STAB Symposium, 6-7 November 2018

Cybermatrix: A novel approach to computationally and collaboration intensive MDO for transport aircraft design

Caslav Ilic, Andrei Merle, Arno Ronzheimer, Mohammad Abu-Zurayk, Jonas Jepsen, Martin Leitner, Matthias Schulze, Andreas Schuster, Michael Petsch, Sebastian Gottfried

Two backgrounds of MDO for aircraft design

- Background 1: classic aircraft design
  - Focus on process automation, many disciplines, data modeling
  - No specific focus on high-performance computing (HPC)
  - No formal optimality criteria, suboptimal designs by construction
- Background 2: mathematical optimization
  - Focus on analysis fidelity, modeling constraints and adding disciplines
  - Explicit consideration of optimality criteria and often high HPC use
  - Simplifed tools, poorly scalable in number of disciplines/experts
- **The present approach aims at balancing the two backgrounds**
  - Developed within the **DLR project VicToria**
  - Optimality criteria explicit but applied in a heuristic manner
  - Parallelism built in ground-up, in participation of experts and use of HPC
  - Assembly (human) and execution (computer) phases with analogous communication and control in a matrix-like structure → **cybermatrix**

Approximate optimal design

- Any design process can be viewed as an **approximate** optimization process:

$$\frac{d\widehat{f(w,p)}}{dp} - \lambda \frac{d\widehat{c(w,p)}}{dp} = 0, \quad c(w,p) = 0, \quad r(w,p) = 0$$

where *f* goal, *c* constraints, *r* consistencies (residuals), *w* states,
p design parameters, *λ* constraint scales (Lagrange multipliers)
→ approximate KKT optimality condition

- Expanded for three disciplines A, B, C and global goal function F:

$$\widehat{\frac{\partial F}{\partial f_A}}\underline{\widehat{\frac{df_A}{dp_A}}} + \widehat{\frac{\partial F}{\partial f_B}}\widehat{\frac{df_B}{dp_A}} + \widehat{\frac{\partial F}{\partial f_C}}\widehat{\frac{df_C}{dp_A}} - \underline{\lambda_A \widehat{\frac{dc_A}{dp_A}}} - \lambda_B \widehat{\frac{dc_B}{dp_A}} - \lambda_C \widehat{\frac{dc_C}{dp_A}} = 0,\ \underline{c_A}=0,\ \underline{r_A}=0$$

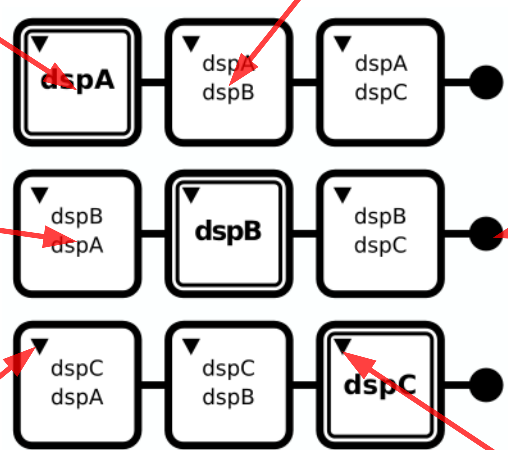
$$\widehat{\frac{\partial F}{\partial f_A}}\widehat{\frac{df_A}{dp_B}} + \widehat{\frac{\partial F}{\partial f_B}}\underline{\widehat{\frac{df_B}{dp_B}}} + \widehat{\frac{\partial F}{\partial f_C}}\widehat{\frac{df_C}{dp_B}} - \lambda_A \widehat{\frac{dc_A}{dp_B}} - \underline{\lambda_B \widehat{\frac{dc_B}{dp_B}}} - \lambda_C \widehat{\frac{dc_C}{dp_B}} = 0,\ \underline{c_B}=0,\ \underline{r_B}=0$$

$$\widehat{\frac{\partial F}{\partial f_A}}\widehat{\frac{df_A}{dp_C}} + \widehat{\frac{\partial F}{\partial f_B}}\widehat{\frac{df_B}{dp_C}} + \widehat{\frac{\partial F}{\partial f_C}}\underline{\widehat{\frac{df_C}{dp_C}}} - \lambda_A \widehat{\frac{dc_A}{dp_C}} - \lambda_B \widehat{\frac{dc_B}{dp_C}} - \underline{\lambda_C \widehat{\frac{dc_C}{dp_C}}} = 0,\ \underline{c_C}=0,\ \underline{r_C}=0$$

