CFD Automation in ANSYS Fluent for a 2D Airfoil Simulation (with .jou and .py)

Description: An automated workflow for ANSYS Fluent simulation for a 2D airfoil, using built in journal tool (.jou) and supplementary Python script.

Keywords: *CFD, AoA, ANSYS, Python, Journal*

Brief Overview

This is an automated workflow for ANSYS Fluent, where you will be able to run a simulation for a user-specific Angle of Attack on a 2D airfoil with just a click of a button. Okay that was an exaggeration, you still need to manually input some values (for the AoA setup) – but you get it, right?

Better to do that than painstakingly find those specific settings in the Fluent UI every time you want to change the angle of attack!

Before You Start and Workflow Brief

Given you have setup the model and meshing, and all the components are updated and ready for simulation – I am assuming you are in the ANSYS Workbench, ready to start editing the 'Setup' of your project. Please note, this tutorial assumes ANSYS and CFD literacy on the readers part and therefore, will not cover critical components of a CFD simulation. You are already familiar with the simulation protocol, alongside key concepts in CFD like boundary conditions and velocity components. You have run your own simulations multiple time and now want to 'automate' some of your mundane tasks so that you can go 'touch some grass' while the machine takes care of those repetitive steps.

That being said, the following is a brief of the workflow we are entertaining –

- Open the journal file(s) and the Python script in your preferred IDE (I used PyCharm)
- Calculate the critical settings values (using velocity and AoA as inputs) using a Python script
- Change those critical values in the journal file(s) and hit Ctrl+S
- Open ANSYS Fluent Setup and 'Read' the journal file
- Touch grass and come back to a complete simulation with some post-processing data (velocity and pressure contours, wall shear stress)

Take screenshots, log the results and you are ready for the next iteration.

Fun!

Please note, I intend to automate the part of logging results in a future update so that we can achieve parametric sweeps in batches with results logged in. But my assistant is on vacation and I am stretched in time since I have to make my own coffee at this moment.

Bummer.

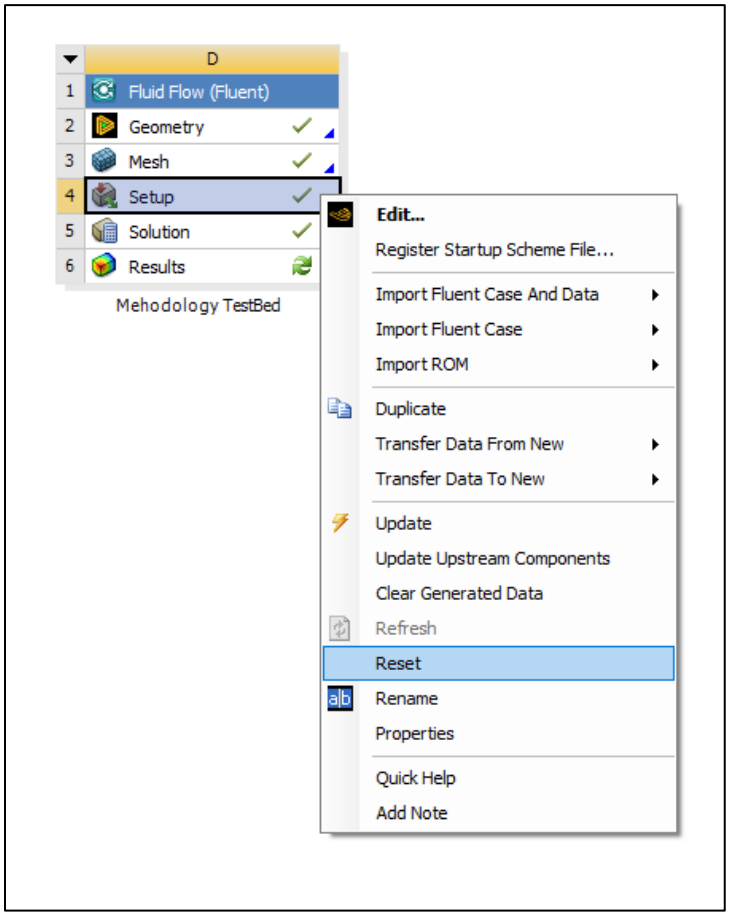
(I don't have any assistant)

Anyways, let's dive deep into the workflow, step by step.

