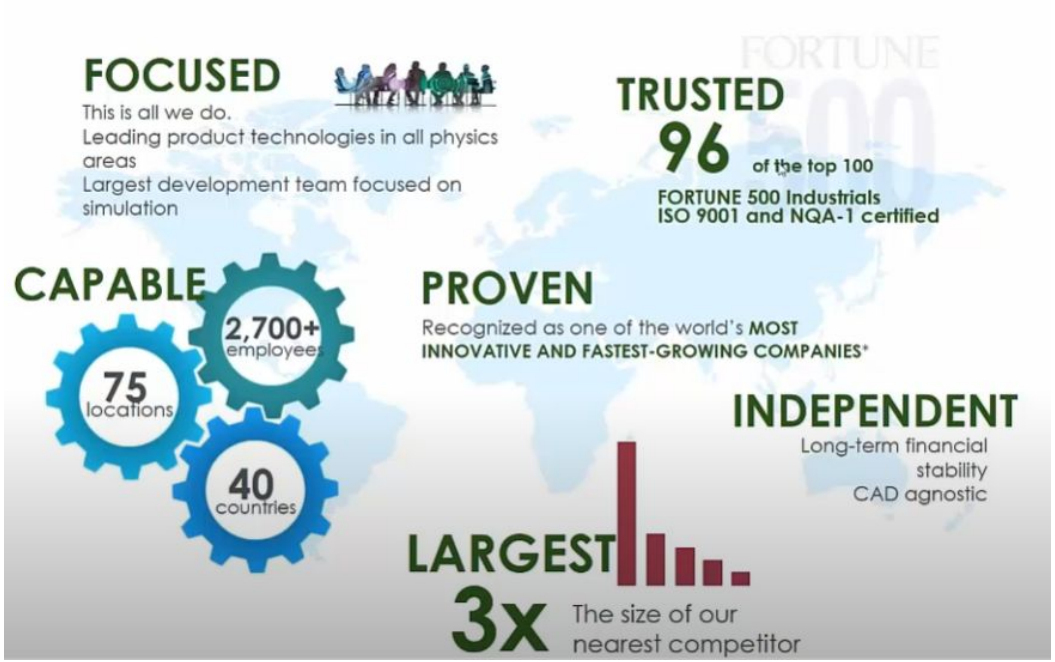
ARES Workshop - Ansys FEA

Cas Kent, Ben van der Veer, Hasan al Banna

Ansys

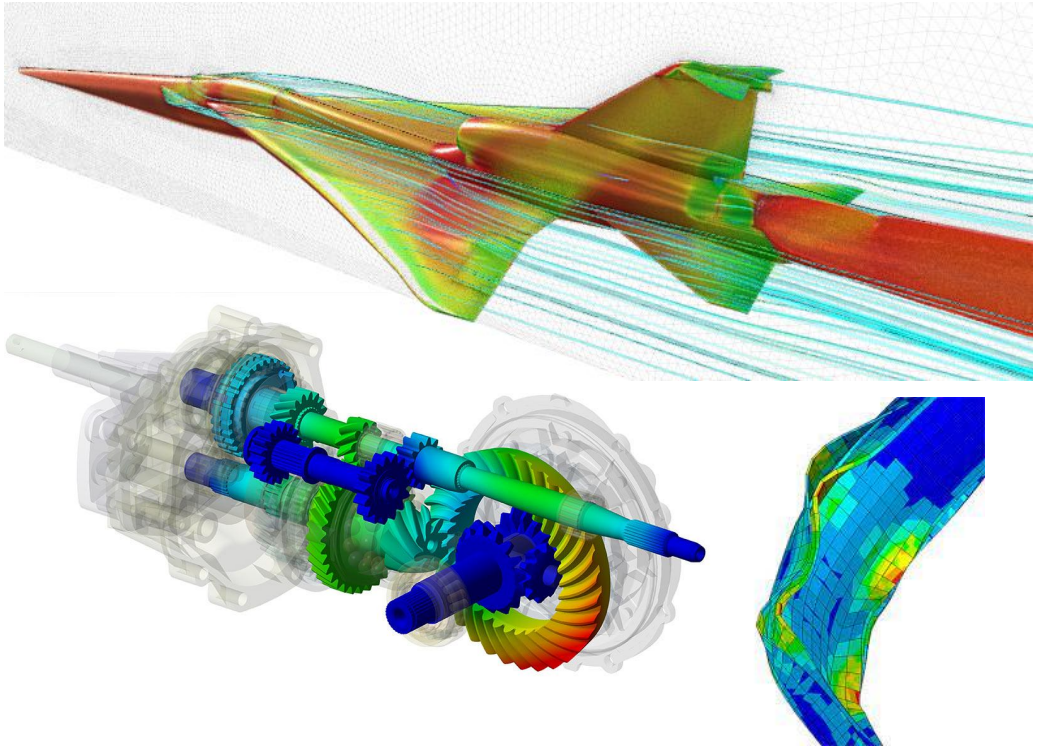
- Industry standard engineering **simulation** software
- Complete package solution
- Global leader in simulation
- Trusted partner of ARES - **LEAP Australia** provide support

leap australia pty ltd



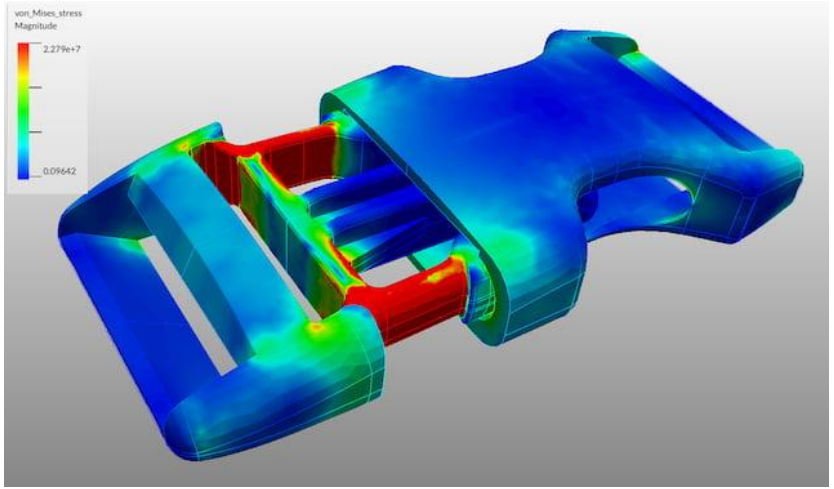
Ansys

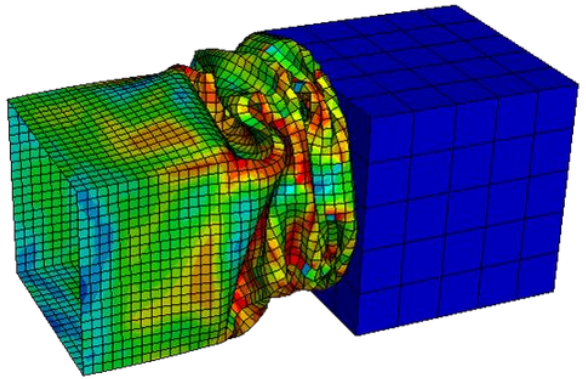
- 3D Design tools (**CAD**)
- Simulation of various physical phenomena, including:
  - Fluid dynamics (**CFD**)
  - Structural mechanics (**FEA**)
  - Thermal
  - Electromagnetic
- Multiphysics simulations
- Model composite materials (carbon fibre)



Why FEA?

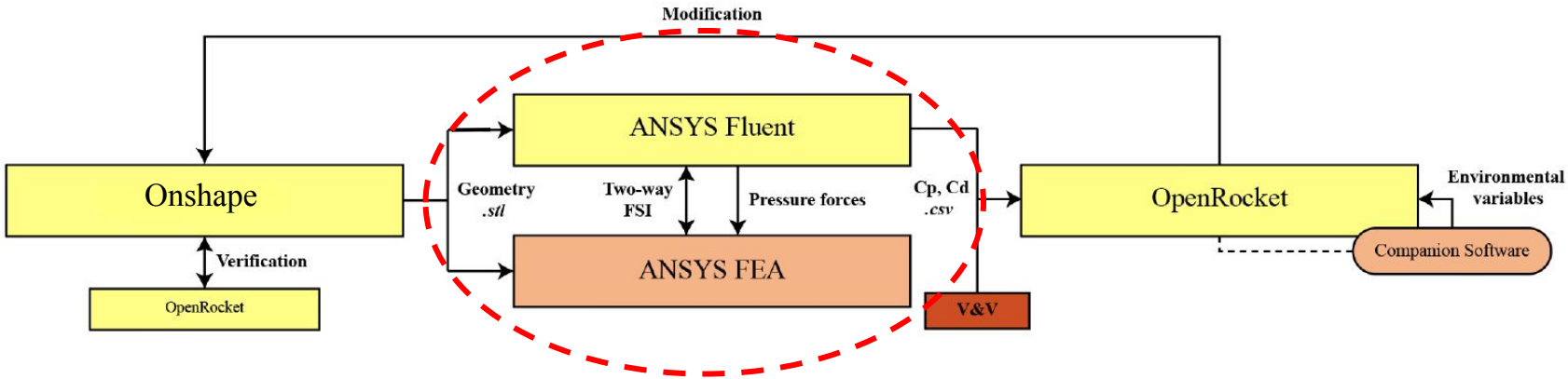
- Numerically solve solid mechanics problems **(avoid a lot of algebra!)**
- Detailed info about **stress & strain distributions,** and deformation of solid bodies
- Can also find **thermal** distributions





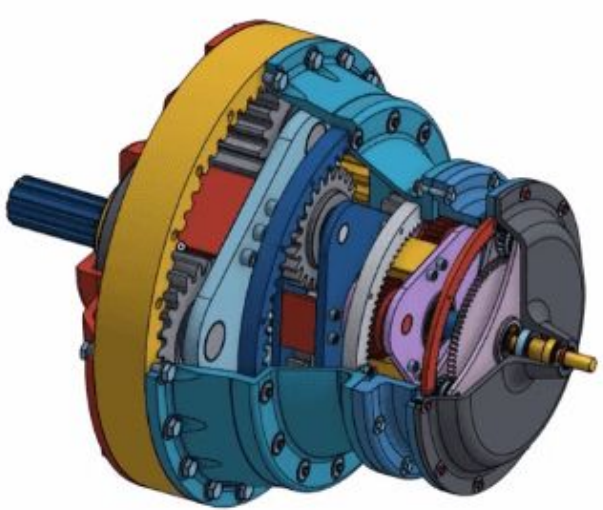
Why CFD?

- **Verify structural integrity** of components under expected **loading conditions**
- Determine the **likely failure modes and locations**
- Reduces effort and cost by avoiding physical experiments
- **Improve accuracy of CFD** analysis by including deflections from FEA



Software you need

- **Onshape**
  - Online 3D modelling (CAD) software
  - ARES uses this for all of our **3D design projects**



Software you need

- Ansys **SpaceClaim** - CAD software
  - Don’t build your model in SpaceClaim!
  - **Prepare the geometry** for CFD
- Ansys **Mechanical** - FEA software
  - Build the **mesh** for FEA
  - Apply loading conditions
  - Solve the FEA

Ansys Mechanical

Design Workflow

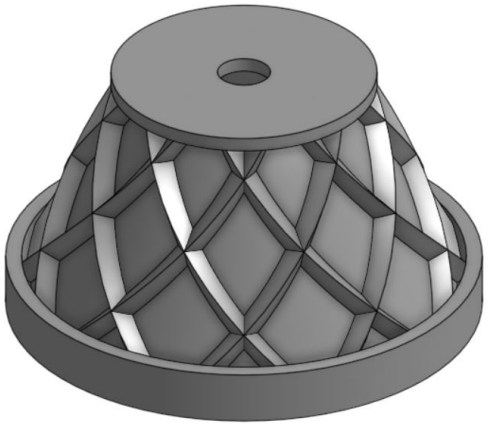
Onshape (full 3D model)
Part Studio

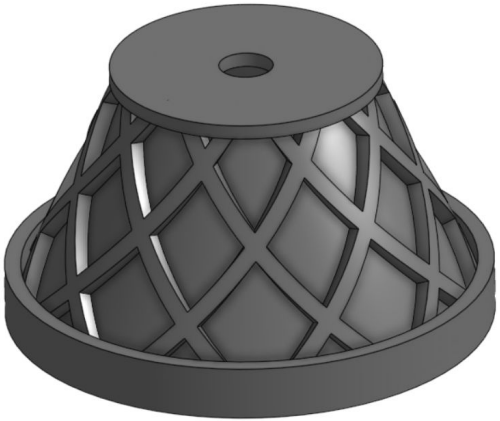
Clean up

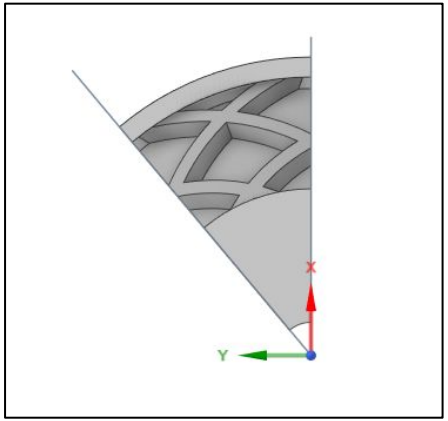
Onshape (simplified)
.sat (ACIS)

Export

SpaceClaim
.scdoc







Design Workflow

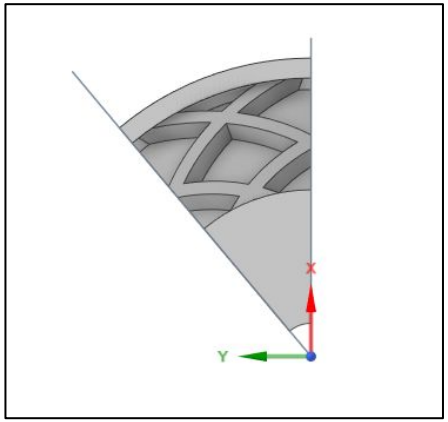
SpaceClaim
.scdoc

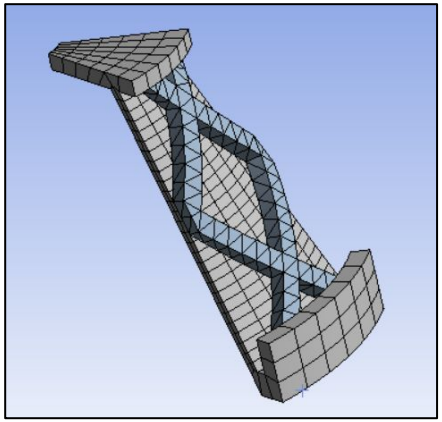
Import to Fluent

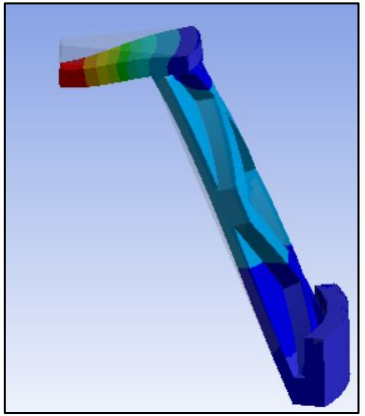
Mechanical (mesh)
.msh

Read mesh

Mechanical
(solver)

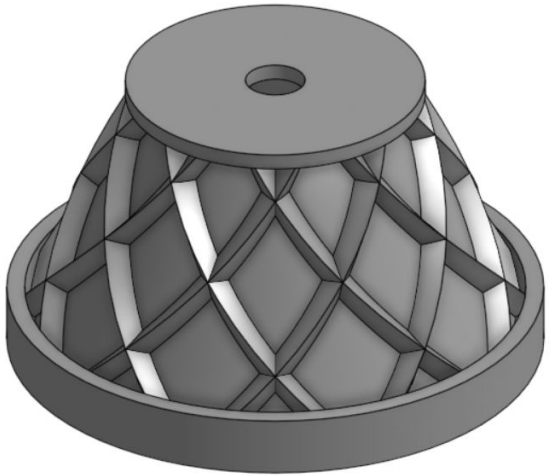


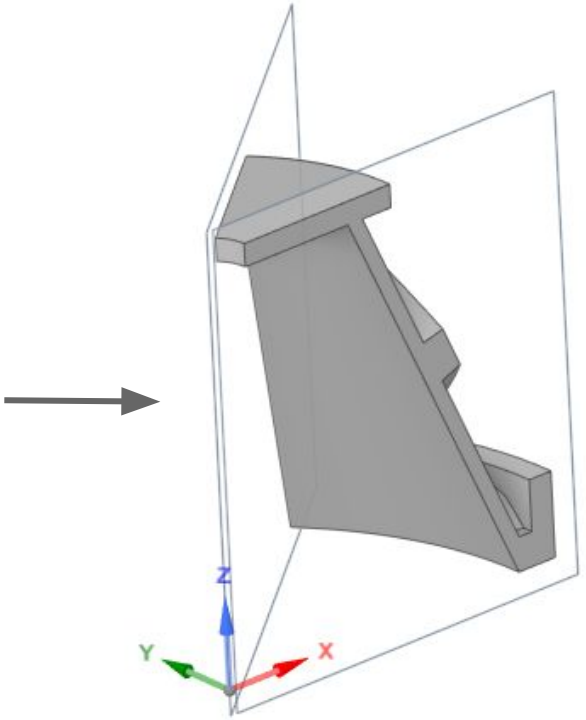




Preparing Geometry

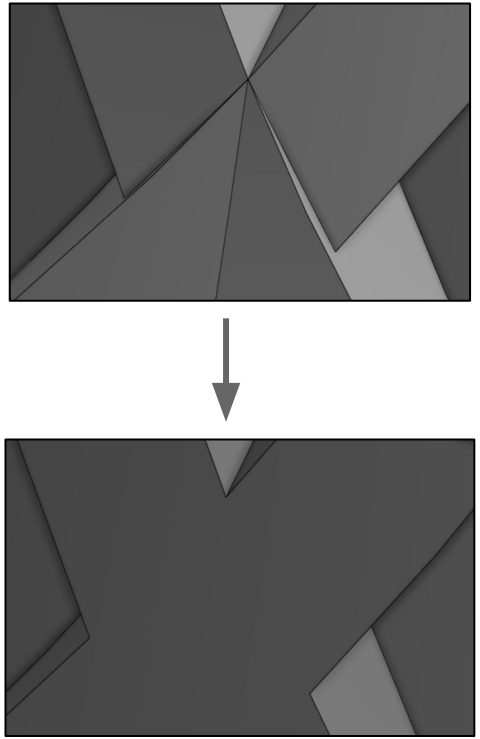
- **FEA can't handle** complex geometry
- Need to **simplify** the geometry for FEA
- Build a **simplified model in Onshape**
- Exporting might cause more issues, so you need to **double check in Ansys**



