aerospacedesignlab

Getting Started With SU^2

Code Installation, RANS Analysis, and Shape Design

July 1st, 2014

Francisco Palacios, Michael Colonno,
Thomas D. Economon, Juan J. Alonso
Department of Aeronautics & Astronautics
Stanford University

GETTING STARTED

BUILDING SU²

ANALYSIS EXAMPLE: ONERA M6 WING

DESIGN EXAMPLE: ONERA M6 WING

The Open-Source CFD Code

GETTING STARTED

BUILDING SU²

ANALYSIS EXAMPLE:
ONERA M6 WING

DESIGN EXAMPLE:
ONERA M6 WING

The Open-Source CFD Code

Getting Started
Downloading the Code

- Check out the download portal on the main website: http://su2.stanford.edu/download.html.
- If that site is blocked, download the V3.2 version (zip) directly from GitHub here: https://github.com/su2code/SU2/releases.
- Register and download binaries or source for major releases there.
- Developer versions of the code are also available on GitHub: http://github.com/su2code.html.
- Suite of test cases also available (includes tutorial files).
- SU2 V3.2 was recently released at AIAA AVIATION on June 17, 2014

Recommended: Register and download the source code and test cases for SU2 V3.2 or get the latest and greatest from GitHub

Getting Started
Documentation and Tutorials

- A large body of documentation is available!

- Main documentation found under “Guides” linked on the main page:
  - http://adl-public.stanford.edu/docs/display/SUSQUARED/SU2+Home
  - Detailed information on installation, input and output files, etc.
  - Contains step-by-step tutorials
  - Wiki-style docs that the developers maintain
  - See current contents to the right…

- About
- Download
- Installation
- Quick Start Tutorial
- User's Guide
  - Running SU2
  - Input and Output Files
    - Configuration file
    - Mesh files
    - Solution and Restart files
    - Non-dimensionalization
- User's Tutorials
  - Tutorial 1 - Bump in a Channel
  - Tutorial 2 - Inviscid ONERA M6
  - Tutorial 3 - Laminar Flat Plate
  - Tutorial 4 - Laminar Cylinder
  - Tutorial 5 - Turbulent Flat Plate
  - Tutorial 6 - Turbulent ONERA M6
  - Tutorial 7 - Optimal Shape Design of a Rotating Airfoil
  - Tutorial 8 - Constrained Optimal Shape Design of a Fixed Wing
  - Tutorial 9 - Inviscid Supersonic Wedge
- Developer's Guide
- FAQs
- References & Citations
- Contact

Getting Started
Documentation and Tutorials

- Additional training materials: http://su2.stanford.edu/training.html.
  - Links to tutorials, presentations, files, and videos

- Active forum on CFD Online: http://www.cfd-online.com/Forums/su2/.

- Recent AIAA publications with some technical detail on physics, numerical methods, and V&V:
  - *Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design,* AIAA Paper 2013-0287
  - *Stanford University Unstructured (SU2): Open-source analysis and design technology for turbulent flows,* AIAA Paper 2014-0243

Tip: If you get stuck or have questions, just drop us a line at our personal emails or susquared-dev@lists.stanford.edu

GETTING STARTED

BUILDING SU^2

ANALYSIS EXAMPLE:
ONERA M6 WING

DESIGN EXAMPLE:
ONERA M6 WING

The Open-Source CFD Code

Building SU^2
Our Open-Source Build Philosophy

- As an open-source code, care has been taken to **simplify the build process** wherever possible:
  - Use standard C++
  - Leverage standard build tools, i.e., autoconf and automake
  - Avoid or automatically include external dependencies
- To build the vanilla version, all you should need is a C++ compiler.
- However, we maintain the **flexibility** to build with additional features:
  - Fully parallel with MPI, specify your MPI flavor with configure
  - METIS ships with SU2 and is automatically built for parallel runs
  - Tecio for binary Tecplot output also ships as an optional feature
  - CGNS mesh input capability (unstructured, single block)

Building SU^2

Customizing and Compiling from Source

1. Download and untar the code, then move into the source directory.

   ```
   $ tar -xzf SU2v3.2.0.tgz
   $ cd SU2
   ```

2. Specify various options, compilers, etc. and run configure. The following options are typical for high-performance/parallel operation.

   ```
   $ ./configure --with-MPI=mpicxx CXXFLAGS='-O3' --prefix=/home/economon/SU2 --with-CGNS-lib=/home/economon/cgns-3.1.4/lib --with-CGNS-include=/home/economon/cgns-3.1.4/include
   ```

3. Make and install the code (location specified by the prefix option above). Note that this could be done in two separate steps, as "make" and "make install" consecutively at the command line.

   ```
   $ make -j 8 install
   ```

4. Update your PATH variables appropriately before executing.

   ```
   $ export SU2_RUN="/home/economon/SU2/bin"
   $ export SU2_HOME="/home/economon/SU2"
   $ export PATH=$SU2_RUN:$PATH
   $ export PYTHONPATH=$SU2_RUN:$PYTHONPATH
   ```

Building SU^2
Tips for Hassle-Free Builds

Recommended: CGNS V3.1.4 without HDF5

Tip: Use parallel make to accelerate the build, i.e., “$ make -j 24 install” to use 24 processes if you have a workstation with 24 cores.

Recommended: For advanced features in SU2, Python scripts are available. Try out the free Python distribution (V2.X distribution) with built-in packages by Enthought: https://www.enthought.com/products/epd/free/

Building SU^2

Tips for Hassle-Free Builds

Tip: Add an alias for your configure command and set the environment variables in your ~/.bashrc file (steps 2 and 4 on the previous slide).

$ alias config_serial="./configure CXXFLAGS='-O3' --prefix=/Users/economon/SU2 --enable-tecio"

$ alias config_mpi="./configure --with-MPI=mpicxx --prefix=/Users/economon/SU2 --enable-tecio CXXFLAGS='-O3' --with-CGNS-lib=/usr/local/lib --with-CGNS-include=/usr/local/include"

Building SU²

Tips for Hassle-Free Builds

```
$ export SU2_RUN="/home/economon/SU2/bin”
$ export SU2_HOME="/home/economon/SU2”
$ export PATH=$SU2_RUN:$PATH
$ export PYTHONPATH=$SU2_RUN:$PYTHONPATH
```

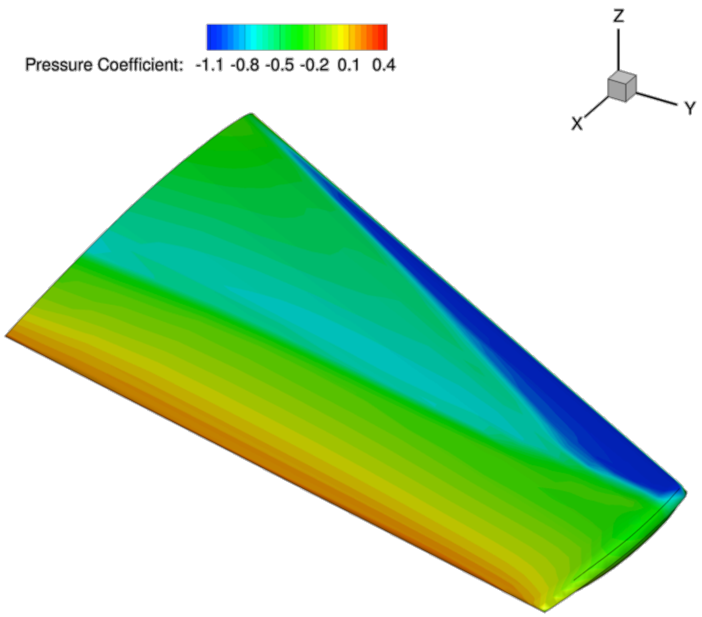
Tip: Note the order of the PATH variable. $SU2_RUN appears before $PATH in this case. If $PATH comes before $SU2_RUN, make sure that there are not any previous versions of SU2 installed in other locations in your PATH, otherwise it may use old versions and cause headaches…

