(x)
  REAL :: x(2), det, y(2), f(2)
  det = 2 + x(1)*x(2)**2
  y(1) = x(2)**2*SIN(x(1))/det
  y(2) = SIN(x(1))/det
  f(1) = y(1)
  f(2) = y(2)*SIN(x(1))
  RETURN
END FUNCTION F
```

Forward AD

```
FUNCTION F(x)
  REAL :: x(2), det, y(2), f(2)
  det = 2 + x(1)*x(2)**2
  y(1) = x(2)**2*SIN(x(1))/det
  y(2) = SIN(x(1))/det
  f(1) = y(1)
  f(2) = y(2)*SIN(x(1))
  RETURN
END FUNCTION F
```

```
FUNCTION F_D(x, xd, f)
  REAL :: x(2), xd(2)
  REAL :: det, detd
  REAL :: y(2), yd(2)
  REAL :: f(2), f_d(2)
  detd = xd(1)*x(2)**2 + x(1)*2*x(2)*xd(2)
  det = 2 + x(1)*x(2)**2
  yd = 0.0
  yd(1) = ((2*x(2)*xd(2)*SIN(x(1))+x(2)**2*xd(1)*COS(x(1)))*det-
      x(2)**2*&
&    SIN(x(1))*detd)/det**2
  y(1) = x(2)**2*SIN(x(1))/det
  yd(2) = (xd(1)*COS(x(1))*det-SIN(x(1))*detd)/det**2
  y(2) = SIN(x(1))/det
  f_d = 0.0
  f_d(1) = yd(1)
  f(1) = y(1)
  f_d(2) = yd(2)*SIN(x(1)) + y(2)*xd(1)*COS(x(1))
  f(2) = y(2)*SIN(x(1))
  RETURN
END FUNCTION F_D
```

Reverse AD

```
FUNCTION F(x)
  REAL :: x(2), det, y(2), f(2)
  det = 2 + x(1)*x(2)**2
  y(1) = x(2)**2*SIN(x(1))/det
  y(2) = SIN(x(1))/det
  f(1) = y(1)
  f(2) = y(2)*SIN(x(1))
  RETURN
END FUNCTION F
```

```
SUBROUTINE F_B(x, xb, fb)
  REAL :: x(2), xb(2),
  REAL :: y(2), yb(2)
  REAL :: f(2), fb(2)
  REAL :: det, detb, tempb, temp
  det = 2 + x(1)*x(2)**2
  y(1) = x(2)**2*SIN(x(1))/det
  y(2) = SIN(x(1))/det
  xb = 0.0
  yb = 0.0
  yb(2) = yb(2) + SIN(x(1))*fb(2)
  xb(1) = xb(1) + y(2)*COS(x(1))*fb(2)
  fb(2) = 0.0
  yb(1) = yb(1) + fb(1)
  xb(1) = xb(1) + COS(x(1))*yb(2)/det
  detb = -(SIN(x(1))*yb(2)/det**2)
  yb(2) = 0.0
  tempb = SIN(x(1))*yb(1)/det
  temp = x(2)**2/det
  xb(2) = xb(2) + 2*x(2)*tempb
  detb = detb - temp*tempb
  xb(1) = xb(1) + x(2)**2*detb + temp*COS(x(1))*yb(1)
  xb(2) = xb(2) + x(1)*2*x(2)*detb
END SUBROUTINE F_B
```

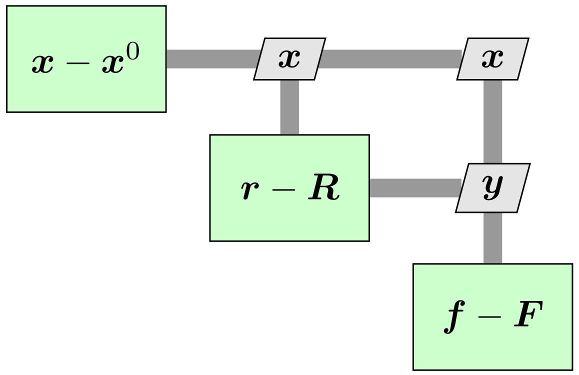
AD Example: Forward and Reverse

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & -2 & 1 & 0 & 0 & 0 & 0 \\ -0.18 & -0.561 & 0.093 & 1 & 0 & 0 & 0 \\ -0.18 & 0 & 0.093 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ -0.152 & 0 & 0 & 0 & -0.841 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 2 \\ 0.087 & 0.374 \\ 0.087 & -0.187 \\ 0.087 & 0.374 \\ 0.224 & -0.157 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -1 & -0.18 & -0.18 & 0 & -0.152 \\ 0 & 1 & -2 & -0.561 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.093 & 0.093 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -0.841 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.087 & 0.224 \\ 0.374 & -0.157 \\ -0.093 & -0.079 \\ 1 & 0 \\ 0 & 0.841 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Analytic Methods

