CONTACT DETECTION ACCELERATION IN PEBBLE FLOW SIMULATION FOR PEBBLE BED REACTOR SYSTEMS

Yanheng Li and Wei Ji*
Department of Mechanical, Aerospace, and Nuclear Engineering
Rensselaer Polytechnic Institute
110 8th street, Troy, New York, USA 12180
liy19@rpi.edu; jiw2@rpi.edu

ABSTRACT

Pebble flow simulation plays an important role in the steady state and transient analysis of thermal-hydraulics and neutronics for Pebble Bed Reactors (PBR). The Discrete Element Method (DEM) and the modified Molecular Dynamics (MD) method are widely used to simulate the pebble motion to obtain the distribution of pebble concentration, velocity, and maximum contact stress. Although DEM and MD present high accuracy in the pebble flow simulation, they are quite computationally expensive due to the large quantity of pebbles to be simulated in a typical PBR and the ubiquitous contacts and collisions between neighboring pebbles that need to be detected frequently in the simulation, which greatly restricted their applicability for large scale PBR designs such as PBMR400. Since the contact detection accounts for more than 60% of the overall CPU time in the pebble flow simulation, the acceleration of the contact detection can greatly enhance the overall efficiency. In the present work, based on the design features of PBRs, two contact detection algorithms, the basic cell search algorithm and the bounding box search algorithm are investigated and applied to pebble contact detection. The influence from the PBR system size, core geometry and the searching cell size on the contact detection efficiency is presented. Our results suggest that for present PBR applications, the bounding box algorithm is less sensitive to the aforementioned effects and has superior performance in pebble contact detection compared with basic cell search algorithm.

Key Words: Pebble-bed reactors, Contact detection, Basic cell search, Bounding box search

1. INTRODUCTION

Pebble Bed Reactor (PBR) designs, including Pebble Bed Very High Temperature Gas-cooled Reactors [1] (PB-VHTR, with helium as the coolant) and Pebble Bed Advanced High Temperature Reactors [2] (PB-AHTR, with molten fluoride salt as the coolant), are promising future generation (Gen IV) reactors with inherently passive safety, high energy efficiency, and flexible online refueling capability. These features root in their unique fuel designs and operation characteristics: hundreds of thousands of fuel pebbles are circulating within reactor core regions which can be considered as a granular flow. To provide highly reliable safety assessment for PBR designs, one needs to perform analyses on the thermal-hydraulics and reactor core physics under steady-state, transient and off-normal operation conditions such as an earthquake [3]. The pebble flow dynamics has been shown to have significant effects on the thermal and neutronic performance within PBR cores [4, 5]. Therefore, the precise knowledge of the pebble distribution

* Corresponding author

in PBR is crucial for a reliable safety analysis for PBR design and development. To gain such knowledge, a high fidelity pebble flow simulation is desired, which has been studied by many researchers.

Some theoretical approximations, such as kinetic theory [6] and kinematic theory [7] have been proposed to quickly analyze the velocity distribution and residence time of pebbles within a PBR core, but most of these theoretical approaches have many limitations and cannot well evaluate some unique and important characteristics of pebble flow such as packing fraction distribution for either steady state or transient state, which prevents these approximations from accurately predict the in-core performance of PBR. Although the pebble distribution can be obtained through a settled static packing simulation using fast algorithms, some essential phenomena such as the near wall ordering and pebble velocity profile cannot be well reproduced by the settling algorithm [7]. Hence we cannot rely on a settled static pebble packing to obtain pebble position and velocity distributions with high accuracy. Instead, a dynamic pebble flow simulation, which preserves all essential pebble contact mechanics, is needed. Discrete Element Methods [3-5,8] and the modified Molecular Dynamics method [9] account for all the essential contact physics and use a fine time resolution to well capture the contacts and collisions with high accuracy. These micro-scale approaches, which need inter-pebble contact information to calculate the contact forces, are widely used in granular flow modeling including the pebble flow simulation in PBRs. However, due to the large number and dense packing of fuel pebbles within the active core, the contact evaluation of pebbles presents a great computational challenge. Since the pebble surface material (graphite) has a very high Young's Modulus (~10Gpa), the time step need to be very small (<1e-5s) in order to capture the contact and collision physics in the dynamic simulations [7]. Also, because the pebbles move slowly within the core, billions of time steps are needed to model the pebble recirculation within the core. In practice, contact detection (including the broad phase spatial sorting and the narrow phase contact detection [10]) account for more than 60% of the overall CPU time [10], therefore the development of acceleration methods to speed up the contact detection is of importance in addressing the computational challenges in high fidelity pebble flow simulations.