GETTING STARTED

BUILDING SU^2

ANALYSIS EXAMPLE: ONERA M6 WING

DESIGN EXAMPLE: ONERA M6 WING

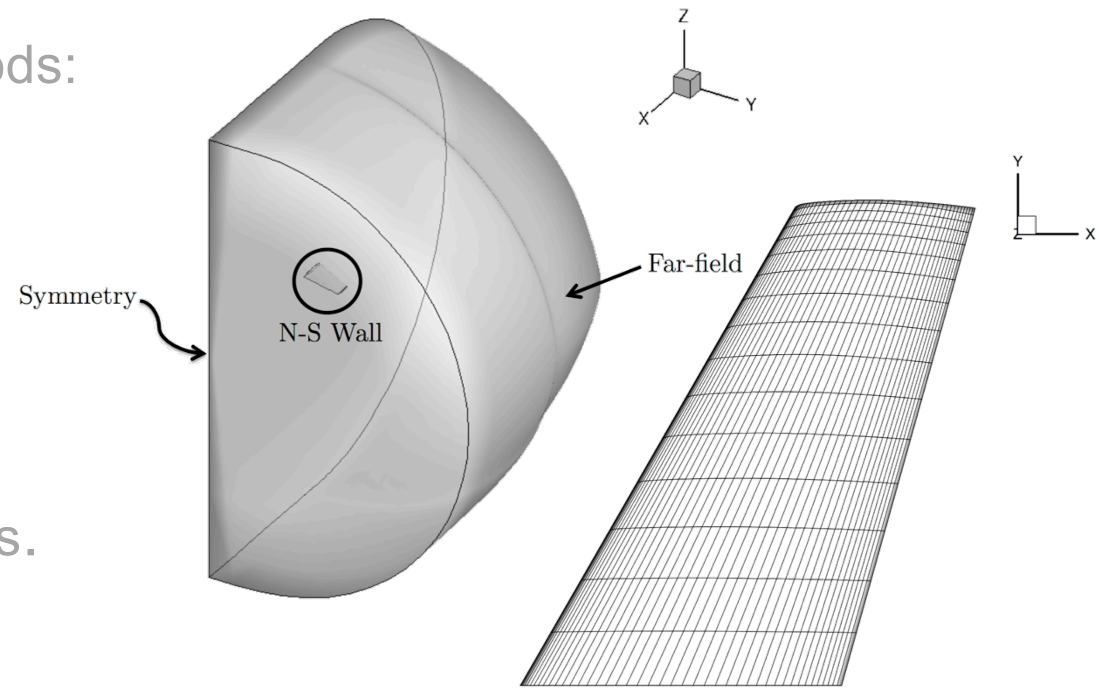


ONERA M6 RANS Analysis

1. Prepare geometry & mesh beforehand.
2. Choose appropriate physics.
3. Set proper conditions for a viscous simulation.
4. Select numerical methods:
   A. Convective terms
   B. Viscous terms
   C. Time Integration
   D. Multi-grid
5. Run the analysis.
6. Post-process the results.

```
% Mesh input file
MESH_FILENAME= mesh_ONERAM6_turb_hexa_43008.su2
```

```
---------------------- Read grid file information ----------------------
Three dimensional problem.
43008 interior elements. 46417 points, and 0 ghost points.
3 surface markers.
2560 boundary elements in index 0 (Marker = FARFIELD).
1408 boundary elements in index 1 (Marker = WING).
2688 boundary elements in index 2 (Marker = SYMMETRY).
```



ONERA M6 RANS Analysis

1. Prepare geometry & mesh beforehand.
2. Choose appropriate physics.
3. Set proper conditions for a viscous simulation.
4. Select numerical methods:
   A. Convective terms
   B. Viscous terms
   C. Time Integration
   D. Multi-grid
5. Run the analysis.
6. Post-process the results.

```
% ------------- DIRECT, ADJOINT, AND LINEARIZED PROBLEM DEFINITION ------------%
%
% Physical governing equations (EULER, NAVIER_STOKES,
%                               TNE2_EULER, TNE2_NAVIER_STOKES,
%                               WAVE_EQUATION, HEAT_EQUATION, LINEAR_ELASTICITY,
%                               POISSON_EQUATION)
PHYSICAL_PROBLEM= NAVIER_STOKES
%
% Specify turbulence model (NONE, SA, SST)
KIND_TURB_MODEL= SA
%
% Mathematical problem (DIRECT, ADJOINT, LINEARIZED)
MATH_PROBLEM= DIRECT
```

ONERA M6 RANS Analysis

1. Prepare geometry & mesh beforehand.
2. Choose appropriate physics.
3. Set proper conditions for a viscous simulation.
4. Select numerical methods:
   A. Convective terms
   B. Viscous terms
   C. Time Integration
   D. Multi-grid
5. Run the analysis.
6. Post-process the results.

How are the flow conditions set inside SU2?

a) Store the gas constants and freestream temperature, then calculate the speed of sound.
b) Calculate and store the freestream velocity from the Mach number & AoA/sideslip angles.
c) Compute the freestream viscosity from Sutherland's law and the supplied freestream temperature.
d) Use the definition of the Reynolds number to find the freestream density from the supplied Reynolds information, freestream velocity, and freestream viscosity from step 3.
e) Calculate the freestream pressure using the perfect gas law with the freestream temperature, specific gas constant, and freestream density from step 4.
f) Perform any required non-dim.

ONERA M6 RANS Analysis

Tip: Meshes should be in units of meters for viscous flows. Try the MESH_SCALE_CHANGE option in the configuration file.

```
% Change the scale of the numerical grid (useful to change the length units
%                                         or to re-scale the grid)
MESH_SCALE_CHANGE= 1.0
%
% Write a new mesh after reading only in serial (NO, YES)
MESH_OUTPUT= NO
%
% Mesh output file
MESH_OUT_FILENAME= mesh_out.su2
```

