LA-UR-21-26436
Approved for public release; distribution is unlimited.

Title: MCNP Unstructured Mesh Overview, Improvement, and Verification & Validation Testing

Author(s): Armstrong, Jerawan Chudoung
Kulesza, Joel A.
Josey, Colin James
Mosher, Scott William
Swaminarayan, Sriram
Kelley, Karen Corzine
Alwin, Jennifer Louise
Spencer, Joshua Bradly
Mehta, Vedant Kiritkumar

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

Issued: 2021-07-07

MCNP Unstructured Mesh Overview, Improvement, and Verification & Validation Testing

Jerawan Armstrong, Joel Kulesza, Colin Josey, Scott Mosher, Sriram Swaminarayan, Karen Kelley, Jen Alwin, Josh Spencer, Vedant Mehta

2021 MCNP User Symposium, July 12-16, 2021

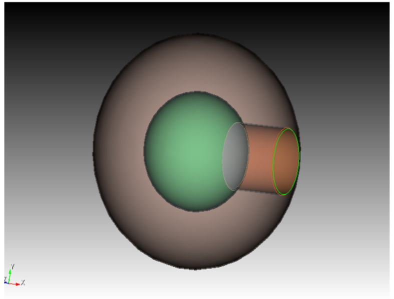
Outline

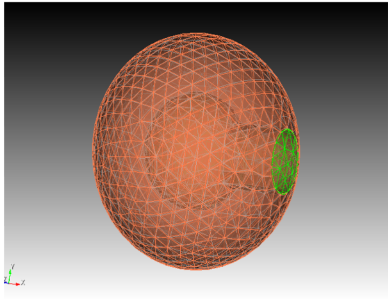
- MCNP Unstructured Mesh Overview
- Code Improvement from 6.2 to 6.3 versions
- Verification & Validation Testing
- MCNP UM Limitations
- Future Work

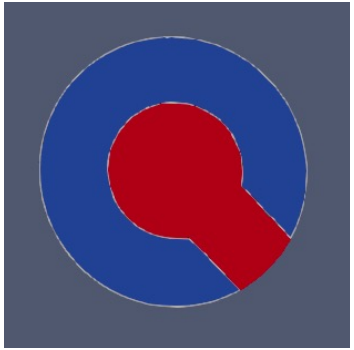
MCNP Geometry Models

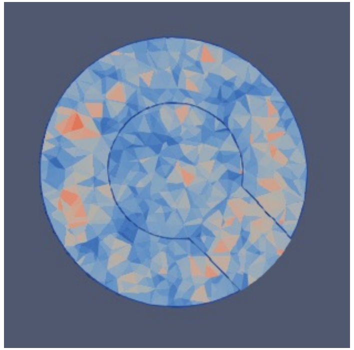
Geometry setup is a crucial step of MCNP simulations!

- **Constructive Solid Geometry (CSG) Model:**
  - Constructed by organizing an arbitrary 3D configuration of materials into geometric cells bounded by surfaces.
- **Hybrid Geometry Model:**
  - Constructed by embedding **finite element meshes (structured or unstructured meshes)** into CSG cells.
  - Finite element meshes are typically generated by host codes or meshing software packages.