Many efforts have been made for the acceleration of contact detection in high fidelity DEM simulations for granular flow applications. Some of these efforts aimed at employing enhanced computation environment, such as MPI/OpenMP-based parallelization [11] and GPU-based parallelization [12], while other efforts tried to seek highly efficient contact detection algorithms, such as the basic cell search algorithm used for PBR simulations [11] and the Octree sorting used by Moore [13] for collision detection in computer animations. In the present work, two grid-based spatial sorting algorithms, the basic cell search algorithm [11] and the bounding box algorithm [12, 14], are investigated and compared in the DEM simulation of PBR pebble flows. Since the contact detection algorithm efficiency can be affected by the system configuration (such as system size, core geometry and pebble distribution) and the algorithm parameters (such as the grid dimension), these effects need to be studied in order to achieve optimal algorithm efficiency. The principles of Discrete Element Method and two contact detection algorithms are briefly introduced first, and then the influence from PBR geometry and searching cell size on the overall simulation efficiency are investigated. Our results suggest that for PBR applications, the bounding box algorithm shows better performance. All the results presented in this work are based on serial computations on a single-thread CPU. Since the algorithms presented in this

paper can be easily modified for parallelization [11, 12], it is promising to further enhance the contact detection efficiency by introducing more advanced computation environment such as multi-threading and GPU-based calculations.

2. DESCRIPTION OF METHODOLOGIES

2.1. Discrete Element Methods for Pebble Flow Simulations

The computation of the dynamics for a large number of pebbles can be generalized as the granular material simulation. The equations of motion for the pebbles are:

$$m_i \frac{d\mathbf{v}_i}{dt} = \boldsymbol{F}_i = \sum_{\substack{j \neq i \\ j=1}}^{N} \boldsymbol{F}_{ij} + \boldsymbol{W}_i + m_i \boldsymbol{g} + \boldsymbol{F}_{f,i}, \tag{1}$$

$$J_i \frac{d\boldsymbol{\omega}_i}{dt} = \boldsymbol{T}_i, \tag{2}$$

where $\mathbf{v}_i$ is the velocity of the ith pebble, $\boldsymbol{F}_i$ is the net force on the ith pebble including $\boldsymbol{F}_{ij}$ the contact force from the jth pebble, $\boldsymbol{W}_i$ the contact force on the ith pebble from the wall, $m_i\boldsymbol{g}$ the gravitational force, and $\boldsymbol{F}_{f,i}$ the fluid-pebble interaction force which is a function of local fluid velocity $\boldsymbol{u}$, pebble velocity $\mathbf{v}_i$ and local porosity ε. $\boldsymbol{T}_i$ is the torque on ith pebble due to the tangential contact (shear) force, and J_i is the moment of inertial and $\boldsymbol{\omega}_i$ is the angular velocity.

The pebble- pebble contact force $\boldsymbol{F}_{ij}$ consists of two components: normal contact force $\boldsymbol{F}_{n,ij}$ and tangential contact force $\boldsymbol{F}_{t,ij}$. Several models have been proposed and studied to mathematically express the contact forces. Among these models, due to high accuracy, the model of Hertzian Contact Law/Non-linear Spring with energy dissipation, which calculates the contact force based on the inter-pebble overlaps and relative velocity between contacting pebbles. The wall-pebble contact force $\boldsymbol{W}_i$ follows the same models as used in the pebble-pebble contact.

