

# CAD\_to\_OpenMC: from CAD design to particle transport

Erik Bergbäck Knudsen<sup>1</sup>  and Lorenzo Chierici<sup>2</sup>

<sup>1</sup> United Neux, Denmark <sup>2</sup> Copenhagen Atomics A/S, Denmark  Corresponding author

DOI: [10.21105/joss.07710](https://doi.org/10.21105/joss.07710)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

---

Editor: Mojtaba Barzegari  

Reviewers:

- [@nassermra](#)
- [@jacobmerson](#)
- [@bonh](#)

Submitted: 24 September 2024

Published: 16 May 2025

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

We present CAD\_to\_OpenMC - a practical tool for allowing geometries designed in CAD-systems to be used in Monte Carlo particle transport simulation systems. Written in Python, and easily installable through pip and/or conda channels, CAD\_to\_OpenMC does so by importing a geometry in the form of a step-file, a common export format among many CAD systems, identifying objects within that file, and running a triangularization algorithm on all surfaces contained within the geometry. The geometry is written to an HDF5 file in a format defined by the DAGMC-library, which is usable by the main particle-transport packages.

## Statement of need

Most Monte Carlo neutronics-, and general particle transport (e.g. OpenMC, GEANT4, MCNP, Fluka ([Agostinelli et al., 2003](#); [Böhlen et al., 2014](#); [Ferrari et al., 2005](#); [Kulesza et al., 2022](#); [Romano & Forget, 2013](#))) software packages are ubiquitous in nuclear science. They're used to assess performance of the system, estimate radiation levels, shielding requirements, among many other uses. Almost all of them use the concept of Constructive Solid Geometry (CSG) to describe the geometry or scene in which computations are to be performed. Computations generally amount to trace a large number of rays through the geometry. Such models range in complexity from simple scenes of only a few geometrical objects to several million, but are generally considered tricky to work with, in that all object boundaries must be constructed from analytical descriptions of geometrical primaries (cylinders, planes, 2nd order surfaces, etc). Most engineering design, however, is performed using CAD tools, which are optimized for ease of design.

These facts form a compelling argument for bridging the gap between these worlds. So far, a number of solutions have been proposed which can mostly be put into two categories.

1. tools that convert CAD models into CSG ([Catalán et al., 2024](#); [Du et al., 2018](#)), and
2. tools that allow ray-tracing on CAD geometries, or close derivatives thereof.

In the second category, one of the most widespread open source solutions is DAGMC ([Wilson et al., 2010](#)), a library with couplings to all OpenMC, MCNP, fluka, and Geant4. DAGMC is proven to be fast and reliable, yet one hurdle remains: DAGMC requires a discretized (meshed) approximation of the CAD geometry. In general, it is not a trivial requirement.

CAD software packages are, for the most part, proprietary packages with little to no focus on being interchangeable. One upshot of this is a notable lack of overlapping standard formatting between packages, the exception being the STEP-format, which is backed by an international standard ([Industrial Automation Systems and Integration—Product Data Representation and Exchange – Part 1, 2024](#)). Thus, a tool should support this format if it is not to be tied to any particular CAD engine.

The CAD\_to\_OpenMC tool eases the process of generating a meshed description of a CAD-generated geometry (in the form of a step file) ready for inclusion in transport codes through DAGMC.

While other active projects exist that target similar problems (e.g., `cad_to_dagmc` ([Shimwell, 2025](#)), and `stellarmesh` ([Koen, 2025](#))), CAD\_to\_OpenMC is designed with generality in mind. This is defined as: First, it is aimed at working for all step-geometries, with no assumptions on geometry. Second, it must be relatively easy to add code for a new backend should one wish to do so. From this requirement stems the code structure as it is, with one frontend and several backend classes. This may be seen as a measure to enhance the longevity of the project. Last, it must be easy to extract, generate, and manipulate material tags from the underlying step model. As models become large and complex, maintaining a separate list of materials, in sequence, is seen as cumbersome and error-prone.

