Initial Implementation of VG Source Terms in FUN3D

Background

- Spent ~3 days on the implementation at a user's recent request – seems to be working, but have not spent significant time evaluating its use
  - Looking for users to try, provide feedback
- Implementation based on these references:
  - Bender, E.E., Anderson, B.H., and Yagle, P.J., "Vortex Generator Modeling for Navier-Stokes Codes," 3rd Joint ASME/JSME Fluids Engineering Conference, San Francisco, CA, 1999.
  - Waithe, K., "Source Term Model for Vortex Generator Vanes in a Navier-Stokes Computer Code," AIAA 2004-1236, January 2004.
  - Jirasek, A., "A Vortex Generator Model and its Application to Flow Control," AIAA 2004-4965, August 2004.

Initial Implementation of VG Source Terms in FUN3D

Details

- Will be available in v12.2
- Only intended for static grid computations currently
- Implementation supports as many vortex generators as user desires (up to 1000)
- Through the new `&vortex_generator` namelist, user provides the number of VG's, the model calibration constant, and the following values for each VG:
  - Planform area
  - Height
  - Approximate locations of two points on the surface defining the base of the VG
  - Intended boundary patches each of these points should be associated to during projection
  - Optional: have FUN3D reverse the orientations of the assumed $\mathbf{t}$ and $\mathbf{n}$ unit vectors
- Implemented for all element types
- Source terms are treated fully implicit

Initial Implementation of VG Source Terms in FUN3D

Sample Namelist

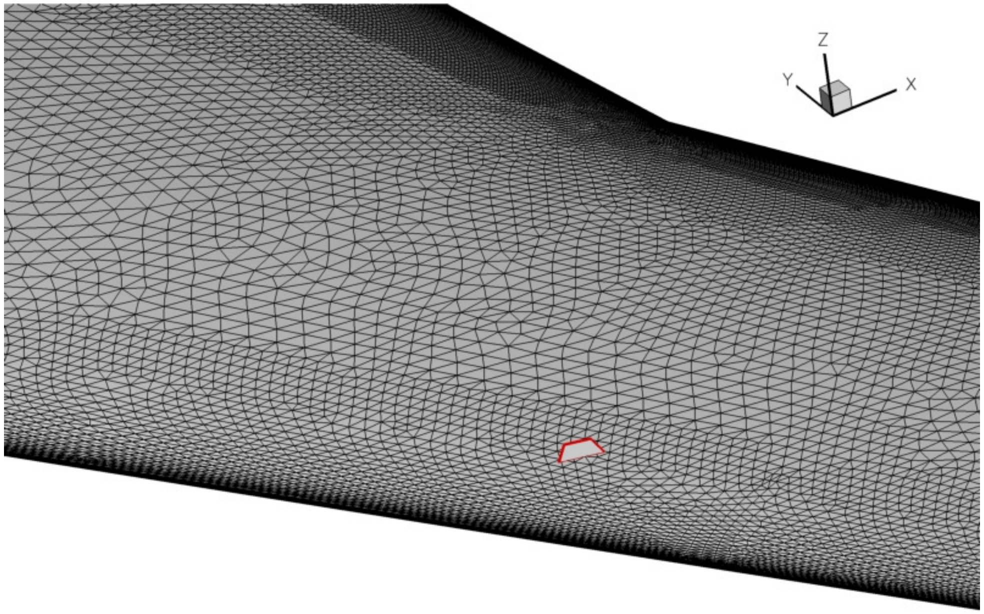
- Used f6fx2b_0.1 VGRID mesh available on DPW-3 website (5.6M nodes)
- Free-handed a rough VG for testing, using the following namelist

```
&vortex_generator
  number_of_vgs           = 1
  calibration_constant(1) = 100.
  planform_area(1)        = 12.0
  height(1)               = 2.5
  boundary_patch1(1)      = 31
  point1_xcoord(1)        = 176.1
  point1_ycoord(1)        = -319.59
  point1_zcoord(1)        = 32.40
  boundary_patch2(1)      = 31
  point2_xcoord(1)        = 181.50
  point2_ycoord(1)        = -322.30
  point2_zcoord(1)        = 33.06
  reverse_t(1)            = .false.
  reverse_n(1)            = .false.
/
```