DLR

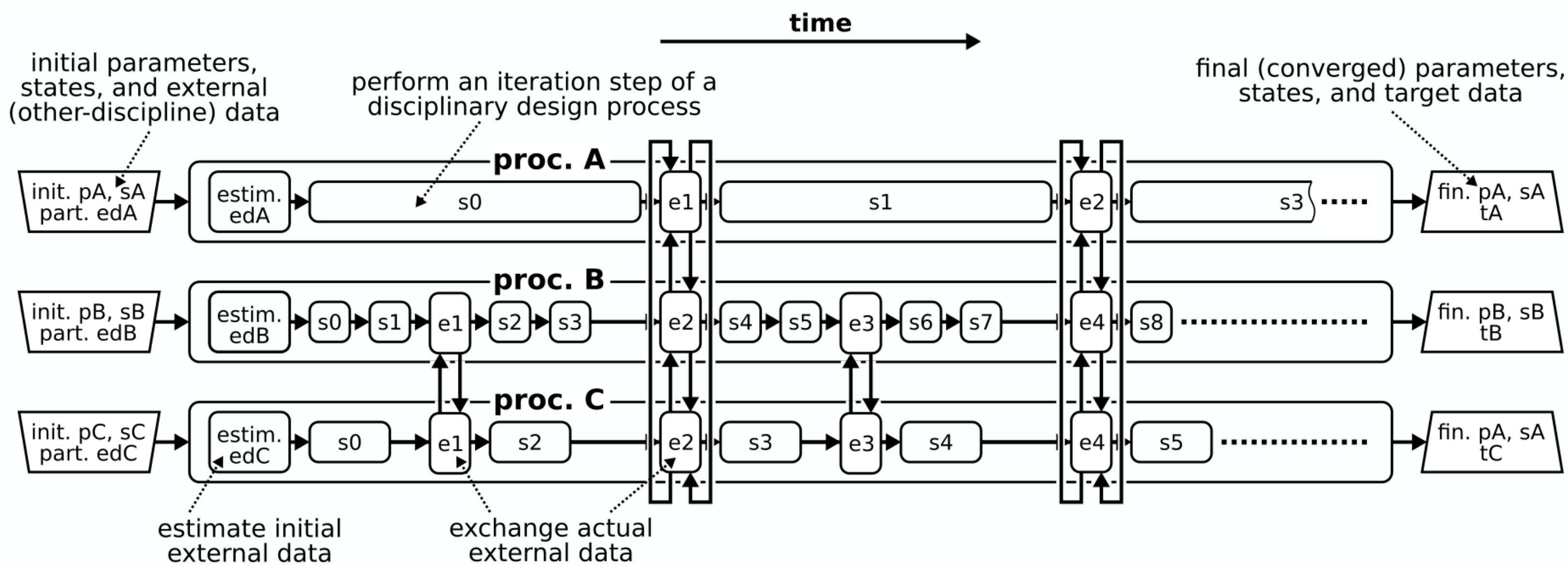
Cybermatrix representation

- Since the design equation is usually implied, use a schematic representation
- Let each row belong to one discipline (all related to its design parameters)



Communication protocol

- Each disciplinary design process can have any form, so long as **iterative**
- Equip it additionally with **data exchange points** and **initial data estimator**