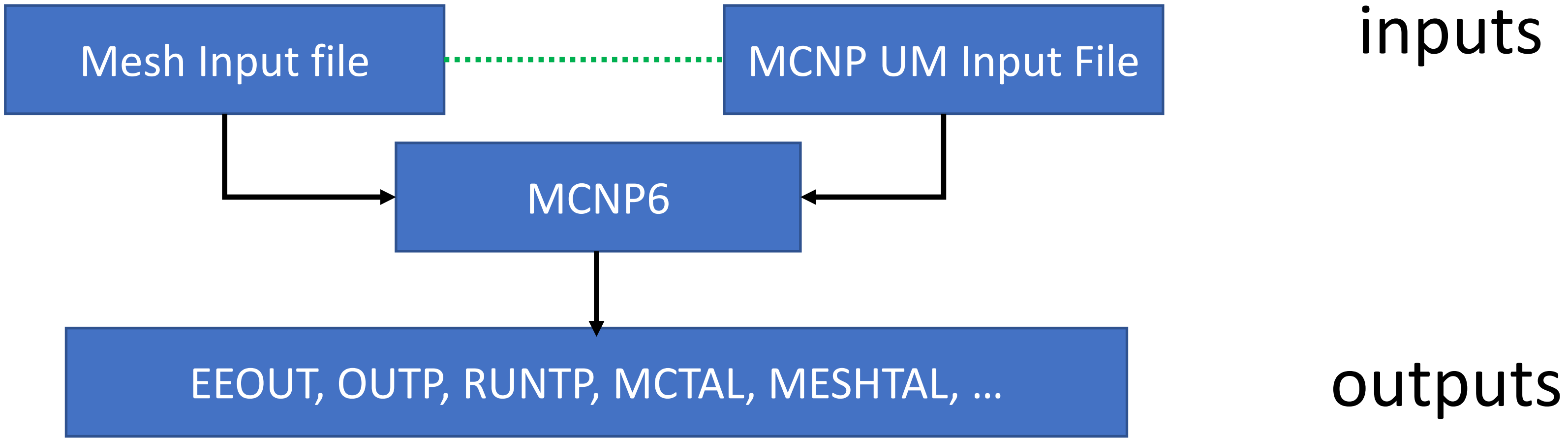








MCNP Unstructured Mesh (UM) Calculations



Mesh Input File Format:

- Abaqus Input [6.0 - 6.3 versions]
- HDF5 [6.3 version]

EEOUT (Element Edit OUTput) File Format:

- *Flat* ASCII or Binary [6.0 - 6.3 versions]
- HDF5 [6.3 version]

HDF5 EEOUT = HDF5 Input + Edit Results

MCNP UM Preprocessing & Postprocessing

- MCNP UM Preprocessing Tools:
  - um_pre_op
  - Python scripts
  - Commercial software
- MCNP UM Postprocessing Tools:
  - um_post_op
  - Python scripts
  - Abaqus scripts [python & C++]
  - Commercial software

- An MCNP UM calculation requires two input file types:
  - MCNP input file, &
  - Mesh input file [Abaqus or HDF5]
- An Abaqus input file must have the correct Abaqus syntax rules and meet additional MCNP requirements.
- Any “code” that can export an Abaqus formatted input file may be used to create UM models for MCNP simulations.

Abaqus UM Input Model

Element type: C3D4, C3D6, C3D8, C3D10, C3D15, C3D20, **SC8**

SC8 is in 6.3 version

part

```
*Part, name=Part-big_block
*Node
+-- 90 lines: 1,          10.,          10.,          90.-
*Element, type=C3D8R
+-- 36 lines: 1, 31, 32, 35, 34,  1,  2,  5,  4----------
*Nset, nset=Set-material_02, generate
  1,  90,   1
*Elset, elset=Set-material_02, generate
  1,  36,   1
*Nset, nset=Set-statistic_02, generate
  1,  90,   1
*Elset, elset=Set-statistic_02, generate
  1,  36,   1
*End Part
```

