Optimal Shape Design Using SU2

Pointwise® and SU2 Joint Workshop

Sept 29th-30th, 2014

Francisco Palacios
Aeronautics & Astronautics Department (Stanford University)

The Open-Source CFD Code

aerospacedesignlab

I) INTRODUCTION TO SHAPE DESIGN
II) 1ST OPTIMIZATION ATTEMPT
III) 2ND OPTIMIZATION ATTEMPT
IV) CONCLUSIONS

SU2
The Open-Source CFD Code
aerospacedesignlab

I) INTRODUCTION TO SHAPE DESIGN

II) 1ST OPTIMIZATION ATTEMPT

III) 2ND OPTIMIZATION ATTEMPT

IV) CONCLUSIONS

SU2

The Open-Source CFD Code

aerospacedesignlab

Shape Design in SU2

There are a lot of moving parts in the design loop…

However, SU2 provides everything you need for design. Futhermore, it is possible to configure each module within a single **config file** (.cfg)

With everything in place, call **shape_optimization.py**

Python automates the shape design loop for you

```
$ shape_optimization.py -f LM-1021.cfg -n 96
```

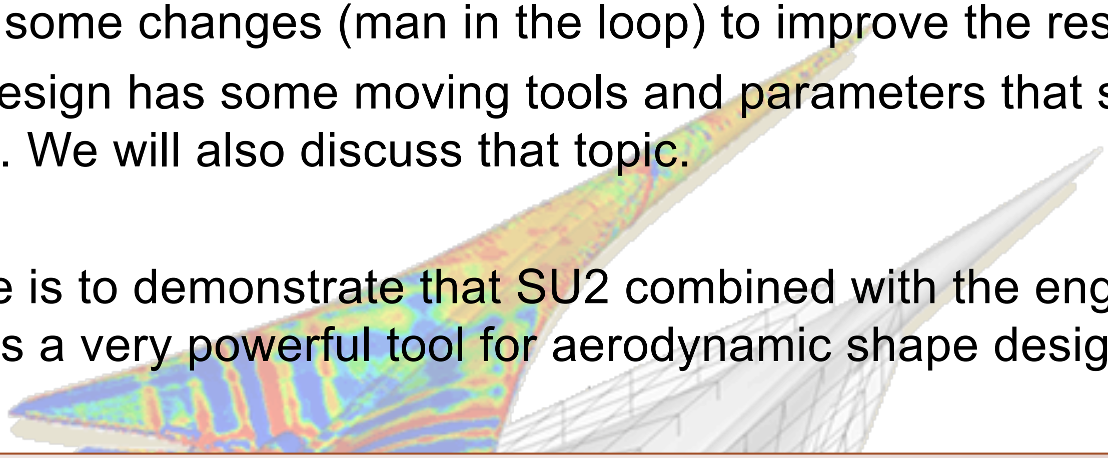
‘-f’ specifies config filename

‘-n’ specifies number of procs for parallel jobs

In this presentation…

We will focus on the redesign of a supersonic aircraft:

- Starting with a basic setting, we will run a quick optimization and then propose some changes (man in the loop) to improve the results.
- Shape design has some moving tools and parameters that should be adjusted. We will also discuss that topic.



My objective is to demonstrate that SU2 combined with the engineering knowledge is a very powerful tool for aerodynamic shape design.

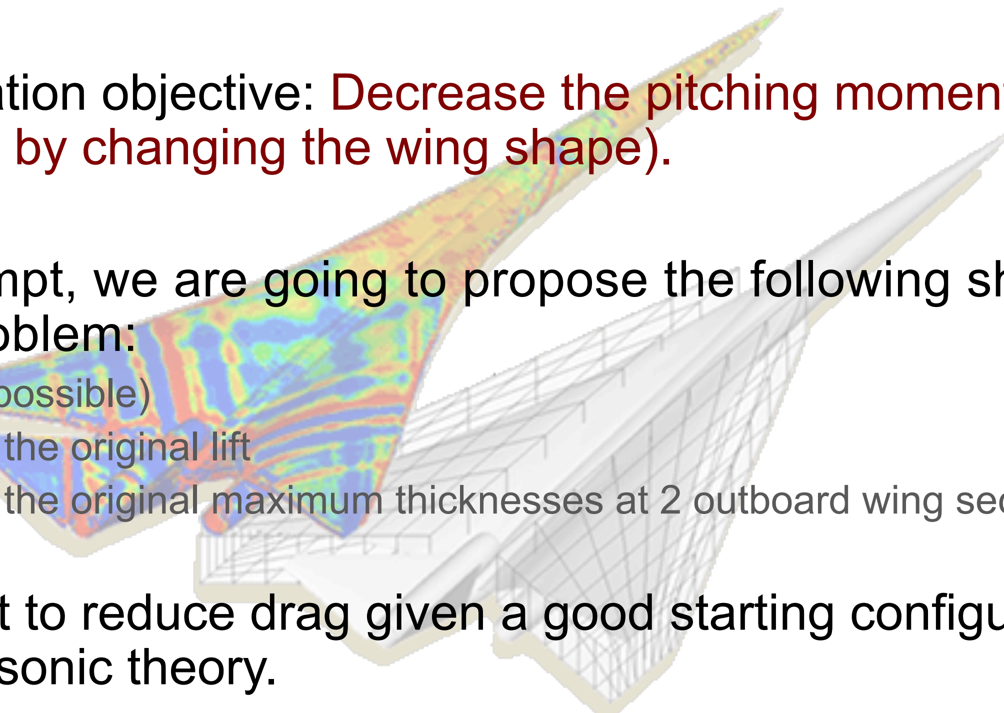
- Supersonic low-boom design is not new! What has changed? computational resources, turbulence modelling, adjoint based design tools, better understanding of the propagation…
- Our goal: to enable routine use of high-fidelity tools in the multi-disciplinary design of complete low-boom configurations.

Image | Lockheed Martin via NASA webpage

Shape Design in SU2

Python orchestrates the automatic shape design loop. Can easily adapt a config file from an existing case (check TestCases folder).

Design optimization objective: Decrease the pitching moment by 25% (trim the aircraft by changing the wing shape).



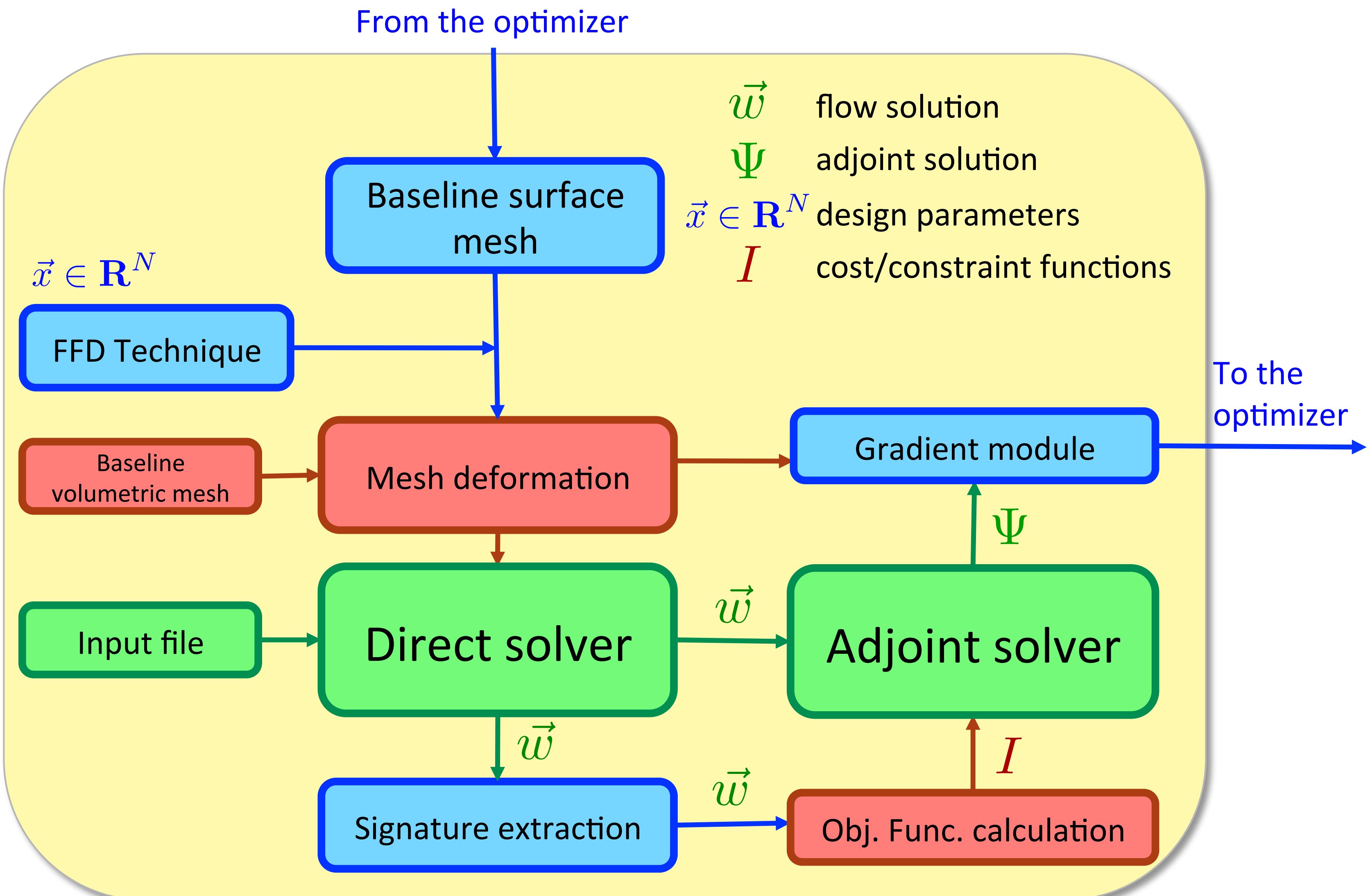
In our first attempt, we are going to propose the following shape optimization problem:

- Reduce drag (if possible)
- Maintain 95% of the original lift
- Maintain 90% of the original maximum thicknesses at 2 outboard wing sections

It can be difficult to reduce drag given a good starting configuration based on supersonic theory.

So, what do you need to know in order to set up your own shape design problems?

Optimization Problem

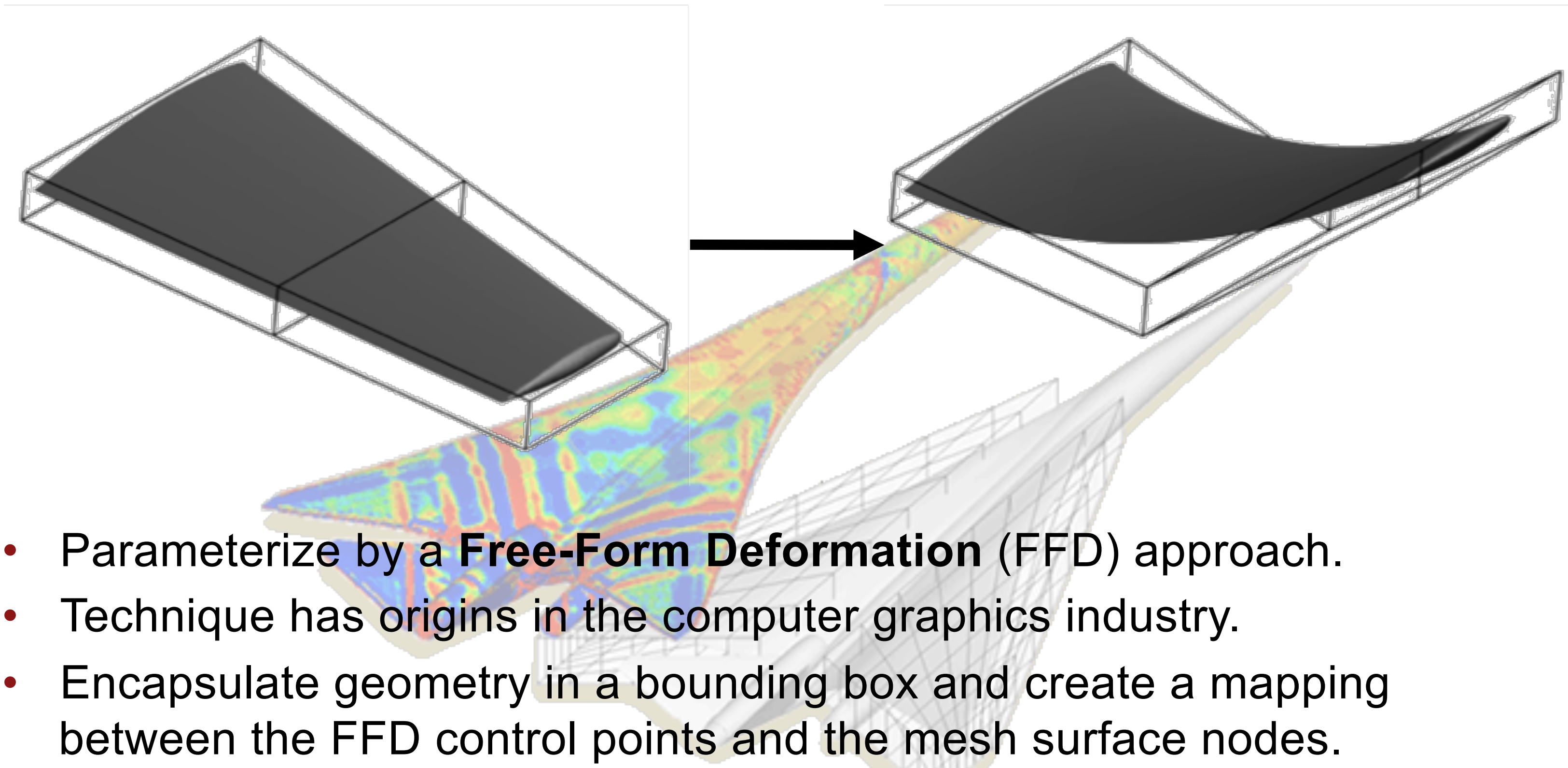


Optimization Problem

- Lastly, define your optimization problem in the config file
- Choose an objective, constraints (including geometric), and specify your design variables
- Full specifications can be found in **config_template.cfg**
- Scaling factor helps optimizer take appropriate first step

```
% Optimization objective function with scaling factor
% ex= Objective * Scale
OPT_OBJECTIVE= DRAG * 0.01
%
% Optimization constraint functions with scaling factors, separated by
semicolons
% ex= (Objective = Value ) * Scale, use '>','<','='
OPT_CONSTRAINT= (MOMENT_Y < 0.0135919515) * 0.01; (LIFT > 0.13212308464) *
0.01; (MAX_THICKNESS_SEC1 > 0.0035208432) * 0.01; (MAX_THICKNESS_SEC4 >
0.0013014216) * 0.01; (MAX_THICKNESS_SEC5 > 0.00052060608) * 0.01
```