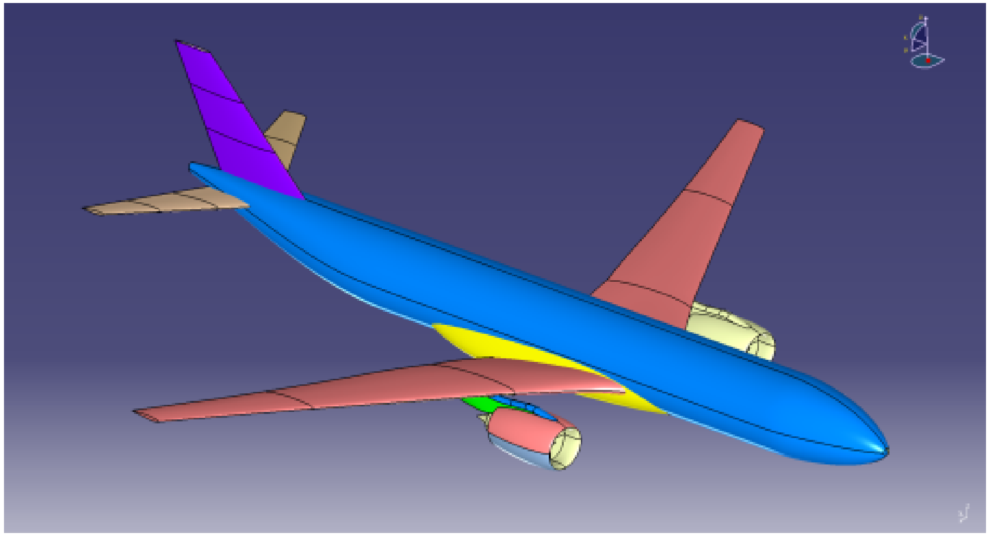
- Different disciplines may have different exchange periods
- Selection of rows and exchange periods recover **all known MDO architectures**
  - **In practical cases always a hybrid architecture**

A realization on HPC clusters

- A cybermatrix process integration framework for HPC clusters in development
  - Starts disciplinary processes, assignes resources, monitors progress
  - Triggers data exchanges and determines global convergence
- Disciplinary experts do not work with the framework directly
  - No need to learn yet another integration framework
  - Only provide **input collector** scripts to copy data from other disciplines
  - The whole MDO process implementation: a directory of input collectors
- Maintainable by standard software engineering tools and practices
  - Set of input collectors under source version control
  - Integration framework is an **interpreter** of the set of collectors and some meta-data (data exchange periods, etc)
- Currently data exchange performed over parallel on-disk file system
  - Parallel in-memory or area-network file system possible in principle
  - No changes to disciplinary processes in any case

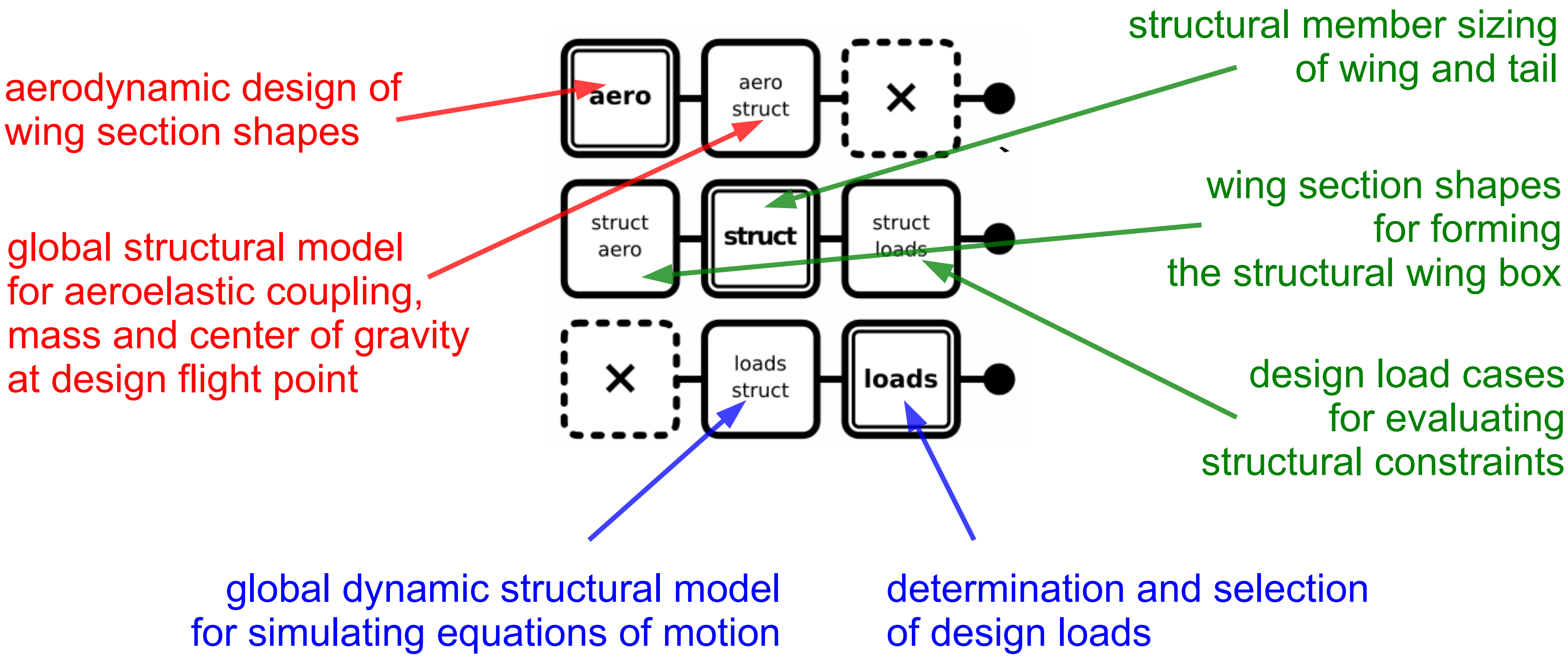
Example: MDO of long-range transport aircraft