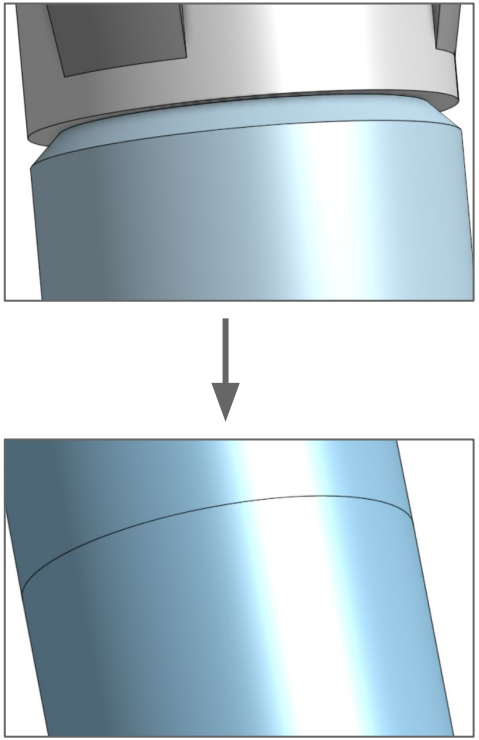


Cleanup - Onshape

Small faces, manufacturing features

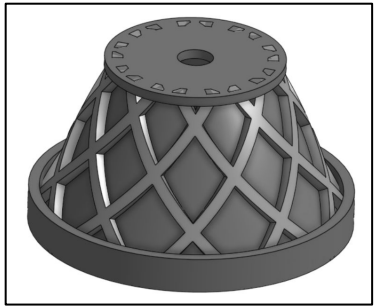


Fillets, chamfers, rounds

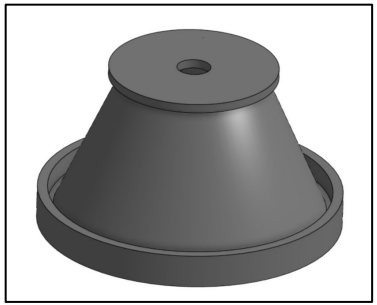


Cleanup - Onshape

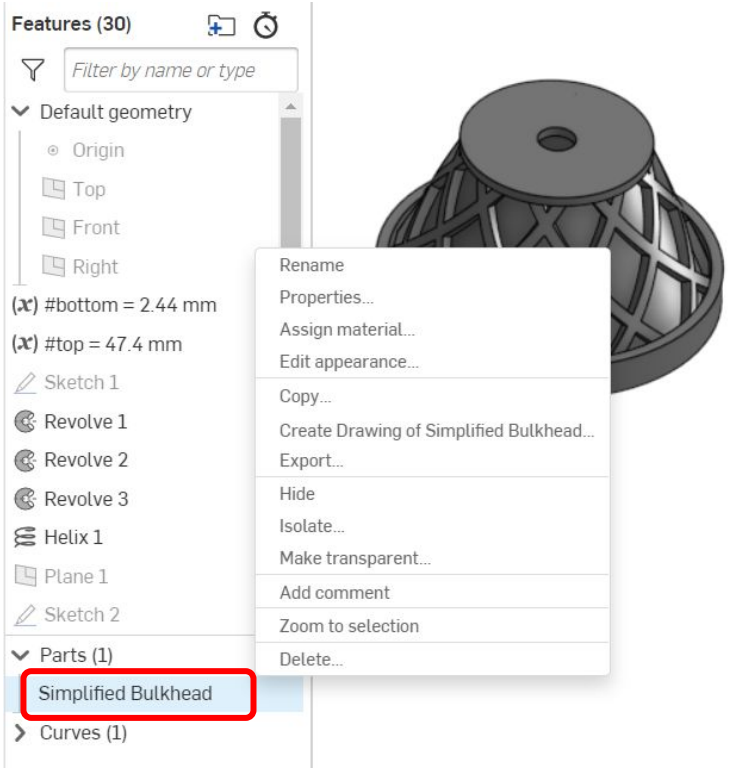
Relevant features?



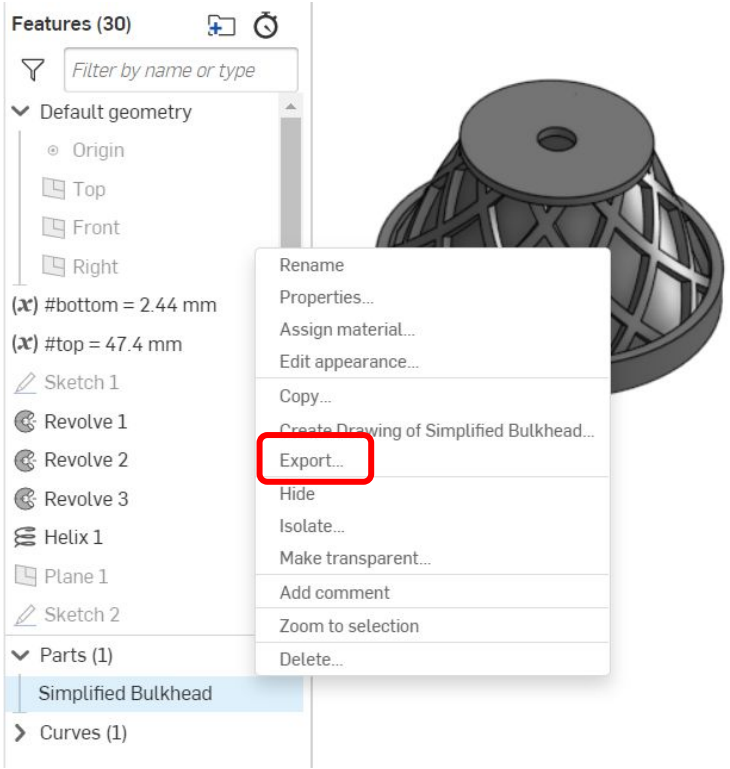
Will this compromise strength?



