

LA-UR-26-27246

Approved for public release; distribution is unlimited.

Title: Two Recent Enhancements to MCNP Unstructured Mesh

Author(s): Armstrong, Jerawan Chudoung

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

Issued: 2026-09-16 (rev.1)



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.

Two Recent Enhancements to MCNP Unstructured Mesh

Jerawan Armstrong
Monte Carlo Codes Group (XCP-3), LANL

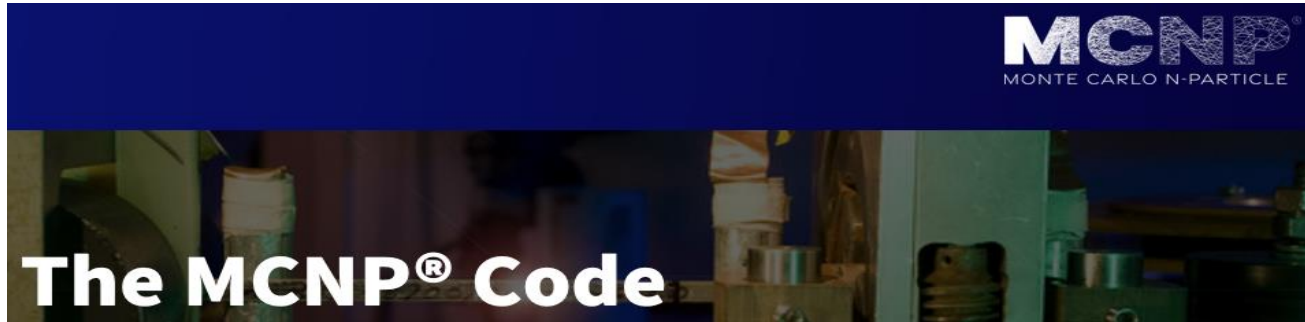
2026 MCNP User Symposium
August 24-26, 2026

LA-UR-26-27246, Rev. 1

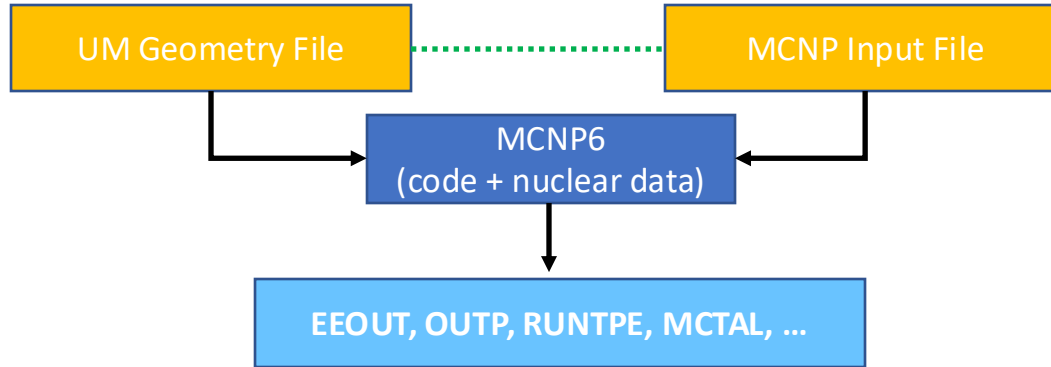
Outline

- **Enhancement 1:** HDF5-Based Processed Mesh Reuse
- **Enhancement 2:** Simplified Continue Runs for Unstructured Mesh Calculations