Geometry Parameterization

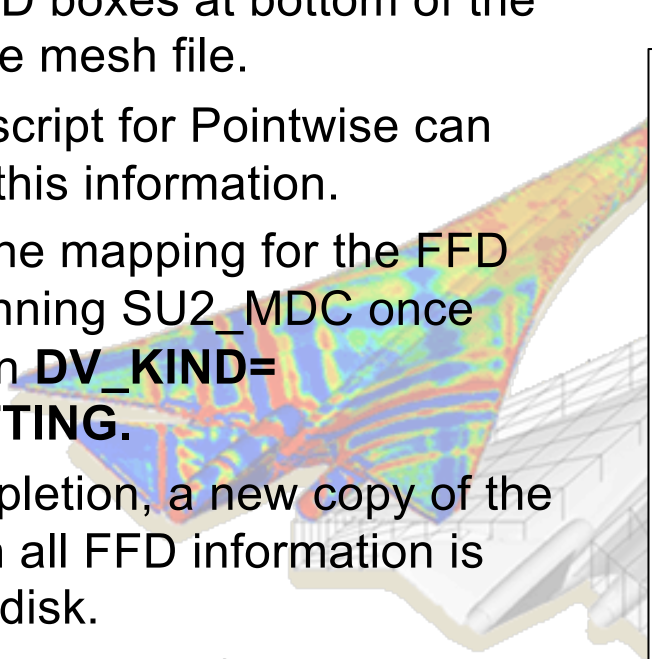


- Parameterize by a **Free-Form Deformation** (FFD) approach.
- Technique has origins in the computer graphics industry.
- Encapsulate geometry in a bounding box and create a mapping between the FFD control points and the mesh surface nodes.
- FFD control points become the design variables (DVs) with the surface inheriting a smooth deformation.
- FFD variables are now available in both 2D & 3D with SU^2 V3.2.2

Geometry Parameterization

SU2
The Open-Source CFD Code

- Set up FFD boxes at bottom of the SU2 native mesh file.
- The new script for Pointwise can generate this information.
- Initialize the mapping for the FFD box by running SU2_MDC once with option **DV_KIND= FFD_SETTING.**
- After completion, a new copy of the mesh with all FFD information is written to disk.
- Use this new mesh for shape optimization.

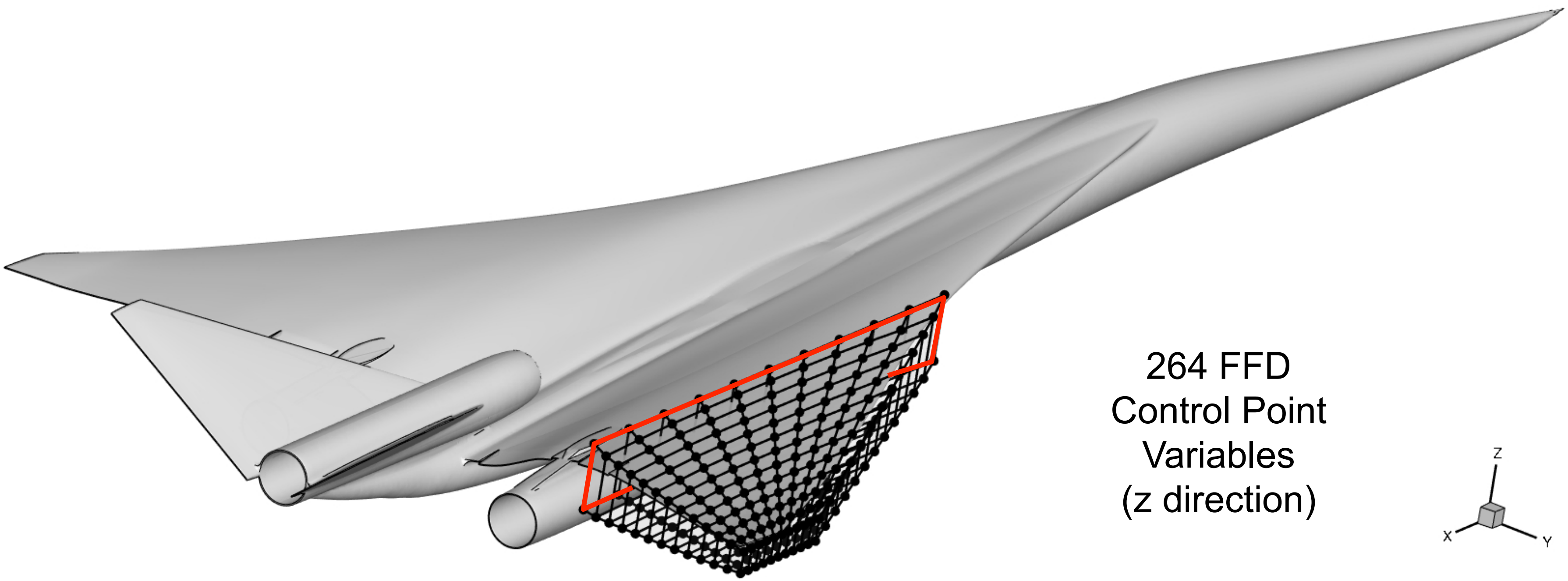


FFD box specification for our LM 1021 problem

```
FFD_NBOX= 1
FFD_NLEVEL= 1
FFD_TAG= WING
FFD_LEVEL= 0
FFD_DEGREE_I= 10
FFD_DEGREE_J= 12
FFD_DEGREE_K= 1
FFD_PARENTS= 0
FFD_CHILDREN= 0
FFD_CORNER_POINTS= 8
0.502544   -0.044427   0.0488
0.352456   -0.044427   0.0488
0.503809   -0.107234   0.06271
0.53585    -0.107234   0.06271
0.502544   -0.044427   0.06534
0.352456   -0.044427   0.06534
0.503809   -0.107234   0.06828
0.53585    -0.107234   0.06828
FFD_CONTROL_POINTS= 0
FFD_SURFACE_POINTS= 0
```

Geometry Parameterization

FFD box for redesigning the Lockheed Martin 1021 configuration



264 FFD
Control Point
Variables
(z direction)

```
DEFINITION_DV= ( 7, 1.0 | aircraft | WING, 0, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 1, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 2, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 3, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 4, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 5, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 6, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 7, 2, 0, 0.0, 0.0, 1.0 ); ...
```

SU2
The Open-Source CFD Code

Flow & Adjoint Problem Setup

Find $S^{\text{min}} \in S_{\text{ad}}$ such that

$$J(S^{\text{min}}) = \min_{S \in S_{\text{ad}}} J(S),$$

where $J(S) = \int_S j(U, \vec{n})\, ds.$

- $U \equiv$ set of flow variables
- $\alpha \equiv$ set of design variables
- Governing equation of the fluid: $R(U, \alpha) = 0$
- Functional to minimize: $J = J(U, \alpha)$

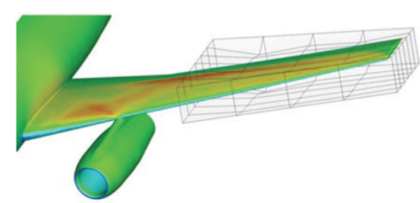
Adjoint approach. What is the adjoint equation?

$$\left.\begin{aligned} \delta J &= \tfrac{\partial J}{\partial U}\delta U + \tfrac{\partial J}{\partial \alpha}\delta\alpha \\ \delta R &= \tfrac{\partial R}{\partial U}\delta U + \tfrac{\partial R}{\partial \alpha}\delta\alpha = 0 \end{aligned}\right\} \delta J = \left(\frac{\partial J}{\partial U} - \Psi^{\mathsf{T}}\frac{\partial R}{\partial U}\right)\delta U + \left(\frac{\partial J}{\partial \alpha} - \Psi^{\mathsf{T}}\frac{\partial R}{\partial \alpha}\right)\delta\alpha$$

Choosing Ψ to satisfy the adjoint equation:

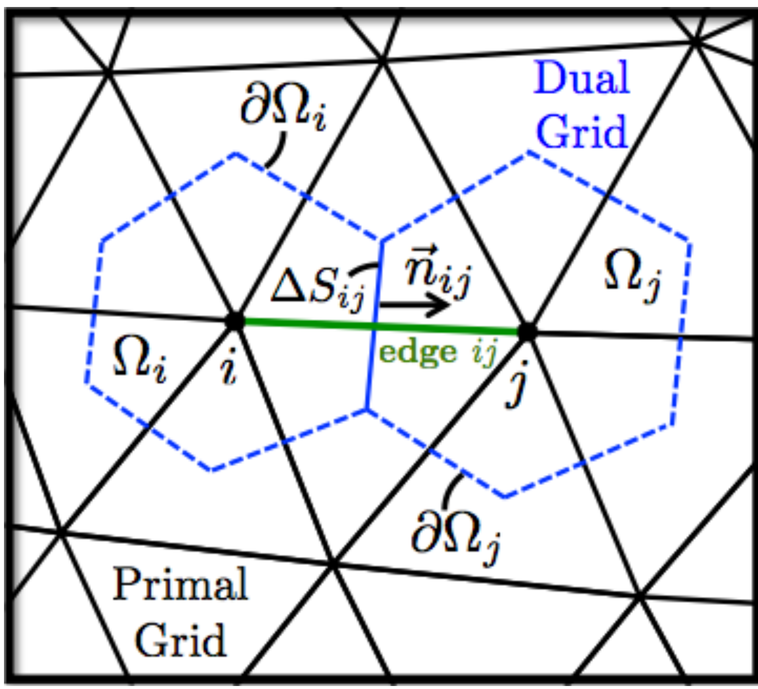
$$\frac{\partial J}{\partial U} - \Psi^{\mathsf{T}}\frac{\partial R}{\partial U} = 0 \quad \Rightarrow \quad \delta J = \left(\frac{\partial J}{\partial \alpha} - \Psi^{\mathsf{T}}\frac{\partial R}{\partial \alpha}\right)\delta\alpha$$

Easy to formulate, complex to solve robustly when dealing with turbulence, shocks, geometries with singularities, surface formulation.



SU2
The Open-Source CFD Code

Flow & Adjoint Problem Setup

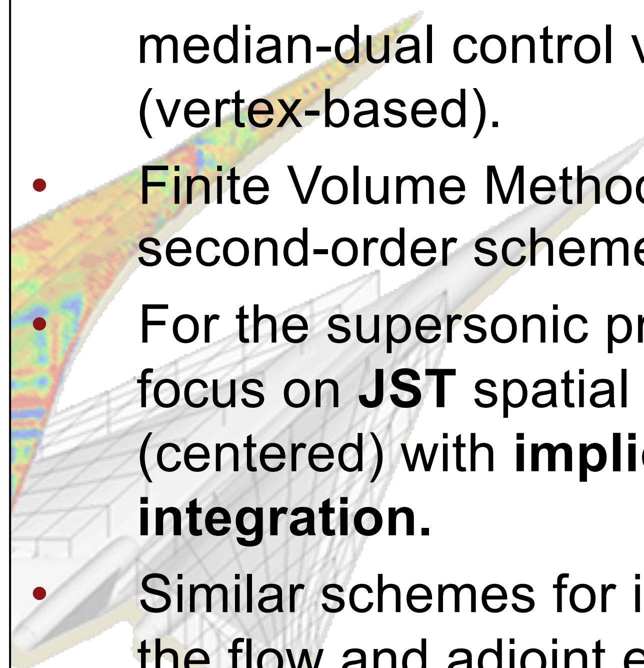


- Unstructured meshes with median-dual control volumes (vertex-based).
- Finite Volume Method with second-order schemes.
- For the supersonic problem, we'll focus on **JST** spatial integration (centered) with **implicit time integration.**
- Similar schemes for integrating the flow and adjoint equations (adjoint scheme is non-conservative).