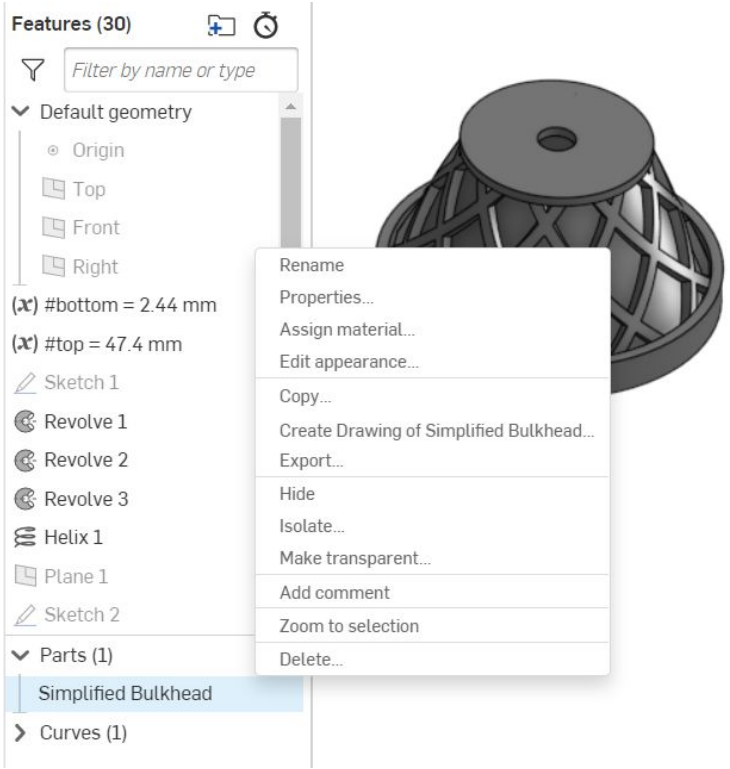
Export Onshape

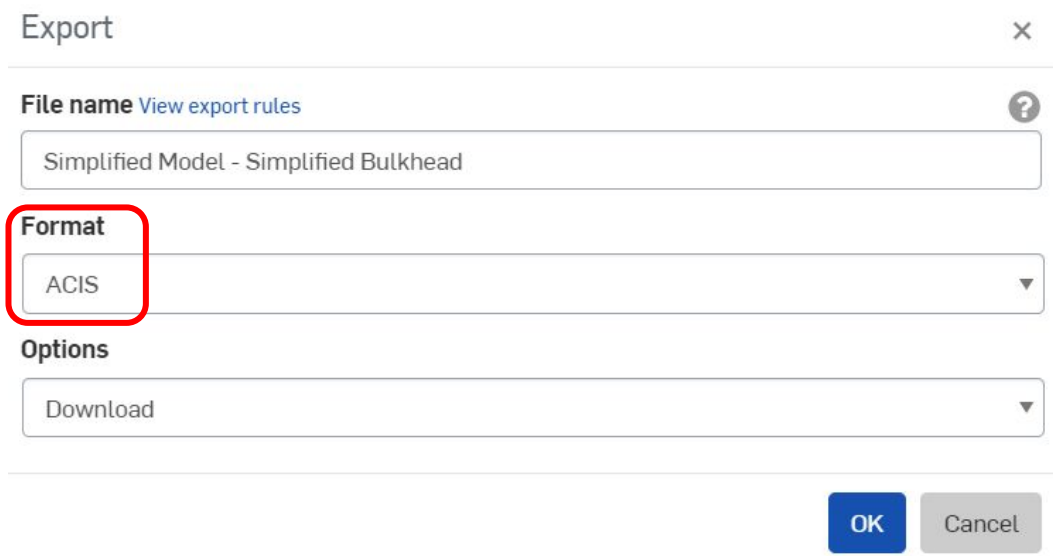


Export Onshape

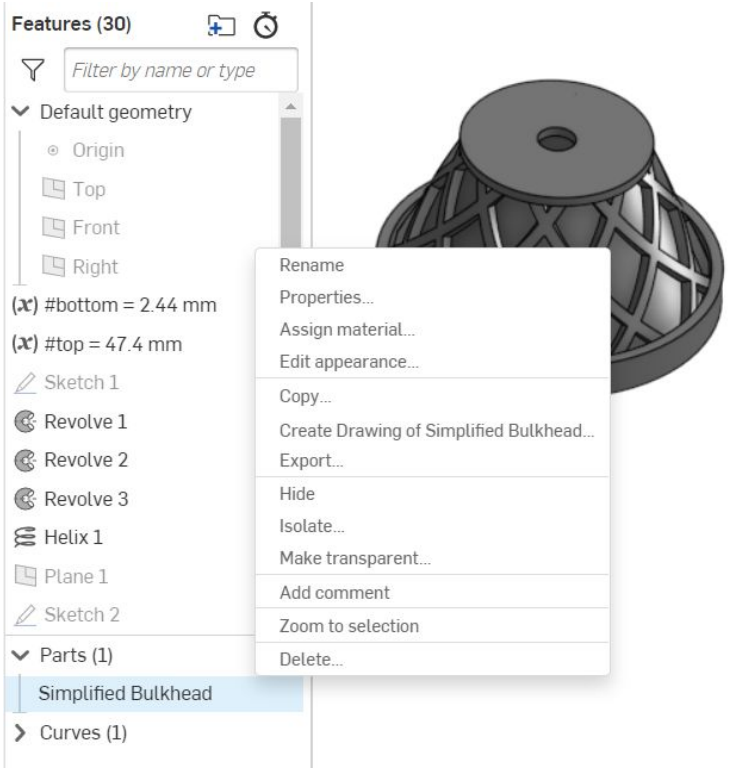


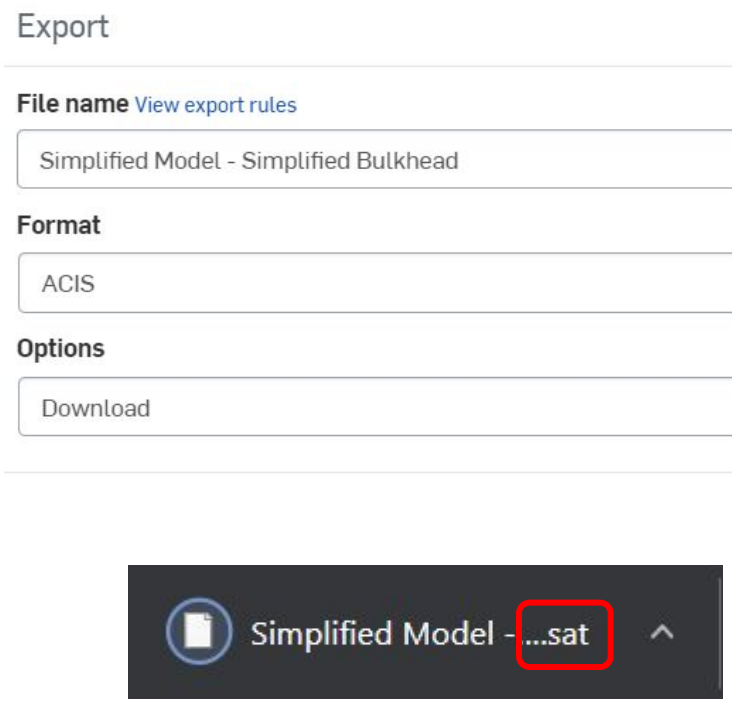
Export Onshape





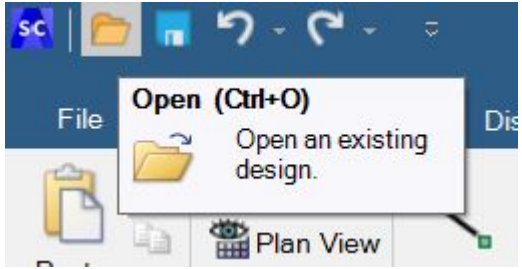
Export Onshape



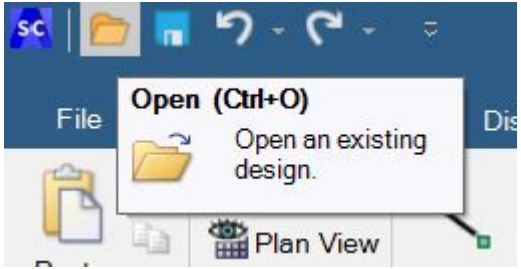


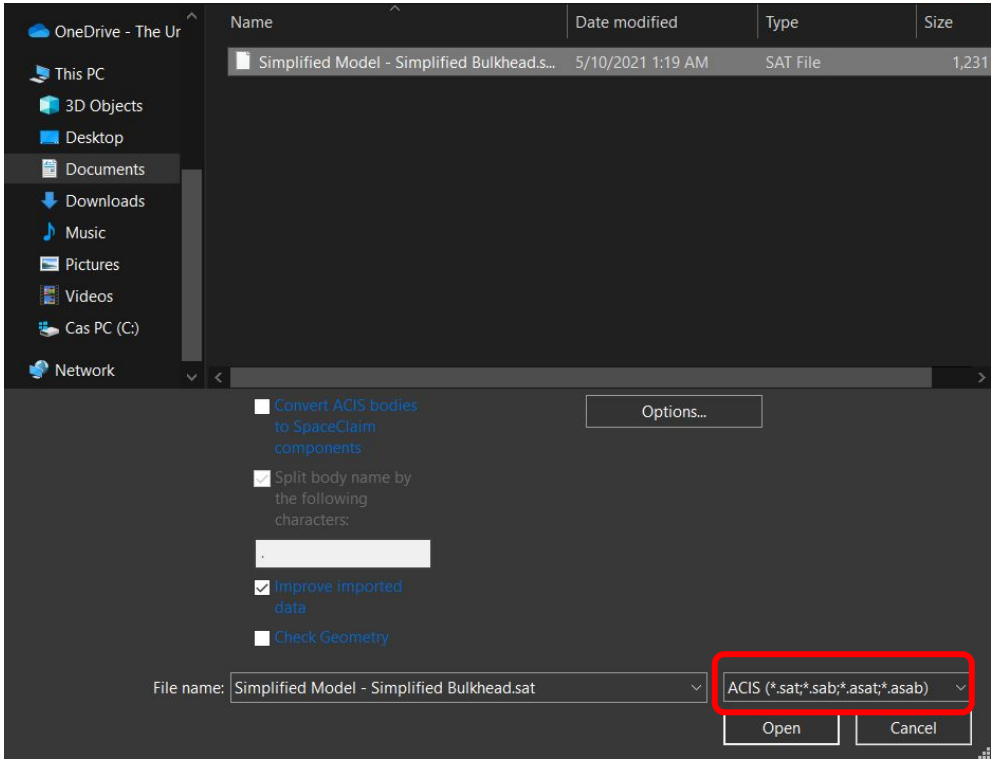
Export Onshape to SpaceClaim

Export Onshape to SpaceClaim

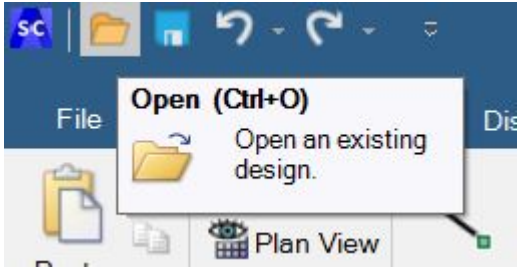


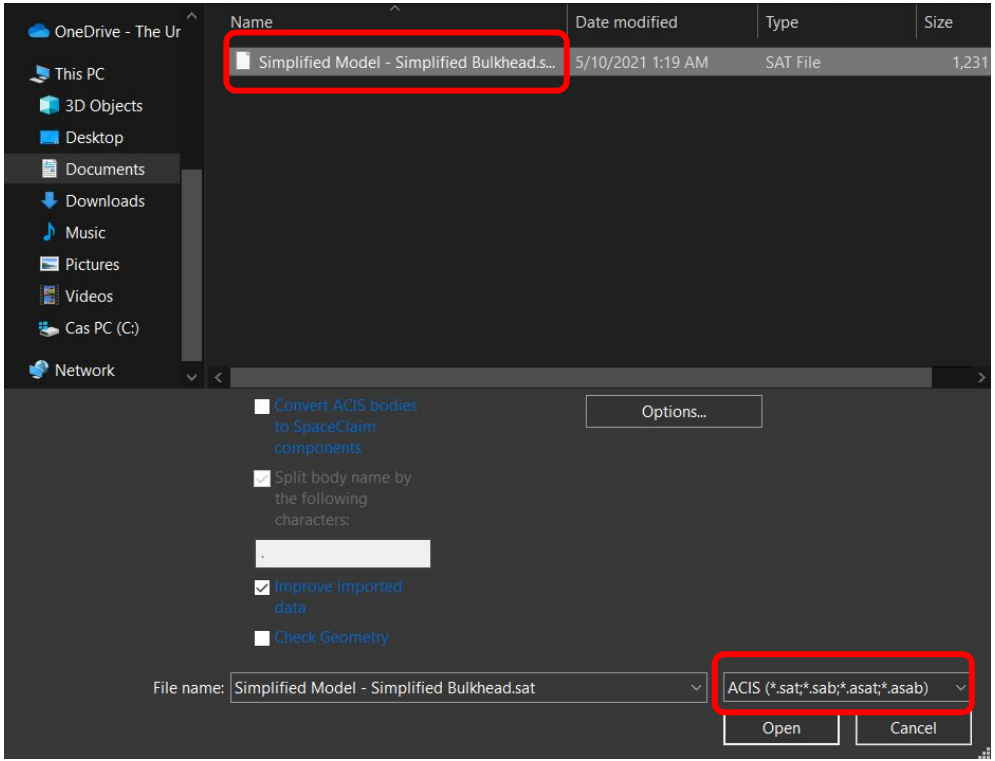
Export Onshape to SpaceClaim



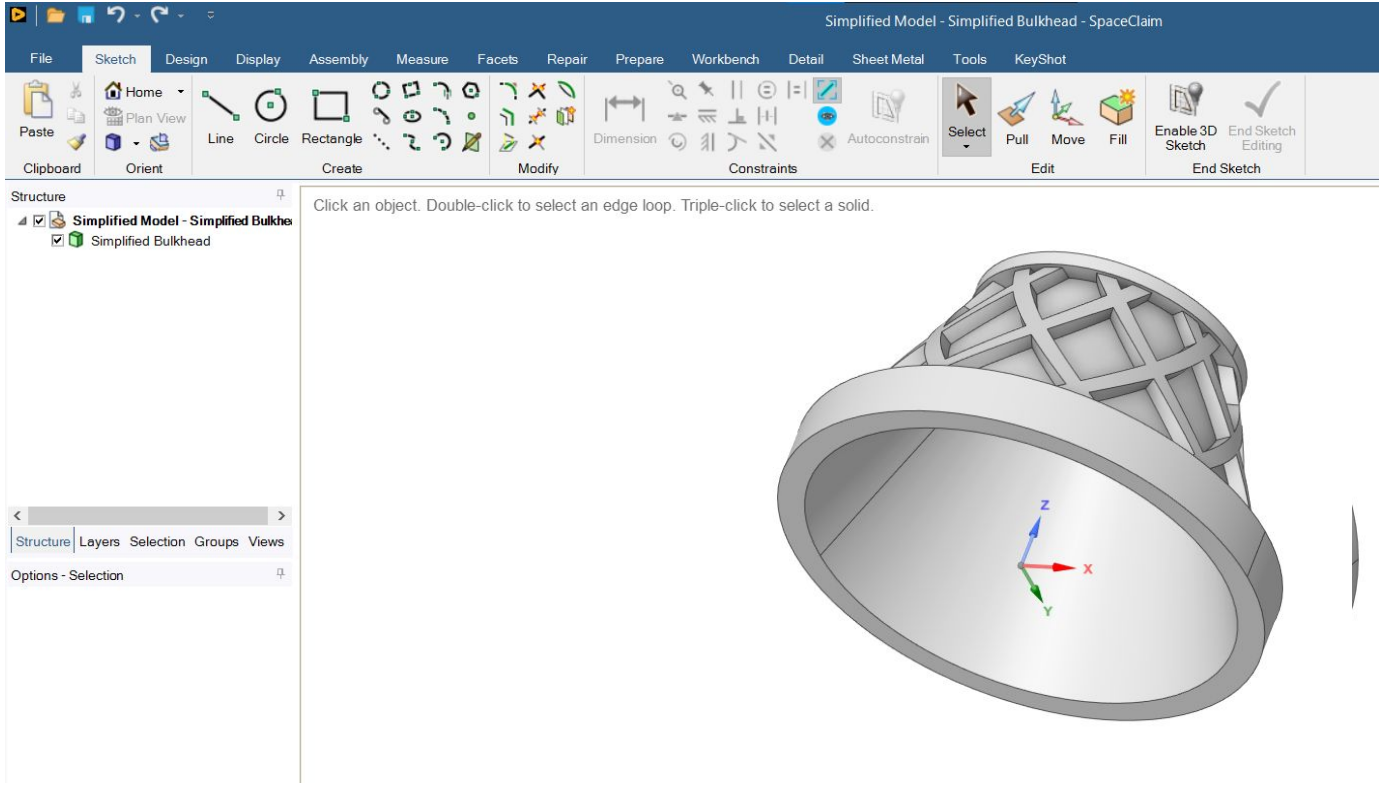


Export Onshape to SpaceClaim

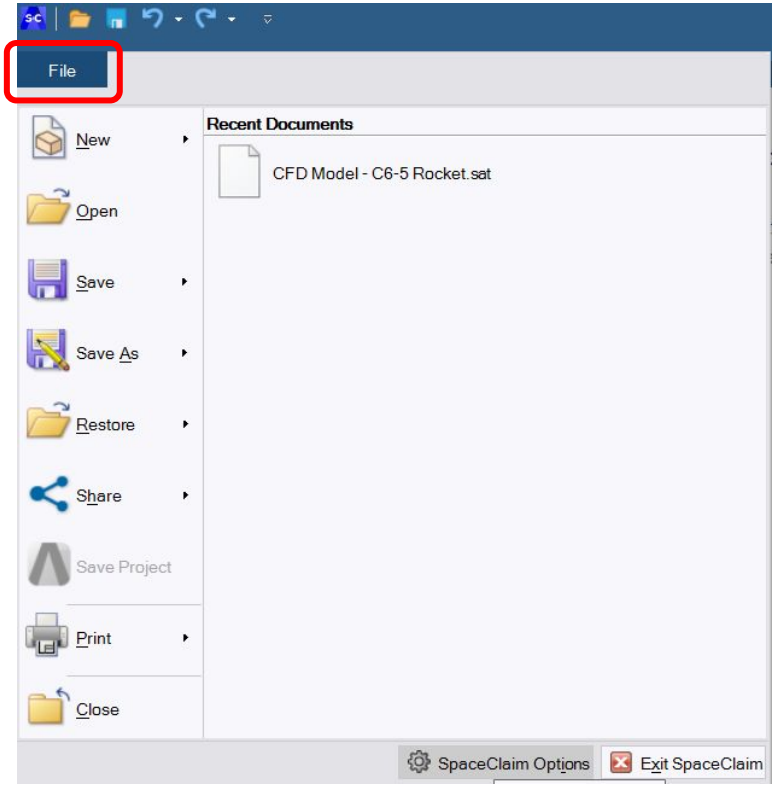




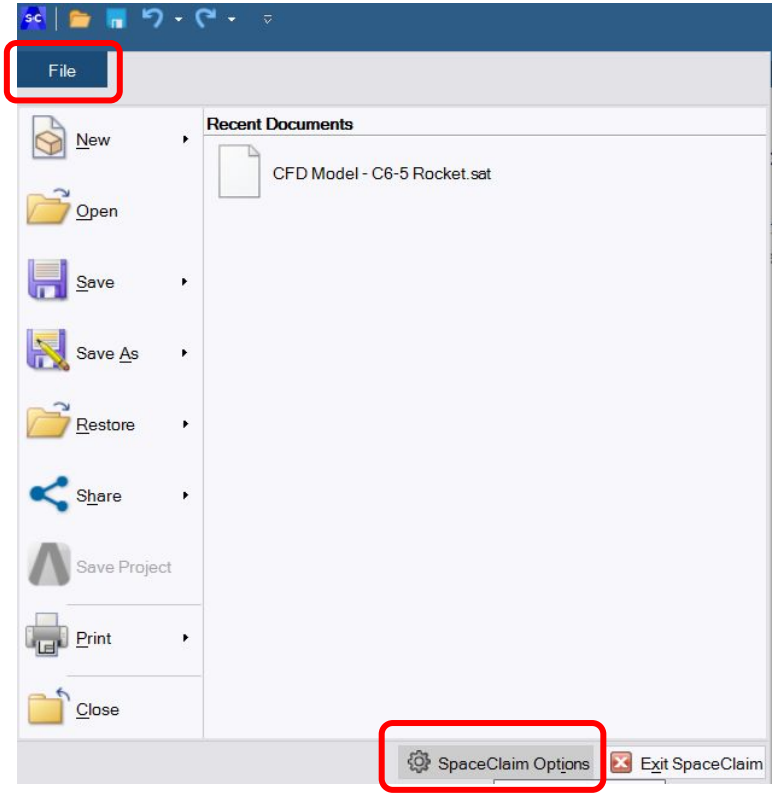
Export Onshape to SpaceClaim



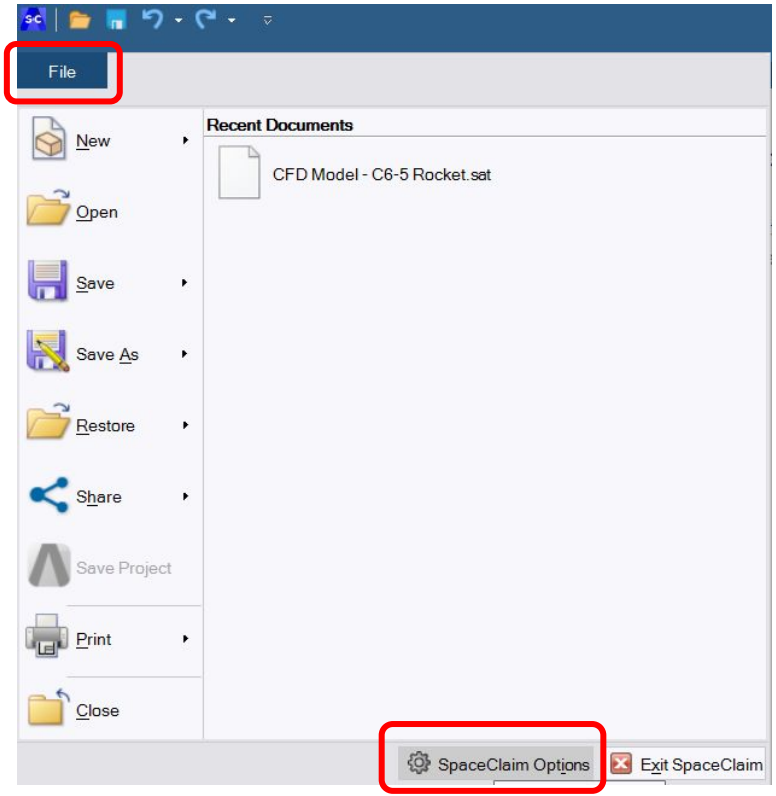
SpaceClaim Mouse Controls

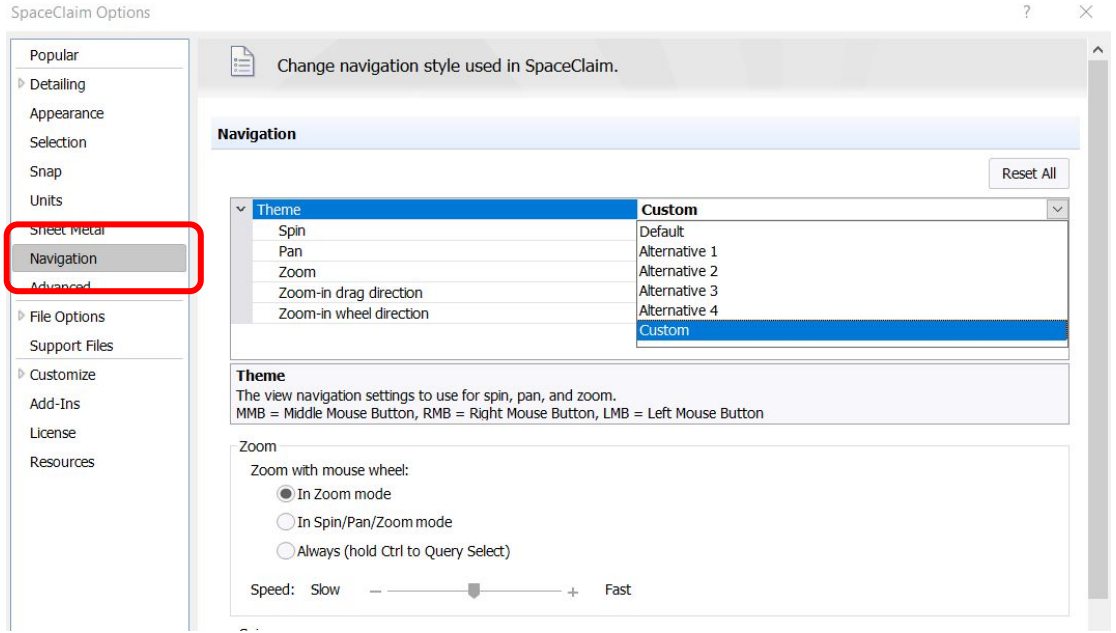


SpaceClaim Mouse Controls

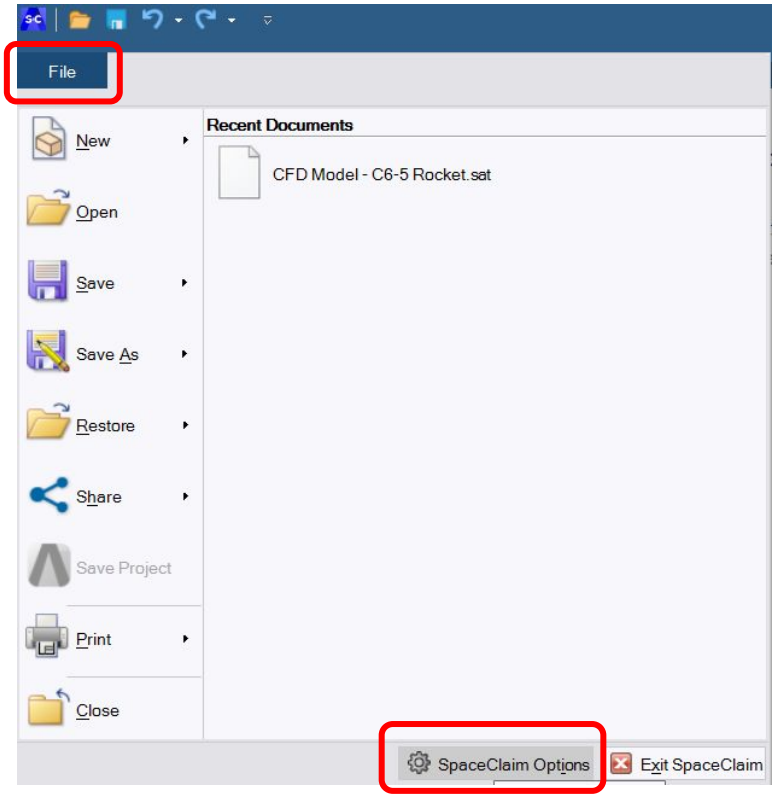


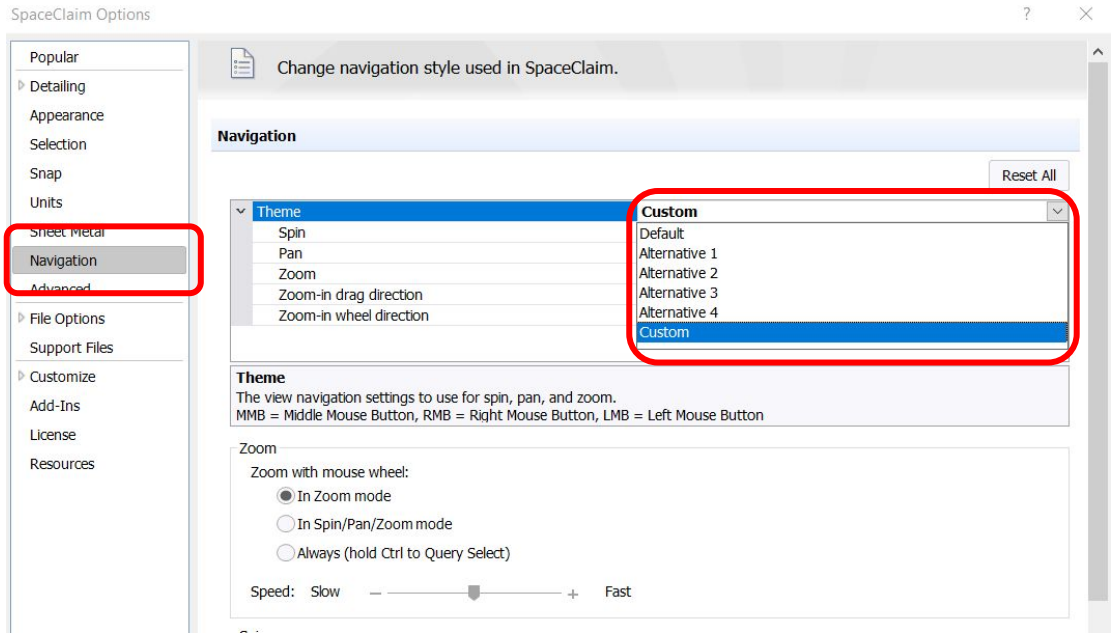
