Annals of Nuclear Energy 76 (2015) 479–484

Contents lists available at ScienceDirect

Annals of Nuclear Energy

journal homepage: www.elsevier.com/locate/anucene

Optimal Spatial Subdivision method for improving geometry navigation performance in Monte Carlo particle transport simulation

Zhenping Chen [a,b], Jing Song [b], Huaqing Zheng [b], Bin Wu [b], Liqin Hu [a,b,*]

[a] University of Science and Technology of China, Hefei, Anhui 230026, China

[b] Institute of Nuclear Energy Safety Technology, Chinese Academy of Sciences, Hefei, Anhui 230031, China

ARTICLE INFO

Article history:
Received 4 September 2014
Received in revised form 20 October 2014
Accepted 22 October 2014
Available online 13 November 2014

Keywords:
Monte Carlo
Particle transport
Geometry navigation
Spatial subdivision

ABSTRACT

Geometry navigation is one of the key aspects of dominating Monte Carlo particle transport simulation performance for large-scale whole reactor models. In such cases, spatial subdivision is an easily-established and high-potential method to improve the run-time performance. In this study, a dedicated method, named Optimal Spatial Subdivision, is proposed for generating numerically optimal spatial grid models, which are demonstrated to be more efficient for geometry navigation than traditional Constructive Solid Geometry (CSG) models. The method uses a recursive subdivision algorithm to subdivide a CSG model into non-overlapping grids, which are labeled as totally or partially occupied, or not occupied at all, by CSG objects. The most important point is that, at each stage of subdivision, a conception of quality factor based on a cost estimation function is derived to evaluate the qualities of the subdivision schemes. Only the scheme with optimal quality factor will be chosen as the final subdivision strategy for generating the grid model. Eventually, the model built with the optimal quality factor will be efficient for Monte Carlo particle transport simulation. The method has been implemented and integrated into the Super Monte Carlo program SuperMC developed by FDS Team. Testing cases were used to highlight the performance gains that could be achieved. Results showed that Monte Carlo simulation runtime could be reduced significantly when using the new method, even as cases reached whole reactor core model sizes.

1. Introduction

Geometry navigation is one of the most fundamental parts of a Monte Carlo particle transport code, which considerably affects particle tracking (i.e., particle location) performance in Monte Carlo simulation. Many Monte Carlo codes (Forrest, 2005; Geant4 Collaboration, 2009) uses Constructive Solid Geometry (CSG) as their geometry representation system, where each geometric object is constructed by Boolean operation of either (1) half-space surfaces defined by mathematical equations or (2) primitive solid bodies. Some previous researches (Nease et al., 2013; She et al., 2011) indicate that geometry navigation typically accounts for 30–80% of total runtime for large-scale whole reactor core Monte Carlo simulations. In particular, the proposition of BEVARS challenge (Smith and Forget, 2013) reveals that geometry navigation will play an important role in dominating Monte Carlo simulation performance for large-scale whole core models. Therefore, if Monte Carlo codes are to be realistically used for whole core analysis and design, the optimization of geometry navigation should be investigated further. To the author's knowledge, several classical optimization techniques such as neighbor search method (Forrest, 2005; She et al., 2011), multi-level geometry hierarchy method (Donovan and Tyburski, 2006; Geant4 Collaboration, 2009), bounding box method (Millman et al., 2013; Wang, 2006) and delta-tracking method (Leppanen, 2010; She et al., 2011) have been widely used in many codes to improve geometry navigation performance. And recently, another new technique named geometric template (Nease et al., 2013) was introduced by MC21 to enhance its geometry navigation for large-scale whole core realistic models.

Generally, particle location calculation to determine in which geometric object one particle located is one of the key aspects of geometry navigation in Monte Carlo simulation. The traditional way is to search all geometric objects one by one without optimization, which will suffer a low-efficient problem for models consisting of huge number of geometric objects. Thus, a dedicated method named Optimal Spatial Subdivision (OSS) was developed for accelerating particle location calculations by minimizing the number of searched geometric objects. In computer graphics, spatial subdivision (MacDonald and Booth, 1990) is widely used to optimize runtime performance of computation-intensive applications (i.e., ray tracing). The idea of the method is to subdivide the 3D space into a set of regular grids, each containing a small number of objects.

* Corresponding author at: University of Science and Technology of China, Hefei, Anhui 230026, China.

