FUN3D Analysis Interface Module (AIM)

Ryan Durscher
AFRL/RQVC

August 1, 2016

Contents

1 Introduction

1.1 FUN3D AIM Overview

A module in the Computational Aircraft Prototype Syntheses (CAPS) has been developed to interact (primarily through input files) with NASA LaRC's unstructured flow solver FUN3D. FUN3D is a parallelized flow analysis and design suite capable of addressing a wide variety of complex aerodynamic configurations by utilizing a mixed-element, node-based, finite volume discretization. The suite can simulate perfect gas (both incompressible and compressible), as well as multi-species equilibrium and non-equilibrium flows. Turbulence effects may be represented through a wide variety of models. Currently only a subset of FUN3D's input options have been exposed in the analysis interface module (AIM), but features can easily be included as future needs arise.

Current issues include:

- A thorough bug testing needs to be undertaken.
- Not all parameters/variables in fun3d.nml are currently available.

1.2 Generating fun3d.nml

FUN3D's primarily input file is a master FORTRAN namelist, fun3d.nml. Currently, the FUN3D AIM relies on a Python script to read and write this file. As such the AIM user needs Python installed so that the AIM may be complied with the Python API library (and header file - Python.h). The logic behind using Python to interact with the namelist file is simple, flexibility. Python allows dynamic creation of input variables, while each option available in C would have to be explicit defined. This would essentially lock the AIM into a particular version of FUN3D, as changes due to the addition and deprecation of variables would become cumbersome. Currently, the FUN3D AIM will first try to read an existing fun3d.nml file in the specified directory. If the file does not exist one will be created. Only modified input variables that have been specified as AIM inputs (currently supported variables are outlined in AIM Inputs are updated in the namelist file.

Alternatively, if the AIM is not built against the Python API a bare-bones fun3d.nml is created using only the variables set in AIM Inputs. For the fun3d.nml file to be created it is essential to set the Overwrite_NML input variable to True. This gives the user ample warning that their fun3d.nml (if it exist) will be over written.

Related pages:

- CFD Boundary Conditions
- Geometry Fidelity
- AIM Inputs
- AIM Outputs
- AIM Shareable Data
- FUN3D Data Transfer

2 Geometry Fidelity

The geometric fidelity for the FUN3D AIM requires that the body(ies) be either a face body(ies) (FACEBODY), solid body(ies) (SOLIDBODY) or non-manifold sheet body(ies) (SHEETBODY). For 2D simulations the attribute caps↩ Fidelity should be set to CFD with the geometric fidelity being either a planar SHEETBODY or FACEBODY, while for 3D simulations it should again be set to CFD but the both SOLIDBODY(ies) or SHEETBODY(ies) are acceptable.

3 AIM Inputs

The following list outlines the FUN3D inputs along with their default value available through the AIM interface. One will note most of the FUN3D parameters have a NULL value as their default. This is done since a parameter in the FUN3D input deck (fun3d.nml) is only changed if the value has been changed in CAPS (i.e. set to something other than NULL).

- **Proj_Name = "fun3d_CAPS"**
  This corresponds to the project_rootname variable in the &project namelist of fun3d.nml. This should be consistent with the meshing AIM.
- **Mach = NULL**
  This corresponds to the mach_number variable in the &reference_physical_properties namelist of fun3d.nml.
- **Re = NULL**
  This corresponds to the reynolds_number variable in the &reference_physical_properties namelist of fun3d.↩ nml.
- **Viscous = NULL**
  This corresponds to the viscous_terms variable in the &governing_equation namelist of fun3d.nml.

