LA-UR-25-26170

Approved for public release; distribution is unlimited.

Title:	Development of an HDF5-based Unstructured Mesh Tracking Model in MCNP6
Author(s):	Armstrong, Jerawan Chudoung
Intended for:	2025 MCNP User Symposium, 2025-07-07/2025-07-10 (Los Alamos, New Mexico, UNITED STATES)
Issued:	2025-07-17 (rev.1)

Los Alamos NATIONAL LABORATORY

Development of an HDF5-based Unstructured Mesh Tracking Model in MCNP6

Jerawan Armstrong

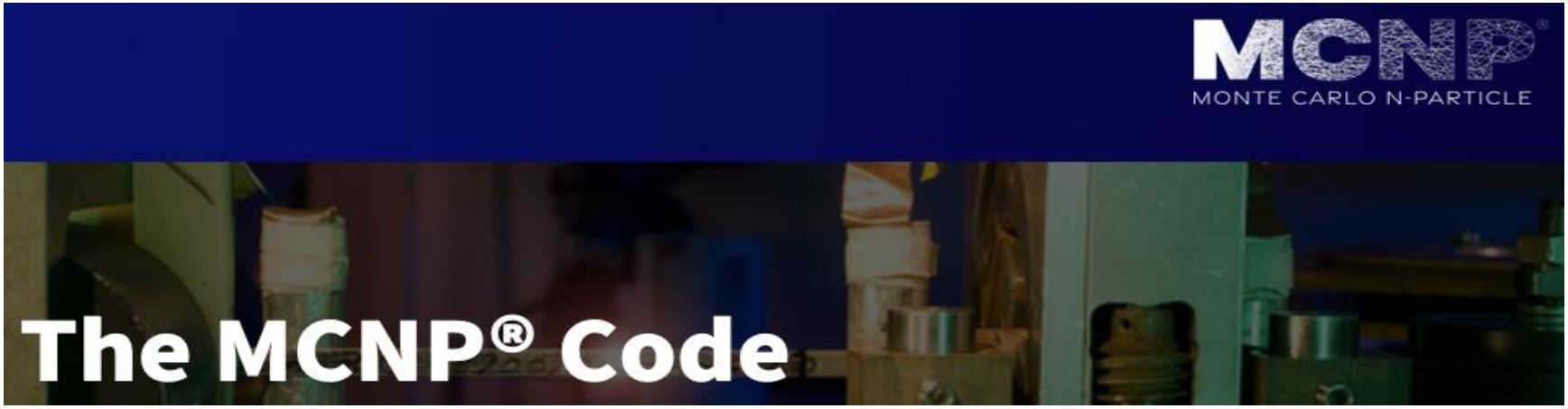
2025 MCNP User Symposium
July 7-10, 2025

