OPTIMIZATION OF MONTE CARLO TRANSPORT SIMULATIONS IN STOCHASTIC MEDIA

Chao Liang and Wei Ji[*]
Department of Mechanical, Aerospace and Nuclear Engineering
Rensselaer Polytechnic Institute
110 8th street, Troy, NY
liangc3@rpi.edu; jiw2@rpi.edu

ABSTRACT

This paper presents an accurate and efficient approach to optimize radiation transport simulations in a stochastic medium of high heterogeneity, like the Very High Temperature Gas-cooled Reactor (VHTR) configurations packed with TRISO fuel particles. Based on a fast nearest neighbor search algorithm, a modified Random Sequential Addition (RSA) method is developed to speed up the generation of the stochastic media systems packed with both mono-sized and poly-sized spheres. A fast neutron tracking method is also developed to optimize the next sphere boundary search in the radiation transport procedure. In order to investigate their accuracy and efficiency, the developed sphere packing and neutron tracking methods are implemented into an in-house continuous energy Monte Carlo code to solve an eigenvalue problem in VHTR unit cells. Comparison with the MCNP benchmark calculations for the same problem indicates that the new methods have considerable higher time efficiency.

Key Words: Monte Carlo, stochastic medium, VHTR, RSA, nearest neighbor search

1. INTRODUCTION

The continuous-energy Monte Carlo method is considered the most accurate method for radiation transport simulations. It plays an important role in benchmarking other approximate deterministic codes used for the routine analysis of nuclear reactor systems. Many Monte Carlo codes have been developed for this purpose, such as MCNP, MC21, and MERCURY etc. [1-3]. While theses codes are used to analyze the stochastic media systems, such as the Very High Temperature Gas-cooled Reactor (VHTR) designs, where a large number of TRISO fuel particles are randomly distributed in a fuel pebble or a fuel compact, and provide benchmark simulations, two major challenges exist for the current modeling and simulation capability. One is the generation of the stochastic media, which requires a fast method to pack tens of thousands of fuel particles in a region. This can be even more challenging in some reactor designs, such as the Fort Saint Vrain design, where the fuel particles have a distribution in size. Normally, users employ some packing method, such as the Random Sequential Addition (RSA) [4] method to first generate the stochastic distribution of the fuel particles. The positions of fuel particles are then written into the input file of a production Monte Carlo code, such as MCNP, for the simulations. This leads to another challenge for current production Monte Carlo codes: in the neutron history tracking procedure, a distance to the boundary of next material is always

[*] Corresponding author, jiw2@rpi.edu Tel: +1 (518) 276 6602; Fax: +1 (518) 276 6025

calculated. In the region consisting of a large number of fuel particles, it is very time-consuming to calculate the distance to every fuel particle when a neutron transports in the background material. Thus the computational efficiency becomes a major concern for the users. To address these challenges, fast algorithms are needed to provide efficient sphere packing and neutron tracking methods for solving radiation transport problems in the stochastic media system.

In this paper, based on the concept of the nearest neighbor search, a generalized sphere packing algorithm is developed for the systems packed with mono-sized spheres or poly-sized spheres. An algorithm of fast neutron tracking is also developed by optimizing the next neighbor sphere search during a transport procedure. In order to demonstrate the efficiency of the two algorithms, they are implemented in a self-developed continuous energy Monte Carlo code and used to solve an eigenvalue problem for the VHTR unit cell configurations (the fuel pebble cell and the fuel compact cell). The same problem is also solved using MCNP by explicitly modeling the stochastic distribution of the fuel particles. By comparing the computational times and solutions, a very high speedup is obtained by the two algorithms without sacrificing any accuracy.

The remainder of the paper is organized as follows: Section 2 specifically describes the developed new algorithms and their capabilities. Section 3 shows the performance of the algorithms is assessed by the calculation of the infinite multiplication factor k_{inf} for the VHTR unit cell configurations. The computational efficiency is compared with the MCNP simulations for the same problems. Section 4 presents the final conclusion and future work.