Flow & Adjoint Problem Setup

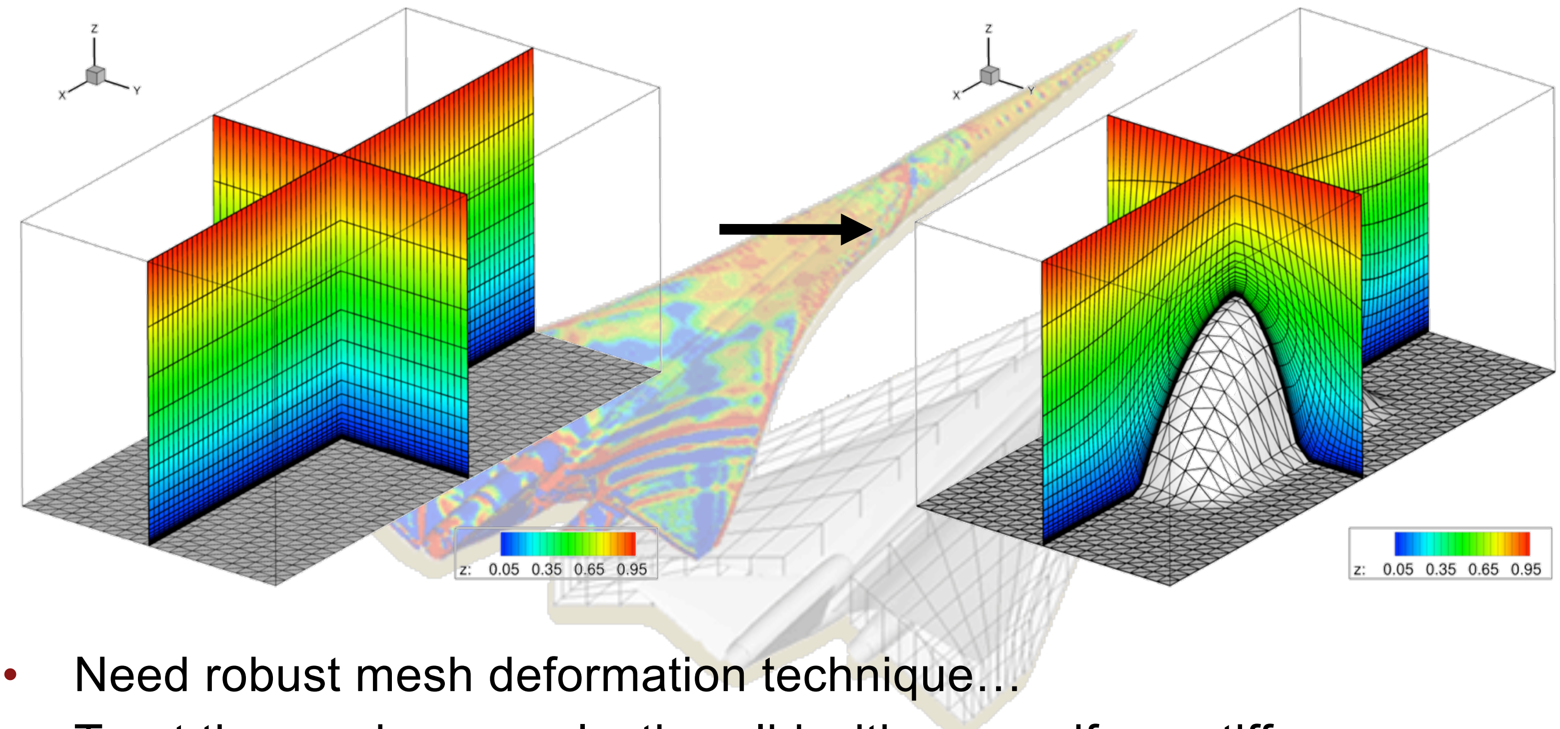
```
% - FLOW NUMERICAL METHOD DEFINITION - %
%
% Convective numerical method (JST, LAX-FRIEDRICH,
CUSP, ROE, AUSM, HLLC, TURKEL_PREC, MSW)
CONV_NUM_METHOD_FLOW= JST
%
% 1st, 2nd and 4th order artificial dissipation
coefficients
AD_COEFF_FLOW= ( 0.15, 0.5, 0.02 )
%
% Time discretization (RUNGE-KUTTA_EXPLICIT,
EULER_IMPLICIT, EULER_EXPLICIT)
TIME_DISCRE_FLOW= EULER_IMPLICIT

% - ADJOINT-FLOW NUMERICAL METHOD DEFINITION -%
%
% Convective numerical method (JST, LAX-FRIEDRICH,
ROE)
CONV_NUM_METHOD_ADJFLOW= JST
%
% 1st, 2nd, and 4th order artificial dissipation
coefficients
AD_COEFF_ADJFLOW= ( 0.15, 0.5, 0.05 )
%
% Time discretization (RUNGE-KUTTA_EXPLICIT,
EULER_IMPLICIT)
TIME_DISCRE_ADJFLOW= EULER_IMPLICIT
%
% Reduction factor of the CFL coefficient in the
adjoint problem
CFL_REDUCTION_ADJFLOW= 0.25
%
% Limit value for the adjoint variable
LIMIT_ADJFLOW= 1E6
```



- Unstructured meshes with median-dual control volumes (vertex-based).
- Finite Volume Method with second-order schemes.
- For the supersonic problem, we'll focus on **JST** spatial integration (centered) with **implicit time integration.**
- Similar schemes for integrating the flow and adjoint equations (adjoint scheme is non-conservative).

Mesh Deformation



- Need robust mesh deformation technique…
- Treat the mesh as an elastic solid with non-uniform stiffness
- Solve the **linear elasticity equations** on the mesh for the nodal displacements using the movement of the boundaries as input

Mesh Deformation

SU2
The Open-Source CFD Code

```
% ------------------------ GRID DEFORMATION PARAMETERS ------------------------%
%
% Number of smoothing iterations for FEA mesh deformation
DEFORM_LINEAR_ITER= 500
%
% Number of nonlinear deformation iterations (surface deformation increments)
DEFORM_NONLINEAR_ITER= 1
%
% Print the residuals during mesh deformation to the console (YES, NO)
DEFORM_CONSOLE_OUTPUT= YES
%
% Factor to multiply smallest cell volume for deform tolerance (0.001 default)
DEFORM_TOL_FACTOR = 0.001
%
% Type of element stiffness imposed for FEA mesh deformation (INVERSE_VOLUME,
%                                           WALL_DISTANCE, CONSTANT_STIFFNESS)
DEFORM_STIFFNESS_TYPE= INVERSE_VOLUME
%
% Visualize the deformation (NO, YES)
VISUALIZE_DEFORMATION= NO
%
% Divide elemetns into triangles and tetrahedra
DIVIDE_ELEMENTS= YES
```

Controls type of cell stiffness. Smaller cells or those near walls will be more rigid under deformation (preserves mesh quality).

Shape Design in SU2

```
$ shape_optimization.py -f LM-1021.cfg -n 96
```

I) INTRODUCTION TO SHAPE DESIGN

II) 1ST OPTIMIZATION ATTEMPT

III) 2ND OPTIMIZATION ATTEMPT

IV) CONCLUSIONS

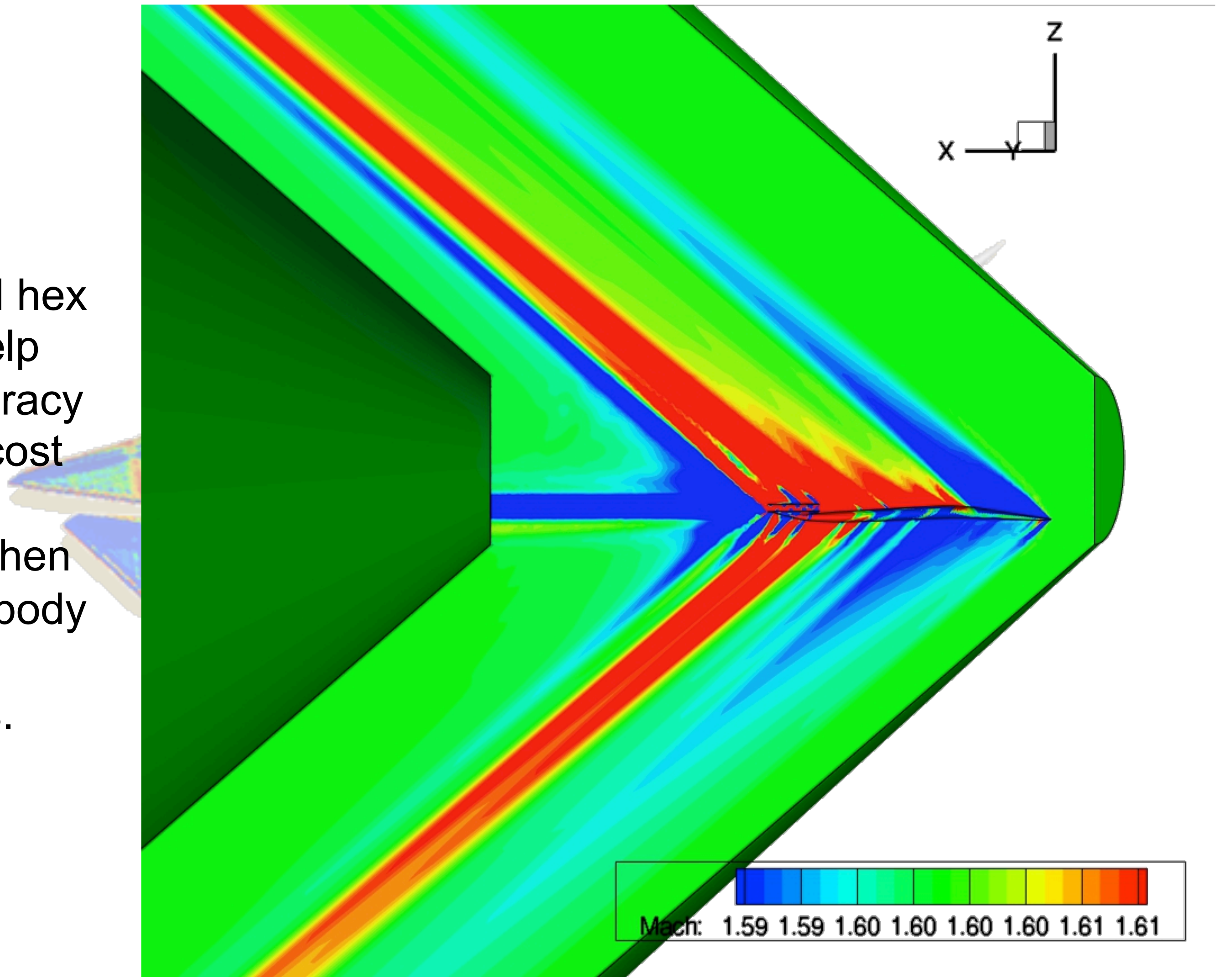
SU2

The Open-Source CFD Code

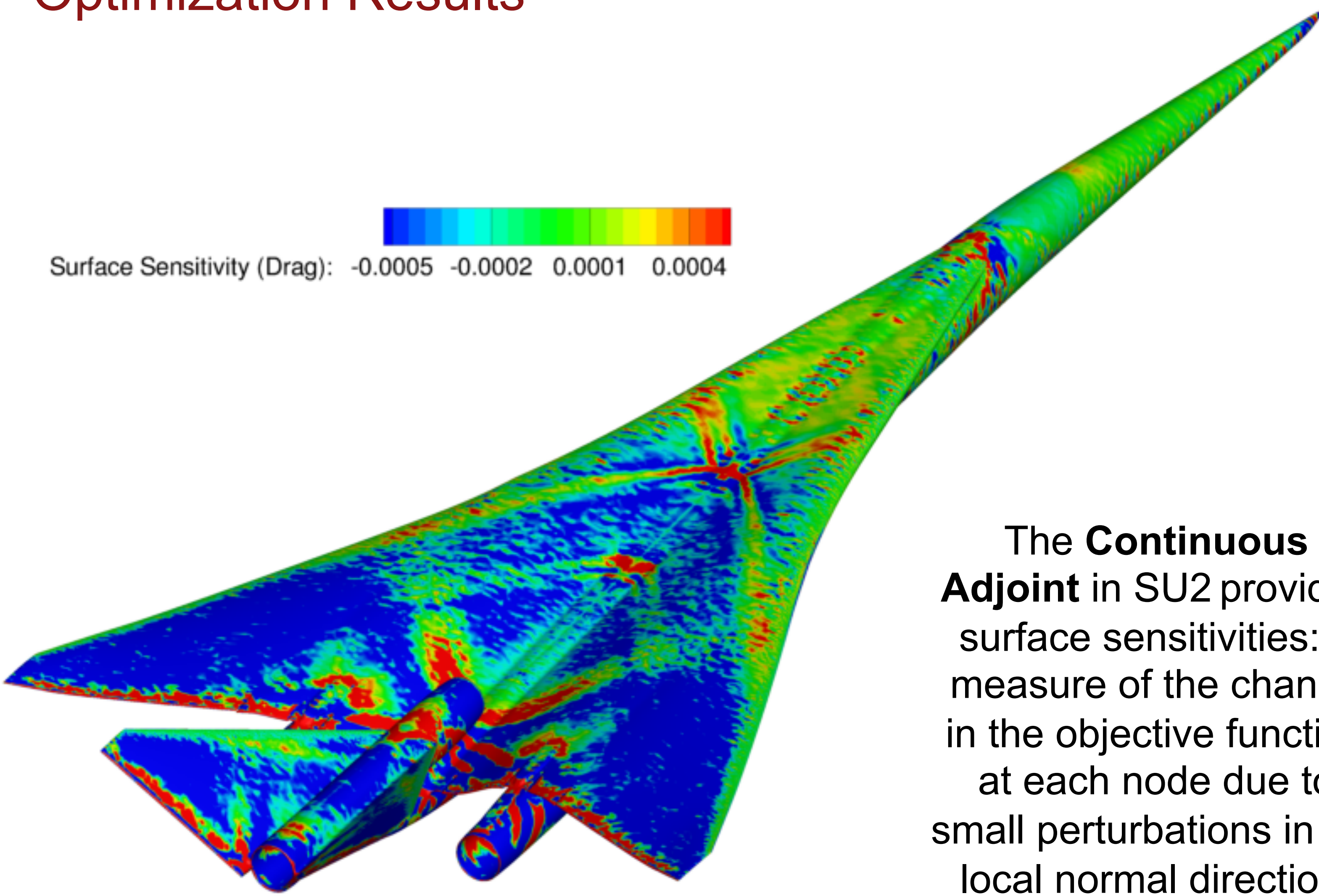
aerospacedesignlab

Optimization Results

Shock-aligned hex cells can help maintain accuracy and reduce cost for boom predictions when extracting off-body pressure signatures.