Variables and Constraints



$$\boldsymbol{v}=\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\\ \boldsymbol{f}\end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{v})=\begin{bmatrix}\boldsymbol{x}-\boldsymbol{x}^0\\ \boldsymbol{r}-\boldsymbol{R}(\boldsymbol{x},\boldsymbol{y})\\ \boldsymbol{f}-\boldsymbol{F}(\boldsymbol{x},\boldsymbol{y})\end{bmatrix}$$

Derivation

$$\left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]=\boldsymbol{I}=\left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]^T\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]^T$$

$$\begin{bmatrix}\frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)}{\partial \boldsymbol{x}} & \frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)}{\partial \boldsymbol{y}} & \frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)}{\partial \boldsymbol{f}}\\ \frac{\partial(\boldsymbol{r}-\boldsymbol{R})}{\partial \boldsymbol{x}} & \frac{\partial(\boldsymbol{r}-\boldsymbol{R})}{\partial \boldsymbol{y}} & \frac{\partial(\boldsymbol{r}-\boldsymbol{R})}{\partial \boldsymbol{f}}\\ \frac{\partial(\boldsymbol{f}-\boldsymbol{F})}{\partial \boldsymbol{x}} & \frac{\partial(\boldsymbol{f}-\boldsymbol{F})}{\partial \boldsymbol{y}} & \frac{\partial(\boldsymbol{f}-\boldsymbol{F})}{\partial \boldsymbol{f}}\end{bmatrix}\begin{bmatrix}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{f}}\\ \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{f}}\\ \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{f}}\end{bmatrix}=\boldsymbol{I}=\begin{bmatrix}\frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)^T}{\partial \boldsymbol{x}} & \frac{\partial(\boldsymbol{r}-\boldsymbol{R})^T}{\partial \boldsymbol{x}} & \frac{\partial(\boldsymbol{f}-\boldsymbol{F})^T}{\partial \boldsymbol{x}}\\ \frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)^T}{\partial \boldsymbol{y}} & \frac{\partial(\boldsymbol{r}-\boldsymbol{R})^T}{\partial \boldsymbol{y}} & \frac{\partial(\boldsymbol{f}-\boldsymbol{F})^T}{\partial \boldsymbol{y}}\\ \frac{\partial(\boldsymbol{x}-\boldsymbol{x}^0)^T}{\partial \boldsymbol{f}} & \frac{\partial(\boldsymbol{r}-\boldsymbol{R})^T}{\partial \boldsymbol{f}} & \frac{\partial(\boldsymbol{f}-\boldsymbol{F})^T}{\partial \boldsymbol{f}}\end{bmatrix}\begin{bmatrix}\frac{\mathrm{d}\boldsymbol{x}^T}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{y}^T}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{x}}\\ \frac{\mathrm{d}\boldsymbol{x}^T}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{y}^T}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{r}}\\ \frac{\mathrm{d}\boldsymbol{x}^T}{\mathrm{d}\boldsymbol{f}} & \frac{\mathrm{d}\boldsymbol{y}^T}{\mathrm{d}\boldsymbol{f}} & \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{f}}\end{bmatrix}$$