Workflow Step by Step

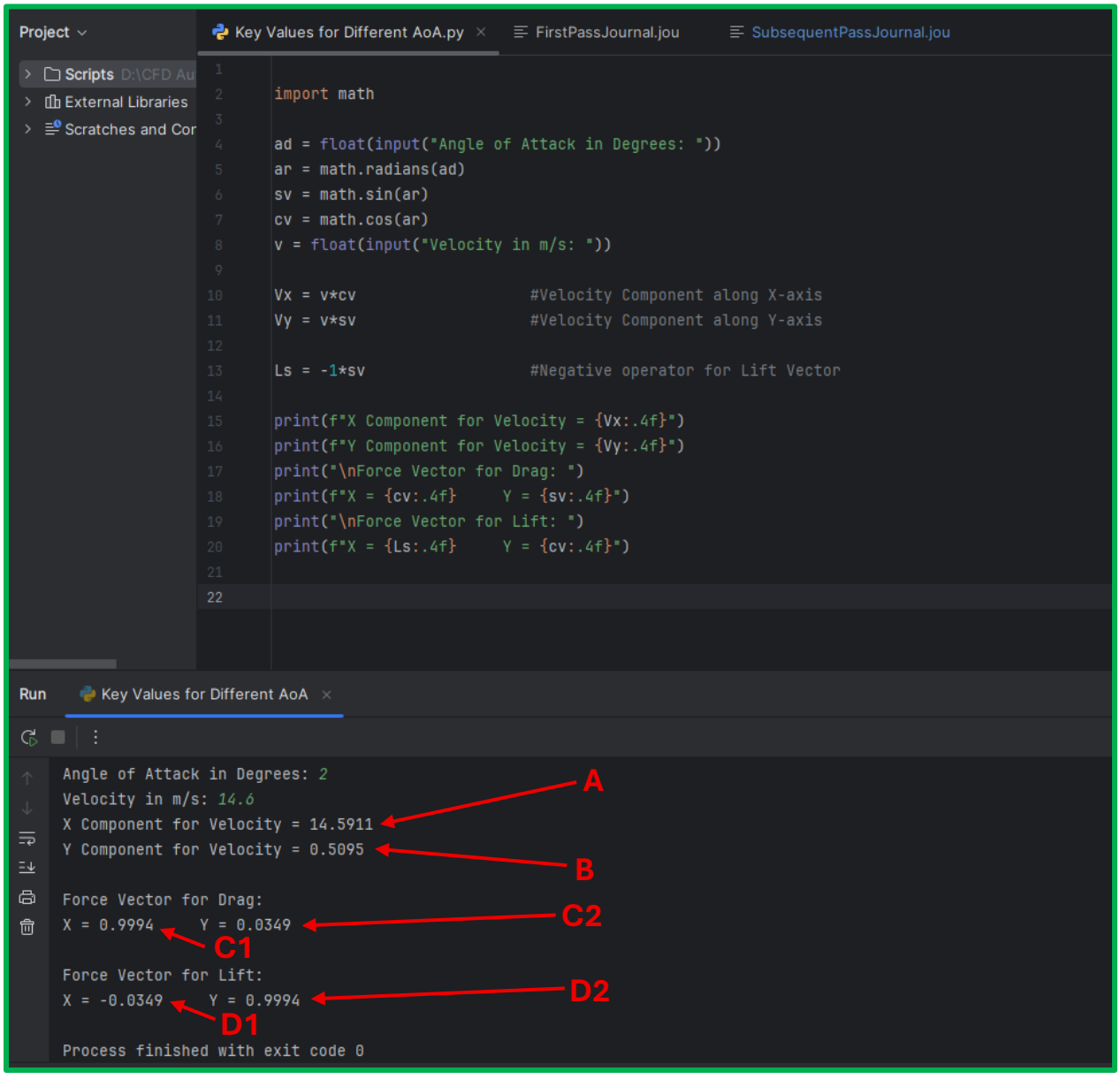
Step 0

If this is your first run, this is the first step for you. Otherwise, you can skip this. Before starting to edit the 'Setup', it is a good practice to reset the protocol. If this is the first time you started your project, you can skip it too. But for good measure, let us stick to it and reset the setup. Then, right click on setup and click edit. Use the appropriate (whatever was convenient to you in your previous studies) solver settings and let Fluent load.



Step 1

While Fluent is loading (for first pass), open the .jou and .py files on the IDE. Please note, there are two .jou file, and use whatever your case is. The name of the files are pretty self-explanatory. Run the Python script and input your desired flow values (velocity and Angle of Attack). You will get 6 critical values. Now, you need to modify the journal file accordingly. For this purpose, let's take 14.6 m/s (Re = 1,000,000 for L = 1m) and 2 degree for our first pass. Please see the following screenshot.