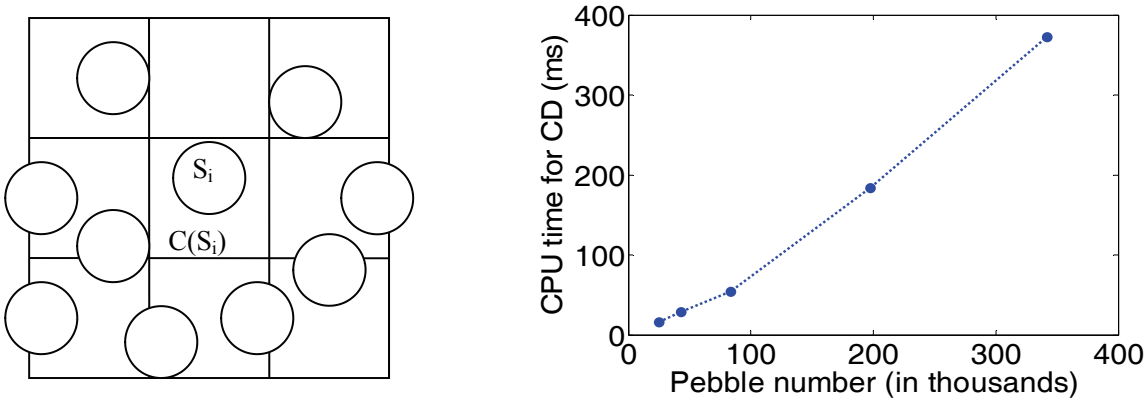
In each time step, the translational and rotational acceleration for each pebble due to the net force, $\boldsymbol{F}_i$, and torque, $\boldsymbol{T}_i$, are calculated using Eqs. (1) and (2). These values are assumed constant over each time step and, after the integration calculation, the new velocities and new positions are obtained. From the new positions, the overlaps between contacting pebbles and the consequent contact forces are evaluated. A vector summation of all forces is used in the next time step to obtain a new value of $\boldsymbol{F}_i$ and $\boldsymbol{T}_i$. Pebble-wall interactions are treated in the same way as pebble- pebble interactions except that the wall is fixed.

For DEM simulation, a critical time step exists which should not be exceeded in order to preserve the essential contact physics and maintain the stability of DEM calculation. The critical time step of DEM is determined based on the pebble material properties studied by Cundall [15] and Mishra [16]. To keep good efficiency and accuracy, and to guarantee the stability of the numerical scheme, the time step Δt is taken as a value close to the critical time step. For the studied PBR applications, a time step size $\Delta t = 10^{-5}$s is chosen based on these criteria [4, 7].

Because the pebble flow moves slowly within PBR cores, it will take billions of DEM steps to fully simulate the pebble recirculation process under realistic conditions. As the contact detection needs to be performed at every single time step, how to accelerate contact detection becomes quite important in enhancing the overall simulation efficiency. A straightforward way to find contacting pebble pairs is to perform a global search on every other pebble for each one over the reactor core region (exhaustive search). Obviously, for a PBR system containing N pebbles, the number of required search and comparison is proportional to N^2, denoted by $O(N^2)$. This is a time-consuming process for systems consisting of a large number of pebbles. More advanced algorithms use techniques based on a broad-phase spatial sorting to reduce the number of possible contacting candidates [10]. In a broad-phase spatial sorting, potential neighbors for each pebble are identified first, and then a narrow-phase contact detection is performed to determine contacting pairs. Two typical spatial sorting algorithms, the basic cell search algorithm and the bounding box search algorithm, are investigated and compared in this work for PBR applications. The basic principles of both algorithms are similar. A system is partitioned into many spatial grid cells. Each pebble is assigned to one or more "home" cells it belongs to. Each cell can be a home cell for one or more pebbles. These form mapping relationships between pebbles and cells. One can search a pebble's potential contact by 1) locating its home cell (for basic cell search) or bounding cells (for bounding box search) using pebble to cell mapping; 2) identifying neighbor cells of the home cell, if necessary; 3) identifying all the pebbles that belong to those cells including the home cell of the pebble under study using cell to pebble mapping; and 4) performing the contact detection. Details of both algorithms and results are given in following sections.

