FUN3D v14.0 Training
Actuator Disk Models

Li Wang

Learning Goals

What we will cover

- Overview of actuator-disk (AD) models for rotorcraft
- AD setup for rotorcraft simulations
- AD setup for adjoint-based sensitivity analysis & design optimization

What we will not cover

- Actuator-blade models (not functional in FUN3D)
- “First principles” articulated-blade rotorcraft simulations
  (referred to the past workshop materials)

Overview (1)

- Time-averaged representation of overall rotor influences to flowfield
  - Utilize momentum and energy equation source terms
  - Original implementation by Dave O’Brien (c.f., GIT Ph.D. dissertation)
- Lower fidelity of rotorcraft simulations compared to use of discrete blade grids
- Rotor blades assumed to be rigid (not flexible)
- Articulated-blade controls (collective, cyclic, flap, and lead/lag harmonics) can be incorporated
- Simplified grid generation
  - AD automatically embedded in computational grid
  - Any number of ADs can be modeled
  - Overset and related mechanics not needed
- Accuracy improved by grid refinement in the vicinity of AD meshes
  - Cell sizes of “background” grid should be similar to cell sizes of AD meshes

Overview (2)

- HI-ARMS implementation (SMEMRD) by Dave O'Brien ARMDEC added trim and ability to use C81 airfoil tables
  - Requires separate library, SMEMRD, not included in FUN3D distribution
- Blade-element based AD models using C81 airfoil tables have been enabled internally in FUN3D
- Efficient modeling of rotors and airframe using steady-state solvers
- Discretely consistent adjoint-based sensitivities calculated w.r.t. various AD design variables (tip speed ratio, collective pitch control angle, disk orientation angles, etc.) are available
- Constrained design optimization enabled with support of AD grouping, rotor antisymmetry enforcement, and couple/decouple properties in multipoint optimization
- Activated by the `--rotor` command line option

Actuator-Disk Loading Options

- Several loading options have been enabled, as specified by `Load Type` in the `rotor.input` file with `Rotor Type = 1`

`Load Type`	Descriptions
1	Constant Δp (specified thrust coefficient C_T)
2	Linearly increasing Δp from blade root to tip (specified C_T)
3	Blade-element based (required to specify coefficients for airfoil lift (C_l) and drag (C_d) curves; computed C_T and torque coefficient C_Q if `swirl = 1`)
4	User specified sources (not recommended)
5	C_T and C_Q radial distributions provided in a file
6	Goldstein distribution with optional swirl (specified C_T and C_Q)
7	Blade-element based (computed C_T and C_Q) • Airfoil aero properties specified via C_l and C_d curves or C81 airfoil tables • Supports radial variations of blade geometry (twist and chord length) and loading properties • Rotor trim can be achieved via FUN3D opt_driver

Setup of AD Models for rotor simulations (1)

- Activate AD models through command line option, `--rotor`
- Major input files in the run directory:
  - `fun3d.nml`
  - `rotor.input`
  - `bladegeom_rotor.dat` (required for blade-element based models, `Load Type = 7`)
  - `C81` airfoil tables if requested (note to the specific table format, referred to https://cibinjoseph.github.io/C81-Interface/page/index.html, also described in https://apps.dtic.mil/sti/pdfs/ADA108246.pdf)

Setup of AD Models for rotor simulations (2)

fun3d.nml sample:

```
&reference_physical_properties
  mach_number         = 0.6387      ! tip Mach as reference condition
  reynolds_number     = 16400000.0 ! free stream condition set by
  angle_of_attack     = 4.50        ! vinf_ratio
  temperature         = 522.81
/

&overset_data
  overset_flag        = .false.
/

&global
  moving_grid         = .false.
  recompute_turb_dist = .false.
/
```

Setup of AD Models for rotor simulations (3)

