SU^2: Advanced Analysis Topics

OpenMDAO-SU^2 Joint Workshop

Stanford University
Tuesday, October 1, 2013

Thomas D. Economon
Department of Aeronautics & Astronautics
Stanford University

The Open-Source CFD Code

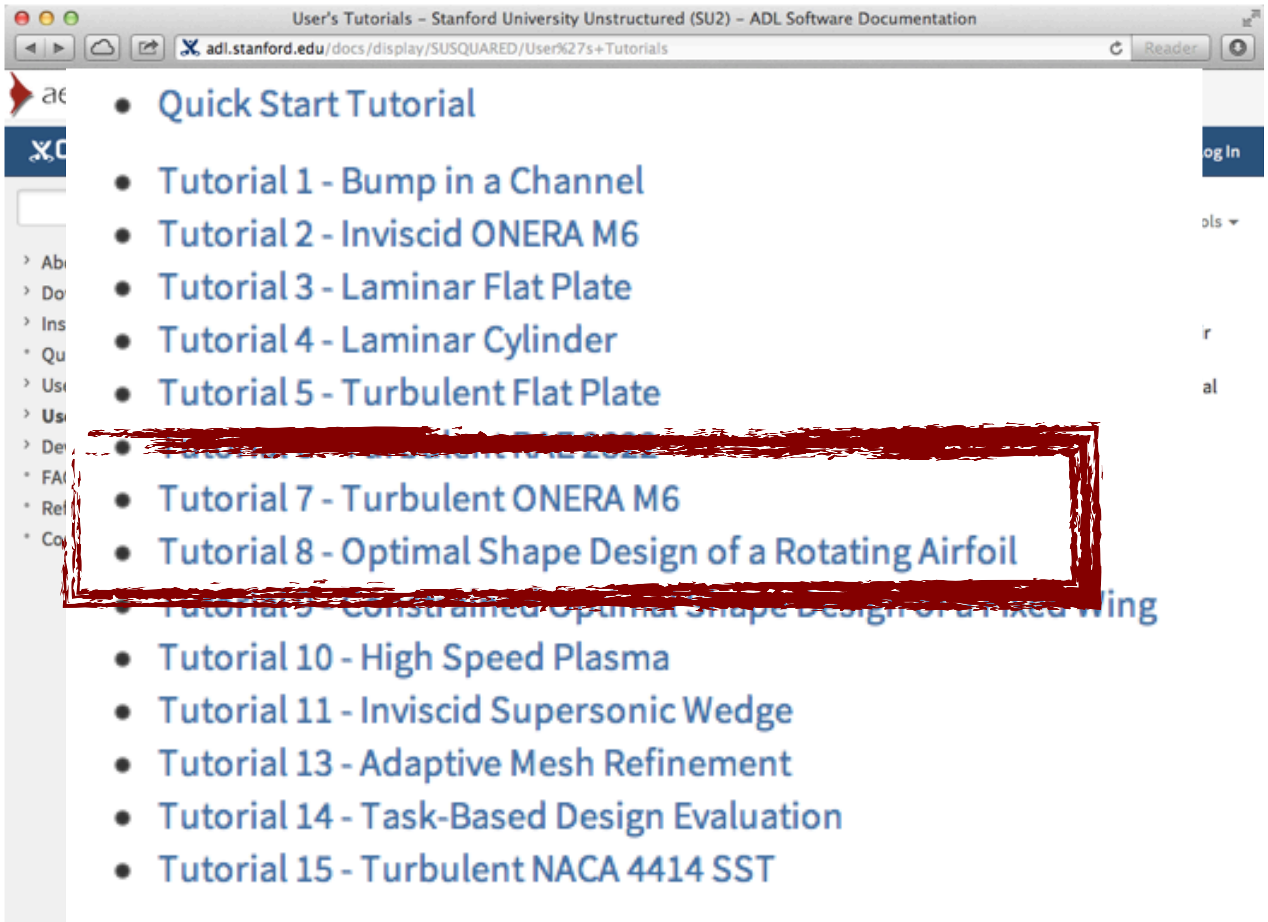
Three Main Topics

- Getting up to speed with SU^2
  - Configuring a RANS simulation
- Advanced Analysis Topics
  - Steady flows with moving walls
  - Unsteady flows with fixed geometry
  - Unsteady flows on dynamic meshes
- Getting started with optimal shape design
  - Setting up a 2-D airfoil optimization

Getting up to Speed with SU^2

- Best way to get going? User Tutorials!

We'll focus here...



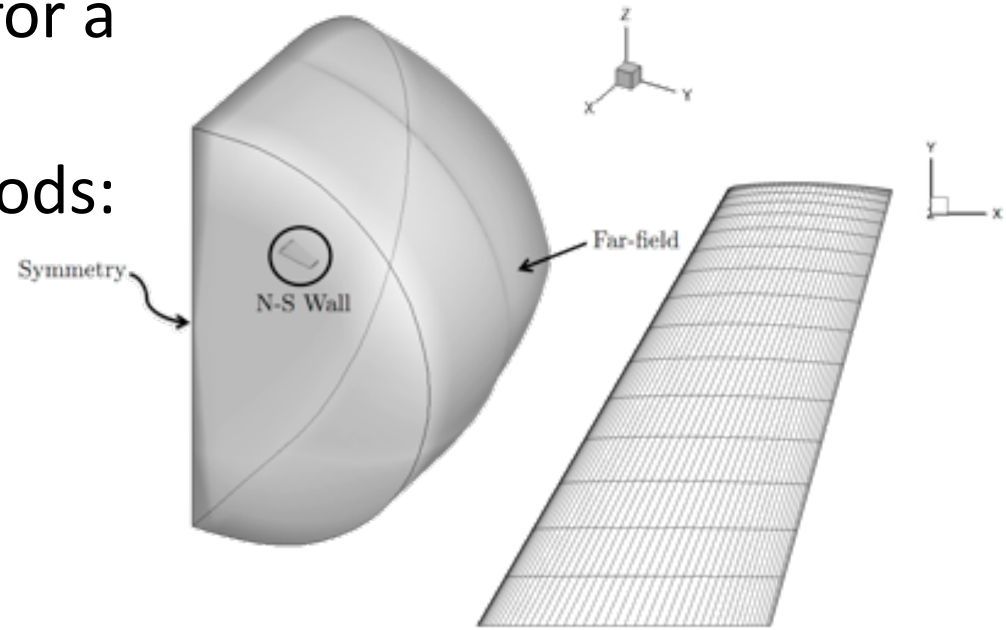
Getting up to Speed with SU^2

The workflow for RANS...

- Prepare geometry & mesh beforehand.
- Choose appropriate physics.
- Set proper conditions for a viscous simulation.
- Select numerical methods:
  - Convective terms
  - Viscous terms
  - Time Integration
  - Multi-grid
- Run the analysis.
- Post-process the results.

```
% Mesh input file
MESH_FILENAME= mesh_ONERAM6_turb_hexa.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).
```



Getting up to Speed with SU^2

- Prepare geometry & mesh beforehand.
- Choose appropriate physics.
- **Set proper conditions for a viscous simulation.**
- Select numerical methods:
  - Convective terms
  - Viscous terms
  - Time Integration
  - Multi-grid
- Run the analysis.
- Post-process the results.