node

```
*Part, name=Part-source_hex
*Node
+-- 12 lines: 1,          10.,          0,          10.-
*Element, type=C3D8R
+--  2 lines: 1,  5,  6,  8,  7,  1,  2,  4,  3----------
*Nset, nset=Set-materilal_01, generate
  1,  12,   1
*Elset, elset=Set-material_01
 1, 2
*Nset, nset=Set-statistic_01, generate
  1,  12,   1
*Elset, elset=Set-statistic_01
 1, 2
*Nset, nset=Set-source_01, generate
  1,  12,   1
*Elset, elset=Set-source_01
 1, 2
*End Part
```

```
*Part, name=Part-source_pent
*Node
+-- 8 lines: 1,          0,          10.,          10.-
*Element, type=C3D6
+-- 2 lines: 1, 4, 2, 1, 8, 6, 5-------------------------
*Nset, nset=Set-material_01, generate
 1,  8,  1
*Elset, elset=Set-material_01
 1, 2
*Nset, nset=Set-statistic_01, generate
 1,  8,  1
*Elset, elset=Set-statistic_01
 1, 2
*Nset, nset=Set-source_01, generate
 1,  8,  1
*Elset, elset=Set-source_01
 1, 2
*End Part
```

instance

```
*Part, name=Part-source_tet
*Node
+-- 9 lines: 1,          10.,          10.,          10.-
*Element, type=C3D4
+-- 12 lines: 1, 2, 8, 9, 5-----------------------------
*Nset, nset=Set-material_01, generate
 1,  9,  1
*Elset, elset=Set-material_01, generate
  1,  12,   1
*Nset, nset=Set-statistic_01, generate
 1,  9,  1
*Elset, elset=Set-statistic_01, generate
  1,  12,   1
*Nset, nset=Set-source_01, generate
 1,  9,  1
*Elset, elset=Set-source_01, generate
  1,  12,   1
*End Part
```

elset

```
** ASSEMBLY
**
*Assembly, name=Assembly
**
*Instance, name=Part-big_block-1, part=Part-big_block
       0.,          0.,          10.
*End Instance
**
*Instance, name=Part-source_hex-1, part=Part-source_hex
*End Instance
**
*Instance, name=Part-source_tet-1, part=Part-source_tet
*End Instance
**
*Instance, name=Part-source_pent-1, part=Part-source_pent
*End Instance
**
*End Assembly
**
** MATERIALS
**
*Material, name=Material-main_material_02
*Density
 0.00014,
*Material, name=Material-source_material_01
*Density
 0.5,
```

material

elset format/name and material name must meet MCNP requirements

MCNP UM Calculations Using Abaqus

SIMULIA

ABAQUS UNIFIED FEA

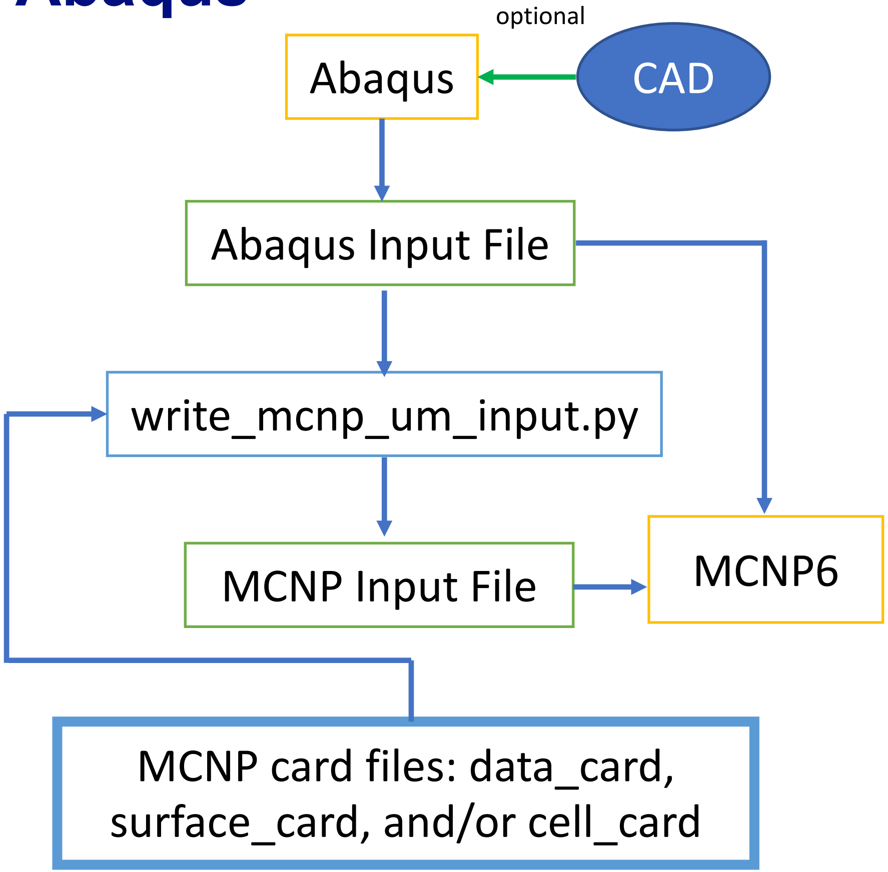
COMPLETE SOLUTIONS FOR REALISTIC SIMULATION