$$\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0}\\ -\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}} & \boldsymbol{0}\\ -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0}\\ \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{r}} & \boldsymbol{0}\\ \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}} & \boldsymbol{I}\end{bmatrix}=\boldsymbol{I}=\begin{bmatrix}\boldsymbol{I} & -\frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{F}^T}{\partial \boldsymbol{x}}\\ \boldsymbol{0} & -\frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{y}} & -\frac{\partial \boldsymbol{F}^T}{\partial \boldsymbol{y}}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{I} & \frac{\mathrm{d}\boldsymbol{y}^T}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{x}}\\ \boldsymbol{0} & \frac{\mathrm{d}\boldsymbol{y}^T}{\mathrm{d}\boldsymbol{r}} & \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{r}}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}$$

Direct method

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}}\frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}}=-\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}$$

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

Adjoint method

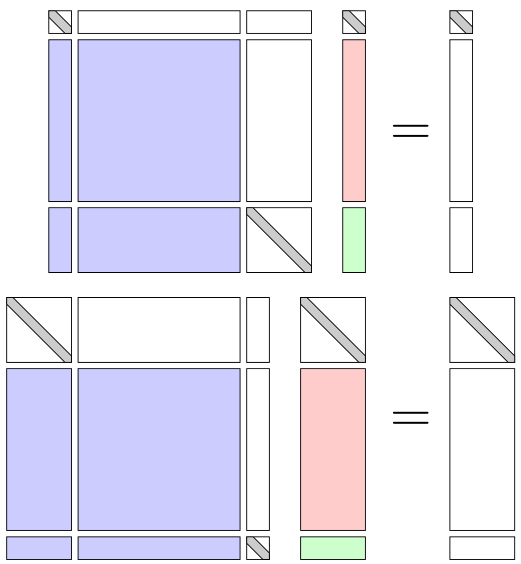
$$\frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{y}}\frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{r}}=-\frac{\partial \boldsymbol{F}^T}{\partial \boldsymbol{y}}$$

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

Analytic Methods: Direct vs. Adjoint

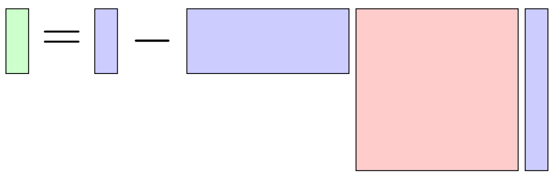
From forward chain rule

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ -\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}} & \boldsymbol{0} \\ -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}} \\ \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}$$

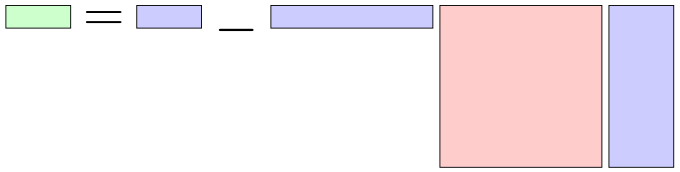


Solution

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



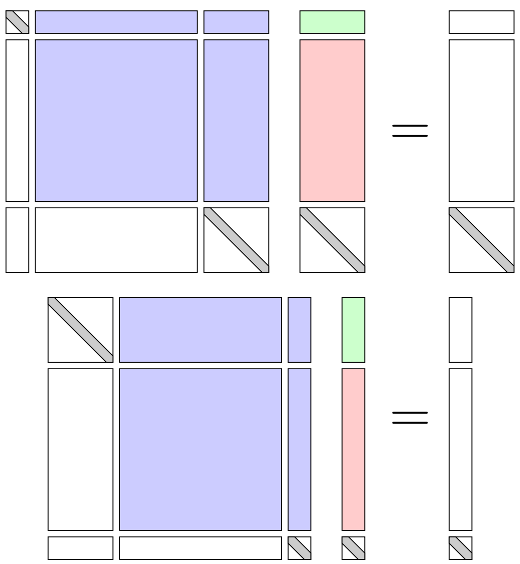
$n_x < n_f$



$n_x > n_f$

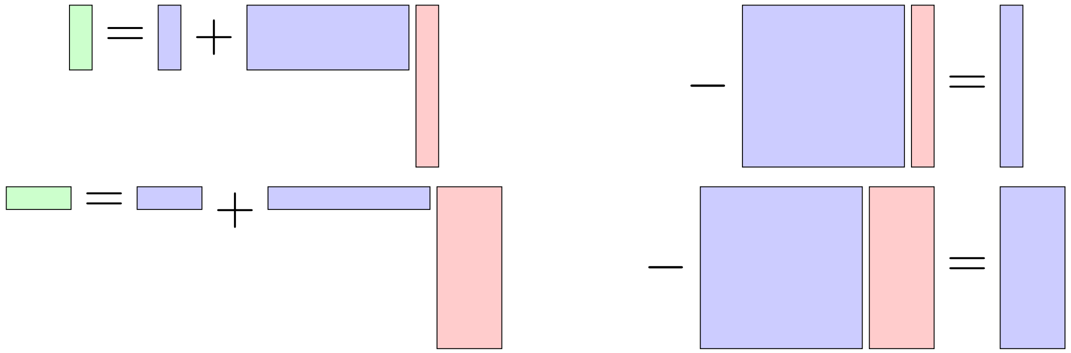
From reverse chain rule

$$\begin{bmatrix} \boldsymbol{I} & -\frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{F}^T}{\partial \boldsymbol{x}} \\ \boldsymbol{0} & -\frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{y}} & -\frac{\partial \boldsymbol{F}^T}{\partial \boldsymbol{y}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{x}} \\ \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{r}} \\ \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}$$



