Xoptfoil-JX is a modified version of Xoptfoil version 1.11.1 – the airfoil optimizer by Daniel Prosser

Xoptfoil-JX Reference

Jochen Guenzel, Matthias Boese February 2020

- see ‘Xoptfoil-JX Description’ for more detailed information.

Aerodynamic target values

With the new optimization types ‘target-drag’ and ‘target-moment‘ one can define a certain ‘target_value‘ for an operating point. With this the Xoptfoil optimization tries to get to the value of the design variable as close as possible to the defined target value. As with the normal optimization types the gravitational force towards the target value is controlled by the weighting of the operating point.

`optimization_type(n)`	Additional types ‘target-drag’ and ‘target-moment’
`target_value(n)`	Numerical value which should be achieved. The special value ‘-1’ takes the value of the seed airfoil as the target which leads to a “try not to change this operating point”.

Example:

```
op_mode(7) = 'spec-cl'
op_point(7) = 0.8
optimization_type(7) = 'target-drag'
target_value(7) = 0.0142
reynolds(7) = 146E+03
weighting(7) = 0.7
```

Geometric target values

If there is the need that the final airfoil should have a certain thickness it is more natural for the optimization process to define the thickness as an optimization target instead being a constraint.

Another use case is to have control over a sometimes bad deformation of the surface on the upper or lower side of the airfoil especially towards the trailing edge.

`&geometric_targets`	New namelist
`ngeotargets`	Number of geometric targets that shall be applied
`target_type(n)`	Type of geometric target – either 'zBot' z-value at the bottom side ‘zTop’ z-value on the top side ‘Thickness’ final thickness of the airfoil
`target_geo(n)`	Numerical value which should be achieved
`x_Pos(n)`	X-Position along the chord for this geometric target. Only needed for ‘zBot’ and ‘zTop’
`weighting_geo(n)`	Weighing of this target within the objective function. Same meaning as ‘weighting(N)’ for an operating point

Example

```
&geometric_targets
   ngeotargets = 1
   x_Pos(1) = 0.7
   target_type(1) = 'zBot'
```

```
target_geo(1) = 0.012
weighting_geo(1) = 0.5
```

New shape_functions ‘camb-thick’

Beside the shape_functions type 'naca' and 'hicks-henne' there is a new type implemented called 'camb-thick'. In this case the seed airfoil will be modified during optimization by the 4 airfoil paramters

- thickness
- x-location of max. thickness
- camber
- x-location of max. camber

The airfoil modification is done by the two xfoil routines THKCAM and HIPNT

`&optimization options`	high-level optimization and parameterization parameters are
`shape_functions`	= ‘naca' = ‘hicks-henne' = ‘camb-thick' **new**

Additional infos during optimization

Additional information about the status of the optimization process can be activated

`&optimization options`	high-level optimization and parameterization parameters are
`show_details`	= .true. show more information (default) = .false do not show

Airfoils surface quality and smoothing

Activate a simple smoothing algorithm to get rid off micro waves (spikes) in the surface of the (seed) airfoil.

`&smoothing_options`	New namelist
`do_smoothing`	= .true. - activate smoothing of the surface
`show_smoothing`	= .true. - show additional informations during the optimization process
`highlow_treshold`	Numerical value to detect highs and lows of the 2nd derivative of the surface curve. The number of high and lows is taken into account to assess the quality of the surface. A good value to start is 0.05
`spike_threshold`	Numerical value to detect highs and lows of the 3nd derivative called spikes. The number of spikes is taken into account to assess the quality of the surface. A good value to start is 2
`weighting_smoothing`	Out of the number of highs, lows and spikes a “perturbation value” is calculated which can be an indication for the surface quality: ‘0’ is excellent, a value greater ‘1’ indicate some problems… If this value should be part of the objective function ‘weighting_smoothing’ should have a value > 0.0.

