TACS Analysis Interface Module (AIM)

Marshall Galbraith MIT and Ryan Durscher AFRL/RQVC

0.1 Introduction

0.1.1 TACS AIM Overview

A module in the Computational Aircraft Prototype Syntheses (CAPS) has been developed to interact (primarily through input files) with the finite element structural solver TACS **[TACS]**.

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

Details of the AIM's automated data transfer capabilities are outlined in TACS Data Transfer

0.2 AIM attributes

The following list of attributes are required for the TACS AIM inside the geometry input.

- **capsDiscipline** This attribute is a requirement if doing aeroelastic analysis within TACS. capsDiscipline allows the AIM to determine which bodies are meant for structural analysis and which are used for aerodynamics. Options are: Structure and Aerodynamic (case insensitive).
- **capsGroup** This is a name assigned to any geometric body to denote a property. This body could be a solid, surface, face, wire, edge or node. Recall that a string in ESP starts with a $. For example, attribute `capsGroup $Wing`.
- **capsLoad** This is a name assigned to any geometric body where a load is applied. This attribute was separated from the `capsGroup` attribute to allow the user to define a local area to apply a load on without adding multiple `capsGroup` attributes. Recall that a string in ESP starts with a $. For example, attribute `capsLoad $force`.
- **capsConstraint** This is a name assigned to any geometric body where a constraint/boundary condition is applied. This attribute was separated from the `capsGroup` attribute to allow the user to define a local area to apply a boundary condition without adding multiple `capsGroup` attributes. Recall that a string in ESP starts with a $. For example, attribute `capsConstraint $fixed`.
- **capsIgnore** It is possible that there is a geometric body (or entity) that you do not want the TACS AIM to pay attention to when creating a finite element model. The capsIgnore attribute allows a body (or entity) to be in the geometry and ignored by the AIM. For example, because of limitations in OpenCASCADE a situation where two edges are overlapping may occur; capsIgnore allows the user to only pay attention to one of the overlapping edges.
- **capsConnect** This is a name assigned to any geometric body where the user wishes to create "fictitious" connections such as springs, dampers, and/or rigid body connections to. The user must manually specify the connection between two `capsConnect` entities using the "Connect" tuple (see AIM Inputs). Recall that a string in ESP starts with a $. For example, attribute `capsConnect $springStart`.
- **capsConnectLink** Similar to `capsConnect`, this is a name assigned to any geometric body where the user wishes to create "fictitious" connections to. A connection is automatically made if a `capsConnectLink` matches a `capsConnect` group. Again further specifics of the connection are input using the "Connect" tuple (see AIM Inputs). Recall that a string in ESP starts with a $. For example, attribute `capsConnect↩ Link $springEnd`.
- **capsResponse** This is a name assigned to any geometric body that will be used to define design sensitivity responses for optimization. Specific information for the responses are input using the "Design_Response" tuple (see AIM Inputs). Recall that a string in ESP starts with a $. For examples, attribute `capsResponse $displacementNode`.

- **capsBound** This is used to mark surfaces on the structural grid in which data transfer with an external solver will take place. See TACS Data Transfer for additional details.

Internal Aeroelastic Analysis

- **capsBound** This is used to mark surfaces on the structural grid in which a spline will be created between the structural and aero-loads.
- **capsReferenceArea** [Optional: Default 1.0] Reference area to use when doing aeroelastic analysis. This attribute may exist on any aerodynamic cross-section.
- **capsReferenceChord** [Optional: Default 1.0] Reference chord to use when doing aeroelastic analysis. This attribute may exist on any aerodynamic cross-section.
- **capsReferenceSpan** [Optional: Default 1.0] Reference span to use when doing aeroelastic analysis. This attribute may exist on any aerodynamic cross-section.

0.3 AIM Inputs

The following list outlines the TACS inputs along with their default value available through the AIM interface. Unless noted these values will be not be linked to any parent AIMs with variables of the same name.

- **Proj_Name = "nastran_CAPS"**
  This corresponds to the project name used for file naming.
- **Property = NULL**
  Property tuple used to input property information for the model, see FEA Property for additional details.
- **Material = NULL**
  Material tuple used to input material information for the model, see FEA Material for additional details.
- **Constraint = NULL**
  Constraint tuple used to input constraint information for the model, see FEA Constraint for additional details.
- **Load = NULL**
  Load tuple used to input load information for the model, see FEA Load for additional details.
- **Analysis = NULL**
  Analysis tuple used to input analysis/case information for the model, see FEA Analysis for additional details.
- **Analysis_Type = "Modal"**
  Type of analysis to generate files for, options include "Modal", "Static", "AeroelasticTrim", "AeroelasticFlutter", and "Optimization". Note: "Aeroelastic" and "StaticOpt" are still supported and refer to "AeroelasticTrim" and "Optimization".
- **File_Format = "Small"**
  Formatting type for the bulk file. Options: "Small", "Large", "Free".
- **Mesh_File_Format = "Small"**
  Formatting type for the mesh file. Options: "Small", "Large", "Free".
- **Design_Variable = NULL**
  The design variable tuple is used to input design variable information for the model optimization, see FEA DesignVariable for additional details.
- **Design_Variable_Relation = NULL**
  The design variable relation tuple is used to input design variable relation information for the model optimization, see FEA DesignVariableRelation for additional details.