Now, change the following values in the .jou file accordingly (first pass).

```
/file/set-tui-version "25.1"
(cx-gui-do cx-set-list-tree-selections "NavigationPane*Frame2*Table1*List_Tree2" (list "Setup|Models|Viscous (SST k-omega)"))
(cx-gui-do cx-list-tree-right-click "NavigationPane*Frame2*Table1*List_Tree2" )
(cx-gui-do cx-activate-item "MenuBar*PopupMenuTree-Viscous (SST k-omega)*Edit...")
(cx-gui-do cx-set-toggle-button2 "Viscous Model*Table1*ToggleBox1(Model)*Spalart-Allmaras (1 eqn)" #t)
(cx-gui-do cx-activate-item "Viscous Model*Table1*ToggleBox1(Model)*Spalart-Allmaras (1 eqn)")
(cx-gui-do cx-activate-item "Viscous Model*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*Frame2*Table1*List_Tree2" (list "Setup|Materials|Fluid|air"))
(cx-gui-do cx-list-tree-right-click "NavigationPane*Frame2*Table1*List_Tree2" )
(cx-gui-do cx-activate-item "MenuBar*PopupMenuTree-air*Edit...")
(cx-gui-do cx-activate-item "Create/Edit Materials*PanelButtons*PushButton3(Change/Create)")
(cx-gui-do cx-activate-item "Create/Edit Materials*PanelButtons*PushButton1(Close)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*Frame2*Table1*List_Tree2" (list "Setup|Boundary Conditions|Inlet|inlet (velocity-inlet, id=7)"))
(cx-gui-do cx-list-tree-right-click "NavigationPane*Frame2*Table1*List_Tree2" )
(cx-gui-do cx-activate-item "MenuBar*PopupMenuTree-inlet (velocity-inlet, id=7)*Edit...")
(cx-gui-do cx-set-list-selections "Velocity Inlet*Frame2*Frame2*Frame1(Momentum)*Table1*DropDownList7(Velocity Specification Method)" '( 1))
(cx-gui-do cx-enable-apply-button "Velocity Inlet")
(cx-gui-do cx-activate-item "Velocity Inlet*Frame2*Frame2*Frame1(Momentum)*Table1*DropDownList7(Velocity Specification Method)")
(cx-gui-do cx-set-expression-entry "Velocity Inlet*Frame2*Frame2*Frame1(Momentum)*Table1*Table17*ExpressionEntry1(X-Velocity)" '("14.5911" . 0))
(cx-gui-do cx-set-expression-entry "Velocity Inlet*Frame2*Frame2*Frame1(Momentum)*Table1*Table18*ExpressionEntry1(Y-Velocity)" '("0.5095" . 0))
(cx-gui-do cx-activate-item "Velocity Inlet*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-activate-item "Velocity Inlet*PanelButtons*PushButton2(Cancel)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*Frame2*Table1*List_Tree2" (list "Setup|Reference Values"))
(cx-gui-do cx-set-list-tree-selections "NavigationPane*Frame2*Table1*List_Tree2" (list "Setup|Reference Values"))
```

A

B

Key Values for Different AoA.py | FirstPassJournal.jou | SubsequentPassJournal.jou

```
(cx-gui-do cx-activate-item "Residual Monitors*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Report Definitions"))
(cx-gui-do cx-list-tree-right-click "NavigationPane*Frame2*Table1*List_Tree2" )
(cx-gui-do cx-activate-item "MenuBar*Force ReportSubMenu*Drag...")
(cx-gui-do cx-set-text-entry "Drag Report Definition*Table1*TextEntry3(Name)" "cd")
(cx-gui-do cx-activate-item "Drag Report Definition*Table1*TextEntry3(Name)")
(cx-gui-do cx-set-real-entry-list "Drag Report Definition*Table1*Table1*Table2*Table2(Force Vector)*RealEntry1(X)" '( 0.9994))
(cx-gui-do cx-set-real-entry-list "Drag Report Definition*Table1*Table1*Table2*Table2(Force Vector)*RealEntry2(Y)" '( 0.0349))
(cx-gui-do cx-set-toggle-button2 "Drag Report Definition*Table1*Table1*Table5(Create)*CheckButton4(Print to Console)" #t)
(cx-gui-do cx-activate-item "Drag Report Definition*Table1*Table1*Table5(Create)*CheckButton4(Print to Console)")
(cx-gui-do cx-set-list-selections "Drag Report Definition*Table1*Table2*List2(Zones)" '( 0))
(cx-gui-do cx-activate-item "Drag Report Definition*Table1*Table2*List2(Zones)")
(cx-gui-do cx-activate-item "Drag Report Definition*PanelButtons*PushButton1(OK)")
(cx-gui-do cx-set-list-tree-selections "NavigationPane*Frame2*Table1*List_Tree2" (list "Solution|Report Definitions"))
(cx-gui-do cx-list-tree-right-click "NavigationPane*Frame2*Table1*List_Tree2" )
(cx-gui-do cx-activate-item "MenuBar*Force ReportSubMenu*Lift...")
(cx-gui-do cx-set-text-entry "Lift Report Definition*Table1*TextEntry3(Name)" "cl")
(cx-gui-do cx-activate-item "Lift Report Definition*Table1*TextEntry3(Name)")
(cx-gui-do cx-set-real-entry-list "Lift Report Definition*Table1*Table1*Table2*Table2(Force Vector)*RealEntry1(X)" '( -0.0349))
(cx-gui-do cx-set-real-entry-list "Lift Report Definition*Table1*Table1*Table2*Table2(Force Vector)*RealEntry2(Y)" '( 0.9994))
(cx-gui-do cx-set-toggle-button2 "Lift Report Definition*Table1*Table1*Table5(Create)*CheckButton4(Print to Console)" #t)
(cx-gui-do cx-activate-item "Lift Report Definition*Table1*Table1*Table5(Create)*CheckButton4(Print to Console)")
(cx-gui-do cx-set-list-selections "Lift Report Definition*Table1*Table2*List2(Zones)" '( 0))
(cx-gui-do cx-activate-item "Lift Report Definition*Table1*Table2*List2(Zones)")
```

