Computational Aircraft Prototype Syntheses

caps

Training Session 7
Meshing for CFD II: Pointwise

ESP v1.19

Marshall Galbraith galbramc@mit.edu
Bob Haimes haimes@mit.edu
Massachusetts Institute of Technology

John F. Dannenhoffer, III
jfdannen@syr.edu
Syracuse University

Overview

- Pointwise and GeomToMesh.glf
- Pointwise invsicid meshing
  - Suggested parameters
  - Proximity detection
- Pointwise viscous meshing
  - Suggested parameters
  - Viscous boundary layer mesh generation
- Suggested Exercises

Pointwise Overview

- Pointwise: commercial software available for Windows, Linux, and Mac.
- General purpose mesh generator for high quality structured, unstructured, and hybrid meshes
- T-Rex extrusion for generation of boundary layer resolving hybrid meshes.
- Glyph: a TCL/TK based scripting language to automate repetitive tasks.

GeomToMesh.glf Overview

- GeomToMesh.glf: glyph script for creating unstructured volume meshes given a clean EGADS geometry file.
- Vision: script to automatically generate a valid, high quality unstructured mesh given clean, closed geometry.
- Using geometry attributions, the scripts will attempt to create a mesh that matches the user's intent.
- Source Box for increased viscous off body resolution
- Elevate on Export generates higher-order curved meshes (not yet in CAPS)

Pointwise Terminology

- Edge: Line connecting two vertexes

Connector

- A collection of mesh vertexes on an `EDGE`.
- `CAPS` requires only one connector per `EDGE`.
- Pointwise supports multiple connectors per `EDGE`.

Domain

- Surface elements on an `FACE` bounded by a set of connectors
- `CAPS` requires one domain per `FACE`
- Pointwise supports “Quilting” where one domain spans multiple `FACES`

Block

- Volume region bounded by a set of domains

Pointwise AIM Inputs and Outputs

Pointwise AIM Documentation

Transport Geometry EGADS Files

- Use pyCAPS to export geometry to EGADS files
- Explore meshing parameters without rebuilding geometry
- DANGER: Decouples geometric and analysis parameters
  - getGeometryVal and getGeometryOutVal are read only

Execute: EGADS/egadsCFD.py

```
# Change to Inviscid CFD view
transport.cfgpmtr.VIEW.Concept     = 0
transport.cfgpmtr.VIEW.CFDInviscid = 1
transport.cfgpmtr.VIEW.CFDViscous  = 0

# Enable just wing
transport.cfgpmtr.COMP.Wing    = 1
transport.cfgpmtr.COMP.Fuse    = 0
transport.cfgpmtr.COMP.Htail   = 0
transport.cfgpmtr.COMP.Vtail   = 0
transport.cfgpmtr.COMP.Pod     = 0
transport.cfgpmtr.COMP.Control = 0

# Save egads file of the geometry
print("==> Generating CFDInviscid_Wing")
transport.save("CFDInviscid_Wing.egads")
```

CFDInviscid_Wing.egads
CFDInviscid_WingPod.egads
CFDInviscid_Transport.egads
CFDViscous_Wing.egads
CFDViscous_WingPod.egads
CFDViscous_Transport.egads

Pointwise Execution

- Pointwise GeomToMesh.glf script available via $CAPS_GLYPH environment variable
- Windows: also uses PW_HOME environment variable
- Try multiple times in case server license is not available

session07/pointwise_01_Defaults.py

```
####### Run pointwise #################
currentDirectory = os.getcwd()   # Get current working directory
os.chdir(pointwise.analysisDir)  # Move into test directory

CAPS_GLYPH = os.environ["CAPS_GLYPH"]
for i in range(60):
    if "Windows" in platform.system():
        PW_HOME = os.environ["PW_HOME"]
        os.system(’"’ + PW_HOME + ’\\win64\\bin\\tclsh.exe ’ + CAPS_GLYPH + \
                  ’\\GeomToMesh.glf" caps.egads capsUserDefaults.glf’)
    else:
        os.system("pointwise -b " + CAPS_GLYPH + "/GeomToMesh.glf caps.egads capsUserDefaults.glf")

    time.sleep(1) # let the harddrive breathe
    if os.path.isfile(’caps.GeomToMesh.gma’) and os.path.isfile(’caps.GeomToMesh.ugrid’): break
    time.sleep(20) # wait and try again

os.chdir(currentDirectory)       # Move back to top directory
#######################################
```

