MANCHESTER 1824
The University of Manchester

# Investigating the relationship between CAD model complexity and performance trade-offs for fusion neutronics

**Raska Soemantoro**, Zeyuan Miao, Lee Margetts

The University of Manchester, Manchester, UK

UK Atomic Energy Authority, Culham, UK

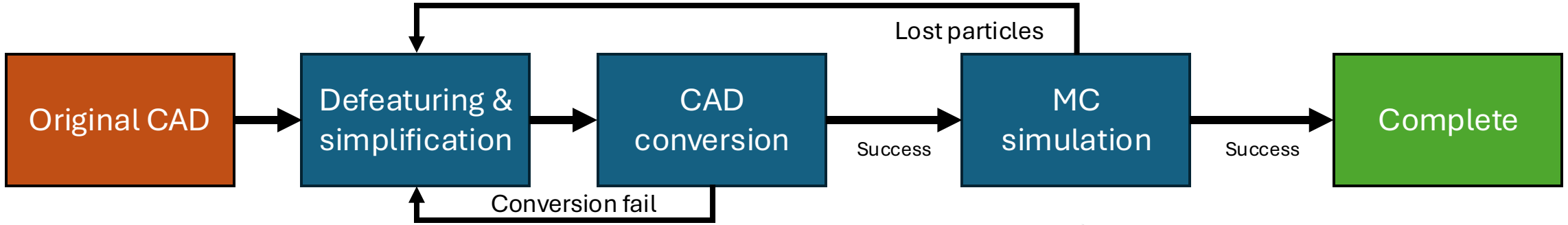
2025 Fusion Neutronics Meeting: ITER and Beyond

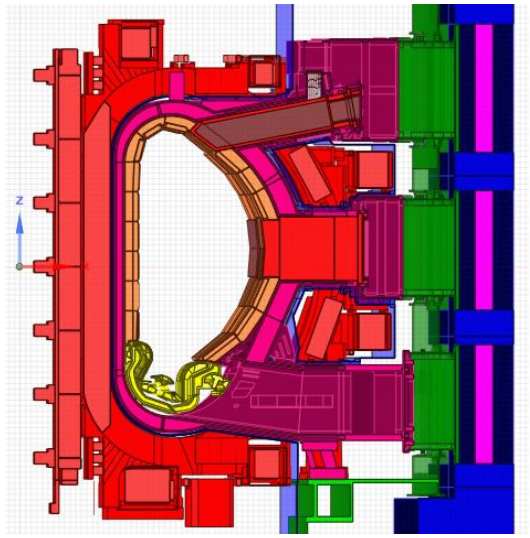
Facultad de Educación - UNED, Madrid, Spain

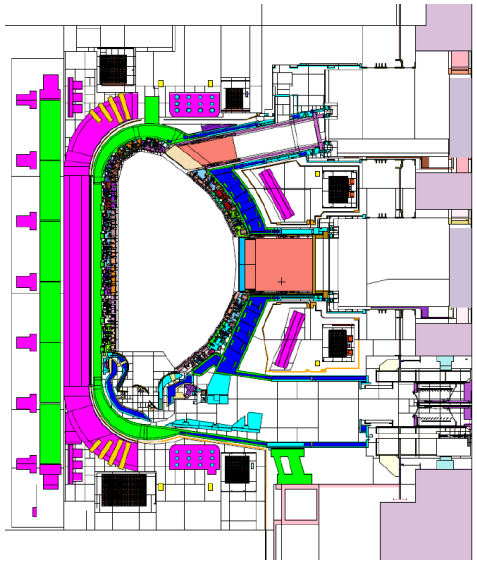
April 2025

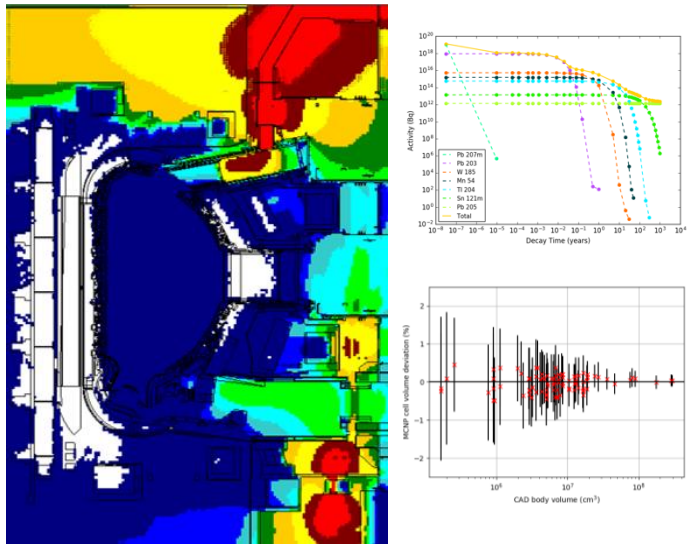
# A typical neutronics workflow





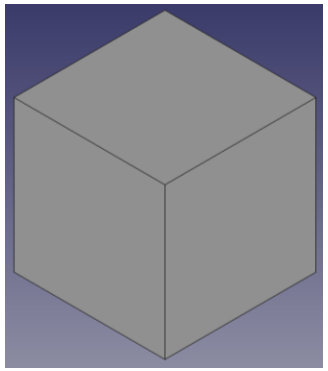




# Measuring complexity in geometry

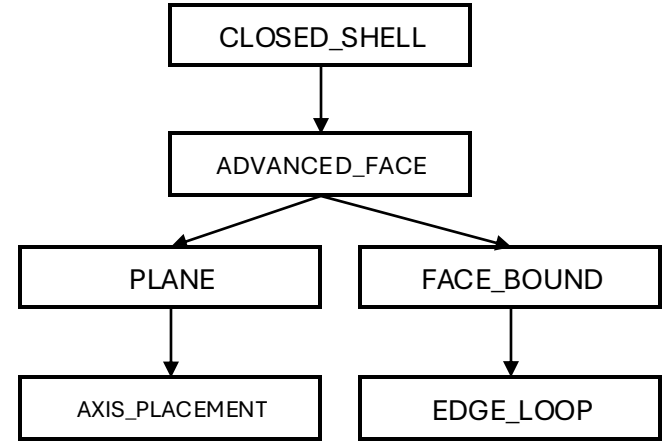
- Real-world models need to be defeatured prior to running a simulation
  - Defeaturing is manual and often error-prone
- We want to automate defeaturing, but that requires understanding geometry complexity
  - We need quantifiable metrics to decide *when* to defeature and *how much*
- To do this..
  - We implement the complexity measurement introduced in [1], comparing quantifiable metrics with so-called expert grading
  - Includes graph- and geometry-based metrics, applying them to STEP files