## Method

CAD\_to\_OpenMC uses OCCT ([OCCT3D Collaborative Development Portal, n.d.](#)) to interact with CAD geometries and its handling of step-files. Once the geometry has been imported, one of several meshing back-ends may be called to create a discretized version of the geometry. In the end, the so-generated geometry is exported into the DAGMC-expected format.

Similarly to most CAD systems, OCCT uses a hierarchical boundary representation model (BREP). In a simplified picture, objects consist of a set of bounded surfaces (faces), formed by a set of boundaries (edges), which themselves consist of line segments connecting points or simply mathematical descriptions of curves. More specifically, the BREP-description used in OCCT also includes notions of shells (the boundary of a volume), loops (a circular set of edges), etc. The BREP concept is subtly but fundamentally different from CSG, where objects are formed by boolean operations on halfspaces. E.g., the OCCT BREP-description allows several more types of operations for forming a geometry, such as extrusions and rotations.

Thus, a need for conversion always exists, regardless of whether one discretizes the geometry or not.

## Triangularization / Surface Meshing

When a curved geometry is to be discretized, the result is an approximation. This leads to potential problems with overlaps between regions if

- a. objects are close to each other, such as for a cylindrical can with thin walls
- b. objects share a surface such as the case for a liquid inside a can.
- c. when objects have surfaces that touch each other.

The first case puts a constraint on the absolute tolerance of the discretization, i.e., triangles have to be small enough not to cause crossing surfaces.

In terms of case b, the discretization process must take care not to re-evaluate any surface that is shared. Instead, it must reuse the triangles of the former instance. This means that objects cannot be independently processed. The whole process is generally handled by the underlying geometry engine ([OCCT3D Collaborative Development Portal, n.d.](#)) through its hierarchical model. Objects simply store links to surfaces and their triangularizations (similar for curves — line segments). CAD\_to\_OpenMC handles this by generating a hash code for each surface upon processing. Each time a surface is encountered, it is examined, and if the surface has already been encountered, it will be reused, not recomputed.

In the latter case (c), imprinting has to be performed. This is the process where the boundary curves of the smaller surface is projected onto the larger, splitting the larger into two or more sub surfaces.

### Meshing backends

CAD\_to\_OpenMC is constructed such that it is flexible in terms of the algorithms used to generate triangularized surfaces that approximate the geometries, known as back-ends. The present release supports a set of backends: `{'gmsh','stl','stl2','db'}`. In most cases, we have encountered that 'stl2' performs well. It uses a very basic algorithm, with no restriction on triangle aspect ratio, but also produces discretized geometry files of moderate size. In practice, the aspect ratio does not appear to be a problem for particle transport. In case the same discretization is to be used for other types of applications, care should be taken.

In some cases, the simple algorithm fails to create a watertight model. In such cases, we recommend using the 'db' backend if available. The 'gmsh' backend consistently produces a practical model, but often has the problem that it is memory hungry. E.g., neither the ARE nor the MSRE (see below) models could be run this way on our available hardware (64GB workstation).

### Material tags

If needed, CAD\_to\_OpenMC can use the CAD-generated part names as material tags. The default behavior is to use the first part of the part name as a material tag. This may be changed by supplying a hash table, i.e. Python dictionary, as the tags-argument. Here, the keys to the table are interpreted as regular expressions, and the values are taken to be material tags to use. Parts, whose names do not match any of the regexp-keys, are tagged as vacuum by default. If, however, a flag is set, then the material tag is extracted from the part name, as if no list of tags is supplied. This allows to re-tag only a subset of the parts. The below example shows how to tag all parts with the name "wall" in them with "concrete" and all parts ending with bellows with "steel".

```
tag_dict = {".*bellows" : "steel", ".*wall.*" : "concrete"}
```

### Implicit complement

CAD\_to\_OpenMC can add an implicit complement to the output file. That is, the material tag that will be applied to any part of the geometry which is *not* claimed by any CAD part. This is done by simply assigning the name of the material as a string to the implicit complement attribute of the base Assembly object. If the attribute is set C2O assigns the extra material tag (with a suffix of '\_comp') to the last part handled. That material tag gets picked up by the DAGMC system and is used for any unclaimed volume.