GeomToMesh.glf Default Values

- $CAPS_GLYPH/GeomToMeshDefaults.glf contains default meshing parameters that control the mesh characteristics
- The default parameter settings mirror values in Pointwise interactive mode
- GeomToMesh.glf input: CAD file and optional “UserDefault.glf”
- The parameters in the “UserDefaults.glf” file will override the settings in the $CAPS_GLYPH/GeomToMeshDefaults.glf

Execute: session07/pointwise_01_Defaults.py

GeomToMesh.glf Suggested Inviscid Values

- Steve Karman: “The values discussed in the following slides are, to some extent, personal preferences evolved over years of experience.”
- These parameters resolve geometry curvature and create high quality surface meshes
- The volume mesh exhibits smooth gradation of element size

session07/pointwise_02_InviscidWing.py

```
# Connector level
pointwise.input.Connector_Turn_Angle      = 10
pointwise.input.Connector_Turn_Angle_Hard = 70
pointwise.input.Connector_Source_Spacing  = True

# Domain level
pointwise.input.Domain_Algorithm    = "AdvancingFront"
pointwise.input.Domain_Max_Layers   = 15
pointwise.input.Domain_TRex_ARLimit = 40.0
pointwise.input.Domain_Decay        = 0.8

# Block level
pointwise.input.Block_Boundary_Decay       = 0.8
pointwise.input.Block_Edge_Max_Growth_Rate = 1.5
```

GeomToMesh.glf Suggested Inviscid Values

- Connector parameters control the mesh operations on EDGEs
- EDGE length and curvature influences the mesh resolution and distribution
- Source Spacing enables proximity checking between connectors

session07/pointwise_02_InviscidWing.py

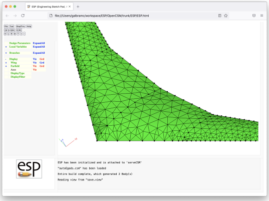
```
# Connector level
pointwise.input.Connector_Turn_Angle      = 10
pointwise.input.Connector_Turn_Angle_Hard = 70
pointwise.input.Connector_Source_Spacing  = True
```

GeomToMesh.glf Suggested Inviscid Values

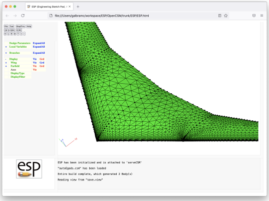
- Domain parameters control the mesh operations on `FACE`s
- Max Layers enables T-Rex surface boundary layer
  - Clustering to high curvature and sharp regions of the geometry

session07/pointwise_02_InviscidWing.py

```
# Domain level
pointwise.input.Domain_Algorithm     = "AdvancingFront"
pointwise.input.Domain_Max_Layers    = 15
pointwise.input.Domain_TRex_ARLimit = 40.0
pointwise.input.Domain_Decay         = 0.8
```



```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```



```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```

GeomToMesh.glf Suggested Inviscid Values

- Block parameters control the mesh operation in the volume, including the extruded viscous mesh portion
- Block_Boundary_Decay: Rate of element size increase away from boundaries
- Block_Edge_Max_Growth_Rate: Controls growth rate along connectors and gradation of the volume mesh

session07/pointwise_02_InviscidWing.py

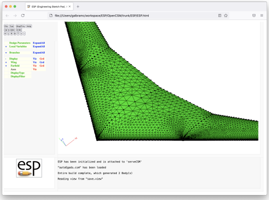
```
# Block level
pointwise.input.Block_Boundary_Decay       = 0.8
pointwise.input.Block_Edge_Max_Growth_Rate = 1.5
```

