

LA-UR-26-27262

Approved for public release; distribution is unlimited.

Title: Native STL Faceted-Surface Tracking in MCNP6® and an Automated STEP-to-MCNP Converter

Author(s): Sweezy, Jeremy Ed

Intended for: Meeting: 2026 MCNP User Symposium, 2026-08-24 - 2026-08-26 (Los Alamos, New Mexico, UNITED STATES)

Issued: 2026-08-12



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

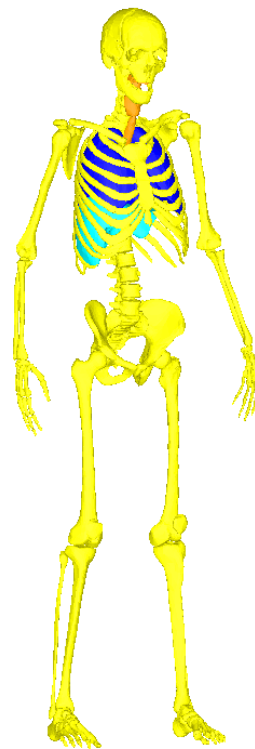
Native STL Faceted-Surface Tracking in MCNP6® and an Automated STEP-to-MCNP Converter

Jeremy Sweezy
Los Alamos National Laboratory

MCNP[®] and Monte Carlo N-Particle[®] are registered trademarks owned by Triad National Security, LLC, manager and operator of Los Alamos National Laboratory. Any third party use of such registered marks should be properly attributed to Triad National Security, LLC, including the use of the [®] designation as appropriate. Any questions regarding licensing, proper use, and/or proper attribution of Triad National Security, LLC marks should be directed to trademarks@lanl.gov.

STL Faceted-Surface Integration in MCNP6

1. **New Capability:** Native support for STL (Stereolithography) geometry as a standard MCNP surface type.
2. **Goal:** Simplify complex geometry modeling by allowing direct import of CAD surface meshes without manual CSG (Constructive Solid Geometry) conversion.
3. **Integration:** Works alongside existing quadratic/analytical surfaces (planes, spheres, cylinders, tori). Works with CSG Boolean operators.



MCNP 3-D rendering showing STL based phantom, by Auer Benjamin, Brigham and Women's Hospital and Harvard Medical School

What is an STL Faceted-Surface?

- **What is an STL file?**
 - Standard file format originating from 3D printing/CAD export (Stereolithography).
 - Describes the **outer boundary surface** of a 3D object using triangular facets.
- **What is a Faceted Surface?**
 - A surface composed of small, flat planar triangles (facets).
 - Each triangle is defined by 3 vertices and an outward-pointing unit normal vector.



Low resolution version of the Stanford Bunny rendered in MCNP6

Defining STL Surfaces in MCNP Syntax

Surface Card form:

```
<id> [tr] stl file[=]<path_to_stl_file>  
      [scale[=]<value>]  
      [check[=]<0|1|2>]
```

Example:

```
1 1 1.0 -1 2 imp:n=1 VOL=1.0 $ low poly bunny  
200 0 (1:-2) -200 imp:n=1 $ void  
300 0 200 imp:n=0 $ Outside world
```

```
1 stl file=bunny.stl
```

```
2 c/y 3 4 2
```

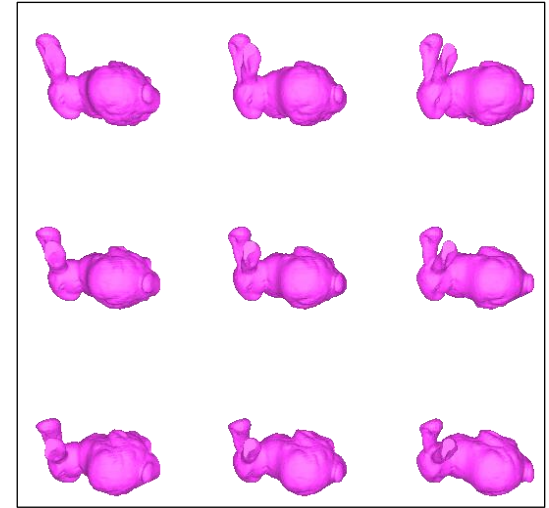
```
200 so 200 $ Outer boundary
```



Stanford Bunny with hole, rendered
in MCNP6

How to use in MCNP

- **Works like an existing MCNP surface**
- **Forms a closed (manifold surface) like:**
 - Sphere
 - Torus
 - Macrobodyes (RCC, RPP, BOX, RHP, ...)
- **Rotations and translations work normally**
- **Lattices work normally**
- **Take care of possible overlap between parts, most CAD conversion to STL will create small gaps and overlaps. Deconflict with standard Boolean CSG operators.**



