LA-UR-21-26365
Approved for public release; distribution is unlimited.

Title: MCNP Unstructured Mesh Visualization & Post-processing Techniques

Author(s): Kulesza, Joel A.; Armstrong, Jerawan Chudoung; Josey, Colin James; Swaminarayan, Sriram; Alwin, Jennifer Louise; McClanahan, Tucker Caden; Spencer, Joshua Bradly; Goorley, John Timothy; Kelley, Karen Corzine; Khalsa, Prabhu Singh; Failla, Gregory A.

Intended for: 2021 MCNP User Symposium, 2021-07-12/2021-07-16 (Online, New Mexico, United States)

Issued: 2021-07-06

Disclaimer:
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.

Los Alamos
NATIONAL LABORATORY

MCNP Unstructured Mesh Visualization & Post-processing Techniques

Joel A. Kulesza, Jerawan C. Armstrong, Colin J. Josey, Sriram Swaminarayan, Jennifer L. Alwin, Tucker C. McClanahan, Joshua B. Spencer, John T. Goorley, Karen C. Kelley, Prabhu S. Khalsa, and Gregory A. Failla

2021 MCNP® User Symposium
July 12–16, 2021

Managed by Triad National Security, LLC, for the U.S. Department of Energy's NNSA.

Acknowledgements

This work is supported by the Department of Energy National Nuclear Security Administration (NNSA) Advanced Simulation and Computing (ASC) Program and Engineering Campaign 7.

Outline

Review of MCNP Unstructured Mesh File Formats

- (Legacy ASCII) EEOUT
- HDF5+XDMF Output (and Input)

ASCII EEOUT File Conversion to VTK, Abaqus, and Tecplot

Post-processing & Visualization Workflows

- Batch
  - Python (`h5py`)
  - ParaView (`pvpython`)
- Interactive
  - HDFView and ViTables
  - ParaView

(Legacy ASCII) EEOUT File Format Review

- MCNP UM **e**lemental **e**dit **out**put file [1]: ASCII or binary
  - Boutique format: unique to the MCNP code (currently version 6)
  - ASCII appears to dominate use in practice

- Stores results and provides restart information, however:
  - Not immediately usable in many other widely available tools [2, 3]
  - Limited precision in ASCII values (5 digits)
  - No intrinsic data compression

- `um_post_op` [4] was developed to process EEOUT files
  - Fortran-based utility that uses MCNP UM routines to process EEOUT
  - Has not received extensive testing [1, §13.8]
  - Minimal success with recent VTK conversion; prompted creating [5]
  - Future support for `um_post_op` is not planned

HDF5+XDMF UM I/O File Format Review

HDF [6]

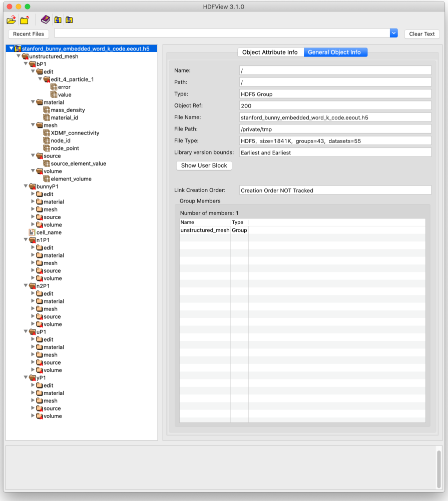
- Hierarchical Data Format; developed by The HDF Group
- BSD-like license, freely available, portable, numerous APIs
- Developed with speed and scalability in mind: compressible binary
  - Also offers serial and parallel I/O
  - Binary, but easy to work with (demonstrated soon)
- Three major objects: groups, datasets, and attributes

XDMF [7, 8]

- ASCII XML file; standard, but not actively developed, file format
- Supports various structured and unstructured geometries
- Can contain data and/or **contain text pointers into HDF5 files**
- MCNP outputs are XDMF version 2