LA-UR-25-26170

Outline

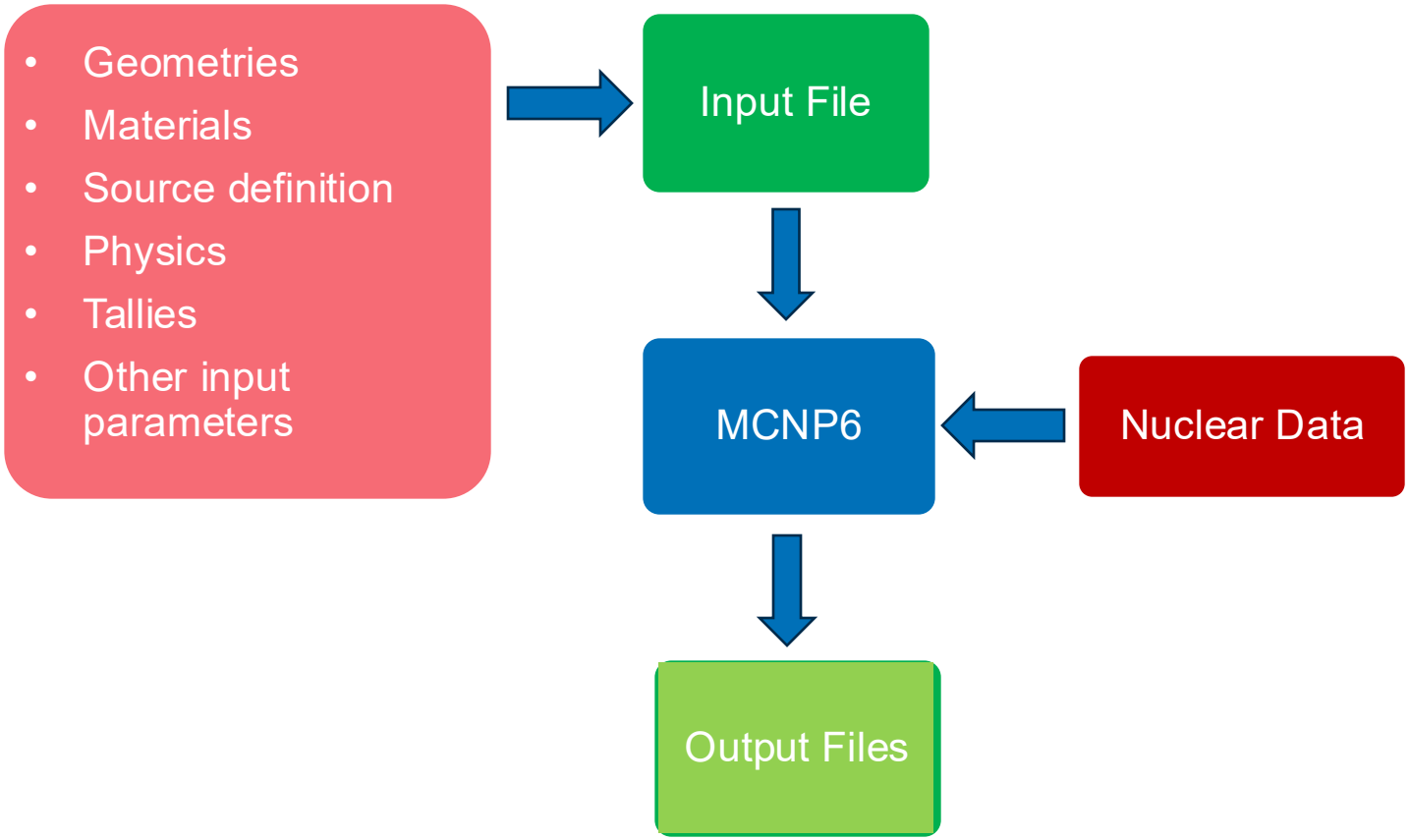
- Unstructured Mesh Input Processing
- HDF5 Mesh Input and Model Data for Particle Tracking

https://mcnp.lanl.gov



MCNP6 Calculation

Do not use MCNP6 as a black box.



An MCNP unstructured mesh (UM) calculation require an input file and external UM geometry file.

Constructive Solid Geometry (CSG) in MCNP

- The CSG approach in MCNP:
  - Describes the geometry of a problem by defining cells using surfaces and Boolean operations.
  - Might not accurately represent a physical system, as a geometry model must be adapted to fit within the limitations of the CSG modeling technique.

- The tools available for creating CSG models for MCNP simulations are limited.
  - The MCNP team does not maintain any tools for creating CSG models.