2.2. Basic Cell Search Algorithm

Basic cell search and the latter mentioned bounding box search algorithm belong to a class of grid cell-based algorithms [10], which have advantages of easy implementation and convenient incorporation with other multi-physics computations and have been employed in various applications [11, 17]. In both algorithms, the system domain is meshed into grid cells, and the broad phase identification of possible contact pairs is performed within neighboring cells instead of the whole domain. In the basic cell search (BC) algorithm, for ith pebble S_i, the home cell where the center of S_i locates is first identified and denoted as $C(S_i)$. By looping over all pebbles the complete pebble to cell mapping is established. Then the adjacent neighbor cells of $C(S_i)$ can be identified as $Nb_j[C(S_i)]$, where j is the index of neighbor cells. For 3-dimensional applications, there are 26 neighbor cells for each cell, hence j=1, 2 … 26. All the pebbles that locate within these neighbor cells $Nb_j[C(S_i)]$ as well as the home cell $C(S_i)$ (excluding S_i itself) are then identified using the cell to pebble mapping. Then the narrow phase contact detection can be performed among these pebbles. The principle of BC is illustrated in Fig. 1a.



(a) Basic cell search in 2D (b) Contact detection time vs. pebble numbers

Figure 1. Basic cell search algorithm

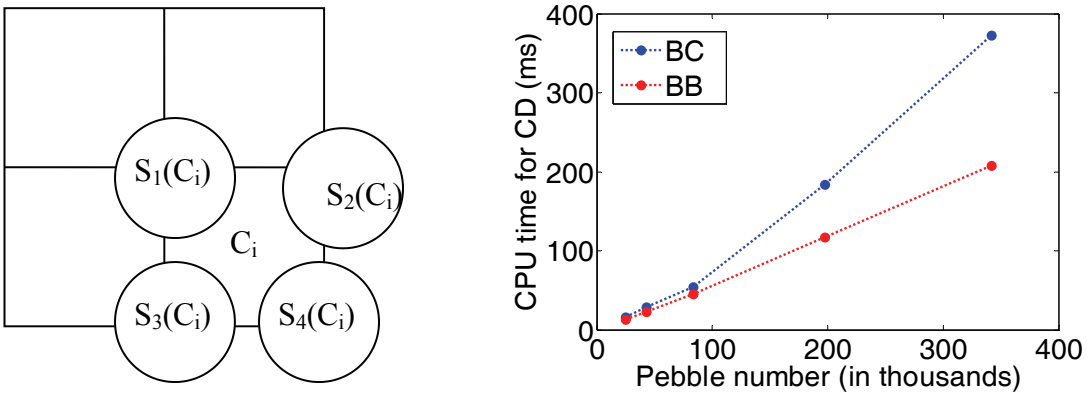
It should be noted that in BC, only the pebble center is considered. Therefore although a pebble may overlap with multiple cells, its center can be found in only one cell. Denote N as the total pebble number, and M as the average pebble number in one cell. If pebbles are evenly distributed and the side length of a cell is one pebble diameter, there are on average 8~9 pebbles in one cell with the maximum value to be 13~14. Therefore in total approximately $27MN$ potential contacts need to be examined. Thus the basic cell search has a complexity of $O(N)$ instead of $O(N^2)$.

In order to verify the relation between system size (or equivalently, total pebble numbers) and the contact detection efficiency, a cylindrical core with a 1:1 diameter-to-height ratio at 61% packing fraction is used. The geometry of the core is scaled up to accommodate more pebbles. 100 steps of DEM simulation are performed and the CPU time for the contact detection is accumulated and averaged over the 100 steps. Figure 1b shows a nearly linear relation between system size and contact detection CPU time, where "ms" in the figure stands for microsecond.

2.3. Bounding Box Search Algorithm

The bounding box search (BB) algorithm is another grid cell based search algorithm. This algorithm restricts potential contacts within an even smaller neighborhood than the BC search algorithm [12, 14]. Different from the basic cell search algorithm that only accounts for cells containing pebble centers, for a pebble S_i, BB identifies all the bounding cells (bounding boxes) it overlaps with, denoted as $C_j(S_i)$. By looping over all pebbles, the pebble-to-bounding cells mapping is established. For a typical cell size (a pebble diameter), there are normally 8 cells overlapping with S_i, hence usually j=1, 2 … 8. After gathering the pebble to bounding cells mapping information from S_i to $C_j(S_i)$ for all the pebbles, a reverse cell to pebble mapping can be established. By using this reverse mapping, for each cell C_j, we can know the pebble IDs within C_j, denoted by $S_k(C_j)$ (there are normally 8~9 pebbles overlaps with one cell in average, and 13~14 in maximum for PBR applications), and the narrow phase contact detection is performed subsequently among these pebbles ($S_k(C_j)$). It is possible that for any two pebbles S_{i1} and S_{i2}, they may overlap with multiple common cells. After the contact check within one cell, the contact information between S_{i1} and S_{i2} should be registered in a global contact array (named as

