Automation of Creating High Fidelity TRISO Particle Fuel Element in MCNP Models

Ben Impson, Zachary Crouch, Braden Goddard, and Zeyun Wu
Department of Mechanical and Nuclear Engineering, Virginia Commonwealth University, 401 W. Main St., Richmond, VA 23284

Introduction

High temperature gas cooled pebble bed reactors are a next generation reactor design currently in development [1]. Their fuel, graphite pebbles containing TRISO particles, are a novel design when compared to traditional fuel pellets. They allow for far higher temperatures in a reactor, providing better efficiencies. However, there are challenges associated with modeling these pebbles and their behavior. There have been previous efforts to model these pebbles, although that work has concentrated on molten-salt coolants [2] [3]. This research explores the creation of input files for radiation transport codes, namely MCNP6.2, as well as the results of those inputs [4].

Creation of the Models

While the creation of a repeating lattice in MCNP6.2 is not especially difficult, it can result it features not present in actual pebbles. The most noticeable feature are cut TRISO particles, shown in Fig. 1. These cut particles can be corrected manually; however, this is extremely time consuming. As such, a computer script using MATLAB was written that would not produce the cut particles and allowing for more elaborate changes to be made quickly. MATLAB was found to be somewhat limiting, so for future purposes this script was adopted to Python.

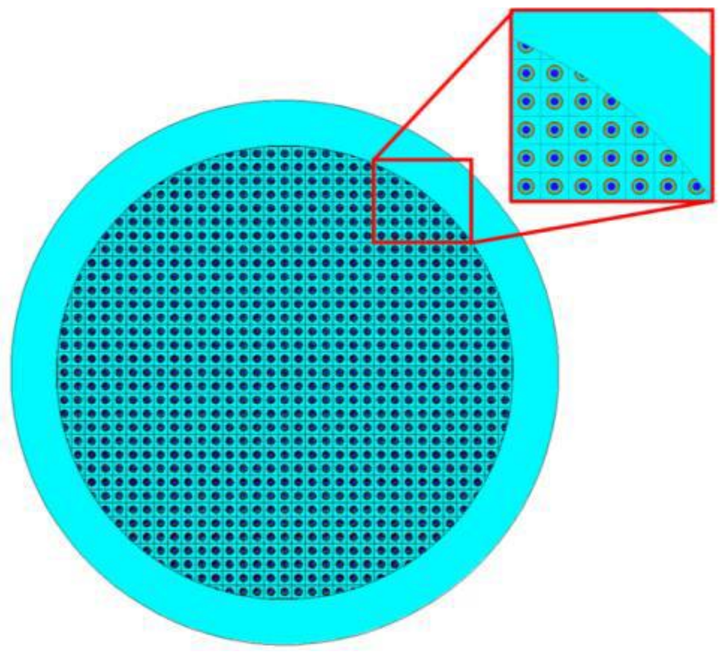


Fig 1. Manually Created Pebble Showing Cut Particles.

The Process

The script, shown in Fig. 2, is a loop that checks the position of a potential particle before moving to the next possible position. The process checks every single possible position within a given radius. The MATLAB version of the script takes approximately 40 seconds to complete, with the Python version taking approximately 45 seconds. Both options are considerably faster than removing the cut particles by hand. The resulting model contains 18,949 particles, only 51 short of the ideal 19,000. The total number of particles is limited by the geometry, 18,949 was the closest number possible.