STEP File



```
. . .
#79 = EDGE_CURVE('',#72,#80,#82,.T.);
#80 = VERTEX_POINT('',#81);
#81 = CARTESIAN_POINT('',(10.,10.,10.));
#82 = LINE('',#83,#84);
#83 = CARTESIAN_POINT('',(10.,10.,0.));
#84 = VECTOR('',#85,1.);
#85 = DIRECTION('',(0.,0.,1.));
#86 = ORIENTED_EDGE('',*,*,#87,.F.);
#87 = EDGE_CURVE('',#64,#80,#88,.T.);
#88 = LINE('',#89,#90);
#89 = CARTESIAN_POINT('',(10.,0.,10.));
. . .
```

Graph representation



# Methodology: quantifying model complexity

The University of Manchester

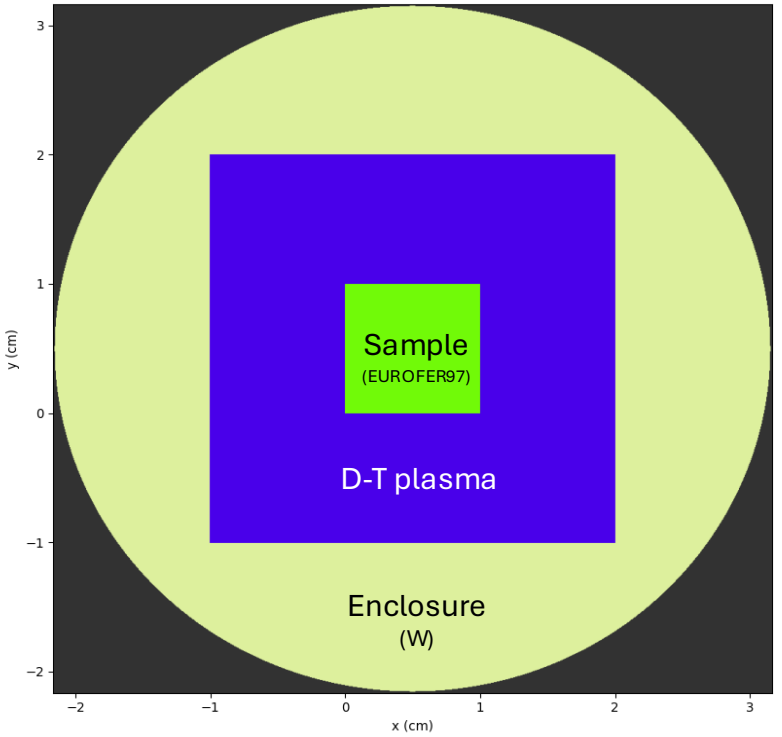
| Metric name | Description | Spearman (ρ) (to practitioner grading) |
|---|---|---|
| Cyclomatic complexity | No. of independent paths through the graph | 0.804 |
| Kolmogorov complexity | String length when the graph is encoded as a binary string | 0.801 |
| Graph entropy | Description of graph uncertainty | 0.799 |
| Graph size | No. of nodes in the graph | 0.792 |
| Number of faces | Total number of individual faces in model | 0.762 |
| Graph dependencies | No. of edges in the graph | 0.755 |
| Number of vertices | Total number of individual vertices in model | 0.713 |
| Cube ratio | 1–aCube/Surface, aCube is area of a cube with the same volume as the model | 0.428 |
| Sphere ratio | 1–aSphere/Surface, aSphere is area of a sphere with the same volume as the model | 0.428 |
| Volume ratio | 1–Volume/bBoxVol, bBoxVol is volume of the bounding box of the model | 0.414 |
| Volume/area ratio | Ratio of surface area/object volume | -0.06 |



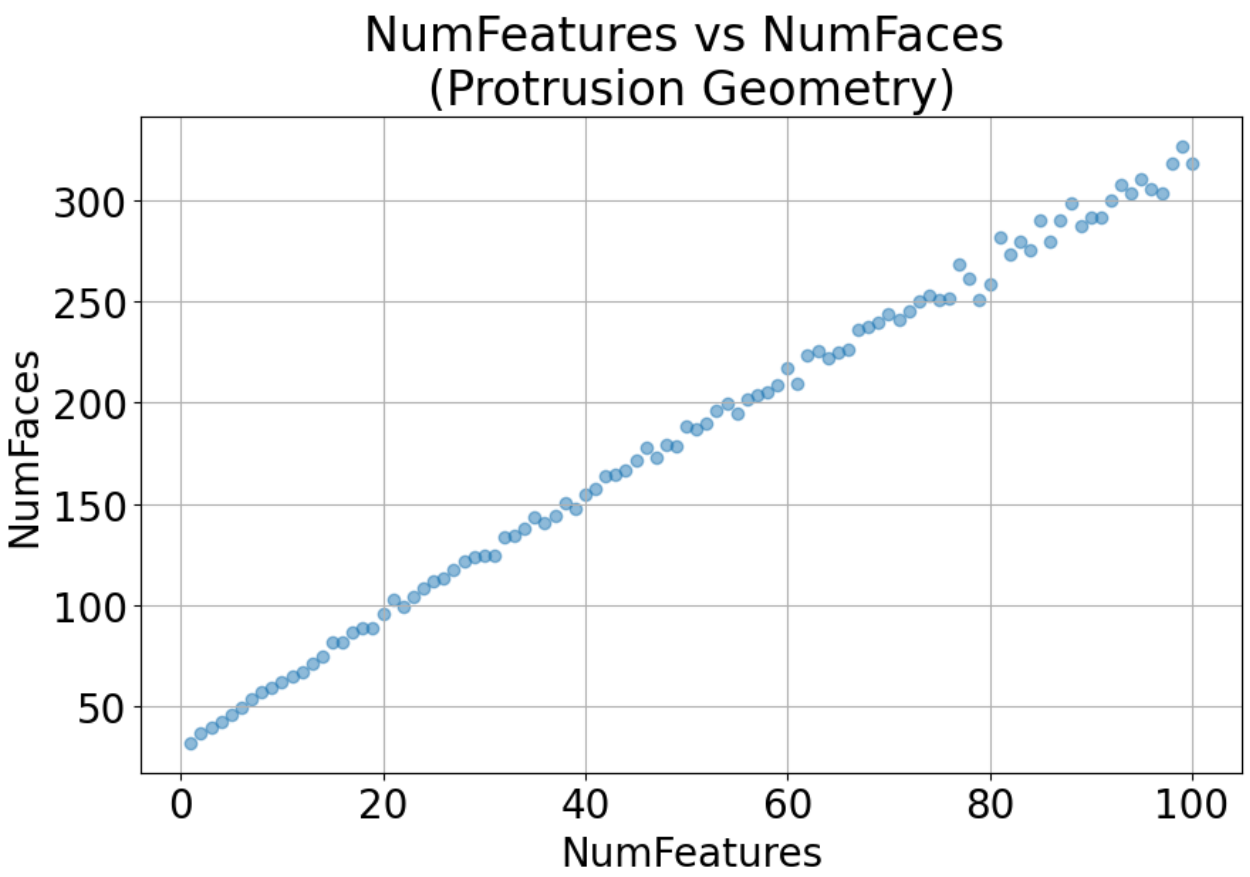
Values from [1]