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.

oneram6 — vim — 86×47

```
%
% Conversion factor for converting the grid to meters
CONVERT_TO_METER= 1.0
%
% Write a new mesh converted to meters (NO, YES)
WRITE_CONVERTED_MESH = NO
```

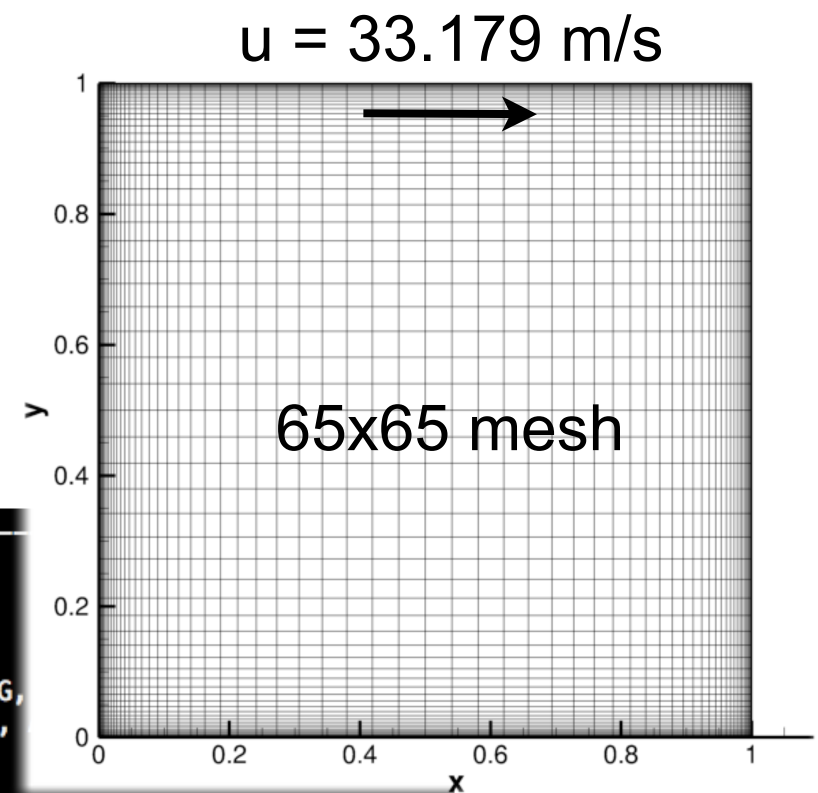
Advanced Analysis Topics

- New "Dynamic Mesh" config options
- Unified approach for:
  - Moving walls & rotating frame (steady)
  - Rigidly transforming meshes (unsteady)
  - Dynamically deforming meshes (unsteady)

```
% ----------------------- DYNAMIC MESH DEFINITION -----------------------------%
%
% Dynamic mesh simulation (NO, YES)
GRID_MOVEMENT= NO
%
% Type of dynamic mesh (NONE, RIGID_MOTION, DEFORMING, ROTATING_FRAME,
%                       MOVING_WALL, FLUID_STRUCTURE, AEROELASTIC, ELASTICITY,
%                       EXTERNAL)
GRID_MOVEMENT_KIND= DEFORMING
%
% Motion mach number (non-dimensional). Used for initializing a viscous flow
% with the Reynolds number and for computing force coeffs. with dynamic meshes.
MACH_MOTION= 0.8
```

