Mphys: Standardizing High-fidelity Optimization with OpenMDAO

Kevin Jacobson
NASA Langley Research Center

2022 OpenMDAO Workshop

State of High-fidelity Coupling in MDAO

Industry regularly performs single discipline high-fidelity modeling in MDAO

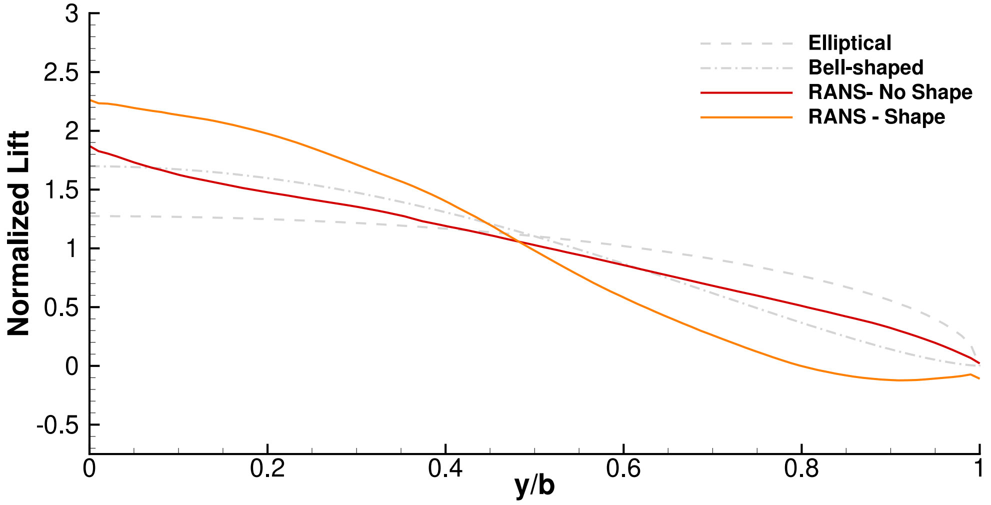
- Sequential aerodynamic shape optimization and structural sizing

State-of-the-art research shows benefit of coupled high-fidelity sensitivities

- Let the optimizer see the interactions of disciplines
- Coupling 2 or 3 disciplines
- Small subset of required design conditions and constraint types

Everyone in high-fidelity MDAO does their own implementation

- Difficult to implement and extend
- Difficult to transition research developments to industry users



$$
\begin{aligned}
\frac{df}{d\mathbf{x}} = &\left\{ \frac{\partial f}{\partial \mathbf{x}_{A,0}} + \sum_{k=1}^{N} \left[\left(\psi_D^{(k)}\right)^T \frac{\partial \mathbf{D}^{(k)}}{\partial \mathbf{x}_{A,0}} + \left(\psi_R^{(k)}\right)^T \frac{\partial \mathbf{R}^{(k)}}{\partial \mathbf{x}_{A,0}} + \left(\psi_E^{(k)}\right)^T \frac{\partial \mathbf{E}^{(k)}}{\partial \mathbf{x}_{A,0}} \right. \right. \\
&\left. \left. + \left(\psi_L^{(k)}\right)^T \frac{\partial \mathbf{L}^{(k)}}{\partial \mathbf{x}_{A,0}} + \left(\psi_G^{(k)}\right)^T \frac{\partial \mathbf{G}^{(k)}}{\partial \mathbf{x}_{A,0}} \right] + \left(\psi_G^{(0)}\right)^T \frac{\partial \mathbf{G}^{(0)}}{\partial \mathbf{x}_{A,0}} \right\} \frac{\partial \mathbf{x}_{A,0}}{\partial \mathbf{x}} \\
&+ \left\{ \frac{\partial f}{\partial \mathbf{x}_{S,0}} + \sum_{k=1}^{N} \left[\left(\psi_D^{(k)}\right)^T \frac{\partial \mathbf{D}^{(k)}}{\partial \mathbf{x}_{S,0}} + \left(\psi_L^{(k)}\right)^T \frac{\partial \mathbf{L}^{(k)}}{\partial \mathbf{x}_{S,0}} + \left(\psi_S^{(k)}\right)^T \frac{\partial \mathbf{S}^{(k)}}{\partial \mathbf{x}_{S,0}} \right] + \left(\psi_S^{(0)}\right)^T \frac{\partial \mathbf{S}^{(0)}}{\partial \mathbf{x}_{S,0}} \right\} \frac{\partial \mathbf{x}_{S,0}}{\partial \mathbf{x}}
\end{aligned}
$$

High-fidelity MDAO

CFD Vision 2030 - roadblocks to overcome:

- Computational cost
- Automation of high-fidelity modeling
  - » Mesh generation, lack of robustness of coupling methods, …
- **Time and expertise required to develop high-fidelity MDAO capability**
  - » **CFD-based MDAO limited by “one-off laborious, nonstandard interfaces”**

Mphys: An OpenMDAO Library

The goals are:

1. Make high-fidelity coupling in MDAO more accessible
2. Create a modular high-fidelity MDAO environment
   - Standardize interfaces and automate coupling