`rotor.input` – sample input available at `/path/to/fun3d/utils/Rotorcraft/.`

```
  # Rotors   Vinf_ratio   Write Soln    Force_ref    Moment_ref
         1    0.1504645          400  0.195079388   0.0975397           ! We set Vref = Vtip
=== Main Rotor =========================================================  ! Vinf_ratio = Vinf/Vref = Vinf/Vtip = AR
Rotor Type    Load Type    # Radial     # Normal  Tip Weight            ! AR = Advance ratio
         1            7         180          360         0.0
  X0_rotor     Y0_rotor    Z0_rotor         phi1        phi2        phi3  ! Force_ref and Moment_ref used to convert output
    0.7652        0.000       0.796         0.00        0.00        0.00
  Vt_ratio   ThrustCoff   PowerCoff         psi0  PitchHinge      DirRot  ! Vt_ratio = Vtip/Vref = 1.0
       1.0      0.00457       0.00.         0.00         0.0           0  ! Negative Vt_ratio indicates CW rotation
  # Blades    TipRadius  RootRadius   BladeChord   FlapHinge   LagHinge.  ! ThrustCoff, PowerCoff not effective for
         4        2.000      0.2400        0.121         0.0         0.0  ! blade-element based models
 LiftSlope   alpha, L=0         cd0          cd1         cd2
      5.79        -1.20       0.013       0.0098       0.004
    CL_max       CL_min      CD_max       CD_min       Swirl              ! Swirl = 1 adds blade drag forces/torque
      1.40        -0.50        1.50       0.0081           1
    Theta0   ThetaTwist     Theta1s      Theta1c  Pitch-Flap
      3.80         0.00       -1.12         1.81        0.00
 # FlapHar        Beta0      Beta1s       Beta1c
         1         2.50       0.000        0.000
    Beta2s       Beta2c      Beta3s       Beta3c
      0.00         0.00        0.00         0.00
  # LagHar       Delta0     Delta1s      Delta1c
         0         0.00        0.00         0.00
   Delta2s      Delta2c     Delta3s      Delta3c
      0.00         0.00        0.00         0.00
```

Key:
Required for constant/linear loading
Addt'l data for blade element or "first principles" simulations
(all items must have a value, even if unused)

Setup of AD Models for rotor simulations (4)

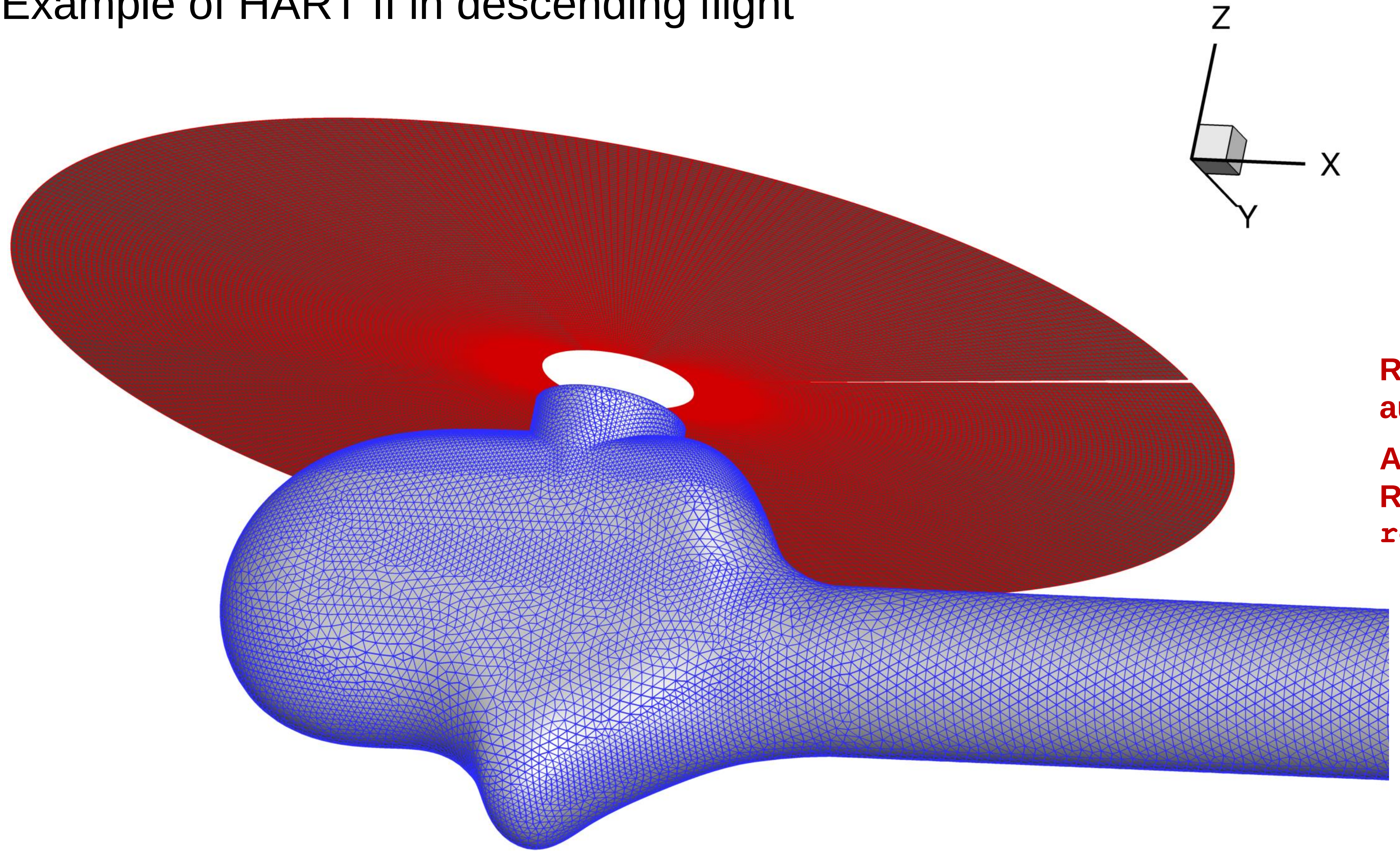
`bladegeom_rotor.dat` – sample input available at `/path/to/fun3d/utils/Rotorcraft/.`

```
===================================== Rotor Index ==========================================
1
Nstations   C81 Table
3           0
r/Rtip Twist(deg) Chord/Rtip LiftSlope a,L=0 cd0 cd1 cd2 CL_max CL_min CD_max CD_min C81 File
0.1504  0.0   0.0605   6.28   0.0   0.013   0.0   0.004   1.5   -0.5   1.5   0.0   none
0.5280  0.0   0.0605   6.28   0.0   0.013   0.0   0.004   1.5   -0.5   1.6   0.0   none
1.0000  0.0   0.0605   6.28   0.0   0.030   0.0   0.020   1.8   -0.5   1.8   0.0   none
===================================== Rotor Index ==========================================
2
Nstations   C81 Table
3           0
r/Rtip Twist(deg) Chord/Rtip LiftSlope a,L=0 cd0 cd1 cd2 CL_max CL_min CD_max CD_min C81 File
0.1504  0.0   0.0605   6.28   0.0   0.013   0.0   0.004   1.5   -0.5   1.5   0.0   none
0.5280  0.0   0.0605   6.28   0.0   0.013   0.0   0.004   1.5   -0.5   1.6   0.0   none
1.0000  0.0   0.0605   6.28   0.0   0.030   0.0   0.020   1.8   -0.5   1.8   0.0   none
```

