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 = "tacs_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 Design Variables 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 Design Constraints for additional details.

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

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

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

- **Design_Equation_Response = NULL**
  The design equation response tuple used to input design sensitivity equation response information, see FEA Design Equation Responses 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 (e.g. "Value" = {"density": 7850, "youngModulus": 120000.0, "poissonRatio": 0.5, "materialType": "isotropic"}) the following keywords (= default values) may be used:

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

- **youngModulus = 0.0**
  Also known as the elastic modulus, defines the relationship between stress and strain. Default if 'shear↩ Modulus' and 'poissonRatio' != 0, youngModulus = 2∗(1+poissonRatio)∗shearModulus

- **shearModulus = 0.0**
  Also known as the modulus of rigidity, is defined as the ratio of shear stress to the shear strain. Default if 'youngModulus' and 'poissonRatio' != 0, shearModulus = youngModulus/(2∗(1+poissonRatio))

- **poissonRatio = 0.0**
  The fraction of expansion divided by the fraction of compression. Default if 'youngModulus' and 'shear↩ Modulus' != 0, poissonRatio = (2∗youngModulus/shearModulus) - 1

- **density = 0.0**
  Density of the material.

- **thermalExpCoeff = 0.0**
  Thermal expansion coefficient of the material.

- **thermalExpCoeffLateral = 0.0**
  Thermal expansion coefficient of the material.

- **temperatureRef = 0.0**
  Reference temperature for material properties.

- **dampingCoeff = 0.0**
  Damping coefficient for the material.

- **yieldAllow = 0.0**
  Yield strength/allowable for the material.

- **tensionAllow = 0.0**
  Tension allowable for the material.

- **tensionAllowLateral = 0.0**
  Lateral tension allowable for the material.

- **compressAllow = 0.0**
  Compression allowable for the material.

- **compressAllowLateral = 0.0**
  Lateral compression allowable for the material.

- **shearAllow = 0.0**
  Shear allowable for the material.

- **allowType = 0**
  This flag defines if the above allowables `compressAllow` etc. are defined in terms of stress (0) or strain (1). The default is stress (0).

- **youngModulusLateral = 0.0**
  Elastic modulus in lateral direction for an orthotropic material

- **shearModulusTrans1Z = 0.0**
  Transverse shear modulus in the 1-Z plane for an orthotropic material

- **shearModulusTrans2Z = 0.0**
  Transverse shear modulus in the 2-Z plane for an orthotropic material

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 `caps←Group` 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 (e.g. "Value" = {"shearMembraneRatio": 0.83, "bendingInertiaRatio": 1.←0, "membraneThickness": 0.2, "propertyType": "Shell"}) the following keywords (= default values) may be used:

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

- **massPerLength = 0.0**
  Non-structural mass per unit length.

- **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.

- **crossSecType = NULL**
  Cross-section type. Must be one of following character variables: BAR, BOX, BOX1, CHAN, CHAN1, CHAN2, CROSS, H, HAT, HEXA, I, I1, ROD, T, T1, T2, TUBE, or Z.

- **crossSecDimension = [0,0,0,....]**
  Cross-sectional dimensions (length of array is dependent on the "crossSecType"). Max supported length array is 10!

- **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. If no material is given and "bendingInertia↩Ratio" is greater than 0, the material name provided in "material" is used.

- **materialShear = "Material Name" (FEA Material)**
  "Material Name" from FEA Material to use for property shear. If no material is given and "shearMembraneRatio" is greater than 0, the material name provided in "material" is used.

- **massPerArea = 0.0**
  Non-structural mass per unit area.

- **zOffsetRel = 0.0**
  Relative offset from the surface of grid points to the element reference plane as a percentage of the thickness. zOffSet = thickness$*$zOffsetRel/100

- **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. Options: "HILL", "HOFF", "TSAI", and "STRN"

- **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 (eg. "Value" = {"groupName": "plateEdge", "dofConstraint": 123456}) the following keywords (= default values) may be used:

- **constraintType = "ZeroDisplacement"**
  Type of constraint. Options: "Displacement", "ZeroDisplacement".