2. METHODOLOGY DESCRIPTION AND CAPABILITY

2.1. A Generalized Algorithm for the Sphere Packing in a Stochastic System

The Random Sequential Addition (RSA) [4] method is usually used for geometry initialization in the stochastic media systems [5,6], which can provide a maximum volume packing fraction up to 38% [7] for a 3-D system. The basic RSA method follows a very simple procedure: 1) uniformly sample one sphere within the container; 2) compare with all the other existing spheres in the container to check if the newly sampled sphere overlaps with any one of them; 3) if there is an overlap, it is rejected and a new sphere is re-sample until no overlap occurs, otherwise, it is accepted. These steps continue until the desired volume packing fraction (VPF) is reached. Due to the global overlap checking process, the time cost of the basic RSA method follows the scale of $O(N^2)$ with the number of particle N, it is therefore inefficient when the packing system has a high density and large size.

Brown [8] has improved the algorithm and reduced the complexity from the scale of $O(N^2)$ to $O(N)$ by introducing a mesh system in the container to localize the overlap checking. This algorithm is only able to handle the mono-sized sphere system due to the required relationship between the sphere radius R and the mesh size h: $h \le 2R/\sqrt{3}$. In this paper, we modify the mesh system and extend the previous developed fast RSA algorithm by Brown to handle the poly-sized sphere system, while maintain the time efficiency to be $O(N)$.

Figure 1 is an illustration of the new mesh system in 2-D geometry. When the sampled disk as #1 is completely located inside of the mesh (2,2), the sphere is marked as belonging to that mesh.

When the sampled disk as #2 intersects with the neighboring two meshes (2,1) and (2,2), it is marked as belonging to both meshes. When the sampled sphere as #3 intersects with the neighboring four meshes (1,2), (1,3), (2,2) and (2,3), it is marked as belonging to these four meshes. This rule also applies to a 3-D mesh system. With the requirement of $h = 2\times\max(R)$ used, it guarantees that no sphere can occupy 3 adjacent meshes lined in one direction.

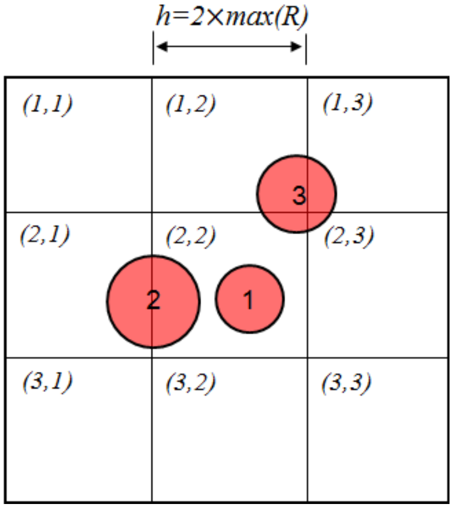


Figure 1. Illustration for the mesh system in 2-D geometry

The algorithm for the modified RSA is as follows:

```
V_total = 0
While V_total < packing fraction × container volume
        1. Uniformly sample (x, y, z) within the container
        2. Sample sphere's radius R from governing probability density function f(R)
        3. Determine the mesh IDs intersecting with the sampled sphere
        4. If the sampled sphere does not overlap with any of the existing spheres that belonging
           to the intersecting meshes
                Mark the sampled sphere as belonging to the intersecting meshes
                $V\_total = V\_total + 4pi/3R^3$
           Otherwise
                Reject the sampled sphere, and go to step 1.
```

With the modified RSA used, the computation scale of *O(N)* is kept, but given the capability of handling poly-sized sphere system.