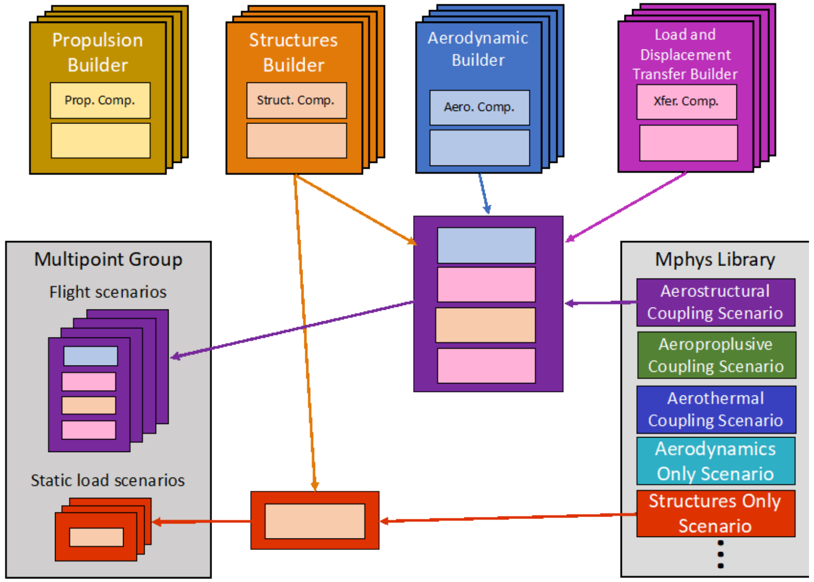
Mphys is a collaborative effort

» Open-source package developed with biweekly telecons

» Participants from government, academia, and industry

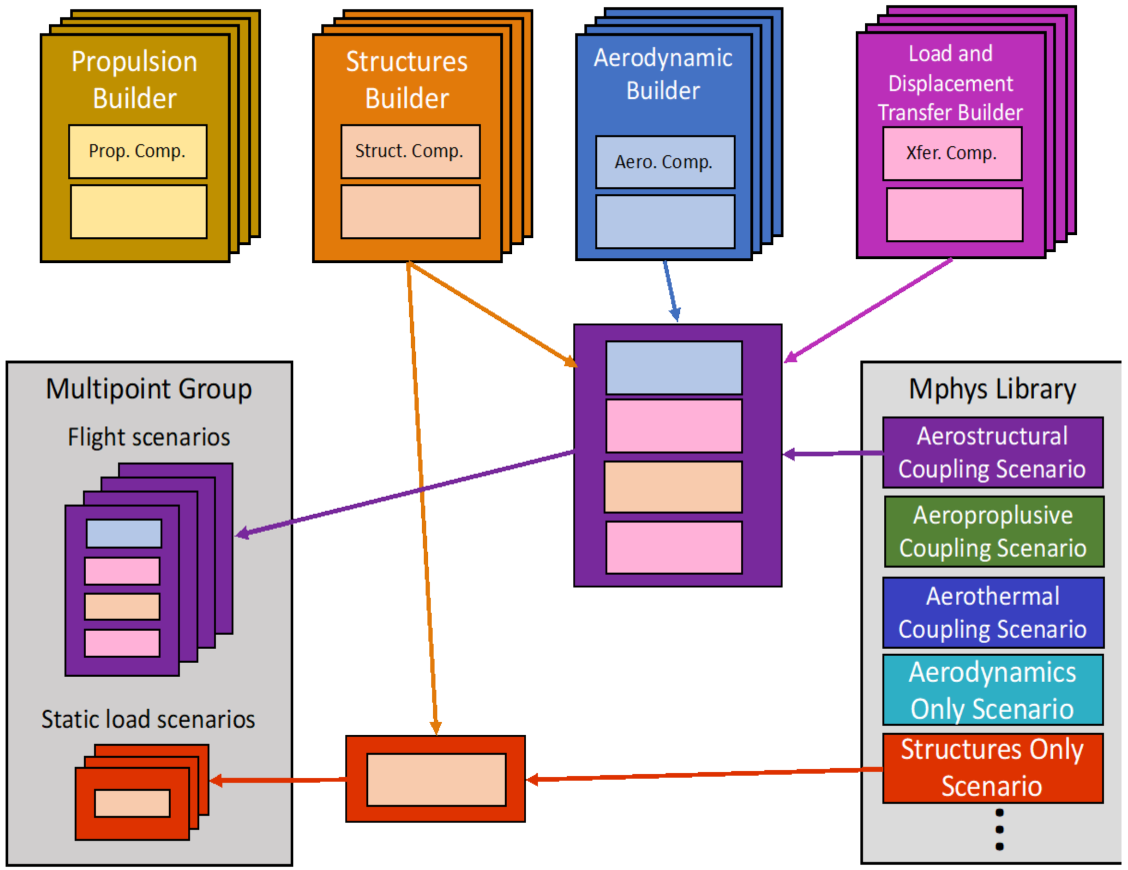
The Mphys package provides:

1. A library of standardized multiphysics problems for OpenMDAO
2. Utilities to set up optimization problems



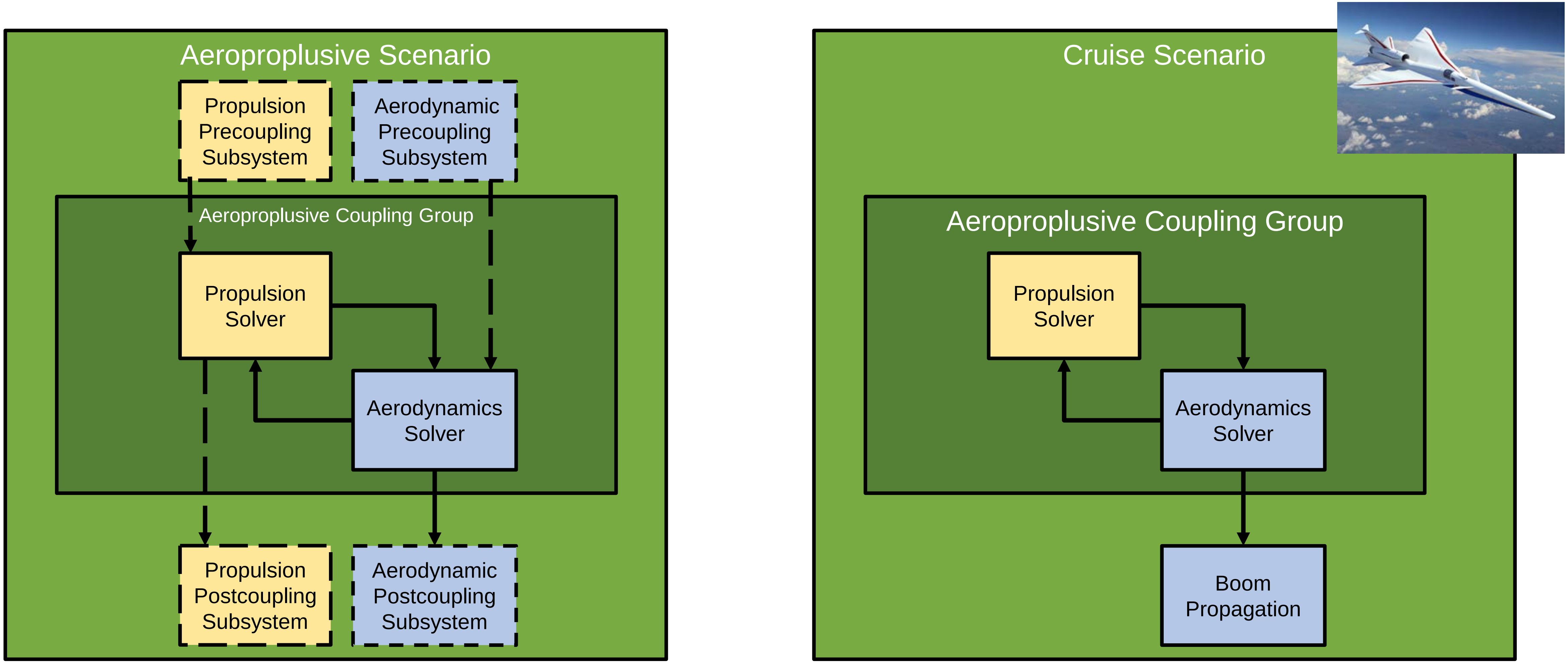
Mphys Models

- **Mphys provides a library of standardized multiphysics Scenarios for OpenMDAO**
- **Multipoint Group holds Scenarios defining the model**
- **Builders are associated with a disciplinary solver**
  - Produce OpenMDAO subsystems to populate the Scenario
- **Model Assembly Process**
  1. Pick which builders to use (select disciplinary solvers)
  2. Give these builders to desired Scenarios
  3. Add Scenarios to the Multipoint Group