## Results

We have chosen 3 reactor models as test systems. A tabletop reactor and two full-scale molten salt reactors. The former (GODIVA IV) model is included in the ICSBEP-benchmark project ([International Handbook of Evaluated Criticality Safety Benchmark Experiments, 2022/2023](#)) as case HEU-MET-FAST-086; the latter two were part of the molten salt reactor program at ORNL.

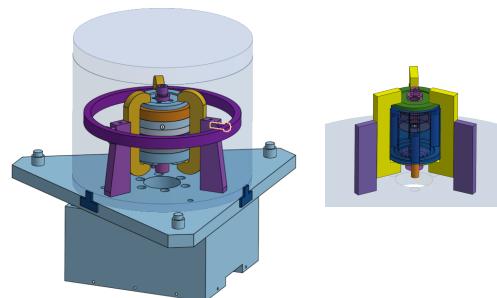
### GODIVA IV

This model was chosen since it exhibits moderate complexity and has a generous set of experimental data to benchmark against. It is detailed enough to be cumbersome to model quickly using CSG but not with CAD. Further, CSG models (for MCNP) may be found in the published benchmark, along with detailed experimental validation data allowing for practical comparison. The reactor consists of a cylindrical core geometry ([Figure 1](#)) held by 3 sets of clamps set at 120 deg. offset. Additionally, the core has three vertical holes (similarly 120 deg.

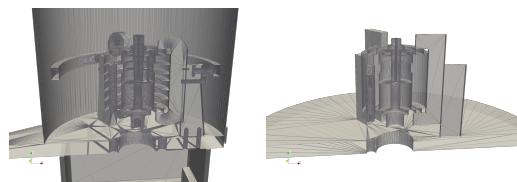
apart), into which control and burst rods may be inserted from below by vertical actuators. The rods themselves are similar in composition to the fuel elements.

The benchmark includes 5 experimental cases, which differ in terms of control- and burst-rod positions. Also, there are 3 geometries described: 1. a detailed model which includes as-built things (curved clamp etc.), 2. simplified model, where all surfaces are along the principal axis, all corners 90 deg. etc., and 3. cylindrical symmetric model, to allow 2D computations. The benchmark reports only experimental results for the 2 first ones, but contains MCNP geometries for all 3 in (*International Handbook of Evaluated Criticality Safety Benchmark Experiments, 2022/2023*). The 3rd model is created in order to use some legacy analysis tools which are purely 1D/2D.

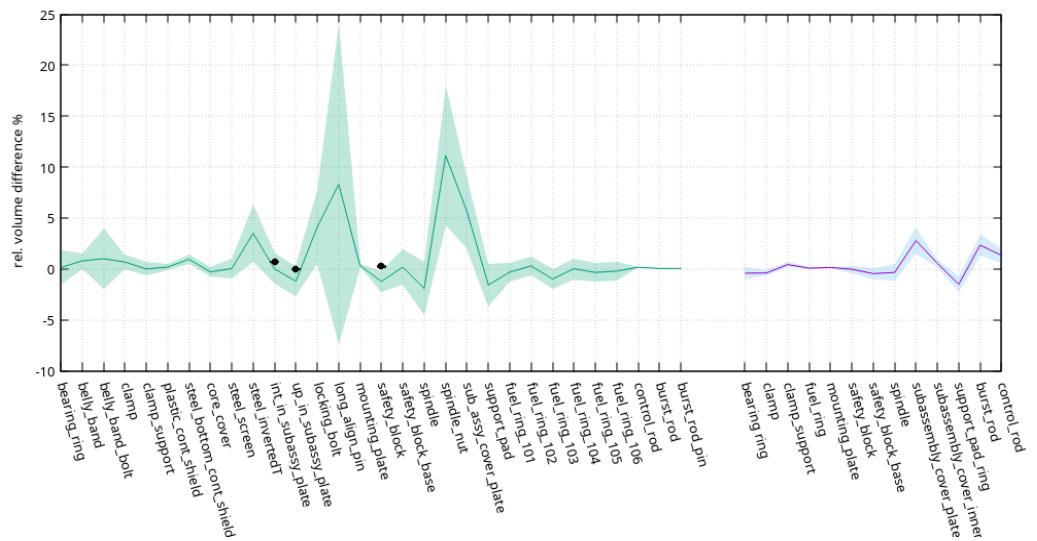
Corresponding to [Figure 1](#), [Figure 2](#) shows the discretized version of the two reactor models used for our further analysis.