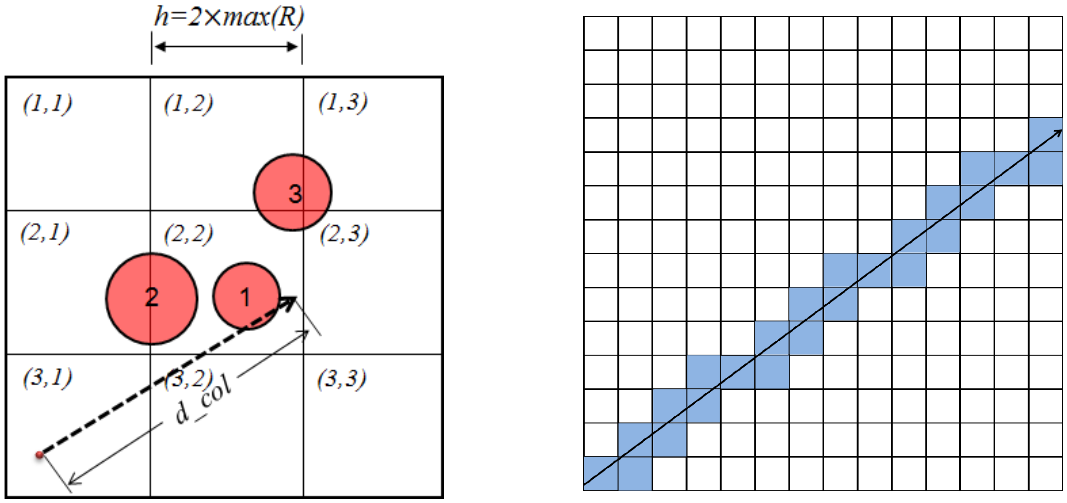
2.2. An Algorithm for the Fast Neutron Tracking

One common feature of Monte Carlo code is that it has to do intensive search calculation during the simulation due to its nature of high discreteness, such as the search for cross section data from libraries or the search for the next cell ID at each step of transport. Both of the search calculations are considerable time-consuming and occupy a large fraction of the total computation time.

When simulating a particle transport problem within a binary stochastic system, the transport procedure in a Monte Carlo code normally follows these steps:

```
1. Calculate the distances of current particle site from all the surface IDs (spheres' outer surface) d_sph within current cell
2. Choose the minimum d_sph, and determine the next possible entering sphere
3. Sample collision distance d_col
4. If d_sph < d_col
       Advance particle to the sphere's surface
   Else
       Advance particle to the collision point in the background material, and go to step 1
```

This procedure becomes extremely inefficient if the simulation is performed for the configurations with a large number of surfaces within one cell, such as the VHTR designs. Considering the situation where most of the computation time is wasted in the first transport step, it can be optimized by introducing the concept of the nearest neighbor search into the sphere search.



(a) Microscopic View **(b) Macroscopic View**

Figure 2. Illustration for fast neutron track

Figure 2(a) shows the illustration for fast neutron tracking in a 2-D system. The mesh system used for modified RSA is still used to symbolize the spheres. Assume the particle starts from the mesh (3,1), the algorithm executes the following the steps:

```
1. Sample collision distance d_col
2. Determine the mesh IDs the particle's trajectory pass through: (3,1), (3,2) and (2,2)
3. Check trajectory's intersecting from the spheres that belong to the meshes
    a) No spheres belong to (3,1)
    b) No spheres belong to (3,2)
    c) Three spheres belong to (2,2)
        i. Sphere #1 intersects with the trajectory
        ii. Spheres #2 and #3 are off the trajectory
```