SpaceClaim Mouse Controls



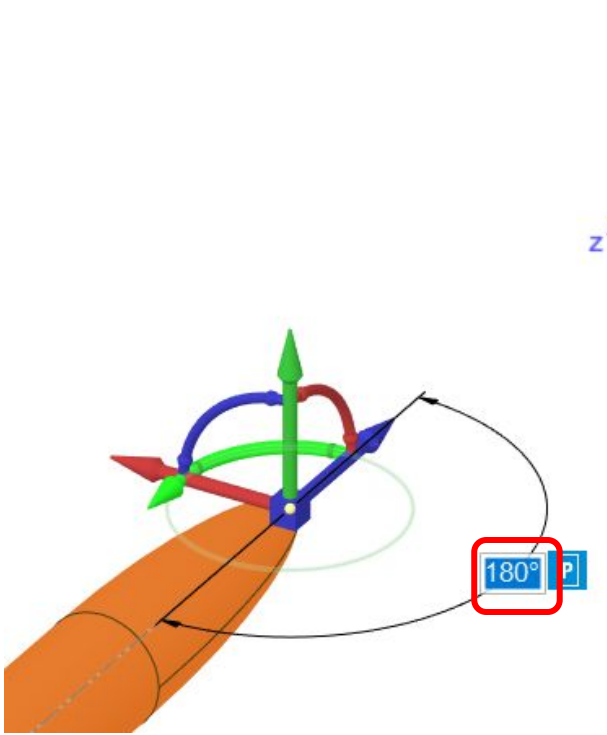


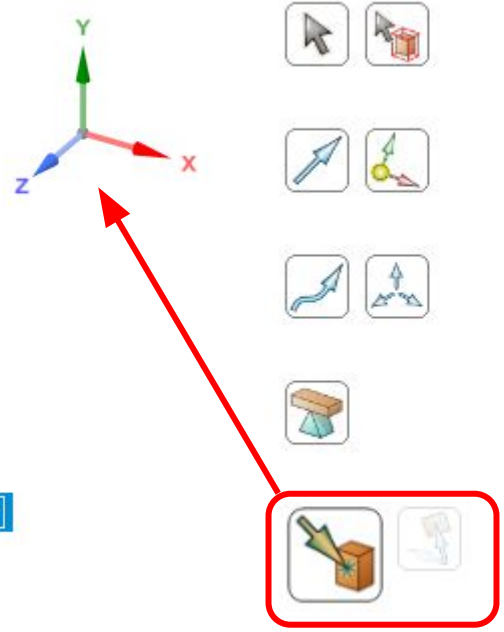
SpaceClaim Mouse Controls



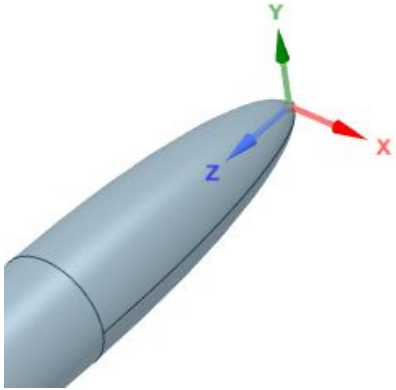


Fix Origin and Orientation

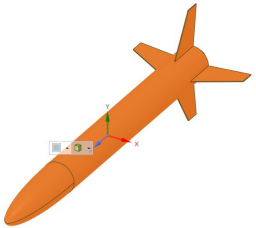


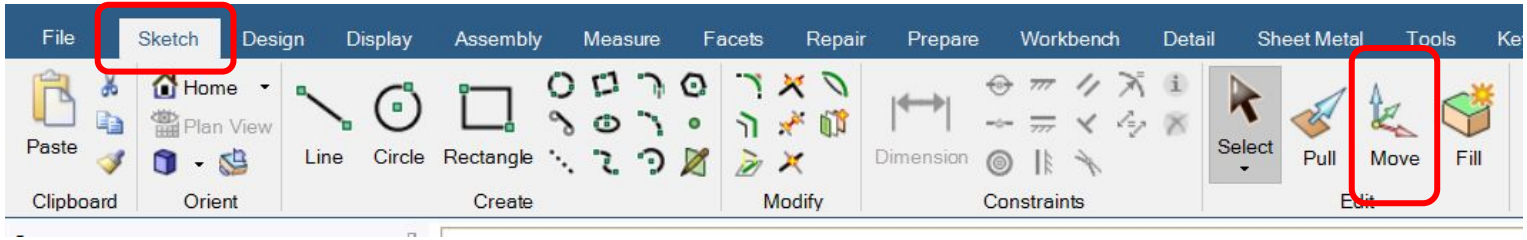


Press **esc** to leave the Move Tool



Fix Origin and Orientation



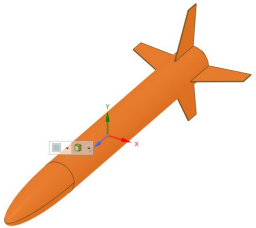


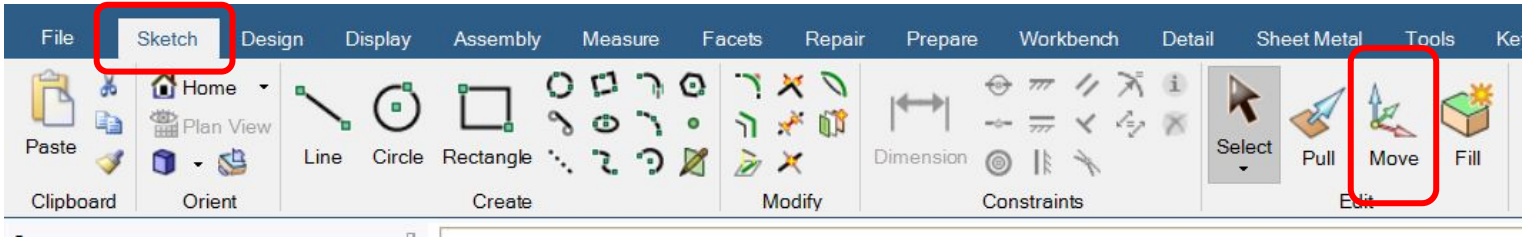
Want gravity (***z***) pointing down the rocket

Want the origin at the **tip of the nose** cone to match OpenRocket

Triple click to select the entire rocket

Fix Origin and Orientation

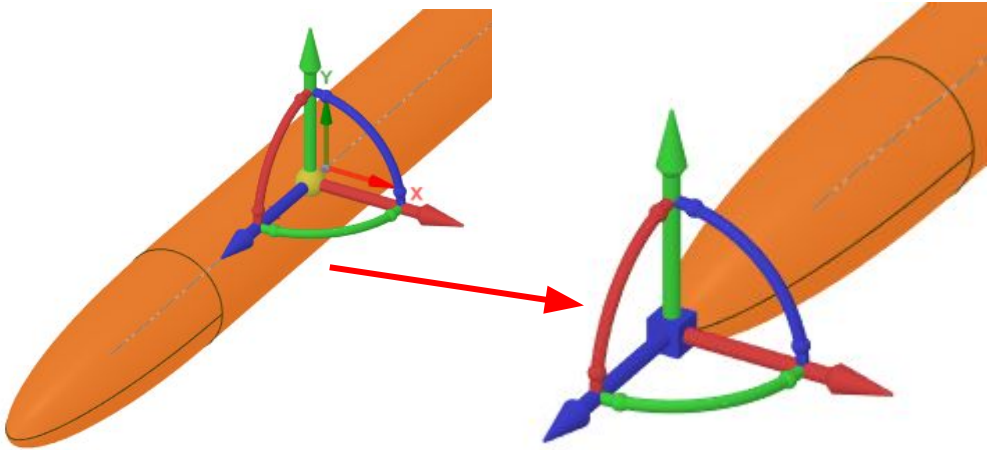




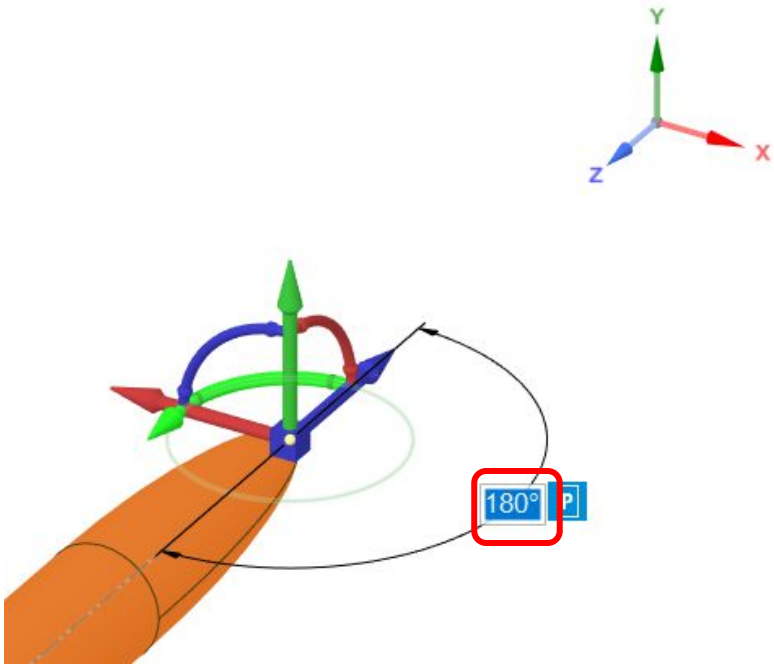
Want gravity (***z***) pointing down the rocket

Want the origin at the **tip of the nose** cone to match OpenRocket

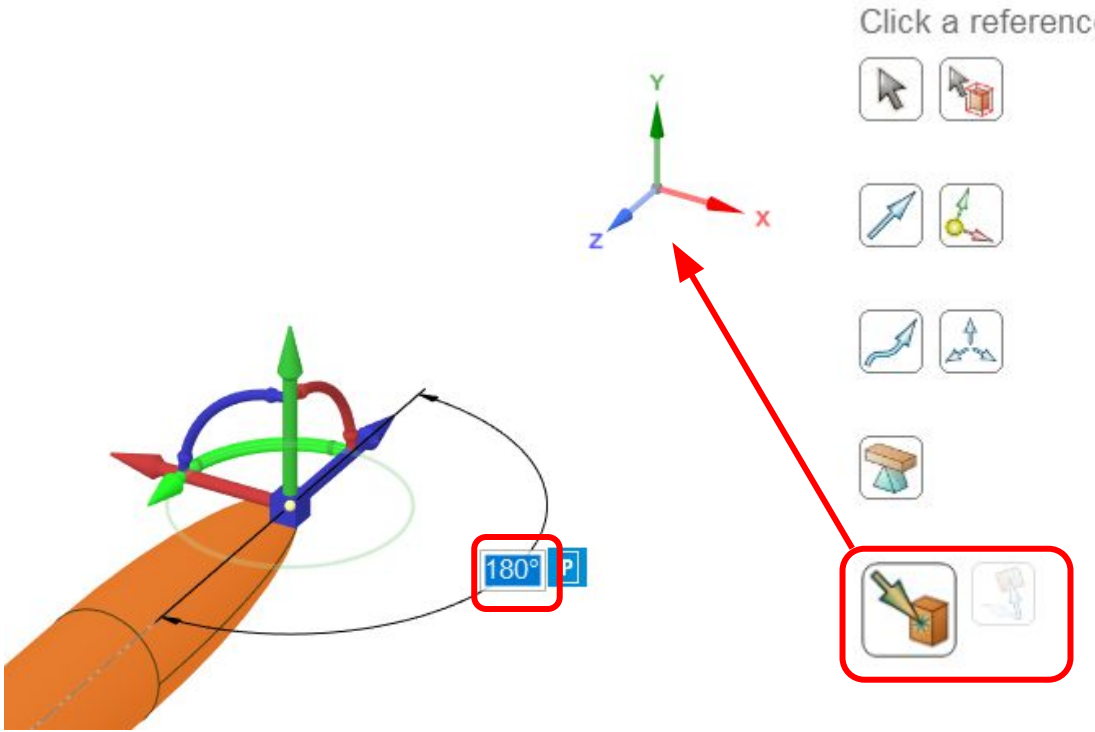
Triple click to select the entire rocket