- **Eqn = NULL**
  This corresponds to the eqn_type variable in the &governing_equation namelist of fun3d.nml. This is chosen automatically based on the capsFidelity attribute.
- **Alpha = NULL**
  This corresponds to the angle_of_attack variable in the &reference_physical_properties namelist of fun3d.↩ nml.
- **Eqn = NULL**
  This corresponds to the angle_of_yaw variable in the &reference_physical_properties namelist of fun3d.nml.
- **Overwrite_NML = NULL**
  If Python is not linked with the FUN3d AIM at compile time this flag gives the AIM permission to overwrite fun3d.nml if present. The namelist produced will solely consist of input variables present and set in the AIM.
- **Mesh_Format = "AFLR3"**
  Mesh output format. By default an AFLR3 mesh will be used.
- **Mesh_ASCII_Flag = True**
  Output mesh in ASCII format, otherwise write a binary file if applicable.
- **Num_Iter = NULL**
  This corresponds to the steps variable in the &code_run_control namelist of fun3d.nml.
- **CFL_Schedule = NULL**
  This corresponds to the schedule_cfl variable in the &nonlinear_solver_parameters namelist of fun3d.nml.
- **CFL_Schedule_Inter = NULL**
  This corresponds to the schedule_iteration variable in the &nonlinear_solver_parameters namelist of fun3d.↩ nml.
- **Restart_Read = NULL**
  This corresponds to the restart_read variable in the &code_run_control namelist of fun3d.nml.
- **Boundary_Condition = NULL**
  See CFD Boundary Conditions for additional details.
- **Use_Python_NML = False**
  By default even if Python has been linked to the FUN3D AIM it is not used unless the this value is set to True.
- **Pressure_Scale_Factor = 1.0**
  Value to scale Cp data during when transferring data. Data is scaled based on Pressure = Pressure_Scale↩ _Factor∗Cp + Pressure_Scale_Offset.
- **Pressure_Scale_Offset = 0.0**
  Value to offset Cp data during when transferring data. Data is scaled based on Pressure = Pressure_Scale↩ _Factor∗Cp + Pressure_Scale_Offset.
- **NonInertial_Rotation_Rate = NULL [0.0, 0.0, 0.0]**
  Array values correspond to the rotation_rate_x, rotation_rate_y, rotation_rate_z variables, respectively, in the &noninertial_reference_frame namelist of fun3d.nml.
- **NonInertial_Rotation_Center = NULL, [0.0, 0.0, 0.0]**
  Array values correspond to the rotation_center_x, rotation_center_y, rotation_center_z variables, respectively, in the &noninertial_reference_frame namelist of fun3d.nml.
- **Two_Dimensional = False**
  Run FUN3D in 2D mode. If set to True, body must be a single "sheet" body in the x-z plane. A 3D mesh will be written out, where the body is extruded a length of 1 in the y-direction.
- **Modal_Aeroelastic = NULL**
  See Modal Aeroelastic Inputs for additional details.
- **Modal_Ref_Velocity = NULL**
  The freestream velocity in structural dynamics equation units; used for scaling during modal aeroelastic simulations. This corresponds to the uinf variable in the &aeroelastic_modal_data namelist of movingbody.input.

- **Modal_Ref_Length = 1.0**
  The scaling factor between CFD and the structural dynamics equation units; used for scaling during modal aeroelastic simulations. This corresponds to the grefl variable in the &aeroelastic_modal_data namelist of movingbody.input.
- **Modal_Ref_Dynamic_Pressure = NULL**
  The freestream dynamic pressure in structural dynamics equation units; used for scaling during modal aeroelastic simulations. This corresponds to the qinf variable in the &aeroelastic_modal_data namelist of movingbody.input.
- **Time_Accuracy = NULL**
  Defines the temporal scheme to use. This corresponds to the time_accuracy variable in the &nonlinear_↩ solver_parameters namelist of fun3d.nml.
- **Time_Step = NULL**
  Non-dimenional time step during time accurate simulations. This corresponds to the time_step_nondim variable in the &nonlinear_solver_parameters namelist of fun3d.nml.
- **Num_Subiter = NULL**
  Number of sub-interations used during a time step in a time accurate simulations. This corresponds to the subiterations variable in the &nonlinear_solver_parameters namelist of fun3d.nml.
- **Temporal_Error = NULL**
  This sets the tolerance for which subiterations are stopped during time accurate simulations. This corresponds to the temporal_err_floor variable in the &nonlinear_solver_parameters namelist of fun3d.nml.

4 AIM Shareable Data

Currently the FUN3D AIM does not have any shareable data types or values. It will try, however, to inherit a "↩ Volume_Mesh" (for 3D simulations) or a "Surface_Mesh" (for 2D simulations) from any parent AIMs. Note that the inheritance of the volume/surface mesh variable is required if the FUN3D aim is to write a suitable grid and *.mapbc (boundary condition file for the AFLR3 mesh format) file.

5 AIM Outputs

The following list outlines the FUN3D outputs available through the AIM interface. All variables currently correspond to values for all boundaries (total) found in the ∗.forces file

Net Forces - Pressure + Viscous:

- **CLtot** = The lift coefficient.
- **CDtot** = The drag coefficient.
- **CMXtot** = The moment coefficient about the x-axis.
- **CMYtot** = The moment coefficient about the y-axis.
- **CMZtot** = The moment coefficient about the z-axis.
- **CXtot** = The force coefficient about the x-axis.
- **CYtot** = The force coefficient about the y-axis.
- **CZtot** = The force coefficient about the z-axis.

