CAPS Friction AIM Documentation

April 4, 2016

1 Goals

FRICTION provides an estimate of laminar and turbulent skin friction and form drag suitable for use in aircraft preliminary design. The Computational Aircraft Prototype Syntheses (CAPS) Analysis Interface & Meshing (AIM) plugin described in this document allows the generation of FRICTION input from Engineering Sketch Pad (ESP) geometry.

2 Inputs

FRICTION requires inputs that describe the geometry and the flow conditions where the the drag is evaluated. The geometry description is in units of *ft*. Unit conversion is handled inside the AIM, thus the ESP input is not required to be in *ft*.

The flow conditions are Mach, Altitude pairs. The altitude units are in *kft*. If altitude is entered in another unit conversion will be automatically handled inside the AIM.

3 CAPS Values

The inputs to the FRICTION AIM are variable length arrays. There are two input values *Mach* and *Altitude*. The units for *Altitude* are *kft*. Different units may be specified and the AIM will handle the unit conversion automatically. The length of the *Mach* and *Altitude* inputs must be the same.

The outputs from the FRICTION AIM are variable length arrays that are sized based on the number of inputs requested. The outputs are *CDtotal*, *CDform* and *CDfric*. Where $CDtotal = CDform + CDfric$.

Table 1: CAPS Values

Inputs	**Outputs**
Mach	CDtotal
Altitude	CDform
BoundaryLayerTransistionPercent	CDfric

The *BoundaryLayerTransistionPercent* Input is a value between 0 and 1. Zero implies a fully turbulent boundary layer, 1 is fully laminar. This value is the same for every section input into friction.

4 Attribution

The length units that the ESP input file is defined as must be defined. These can be set at the beginning of the file using the following command.

attribute capsLength $m

The fidelity of the geometry must also be defined. In the case of FRICTION the appropriate fidelity is very low. Only cross sections must be generated. Options to rule, blend, loft, union, intersect, subtract, etc. are not required. When a section is defined it must be attributed with the fidelity information.

attribute capsFidelity 13

Table 2: capsGroup attributes

Lifting Surfaces	**Body of Revolution**
Wing	Fuselage
Tail	Fuse
HTail	Store
VTail	
Horizontal_Tail	
Vertical_Tail	
Cannard	

Aircraft components are defined as cross sections in the low fidelity geometry definition. To be able to logically group the cross sections into wings, tails, fuselage, etc they must be given a grouping attribute. This attribute defines a logical group along with identifying a set of cross sections as a lifting surface or a body of revolution. The format is as follows.

attribute capsGroup $Wing

The options for group attributes have been defined within the AIM and are given in Table 2.

Finally the reference area used in the drag coefficient calculation can be assigned the geometry via attribution. To do this any of the cross sections should contain the attribute *capsReferenceArea*. If this is not done a value of 1.0 will be used for the reference area. The attribute is as follows.

attribute capsReferenceArea RealValue

5 Limitations

Currently this AIM only works for Fidelity level 13 geometry.

6 Source Code

frictionAIM.c **and its associated** *Makefile*

frictionTest.c **and its associated make file** *frictionTest.make*

frictionAIM.csm **an example ESP input to test the AIM**

friction_eja_mod.f **FRICTION source code**

6.1 Modifications to FRICTION

FRICTION has been modified to allow longer input and output file name lengths. In addition the IO for has been modified. The AIM assumes that FRICTION can be called using the following syntax.

»friction inputfilename.txt outputfilename.txt

To install friction use the following command and ensure that the location is in your *$PATH*.

»gfortran -o friction friction_eja_mod.f

7 Running the AIM

To execute the example problem create a scratch directory *fric*. Use the command bellow to execute the example.

»frictionTest frictionAIM.csm fric