LA-UR-21-25545

Approved for public release; distribution is unlimited.

Title:	MCNP® Visualization Approaches: An Overview
Author(s):	Kulesza, Joel A.
Intended for:	MCNP Steering Committee Meeting Presentation
Issued:	2021-06-14

MCNP® Visualization Approaches: An Overview

Joel A. Kulesza

Monte Carlo Codes Group (XCP-3)
Los Alamos National Laboratory

MCNP Steering Committee Meeting
June 10, 2021

Outline

Introduction & Motivation

Objective: broad overview of non-traditional MCNP visualization approaches

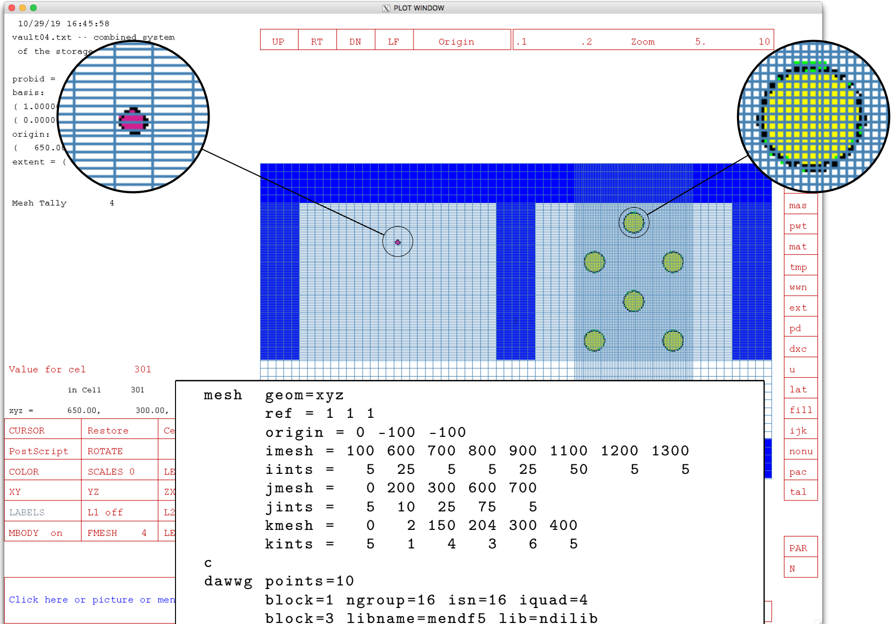
- The MCNP [1] plotter originated in the late 1970s
  - – Dated, and potentially unintuitive, interface (table vs. button)
  - – Limited interactivity—no click-and-drag capabilities
  - – Relies on X11 (unconventional setup for Windows users)
  - + 2-D slice plotter is precise (faithful zoomed geometry)
  - + Headless (`notek`) and batch-processing capability: speed & efficiency
  - + Cross-platform execution; multiple output capabilities (`file`, `printpts`)

↔

```
mcplot> help term
> Syntax: term n m
  Device type.
     ...
   n=1 specifies Tektronix 4010 using CGS.
   n=2 specifies Tektronix 4014 using CGS.
   n=3 specifies Tektronix 4014E using CGS. (DEFAULT)
   n=4115 specifies Tektronix using GKS and UNICOS.(DEFAULT)
   n=1 specifies Tektronix using the AIX PHIGS GKS library
     (DEFAULT) Check with your vendor for the proper
     terminal type if you are using a GKS library.
  The baud rate of the terminal is specified by m.
    (DEFAULT=9600)
```

https://en.wikipedia.org/wiki/Tektronix_4010.

LNK3DNT from MCNP CSG (`mcnp6 m i= vault04.txt`)




```
mesh   geom=xyz
       ref = 1 1 1
       origin = 0 -100 -100
       imesh = 100 600 700 800 900 1100 1200 1300
       iints =   5  25   5   5  25   50    5    5
       jmesh =   0 200 300 600 700
       jints =   5  10  25  75   5
       kmesh =   0   2 150 204 300 400
       kints =   5   1   4   3   6   5
c
dawwg  points=10
       block=1 ngroup=16 isn=16 iquad=4
       block=3 libname=mendf5 lib=ndilib
       block=5 trcor=diag srcacc=dsa diffsol=mg isct=2
       block=6 massed=1 edoutf=3
```

Geometry: `vault04.txt` from MCNP Criticality Class.