Direct method

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}} \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}} \qquad -\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{y}} \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}$$

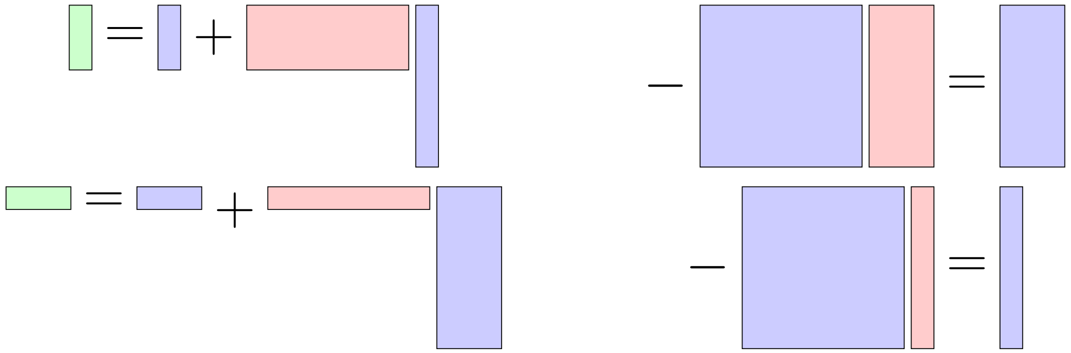


$n_x < n_f$

$n_x > n_f$

Adjoint method

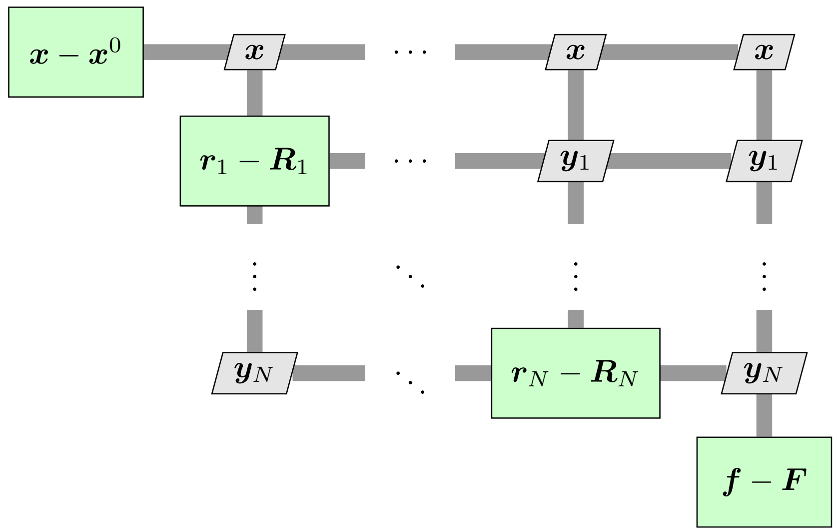
$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} + \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}} \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}} \qquad -\frac{\partial \boldsymbol{R}^T}{\partial \boldsymbol{y}} \frac{\mathrm{d}\boldsymbol{f}^T}{\mathrm{d}\boldsymbol{r}} = \frac{\partial \boldsymbol{F}^T}{\partial \boldsymbol{y}}$$



[Martins and Hwang, *AIAAJ*, 2013]

Coupled Analytic Methods: Residual Form

Variables and Constraints



$$
\boldsymbol{v} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_N \\ \boldsymbol{f} \end{bmatrix}
\qquad
\boldsymbol{C}(\boldsymbol{v}) = \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}^0 \\ \boldsymbol{r}_1 - \boldsymbol{R}_1(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_N) \\ \vdots \\ \boldsymbol{r}_N - \boldsymbol{R}_N(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_N) \\ \boldsymbol{f} - \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_N) \end{bmatrix}
$$

Derivation

$$
\left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right] = \boldsymbol{I} = \left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]^T\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]^T
$$

$$
\begin{bmatrix}
\boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\
-\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{y}_1} & \cdots & -\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{y}_N} & \boldsymbol{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
-\frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{y}_1} & \cdots & -\frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{y}_N} & \boldsymbol{0} \\
-\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_1} & \cdots & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_N} & \boldsymbol{I}
\end{bmatrix}
\begin{bmatrix}
\boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\
\frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{r}_1} & \cdots & \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{r}_N} & \boldsymbol{0} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{r}_1} & \cdots & \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{r}_N} & \boldsymbol{0} \\
\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}_1} & \cdots & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}_N} & \boldsymbol{I}
\end{bmatrix}
= \boldsymbol{I} =
\begin{bmatrix}
\boldsymbol{I} & -\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{x}}^T & \cdots & -\frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{x}}^T & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}^T \\
\boldsymbol{0} & -\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{y}_1}^T & \cdots & -\frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{y}_1}^T & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_1}^T \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\boldsymbol{0} & -\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{y}_N}^T & \cdots & -\frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{y}_N}^T & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_N}^T \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}
\end{bmatrix}
\begin{bmatrix}
\boldsymbol{I} & \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{x}}^T & \cdots & \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{x}}^T & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}^T \\
\boldsymbol{0} & \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{r}_1}^T & \cdots & \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{r}_1}^T & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}_1}^T \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\boldsymbol{0} & \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{r}_N}^T & \cdots & \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{r}_N}^T & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}_N}^T \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}
\end{bmatrix}
$$

Coupled direct: residual form

$$
\begin{bmatrix}
\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{y}_1} & \cdots & \frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{y}_N} \\
\vdots & \ddots & \vdots \\
\frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{y}_1} & \cdots & \frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{y}_N}
\end{bmatrix}
\begin{bmatrix}
\frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{x}} \\ \vdots \\ \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{x}}
\end{bmatrix}
= -
\begin{bmatrix}
\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{x}} \\ \vdots \\ \frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{x}}