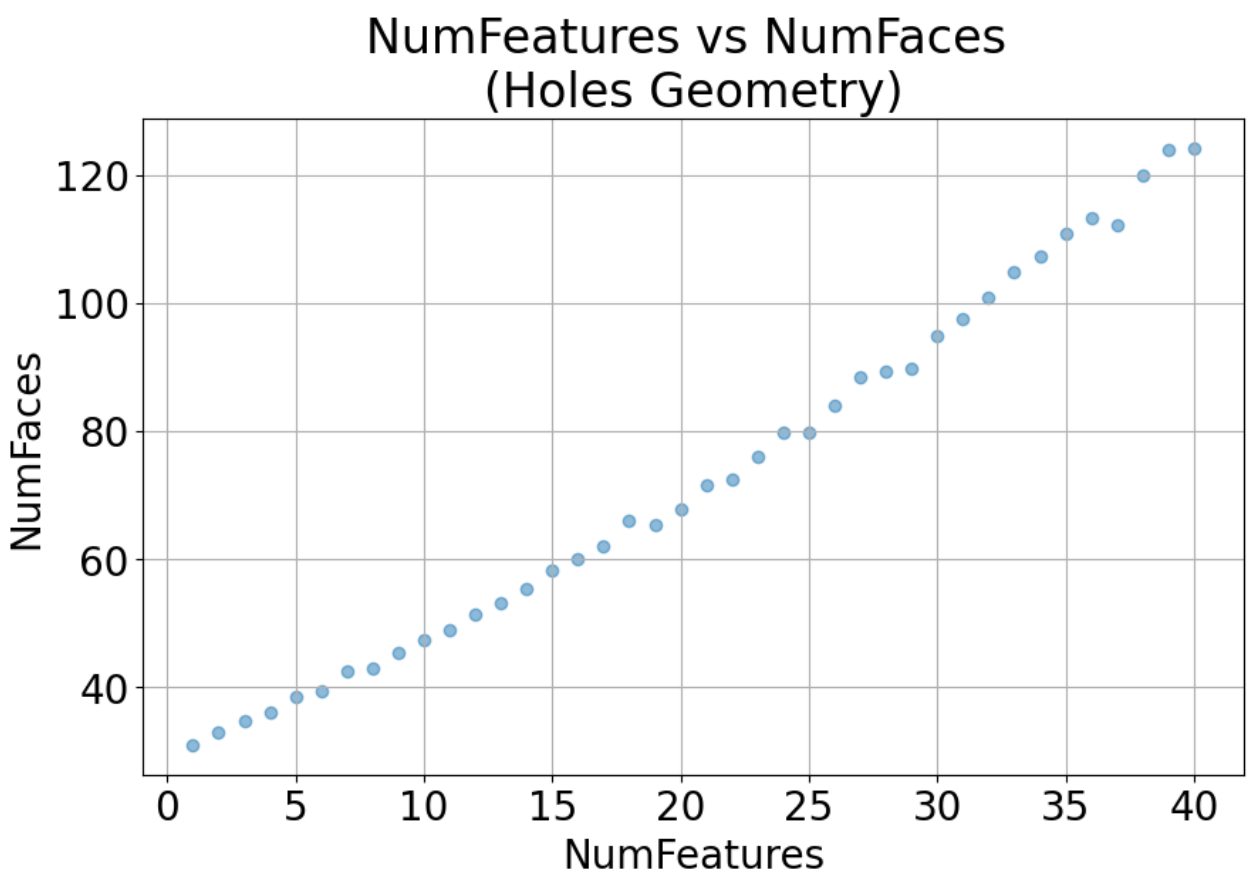
# Methodology: simulation of a toy case

- Case: A ‘box’ of plasma with the sample geometry at centre
- Input:
  - Geometry from the MFCAD++ dataset [2], converted to CSG using GEOUNED[3]
  - 2 sets: one with protrusions and another with holes
- Simulation configurations
  - Source: Isotropic IndependentSource, located at (0,0,0)
    - Energy: Muir distribution @14.1MeV mean, $M_{rat}$=5AMU, $k_t$=20000eV
  - Materials:
    - Enclosure: W (100%),
    - Sample: EUROFER97 (composition as per [4])
  - Environment:
    - ARCHER2, standard partition, 16 cores per task
- Measured simulation outcome:
  - Tallies:
    - Flux: VITAMIN-J-175 energy group
  - Runtime
    - Measured as per OpenMC timing variables