Current HDF5 File Hierarchy

- This is a representative illustration
- Some components are optional



```
▼ stanford_bunny_embedded_word_k_code.eeout.h5
  ▼ unstructured_mesh
    ▼ bP1
      ▼ edit
        ▼ edit_4_particle_1
            error
            value
      ▼ material
          mass_density
          material_id
      ▼ mesh
          XDMF_connectivity
          node_id
          node_point
      ▼ source
          source_element_value
      ▼ volume
          element_volume
    ▼ bunnyP1
      ▶ edit
      ▶ material
      ▶ mesh
      ▶ source
      ▶ volume
      cell_name
    ▼ n1P1
      ▶ edit
      ▶ material
      ▶ mesh
      ▶ source
      ▶ volume
    ▼ n2P1
      ▶ edit
      ▶ material
      ▶ mesh
      ▶ source
      ▶ volume
    ▼ uP1
      ▶ edit
      ▶ material
      ▶ mesh
      ▶ source
      ▶ volume
    ▼ yP1
      ▶ edit
      ▶ material
      ▶ mesh
      ▶ source
      ▶ volume
```

Name: /

Path: /

Type: HDF5 Group

Object Ref: 200

File Name: stanford_bunny_embedded_word_k_code.eeout.h5

File Path: /private/tmp

File Type: HDF5, size=1841K, groups=43, datasets=55

Library version bounds: Earliest and Earliest

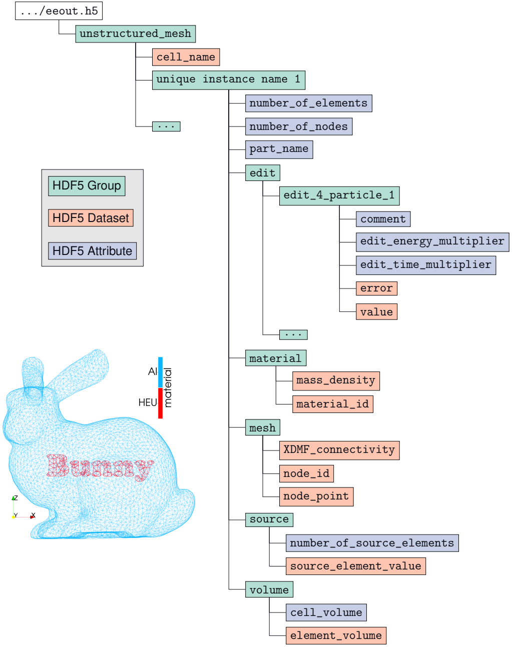
Show User Block

Link Creation Order: Creation Order NOT Tracked

Group Members

Number of members: 1

Name	Type
unstructured_mesh	Group



```
.../eeout.h5
  unstructured_mesh
    cell_name
    unique instance name 1
      number_of_elements
      number_of_nodes
      part_name
      edit
        edit_4_particle_1
          comment
          edit_energy_multiplier
          edit_time_multiplier
          error
          value
        ...
      material
        mass_density
        material_id
      mesh
        XDMF_connectivity
        node_id
        node_point
      source
        number_of_source_elements
        source_element_value
      volume
        cell_volume
        element_volume
    ...
```

🟩 HDF5 Group

🟧 HDF5 Dataset

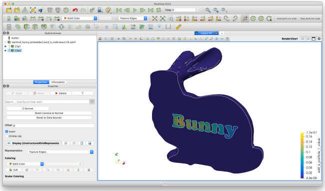
🟦 HDF5 Attribute

Geometry adapted from [9].