https://www.3ds.com

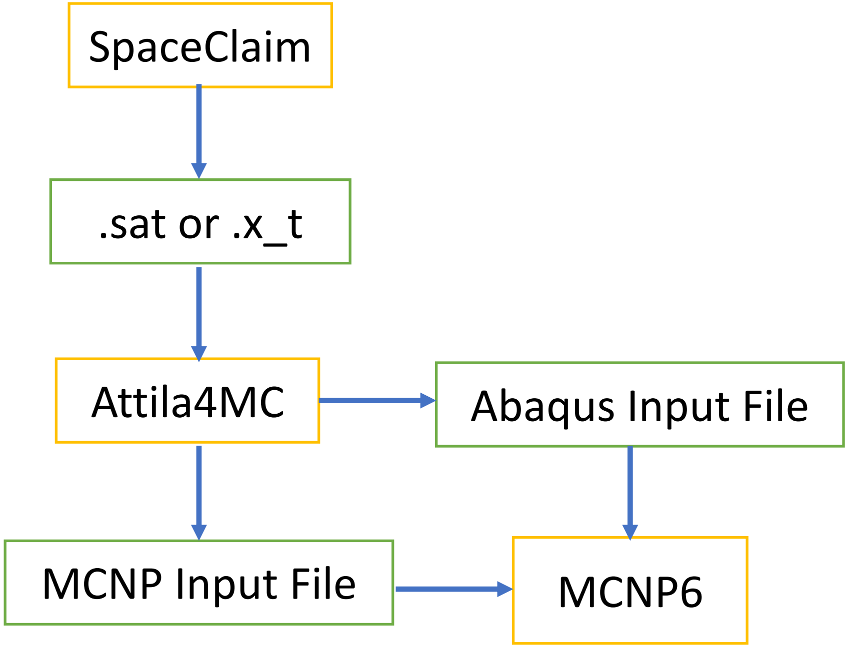
Abaqus Element Types that MCNP 6.0 -6.2 versions can process:

- 1^{st} order tet, pent, hex elements
- 2^{nd} order tet, pent, hex elements

- **write_mcnp_um_input.py**
  - performs extensively error checking on an Abaqus input file format.
- **um_pre_op:**
  - developed to write an MCNP skeleton input file.



MCNP UM Calculations Using Attila4MC



MCNP UM Input Setup Using Attila4MC

https://silverfirsoftware.com

Only 1st order tet elements

"Attila4MC provides an easy-to-use graphical interface, allowing novice and advanced MCNP users to easily set up, run, and visualize MCNP solutions from CAD data."