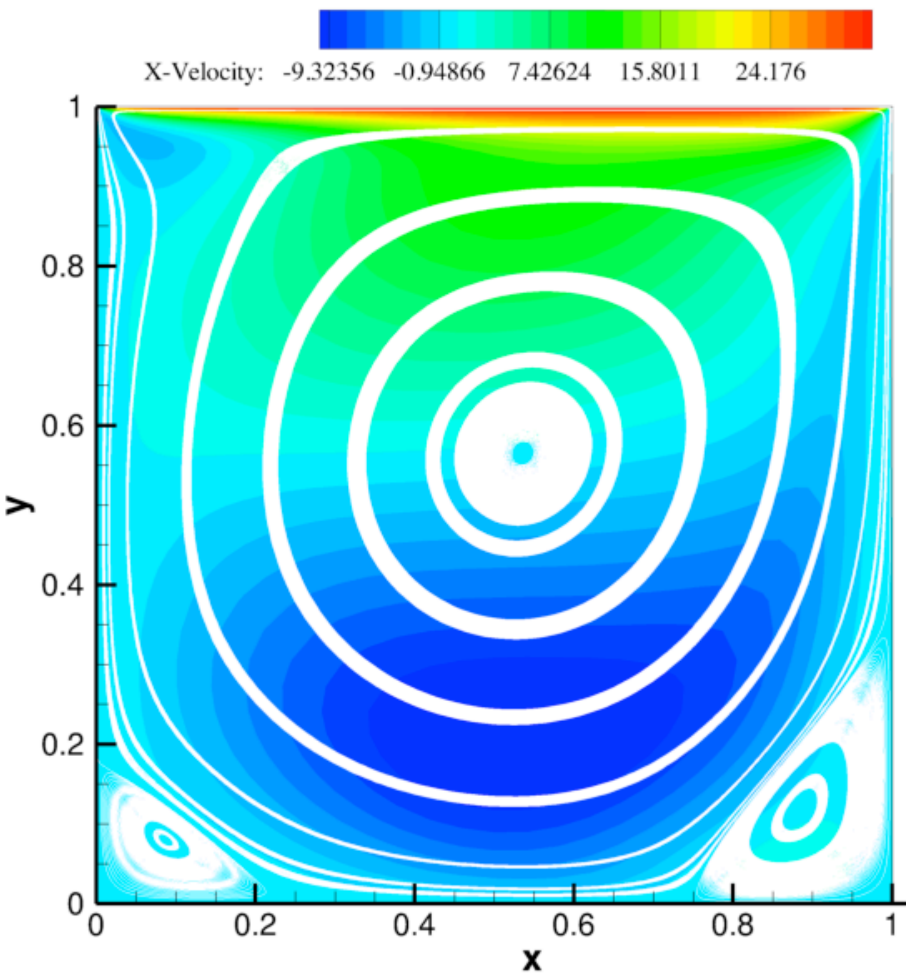
Moving Walls: Lid-driven Cavity

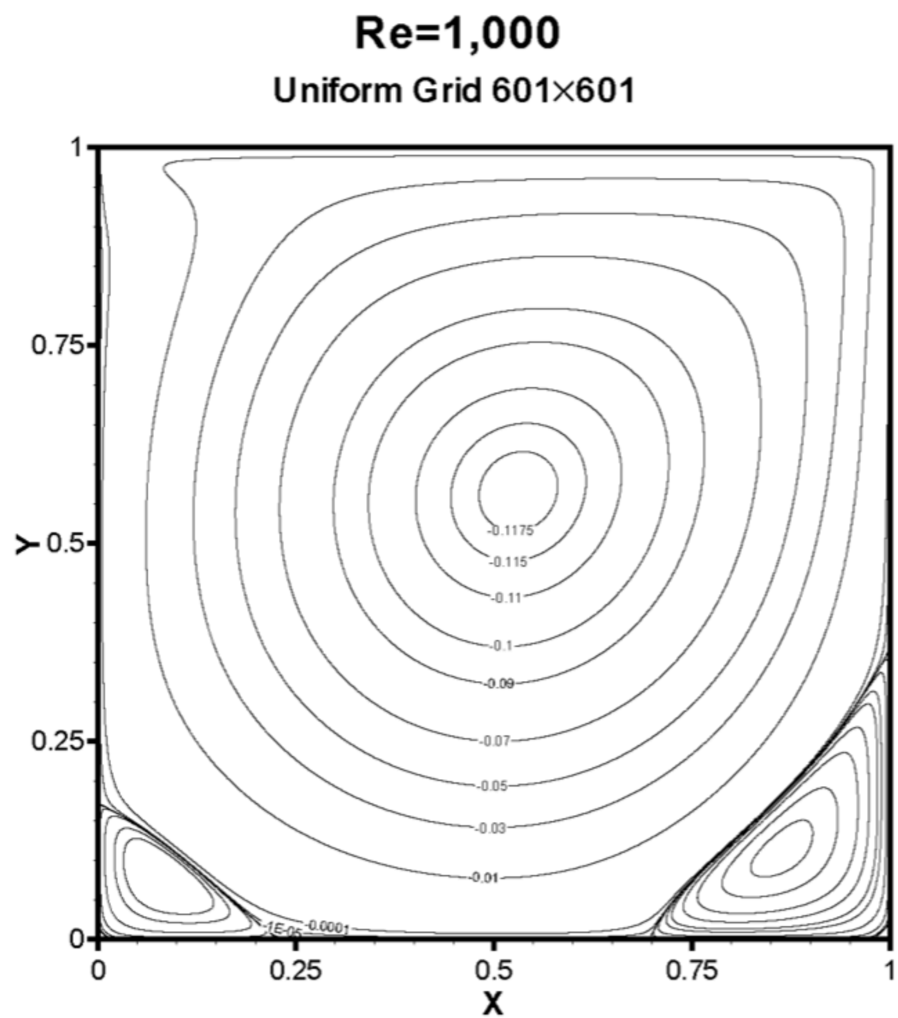
- Select MOVING_WALL
- Input the name of the moving marker(s)
- Enter the translation rate for marker(s)




```
% ------------------------ DYNAMIC MESH DEFINITION ----
%
% Dynamic mesh simulation (NO, YES)
GRID_MOVEMENT= YES
%
% Type of dynamic mesh (NONE, RIGID_MOTION, DEFORMING,
%                       MOVING_WALL, FLUID_STRUCTURE,
%                       EXTERNAL)
GRID_MOVEMENT_KIND= MOVING_WALL
%
% Motion mach number (non-dimensional). Used for initializing a viscous flow
% with the Reynolds number and for computing force coeffs. with dynamic meshes.
MACH_MOTION= 0.1
%
% Moving wall boundary marker(s) (NONE = no marker, ignored for RIGID_MOTION)
MARKER_MOVING= ( upper )
%
% Translational velocity (m/s) in the x, y, & z directions
TRANSLATION_RATE_X= 33.179
TRANSLATION_RATE_Y= 0.0
TRANSLATION_RATE_Z= 0.0
```

Moving Walls: Lid-driven Cavity

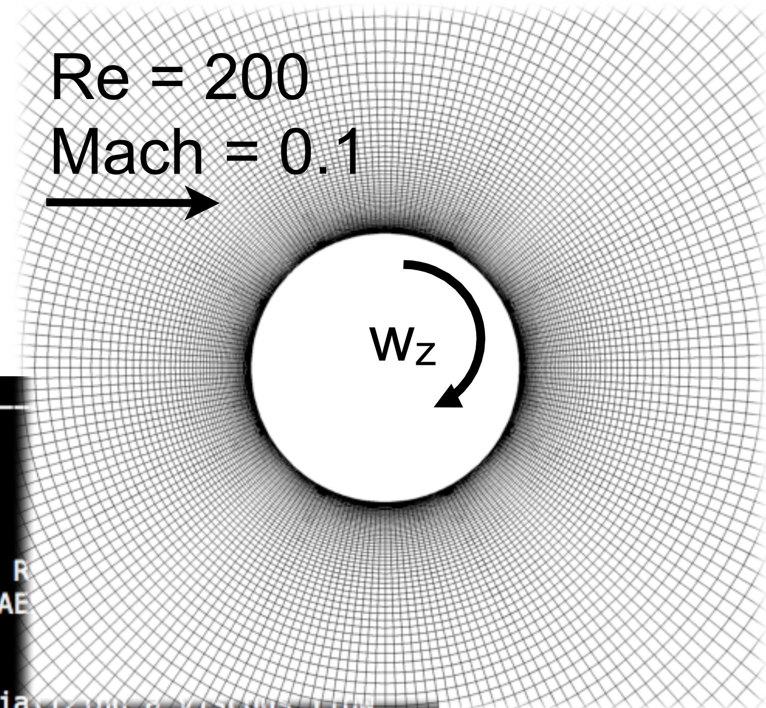




Erturk, Corke, and Gokcol (2005), "Numerical Solutions of 2-D Steady Incompressible Driven Cavity Flow at High Reynolds Numbers", International Journal for Numerical Methods in Fluids, Vol. 48, pp. 747-774.