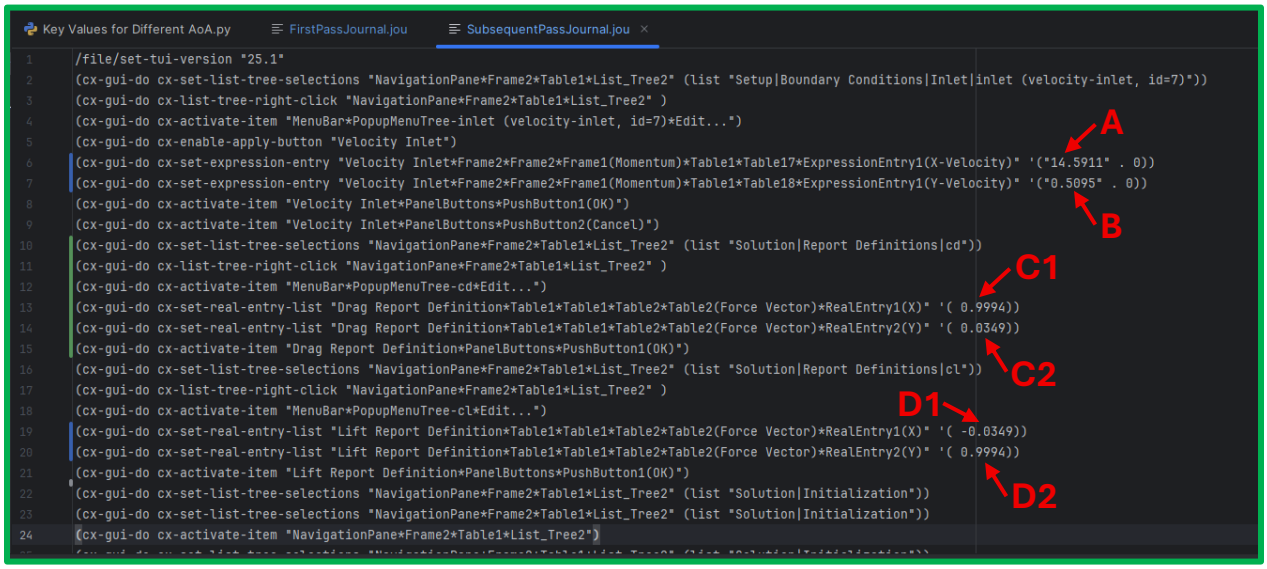
C1

C2

D1

D2

And for subsequent passes, consult the screenshot below.



Save the .jou files. Now, at this point, ANSYS Fluent will be loaded completely. This is a good time to make sure your journal files are at the right path. Usually, the path you want your .jou files to be available looks something like this –
NACA 0012\NACA0012 Project Files\NACA0012_files\dp0\FFF-5\Fluent
Before doing anything else, it is very important to put your files in this path. Copy those .jou files in this directory and open them from that directory for future reference.

Step 2

Once your files are on that specific directory (it might change slightly, say FFF-2 in stead of FFF-5), open them in ANSYS Fluent. To do that, click File on top left corner of the Fluent window, go to Read, and select Journal...
By default, ANSYS Fluent will open a specific directory (that is the default directory for your project file, where you should be pasting the .jou files) in the pop-up menu. Select the .jou file (based on first or subsequent runs) and go grab a cup of coffee!

> ***Pro Tip*** *– After the runs (and putting the files in that specific directory for the first time), it is a good practice to properly close Fluent (File – Close Fluent), go to workbench and save the project. Otherwise, the .jou files might be removed for future studies.*

Omar Saif
CFD Enthusiast
October 15, 2025