Converting LNK3DNT to VTK/XDMF+HDF5 [5–8]

```
> l3d2vtk -V linkout
Processing file: linkout
  Found 3-D Cartesian (x,y,z) geometry.
  Reading materials...
  Constructing VTS file...
  Writing VTS file...
  Done.
```

```
> l3d2xdmf -V linkout
Processing file: linkout
  Found 3-D Cartesian (x,y,z) geometry.
  Reading materials... Done.
  Number of x mesh: 125
  Number of y mesh: 120
  Number of z mesh: 24
  Total number of mesh: 360000
  Writing XDMF file... Done.
  Processing mesh and writing HDF5 file...
    Finished 125 of 125 mesh layer iterations... Done.
```

File Sizes

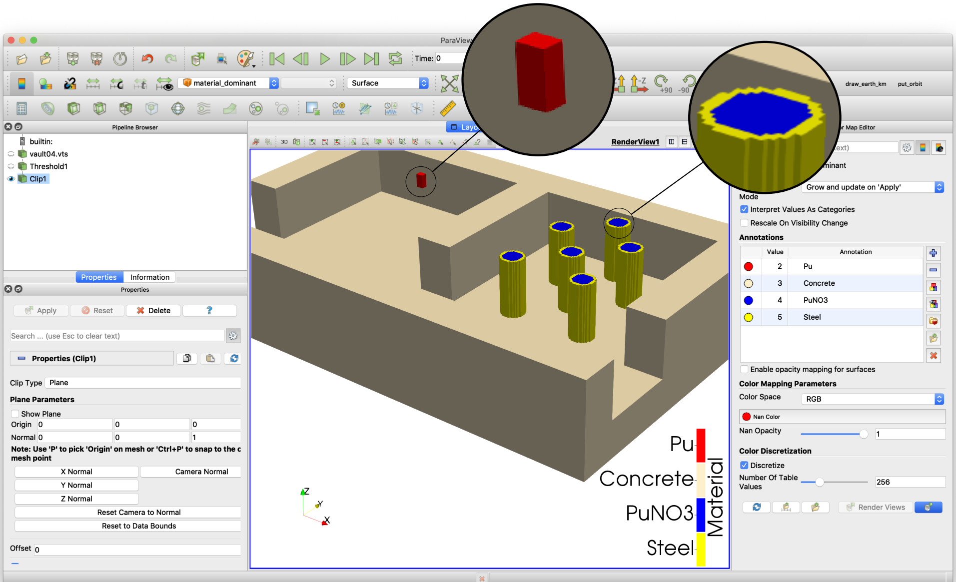
`linkout`: 8.3M

`lnk3dnt.vts`: 27M

`l3dfile.xdmf.h5`: 688k

Top: `l3d2vtk` (MCNPTools [2] v5.1.0+) [3].
Bottom: `l3d2xdmf` (LANL Only) [4].

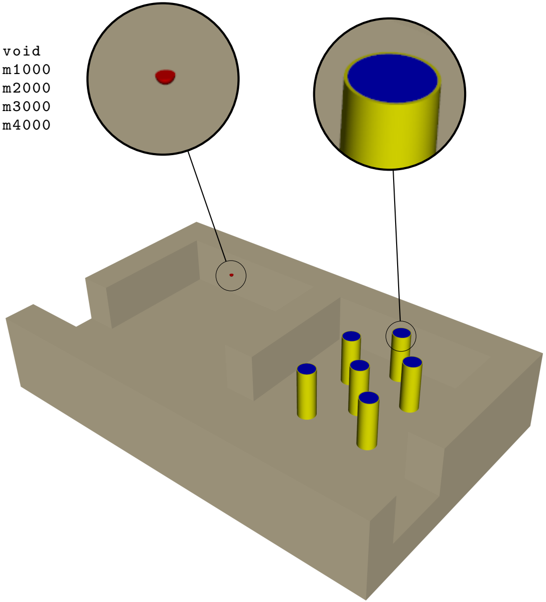
Interactively Viewing LNK3DNT Files [3]



Interface: ParaView [9], which is widely available at LANL for use on on desktop computers and/or through client-server connections to LANL HPC systems.

Compare with: Ray-traced CSG via Radiant [10]

```
colors {
  material     0     0    0    0  255   ! 0 void
  material    -1   255    0    0  255   ! 1 m1000
  material    -2   255  240  200  255   ! 2 m2000
  material    -3     0    0  255  255   ! 3 m3000
  material    -4   255  255    0  255   ! 4 m4000
}

visibility {
  materials_off  0
}

lights {
  light 0.0 0.0 -1.0  1.0
}

clipping {
  plane 0  0 0 150  0.0 0.0 1.0 false
}

image3d {
  background     255 255 255 255
  camera_pos     2048 -1250 1522
  view_angle     30.0
  view_focus     650 300 0
  view_pan       0 0
  view_up        -.34 0.43 0.83
  view_zoom       1.0
}
```

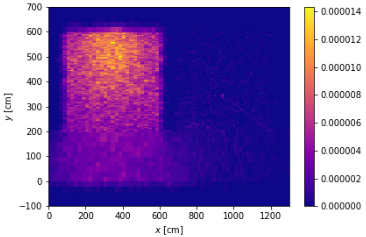


Generated with `radiant -t 4 -r 2560x2048 -a 4 radiant.inp runtpe`.