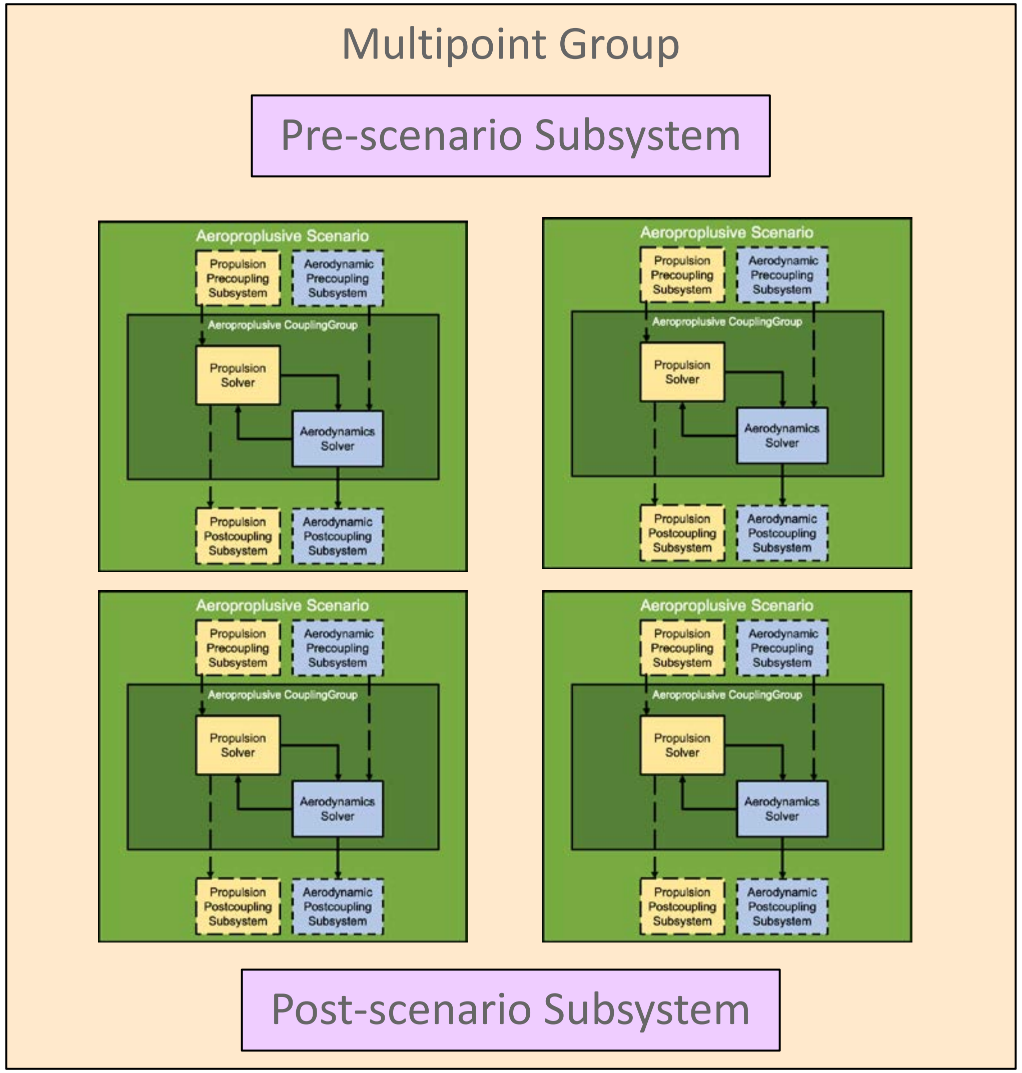
Mphys Model Hierarchy

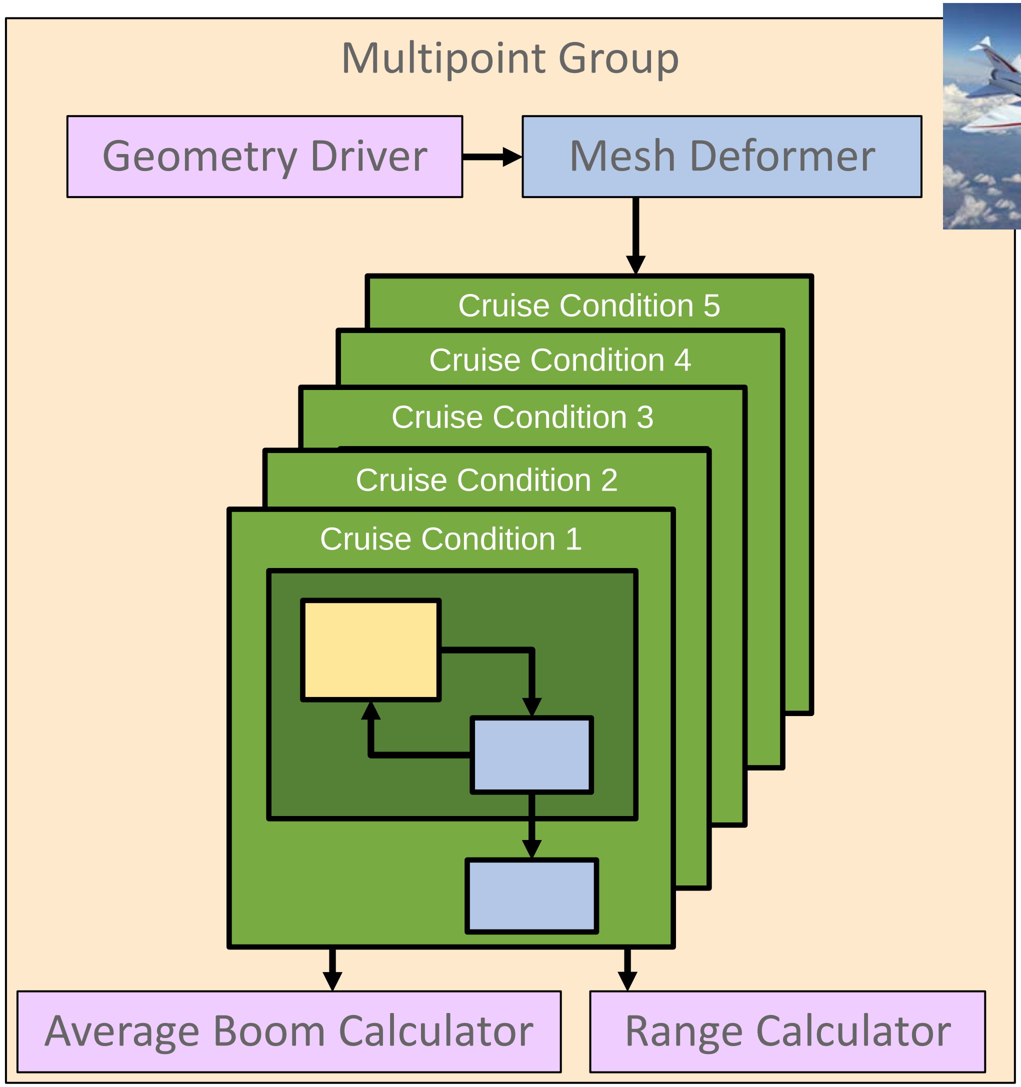
The Scenario groups hold components associated with a particular analysis condition



Mphys Model Hierarchy

Multipoint Groups hold a set of Scenarios and subsystems that interact with multiple Scenarios





How Mphys works under the hood

- **Coupling variable named according to the Mphys standard**
- **Variables to be connected outside of a discipline’s subsystem are tagged**
  - “mphys_coupling”, “mphys_coordinates”, “mphys_input”, or “mphys_result”

```
self.add_input('aoa',  desc = 'Angle of attack', units='deg', tags=['mphys_input'])
```