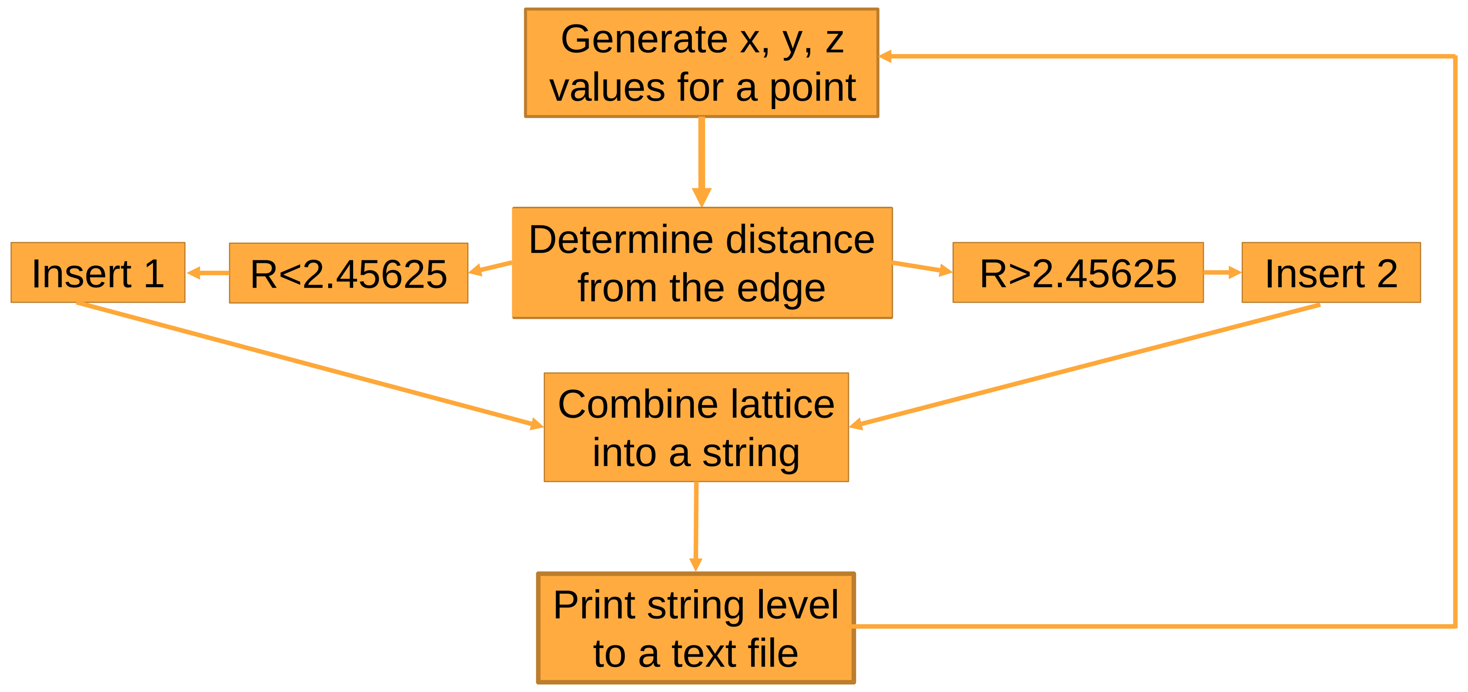
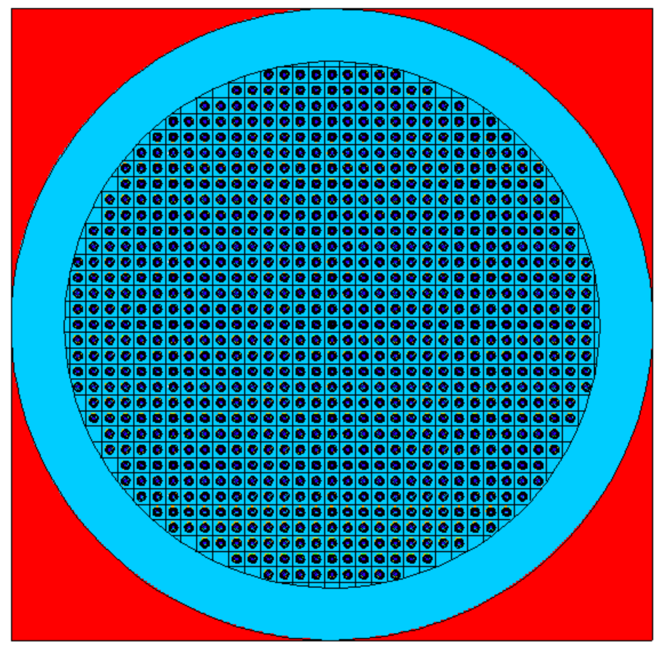


Fig 2. Diagram of the Script.

Pebble Variations

Semi-Random

Once the heterogeneous pebble model had been created, shown in Fig 3., the Python script was modified to produce different particle distributions. The first was a pebble with semi-randomly distributed particles, shown in Fig. 3. This was accomplished by producing a model that had approximately five times more particles than a pebble should contain. The program then removed 80% of those particles at random.