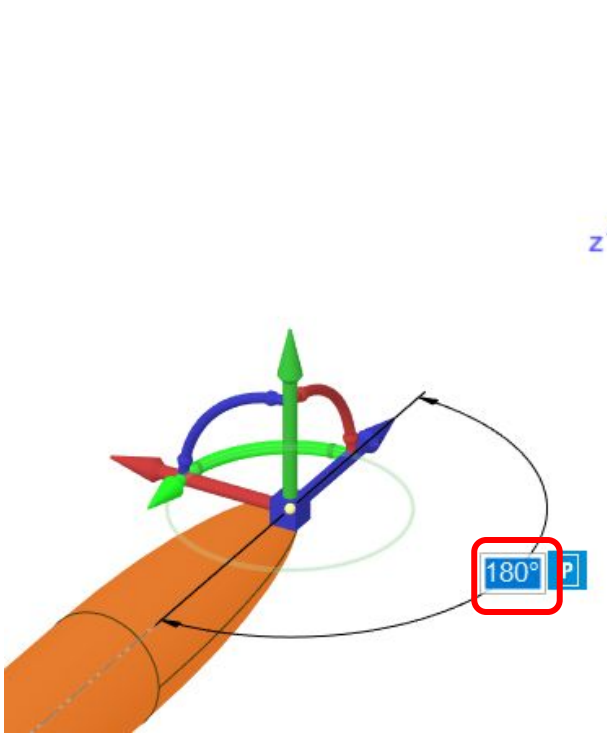
Fix Origin and Orientation

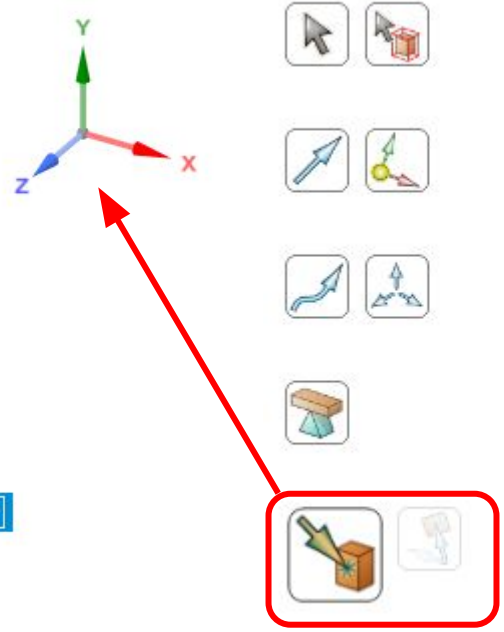


Fix Origin and Orientation

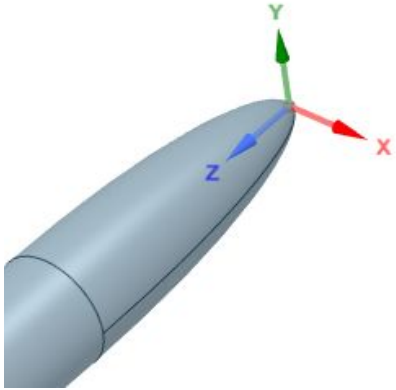


Fix Origin and Orientation

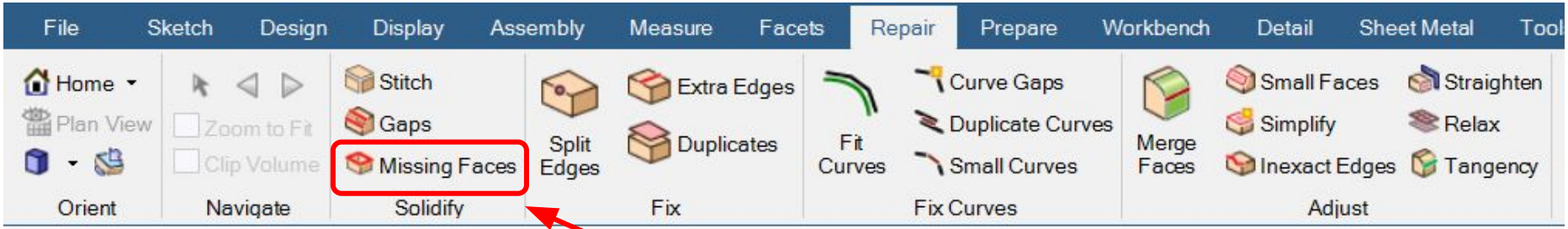




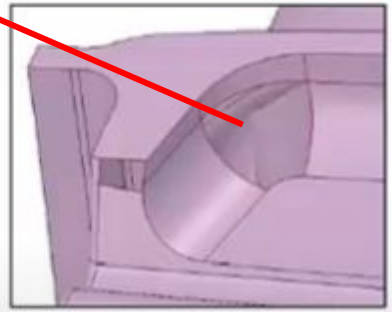
Press **esc** to leave the Move Tool



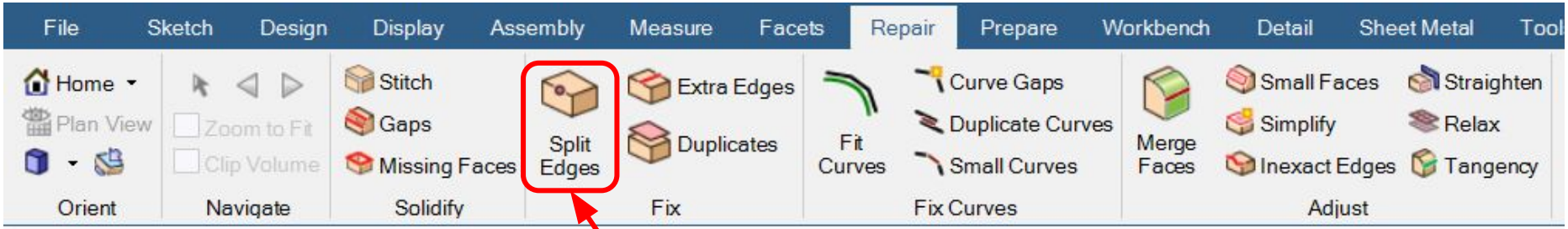
Cleanup - SpaceClaim



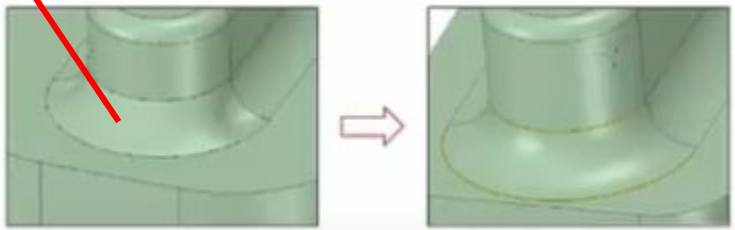
Missing faces



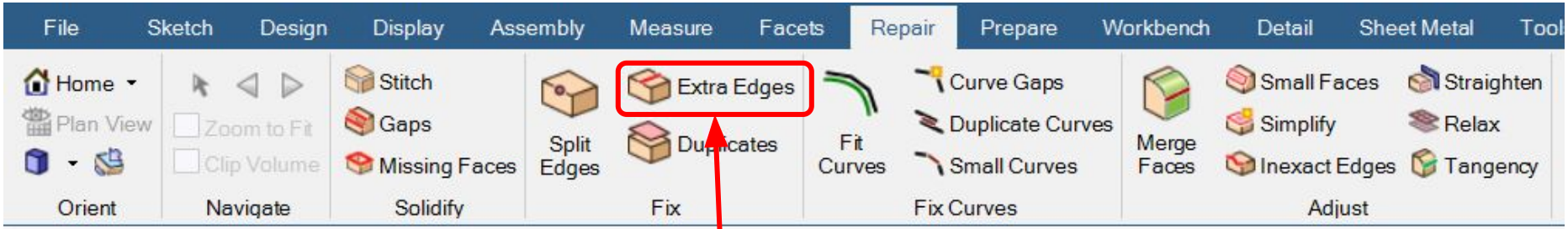
Cleanup - SpaceClaim

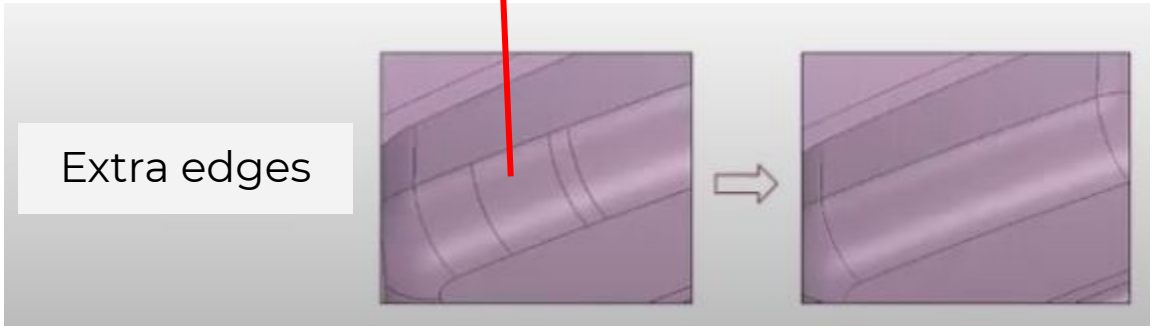


Split edges

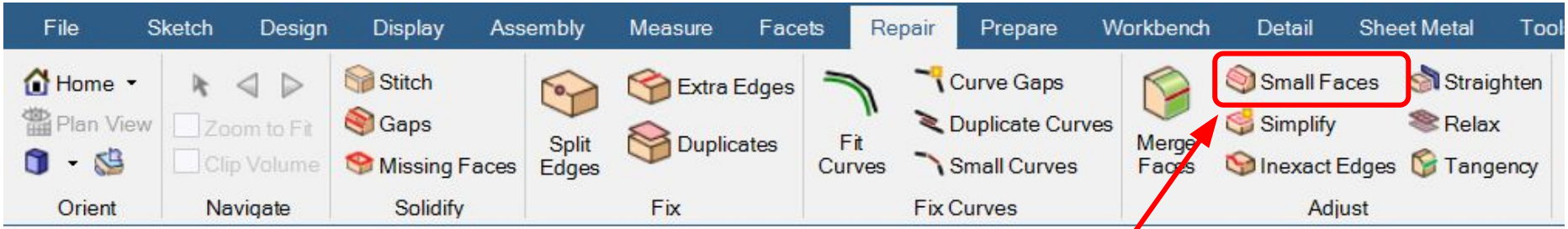


Cleanup - SpaceClaim



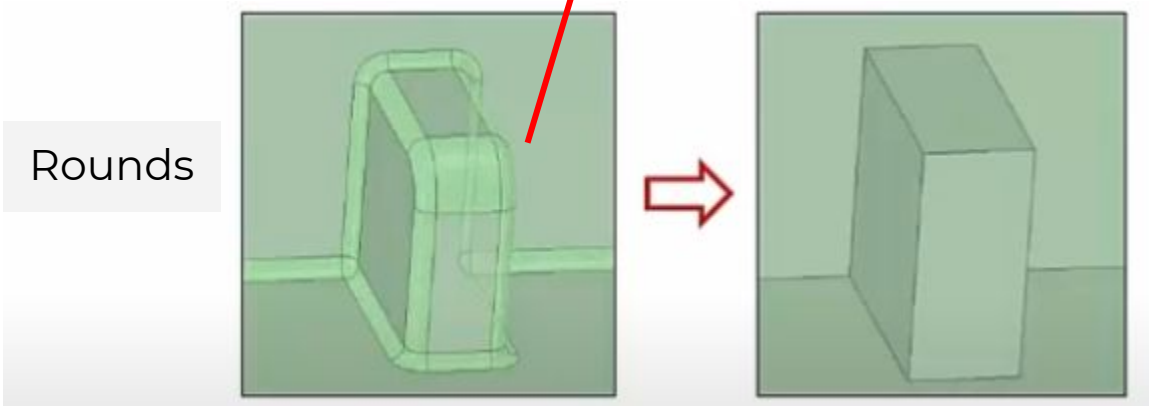


Cleanup - SpaceClaim

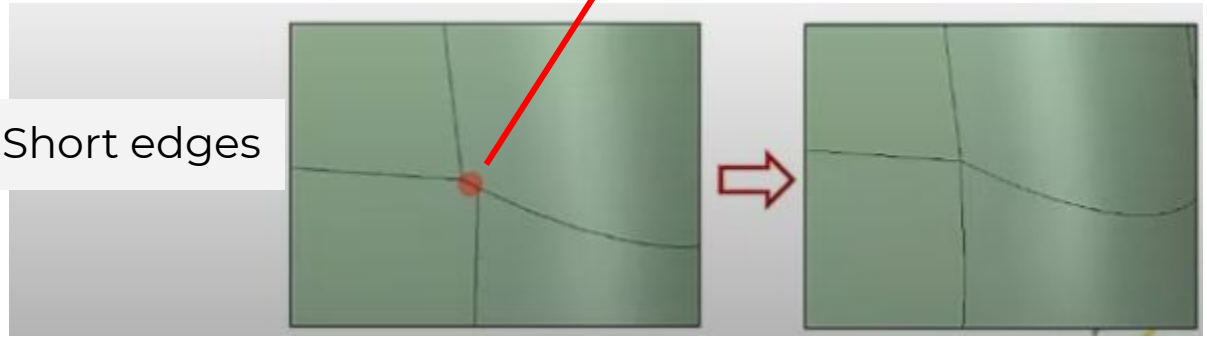


Cleanup - SpaceClaim

Rounds

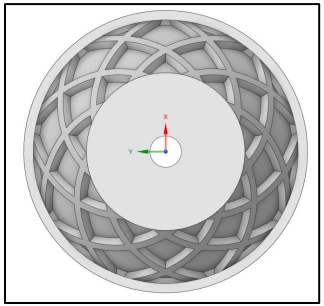


Cleanup - SpaceClaim



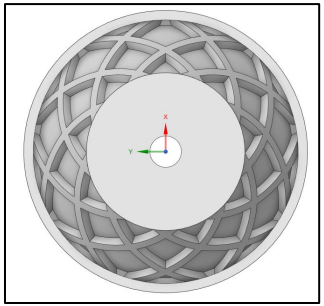
Cleanup - SpaceClaim

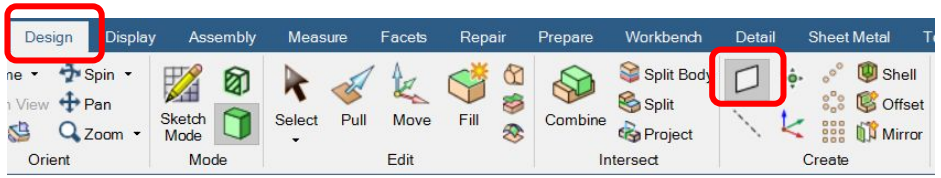
Symmetric object + symmetric **loading condition**?



Cleanup - SpaceClaim

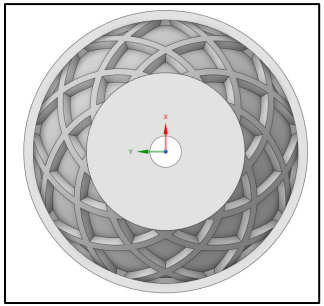
Symmetric object
+ symmetric
loading condition?

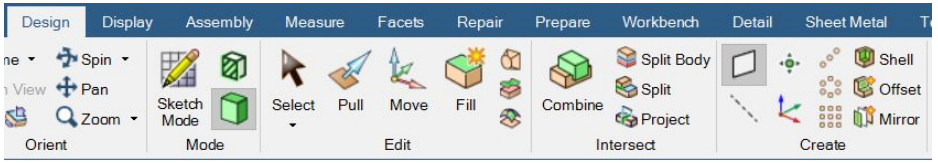


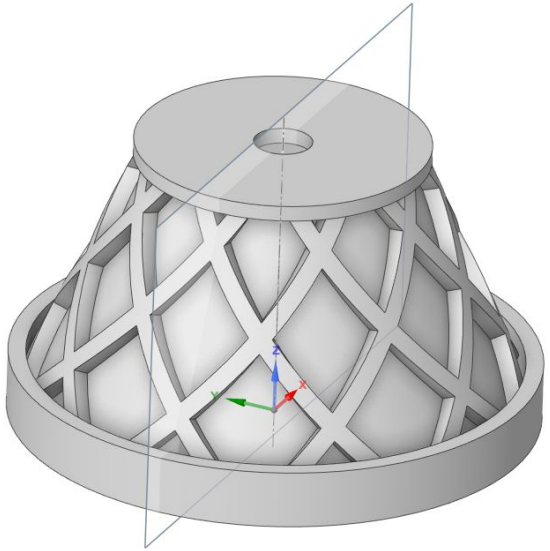


Cleanup - SpaceClaim

Symmetric object
+ symmetric
loading condition?

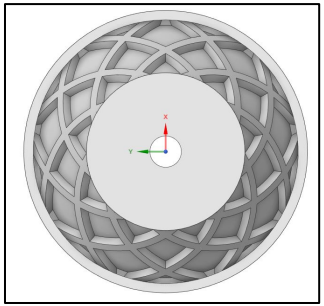


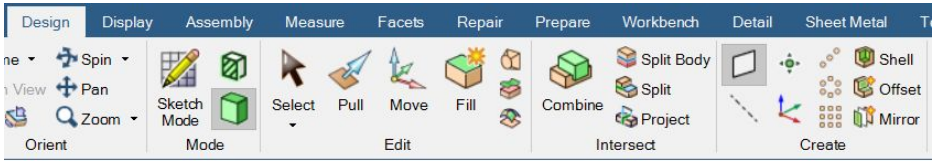


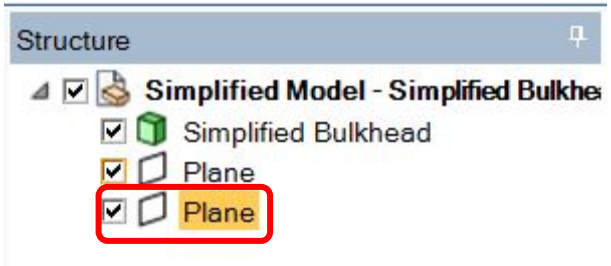


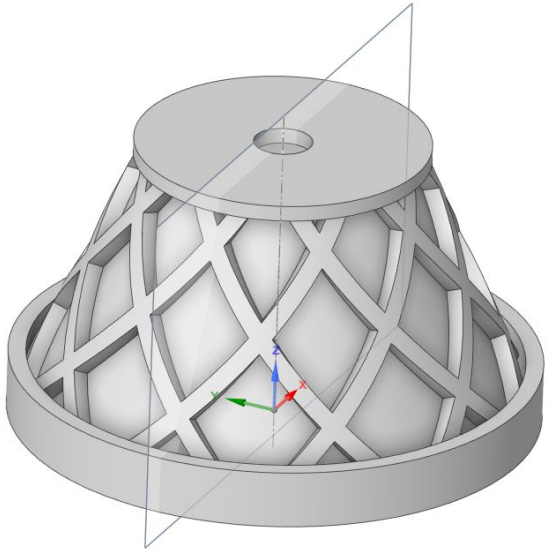
Cleanup - SpaceClaim

Symmetric object
+ symmetric
loading condition?



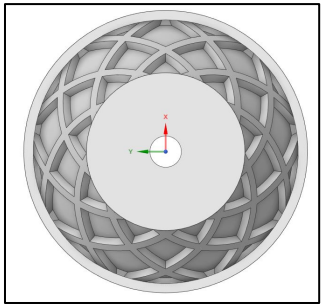


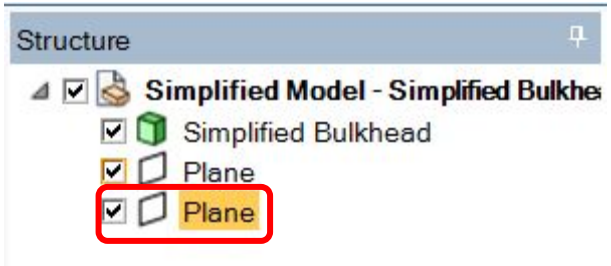


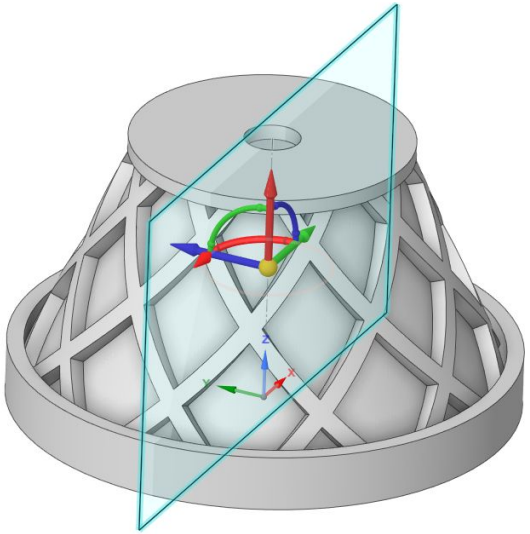


Cleanup - SpaceClaim

Symmetric object
+ symmetric
loading condition?

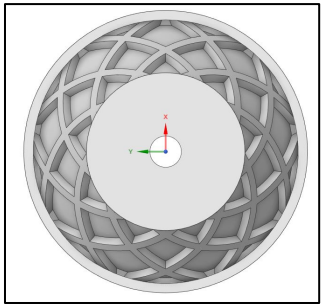


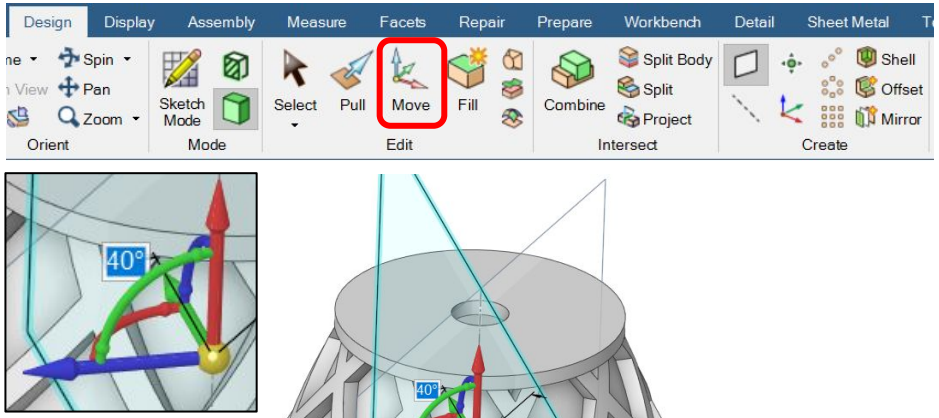


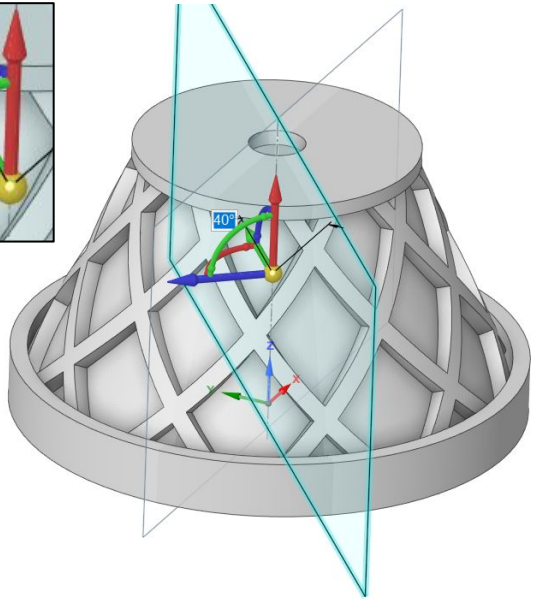


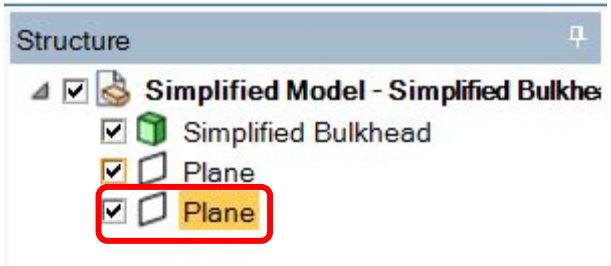
Cleanup - SpaceClaim

Symmetric object
+ symmetric
loading condition?