Lattice of Stanford Bunnies rendered in MCNP6

Overlaps

- **You define the overlap: one takes precedence**

```
1 1 1.0 -1 2 imp:n=1 VOL=1.0 $ wrench
```

```
2 2 1.0 -2 imp:n=1 VOL=1.0 $ bolt
```

```
100 0 -200 2 1 imp:n=1 $ world
```

```
300 0 200 imp:n=0 $ Outside world
```

```
1 stl file=wrench.stl
```

```
2 stl file=bolt.stl
```

```
200 so 200 $ Outer boundary
```



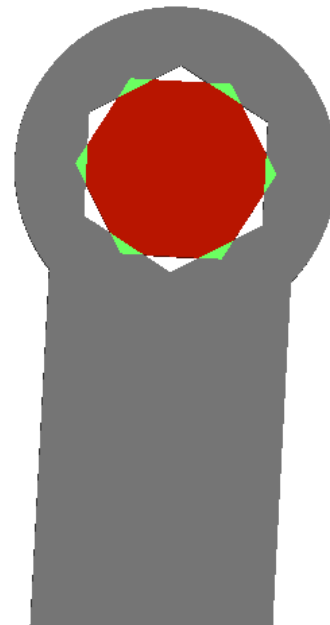
Wrench and bolt defined with STLs. Bolt turned 90 degrees. Bolt takes precedence.

Overlaps

- You define the overlap: overlap region**

```
1 1 1.0 -1 2 imp:n=1 VOL=1.0 $ wrench
2 2 1.0 -2 1 imp:n=1 VOL=1.0 $ bolt
3 3 1.0 -2 -1 imp:n=1 VOL=1.0 $ overlap
100 0 -200 2 1 imp:n=1 $ world
300 0 200 imp:n=0 $ Outside world

1 stl file=wrench.stl
2 stl file=bolt.stl
200 so 200 $ Outer boundary
```



Wrench and bolt defined with STLs.
Bolt turned 90 degrees. Green –
wrench bolt overlap.

Manifold STL Requirements

- **STL must be manifold:**

- Water-tight
- No holes or gaps
- Normals are oriented outwards
- All edges must connect to two faces.

- **MCNP quality checks:**

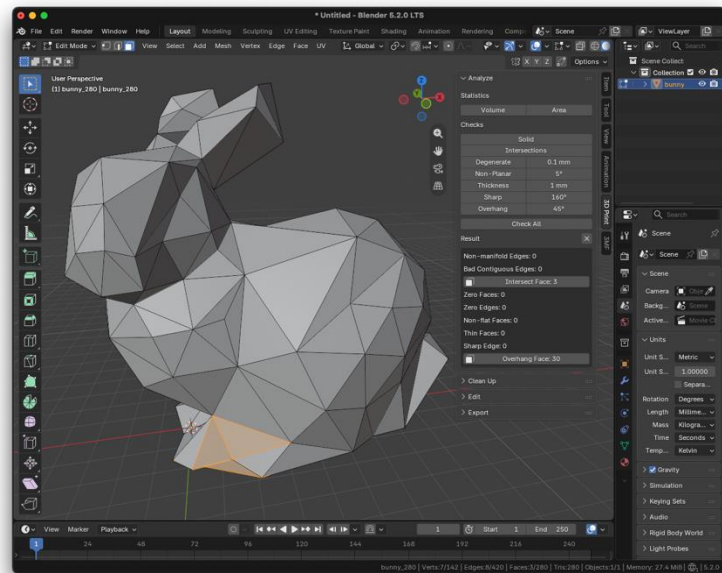
- Flipped Normals
- Self-overlaps
- Non-manifold edges
- Self-intersecting triangles

```
>>> STL GEOMETRY: bunny.stl <<<
Triangles          : 280
Building BVH       : [=====] 100% (70/70) -- 0.00 seconds
Checking overlaps  : [=====] 100% (280/280) -- 0.00 seconds
Checking intersects: [=====] 100% (70/70) -- 0.00 seconds
Checking manifold  : [=====] 100% (280/280) -- 0.00 seconds

Validation Summary:
Triangles checked: 280
Flipped normals: 0
Sliver exclusions: 0 (flipped-normal flags excused; aspect ratio > 200.00)
Self-overlaps: 0
Non-manifold edges: 0
Boundary edges: 0
Non-contiguous edges: 0
Self-intersecting triangles: 0 (0 pairs)
Error rate: 0.0%
Validation time: 0.00 seconds
Valid: YES
<<< STL GEOMETRY RESULT: bunny.stl (OK) <<<
```