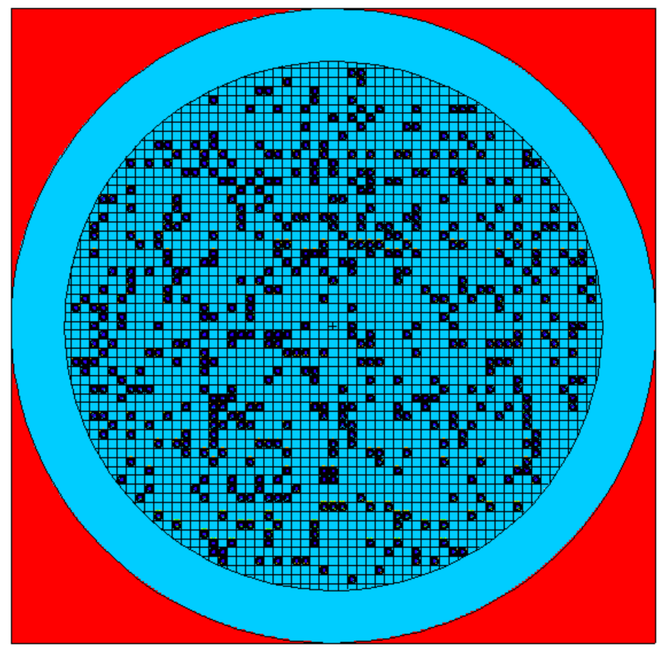
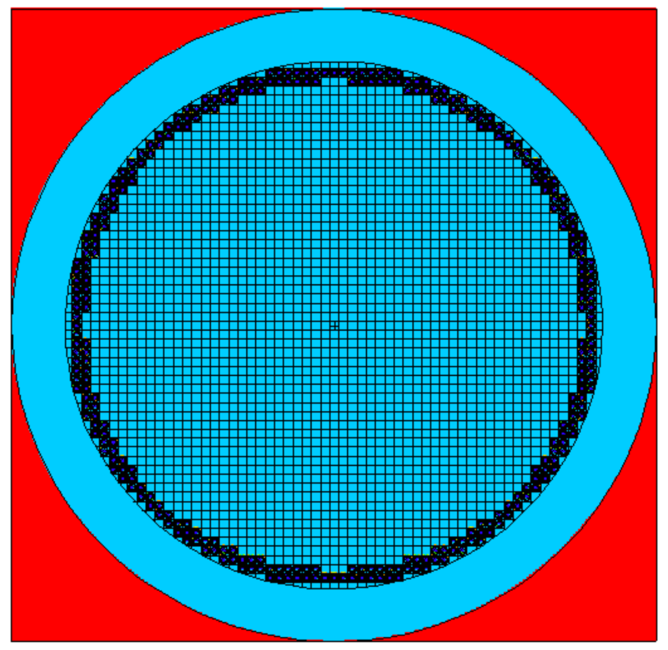
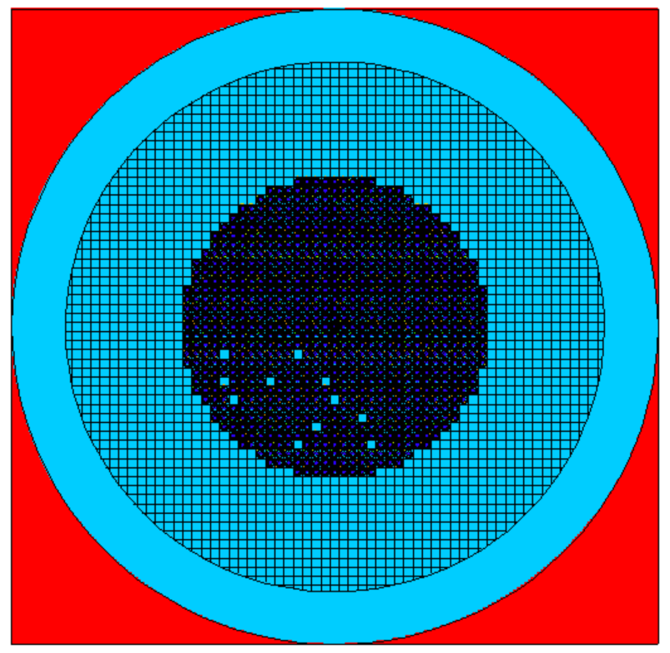


Fig 3. The Heterogeneous (left) and Semi-Random (right) Pebbles.

Iris and Pupil and Bottom

The next two pebbles developed were the 'Iris' and 'Pupil' models, shown in Fig. 4. The 'Iris' is modeled so that all the particles lie along the outer edge of the pebble's fuel region. The 'Pupil' is the opposite, with the particles packed into the center. The "Bottom" Pebble, shown in Fig. 4, only has particles in the lowest levels of the lattice.





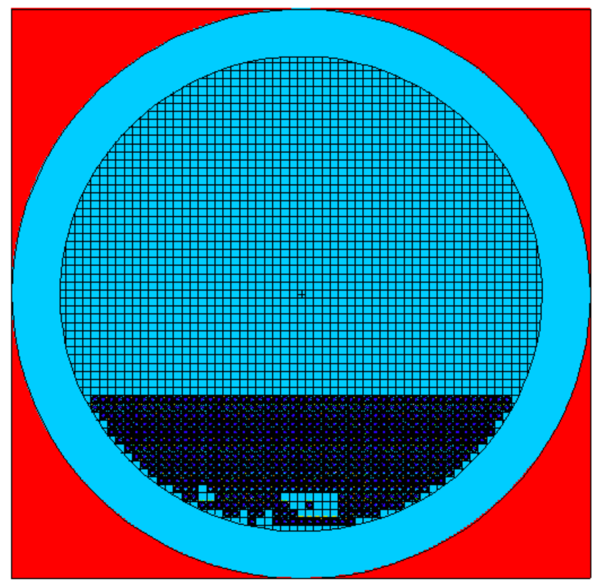


Fig 4. Iris (left), Pupil (right), and Bottom (bottom) Pebbles.