contact list) if they are contacting each other. There will be no further contact check for these two pebbles in other cells by checking with these two pebbles' IDs in the contact list. The principle of BB is shown in Fig. 2a. Since the total cell number is N/M, and BB performs $M(M-1)/2$ contact checks within one cell, the complexity of BB search can be estimated as $(M-1)N/2$ which is also $O(N)$. Although the complexity will be increased by frequently storing and accessing the global contact list (which registers all identified contact pairs) but we can still expect BB search has less complexity compared with BC. The linearity with the pebble number N is demonstrated in Fig. 2b, which uses the same geometry as in Fig. 1b. From the figure we can see that, as expected, the efficiency of BB search is improved over BC search by 15% in the N~1e4 range and more than 40% less CPU time in the N~1e5 range. In addition, BB search has better linear CPU time vs pebble number relation, which means that the CPU time is more predictable for long term simulation of large PBR systems.

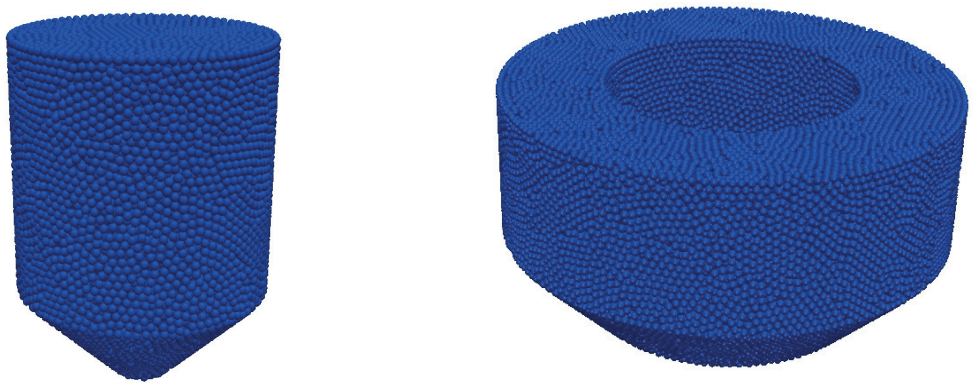


(a) Bounding box principle in 2D (b) Contact detection time vs. pebble number

Figure 2. Bounding box search algorithm

3. SIMULATION RESULTS AND COMPARISONS

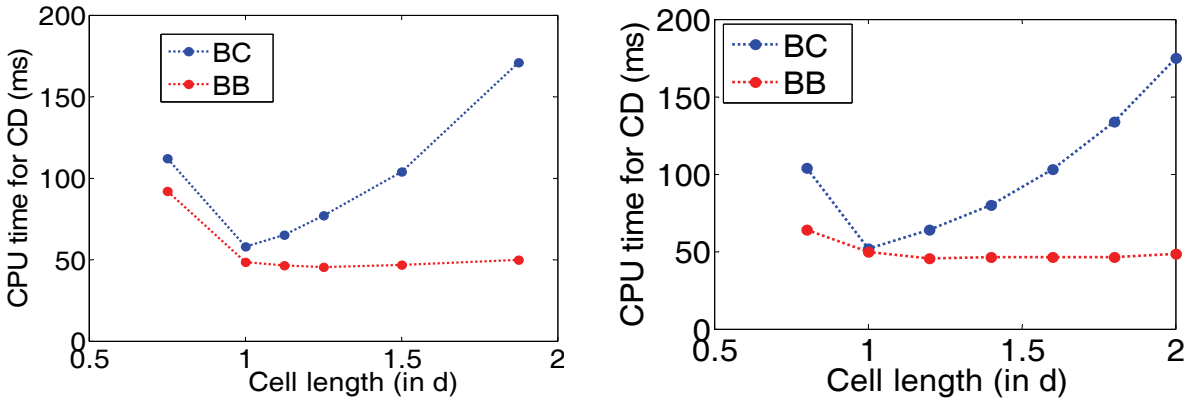
In pebble bed reactors, fuel pebbles are circulating within the core, forming a dense granular pebble flow which usually has a packing fraction above 60% [7]. There are typically two types of PBR core geometries, a cylindrical core with a conic outlet (such as the HTR10 design [18], shown in Fig. 3a), and an annular core with a conic outlet (such as the PBMR design [19], shown in Fig. 3b). For the cylindrical core, the ratio of diameter to height is set to be 1:1 and for the annular core, the ratio of outer diameter to inner diameter is set to be 1.85 to 1, based on the realistic design. In the DEM simulation, once a pebble moves out of the bottom of the core, another new pebble is inserted from top, therefore the total pebble number is fixed and the average packing fraction of pebbles within both