LA-UR-16-23286

Approved for public release; distribution is unlimited.

Title:	The MCNPUM Capability for MCNP6's Unstructured Mesh Feature
Author(s):	Roger L. Martz
Intended For:	General Reference / MCNP® Website
Issued:	April 2016

![Los Alamos National Laboratory logo — EST.1943]

This page intentionally left blank.

The MCNPUM Capability for MCNP6's Unstructured Mesh Feature

Roger Martz
Monte Carlo Methods, Codes, and Applications
X-Computational Physics Division
Los Alamos National Laboratory

April 2016

1   Introduction

This report documents a new file type that has recently been added to MCNP6 [1] for use with the unstructured mesh (UM) feature. This capability allows the user to request that the code convert an Abaqus [2] inp file to a new format called the mcnpum format; MCNP6 can also read this new UM file format. Changes to the `EMBED` card are required to accommodate this capability. This report documents the verification of this capability.

2   Discussion

The Abaqus input file contains some basic information about the unstructured mesh, but does not contain everything that MCNP6 needs. Once MCNP6 reads this file, it uses the Abaqus data to generate other information that it needs in its tracking routines such as nearest neighbor lists. That is, it converts the Abaqus information to its internal data structures. Even with the parallel input processing, Reference [3], significant computer time can be required to regenerate these internal data structures for every MCNP calculation that uses the Abaqus inp unstructured mesh file. This is extremely burdensome when large UM files are needed in repeated calculations.

The mcnpum file type was created to contain all of the unstructured mesh data structures that MCNP needs, thus eliminating the need to "input process" the Abaqus input file every time the code is run, including continue runs. MCNP can generate this file (primarily after processing the Abaqus input file) by simply including the `mcnpumfile` option on the embed card. MCNP can use the mcnpum file when the `mcnpum` keyword is supplied to the `MESHGEO` parameter on the `EMBED` card. See the input cards section in Reference [3].

Details on the structure of the mcnpum file type are beyond the scope of the document. The truly interested reader should look at the MCNP source code. It is possible that the structure of this file may change over time.

3   Verification

Verification of this capability is in several parts and includes initial runs, continue runs, creating the mcnpum file, volume sources, and mixing use with the traditional Abaqus inp file in a multi-mesh run. More details are provided in the verification sections that follow.

3.1 Verification With 1001 Problem Geometry

Regression problem 1001 from MCNP6's REGRESSION test suite contains 6 parts (pseudo-cells) with each part constructed from one of the six allowable finite element types. See Reference [3] for more on these element types. It is crucial that these six element types are tested since internal to MCNP6 their information is contained in different data structures and consequently represent different (non-trivial) sections of both the mcnpum and eeout files.

Problem 1001 is a fixed source ray-tracing problem. The source particles are started from a spherical surface that surrounds all three universes and are directed inward. The `VOID` card is used to void out all materials in the problem. Consequently, the pedigree of the cross sections is not important. When the `WGT` parameter on the `SDEF` card is set to a value corresponding to the surface area of the sphere, this problem becomes one of stochastically estimating the volumes of each cell or UM element as discussed in Reference [4]'s section on Stochastic Volume and Area Calculations.

Problem 1001 is a neutron only problem with two edits. The flux edit produces meaningful flux edits given the small number of histories (10,000) run while the energy deposition edit produces all zeros since the materials have been voided in each cell and no energy deposition takes place.

3.1.1 Three Part Verification

Verification with the 1001 problem geometry is done in three parts, demonstrating three important scenarios required by MCNP6. First, problem 1043 generates a mcnpum version of the problem 1001 Abaqus inp file, demonstrating the ability to correctly process the mcnpumfile input parameter and generate a mcnpum file. Next, problem 1044 uses the mcnpum file generated with problem 1043 and performs an initial calculation with it while running a total of 10,000 histories. This demonstrates the code's ability to read the new file type correctly and successfully perform the calculation. Lastly, problem 1045 is a continue run of problem 1044 for a total of 20,000 histories and demonstrates the code's ability to successfully use the new file type under this set of circumstances. The MCNP6 inputs for these three REGRESSION suite test problems are given in the sections that follow.