Testing the Pebbles

Running MCNP6.2 Calculations

Each of the pebble models were run using k-code in MCNP6.2. The pebbles were each surrounded by a cube of helium, as well as a reflective barrier to prevent leakage. They produced a variety of different k_∞ values, varying considerably between the geometries and pebble variations. The results are shown as follow.

Geometry	k_∞	Standard Deviation
Heterogeneous	1.50816	0.00083
Random	1.51353	0.00101
Iris	1.52369	0.00080
Pupil	1.62360	0.00083
Bottom	1.61094	0.00075

Examining the Differences

The cause of these extreme variations in k_∞ values was not immediately apparent. As such, the flux energy spectrum at the standard SCALE 252 energy groups were calculated and examined for each pebble case with the same models as above. Fig. 5 shows the pebble averaged flux spectra of each pebble case investigated in this work. The distinguishable variations appearing in the spectra agree well with the k_∞ values.

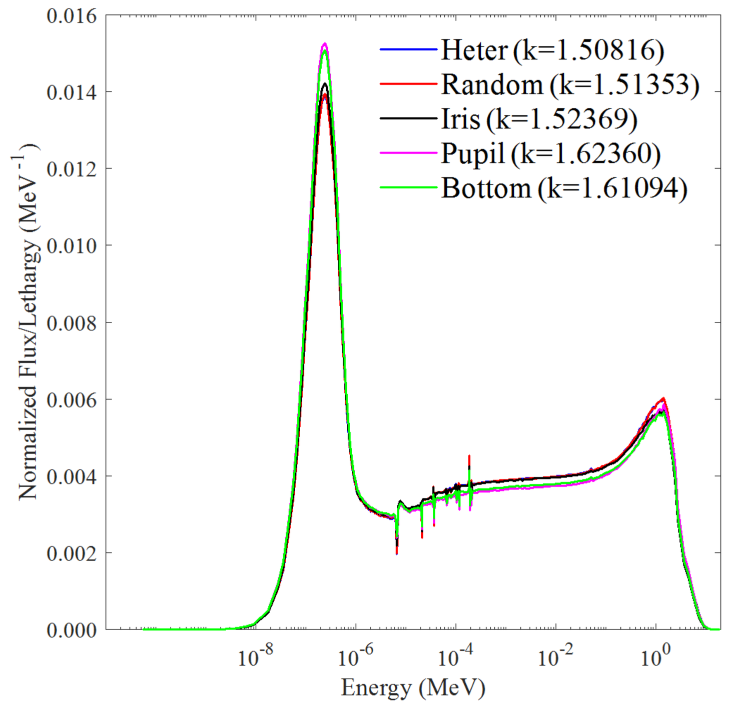


Fig 5. Flux Spectra for Various Pebbles.

Conclusion and Future Work

It has been clearly demonstrated that generating lattices for MCNP6.2 inputs can be streamlined successfully using looping scripts. This allows for the automation of modeling, taking far less time than any manual design. It is also clear that there is a considerable effect on the actual properties of a pebble from how its particles are distributed. The exact cause of this will need to be further explored, not only to determine the cause, but to determine if distributing the particles evenly is even the most ideal scenario.

References

1. Generation IV Systems, Gen IV International Forum, Retrieved January 3, 2023, from https://www.gen-4.org/gif/jcms/c_40465/generation-iv-systems
2. Auwerda, G. J., Kloosterman, J. L., Lathouwers, D., and Hagen, T. H. J. J. van der, "Effects of random pebble distribution on the multiplication factor in HTR pebble bed reactors," Annals of Nuclear Energy (2010)
3. Li, Z., Cao, L., Wu, H., He, Q., and Shen, W., "On the improvements in neutronics analysis of the unit cell for the pebble-bed fluoride-salt-cooled high-temperature reactor," Progress in Nuclear Energy (2016)
4. C.J. Werner, J.S. Bull, C.J. Solomon, et all., "MCNP6.2 Release Notes," LA-UR-18-20808 (2018)

Acknowledgement of funding statement, This work is performed with the support of U.S. Department of Energy's Nuclear Energy University Program (NEUP) with the Award No. DE-NE0009304.