4. Advance particle to sphere #1's surface

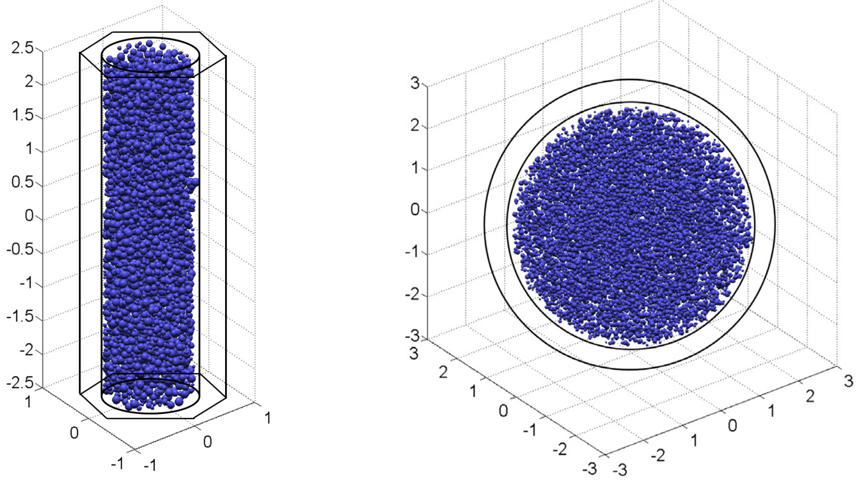
Once the neutron enters the sphere #1, the conventional Monte Carlo procedure applies until it leaks out from the sphere. In order to evaluate the performance of the nearest neighbor search used, a macroscopic view of a void mesh system in 2-D is shown in Figure 2(b). The particle's trajectory passes through 24 out of the 14×14 meshes. The search for the next entering sphere only applies within the 24 meshes. Similarly when the rule is applied to a void 3-D system, the percentage of the meshes along the particle's trajectory out of the total number of meshes is reduced to around 1%. With the algorithm used in a packed system, the search algorithm stops once it find the entering sphere, and does not have to go through all the meshes along the trajectory. The percentage of the meshes needs for check is reduced further to much less than 1%. In general, the cost of the computation for searching the next entering sphere could be significantly reduced by narrowing the scope of the search.

3. NUMERICAL RESULTS AND ANALYSIS

In order to evaluate the effectiveness and applicability of the developed algorithms, a continuous energy Monte Carlo code is developed to apply the algorithms to analyzing VHTR unit cells using the ENDF cross section libraries. Section 3.1 shows the CPU time efficiency for creating VHTR unit cell configurations with poly-sized fuel particles by using the modified RSA versus the basic RSA. Section 3.2 investigates the time efficiency of the fast neutron tracking algorithm in solving the eigenvalue problems for the VHTR unit cells. The cells are packed with mono-sized fuel particles and are analyzed by the new code and MCNP, respectively. The time efficiency and accuracy are compared.

3.1 Time Efficiency for the Stochastic System Packing

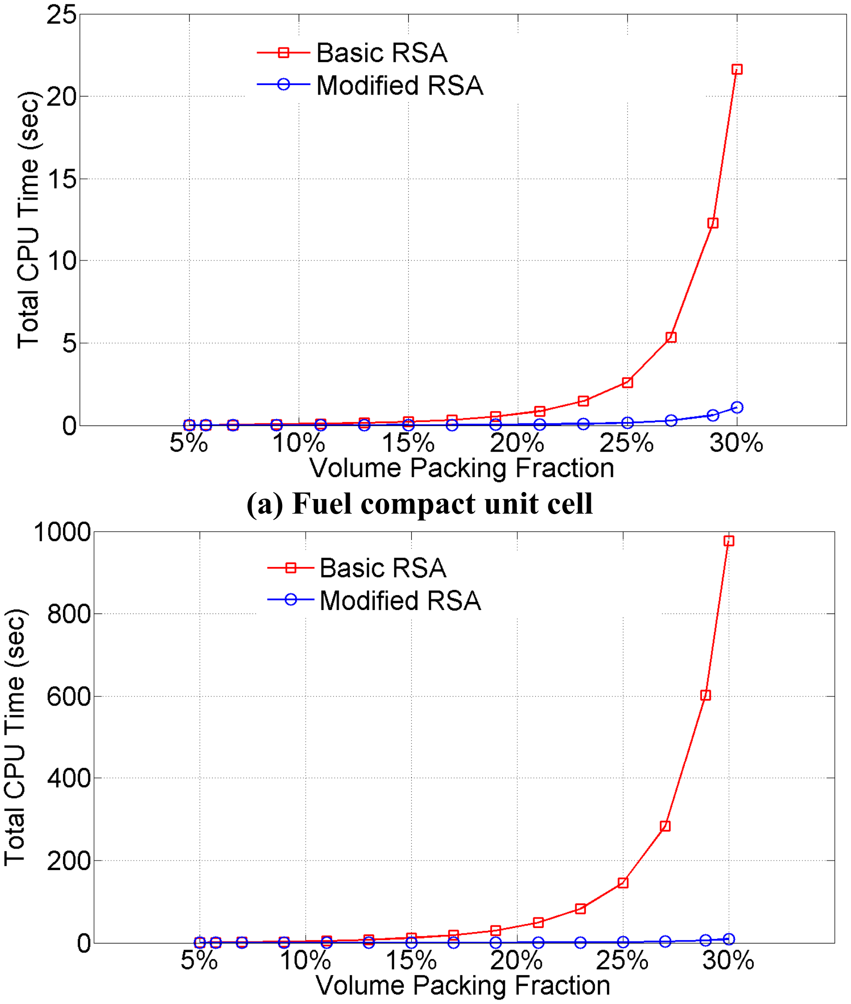
Figure 3 shows a physical realization of the stochastic distribution of poly-sized fuel particles within VHTR unit cells. The fuel kernel radius uniformly distributed within [0.0225cm, 0.0675cm]. By using the same random number seed, the basic and the modified RSA methods can achieve the same packing system without any difference in geometry.