# Results: starting simple

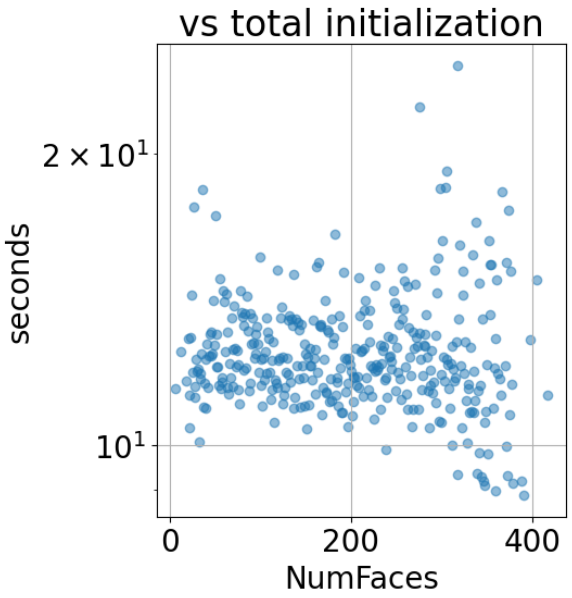


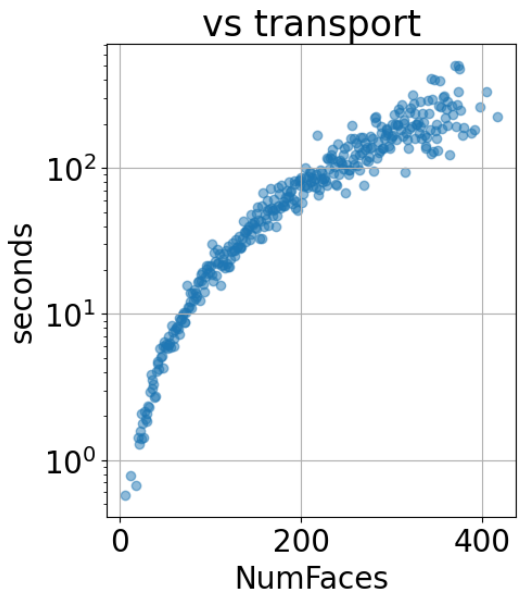



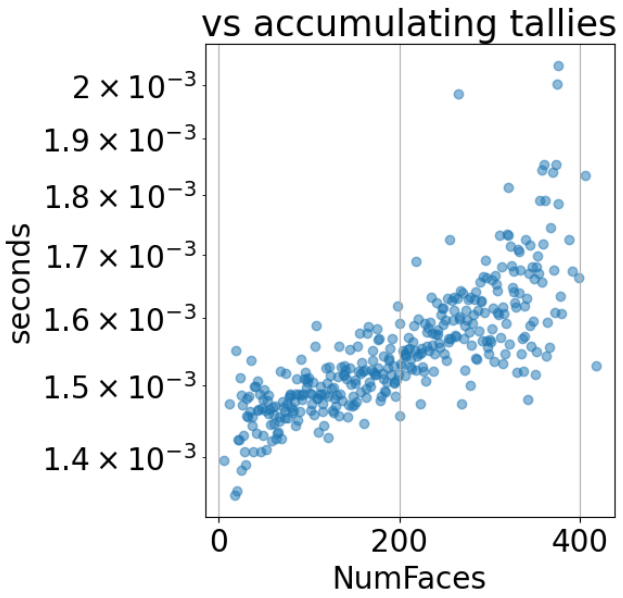

More features -> more faces (easy enough to understand!)

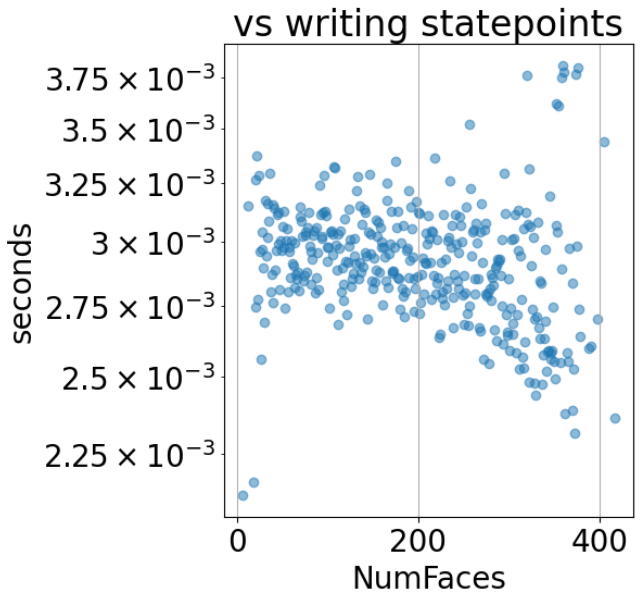
# Results: runtime (protrusions)

NumFaces vs..