- Configuration:
  twin-engine wide-body
  long-range transport aircraft
  - Wing-body-tail-pylon-flow through nacelle
  - 250 t max. take-off mass class
- Global goal function:
  minimize fuel consumption
- Constraints: all local
  (assigned to disciplines)



- Involved disciplinary processes:
  - Aerodynamic design of wing section shape (**aero**)
  - Structural member sizing of wing and tail (**struct**)
  - Determination and evaluation of design loads (**loads**)

Example: Matrix setup

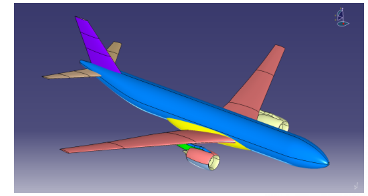


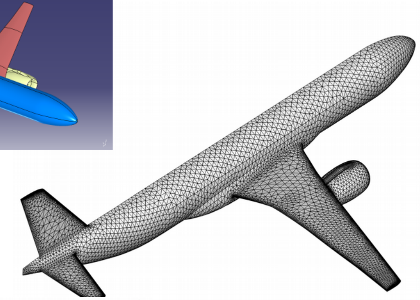
Only consistency dependencies, no design dependencies

Example: Disciplinary subprocesses

- **aero**:

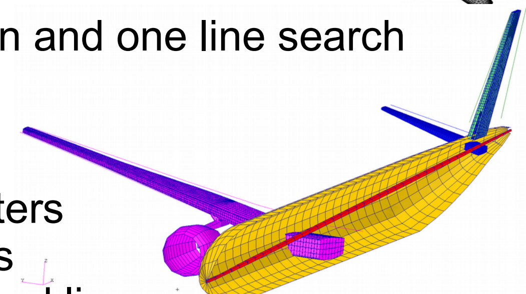
  Adjoint gradient-based static-aeroelastic optimization
  Design point: M = 0.83, h = 11000 m
  CAD+ROM B-spline airfoil definition, 126 design parameters
  Hybrid-unstructured RANS CFD mesh, 544,000 pts, 1,130,000 els
  Goal: minimize drag; constraints: trimmed flight
  Between data exchanges: one gradient evaluation and one line search

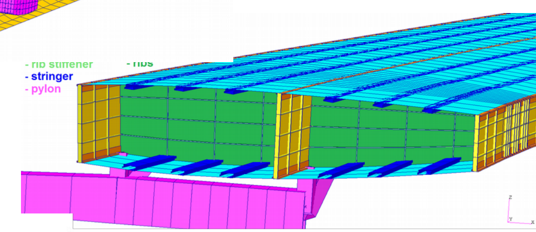




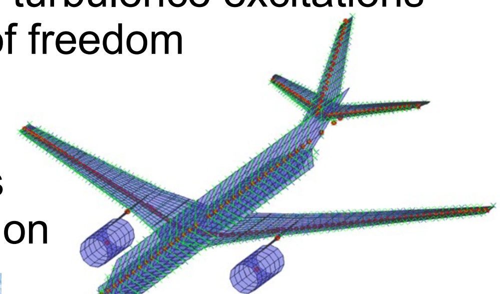
- **struct**:

