Session 8:
Aeroelastic Simulations

Bob Biedron

Learning Goals

- What this will teach you
  - The two methods of aeroelastic coupling with FUN3D
    - Static coupling with an external structural solver (linear or nonlinear structures)
    - Dynamic coupling to a self contained, mode-based, linear structures model
- What you will not learn
  - Projection of forces/displacements to/from CFD and FEM
  - Structural modeling or FEM usage
- What should you already know
  - Basic steady-state, time-dependent, and dynamic-mesh solver operation and control, especially as pertains to deforming meshes
  - Basic flow visualization

Setting

- Background
  - Aeroelastic problems of interest that can be tackled with FUN3D fall into 2 general categories
    - Static: weak interaction between structures and aero; structural displacement asymptotes to a fixed level; coupling between CFD and CSD can be done infrequently - typically interested in accounting for the structural displacement on (say) cruise performance
    - Dynamic: the change in aero affects the structural deformation to the extent that there is an unsteady coupling between the two; coupling between CFD and CSD must be done frequently - prediction of flutter onset is the classic example
- Compatibility
  - Compatible with compressible flow; mixed elements; 2D/3D
- Status
  - Dynamic (flutter) analysis used fairly routinely; static much less so

Static Aeroelastic Coupling (1/4)

- Basic process
  1. Solver starts with a grid extending around a user-defined aeroelastic surface (collection of one or more grid boundaries)
  2. Solver reads in a new surface shape and deforms the mesh to fit
  3. Solver performs the requested number of iterations, and dumps out aerodynamic loads: Cp Cfx, Cfy, Cfz (dT/dn also dumped out) to a Tecplot file
  4. Middleware (*not provided*), maps the aerodynamic loads at CFD grid points onto the mesh points used by the structural solver
  5. Structural solver computes new displacements from the airloads
  6. Middleware maps structural displacements onto CFD surface, and updates a specially-named Tecplot file (next slide)
  7. Back to step 2; repeat until converged - airloads and displacements
- Jamshid Samareh of NASA Langley provides middleware (“Discrete Data transFer” - DDF) for this loads and deflection transfer

Static Aeroelastic Coupling (2/4)

- Surface file for FUN3D to read (overwritten/updated by middleware)
  - Naming convention: **`[project].body1_timestep1`**
    - Note: *must* be **`body1_timestep1`** for static aeroelastics
  - Tecplot formatted, FEPOINT style; webpage gives details, but file must contain the following variables: x, y, z, id
    - id is the index of the point in the global (raw) mesh (more later)
    - Use CLO **`--read_surface_from_file`** to have solver read file
    - Can use **`--write_massoud_file`** (more later) to generate the starting (undeformed) version of this file
- Airloads file output by FUN3D (to be read by middleware)
  - Naming convention: **`[project]_ddfdrive_bndryN.dat`**
  - Tecplot (ASCII), FEPOINT style; webpage gives details; the output variables are x, y, z, id, cp, cfx, cfy, cfz, T (temperature), and dT/dn
  - Use **`--write_aero_loads_to_file`** to have solver write this file

Static Aeroelastic Coupling (3/4)

- By default, **`--write_aero_loads_to_file`** will write one file per solid boundary; may group boundaries into one body with the following in **`fun3d.nml`**:

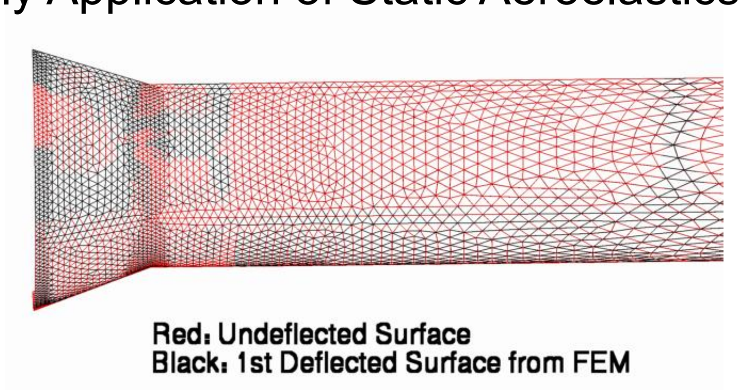
```
&aero_loads_output
n_bodies    = 1
nbndry(1)  = 3
boundary_list(1) = '5 7 9'
/
```