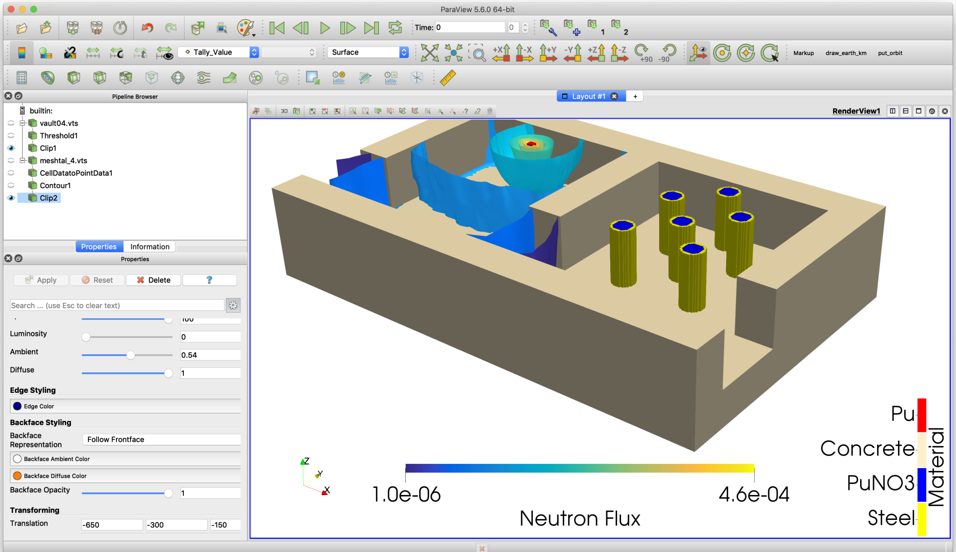
Mesh Tallies (fmesh)

- Several paths to a file that can be visualized interactively in 3-D space
  - MCNP6.2 and before: VTK via `meshtal2vtk` (MCNPTools v5.1.0)
  - MCNP6.3 New Option: `out=xdmf` (XDMF+HDF5)
- XDMF+HDF5 output in `runtpe.h5:/results/mesh_tally`
  - XDMF file must be kept with the runtape file
  - HDF5 results are useful independent of XDMF

```python
import h5py
import matplotlib.pyplot as plt
import numpy as np

f = h5py.File("vault04.txtr.h5", "r")
grid_x = f["/results/mesh_tally/mesh_tally_4/grid_x"]
grid_y = f["/results/mesh_tally/mesh_tally_4/grid_y"]
X, Y = np.meshgrid(grid_x, grid_y)
values = f["/results/mesh_tally/mesh_tally_4/mean"]
Z = values[0, 0, 5, :, :]
plt.pcolormesh(X, Y, Z, cmap="plasma")
plt.colorbar()

plt.xlabel("$x$ [cm]")
plt.ylabel("$y$ [cm]")
plt.tight_layout()

plt.savefig("vault04.png")
f.close()
```



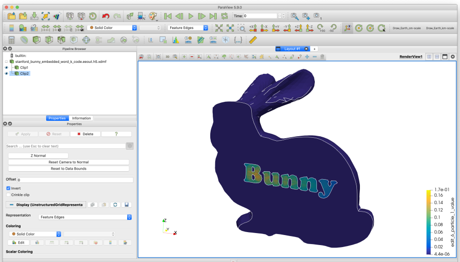
3-D, Interactive, Mesh Tallies + Geometry [3]



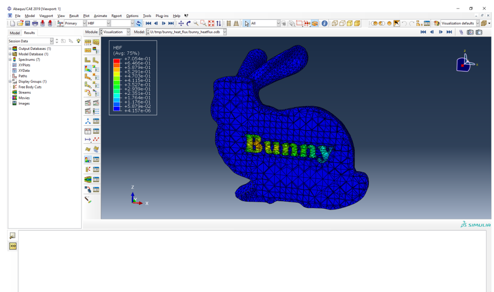
Mesh Tally Surfaces: Cell-to-point Data and Contour Filters.