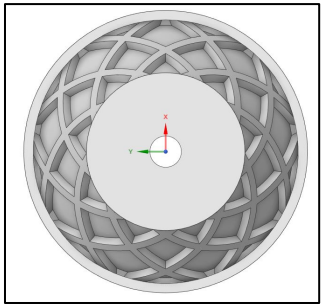


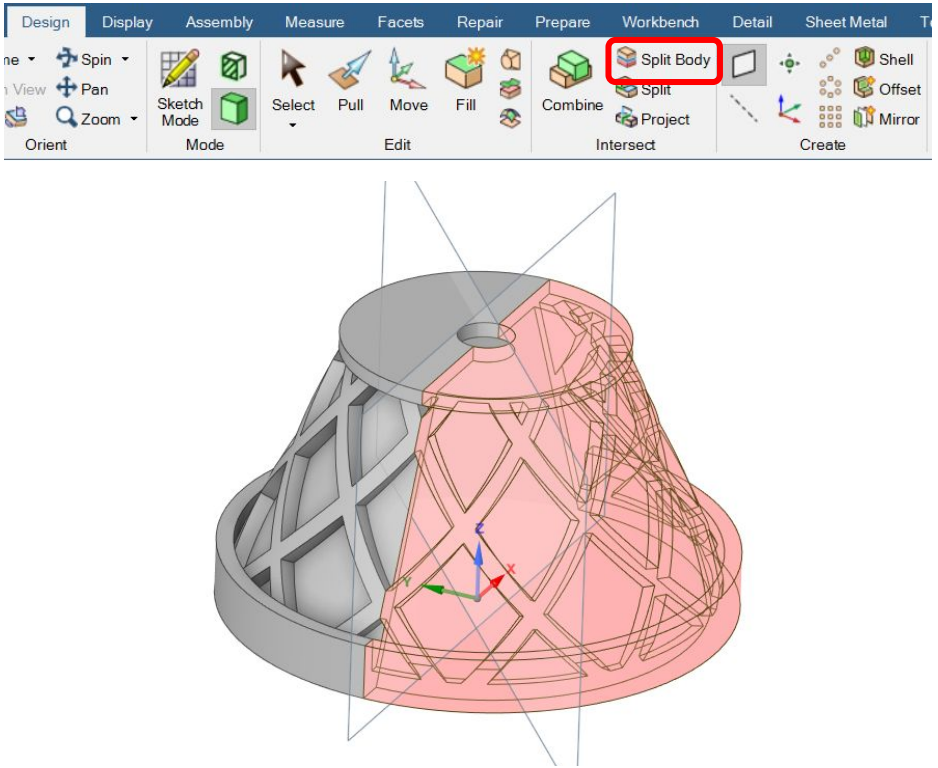




Cleanup - SpaceClaim

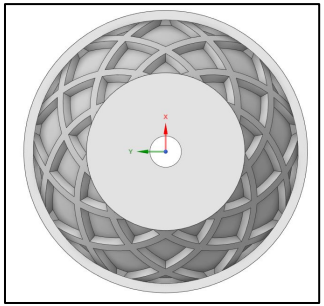
Symmetric object
+ symmetric
loading condition?

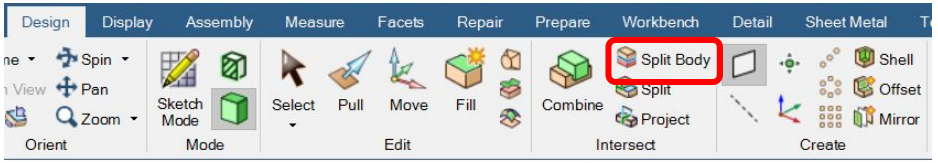


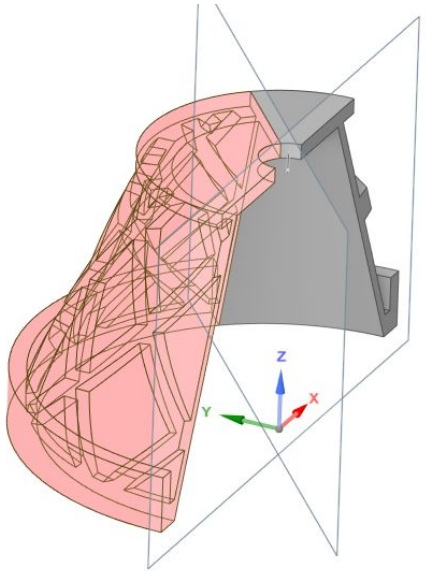


Cleanup - SpaceClaim

Symmetric object
+ symmetric
loading condition?

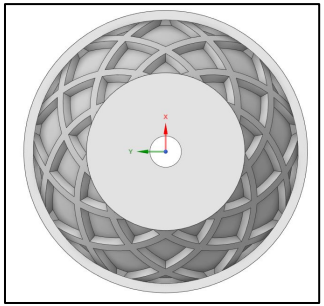


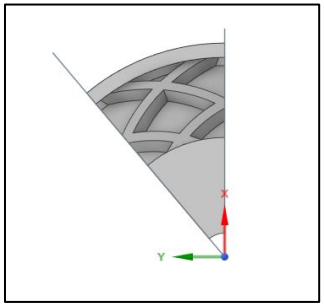


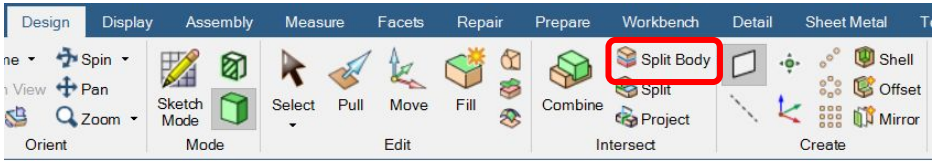


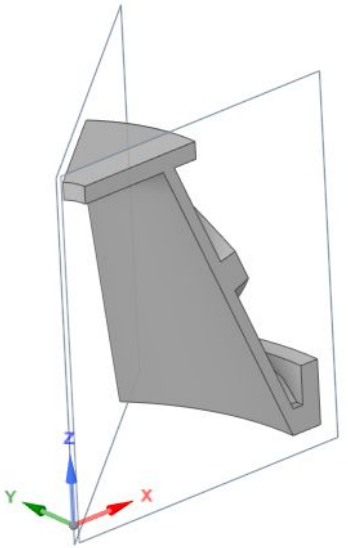
Cleanup - SpaceClaim

Symmetric object
+ symmetric
loading condition?

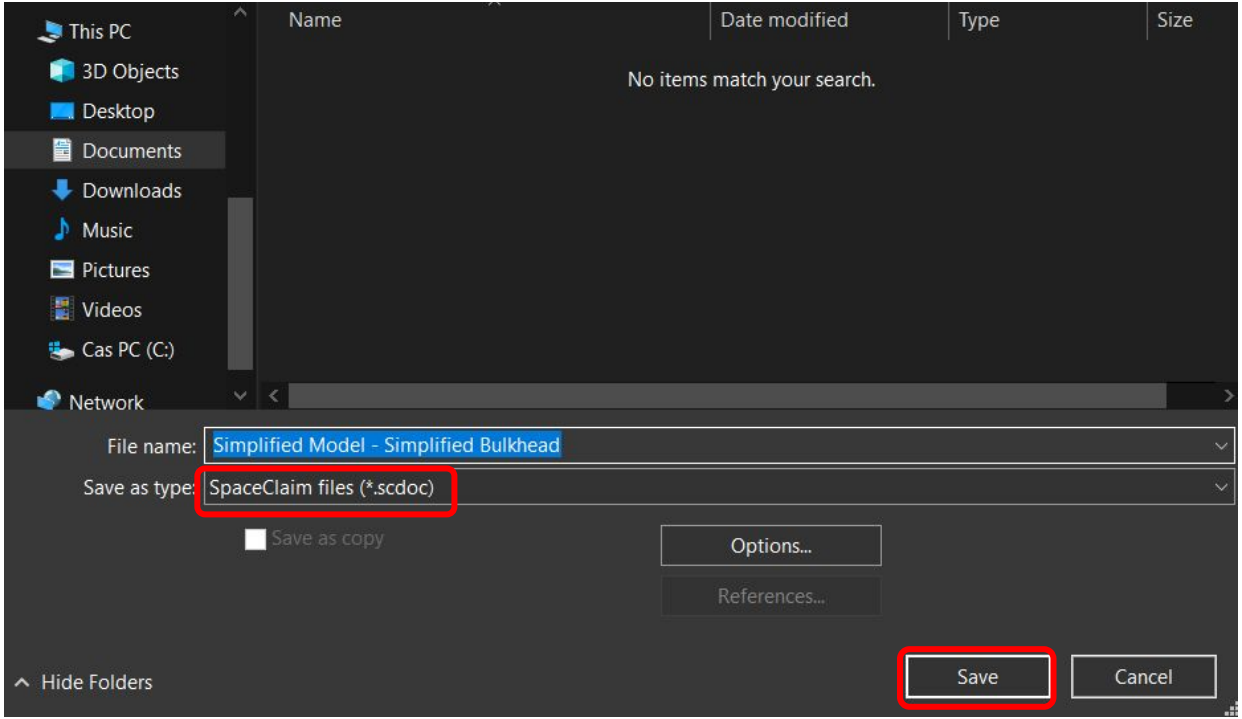


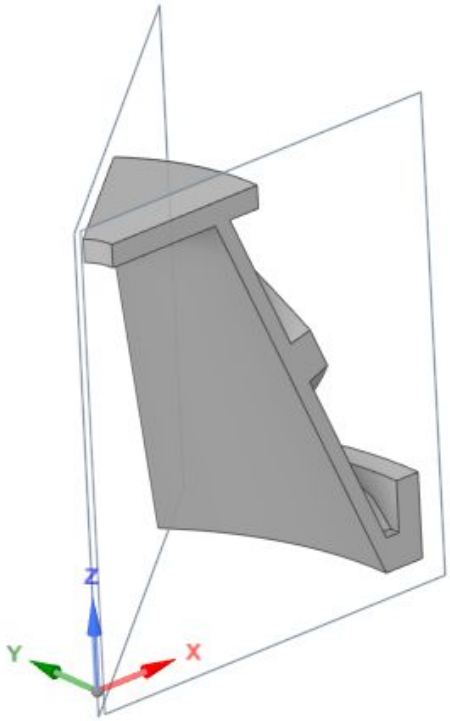




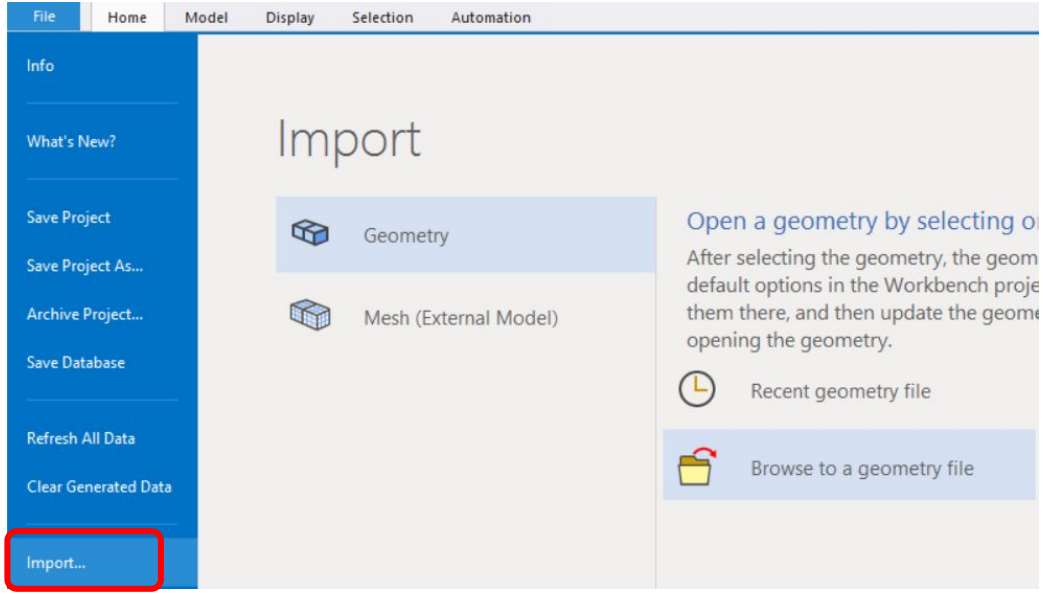


Export to Mechanical

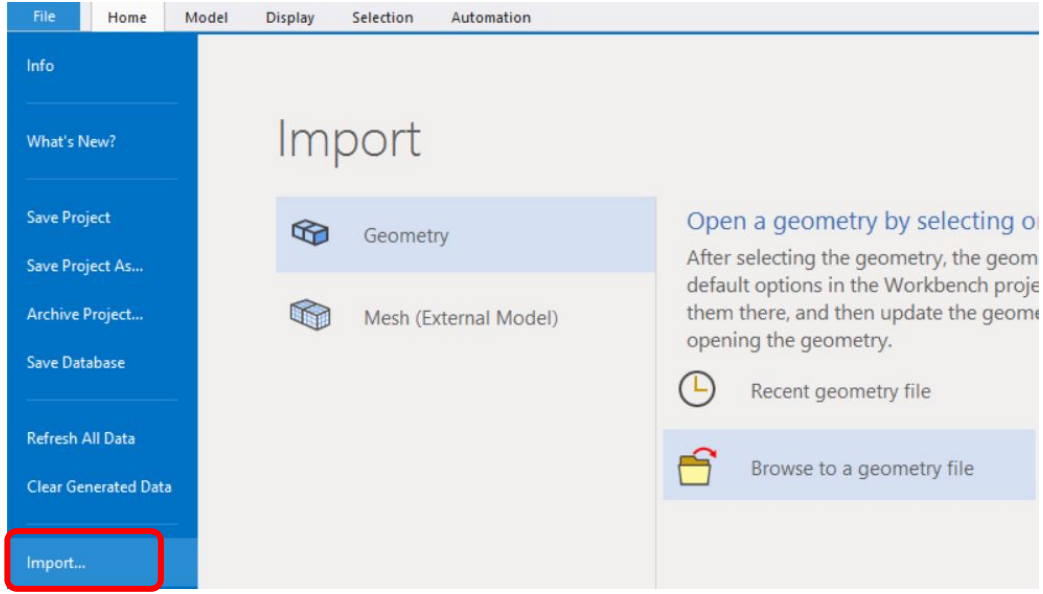


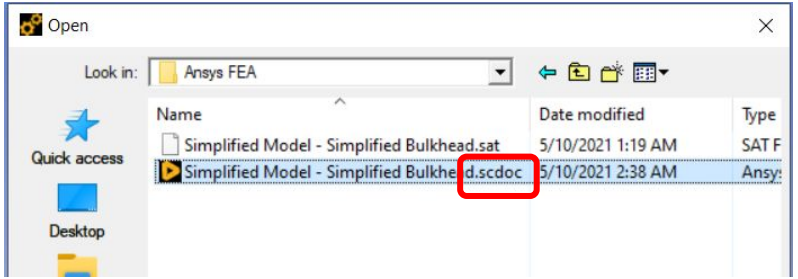


Export to Mechanical

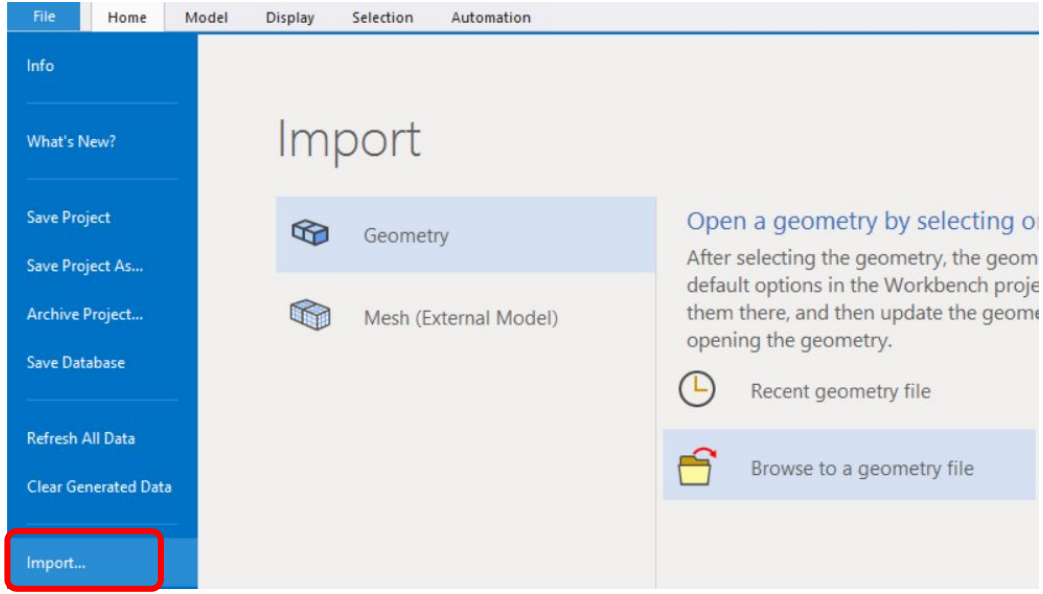


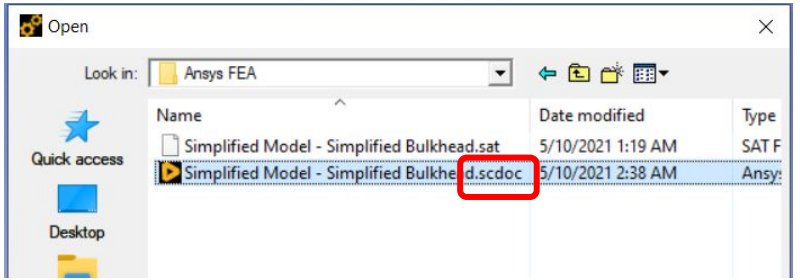
Export to Mechanical

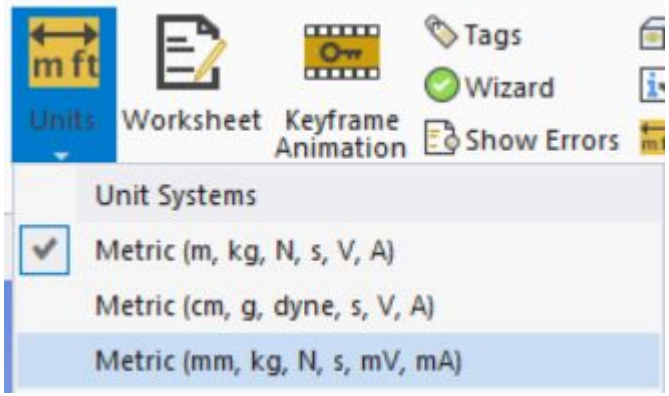




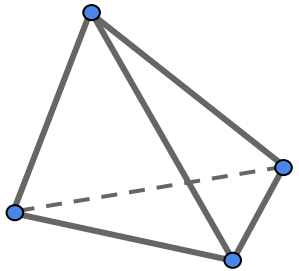
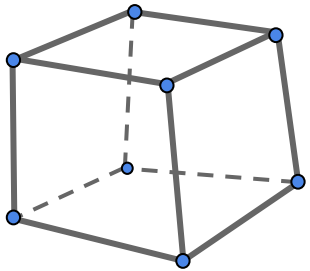
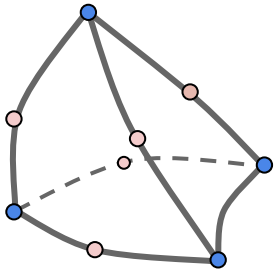
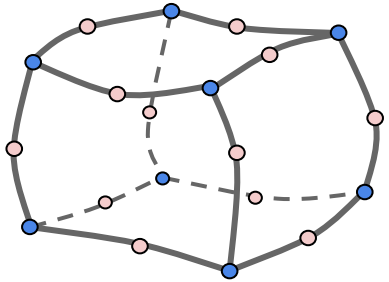
Export to Mechanical