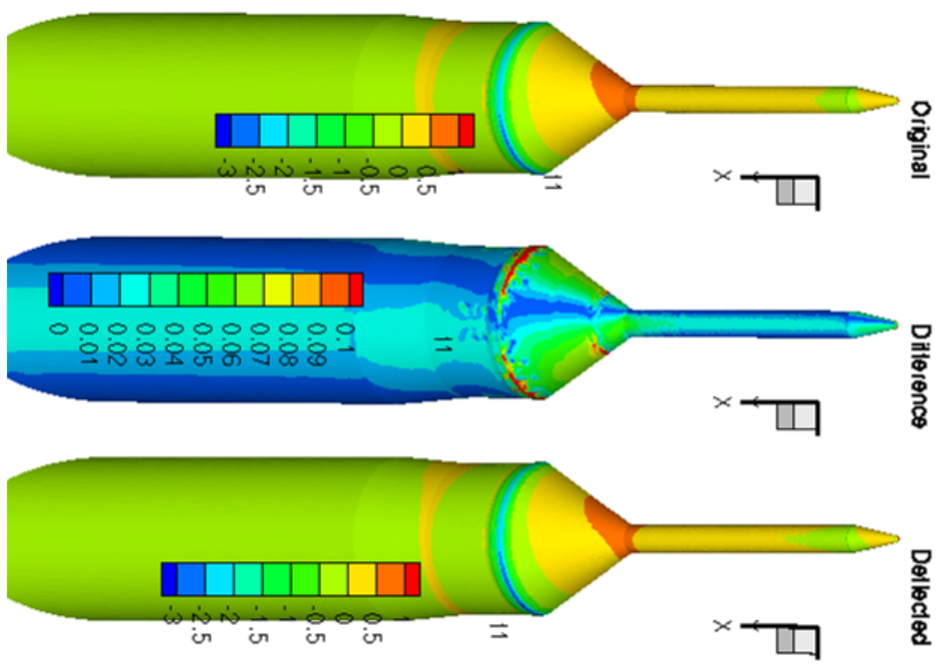
  will result in a file named **`[project]_ddfdrive_bodyM.dat`** (M=1)

- To run a loosely coupled static aeroelastic case
  - Use **`--aeroelastic_external`**
    - Turns on both **`--read_surface_from_file`** and **`--write_aero_loads_to_file`**
  - Run solver in steady-state mode
- In principle, process will work if applied every time step of a time-accurate simulation; however, current file I/O method is very inefficient for this

Static Aeroelastic Coupling (4/4)

Early Application of Static Aeroelastics: Deflection of Launch Vehicle WT Model



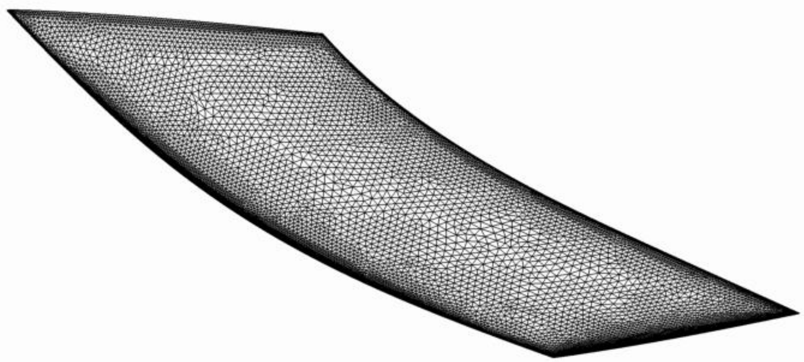


Dynamic Aeroelastic Coupling (1/6)