(a) Fuel compact cell at 28.92% VPF (b) Fuel pebble cell at 5.76% VPF

Figure 3. VHTR unit cell configurations with poly-sized TRISO fuel particles

Figure 4 shows the CPU time for the packing procedure in the fuel compact cell and the fuel pebble cell. Packing fractions from 5% to 30% are tested with both the basic and modified RSA methods. From the Figure, a significant speedup for the sphere packing is achieved with the modified RSA method.



(a) Fuel compact unit cell

(b) Fuel pebble unit cell

Figure 4. CPU time for packing fuel particles in VHTR unit cells

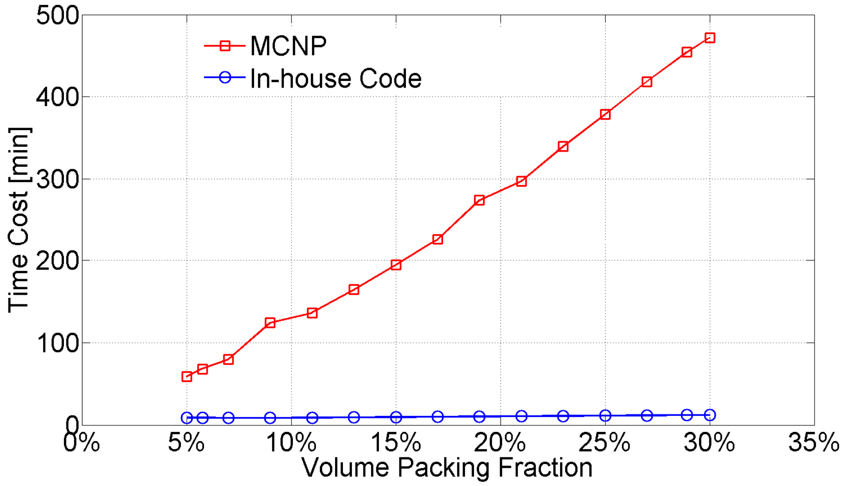
The packing CPU time for the basic RSA follows the scaling of $O(N^2)$, while the modified RSA follows the scaling of $O(N)$. At the low packing fraction less than about 15%, when the sphere number is small, both algorithms exhibit good performance with several seconds to finish the packing. However, as the packing fraction increases, the CPU time for the basic RSA climbs rapidly while that from the modified RSA remains low. This result is similar to the result found in Ref. [8], but in this paper, the capability of packing extends to the poly-sized spheres within the stochastic medium.