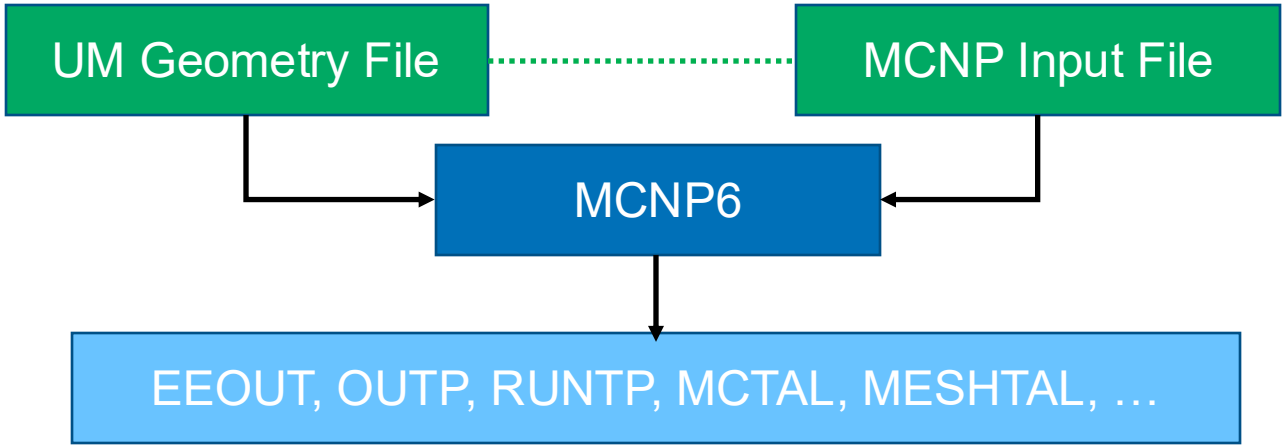
Creating an accurate CSG model for a complex physical system is very difficult.

Unstructured Mesh (UM) Geometry in MCNP6

- MCNP6 supports embedding an UM geometry model into a CSG cell.
  - UM geometry files are typically created from CAD files.

- Two main factors have contributed to the MCNP UM feature becoming widely used in many applications.
  - **Modern CAD and meshing tools** enable automated workflows for generating high fidelity 3D solid models and their corresponding UM representations, significantly reducing model preparation times.
  - MCNP6 produces **element-level quantities of interest** – such as flux and energy deposition, proving high-fidelity results written into an elemental edit output (EEOUT) file.

MCNP Unstructured Mesh (UM) Calculation



UM Geometry File Types:

- Abaqus Input: ASCII, MCNP6.0+
- MCNPUM: ASCII or binary, MCNP6.2–6.3, deprecated in 6.3
- HDF5 Input: binary, MCNP6.3+

EEOUT (Elemental Edit OUTput) File Types:

- Legacy Flat File: ASCII or binary, MCNP6.0 - 6.3, deprecated in 6.3
- HDF5 EEOUT: binary, MCNP6.3+
  - HDF5 EEOUT = HDF5 Input + Edit Results (i.e., element-level results)

Benefits of Using HDF5 Format in MCNP6

- **Smaller file size** for large output data - EEOUT, RUNTPE, FMESH, PTRAC.
- **Faster read/write performance** compared to text-based formats.
- **Works well with other tools** like Python (h5py) and ParaView.
- **Easy to extract data** for post-processing.
- **Robust format that works across different computer systems** – create files on one machine and use them on another without issues.

MCNP HDF5 UM input and EEOUT files can be visualized directly in ParaView and VisIt.

UM Geometry: Part and Assembly Concept

- The geometry of a model is defined by organizing into parts which are positioned relative to one another in an assembly.
- The concept of parts and assemblies is common in many CAD software packages since a physical model is typically created by assembling various components.
- The concept of model created by assembling various parts allows reuse of part definitions, which is valuable for creating a large complex model.

MCNP can only process Abaqus input files structured using parts and assemblies.

Abaqus Input File

- An Abaqus input file for an MCNP simulation must follow the correct Abaqus syntax and meet the additional requirements imposed by the MCNP code.

- Several codes (**excluding MCNP**) can import a CAD file, generate a UM representation of the model, and export it as an Abaqus input file.

- The Abaqus input file and the MCNP input file must be consistent with each other.

- Generating an MCNP input file from a large Abaqus input file can be tedious and time-consuming.
  - The Python code distributed with MCNP6.3 was developed to read an Abaqus input file, check its syntax, and generate a skeleton MCNP input file.

A common issue in MCNP UM simulations is poorly formatted or incorrect Abaqus input files.