Viewing UM Geometry & Results

- `um_post_op` could convert EEOUT to VTK [11]; recently unsuccessful
- Multiple post-processing capabilities exist, which include:
  - Attila4MC EEOUT-to-Tecplot [12, 13]
  - Python-based EEOUT-to-VTK [14]
  - Python-based EEOUT-to-Abaqus [15, 16]
- MCNP6.3: directly produce XDMF+HDF5 files from a UM calculation



EEOUT-to-VTK Converter



EEOUT-to-Abaqus Converter

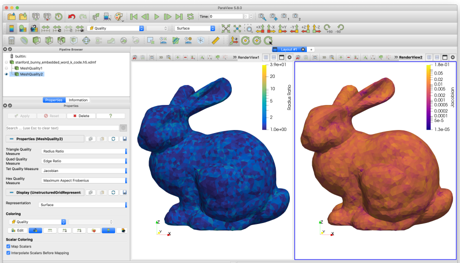
Los Alamos NATIONAL LABORATORY

Geometry adapted from [17].

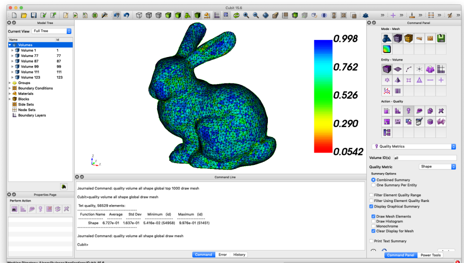
Assessing UM Elemental Quality

- MCNP6.3: Elemental quality assessment added to UM input processing
  - Default enabled with opt-out capability provided
  - The metrics assessed [18] are a subset of those in Verdict [19]
  - Goal: provide enough information to know when more is needed

Metric	Rec. Min.	Rec. Max.	Mean	Std. Dev.	Calc. Min.	Calc. Max.	# Outside	# Checked	% Outside
...									
Jacobian	1.000E-30	1.000E+30	1.363E-02	1.583E-02	1.335E-05	1.758E-01	0	56529	0.000
Radius Ratio	1.000E+00	3.000E+00	1.892E+00	1.084E+00	1.003E+00	3.874E+01	4674	56529	8.268
...									



ParaView



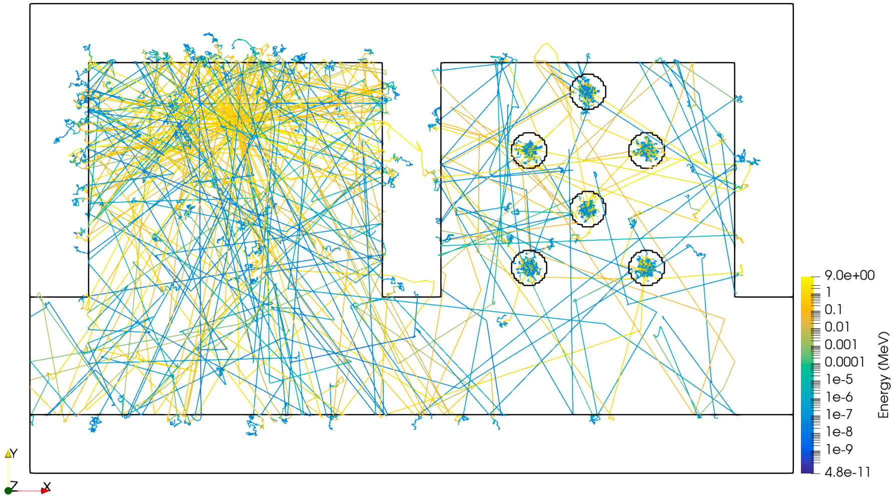
Cubit [20]

Los Alamos NATIONAL LABORATORY

Both ParaView and Cubit provide the same elemental quality metrics as available in Verdict. Geometry adapted from [17].

`ptrac`: Processing & Viewing

- Legacy ASCII & Binary `ptrac` file parsing with MCNPTools
- MCNP6.3: Produce HDF5 ptrac files; works with multiprocessing
  - Can be directly parsed, or interpreted with MCNPTools v5.2.0+



Energy-shaded particle tracks from first 1000 histories.

Summary & Future Work

Summary

- Demonstrated several approaches to visualization beyond MCNP plotter
  - Looked at various MCNP data in isolation and combination
- Explored various interfaces: ParaView, matplotlib, Abaqus, Cubit
- Provided code snippets and supplemental references

Future Work

- Improved approach(es) for parsing `ptrac` data for visualization
- Real-time, interactive, CSG (active research topic)
  - In the meantime, incorporate faithful CSG with results
- Migration of other data to (XDMF+)HDF5 (`rssa`, `srctp`, etc.)

