Abaqus Analysis Interface Module (AIM)

Ryan Durscher
AFRL/RQVC

0.1 Introduction

0.1.1 Abaqus 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 Abaqus [1].

Current issues include:

- A thorough bug testing needs to be undertaken.

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

Details of the AIM's shareable data structures are outlined in sharableDataAbaqus if connecting this AIM to other AIMs in a parent-child like manner.

Details of the AIM's automated data transfer capabilities are outlined in Data transfer from Abaqus

0.2 Abaqus AIM attributes

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

- **capsGroup** This is a name assigned to any geometric body. 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 Astros 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.

0.3 AIM Inputs

The following list outlines the Abaqus 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 = "Abaqus_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".
- **Mesh_Morph = False**
  Project previous surface mesh onto new geometry.
- **Mesh = NULL**
  A Mesh link.

0.4 AIM Outputs

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

None to date.

- **Tmax** = Maximum displacement.
- **T1max** = Maximum x-coordinate displacement.
- **T2max** = Maximum y-coordinate displacement.
- **T3max** = Maximum z-coordinate displacement.
- **vonMises_Grid** = Grid coordinate von Mises stress (dynamic output, only static analysis).
- **Displacement** = Grid coordinate displacement (dynamic output, only static analysis).

0.5 Abaqus Data Transfer

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

0.5.1 Data transfer from Abaqus

- **"Displacement"**
  Retrieves nodal displacements from the ∗.fil file.

0.5.2 Data transfer from Abaqus

- **"Pressure"**
  Writes appropriate load cards using the provided pressure distribution.

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.

- **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

- **kappa = 0.0**
  Thermal conductivity for an isotropic solid

- **K = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]**
  Thermal conductivity for an anisotropic solid (KXX, KXY, KXZ, KYY, KYZ, KZZ)

- **specificHeat = 0.0**
  Specific heat constant pressure (per unit mass) for an isotropic solid

Something else ....

- **honeycombCellSize = NULL**
  Honeycomb sandwich core cell size. Required if material defines the core of a honeycomb sandwich and dimpling stability index is desired

- **honeycombYoungModulus = NULL**
  Honeycomb sandwich core Young's modulus used for stability index analysis

- **honeycombShearModulus = NULL**
  Honeycomb sandwich core shear modulus used for stability index analysis

- **fractureAngle = NULL**
  Fracture angle for uniaxial transverse compression in degrees. Used in the NASA LaRC02 failure theory only

- **interlaminarShearAllow = NULL**
  Allowable inter-laminar shear stress of the composite laminate bonding material

- **fiberYoungModulus = NULL**
  Modulus of elasticity of fiber

- **fiberPoissonRatio = NULL**
  Poisson's ratio of fiber

- **meanStressFactor = NULL**
  Mean stress magnification factor

- **transTensionSlope = NULL**
  Failure envelop slope parameter for transverse tension

- **transCompressionSlope = NULL**
  Failure envelop slope parameter for transverse compression

- **compositeFailureTheory = NULL**
  Composite failure theory (string value)

- **interlaminarNormalStressAllow = NULL**
  Allowable inter-laminar normal stress of the composite laminate bonding material (allowable interlaminar normal stress)

- **youngModulusThick = NULL**
  Modulus of elasticity in thickness direction, also defined as the matrix direction or 3-direction

- **poissonRatio23 = 0.0**
  Poisson's ratio (for uniaxial loading in 2-direction)

- **poissonRatio31 = 0.0**
  Poisson's ratio (for uniaxial loading in 3-direction)

- **youngModulusFactor = NULL**
  Longitudinal modulus of elasticity reduction scale factor for nonlinear composite Progressive Ply Failure Analysis (PPFA)

- **youngModulusLateralFactor = NULL**
  Lateral modulus of elasticity reduction scale factor for nonlinear composite Progressive Ply Failure Analysis (PPFA)

- **shearModulusFactor = NULL**
  In-plane shear modulus reduction scale factor for nonlinear composite Progressive Ply Failure Analysis (PPFA)

- **shearModulusTrans1ZFactor = NULL**
  Transverse shear modulus reduction scale factor in 1-Z plane for nonlinear composite Progressive Ply Failure Analysis (PPFA)

- **shearModulusTrans2ZFactor = NULL**
  Transverse shear modulus reduction scale factor in 2-Z plane for nonlinear composite Progressive Ply Failure Analysis (PPFA)

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:

Something else ....

Something else ....

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

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

v Something else ....

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 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 (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: "Pressure", "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.

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

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

- **pressureForce = 0.0**
  Uniform pressure force for a "Pressure" 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, "eigenNormalization": "MASS", "numEstEigenvalue": 1, "extractionMethod": "GIV", "frequencyRange": [0, 10000]}) the following keywords (= default values) may be used:

- **analysisType = "Modal"**
  Type of load. Options: "Modal", "Static"

- **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",...].

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

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

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

- **eigenNormalization = "(no default)"**
  Method of eigenvector renormalization. Options: "DISPLACEMENT", "MASS"

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).

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

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).

0.17 FEA Mass Increments

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

0.18 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).

0.18.1 List of equation strings

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

0.19 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.

0.20 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).

0.20.1 JSON String Dictionary

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

0.21 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).

0.21.1 JSON String Dictionary

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

0.22 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.

0.23 FEA Aerodynamic References

Tuple of the aerodynamic reference input (see Section JSON String Dictionary).

0.23.1 JSON String Dictionary

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

Bibliography

[1] Dassault Systèmes. http://www.3ds.com/products-services/simulia/products/abaqus/. Accessed: May 2016. 1