Missile DATCOM Analysis Interface Module (AIM)

Marshall Galbraith MIT and Ryan Durscher
AFRL/RQVC

September 17, 2025

0.1 Introduction

0.1.1 Missile Datcom AIM Overview

A module in the Computational Aircraft Prototype Syntheses (CAPS) has been developed to interact with Missile Datcom.

Details on the use of units are outlined in AIM Units.

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

0.1.2 Geometry Requirements and Assumptions

The Missile DATCOM coordinate system assumption (X – downstream, Y – out the right wing, Z – up) needs to be followed.

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.

A *NodeBody* may be used for a sharp fin tip.

Missile DATCOM also supports the the specific HEX, ARC, USER, and NACA fin sets. The fin set and corresponding ESP airfoil section are listed in the table below.

missile DATCOM Airfoil	ESP Airfoil
HEX	misdathex
ARC	biconvex
USER	Any airfoil UDP
NACA	See FinSet SecNACA in AIM Inputs

The misdathex UDP has the same arguments as the missile DATCOM HEX airfoil, i.e. zupper, zlower, lmaxu, lmaxl, lflatu, and lflatl

0.1.3 Notes

- DEXIT is calculated based on any inner loop found on the last FaceBody of the missile's body using the following: $D_{Exit} = \sqrt{A_{Loop} * 4/\pi}$

0.1.4 ToDO

- Add Base-Jet Plume Interaction Inputs
- Add fin controls.

0.1.5 Code Distribution

DISTRIBUTION STATEMENT A. Approved for public release: distribution is unlimited. PA#: AFRL-2024-6115.

0.2 AIM Attributes

The following list of attributes drives the Missile Datcom geometric definition. Attributes which are required and those that are optional (note: Missile Datacom will provided default values inherently for optional arguments) are marked so in the description:

- **capsType** This string attribute labels the *FaceBody* as to which type the section is assigned. Can either be "Body" or "Fin".
- **capsGroup** This string attribute labels the *FaceBody* as to which "Fin" sections need to be grouped together. Note: The capsGroup string for Fins should have a FinSet# prefix to group Fins into FinSets. For example: FinSet1_1, FinSet1_2, FinSet2_1, FinSet2_2
  All Fins within a FinSet are assumed to have the same airfoil sections without checking.
- **capsReferenceArea** [Optional] This attribute may exist on any *Body*. Its value will be used as the SREF entry for reference area.
- **capsReferenceChord** [Optional] This attribute may exist on any *Body*. Its value will be used as the LREF entry for longitudinal reference length.
- **capsReferenceSpan** [Optional] This attribute may exist on any *Body*. Its value will be used as the LATREF entry for lateral reference length.
- **capsReferenceX** [Optional] This attribute may exist on any *Body*. Its value will be used as the XCG entry for longitudinal position of C.G.
- **capsReferenceY** [Optional] This attribute may exist on any *Body*. Its value will be used as the ZCG entry for vertical position of C.G.

0.3 Missile DATCOM FinSet

Structure for the constraint tuple = ("FinSet Name", "Value"). "Constraint Name" defines the reference name for the constraint being specified. The "Value" can either be a JSON String dictionary (see Section JSON String Dictionary) or a single string keyword (see Section keyStringFinSet).

0.3.1 JSON String Dictionary

If "Value" is JSON string dictionary (eg. "Value" = {"Translation": "free", "Rotation":"free"}) the following keywords (= default values) may be used:

- **SecNACA = ""**
  Airfoil section type for the FinSet.
  For NACA, the full NACA control card should be specifed (without the FinSet index), e.g.:
  SecTyp = "NACA-4-0012"
  for a NACA 4-series airfoil. See the Missile Datcom NACA Control Card documentation for more info.

- **phif = <from geometry by default>**
  List of roll angle of each fin measured clockwise from top vertical center looking forward.
  This overrides values otherwise computed from the geometry.
  Note: If both phif and gam are specified they must be of the same length.

- **gam = <from geometry by default>**
  List of roll angle of each fin measured clockwise from top vertical center looking forward This overrides values otherwise computed from the geometry.
  Note: If both phif and gam are specified they must be of the same length.

