IPW2 - Tecplot Post-Process Tools

Polytechnique Montréal

January 9, 2023

Table of content

Objectives

Important points

- Tecplot tools (*cutTool_ipw.py*, *iceThicknessTool_ipw.py* & *mccsTool_ipw.py*) will be sent to participants;
- Participants must install PyTecplot (link);
- Take as input a file of any type (*.dat*, *.cgns*, *.txt*, ...);
- For some tools, the participant must indicate the indices of the columns in which the data are located;
- Participants will send a *.dat* type file to the post-processing team.

To define

Possible requirements

- Automated computation of the curvilinear distance?;
- Position of the required cuts along the span;
- Normalization by a reference length of the distance variables;
- The highlight position (ex: LE (x min) of the slice at $AoA = 0$ deg);
- The rotation of the clean mesh at $AoA = 0$ deg;
- The min/max of the curvilinear distance (must be the same for all participants);
- For data comparison, participants must provide the type of solver for each module (flow, droplet, thermo, geo) from pre-defined categories as auxiliary data in *.dat* files;
- A *.sh* file to run the PyTecplot tools.

Tool 1: *cutTool_ipw.py*

Important points

- Extract data for predefined slices from the 3D file;
- Compute the curvilinear distance from a reference point;
- *CoordinateX*, *CoordinateY*, *CoordinateX* must be used to define the clean coordinates;
- *CoordinateX_Iced*, *CoordinateY_Iced*, *CoordinateX_Iced* can be used to define the ice coordinates (if desired);
- Input can be used to rotate and scale the clean geometry;
- Data must be stored at nodes;
- Variables names must be "CollectionEfficiency", "Cp", "HTC", "FreezingFraction";
- For glaze cases, the horns parameters are computed and stored as aux data if the ice coordinates are given by the user;
- Variable name/index could be defined by the user (to do).

Tool 1: Inputs

Inputs

- –**axis**: Axis direction to perform the slice(s) (x, y, z);
- –**pos**: Position along the axis to perform the slice(s);
- –**normal**: Normal of the plane cut $< nx, ny, nz >$;
- –**nslice**: Number of slices to be performed using extremums as star and end position;
- –**tecplotFiles**: Tecplot data files;
- –**cleanGrid**: Tecplot data file containing clean grid;
- –**output**: Output file;
- –**chord**: Chord for lift adim if you want to force it (don't define if you want it to be computed automatically);
- –**angle**: Rotation angle to apply to geometry before computing the ice data;
- –**scale**: Scale to apply to coordinates when writing output;
- –**highlight**: High-light point to project on surface $< x, y, z >$.

Example of command

python cutTool_ipw.py -axis y -normal 0.5 0.8 0.0 -pos 0.7 -scale 0.0254 -tecplotFiles 'solSurf.cgns' -output'cut.dat'

Tool 2: *iceThicknessTool_ipw.py*

Important points

- Compute the ice height along the curvilinear distance for a 2D slice;
- Can use the previous ice layer or the clean geometry as the reference;
- Variable name/index is defined by the user;
- For glaze cases, the horns parameters are computed and stored as aux data if the ice coordinates are given by the user.

Tool 2: Inputs

Inputs

- –**iceCut**: Tecplot data file containing ice shape. Mandatory;
- –**cleanCut**: Tecplot data file containing clean grid. Optional or can be the same file as ice cut if all variables are included in the file;
- –**output**: Output file name. Mandatory;
- –**xIce**: Index or Name of variable containing ice X-coordinates. Mandatory;
- –**yIce**: Index or Name of variable containing ice X-coordinates. Mandatory;
- –**xClean**: Zero-base Index or Name of variable containing clean X-coordinates. Mandatory;
- –**yClean**: Zero-base Index or Name of variable containing clean Y-coordinates. Mandatory;
- –**sClean**: Zero-base Index or Name of variable containing clean wrap distance. Optional;
- –**highlight**: High-light point to project on surface $< x, y, z >$.