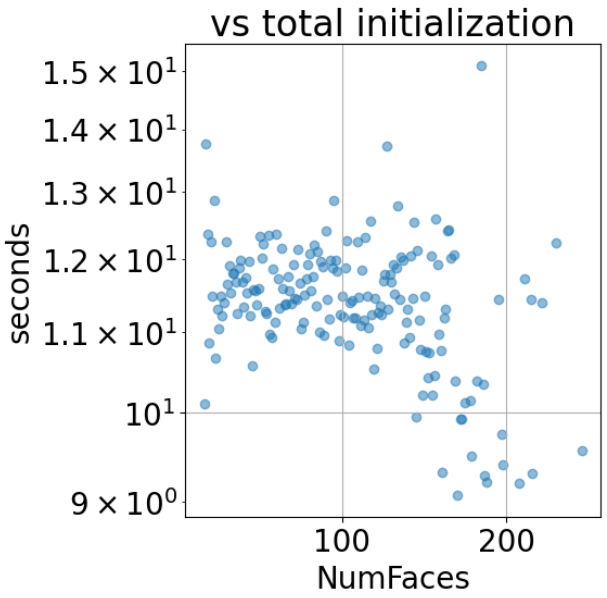



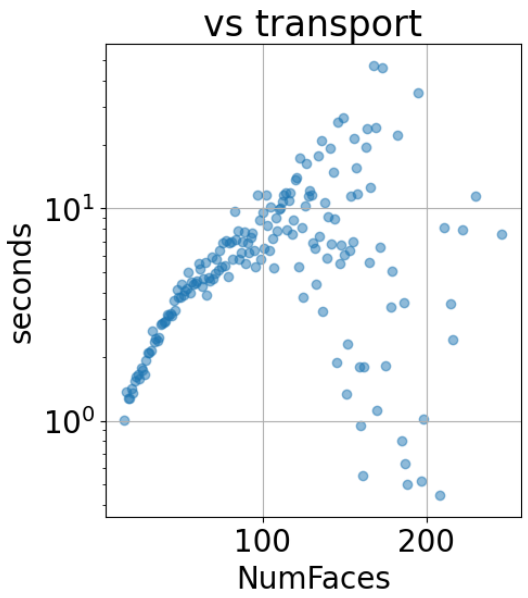



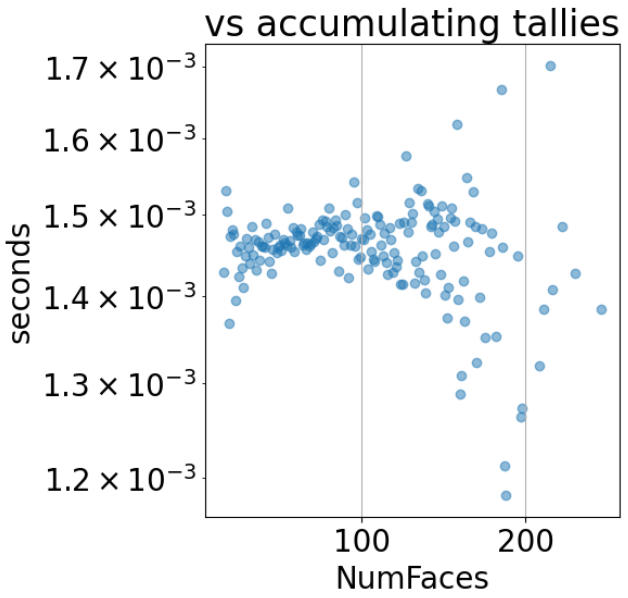

More faces -> more runtime (but only in transport-related phase of simulation)

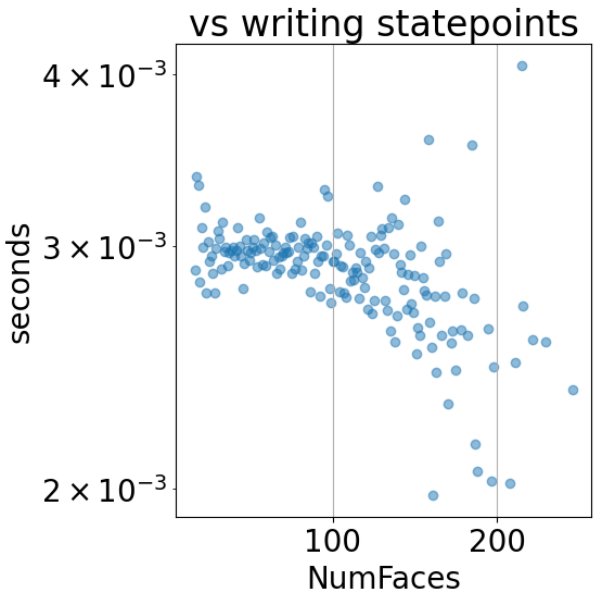
# Results: runtime (holes)

NumFaces vs..



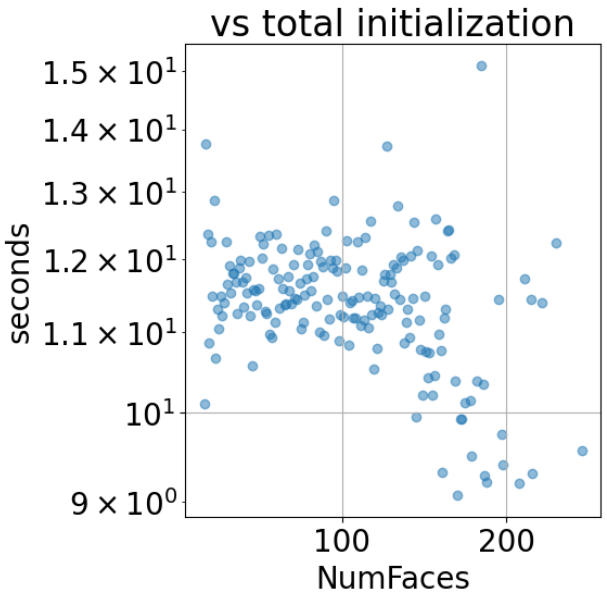



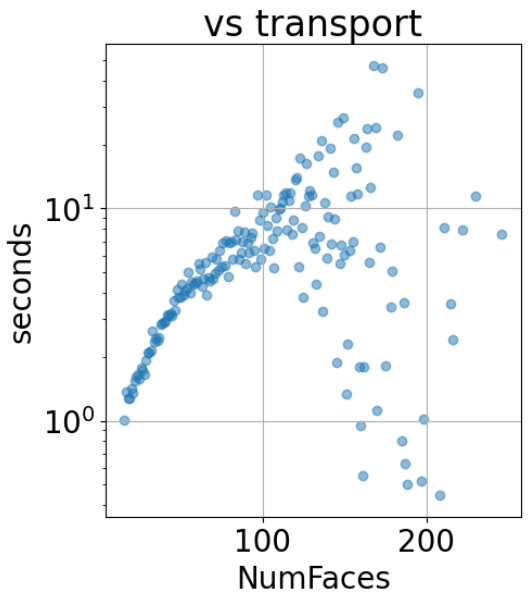



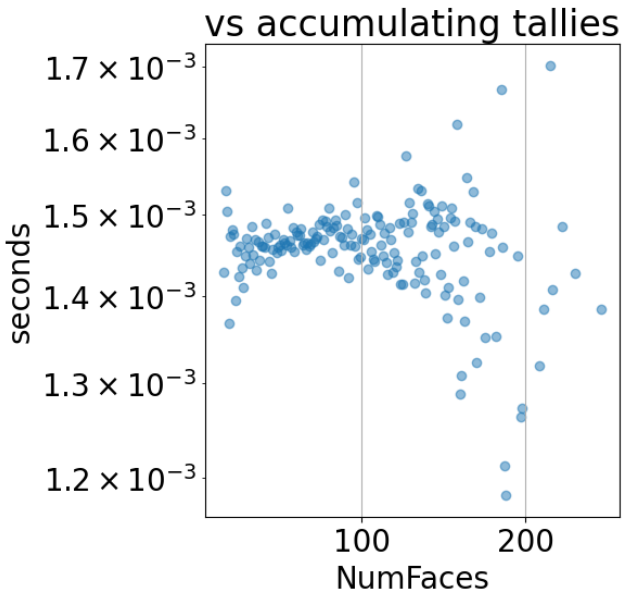

Similar phenomenon here, but transport runtime drops at a steep gradient

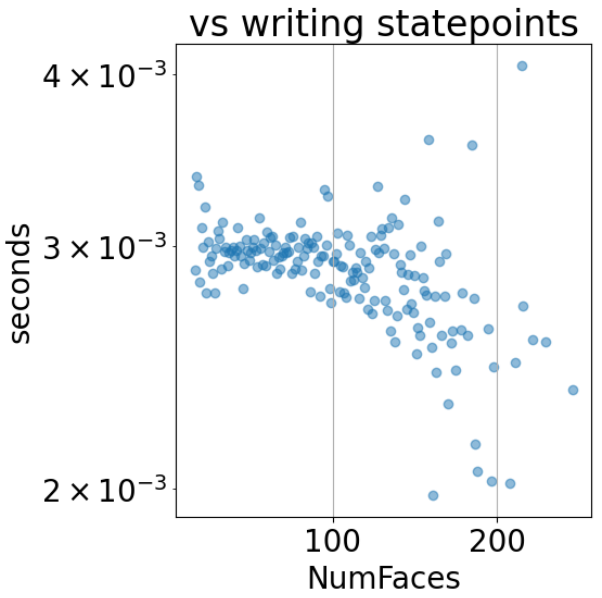
# Results: runtime (holes)

NumFaces vs..



