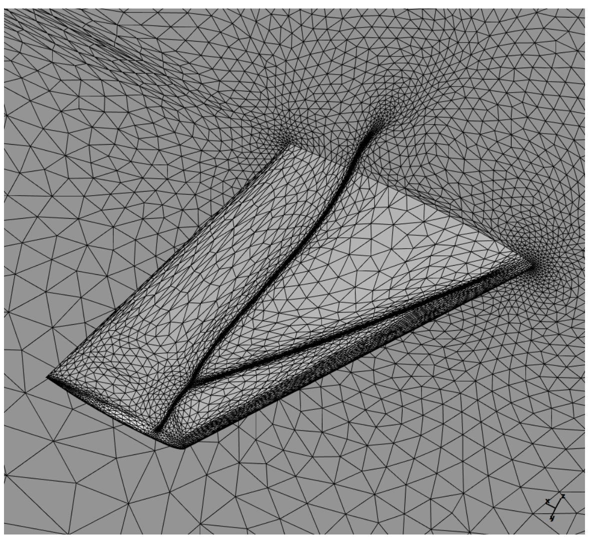
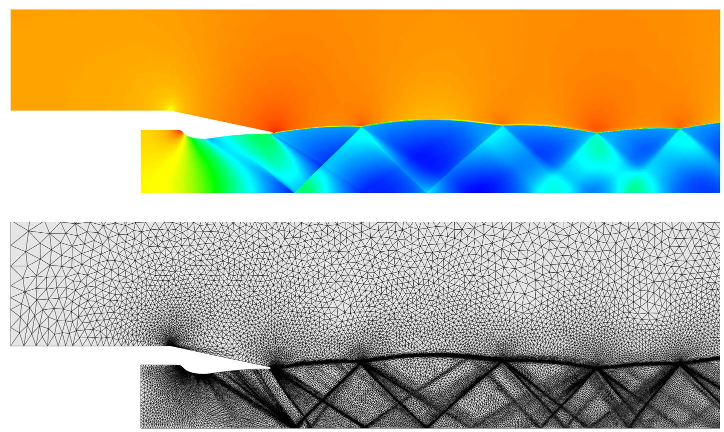
Mesh Adaptation for SU2 with the INRIA AMG Library

1st Annual SU2 Developers Meeting
TU Delft, The Netherlands
September 5, 2016

Juan J. Alonso, Thomas D. Economon, Victorien Menier
Department of Aeronautics & Astronautics
Stanford University





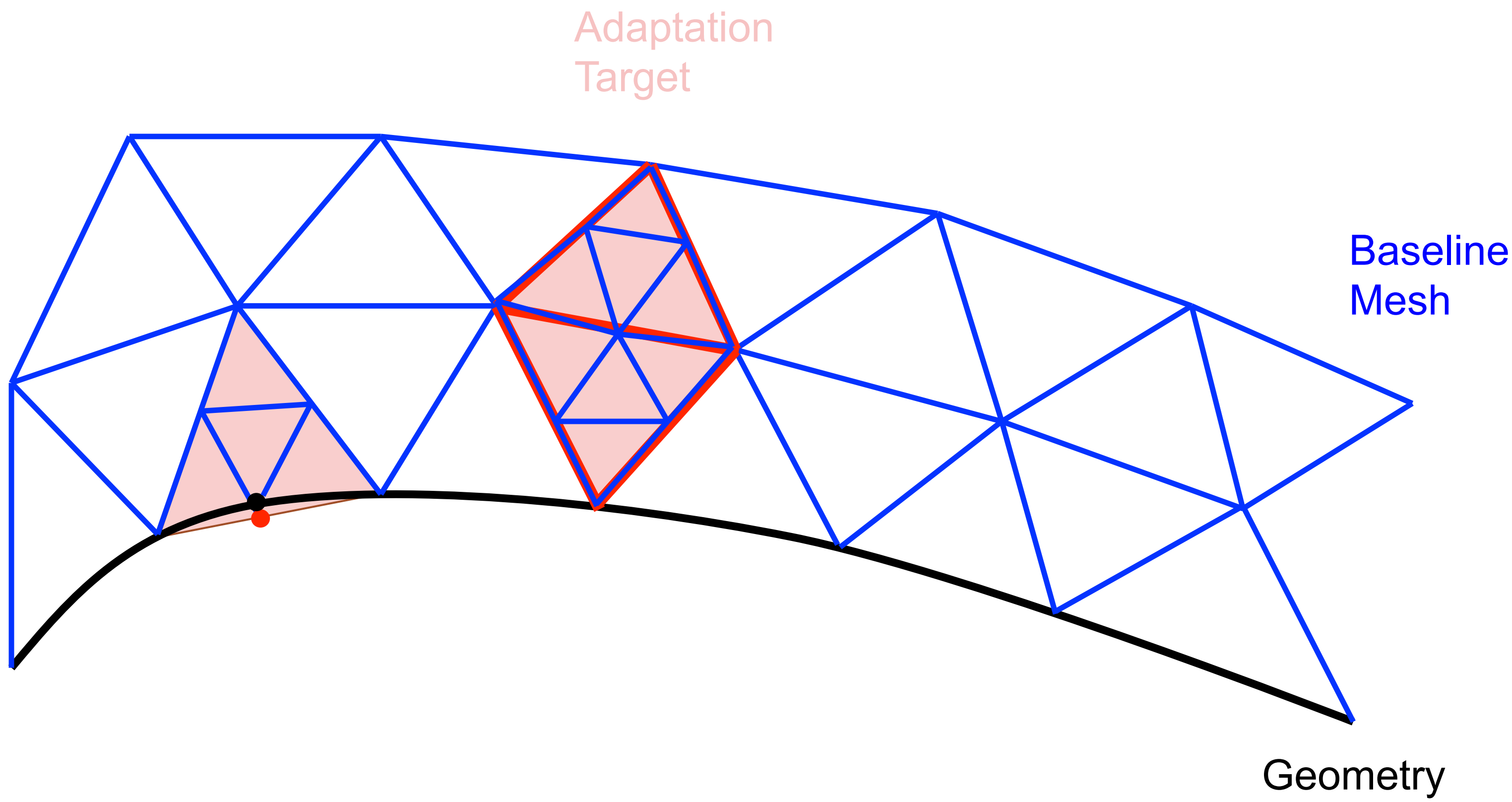
Why is SU2 based on unstructured grids?

- The choice of an underlying unstructured topology in SU2 was carefully decided on for several reasons:
  - Ability to quickly mesh complex (entire aircraft) configurations

  …and…

  - Ability to perform **solution-based mesh adaptation**
- In the early releases of SU2, a rudimentary mesh adaptation code (SU2_MAC) was included and exercised in some simple situations
- SU2_MAC was rolled into SU2_MSH (in current github repository)
- But we have not pushed the boundaries in earnest
- Time to start doing something about it!

Grid Adaptation 101



Mesh adaptation approaches

- Traditionally based on local gradients of quantities of interest: pressure, density, or locally flow-aligned gradients such as:

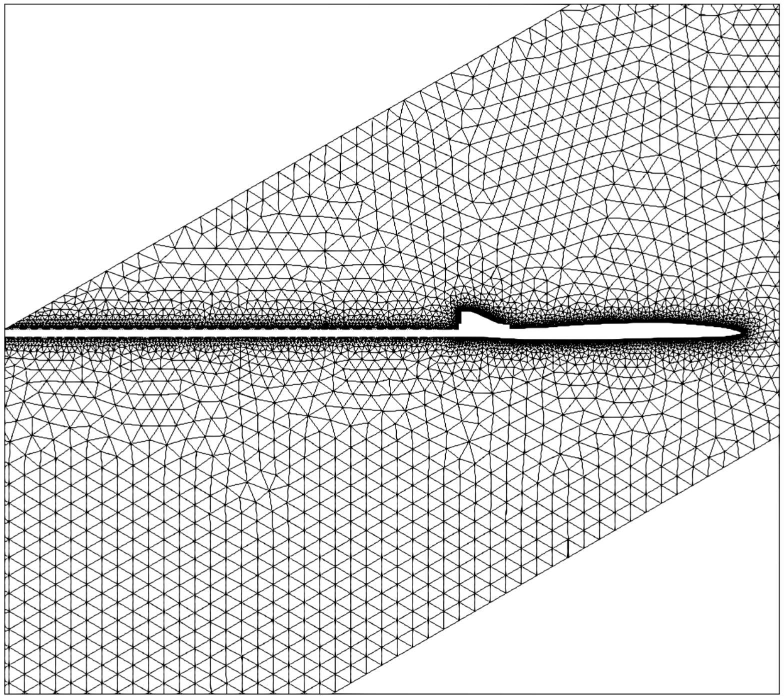
$$\epsilon = \frac{\mathbf{V}}{c} \cdot \frac{\nabla p}{|\nabla p|} \triangle x$$

- Functional-based solution adaption is based on the theory of numerical error estimation and is based on adjoint formulations (available in SU2!)