Leading Edge Spacing: Connector_Turn_Angle

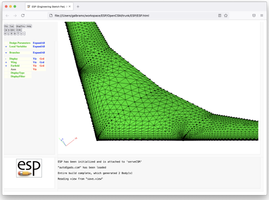
- Turning Angle resolves `EDGE`s to match the specified degree
- Lower angles increases `EDGE` resolution in high curvature regions

session07/pointwise_03_TurnAngle.py

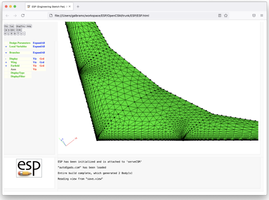
```python
# Demonstrate the impact of Connector_Turn_Angle
for conTurnAngle in [5, 10, 20]:
    # Modify the turn angle
    pointwise.input.Connector_Turn_Angle = conTurnAngle
```



```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```



```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```



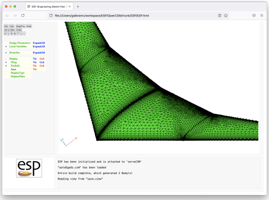
```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```

Trailing Edge Spacing: Connector_Turn_Angle_Hard

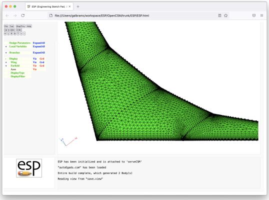
- Turning Angle Hard resolves EDGE with acute angle between connected FACEs
- Lower angles increases EDGE resolution

session07/pointwise_04_TurnAngleHard.py

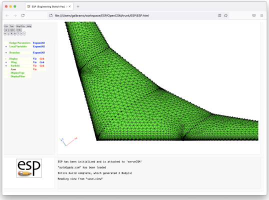
```
# Demonstrate the impact of Connector_Turn_Angle
for conTurnAngleHard in [10, 30, 50]:
    # Modify the hard turn angle
    pointwise.input.Connector_Turn_Angle_Hard = conTurnAngleHard
```



```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```



```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```



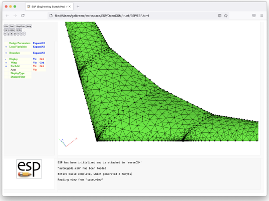
```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```

Impact of Domain_Decay

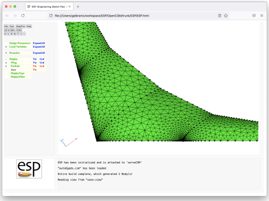
- Domain decay controls gradation of element sizes away from the EDGEs a surface mesh patch.
- Values near 1.0 give gradual increase in element size.
- Values (0.5 or less) gives rapid increase is element size.

session07/pointwise_05_DomainDecay.py

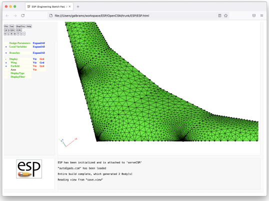
```python
# Demonstrate the impact of Domain_Decay
for domDecay in [0.1, 0.6, 0.95]:
    # Modify the domain decay
    pointwise.input.Domain_Decay = domDecay
```



```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```



```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```



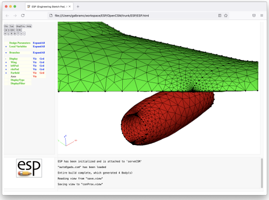
```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 2 Body(s)
Reading view from "save.view"
```

Proximity Detection

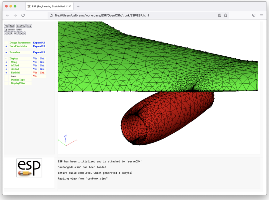
- Pod includes `EDGE` geometry close to the wing leading edge
- Connector Source Spacing enables connectors-to-connector proximity detection
- The proximity test also uses Block level parameters

session07/pointwise_06_ConnectorProximity.py

```
# Demonstrate the impact of Connector_Source_Spacing
for conSourceSpace in [False, True]:
    # Modify the source spacing
    pointwise.input.Connector_Source_Spacing = conSourceSpace
```



```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 4 Body(s)
Reading view from "save.view"
Saving view to "conProx.view"
```