Since there are a total of 331 elements in this UM geometry, it is tedious to provide in this document results for each element. In the process of this verification, two sets of comparisons were made between the eeout and fmesh output files produced by the code for each of the runs. Running a Linux diff utility between appropriate files for the initial (problems 1001 and 1044) and continue (problems 1002 and 1045) runs showed that the results were identical in all cases.

3.1.2 Problem 1043 Input File

```
block01: 8x8x6; 6 parts, 6 element types
c
c  -------------------------------
c  - MCNP / REGRESSION Test Suite  -
c  - Unstructured Mesh Problem #43 -
c  - Roger Martz                   -
c  - 10/13/2015                    -
c  -------------------------------
c
c  Creating mcnpum file type with MCNP; test with 1044 & 1045.
c
10  1   -2.03   0        u=2
11  1   -2.03   0        u=2
12  1   -2.03   0        u=2
13  1   -2.03   0        u=2
14  1   -2.03   0        u=2
15  1   -2.03   0        u=2
21  0           0        u=2 $ background
```

```
30  0           -99     fill=2 $ fill cell
40  0            99

99  sph   0.  0. 3. 10.

c  the following is pseudo-concrete m1    1001 -0.02  8016 -0.60  14000 -0.38
c
sdef   pos= 0 0 3    erg=2   nrm= -1
       rad=10          sur=99  wgt= 314.159265
c
mode n
imp:n  1 7r  0
c
embed2 meshgeo=abaqus
       mgeoin=um1001.inp
       meeout=inp1043u
       mcnpumfile=um1044.inp
       filetype=ascii
       background= 21
       matcell= 1 10  2 11  3 12  4 13  5 14  6 15
c
embee4:n  embed=2
embtb4  1e+39
embtm4  1.0
embeb4  1e+10
embem4  1.0
c
embee6:n  embed=2
embtb6  1e+39
embtm6  1.0
embeb6  1e+10
embem6  1.0
c
fmesh4:n  geom=  xyz  origin= -12. -12. 0.
          imesh=  12  iints=6
          jmesh=  12  jints=6
          kmesh=  40  kints=10
c
f14:n  10
f24:n  11
f34:n  12
f44:n  13
f54:n  14
f64:n  15
c
cut:n J 1.0e-6
nps 10000
print
void
prdmp 10000000 10000000  -1
```

3.1.3   Problem 1044 Input File

```
block01: 8x8x6; 6 parts, 6 element types
c
c   --------------------------------
c   - MCNP / REGRESSION Test Suite  -
c   - Unstructured Mesh Problem #44 -
c   - Roger Martz                   -
c   - 10/13/2015                    -
c   --------------------------------
c
c   Using mcnpum file created by #1043
c
10   1    -2.03   0         u=2
11   1    -2.03   0         u=2
12   1    -2.03   0         u=2
13   1    -2.03   0         u=2
14   1    -2.03   0         u=2
15   1    -2.03   0         u=2
21   0            0         u=2 $ background
30   0           -99     fill=2 $ fill cell
40   0            99

99   sph   0.  0. 3. 10.

c   the following is pseudo-concrete m1   1001 -0.02  8016 -0.60   14000 -0.38
c
sdef    pos= 0 0 3    erg=2   nrm= -1
        rad=10         sur=99  wgt= 314.159265
c
mode n
imp:n  1 7r   0
c
embed2 meshgeo=mcnpum
       mgeoin=um1044.inp
       meeout=inp1044u
       gmvfile=inp1044g
       filetype=ascii
       background= 21
       matcell= 1 10   2 11   3 12   4 13   5 14   6 15
c
embee4:n   embed=2
embtb4   1e+39
embtm4   1.0
embeb4   1e+10
embem4   1.0
c
embee6:n   embed=2
embtb6   1e+39
embtm6   1.0
embeb6   1e+10
embem6   1.0
c
```

```
fmesh4:n  geom=  xyz  origin= -12. -12. 0.
          imesh=  12  iints=6
          jmesh=  12  jints=6
          kmesh=  40  kints=10
c
f14:n  10
f24:n  11
f34:n  12
f44:n  13
f54:n  14
f64:n  15
c
cut:n J 1.0e-6
nps 10000
print
void
prdmp 10000000 10000000  -1
```