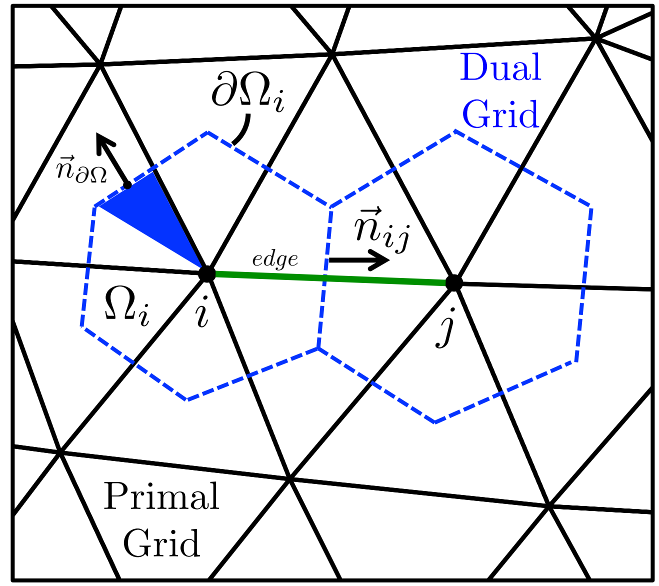
ONERA M6 RANS Analysis

1. Prepare geometry & mesh beforehand.
2. Choose appropriate physics.
3. Set proper conditions for a viscous simulation.
4. Select numerical methods:
   A. Convective terms
   B. Viscous terms
   C. Time Integration
   D. Multi-grid
5. Run the analysis.
6. Post-process the results.

```
% ------------------- FLOW NUMERICAL METHOD DEFINITION ----------------------%
%
% Convective numerical method (JST, LAX-FRIEDRICH, CUSP, ROE, AUSM, HLLC,
%                              TURKEL_PREC, MSW)
CONV_NUM_METHOD_FLOW= ROE
%
% Spatial numerical order integration (1ST_ORDER, 2ND_ORDER, 2ND_ORDER_LIMITER)
SPATIAL_ORDER_FLOW= 2ND_ORDER_LIMITER
%
% Slope limiter (VENKATAKRISHNAN, MINMOD)
SLOPE_LIMITER_FLOW= VENKATAKRISHNAN
%
% Coefficient for the limiter (smooth regions)
LIMITER_COEFF= 10.0
%
% 1st, 2nd and 4th order artificial dissipation coefficients
AD_COEFF_FLOW= ( 0.15, 0.5, 0.02 )
%
% Viscous numerical method (AVG_GRAD, AVG_GRAD_CORRECTED, GALERKIN)
VISC_NUM_METHOD_FLOW= AVG_GRAD_CORRECTED
%
% Source term numerical method (PIECEWISE_CONSTANT)
SOUR_NUM_METHOD_FLOW= PIECEWISE_CONSTANT
%
% Time discretization (RUNGE-KUTTA_EXPLICIT, EULER_IMPLICIT, EULER_EXPLICIT)
TIME_DISCRE_FLOW= EULER_IMPLICIT

% ------------------- TURBULENT NUMERICAL METHOD DEFINITION -----------------%
%
% Convective numerical method (SCALAR_UPWIND)
CONV_NUM_METHOD_TURB= SCALAR_UPWIND
%
% Spatial numerical order integration (1ST_ORDER, 2ND_ORDER, 2ND_ORDER_LIMITER)
SPATIAL_ORDER_TURB= 1ST_ORDER
%
% Slope limiter (VENKATAKRISHNAN, MINMOD)
SLOPE_LIMITER_TURB= VENKATAKRISHNAN
%
% Viscous numerical method (AVG_GRAD, AVG_GRAD_CORRECTED)
VISC_NUM_METHOD_TURB= AVG_GRAD_CORRECTED
%
% Source term numerical method (PIECEWISE_CONSTANT)
SOUR_NUM_METHOD_TURB= PIECEWISE_CONSTANT
%
% Time discretization (EULER_IMPLICIT)
TIME_DISCRE_TURB= EULER_IMPLICIT
```

ONERA M6 RANS Analysis

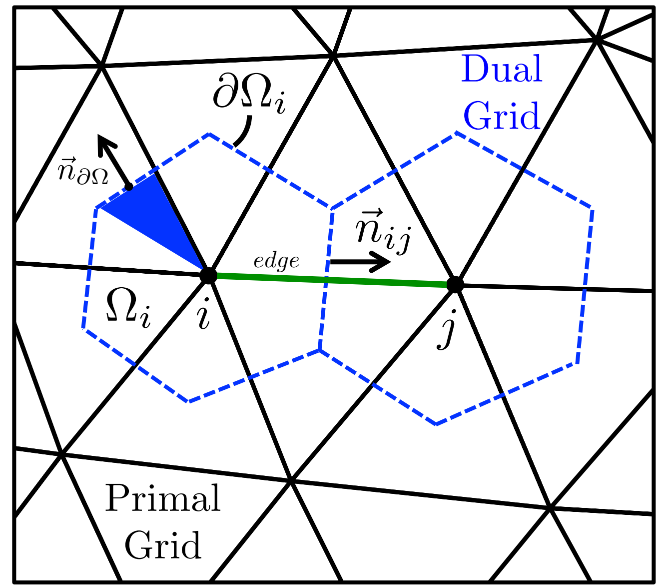
1. Prepare geometry & mesh beforehand.
2. Choose appropriate physics.
3. Set proper conditions for a viscous simulation.
4. Select numerical methods:
   - A. Convective terms
   - B. Viscous terms
   - C. Time Integration
   - D. Multi-grid
5. Run the analysis.
6. Post-process the results.




```
% Convective numerical method (JST, LAX-FRIEDRICH, CUSP, ROE, AUSM, HLLC,
%                              TURKEL_PREC, MSW)
CONV_NUM_METHOD_FLOW= ROE
%
% Spatial numerical order integration (1ST_ORDER, 2ND_ORDER, 2ND_ORDER_LIMITER)
SPATIAL_ORDER_FLOW= 2ND_ORDER_LIMITER
%
% Slope limiter (VENKATAKRISHNAN, MINMOD)
SLOPE_LIMITER_FLOW= VENKATAKRISHNAN
%
% Coefficient for the limiter (smooth regions)
LIMITER_COEFF= 10.0

% Convective numerical method (SCALAR_UPWIND)
CONV_NUM_METHOD_TURB= SCALAR_UPWIND
%
% Spatial numerical order integration (1ST_ORDER, 2ND_ORDER, 2ND_ORDER_LIMITER)
SPATIAL_ORDER_TURB= 1ST_ORDER
```

ONERA M6 RANS Analysis

1. Prepare geometry & mesh beforehand.
2. Choose appropriate physics.
3. Set proper conditions for a viscous simulation.
4. Select numerical methods:
   - A. Convective terms
   - B. Viscous terms
   - C. Time Integration
   - D. Multi-grid
5. Run the analysis.
6. Post-process the results.




```
% Numerical method for spatial gradients (GREEN_GAUSS, WEIGHTED_LEAST_SQUARES)
NUM_METHOD_GRAD= GREEN_GAUSS

% Viscous numerical method (AVG_GRAD, AVG_GRAD_CORRECTED, GALERKIN)
VISC_NUM_METHOD_FLOW= AVG_GRAD_CORRECTED
% Viscous numerical method (AVG_GRAD, AVG_GRAD_CORRECTED)
VISC_NUM_METHOD_TURB= AVG_GRAD_CORRECTED
```

ONERA M6 RANS Analysis

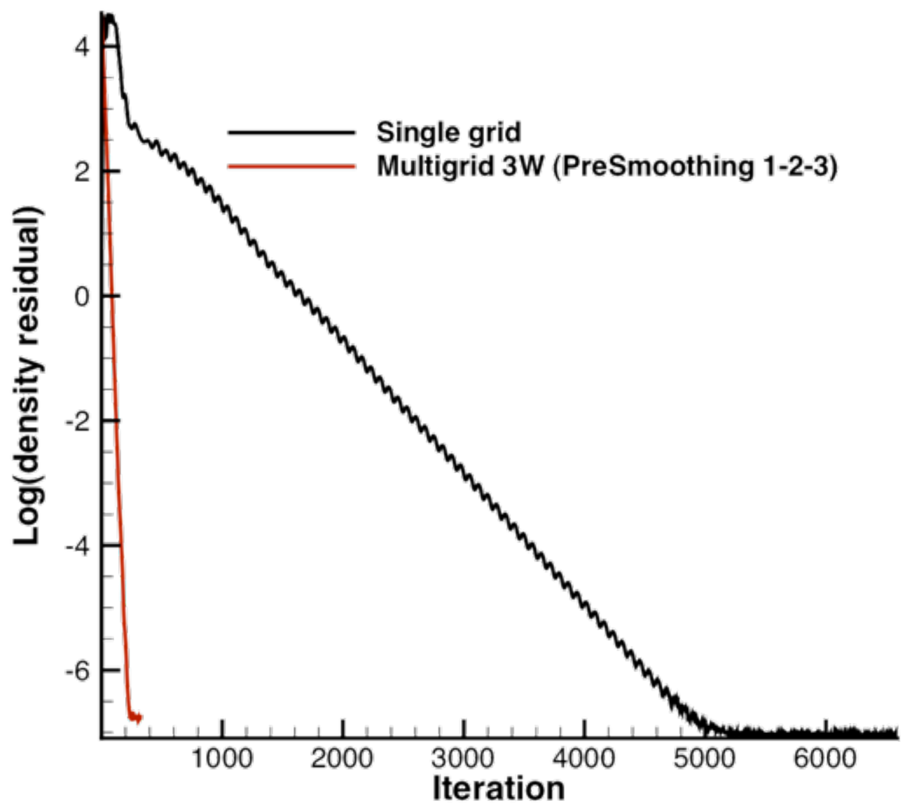
1. Prepare geometry & mesh beforehand.
2. Choose appropriate physics.
3. Set proper conditions for a viscous simulation.
4. Select numerical methods:
   A. Convective terms
   B. Viscous terms
   C. Time Integration
   D. Multi-grid
5. Run the analysis.
6. Post-process the results.

```
% Time discretization (RUNGE-KUTTA_EXPLICIT,
                       EULER_IMPLICIT, EULER_EXPLICIT)
TIME_DISCRE_FLOW= EULER_IMPLICIT

% Time discretization (EULER_IMPLICIT)
TIME_DISCRE_TURB= EULER_IMPLICIT
```

```
% ------------------------ LINEAR SOLVER DEFINITION --------------------------%
%
% Linear solver for the implicit (or discrete adjoint) formulation (BCGSTAB, FGMRES)
LINEAR_SOLVER= FGMRES
%
% Preconditioner of the Krylov linear solver (NONE, JACOBI, LINELET)
LINEAR_SOLVER_PREC= LU_SGS
%
% Min error of the linear solver for the implicit formulation
LINEAR_SOLVER_ERROR= 1E-4
%
% Max number of iterations of the linear solver for the implicit formulation
LINEAR_SOLVER_ITER= 5
```