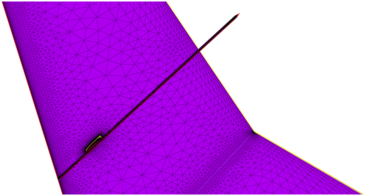
```
ESP has been initialized and is attached to 'serveCSM'
"autoEgads.csm" has been loaded
Entire build complete, which generated 4 Body(s)
Reading view from "conProx.view"
```

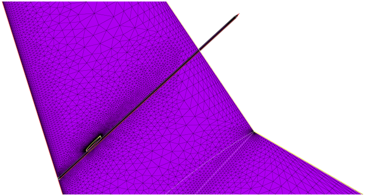
Proximity Detection

- Domain-to-domain proximity detection enabled with
  Connector_Adapt_Sources
  Domain_Adapt
- **FACE**s must be attributed with

```
ATTRIBUTE PW:DomainAdaptSource $true
ATTRIBUTE PW:DomainAdaptTarget $true
```

- Source domains will refine Target domains
- A single domain can be tagged both source and target

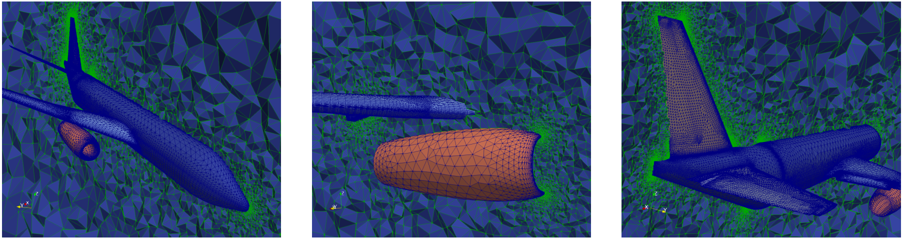




Full Inviscid Transport Example

- Inviscid surface mesh for the full transport configuration using suggested parameters
  - ∼ 6 min
  - 760k Nodes
  - 4.4M Elements

session07/pointwise_07_InviscidTransport.py



Overview

- Pointwise and GeomToMesh.glf
- Pointwise invsicid meshing
  - Suggested parameters
  - Proximity detection
- Pointwise viscous meshing
  - Suggested parameters
  - Viscous boundary layer mesh generation
- Suggested Exercises

GeomToMesh.glf Suggested Viscous Values

- Recommended viscous values are similar to the inviscid
- Block level parameters to control boundary layer meshing

session07/pointwise_08_ViscousWing.py

```
# Connector level
pointwise.input.Connector_Turn_Angle       = 10
pointwise.input.Connector_Prox_Growth_Rate = 1.2
pointwise.input.Connector_Source_Spacing   = True

# Domain level
pointwise.input.Domain_Algorithm    = "AdvancingFront"
pointwise.input.Domain_Max_Layers   = 15
pointwise.input.Domain_Growth_Rate  = 1.25
pointwise.input.Domain_TRex_ARLimit = 40.0
pointwise.input.Domain_Decay        = 0.8

# Block level
pointwise.input.Block_Boundary_Decay       = 0.8
pointwise.input.Block_Collision_Buffer     = 1.0
pointwise.input.Block_Max_Skew_Angle       = 160.0
pointwise.input.Block_Edge_Max_Growth_Rate = 1.5
pointwise.input.Block_Full_Layers          = 1
pointwise.input.Block_Max_Layers           = 100

# Set wall spacing for capsMesh == leftWing and capsMesh == riteWing
viscousWall  = {"boundaryLayerSpacing" : 0.001}
pointwise.input.Mesh_Sizing = {"leftWing": viscousWall,
                               "riteWing": viscousWall}
```

GeomToMesh.glf Suggested Viscous Values

- The suggested connector attributes very similar to inviscid values.
- Proximity growth rate has been slightly reduced.

session07/pointwise_08_ViscousWing.py

```
# Connector level
pointwise.input.Connector_Turn_Angle       = 10
pointwise.input.Connector_Prox_Growth_Rate = 1.2
pointwise.input.Connector_Source_Spacing   = True
```