Optimization Results



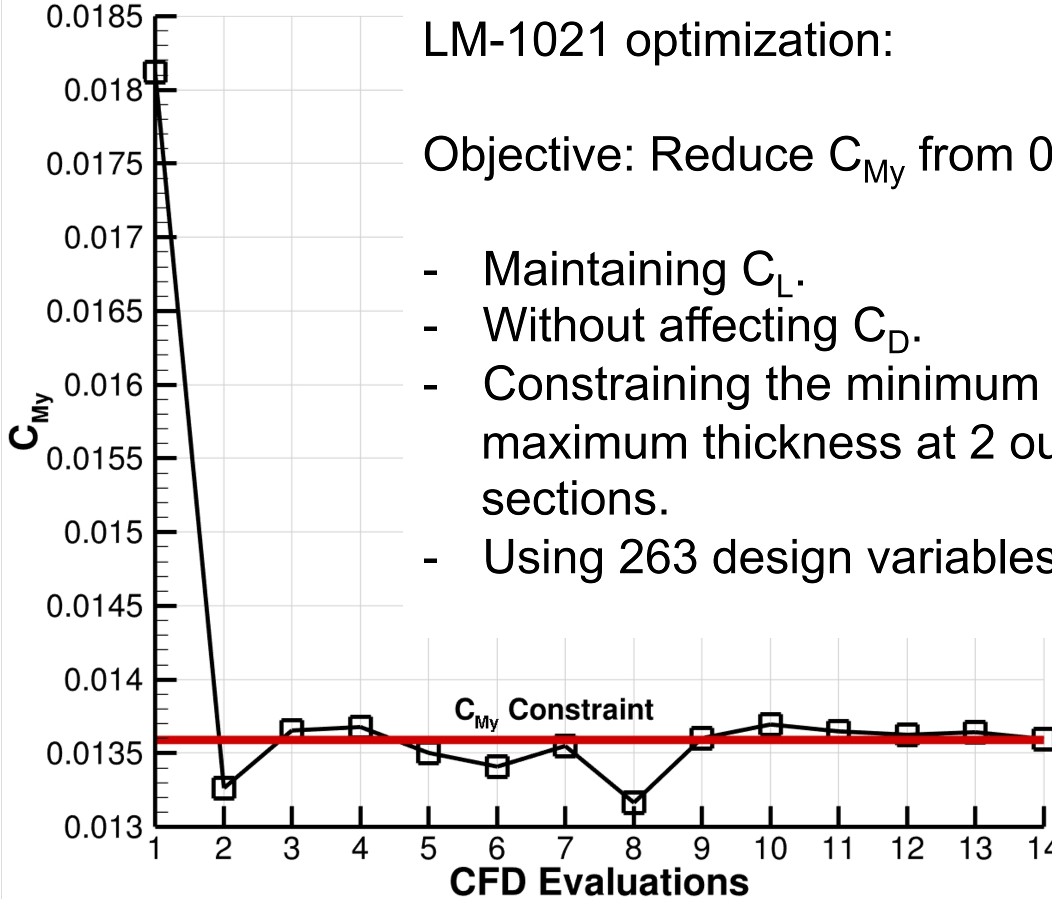
The **Continuous Adjoint** in SU2 provides surface sensitivities: a measure of the change in the objective function at each node due to small perturbations in the local normal direction.

Pitching Moment Constraint

LM-1021 optimization:

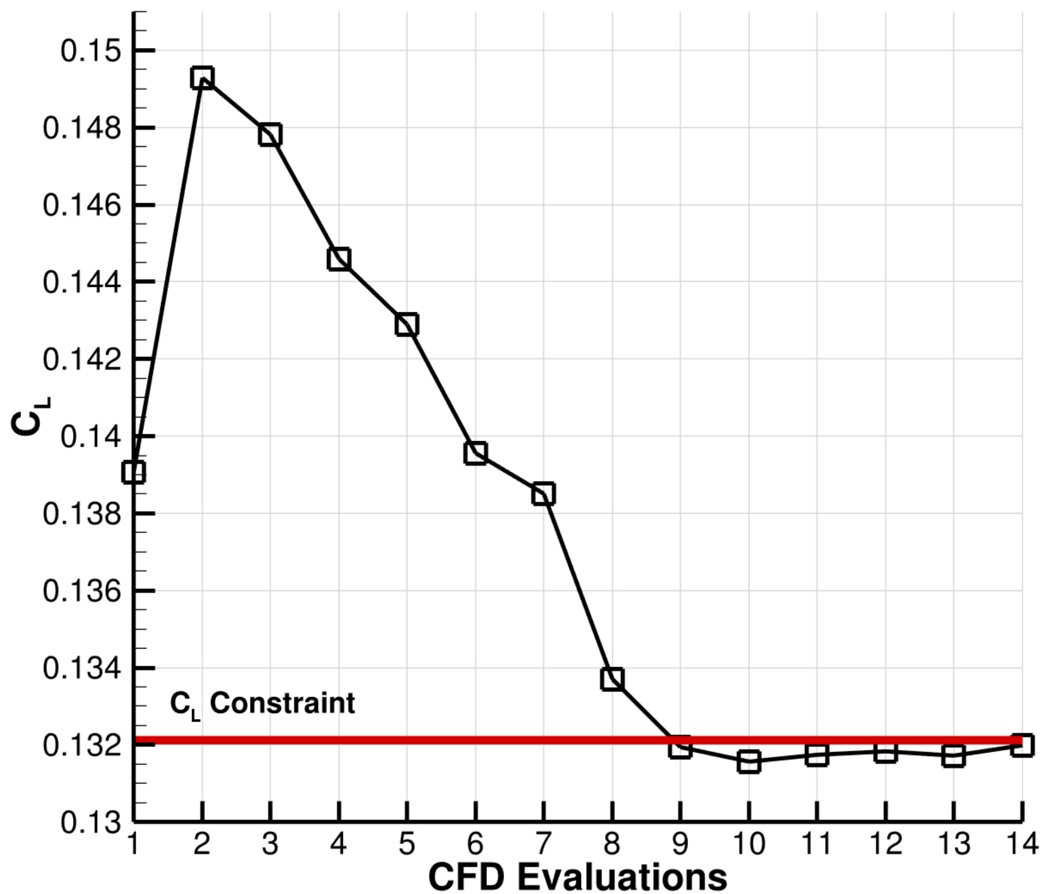
Objective: Reduce C_{My} from 0.018 to 0.0135.

- Maintaining C_L.
- Without affecting C_D.
- Constraining the minimum value of the maximum thickness at 2 outboard sections.
- Using 263 design variables.



