Tutorial 2:
Optimization & Python Scripts

SU2 Workshop Feb 3rd 2017
Heather Kline

Modified from presentations by
Francisco Palacios & T.D Economon

Shape Optimization

- Tutorial Files & Settings
- Start the Simulation
- Introduction to Optimization
- Optimization Problem Settings & Workflow
- Shape_optimization.py and Other Python Scripts

Tutorial Files And Required Settings

- Set PYTHONPATH (if not already done):
  export SU2_RUN=<...../bin/> (path to SU2_CFD, etc)
  export PYTHONPATH=$PYTHONPATH:$SU2_RUN
  - Python scripts require the path in order to find all the functions that are defined in subfolders (Advanced: see SU2_PY/SU2/).
  - Python scripts can now be called from any folder without moving the scripts.
- Get and extract configuration, mesh and solution files:
  wget su2.stanford.edu/documents/WorkshopFeb2017/WorkshopTutorial2naca.zip
  unzip WorkshopTutorial2naca.zip

  3D case: ...WorkshopFeb2017/WorkshopTutorial2oneram6.zip
- Move to the new directory:
  cd WorkshopTutorial2naca/
  - Similar to files needed for SU2_CFD analysis.
  - Configuration file now has optimization options that are read by the python script.
  - 3D case: mesh file now has free-form deformation box definition that is needed for some design variables.

Starting the Optimization Problem

$ shape_optimization.py   -f inv_NACA0012_adv.cfg -n 2 > opt.out &

Python script located in the SU2-5.0.0/bin/ folder

-f < file name > specifies the configuration file

-n <np> specifies the number of processors

> opt.out redirects the output,
& at the end 'backgrounds' the process

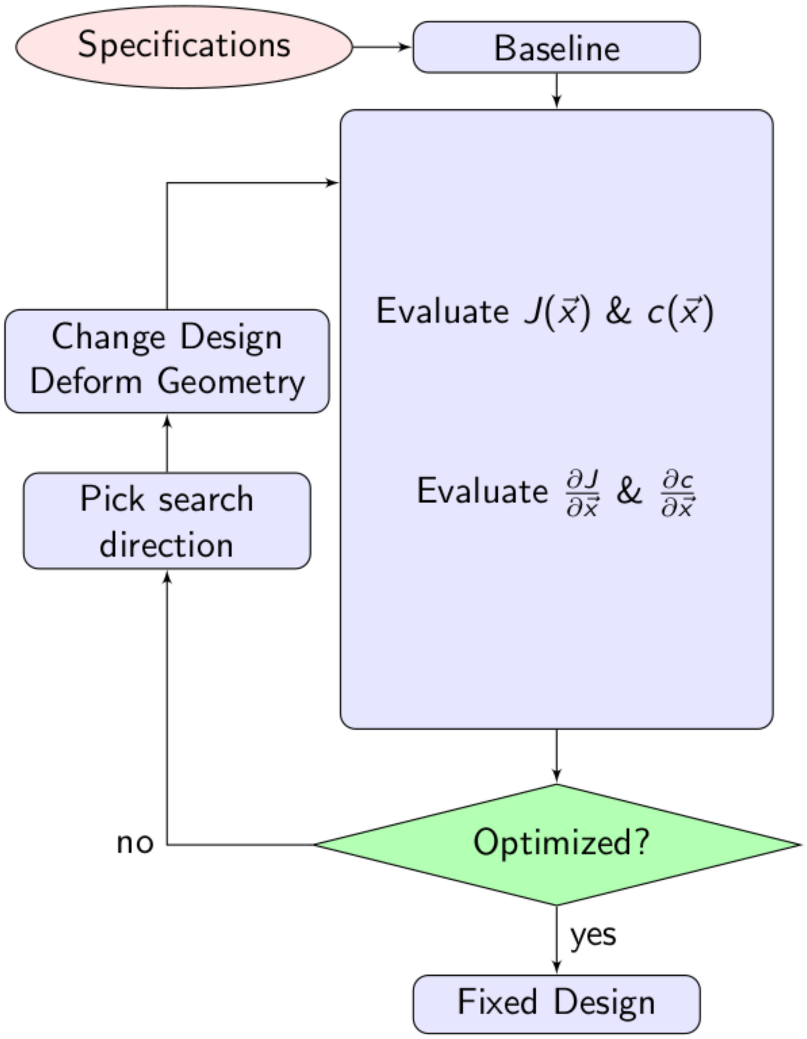
To verify the location of the script:
$ which shape_optimization.py

To check the number of available processors:
$ nproc

More about shape_optimization.py

- shape_optimization.py -h
- open shape_optimization.py in a text editor
- Other python scripts are available in SU2/bin/

Introduction to Optimization



Non-Linear Program:

$$\begin{array}{lll} \text{minimize} & J(\vec{x}) & \\ \text{with respect to} & \vec{x} \in \mathbb{R}^n & \\ \text{subject to} & \hat{c}_j(\vec{x}) = 0, & j = 1, \ldots, \hat{m} \\ & c_k(\vec{x}) \geq 0, & k = 1, \ldots, m \end{array}$$

$\vec{x}$: bump functions, FFD control points
J : an evaluation of SU2_CFD
c : an evalution of SU2_CFD or SU2_GEO

Optimization Algorithm: SciPy SLSQP
Gradient Techniques: continuous adjoint, finite difference, discrete adjoint.

Optimization Problem Settings & Workflow

Define and run the physical problem.