<https://mcnp.lanl.gov>



MCNP Unstructured Mesh (UM) Calculations



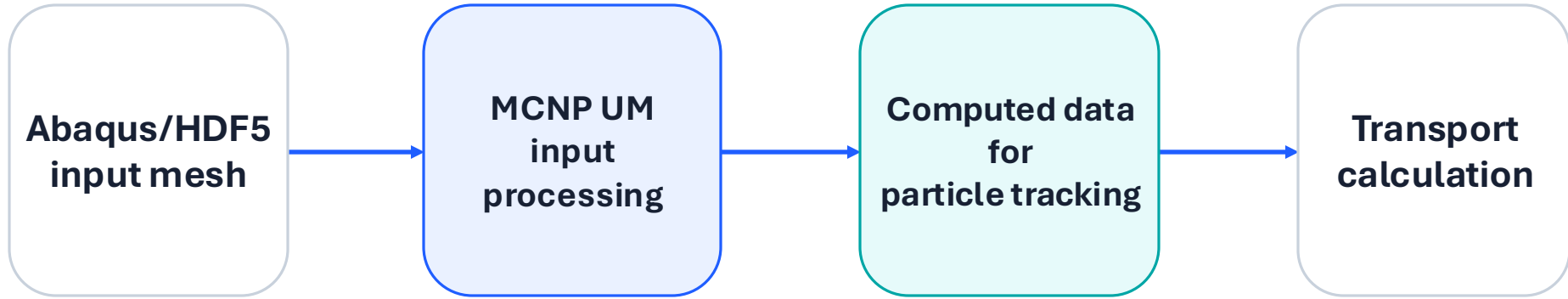
UM Geometry File :

- Define on EMBED card
 - **MESHGEO** for file format
 - **MGEOIN** for file name
- Supported file formats:
 - Abaqus input
 - MCNPUM
 - HDF5 mesh input

HDF5 EEOUT (Elemental Edit OUTput) File:

- Specify file name on EMBED card
 - **HDF5FILE** for EEOUT file name
- HDF5 mesh input + element-level results

Unstructured Mesh (UM) Setup Work



Different MCNP calculations may share the same mesh model but still trigger redundant setup

Cost of Unstructured Mesh Input Processing [1]

- Requires substantial memory and computational effort for large mesh models
- Abaqus and HDF5 mesh input files contain node positions and element connectivity
- For Abaqus mesh input files, pseudo-cell data are computed from part and instance data
- Mesh input files do not contain data required for particle tracking, including
 - Element information (volumes, centroid, bounding boxes, faces, nearest neighbors)
 - Element face information (centroid, bounding boxes, adjacent elements)
 - Cell bounding boxes
 - Spatial search structures such as SKD-Trees
 - Elemental edit data structures for storing results

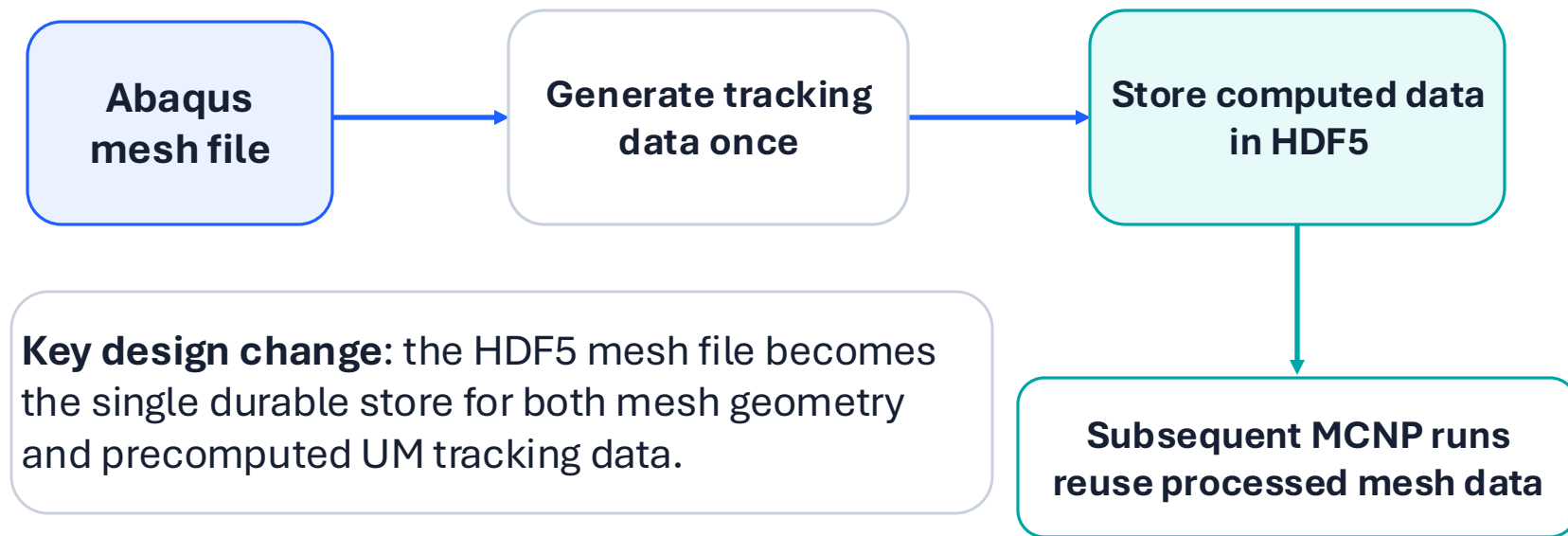
Cost of Unstructured Mesh Input Processing [2]