3.1.4   Problem 1045 Input File

```
CONTINUE
c
c  --------------------------------
c  - MCNP / REGRESSION Test Suite  -
c  - Unstructured Mesh Problem #45 -
c  - Roger Martz                   -
c  - 10/13/2015                    -
c  --------------------------------
c
c  Testing mcnpum with continuation run.
c
embed2 meshgeo=mcnpum
       mgeoin=um1044.inp
       meein=inp1044u
       meeout=inp1045u
       gmvfile=inp1045g
       filetype=ascii
       background= 21
       matcell= 1 10  2 11  3 12  4 13  5 14  6 15
nps 20000
```

3.2   Verification With Volume Sources

Verification of this capability with UM volume sources is essential since 1) the code must use the mesh information to construct the volume source data structures and 2) the code must correctly write/read these data structures to/from the mcnpum file. Volume source testing is done with three problems.

The first problem is 1046 (mcnpum equivalent of 1012) and tests a single volume source. This fixed-source, neutron-only problem has allof the first order element types. The second problem is 1047 (mcnpum equivalent of 1020) and tests two volumes sources (source cells) where the UM pseudo-cells are sampled from a distribution on the SDEF card before the final position is selected within the chosen pseudo-cell. This is a fixed-source, neutron-only problem. The third problem is 1053 and is a continue run of 1047 for an additional 10,000 histories. Until 1053 was added to the test suite, none of the other REGRESSION problems tested the volume source in a continue run.

Since there are many elements in these UM geometries, it is tedious to provide results for each element. For this set of verification problems, two sets of comparisons were made between the eeout and mctal output files produced by the code for each of the runs. Running a Linux diff utility between appropriate files for the problems using the Abaqus inp files (problems 1012 and 1020) and problems using the mcnpum files (problems 1046 and 1047) runs showed that the results were identical in all cases. A continue run of 1020 was made and appropriate files where compared to those generated by 1053; again, results were identical for both the eeout and mctal files.

3.2.1 Problem 1046 Input File

```
slabmixed01mg: 20x20x100; 2 parts; mixed el types; volume source
c
c  ------------------------------
c  - MCNP / REGRESSION Test Suite  -
c  - Unstructured Mesh Problem #46 -
c  - Roger Martz               -
c  - 10/16/2015                -
c  ------------------------------
c
c  Testing mcnpum file type with um volume source.
c  Testing creation of mcnpum file type.
c
20  0              0      u=2    $ background
11  1    4.921e-5  0      u=2
12  1    4.921e-5  0      u=2
13  1    4.921e-5  0      u=2
10  2   -0.50      0      u=2
c
30  0             -99      fill=2
40  0              99

99  sph    0.  0. 50. 100.

c ** Modified materials for regression test library **
c  air
m1   8016.40c -75.6
c  pseudo-concrete
m2   1001.60c -0.02  8016.40c -0.60
c
sdef   pos= volumer   erg=2
c
mode n
imp:n  1  1  1  1  1  1  0
c
embed2 meshgeo=mcnpum
       mgeoin=um1046.inp
       meeout=inp1046u
       mcnpumfile=inp1046g
       background= 20
       matcell=  1 10  2 11  3 12  4 13
c
c
embee4:n  embed=2
```

```
embtb4  1e+39
embtm4  1.0
embeb4  1e+10
embem4  1.0
c
embee6:n  embed=2
embtb6  1e+39
embtm6  1.0
embeb6  1e+10
embem6  1.0
c
f14:n 11 12 13 f24:n 10
c
cut:n J 1.0e-6
nps 100000
dbcn  41j  1.
print
prdmp 10000000 10000000  -1 j 8000
```