Pressure Forces:

- **CLtot_p** = The lift coefficient - pressure contribution only.

- **CDtot_p** = The drag coefficient - pressure contribution only.
- **CMXtot_p** = The moment coefficient about the x-axis - pressure contribution only.
- **CMYtot_p** = The moment coefficient about the y-axis - pressure contribution only.
- **CMZtot_p** = The moment coefficient about the z-axis - pressure contribution only.
- **CXtot_p** = The force coefficient about the x-axis - pressure contribution only.
- **CYtot_p** = The force coefficient about the y-axis - pressure contribution only.
- **CZtot_p** = The force coefficient about the z-axis - pressure contribution only.

Viscous Forces:

- **CLtot_v** = The lift coefficient - viscous contribution only.
- **CDtot_v** = The drag coefficient - viscous contribution only.
- **CMXtot_v** = The moment coefficient about the x-axis - viscous contribution only.
- **CMYtot_v** = The moment coefficient about the y-axis - viscous contribution only.
- **CMZtot_v** = The moment coefficient about the z-axis - viscous contribution only.
- **CXtot_v** = The force coefficient about the x-axis - viscous contribution only.
- **CYtot_v** = The force coefficient about the y-axis - viscous contribution only.
- **CZtot_v** = The force coefficient about the z-axis - viscous contribution only.

6 FUN3D Data Transfer

Data transfer blah, blah, blah....

7 CFD Boundary Conditions

Structure for the boundary condition tuple = ("CAPS Group Name", "Value"). "CAPS Group Name" defines the capsGroup on which the boundary condition 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)

7.1 JSON String Dictionary

If "Value" is a JSON string dictionary (eg. "Value" = {"bcType": "Viscous", "wallTemperature": 1}) the following keywords (= default values) may be used:

- **bcType = "Inviscid"**
  Boundary condition type. Options: Inviscid, Viscous, Farfield, Extrapolate, Freestream, BackPressure, Symmetry, SubsonicInflow, SubsonicOutflow, MassflowIn, MassflowOut, FixedInflow, FixedOutflow.

7.1.1 Wall Properties

- **wallTemperature = 0.0**
  The ratio of wall temperature to reference temperature for inviscid and viscous surfaces. Adiabatic wall = -1

7.1.2 Wall Properties

7.1.3 Stagnation Properties

- **totalPressure = 0.0**
  Ratio of total pressure to reference pressure on a boundary surface.

- **totalTemperature = 0.0**
  Ratio of total temperature to reference temperature on a boundary surface.

7.1.4 Static Properties

- **staticPressure = 0.0**
  Ratio of static pressure to reference pressure on a boundary surface.

7.1.5 Velocity Components

- **machNumber = 0.0**
  Mach number on boundary.

7.1.6 Massflow Properties

- **massflow = 0.0**
  Massflow through the boundary in units of grid units squared.

7.2 Single Value String

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

- "Inviscid" (default)
- "Viscous"
- "Farfield"
- "Extrapolate"
- "Freestream"
- "SymmetryX"
- "SymmetryY"
- "SymmetryZ"

8 Modal Aeroelastic Inputs

Structure for the modal aeroelastic tuple = ("EigenVector_#", "Value"). The tuple name "EigenVector_#" defines the eigen-vector in which the supplied information corresponds to, where "#" should be replaced by the corresponding mode number for the eigen-vector (eg. EigenVector_3 would correspond to the third mode, while EigenVector_6 would be the sixth mode). This notation is the same as found in FUN3D Data Transfer. The "Value" must be a JSON String dictionary (see Section JSON String Dictionary).

8.1 JSON String Dictionary

If "Value" is a JSON string dictionary (eg. "Value" = {"generalMass": 1.0, "frequency": 10.7}) the following keywords (= default values) may be used:

- **frequency = 0.0**
  This is the frequency of specified mode, in rad/sec.

- **damping = 0.0**
  The critical damping ratio of the mode.

- **generalMass = 0.0**
  The generalized mass of the mode.

- **generalDisplacement = 0.0**
  The generalized displacement used at the starting time step to perturb the mode and excite a dynamic response.

- **generalVelocity = 0.0**
  The generalized velocity used at the starting time step to perturb the mode and excite a dynamic response.

- **generalForce = 0.0**
  The generalized force used at the starting time step to perturb the mode and excite a dynamic response.