LLNL-ABS-680629

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

A Solid of Revolution Time Study using COG11.1 and MCNP6

W. J. Zywiec, D. P. Heinrichs

January 7, 2016

2016 ANS Annual Meeting
New Orleans, LA, United States
June 12, 2016 through June 16, 2016

A Solid of Revolution Time Study using COG11.1 and MCNP6.1

William Zywiec and David Heinrichs

Lawrence Livermore National Laboratory
7000 East Avenue, L-198, Livermore, California, 94550, zywiec1@llnl.gov, heinrichs1@llnl.gov

INTRODUCTION

The performance of two continuous energy Monte Carlo transport codes, COG11.1 and MCNP6.1, has been compared for the ^{239}Pu Jezebel Benchmark [1] using cross sections based on the ENDF/B-VII.1 evaluated nuclear data library. The purpose of this study is to determine the length of time it takes each code to complete a series of criticality calculations using two different methods to model a solid of revolution.

A solid of revolution is essential in many fields of engineering and mathematics, and has been a standard feature in computer-aided design and manufacturing since the 1970s. In the field of criticality safety, a solid of revolution can be used to reproduce and simplify complex systems that would otherwise require excessive use of geometric primitives and computational resources to achieve a comparable level of fidelity.

There are two basic methods to model a solid of revolution:

Method 1 (COG)
Specify a set of x-y pairs or polar coordinates and rotate the curve about the x-axis. The example shown in Fig. 1 consists of a single curve that has nine x-y pairs or polar coordinates.

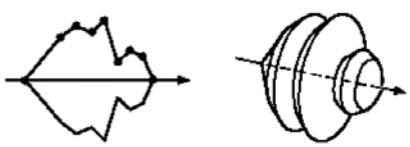

Fig. 1. Example of Method 1

Method 2 (COG and MCNP)
Specify a series of end-to-end conical frustums (and/or cylinders) that mimic Method 1. The example shown in Fig. 1 consists of eight conical frustums. This method requires the code user to redundantly specify x-y pairs, and, for MCNP, calculate the slope and x-intercept for all but two conical frustums.

Method 1 is only available in COG and can be implemented using one surface and sector specification. Method 2 can be implemented in both COG and MCNP.

This time study was initially conceived as a result of performing criticality calculations for a complex system with a high degree of rotational symmetry. Each COG input deck typically took 1-2 days to perform a single criticality calculation in serial mode. COG input decks were then converted to MCNP format and run on the same computer cluster. Instead of expected 1-2 day code runtimes, which would have been slightly faster than COG11.1, it took MCNP6.1 more than one week to complete each criticality calculation. Since MCNP typically runs faster than many codes, these lengthy runtimes were considered unusual. This study attempts to investigate the runtime discrepancy between the two codes. The benchmark in this time study was selected to provide a consistent, generalized approach to modeling an arbitrary solid of revolution. While there are many different ways to model the ^{239}Pu Jezebel Benchmark, the focus of this study is on investigating the practical difference between the two basic methods to model a solid of revolution.

DESCRIPTION OF THE ACTUAL WORK

MCNP and COG input deck specifications were taken from the ICSBEP Handbook [1] and are shown in Fig. 2 and Fig. 3, respectively.

```
PU-MET-FAST-001  239Pu Jezebel
1  1  0.040290  -1  imp:n = 1
2  0  1  imp:n = 0
1  so 6.3849
kcode  5000  1  100  500
ksrc   2  0  0
m1     31069  8.2663e-4  31071  5.4857e-4
        94239  3.7047e-2  94240  1.7512e-3
       94241  1.1674e-4
```

Fig. 2. MCNP Input Deck

```
PU-MET-FAST-001  239Pu Jezebel
BASIC  neutron delayedn URRPT
SURFACES  1 sphere 6.3849
GEOMETRY  sector 1 alloy -1
           boundary vacuum +1
CRITICALITY  npart=5000 nbatch=500 sdt=0.0001
              nfirst=100 norm=1 nsource=1 0 0 0
MIX  nlib=MCNP.71nc ptlib=PT.MCNP.71nc
     mat=1 bunches  ga    1.3752-3  pu239 3.7047-2
                    pu240 1.7512-3  pu241 1.1674-4
END
```

Fig. 3. COG Input Deck

Each sphere was sliced into several segments. A *segment* is defined as one x-y pair (Method 1) or one conical frustum (Method 2). Examples of 10 and 100-segment geometries are shown in Fig. 4 and Fig. 5, respectively.