- Rotor ordering must be consistent with the rotor ordering in `rotor.input`
- This input can only provide rotors specified with `Load Type = 7`
- Linear interpolation performed between two adjacent stations
- Set the value for `C81 Table` to 1 and provide C81 table file names accordingly
- Parameters provided in `bladegeom_rotor.dat` overwrite the inputs in `rotor.input`

Rotor Simulation Example (1)

Example of HART II in descending flight



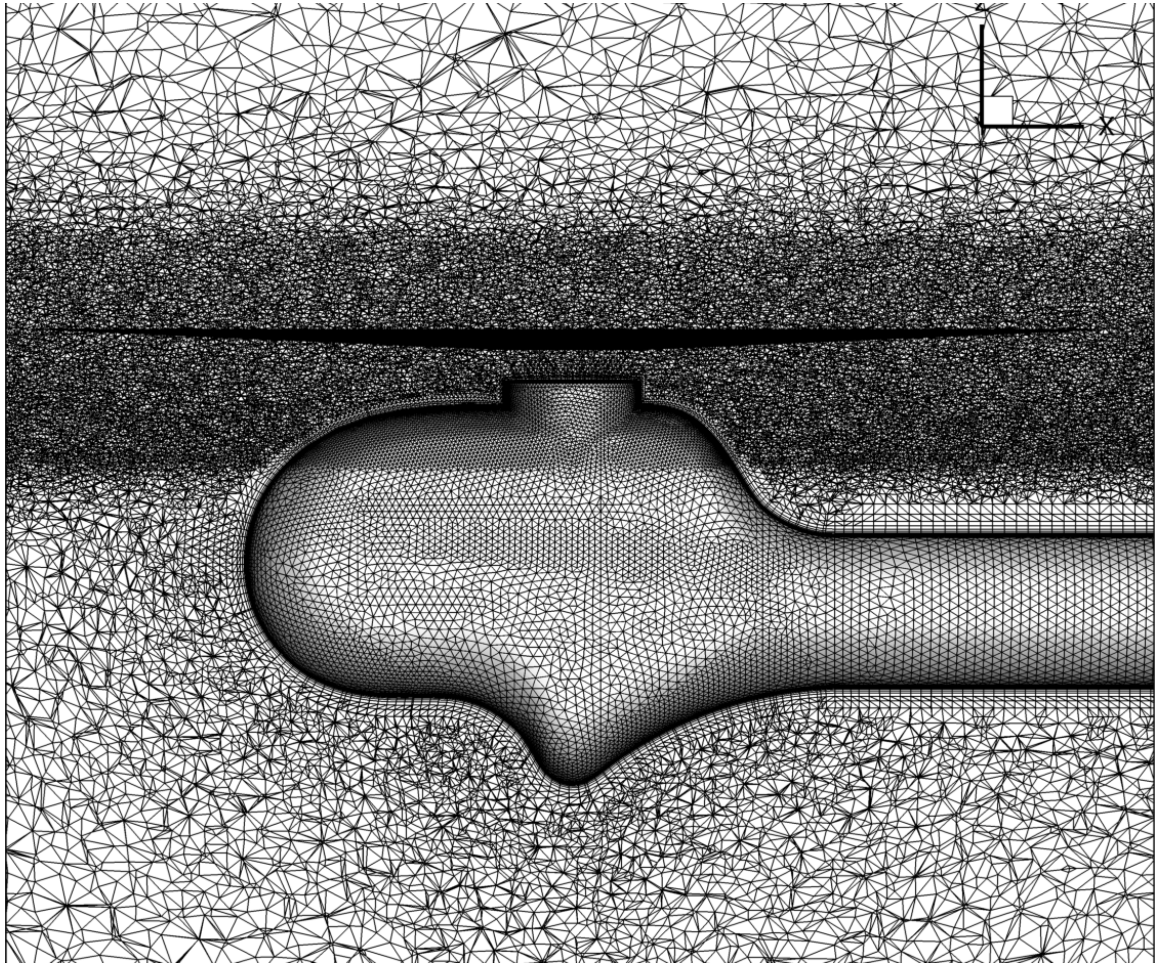
Red mesh is the AD mesh generated automatically with the solver