**Figure 1:** CAD-drawings of the Godiva IV reactor, detailed (left) and simplified (right) versions. Note the rectangular clamps and supports in the simplified version. Visible are also the set of control and burst rods.



**Figure 2:** Discretized Godiva IV-models, detailed (left) and simplified (right) versions.



**Figure 3:** Part-by-part comparison between volume calculations using stochastic volume estimation in OpenMC compared with direct volume calculations reported by CAD-software for the detailed model (green) and simplified benchmark model (magenta). The indicated intervals are the computational error margins, almost exclusively stemming from the estimated error in the stochastic volume computation. The additional black circles denote the relative difference between the extracted benchmark CSG-model (as computed by stochastic volume runs in OpenMC) and the CAD-model.

Figure 3 shows differences in volumes between the discretized models and the exact CAD model for the various objects making up the parts. Volumes have been calculated using a built-in feature of our CAD package, whereas volumes from the discretized models have been computed using the stochastic volume computation feature of OpenMC (Romano & Forget, 2013). In the latter case, volumes are being computed by sampling a (large) number of points within a set boundary, while recording which volume each point falls within. The precision of the algorithm is directly governed by the number of points sampled, meaning a direct dependence on run time. In practice, the calculation can be run to an arbitrary set target tolerance.

Generally, differences in volume have a much bigger influence on the neutronics of a reactor than do small boundary changes (with constant volume). Hence, this is a useful measure for performance. Note that the errors found (fig. 3) are dominated by the error in the stochastic volume estimator, not the volume error itself. This is evidenced by the very small error in burst- and control-rod volume for the detailed model, which was run with smaller tolerances. Additionally, we have used the benchmark CSG-model to verify a few volumes in the detailed model. A CSG model generally yields shorter runtimes, which allows tighter tolerance while also remaining practical. We find that the CSG-benchmark deviates slightly from the CAD-model constructed from drawings, suggesting there is an underlying discrepancy internally in the benchmark, which may have to be addressed.

The considered cases have different settings for the control- and burst-rods each (see tables 1 and 2).

**Table 1:** Control rod (CR) and burst rod (BR) positions for the 5 cases of the Godiva-IV benchmark/simplified model from HEU-MET-FAST-086 (Goda et al., 2021; *International Handbook of Evaluated Criticality Safety Benchmark Experiments*, 2022/2023). Measures in inches withdrawn from the fully inserted position. The two rightmost columns contain criticality numbers for the device. MC refers to simulated Monte Carlo estimates, whereas Lit. refers to numbers drawn from the benchmark report. CAD0 is a single-mesh model, whereas CAD refers to a model where core, control rods, and burst rods have been discretized separately, then put together as separate universes in OpenMC.

case	CR 1 top	CR 2 top	BR top	$k_{eff}$ CAD0	$k_{eff}$ CAD	$k_{eff}$ CSG
1	-4.001	-0.449	0.0	0.98326	0.98026	0.98187
2	-1.998	-1.666	0.0	0.98427	0.98101	0.98185
3	-0.493	-3.794	0.0	0.98460	0.98124	0.98297
4	-0.469	-0.447	-2.970	0.98446	0.98745	0.98359
5	-0.319	-0.656	0.0	0.98980	0.98706	0.98844