Evaluate geometry (thickness, AoA, etc).

Define design variables.

- Create the FFD box (.su2 file).
- FFD design variables preprocessing.

Define the optimization problem

- Objective function.
- Constraints (flow and geometry).
- Design variables.

Checks before the optimization (optional).

- Compute C_D, and C_L gradients.
- Compute geometric gradients.

Final checks (optional).

- Restart files are available.
- The grid contains the FFD information.
- The stop criteria is reasonable.
- The proposed optimization problem makes sense (scaling).

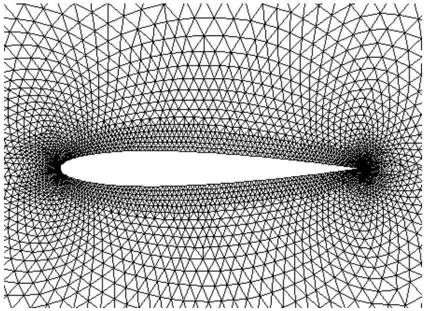
Run the optimization.

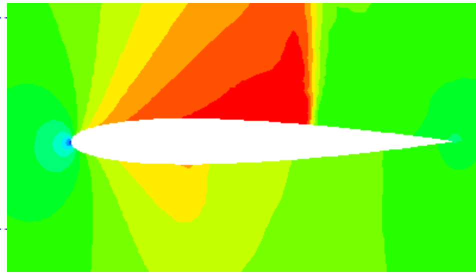
Analyze the solution.

- Folder structure and history_project file.
- Restart capability.

```
parallel_computation.py -f inv_NACA0012_adv.cfg -n 2
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                              %
% SU2 configuration file                                                       %
% Case description: Transonic inviscid optimization of a NACA0012 airfoil      %
% Author: Francisco Palacios                                                   %
% Institution: Stanford University                                             %
% Date: 2013.09.29                                                             %
% File Version 5.0.0 "Raven"                                                 %
%                                                                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ------------- DIRECT, ADJOINT, AND LINEARIZED PROBLEM DEFINITION ------------%
%
% Physical governing equations (EULER, NAVIER_STOKES,
%                               WAVE_EQUATION, HEAT_EQUATION, FEM_ELASTICITY,
%                               POISSON_EQUATION)
PHYSICAL_PROBLEM= EULER
%
% Mathematical problem (DIRECT, CONTINUOUS_ADJOINT)
MATH_PROBLEM= DIRECT
%
% Restart solution (NO, YES)
RESTART_SOL= YES

% -------------------- COMPRESSIBLE FREE-STREAM DEFINITION --------------------
%
% Mach number (non-dimensional, based on the free-stream values)
MACH_NUMBER= 0.8
%
% Angle of attack (degrees)
AOA= 1.25
%
% Free-stream pressure (101325.0 N/m^2 by default, only Euler flows)
FREESTREAM_PRESSURE= 101325.0
%
% Free-stream temperature (288.15 K by default)
FREESTREAM_TEMPERATURE= 288.15

% ---------------------- REFERENCE VALUE DEFINITION ---------------------------
%
% Reference origin for moment computation
REF_ORIGIN_MOMENT_X = 0.25
REF_ORIGIN_MOMENT_Y = 0.00
REF_ORIGIN_MOMENT_Z = 0.00
%
% Reference length for pitching, rolling, and yawing non-dimensional moment
REF_LENGTH_MOMENT= 1.0
%
% Reference area for force coefficients (0 implies automatic calculation)
REF_AREA= 1.0
%
% Flow non-dimensionalization (DIMENSIONAL, FREESTREAM_PRESS_EQ_ONE,
%                              FREESTREAM_VEL_EQ_MACH, FREESTREAM_VEL_EQ_ONE)
REF_DIMENSIONALIZATION= FREESTREAM_PRESS_EQ_ONE

% ----------------------- BOUNDARY CONDITION DEFINITION -----------------------%
%
% Marker of the Euler boundary (0 = no marker)
MARKER_EULER= ( airfoil )
%
% Marker of the far field (0 = no marker)
MARKER_FAR= ( farfield )

% ------------------------ SURFACES IDENTIFICATION ----------------------------%
%
% Marker of the surface which is going to be plotted or designed
MARKER_PLOTTING= ( airfoil )
%
% Marker of the surface where the functional (Cd, Cl, etc.) will be evaluated
MARKER_MONITORING= ( airfoil )
```





Optimization Problem Settings & Workflow

Define and run the physical problem.
Evaluate geometry (thickness, AoA, etc).
Define design variables.
- Create the FFD box (.su2 file).
- FFD design variables preprocessing.

Define the optimization problem
- Objective function.
- Constraints (flow and geometry).
- Design variables.

Checks before the optimization (optional).
- Compute C_D, and C_L gradients.
- Compute geometric gradients.

Final checks (optional).
- Restart files are available.
- The grid contains the FFD information.
- The stop criteria is reasonable.
- The proposed optimization problem makes sense (scaling).

Run the optimization.
Analyze the solution.
- Folder structure and history_project file.
- Restart capability.