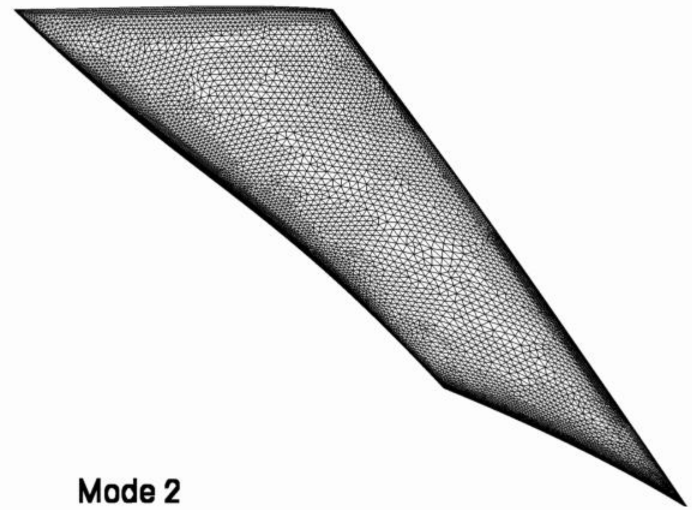
- For tightly coupled, time-accurate aeroelastic modeling, FUN3D relies on a modal decomposition approach - *same implementation as CFL3D*
  - "Small" deflections are assumed, allowing the deflected shape to be constructed as a *linear* combination of "mode shapes"- sutable for flutter *onset*, but not large deflections that occur as flutter escalates
  - A *nonlinear* aerodynamics model is used (FUN3D), so effects of shocks and viscosity can be captured
  - Structural dynamic response is decomposed into eigenmodes of known frequency (extracted a priori from an FEM model)
  - Typically only a limited set of the "important" eigenmodes is retained for dynamic aeroelastic analysis
  - Middleware (*not provided*) maps eignenmodes onto CFD surface in a one-time preprocessing step; at startup FUN3D reads these
  - Aerodynamics at current time step determine the weight applied to each eigenmode; current shape is weighted sum of eigenmodes

Dynamic Aeroelastic Coupling (2/6)

AGARD Wing 445 Mode Shapes

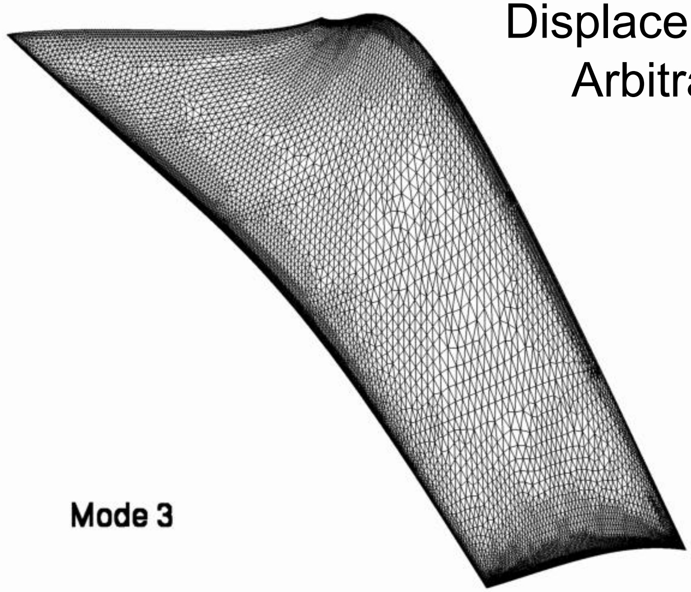


Mode 1

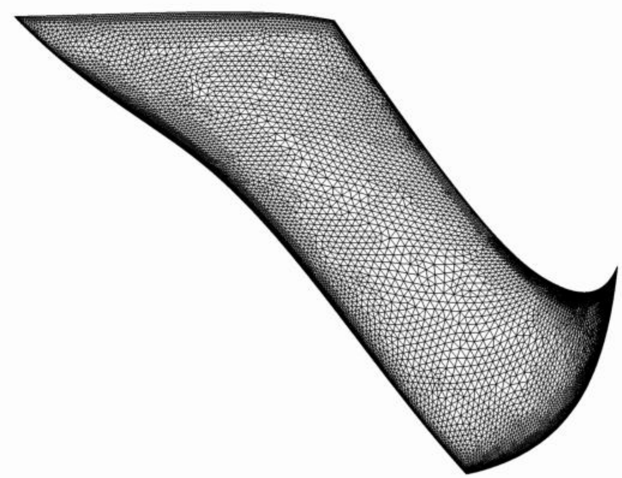


Mode 2

Vertical
Displacements
Arbitrary



Mode 3



Mode 4

Dynamic Aeroelastic Coupling (3/6)