ONERA M6 RANS Analysis

1. Prepare geometry & mesh beforehand.
2. Choose appropriate physics.
3. Set proper conditions for a viscous simulation.
4. Select numerical methods:
   A. Convective terms
   B. Viscous terms
   C. Time Integration
   D. Multi-grid
5. Run the analysis.
6. Post-process the results.



Multigrid example with the inviscid ONERA M6

ONERA M6 RANS Analysis

```
% ------------------------- MULTIGRID PARAMETERS ----------------------------%
%
% Multi-Grid Levels (0 = no multi-grid)
MGLEVEL= 3
%
% Multi-Grid Cycle (0 = V cycle, 1 = W Cycle)
MGCYCLE= 1
%
% Reduction factor of the CFL coefficient in the coarse levels
MG_CFL_REDUCTION= 0.9
%
% Maximum number of children in the agglomeration stage
MAX_CHILDREN= 250
%
% Maximum length of an agglomerated element (compared with the domain)
MAX_DIMENSION= 0.15
%
% Multi-Grid PreSmoothing Level
MG_PRE_SMOOTH= ( 1, 2, 3, 3 )
%
% Multi-Grid PostSmoothing Level
MG_POST_SMOOTH= ( 0, 0, 0, 0 )
%
% Jacobi implicit smoothing of the correction
MG_CORRECTION_SMOOTH= ( 0, 0, 0, 0 )
%
% Damping factor for the residual restriction
MG_DAMP_RESTRICTION= 0.9
%
% Damping factor for the correction prolongation
MG_DAMP_PROLONGATION= 0.9
```

Tip: There are many knobs available for tuning the agglomeration multigrid in the configuration file.

ONERA M6 RANS Analysis

1. Prepare geometry & mesh beforehand.
2. Choose appropriate physics.
3. Set proper conditions for a viscous simulation.
4. Select numerical methods:
   - A. Convective terms
   - B. Viscous terms
   - C. Time Integration
   - D. Multi-grid
5. Run the analysis.
6. Post-process the results.

The files for this case can be found in TestCases/rans/oneram6/. Run the case with:

$ SU2_CFD turb_ONERAM6.cfg

```
----------------------------- Begin Solver -----------------------------

Maximum residual: -0.0721104, located at point 44935.

Iter    Time(s)     Res[Rho]      Res[nu]   CLift(Total)   CDrag(Total)
   0   2.030245    -1.922382    -6.595337       0.235584       0.138501
   1   2.028815    -1.938899    -6.745390       0.273893       0.168589
   2   2.021844    -2.103530    -6.731876       0.264675       0.174642
   3   2.025663    -2.258632    -6.642393       0.248635       0.172972
   4   2.026968    -2.374254    -6.597767       0.237385       0.170551
   5   2.026089    -2.412192    -6.579448       0.231731       0.168415
   6   2.025900    -2.415528    -6.575112       0.229884       0.166201
   7   2.024740    -2.406839    -6.577105       0.229701       0.163874
   8   2.027044    -2.391325    -6.580680       0.229969       0.161399
   9   2.025022    -2.369791    -6.583364       0.230239       0.158709
  10   2.025010    -2.343659    -6.584583       0.230390       0.155769
  11   2.025580    -2.321644    -6.584342       0.230408       0.152550
  12   2.029821    -2.300447    -6.582755       0.230313       0.149018
  13   2.033597    -2.276003    -6.580333       0.230019       0.145140
  14   2.038047    -2.259261    -6.577723       0.229501       0.140885
  15   2.042814    -2.241444    -6.574115       0.228783       0.136243
  16   2.047098    -2.225648    -6.571125       0.227888       0.131226
  17   2.047228    -2.213251    -6.567373       0.226836       0.125867
  18   2.050010    -2.205186    -6.564792       0.225628       0.120227
  19   2.052010    -2.191509    -6.560882       0.224339       0.114373
```

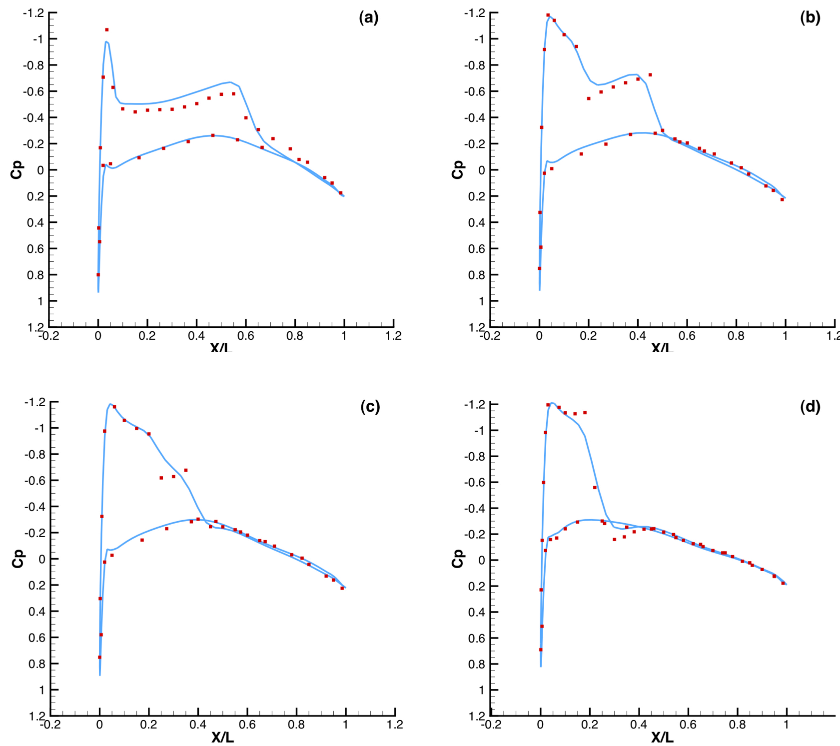
ONERA M6 RANS Analysis

Tip: Convergence can be controlled based on the residual (RESIDUAL) or by monitoring coefficients (CAUCHY).

```
% --------------------------- CONVERGENCE PARAMETERS --------------------------%
%
% Convergence criteria (CAUCHY, RESIDUAL)
%
CONV_CRITERIA= CAUCHY
%
% Residual reduction (order of magnitude with respect to the initial value)
RESIDUAL_REDUCTION= 5
%
% Min value of the residual (log10 of the residual)
RESIDUAL_MINVAL= -10
%
% Start convergence criteria at iteration number
STARTCONV_ITER= 10
%
% Number of elements to apply the criteria
CAUCHY_ELEMS= 100
%
% Epsilon to control the series convergence
CAUCHY_EPS= 1E-6
%
% Function to apply the criteria (LIFT, DRAG, NEARFIELD_PRESS, SENS_GEOMETRY,
%                                 SENS_MACH, DELTA_LIFT, DELTA_DRAG)
CAUCHY_FUNC_FLOW= DRAG
CAUCHY_FUNC_ADJFLOW= SENS_GEOMETRY
```