silverfirsoftware.com

MCNP UM Calculations Using CUBIT

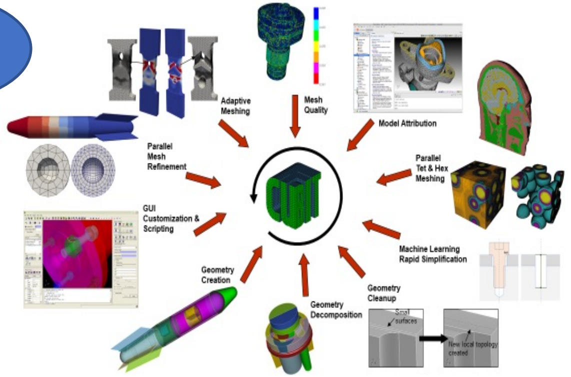
CUBIT is the Sandia National Laboratory automated mesh generation toolkit

CAD → (optional) Cubit → Abaqus Input File → cubit_to_mcnp.py → Abaqus Input File for MCNP → write_mcnp_um_input.py → MCNP Input File → MCNP6

Volume-Material Filles → cubit_to_mcnp.py

MCNP card files: data_card, surface_card, and/or cell_card → write_mcnp_um_input.py

Abaqus Input File for MCNP → MCNP6



https://cubit.sandia.gov

Version 15.6: 1^{st} order tet & hex and 2^{nd} order tet elements.

MCNP UM Input Setup Using Cubit

MCNP CSG & UM Input files

```
Leakage from Al (40cm dia) sphere 3-d surface tally
c
1  1    -0.001288  (  -3 -8):(8   -1 -6)
2  3       -7.824  (3 -4 -8):(8 1 -2 -6)
3  2       -1.223  (4 -5 -8):(8 2    -5)
4  3       -7.824  (5 -6 -8):(8 2  5 -6)
5  0                 6 -7
6  0                 7

1  cx    5.55
2  cx    5.75
3  so   10.0
4  so   10.2
5  so   19.75
6  so   19.95
7  so   25.0
8  px    8.32

mode   n p
imp:n 1 4r 0
imp:p 1 4r 0
sdef   pos=0 0 0  cel=1  erg=d1
nps    1e8
prdmp  2j 1 2 5e6
```

```
Oktavian Al - UM Model
c
1        1     -0.001288      0  u=1  $ material_01
2        3     -7.824000      0  u=1  $ material_03
3        2     -1.223000      0  u=1  $ material_02
4        3     -7.824000      0  u=1  $ material_03
5        0                    0  u=1  $ background
6        0                 -999  fill=1 $ fill cell
7        0                  999

999 sph  0  0  0 25.0

mode    n p
imp:n 1 5r 0
imp:p 1 5r 0
sdef  erg=d1 pos=0.0001 0.0001 0.0001
nps    1e8
prdmp  2j 1 2 5e6
embed1 meshgeo=abaqus
       mgeoin=oktaviansphere.abaqus.inp
       meeout=oktaviansphere.eeout
       length=1.000000E+00
       background=5
       matcell=1 1  $ material_01
               2 2  $ material_03
               3 3  $ material_02
               4 4  $ material_03
c   Composition 1
c     Al 0.997 Si 0.0015
```

meeout=
elemental edit output file

mgeoin=
mesh input file

```
******************************** P A R T S ********************************
*PART, NAME=Part-Default
**
******************************** N O D E S ********************************
*NODE, NSET=ALLNODES
+--50117 lines: 1,    1.916246e+01,    6.796790e-16,    5.550000e+00----------
**
******************************** E L E M E N T S ****************************
*ELEMENT, TYPE=C3D4
+--154500 lines: 1,   24525,   38348,   38270,   38483-----------------------
**
******************************** P R O P E R T I E S ***********************
** Cell 1: element number 84400-86393,  material 1
** Cell 2: element number 86394-143938, material 3
** Cell 3: element number 143929-15400, material 2
** Cell 4: element number 1-84399,      material 3
*ELSET, elset=Set-statistic-04, generate
 1,  84399, 1
*ELSET, elset=Set-material-03, generate
 1,  84399, 1
*ELSET, elset=Set-material-statistic-01, generate
 84400,  86393, 1
*ELSET, elset=Set-statistic-02, generate
 86394, 143928, 1
*Elset, elset=Set-material-03, generate
 86394, 143928, 1
*ELSET, elset = Set-statistic-03, generate
 143929, 154500, 1
*Elset, elset=Set-material-02, generate
 143929, 154500, 1
**
*END PART
**
```