- File nomenclature / format for mode shape input files
  - Can have multiple aeroelastic bodies (in contrast to static case) though usually only one is used
  - For each aeroelastic body, each mode shape is in a different file: **`[project].bodyB_modeM`**
  - Files are ASCII Tecplot files, similar to those input for static aeroelastic analysis, only now have modal amplitudes tacked on:

```
TITLE="wing-445.6 Mode 1"
VARIABLES= "x" "y" "z" "id" "xmd" "ymd" "zmd"
ZONE  I= 57286 ,  J= 101359 , F=FEPOINT
 0.109050E+01 -0.650348E+00 -0.294021E-01 17  0.000000E+00  0.000000E+00  0.869050E-01
 0.691189E+00 -0.650348E+00  0.000000E+00 18  0.000000E+00  0.000000E+00  0.448300E-01
 0.000000E+00  0.000000E+00  0.000000E+00 23  0.000000E+00  0.000000E+00 -0.276958E-02
 0.438172E+00  0.000000E+00 -0.322640E-01 24  0.000000E+00  0.000000E+00  0.365961E-03
```

- Can use **`--write_massoud_file`** (more later) to generate a template file with x,y,z, and id to which the middleware can append modal amplitudes

Dynamic Aeroelastic Coupling (4/6)

- Highlights of **`moving_body.input`** file – see webpage for all the details

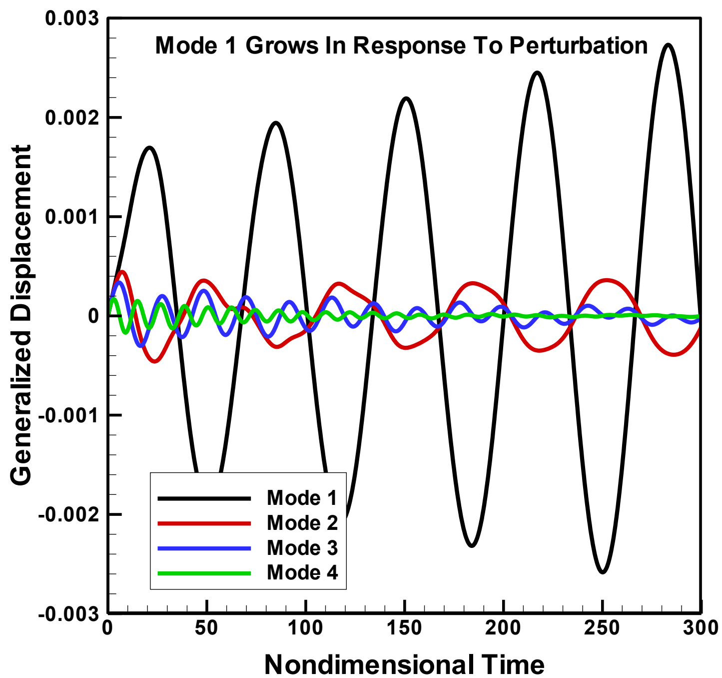
```
&body_definitions
…                        ! define bodies as collection of surfaces; b = body #
  motion_driver(b) = 'aeroelastic'
  mesh_movement(b) = 'deform'
/
&aeroelastic_modal_data  ! below, b = body #, m = mode number
  plot_modes = .true.    ! can tecplot to verify mode shapes read correctly
  nmode(b)   = 4         ! 4 modes for this body
  uinf(b)    = 973.4     ! free stream velocity (ft/s)
  grefl(b)   = 1.00      ! scale factor ft/grid unit i.e L*ref/Lref
  qinf(b)    = 89.3      ! free stream dynamic pressure, psf
  freq(m,b) = 60.3135016 ! mode frequency (rad/s)
…                        ! skip remaining 3 modes for space
  gmass(m,b)  = 0.08333  ! generalized mass (nondim)
…
  damp(m,b)   = 0.000    ! Critical damping ratio, z (nondim) (use large value
…                        ! 0.99999 to obtain static aeroelastic deflections)
  gvel0(m,b) = 0.1       ! nonzero initial velocity to kick off dynamic
…                        ! response; set = 0 on restart - don't kick me twice
 /
```

Dynamic Aeroelastic Coupling (5/6)

