ZAero Analysis Interface Module (AIM) Manual

Marco De Lannoy Kobayashi AFRL/RQVC and Marshall Galbraith MIT

April 27, 2024

0.1 Introduction

0.1.1 ZAero AIM Overview

A module in the Computational Aircraft Prototype Syntheses (CAPS) has been developed to interact (primarily through input files) with Zona's ZAero. ZAero is designed to multiplie different aeroelastic analysis disciplines. Currently only a subset of ZAero's input options have been exposed in the analysis interface module (AIM), but features can easily be included as future needs arise.

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

Geometric attributes recognized by the AIM are provided in Attribution.

0.2 Attribution

The following list of attributes drives the ZAero geometric definition.

- **capsLength** This attribute defines the length units that the $*$.csm file is generated in. ZAero grids MUST be in units of meter, as such the geometry is scaled accordingly based on this value.
- **capsReferenceArea** [Optional] This attribute may exist on any *Body*. Its value will be used as the reference area in ZAero's input file with its units assumed to be consistent with the attribute "capsLength". No conversion takes place if "capsLength" isn't set. This value may be alternatively set through an input value, "Reference↩ Area" (see AIM Inputs)
- **capsReferenceChord** and **capsReferenceSpan** [Optional] These attribute may exist on any *Body*. Their value will be used as the reference moment lengths in ZAero's input file with their units assumed to be consistent with the attribute "capsLength". No conversion takes place if "capsLength" isn't set. These values may be alternatively set through an input value, "Moment_Length" (see AIM Inputs)
- **capsReferenceX**, **capsReferenceY**, and **capsReferenceZ** [Optional] These attribute may exist on any *Body*. Their value will be used as the center of gravity (CG) location in ZAero's input file with their units assumed to be consistent with the attribute "capsLength". No conversion takes place if "capsLength" isn't set. These values may be alternatively set through an input value, "Moment_Center" (see AIM Inputs)

0.3 ZAero Executive Control Section

The following lists the input for the Executive Control Section of the ZAero input file

0.3.1 FEM JSON String Dictionary

FEM input must be a JSON string dictionary (e.g. {"boundary": "sym", "suport": [1,3], "print": 1} where the following keywords (= default values) may be used:

- **fem = ""**
  Optional Absolute file path to f06 file from a structural analysis. Cannot be specified if F06 is linked.

- **form = ""**
  Optional string describing solver used to generate FEM file. If "fem" is specified then "form" must be one of: 'MSC' generated by MSC.NASTRAN or NX.NASTRAN 'NE' generated by NE/NASTRAN 'ASTROS' generated by ASTROS 'IDEAS' generated by I-DEAS 'ELFINI' generated by ELFINI 'GENESIS' generated by GENESIS 'ABAQUS' generated by ABAQUS 'ALTAIR' generated by ALTAIR's RADIOSS 'FREE ' stored according to the input instruction described in Remark 9 of ZAero manual

- **boundary = "SYM"**
  "boundary" indicates the boundary condition of the structural finite element model.
  'SYM' for symmetric boundary condition
  'ANTI' for anti-symmetric boundary condition
  'ASYM' for asymmetric boundary condition

- **support = [0, 0]**
  "support" up to length 2 array for the m/L ZAero support input. See ZAero manual for details.

- **print = 0**
  Print options to the standard output file; where n is an integer.
  n = 0 \t no printout of the imported structural free vibration solution.
  | n | >= 1 \t print out the structural grid point locations in the aerodynamic coordinate system.
  n >= 2 \t print out the modal data (mode shapes) at the structural grid points in the aerodynamic coordinate system.
  n <= -2 \t print out the interpolated modal data at the control points of the aerodynamic boxes in the aerodynamic coordinate system.
  n = 3 \t print all of the above

0.4 ZAero Unified Aerodynamic Influence Coefficients

Structure for the UAIC tuple = ("UAIC Name", "Value"). "UAIC Name" defines the reference name for the UAIC being specified. The "Value" must be a JSON String dictionary.

0.4.1 UAIC JSON String Dictionary

For the JSON string "Value" dictionary (e.g. "Value" = {"machNumber": 0.5, "method": 120000.0, "poissonRatio": 0.5, "materialType": "isotropic"}) the following keywords (= default values) may be used:

- **machNumber**
  Mach number.

- **method**
  Integer aerodynamic method
  method = 0 for the ZONA6/ZONA7/ZSAP method
  method = 1 for the ZTAIC method
  method = +/- 2 for the ZONA7U method
  method = 3 for the ZTRAN method

- **reducedFreq**
  List of real reduced frequencies

- **print = 0**
  Integer controlling print output

0.5 ZAero Spline Module

Structure for the spline tuple is a tuple of key value pairs.