- **dofConstraint = 0**
  Component numbers / degrees of freedom that will be constrained (123 - zero translation in all three directions).

- **gridDisplacement = 0.0**
  Value of displacement for components defined in "dofConstraint".

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 (eg. "Value" = {"groupName": "plateEdge", "dofSupport": 123456}) the following keywords (= default values) may be used:

- **groupName = "(no default)"**
  Single or list of `capsConstraint` names on which to apply the support (e.g. "Name1" or ["Name1","↩
  Name2",...]. If not provided, the constraint tuple name will be used.

- **dofSupport = 0**
  Component numbers / degrees of freedom that will be supported (123 - zero translation in all three directions).

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 (e.g. "Value" = {"dofDependent": 1, "propertyType": "RigidBody"}) the following keywords (= default values) may be used:

- **connectionType = RigidBody**
  Type of connection to apply to a given capsConnect pair defined by "Connection Name" and the "groupName". Options: Mass (scalar), Spring (scalar), Damper (scalar), RigidBody, RigidBodyInterpolate.

- **dofDependent = 0**
  Component numbers / degrees of freedom of the dependent end of rigid body connections (ex. 123 - translation in all three directions).

- **componentNumberStart = 0**
  Component numbers / degrees of freedom of the starting point of the connection for mass, spring, and damper elements (scalar) (0 <= Integer <= 6).

- **componentNumberEnd= 0**
  Component numbers / degrees of freedom of the ending point of the connection for mass, spring, damper elements (scalar), and rigid body interpolative connection (0 <= Integer <= 6).

- **stiffnessConst = 0.0**
  Stiffness constant of a spring element (scalar).

- **dampingConst = 0.0**
  Damping coefficient/constant of a spring or damping element (scalar).

- **stressCoeff = 0.0**
  Stress coefficient of a spring element (scalar).

- **mass = 0.0**
  Mass of a mass element (scalar).

- **weighting = 1**
  Weighting factor for a rigid body interpolative connections.

- **groupName = "(no default)"**
  Single or list of `capsConnect` names on which to connect the nodes found with the tuple name ("↩
  Connection Name") to. (e.g. "Name1" or ["Name1","Name2",...].

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 (e.g. "Value" = {"groupName": "plate", "loadType": "Pressure", "pressureForce": 2000000.0}) the following keywords (= default values) may be used:

- **loadType = "(no default)"**
  Type of load. Options: "GridForce", "GridMoment", "Rotational", "Thermal", "Pressure", "PressureDistribute", "PressureExternal", "Gravity".

- **groupName = "(no default)"**
  Single or list of `capsLoad` names on which to apply the load (e.g. "Name1" or ["Name1","Name2",...]. If not provided, the load tuple name will be used.

- **loadScaleFactor = 1.0**
  Scale factor to use when combining loads.

- **forceScaleFactor = 0.0**
  Overall scale factor for the force for a "GridForce" load.

- **directionVector = [0.0, 0.0, 0.0]**
  X-, y-, and z- components of the force vector for a "GridForce", "GridMoment", or "Gravity" load.

- **momentScaleFactor = 0.0**
  Overall scale factor for the moment for a "GridMoment" load.

- **gravityAcceleration = 0.0**
  Acceleration value for a "Gravity" load.

- **pressureForce = 0.0**
  Uniform pressure force for a "Pressure" load.

- **pressureDistributeForce = [0.0, 0.0, 0.0, 0.0]**
  Distributed pressure force for a "PressureDistribute" load. The four values correspond to the 4 (quadrilateral elements) or 3 (triangle elements) node locations.

- **angularVelScaleFactor = 0.0**
  An overall scale factor for the angular velocity in revolutions per unit time for a "Rotational" load.

- **angularAccScaleFactor = 0.0**
  An overall scale factor for the angular acceleration in revolutions per unit time squared for a "Rotational" load.

- **coordinateSystem = "(no default)"**
  Name of coordinate system in which defined force components are in reference to. If no value is provided the global system is assumed.