2D:

```
SU2_GEO inv_naca0012_adv.cfg
```

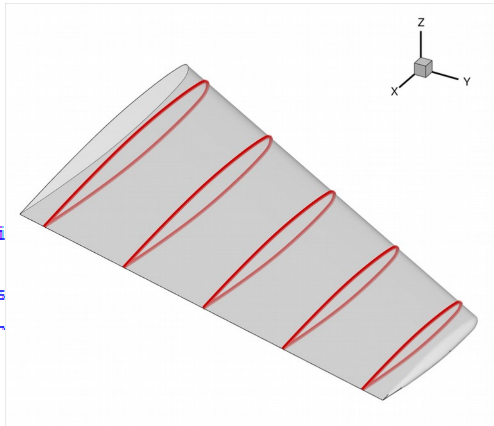
```
% ----------------------- GEOMETRY EVALUATION PARAMETERS ---------------------%
%
% Geometrical evaluation mode (FUNCTION, GRADIENT)
GEO_MODE= FUNCTION
%
% Marker(s) of the surface where geometrical based func. will be evaluated
GEO_MARKER= ( airfoil )
```

```
hlkline:WorkshopTutorial2v2$ more of_func.dat
TITLE = "SU2_GEO Evaluation"
VARIABLES = "MAX_THICKNESS","1/4_THICKNESS","1/3_THICKNESS","1/2_THICKNESS","2/3_THICKN
ZONE T= "Geometrical variables (value)"
0.120011, 0.118815, 0.119519, 0.105723, 0.0791089, 0.0624088, 0.0816925, 0, 1
```

3D:

```
SU2_GEO inv_ONERAM6.cfg
```

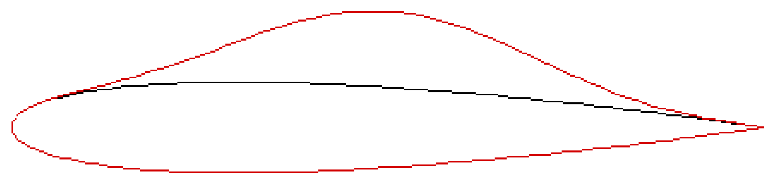
```
% ----------------------- GEOMETRY EVALUATION PARAMETERS ---------------------%
%
% Geometrical evaluation mode (FUNCTION, GRADIENT)
GEO_MODE= FUNCTION
%
% Marker(s) of the surface where geometrical based
GEO_MARKER= ( WING )
%
% Orientation of airfoil sections (X_AXIS, Y_AXIS,
GEO_AXIS_STATIONS= Y_AXIS
%
% Coordinate of the sections
GEO_LOCATION_STATIONS= (0.0, 0.2, 0.4, 0.6, 0.8)
%
% Location (coordinate) of the airfoil sections (Mi
GEO_WING_BOUNDS= (0.0806, 1.1284)
%
% Plot loads and Cp distributions on each airfoil s
GEO_PLOT_STATIONS= NO
```



Optimization Problem Settings & Workflow

Define and run the physical problem.
Evaluate geometry (thickness, AoA, etc).
Define design variables.
  Create the FFD box (.su2 file).
  FFD design variables preprocessing.
Define the optimization problem
  Objective function.
  Constraints (flow and geometry).
  Design variables .
Checks before the optimization (optional).
  Compute C_D, and C_L gradients.
  Compute geometric gradients.
Final checks (optional).
  Restart files are available.
  The grid contains the FFD information.
  The stop criteria is reasonable.
  The proposed optimization problem makes sense (scaling).
Run the optimization.
Analyze the solution.
  Folder structure and history_project file.
  Restart capability.

```
% ----------------------- DESIGN VARIABLE PARAMETERS ------------------------%
%
% Kind of deformation (FFD_SETTING, HICKS_HENNE, HICKS_HENNE_NORMAL, PARABOLIC,
%                      HICKS_HENNE_SHOCK, NACA_4DIGITS, DISPLACEMENT, ROTATION,
%                      FFD_CONTROL_POINT, FFD_DIHEDRAL_ANGLE, FFD_TWIST_ANGLE,
%                      FFD_ROTATION)
DV_KIND= HICKS_HENNE
%
% Marker of the surface in which we are going apply the shape deformation
DV_MARKER= ( airfoil )
%
% Parameters of the shape deformation
%       - HICKS_HENNE_FAMILY ( Lower(0)/Upper(1) side, x_Loc )
%       - NACA_4DIGITS ( 1st digit, 2nd digit, 3rd and 4th digit )
%       - PARABOLIC ( 1st digit, 2nd and 3rd digit )
%       - DISPLACEMENT ( x_Disp, y_Disp, z_Disp )
%       - ROTATION ( x_Orig, y_Orig, z_Orig, x_End, y_End, z_End )
DV_PARAM= ( 1, 0.5 )
%
% Value of the shape deformation deformation
DV_VALUE= 1.0
```



```
% Optimization design variables, separated by semicolons
DEFINITION_DV= ( 1, 1.0 | airfoil | 0, 0.05 ); ( 1, 1.0 | airfoil | 0, 0.10 ); ( 1, 1.0 |
airfoil | 0, 0.15 ); ( 1, 1.0 | airfoil | 0, 0.20 ); ( 1, 1.0 | airfoil | 0, 0.25 ); ( 1,
1.0 | airfoil | 0, 0.30 ); ( 1, 1.0 | airfoil | 0, 0.35 ); ( 1, 1.0 | airfoil | 0, 0.40 );
```