\end{bmatrix}
$$

$$
\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} + \left[\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_1} \cdots \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_N}\right]
\begin{bmatrix}
\frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{x}} \\ \vdots \\ \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{x}}
\end{bmatrix}
$$

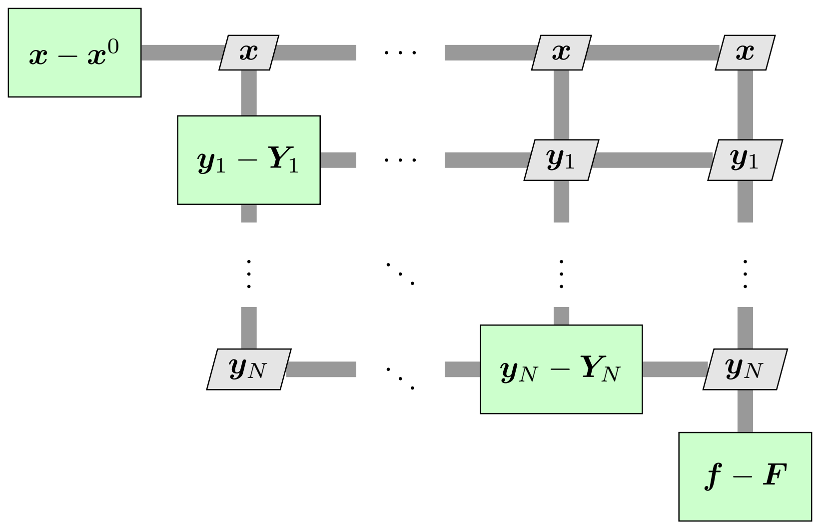
Coupled adjoint: residual form

$$
\begin{bmatrix}
\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{y}_1}^T & \cdots & \frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{y}_1}^T \\
\vdots & \ddots & \vdots \\
\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{y}_N}^T & \cdots & \frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{y}_N}^T
\end{bmatrix}
\begin{bmatrix}
\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}_1}^T \\ \vdots \\ \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}_N}^T
\end{bmatrix}
= -
\begin{bmatrix}
\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_1}^T \\ \vdots \\ \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_N}^T
\end{bmatrix}
$$

$$
\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} + \left[\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}_1} \cdots \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{r}_N}\right]
\begin{bmatrix}
\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{x}} \\ \vdots \\ \frac{\partial \boldsymbol{R}_N}{\partial \boldsymbol{x}}
\end{bmatrix}
$$

Coupled Analytic Methods: Functional Form

Variables and Constraints



$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_N \\ \boldsymbol{f} \end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{v}) = \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}^0 \\ \boldsymbol{y}_1 - \boldsymbol{Y}_1(\boldsymbol{x}, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N) \\ \vdots \\ \boldsymbol{y}_N - \boldsymbol{Y}_N(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N-1}) \\ \boldsymbol{f} - \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_N) \end{bmatrix}$$

Derivation

$$\left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]\left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right] = \boldsymbol{I} = \left[\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{v}}\right]^T \left[\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}\boldsymbol{c}}\right]^T$$