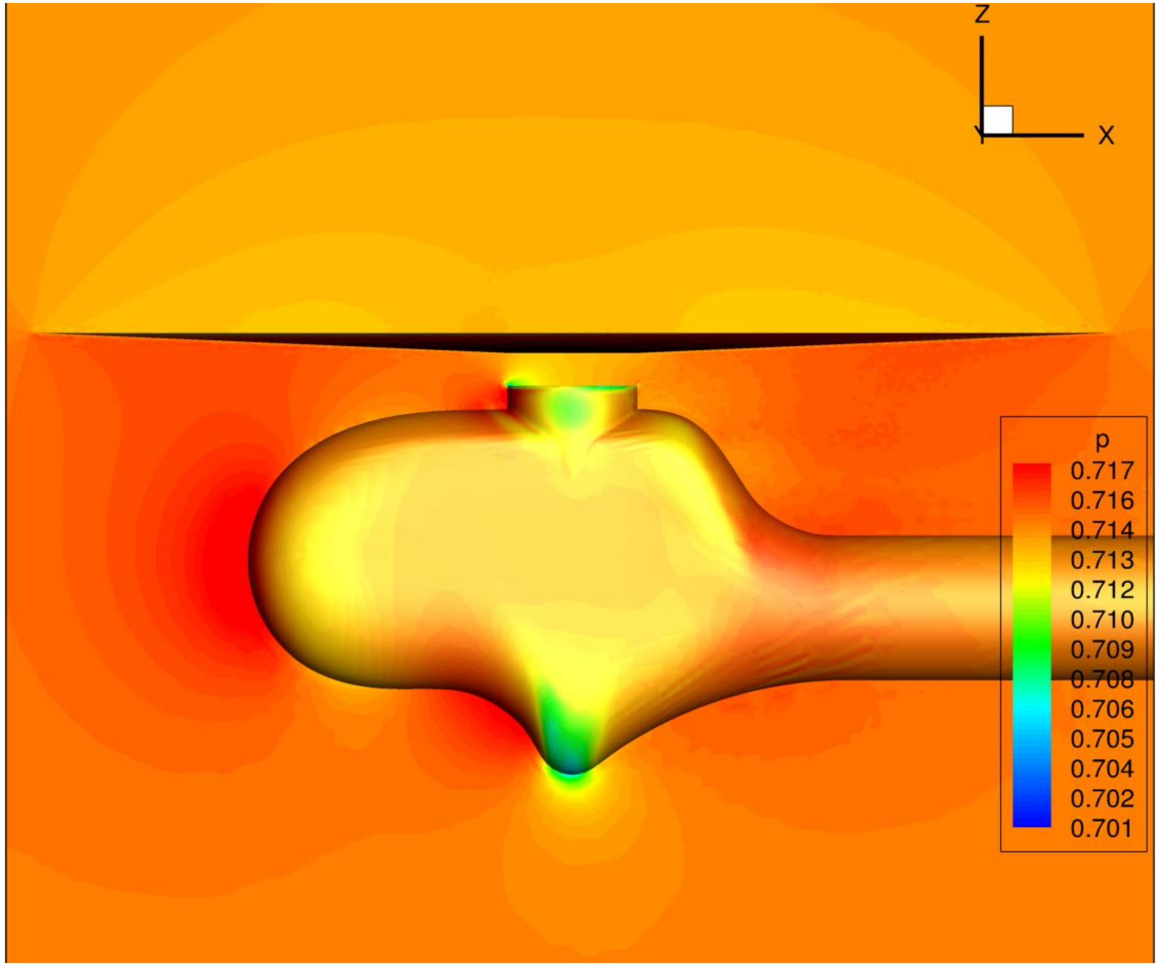
AD mesh resolution is determined by Radial X Normal specified in `rotor.input`