Fig. 4. 10-Segment Geometry

Fig. 5. 100-Segment Geometry

COG and MCNP input decks were created for spheres ranging from 10 to 990 segments, in 10-segment increments. Each input deck was run in serial mode, with 5000 particles in 500 batches, ignoring the first 100 batches. Python scripts were written and used to create input decks based on the benchmark, run them, and log the elapsed time for each run in separate a separate output file.

RESULTS

Time study results are shown in Fig. 6 and Fig. 7. Above 160 segments, runtimes start to diverge and the performance of spheres modeled as conical frustums decreases exponentially. At 990 segments, the following runtimes were observed:

Solid of Revolution (Method 1)
COG11.1 - 22 minutes 10 seconds

Conical Frustums (Method 2)
COG11.1 - 795 minutes 23 seconds
MCNP6.1 - 201 minutes 43 seconds

Throughout this study, the only code that maintains reasonably low runtimes is COG11.1 (Method 1). Based on the observed performance of each code and method,

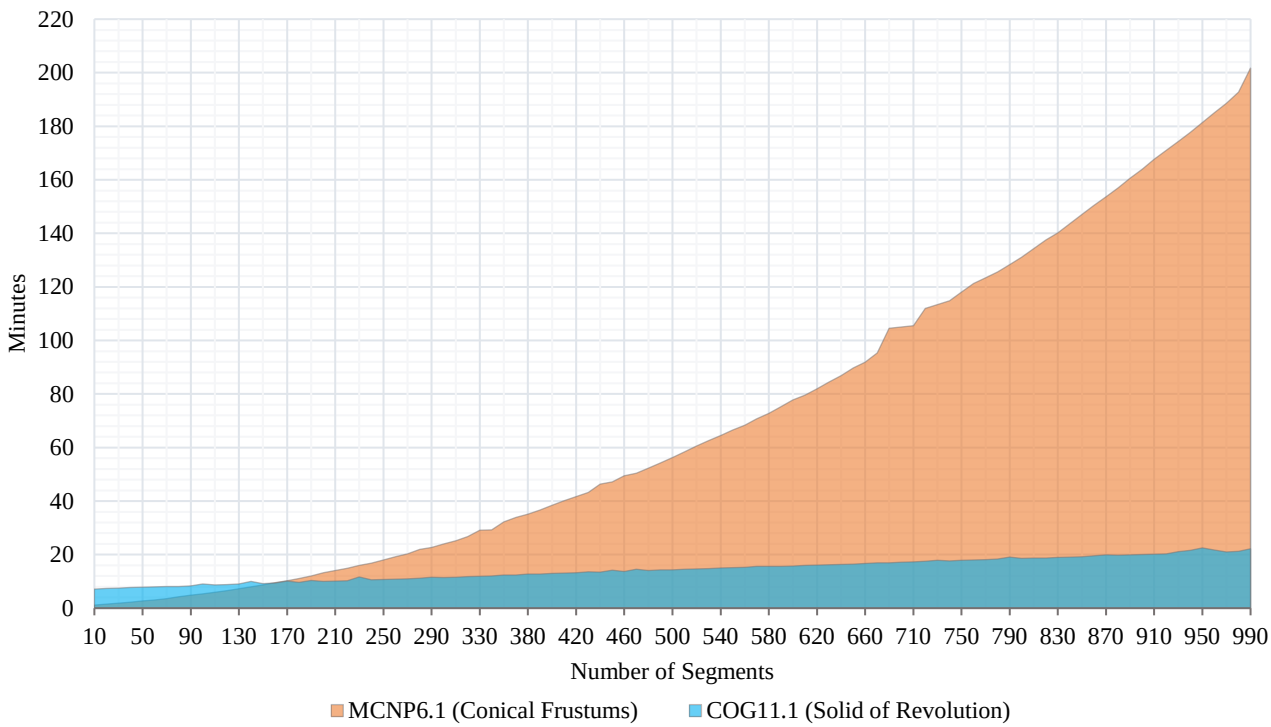

Fig. 6. Solid of Revolution Time Study Results

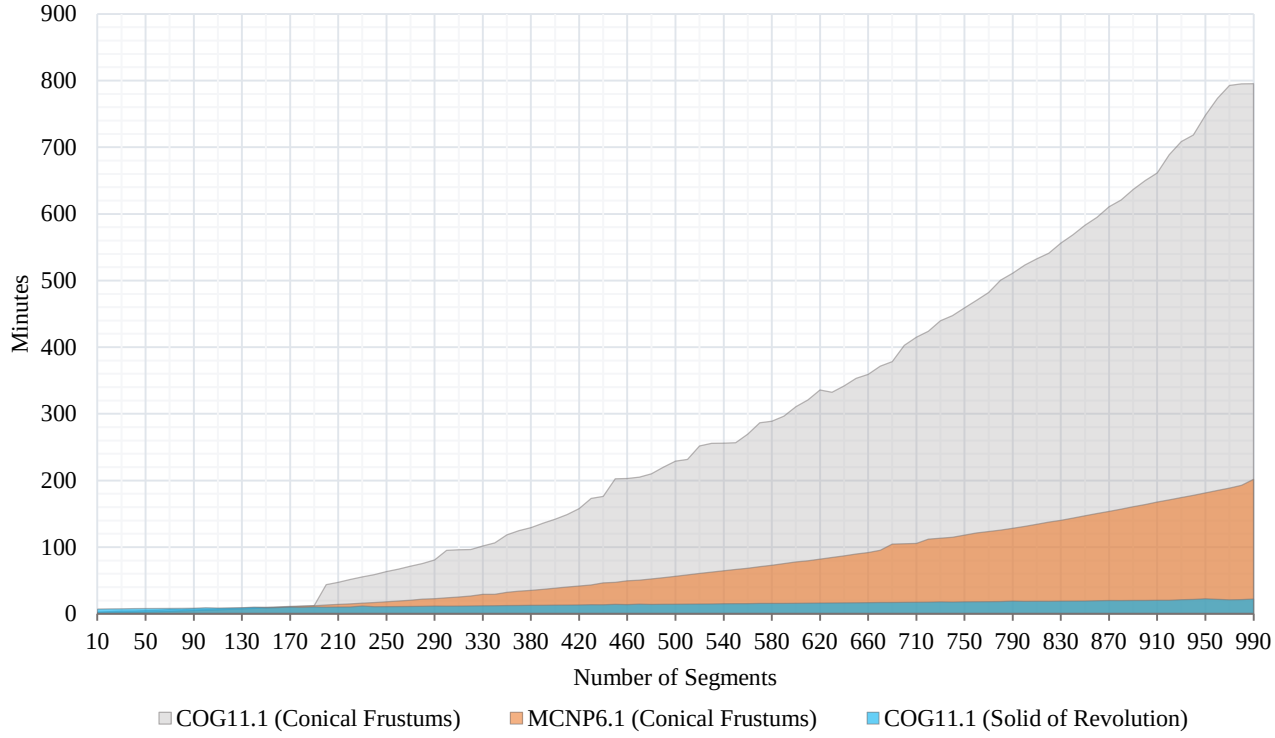

Fig. 7. Solid of Revolution Time Study Results

it is likely that the divergent runtimes are the result of tracking particles across planes that lie between conical frustums (Method 2), as opposed to defining the object boundary as a single surface using a solid of revolution (Method 1). Slight fluctuations in graphed results were also observed, due to random sampling and CPU loading. This was confirmed by re-running input decks and comparing average runtimes to a preliminary curve fit. These additional runs are not shown in Fig. 6 and Fig. 7.

From the perspective of a criticality safety engineer, the results of this study are expected to contribute towards understanding one of the practical trade-offs between codes and methods as they apply to model fidelity and ease-of-use. Ultimately, it's about choosing the right tool for the job.

REFERENCES

1 J.A. Favorite, R.W. Brewer, PU-MET-FAST-001, *International Handbook of Evaluated Criticality Safety Benchmark Experiments*, NEA/OECD (2013).

2 R.A. Buck, E.M. Lent, *COG User's Manual*, Lawrence Livermore National Laboratory (2002).

3 D.B. Pelowitz, *MCNP6.1 User's Manual*, Los Alamos National Laboratory (2013).

Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.