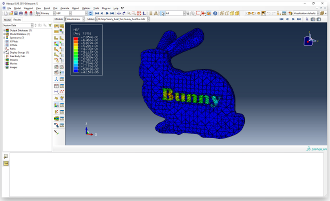
ASCII EEOUT File Conversion to VTK, Abaqus, and Tecplot

- Attila4MC EEOUT-to-Tecplot [3, 10]
- Python-based EEOUT-to-VTK [5]
- Python-based EEOUT-to-Abaqus [11, 12]

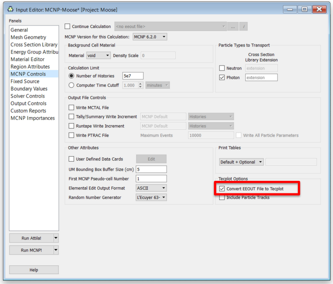
```
> ./Convert_MCNP_eeout_to_VTK.py stanford_bunny_embedded_word_k_code.eeout
Processing stanford_bunny_embedded_word_k_code.eeout...
  Found 1 edit(s).
  Processing FLUX_6 edit...
    Processing & Validating EDIT_6_RESULT_...
    Maximum          value: 1.71341e-01
    Minimum positive value: 4.43231e-06
    Minimum          value: 0.00000e+00
```



ParaView



Abaqus



Input Editor: MCNP-Moose* [Project: Moose]

Panels: General, Mesh Geometry, Cross Section Library, Energy Group Attribu, Material Editor, Region Attributes, MCNP Controls, Fixed Source, Boundary Values, Solver Controls, Output Controls, Custom Reports, MCNP Importances

□ Continue Calculation <no eeout file>

MCNP Version for this Calculation: MCNP 6.2.0

Background Cell Material: Material void, Density Scale 0

Calculation Limit: ● Number of Histories 5e7; ○ Computer Time Cutoff 1.000 minutes

Particle Types to Transport (Cross Section Library Extension): □ Neutron extension; ☑ Photon extension

Output File Controls: □ Write MCTAL File; □ Tally/Summary Write Increment MCNP Default Histories; □ Runtape Write Increment MCNP Default Histories; □ Write PTRAC File, Maximum Events 10000, □ Write All Particle Parameters

Other Attributes: □ User Defined Data Cards (Edit); UM Bounding Box Buffer Size (cm) 5; First MCNP Pseudo-cell Number 1; Elemental Edit Output Format ASCII; Random Number Generator L'Ecuyer 63-

Print Tables: Default + Optional

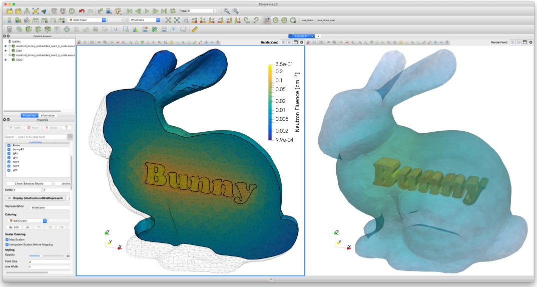
Tecplot Options: ☑ Convert EEOUT File to Tecplot; □ Include Particle Tracks

Attila4MC

Batch Use: Python (h5py)