Abaqus Input Processing in MCNP6

- An Abaqus input file must include part definitions, an assembly, and material definitions.
  - **Part**: name, nodes, elements, element sets (elset)
  - **Assembly**: instances
  - **Material**: name and density
- Instances in an assembly and element sets in parts are used to create **pseudo-cells**. Element and node definitions in parts are used to define pseudo-cell geometries.
- Material and pseudo-cell information must also be assigned in the MCNP input file.
- An **EMBED** card must be used to map pseudo-cells between the Abaqus and MCNP input files.
- MCNP6 reads information from an Abaqus input file and **creates the model data for particle tracking in UM geometry**.

Processing a large Abaqus input file in MCNP6 can be computationally expensive and memory extensive.

MCNPUM File

- The MCNPUM file type feature was developed in MCNP6.2.
  - Read an Abaqus input file.
  - Compute model data required for particle tracking.
  - Write model into a file for use in subsequent calculations.

- Advantage of the MCNPUM file type approach:
  - Significantly reduce input processing time and memory usage in subsequent calculations

- Limitations of the MCNPUM file type approach:
  - Generate large binary or flat files without a defined structure.
  - The code implementation is difficult to maintain.

The MCNPUM file type feature is deprecated in 6.3.

HDF5 Mesh Input File

- The HDF5-based mesh input file and output file feature was developed in MCNP6.3.
  - **HDF5FILE** keyword on EMBED card.

- The **HDF5FILE** keyword can be used to convert an Abaqus input file into an HDF5-based mesh input file, which can be used in the subsequent simulations where the UM geometry remains unchanged.
  - Run MCNP6 using the input-only command to generate an HDF5-based mesh input file from an Abaqus input file (i.e., **mcnp6 i i=test.i**).

- A size of HDF5-based mesh input file is smaller than that of the original Abaqus input file or the MCNPUM file converted from an Abaqus input file.

- MCNP6 must read and process the HDF5-based mesh input file, as it does not contain the model data required for particle tracking.
  - **The input processing time and memory usage of an HDF5-based mesh input file are greater than those of an MCNPUM file.**

Abaqus Input File Conversion

- The following procedure can be used to create both an HDF5-based mesh file and an MCNPUM file from an Abaqus input file
  - Use **HDF5FILE** and **MCNPUMFILE** keywords on EMBED card.
  - Run MCNP6 using the input-only command, i.e., **mcnp6 i inp=candu.i**

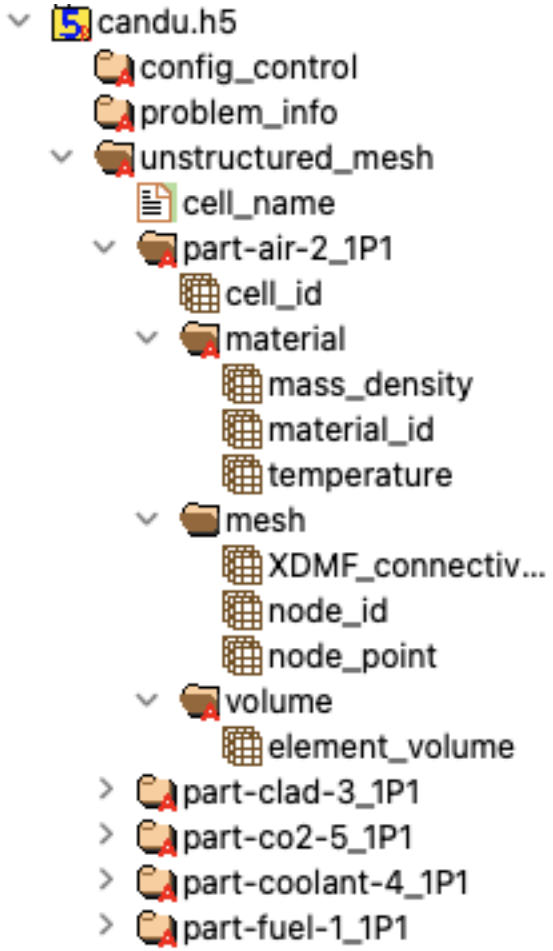
```
embed1 meshgeo=abaqus
       mgeoin=candu.inp
       mcnpumfile=candu.txt
       hdf5file=candu.h5
       background=6
       matcell=1 1  $ Coolant-4
                 2 2  $ NU-1
                 3 3  $ Air-2
                4 4  $ Clad-3
                5 5  $ CO2-5
```

```
embed1 meshgeo=abaqus
       mgeoin=candu.inp
       mcnpumfile=candu.bin
       hdf5file=candu.h5
       filetype=binary
       background=6
       matcell=1 1  $ Coolant-4
                 2 2  $ NU-1
                 3 3  $ Air-2
                4 4  $ Clad-3
                5 5  $ CO2-5
```