CSG Input File

1 input file type

UM Input Files

2 input file types

Three Options to Produce EEOUT Files: Option I

- **Option 1: create an ASCII EEOUT file**

```
embed2 meshgeo=abaqus
       mgeoin=example.inp
       meeout=example.eeout
       background= 20
       matcell=  1 10  2 11  3 12  4 13
```

Produce an ASCII EEOUT file:

example.eeout

- Letters in file names on EMBED card must be lowercase letters.
- MCNP overwrites an old ASCII EEOUT file.

Three Options to Produce EEOUT Files: Option II

- **Option 2: create a binary HDF5 EEOUT file**

New feature in MCNP 6.3 version

```
embed2 meshgeo=abaqus
      mgeoin=example.inp
      hdf5file=example.eeout.h5
      background= 20
      matcell=  1 10  2 11  3 12  4 13
```

Produce two output files:
- example.eeout.h5 (binary HDF5 file)
- example.eeout.h5.xdmf (ASCII XML file)

- Letters in file names on EMBED card must be lowercase letters.
- MCNP overwrites an old HDF5 EEOUT file [& XML file].

Three Options to Produce EEOUT Files: Option III

- **Option 3: create an ASCII EEOUT file & a binary HDF5 EEOUT file**

New feature in MCNP 6.3 version

This option is not recommended for a large calculation.

```
embed2 meshgeo=abaqus
    mgeoin=example.inp
    meeout=example.eeout
    hdf5file=example.eeout.h5
    background= 20
    matcell=  1 10  2 11  3 12  4 13
```

Produce three output files:

- **example.eeout** (ASCII EEOUT file)
- example.eeout.h5 (binary HDF5 file)
- example.eeout.h5.xdmf (ASCII XML file)

- Letters in file names on EMBED card must be lowercase letters.
- MCNP overwrites the old ASCII & HDF5 EEOUT files [& XML file].

MCNP UM Postprocessing & Visualization

- 2021 MCNP User Symposium [July 14, 10:55-11:15]:

MCNP Unstructured Mesh Visualization & Post-processing Techniques

Joel A. Kulesza[1], Jerawan C. Armstrong[1], Colin J. Josey[1], Sriram Swaminarayan[2], Jennifer L. Alwin[3], Tucker C. McClanahan[3], Joshua B. Spencer[3], John T. Goorley[3], Karen C. Kelley[4], Prabhu S. Khalsa[4], and Gregory A. Failla[5]

[1]*Los Alamos National Laboratory, Monte Carlo Codes Group, Los Alamos, NM*
[2]*Los Alamos National Laboratory, Applied Computer Science Group, Los Alamos, NM*
[3]*Los Alamos National Laboratory, Radiation Transport Applications Group, Los Alamos, NM*
[4]*Los Alamos National Laboratory, Advanced Engineering Analysis Group, Los Alamos, NM*
[5]*Silver Fir Software, Gig Harbor, WA*
Corresponding Author Email: jkulesza@lanl.gov

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.