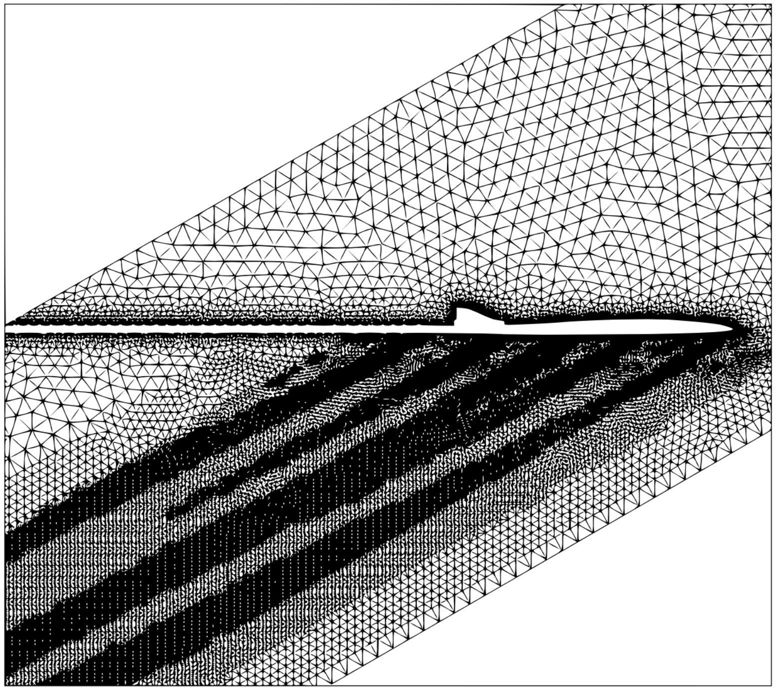
- Isotropic vs. non-isotropic methods

- Inviscid vs RANS

Mesh adaptation



Baseline Grid



Adapted Grid

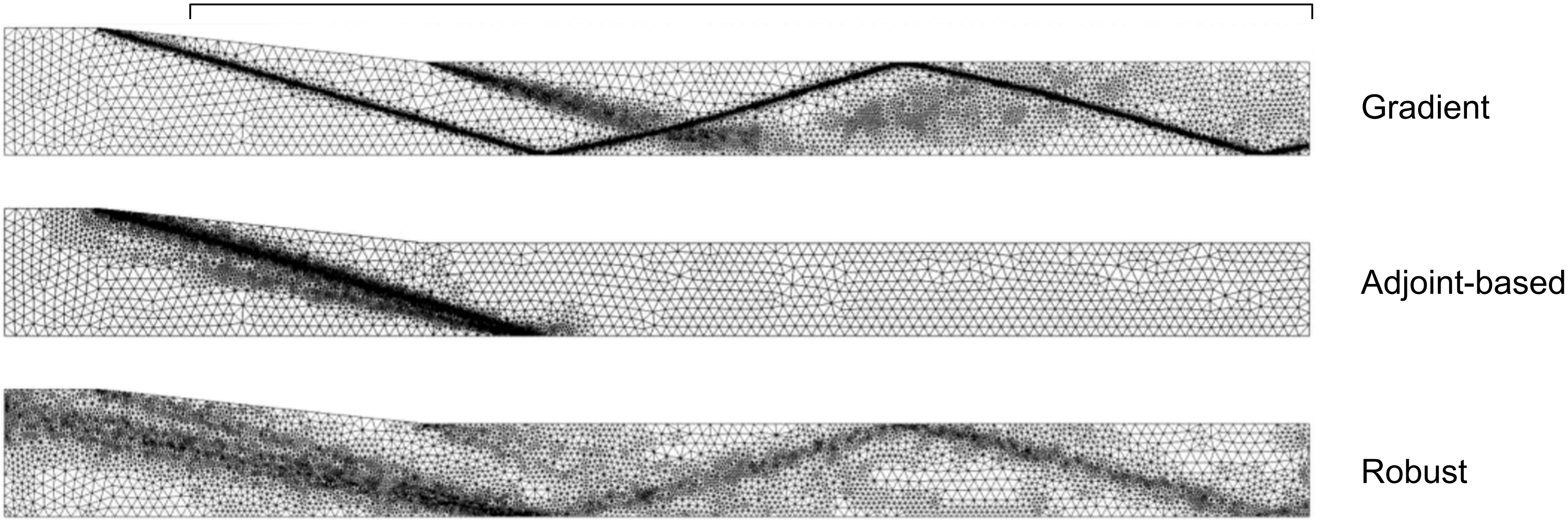
- Adapt meshes were only where required
- Adaption indicator must be constructed carefully

Choi, Alonso, van der Weide, “Numerical and Mesh Resolution Requirements for Accurate Sonic Boom Prediction”
AIAA Journal of Aircraft, vol. 46, 2009

Some previous efforts

SU2
The Open-Source CFD Code

- Some efforts were pursued to ensure that, during UQ simulations, the mesh only had to be adapted once
- The adaption estimator ensured that the change in numerical error as minimized

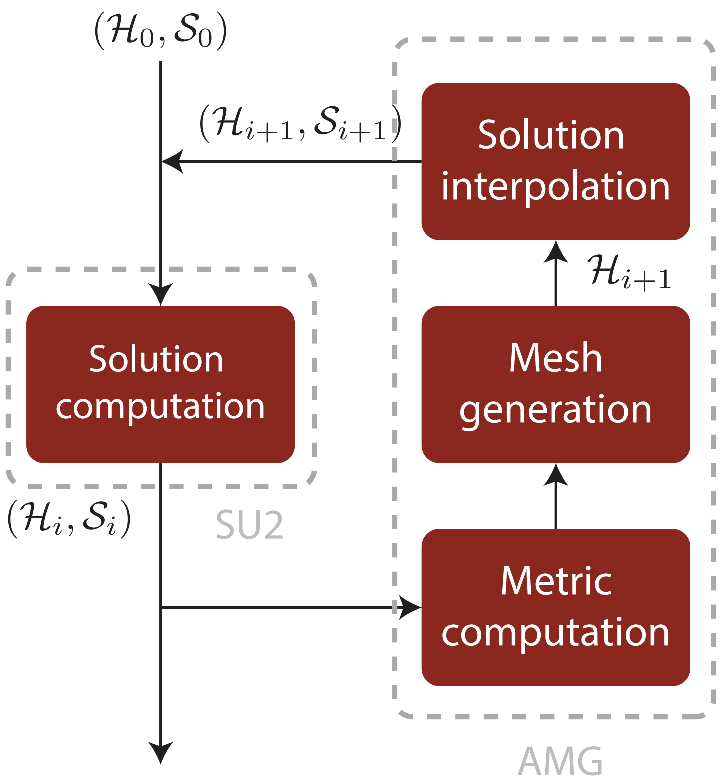


technique is proposed that is aimed at minimizing the numerical error over a range of variations of the uncertain parameters of interest about a nominal state. Using this approach, it is possible to generate computational grids that are insensitive to small variations of the uncertain parameters that can both locally and globally change the solution and, as a result, the error distribution. This is in contrast with classical adjoint techniques, which seek to adapt the grid with the aim of minimizing numerical errors for a specific flow condition (and geometry). It is demonstrated that flow computations on these robust grids result in low numerical errors under the expected range of variations of the uncertain input parameters. The effectiveness of this strategy is demonstrated in problems involving the Poisson equation and the Euler equations at transonic and supersonic/hypersonic speeds.