Examples of EMBED card for Abaqus input file conversion

MCNP HDF5 Mesh Input File

An MCNP HDF5 mesh input file includes an **unstructured_mesh** group that organizes data into cell subgroups.



Example of an MCNP HDF5 mesh input file

File Size and Input Processing Time Comparison

Number of elements = 1086339, Number of parts = 5

Mesh Input File	Type	File Size	Memory Use (High Water Mark)	Memory Use (Estimated)	Input Processing Time (s)
candu.inp	Abaqus	143 MB	1209 MB	147 MB	131
candu.h5	HDF5	21 MB	1009 MB	141 MB	126
candu.txt	MCNPUM (ASCII)	1.3 GB	592 MB	137 MB	9
candu.bin	MCNPUM (binary)	459 MB	592 MB	137 MB	1

1GB = 1024 MB

Common Use Cases for MCNPUM File Type

- **Use Case I: Multiple MCNP simulations using the same UM geometry**
  - Convert the Abaqus input file to a binary MCNPUM file.
  - Use the MCNPUM binary file as input for many simulations with different physics, materials, and/or variance reduction techniques.

- **Use Case II: Large Abaqus input file cannot be run with MPI due to memory limits**
  - Convert the Abaqus input file to a binary MCNPUM file by running MCNP via the input-only command in a sequential mode.
  - Use the binary MCNPUM file to run the simulation in a parallel mode.

A new feature with functionality similar to the MCPUM file type must be implemented before the existing MCNPUM file type feature is removed from the MCNP code.

HDF5-based UM Tracking Model

- The ”MCNPUM file type-like” feature has been implemented as a research code version.

- Two new keywords on the EMBED card:
  - **MODELWRITE**
  - **MODELREAD**

HDF5-based UM Tracking Model: MODELWRITE

- The **MODELWRITE** and **HDF5FILE** keywords must be used to convert an Abaqus input file to an HDF5 file that can be used as input for subsequent simulations.
  - MODELWRITE=YES for writing both **unstructured_mesh** and **model_data** groups.
  - MODELWRITE=NO for writing only **unstructured_mesh** group (default).

- **MODELWRITE=YES** is not allowed for a continue run.
  - The MODELWRITE keyword is used during the input processing, and it is only allowed in an initial run. This approach reduces code implementation complexity.

- **MODELWRITE=YES** must be used with **MESHGEO=ABAQUS**.
  - The MCNP code is currently the only tool that can generates an HDF5 mesh input file by translating an Abaqus input file. Therefore **MODELWRITE=YES** is only allowed with **MESHGO=ABAQUS**.