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \ldots & \boldsymbol{0} & \boldsymbol{0} \\ -\frac{\partial \boldsymbol{Y}_1}{\partial \boldsymbol{x}} & \boldsymbol{I} & \ldots & -\frac{\partial \boldsymbol{Y}_1}{\partial \boldsymbol{y}_N} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\frac{\partial \boldsymbol{Y}_N}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{Y}_N}{\partial \boldsymbol{y}_1} & \ldots & \boldsymbol{I} & \boldsymbol{0} \\ -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_1} & \ldots & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_N} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \ldots & \boldsymbol{0} & \boldsymbol{0} \\ \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{x}} & \boldsymbol{I} & \ldots & \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{y}_N} & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{y}_1} & \ldots & \boldsymbol{I} & \boldsymbol{0} \\ \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}_1} & \ldots & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}_N} & \boldsymbol{I} \end{bmatrix} = \boldsymbol{I} = \begin{bmatrix} \boldsymbol{I} & -\frac{\partial \boldsymbol{Y}_1}{\partial \boldsymbol{x}}^T & \ldots & -\frac{\partial \boldsymbol{Y}_N}{\partial \boldsymbol{x}}^T & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}^T \\ \boldsymbol{0} & \boldsymbol{I} & \ldots & -\frac{\partial \boldsymbol{Y}_N}{\partial \boldsymbol{y}_1}^T & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_1}^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & -\frac{\partial \boldsymbol{Y}_1}{\partial \boldsymbol{y}_N}^T & \ldots & \boldsymbol{I} & -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_N}^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{x}}^T & \ldots & \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{x}}^T & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}}^T \\ \boldsymbol{0} & \boldsymbol{I} & \ldots & \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{y}_1}^T & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}_1}^T \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{0} & \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{y}_N}^T & \ldots & \boldsymbol{I} & \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}_N}^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

Coupled direct: functional form

$$\begin{bmatrix} \boldsymbol{I} & \ldots & -\frac{\partial \boldsymbol{Y}_1}{\partial \boldsymbol{y}_N} \\ \vdots & \ddots & \vdots \\ -\frac{\partial \boldsymbol{Y}_N}{\partial \boldsymbol{y}_1} & \ldots & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{x}} \\ \vdots \\ \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \boldsymbol{Y}_1}{\partial \boldsymbol{x}} \\ \vdots \\ \frac{\partial \boldsymbol{Y}_N}{\partial \boldsymbol{x}} \end{bmatrix}$$