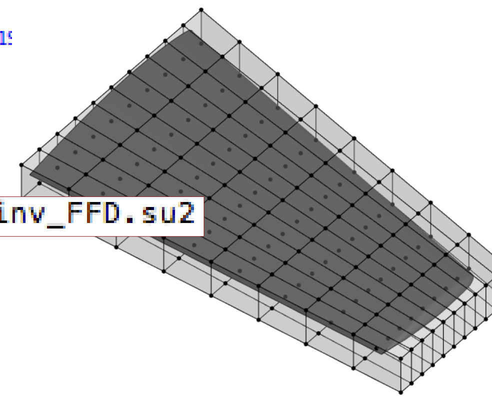
Optimization Problem Settings & Workflow

- Define and run the physical problem.
- Evaluate geometry (thickness, AoA, etc).
- Define design variables.
  - Create the FFD box (.su2 file).
  - FFD design variables preprocessing.
- Define the optimization problem
  - Objective function.
  - Constraints (flow and geometry).
  - Design variables based on FFD box.
- Checks before the optimization (optional).
  - Compute C_D, and C_L gradients.
  - Compute geometric gradients.
- Final checks (optional).
  - Restart files are available.
  - The grid contains the FFD information.
  - The stop criteria is reasonable.
  - The proposed optimization problem makes sense (scaling).
- Run the optimization.
- Analyze the solution.
  - Folder structure and history_project file.
  - Restart capability.

```
% -------------------- FREE-FORM DEFORMATION PARAMETERS ----------------------%
%
% Tolerance of the Free-Form Deformation point inversion
FFD_TOLERANCE= 1E-10
%
% Maximum number of iterations in the Free-Form Deformation point inversion
FFD_ITERATIONS= 500
%
% FFD box definition: 3D case (FFD_BoxTag, X1, Y1, Z1, X2, Y2, Z2, X3, Y3, Z3,
X4, Y4, Z4,
%                              X5, Y5, Z5, X6, Y6, Z6, X7, Y7, Z7, X8, Y8, Z8)
%                     2D case (FFD_BoxTag, X1, Y1, 0.0, X2, Y2, 0.0, X3, Y3,
0.0, X4, Y4, 0.0,
%                              0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0)
FFD_DEFINITION= (MAIN_BOX, -0.0403, 0.0, -0.04836, 0.8463, 0.0, -0.04836,
1.209, 1.2896, -0.04836, 0.6851, 1.2896, -0.04836, -0.0403, 0.0, 0.04836,
0.8463, 0.0, 0.04836, 1.209, 1.2896, 0.04836, 0.6851, 1.2896, 0.04836)
%
% FFD box degree: 3D case (x_degree, y_degree, z_degree)
%                 2D case (x_degree, y_degree, 0)
FFD_DEGREE= (10, 8, 1)
%
% Surface continuity at the intersection with the FFD (1S
2ND_DERIVATIVE)
FFD_CONTINUITY= 2ND_DERIVATIVE
```

```
SU2_DEF inv_ONERAM6.cfg
```

```
mv mesh_out.su2 mesh_ONERAM6_inv_FFD.su2
```



```
% Optimization design variables, separated by semicolons
DEFINITION_DV= ( 7, 1.0 | WING | MAIN_BOX, 0, 3, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING |
MAIN_BOX, 1, 3, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING | MAIN_BOX, 2, 3, 0, 0.0, 0.0, 1.0 );
( 7, 1.0 | WING | MAIN_BOX, 3, 3, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING | MAIN_BOX, 4, 3, 0,
0.0, 0.0, 1.0 ); ( 7, 1.0 | WING | MAIN_BOX, 5, 3, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING |
MAIN_BOX, 6, 3, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING | MAIN_BOX, 7, 3, 0, 0.0, 0.0, 1.0 );
( 7, 1.0 | WING | MAIN_BOX, 8, 3, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING | MAIN_BOX, 9, 3, 0,
0.0, 0.0, 1.0 ); ( 7, 1.0 | WING | MAIN_BOX, 10, 3, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING |
MAIN_BOX, 0, 4, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING | MAIN_BOX, 1, 4, 0, 0.0, 0.0, 1.0 );
( 7, 1.0 | WING | MAIN_BOX, 2, 4, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING | MAIN_BOX, 3, 4, 0,
0.0, 0.0, 1.0 ); ( 7, 1.0 | WING | MAIN_BOX, 4, 4, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING |
MAIN_BOX, 5, 4, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | WING | MAIN_BOX, 6, 4, 0, 0.0, 0.0, 1.0 );
```

Optimization Problem Settings & Workflow

Define and run the physical problem.
Evaluate geometry (thickness, AoA, etc).
Define design variables.
- Create the FFD box (.su2 file).
- FFD design variables preprocessing.

Define the optimization problem
- Objective function.
- Constraints (flow and geometry).
- Design variables.

Checks before the optimization (optional).
- Compute C_D, and C_L gradients.
- Compute geometric gradients.

Final checks (optional).
- Restart files are available.
- The grid contains the FFD information.
- The stop criteria is reasonable.
- The proposed optimization problem makes sense (scaling).