GeomToMesh.glf Suggested Viscous Values

- The suggested domain attributes also similar to inviscid values.
- The growth rate has been slightly reduced.

session07/pointwise_08_ViscousWing.py

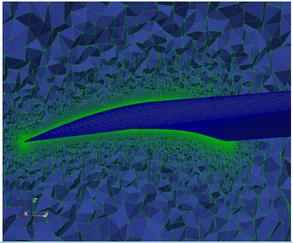
```
# Domain level
pointwise.input.Domain_Algorithm    = "AdvancingFront"
pointwise.input.Domain_Max_Layers   = 15
pointwise.input.Domain_Growth_Rate  = 1.25
pointwise.input.Domain_TRex_ARLimit = 40.0
pointwise.input.Domain_Decay        = 0.8
```

GeomToMesh.glf Suggested Viscous Values

- Increased collision buffer to allow gap between fronts
- Skew angle stops T-Rex locally when elements exceed angle
- Full Layers is the desired minimum number of full viscous layers
- Max Layer is upper bounds on the number of layers
- T-Rex automatically halts locally when elements approach isotropy

session07/pointwise_08_ViscousWing.py

```
# Block level
pointwise.input.Block_Boundary_Decay       = 0.8
pointwise.input.Block_Collision_Buffer     = 1.0
pointwise.input.Block_Max_Skew_Angle       = 160.0
pointwise.input.Block_Edge_Max_Growth_Rate = 1.5
pointwise.input.Block_Full_Layers          = 1
pointwise.input.Block_Max_Layers           = 100
```



Colliding Boundary Layers

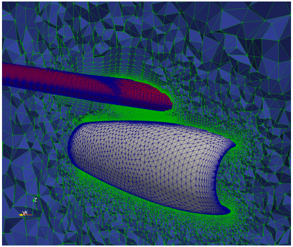
- Viscous extrusion with `FACE` attribute

  `ATTRIBUTE PW:WallSpacing 0.001`

- Set with Mesh_Sizing `boundaryLayerSpacing` using `capsMesh` attribute
  - `boundaryLayerSpacing` scaled by `capsMeshLength`
- Boundary layer wall spacing can differ between `FACEs`

session07/pointwise_09_ViscousWingPod.py

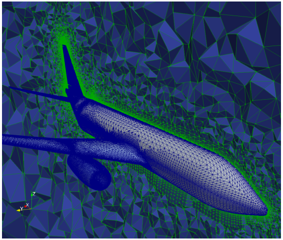
```
# Set wall spacing for capsMesh == leftWing, riteWing, Pod
wingWall = {"boundaryLayerSpacing" : 0.001}
podWall  = {"boundaryLayerSpacing" : 0.003}
pointwise.input.Mesh_Sizing = {"leftWing" : wingWall,
                               "riteWing": wingWall,
                               "Pod"     : podWall}
```

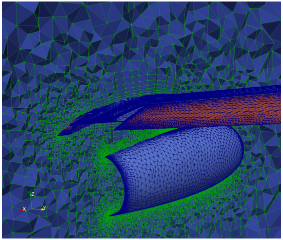


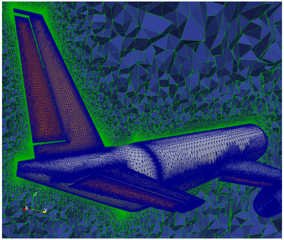
Full Transport Example

- Viscous surface mesh for the full transport configuration using suggested parameters[1]
  - ∼ 20 min
  - 2.5M Nodes
  - 14.6M Elements

session07/pointwise_10_ViscousTransport.py







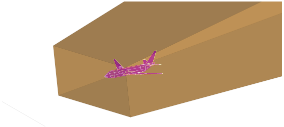
[1] NOTE: Unreasonably coarse boundary layers in examples