- MPI scalability limitations:
 - Efficiency depends on the number of parts and pseudo-cells
 - Excess MPI ranks may remain idle during input processing
 - More ranks per node reduce memory available per rank
 - Mesh data must be distributed, collected, combined, and redistributed before particle tracking
- OpenMP threading limitations:
 - Elemental edit array sizes increase with the number of threads
 - Additional threads increase memory usage
 - Threading across HPC compute nodes can degrade performance

Legacy MCNPUM File Format

- Introduced in MCNP6.2
- Reads and processes an Abaqus input file
- Computes mesh-derived tracking data and stores for reuse in subsequent calculations
- **Advantage:**
 - Significantly reduce input processing time and memory usage in subsequent runs
- **Limitations:**
 - Produced large binary or flat files with no well-defined data structure
 - The implementation is difficult to maintain
 - Deprecated in MCNP6.3 and scheduled for removal in MCNP6.4

Replace MCNPUM with HDF5-Based Mesh Reuse



Eliminates: the legacy MCNPUM file format

Preserves: processed-data reuse across calculations sharing the same mesh model

Status: completed, reviewed, and merged into the main MCNP code base

Convert Abaqus Mesh to HDF5-Based Processed Mesh

- Use the new **DATAWRITE** keyword on the **EMBED** card
- Specify both **HDF5FILE** and **DATAWRITE=YES**
- **DATAWRITE=YES** writes the **unstructured_mesh/computed_data** group to the HDF5 file specified by the **HDF5FILE** keyword
- Run MCNP in input-only mode:
 - **mcnp6 i i=<mcnp_input_file>**
- HDF5-based mesh files created by running MCNP in input-only mode do not contain edit groups

```
embed1 meshgeo=abacus
      mgeoin=candu.inp
      hdf5file=candu2.h5
      datawrite=yes
      background=6
      matcell=1 1 $ Coolant-4
              2 2 $ NU-1
              3 3 $ Air-2
              4 4 $ Clad-3
              5 5 $ CO2-5
```

*EMBED card for writing
unstructured_mesh/computed_data
group to candu2.h5*

HDF5-Based Unstructured Mesh Use Cases [1]

Use Case I: Reuse the Same UM Geometry

- Convert the Abaqus input file to HDF5 by setting **DATAWRITE=YES** and running MCNP in the input-only mode
- Reuse the resulting HDF5 mesh file in subsequent MCNP calculations that share the same mesh model but use different input settings, such as
 - Source definitions
 - Physics options
 - Material definitions
 - Variance-reduction techniques

```
embed1 meshgeo=hdf5
      mgeoin=candu2.h5
      hdf5file=candu2.eeout.h5
      background=6
      matcell=1 1 $ Coolant-4
              2 2 $ NU-1
              3 3 $ Air-2
              4 4 $ Clad-3
              5 5 $ CO2-5
```

EMBED card for using an HDF5 mesh input file. candu2.h5 is created with datawrite = yes; thus the unstructured_mesh/computed_data group is in the candu2.h5 file.