Run the optimization.
Analyze the solution.
- Folder structure and history_project file.
- Restart capability.

```
% -------------------- OPTIMAL SHAPE DESIGN DEFINITION ---------------------%
% Available flow based objective functions or constraint functions
%    DRAG, LIFT, SIDEFORCE, EFFICIENCY,
%    FORCE_X, FORCE_Y, FORCE_Z,
%    MOMENT_X, MOMENT_Y, MOMENT_Z,
%    THRUST, TORQUE, FIGURE_OF_MERIT,
%    EQUIVALENT_AREA, NEARFIELD_PRESSURE,
%    FREE_SURFACE
%
% Available geometrical based objective functions or constraint functions
%    MAX_THICKNESS, 1/4_THICKNESS, 1/2_THICKNESS, 3/4_THICKNESS, AREA, AOA, CHORD,
%    MAX_THICKNESS_SEC1, MAX_THICKNESS_SEC2, MAX_THICKNESS_SEC3, MAX_THICKNESS_SEC4, MAX_THICKNESS_SEC5,
%    1/4_THICKNESS_SEC1, 1/4_THICKNESS_SEC2, 1/4_THICKNESS_SEC3, 1/4_THICKNESS_SEC4, 1/4_THICKNESS_SEC5,
%    1/2_THICKNESS_SEC1, 1/2_THICKNESS_SEC2, 1/2_THICKNESS_SEC3, 1/2_THICKNESS_SEC4, 1/2_THICKNESS_SEC5,
%    3/4_THICKNESS_SEC1, 3/4_THICKNESS_SEC2, 3/4_THICKNESS_SEC3, 3/4_THICKNESS_SEC4, 3/4_THICKNESS_SEC5,
%    AREA_SEC1, AREA_SEC2, AREA_SEC3, AREA_SEC4, AREA_SEC5,
%    AOA_SEC1, AOA_SEC2, AOA_SEC3, AOA_SEC4, AOA_SEC5,
%    CHORD_SEC1, CHORD_SEC2, CHORD_SEC3, CHORD_SEC4, CHORD_SEC5
%
% Available design variables
%    HICKS_HENNE        (  1, Scale | Mark. List | Lower(0)/Upper(1) side, x_Loc )
%    SPHERICAL          (  3, Scale | Mark. List | ControlPoint_Index, Theta_Disp, R_Disp )
%    NACA_4DIGITS       (  4, Scale | Mark. List |  1st digit, 2nd digit, 3rd and 4th digit )
%    DISPLACEMENT       (  5, Scale | Mark. List | x_Disp, y_Disp, z_Disp )
%    ROTATION           (  6, Scale | Mark. List | x_Axis, y_Axis, z_Axis, x_Turn, y_Turn, z_Turn )
%    FFD_CONTROL_POINT  (  7, Scale | Mark. List | FFD_BoxTag, i_Ind, j_Ind, k_Ind, x_Mov, y_Mov, z_Mov )
%    FFD_TWIST  (  9, Scale | Mark. List | FFD_BoxTag, x_Orig, y_Orig, z_Orig, x_End, y_End, z_End )
%    FFD_ROTATION       ( 10, Scale | Mark. List | FFD_BoxTag, x_Orig, y_Orig, z_Orig, x_End, y_End, z_End )
%    FFD_CAMBER         ( 11, Scale | Mark. List | FFD_BoxTag, i_Ind, j_Ind )
%    FFD_THICKNESS      ( 12, Scale | Mark. List | FFD_BoxTag, i_Ind, j_Ind )
%    FFD_VOLUME         ( 13, Scale | Mark. List | FFD_BoxTag, i_Ind, j_Ind )
%    FOURIER            ( 14, Scale | Mark. List | Lower(0)/Upper(1) side, index, cos(0)/sin(1) )
%
% Optimization objective function with scaling factor
% ex= Objective * Scale
OPT_OBJECTIVE= DRAG * 0.001
%
% Optimization constraint functions with scaling factors, separated by semicolons
% ex= (Objective = Value ) * Scale, use '>','<','='
OPT_CONSTRAINT= ( LIFT > 0.327 ) * 0.001; ( MOMENT_Z > 0.0 ) * 0.001; ( MAX_THICKNESS > 0.12 ) * 0.001
%
% Maximum number of optimizer iterations
OPT_ITERATIONS= 100
%
% Requested accuracy
OPT_ACCURACY= 1E-6
%
% Upper bound for each design variable
OPT_BOUND_UPPER= 0.1
%
% Lower bound for each design variable
OPT_BOUND_LOWER= -0.1
%
% Optimization design variables, separated by semicolons
DEFINITION_DV= ( 1, 1.0 | airfoil | 0, 0.05 ); ( 1, 1.0 | airfoil | 0, 0.10 ); ( 1, 1.0 | airfoil | 0, 0.15 ); ( 1, 1.0 | airfoil | 0, 0.20 ); ( 1, 1.0 |
airfoil | 0, 0.25 ); ( 1, 1.0 | airfoil | 0, 0.30 ); ( 1, 1.0 | airfoil | 0, 0.35 ); ( 1, 1.0 | airfoil | 0, 0.40 ); ( 1, 1.0 | airfoil | 0, 0.45 ); ( 1,
1.0 | airfoil | 0, 0.50 ); ( 1, 1.0 | airfoil | 0, 0.55 ); ( 1, 1.0 | airfoil | 0, 0.60 ); ( 1, 1.0 | airfoil | 0, 0.65 ); ( 1, 1.0 | airfoil | 0, 0.70
); ( 1, 1.0 | airfoil | 0, 0.75 ); ( 1, 1.0 | airfoil | 0, 0.80 ); ( 1, 1.0 | airfoil | 0, 0.85 ); ( 1, 1.0 | airfoil | 0, 0.90 ); ( 1, 1.0 | airfoil |
0, 0.95 ); ( 1, 1.0 | airfoil | 1, 0.05 ); ( 1, 1.0 | airfoil | 1, 0.10 ); ( 1, 1.0 | airfoil | 1, 0.15 ); ( 1, 1.0 | airfoil | 1, 0.20 ); ( 1, 1.0 |
airfoil | 1, 0.25 ); ( 1, 1.0 | airfoil | 1, 0.30 ); ( 1, 1.0 | airfoil | 1, 0.35 ); ( 1, 1.0 | airfoil | 1, 0.40 ); ( 1, 1.0 | airfoil | 1, 0.45 ); ( 1,
1.0 | airfoil | 1, 0.50 ); ( 1, 1.0 | airfoil | 1, 0.55 ); ( 1, 1.0 | airfoil | 1, 0.60 ); ( 1, 1.0 | airfoil | 1, 0.65 ); ( 1, 1.0 | airfoil | 1, 0.70
); ( 1, 1.0 | airfoil | 1, 0.75 ); ( 1, 1.0 | airfoil | 1, 0.80 ); ( 1, 1.0 | airfoil | 1, 0.85 ); ( 1, 1.0 | airfoil | 1, 0.90 ); ( 1, 1.0 | airfoil |
1, 0.95 )
```