3.2. Speedup for the Fast Neutron Tracking Algorithm

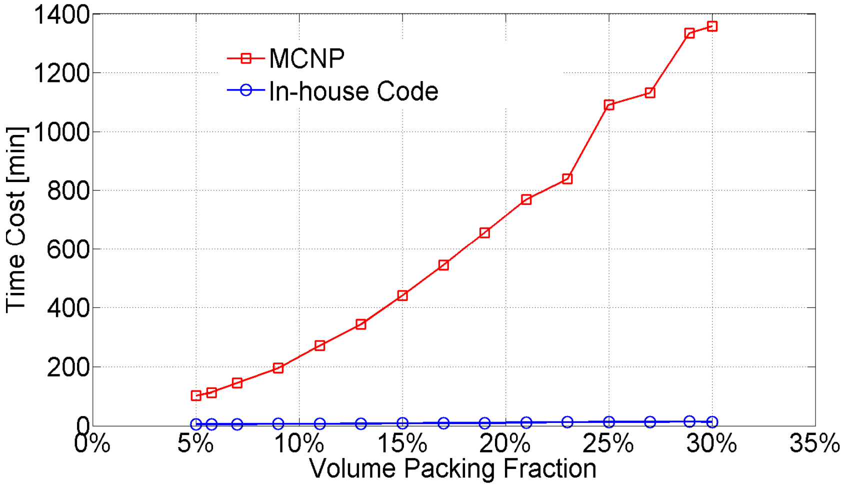
The nearest neighbor search algorithm is used during a neutron transport procedure. In order to evaluate the efficiency of the developed algorithm, eigenvalue problems for VHTR unit cells are solved as the case study. White boundaries are applied to the fuel pebble cell and reflecting boundaries are applied to the fuel compact cell. Mono-sized fuel particles are packed in the fuel pebble and fuel compact. The material composition and geometry parameters are based on Refs. [9] and [10].

In the Monte Carlo simulations for the eigenvalue problems, a total of 20 cycles with 10 inactive cycles, and a total of 5,000 histories per cycle are used to predict k_{inf}. These values are selected for the simulation parameters only for time efficiency evaluation so smaller cycle and history number would be fine. It is also confirmed from the simulation results that by keeping the seed of the random numbers and the cross section data the same, the simulation results from the in-house Monte Carlo code and MCNP are exactly the same.

The CPU time usage of a single realization simulation for the eigenvalue problem using both the in-house Monte Carlo code and MCNP is shown in Figure 5. From the Figure, it is found the CPU time of MCNP dramatically increases with the packing fraction, and in an almost linear fashion. While CPU time of the new code is always small and independent of the packing fraction.



(a) Fuel compact unit cell



(b) Fuel pebble unit cell

Figure 5. CPU time for eigenvalue problem simulations in VHTR unit cells

In order to make a direct comparison in terms of computation efficiency, Figure 6 plots the speedup, defined as the CPU time ration between MCNP and the in-house code, as a function of the volume packing fraction. Generally speaking, the speedup increases as the packing fraction increases. This is because in MCNP, the CPU time used to find the next entering sphere is proportional to the number of spherical particles. As for the in-house code, the search for the next entering sphere is restricted within the meshes along the neutron's flying trajectory. Moreover, the search does not go through all the meshes along the trajectory, instead it stops when it finds the entering sphere. With the increase of the spheres number, it has a higher possibility to find the entering sphere in a closer mesh. Thus, the computation cost varies in the opposite direction as MCNP, declining with the increase of sphere number.