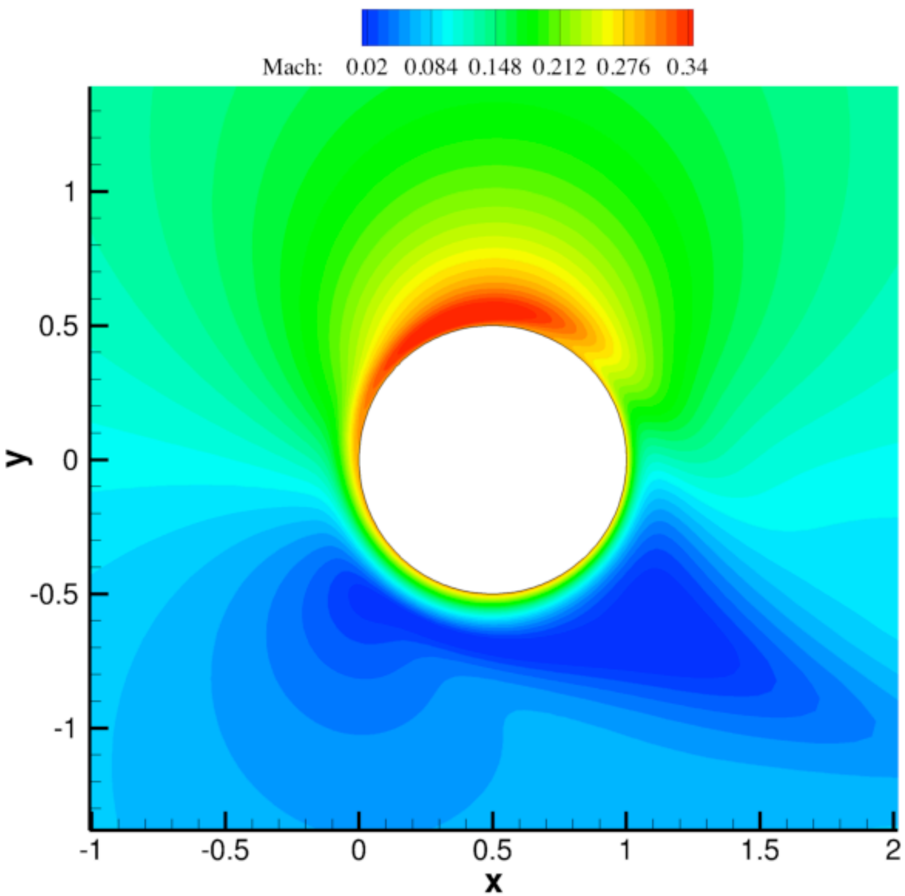
Moving Walls: Spinning Cylinder

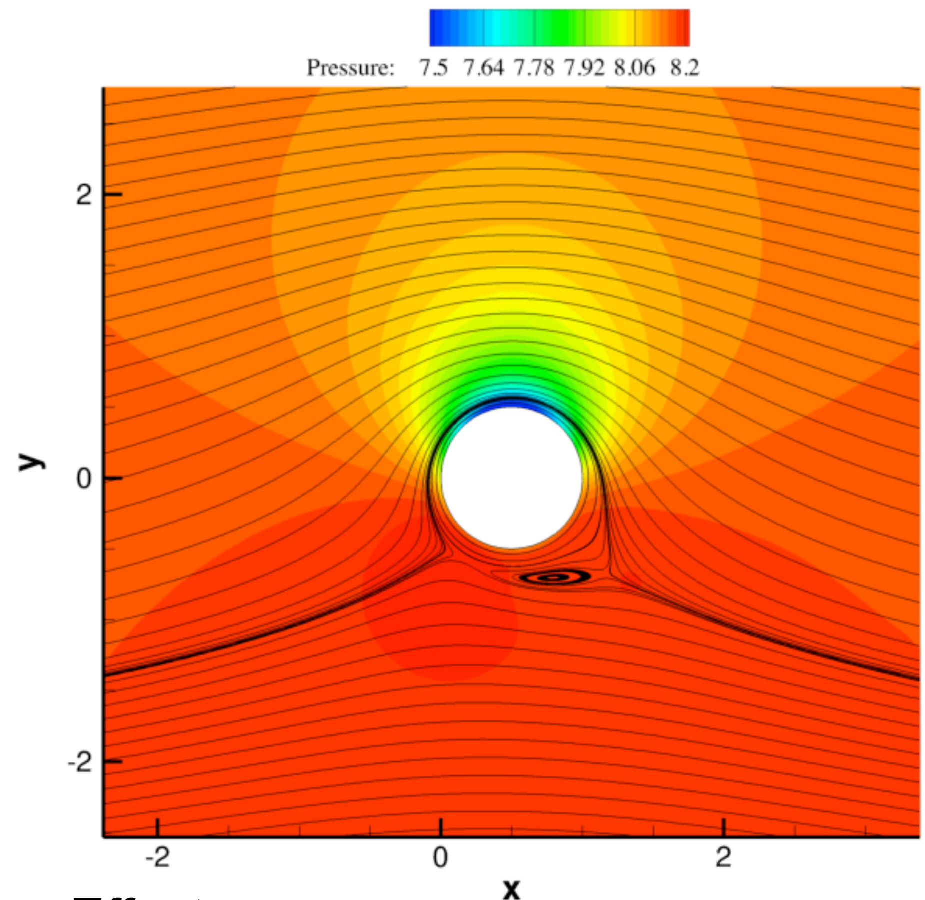
- Select MOVING_WALL again
- Input the name of the rotating marker(s), center(s) or rotation, and rotation rate(s)




```
% ----------------------- DYNAMIC MESH DEFINITION ------
%
% Dynamic mesh simulation (NO, YES)
GRID_MOVEMENT= YES
%
% Type of dynamic mesh (NONE, RIGID_MOTION, DEFORMING, R
%                       MOVING_WALL, FLUID_STRUCTURE, AE
GRID_MOVEMENT_KIND= MOVING_WALL
%
% Motion mach number (non-dimensional). Used for intitializing a viscous flow
% with the Reynolds number and for computing force coeffs. with dynamic meshes.
MACH_MOTION= 0.1
%
% Moving wall boundary marker(s) (NONE = no marker, ignored for RIGID_MOTION)
MARKER_MOVING= ( cylinder )
%
% Coordinates of the motion origin
MOTION_ORIGIN_X= 0.5
MOTION_ORIGIN_Y= 0.0
MOTION_ORIGIN_Z= 0.0
%
% Angular velocity vector (rad/s) about the motion origin
ROTATION_RATE_X = 0.0
ROTATION_RATE_Y = 0.0
ROTATION_RATE_Z = -199.0738
```

Moving Walls: Spinning Cylinder





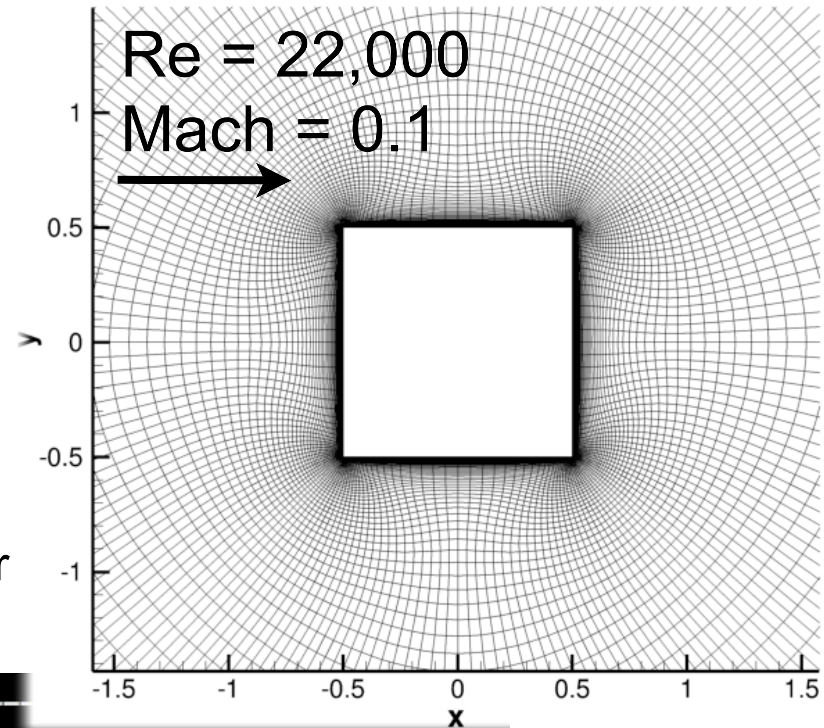
Magnus Effect...