HDF5-based UM Tracking Model: MODELWRITE

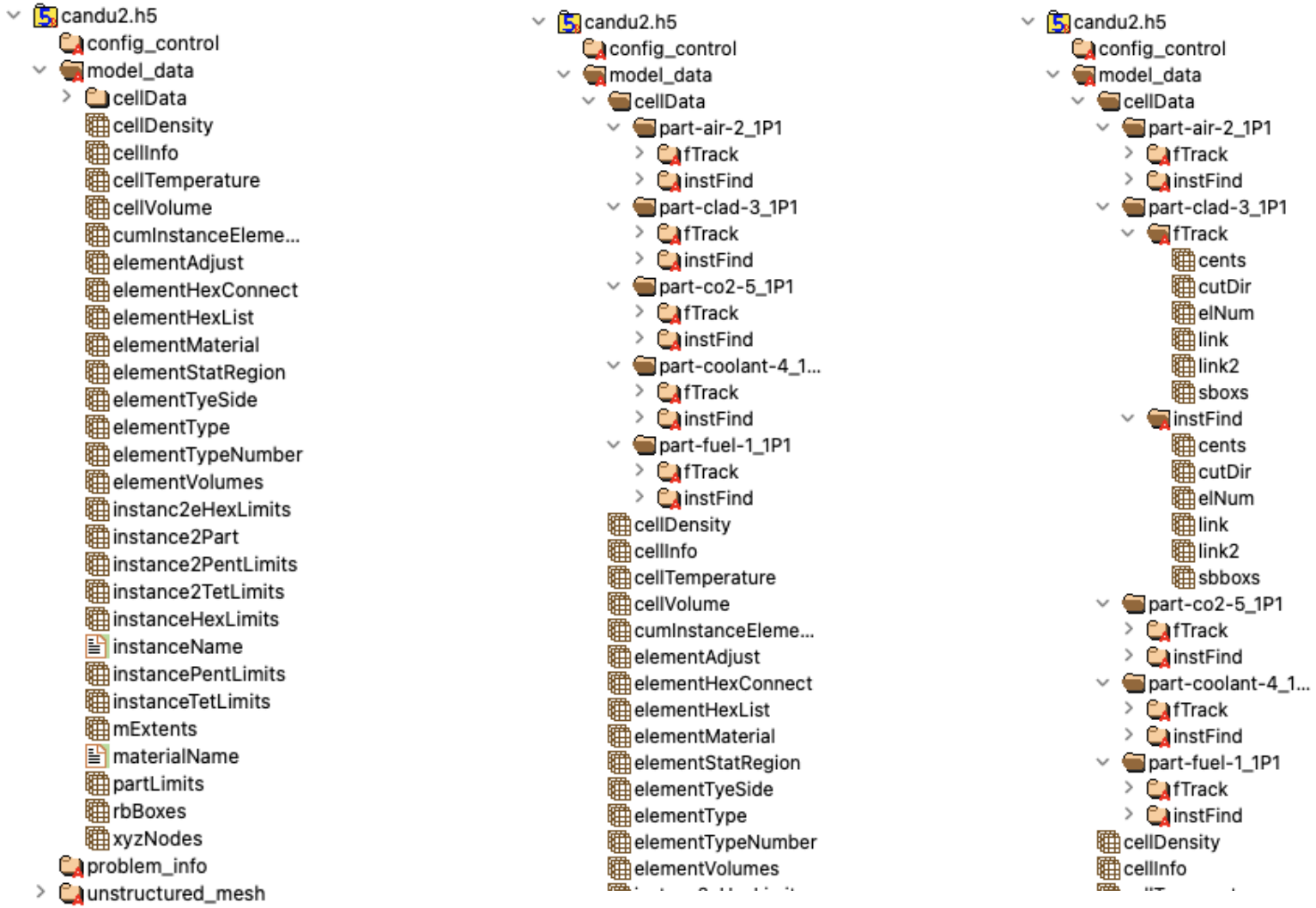
- The following procedure can be used to create an HDF5 input file from an Abaqus input file.
  - Use **HDF5FILE** and **MODELWRITE=YES** on the EMBED card.
  - Run MCNP6 using the input-only command, i.e., **mcnp6 i inp=candu2.i**

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

Example of EMBED card for generating an HDF5 file

MODELWRITE=YES is for writing both **unstructured_mesh** and **model_data** groups to an HDF5 file (candu2.h5).