Blue mesh is the FUN3D input grid (mesh resolution determined in grid generation process)

Rotor Simulation Example (2)

Example of HART II in descending flight





Precone = 2.5˚

Main AD Outputs (1)

Iterative process shown in stdout of the section "`Rotor Force Summary`"

Sample:

```
    41  0.492044548019175E-05  0.18210E-02  0.13678E+02 -0.93267E+00  0.63539E+00
        0.171543563838665E-02  0.42647E+00  0.16816E+02 -0.22321E+01 -0.13209E+01
        41 HANIM 1 Prec(S mf 10 0.188E+00 t 50 0.175E+00) GCR(S 8  0.181E+00) Realz(S) Contrl(F) CFL 0.137E+03
        41 HANIM 2 Prec(S mf 10 0.109E+00 t 10 0.102E+00) GCR(S 2  0.155E+00) Realz(S) Contrl(S) CFL 0.275E+03
        Lift  0.569091141911458E-02         Drag  0.758553275679323E-02
 Rotor Force Summary:
 Rotor  1   Grid Forces:Fx=  2.7816E-04 Fy -1.7830E-04 Fz  5.2420E-03
 Rotor  1  Grid Moments:Mx=  1.3421E-03 My -1.1875E-04 Mz -3.2152E-04
 Rotor  1  Shaft Forces:H =  2.7816E-04 Y  -1.7830E-04 T=  5.2420E-03
 Rotor  1 Shaft Moments:Mh=  1.3421E-03 My -1.1875E-04 Q= -3.2152E-04
    42  0.393132417640741E-05  0.22027E-02  0.13678E+02 -0.93267E+00  0.63539E+00
        0.155774141679910E-02  0.41891E+00  0.14415E+02 -0.47352E+00  0.57841E+00
        42 HANIM 1 Prec(S mf 5 0.197E+00 t 10 0.185E+00) GCR(S 3  0.152E+00) Realz(S) Contrl(S) CFL 0.550E+03
        Lift  0.633122993942016E-02         Drag  0.569688957133075E-02

 Rotor Force Summary:
 Rotor  1   Grid Forces:Fx=  2.6332E-04 Fy -1.5322E-04 Fz  4.9658E-03
 Rotor  1  Grid Moments:Mx=  1.1135E-03 My -5.1323E-05 Mz -3.1536E-04
 Rotor  1  Shaft Forces:H =  2.6332E-04 Y  -1.5322E-04 T=  4.9658E-03
 Rotor  1 Shaft Moments:Mh=  1.1135E-03 My -5.1323E-05 Q= -3.1536E-04
```

Rotor load components are output in the Cartesian coordinate system (FUN3D body frame) and the shaft frame

Final disk loads output in `[project_rootname]_ad_loading.dat`

Main AD Outputs (2)

Iterative process shown in stdout of the section "`Rotor Force Summary`"

Final disk loads output in `[project_rootname]_ad_loading.dat`

Force ref	Moment ref	Rotor Force Summary
1.0	1.0	Standard FUN3D nondimensional output for total disk forces and moments
$(L_{ref}^2 a_{ref}^2)/(\pi R_{tip}^2 V_{tip}^2)$	$(L_{ref}^3 a_{ref}^2)/(\pi R_{tip}^3 V_{tip}^2)$	Standard rotorcraft nondimensionalization • L_{ref} is the ratio of physical characteristic length to the corresponding grid units • a_{ref} is the nondimensional speed of sound in FUN3D (unity) • R_{tip} is the blade tip radius in CFD grid • V_{tip} is the nondimensional tip velocity, which is equal to the tip Mach number
$\rho_{ref}^* (a_{ref}^*)^2 L_{ref}^2$	$\rho_{ref}{}^* (a_{ref}^*)^2 L_{ref}^3$	Dimensional quantities of forces and moments • L_{ref} is the ratio of physical characteristic length to the corresponding grid units • a_{ref}^* is the dimensional speed of sound (e.g., m/s, ft/s) • ρ_{ref}^* is the dimensional air density (e.g., kg/m^3, slug/ft^3)

Main AD Outputs (3)

The disk geometry and loading are output in plot3d format in files:

`source_grid_iteration#.p3d` and **`source_data_iteration#.p3d`**

respectively, where **#** is the nonlinear iteration index (output freq. controlled by `Write_Soln` input parameter in `rotor.input`).