- **temperature = 0.0**
  Temperature at a given node for a "Temperature" load.

- **temperatureDefault = 0.0**
  Default temperature at a node not explicitly being used for a "Temperature" load.

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 (e.g. "Value" = {"numDesiredEigenvalue": 10, "eigenNormaliztion": "MASS", "numEstEigenvalue": 1, "extractionMethod": "GIV", "frequencyRange": [0, 10000]}) the following keywords (= default values) may be used:

- **analysisType = "Modal"**
  Type of load. Options: "Modal", "Static", "AeroelasticTrim", "AeroelasticFlutter" Note: "AeroelasticStatic" is still supported but refers to "AeroelasticTrim" Note: "Optimization" and "StaticOpt" are not valid - Optimization is initialized by the Analysis_Type AIM Input

- **analysisLoad = "(no default)"**
  Single or list of "Load Name"s defined in FEA Load in which to use for the analysis (e.g. "Name1" or ["↩
  Name1","Name2",...].

- **analysisConstraint = "(no default)"**
  Single or list of "Constraint Name"s defined in FEA Constraint in which to use for the analysis (e.g. "Name1" or ["Name1","Name2",...].

- **analysisSupport = "(no default)"**
  Single or list of "Support Name"s defined in FEA Support in which to use for the analysis (e.g. "Name1" or ["Name1","Name2",...].

- **analysisDesignConstraint = "(no default)"**
  Single or list of "Design Constraint Name"s defined in FEA Design Constraints in which to use for the analysis (e.g. "Name1" or ["Name1","Name2",...].

- **extractionMethod = "(no default)"**
  Extraction method for modal analysis.

- **frequencyRange = [0.0, 0.0]**
  Frequency range of interest for modal analysis.

- **numEstEigenvalue = 0**
  Number of estimated eigenvalues for modal analysis.

- **numDesiredEigenvalue = 0**
  Number of desired eigenvalues for modal analysis.

- **eigenNormaliztion = "(no default)"**
  Method of eigenvector renormalization. Options: "POINT", "MAX", "MASS"

- **gridNormaliztion = 0**
  Grid point to be used in normalizing eigenvector to 1.0 when using eigenNormaliztion = "POINT"

- **componentNormaliztion = 0**
  Degree of freedom about "gridNormalization" to be used in normalizing eigenvector to 1.0 when using eigen↩
  Normaliztion = "POINT"

- **lanczosMode = 2**
  Mode refers to the Lanczos mode type to be used in the solution. In mode 3 the mass matrix, Maa,must be nonsingular whereas in mode 2 the matrix K aa - sigma∗Maa must be nonsingular

- **lanczosType = "(no default)"**
  Lanczos matrix type. Options: DPB, DGB.

- **machNumber = 0.0 or [0.0, ..., 0.0]**
  Mach number used in trim analysis OR Mach numbers used in flutter analysis..

- **dynamicPressure = 0.0**
  Dynamic pressure used in trim analysis.

- **density = 0.0**
  Density used in trim analysis to determine true velocity, or flutter analysis.

- **aeroSymmetryXY = "(no default)"**
Aerodynamic symmetry about the XY Plane. Options: SYM, ANTISYM, ASYM. Aerodynamic symmetry about the XY Plane. Options: SYM, ANTISYM, ASYM. SYMMETRIC Indicates that a half span aerodynamic model is moving in a symmetric manner with respect to the XY plane. ANTISYMMETRIC Indicates that a half span aerodynamic model is moving in an antisymmetric manner with respect to the XY plane. ASYMMETRIC Indicates that a full aerodynamic model is provided.

- **aeroSymmetryXZ = "(no default)"**
Aerodynamic symmetry about the XZ Plane. Options: SYM, ANTISYM, ASYM. SYMMETRIC Indicates that a half span aerodynamic model is moving in a symmetric manner with respect to the XZ plane. ANTISYMMETRIC Indicates that a half span aerodynamic model is moving in an antisymmetric manner with respect to the XZ plane. ASYMMETRIC Indicates that a full aerodynamic model is provided.