C_l = 9.961930

C_d = 0.093797

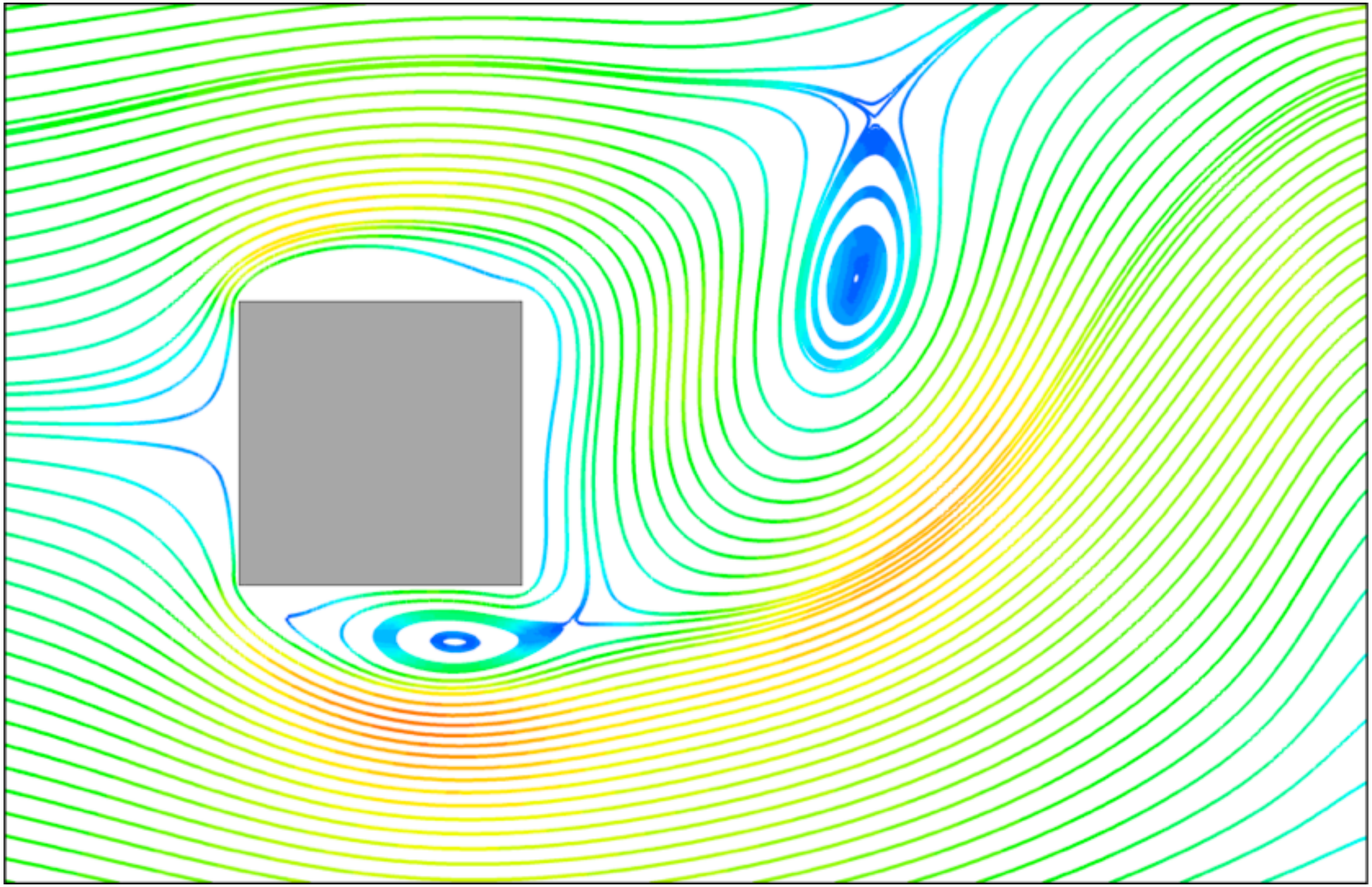
Unsteady RANS: Square Cylinder

- Dual time strategy (1st- or 2nd-order)
- Choose a physical time step using UNST_TIMESTEP
- Set the maximum physical time with UNST_TIME
- UNST_INT_ITER: maximum number of pseudo-time steps
- Regular convergence criteria apply for converging each physical time step




```
% ------------------------- UNSTEADY SIMULATION ------
%
% Unsteady simulation (NO, TIME_STEPPING, DUAL_TIME_STEPPING-1ST_ORDER,
%                      DUAL_TIME_STEPPING-2ND_ORDER, TIME_SPECTRAL)
UNSTEADY_SIMULATION= DUAL_TIME_STEPPING-2ND_ORDER
%
% Time Step for dual time stepping simulations (s)
UNST_TIMESTEP= 0.0015
%
% Total Physical Time for dual time stepping simulations (s)
UNST_TIME= 3.75
% 2500 iterations
%
% Number of internal iterations (dual time method)
UNST_INT_ITER= 2000
```

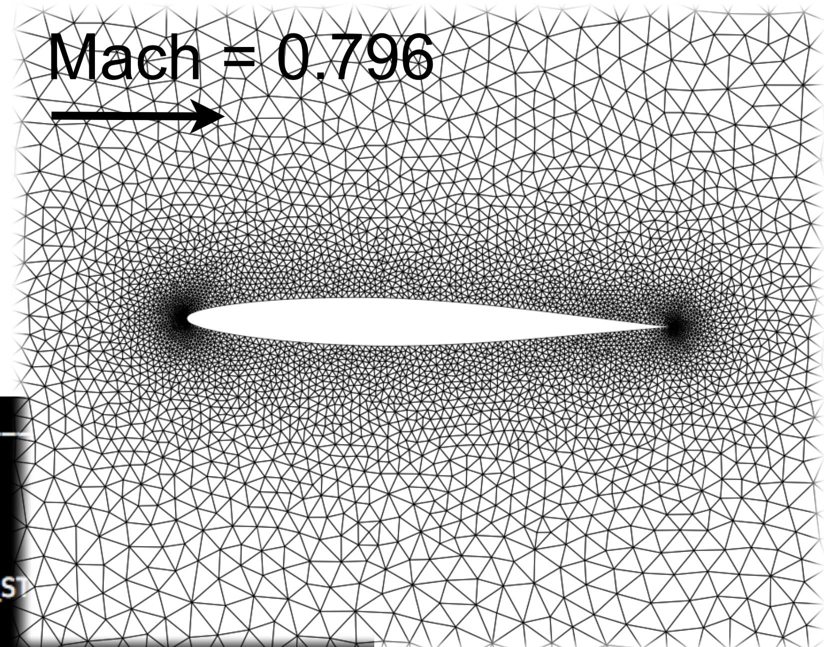
Unsteady RANS: Square Cylinder



Time-averaged C_d = 2.16 agrees well with similar results reported in Iaccarino, G., et al. "Reynolds averaged simulation of unsteady separated flow." *International Journal of Heat and Fluid Flow* 24.2 (2003): 147-156.

