

Computational Aircraft Prototype Syntheses



Training Session 5 Geometry Analysis Views ESP v1.22

Marshall Galbraith

galbramc@mit.edu

Massachusetts Institute of Technology

Bob Haimes

haimes@mit.edu

John F. Dannenhoffer, III

jfdannen@syr.edu

Syracuse University

- Multi-analysis Models
 - transport.csm Example
 - Geometric Analysis Views
 - Attribution Views
 - Coupled Analysis View
- Suggested Exercises

Single-analysis Models

- Only single analysis geometric models considered so far:
f118-C.csm → masstranAIM
naca.csm → xfoilAIM
- Different parameterizations for each model

Multi-analysis Models

- Single model parameterizations for multi-analysis
avlAIM
transport.csm → su2AIM
astrosAIM
- Each analysis requires different geometric representation

ESP/fusePmtrs.udc

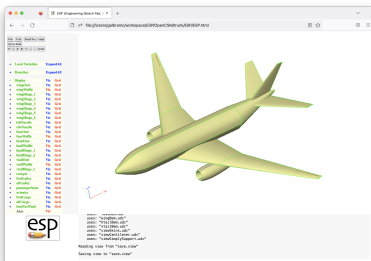
DESPMTR	fuse:fwdLength	28	# length of forward fuselage
DESPMTR	fuse:noseHeight	0	# zloc of center of nose

ESP/wingPmtrs.udc

DESPMTR	wing:area	4240	# area
DESPMTR	wing:aspect	9.00	# aspect ratio

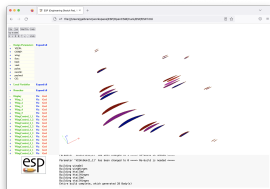
ESP/htailPmtrs.udc

DESPMTR	htail:vc	0.55	# htail volume coefficient
DESPMTR	htail:length	85	# distance between htail root and wing root

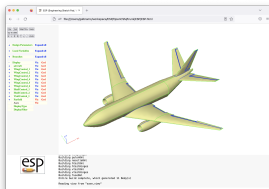


- Views construct analysis specific geometry
- Implemented as user-defined components (UDCs)
 - `viewVlm.udc` → `avlAIM`
 - `transport.csm` → `viewCfdViscous.udc` → `su2AIM`
 - `viewCantilever.udc` → `astrosAIM`
- View attributes geometry with suitable CAPS attributes

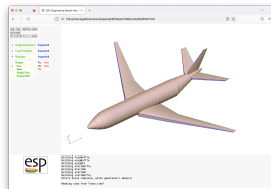
ESP/transport.csm



ESP/viewVlm.udc



ESP/viewCfdViscous.udc



ESP/viewCantilever.udc

- Switches for VIEWs (ESP/transport.csm)

```
# define the views
CFGPMTR  VIEW:Concept      1
CFGPMTR  VIEW:Vlm          0

CFGPMTR  VIEW:CfdInviscid   0
CFGPMTR  VIEW:CfdViscous    0
```

- Switches for COMPONENTs (ESP/transport_init.udc)

```
# make a list of the components
CFGPMTR  COMP:wing          1
CFGPMTR  COMP:fuse          1
CFGPMTR  COMP:htail         1
CFGPMTR  COMP:vtail         1
```

- Design Parameters: fusePmtrs.udc, wingPmtrs.udc...
- Compute derived quantities: fuseCalc.udc, wingCalc.udc...

Models

- Geometric models, created by combinations of primitives
 - wingVlm.udc – cross-sectional cuts
 - wingCfdInviscis.udc – outer mold line, deflected control surfaces
 - wingCfdViscous.udc – outer mold line, free-flying control surfaces
 - wingBem.udc – built-up element model, built by intersecting a waffle with OML
- Assign CAPS attributes using attribute “tags”

Primitives

- Lowest-level geometries
 - wingOml.udc – outer mold line
 - wingWaffle.udc – arrangement of spars and ribs
 - wingHinges.udc – location of hinge lines for control surfaces

Component, Tag, and Index Attributes

- Attributes placed on pimitives during construction
- FACEs
 - tagComp with value \$leftWing or \$riteWing
 - tagType with value \$tip, \$upper, \$lower, or \$trailingEdge
 - tagIndex with value \$1 or \$2
- EDGEs
 - tagComp with value \$leftWing or \$riteWing
 - tagType with value \$root, \$leadingEdge or \$trailingEdge

CAPS Attributes

- Attributes used in views to select entities for CAPS attribution
- “tag” attributes simplify otherwise complex selections

- “tag” attributes on primitives

ESP/wingOml.udc

```
RULE
  ATTRIBUTE tagComp $leftWing
```

```
SELECT    FACE ruledBody 2
  ATTRIBUTE tagType $tip
  ATTRIBUTE tagIndex $1
```

```
SELECT    FACE ruledBody 5
SELECT    ADD ruledBody 6
  ATTRIBUTE tagType $lower
```

ESP/wingWaffle.udc

```
# spars
POINT M ON GH XSECT KL
LINE MH M H tagType=spar tagIndex=1
```

- CAPS attributes on model

ESP/viewCantilever.udc

```
# define point load location(s)
UDPRIM editAttr filename <<
  NODE ADJ2FACE tagComp=leftWing tagType=tip
  AND ADJ2FACE tagType=lower
  AND ADJ2FACE tagType=spar tagIndex=1
  SET capsLoad=leftPointLoad

  NODE ADJ2FACE tagComp=riteWing tagType=tip
  AND ADJ2FACE tagType=lower
  AND ADJ2FACE tagType=spar tagIndex=1
  SET capsLoad=ritePointLoad
>>
```

- Coupled analysis requires multiple simultaneous analysis geometries
- Achieved with multiple active views

CFGPMTR VIEW:Cantilever 1



- Training UDC views are flexible, but not universal
 - Designed for ESP/transport.csm
- Views are a powerful method for organizing multi-analysis geometry
 - Views should be customized for projects
- More details about the transport.csm in **ESP** training session10

Transport Views

- Use the ESP GUI (not editor) with ESP/transport.csm to:
 - Toggle each view one at a time

transport Views

- Using the ESP GUI (not the editor), toggle the views:

`VIEW:Concept` `0`

`VIEW:SimplySupport` `1`

- Toggle the **Viz** for the Nodes to turn them on
- Use the ESP GUI DisplayFilter to inspect the attribute names:
 - `capsGroup`
 - `capsConnect`
 - `capsConnectLink`
 - `capsLoad`

- Create your own (optionally share it galbramc@mit.edu)