Questions?

Contact Information

Joel A. Kulesza, Ph.D., P.E.

Office: +1 (505) 667-5467

jkulesza@lanl.gov

Introduction & Motivation

LNK3DNT
- Generating from MCNP CSG, Converting to VTK/XDMF+HDF5, and Viewing
- Quick Comparison with Ray-traced CSG via Radiant

Mesh Tallies (`fmesh`)
- Converting to VTK, Producing as XDMF+HDF5, and Viewing

Unstructured Mesh
- Viewing UM Geometry & Results, Assessing Elemental Quality
- HDF5 Batch-processing Example

Particle Track (`ptrac`)
- New HDF5 Format; Example Processing & Viewing

Summary & Future Work

Backup Slides

References

[1] C. J. Werner, J. S. Bull, C. J. Solomon, F. B. Brown, G. W. McKinney, M. E. Rising, D. A. Dixon, R. L. Martz, H. G. Hughes, L. J. Cox, A. J. Zukaitis, J. C. Armstrong, R. A. Forster, and L. Casswell, "MCNP® Version 6.2 Release Notes," Tech. Rep. LA-UR-18-20808, Los Alamos National Laboratory, Los Alamos, NM, USA, 2018.

[2] C. J. Solomon, C. R. Bates, and J. A. Kulesza, "The MCNPTools Package: Installation and Use," Tech. Rep. LA-UR-17-21779, Los Alamos National Laboratory, Los Alamos, NM, USA, Mar. 2017.

[3] J. A. Kulesza, J. L. Alwin, J. D. Hutchinson, E. F. Shores, and R. C. Little, "l3d2vtk: An MCNPTools Utility to Enable LNK3DNT File Visualization & Post-processing," in *Transactions of the Winter 2019 American Nuclear Society Meeting*, vol. 121, pp. 1233–1236, Nov. 2019.

[4] J. A. Kulesza, J. L. Alwin, E. F. Shores, J. D. Hutchinson, and R. C. Little, "`l3d2xdmf`: Requirements, Testing, and Use," Tech. Rep. LA-UR-19-29667, Los Alamos National Laboratory, Los Alamos, NM, USA, Sept. 2019.

References

[5] L. S. Avila, U. Ayachit, S. Barré, J. Baumes, F. Bertel, R. Blue, D. Cole, D. DeMarle, B. Geveci, W. A. Hoffman, B. King, K. Krishnan, C. C. Law, K. M. Martin, W. McLendon, P. Pebay, N. Russell, W. J. Schroeder, T. Shead, J. Shepherd, A. Wilson, and B. Wylie, *The VTK User's Guide*. Kitware, Inc., 11 ed., 2010.

[6] "XDMF Model and Format." Website, Mar. 2019.

[7] J. A. Clarke and E. R. Mark, "Enhancements to the eXtensible Data Model and Format (XDMF)," in *HPCMP User's Group Conference 2007. High Performance Computing Modernization Program: A Bridge to Future Defense*, (Pittsburgh, PA, USA; June 18–21), pp. 322–327, 2007.

[8] The HDF Group, "Hierarchical Data Format, version 5." Website, 1997–2020. Last Accessed: Feb. 2020.

[9] U. Ayachit, *The ParaView Guide*. Kitware, Inc., community ed., June 2018.

[10] RSICC, "ADVANTG 3.2.0: AutomateD VAriaNce reducTion Generator," RSICC Code Package CCC-854, Oak Ridge National Laboratory, Oak Ridge, TN, USA, Aug. 2019.

References

[11] R. L. Martz, “Unstructured Mesh User’s Startup Guide,” Tech. Rep. LA-UR-12-00795, Los Alamos National Laboratory, Los Alamos, NM, USA, Feb. 2012.

[12] “Attila4MC 10.2 Overview of Core Functions,” Tech. Rep. SFSW-UR-2020-OCF102, Silver Fir Software, Inc., Gig Harbor, WA, USA, 2020.

[13] Tecplot, Inc., Bellevue, WA, USA, *Tecplot 360 EX User’s Manual*, Tecplot 360 EX 2018 Release 2 ed., Oct. 2018.

[14] J. A. Kulesza and T. C. McClanahan, “A Python Script to Convert MCNP Unstructured Mesh Elemental Edit Output Files to XML-based VTK Files,” Tech. Rep. LA-UR-19-20291, Rev. 2, Los Alamos National Laboratory, Los Alamos, NM, USA, Nov. 2019.