Element Types

	Tetrahedral (tet)		Hexahedral (hex)	
Linear		4 faces 4 nodes		6 faces 8 nodes
Quadratic		4 faces 8 nodes		6 faces 20 nodes

Linear vs Quadratic Mesh

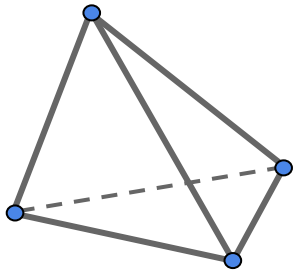
Linear elements suffer from **shear locking** - unable to represent pure bending

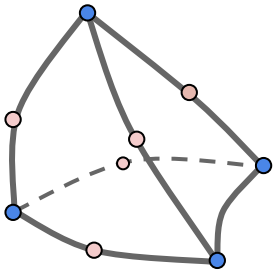
Linear will have **larger error**, especially for **curved geometry** and bending

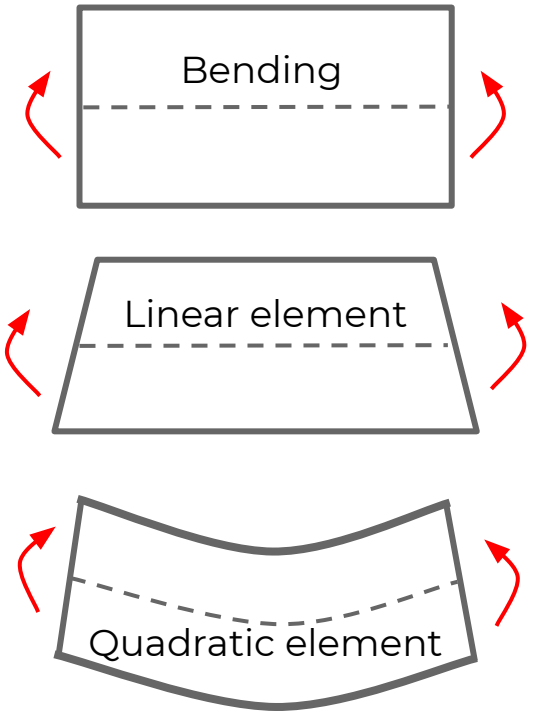
Quadratic may **converge more quickly** as mesh size reduces

Quadratic more **computationally expensive** than linear

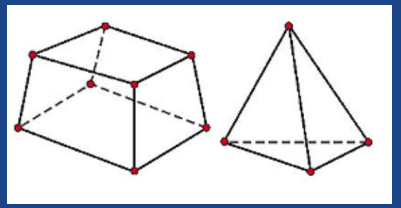
Quadratic **may be overkill** - if the variable in question is linear, use linear elements!







Hex vs. Tet Mesh



Hex can be **structured** (uniform), tet is unstructured.

Hex is generally **more accurate** than tet for the same cell count.

For complex geometry, hex mesh is **more difficult** to achieve than tet.

Hex mesh can be **quicker** to make.

Harder to get high quality hex mesh than tet.

Can also make hybrid meshes with both hex and tet.

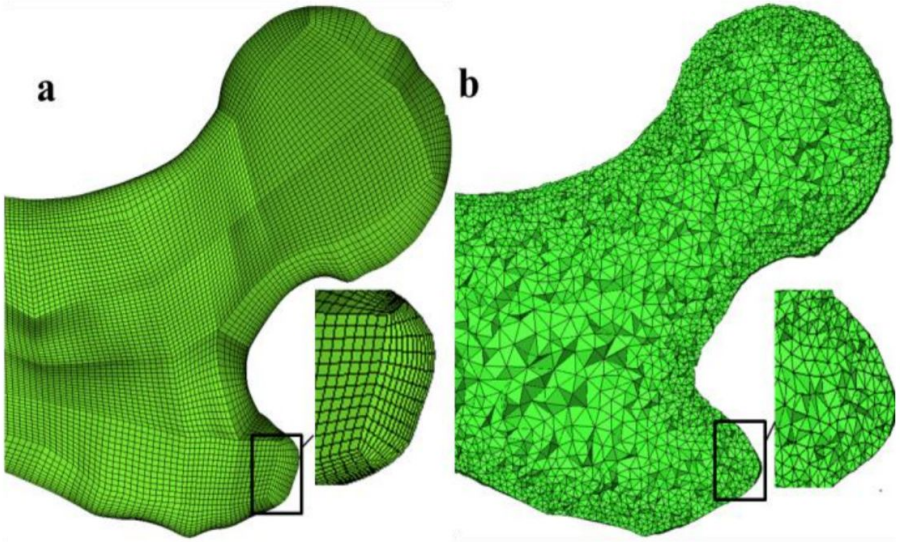
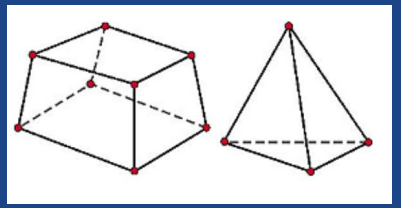


Figure 1: a) Hex, and b) Tet meshes for a femur.

Hex vs. Tet Mesh



ARES

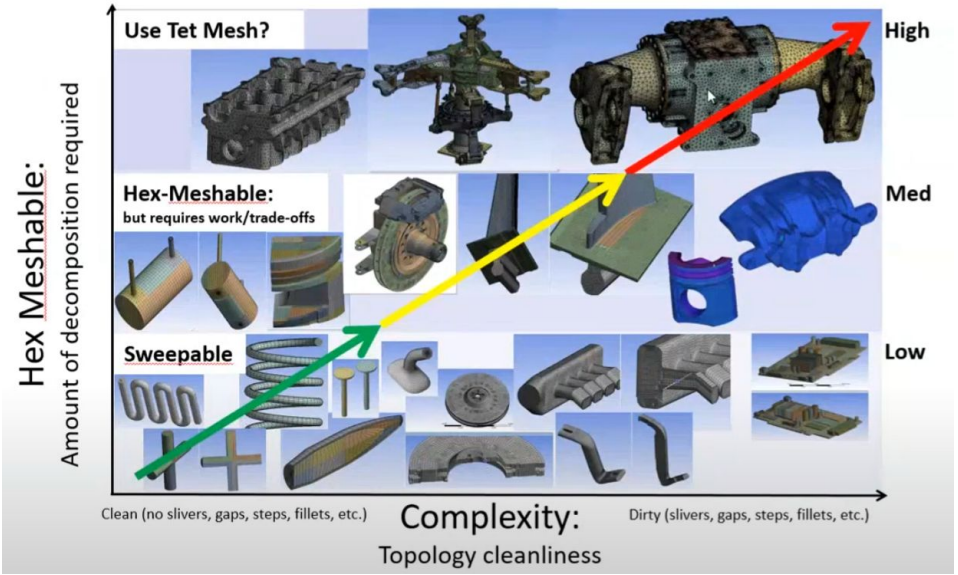
Hex can be **structured** (uniform), tet is unstructured.

Hex is generally **more accurate** than tet for the same cell count.

For complex geometry, hex mesh is **more difficult** to achieve than tet.

Hex mesh can be **quicker** to make.

Can also make hybrid meshes with both hex and tet.



Meshing Advice

Mesh **should match the physics of the problem**

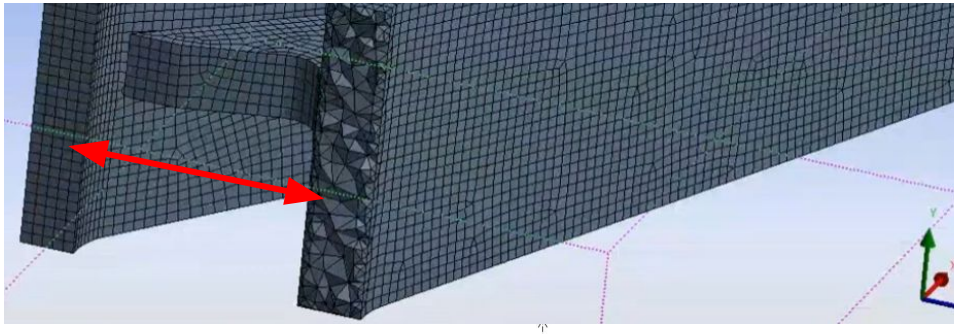
Check the **internal mesh** quality (sections)

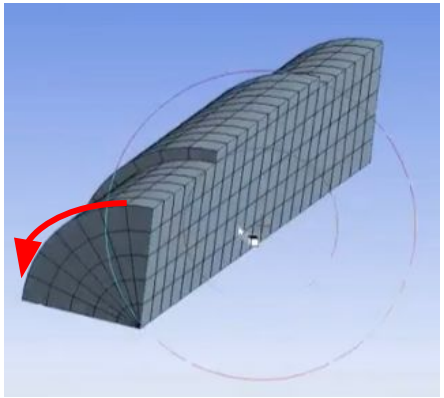
Convert long parts into **1D beam elements** and thin-walled parts into **surface elements**

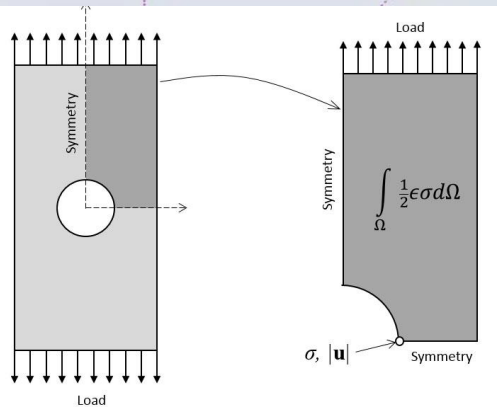
Huge mesh with lots of elements isn't necessarily good

Look for symmetrical and **sweepable** surfaces

Remember **symmetry**, and consider **2D solution** for problems which allow it



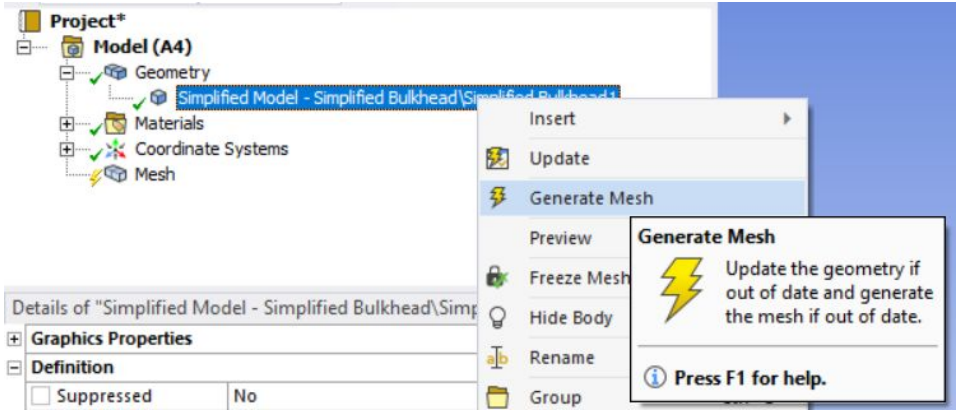


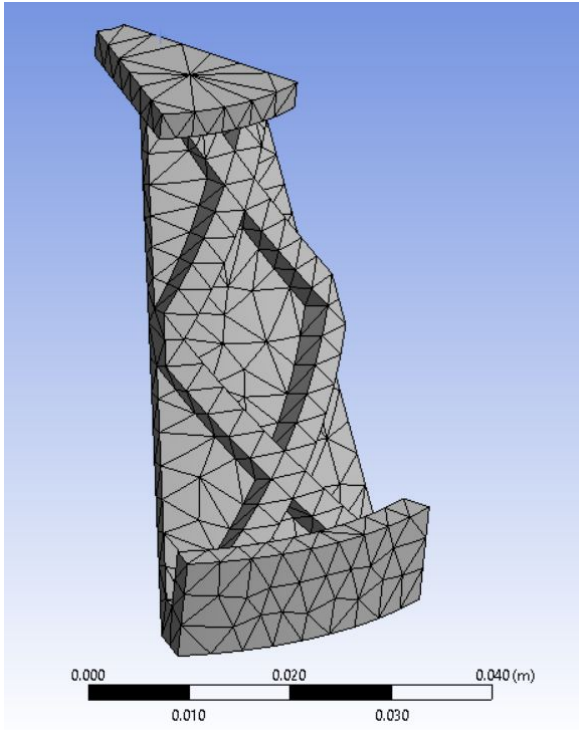


Default Mesh

Can use the **default settings** (without any local fine-tuning to match the part geometry)

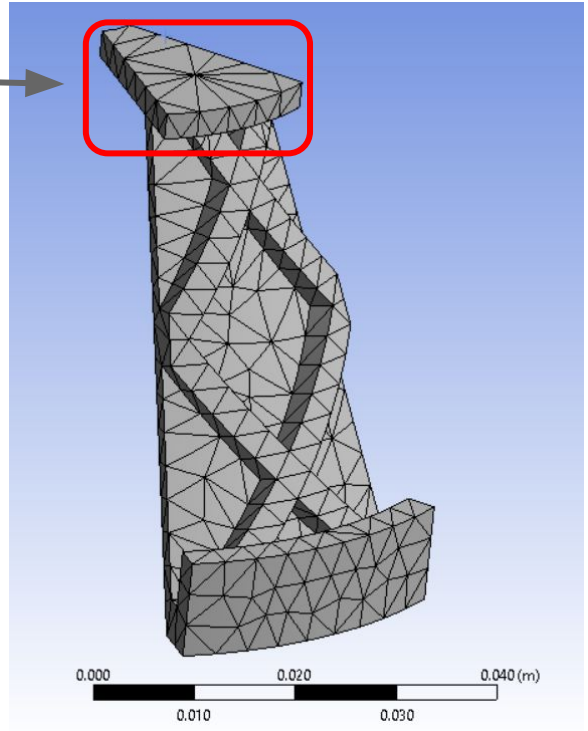
This mesh is generally quite **non-uniform** or poorly well-aligned with the part and loading





Default Mesh

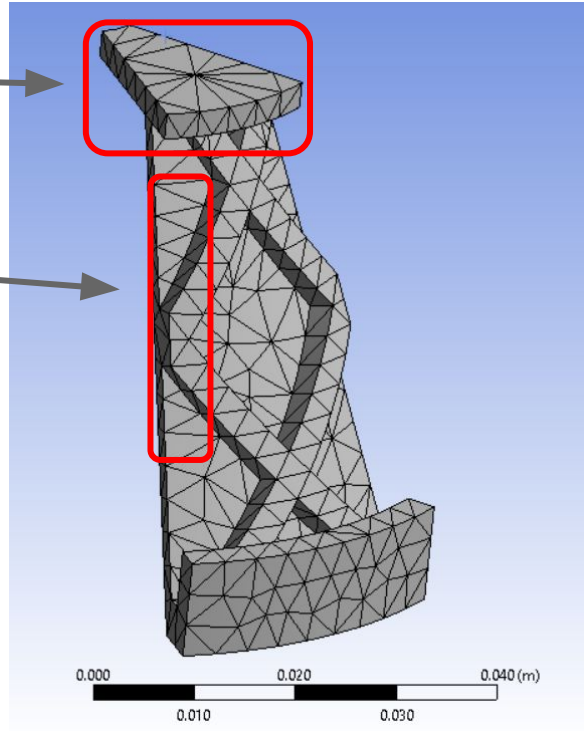
Doesn't utilise **sweeping**



Default Mesh

Doesn't utilise **sweeping**

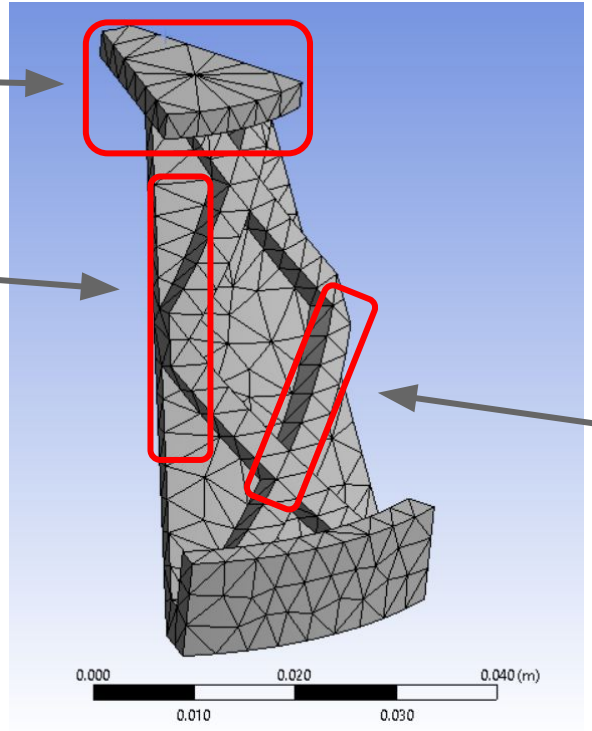
Doesn't align mesh with **loading direction**