geometries is maintained at 61%, which are based on realistic HTR10 and PBMR400 designs. The contact detection time is also recorded and averaged over 100 steps of DEM simulation at the steady state. Different core geometry may affect the pebble contact detection efficiency. Besides, the efficiencies of both searching algorithms are also affected by the grid cell size. In this section, the effect of core geometry and cell size on the efficiency of BC and BB search algorithms is investigated.



(a) HTR-10: cylindrical core geometry (b) PBMR-400: annular core geometry

Figure 3. Two PBR core geometries

First, the cell size effect on the efficiency of both algorithms is studied, which is shown in Fig. 4a (cylinder core) and Fig. 4b (annular core). The total pebble number is fixed at 1e5 for both algorithms in both geometries. For the basic cell search algorithm, the highest efficiency is achieved when the cell size $a=d$, where d is the pebble diameter (6cm in practice and also in this study). The influence from cell size on BC search efficiency behaves similarly for both cylindrical and annular core shapes. For the bounding box search algorithm, when $d<a<2d$, the CPU time is insensitive to the variation of cell size with a slight increasing as the cell size goes larger than $1.5d$. Hence we can see that the best cell size for both BC and BB search algorithms occur at approximately $a=d$, although the actual optimal value for BB search is greater than d. And the insensitivity to cell size enables the bounding box search to be more consistent in efficiency regardless of cell size and hence more promising in other applications such as modeling for poly-dispersed sphere system.



a. Cylindrical core b. Annular core

Figure 4. Cell size influence on search algorithms

By setting the cell size at $a=d$ at which both algorithms have optimal or near optimal performance, the geometry influence on the search algorithm efficiency can be investigated. For the cylindrical core, the result has been shown in Fig. 2b. A complete comparison for both algorithms in both cylindrical and annular core shapes is shown in Fig. 5.