3.2.2 Problem 1047 Input

```
 Rectangular source block for sdef distribution testing
c
c  Testing cell-dependent volume sources #2
c
c  Testing mcnpum file type with um volume source.
c  Testing creation of mcnpum file type.
c
c  -------------------------------
c  - MCNP / REGRESSION Test Suite  -
c  - Unstructured Mesh Problem #47 -
c  - Roger Martz                   -
c  - 10/16/2015                    -
c  -------------------------------
c
c  Created from file    : um_source_distro_block_hex_01.inp
c  Created on           :  3-27-2013 @  9:29: 2
c
c
c LEGACY CELLS
5       0          -98       fill=1 imp:n=1
6       0          98 -99        imp:n=1
7       0          99             imp:n=0
c
c PSEUDO CELLS
103       0          0  u=1   imp:n=1
102       0          0  u=1   imp:n=1
101       0          0  u=1   imp:n=1
104       0          0  u=1   imp:n=1
c

c
c SURFACES
```

```
98   sph    0.00000E+00    0.00000E+00    2.50000E+00    1.37477E+01
99   sph    0.00000E+00    0.00000E+00    2.50000E+00    2.00000E+01
c
c  DATA CARDS
embed1 meshgeo=mcnpum
       mgeoin=um1047.inp
       meeout=inp1047u
       mcnpumfile=inp1047g
       length= 1.00000E+00
       background=  104
       matcell=  1   101  2   102  3   103
c
sdef pos=fcel=d1   cel=d2
ds1 L volumer   volumer
si2 L 101 103
sp2    .4   .6
c
F2:n 98
nps 20000
print
prdmp 10000000 10000000  -1 j 1000
```

3.2.3   Problem 1053 Input

```
CONTINUE
c
c  -------------------------------
c  - MCNP / REGRESSION Test Suite  -
c  - Unstructured Mesh Problem #53 -
c  - Roger Martz                  -
c  -  4/14/2016                   -
c  -------------------------------
c
c  Testing mcnpum with volume source & continuation run.
c
embed1 meshgeo=mcnpum
       mgeoin=um1047.inp
       meein=inp1047u
       meeout=inp1053u
       mcnpumfile=inp1053g
       length= 1.00000E+00
       background=  104
       matcell=  1   101  2   102  3   103 c nps 30000
```

3.3   Verification Of Mixed File Types

One verification problem, 1048, was added to the REGRESSION test suite to demonstrate that the code can read both the Abaqus inp file and the MCNP6 mcnpum file in the same calculation and produce the correct results. This is significant in that it shows that the internal data structures can be created and populated by both mechanisms without interference. Problem 1048 is a modification of REGRESSION problem 1037; its input appears below. These problems are ray-tracing problems like those describe in Section 3.1.

As with the other problems discussed in this document, comparisons were made between the eeout and mctal files created from these two calculations. As before, running a Linux diff utility between appropriate files for the problems showed that the results were identical in all cases.