Note that in V14.0, the plot3d formatted files are also output at the end of simulation

`source_data_iteration#.p3d`

`Load Type 1-2 and 4-6`

F1V1	F_x
F1V2	F_y
F1V3	F_z
F1V4	*Area*

\+

`Load Type 3 and 7` (with additional variables)

F1V5	α_{eff}	F1V10	v_{vel}
F1V6	C_l	F1V11	w_{vel}
F1V7	C_d	F1V12	ρ
F1V8	θ	F1V13	P
F1V9	u_{vel}	F1V14	P_o

Rotor Optimization with AD models (1)

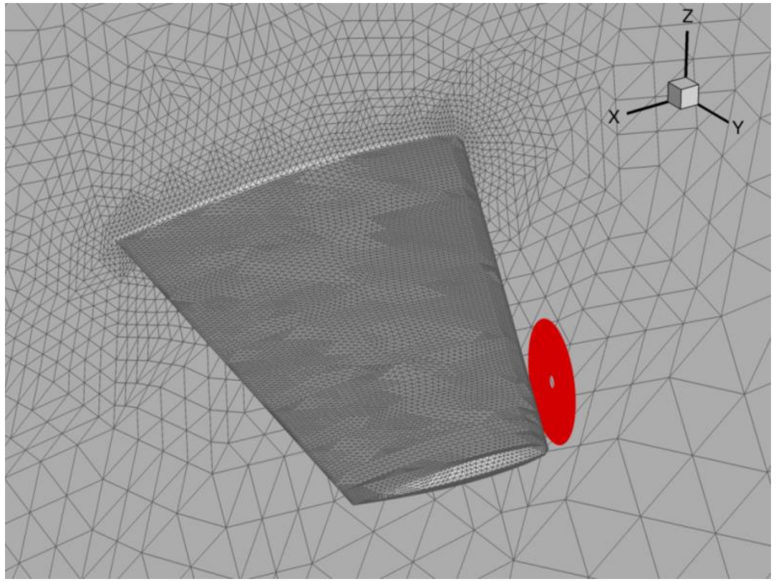
- Supported AD models: constant/linear disk loading (`Load Type = 1` or `2`) and blade-element disk loading (`Load Type = 7`)
- Currently supported objective/constraint functions

Keyword	Function
ad_power	disk power
ad_propeff	propeller efficiency
ad_x_force, ad_y_force, ad_z_force	x-/y-/z-component force
ad_xmoment, ad_ymoment, ad_zmoment	x-/y-/z-component moment

Rotor Optimization with AD models (2)

- Verification of adjoint-based sensitivities to AD design variables

Design Parameter	Output Function		
	x-component Force	Rolling Moment	Pitching Moment
α	A: -0.00404991653417 C: -0.00404991653418	A: 0.24959252000819 C: 0.24959252002799	A: 0.00221866040857 C: 0.00221866040858
ϕ_2	A: -0.00000300774859 C: -0.00000300774859	A: -0.00014611538523 C: -0.00014611538523	A: 0.00000040831518 C: 0.00000040831518
x_0	A: 0.00537999079906 C: 0.00537999079905	A: -0.01974404055856 C: -0.01974404055961	A: 0.00012710885914 C: 0.00012710885914
θ_0	A: -0.00047607050407 C: -0.00047607050407	A: 0.00072200644688 C: 0.00072200644690	A: 0.00000374029027 C: 0.00000374029027
V_{tip}	A: -0.01920622203027 C: -0.01920622203027	A: 0.01748002461877 C: 0.01748002461918	A: 0.00004176550665 C: 0.00004176550665
R	A: -0.10823285939218 C: -0.10823285939218	A: 0.02204710317501 C: 0.02204710317086	A: -0.00144530841724 C: -0.00144530841724
t_w	A:-0.00027889375655 C:-0.00027889375655	A: 0.00042065745972 C: 0.00042065745973	A: 0.00000134609412 C: 0.00000134609412
c	A: -0.04369278291250 C: -0.04369278291250	A: 0.03365913349219 C: 0.03365913349237	A: -0.00018097832250 C: -0.00018097832250
Shape	A: -0.00022033285511 C: -0.00022033285511	A: 0.00388294970217 C: 0.00388294970237	A: 0.00013344869809 C: 0.00013344869809