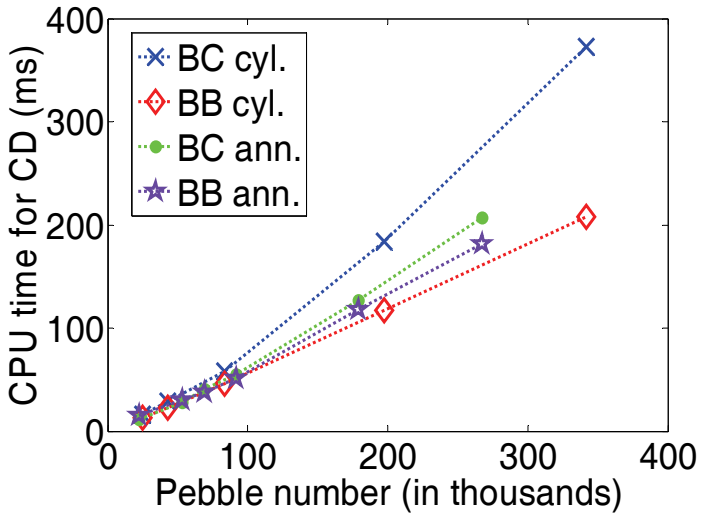


Figure 5. Dependence of contact detection efficiency on system geometry

From Fig. 5, it can be seen that geometry has a significant influence on the efficiency of both algorithms. The basic cell search algorithm shows a better efficiency in the annular core than in the cylindrical core, which is because in the annular core, a pebble has on average fewer neighbors than a pebble in the cylindrical core at the same packing fraction. However, for the bounding box search algorithm, the result is reversed. This is because the bounding box algorithm searches each cell over the system. For a cell with M pebbles, the bounding box algorithm performs $M(M-1)/2$ contact checks, as mentioned in section 2.3. The local pebble distribution in the annular core is not as even as that in the cylinder core [17], which can bring in an appreciable deviation from the nominal number of contact check ($M(M-1)/2$) among all the cells and significantly decrease the efficiency of the BB search. For the BC search, it uses a larger neighborhood (27 cells) for the search which has a smaller deviation compared with BB search and will not be affected significantly by the uneven pebble distribution in the annular core. Although BB search performs less efficiently for the annular core, it is still superior to the BC search according to Fig. 5.

For both BB and BC search algorithms, they establish a containing (or overlapping) pebble list for each cell, and then they perform contact checks pebble by pebble (within 27 neighboring cells for BC) or cell by cell (BB). Multi-node/multi-thread implementations are well suited for all the procedures performed by both algorithms [11, 12]. Because a pebble may overlap with up to 8 bounding cells for cell size $a=d$ while its center can be contained within only one cell, the BB search requires much more memory to store the mapping from pebble to its related cells than the BC search. In addition, in the BB search, two contact pebbles may share multiple common

cells. Hence for each contact pair, it requires a check against a global contact list to prevent repeated registration. On the other end, the BC search registers every contact pair twice therefore it does not need this large global contact list which prevents repeated contact pair registration. Due to above reasons, the BB search requires more memory storage and access than the BC search (more than 3 times as much as that of BC search for the scenarios studied in this work). Although the parallelization of either algorithm is not studied in this work, based on the above analysis it can be expected that the BB search will still perform better than the BC search for multi-node/multi-thread implementations with the speed up for the operations of memory storage and access.

4. CONCLUSIONS

In the present work, two commonly used grid cell-based contact detection algorithms, the basic cell search algorithm and the bounding box search algorithm, are investigated and compared in PBR applications for fuel pebble contact detections. Numerical results suggest that both algorithms have the complexity of O(*N*), which agrees with the theoretical estimation. The bounding box search algorithm has a better linearity with the pebble number N, enabling a better prediction of CPU time for pebble flow simulations in any given system size. The cell size effect on computational efficiencies of both algorithms is investigated by applying the algorithms to simulating pebble flows in both cylindrical and annular core shapes. The investigation suggests that the cell size should be chosen to be one pebble diameter (for BC search) or a slightly greater than one pebble diameter (for BB search) in order to achieve the best efficiency. Compared with the BB search, the BC search efficiency is much more sensitive to the cell size, which means that the optimal cell size should be strictly enforced in the utilization of the BC search algorithm, meanwhile in the application of the BB search algorithm, it is more flexible for the cell size selection. By fixing the cell size at one pebble diameter which is optimal or near optimal for both algorithms, the efficiencies of two algorithms for both cylindrical and annular core geometry are compared. Results show that the BB search algorithm has slightly better performance at the degree of N~1e4 and much better performance (40% less CPU time for cylinder core and 15% less CPU time for annular core) at the degree of N~1e5, than the BC search algorithm. Considering that a realistic PBR design, such as the PBMR400 design, has a pebble quantity of about N~4e5 [19], it is suggested that the BB search algorithm should be adopted for the contact detection calculation in the pebble flow simulation for the full core design. Although this work is based on a single-thread CPU calculation, the algorithms can be implemented in parallel multi-core computers. By combining with other acceleration techniques [20], the overall efficiency of pebble flow simulations in a PBR core can be greatly enhanced.