Initial Implementation of VG Source Terms in FUN3D

Setup Process

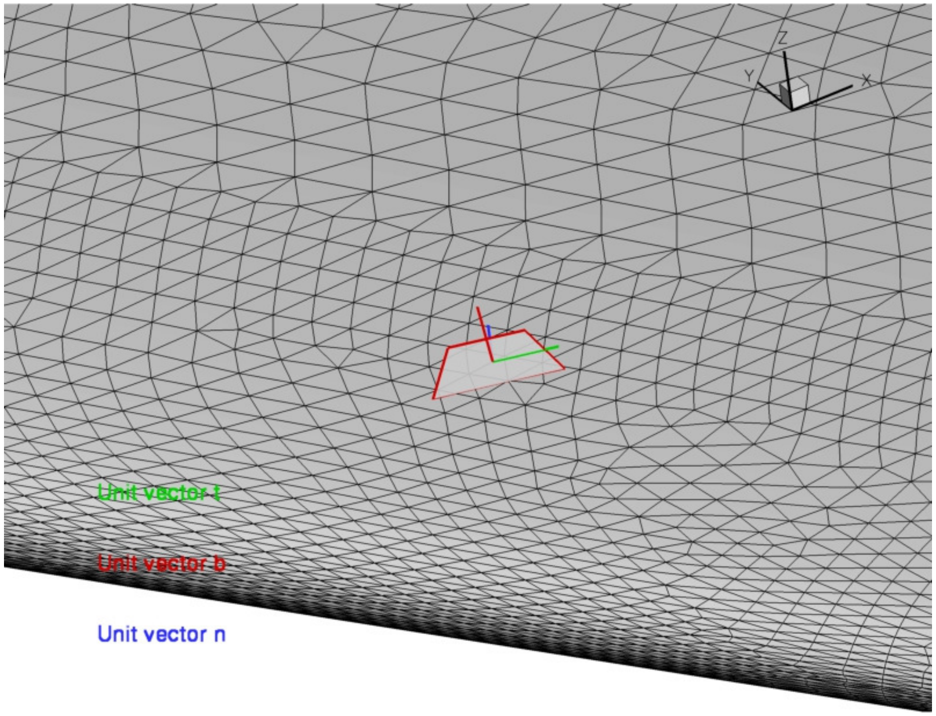
- After setting up the namelist, do a single cycle with the CLO '—animation_freq -1' and look at the provided data for the VG's
  - Load in f6fx2b_0.1_tec_boundary.dat
  - Then load in f6fx2b_vg_geometry.dat on top of it, which contains the geometries FUN3D constructed for each of your VG's