Recent Efforts

- Recently we have been collaborating with INRIA (Gamma group, Adrien Loseille) to leverage their AMG adaptation library
- Pursued 2 separate geometry tools for point projection:
  - GE Lite library from Pointwise
  - EGADS (Engineering Geometry Aircraft Design System) from Haimes / Dannenhoffer
- Functional prototype in place being improved over the next 6 months
- Final objective is to be able to do shape optimization with mesh adaptation in the loop

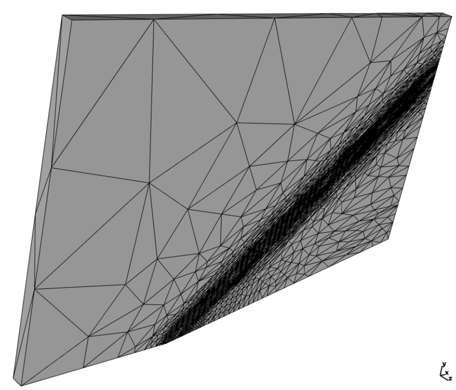
SU2/AMG mesh adaptation process



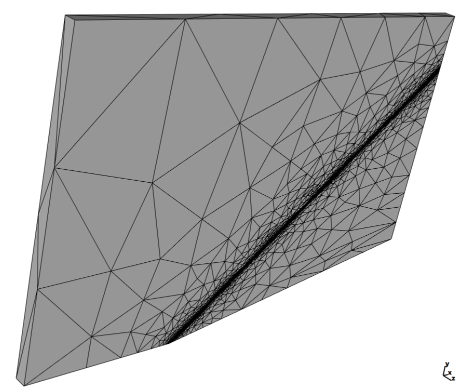
Mesh adaptation : a non-linear problem where both the mesh and the solution are converged to an optimal state.

Mesh sizes are progressively increased at each iteration.

Example: 3D supersonic wedge:



Iter 1



Iter 3

Required tools:

- **SU2_CFD:** performs the flow simulation
- **AMG (INRIA) :** computes a metric from the solution, remeshes, and interpolates the solution onto the new adapted mesh
- **mesh_adaptation.py:** automates the entire mesh adaptation process by executing SU2 and AMG according to a simple configuration file

AMG/SU2 interface

Command line:

```
$ python mesh_adaptation.py -f config.cfg
```

Specific parameters to be added to the configuration file:

- Mesh sizes: what approximate numbers of vertices are wanted at each iteration? How many sub-iterations for each mesh size?
- Metric computation parameters: which sensor? What error estimate?
- Surface re-projection: use a CAD (GELite or EGADS?) representation of the geometry? Or a fine background mesh?
- Remesher: minimal/maximal edge sizes, gradation etc.

config.cfg

```
% ------- MESH ADAPTATION PARAMETERS ------

% Desired mesh sizes (appr. number of vertices)
ADAP_mesh_sizes= (1e6, 2e6)

% Number of sub-iterations for each mesh size
ADAP_SUBITE= (3, 3)

% Use an initial solution? (YES or NO)
ADAP_RESTART= NO
% Name of the initial mesh
ADAP_INI_MESH_FILE= m6_wing.meshb
% (if YES) name of the initial SU2 restart solution
ADAP_INI_RESTART_FILE= m6_wing_restart.solb
% (if YES) name of the initial sensor solution
ADAP_INI_SENSOR_FILE= m6_wing_mach.solb

% Surface reprojection (CAD/BACK_MESH/NONE)
ADAP_PROJ_METHOD= CAD
% If not NONE, specify the name of the cad model/back mesh file
ADAP_BACK_NAME= m6_wing_cad.iges

% Maximal edge sized allowed through the mesh adaptation
ADAP_HMAX= 70
% Minimal edge size
ADAP_HMIN= 0.00001
% Required mesh gradation
ADAP_HGRAD= 1.3
```

Adaptive remesher: AMG (Inria)

SU2
The Open-Source CFD Code

Capabilities:

- Anisotropic remeshing through local modifications of the mesh [Loseille and Menier, IMR 2013]
- Metric computation
- Metric field correction using anisotropic gradation
- Solution interpolation
- Parallel anisotropic mesh adaptation: 1 billion tetrahedra in less than 20 minutes on 120 cores. [Loseille, Menier, Alauzet, IMR 2015]

Nozzle Aero-Thermal-Structural Design

Inspired by the X-47B aircraft

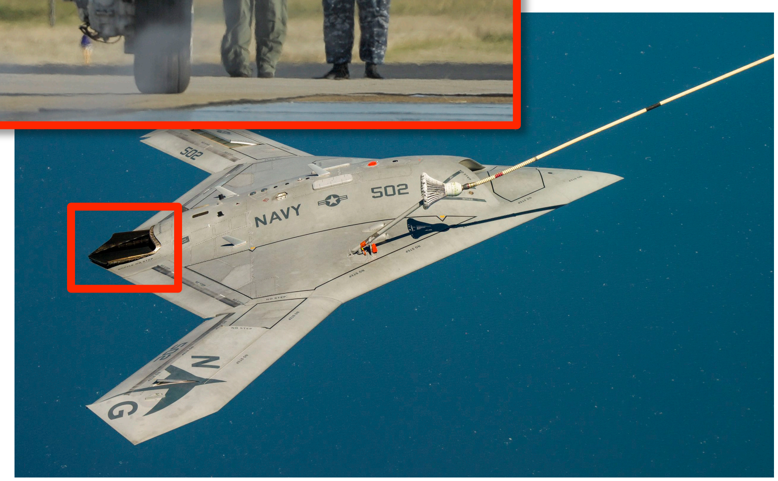
The Open-Source CFD Code

50
215
6 - 15.6
h subsonic
000
0-PW-20

system weight

- Uncertainties in all areas of multi-physics problem

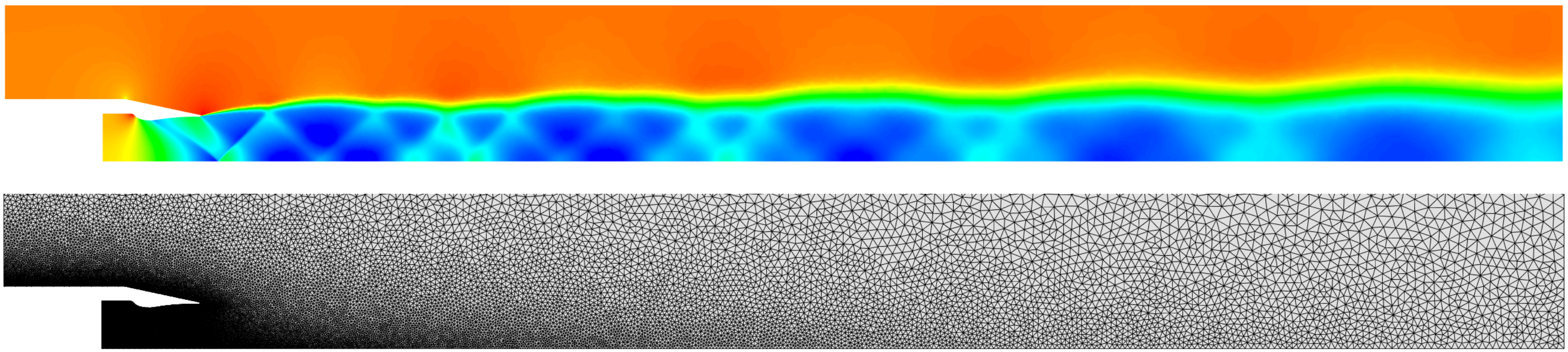
IHS Jane's Unmanned Aerial Vehicles and Targets, 2015
Ferguson et al, Virginia Tech X-47 A/B student presentation, 2015
bottom image: Northrop Grumman X-47B UCAS Data-sheet, 2015