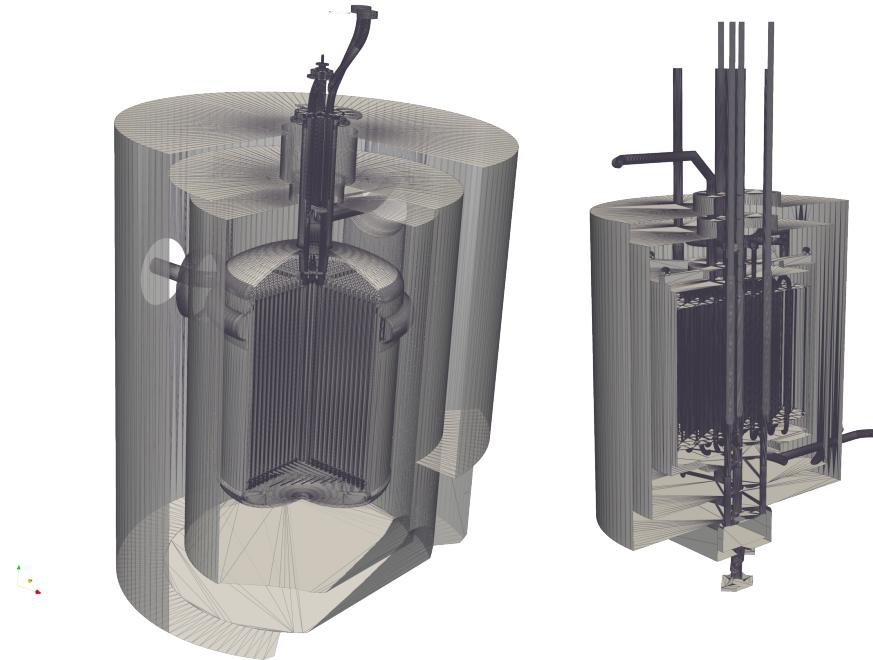
**Table 2:** Control rod (CR) and burst rod (BR) positions for the 5 cases of the detailed Godiva-IV model from HEU-MET-FAST-08 (Goda et al., 2021; *International Handbook of Evaluated Criticality Safety Benchmark Experiments*, 2022/2023). Measures in inches withdrawn from the fully inserted position. The two rightmost columns contain criticality numbers for the device. MC refers to simulated Monte Carlo estimates, whereas Lit. refers to numbers drawn from the benchmark report. In the detailed case, for simplicity, only a separately discretized model was run.

case	CR 1 top	CR 2 top	BR top	$k_{eff}$ CAD	$k_{eff}$ CSG	$k_{eff}$ Lit.
1	-4.001	-0.449	0.0	0.97905	0.98303	0.9880
2	-1.998	-1.666	0.0	0.98390	0.98275	0.9880
3	-0.493	-3.794	0.0	0.97885	0.98330	0.9887
4	-0.469	-0.447	-2.970	0.98352	0.98426	0.9897
5	-0.319	-0.656	0.0	0.98390	0.98969	0.9945

Each experimental case was modelled in CAD in two ways. First, the entire reactor, including burst- and control-rods in respective positions, was drawn in CAD, exported and converted to a transport-compatible geometry. Second, the core geometry, burst, and control-rod were exported and converted individually. In the latter case, the full reactor model is assembled when the transport geometry is described within the confines of an OpenMC geometry. This yields the important advantage of a model in which the rods can be moved dynamically, i.e., not restricted to predefined cases locked in by the discretized geometry. In the case of the detailed model, minor adjustments had to be made to the stated measurements in order for the model to fit. The locking bolts had to be shortened slightly, and the height of the shelf of the inner intermediate sub-assembly plate was reduced. In both cases, the edits were  $\sim= 1$  mm and will not affect the results. We assume the errors are mere misprints in the drawings.