STL Advantages

- Common format
- Output options from most CAD/CAM software
- Opensource tools available to modify and heal non-manifold STLs.
 - Blender (can script with python)
- Python packages available for creating or manipulating STLs
 - trimesh
 - Manifold3d
 - numpy-stl
 - PyMeshLab
 - meshio (mesh format converter)
 - CadQuery (parametric construction and import)
 - Open3D (convert point cloud to STL)



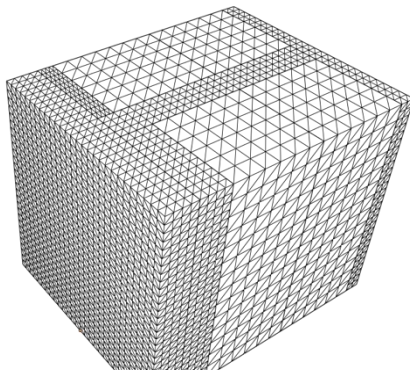
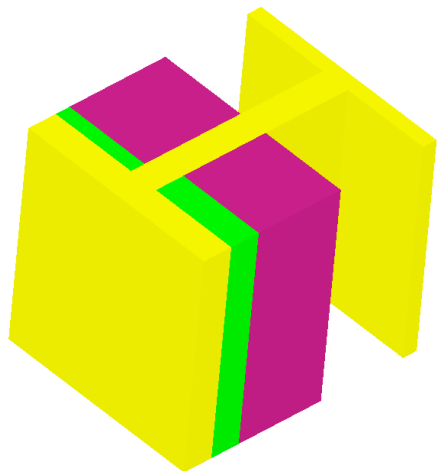
STL Surfaces vs. MCNP Unstructured Mesh (UM)

Feature	STL Surface Tracking	Unstructured Mesh (UM)
Primary Domain	Surface (Boundary) Representation	Volume Mesh
Elements	2D Triangular Facets (Boundary shell)	3D Polyhedra
Role	Functions as a Surface Card to define cell boundaries	Defines Volumetric Cells & material regions directly
CSG Integration	Full CSG: Use STL surface IDs in standard cell cards	Replaces standard CSG cells for transport/tallies
Use Case	Importing complex CAD surface shapes into standard MCNP models when detailed volumetric tallies are NOT needed.	Finite element analysis , detailed volumetric tallies

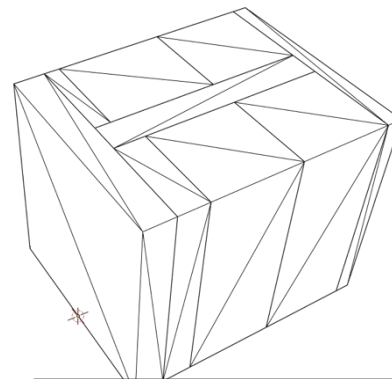
Performance Comparison to UM

- **Bulkhead shielding UM problem**
 - Test from our MCNP UM class
 - 9 STL surfaces, 9 UM regions

Simulation	Performance relative to UM
Naïve Translation (18,960 faces)	0.5
Cleaned STLs (108 faces)	3.2