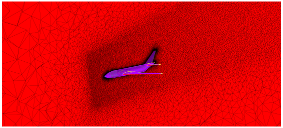
Source Box

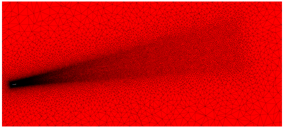
- Box is 20% larger than bounding box of all viscous surfaces
- Length Scaled multiplied along the direction vector
- Widens using the Angle parameter
- Scalar size field grows from surface size along the box

session07/pointwise_11_SourceBox.py

```
# General source box parameters
pointwise.input.Gen_Source_Box_Length_Scale = 2.0 # Double size in Box Direction
pointwise.input.Gen_Source_Box_Direction    = [ 0.9848077, 0, 0.1736482 ]
pointwise.input.Gen_Source_Box_Angle        = 10.0
pointwise.input.Gen_Source_Growth_Factor    = 40.0
```

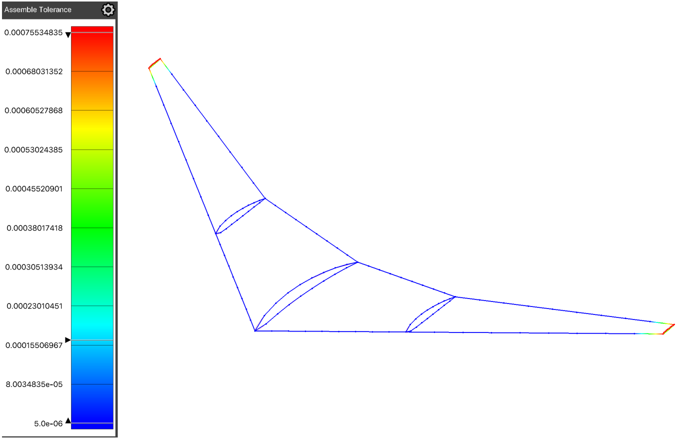






Full Disclosure: Poor Assemble Tolerance

- EGADS files in Pointwise may have poor assemble tolerance
- Can cause meshing failures or errors exporting to CAPS
- Fix exists and will be in future Pointwise release



Inputs Scaled by `capsMeshLength`

Mesh_Sizing Parameters

- nodeSpacing: Spacing around a `NODE`
- minSpacing, maxSpacing, avgSpacing: `EDGE` or `FACE`
- maxDeviation: `EDGE` or `FACE` deviation
- boundaryLayerSpacing: `FACE` initial boundary layer spacing

Global Input Parameters

- Domain_Min_Edge: Domain minimum Edge length
- Domain_Max_Edge: Domain maximum Edge length
- Domain_Wall_Spacing: Initial boundary layer spacing on `FACE`s with

```
ATTRIBUTE PW:WallSpacing $Wall
```

Primary Mesh Resolution Parameters

Surface Mesh Resolution

- Connector_Turn_Angle: High curvature of connector
- Connector_Prox_Growth_Rate: Connector-to-connector prox.
- Connector_Turn_Angle_Hard: Acute angles between `FACE`s
- Domain_TRex_ARLimit: Spanwise resolution of surface TRex
- Block_Edge_Max_Growth_Rate: Growth rate along connectors and gradation of volume mesh

Volume Mesh Resolution

- Block_Boundary_Decay: Rate of element size increase away from boundaries
- Block_Growth_Rate : Growth rate of viscous boundary layer

Suggested Exercises

Inviscid Mesh Sequence

- For the InviscidTransport, generate surface meshes with approximate element counts of:
    - 150,000
    - 250,000
    - 300,000

Domain-to-Domain Proximity Inviscid Wing with Pods

- Add combinations of Source and Target attributes to the Wing and/or Pod in ESP/viewCFDInviscid.udc (do not use EGADS/CFDInviscid_WingPod.egads)

```
ATTRIBUTE PW:DomainAdaptSource $true
```

```
ATTRIBUTE PW:DomainAdaptTarget $true
```

- Toggle Connector_Adapt_Sources and Domain_Adapt

Suggested Exercises Cont.

Other AIM Inputs

- Explore the impact of Pointwise AIM input parameters
  - Connector, Domain, Block inputs

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