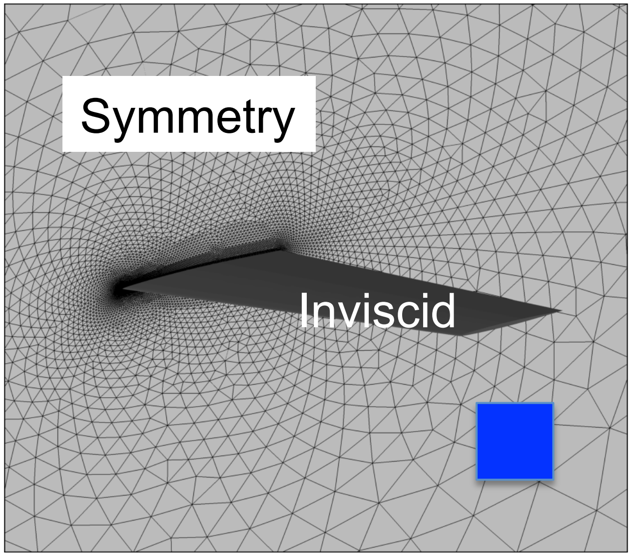
- Output of generalized force, displacement and velocity into files: **`aehist_bodyN_modeM.dat`** (ASCII Tecplot)

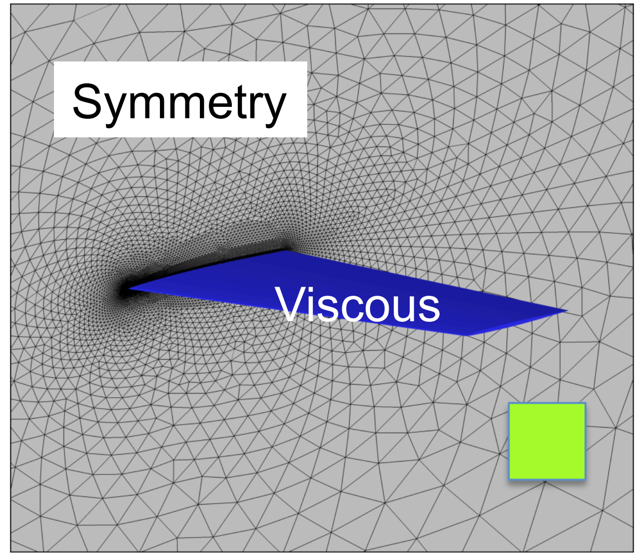
```
#  qinf =  8.93000E+01  uinf =  9.73400E+02  Mach =  9.00000E-01
variables = "time", "gdisp", "gvel", "gforce"
zone t = "modal history for airfoil, mode   1"
    0.00000E+00     0.00000E+00     1.00000E-01     0.00000E+00
    3.00000E-01     2.77139E-05     9.98227E-02    -9.81176E-02
    6.00000E-01     5.53548E-05     9.94742E-02    -8.60835E-02
...
```

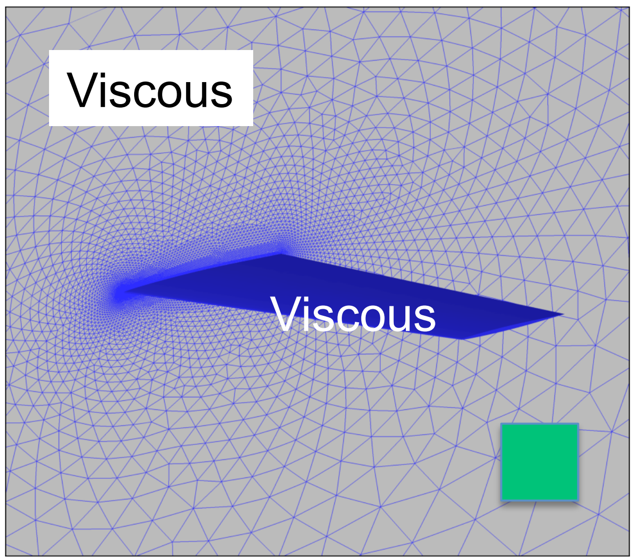
Typical plot to assess dynamic response to disturbance (initial perturbation in gvel)