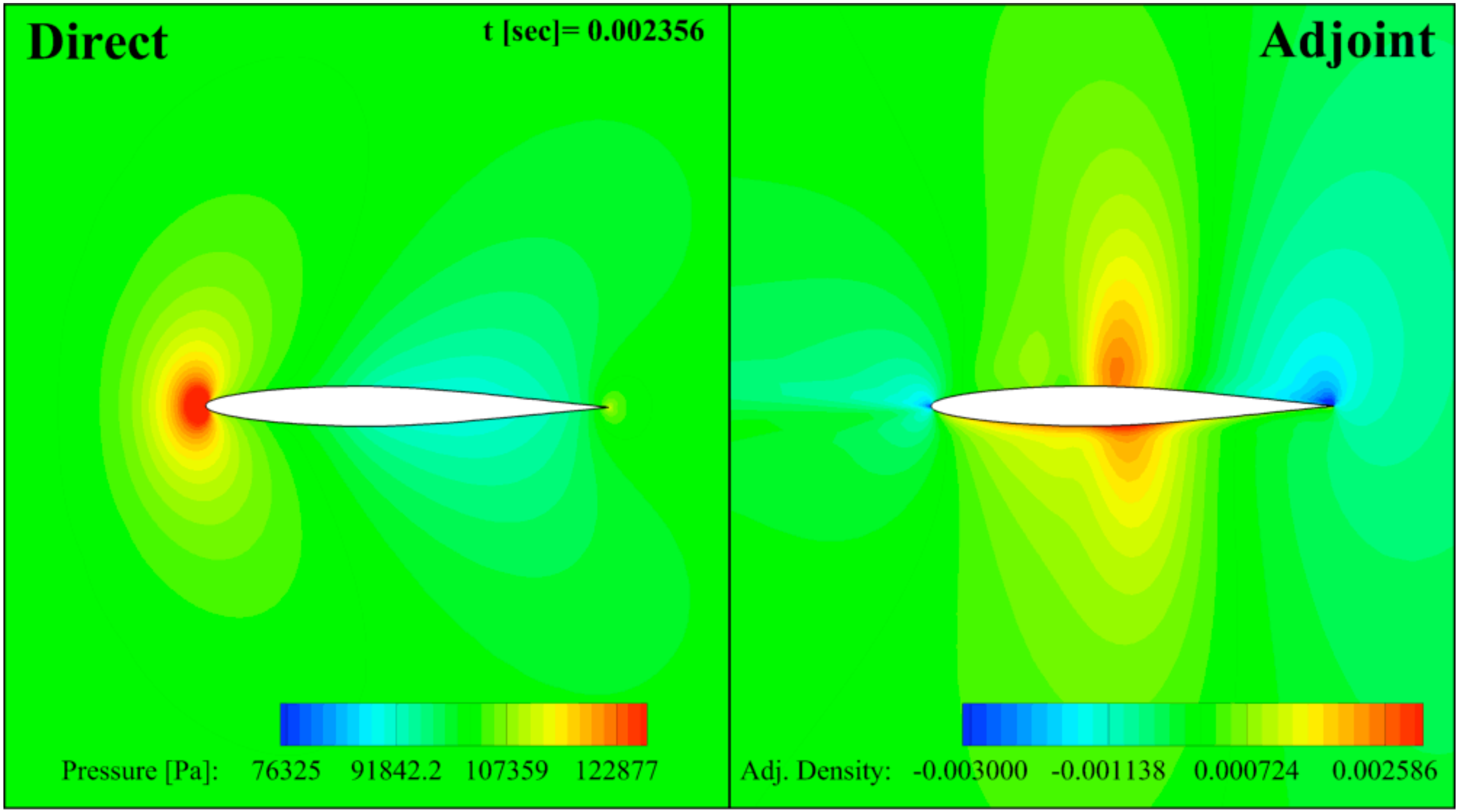
Unsteady Flow: Pitching NACA 64A010

- Set up dual-time strategy as before
- Use RIGID_MOTION option
- Need to give a pitching origin, frequency, amplitude, and possibly phase offset




```
% ----------------------- DYNAMIC MESH DEFINITION ----------
%
% Dynamic mesh simulation (NO, YES)
GRID_MOVEMENT= YES
%
% Type of mesh motion (NONE, FLUTTER, RIGID_MOTION, FLUID_ST
GRID_MOVEMENT_KIND= RIGID_MOTION
%
MACH_MOTION= 0.796
%
% Coordinates of the rigid motion origin
MOTION_ORIGIN_X= 0.248
MOTION_ORIGIN_Y= 0.0
MOTION_ORIGIN_Z= 0.0
%
% Pitching angular freq. (rad/s) about x, y, & z axes (RIGID_MOTION only)
PITCHING_OMEGA_X= 0.0
PITCHING_OMEGA_Y= 0.0
PITCHING_OMEGA_Z= 106.69842
%
% Pitching amplitude (degrees) about x, y, & z axes (RIGID_MOTION only)
PITCHING_AMPL_X= 0.0
PITCHING_AMPL_Y= 0.0
PITCHING_AMPL_Z= 1.01
```