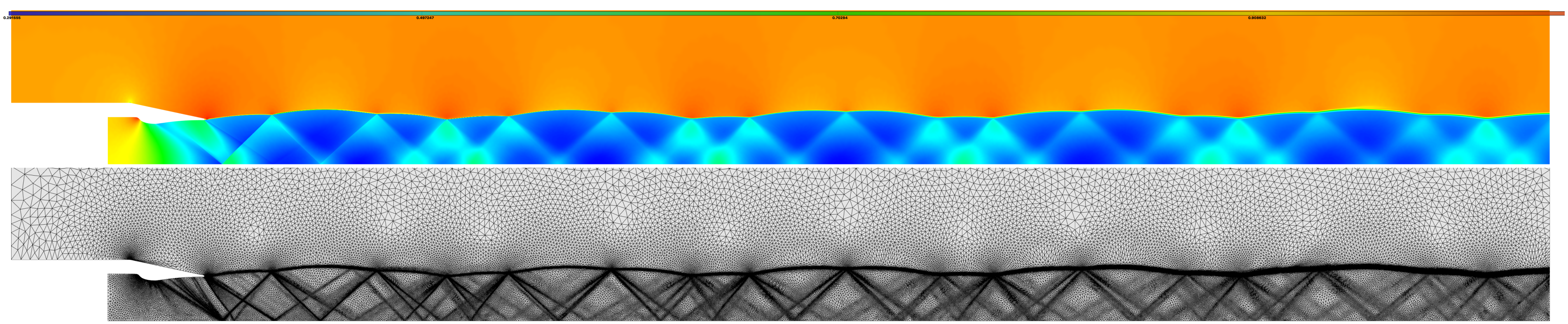
DARPA EQUiPS Nozzle Problem

Hessian-based mesh adaptation of the Mach number.
Using SU2 and AMG
10 adaptive iterations, 20 minutes on a laptop