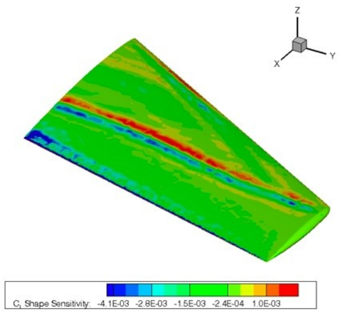
- Do not forget to check the name of the mesh.
- There are scripts to generate the list of design variables.

Optimization Problem Settings & Workflow

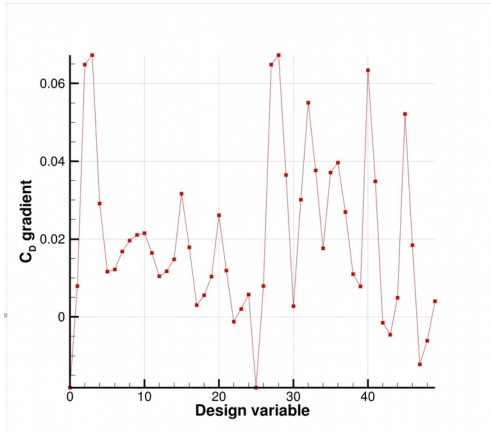
- Define and run the physical problem.
- Evaluate geometry (thickness, AoA, etc).
- Define design variables.
  - Create the FFD box (.su2 file).
  - FFD design variables preprocessing.
- Define the optimization problem
  - Objective function.
  - Constraints (flow and geometry).
  - Design variables.
- Checks before the optimization (optional).
  - Compute C_D, and C_L gradients.
  - Compute geometric gradients.
- Final checks (optional).
  - Restart files are available.
  - The grid contains the FFD information.
  - The stop criteria is reasonable.
  - The proposed optimization problem makes sense (scaling).
- Run the optimization.
- Analyze the solution.
  - Folder structure and history_project file.
  - Restart capability.

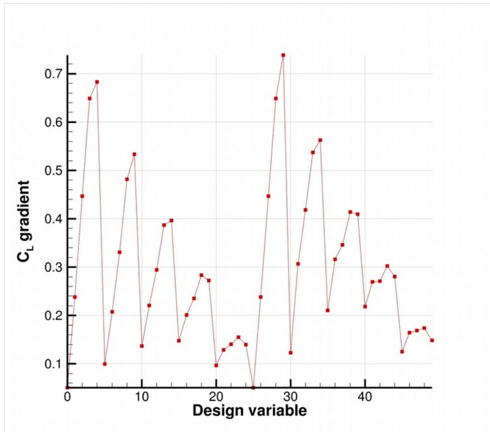
inv_ONERAM6.cfg

```
% -------------------- FLOW NUMERICAL METHOD DEFINITION ---------------------%
% Convective numerical method: (JST, LAX-FRIEDRICH, ROE-1ST_ORDER,
%                               ROE-2ND_ORDER)
CONV_NUM_METHOD_FLOW= JST
%
% Slope limiter: (VENKATAKRISHNAN)
SLOPE_LIMITER_FLOW= VENKATAKRISHNAN
%
% 1st, 2nd and 4th order artificial dissipation coefficients
AD_COEFF_FLOW= ( 0.15, 0.5, 0.04 )
%
% 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-1ST_ORDER,
%                               ROE-2ND_ORDER)
CONV_NUM_METHOD_ADJFLOW= JST
%
% Slope limiter: (VENKATAKRISHNAN, SHARP_EDGES)
SLOPE_LIMITER_ADJFLOW= VENKATAKRISHNAN
%
% 1st, 2nd, and 4th order artificial dissipation coefficients
AD_COEFF_ADJFLOW= ( 0.15, 0.5, 0.01 )
%
% Reduction factor of the CFL coefficient in the adjoint problem
CFL_REDUCTION_ADJFLOW= 0.25
%
% Time discretization (RUNGE-KUTTA_EXPLICIT, EULER_IMPLICIT)
TIME_DISCRE_ADJFLOW= EULER_IMPLICIT
```



```
continuous_adjoint.py -n 4 -f inv_ONERAM6.cfg
```