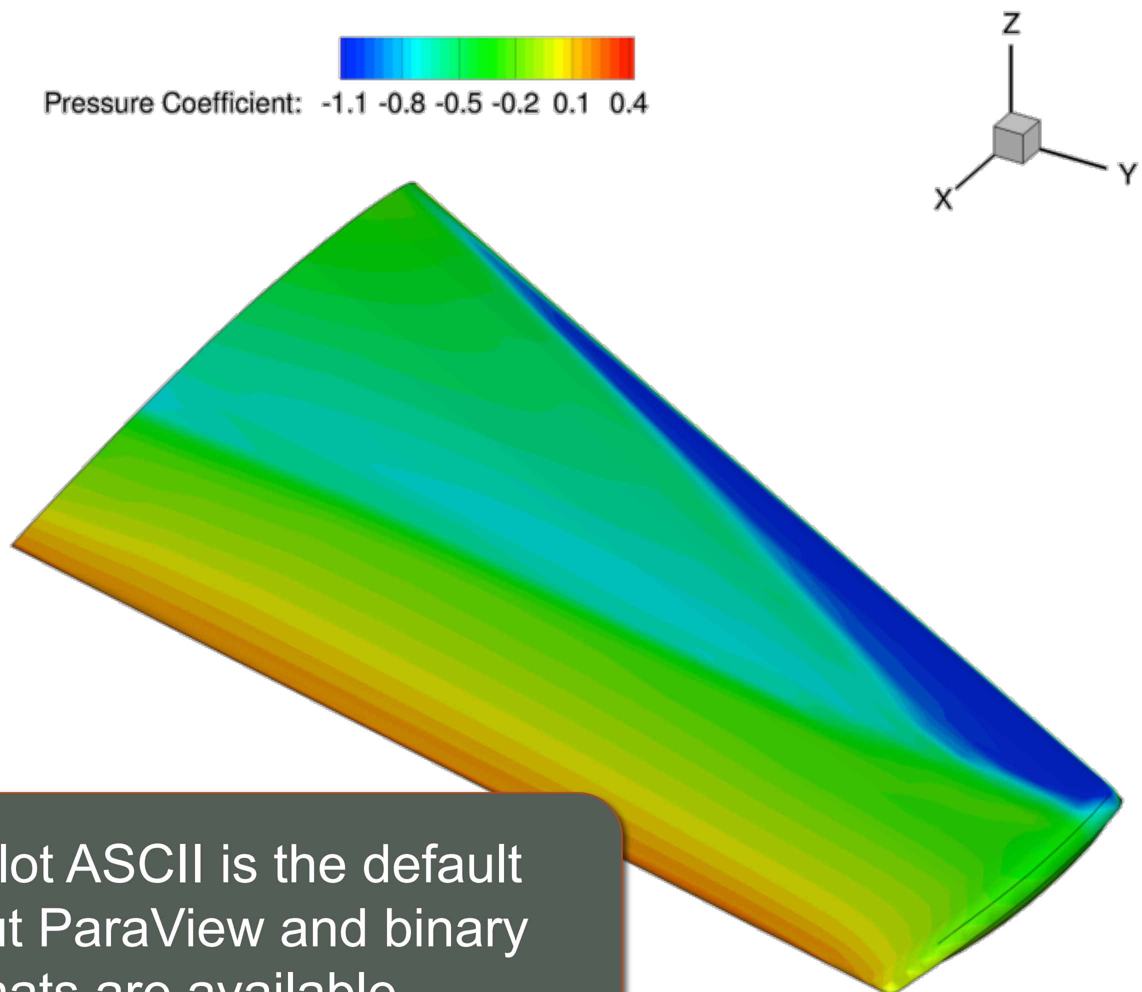
ONERA M6 RANS Analysis

1. Prepare geometry & mesh beforehand.
2. Choose appropriate physics.
3. Set proper conditions for a viscous simulation.
4. Select numerical methods:
   A. Convective terms
   B. Viscous terms
   C. Time Integration
   D. Multi-grid
5. Run the analysis.
6. **Post-process the results.**



Comparison of Cp profiles of the experimental results of Schmitt and Carpin (red squares) against SU2 computational results (blue line) at different sections along the span of the wing. (a) y/b = 0.2, (b) y/b = 0.65, (c) y/b = 0.8, (d) y/b = 0.95

ONERA M6 RANS Analysis



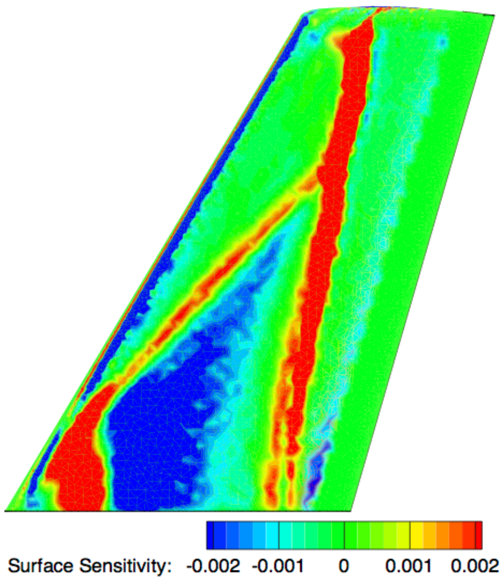
Tip: Tecplot ASCII is the default output, but ParaView and binary formats are available aerospacedesignlab

GETTING STARTED

BUILDING SU^2

ANALYSIS EXAMPLE:
ONERA M6 WING

**DESIGN EXAMPLE:
ONERA M6 WING**



ONERA M6 Shape Optimization (Euler and RANS)

Define and run the physical problem.
Evaluate geometry (thickness, AoA, etc).
Define 3D design variables.
- Create the FFD box (.su2 file).
- FFD design variables preprocessing.

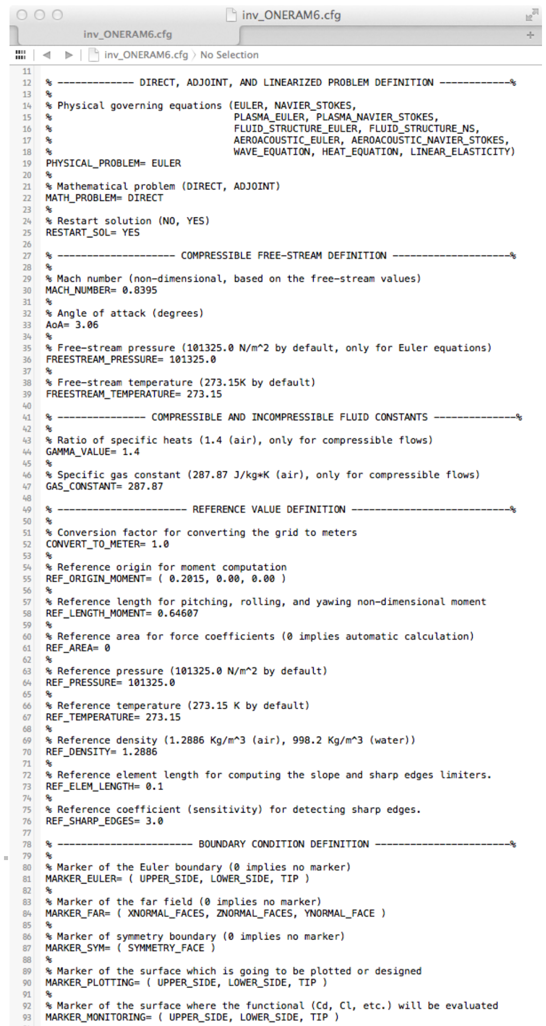
Define the optimization problem
- Objective function.
- Constraints (flow and geometry).
- Design variables based on FFD box.