- **Design_Constraint = NULL**
  The design constraint tuple is used to input design constraint information for the model optimization, see FEA DesignConstraint for additional details.

- **Design_Equation = NULL**
  The design equation tuple used to input information defining equations for use in design sensitivity, see FEA DesignEquation for additional details.

- **Design_Table = NULL**
  The design table tuple used to input table of real constants used in equations, see FEA TableConstant for additional details.

- **Design_Response = NULL**
  The design response tuple used to input design sensitivity response information, see FEA DesignResponse for additional details.

- **Design_Equation_Response = NULL**
  The design equation response tuple used to input design sensitivity equation response information, see FEA DesignEquationResponse for additional details.

- **Design_Opt_Param = NULL**
  The design optimization parameter tuple used to input parameters used in design optimization.

- **Support = NULL**
  Support tuple used to input support information for the model, see FEA Support for additional details.

- **Connect = NULL**
  Connect tuple used to define connection to be made in the, see FEA Connection for additional details.

- **Parameter = NULL**
  Parameter tuple used to define PARAM entries. Note, entries are output exactly as inputed, that is, if the PARAM entry requires an integer entry the user must input an integer!

- **Mesh = NULL**
  A Mesh link.

0.4 AIM Outputs

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

0.5 TACS Data Transfer

The TACS AIM has the ability to transfer displacements and eigenvectors from the AIM and pressure distributions to the AIM using the conservative and interpolative data transfer schemes in CAPS.

0.6 FEA Material

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

0.6.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

- **materialType = "Isotropic"**
  Material property type. Options: Isotropic, Anisothotropic, Orthotropic, or Anisotropic.

0.6.2 Single Value String

If "Value" is a string, the string value may correspond to an entry in a predefined material lookup table. NOT YET IMPLEMENTED!!!!

0.7 FEA Property

Structure for the property tuple = ("Property Name", "Value"). "Property Name" defines the reference `capsGroup` for the property being specified. The "Value" can either be a JSON String dictionary (see Section JSON String Dictionary) or a single string keyword (see Section Single Value String).

0.7.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

- **propertyType = No Default value**
  Type of property to apply to a give capsGroup `Name`. Options: ConcentratedMass, Rod, Bar, Shear, Shell, Membrane, Composite, and Solid

- **material = ‘Material Name’ (FEA Material)**
  ‘Material Name’ from FEA Material to use for property. If no material is set the first material created will be used

- **crossSecArea = 0.0**
  Cross sectional area.

- **torsionalConst = 0.0**
  Torsional constant.

- **torsionalStressReCoeff = 0.0**
  Torsional stress recovery coefficient.

- **zAxisInertia = 0.0**
  Section moment of inertia about the element z-axis.

- **yAxisInertia = 0.0**
  Section moment of inertia about the element y-axis.

- **yCoords[4] = [0.0, 0.0, 0.0, 0.0]**
  Element y-coordinates, in the bar cross-section, of four points at which to recover stresses

- **zCoords[4] = [0.0, 0.0, 0.0, 0.0]**
  Element z-coordinates, in the bar cross-section, of four points at which to recover stresses

- **areaShearFactors[2] = [0.0, 0.0]**
  Area factors for shear.

- **crossProductInertia = 0.0**
  Section cross-product of inertia.

- **membraneThickness = 0.0**
  Membrane thickness.

- **bendingInertiaRatio = 1.0**
  Ratio of actual bending moment inertia to the bending inertia of a solid plate of thickness "membrane↩ Thickness"

- **shearMembraneRatio = 5.0/6.0**
  Ratio shear thickness to membrane thickness.

- **materialBending = "Material Name" (FEA Material)**
  "Material Name" from FEA Material to use for property bending.

- **materialShear = "Material Name" (FEA Material)**
  "Material Name" from FEA Material to use for property shear.

- **compositeMaterial = "no default"**
  List of "Material Name"s, ["Material Name -1", "Material Name -2", ...], from FEA Material to use for composites.

- **shearBondAllowable = 0.0**
  Allowable interlaminar shear stress.

- **symmetricLaminate = False**
  Symmetric lamination option. True- SYM only half the plies are specified, for odd number plies 1/2 thickness of center ply is specified with the first ply being the bottom ply in the stack, default (False) all plies specified

- **compositeFailureTheory = "(no default)"**
  Composite failure theory.

- **compositeThickness = (no default)**
  List of composite thickness for each layer (e.g. [1.2, 4.0, 3.0]). If the length of this list doesn't match the length of the "compositeMaterial" list, the list is either truncated [$>$length("compositeMaterial")] or expanded [$<$length("compositeMaterial")] in which case the last thickness provided is repeated.