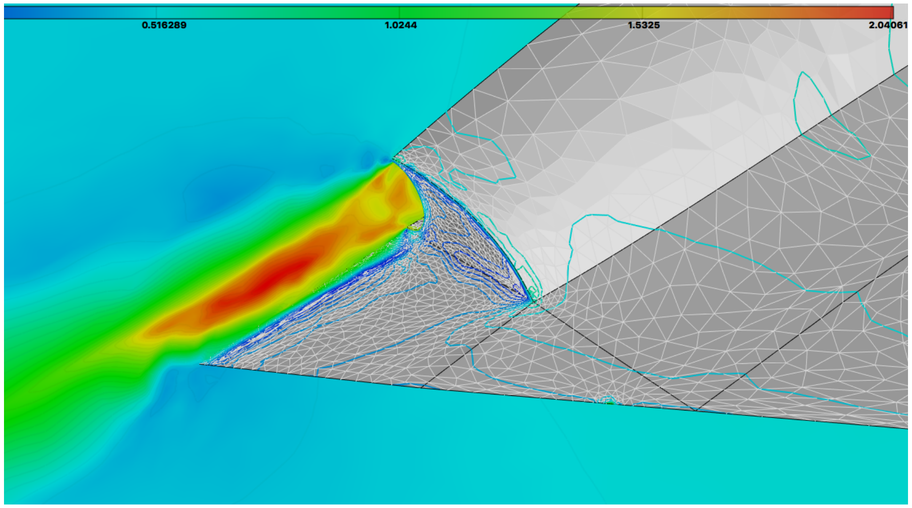
- Baseline mesh and solution

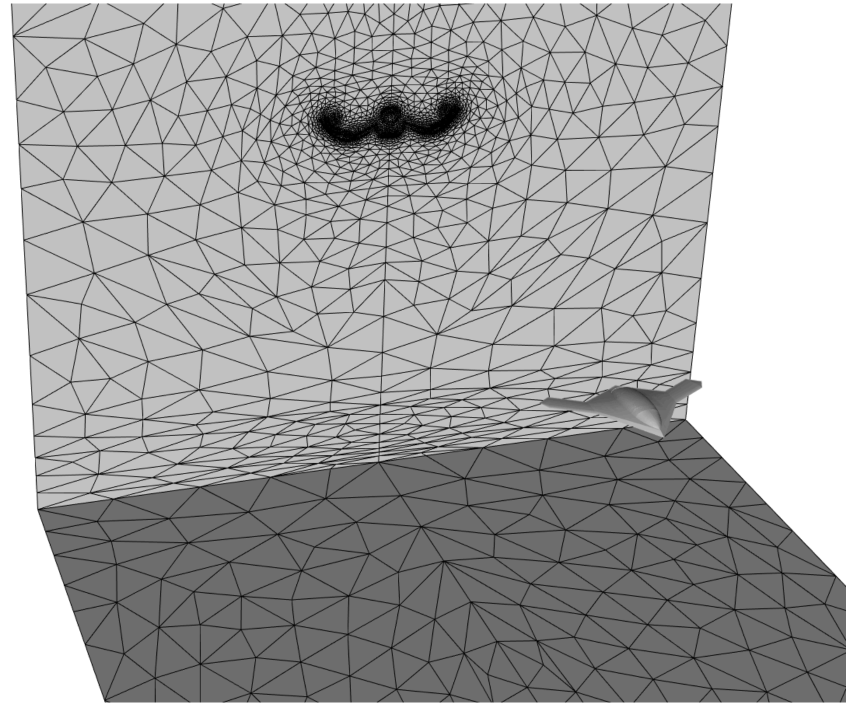


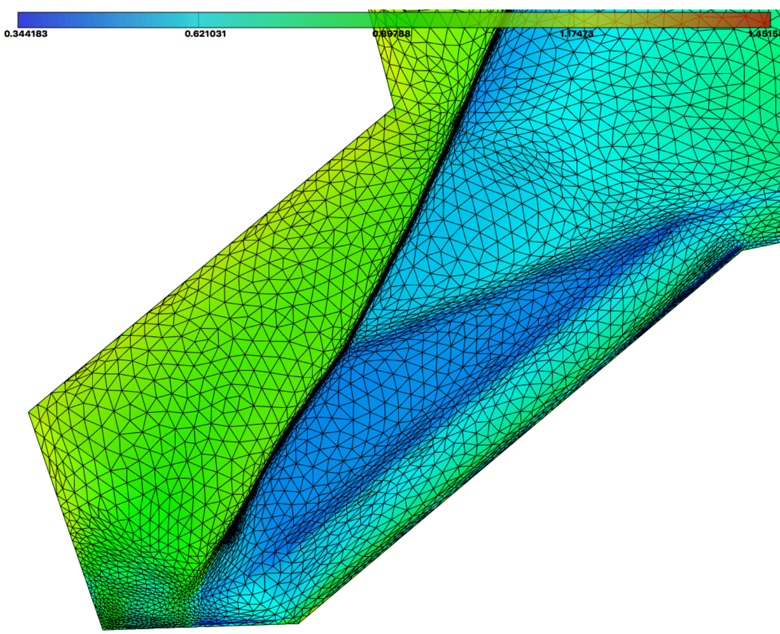
- Adapted mesh and solution

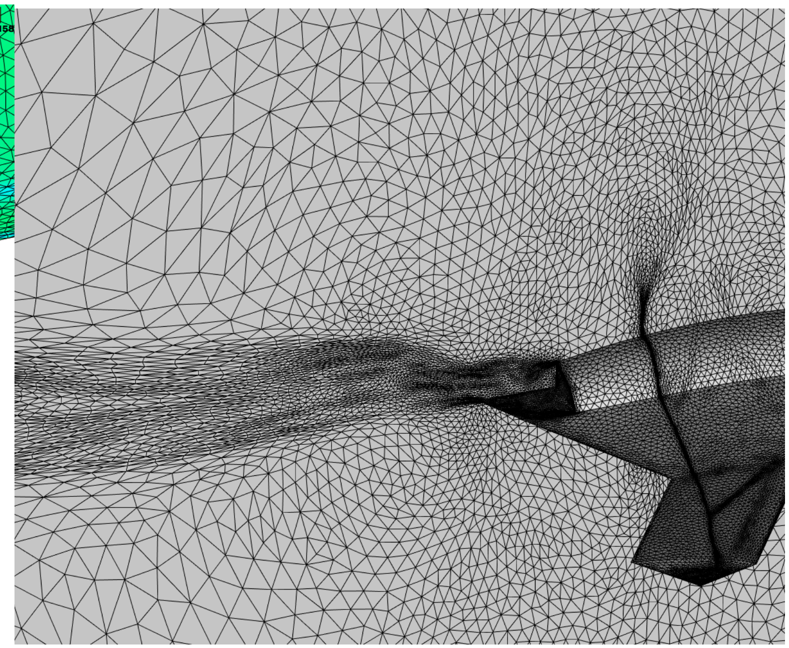


Example : X-47B





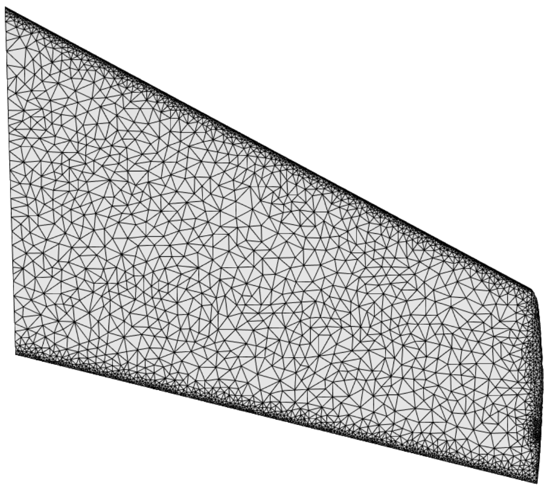


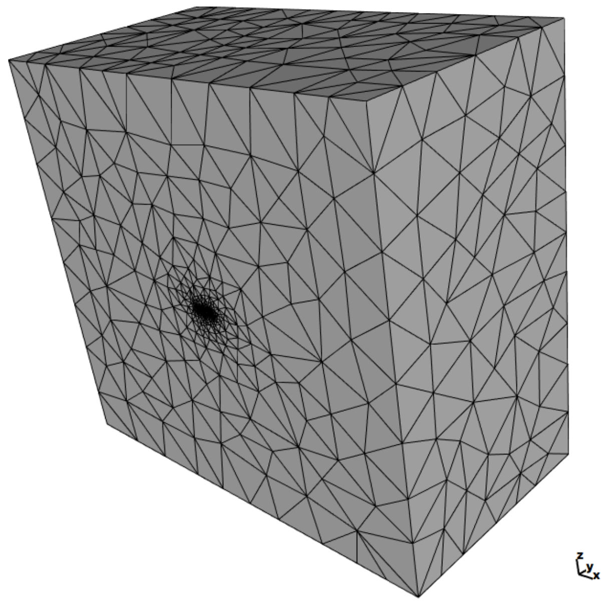


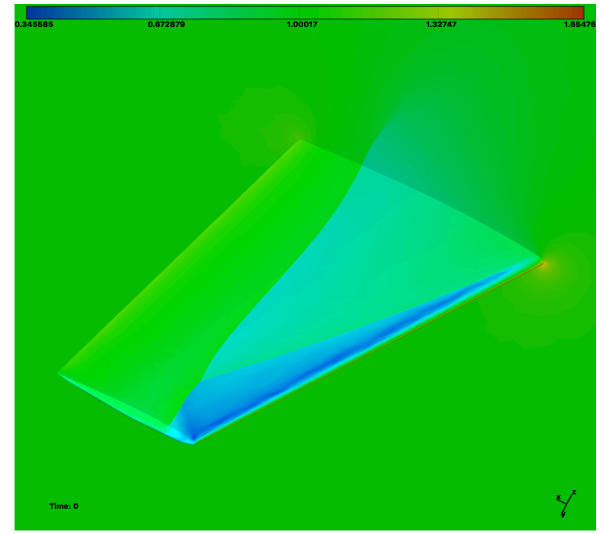
ONERA M6 wing

SU2 The Open-Source CFD Code

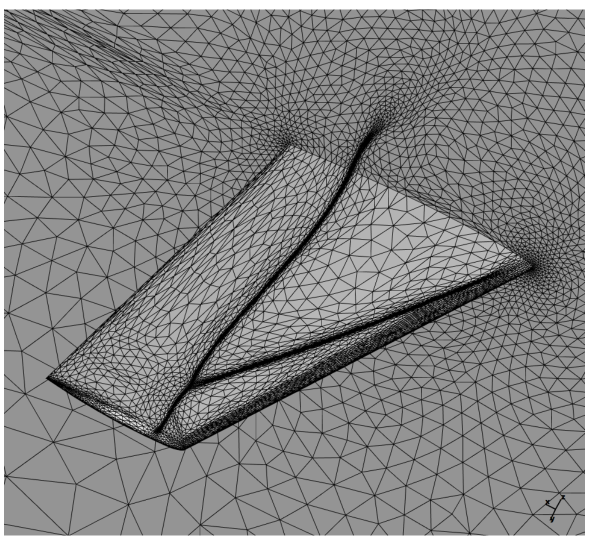