Initial Implementation of VG Source Terms in FUN3D

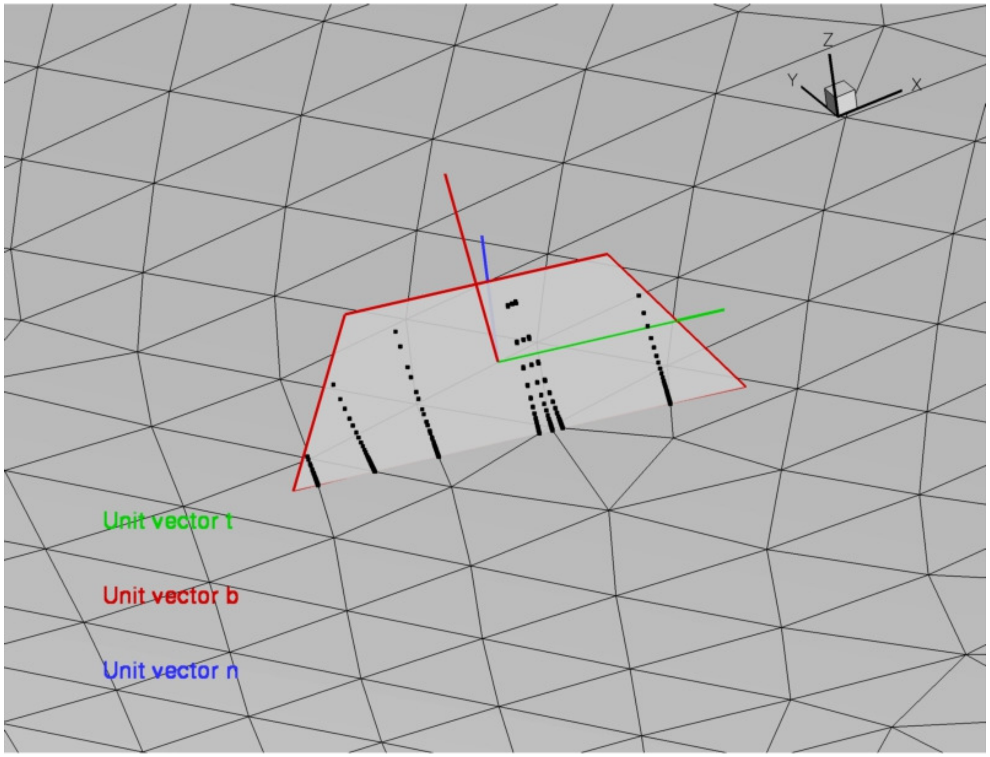
Setup Process

- Now verify that the unit vectors for the source term formulation are oriented as intended
    - Load in f6fx2b_vg_vectors.dat on top of everything so far – this contains the unit vectors $\mathbf{b}$, $\mathbf{t}$, and $\mathbf{n}$ that will be used for the formulation in the references
    - The $\mathbf{b}$ vector is uniquely defined by the local boundary orientation
    - However, the directions of $\mathbf{t}$ and $\mathbf{n}$ are guessed by FUN3D based on the freestream direction – they may be backwards if the local flow direction is substantially different. Use the namelist inputs to flip these two vectors as needed.



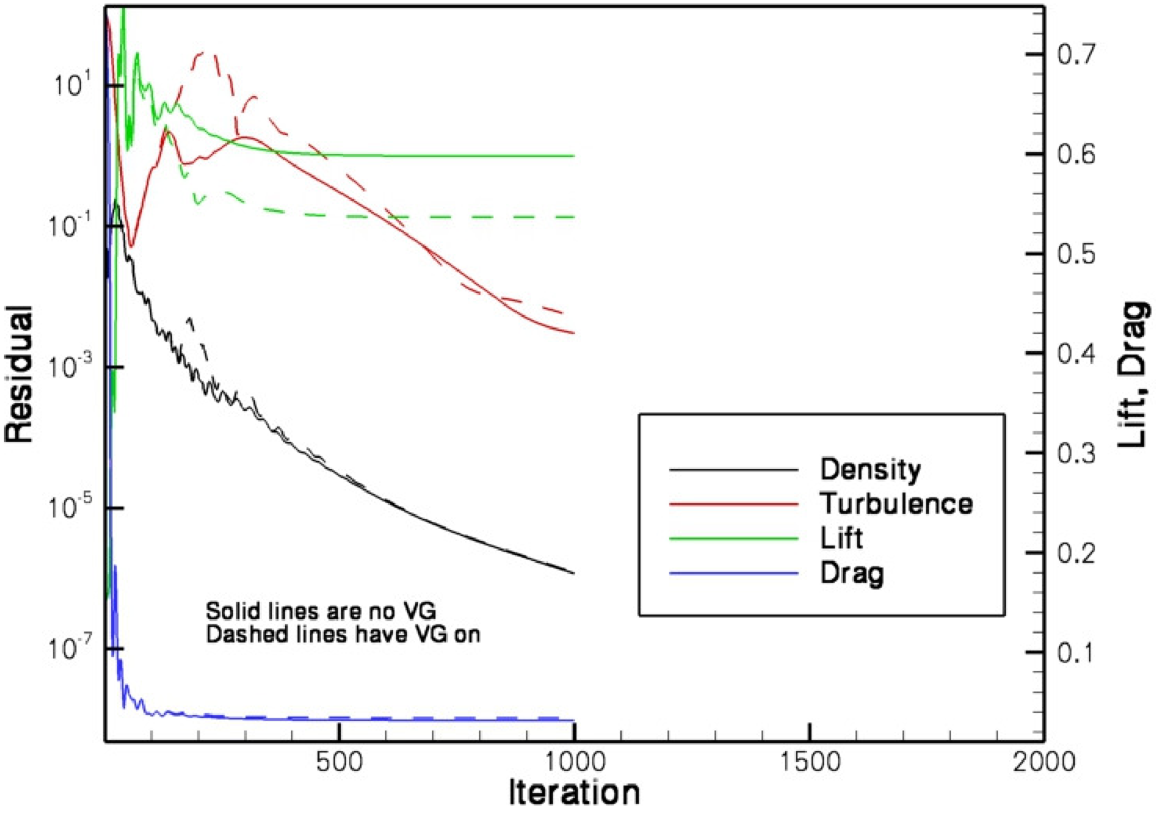
Initial Implementation of VG Source Terms in FUN3D
Setup Process

- Finally, if desired, load in f6fx2b_vg_source_locations.dat – this file can be used to scatter plot the actual locations where the source terms are computed.
- These locations are determined by intersections of grid edges with the VG geometry. The source terms are computed here and distributed to residuals at either end of the intersecting edge.



Initial Implementation of VG Source Terms in FUN3D
Execution

- Run solution to desired convergence as normal. Plot below shows convergence histories with and without VG turned on for this case.



Initial Implementation of VG Source Terms in FUN3D

Visualization

- Physical interpretation, validation left as an exercise for the reader ☺
- Would think significant mesh resolution would still be needed even to capture the effects of the modeled VG's (?)

Density contours with and without VG