- **rigidVariable = ["no default"]**
List of rigid body motions to be used as trim variables during a trim analysis. Nastran valid labels are: ANGLEA, SIDES, ROLL, PITCH, YAW, URDD1, URDD2, URDD3, URDD4, URDD5, URDD6

- **rigidConstraint = ["no default"]**
List of rigid body motions to be used as trim constraint variables during a trim analysis. Nastran valid labels are: ANGLEA, SIDES, ROLL, PITCH, YAW, URDD1, URDD2, URDD3, URDD4, URDD5, URDD6

- **magRigidConstraint = [0.0 , 0.0, ...]**
List of magnitudes of trim constraint variables. If none and 'rigidConstraint'(s) are specified then 0.0 is assumed for each rigid constraint.

- **controlConstraint = ["no default"]**
List of controls surfaces to be used as trim constraint variables during a trim analysis.

- **magControlConstraint = [0.0 , 0.0, ...]**
List of magnitudes of trim control surface constraint variables. If none and 'controlConstraint'(s) are specified then 0.0 is assumed for each control surface constraint.

- **reducedFreq = [0.1, ..., 20.0], No Default Values are defined.**
Reduced Frequencies to be used in Flutter Analysis. Up to 8 values can be defined.

- **flutterVel = [0.1, ..., 20.0]**
Velocities to be used in Flutter Analysis. If no values are provided the following relation is used

  v = sqrt(2∗dynamicPressure/density) dv = (v∗2 - v/2) / 20;

  flutterVel[0] = v/10 flutterVel[i] = v/2 + i∗dv; where i = 1....21 flutterVel[22] = v∗10;

- **visualFlutter = False**
Turn on flutter visualization f06 output.

- **analysisResponse = "(no default)"**
Single or list of "DesignResponse Name"s defined in FEA Design Responses to use for the analysis response spanning sets (e.g. "Name1" or ["Name1","Name2",...].

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 Design Variables

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). In Nastran the DesignVariable Name will be the LABEL used in the DESVAR input. For this reason the user should keep the length of this input to a minimum number of characters, ideally 7 or less.

- `DESVAR ID LABEL XINIT XLB XUB DELXV DDVAL`

0.13.1 JSON String Dictionary

If "Value" is JSON string dictionary (eg. "Value" = {"initialValue": 5.0, "upperBound": 10.0}) the following keywords (= default values) may be used:

- **initialValue = 0.0**
  Initial value for the design variable.

- **lowerBound = 0.0**
  Lower bound for the design variable.

- **upperBound = 0.0**
  Upper bound for the design variable.

- **maxDelta = 0.0**
  Move limit for the design variable.

- **discreteValue = 0.0**
  List of discrete values to use for the design variable (e.g. [0.0,1.0,1.5,3.0].

- **independentVariable = "(no default)"**
  Single or list of "DesignVariable Name"s (that is the Tuple name) used to create/designate a dependent design variable.
  - independentValue = variableWeight[1] + variableWeight[2] $*$ SUM{independentVariableWeight[i] $*$ independentVariable[i]}

- **independentVariableWeight = 1.0 or [1.0, 1.0, ...]**
  Single or list of weighting constants with respect to the variables set for "independentVariable". If the length of this list doesn't match the length of the "independentVariable" list, the list is either truncated [$>$length("independentVariable")] or expanded [$<$length("independentVariable")] in which case the **last weight is repeated**.

- **variableWeight = [1.0, 1.0]**
  Weighting constants for a dependent variable - used if "independentVariable"(s) have been provided.

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 (eg. "Value" = {"componentType": "Property", "componentName": "plate", "fieldName": "TM", "variableName": "MyDesVar"}) the following keywords (= default values) may be used:

- **componentType = "Property"**
  The type of component for this design variable relation. Options: "Material", "Property", "Element".