Aeroelastic Analysis of AGARD 445.6 Wing





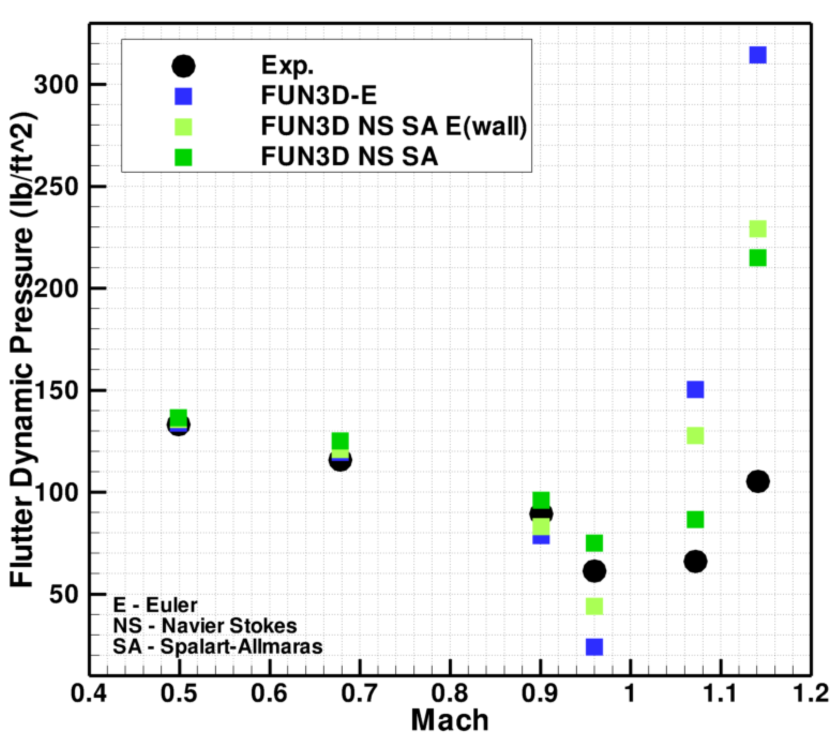


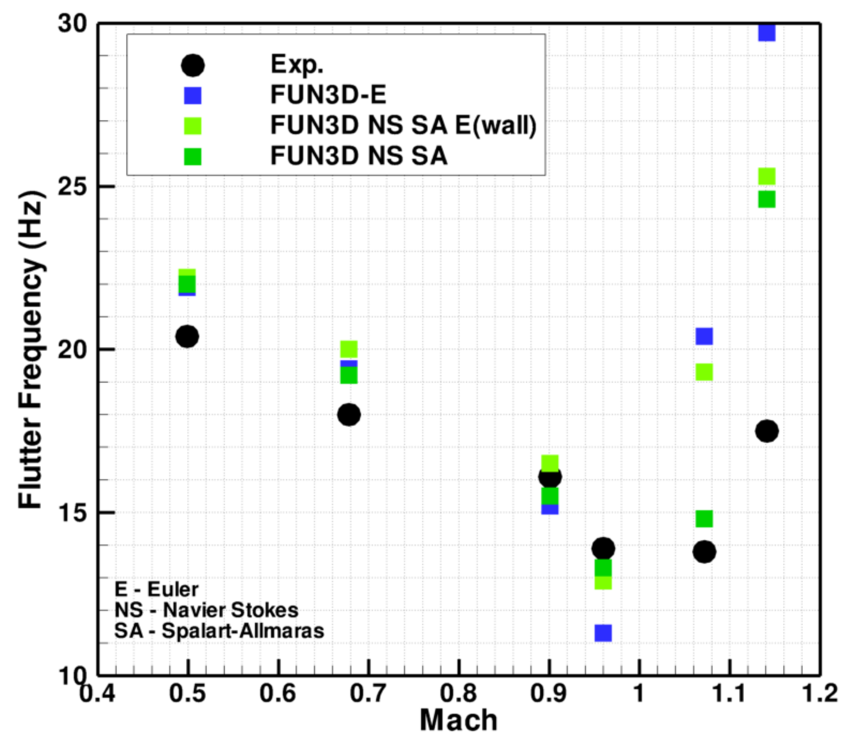
Results Courtesy
Pawel Chwalowski,
Aeroelasticity Branch
NASA Langley