Optimization Problem Settings & Workflow

- Define and run the physical problem.
- Evaluate geometry (thickness, AoA, etc).
- Define design variables.
  - Create the FFD box (.su2 file).
  - FFD design variables preprocessing.
- Define the optimization problem
  - Objective function.
  - Constraints (flow and geometry).
  - Design variables.
- Checks before the optimization (optional).
  - Compute C_D, and C_L gradients.
  - Compute geometric gradients.
- **Final checks (optional).**
  - Restart files are available.
  - The grid contains the FFD information.
  - The stop criteria is reasonable.
  - The proposed optimization problem makes sense (scaling).
- Run the optimization.
- Analyze the solution.
  - Folder structure and history_project file.
  - Restart capability.

```
solution_adj_cd.dat
solution_adj_cl.dat
solution_flow.dat
```

```
% Restart solution (NO, YES)
RESTART_SOL= YES
```

mesh_ONERAM6_inv_FFD.su2

```
FFD_CORNER_POINTS= 8
-0.0403 0       -0.04836
0.8463  0       -0.04836
1.209   1.2896  -0.04836
0.6851  1.2896  -0.04836
-0.0403 0       0.04836
0.8463  0       0.04836
1.209   1.2896  0.04836
0.6851  1.2896  0.04836
FFD_CONTROL_POINTS= 60
0       0       0       -0.0403 0       -0.04836
0       0       1       -0.0403 0       0.04836
0       1       0       0.14105 0.3224  -0.04836
0       1       1       0.14105 0.3224  0.04836
0       2       0       0.3224  0.6448  -0.04836
0       2       1       0.3224  0.6448  0.04836
-------------------------------------------------
5       3       1       1.118325        0.9672  0.04836
5       4       0       1.209   1.2896  -0.04836
5       4       1       1.209   1.2896  0.04836
FFD_SURFACE_POINTS= 19893
LOWER_SIDE      13342   8.800257598459620e-02  5.495278239250181e-01  3.716563358902933e-01
LOWER_SIDE      13341   8.482803129010713e-02  5.525066256523129e-01  3.856879894932108e-01
LOWER_SIDE      12920   8.995567784976399e-02  5.521590113639829e-01  3.658012350400292e-01
LOWER_SIDE      7860    7.335924884774424e-01  1.065162196755406e-01  2.847658395767208e-01
```

inv_ONERAM6.cfg

inv_ONERAM6.cfg 〉 No Selection

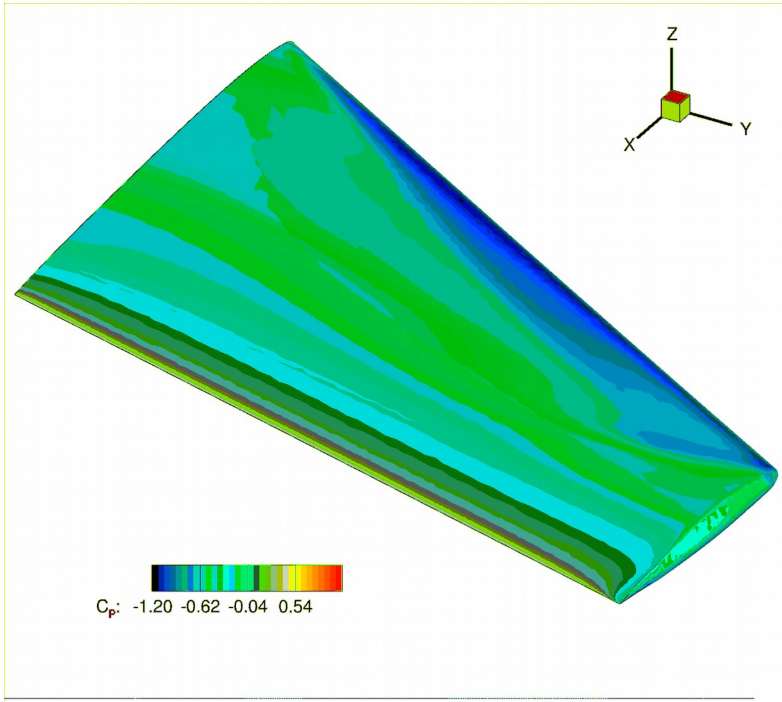
```
% ------------------------- INPUT/OUTPUT INFORMATION -------------------------%
% Mesh input file
MESH_FILENAME= mesh_ONERAM6_inv_FFD.su2
%
% Mesh output file
MESH_OUT_FILENAME= mesh_out.su2
%
% Restart flow input file
SOLUTION_FLOW_FILENAME= solution_flow.dat
%
% Restart adjoint input file
SOLUTION_ADJ_FILENAME= solution_adj.dat
%
% Mesh input file format (SU2)
MESH_FORMAT= SU2
%
% Output file format (PARAVIEW, TECPLOT)
OUTPUT_FORMAT= TECPLOT
```

```
% Optimization objective function with scaling factor
% ex= Objective * Scale
OPT_OBJECTIVE= DRAG * 0.1
%
% Optimization constraint functions with scaling factors, separated by semicolons
% ex= (Objective = Value ) * Scale, use '>','<','='
OPT_CONSTRAINT= (LIFT > 0.2864) * 0.1; (MAX_THICKNESS_SEC1 > 0.0570) * 0.1;
    (MAX_THICKNESS_SEC2 > 0.0513) * 0.1; (MAX_THICKNESS_SEC3 > 0.0457) * 0.1;
    (MAX_THICKNESS_SEC4 > 0.0399) * 0.1; (MAX_THICKNESS_SEC5 > 0.0343) * 0.1
```