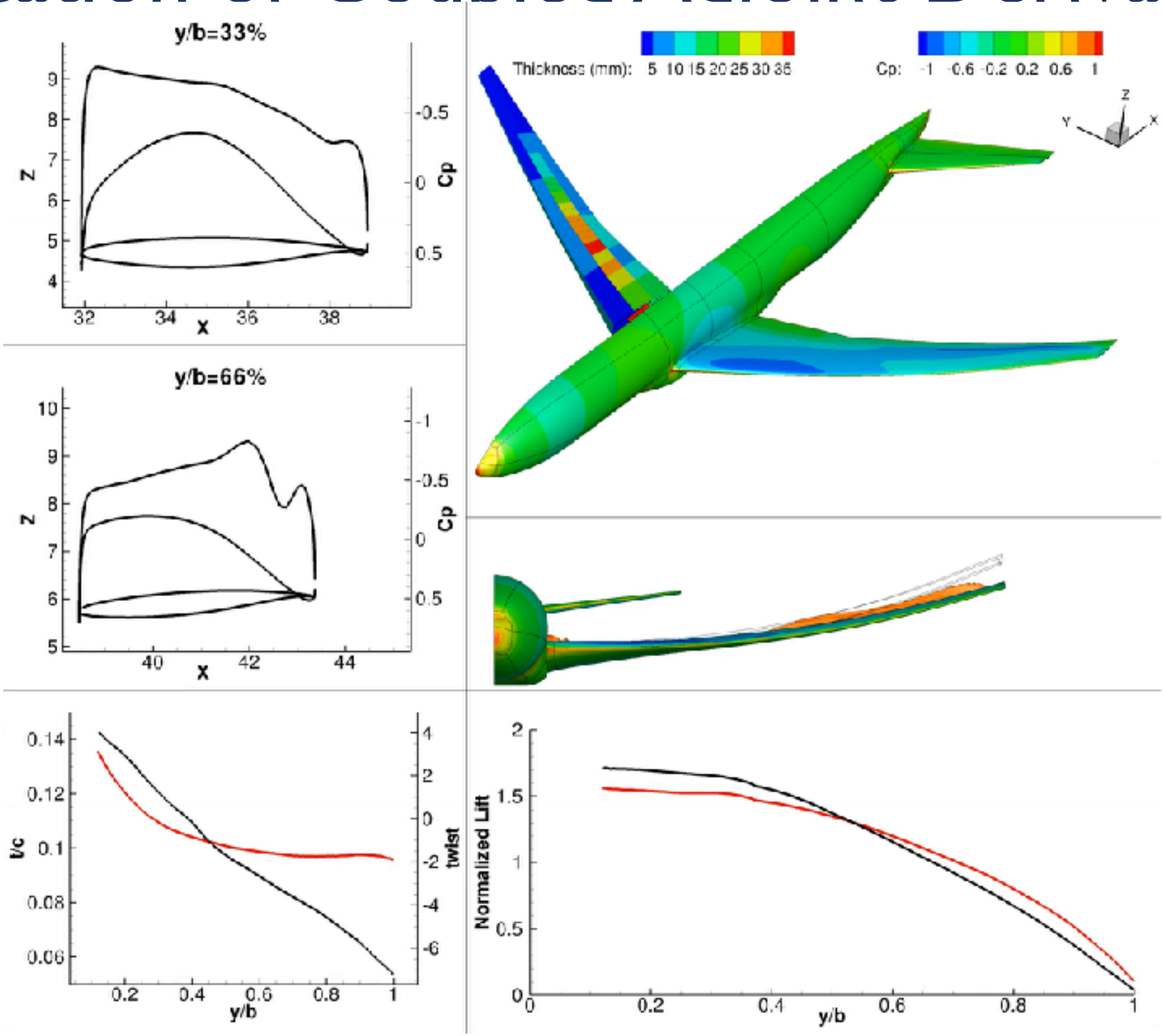
$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} + \begin{bmatrix} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_1} \ldots \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_N} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\boldsymbol{y}_1}{\mathrm{d}\boldsymbol{x}} \\ \vdots \\ \frac{\mathrm{d}\boldsymbol{y}_N}{\mathrm{d}\boldsymbol{x}} \end{bmatrix}$$

Coupled adjoint: functional form

$$\begin{bmatrix} \boldsymbol{I} & \ldots & -\frac{\partial \boldsymbol{Y}_N}{\partial \boldsymbol{y}_1}^T \\ \vdots & \ddots & \vdots \\ -\frac{\partial \boldsymbol{Y}_1}{\partial \boldsymbol{y}_N}^T & \ldots & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}_1}^T \\ \vdots \\ \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}_N}^T \end{bmatrix} = \begin{bmatrix} \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_1}^T \\ \vdots \\ \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{y}_N}^T \end{bmatrix}$$

$$\frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{x}} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} + \begin{bmatrix} \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}_1} \ldots \frac{\mathrm{d}\boldsymbol{f}}{\mathrm{d}\boldsymbol{y}_N} \end{bmatrix} \begin{bmatrix} \frac{\partial \boldsymbol{Y}_1}{\partial \boldsymbol{x}} \\ \vdots \\ \frac{\partial \boldsymbol{Y}_N}{\partial \boldsymbol{x}} \end{bmatrix}$$

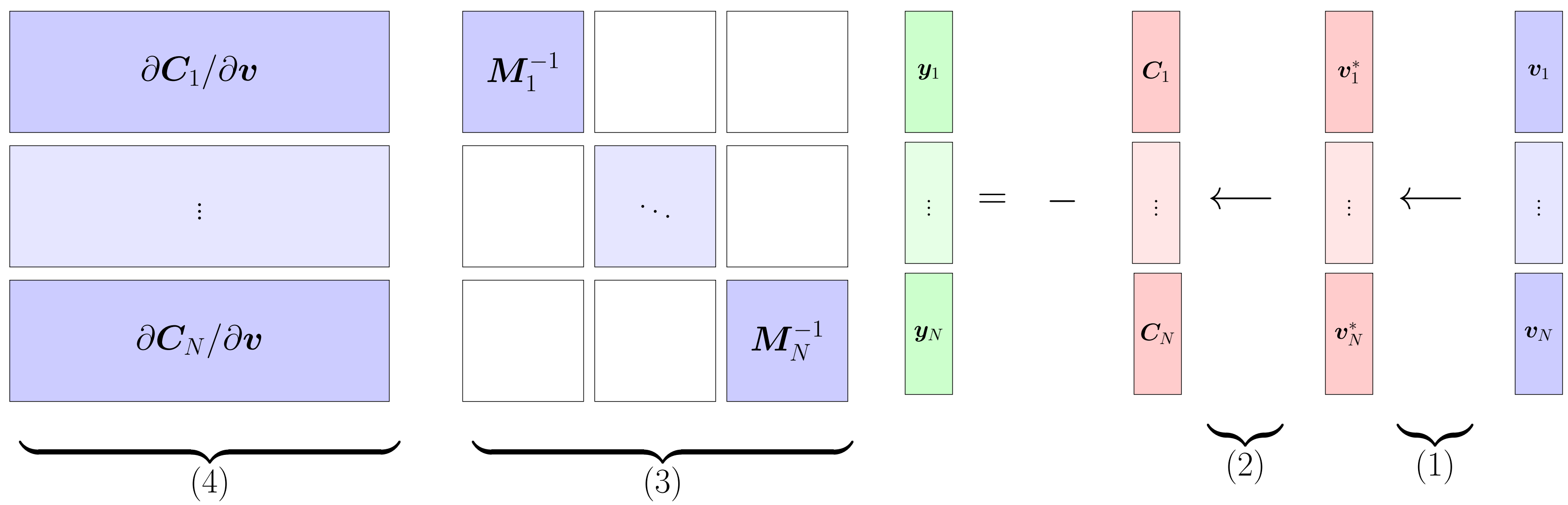
Application of Coupled Adjoint Derivatives



[Kenway, Kennedy and Martins, *AIAA Journal*, 2013]

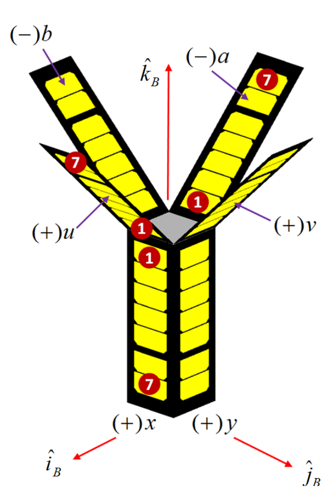
Application to MDO of small satellite

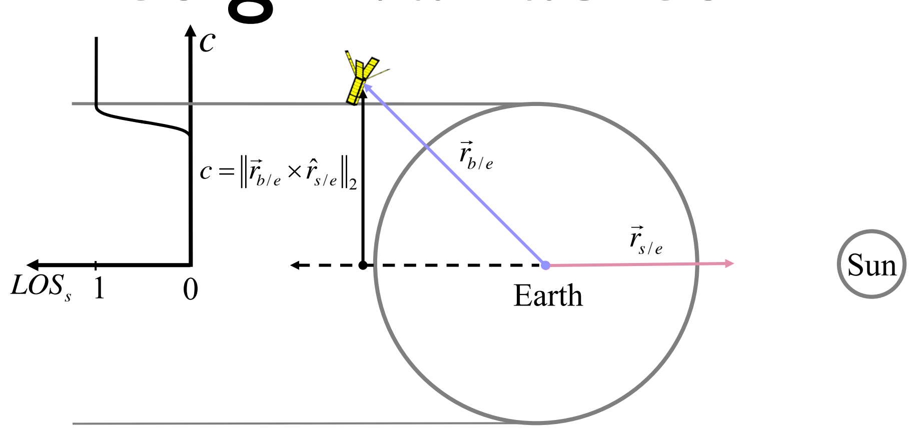
Computational framework for gradient-based MDAO

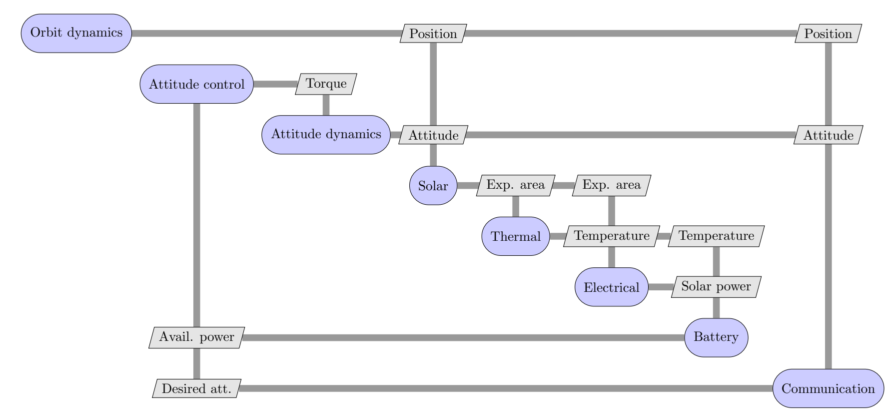


Component	*Jacobian* (4)	*Preconditioner* (3)	*solve* (1)
Triangular	Exact	Back subst.	Exact solve
Preconditioned	Exact	Preconditioner	No action
Factorized	Exact	Exact inverse	No action
Jacobian-free	Directional derivative		No action

Used to solve for millions of states and tens of thousands of design variables







[Hwang et al., *AIAA SDM*, 2013]

Further Reading

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

Martins and Lambe, "Multidisciplinary Design Optimization : A Survey of Architectures", *AIAAJ*, 2013 (In press)

Martins and Hwang, "Review and Unification of Discrete Methods for Computing Derivatives of Single- and Multi-disciplinary Computational Models", *AIAAJ*, 2013 (In press)

Other Relevant 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.