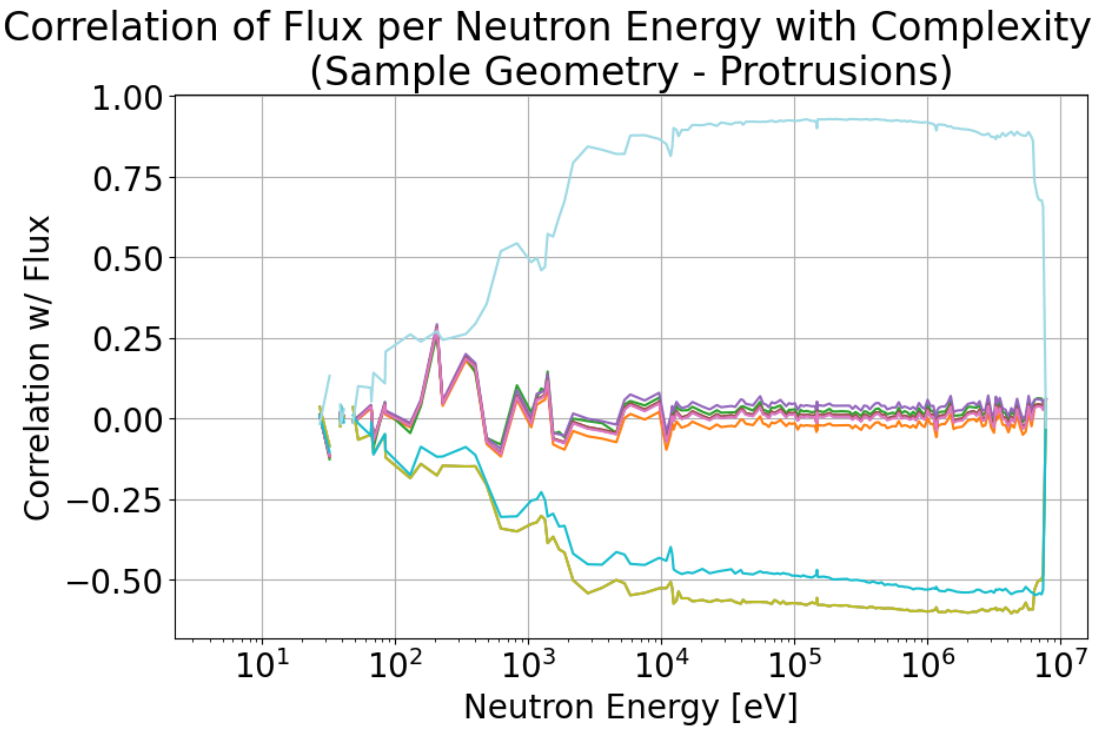



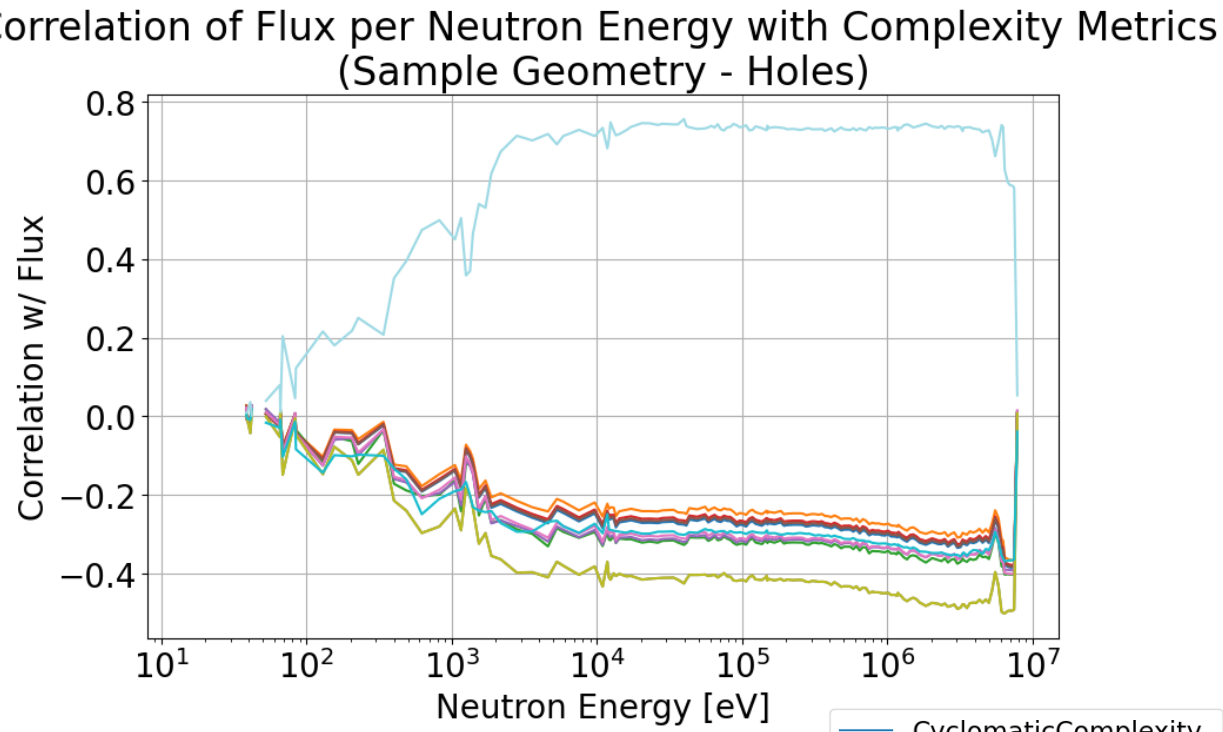

In summary, more features -> more faces -> more runtime.