It is clear from tables 1 and 2 that the agreement between model and experiment is not perfect, yet the difference between CAD-based and CSG-based models is small (generally on the order of 5e-3). We take this as proof that the geometry generation works as intended.

## Molten Salt Reactors



**Figure 4:** Rendering of triangularized models of the ARE- and MSRE-reactors as generated using CAD\_to\_OpenMC. The MSRE model (left) includes both reactor core, liquid fuel contained therein, graphite moderator stringers, as well as thermal shielding and reactor enclosure. The reactor pit is included in the model, but we have excluded it from the image to make the core more visible. The ARE model (right) includes the set of three safety/shim rods and the regulating rod in the center.

[Figure 4](#) (left) shows pictures of the meshed MSRE and ARE geometries, including reactor enclosure control rods, etc. The Aircraft Reactor Experiment (ARE) and Molten Salt Reactor Experiment (MSRE) were two molten salt reactor experiments carried out at Oak Ridge National Laboratory; ARE in November '59 and MSRE was run between '65 and 70. In the confines of this article, this pair serves as examples of complex reactor geometries that can be handled by CAD\_to\_OpenMC. Very detailed CAD models and example scripts used to compute these numbers are available for the set, which we have used as inputs ([Bergbäck Knudsen & Atomics, 2024, 2025](#)). [Table 3](#) tabulates the  $k_{eff}$ -values computed using a materials composition set to mimic the reported values as closely as possible. It is clear that, similar to the case for the GODIVA IV device, the modelled values are not in complete agreement with the reported ones, yet this may be explained by possible discrepancies between the drawings in accessible reports and the actual experiment. Any engineering realities not written down in these reports are likely lost. This is particularly true for ARE, where the discrepancy is largest, and details are scarce. It should be noticed that the calculated  $k_{eff}$  for the MSRE case has better agreement than any other MSRE criticality benchmarks known to the authors.

**Table 3:** Criticality numbers,  $k_{eff}$ , for two molten salt reactors as computed by the combination of CAD-model, CAD\_to\_OpenMC, and OpenMC, using ENDF v8.0 cross-sections. Also tabled are corresponding (albeit trivial) experimental values found in literature,  $k_{eff,lit}$  ([Cottrell et al., 1955; Robertson, 1965](#)).

model	$k_{eff}$	$k_{eff,lit}$
ARE	1.06582	1.0
MSRE	1.00549	1.0

## Discussion and Conclusion

We submit that we have shown that the tool presented in this paper is a convenient tool for making CAD geometries available for Monte Carlo particle transport. By utilizing the DAGMC-layer, the resulting geometries are not restricted to OpenMC, but in fact may be used also in MCNP, fluka, etc. Experience has shown that a particularly useful feature is to extract tags from CAD-defined parts and interpret them as material tags for transport. This enables a consistent material naming scheme throughout the entire modeling procedure.

Finally, as noted, other active projects exist targeting the problem. In the interest of efficiency and resource management, there are active efforts aiming unification, which may bear fruit in coming releases.