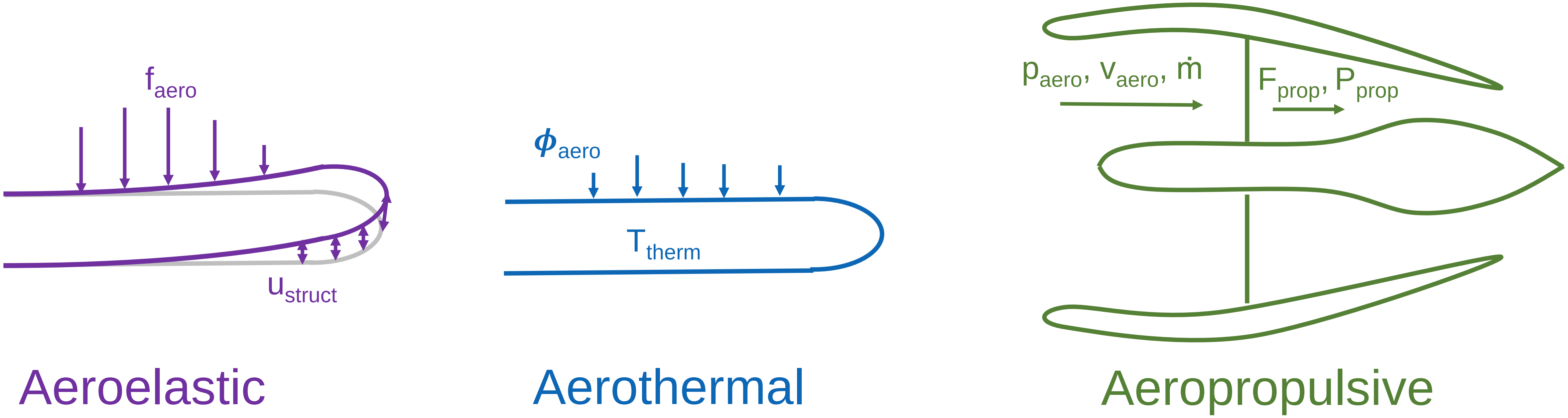
- **MphysGroup.mphys_add_subsystem()**
  - Same as standard Group.add_subsystem() but tracks subsystem to promote tagged variables during configure

```
self.mphys_add_subsystem('disp_xfer', disp_xfer)
self.mphys_add_subsystem('geo_disp', geo_disp)
self.mphys_add_subsystem('aero', aero)
self.mphys_add_subsystem('load_xfer', load_xfer)
self.mphys_add_subsystem('struct', struct)
```

- **The CouplingGroup and Scenario use this tagged promotion**
  - Forms connections within the scenario
  - Creates consistent access for inputs and results
    - » Raise inputs and results to Scenario level for easy and single-point connection by user: “cruise.aoa”

Standardized Interfaces in Mphys

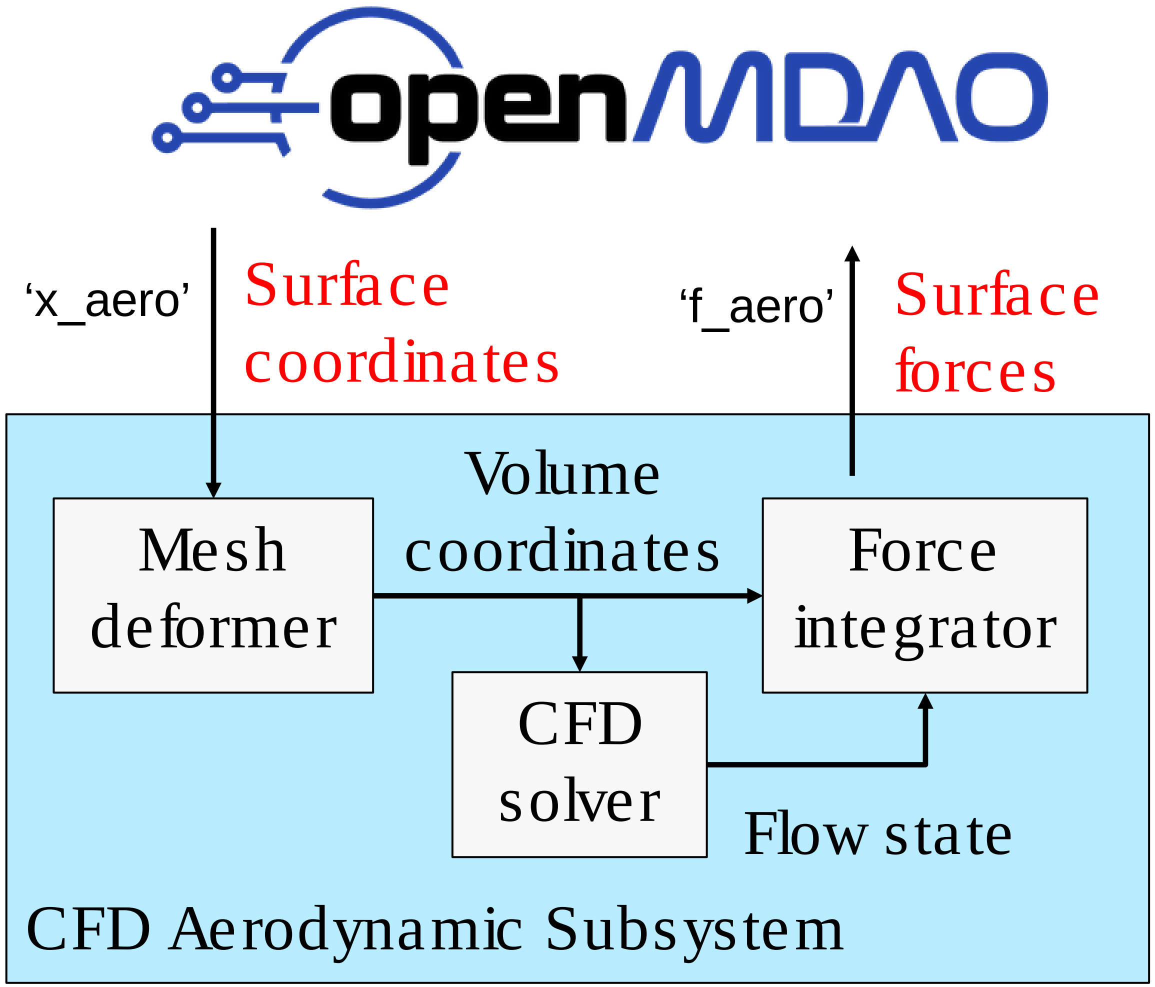
- **Types of coupled problems we're looking at so far:**