3.3.1 Problem 1048 Input

```
simple cube model used 3 times in 3 translated mesh universes
c
c  8 1st order hex elements; 1 per octant
c  each element is a statistical set, 8 total
c  fixed source, ray tracing from inward directed surface source.
c  Tally and edit results should be the element volume * source fraction.
c  Tests 4 particles & 3 tracking loops:
c    neutral particle low energy (neutrons & photons)
c    low mass charged (electrons)
c    heavy mass charged (protons)
c
c  Same as 1037, but tesing mixture of abaqus and mncpum files together.
c
c  -------------------------------
c  - MCNP / REGRESSION Test Suite  -
c  - Unstructured Mesh Problem #48 -
c  - Roger Martz                   -
c  - 10/23/2015                    -
c  -------------------------------
c
c ---cell cards---
11   1  4.7984e-02   0        u=2
12   1  4.7984e-02   0        u=2
13   1  4.7984e-02   0        u=2
14   1  4.7984e-02   0        u=2
15   1  4.7984e-02   0        u=2
16   1  4.7984e-02   0        u=2
17   1  4.7984e-02   0        u=2
18   1  4.7984e-02   0        u=2
19   0             0        u=2
121  0           -41    fill=2(1)
c
21   1  4.7984e-02   0        u=4
22   1  4.7984e-02   0        u=4
23   1  4.7984e-02   0        u=4
24   1  4.7984e-02   0        u=4
25   1  4.7984e-02   0        u=4
26   1  4.7984e-02   0        u=4
27   1  4.7984e-02   0        u=4
28   1  4.7984e-02   0        u=4
29   0             0        u=4
221  0           -42    fill=4(2)
c
31   1  4.7984e-02   0        u=6
32   1  4.7984e-02   0        u=6
33   1  4.7984e-02   0        u=6
34   1  4.7984e-02   0        u=6
```

```
35   1  4.7984e-02   0          u=6
36   1  4.7984e-02   0          u=6
37   1  4.7984e-02   0          u=6
38   1  4.7984e-02   0          u=6
39   0               0          u=6
321  0                    -43    fill=6(3)
c
998  0              41 42 43 -100
999  0                       100

c ---surface cards---
c
41   sx   14 9
42   sx  -14 9
43   sy   14 9
c
50   so 20.0
100  so 30.0

c --- DATA CARDS ---
c
c  Transformations
 tr1    14  0 -5
*tr2  -14  0 -5    180 -90 90   90 180 90   90 90 0
*tr3    0 14 -5    270   0 90   180 270 90   90 90 0
c
c ---material cards---
c   Material 1: HEU (Godiva)
c    atom density:  4.7984e-02 at/b-cm
 m1    92235  4.4994e-02
       92238  2.4984e-03
       92234  4.9184e-04
c
c ---tally cards---
f14:n 11 12 13 14 15 16 17 18 T
c
f24:n 21 22 23 24 25 26 27 28 T
c
f34:n 31 32 33 34 35 36 37 38 T
c
embee24:n   embed=2  errors=yes
c
embee114:p  embed=4  errors=yes
c
embee14:n   embed=4  errors=yes
c
embee124:p  embed=2  errors=yes
c
embee34:n   embed=6  errors=yes
c
embee134:p  embed=6  errors=yes
c
```

```
embee234:e  embed=6  errors=yes
c
embee334:h  embed=6  errors=yes
c
embee224:e  embed=2  errors=yes
c
embee324:h  embed=2  errors=yes
c
embee214:e  embed=4  errors=yes
c
embee314:h  embed=4  errors=yes
c
sdef pos= 0 0 0  erg=14   nrm= -1  par= d1
      rad= 30     sur=100  wgt= 2827.4334
c
si1 L  1    2     3     h
sp1   0.25 0.25  0.25  0.25
c
totnu  no
c
nps 1e5
prdmp J 10000000  -1  j 8000
c
mode n p e h
imp:n 1 30r 0
imp:p 1 30r 0
c
embed2 meshgeo=abaqus
       mgeoin=um1037.inp
       meeout=inp1048u2
       gmvfile=inp1048g2
       filetype=ascii
       background= 19
       matcell= 1 11  2 12  3 13  4 14  5 15  6 16  7 17  8 18
c
embed4 meshgeo=mcnpum
       mgeoin=um1048.inp
       meeout=inp1048u
       mcnpumfile=inp1048g
       filetype=ascii
       background= 29
       matcell= 1 21  2 22  3 23  4 24  5 25  6 26  7 27  8 28
c
embed6 meshgeo=abaqus
       mgeoin=um1037.inp
       meeout=inp1048u3
       gmvfile=inp1048g3
       filetype=ascii
       background= 39
       matcell= 1 31  2 32  3 33  4 34  5 35  6 36  7 37  8 38
c
print -85 -86  -130
```

```
void  11 12 13 14 15 16 17 18
      21 22 23 24 25 26 27 28
      31 32 33 34 35 36 37 38
```

4 Summary

This document provides verification that the UM flux mcnpum capability is functioning as expected. The verification procedure for the mcnpum file type capability was to create mcnpm equivalent problems from other UM test problems, run these calculations as before, and compare eeout, fmesh, and/or mctal output files using a Linux diff utility. In all such comparisons, there was identical agreement. This work demonstrates that the mcnpum file capability is working correctly for initial runs, continue runs, creating mcnpum files, volume sources, and mixed use with the traditional Abaqus inp file in a multi-mesh run.

Provided here within are the MCNP6 input listings used in this verification. The problems described is this documents are regression problems 1043-1048 and 1053 in the MCNP6 REGRESSION test suite.

References

[1] "MCNP6 User's Manual: Code Version 6.1.1 beta," Denise B. Pelowitz, Andrew J. Fallgren, and Garrett E. McMath (editors), Los Alamos National Laboratory manual LA-CP-00745, Rev. 0 (June 2014).

[2] "ABAQUS USER MANUALS, Version 6.14," Dessault Systemes Simulia, Inc., Providence, RI (2014).

[3] Roger L. Martz, "The MCNP6 Book On Unstructured Mesh Geometry: User's Guide," Los Alamos National Laboratory, LA-UR-11-05668, Rev 10 (November 2015).

[4] "MCNP - A General Monte Carlo N-Particle Transport Code, Version 5, Volume I: Overview and Theory," X-5 Monte Carlo Team, Los Alamos National Laboratory manual LA-UR-03-1987 (2008).