- **nvor = 1**
  Number of vortices shed from each panel, up to a maximum of 20

0.4 AIM Units

A unit system may be optionally specified during AIM instance initiation. If a unit system is provided, all AIM input values which have associated units must be specified as well. If no unit system is used, AIM inputs, which otherwise would require units, will be assumed unit consistent. Missile Datcom only supports modifying the length unit, which may be set with a JSON string: unitSys = "{"length": "m"}"

- **length = "None"**
  Length units - must be one of "in", "ft", "cm", "m"

0.5 AIM Inputs

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

- **missileDatcom = "misdat"**
  File name of missileDATCOM executable
- **BODoption = int (1 or 2)**
  Generate AXIBOD/ELLBOD Option 1 or Option 2 geometry
- **BODtnose = "OGIVE"**
  Nose shape:
  "CONICAL" or "CONE" (cone)
  "OGIVE" (tangent ogive) "POWER" (power law) "HAACK" (L-V constrained) "KARMAN" (L-D constrained)

- **BODpower = 0**
  Exponent, n, for nose power law shape: $(r/R)=(x/L)^n$

- **BODbnose = 0**
  Nose bluntness radius or radius of truncation

- **BODtrunc = False**
  Truncation flag: True if nose if truncated, False is nose is not truncated

- **BODtaft = "CONICAL"**
  Afterbody shape:
  "CONICAL" or "CONE" (cone)
  "OGIVE" (tangent ogive)

- **FinSet = NULL**
  Tuple used to input FinSet information, see Missile DATCOM FinSet for additional details.

- **Mach = double**
  OR

- **Mach = [double, ... , double]**
  Mach number.

- **Ren = double**
  OR

- **Ren = [double, ... , double]**
  Reynolds numbers per unit length

- **Vinf = double**
  OR

- **Vinf = [double, ... , double]**
  Freestream velocities

- **Tinf = double**
  OR

- **Tinf = [double, ... , double]**
  Freestream static temperatures

- **Pinf = double**
  OR

- **Pinf = [double, ... , double]**
  Freestream static pressures

- **Alpha = double**
  OR

- **Alpha = [double, ... , double]**

  Angle of attack [degree].

- **Beta = 0.0**
  Sideslip angle [degree].

- **Altitude = double**
  OR

- **Altitude = [double, ... , double]**
  Flight altitude.

- **BL_Type = "TURB"**
  Boundary layer type: "TURB" for fully turbulent and "NATURAL" for natural transition

- **Surface_Roughness = None**
  Surface roughness height. Only use one of "Surface_Roughness" (will take precedence) or "Roughness_Rating".

- **Roughness_Rating = None**
  Roughness height rating. Only use one of "Surface_Roughness" (will take precedence) or "Roughness_Rating".

Note: Any of the following combinations satisfy the minimum requirements for calculating atmospheric conditions (Mach and Reynolds number):

1. MACH and REN
2. MACH and ALT
3. MACH and TINF and PINF
4. VINF and ALT
5. VINF and TINF and PINF
6. VINF and TINF and REN

0.6 AIM Execution

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

```
misdat > missileDatcomOutput.txt
```

where preAnalysis generated the file for005.dat.

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 missileDATCOM AIM object. In this mode, caps_execute and Analysis.runAnalysis can be used to run the analysis, or missileDatcom can be executed by calling preAnalysis, system call, and posAnalysis as demonstrated below with a pyCAPS example:

```
print ("\n\npreAnalysis......")
missileDatcom.preAnalysis()

print ("\n\nRunning......")
missileDatcom.system("missileDatcom > missileDatcomOutput.txt"); # Run via system call

print ("\n\npostAnalysis......")
missileDatcom.postAnalysis()
```

0.7 AIM Outputs

The following list outlines the Missile Datcom outputs available through the AIM interface.

Forces and moments:

- **CNtot** = Normal Force Coefficient (body axis).
- **CAtot** = Axial Force Coefficient (body axis).
- **CYtot** = Side Force Coefficient (body axis).
- **Cmtot** = Pitching Moment Coefficient (body axis).
- **Cntot** = Yawing Moment Coefficient (body axis).
- **Cltot** = Rolling Moment Coefficient (body axis).
- **CLtot** = Lift Coefficient (wind axis).
- **CDtot** = Drag Coefficient (wind axis).
- **Xcp** = Center of Pressure in calibers from the moment reference center.

Derivatives - Alpha:

- **CNa** = Normal force coefficient derivative with angle of attack.
- **Cma** = Pitching moment coefficient derivative with angle of attack.

Derivatives - Beta:

- **CYb** = Side force coefficient derivative with sideslip angle.
- **Cnb** = Yawing moment coefficient derivative with sideslip angle (body axis).
- **Clb** = Rolling moment coefficient derivative with sideslip angle (body axis).

Echo inputs for case specific information:

- **Alpha** = Angle of attack from output file.
- **Mach** = Mach number from output file.
- **Altitude** = Altitude from output file.

Index