## References

- Agostinelli, S., Allison, J., Amako, K., Apostolakis, J., Araujo, H., Arce, P., Asai, M., Axen, D., Banerjee, S., Barrand, G., Behner, F., Bellagamba, L., Boudreau, J., Broglia, L., Brunengo, A., Burkhardt, H., Chauvie, S., Chuma, J., Chytracek, R., ... Zschiesche, D. (2003). Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3), 250–303. [https://doi.org/10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8)
- Bergbäck Knudsen, E., & Atomics, C. (2024). *MSRE* [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.15000209>
- Bergbäck Knudsen, E., & Atomics, C. (2025). *ARE* (Version 1.0.0) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.15000991>
- Böhlen, T. T., Cerutti, F., Chin, M. P. W., Fassò, A., Ferrari, A., Ortega, P. G., Mairani, A., Sala, P. R., Smirnov, G., & Vlachoudis, V. (2014). The FLUKA code: Developments and challenges for high energy and medical applications. *Nuclear Data Sheets*, 120, 211–214. <https://doi.org/10.1016/j.nds.2014.07.049>
- Catalán, J. P., Sauvan, P., García, J., Alguacil, J., Ogando, F., & Sanz, J. (2024). GEOUNED: A new conversion tool from CAD to monte carlo geometry. *Nuclear Engineering and Technology*, 56(6), 2404–2411. <https://doi.org/10.1016/j.net.2024.01.052>
- Cottrell, W. B., Hungerford, H. E., Leslie, J. K., & Meem, J. L. (1955). *OPERATION OF THE AIRCRAFT REACTOR EXPERIMENT* (ORNL-1845(Del.), 4237975; pp. ORNL-1845(Del.), 4237975). <https://doi.org/10.2172/4237975>
- Du, T., Inala, J. P., Pu, Y., Spielberg, A., Schulz, A., Rus, D., Solar-Lezama, A., & Matusik, W. (2018). InverseCSG: Automatic conversion of 3D models to CSG trees. *ACM Transactions on Graphics*, 37(6), 1–16. <https://doi.org/10.1145/3272127.3275006>
- Ferrari, A., Sala, P. R., Fasso, A., & Ranft, J. (2005). *FLUKA: A multi-particle transport code (Program version 2005)*. <https://doi.org/10.2172/877507>
- Goda, J., Bravo, C., Cutler, T., Grove, T., Hayes, D., Hutchinson, J., McKenzie, G., McSpaden, A., Myers, W., Sanchez, R., & and, J. W. (2021). A new era of nuclear criticality experiments: The first 10 years of godiva IV operations at NCERC. *Nuclear Science and Engineering*, 195(sup1), S55–S79. <https://doi.org/10.1080/00295639.2021.1947103>
- Industrial automation systems and integration—product data representation and exchange – part 1: Overview and fundamental principles (standard no. ISO 10303-1:2024)*. (2024). <https://www.iso.org/standard/83105.html>
- International handbook of evaluated criticality safety benchmark experiments* (No. 7645). (2022/2023). OECD Nuclear Energy Agency.

- Koen, et. al., A. (2025). <https://github.com/stellarmesh/stellarmesh>
- Kulesza, J. A., Adams, T. R., Armstrong, J. C., Bolding, S. R., Brown, F. B., Bull, J. S., Burke, T. P., Clark, A. R., Forster, R. A., III, Giron, J. F., Grieve, T. S., Josey, C. J., Martz, R. L., McKinney, G. W., Pearson, E. J., Rising, M. E., Solomon, C. J., Jr., Swaminarayan, S., Trahan, T. J., ... Kulesza, J. A. (2022). *MCNP code version 6.3.0 theory & user manual* (LA-UR-22-30006, Rev. 1). Los Alamos National Laboratory. <https://doi.org/10.2172/1889957>
- OCCT3D collaborative development portal.* (n.d.). Retrieved September 18, 2024, from <https://dev.opencascade.org/>
- Robertson, R. C. (1965). *MSRE design and operations report, part i, description of reactor design* (ORNL-TM-0728). Oak Ridge National Laboratory.
- Romano, P. K., & Forget, B. (2013). The OpenMC monte carlo particle transport code. *Annals of Nuclear Energy*, 51, 274–281. <https://doi.org/10.1016/j.anucene.2012.06.040>
- Shimwell, J. (2025). [https://github.com/fusion-energy/cad\\_to\\_dagmc](https://github.com/fusion-energy/cad_to_dagmc)
- Wilson, P. P. H., Tautges, T. J., Kraftcheck, J. A., Smith, B. M., & Henderson, D. L. (2010). Acceleration techniques for the direct use of CAD-based geometry in fusion neutronics analysis. *Fusion Engineering and Design*, 85(10), 1759–1765. <https://doi.org/10.1016/j.fusengdes.2010.05.030>