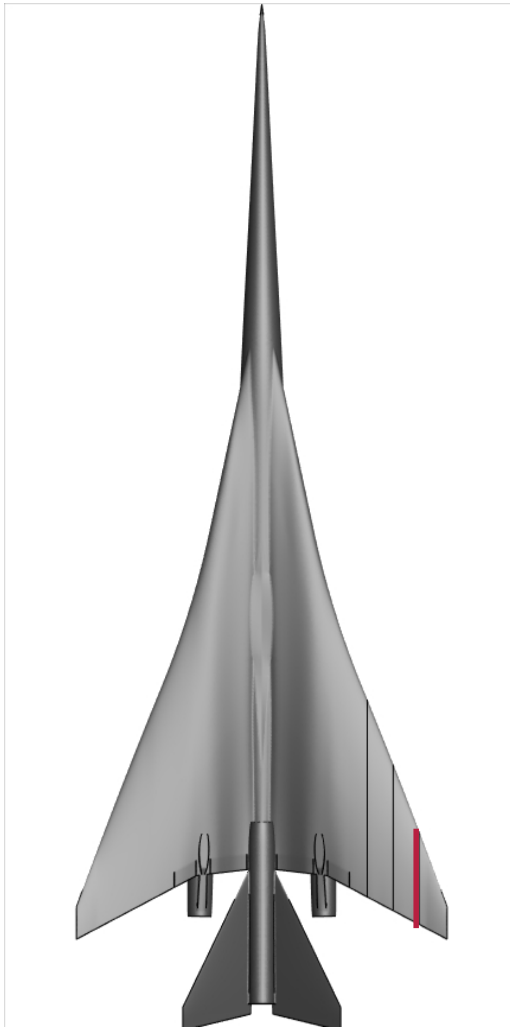
✔ The main objective has been satisfied

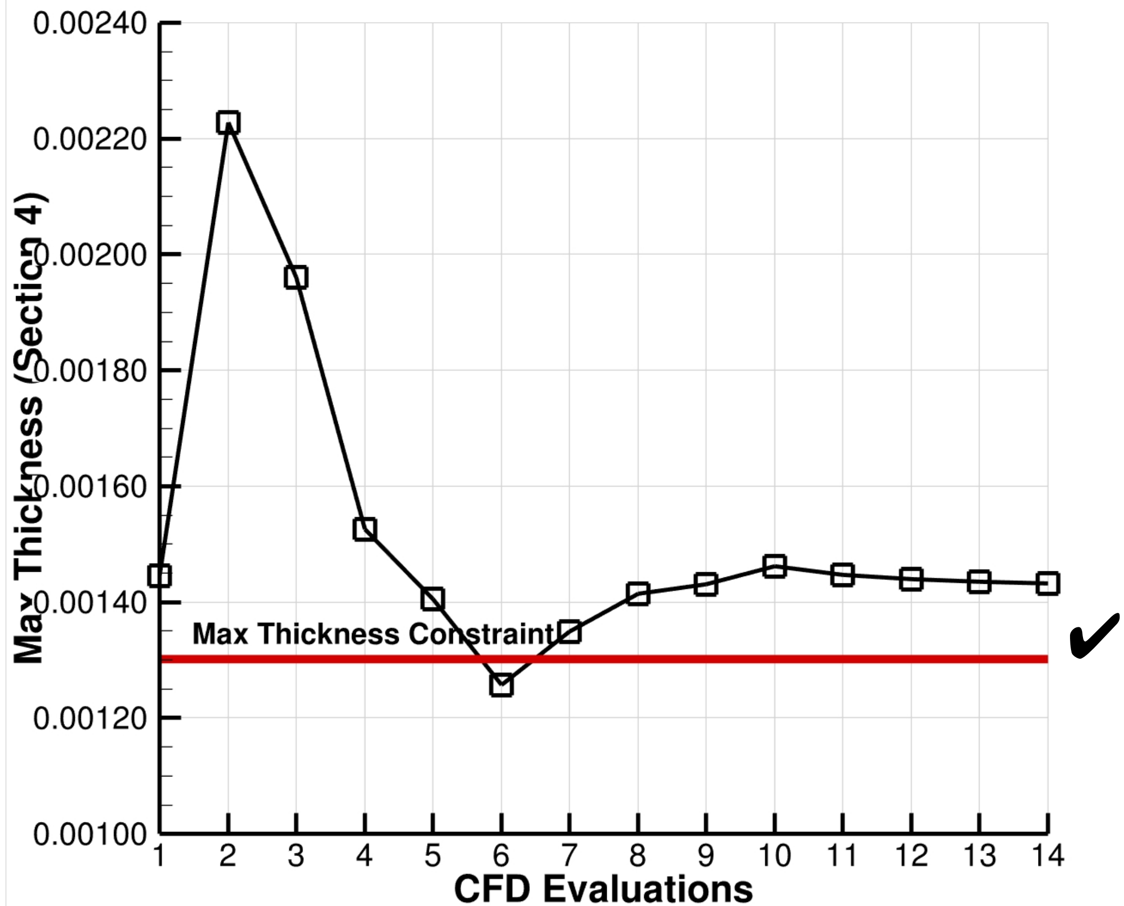
Lift constraint



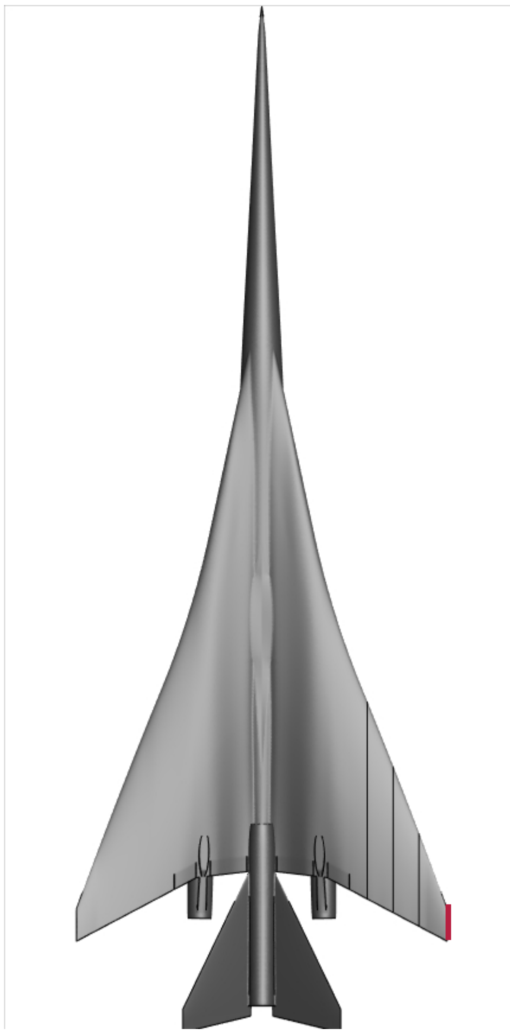
The C_L is acceptable ✔

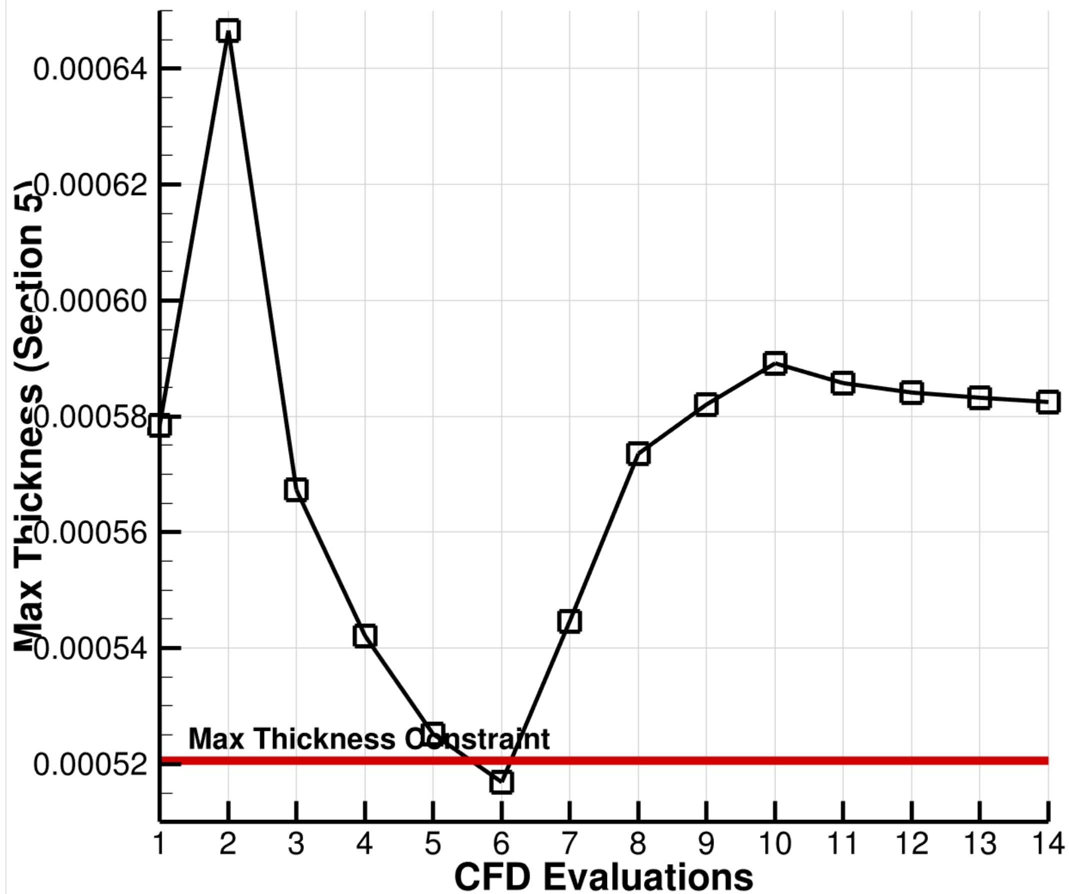
Maximum thickness (y=0.08675m) constraint





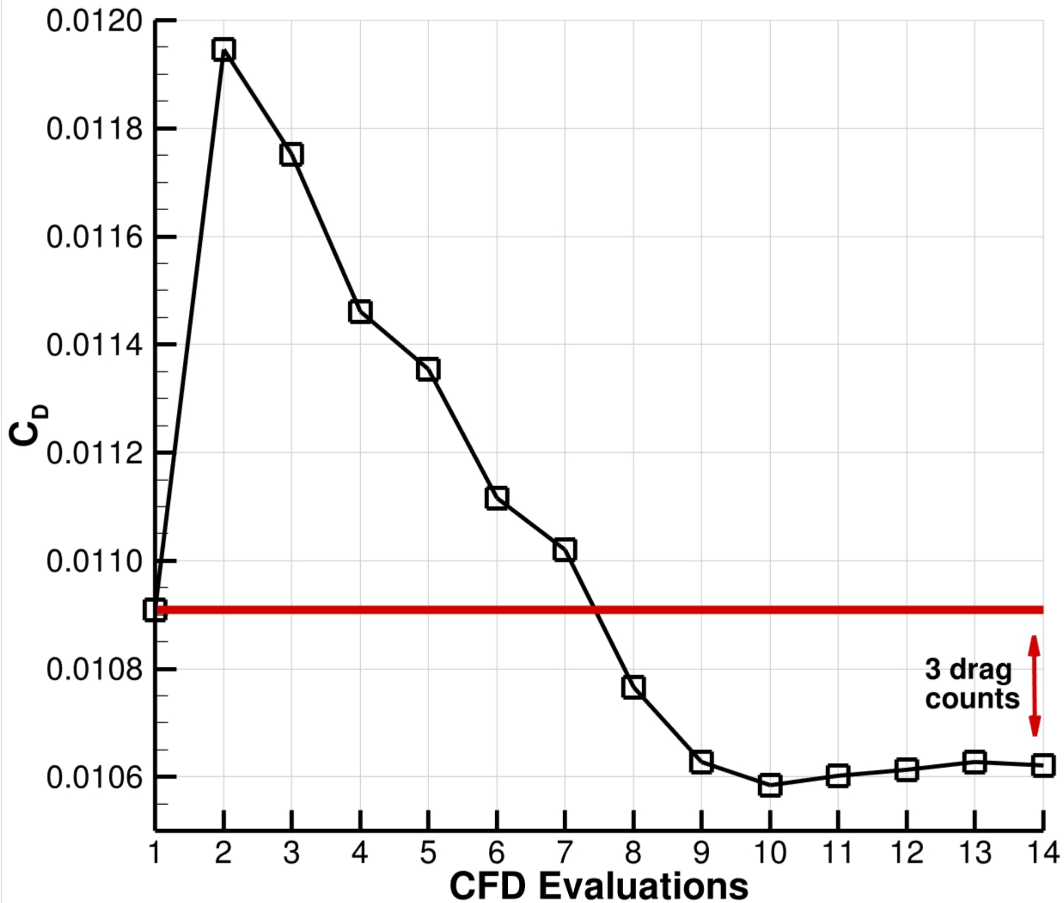
Maximum thickness (y=0.10100m) constraint





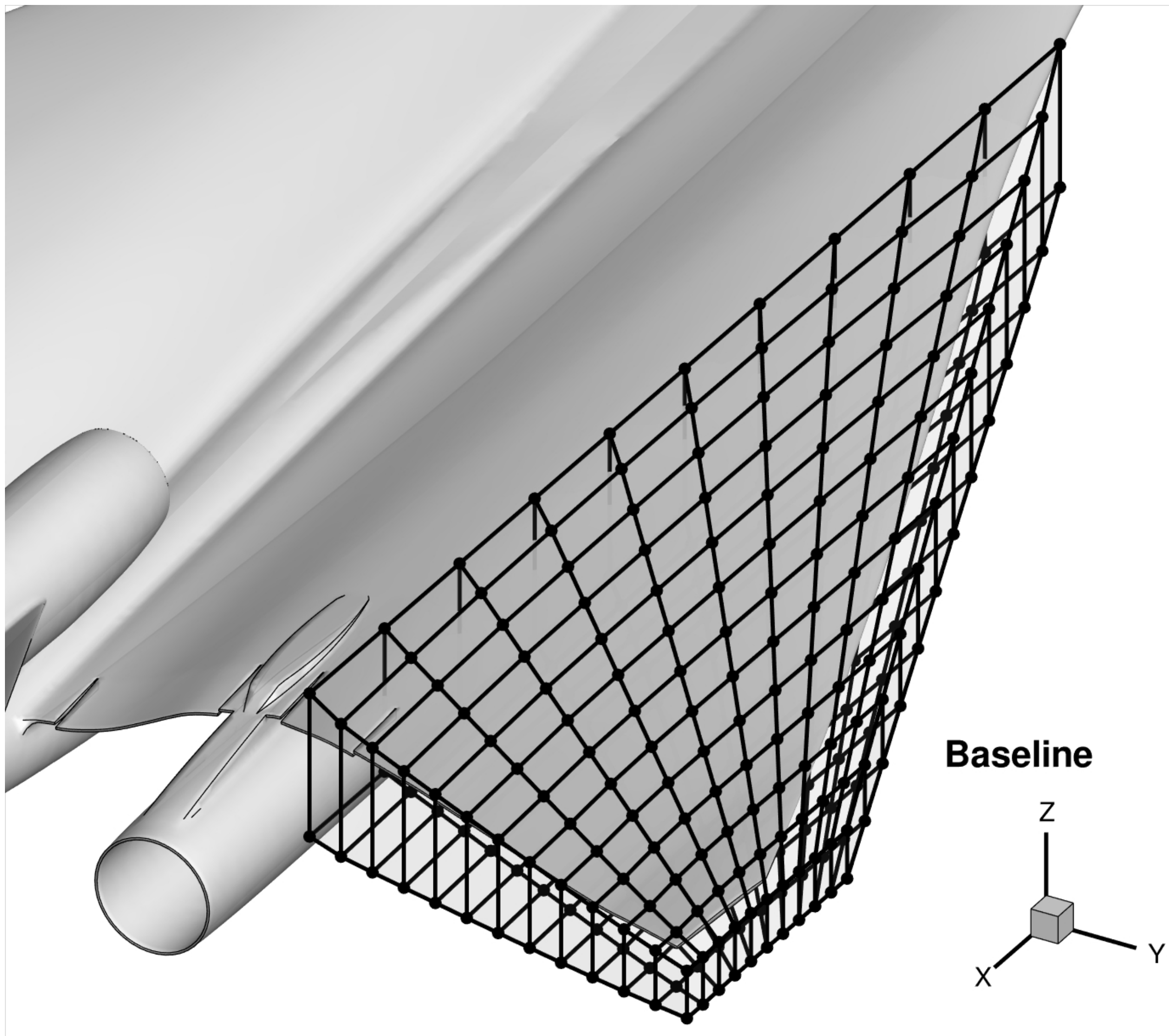
✔

Drag objective function (min)