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

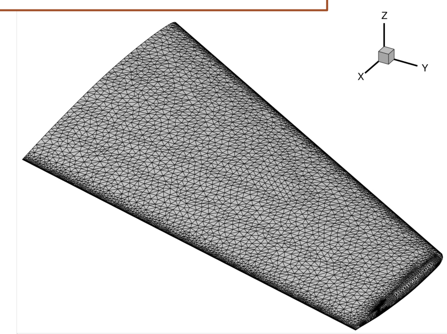
Final checks (optional).
- Restart files are available.
- The grid contains the FFD information.
- The stop criteria is reasonable.
- The proposed optimization problem makes sense (scaling).

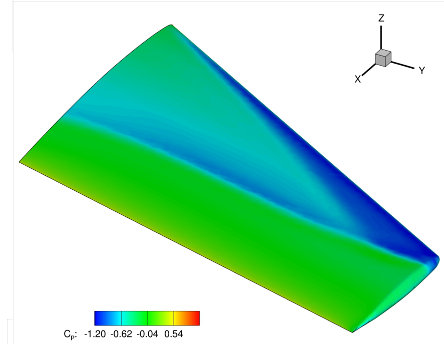
Run the optimization.
Analyze the solution.
- Folder structure and history_project file.
- Restart capability.

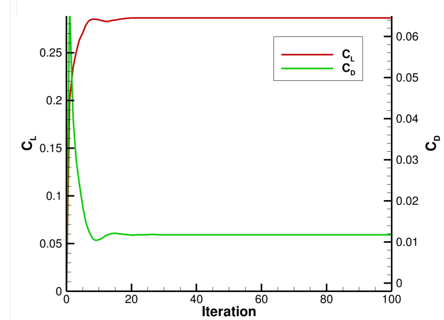
```
parallel_computation.py -n 4 -f inv_ONERAM6.cfg
```


inv_ONERAM6.cfg
inv_ONERAM6.cfg › No Selection

```
% ------------- DIRECT, ADJOINT, AND LINEARIZED PROBLEM DEFINITION ------------%
%
% Physical governing equations (EULER, NAVIER_STOKES,
%                               PLASMA_EULER, PLASMA_NAVIER_STOKES,
%                               FLUID_STRUCTURE_EULER, FLUID_STRUCTURE_NS,
%                               AEROACOUSTIC_EULER, AEROACOUSTIC_NAVIER_STOKES,
%                               WAVE_EQUATION, HEAT_EQUATION, LINEAR_ELASTICITY)
PHYSICAL_PROBLEM= EULER
%
% Mathematical problem (DIRECT, 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= 0.8395
%
% Angle of attack (degrees)
AoA= 3.06
%
% Free-stream pressure (101325.0 N/m^2 by default, only for Euler equations)
FREESTREAM_PRESSURE= 101325.0
%
% Free-stream temperature (273.15K by default)
FREESTREAM_TEMPERATURE= 273.15

% -------------- COMPRESSIBLE AND INCOMPRESSIBLE FLUID CONSTANTS --------------%
%
% Ratio of specific heats (1.4 (air), only for compressible flows)
GAMMA_VALUE= 1.4
%
% Specific gas constant (287.87 J/kg*K (air), only for compressible flows)
GAS_CONSTANT= 287.87

% ---------------------- REFERENCE VALUE DEFINITION ---------------------------%
%
% Conversion factor for converting the grid to meters
CONVERT_TO_METER= 1.0
%
% Reference origin for moment computation
REF_ORIGIN_MOMENT= ( 0.2015, 0.00, 0.00 )
%
% Reference length for pitching, rolling, and yawing non-dimensional moment
REF_LENGTH_MOMENT= 0.64607
%
% Reference area for force coefficients (0 implies automatic calculation)
REF_AREA= 0
%
% Reference pressure (101325.0 N/m^2 by default)
REF_PRESSURE= 101325.0
%
% Reference temperature (273.15 K by default)
REF_TEMPERATURE= 273.15
%
% Reference density (1.2886 Kg/m^3 (air), 998.2 Kg/m^3 (water))
REF_DENSITY= 1.2886
%
% Reference element length for computing the slope and sharp edges limiters.
REF_ELEM_LENGTH= 0.1
%
% Reference coefficient (sensitivity) for detecting sharp edges.
REF_SHARP_EDGES= 3.0

% ---------------------- BOUNDARY CONDITION DEFINITION ----------------------%
%
% Marker of the Euler boundary (0 implies no marker)
MARKER_EULER= ( UPPER_SIDE, LOWER_SIDE, TIP )
%
% Marker of the far field (0 implies no marker)
MARKER_FAR= ( XNORMAL_FACES, ZNORMAL_FACES, YNORMAL_FACE )
%
% Marker of symmetry boundary (0 implies no marker)
MARKER_SYM= ( SYMMETRY_FACE )
%
% Marker of the surface which is going to be plotted or designed
MARKER_PLOTTING= ( UPPER_SIDE, LOWER_SIDE, TIP )
%
% Marker of the surface where the functional (Cd, Cl, etc.) will be evaluated
MARKER_MONITORING= ( UPPER_SIDE, LOWER_SIDE, TIP )
```







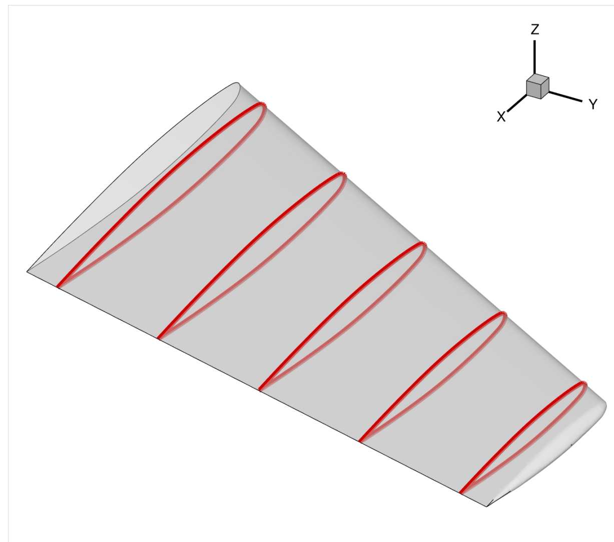
ONERA M6 Shape Optimization (Euler and RANS)

- Define and run the physical problem.
- **Evaluate geometry (thickness, AoA, etc).**
- Define 3D 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.

inv_ONERAM6.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= ( UPPER_SIDE, LOWER_SIDE, TIP )
%
% Number of airfoil sections
GEO_NUMBER_SECTIONS= 5
%
% Orientation of airfoil sections (X_AXIS, Y_AXIS, Z_AXIS)
GEO_ORIENTATION_SECTIONS= Y_AXIS
%
% Location (coordinate) of the airfoil sections (MinValue, MaxValue)
GEO_LOCATION_SECTIONS= (0.0806, 1.1284)
%
% Plot loads and Cp distributions on each airfoil section
GEO_PLOT_SECTIONS= NO
%
```