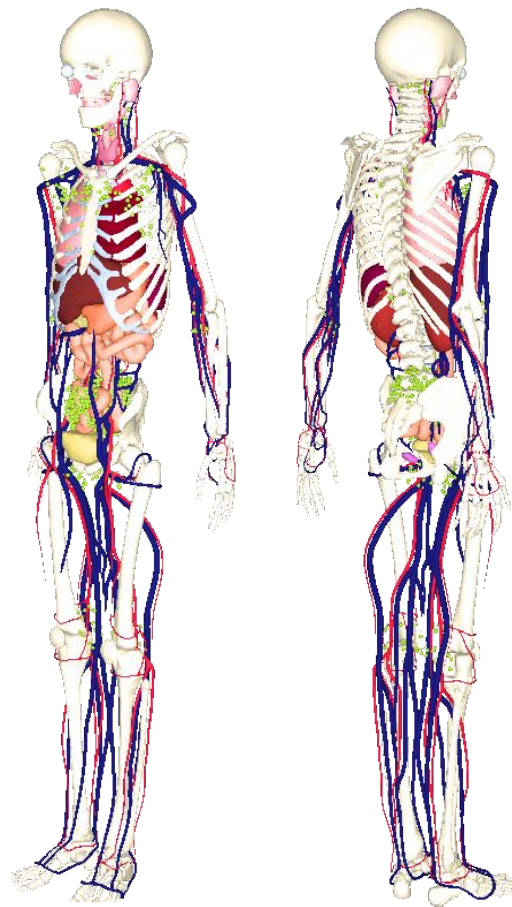
Naïve Translation(18,960 faces)



Cleaned (108 faces)

Performance Comparison to UM

- **ICRP 145 Human Phantom**
 - 173 STL surfaces
- **Initialization**
 - UM initializes in 20.3 minutes
 - STL initializes in 14 secs (3.6 secs without checks)
 - 84.7x faster with STL checks
 - 339x faster without STL checks
- **Transport**
 - 58x faster
- **Memory**
 - 22x less memory

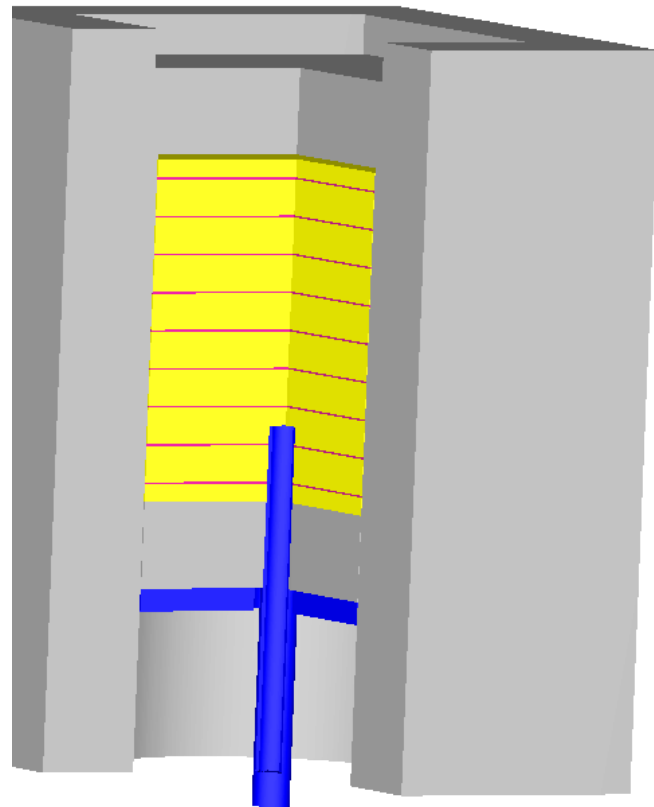


STL model of ICRP 145 Female phantom rendered with MCNP6.

Performance Comparison to CSG

- **ZEUS2 Criticality Experiment**
 - Cylinders and Planes
 - 38 STL surfaces
 - 39 CSG Cells and 155 CSG Surfaces