Optimization Problem Settings & Workflow

- Define and run the physical problem.
- Evaluate geometry (thickness, AoA, etc).
- Define design variables.
    - Create the FFD box (.su2 file).
    - FFD design variables preprocessing.
- Define the optimization problem
    - Objective function.
    - Constraints (flow and geometry).
    - Design variables.
- Checks before the optimization (optional).
    - Compute C_D, and C_L gradients.
    - Compute geometric gradients.
- Final checks (optional).
    - Restart files are available.
    - The grid contains the FFD information.
    - The stop criteria is reasonable.
    - The proposed optimization problem makes sense (scaling).
- Run the optimization.
- Analyze the solution.
    - Folder structure and history_project file.
    - Restart capability.

```
shape_optimization.py -n 4 -f inv_ONERAM6.cfg
```



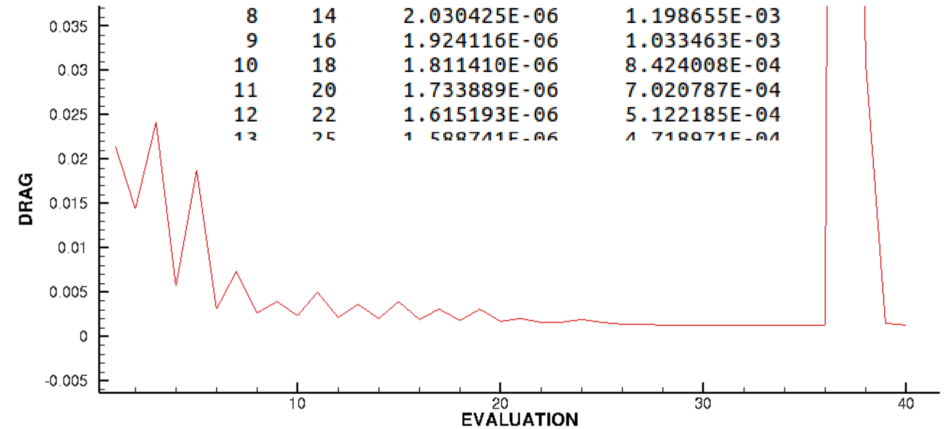
aerospacedesignlab

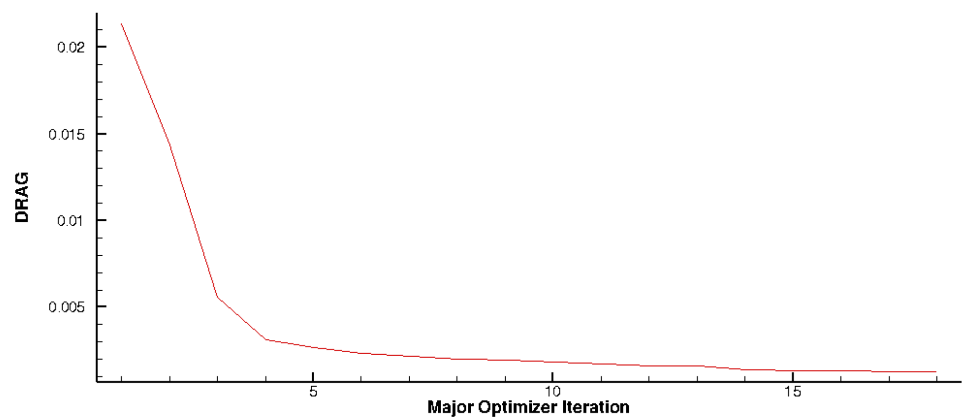
Stanford University

Optimization Problem Settings & Workflow

- Define and run the physical problem.
- Evaluate geometry (thickness, AoA, etc).
- Define design variables.
  - Create the FFD box (.su2 file).
  - FFD design variables preprocessing.
- Define the optimization problem
  - Objective function.
  - Constraints (flow and geometry).
  - Design variables.
- Checks before the optimization (optional).
  - Compute C_D, and C_L gradients.
  - Compute geometric gradients.
- Final checks (optional).
  - Restart files are available.
  - The grid contains the FFD information.
  - The stop criteria is reasonable.
  - The proposed optimization problem makes sense (scaling).
- Run the optimization.
- Analyze the solution.
  - Folder structure and history_project file.
  - Restart capability.

NIT	FC	OBJFUN	GNORM
1	1	2.134974E-05	3.829535E-03
2	2	1.435798E-05	3.560847E-03
3	4	5.605162E-06	2.564817E-03
4	6	3.106681E-06	2.047571E-03
5	8	2.682310E-06	1.835039E-03
6	10	2.314541E-06	1.541461E-03
7	12	2.179186E-06	1.392628E-03
8	14	2.030425E-06	1.198655E-03
9	16	1.924116E-06	1.033463E-03
10	18	1.811410E-06	8.424008E-04
11	20	1.733889E-06	7.020787E-04
12	22	1.615193E-06	5.122185E-04
13	25	1.588741E-06	4.718971E-04





Explore the Optimization Results

- Track log files
  - opt.out
  - DESIGNS/DSN_001/DIRECT/log_Direct.out
- Visualize optimization history
  - history_project.dat
- Visualize geometry
- Stop, restart optimization
- $ % (this ‘foregrounds’ the ‘backgrounded’ job)
- $ Ctrl-C (this stops the job)
- $ shape_optimization.py -f <...>.cfg -n <np> -r project.pkl

Questions?