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

import h5py
import numpy as np

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

fluence = 0.0
volume = 0.0
for g in f["unstructured_mesh"].items():
    if isinstance(g[1], h5py.Group): # ignore ‘cell_name‘
        e = g[1]["edit/edit_4_particle_1/value"]
        v = g[1]["volume/element_volume"]
        fluence += np.inner(e, v)
        volume += np.sum(v)
print(f"Avg. Fluence: {fluence / volume:.3e}")
```

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

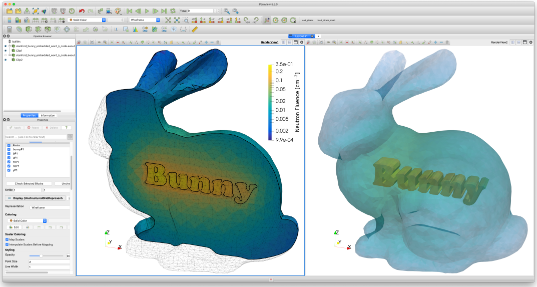


Los Alamos
NATIONAL LABORATORY

Batch Use: ParaView (`pvpython`)

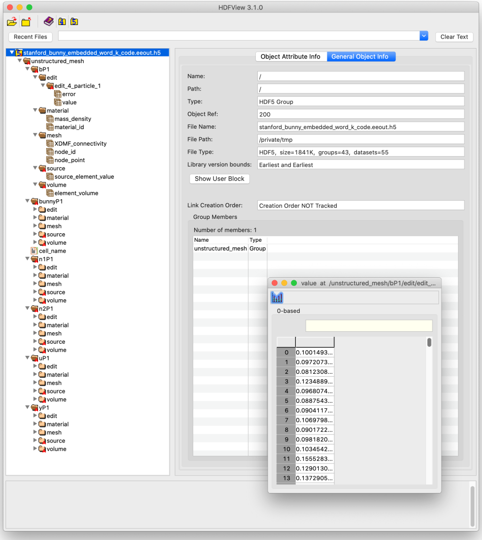
```
#!/Users/jkulesza/Applications/ParaView-5.9.1.app/Contents/bin/pvpython

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.eeout.h5.xdmf"])

# Integrate and divide.
iv1 = IntegrateVariables(Input=bunny)
numer = np.array(pvsm.Fetch(iv1).GetCellData().GetArray("edit_4_particle_1_value"))[0]
denom = np.array(pvsm.Fetch(iv1).GetCellData().GetArray("Volume"))[0]
print("Avg. Fluence: {:.3e}".format(numer / denom))
```

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



Los Alamos NATIONAL LABORATORY

Interactive Use: HDFView [13] & ViTables [14]



Name: /

Path: /

Type: HDF5 Group

Object Ref: 200

File Name: stanford_bunny_embedded_word_k_code.eeout.h5

File Path: /private/tmp

File Type: HDF5, size=1841K, groups=43, datasets=55

Library version bounds: Earliest and Earliest

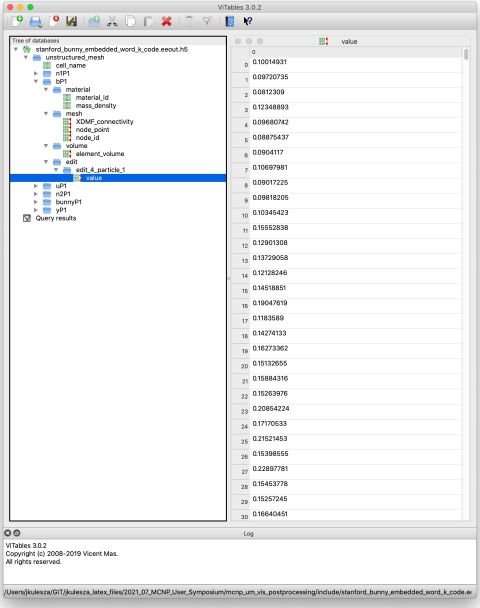
Link Creation Order: Creation Order NOT Tracked

Group Members

Number of members: 1

Name	Type
unstructured_mesh	Group

0	0.1001493...
1	0.0972073...
2	0.0812308...
3	0.1234889...
4	0.0968074...
5	0.0887543...
6	0.0904117...
7	0.1069798...
8	0.0901722...
9	0.0981820...
10	0.1034542...
11	0.1555283...
12	0.1290130...
13	0.1372905...



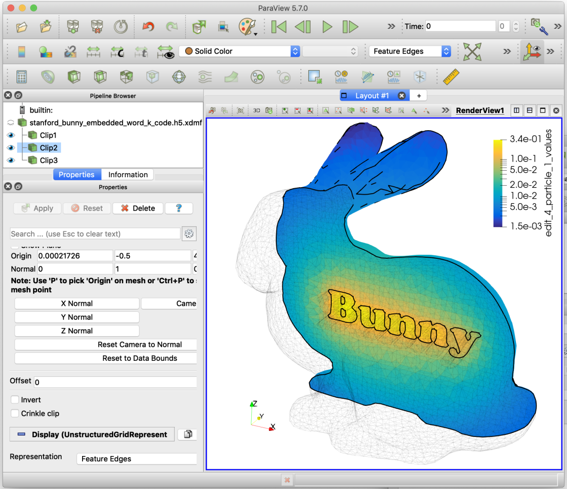
	0
0	0.10014931
1	0.09720735
2	0.0812309
3	0.12348893
4	0.09680742
5	0.08875437
6	0.0904117
7	0.10697981
8	0.09017225
9	0.09818205
10	0.10345423
11	0.15552838
12	0.12901308
13	0.13729058
14	0.12128246
15	0.14518851
16	0.19047619
17	0.1183589
18	0.14274133
19	0.16273362
20	0.15132655
21	0.15884316
22	0.15263976
23	0.20854224
24	0.17170533
25	0.21521453
26	0.15398555
27	0.22897781
28	0.15453778
29	0.15257245
30	0.16640451