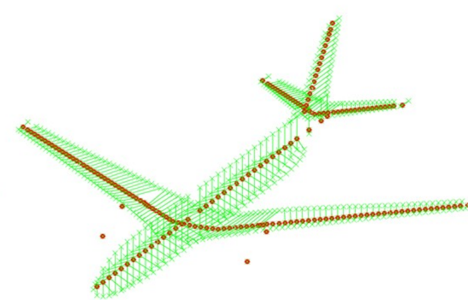
  Gradient-based sizing of structural regions
  Referent region thicknesses, 364 design parameters
  Global FE model, 18,000 nodes, 42,000 elements
  Goal: minimize mass; constraints: strength and buckling
  Between data exchanges: one full sizing



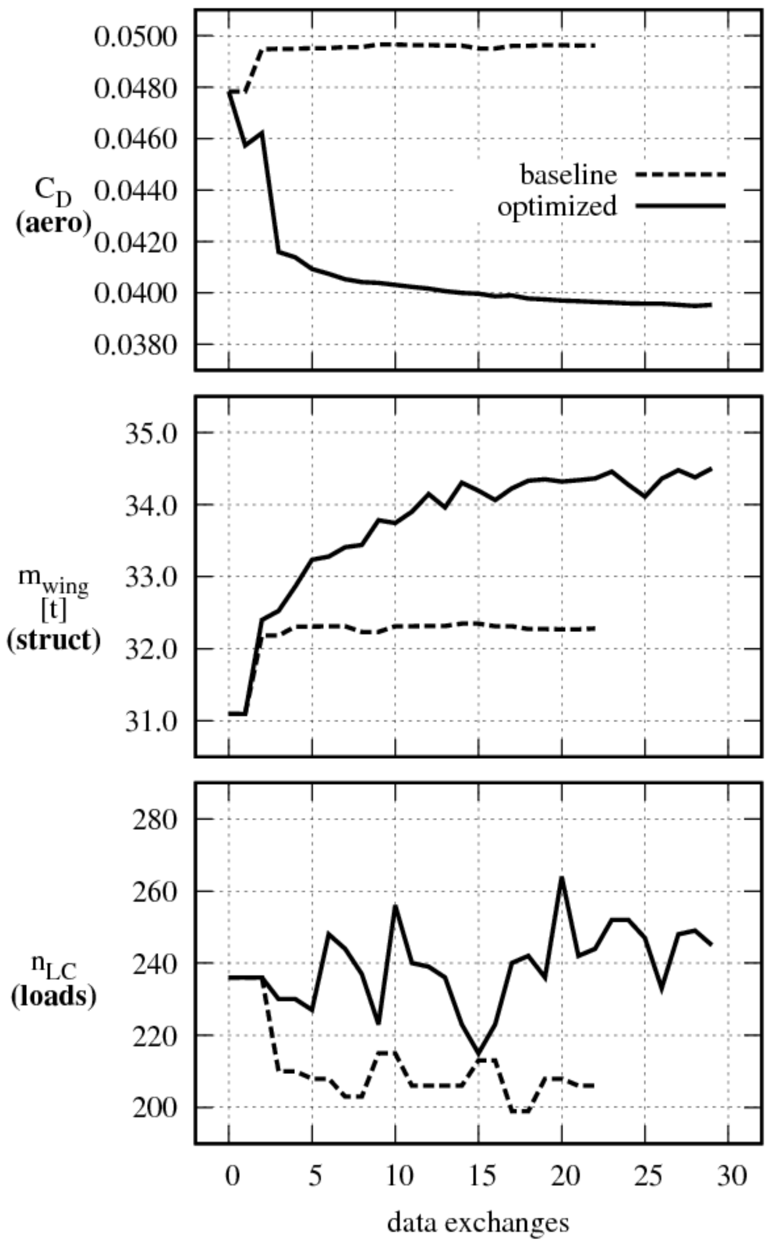


- **loads**:
  Transient dynamic simulations of gust and turbulence excitations
  Dynamic structural model, 1068 degrees of freedom
  Panel aerodynamic model, 1163 boxes
  1284 load cases and 2 mass cases
  No goal/constraints, no design parameters
  Between data exchanges: one full evaluation