- Baseline mesh

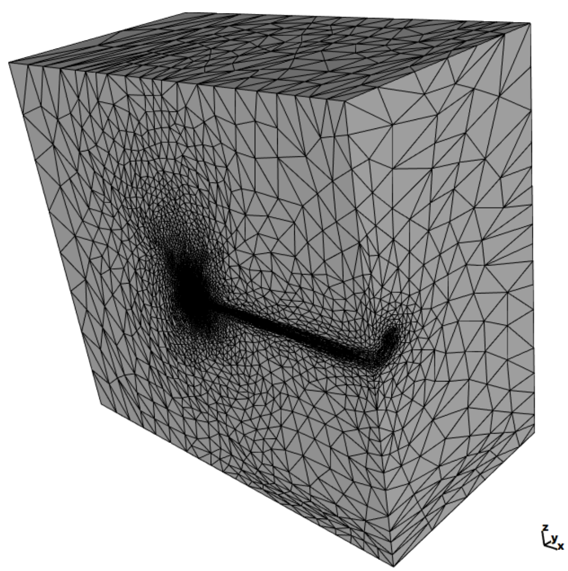


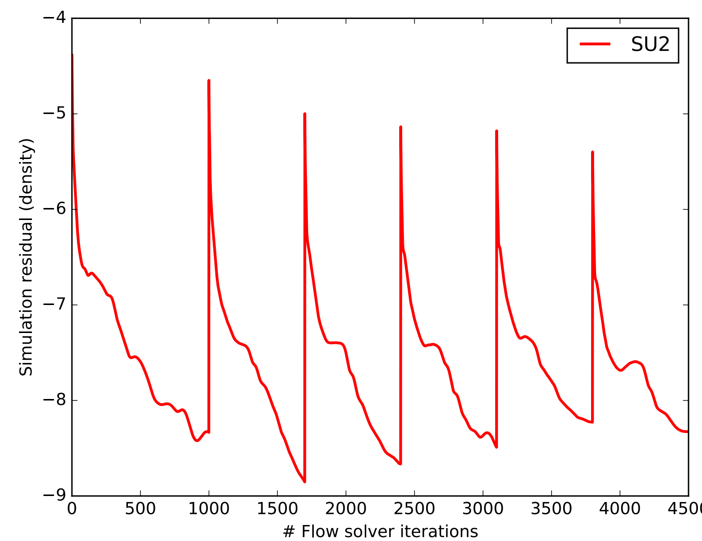




- Adaptation

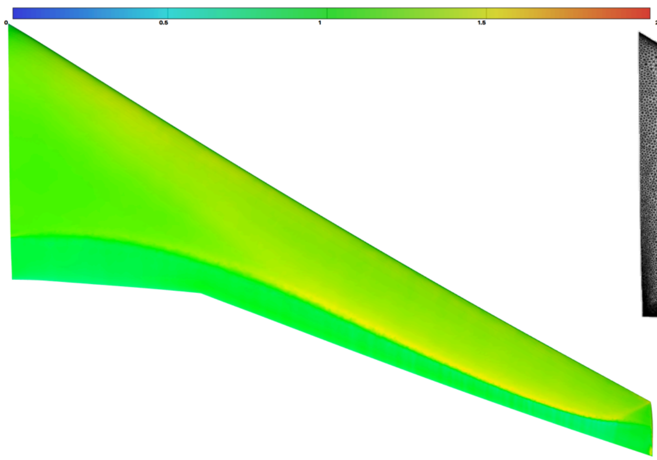


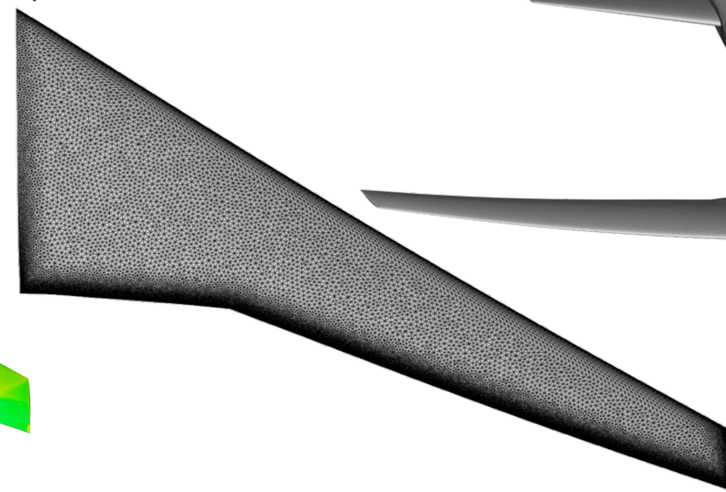


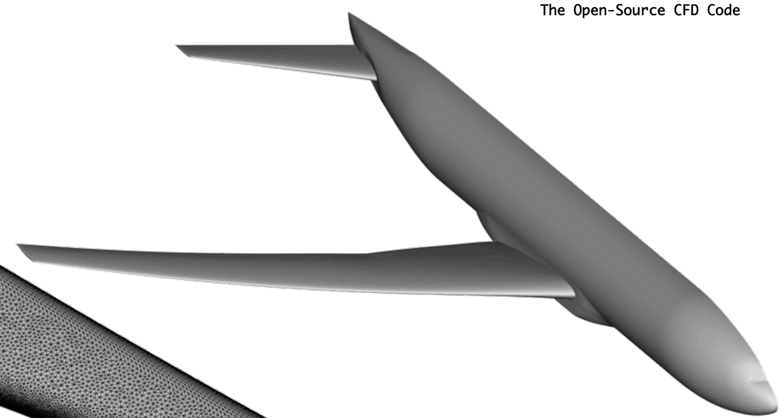


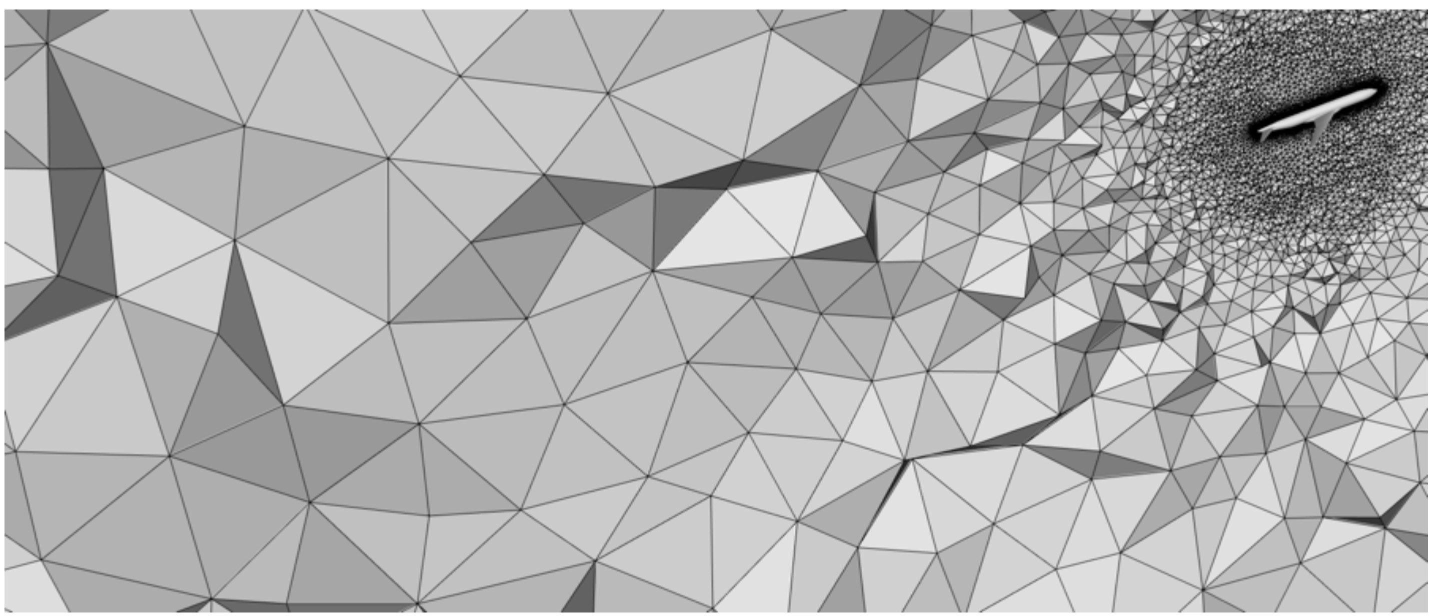
CRM : baseline

SU2
The Open-Source CFD Code



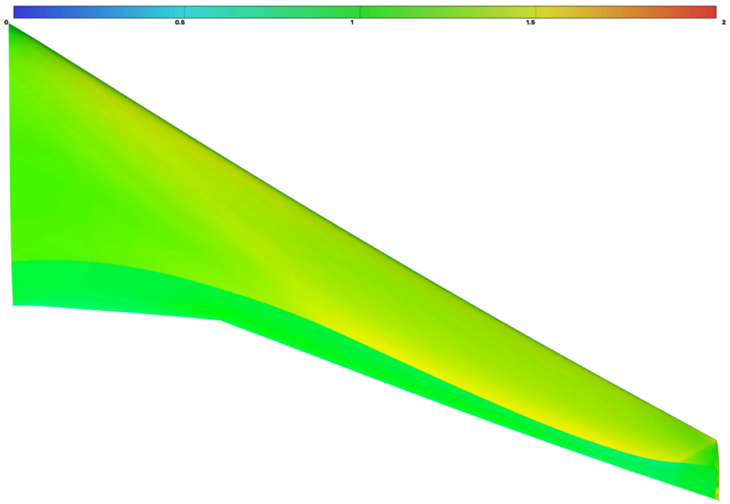


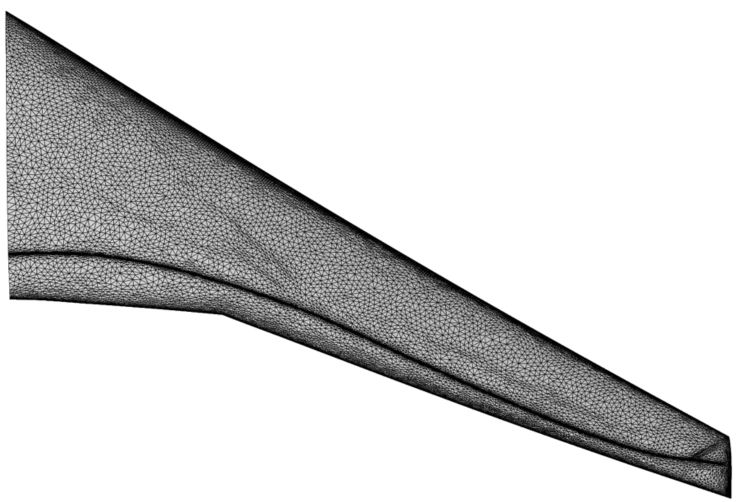


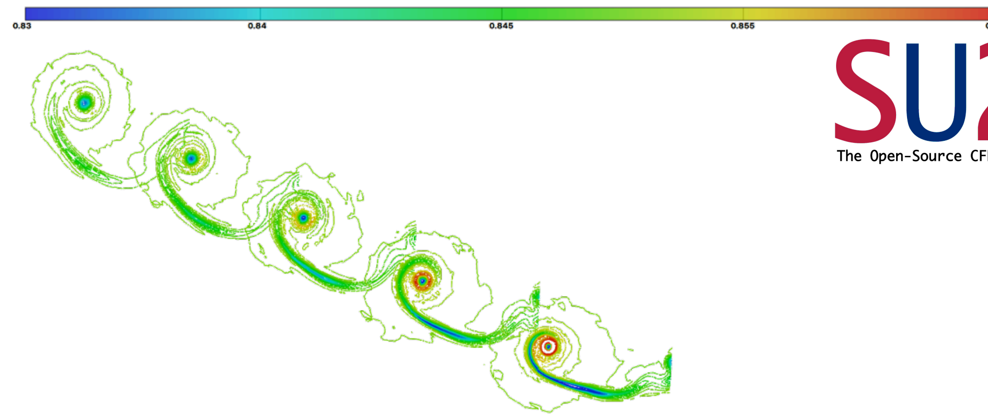