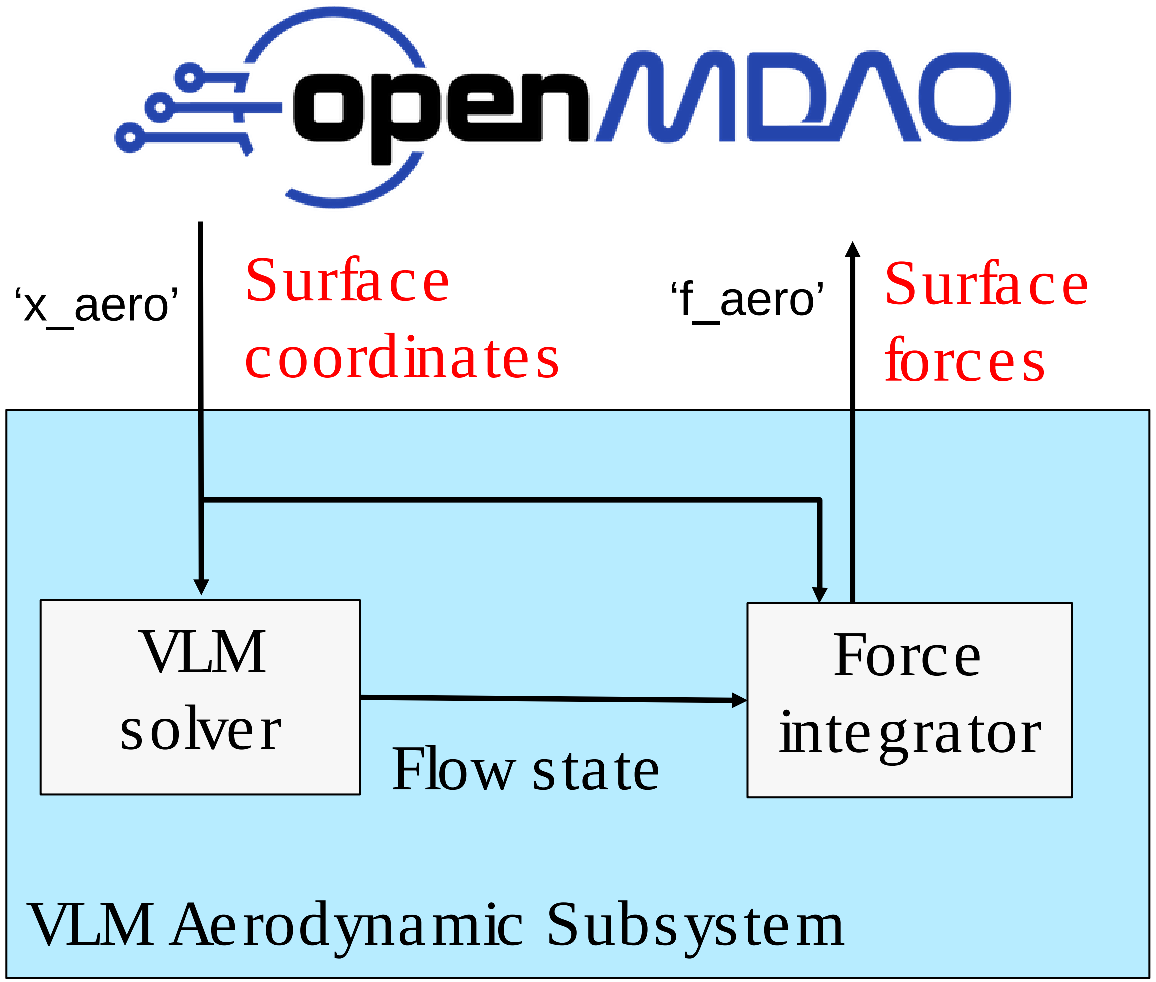


- **Interfaces are defined as vectors of coupling states at domain boundaries**
  - Projection of these vectors from one mesh to another is handled by transfer components when necessary

Standardization of Interface Definitions

- **Minimum set of data to allow as many solvers to fit interface as possible**





Modularity with Mphys

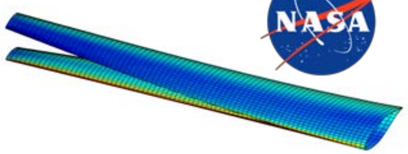
Flow Solvers:
FUN3D (SFE, FV*)
Cart3D*
ADflow
SU2*
DAFOAM
VLM
OpenAeroStruct VLM

Structural Solvers:
TACS
pyshell
Modal Structural Solver
NASTRAN*

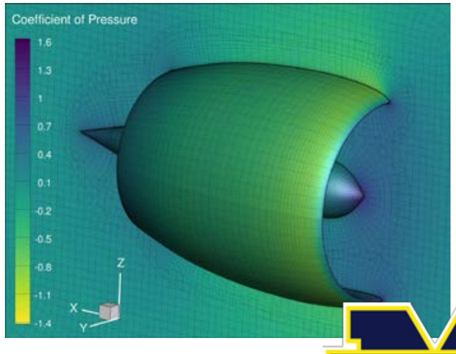
Propulsion:
pyCycle

Load and Displacement Transfers:
FUNtoFEM (MELD, RBF)
Rigid Link Transfer

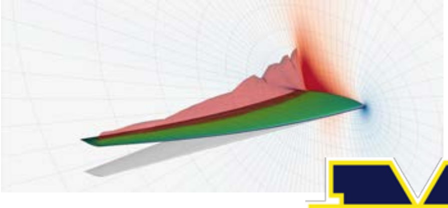
Geometry:
pyGeo (wraps OpenVSP, ESP, FFD)



VLM+MELD+TACS

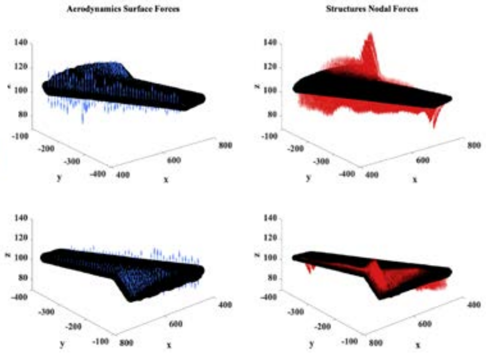


ADflow+pyCycle

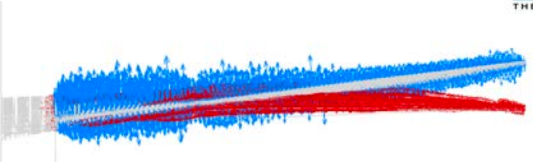


ADflow+RLT+TACS

FUN3D+MELD+TACS



SU2+MELD+NASTRAN

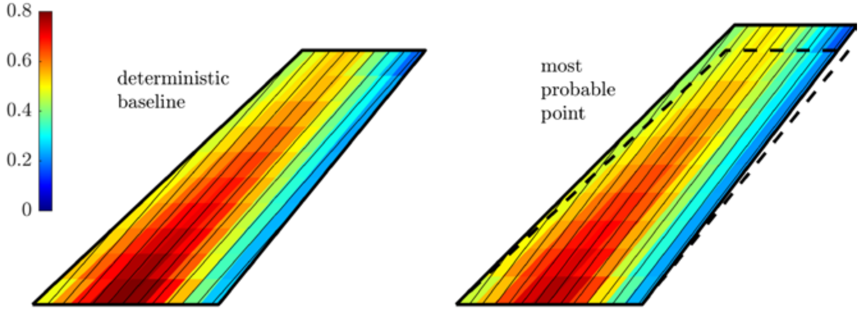


SU2+MELD+TACS

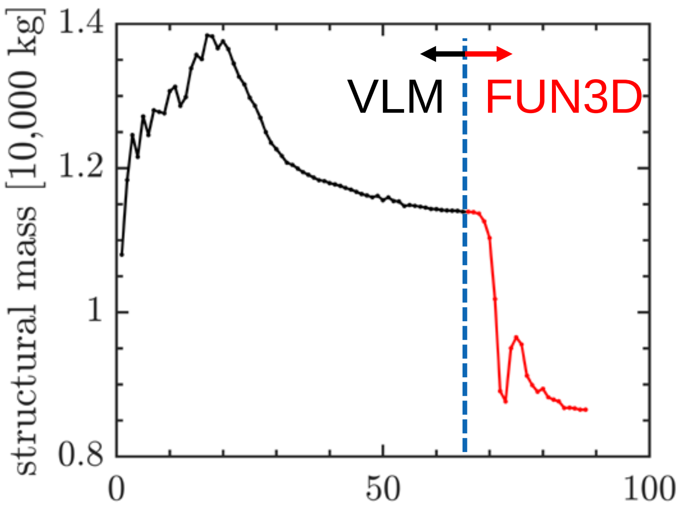
*Partially integrated

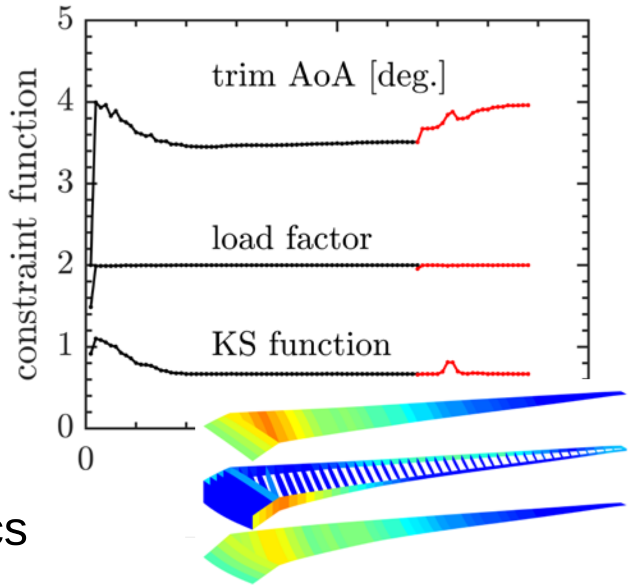
What can you do with the modularity?