Example: Optimization result

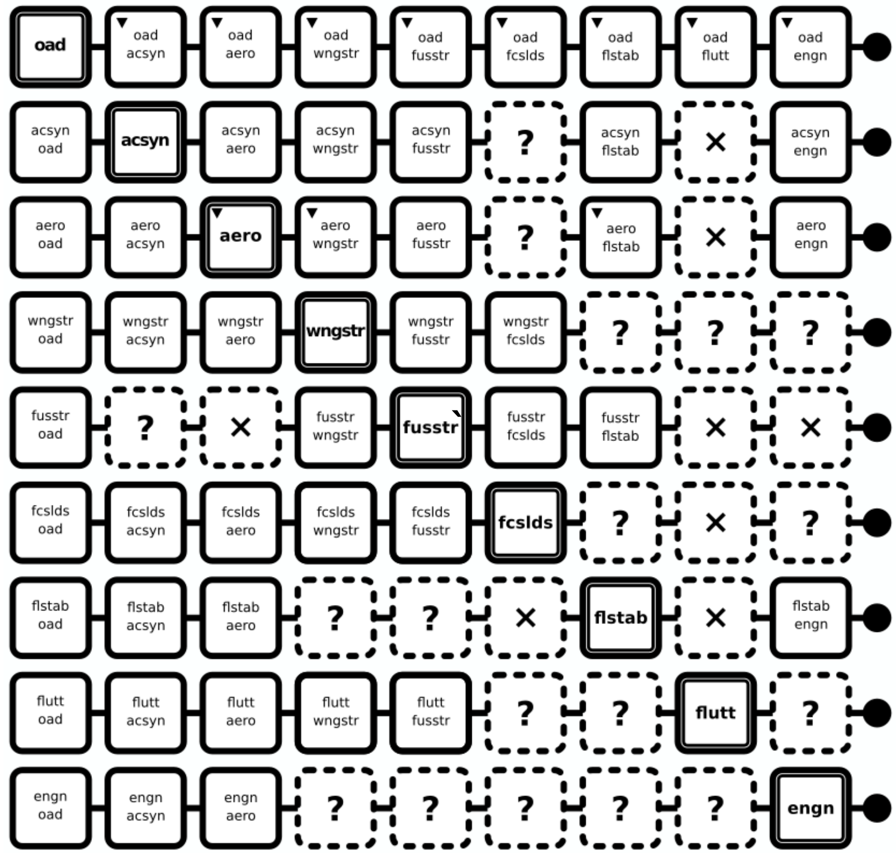


- Total run time:**110 hours** on **64 cores**
  - Time between data exchanges: 4.1 h avg
  - Set by **aero** on 48 c; **struct** 2.2 h on 4 c; **loads** 1.2 h on 12 c
- Main effect: high drag reduction (**-16%**) for modest increase in mass (**4.6%** wing, **0.8%** total)
  - Wing sections slightly retwisted and reshaped to reduce shocks (drag level high due to coarse mesh)
  - Somewhat less favorable spanwise load distribution results in higher design loads
  - Variation in number of design load cases not large, but not negligible
- What is the baseline for comparison?
  - "0" on data exchange axis has no meaning
  - Intention-dependent: here result of an optimization with fixed aerodynamic des. par.
- Global goal function (fuel consumption) not explicitly considered in any discipline
  - A missing design dependency
  - Local goal functions may increase after data exchanges

Conclusions and Outlook

- A core of a cybermatrix-based MDO process demonstrated
  - Aero-structural approximate optimization with variable number of design load cases
  - CAD-based shape parametrization through reduced order modeling
  - Realistic loads process following certification regulations
- Improvement to the core process
  - More flight points and powerd engine for aerodynamic design
  - Control laws and high-fidelity corrections for loads
  - Design dependencies (Jacobian-like information)
- Beyond the core process
  - Higher fidelity structural modeling (separate wing/fuselage disciplines)
  - Tighter geometry and mass synthesis (aircraft synthesis discipline)
  - Modification of wing planform shape (overall aircraft design discipline)
  - Flutter analysis (eliminate planforms exhibiting inherent flutter)

Thank you for your attention!



this time
next year
(more-or-less)