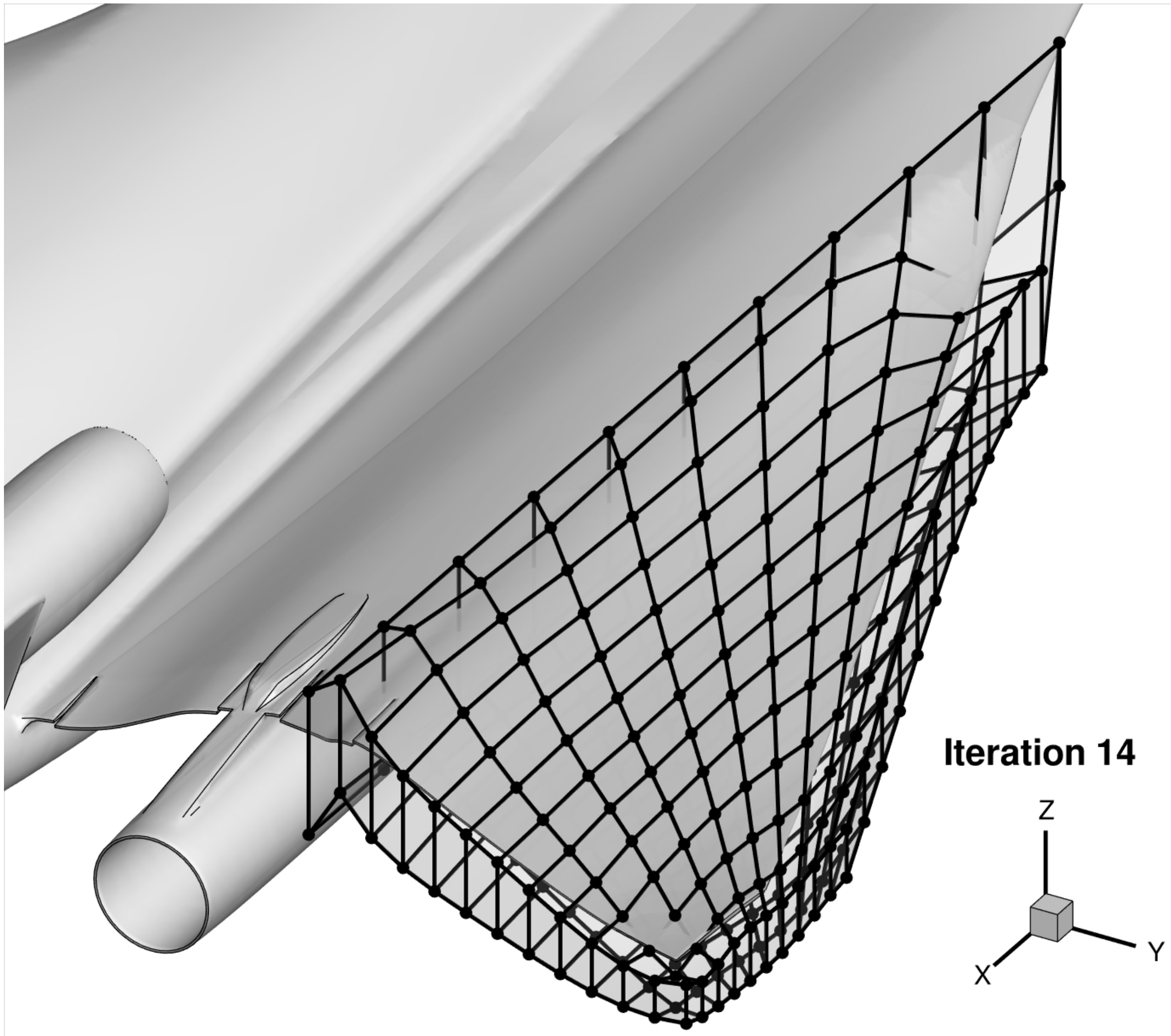


✔ Small decrease in C_D

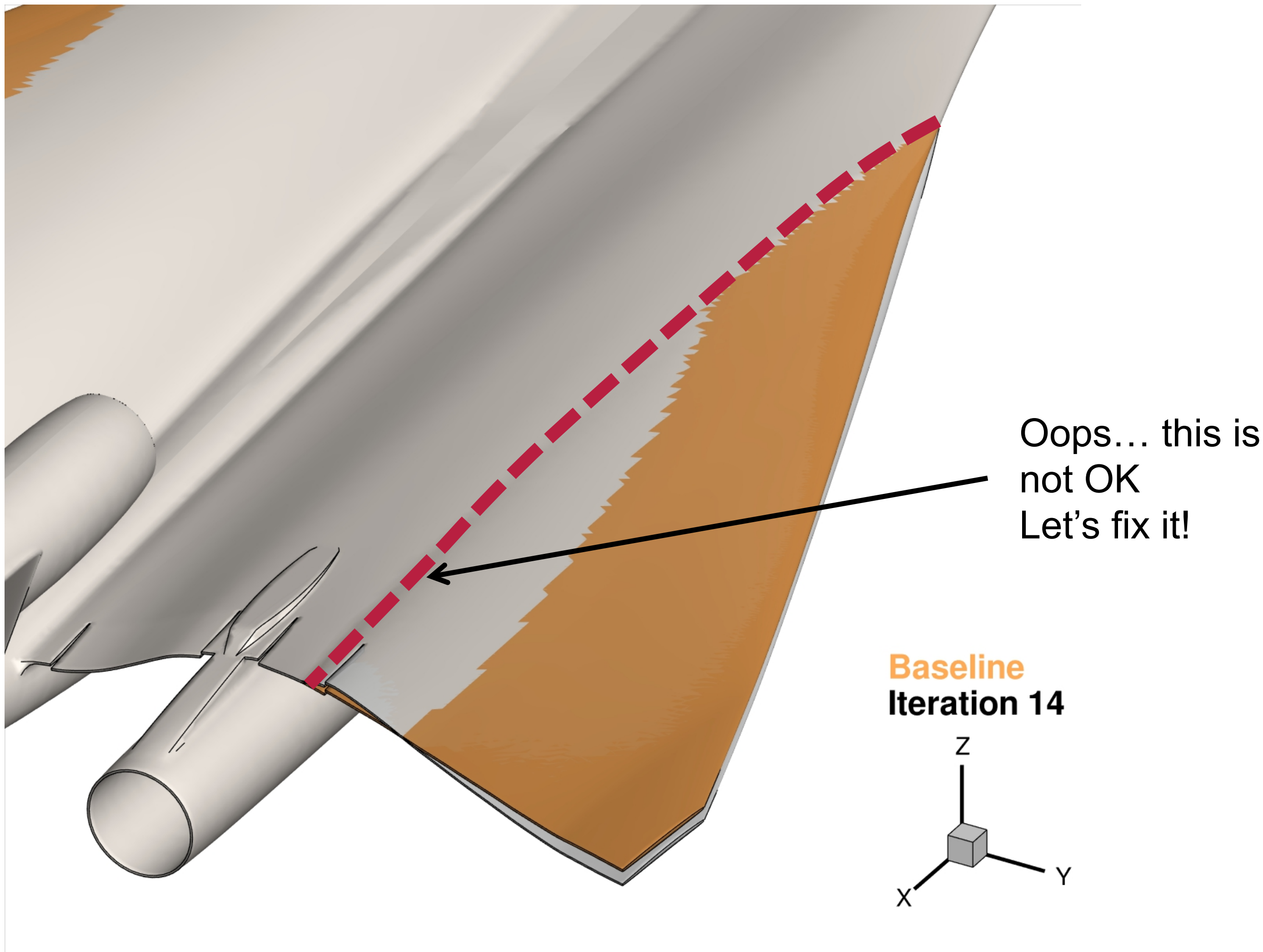
Baseline (LM-1021) geometry and design variables



Optimized (iteration 14) geometry and control points loc.



Geometry comparison



I) INTRODUCTION TO SHAPE DESIGN

II) 1ST OPTIMIZATION ATTEMPT

III) 2ND OPTIMIZATION ATTEMPT

IV) CONCLUSIONS

SU2

The Open-Source CFD Code

aerospacedesignlab

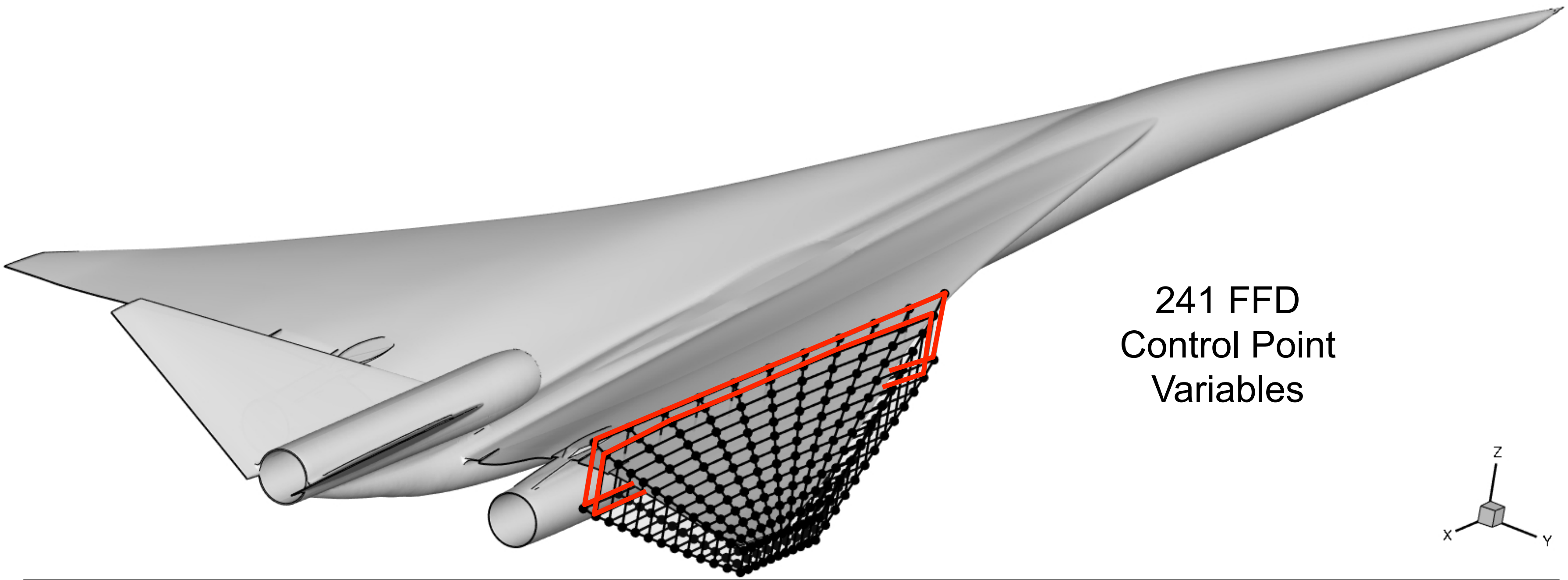
Redefine the Optimization Problem

- We should redefine our optimization problem.
- Modify the design variables to guarantee the continuity of the 1st surface derivative.
- Add new constraints: Max thickness in the inboard and outboard sections (90%) and minimum twist angle (3.4deg)

```
% Optimization objective function with scaling factor
% ex= Objective * Scale
OPT_OBJECTIVE= DRAG * 0.01
%
% Optimization constraint functions with scaling factors, separated by
semicolons
% ex= (Objective = Value ) * Scale, use '>','<','='
OPT_CONSTRAINT= (MOMENT_Y < 0.0135919515) * 0.01; (LIFT > 0.13212308464) *
0.01; (MAX_THICKNESS_SEC1 > 0.0035208432) * 0.01; (MAX_THICKNESS_SEC2 >
0.0027518184) * 0.01; (MAX_THICKNESS_SEC3 > 0.0020452032) * 0.01;
(MAX_THICKNESS_SEC4 > 0.0013014216) * 0.01; (MAX_THICKNESS_SEC5 >
0.00052060608) * 0.01; (AOA_SEC1 > -3.3) * 0.01; (AOA_SEC2 > -3.3) * 0.01;
(AOA_SEC3 > -3.3) * 0.01; (AOA_SEC4 > -3.3) * 0.01; (AOA_SEC5 > -3.3) * 0.01
```

New Geometry Parameterization

FFD box for redesigning the Lockheed Martin 1021 configuration

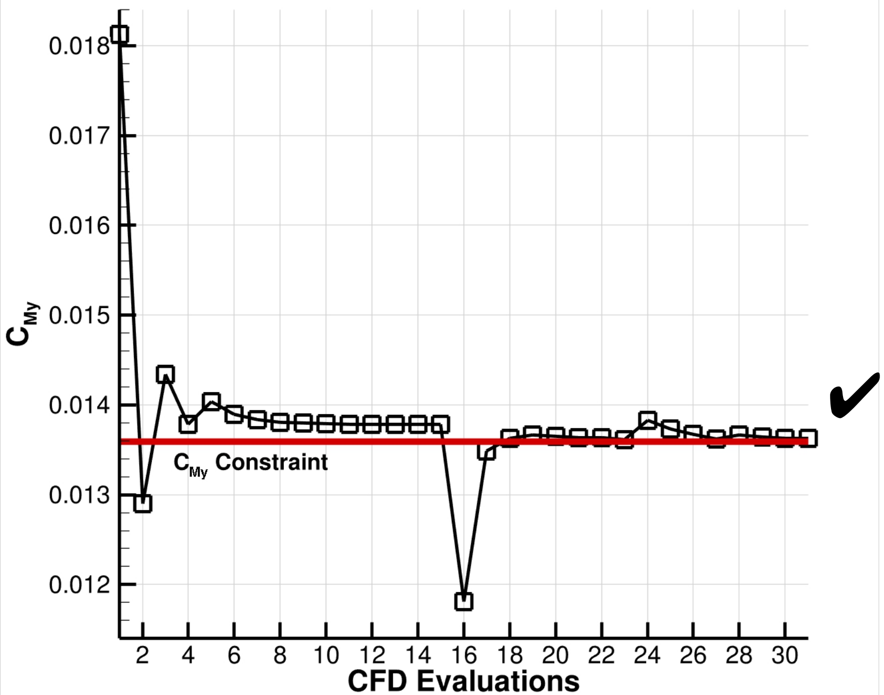


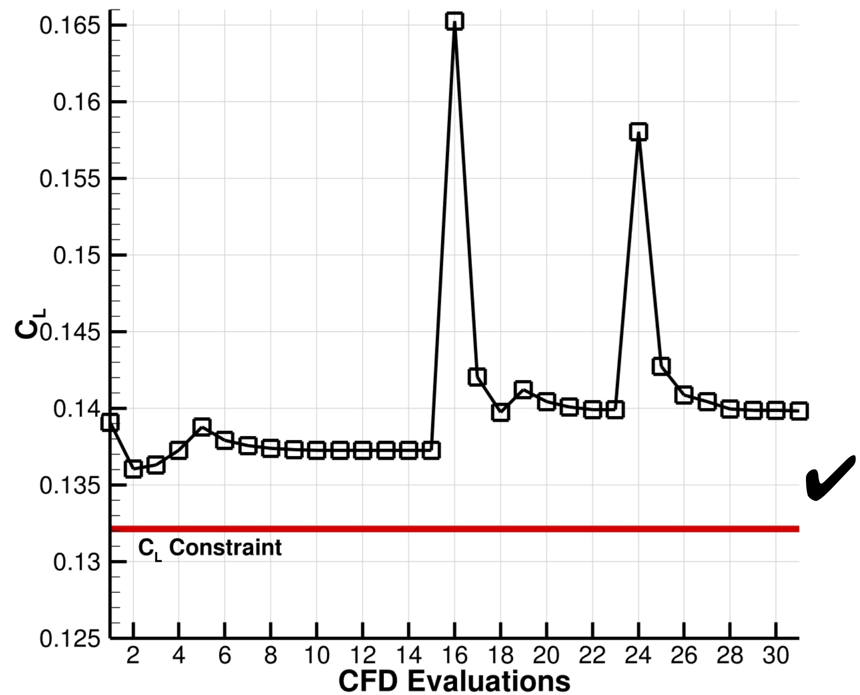

```
DEFINITION_DV= ( 7, 1.0 | aircraft | WING, 0, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft
| WING, 1, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 2, 2, 0, 0.0, 0.0, 1.0 );
( 7, 1.0 | aircraft | WING, 3, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 4, 2,
0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 5, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 |
aircraft | WING, 6, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | aircraft | WING, 7, 2, 0, 0.0, 0.0,
1.0 ); ...
```

Lift and Pitching Moment Constraint

Objective: Reduce C_{My} from 0.018 to 0.0135.

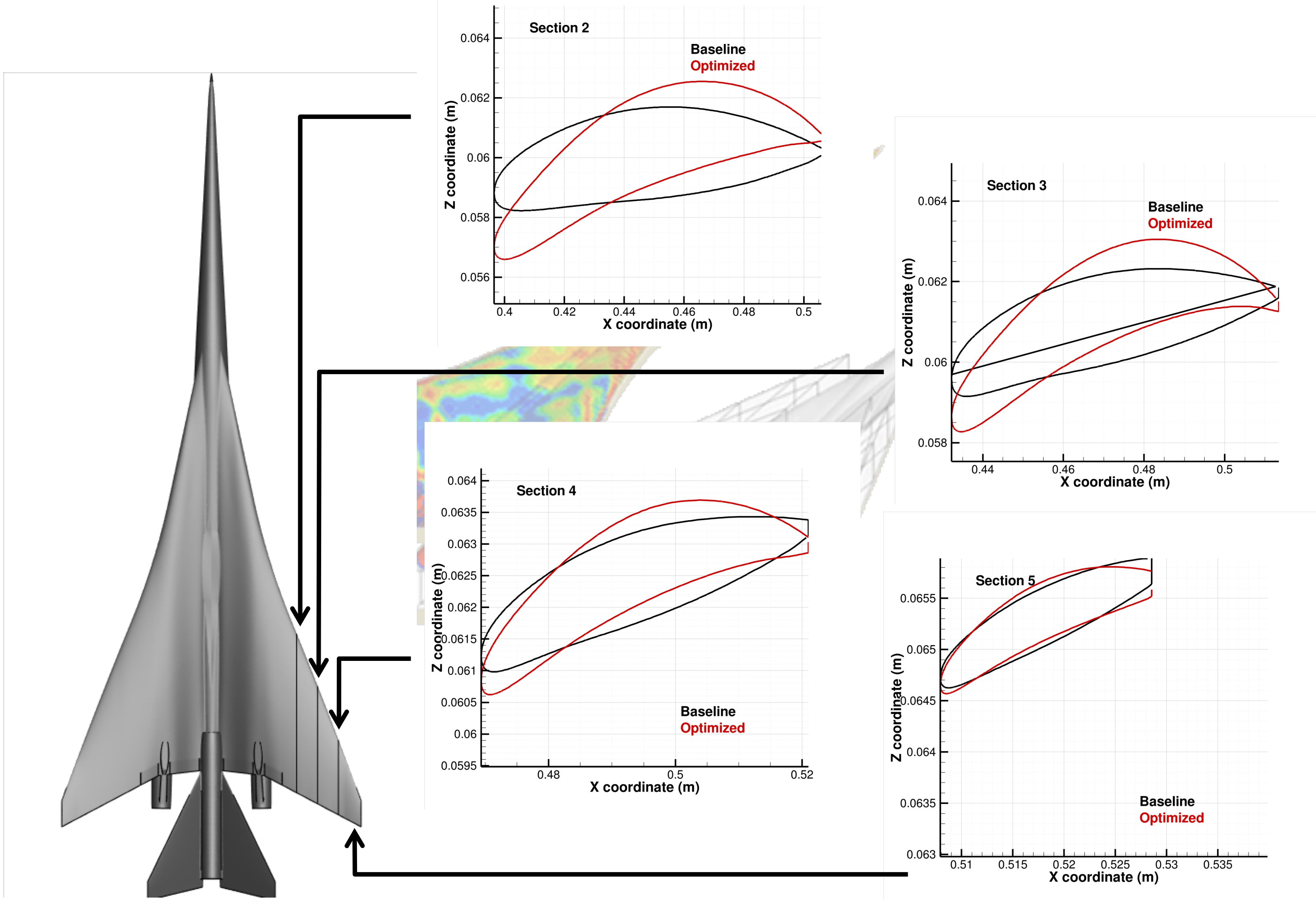
- Maintaining C_L (with a minimal impact in C_D). Unfortunately, at the end, we have 5 extra drag counts.
- Constraining the minimum value of the maximum thickness and twist angle at 4 sections.
- Using 241 design variables.