HDF5-Based Unstructured Mesh Use Cases [2]

Use Case II: Overcoming MPI Memory Limitations

- When a large Abaqus input file cannot be processed with MPI because of memory limits:
 - Convert the Abaqus file to HDF5 in sequential, input-only mode using **DATAWRITE=YES**
 - Use the resulting HDF5 mesh file for the parallel MCNP simulation

File Size and Input Processing Time Comparison [1]

Candu Model: Number of elements = 1,086,339; Number of parts = 5

Mesh Input File	Type	File Size	Memory Use (High Water Mark)	Memory Use	Input Processing Time (s)
candu.inp	Abaqus	149 MB	1145 MB	521 MB	84
candu1.h5	HDF5	17.4 MB	870 MB	515 MB	73
candu2.h5	HDF5	179.5 MB	515 MB	515 MB	2
candu.txt	MCNPUM (ASCII)	1.3 GB	515 MB	515 MB	7
candu.bin	MCNPUM (binary)	476.5 MB	515 MB	515 MB	2

candu1.h5 does not contain **unstructured_mesh/computed_data** group

candu2.h5 contains **unstructured_mesh/computed_data** group

File Size and Input Processing Time Comparison [2]

ICRP145, MRCP-AM-Internal: Number of elements = 8,233,413; Number of parts = 187

Mesh Input File	Type	File Size	Memory Use (High Water Mark)	Memory Use	Input Processing Time (s)
mrcp-am.inp	Abaqus	442 MB	16803 MB	3257 MB	1326
mrcp-am1.h5	HDF5	161.3 MB	13907 MB	3247 MB	1368
mrcp-am2.h5	HDF5	1728.5 MB	3247 MB	3247 MB	15
mrcp-am.txt	MCNPUM (ASCII)	8.2 GB	4156 MB	3247 MB	49
mrcp-am.bin	MCNPUM (binary)	3021.7 MB	4156 MB	3247 MB	12

mrcp-am1.h5 does not contain **unstructured_mesh/computed_data** group

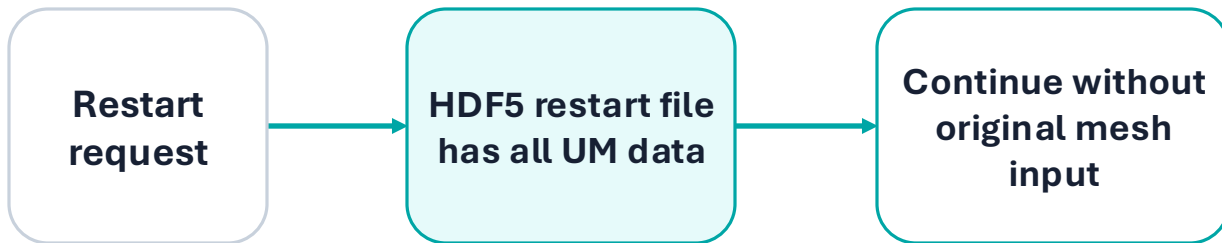
mrcp-am2.h5 contains **unstructured_mesh/computed_data** group

Restart data moves into HDF5

Current continue run



New continue run (under development)



Result: UM restart behavior aligns with traditional MCNP continue-run workflow while avoiding repeated input-processing overhead

Overview of Implementation

HDF5 processed mesh reuse

Write computed UM tracking data into the mesh HDF5 file;
Load **computed_data** group in subsequent calculations

Legacy format retirement

Remove dependence on the MCNPUM file while preserving the reuse behavior users rely on

Restart data capture

Serialize all UM state needed for continue runs into the HDF5 restart file

Continue-run load path

Restore UM state from restart data without requiring EMBED card or original mesh input

Benefits for MCNP Unstructured Mesh Users

- Less setup overhead:
 - Reuse processed mesh data across calculations that share the same mesh model
- Simpler file workflow:
 - No separate MCNPUM artifact to manage for processed mesh reuse
- Cleaner restarts:
 - Continue runs can proceed from the HDF5 restart file alone
- More familiar behavior
 - UM restart workflow better matches traditional MCNP continue-run expectations

Both enhancements remove redundant UM input processing and make HDF5 the consistent container for reusable mesh and restart data

Questions?

MCNP Forum: <https://mcnp.discourse.group>

Legal Notice

MCNP[®] and Monte Carlo N-Particle[®] are registered trademarks owned by Triad National Security, LLC, manager and operator of Los Alamos National Laboratory for the U.S. Department of Energy. 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. For the purposes of visual clarity, the registered trademark symbol is assumed for all references to MCNP within this report.