```
SU2_GEO inv_ONERAM6.cfg
```



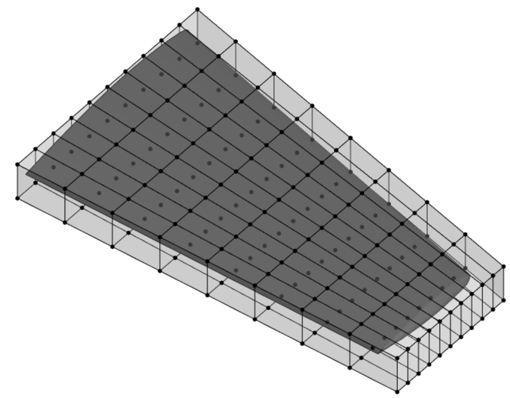

```
Section 1. Plane (yCoord): 0.0806.
Maximum thickness: 0.0760726.
1/4 chord thickness: 0.0722079.
1/3 chord thickness: 0.075645.
1/2 chord thickness: 0.0718373.
2/3 chord thickness: 0.054966.
3/4 chord thickness: 0.0436146.
Area: 0.0417745.
Angle of attack: 0.
Chord: 0.782102.

Section 2. Plane (yCoord): 0.34255.
Maximum thickness: 0.0684805.
1/4 chord thickness: 0.0650278.
1/3 chord thickness: 0.0681138.
1/2 chord thickness: 0.0646917.
2/3 chord thickness: 0.0495041.
3/4 chord thickness: 0.0392899.
Area: 0.0338872.
Angle of attack: 0.
Chord: 0.704434.

Section 3. Plane (yCoord): 0.6045.
Maximum thickness: 0.060948.
1/4 chord thickness: 0.0578653.
1/3 chord thickness: 0.060569.
1/2 chord thickness: 0.0575322.
2/3 chord thickness: 0.0440336.
3/4 chord thickness: 0.0349501.
Area: 0.0268216.
Angle of attack: 0.
Chord: 0.626766.

Section 4. Plane (yCoord): 0.86645.
Maximum thickness: 0.0533332.
1/4 chord thickness: 0.0506826.
1/3 chord thickness: 0.0531113.
1/2 chord thickness: 0.0504042.
2/3 chord thickness: 0.0385875.
3/4 chord thickness: 0.0306302.
Area: 0.0205833.
Angle of attack: 0.
Chord: 0.549098.

Section 5. Plane (yCoord): 1.1284.
Maximum thickness: 0.0458274.
1/4 chord thickness: 0.0435065.
1/3 chord thickness: 0.0456027.
1/2 chord thickness: 0.0432516.
2/3 chord thickness: 0.0331341.
3/4 chord thickness: 0.0262889.
Area: 0.0151681.
Angle of attack: 0.
Chord: 0.47143.
```

ONERA M6 Shape Optimization (Euler and RANS)

- Define and run the physical problem.
- Evaluate geometry (thickness, AoA, etc).
- Define 3D 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.

mesh_ONERAM6_inv.su2 — Edited

```
5       9202    9205    9201
5       483     1677    8773
5       4678    19336   19323
FFD_NBOX= 1
FFD_NLEVEL= 1
FFD_TAG= WING
FFD_LEVEL= 0
FFD_DEGREE_I= 10
FFD_DEGREE_J= 8
FFD_DEGREE_K= 1
FFD_PARENTS= 0
FFD_CHILDREN= 0
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= 0
FFD_SURFACE_POINTS= 0
```



inv_ONERAM6.cfg

```
% ---------------------- DESIGN VARIABLE PARAMETERS -------------------------%
%
% Kind of deformation (FFD_SETTING, FFD_CONTROL_POINT_2D, FFD_CAMBER_2D, FFD_THICKNESS_2D,
%                      HICKS_HENNE, COSINE_BUMP, PARABOLIC,
%                      NACA_4DIGITS, DISPLACEMENT, ROTATION, FFD_CONTROL_POINT,
%                      FFD_DIHEDRAL_ANGLE, FFD_TWIST_ANGLE, FFD_ROTATION,
%                      FFD_CAMBER, FFD_THICKNESS, FFD_CONTROL_SURFACE, SURFACE_FILE, AIRFOIL)
DV_KIND= FFD_SETTING
%
% Marker of the surface in which we are going apply the shape deformation
DV_MARKER= ( UPPER_SIDE, LOWER_SIDE, TIP )
%
% Parameters of the shape deformation
%       - FFD_CONTROL_POINT ( Chunk, i_Ind, j_Ind, k_Ind, x_Disp, y_Disp, z_Disp )
%       - FFD_DIHEDRAL_ANGLE ( Chunk, x_Orig, y_Orig, z_Orig, x_End, y_End, z_End )
%       - FFD_TWIST_ANGLE ( Chunk, x_Orig, y_Orig, z_Orig, x_End, y_End, z_End )
%       - FFD_ROTATION ( Chunk, x_Orig, y_Orig, z_Orig, x_End, y_End, z_End )
%       - FFD_CAMBER ( Chunk, i_Ind, j_Ind )
%       - FFD_THICKNESS ( Chunk, i_Ind, j_Ind )
%       - FFD_VOLUME ( Chunk, i_Ind, j_Ind )
DV_PARAM= ( WING, 1, 0, 0, 0.0, 0.0, 1.0 )
%
% New value of the shape deformation
DV_VALUE= 0.0
```

```
SU2_DEF inv_ONERAM6.cfg
```

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

ONERA M6 Shape Optimization (Euler and RANS)

- Define and run the physical problem.
- Evaluate geometry (thickness, AoA, etc).
- Define 3D 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.

inv_ONERAM6.cfg

inv_ONERAM6.cfg 〉 No Selection