MCNP UM New Features in 6.3 version

- Mesh Quality Metric Tables
  - "MCNP UM Elemental Quality Assessment" Presentation by Joel Kulesza
- New Element Type: Abaqus SC8
  - pure SC8 in a part
  - mixed SC8 and C3D8 in a part
  - using same tracking algorithm for SC8 and C3D8 elements
- Convert an Abaqus input file to an HDF5 mesh input file
  - run MCNP input option [mcnp6 i inp or mpirun –np <n> mcnp6.mpi i inp] using "hdf5file" on EMBED card to create an HDF5 mesh input file

```
embed2 meshgeo=abaqus
        mgeoin=example.inp
        hdf5file=example.h5
        background= 20
        matcell=  1 10  2 11  3 12  4 13
```

- HDF5 Mesh Input File
- HDF5 EEOUT File
  - restart using HDF5 EEOUT file

```
embed2 meshgeo=hdf5
        mgeoin=example.h5
        hdf5file=example.eeout.h5
       background= 20
       matcell=  1 10  2 11  3 12  4 13
```

HDF5 UM Input/Output

- Why is an HDF5 file chosen?
  - Becomes I/O library of choice for NNSA Labs.
  - Designed to manage large complex data collections.
  - Portable among different computing flatforms.
  - Easy to view, edit, and analyze using public available software tools or Python scripts.

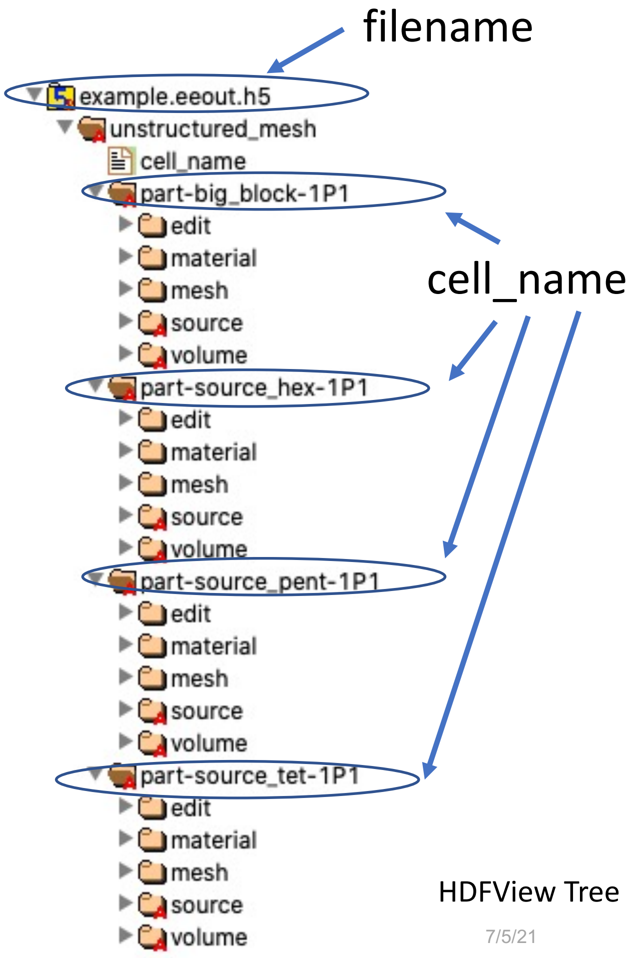
- An HDF5 file is a container for an organized collection of objects where each object must have a unique identity within an HDF5 file and can be accessed only by it name within the hierarchy of the file.
  - HDF5 objects: **attribute, dataset, group**
  - HDF5 link: **unstructured_mesh/cell_name**
- See MCNP 6.3 User Manual (Chapter 8.10) for an HDF5 file format used to store an unstructured mesh model and element edit outputs.

<filename>.h5: /unstructured_mesh

Name	HDF5 Object	Data Type
model_description	attribute	string
total_cells	attribute	integer
total_elements	attribute	integer
total_parts	attribute	integer
total_part_elements	attribute	integer
cell_name	dataset	1D string array
<unique_cell_name>	group	Attributes & groups

groups in each <unique_cell_name>:

- material, mesh, volume, source, edit
- source is an optional group
- edit is a group in an output file