Example of command

python iceThicknessTool_ipw.py -iceCut 'solSurf.cgns' -output 'iceThickness.dat' -xIce 51 -yIce CoordinateZ_Iced -xClean CoordinateX -yClean 2

Tool 2: Example of output file

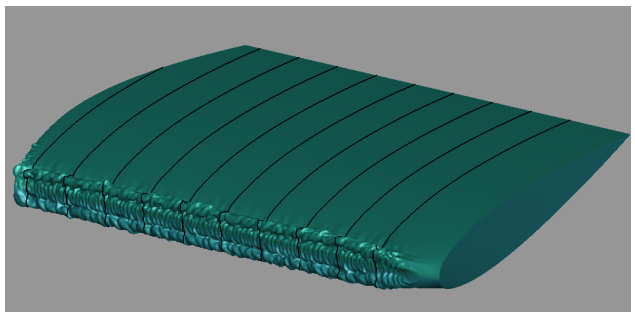
```
 1 TITLE = "Ice thickness"
 2 VARIABLES = "X2D_iced" "Y2D_iced" "sClean" "hIce"
 3 ZONE T="step0_WALL"
 4 AUXDATA HIGHLIGHT_X  = "-0.001281703976"
 5 AUXDATA HIGHLIGHT_Y  = "0.006041527578"
 6 AUXDATA HIGHLIGHT_Z  = "0.000000000000"
 7 AUXDATA LOWER_HORN_HEIGHT  = "0.001061662729"
 8 AUXDATA LOWER_HORN_ANGLE  = "218.945692335329"
 9 AUXDATA UPPER_HORN_HEIGHT  = "0.001054506849"
10 AUXDATA UPPER_HORN_ANGLE  = "173.750800953265"
11 AUXDATA INTEGRATED_ICE_MASS  = "0.000000000000"
12 0.455864568074 -0.012011323656 -0.464439026456 0.000000000000
13 0.455861475055 -0.012099896128 -0.464350399995 0.000000000000
14 0.455858382037 -0.012188468601 -0.464261773533 0.000000000000
15 0.455855289018 -0.012277041074 -0.464173147072 0.000000000000
16 0.455852195999 -0.012365613546 -0.464084520610 0.000000000000
```

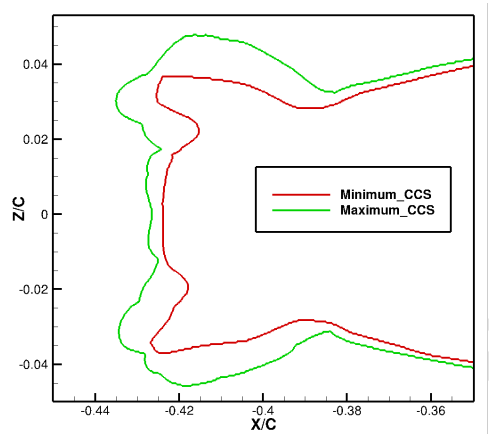
First lines of *.dat* output file for the ice cut with aux data.

Tool 3: *mccsTool_ipw.py*

Info

- The maximum and minimum combined cross-sections are defined;
- A box needs to be specified by the user around the geometry to compute the MCCS;
- The positions of the slices are specified by the user as well as the axis on which to perform the slices;
- The resolution of the box characterizes the resolution of the MCCS;
- A sweep angle can be specified, $x_{min_{clean}}$ at the LE should be zero;





Tool 3: Inputs

Inputs

- –**MCCSAxis**: Axis direction to perform the MCCS slices (x, y, z);
- –**MCCSExtremums**: Extremums along the axis to perform the MCCS slices;
- –**tecplotFile**: Tecplot data file (.dat, .plt or .cgns);
- –**MCCSOutputFile**: MCCS output file name;
- –**nslice**: Number of slices to be performed using extremums as star and end position;
- –**verticalAxis**: vertical axis (x, y, z);
- –**sweepAngle**: Rotation angle to apply to geometry before computing the ice data (see impact on variables);
- –**nx**: Number of elements along the X axis for the rectangular slices;
- –**ny**: Number of elements along the Y axis for the rectangular slices;
- –**nz**: Number of elements along the Z axis for the rectangular slices;
- –**rectangularZoneXLimits**: X coordinate limits for the rectangular slices (xMin, xMax);
- –**rectangularZoneYLimits**: Y coordinate limits for the rectangular slices (yMin, yMax);
- –**rectangularZoneZLimits**: Z coordinate limits for the rectangular slices (zMin, zMax).

Example of command

```
python mccsTool_ipw.py -MCCSAxis y -MCCSExtremums 0.0 1.0 -tecplotFile 'surface.plt' -MCCSOutputFile 'MCCS.dat' -nslice 10 -sweepAngle 30.0 -rectangularZoneXLimits -0.5 -0.35 -rectangularZoneYLimits 0.1 0.9 -rectangularZoneZLimits -0.2 0.2
```