TSFOIL Analysis Interface Module (AIM)

Ryan Durscher
AFRL/RQVC

September 17, 2025

0.1 Introduction

0.1.1 TSFOIL AIM Overview

A module in the Computational Aircraft Prototype Syntheses (CAPS) has been developed to interact (through input files) with the transonic airfoil analysis tool TSFOIL. TSFOIL can be downloaded from http://www.dept.aoe.vt.edu/~mason/Mason_f/MRsoft.html .

Note: In the tsfoil2.f file is may be necessary to comment out line 38 - "USE DFPORT"

An outline of the AIM's inputs and outputs are provided in AIM Inputs and AIM Outputs, respectively.

Upon running preAnalysis the AIM generates two files: 1. "tsfoilInput.txt" which contains instructions for TSFOIL to execute and 2. "caps.tsfoil" which contains the geometry to be analyzed. The TSFOIL AIM can automatically execute TSFOIL, with details provided in AIM Execution.

0.1.2 Assumptions

TSFOIL inherently assumes the airfoil cross-section is in the x-y plane, if it isn't an attempt is made to automatically rotate the provided body.

0.1.2.1 Airfoils in ESP

Within **OpenCSM** there are a number of airfoil generation UDPs (User Defined Primitives). These include NACA 4 series, a more general NACA 4/5/6 series generator, Sobieczky's PARSEC parameterization and Kulfan's CST parameterization. All of these UDPs generate **EGADS** *FaceBodies* where the *Face*'s underlying *Surface* is planar and the bounds of the *Face* is a closed set of *Edges* whose underlying *Curves* contain the airfoil shape.

Important Airfoil Geometry Assumptions

- There must be a *Node* that represents the *Leading Edge* point
- For a sharp trailing edge, there must be a *Nodes* at the *Trailing Edge*
- For a blunt trailing edge, the airfoil curve may be open, or closed by a single *Edge* connecting the upper/lower *Nodes*
- For a *FaceBody*, the airfoil coordinates traverse counter-clockwise around the *Face* normal. The **OpenCSM** *REORDER* operation may be used to flip the *Face* normal.
- For a *WireBody*, the airfoil coordinates traverse in the order of the loop

Note: Additional spurious *Nodes* on the upper and lower *Edges* of the airfoil are acceptable.

0.2 AIM Inputs

The following list outlines the TSFOIL inputs along with their default values available through the AIM interface.

- **Mach = 0.75**
  Mach number. Valid range for TSFOIL is 0.5 to 2.0 .
- **Re = 0.0**
  Reynolds number based on chord length.
- **Alpha = 0.0**
  Angle of attack [degree].

0.3 AIM Execution

If auto execution is enabled when creating an TSFOIL AIM, the AIM will execute TSFOIL just-in-time with the command line:

```
tsfoil2 < tsfoilInput.txt > Info.out
```

where preAnalysis generated the file "frictionInput.txt" which contains the input information.

The analysis can be also be explicitly executed with caps_execute in the C-API or via Analysis.runAnalysis in the pyCAPS API.

Calling preAnalysis and postAnalysis is NOT allowed when auto execution is enabled.

Auto execution can also be disabled when creating an TSFOIL AIM object. In this mode, caps_execute and Analysis.runAnalysis can be used to run the analysis, or TSFOIL can be executed by calling preAnalysis, system call, and posAnalysis as demonstrated below with a pyCAPS example:

```
print ("\n\preAnalysis......")
tsfoil.preAnalysis()

print ("\n\nRunning......")
tsfoil.system("tsfoil2 < tsfoilInput.txt > Info.out"); # Run via system call

print ("\n\postAnalysis......")
tsfoil.postAnalysis()
```

0.4 AIM Outputs

The following list outlines the TSFOIL outputs available through the AIM interface.

- **CL =** Coefficient of lift value.
- **CD =** Coefficient of drag value. (Calculated from momentum integral)
- **CD_Wave =** Wave drag coefficient value.
- **CM =** Moment coefficient value.
- **Cp_Critical =** Critical pressure coefficient (M = 1).

Index