Adjoint (A) and complex-step (C) sensitivities

Rotor Optimization with AD models (3)

- Design directory structure (setup referred to previous workshop materials)

ammo

- ***opt_driver***
- *design.nml*

description.1

- *fun3d.nml*
- *rotor.input*
- *bladegeom_rotor.dat*
- *rubber.data*
- *actuator_disk.data*
- *actuator_disk_group.data*
- *body_group.data*
- *project.lb8.ugrid*
- *project.mapbc*

model.1

Rubberize

- ***rutro***
- *surface_history/*
- *massoud.#*

Adjoint

- ***dual_mpi***
- *links to c81 tables if requested*

Flow

- ***nodet_mpi***
- *c81 tables if requested*

- Adjoint sensitivities alone can be computed by executing `dual_mpi` after design files are properly populated and flow solve is complete

Inputs for AD Optimization (1)

- `rubber.data` – define design outputs and global/shape design variables

```
########################################################################
############################ Function Information ######################
########################################################################
Number of composite functions for design problem statement
    5
########################################################################
Cost function (1) or constraint (2)
    1
If constraint, lower and upper bounds
              0.000000000000000              0.000000000000000
Number of components for function   1
    1
Physical timestep interval where function is defined
    1     1
Composite function weight, target, and power
         1.000000000000000         0.000000000000000         1.000000000000000
Components of function   1: boundary id (0=all)/name/value/weight/target/power
    0 ad_power   0.0000000000000000E+00   0.1000000000000000E+01   0.0000000000000000E+00   0.2000000000000000E+01
```

- Components of function
  - Value 0 indicates AD power calculated from all disks
  - Individual rotors can be set by specifying disk indices

Inputs for AD Optimization (2)

- `rubber.data` – define design outputs and global/shape design variables

```
########################################################################
########################### Function Information #######################
########################################################################
Number of composite functions for design problem statement
    5
########################################################################
Cost function (1) or constraint (2)
    1
If constraint, lower and upper bounds
              0.000000000000000              0.000000000000000
Number of components for function   1
    2
Physical timestep interval where function is defined
    1     1
Composite function weight, target, and power
         1.000000000000000        0.000000000000000        1.000000000000000
Components of function   1: boundary id (0=all)/name/value/weight/target/power
    1 ad_power    0.0000000000000000E+00    0.1000000000000000E+01    0.0000000000000000E+00    0.2000000000000000E+01
    2 ad_power    0.0000000000000000E+00    0.2000000000000000E+01    0.0000000000000000E+00    0.2000000000000000E+01
```

- Components of function
  - Value 0 indicates AD power calculated from all disks
  - Individual rotors can be set by specifying disk indices

Inputs for AD Optimization (3)

- `actuator_disk.data` – setting for AD design variables (sample provided in `Adjoint` directory)

```
##############################################################################
########################## Design Variable Information #######################
##############################################################################
Number of rotors
    2
Number of variables for rotor 1
    18
Index Active         Value                 Lower Bound              Upper Bound
    1     0   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    2     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    3     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    4     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    5     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    6     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    7     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    8     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    9     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
   10     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
   11     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
   12     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
   13     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
   14     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
   15     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
   16     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
   17     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
   18     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
Number of variables for rotor 2
    18
Index Active         Value                 Lower Bound              Upper Bound
    1     0   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    2     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    3     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
    4     1   0.100000000000000E+01  0.000000000000000E+00  0.500000000000000E+01
```

[... more in the file not shown]

Active	Design Variable Property
0	Inactive. Values unchanged in optimization
1	Active. Synced for grouped disks if requested
2	Active. Antisymmetry enforced for grouped disks
-1	Active. **Negative values** indicate that DVs are **decoupled** among design points for multipoint optimization
-2	Active. Antisymmetry enforced for grouped disks. Decoupled among design points for multipoint optimization