REFERENCES

1. H. D. Gougar and C. B. Davis, “Reactor Pressure Vessel Temperature Analysis for Prismatic and Pebble-bed VHTR Designs,” INL/EXE-06-11057 (2006).
2. C. W. Forsberg, P. F. Peterson and R. A. Kochendarfer, “Design Options for the Advanced High-Termperature Reactor,” *Proc. Int. Congress Advances in Nuclear Power Plants (ICAPP ’08) , Anaheim, California, June 8–12, 2008, American Nuclear Society* (2008).

3. A. M. Ougouag, J. Ortensi, and H. Hiruta, "Analysis of an Earthquake-Initiated-Transient in a PBR," *Proceedings of International Conference on Mathematics, Computational Methods and Reactor Physics* (*M&C 2009*), Saratoga Springs, New York, May 3-7 (2009).
4. Y. Li, and W. Ji, "Pebble Flow and Coolant Flow Analysis Based on a Fully Coupled Multi-Physics Model," *Nuclear Science and Engineering*, **173,** (2013).
5. Y. Li and W. Ji, "Pebble Flow Simulation Based on a Multi-Physics Model," *Trans. Am. Nucl. Soc.*, **103,** pp. 323-325 (2010).
6. D. Gidaspow, *Multiphase Flow and Fluidization*. Academic Press, San Diego (1994).
7. C. H. Rycroft, et al. "Analysis of Granular Flow in A Pebble Bed Nuclear Reactor," *Phys. Rev. E*, **74**, 021306 (2006)
8. J. J. Cogliati and A. M. Ougouag, "PEBBLES: A Computer Code for Modeling Packing, Flow and Recirculation of Pebbles in a Pebble Bed Reactor," *Proceedings of 3rd International Topical Meeting on High Temperature Reactor Technology*, October 1-4, 2006, Johannesburg, South Africa. "Spallation Neutron Source: The next-generation neutron-scattering facility for the United States," (2006).
9. K. Lee, "Particle Tracking using Molecular Dynamics Simulation for Pebble Bed Reactors," PHD dissertation, North Carolina State University (2011).
10. J. R. Williams and R. O'Connor, "Discrete Element Simulation and the Contact Problem," *Archives of Computational Methods in Engineering*, **6**, pp.279-304 (1999).
11. J. J. Cogliati and A. M. Ougouag, "PEBBLES Mechanics Simulation Speedup," Technical Report, No. INL/CON-09-17136. Idaho National Laboratory (INL) (2010).
12. J. Zheng, X. An, and M. Huang, "GPU-based Parallel Algorithm for Particle Contact Detection and its Application in Self-compacting Concrete Flow Simulations," *Computers & Structures*, **112**, pp.193-204 (2012).
13. M. Moore and J. Wilhelms, "Collision Detection and Response for Computer Animation," *SIGGRAPH Computer Graphics*, **22**, pp.289-298 (1988).
14. R. Bridson, R. Fedkiw, and J. Anderson, "Robust Treatment of Collisions, Contact and Friction for Cloth Animation," *ACM Transactions on Graphics* (*TOG*), **21**, pp.594-603 (2002).
15. P. A. Cundall and O. D. L. Strack, "A Discrete Numerical Model for Granular Assemblies," *Geotechnique*, **29**, pp.47-65 (1979).
16. B. K. Mishra, C. V. R. Murty, "On the Determination of Contact Parameters for Realistic DEM Simulations of Ball Mills," *Powder Technology*, **115**, pp.290-297 (2001).
17. Y. Li, and W. Ji, "A Collective Dynamics-based Method for Initial Pebble Pacing in Pebble Flow Simulation," *Nuclear Engineering and Design*, **250**, pp.229-236 (2012).
18. Z. Gao and L. Shi, "Thermal Hydraulic Calculation of the HTR-10 for the Initial and Equilibrium Core," *Nuclear Engineering and Design* **218**, pp.51-64 (2002).
19. F. Reitsma, et al., "The PBMR Steady-state and Coupled Kinetics Core Thermal-Hydraulics Benchmark Test Problems," *Nuclear Engineering and Design*, **236**, pp.657-668 (2006).
20. Li, Y. and W. Ji, "Acceleration of Coupled Granular Flow and Fluid Flow Simulations in Pebble Bed Energy Systems," *Nuclear Engineering and Design*, **258**, pp.275-283 (2013).