Example

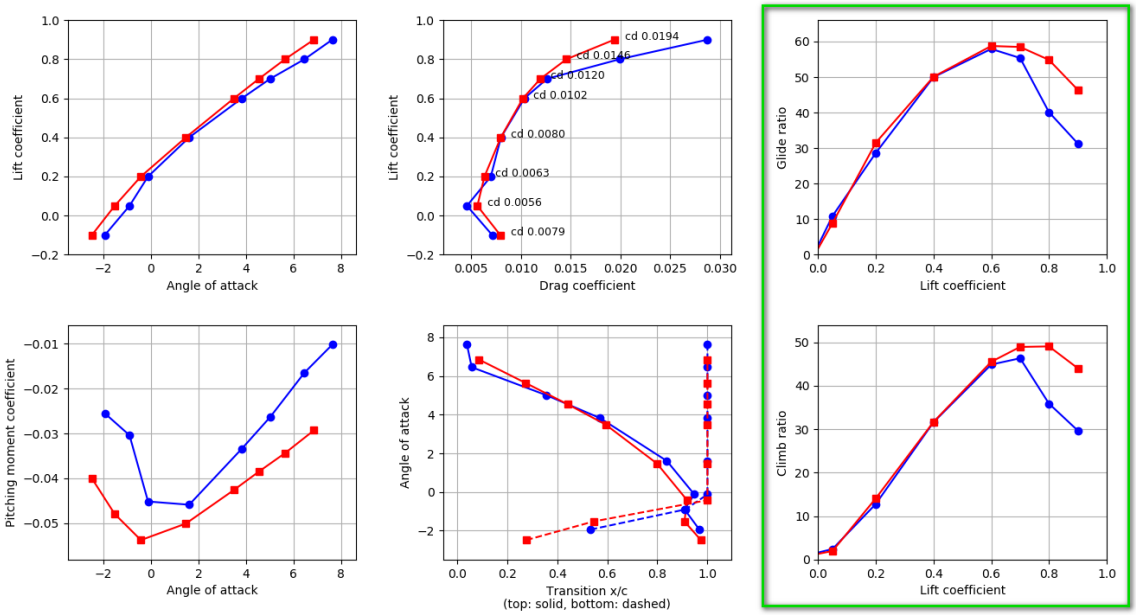
```
&smoothing_options

  do_smoothing     = .true.
  show_smoothing   = .true.
  highlow_treshold = 0.04
  spike_threshold  = 1.5
  weighting_smoothing = 0.1
```

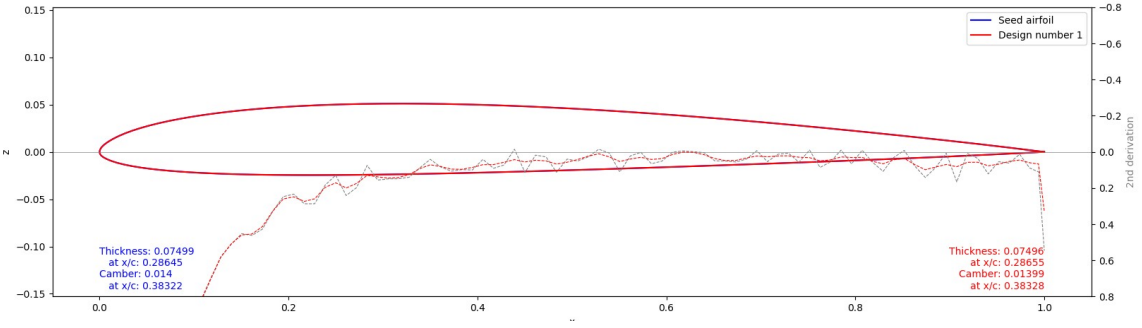
xoptfoil_visualizer-jx.py - Modifications of the Visualizer -

To get an earlier indication whether the optimized airfoil will have the desired aerodynamic properties the polar window of the visualizer is complemented by two sub views for glide ratio and climb ratio.

The layout is optimized for a monitor resolution of 1920 x 1080.



To visualize the smoothing process and to get information about the surface quality additional information can be displayed in the airfoil window:



- 2nd derivation of the airfoil surface
- 3rd derivation of the airfoil surface
- z-values applied by xoptfoil (sum of all shape functions)

This is done by setting these variables in the Python script

```
490   plot_2nd_deriv = True
491   plot_3rd_deriv = False
492   plot_delta_y   = True
```

Have fun!