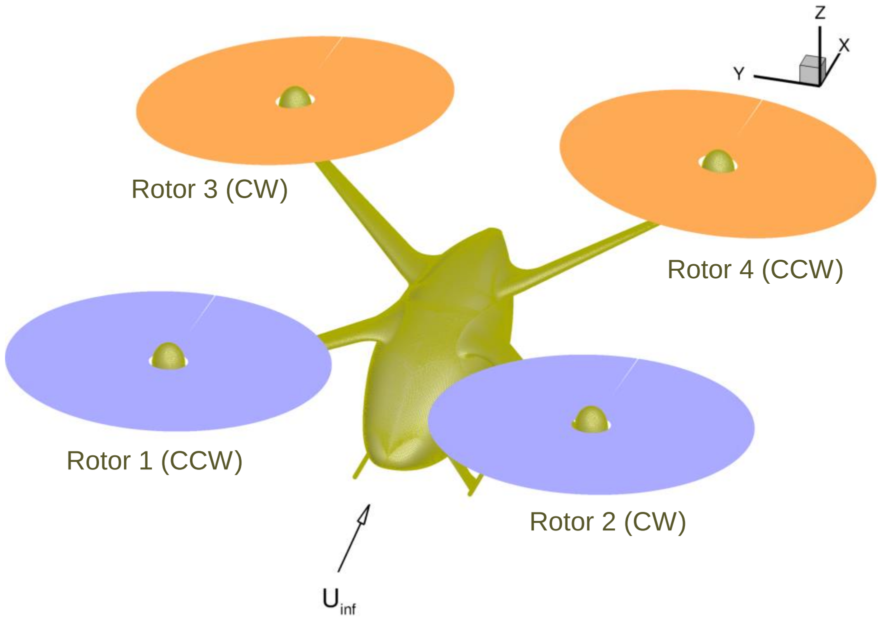
Index	Design Variables
1	Thrustcoeff, C_T
2 – 4	AD orientation angles, phi1, phi2, phi3
5	Tip speed ratio, Vtip_ratio
6	Blade radius, R
7 – 9	AD center positions, X0, Y0, Z0
10	Collective pitch angle
11 – 10+n	Twist angle at specified stations (n = # stations)
11+n – 10+2n	Chord length at specified stations

Inputs for AD Optimization (3)

- `actuator_disk_grouping.data` – identical format to body_grouping.data
  - Used to group ADs in which the AD design variables within the same group are changed uniformly
  - Adjoint sensitivities for group disks are combined as composite derivatives

```
Number of groups to create
2
Number of rotors in group, list of rotors
2
1 2
2
3 4
```

- Rotor index in each group must be consistent with the rotor index defined in `rotor.input`



Example: Quadrotor VTOL configuration

Sensitivity Output in Adjoint Run

- Adjoint-based sensitivities to AD design variables provided at the *end* of stdout in adjoint run

```
Reduced Ct derivative            =    0.00000000000000E+00
Reduced phi1 derivative          =    0.50087363835414E-01
Reduced phi2 derivative          =    0.87934299303692E+00
Reduced phi3 derivative          =    0.32227307548521E+00
Reduced Vt derivative            =    0.63961616887175E+02
Reduced rtip derivative          =    0.12525358935745E+01
Reduced x0 derivative            =   -0.62486683192166E-01
Reduced y0 derivative            =    0.52041389267626E+00
Reduced z0 derivative            =   -0.91451056022092E+00
Reduced theta0 derivative        =    0.82505862467155E+01
Reduced twist derivative at station 1 =    0.15763442291520E+01
Reduced twist derivative at station 2 =    0.44955338129188E+01
Reduced twist derivative at station 3 =    0.15546790622676E+01
Reduced twist derivative at station 4 =    0.62402914237695E+00
Reduced normalized chord derivative at station 1 =    0.32374903083626E+02
Reduced normalized chord derivative at station 2 =    0.62410402053591E+02
Reduced normalized chord derivative at station 3 =    0.28369166382044E+01
Reduced normalized chord derivative at station 4 =   -0.20751434804672E+01
Reduced Ct derivative            =    0.00000000000000E+00
Reduced phi1 derivative          =    0.23923830357712E-05
Reduced phi2 derivative          =   -0.66747086822248E-04
Reduced phi3 derivative          =   -0.98459777061071E-06
Reduced Vt derivative            =   -0.17615187563103E-02
Reduced rtip derivative          =   -0.12101755755192E-03
Reduced x0 derivative            =   -0.93280202720623E-04
Reduced y0 derivative            =   -0.35257413080485E-06
Reduced z0 derivative            =    0.96348506757503E-06
```

[...]

Adjoint sensitivities are output for each design output function, in the order consistent with the function definitions in `rubber.data`

What We Learned

- Overview of actuator-disk (AD) models for rotorcraft
  - Lower fidelity rotor modeling
  - Various load options
- AD setup for rotorcraft simulations in FUN3D
  - Input decks for blade-element AD models
  - Main AD outputs
- General AD setup for adjoint sensitivities and gradient-based optimization
  - Directory structures
  - Additional input decks for AD models used in optimization
  - Setup AD design variables