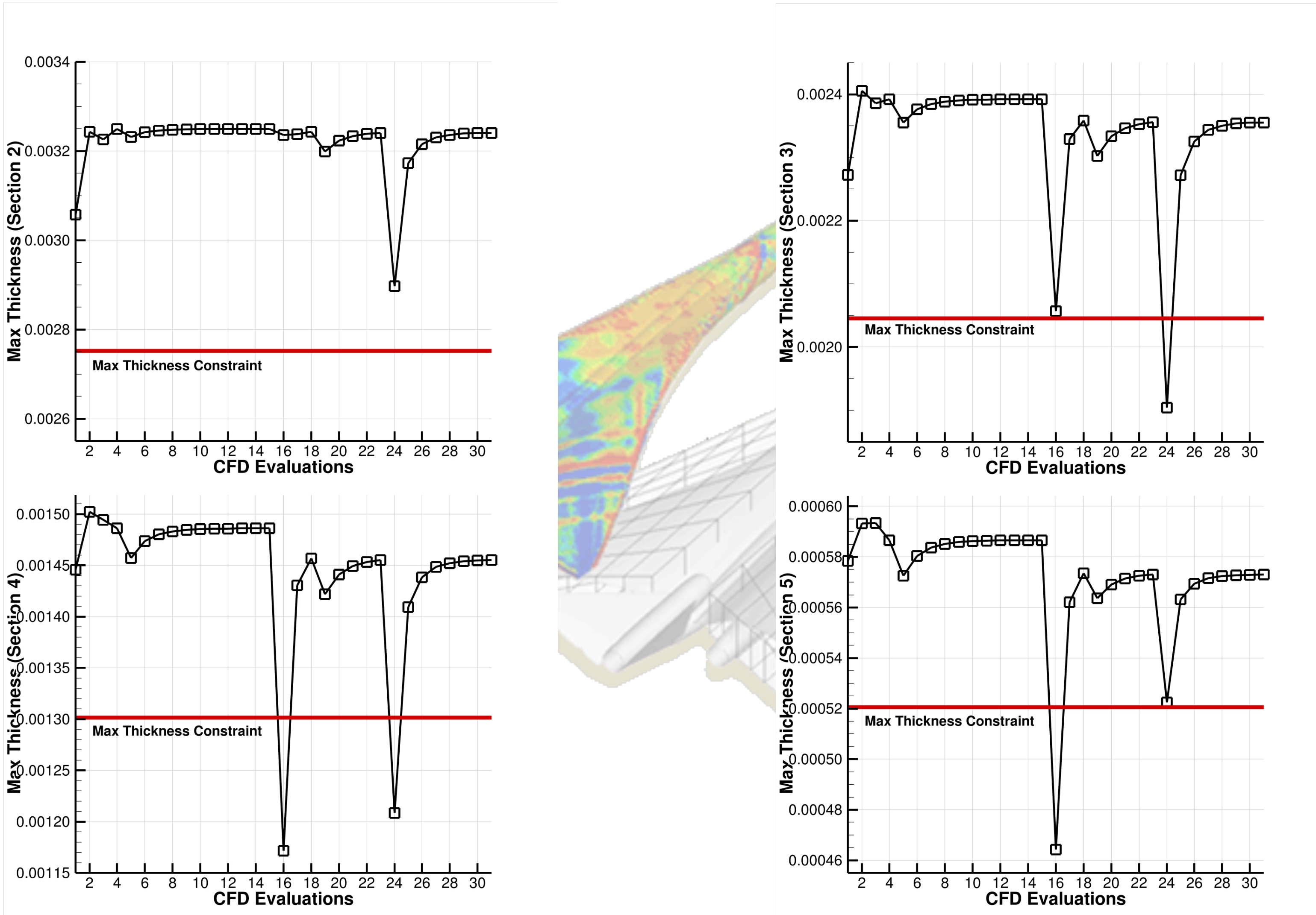
SU2
The Open-Source CFD Code

Geometric constraints

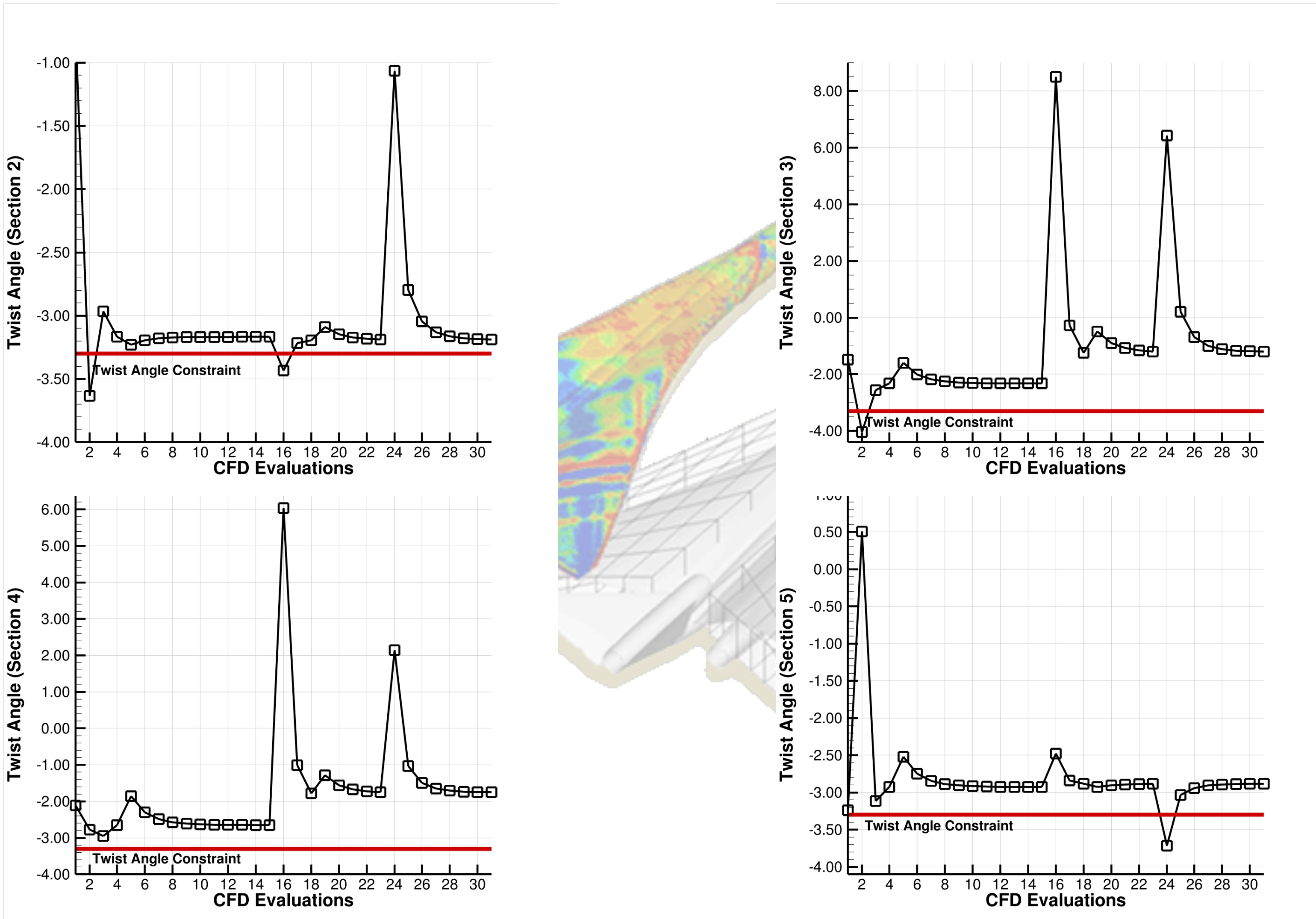


SU2 The Open-Source CFD Code

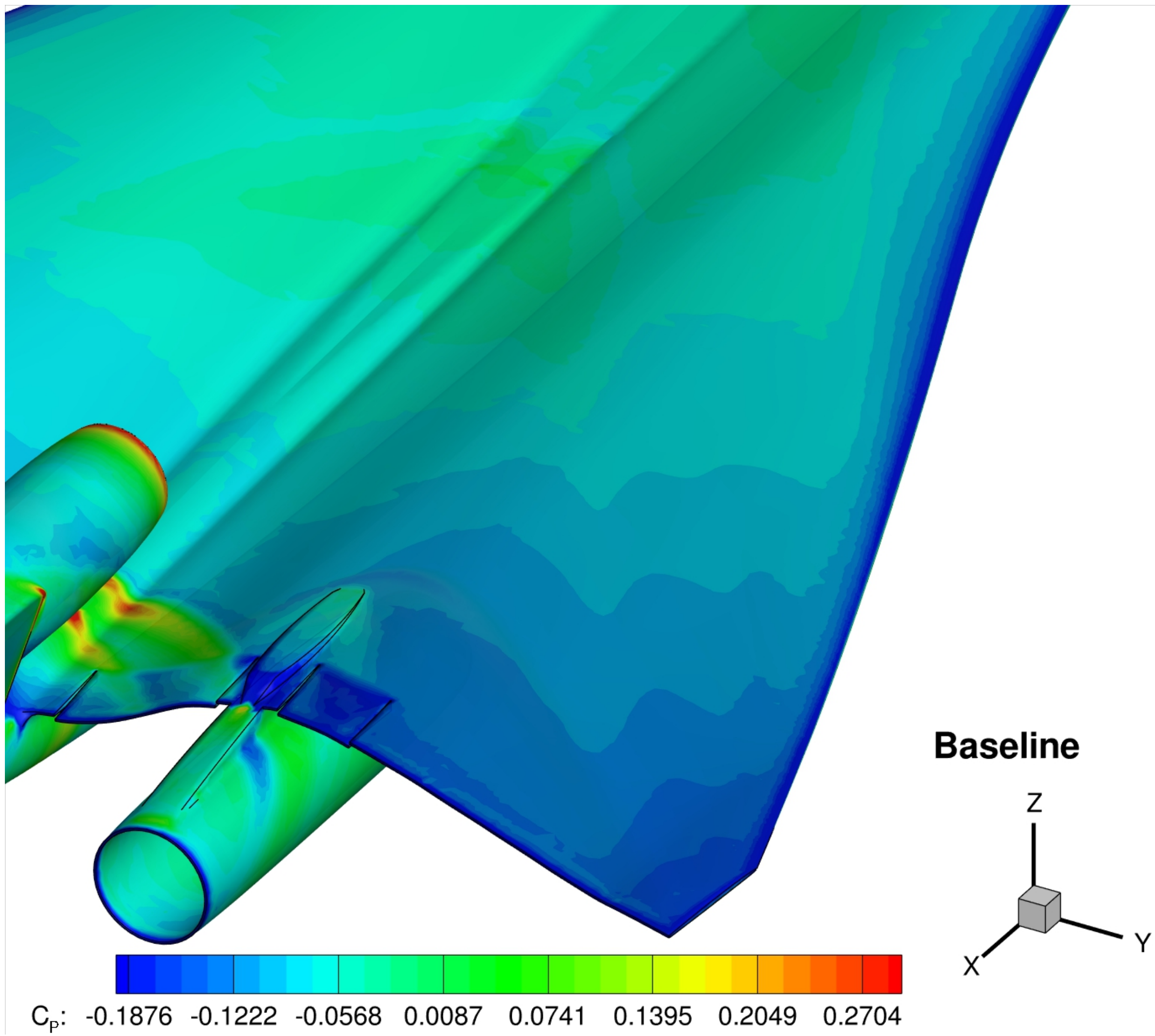
Maximum thickness constraint



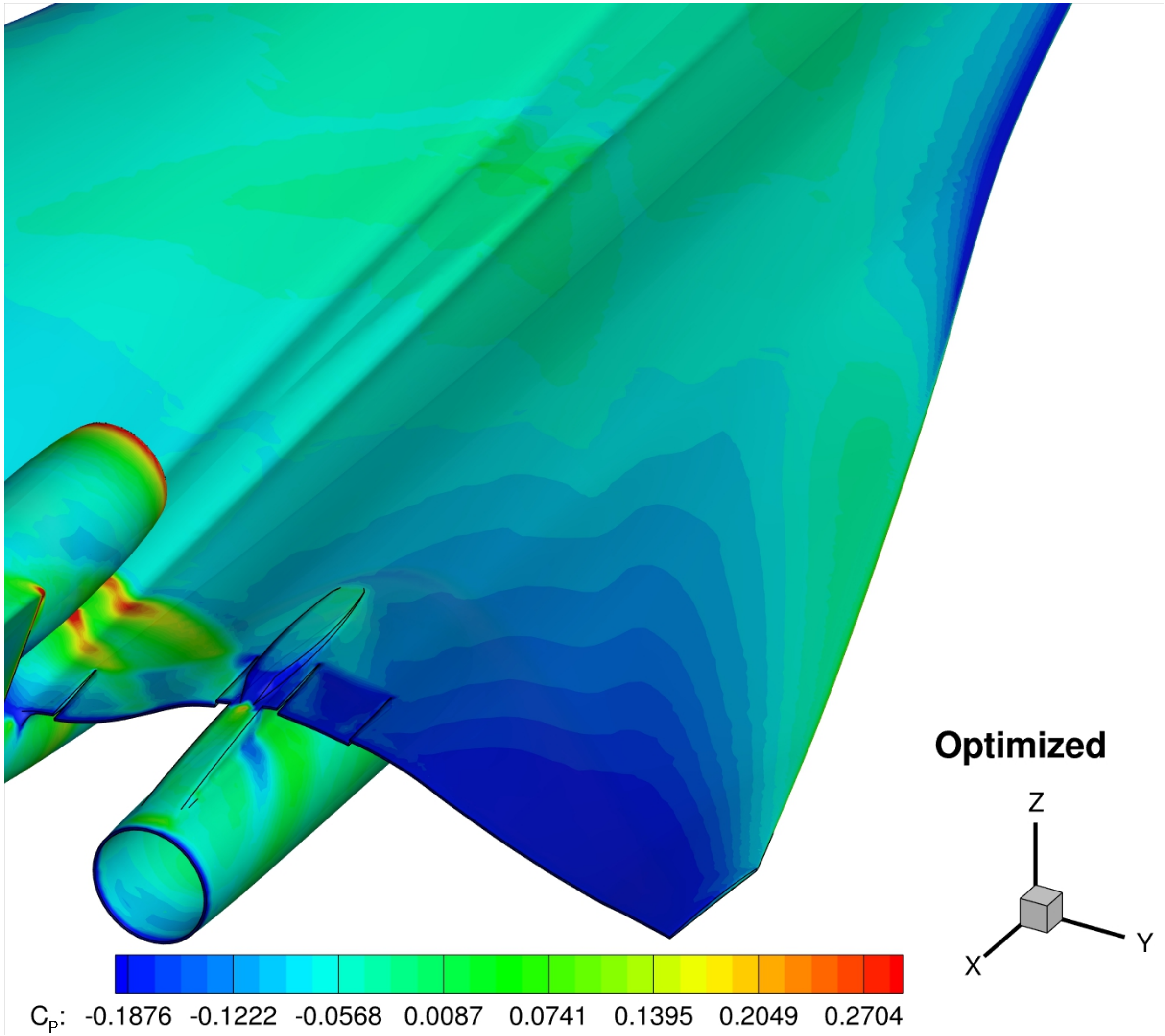
Twist angle constraint



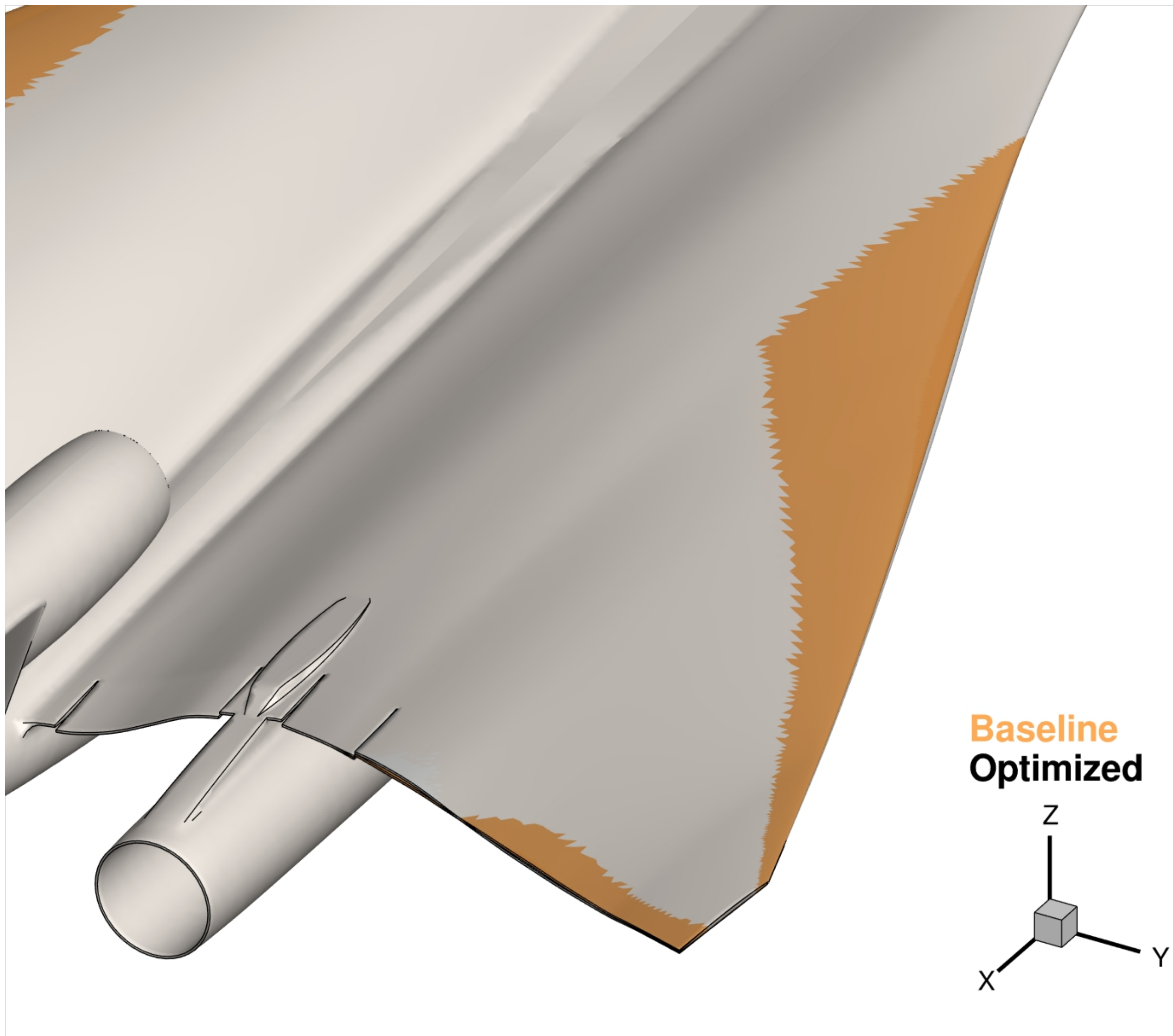
Baseline configuration. C_p distribution



Optimized configuration. C_p distribution



Geometry comparison

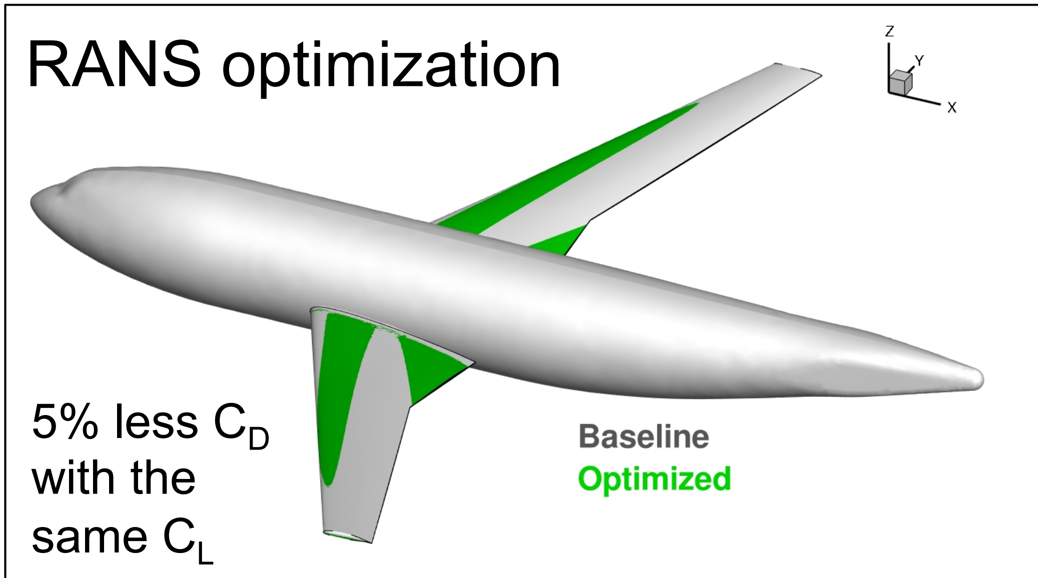


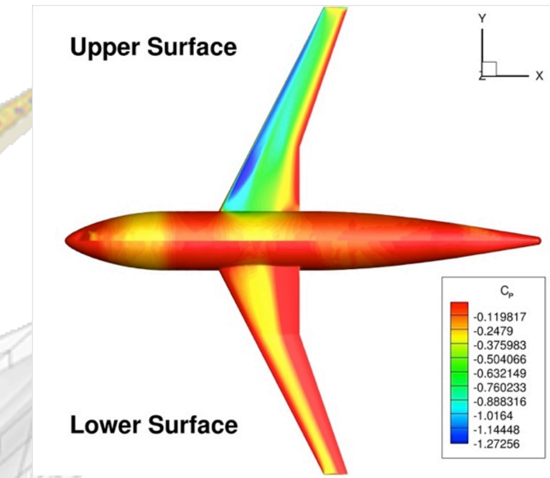
I) INTRODUCTION TO SHAPE DESIGN
II) 1ST OPTIMIZATION ATTEMPT
III) 2ND OPTIMIZATION ATTEMPT
IV) CONCLUSIONS

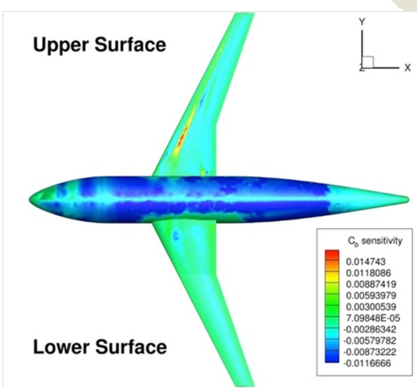
The Open-Source CFD Code

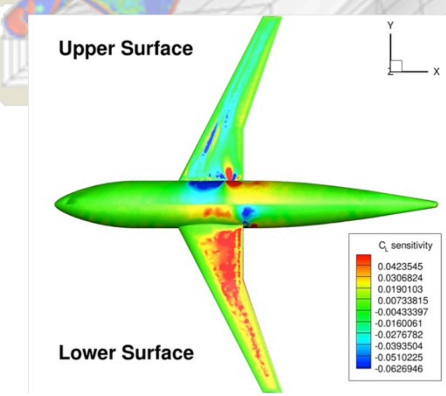
aerospacedesignlab

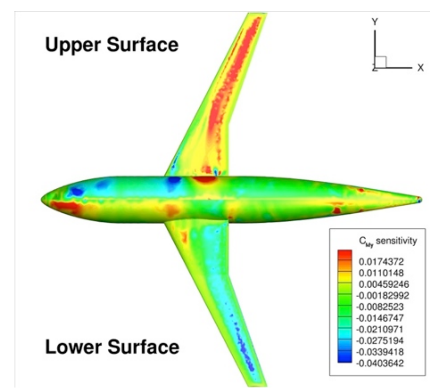