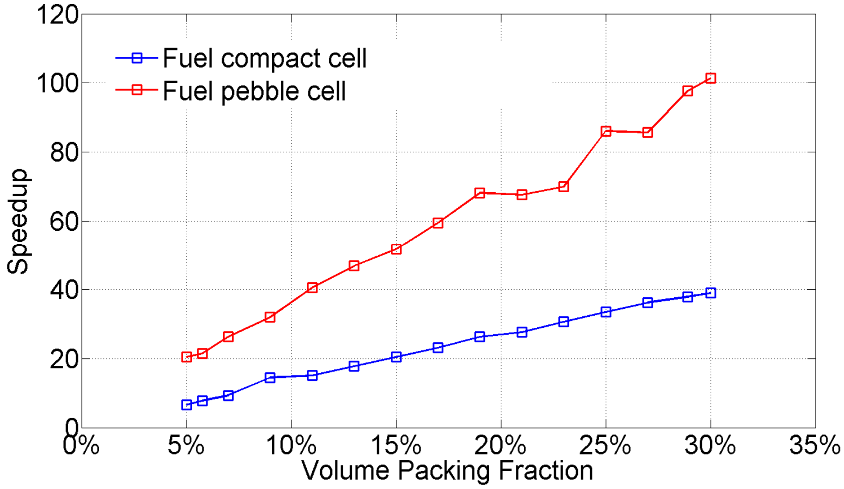


Figure 6. Speedup for the fast neutron tracking algorithm

Also from Figure 6, the speedup in the fuel pebble cell is always higher than that in the fuel compact cell at each packing fraction. The only difference between the two configurations that could affect the neutron tracking is the fuel cell size. For a fuel pebble cell, the size becomes larger and contains a larger number of spheres at the same packing fraction. Consequentially, it needs a larger computation time if MCNP code is used. However, the developed code shows an enormous advantage that the search for the next entering sphere is irrelevant to the size of the system due to its localized feature.

5. CONCLUSIONS

This paper presents an approach to optimize the transport simulation with Monte Carlo method in stochastic medium packed with either mono-sized or poly-sized spheres. Based on the concept of nearest neighbor search, two algorithms are introduced in this paper. The first is a modified RSA algorithm to pack spheres in a stochastic medium. With this method, the packing capability extends from mono-sized spheres to poly-sized spheres, and the CPU time scale at *O(N)* is kept. The second is the fast neutron tracking algorithm optimizing the sphere search from the brutal search, which is normally used in the current production Monte Carlo codes. The two algorithms are implemented and incorporated into a self-developed continuous energy Monte Carlo code. An eigenvalue problem for the VHTR unit cell configurations is used as the case study. Numerical simulation shows the new algorithm has a significant speedup in CPU time over the benchmark MCNP code, especially when applied in a system with large size and large packing fraction. For the practical application in VHTR simulation and neutronic analysis, the speedup could reach up to 35 for the compact design at the packing fraction of 28.92%, and up to 22 for the pebble-bed design at the packing fraction of 5.76%.

ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Nuclear Regulatory Commission Faculty Development Program under contract NRC-38-08-950.

REFERENCES

1.J. F. Briesmeister, “MCNP--A General Monte Carlo Code for Neutron and Photon Transport,” Los Alamos National Laboratory. LA-7396-M (1986).
2. T. M. Sutton, et al., “The MC21 Monte Carlo Transport Code,” *Proceeding of the Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007)*, LaGrange Park, IL, April 15-19, (2007).
3. R. J. Procassini, et al., “Design, Implementation and Testing of MERCURY: A Parallel Monte Carlo Transport Code,” *Proceeding of the Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew, Gatlinburg, Tennessee,* April 6-11, (2003).
4. B. Widom, “Random Sequential Addition of Hard Spheres to a Volume,” *J. Chem. Phys.*, **44**, pp. 3888-3894 (1966).
5. I. Murata, et al., “Continuous energy Monte Carlo calculations of randomly distributed spherical fuels in high-temperature gas-cooled reactors based on a statistical geometry model,” *Nuclear Science and Engineering*, **123,** pp.96-109 (1996).
6. W. Ji, et al., “Explicit Modeling of Particle Fuel for the Very-High Temperature Gas-Cooled Reactor,” *Trans. Am. Nucl. Soc.,* **92**, pp.236-238 (2005)
7. S. Torquato, “Random Heterogeneous Materials: Microstructure and Macroscopic Properties,”

Springer-Verlag (2002).
8. F. B. Brown, “A Fast RSA Method for Fuel Particle Packing,” *Trans. Am. Nucl. Soc.*, **94**, pp.571-573 (2006).
9. P. E. MacDonald, et al., “NGNP Preliminary Point Design – Results of the Initial Neutronics and Thermal-Hydraulic Assessments during FY-03,” INEEL/EXT-03-00870 Rev. 1, Idaho National Engineering and Environmental Laboratory (2003).
10. N. H. Hudson and A. Ougouag, “A Pebble Bed Reactor cross section methodology,” *Annals of Nuclear Energy*, **36**, pp.1138-1150 (2009).