- **compositeOrientation = (no default)**
  List of composite orientations (angle relative element material axis) for each layer (eg. [5.0, 10.0, 30.0]). If the length of this list doesn't match the length of the "compositeMaterial" list, the list is either truncated [$>$length("compositeMaterial")] or expanded [$<$length("compositeMaterial")] in which case the last orientation provided is repeated.

- **mass = 0.0**
  Mass value.

- **massOffset = [0.0, 0.0, 0.0]**
  Offset distance from the grid point to the center of gravity for a concentrated mass.

- **massInertia = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]**
  Mass moment of inertia measured at the mass center of gravity.

0.7.2 Single Value String

If "Value" is a string, the string value may correspond to an entry in a predefined property lookup table. NOT YET IMPLEMENTED!!!!

0.8 FEA Constraint

Structure for the constraint tuple = ("Constraint 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 Single Value String).

0.8.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

0.8.2 Single Value String

If "Value" is a string, the string value may correspond to an entry in a predefined constraint lookup table. NOT YET IMPLEMENTED!!!!

0.9 FEA Support

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

0.9.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

0.9.2 Single Value String

If "Value" is a string, the string value may correspond to an entry in a predefined support lookup table. NOT YET IMPLEMENTED!!!!

0.10 FEA Connection

Structure for the connection tuple = ("Connection Name", "Value"). "Connection Name" defines the reference name to the capsConnect being specified and denotes the "source" node for the connection. The "Value" can either be a JSON String dictionary (see Section JSON String Dictionary) or a single string keyword (see Section Single Value String).

0.10.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

0.10.2 Single Value String

If "Value" is a string, the string value may correspond to an entry in a predefined connection lookup table. NOT YET IMPLEMENTED!!!!

0.11 FEA Load

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

0.11.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

0.11.2 Single Value String

If "Value" is a string, the string value may correspond to an entry in a predefined load lookup table. NOT YET IMPLEMENTED!!!!

0.12 FEA Analysis

Structure for the analysis tuple = (‘Analysis Name’, ‘Value’). 'Analysis Name' defines the reference name for the analysis being specified. The "Value" can either be a JSON String dictionary (see Section JSON String Dictionary) or a single string keyword (see Section Single Value String).

0.12.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

0.12.2 Single Value String

If "Value" is a string, the string value may correspond to an entry in a predefined analysis lookup table. NOT YET IMPLEMENTED!!!!

0.13 FEA DesignVariable

Structure for the design variable tuple = ("DesignVariable Name", "Value"). "DesignVariable Name" defines the reference name for the design variable being specified. This string will be used in the FEA input directly. The "Value" must be a JSON String dictionary (see Section JSON String Dictionary).

0.13.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

0.14 FEA DesignVariableRelation

Structure for the design variable tuple = ("DesignVariableRelation Name", "Value"). "DesignVariableRelation Name" defines the reference name for the design variable being specified. This string will be used in the FEA input directly. The "Value" must be a JSON String dictionary (see Section JSON String Dictionary).

0.14.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

0.15 FEA DesignConstraint

Structure for the design constraint tuple = (‘DesignConstraint Name’, ‘Value’). 'DesignConstraint Name' defines the reference name for the design constraint being specified. The "Value" must be a JSON String dictionary (see Section JSON String Dictionary).

0.15.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

0.16 FEA DesignEquation

Structure for the design equation tuple = ("DesignEquation Name", ["Value1", ... , "ValueN"]). "DesignEquation Name" defines the reference name for the design equation being specified. This string will be used in the FEA input directly. The values "Value1", ... , "ValueN" are a list of strings containing the equation defintions. (see Section List of equation strings).

0.16.1 List of equation strings

Each design equation tuple value is a list of strings containing the equation definitions

0.17 FEA TableConstant

Structure for the table constant tuple = ("TableConstant Name", "Value"). "TableConstant Name" defines the reference name for the table constant being specified. This string will be used in the FEA input directly. The "Value" is the value of the table constant.

0.18 FEA DesignResponse

Structure for the design response tuple = ("DesignResponse Name", "Value"). "DesignResponse Name" defines the reference name for the design response being specified. This string will be used in the FEA input directly. The "Value" must be a JSON String dictionary (see Section JSON String Dictionary).

0.18.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

0.19 FEA DesignEquationResponse

Structure for the design equation response tuple = ("DesignEquationResponse Name", "Value"). "DesignEquation↩ Response Name" defines the reference name for the design equation response being specified. This string will be used in the FEA input directly. The "Value" must be a JSON String dictionary (see Section JSON String Dictionary).

0.19.1 JSON String Dictionary

If "Value" is JSON string dictionary the following keywords (= default values) may be used:

0.20 FEA DesignOptParam

Structure for the design optimization parameter tuple = ("DesignOptParam Name", "Value"). "DesignOptParam Name" defines the reference name for the design optimization parameter being specified. This string will be used in the FEA input directly. The "Value" is the value of the design optimization parameter.