```
ViTables 3.0.2
Copyright (c) 2008-2019 Vicent Mas.
All rights reserved.
```

/Users/jkulesza/GIT/jkulesza_latex_files/2021_07_MCNP_User_Symposium/mcnp_um_vis_postprocessing/include/stanford_bunny_embedded_word_k_code.e

Los Alamos NATIONAL LABORATORY

Interactive Use: ParaView



ParaView 5.7.0

Time: 0

Solid Color

Feature Edges

Pipeline Browser

- builtin:
- stanford_bunny_embedded_word_k_code.h5.xdmf
  - Clip1
  - Clip2
  - Clip3

Properties | Information

Properties

Apply | Reset | Delete | ?

Search ... (use Esc to clear text)

Origin 0.00021726 | -0.5

Normal 0 | 1

Note: Use 'P' to pick 'Origin' on mesh or 'Ctrl+P' to : mesh point

X Normal | Came

Y Normal

Z Normal

Reset Camera to Normal

Reset to Data Bounds

Offset 0

□ Invert

□ Crinkle clip

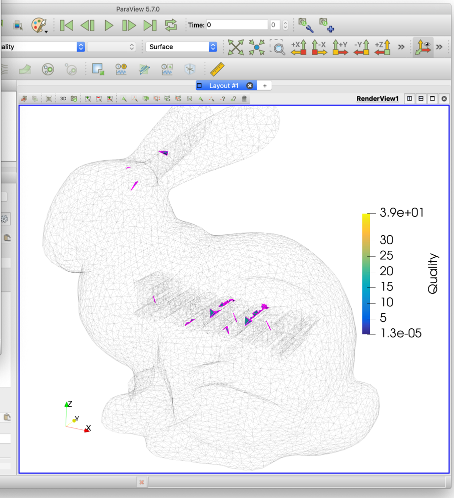
Display (UnstructuredGridRepresent

Representation: Feature Edges

Layout #1

RenderView1

Clipping; feature edges



ParaView 5.7.0

Time: 0

Surface

Layout #1

RenderView1

□ Preserve Topology

Display (UnstructuredGridRepresent

Representation: Surface

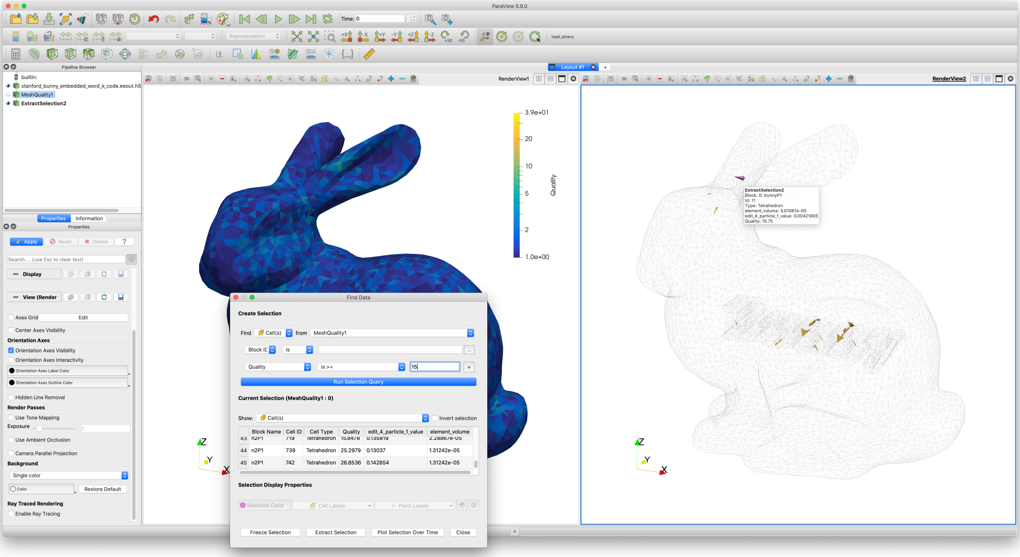
Coloring

Quality

Los Alamos NATIONAL LABORATORY

Segregating mesh quality (see next slide)

Interactive Use: ParaView, cont.



Pipeline Browser:
- builtin:
- stanford_bunny_embedded_word_k_code.eeout.h5
- MeshQuality1
- ExtractSelection2

Properties:
- □ Axes Grid
- □ Center Axes Visibility
- **Orientation Axes**
- ☑ Orientation Axes Visibility
- □ Orientation Axes Interactivity
- Orientation Axes Label Color
- Orientation Axes Outline Color
- □ Hidden Line Removal
- **Render Passes**
- □ Use Tone Mapping
- Exposure 1
- □ Use Ambient Occlusion
- □ Camera Parallel Projection
- **Background**
- Single color
- Color
- **Ray Traced Rendering**
- □ Enable Ray Tracing

Find Data

Create Selection