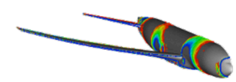
CRM: adapted

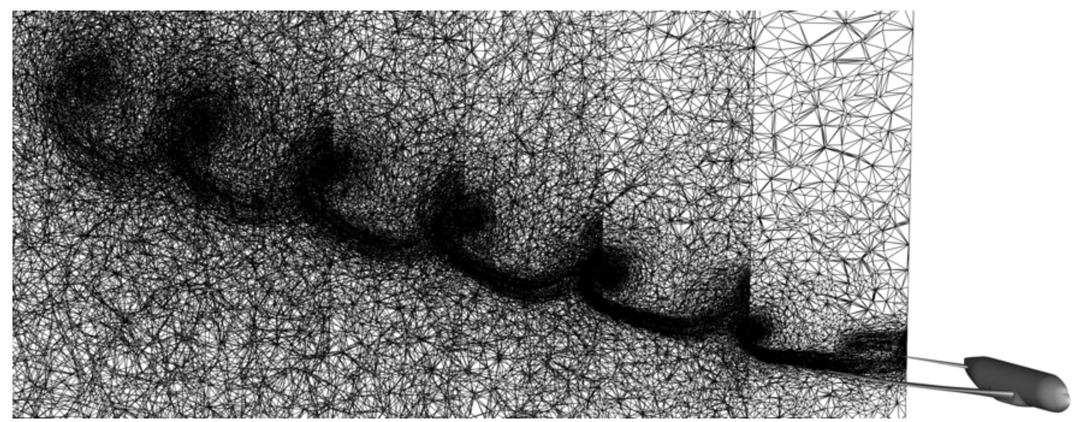
SU2
The Open-Source CFD Code

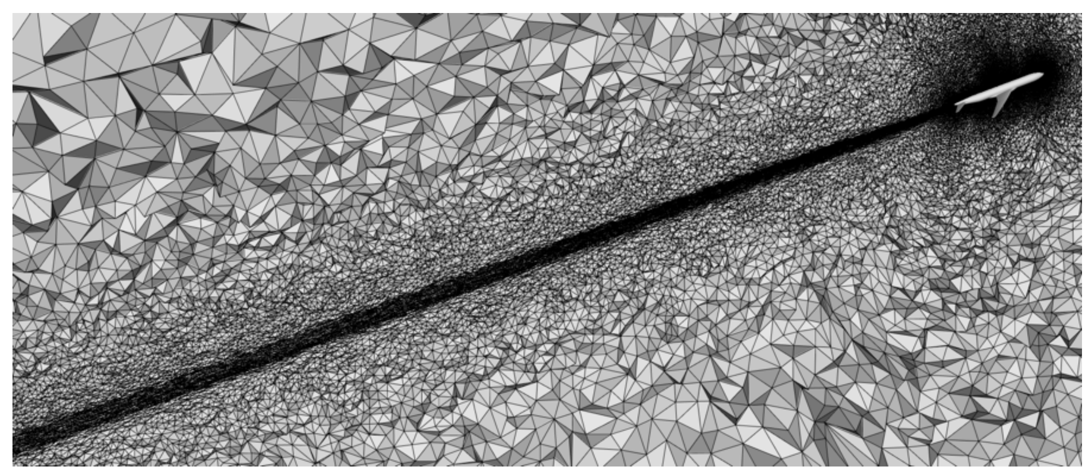






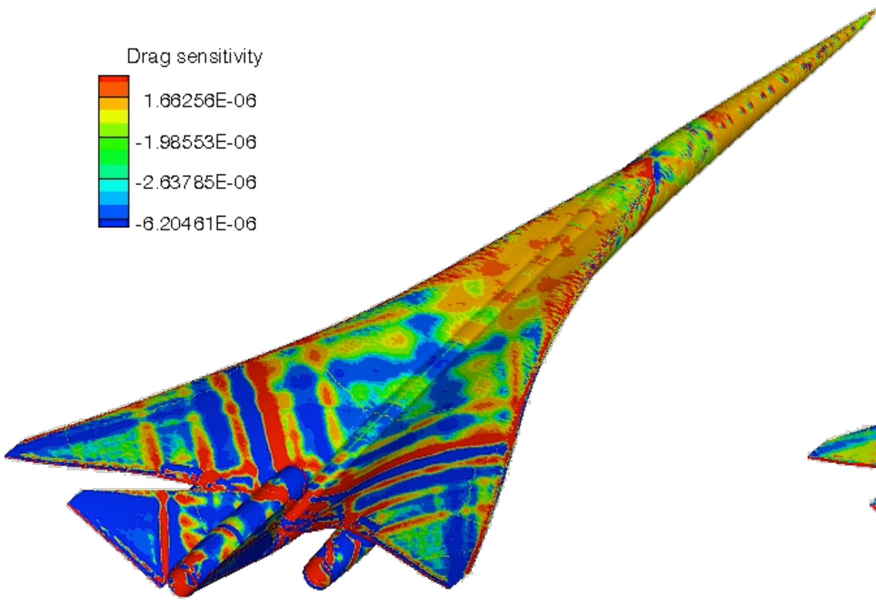




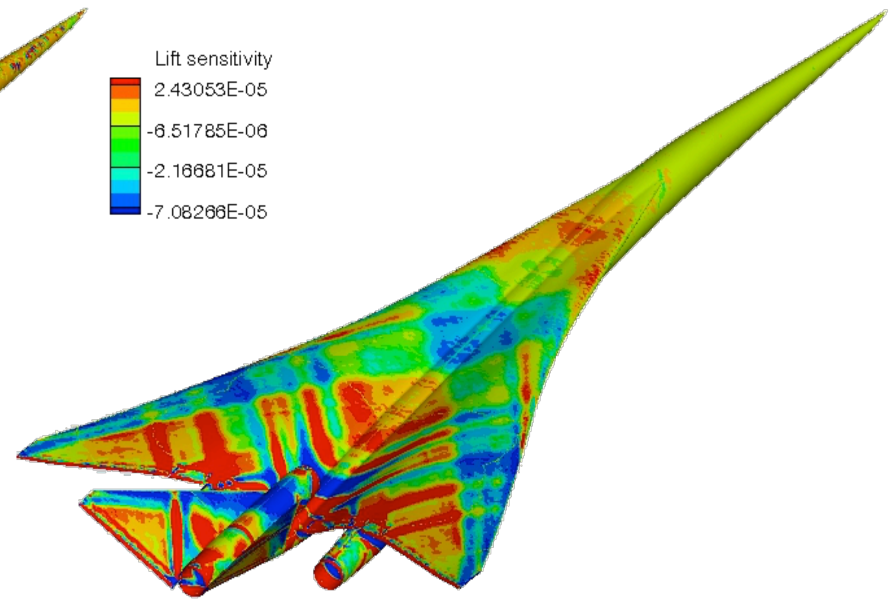


Supersonic aircraft design
Surface drag and lift sensitivities using the adjoint method

Magnitude of surface sensitivity represents changes in cost function caused by changes in geometry.