Color boxes correspond to symbols on next page

Aeroelastic Analysis of AGARD 445.6 Wing

Results Courtesy Pawel Chwalowski, Aeroelasticity Branch, NASA Langley

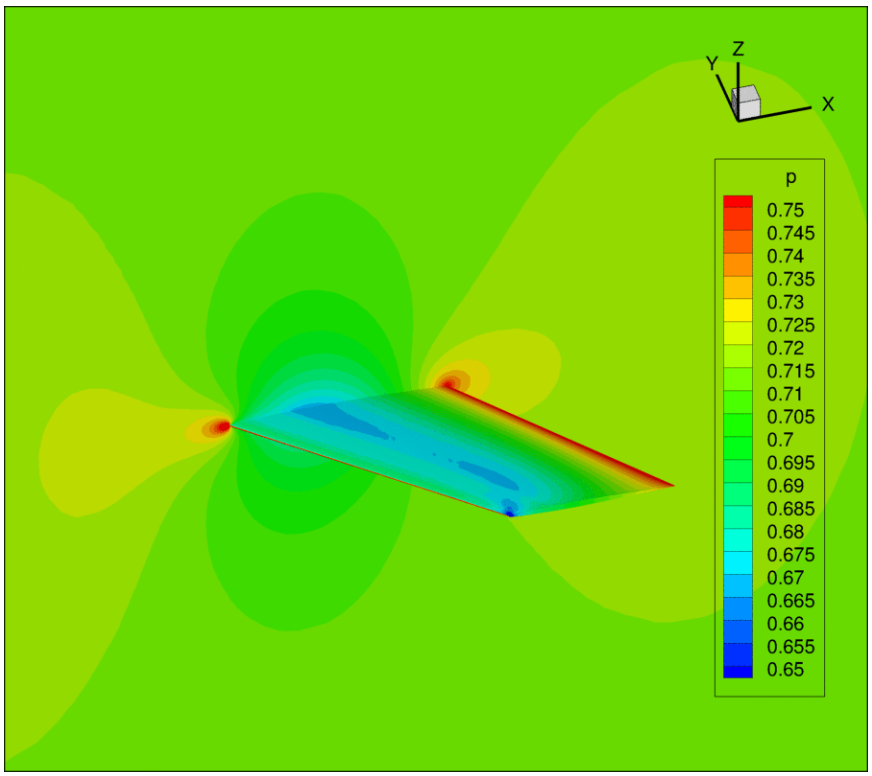


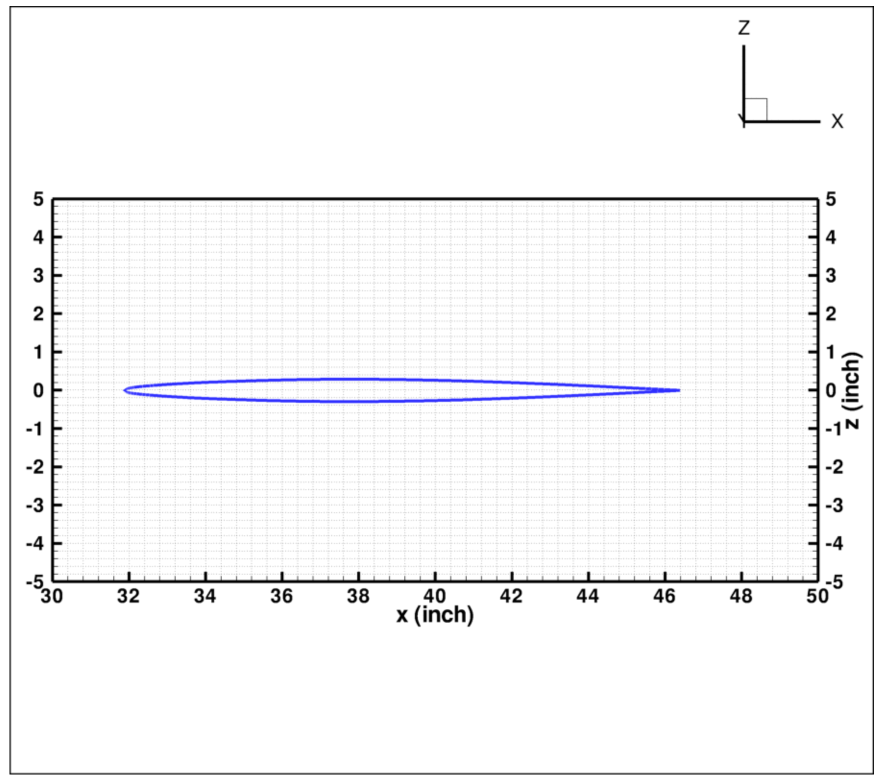


Aeroelastic Analysis of AGARD 445.6 Wing

Results Courtesy Pawel Chwalowski, Aeroelasticity Branch, NASA Langley

Inviscid Flow Mach=0.901, Flutter condition, Q = 78.6 psf





Sample Script: Mode-Shapes From NASTRAN

Samareh, J. A., "Discrete Data Transfer Technique for Fluid–Structure Interaction," 18th AIAA Computational Fluid Dynamics Conference, AIAA Paper 2007-4309, Miami, FL, June 25-28, 2007.

