CAPS AWAVE AIM Documentation

February 10, 2016

1 Goals

AWAVE provides an estimation for wave drag at supersonic Mach numbers at various angles of attack. Inside AWAVE all configurations are assumed to be symmetric with respect to the X-Z plane. Only the +Y axis portion of a given model is used to generate the AWAVE input. This AIM automatically finds the proper portions of the model to create the input. However, it assumes that the model is oriented with the X-axis as the flow direction and the Y-axis out the right side wing from the pilots perspective.

2 Inputs

AWAVE is unit less. The geometry description must be input in consistent units.

3 CAPS Values

The inputs to the AWAVE AIM are variable length arrays. The flow conditions are Mach, Angle of Attack (AoA) pairs. The AoA units are in *degrees*. There are two input values required to execute the AWAVE AIM, *Mach* and *AoA*. The length of the *Mach* and *AoA* inputs must be the same.

The outputs from the AWAVE AIM are variable length arrays that are sized based on the number of inputs requested. In some cases all of the requested inputs will not produce a result. In this case no output will be produced. For this reason the input values are also produced as output CAPS values. These will match the wave drag estimations produced. The outputs are *CDwave*, *MachOut* and *AoAOut*.

Table 1: CAPS Values

Inputs	**Outputs**
Mach	CDwave
AoA	AoAOut
	MachOut

4 Attribution

The fidelity of the geometry must also be defined. In the case of AWAVE 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	
Cannard	
Fin	

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 capsGroup attributes have been defined within the AIM and are given in Table 2.

In addition to the group attribute a unique name can be given to each component within a group. This name only needs to be unique and is defined by the user. This will be used to separate components with the same group attribute. For example if two components with group attribute *Tail* are defined one could be given the name attribute *Tail1* and the other Tail2. The format follows as:

attribute name $UniqueName

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

awaveAIM.c **and its associated** *Makefile*

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

awavemod.f **aware source code**

6.1 Modifications to AWAVE

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

»awave inputfilename.txt outputfilename.txt

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

»gfortran -o friction awavemod.f

7 Running the AIM

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

»awaveTest awaveAIM.csm scratch