- **componentName = "(no default)"**
  Single or list of FEA Property(ies), or FEA Material name(s) linked to the design variable relation (e.g. "↩ Name1" or ["Name1","Name2",...].
  - For `componentType` Property a FEA Property name (or names) is given.
  - For `componentType` Material a FEA Material name (or names) is given.
  - For `componentType` Element a `capsGroup` Name (or names) is given.

- **variableName = "(no default)"**
  Single or list of names of design variables linked to this relation

- **fieldName = "(no default)"**
  Fieldname of variable relation (e.g. "E" for Young's Modulus). Design Variable Relations can be defined as three types based on the `variableType` value. These are Material, Property, or Element. This means that an aspect of a material, property, or element input can change in the optimization problem. This input specifies what aspect of the Material, Property, or Element is changing.

  1. **Material Types** Selected based on the material type (see FEA Material, materialType) referenced in the `componentName` above.
     - **`MAT1`**, `materialType = "Isotropic"`
       * "E", "G", "NU", "RHO", "A"
     - **`MAT2`**, `materialType = "Anisothotropic"`
       * "G11", "G12", "G13", "G22", "G23", "G33", "RHO", "A1", "A2", "A3"
     - **`MAT8`**, `materialType = "Orthotropic"`
       * "E1", "E2", "NU12", "G12", "G1Z", "G2Z", "RHO", "A1", "A2"
     - **`MAT9`**, `materialType = "Anisotropic"`
       * "G11", "G12", "G13", "G14", "G15", "G16"
       * "G22", "G23", "G24", "G25", "G26"
       * "G33", "G34", "G35", "G36"
       * "G44", "G45", "G46"
       * "G55", "G56", "G66"
       * "RHO", "A1", "A2", "A3", "A4", "A5", "A6"
  2. **Property Types** (see FEA Property)
     - **`PROD`** `propertyType = "Rod"`
       * "A", "J"
     - **`PBAR`** `propertyType = "Bar"`

        - "A", "I1", "I2", "J"
    - **PSHELL** `propertyType = "Shell"`
        - "T"
    - **PCOMP** `propertyType = "Composite"`
        - "T1", "THETA1", "T2", "THETA2", ... "Ti", "THETAi"
    - **PSOLID** `propertyType = "Solid"`
        - not supported

3. **Element Types**
    - **CTRIA3, CQUAD4** `propertyType = "Shell"`
        - "ZOFFS"

- **fieldPosition = 0**
  This input is ignored if not defined. The user may use this field instead of the `fieldName` input defined above to relate design variables and property, material, or elements. This requires knowledge of Nastran bulk data input format for material, property, and element input cards.

- **constantCoeff = 0.0**
  Constant term of relation.

- **linearCoeff = 1.0**
  Single or list of coefficients of linear relation. Must be same length as `variableName`.

0.15 FEA Design Constraints

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 (eg. "Value" = {"groupName": "plate", "upperBound": 10.0}) the following keywords (= default values) may be used:

- **groupName = "(no default)"**
  Single or list of `capsGroup` name(s) to the design variable (e.g. "Name1" or ["Name1","Name2",...].The property (see FEA Property) also assigned to the same `capsGroup` will be automatically related to this constraint entry.

- **constraintType = "Property"**
  The type of design constraint. Options: "Property", "Flutter"

- **lowerBound = 0.0**
  Lower bound for the design constraint.

- **upperBound = 0.0**
  Upper bound for the design constraint.

- **responseType = "(no default)"**
  Response type options for DRESP1 Entry (see Nastran manual).
  - Implemented Options
    1. `STRESS`, for `propertyType = "Rod" or "Shell"` (see FEA Property)
    2. `CFAILURE`, for `propertyType = "Composite"` (see FEA Property)