HDFView Tree

Easy to process HDF5 UM Input/Output File using Python

```
import h5py

def printall(name, obj):
    print(name, dict(obj.attrs))

name = "nestedcylinder_electrontestv3.eeout.h5"
with h5py.File(name,'r') as hf: hf.visititems(printall)
```

```
import numpy as np
filename = "nestedcylinder_electrontestv3.eeout.h5"
cell_data, cell_name = get_mesh_data_HDF5(filename)
for cname in cell_name:
    volume = cell_data.get(cname)[1]
    volume = np.array(volume)
    indx   = np.where(volume <= 1.E-6)[0]
    print("{:>s}".format(cname))
    for i in indx:
        print(" {:20d} {:20.5e}".format(i, volume[i]))
```

```
import h5py

def get_mesh_data_HDF5(eeout_filename):
    f = h5py.File(eeout_filename,'r')

    um = '/unstructured_mesh'
    cell_label = list(f[um+'/cell_name'])

    path_name = []
    cell_name = []
    for c in cell_label:
        name = c.decode('utf-8').strip()
        path_name.append(um+'/'+ name)
        cell_name.append(name)

    cell_data = {}
    for pname, cname in zip(path_name, cell_name):
        k    = pname + '/material/mass_density'
        density = list(f[k])

        k = pname +'/volume/element_volume'
        volume = list(f[k])

        cell_data[cname] = [density, volume]

    f.close()
    return  cell_data, cell_name
```

Some MCNP UM Code Enhancements in 6.3 Version

- Improved UM Abaqus Input preprocessing
  - reduce memory, faster, & more robust
- Fixed poor code performance
  - significantly faster for large calculations
- Fixed codes so that all UM regression/feature testing problems can be run with the executable build with more restrictive flags :
  - Fortran_FLAGS="-check all, noshape,noarg_temp_created"
- Fixed neutral particle tracking bugs:
  - "collision in void in colidn routine"
  - "photon transport with all-zero photoatomic cross section"
- Other UM fixed bugs and code enhancements will be listed in the release notes of MCNP 6.3 version.

do not build MCNP with this option for production calculations

MCNP UM Code Verification & Validation

- **Motivation**:
  - MCNP code V&V gives users confidence in its calculated results. Several MCNP V&V suites are distributed with MCNP code release.
  - Despite the MCNP UM feature being increasingly used for new applications, there is no UM V&V suite distributed with the code.
- **Verification**:
  - converted Oktavian testing problems in MCNP CSG VERIFICATION_SHLD_SVDM Suite into UM models and verify MCNP CSG & UM results.
- **Validation**:
  - converted Godiva sphere into 4 models [1st order tet & hex; 2nd order tet & hex] and validate the calculated keff values with experimental value.

MCNP UM Godiva & Oktavian test problems will be released with MCNP 6.3 version (15 MCNP and Abaqus input files).

MCNP UM Limitations

Currently, UM Capability is not fully integrated with all of the pre-existing MCNP features. See MCNP 6.3 User Manual for other UM Limitations.

- Limited testing on the following features:
  - Non-void background cell
  - PTRAC
  - SDEF options
  - Average & Entry Overlap model
  - 2nd order tet, pent, hex elements
  - Neutron/Photon/Electron, Photon/Electron, and charged particle transport calculations
- Known issues:
  - Incorrect results for mixed void and non-void pseudo cells
  - Negative energy depositions for electrons
- Should not use for magnetic fields
- No code implementation for forced collision on UM
- No testing on UM Utilities Program
- Surface tallies are not permitted in the background cell and pseudo cells

Future Work

- Refactor codes to reduce memory & speed up the calculations [continuing work]
  - Replace inefficient data structure
  - Replace inefficient tracking algorithms
  - Replace algorithms used to calculate edits
- HDF5 parallel reading/writing input/output files
- Improvement for Photon/Electron transport calculations
- Remove UM Utilities from MCNP code [Fortran Code]
  - Develop Python scripts to replace UM Utilities
  - Remove MCNPUM format from MCNP code
- MCNP UM V&V

Questions?