The following keywords (= default values) may be used:

- **method = 1**
  0 : Imposes zero-displacement condition on aerodynamic boxes. 1 : Defines a surface spline method (Infinite Plate Spline method) for CAERO7. 2 : Defines a beam spline method for CAERO7 / BODY7. 3 : Defines a 3-D spline (Thin Plate Spline method) for CAERO7 / BODY7.

- **attachFlex = 0**
  Linear attachment flexibility (for spline method 1)

- **eps = 1e-6**
  Multiplication factor to obtain a small tolerance to detect any duplicated location of structural grid points. The tolerance is computed by EPS×REFC, where REFC is the reference chord defined in the AEROZ bulk data card (for spline method 1 or 3)

0.6 ZAero Analysis

Structure for the Analysis tuple = ("Case Name", "Value"). "Case Name" defines the reference name for the subcase being specified. The "Value" must be a JSON String dictionary.

0.6.1 Analysis JSON String Dictionary

For the JSON string "Value" dictionary (e.g. "Value" = {"discipline": "LinearFlutter", "uaic": "cruise"}) the following keywords (= default values) may be used:

- **discipline = ""**
  Analysis discipline JSON string. Options :
  LinearFlutter or flutter - Linear flutter
  ParamFlutter or fltpram - Parametric flutter.
  Aeroservoelastic or ase - Asymmetric parametric flutter.
  StaticAeroelastic or trim - Static areoelastic (see StaticAeroelastic or Trim Discipline JSON String Dictionary).
  EjectionLoads or eloads - Transient ejection loads.
  ManeuverLoads or mloads - Transient manouver loads.
  GustLoads or gloads - Discrete gust load.
  MFTGustLoads or mftgust - Continuous gust load.
  NonLinearFlutter or nlfltr - Nonlinear flutter.

- **uaic = ""**
  Name of the UAIC module for the subcase

0.6.2 StaticAeroelastic or Trim Discipline JSON String Dictionary

For the JSON string "Value" dictionary (e.g. "Value" = {"dynamicPressure": rho∗V^2/2, "gravityAcceleration": 9.81 m/s^2}) the following keywords (= default values) may be used:

- **dynamicPressure**
  Dynamic pressure

- **vectorToCG = [0,0,0]**
  Vector to offset the CG

- **gravityAcceleration = NULL**
  Gravity acceleration

- **weight = NULL**
  Weight of the overall aircraft if specified

- **weightMomentOfInertia = NULL**
  Moments of inertia of the full aircraft if specified: [IXX, IXY, IYY, IXZ, IYZ, IZZ]

0.7 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. A unit system may be specified via a JSON string dictionary for example: unitSys = "{"mass": "kg", "length": "m", "time":"seconds", "temperature": "K"}"

0.7.1 JSON String Dictionary

The key arguments of the dictionary are described in the following:

- **mass = "None"**
  Mass units - e.g. "kilogram", "k", "slug", ...

- **length = "None"**
  Length units - e.g. "meter", "m", "inch", "in", "mile", ...

- **time = "None"**
  Time units - e.g. "second", "s", "minute", ...

- **temperature = "None"**
  Temperature units - e.g. "Kelvin", "K", "degC", ...

0.8 AIM Inputs

The following list outlines the Zaero inputs along with their default value available through the AIM interface.

- **Proj_Name = "zaero_CAPS"**
  This corresponds to the project name used for file naming.
- **Analysis = NULL**
  Analysis tuple used to input analysis/case information for the model.
- **File_Format = "Small"**
  Formatting type for the bulk file. Options: "Small", "Large", "Free".
- **FEM_1 = NULL**
  JSON dictionary for first ZAero ASSIGN FEM inputs in Executive Control Section (See ZAero Executive Control Section)
- **F06_1 = NULL**
  Link for F06 file from from a structural analysis AIM for first ZAero ASSIGN FEM. zaeroAIM will attempt to extract / determine as many analysis parameters from the F06 file as possible.
- **FEM_2 = NULL**
  JSON dictionary for first ZAero ASSIGN FEM inputs in Executive Control Section (See ZAero Executive Control Section)
- **F06_2 = NULL**
  Link for F06 file from from a structural analysis AIM for second ZAero ASSIGN FEM. zaeroAIM will attempt to extract / determine as many analysis parameters from the F06 file as possible.
- **CPU = 1**
  Defines the number of processors for parallel computation.
- **Memory = "1600MB"**
  Maximum memory in terms of megabytes that is allocable by ZAERO from the heap space.
- **Smart_Restart = True**
  If True, zaeroAIM will try to detect whether the ZAERO restart capability can be used and configure the ZAERO input to load existing AIC matrices. If False, new AIC matrices are always generated.
- **Echo = "sort"**
  Controls echo (printout) of the Bulk Data Section
- **Output**
  Output tuple used to define analysis/case outputs for plotting, etc.
- **HFG**
  JSON dictionary used to define HFG module data
- **UAIC**
  UAIC tuple used to define UAIC configurations for unsteady aerodynamics data generation, see ZAero Unified Aerodynamic Influence Coefficients for additional details.
- **Spline**
  JSON dictionary used to define SPLINE module data, see ZAero Spline Module for additional details.
- **VLM_Surface = NULL**
  Vortex lattice method tuple input, see Vortex Lattice Surface for additional details.
- **VLM_Control = NULL**
  Vortex lattice method control surface tuple input, see Vortex Lattice Control Surface for additional details.
- **Trim_Variable**
  Trim_Variable tuple used to define Trim variables and/or constraints, see ZAero Trim_Variable for additional details.