Default Mesh

Doesn't utilise **sweeping**

Doesn't align mesh with **loading direction**



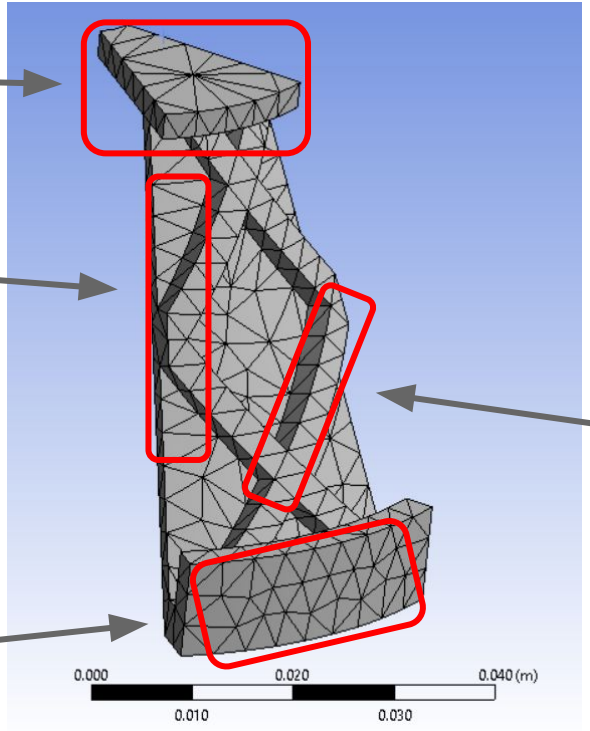
Doesn't **split geometry** into sweepable parts

Default Mesh

Doesn't utilise **sweeping**

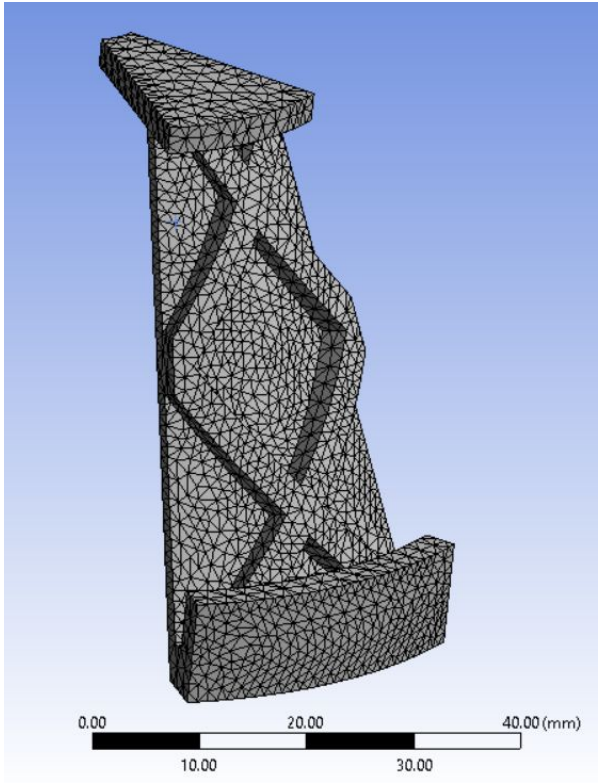
Doesn't align mesh with **loading direction**

Doesn't utilise **hex mesh**



Doesn't **split geometry** into sweepable parts

Global Mesh Controls

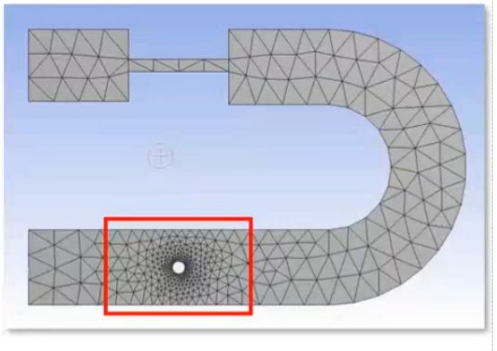


Curvature and Proximity

Curvature sizing limits the angle between adjacent elements

Best used for curved surfaces

Growth rate determines transition to larger mesh away from curvature



Curvature and Proximity

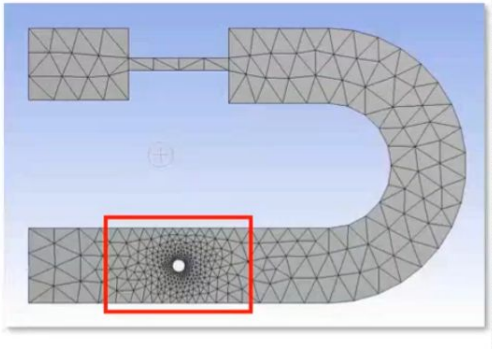
Curvature sizing limits the angle between adjacent elements

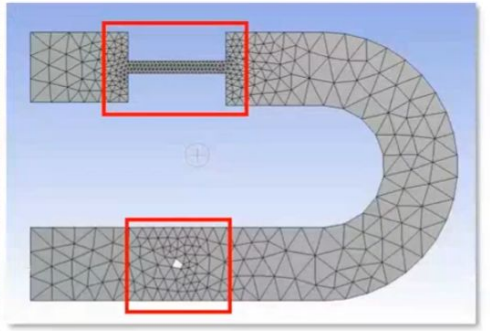
Best used for curved surfaces

Growth rate determines transition to larger mesh away from curvature

Proximity sizing ensures a minimum number of elements across thin faces/narrow gaps

Note this does not apply to the curved surface





Sizing	
Use Adaptive Sizing	No
Growth Rate	Default (1.85)
Max Size	Default (16.0 mm)
Mesh Defeaturing	No
Capture Curvature	Yes
Curvature Min Size	Default (8.e-002 mm)
Curvature Normal Angle	Default (70.395°)
Capture Proximity	Yes
Proximity Min Size	4.0 mm
Num Cells Across Gap	Default (3)
Proximity Size Function Sources	Faces and Edges

Defeaturing

Defeaturing ignores features below a certain size

Default feature size is half the minimum **local** element size

In areas of fine local mesh, features are kept

In areas of coarse local mesh, features are omitted

Defeaturing

Defeaturing ignores features below a certain size

Default feature size is half the minimum **local** element size

In areas of fine local mesh, features are kept

In areas of coarse local mesh, features are omitted

Midsurfacing (Shell Meshes)

3D meshing needs uniform elements across each feature (ideally **3-4 elements across**)

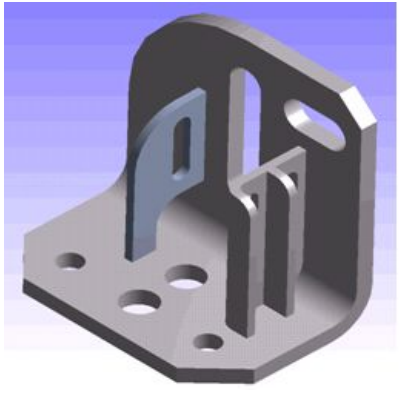
Geometry with **very thin shells → huge number of tiny elements** required for 3D mesh

Modelling thin shells as 3D mesh actually leads to **large errors** in stress, deformation etc.

Much cheaper, **faster** (and more accurate) to model thin shells as **2D elements**

SpaceClaim calls this **“midsurfacing”**

Then use the **extend tool** to meet the faces



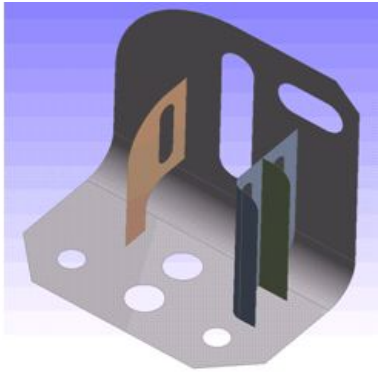
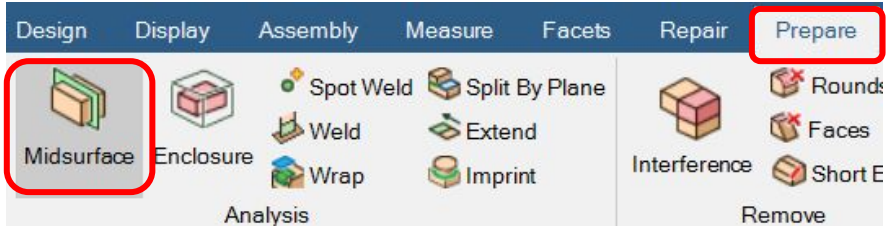


Image source: Ozen Engineering: Automation with the Flexibility to Customize
ozeninc.com/automation-wth-the-flexibility/



Midsurfacing (Shell Meshes)

3D meshing needs uniform elements across each feature (ideally **3-4 elements across**)

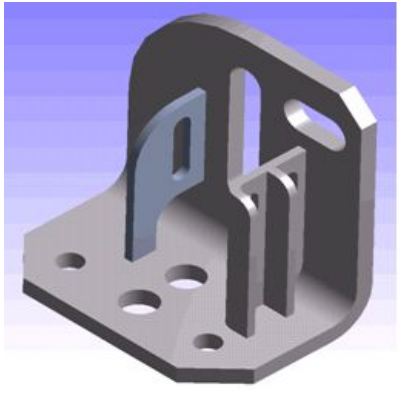
Geometry with **very thin shells → huge number of tiny elements** required for 3D mesh

Modelling thin shells as 3D mesh actually leads to **large errors** in stress, deformation etc.

Much cheaper, **faster** (and more **accurate**) to model thin shells as **2D elements**

SpaceClaim calls this **“midsurfacing”**

Then use the **extend tool** to meet the faces



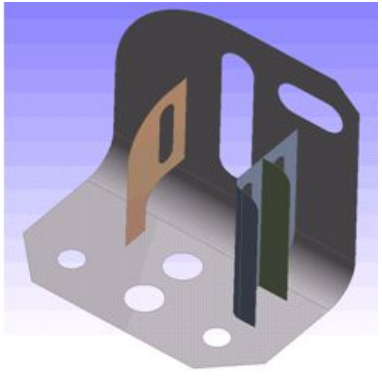
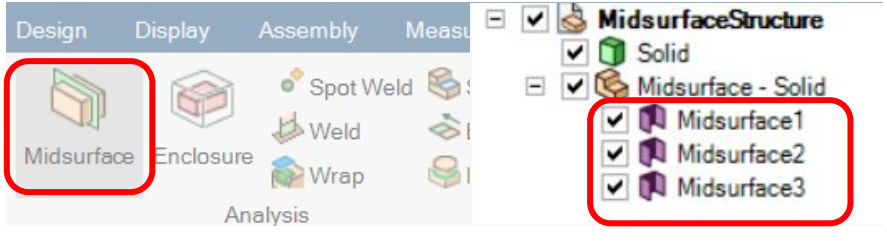


Image source: Ozen Engineering: Automation with the Flexibility to Customize
ozeninc.com/automation-wth-the-flexibility/



Midsurfacing (Shell Meshes)

3D meshing needs uniform elements across each feature (ideally **3-4 elements across**)

Geometry with **very thin shells → huge number of tiny elements** required for 3D mesh

Modelling thin shells as 3D mesh actually leads to **large errors** in stress, deformation etc.

Much cheaper, **faster** (and more accurate) to model thin shells as **2D elements**

SpaceClaim calls this **“midsurfacing”**

Then use the **extend tool** to meet the faces

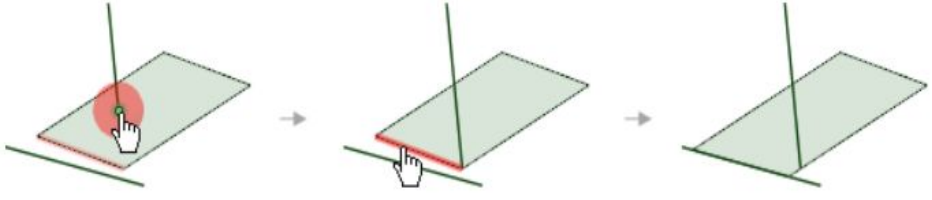


Image source: Ansys SpaceClaim Online Help
http://help.spaceclaim.com/2015.0.0/en/Content/Extend.htm

Beam Extraction

Beams FEA Guide (courtesy Leap Australia)

Trying to model long, thin beams as a solid 3D mesh is undesirable:

- Requires a tiny mesh with many elements. This means **huge solving time**
- Solid mesh gives **massive FEA error** for very long and thin beams

Much more efficient to model beams as **1D objects** which include profile and thickness as internal **parameters**

- More **accurate**
- **Faster** solving, meshing and post-processing

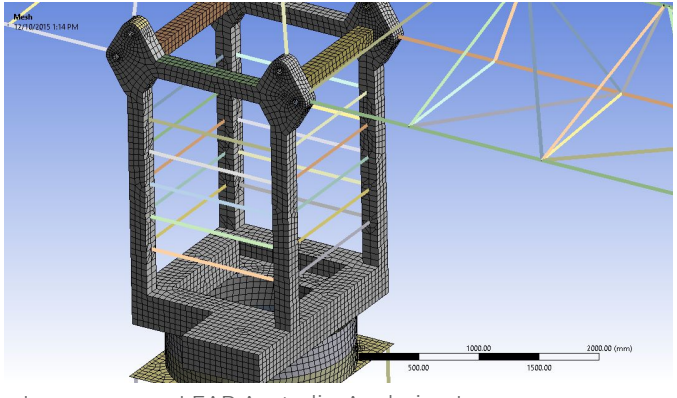
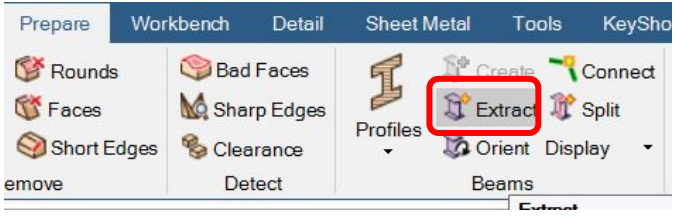
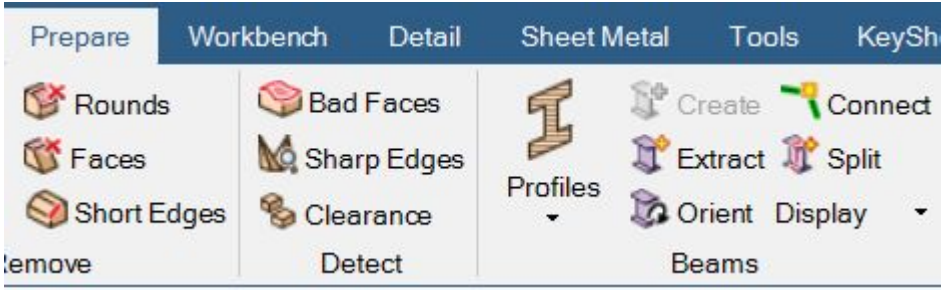


Image source: LEAP Australia: Analysing Large Fabricated Structures finiteelementanalysis.com.au/featured/analysing-large-fabricated-structures/



Beam Extraction

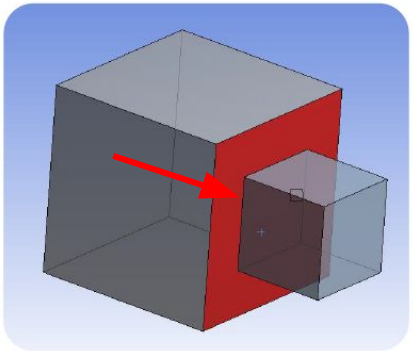
Ansys provides **Beam Extraction** tools



Sharing Topology

For designs with multiple parts, it's best to **share** any contacting surfaces

When two parts don't share the contacting faces, the mesh will be **discontinuous**



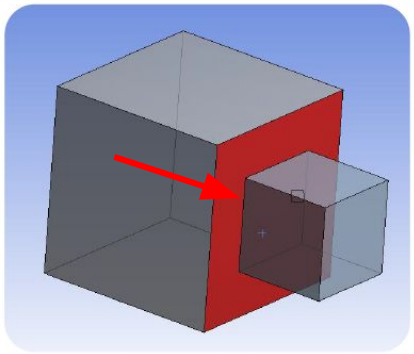
Sharing Topology

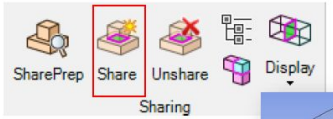
For designs with multiple parts, it's best to **share** any contacting surfaces

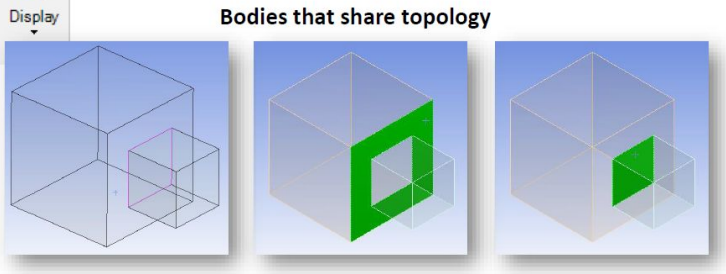
When two parts don't share the contacting faces, the **mesh will be discontinuous**

When the parts share surfaces, the **mesh will be continuous** and the shared surface will be an **internal** surface

This is called "conformally meshed"







Next week - topic

See you next week! :)