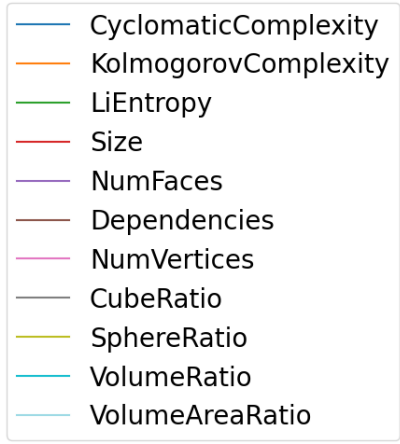
But also remember that <u>number of faces ties in with practitioner-based complexity grading</u>

(what CAD practitioners think a complex geometry is)

# Results: flux on material sample

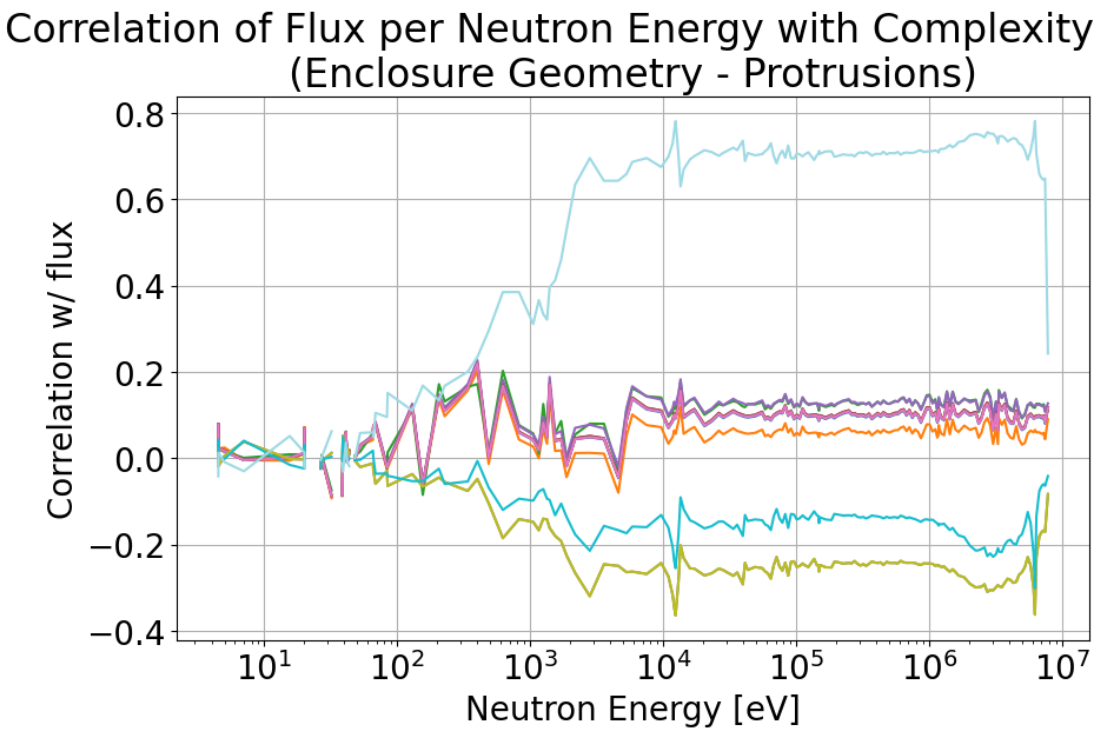


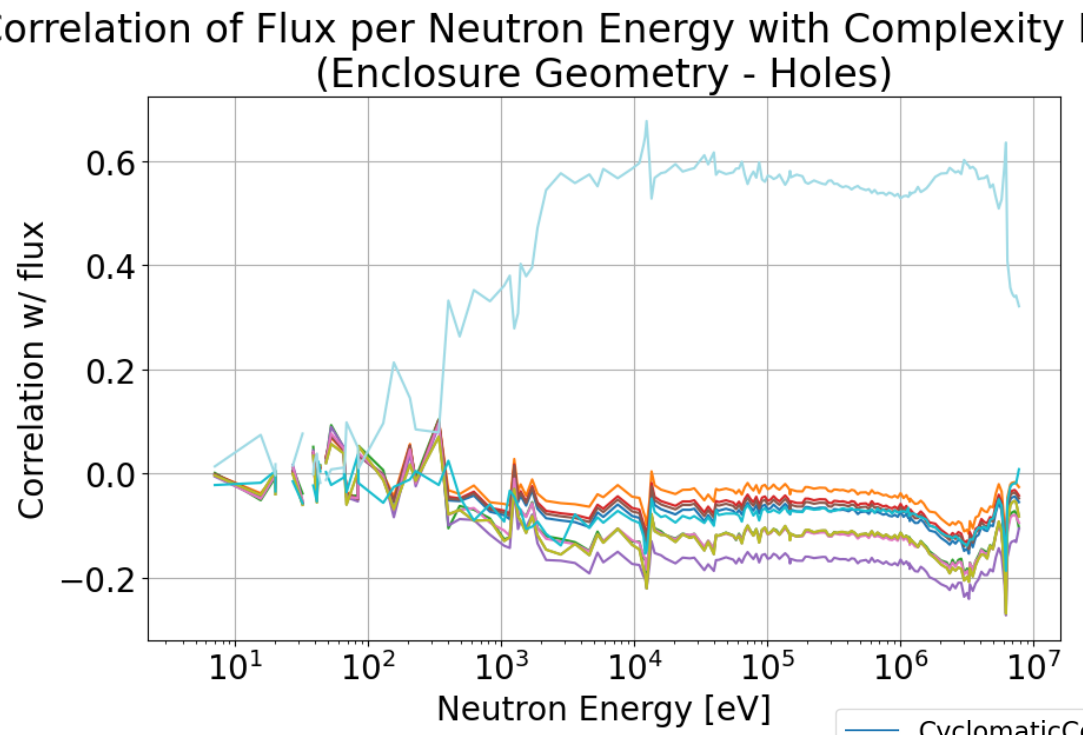



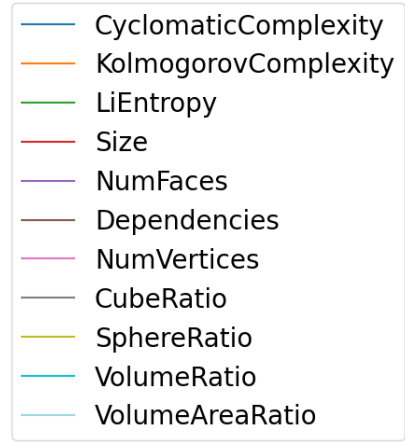




Pearson correlation coefficient: $\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$

# Results: flux on enclosure




- VolumeAreaRatio: Higher volume-to-surface-area ratio correlates with higher flux at higher energies.
- Cube & Sphere Ratio: Negative correlation - deviation from simple shapes may encourage less flux retention.