[15] V. K. Mehta and J. C. Armstrong, “Processing MCNP Elemental Edit Outputs,” Tech. Rep. LA-UR-20-24025, Los Alamos National Laboratory, Los Alamos, NM, USA, June 2020.

References

[16] Dassault Systèmes Simulia Corp., “Abaqus/CAE 6.12 Online Documentation,” 2012.

[17] Stanford Computer Graphics Laboratory, “The Stanford 3D Scanning Repository,” Aug. 2017.

[18] J. A. Kulesza, “MCNP® Code Version 6.3.0 Unstructured-mesh Quality Metrics & Assessment,” Tech. Rep. LA-UR-20-27150 (Draft), Los Alamos National Laboratory, Los Alamos, NM, USA, Sept. 2020.

[19] C. J. Stimpson, C. D. Ernst, P. K. and. P. P. Pébay, and D. Thompson, “The Verdict Geometric Quality Library,” Tech. Rep. SAND2007-1751, Sandia National Laboratories, Albuquerque, NM, USA, Mar. 2007.

[20] Sandia National Laboratories, “CUBIT Toolkit.”

[21] J. A. Kulesza, “Recent Work on HDF5 and XDMF-based MCNP Unstructured Mesh Output Files,” in *Proceedings of 14th International Conference on Radiation Shielding (ICRS14) and 21st Topical Meeting of the Radiation Protection and Shielding Division (RPSD2020)*, (Seattle, WA, USA; September 13–17), 2020.

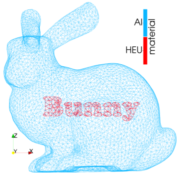
References

[22] J. A. Kulesza, “Visualization Advancements for the MCNP Code (and a Post-processing Wish List),” in *Proceedings of Department of Energy Computer Graphics Forum*, (Santa Fe, NM, USA; April 28–30), 2020. Also: LA-UR-20-22891.

[23] J. L. Alwin, J. B. Spencer, and G. A. Failla, “Criticality Accident Alarm System Analysis using MCNP6.2 Constructive Solid Geometry/ Unstructured Mesh Hybrid,” in *Proceedings of 11th International Conference on Nuclear Criticality Safety (ICNC2019)*, (Paris, France, September 15–20), 2019.

[24] J. Ellson, E. Gansner, L. Koutsofios, S. North, and G. Woodhull, “Graphviz—Open Source Graph Drawing Tools,” in *Lecture Notes in Computer Science*, pp. 483–484, Springer-Verlag, 2001.

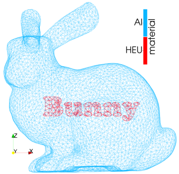
MCNP UM HDF5 Batch Use—Python (h5py)




```
#!/usr/bin/env python

import h5py
import numpy as np

f = h5py.File("stanford_bunny_embedded_word_k_code.h5", "r")

fluence = 0.0
volume = 0.0
for i in f["unstructured_mesh"]["instances"].items():
    e = i[1]["edits"]["edit_4_particle_1"]["values"]
    v = i[1]["volume"]
    fluence += np.inner(e, v)
    volume += np.sum(v)
print(f"Avg. Fluence: {fluence / volume:.3e}")
```

Avg. Fluence: $$\frac{\int \phi(\vec{x}) \mathrm{d}V}{\int \mathrm{d}V}$$

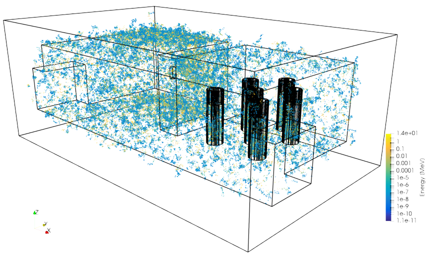
MCNP UM HDF5 Batch Use—ParaView (pvpython)

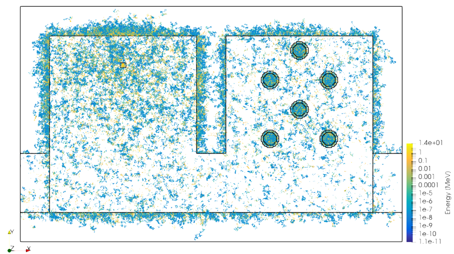



```
#!/Users/jkulesza/Applications/ParaView-5.8.0.app/Contents/bin/pvpython

import numpy as np
from paraview.simple import *
from paraview import servermanager as pvsm

# Setup ParaView.
paraview.simple._DisableFirstRenderCameraReset()
renderView1 = GetActiveViewOrCreate("RenderView")

# Load data.
bunny = XDMFReader(FileNames=["stanford_bunny_embedded_word_k_code.h5.xdmf"])

# Integrate and divide.
iv1 = IntegrateVariables(Input=bunny)
numer = np.array(pvsm.Fetch(iv1).GetCellData().GetArray("edit_4_particle_1_values"))[0]
denom = np.array(pvsm.Fetch(iv1).GetCellData().GetArray("Volume"))[0]
print("Avg. Fluence: {:.3e}".format(numer / denom))
```