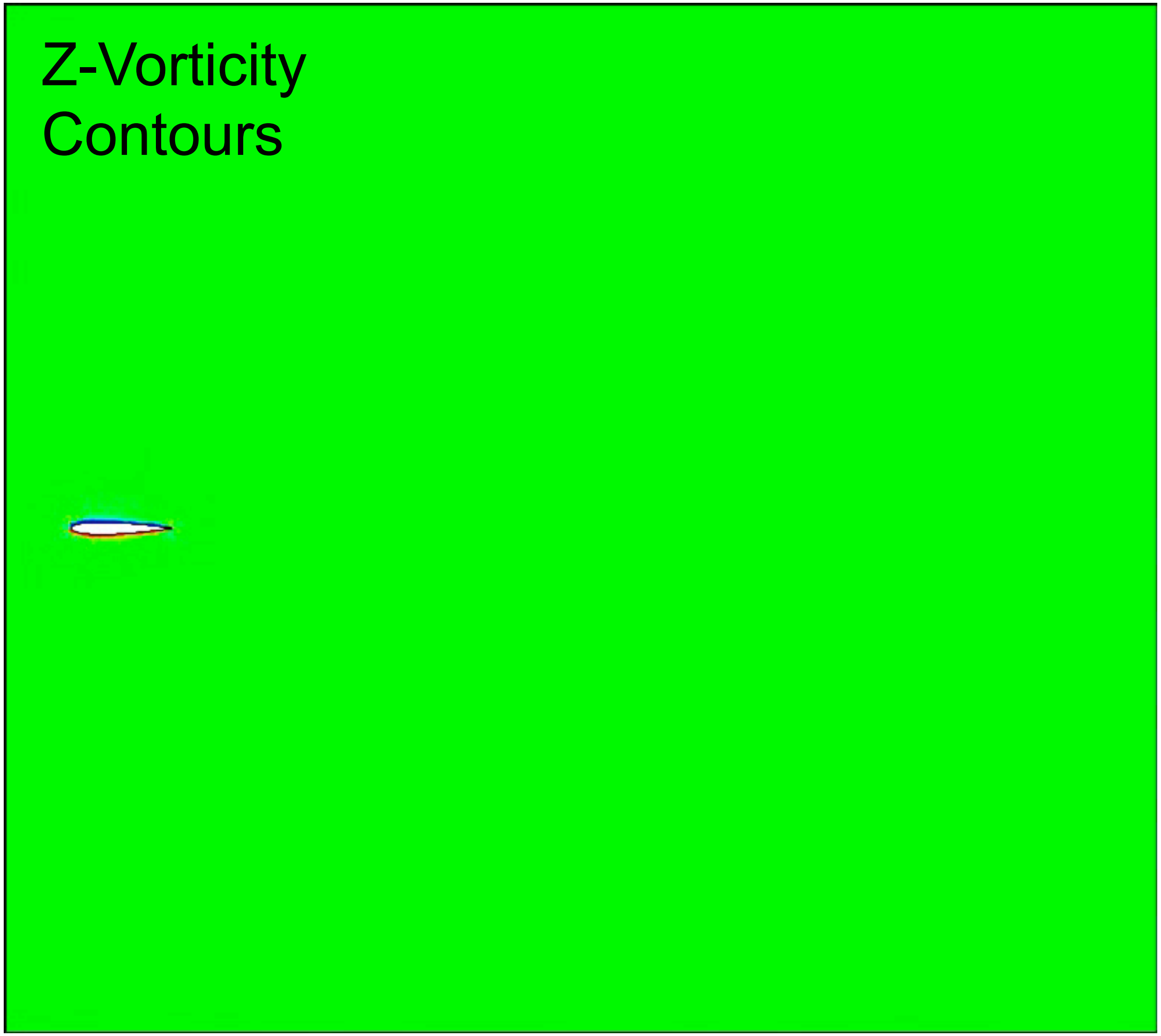
Unsteady Flow: Pitching NACA 64A010



(Time-averaged drag functional)