HDF5-based UM Tracking Model: MODEL_DATA Group



Example of group structures in an HDF5 input file

HDF5-based UM Tracking Model: MODELREAD

- The **MODELREAD** and **HDF5FILE** keywords must be used to read an HDF5 file created from **MODELWRITE=YES**.
  - MODELREAD=YES for reading both **unstructured_mesh** and **model_data** groups.
  - MODELREAD=NO for reading only **unstructured_mesh** group (default).

- Both **MODELWRITE=YES** and **MODELREAD=YES** are not allowed in the same EMBED card.
  - Reduce code implementation complexity.

- If **MODELREAD=YES** and an HDF5 file contains only an **unstructured_mesh** group, then an error is thrown during the input checking step.

HDF5-based UM Tracking Model: MODELREAD

- Use both **HDF5FILE** and **MODELREAD=YES** to read an HDF5 UM input file for MCNP simulations.

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

Examples of EMBED card for reading an HDF5 file

- **MODELREAD=YES** is for reading both **unstructured_mesh** and **model_data** groups from an HDF5 file (candu2.h5).
- The EEOUT file (candu2.eeout.h5) contains only **unstructured_mesh** group.
- If **MODELREAD=YES**, **MESHGEO** is still required but its value is ignored since the mesh file type will be read from the **model_data** group.

File Size and Input Processing Time Comparison

Number of elements = 1086339, Number of parts = 5

Mesh Input File	Type	File Size	Memory Use (High Water Mark)	Memory Use (Estimated)	Input Processing Time (s)
candu.inp	Abaqus	143 MB	1209 MB	147 MB	131
candu.h5	HDF5	21 MB	1009 MB	141 MB	126
candu.txt	MCNPUM (ASCII)	1.3 GB	592 MB	137 MB	9
candu.bin	MCNPUM (binary)	459 MB	592 MB	137 MB	1
candu2.h5	HDF5	225 MB	597 MB	152 MB	1

candu.h5 contains **unstructured_mesh** group
candu2.h5 contains **unstructured_mesh** and **model_data** groups

Conclusions

- The MODELWRITE=YES/MODELREAD=YES keywords are introduced to write/read model_data group to/from an HDF5 mesh input file.

- The testing results show that:
  - The input processing time is comparable between the new HDF5 write/read feature and the existing MCNPUM read/write feature.
  - The file size of the HDF5 file is smaller than that of the binary MCNPUM file.
  - The memory usage of the HDF5 mesh input is slightly larger than that of the MCNPUM input.
  - The code implementation of the new HDF5 write/read feature is significantly simpler than that of the MCNPUM write/read feature.

- Current work:
  - Make code changes to reduce the memory usage of the MODELWRITE/MODELREAD feature.

- Future work:
  - The MODELWRITE/MODELREAD feature will undergo testing and review before being merged into the MCNP repository.

As the developer of the new MODELWRITE/MODELREAD feature, I'm actively seeking inputs from users familiar with the MCNPUM file type. If you have test problems or feedback, please consider sharing them. Your contributions will help me test and refine this feature before its official release.

Questions?

References

- J. A. Kulesza et al., *MCNP6.3 Code Version 6.3 Theory & User Manual*, LA-UR-22-30006 Rev. 1 (LANL, September 2022).
- J. C. Armstrong et al., *MCNP Unstructured Mesh Overview, Improvement, and Verification & Validation Testing*, LA-UR-21-26436 (LANL, July 2021).
- J. C. Armstrong & K. C. Kelley, *Creating Unstructured Mesh Models for MCNP Simulations*, LA-UR-22-30661 (LANL, October 2022).
- J. C. Armstrong & K. C. Kelley, *Generating MCNP Input Files for Unstructured Mesh Geometries*, LA-UR-23-28515 (LANL, July 2023).
- J. A. Kulesza et al., *MCNP Unstructured Mesh Visualization & Post-processing Techniques*, LA-UR-21-26365 (LANL, July 2021).
- E. Gonzalez, *MCNP6 Candu Input Files*, LA-UR-24-31558 (LANL, October 2024).

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.