1. **Easier transfer of technology**
2. **Work out high-fidelity optimization kinks with lower fidelities**
   - Scaling of DVs
   - Relative weight of competing objectives
   - Identify missing constraints
3. **Reduce cost of optimization (use HiFi only when necessary)**
   - Sequence physics during optimization
   - Multifidelity optimization
4. **UQ – Gradient-enhanced Entropy Contour Location**

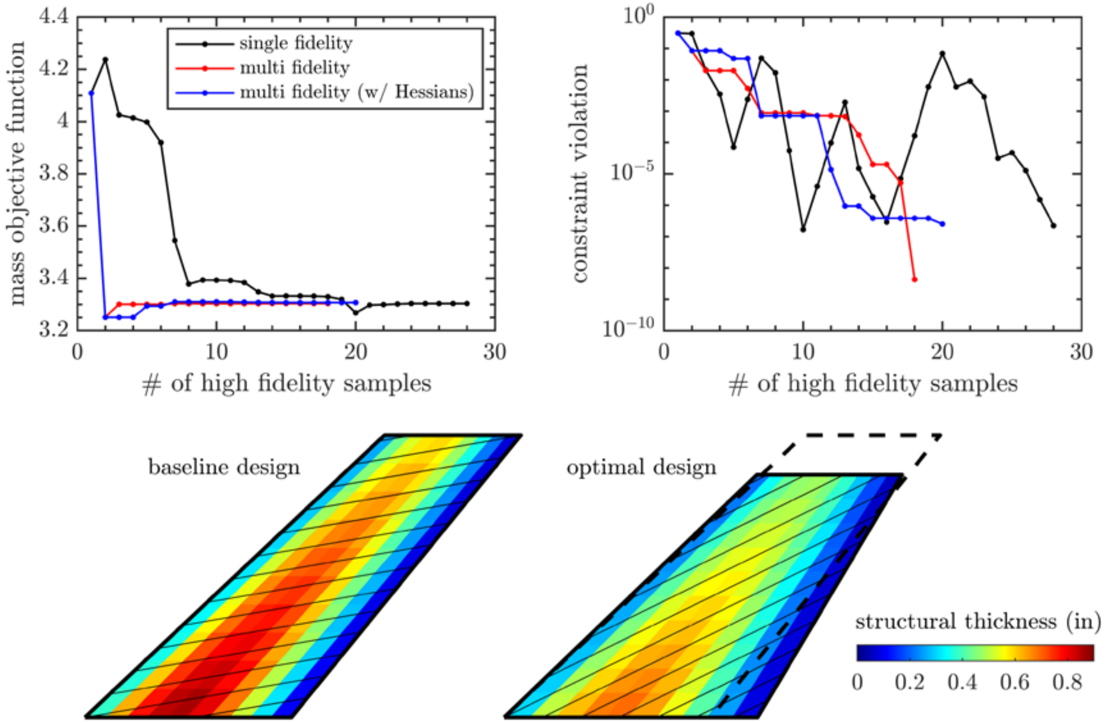


Gradient-enhanced Flutter UQ





Sequencing of physics



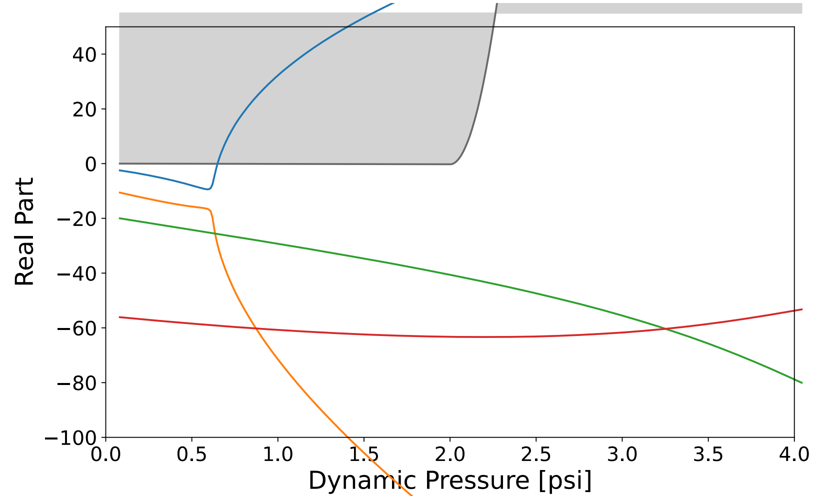
Multifidelity flutter-constrained optimization