Script Courtesy P. Chwalowski,
NASA Langley Aeroelasticity Branch

```
#!/bin/csh
echo "-----convert source mesh to ddf -----"
./toFromDDF -nastran weak3.bdf -ddf grid1a.ddf
./toFromDDF -move moveFile grid1a.ddf -ddf grid1.ddf
echo "-----convert source solution to ddf -----"
./toFromDDF -ddf grid1.ddf -nastranModes weak3.f06 10 50 -ddf gridSolu1.ddf
echo "----- convert target mesh to ddf -----"
./toFromDDF -tecplot wing4456_massoudin.tec.1 -ddf grid2.ddf
echo "----- run ddfDrive -----"
foreach i (mode???.ddf)
/bin/cp $i gridSolu1a.ddf
./toFromDDF -move moveFile1 gridSolu1a.ddf  ddf gridSolu1.ddf
./ddfDrive < ddfDrive.input >! ddfDrive.out
./toFromDDF -ddf outputResult.ddf -tecplot wing4456.bodv1_$i.p3df
end
echo "----- END -----"
```

Geometry extraction and alignment with CFD grid

Mode shape extraction from NASTRAN

CFD surface mesh

Mode shape magnitude scaling, interpolation into CFD surface, and output

Note: toFromDDF and ddfDrive software need to be requested from J. Samareh

Creating A “MASSOUD” File For DDF

- Either approach for aeroelastic simulations requires that the surface file or mode-shape file read by FUN3D have an identifier (“`id`”) that indicates the number of that surface point in the global (raw) grid; middleware needs to preserve this id when updating surface x,y,z values
- Use the command line option **`--write_massoud_file`**
  - By default, this will result in a separate file for each solid boundary named **`[project]_massoud_bndryN.dat`** (ASCII Tecplot file)
  - In the **`fun3d.nml`** file, the user may group selected boundaries into “bodies” for output with (e.g.):

```
&massoud_output
n_bodies   = 2
nbndry(1)  = 3
boundary_list(1) = '5 7 9'
nbndry(2)  = 2
boundary_list(2) = '3 4'
/
```

which gives **`[project]_massoud_bodyM.dat`** for each body - can be used as a template for the input surface file

Additional Considerations

- Be especially careful with dimensions since at one point or another exchange must be done between CFD and FEM - need to ensure consistency! CFD meshes and FEM meshes often don't occupy the same space. DDF code is designed to handle this.
- Note that frequencies increase in the higher modes; choose time steps accordingly

List of Key Input/Output Files

- Beyond basics like **`fun3d.nml`**, **`[project]_hist.tec`**, etc.:
- Input
  - **`moving_body.input`**
  - **`[project].body1_timestep1`** (with CLO; static aeroelastic)
  - **`[project].bodyB_modeM`** (modal structures only)
- Output
  - **`aehist_bodyB_modeM.dat`** (modal structures only)
  - **`[project]_ddfdrive_bndryN.dat`** (with CLO)

Troubleshooting/FAQ

- Is there anything I can do to speed up mesh deformation?
  - Try using `--restart_deformation` - restarts deformation at each step from previous elasticity solution; probably not a good idea for large deformations, but for small deformations can be OK
- Is there anything I can do to overcome negative volumes when the grid is deformed?
  - Best: build FUN3D against Mike Park's "refine" library to bring in "untangling" capability (untangling attempted automatically)
  - Might work in a few cases: lower the `tol` parameter in the `&elasticity_gmres` namelist (but pay the price for tighter convergence every time step)

What We Learned

- Overview of two approaches for aeroelastic analysis with FUN3D
  - Steady-state loose coupling for static aeroelastic simulations – objective being to include the effect of the final deflected shape on the aerodynamics
  - Time-dependent tight coupling (with linear structural model using mode shapes) – objective typically being flutter-onset prediction
- Broad-brush treatment here; the important step of FEM to CFD data transfer only touched upon