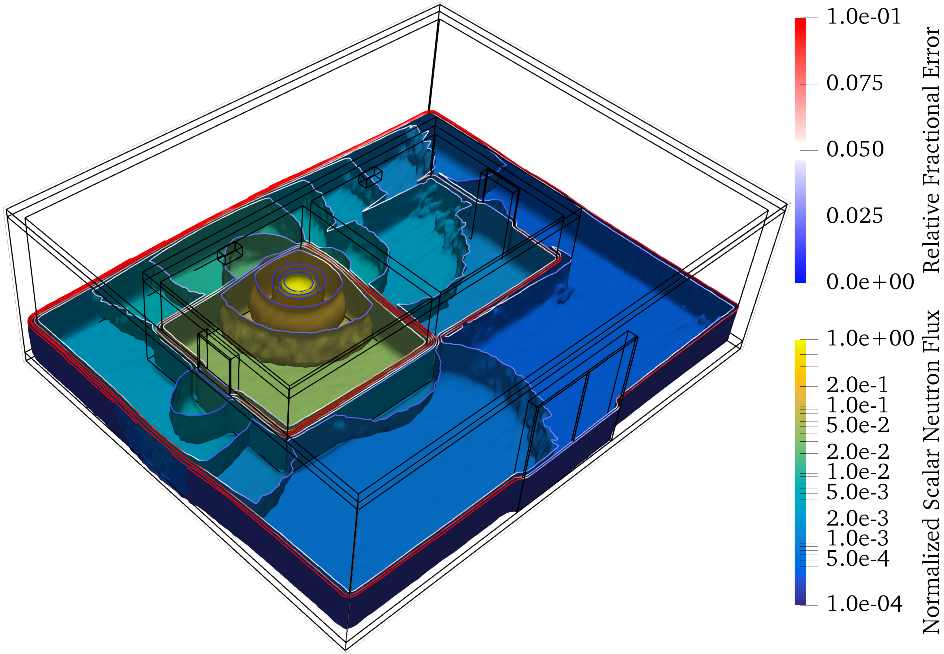
ptrac HDF5 Batch Use—Python (h5py)

```
#!/usr/bin/env python

import h5py
import numpy as np

f = h5py.File("ptrac.h5", "r")
x = f["ptrack/Collision"]["x"]
y = f["ptrack/Collision"]["y"]
z = f["ptrack/Collision"]["z"]
e = f["ptrack/Collision"]["energy"]
d = np.vstack((x, y, z, e))
np.savetxt("Collisions.csv", d.T, delimiter=",", header="x,y,z,e")
```





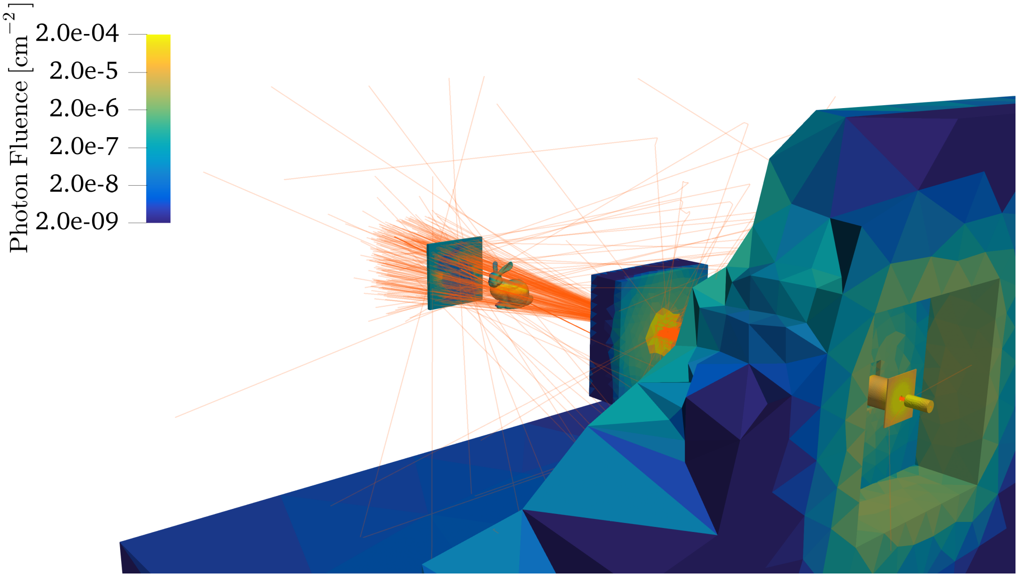
Example Visualization: Criticality Accident Alarm System