Animation

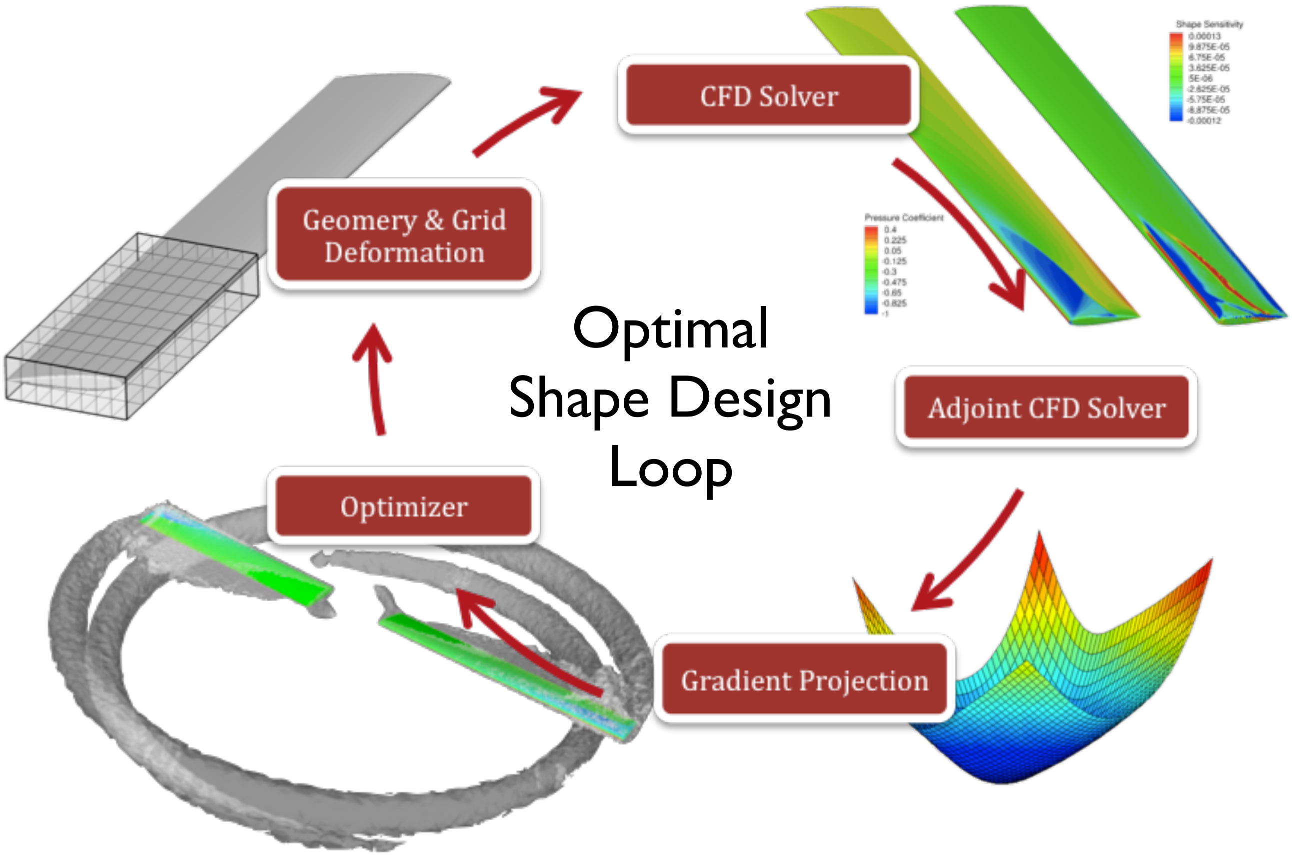
Pitching + Plunging NACA 0012



Plunging is also available... Mach = 0.3, Re = 1000

Animation

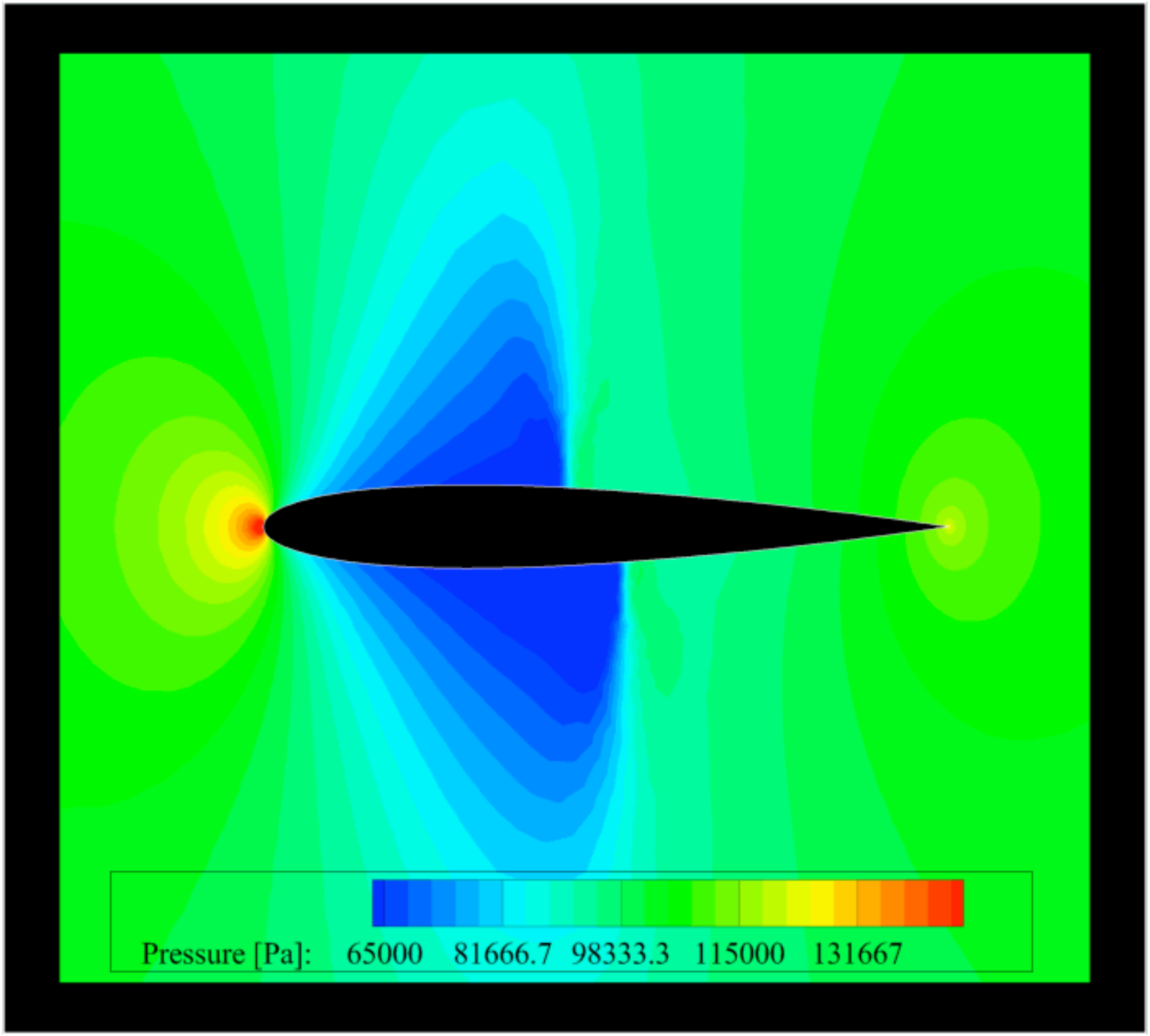
Getting Started with Shape Design



Shape Optimization is a defining feature of SU²!

– How does one set up an optimization problem?
– How does one use Hicks-Henne bumps?

Shape Design of a Rotating Airfoil



Animation

Shape Design of a Rotating Airfoil

The shape design problem...

```
% 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= NONE
%
% Optimization design variables, separated by semicolons
DEFINITION_DV= ( 1, 1.0 | airfoil | 0, 0.961538461538 ); ( 1, 1.0 | airfoil | 0, 0.923076923077 ); (
 1, 1.0 | airfoil | 0, 0.884615384615 ); ( 1, 1.0 | airfoil | 0, 0.846153846154 ); ( 1, 1.0 | airfoi
l | 0, 0.807692307692 ); ( 1, 1.0 | airfoil | 0, 0.769230769231 ); ( 1, 1.0 | airfoil | 0, 0.7307692
30769 ); ( 1, 1.0 | airfoil | 0, 0.692307692308 ); ( 1, 1.0 | airfoil | 0, 0.653846153846 ); ( 1, 1.
0 | airfoil | 0, 0.615384615385 ); ( 1, 1.0 | airfoil | 0, 0.576923076923 ); ( 1, 1.0 | airfoil | 0,
 0.538461538462 ); ( 1, 1.0 | airfoil | 0, 0.5 ); ( 1, 1.0 | airfoil | 0, 0.461538461538 ); ( 1, 1.0
 | airfoil | 0, 0.423076923077 ); ( 1, 1.0 | airfoil | 0, 0.384615384615 ); ( 1, 1.0 | airfoil | 0,
0.346153846154 ); ( 1, 1.0 | airfoil | 0, 0.307692307692 ); ( 1, 1.0 | airfoil | 0, 0.269230769231 )
; ( 1, 1.0 | airfoil | 0, 0.230769230769 ); ( 1, 1.0 | airfoil | 0, 0.192307692308 ); ( 1, 1.0 | air
foil | 0, 0.153846153846 ); ( 1, 1.0 | airfoil | 0, 0.115384615385 ); ( 1, 1.0 | airfoil | 0, 0.0769
230769231 ); ( 1, 1.0 | airfoil | 0, 0.0384615384615 ); ( 1, 1.0 | airfoil | 1, 0.0384615384615 ); (
 1, 1.0 | airfoil | 1, 0.0769230769231 ); ( 1, 1.0 | airfoil | 1, 0.115384615385 ); ( 1, 1.0 | airfo
il | 1, 0.153846153846 ); ( 1, 1.0 | airfoil | 1, 0.192307692308 ); ( 1, 1.0 | airfoil | 1, 0.230769
230769 ); ( 1, 1.0 | airfoil | 1, 0.269230769231 ); ( 1, 1.0 | airfoil | 1, 0.307692307692 ); ( 1, 1
.0 | airfoil | 1, 0.346153846154 ); ( 1, 1.0 | airfoil | 1, 0.384615384615 ); ( 1, 1.0 | airfoil | 1
, 0.423076923077 ); ( 1, 1.0 | airfoil | 1, 0.461538461538 ); ( 1, 1.0 | airfoil | 1, 0.5 ); ( 1, 1.
0 | airfoil | 1, 0.538461538462 ); ( 1, 1.0 | airfoil | 1, 0.576923076923 ); ( 1, 1.0 | airfoil | 1,
 0.615384615385 ); ( 1, 1.0 | airfoil | 1, 0.653846153846 ); ( 1, 1.0 | airfoil | 1, 0.692307692308
); ( 1, 1.0 | airfoil | 1, 0.730769230769 ); ( 1, 1.0 | airfoil | 1, 0.769230769231 ); ( 1, 1.0 | ai
rfoil | 1, 0.807692307692 ); ( 1, 1.0 | airfoil | 1, 0.846153846154 ); ( 1, 1.0 | airfoil | 1, 0.884
615384615 ); ( 1, 1.0 | airfoil | 1, 0.923076923077 ); ( 1, 1.0 | airfoil | 1, 0.961538461538 )
```

Hicks-Henne Format: (1, Scale | Mark. List | Lower(0)/Upper(1) side, x_Loc)

Shape Design of a Rotating Airfoil

DEMO