Drag sensitivity (Mach 1.6, AoA 2.3deg)

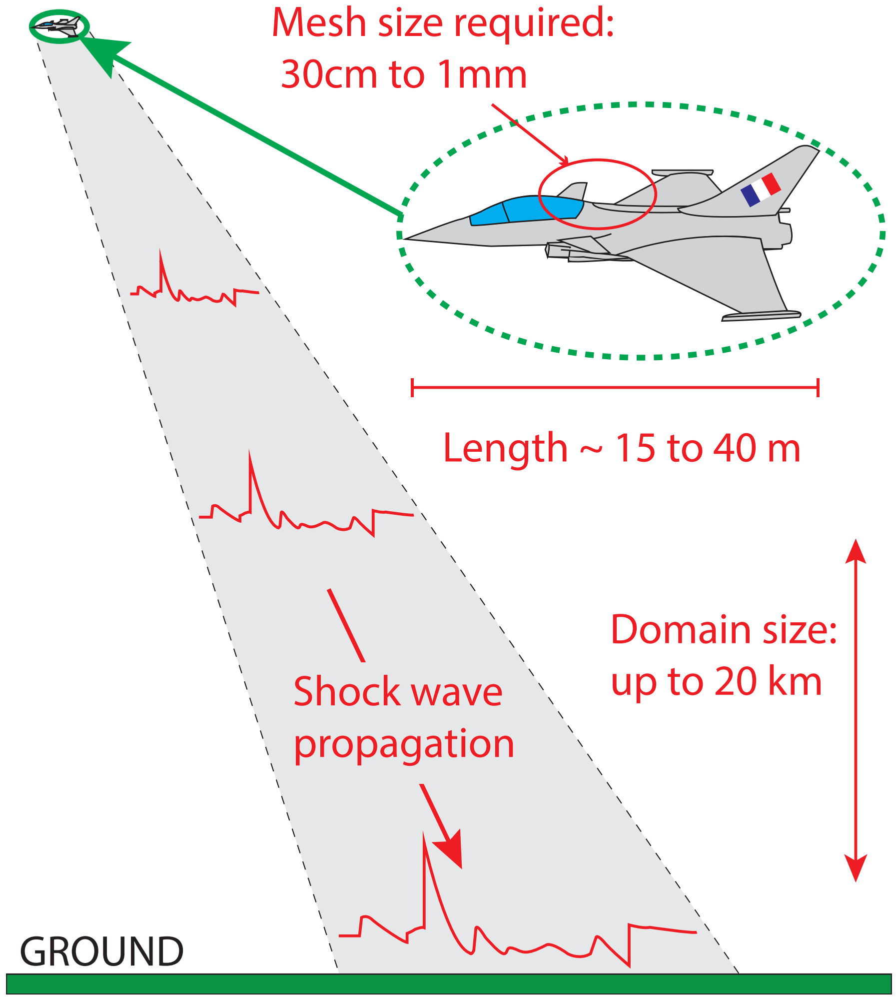


Lift sensitivity (Mach 1.6, AoA 2.3deg)

Designers can use this sensitivity information to determine appropriate parameterizations of the configuration prior to optimization.

Sonic boom prediction

SU2
The Open-Source CFD Code

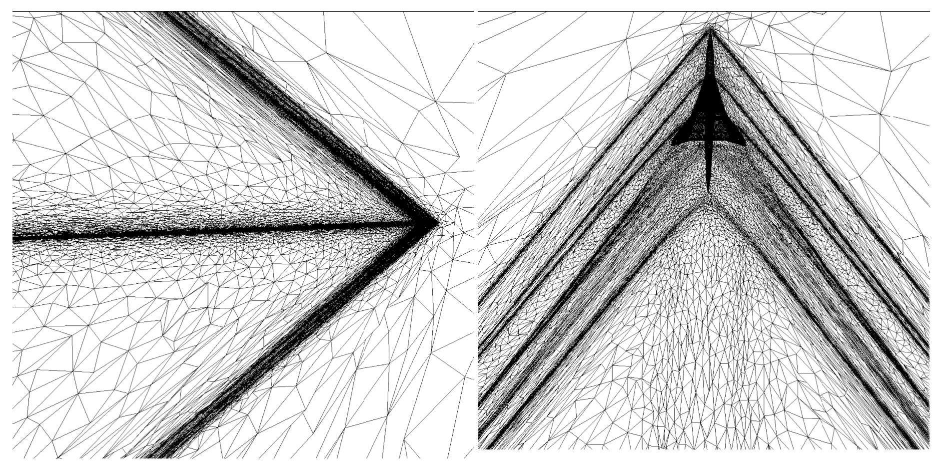


Without mesh adaptation :

1m precision ➔ 200 Billion DoF

Using mesh adaptation

➔ 0.1 Billion DoF



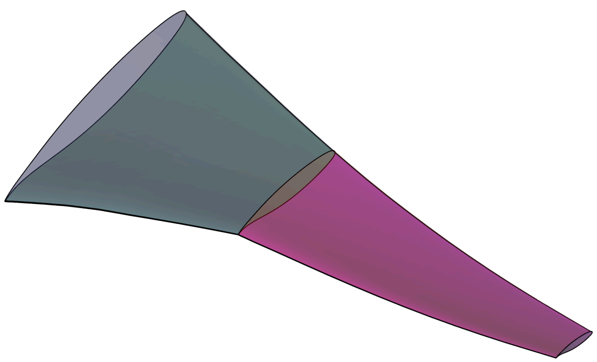
The problem complexity is decreased by 3 orders of magnitude just by modifying the discretization.

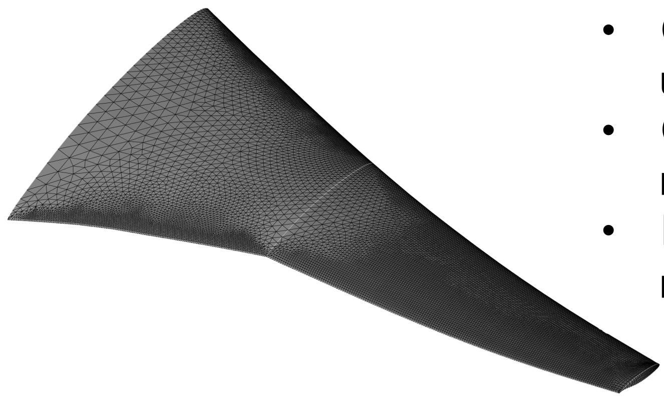
On-going: mesh adaptive shape optimization

Optimal shape algorithm coupled with mesh adaptation:

- The quantities of interest are computed on adapted meshes.
- A mesh adaptation loop is embedded inside each optimizer iteration.

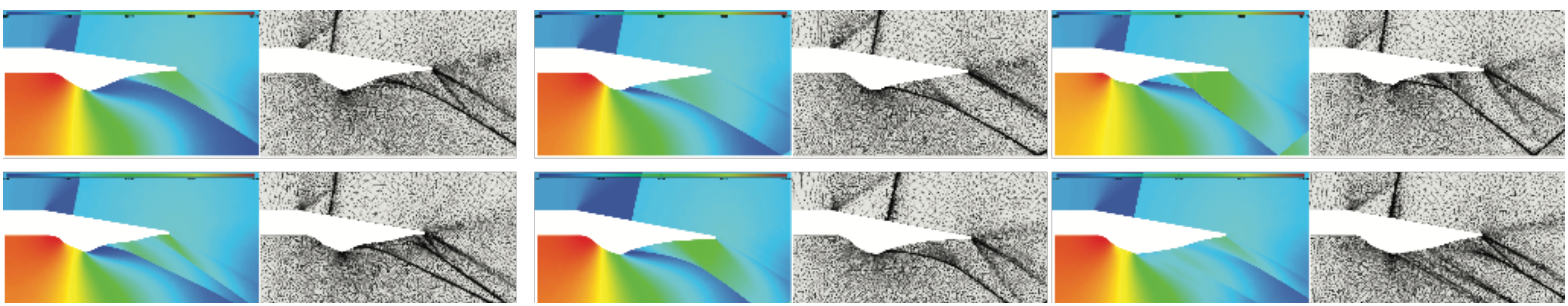
- Preliminary work : optimal shape of a CRM wing





- CAD model created from scratch using EGADS.
- Comparison of surface projection methods (EGADS, GELite).
- Feature-based and goal-oriented mesh adaptation.

- On-going: Design Under Uncertainty of a Nozzle geometry



SU2

The Open-Source CFD Code

https://su2.stanford.edu

https://github.com/su2code

intel

Argonne NATIONAL LABORATORY

SU2
The Open-Source CFD Code

TU Delft

UNIVERSITY OF TWENTE.

Stanford | ENGINEERING
Aeronautics & Astronautics

POLITECNICO MILANO

LOCKHEED MARTIN

Imperial College London

Scientific Computing
Technische Universität Kaiserslautern

Universitas Leodiensis

BOEING

Inria
INVENTEURS DU MONDE NUMÉRIQUE

BOSCH

EMBRAER