Los Alamos NATIONAL LABORATORY

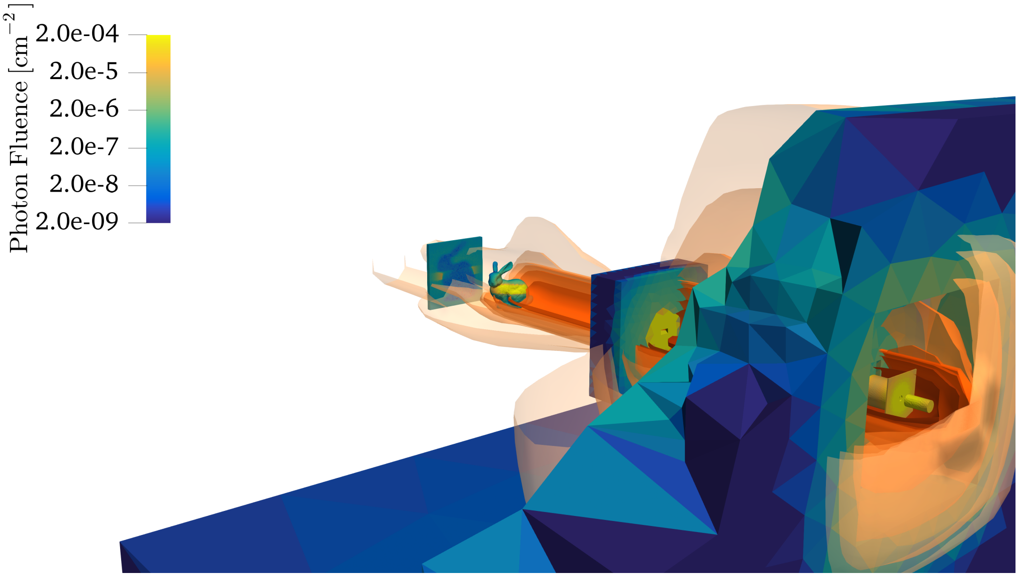
From [14, 22, 23].

Example Visualization: Notional Radiograph Facility



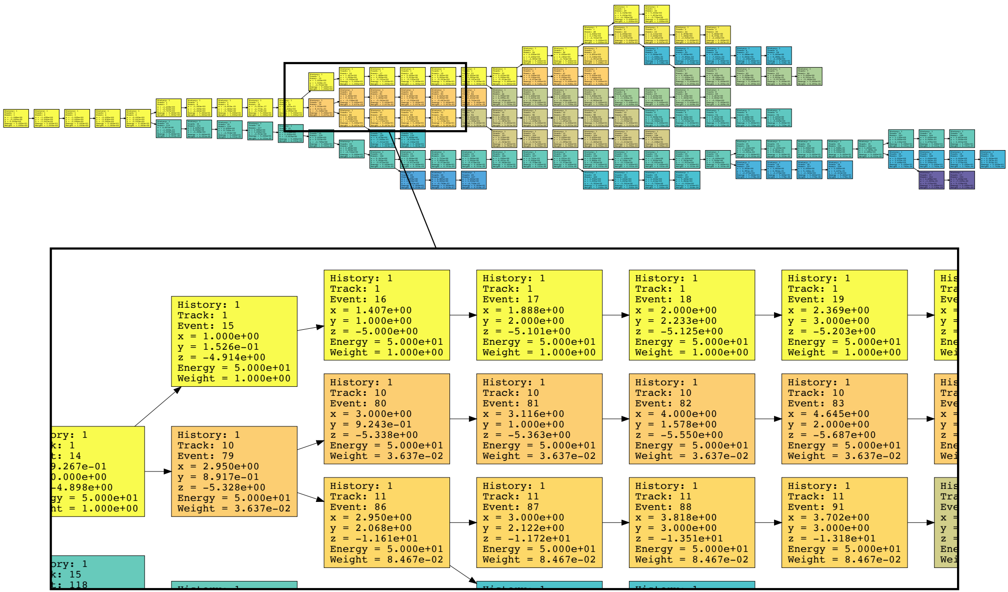
From [22].

Example Visualization: Notional Radiograph Facility, cont.



From [22].

Example Visualization: Particle Evolution via Graphviz [24]



From [22].