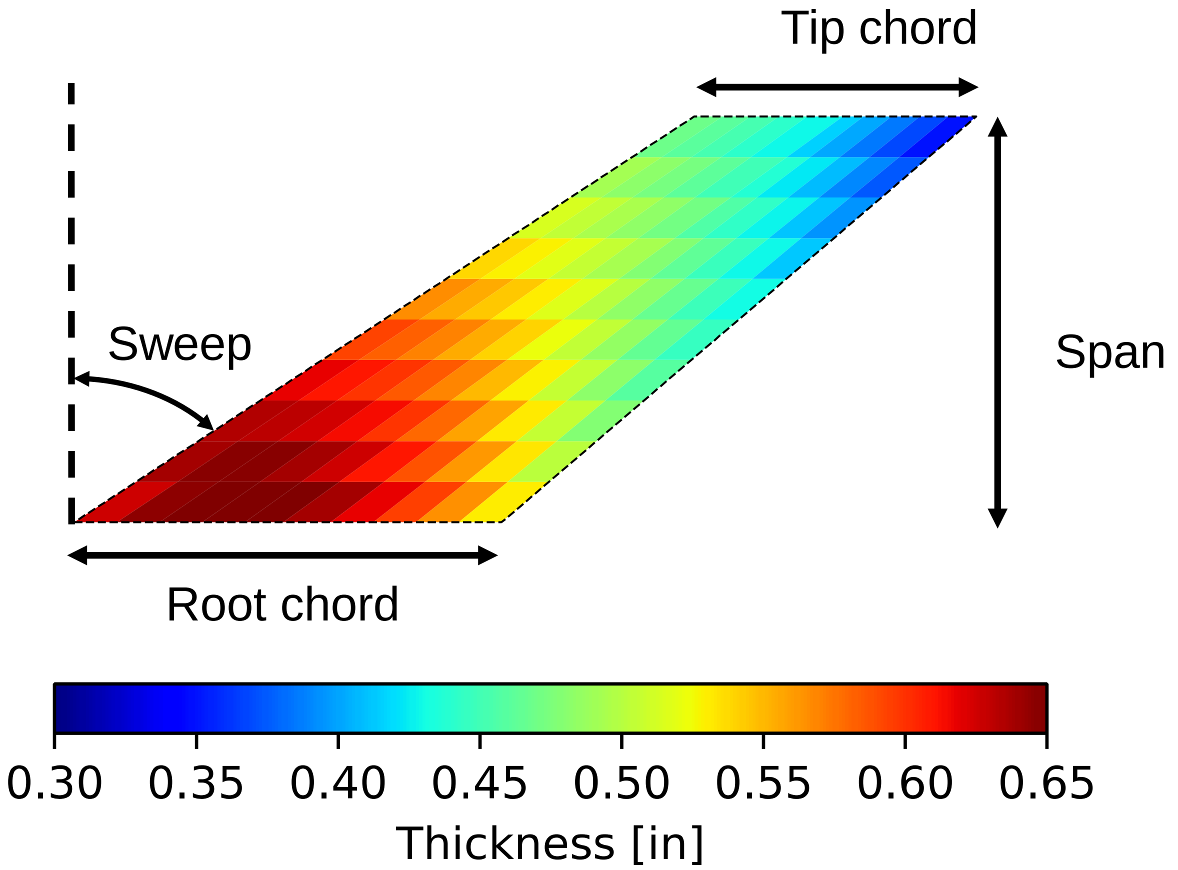
AGARD 445.6 Wing – Optimization Problem

- **Inviscid, symmetric wing at Mach 0.5**
  - Performed with Linearized Frequency Domain CFD (LFD) and Doublet Lattice Method (DLM) flutter analysis
- **Minimize the mass subject to flutter constraint and fixed planform area**

$$\begin{aligned} \min \quad & mass \\ \text{subject to:} \quad & KS_{flutter} \leq 0 \\ & A = A_0 \end{aligned}$$

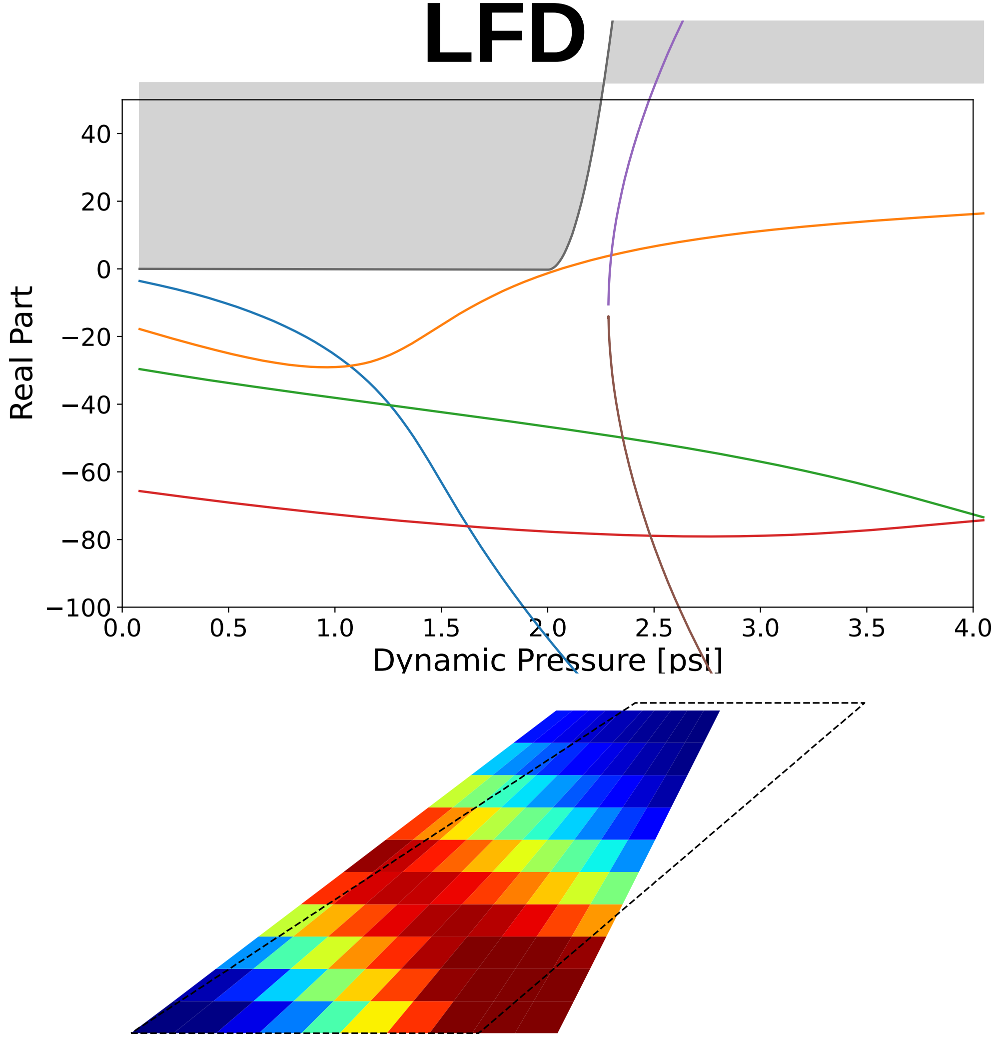
- **104 design variables**
  - 100 structural thicknesses
  - 4 geometric (affect aero. and struct. meshes)