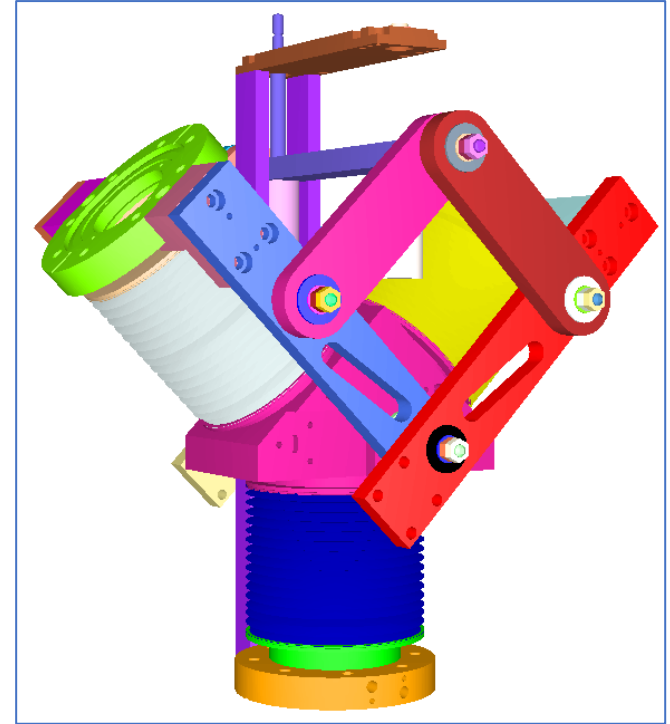
Angle Resolution (degrees)	Performance relative to CSG
1°	0.50
2°	0.55
4°	0.75
8°	0.65



STL model of ZEUS2 criticality benchmark rendered with MCNP6.

BobCAD to STEP-to-MCNP input

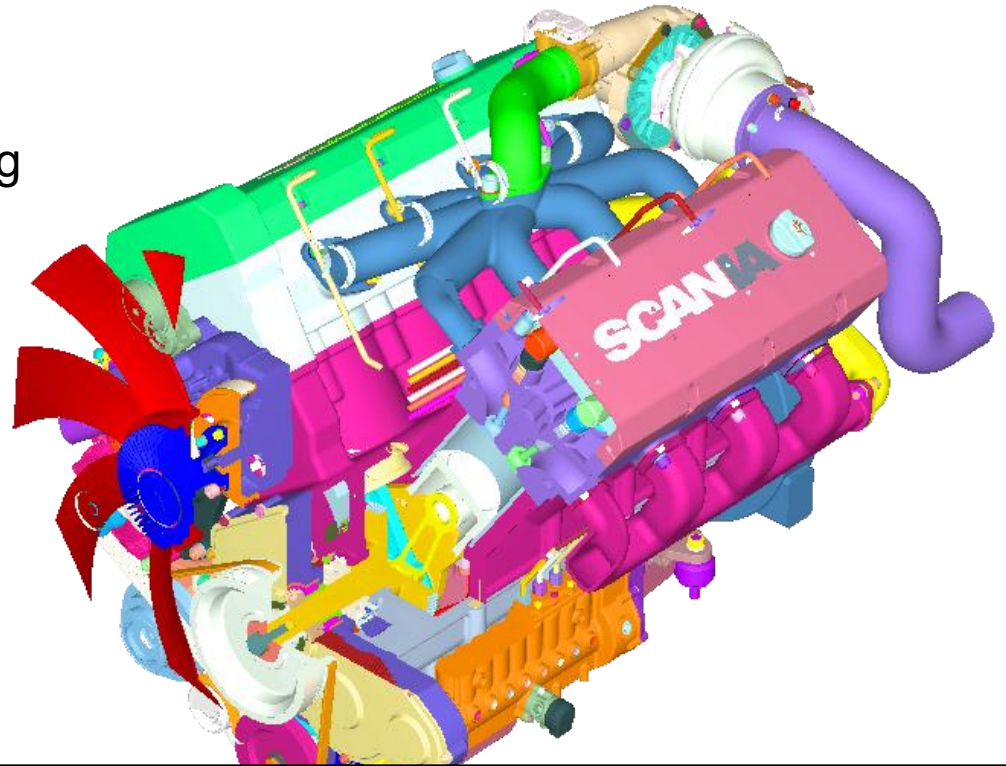
- BobCAD is a LANL product to query and manipulate CAD geometry models
- It has been extended to convert STEP files into MCNP input along with associated STL files.



Weldment STEP Test with 34 unique parts and 108 components converted with BobCAD and rendered in MCNP6

Future work:

- Testing, testing, and more testing
- Optimization
- Memory reduction



Engine V8 Turbo STEP Test from Autodesk Inventor 2018 with 158 unique parts and 726 components converted with BobCAD and rendered in MCNP6.

Model by: Ivo Jardim
Source: steptools.com

Questions

Jeremy Sweezy

jsweezy@lanl.gov



A bobblehead of the author rendered using MCNP6.