RANS optimization











SU2
The Open-Source CFD Code

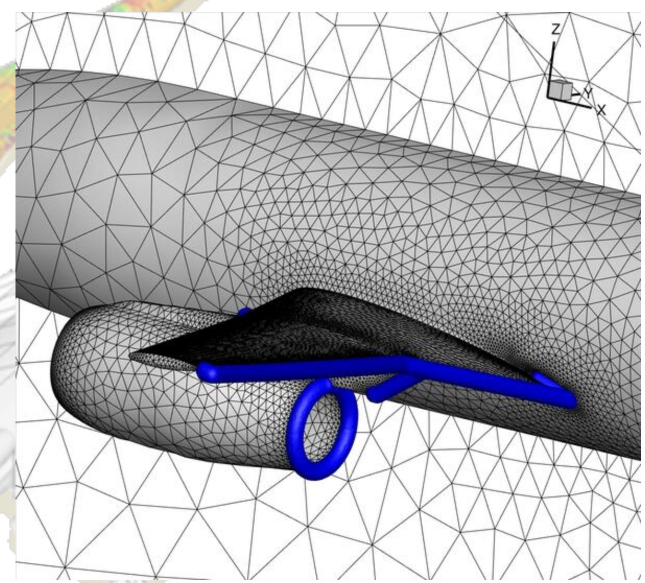
We’ve Only Scratched the Surface…

Objective Functions and Constraints:

- Drag, lift, moments.
- Stability derivatives.
- Load distribution on the wing.
- Free-surface problems.
- Near-field pressure.
- Equivalent area distribution.
- Multiple geometrical constraints using unstructured grids.

- Compressible and incompressible.
- Euler and RANS problems.
- Steady and unsteady applications.

Exclusive numerical schemes designed to solve the adjoint equations.



Once a design is finalized, SU2_MDC can export .stl files of the geometry to reintegrate with CAD/manufacturing

THANKS A LOT FOR YOUR ATTENTION!

QUESTIONS & ANSWERS

SU2

The Open-Source CFD Code

aerospacedesignlab