Find Cell(s) from MeshQuality1

Block ID is

Quality is >= 15

Run Selection Query

Current Selection (MeshQuality1 : 0)

Show: Cell(s) □ Invert selection

	Block Name	Cell ID	Cell Type	Quality	edit_4_particle_1_value	element_volume
43	n2P1	719	Tetrahedron	15.8478	0.135819	2.28867e-05
44	n2P1	739	Tetrahedron	25.2979	0.13037	1.31242e-05
45	n2P1	742	Tetrahedron	26.8536	0.142854	1.31242e-05

Selection Display Properties

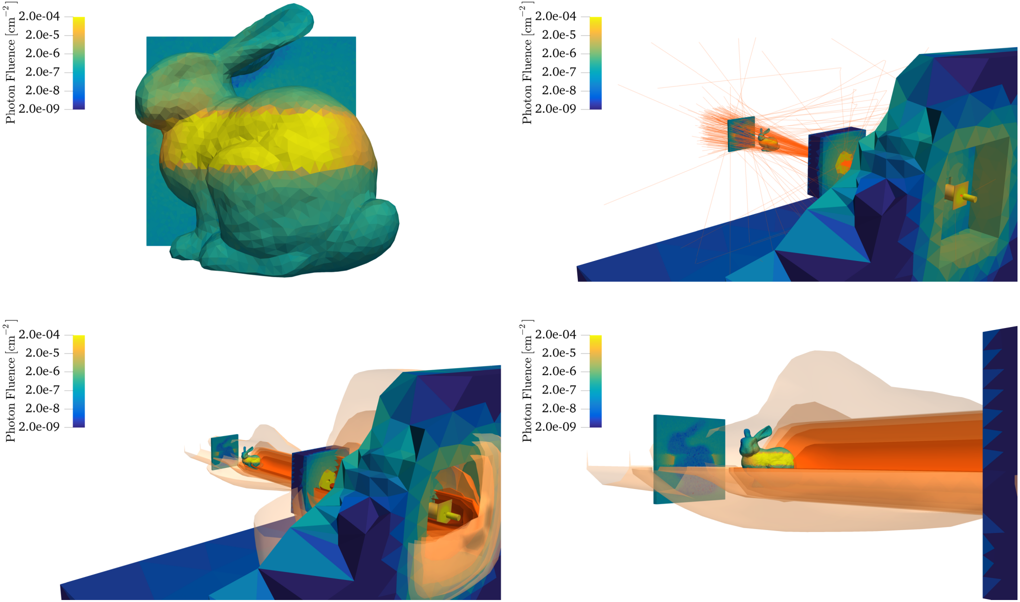
Selection Color | Cell Labels | Point Labels

Freeze Selection | Extract Selection | Plot Selection Over Time | Close

ExtractSelection2
Block: 0: bunnyP1
Id: 11
Type: Tetrahedron
element_volume: 5.01061e-05
edit_4_particle_1_value: 0.00421805
Quality: 15.75

Los Alamos
NATIONAL LABORATORY

Interactive Use with other Results



Adobe Acrobat: click image for animation (hopefully; alternative: right-click and Save Video As...).

Summary & Future Work

Summary

- Reviewed legacy and forthcoming MCNP UM files
- Several batch and interactive, qualitative and quantitative, workflows
  - A variety of freely available and commercial software were explored
- Provided code snippets and supplemental references

Future Work

- What more can be done to increase ease and/or adoption?
- Would a hands-on workshop or other in-depth presentation be helpful?
  - If so, what is the right venue, duration, level of detail?
- R&D: what can be done with CSG to complement this work with UM?

Questions?

Review of MCNP Unstructured Mesh File Formats

- (Legacy ASCII) EEOUT
- HDF5+XDMF Output (and Input)

ASCII EEOUT File Conversion to VTK, Abaqus, and Tecplot

Post-processing & Visualization Workflows

- Batch
  - Python (`h5py`)
  - ParaView (`pvpython`)
- Interactive
  - HDFView and ViTables
  - ParaView

Backup Slides

References