```
% -------------------- 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 )
%    COSINE_BUMP  (  2, Scale | Mark. List | Lower(0)/Upper(1) side, x_Loc, x_Size )
%    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 | Chunk, i_Ind, j_Ind, k_Ind, x_Mov, y_Mov, z_Mov )
%    FFD_DIHEDRAL_ANGLE (  8, Scale | Mark. List | Chunk, x_Orig, y_Orig, z_Orig, x_End, y_End, z_End )
%    FFD_TWIST_ANGLE  (  9, Scale | Mark. List | Chunk, x_Orig, y_Orig, z_Orig, x_End, y_End, z_End )
%    FFD_ROTATION   ( 10, Scale | Mark. List | Chunk, x_Orig, y_Orig, z_Orig, x_End, y_End, z_End )
%    FFD_CAMBER   ( 11, Scale | Mark. List | Chunk, i_Ind, j_Ind )
%    FFD_THICKNESS  ( 12, Scale | Mark. List | Chunk, i_Ind, j_Ind )
%    FFD_VOLUME   ( 13, Scale | Mark. List | Chunk, 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 * 0.1
%
% 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 design variables, separated by semicolons
DEFINITION_DV= ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE, TIP | WING, 0, 1, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE, TIP | WING,
  1, 1, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE, TIP | WING, 2, 1, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE,
  TIP | WING, 3, 1, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE, TIP | WING, 4, 1, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | UPPER_SIDE,
  LOWER_SIDE, TIP | WING, 5, 1, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE, TIP | WING, 6, 1, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 |
  UPPER_SIDE, LOWER_SIDE, TIP | WING, 7, 1, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE, TIP | WING, 8, 1, 0, 0.0, 0.0,
  1.0 ); ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE, TIP | WING, 9, 1, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE, TIP | WING, 10, 1,
  0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE, TIP | WING, 0, 2, 0, 0.0, 0.0, 1.0 ); ( 7, 1.0 | UPPER_SIDE, LOWER_SIDE, TIP |
```

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

Building SU²

Tips for Hassle-Free Builds

Tip: Use the set_ffd_design_var.py script in order to easily define your set of FFD design variables in the configuration file.

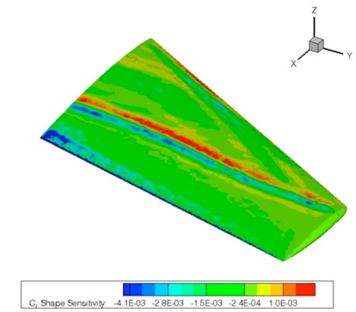
```
$ ./set_ffd_design_var.py -i 30 -j 1 -b MAIN_BOX -m airfoil -s 1.0
```

ONERA M6 Shape Optimization (Euler and RANS)

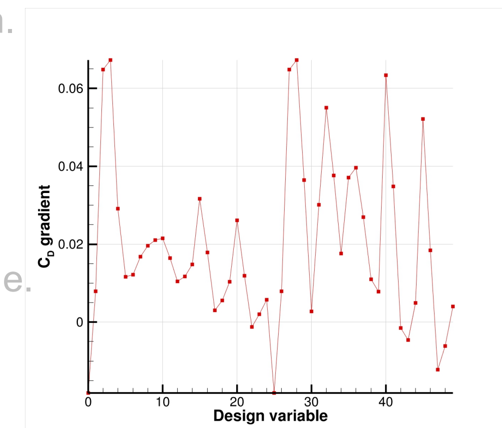
- Define and run the physical problem.
- Evaluate geometry (thickness, AoA, etc).
- Define 3D design variables.
  - Create the FFD box (.su2 file).
  - FFD design variables preprocessing.
- Define the optimization problem
  - Objective function.
  - Constraints (flow and geometry).
  - Design variables 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.

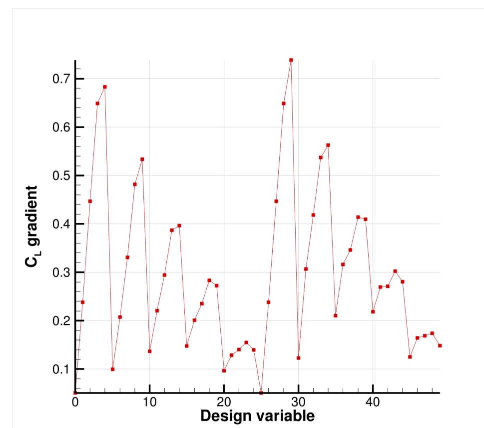
inv_ONERAM6.cfg

```
% -------------------- 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
```





ONERA M6 Shape Optimization (Euler and RANS)

- Define and run the physical problem.
- Evaluate geometry (thickness, AoA, etc).
- Define 3D 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.

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
```

ONERA M6 Shape Optimization (Euler and RANS)

Define and run the physical problem.
Evaluate geometry (thickness, AoA, etc).
Define 3D design variables.
- Create the FFD box (.su2 file).
- FFD design variables preprocessing.

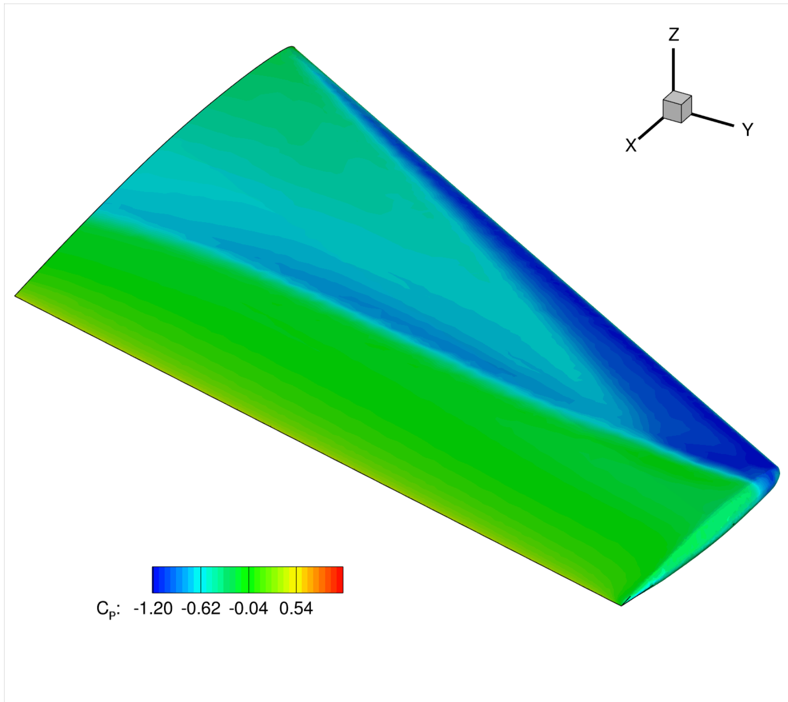
Define the optimization problem
- Objective function.
- Constraints (flow and geometry).
- Design variables based on FFD box.

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.

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



ONERA M6 Shape Optimization (Euler and RANS)

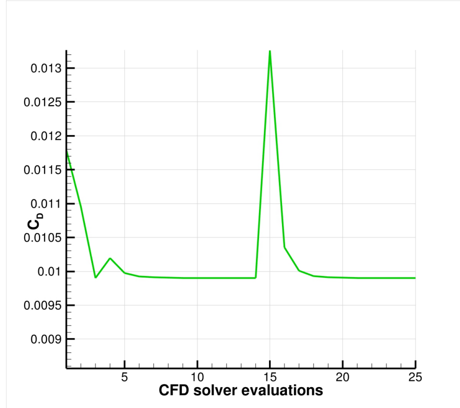
Define and run the physical problem.
Evaluate geometry (thickness, AoA, etc).
Define 3D design variables.
- Create the FFD box (.su2 file).
- FFD design variables preprocessing.

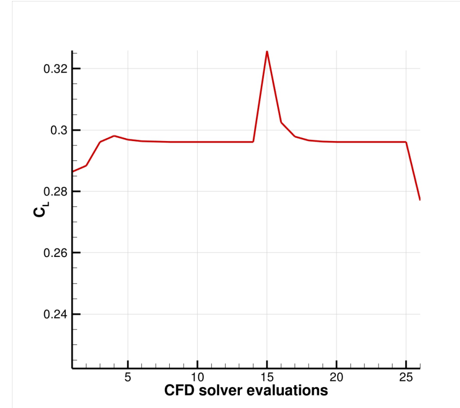
Define the optimization problem
- Objective function.
- Constraints (flow and geometry).
- Design variables based on FFD box.

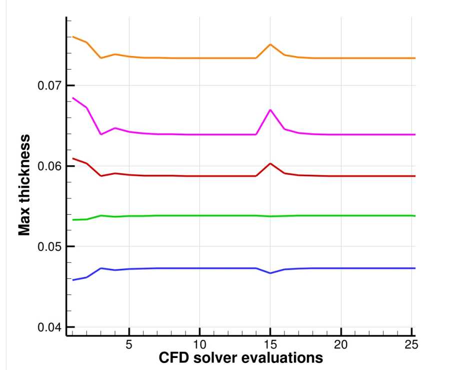
Checks before the optimization (optional).
- Compute C_D, and C_L gradients.
- Compute geometric gradients.

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.







RANS optimization (49 design variables)
Baseline C_D= 0.0197, C_L = 0.2684
Final C_D= 0.0179, C_L = 0.2731

Euler optimization (175 design variables)
Baseline C_D= 0.0118, C_L = 0.2864
Final C_D= 0.0099, C_L = 0.2960

SU²

The Open-Source CFD Code

Thanks a lot for your attention!

Questions & Answers

More details in http://su2.stanford.edu/