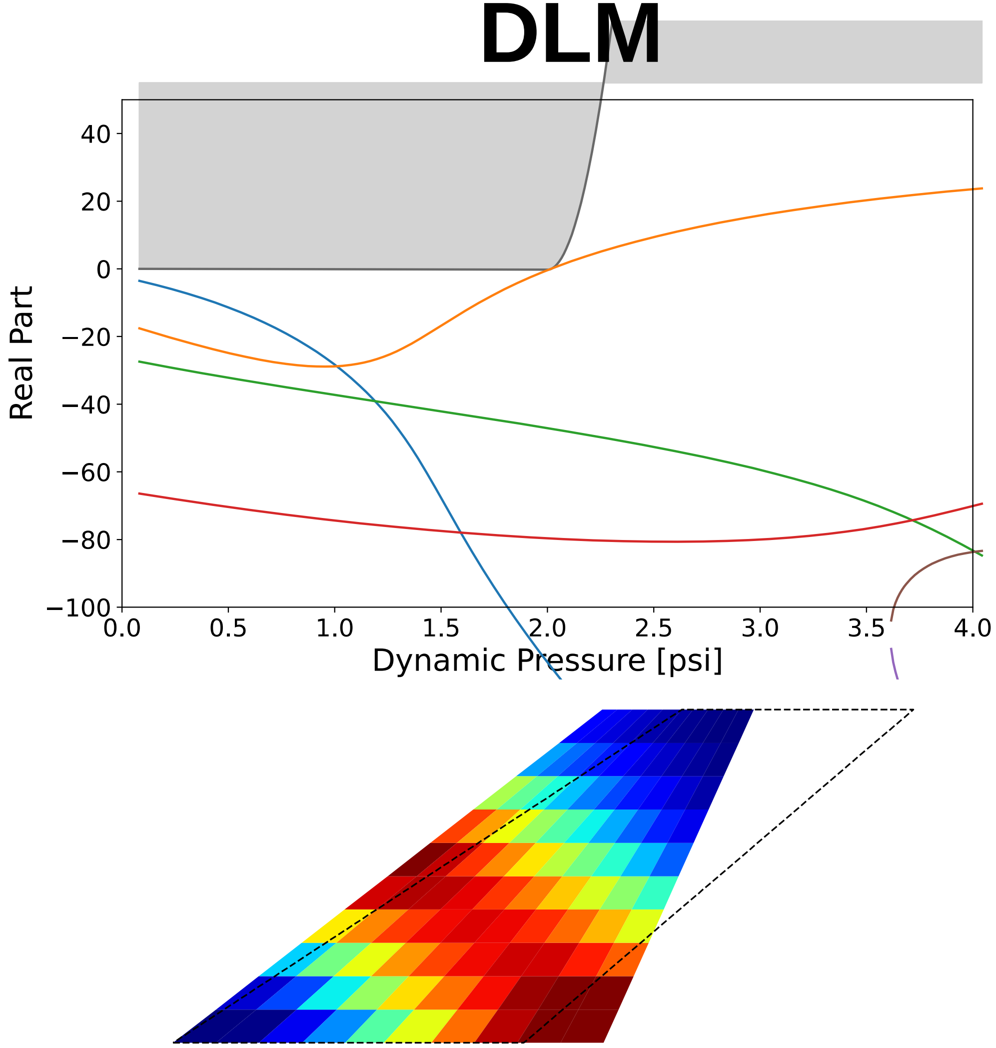


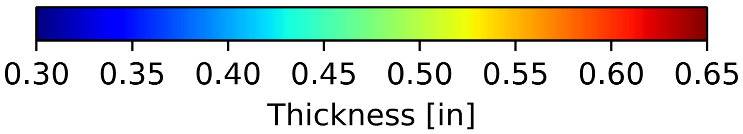
AGARD 445.6 – Optimized Designs

LFD



DLM





Impact of Mphys on OpenMDAO

- **Mphys pushes the envelope for large parallel models in OpenMDAO**
- **Enhancements to OM due to Mphys**

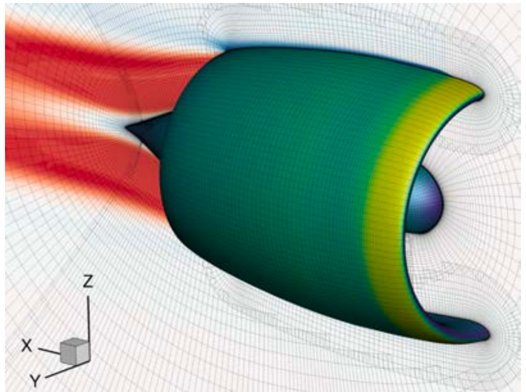
1. Distributed variables instead of components
2. Jacobian-vector product method at driver level for time-domain coupling
3. Change to parallel reduction order to have consistent dual states when using different number of processors
4. Redesign of recursive linear solver algorithms to avoid excess derivative computations
5. Shape by connection for inputs and outputs
6. Random vectors for directional derivatives
7. Aitken relaxation in linear block Gauss-Seidel solver

\+ Many bug reports

Conclusions

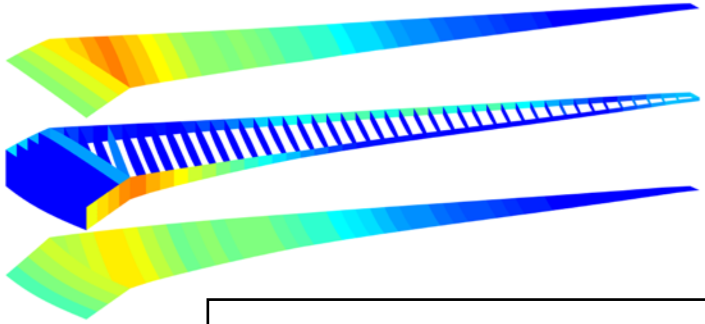
Mphys is a collaborative effort to standardize coupling in high-fidelity MDAO

- Address “one-off laborious, nonstandard interfaces”
- Uses OpenMDAO for automated coupling analysis and sensitivities
- Provides utilities to simplify the optimization problem setup



Mphys provides flexibility

- Analyses and optimizations of the same problems with different solvers
  - » Swap different fidelities for the same discipline
- Different types of multiphysics problems

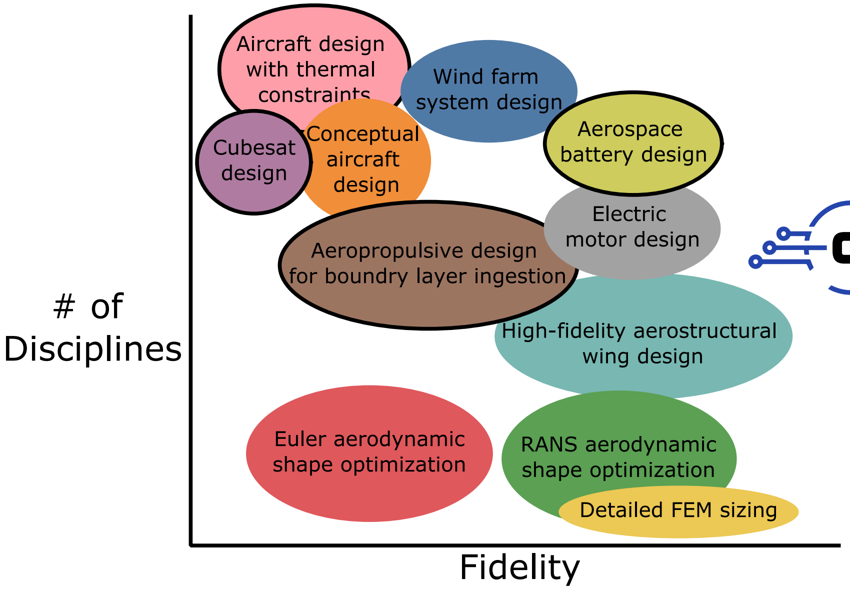


Learn more at the Mphys workshop tomorrow

Backup Slides

Why use OpenMDAO for Mphys?

- **Automated coupling and coupled sensitivities**
  - Easier extension to new disciplines
  - Separation of responsibility and expertise



- **Low barrier to entry**
  - Component development is simple for users
    - Custom objectives or constraints
  - Provides diagnostic tools for visualizing and checking a problem setup

- **Easier integrate into larger MDAO processes compared to previous one-off frameworks**

Isogai Airfoil – Flutter-Constrained Optimization

- **By changing the post-coupling groups, we can do flutter constraints**

Aeroelastic Optimization with Physics Sequencing

Objective:

- Minimize the mass of the wingbox

Constraints at maneuver condition:

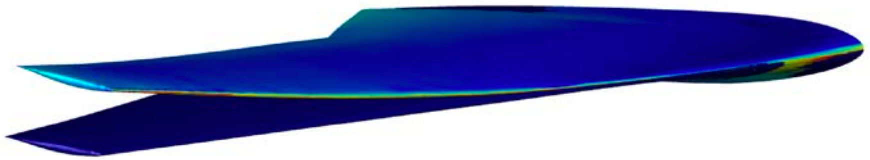
- Trim: L=2.0W
- Stress: 1.5 * KS failure < 1.0

Design variables:

- Angle of attack (1)
- Wingbox panel thicknesses (240)

Optimization process:

1. Perform optimization with VLM
2. Use VLM result as initial condition for CFD aerodynamics



FUN3D-SFE Euler + MELD + TACS