[1] R. L. Martz, "The MCNP6 Book On Unstructured Mesh Geometry: User's Guide For MCNP 6.2," Tech. Rep. LA-UR-17-22442, Los Alamos National Laboratory, Los Alamos, NM, USA, Mar. 2017.

[2] R. L. Martz, "TranzViz User's Guide Version 0.7.4," Tech. Rep. SSS-UM-001, Silver Fir Software, Wadley, GA, USA, Jan. 2020.

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

[4] 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.

[5] 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.

References

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

[7] “XDMF Model and Format.” Website, Mar. 2019.

[8] 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.

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

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

[11] 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

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

[13] The HDF Group, “HDFView Software.” Website, Feb. 2020.

[14] V. Mas, “ViTables Software.” Website, Feb. 2020.

Abstract

This talk describes several techniques for post processing MCNP6 ASCII unstructured mesh (UM) elemental-edit output (EEOUT) files as well as HDF5 EEOUT files expected to be present in the upcoming release of the MCNP code, version 6.3.

The talk begins with a brief overview of the ASCII output format and a description of the new HDF5 UM EEOUT file. Several readily available utilities and approaches to working with the ASCII files are reviewed that current code users may be unfamiliar with. Next, ways to interrogate the upcoming HDF5 files with GUI applications and the Python programming language are described. The talk concludes with suggested approaches to interactively viewing and querying the HDF5 files through an accompanying XDMF file, which is expected to be generated directly alongside the upcoming HDF5 files.