E-mail address: liqin.hu@fds.org.cn (L. Hu).

http://dx.doi.org/10.1016/j.anucene.2014.10.028
0306-4549/

The grids are normally organized by a hierarchical data structure which represents the geometric topological relationship among those objects. It is a kind of divide-and-conquer approach that divides the 3D space consisting of many geometric objects and finds fewer candidate objects for intersection tests. For geometry navigation, selecting fewer candidate objects for testing when performing particle location calculations is the main idea to optimize the run-time performance. Thus, the motivation is that the similarity between ray tracing and geometry navigation makes it possible to make use of spatial subdivision for accelerating particle location calculations. Therefore, reduction in number of searched geometric objects for performing particle location calculation is the key aspect for optimization in this paper. The method has been implemented and integrated into the Super Monte Carlo program (SuperMC) (Song et al., 2014; Sun et al., 2013; Wu et al., 2014), which is a CAD-based multi-functional simulation system for nuclear and radiation process (Wu et al., 2011; Wu, 2008, 2007). The program SuperMC, which integrates automatic modeling (Hu et al., 2007; Li et al., 2007; Wu, 2009), multi-physics simulation and visualized analysis as a systematical system, based on hybrid Monte Carlo and deterministic transport methods (Chen et al., 2014; Wu et al., 1999), is designed to perform radiation transport, isotope burn-up and material activation simulations (Huang et al., 2009, 2004).

The structure of this paper is as follows. Section 2 describes the derivation and implementation of the Optimal Spatial Subdivision method. The implementation of geometry navigation with the method in SuperMC is presented in Section 3. The performance evaluation is assessed with specifically defined testing cases in Section 4. Section 5 is left for the conclusion.

2. Optimal Spatial Subdivision method

2.1. General idea of the OSS method

Spatial subdivision includes uniform scheme and non-uniform scheme. The advantage of uniform scheme is that it is easy to be implemented and provides $O(1)$ complexity for grid location, but it is an object-independent method which still suffers a low-efficient problem when the objects in the 3D space are not distributed uniformly. Non-uniform scheme is an object-dependent method which usually uses an octree or kd-tree (Hapala and Havran, 2011) due to its ability of matching the subdivided grids with the distribution of geometric objects in the 3D space. However, the non-uniform scheme is more complex and time-consumed than uniform scheme for grid location. For example, given a 3D space within N geometric objects, suppose a kd-tree subdivision partitions it into M leaf grids. Each traversal of the kd-tree takes $O(\log M)$ search steps (i.e., memory transactions involved or the number of visited tree nodes) for grid location. In Monte Carlo application, this will be inefficient because the number of subdivided grids M is usually extremely large.

Therefore, a novel method named Optimal Spatial Subdivision (OSS), specifically developed for Monte Carlo application, was derived as a combination of the uniform and non-uniform schemes. The new method combining both advantages of the two schemes not only provides $O(1)$ complexity for grid location as that of uniform scheme, but also matches the subdivided grids with the distribution of the geometric objects. Consequently, the method will be robust and efficient for dealing with geometry models with objects arbitrarily distributed.

2.2. Implementation of the OSS method

The Optimal Spatial Subdivision uses a recursive subdivision algorithm as that of construction for kd-trees to divide CSG model into non-overlapping grid model. The difference is that the grid model in SuperMC is built with a cost estimation function described by geometric object density heuristic rather than SAH heuristic (Havran et al., 2006) used in kd-tree scheme. The newly developed geometric object density heuristic can optimize the spatial subdivision strategies for generating numerically optimal Monte Carlo spatial grid models. As mentioned in Section 2.1, the optimal subdivision combines both characteristics of the uniform and kd-tree schemes. It means that, at each stage of subdivision for a certain space or sub-space, the subdivision is uniform with identical grid size. However, subdivision for different sub-spaces is actually non-uniform similar to that of kd-tree subdivision, which makes the scheme match the grids with the distribution of geometric objects in the Monte Carlo CSG model.

Algorithm 1. GN_OptimalSubdivision (Space *BB*, List *GE*, Grid *RT*): Given an initial spatial space (i.e., CSG model) with its bounding box *BB*, geometric objects list *GE* contained within *BB*, and *RT* is an grid array, each *RT*[i] is an interior node or leaf node, the optimal grid model returned by root grid *RT*.

```
// Recursively subdivide spatial space BB along Cartesian axes
for axis = X_axis, Y_axis, Z_axis
    // Select an axis (no subdivision before) for subdivision
    switch (axis)
        case X_axis:
            if (IsSubdividedX (BB)==false) then
                tmp_axis = X_axis
            else
                continue
            end if
        end case
        case Y_axis:
            if (IsSubdividedY (BB)==false) then
                tmp_axis = Y_axis
            else
                continue
            end if
        end case
        case Z_axis:
            if (IsSubdividedZ (BB)==false) then
                tmp_axis = Z_axis
            else
                continue
            end if
        end case
    end switch
    // Uniform subdivision for BB with planes
    // perpendicular to Cartesian axis tmp_axis
    TmpRT = UniformSubdivision(BB, GE, tmp_axis)

    // Calculate the quality factor (QF) based on a cost
    // estimation function
    QF = CalculateQualityFactor(TmpRT)

    // Select the maximum QF model as the optimal model
    if (QF > OptimalQF) then
        OptimalQF = QF
        RT = TmpRT
    end if
end for

// Further recursive subdivision for each interior
// node RT[i]
for node = RT[1], RT[2], ⋯, RT[N]    // N = RT.size()
    if (IsTerminateSubdivision (node) == false) then
        GN_OptimalSubdivision (node, node.GE(), RT[i])
    end if
end for

return
```