- **fieldName = "(no default)"**
  For constraints, this field is only used currently when applying constraints to composites. This field is used to identify the specific lamina in a stacking sequence that a constraint is being applied too. Note if the user has design variables for both THEATA1 and T1 it is likely that only a single constraint on the first lamina is required. For this reason, the user can simply enter LAMINA1 in addition to the possible entries defined in the FEA Design Variables section. Additionally, the `fieldPosition` integer entry below can be used. In this case `"LAMINA1" = 1`.
  - -# **Property Types** (see FEA Property)
    - **PCOMP** `propertyType = "Composite"`
      - "T1", "THETA1", "T2", "THETA2", ... "Ti", "THETAi"
      - "LAMINA1", "LAMINA2", ... "LAMINAi"

- **fieldPosition = 0**
  This input is ignored if not defined. The user may use this field instead of the `fieldName` input defined above to identify a specific lamina in a composite stacking sequence where a constraint is applied. Please read the `fieldName` information above for more information.

0.16 FEA Optimization Control

Structure for the optimization control dictionary = 'Value'. The "Value" must be a JSON String dictionary (see Section JSON String Dictionary).

- **fullyStressedDesign = 0**
  Number of iterations with fully stressed design.

- **mathProgramming = 30**
  Number of iterations for math programming methods.

- **maxIter = 30**
  Maximum number of optimization iterations.

- **constraintRetention = 1.5**
  Constraint retention factor. Will be at least 1.5 times the number of design variables

- **eps = 1.0**
  Constraint retention parameter in which all constraints having a value greater than "eps" will be considered active.

- **moveLimit = 1.0**
  Move limit bound.

0.17 FEA Design Equations

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).

0.17.1 List of equation strings

Each design equation tuple value is a list of strings containing the equation definitions (eg. ["dispsum3(s1,s2,s3)=sum(s1,s2,s3)"]

0.18 FEA Table Constants

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. In Nastran the TableConstant Name will be the LABLi used in the DTABLE input. For this reason the user should keep the length of this input to a minimum number of characters, ideally 7 or less.

- `DTABLE LABL1 VALU1 LABL2 VALU2 LABL3 VALU3 -etc-`

0.19 FEA Design Responses

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). In Nastran the DesignResponse Name will be the LABEL used in the DRESP1 input. For this reason the user should keep the length of this input to a minimum number of characters, ideally 7 or less.

- `DRESP1 ID LABEL RTYPE PTYPE REGION ATTA ATTB ATT1 ATT2 -etc-`

0.19.1 JSON String Dictionary

If "Value" is JSON string dictionary (eg. "Value" = {"responseType": "DISP", groupName": "plate", "component": 3}) the following keywords (= default values) may be used:

- **responseType**
  Type of design sensitivity response. For options, see NASTRAN User Guide DRESP1 Design Sensitivity Response Attributes table.
- **component = "(no default)"**
  Component flag.
- **groupName = "(no default)"**
  Defines the reference `capsGroup` for the node being specified for the response.

0.20 FEA Design Equation Responses

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). In Nastran the DesignEquationResponse Name will be the LABEL used in the DRESP2 input. For this reason the user should keep the length of this input to a minimum number of characters, ideally 7 or less.

- `DRESP2 ID LABEL EQID REGION ...`

0.20.1 JSON String Dictionary

If "Value" is JSON string dictionary (eg. "Value" = {"equation": "EQ1", "constant": ["PI", "YM", "L"]}) the following keywords (= default values) may be used:

- **equation**
  The name of the equation referenced by this equation response.

- **variable = "(no default)"**
  Single or list of names of design variable equation parameters.

- **constant = "(no default)"**
  Single or list of names of table constant equation parameters.

- **response = "(no default)"**
  Single or list of names of design response equation parameters.

- **equationResponse = "(no default)"**
  Single or list of names of design equation response equation parameters.

0.21 FEA Design Optimization Parameters

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. In Nastran the DesignOpt↩ Param Name will be the PARAMi used in the DOPTPRM input. For this reason the user should keep the length of this input to a minimum number of characters, ideally 7 or less.

- `DOPTPRM PARAM1 VAL1 PARAM2 VAL2 PARAM3 VAL3 -etc-`

0.22 FEA Aerodynamic References

The aerodynamic reference input must be a JSON String dictionary (see Section JSON String Dictionary).

0.22.1 JSON String Dictionary

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