- **ReferenceArea = NULL**
  This sets the reference area for used in force and moment calculations. Alternatively, the geometry (body) attribute (see Attribution) "capsReferenceArea" maybe used to specify this variable (note: values set through the AIM input will supersede the attribution value).

- **ReferenceChord = NULL**
  This sets the reference chord for used in force and moment calculations. Alternatively, the geometry (body) attribute (see Attribution) "capsReferenceChord" maybe used to specify this variable (note: values set through the AIM input will supersede the attribution value).

- **ReferenceSpan = NULL**
  This sets the reference span for used in force and moment calculations. Alternatively, the geometry (body) attribute (see Attribution) "capsReferenceSpan" maybe used to specify this variable (note: values set through the AIM input will supersede the attribution value).

- **Moment_Center = [0.0, 0.0, 0.0] (NULL)**
  Array values correspond to the x, y, and z center of gravity (CG) locations [meter]. Alternatively, the geometry (body) attributes (see Attribution) "capsReferenceX", "capsReferenceY", and "capsReferenceZ" may be used to specify the center of gravity, respectively (note: values set through the AIM input will supersede the attribution values).

- **MassPropLink = NULL**
  Mass properties linked from structural analysis for eigen value analysis Must be in units of kg, m, and kg∗m^2 if unitSystem (see AIM Units) is not specified.

0.9 ZAero Trim_Variable

Structure for the Trim_Variable tuple = ("Trim Variable Name", "Value"). "Trim Variable Name" defines the reference name for the Trim being specified. The "Value" must be a JSON String dictionary.

0.9.1 Trim_Variable JSON String Dictionary

For the JSON string "Value" dictionary (e.g. "Value" = {"label": "ALPHA", "value": "free"}) the following keywords (= default values) may be used:

- **label**
  The trim label

- **value**
  Trime value.

0.10 Vortex Lattice Surface

Structure for the Vortex Lattice Surface tuple = ("Name of Surface", "Value"). "Name of surface defines the name of the surface in which the data should be applied. The "Value" can either be a JSON String dictionary (see Section JSON String Dictionary) or a single string keyword string (see Section Single Value String).

0.10.1 JSON String Dictionary

If "Value" is a JSON string dictionary (eg. "Value" = {"numChord": 5, "spaceChord": 1.0, "numSpan": 10, "space↩ Span": 0.5}) the following keywords (= default values) may be used:

- **groupName = "(no default)"**
  Single or list of *capsGroup* names used to define the surface (e.g. "Name1" or ["Name1","Name2",...]. If no groupName variable is provided an attempted will be made to use the tuple name instead;

- **numChord = 10**
  The number of chordwise horseshoe vortices placed on the surface.

- **spaceChord = 0.0**
  The chordwise vortex spacing parameter.

- **numSpanTotal = 0**
  Total number of spanwise horseshoe vortices placed on the surface. The vorticies are 'evenly' distributed across sections to minimize jumps in spacings. numpSpanPerSection must be zero if this is set.

- **numSpanPerSection = 0**
  The number of spanwise horseshoe vortices placed on each section the surface. The total number of spanwise vorticies are (numSection-1)∗numSpanPerSection. The vorticies are 'evenly' distributed across sections to minimize jumps in spacings. numSpanTotal must be zero if this is set.

- **spaceSpan = 0.0**
  The spanwise vortex spacing parameter.

0.10.2 Single Value String

If "Value" is a single string the following options maybe used:

- (NONE Currently)

0.11 Vortex Lattice Control Surface

Structure for the Vortex Lattice Control Surface tuple = ("Name of Control Surface", "Value"). "Name of control surface defines the name of the control surface in which the data should be applied. The "Value" must be a JSON String dictionary (see Section JSON String Dictionary).

0.11.1 JSON String Dictionary

If "Value" is a JSON string dictionary (e.g. "Value" = {"deflectionAngle": 10.0}) the following keywords (= default values) may be used:

0.11.2 Single Value String

If "Value" is a single string, the following options maybe used:

- (NONE Currently)