# Summary

- What CAD practitioners think a ‘complex geometry’ is may not significantly impact actual neutronics results.
  - The strongest correlated metrics to expert-based grading do not affect neutron transport results.
  - However, **the more complex a geometry is, the longer the runtime will be** during the transport-related phase.
- Results may be influenced by overall shape compactness and deviation from simple forms like spheres and cubes

MANCHESTER 1824
The University of Manchester

End of presentation

# Thank you and questions

**Investigating the relationship between CAD model complexity and performance trade-offs for fusion neutronics**

**Raska Soemantoro**, Zeyuan Miao, Lee Margetts

2025 Fusion Neutronics Meeting: ITER and Beyond

Facultad de Educación - UNED, Madrid, Spain

April 2025

# References


1. Contero et al. (2023), Adv. Eng. Informatics, 56, 101970
2. Collivan et al. (2022), Computer-Aided Design, 147, 103226
3. Catalan et al. (2024), Nucl. Eng. and Tech., 56(6), 2404-2411
4. Shimwell et al (2024), github.com/fusion-energy/neutronics_material_maker

# Extra slides

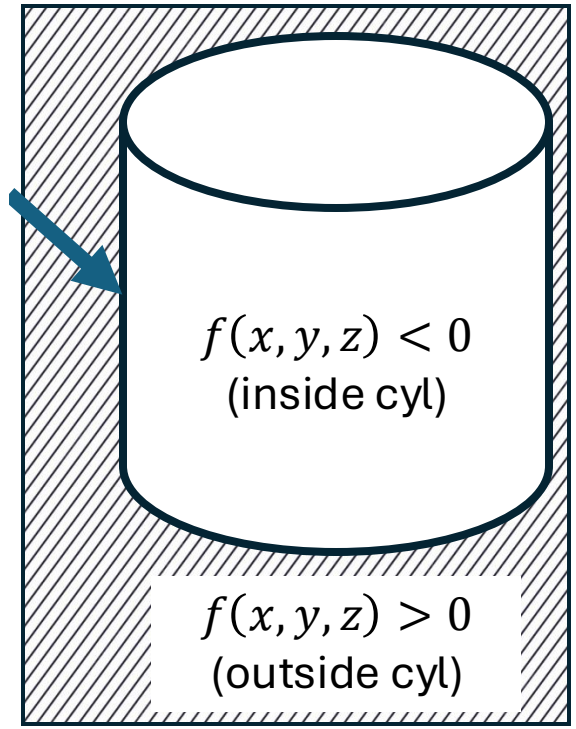
- Extra slides

# CSG vs DAGMC neutronics

- In **CSG**, each 'cell' is made up of regions defined in mathematical functions.
  - Cells contain material property (composition, density, temperature, etc)
- Take a cell made up of an infinite cylinder parallel to the x-axis:

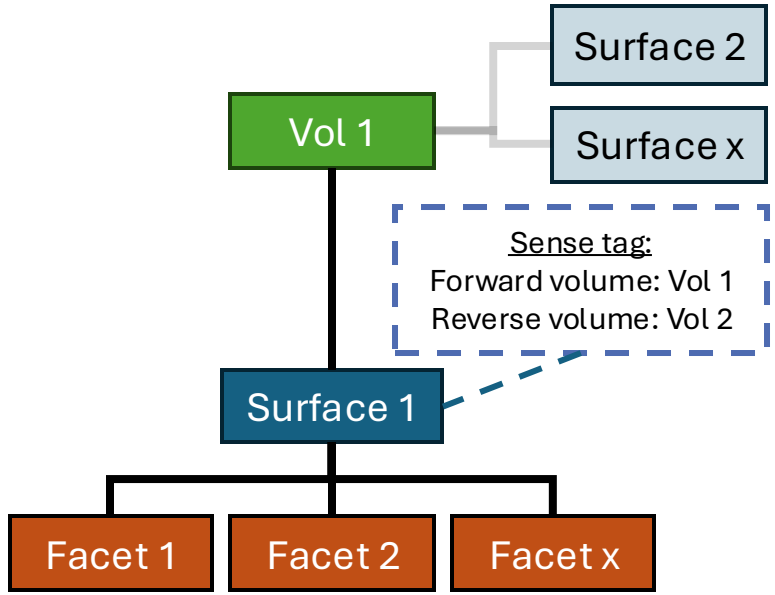
$$f(x,y,z) = (y-y_0)^2 + (z-z_0)^2 - R^2 = 0$$

- We plug in:
  - Cylinder's origin for $y_0$ and $z_0$, and radius for $R$
  - Particle's coordinates $(x, y, z)$
- If the result is less than zero, particle is inside the cylinder (*'negative half-space'*)
  - Otherwise, it is outside (*'positive half-space'*)
- Particles can be lost due to:
  - Cell overlaps
  - Gaps of undefined regions
- Interaction events (collision, absorption, fission) calculated based on material property of the cylinder region
- Conversion from CAD is not straightforward – many CAD geometry has splines!
  - Requires defeaturing



# CSG **vs** **DAGMC** neutronics

- In **DAGMC**, each 'cell' is defined by a volume bound by a surface mesh[1]
  - Cells contain material property (composition, density, temperature, etc)
- Each mesh facet belongs to a surface, which belongs to a volume
  - Surfaces contain 'sense tags' with indicate previous & next volume
- We determine a particle's location by recording which facet it has crossed
  - From there, we can infer the surface and therefore the volume it is in
  - Has added benefit of not requiring negative space definition
- Particles can be lost due to:
  - Overlaps & undefined regions
  - Non-manifold geometry
  - Poor mesh quality
- Interaction events (collision, absorption, fission) calculated based on material property of the volume region
- Conversion from CAD relatively simple – only meshing & file conversion needed




TRIPOLI-4® FLUKA MCNP6 OpenMC

1. DAGMC Team (2022). Developer's Theory Guide to DAGMC. https://svalinn.github.io/DAGMC/contribute/devtheory.html