Efficient Modeling Methods of Large-scale Model for Monte Carlo Transport Simulation

Guiming Qin
Institute of Applied Physics and Computed Mathematics,
Beijing 100094, China
qin_guiming@hotmail.com

Yan Ma
Institute of Applied Physics and Computed Mathematics,
Beijing 100094, China
ma-yan@iapcm.ac.cn

Yuanguang Fu
CAEP Software Center for High Performance Numerical Simulation, Beijing, China
Fu-yuanguang@iapcm.ac.cn

ABSTRACT

Monte Carlo methods are widely used in simulation of the full-core reactor. They are usually adopted to deal with the geometry model based on the Constructive Solid Geometry. A visual modeling software for the automatic particle transport program, called JLAMT, is developed by the institute of applied physics and computation mathematics. It provides the computing model for Monte Carlo simulation codes, such as Geant4(a software toolkits developed by CERN) and JMCT(a 3D Monte Carlo neutron and photon transport code). To get a better result, the detailed model is needed. For devices as complex as full-core reactors, tens of thousands solids are needed to represent the model. This paper brings up efficient modeling methods of implicit modeling and layer-based modeling for solving this problem. And the effects to the overlap checking are discussed. Taking the full-core reactor of Daya bay power station as an example, experiments show that, by using the efficient modeling methods, both the amount of solids and the time of the overlap checking are reduced.

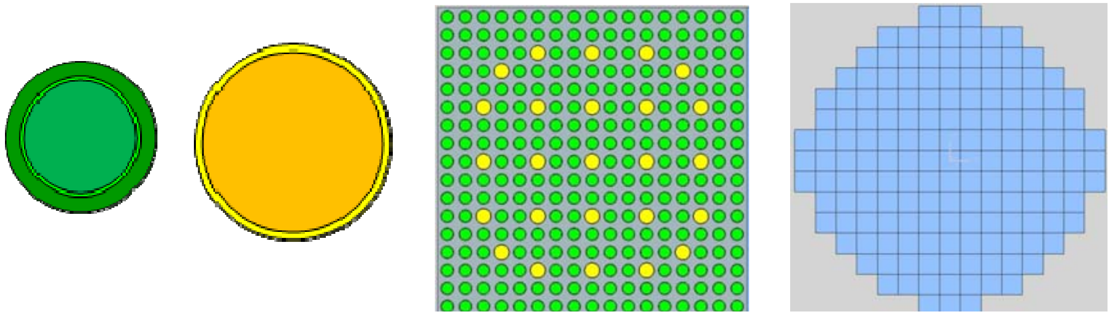
Keywords

implicit modeling, large-scale model, reactor simulation, Monte Carlo.

1. INTRODUCTION

Monte Carlo methods are a broad class of computational algorithms, born in the last century, and widely used to analyze full-core reactor configurations with exact representations of the complex geometry and physical phenomena[Met12] to observe the impact on key reactor parameters like nuclide concentrations, conversion ratio or distribution of generated power concerning, or neutron cross-sections and so on [Kep12, Mar12]. The production codes using Monte Carlo methods describe the geometry usually based on Constructive Solid Geometry (CSG) in Geant4 [Ago03]. A 3D Monte Carlo neutron and photon transport code, called JMCT, is developed for the simulation of the collision of particles with multi-group energy or continuous energy. It also uses CSG to represent geometry. All these codes need a text-based geometry input.

CSG is a solid modeling technique used to allow modelers to create a complex surface or object by using Boolean operators to combine simple objects [Wie05]. The simplest solid objects are called primitives of simple shapes, such as the cuboid, cylinder, sphere and so on.CSG is widely used in computer aided design (CAD), since it has many advantages: 1) modelling using primitives and Boolean operations is much more intuitive than specifying B-rep surfaces directly; 2) the primitives can be associated with additional information [Gui05]. To obtain a simulation with high precision, the geometry models need to be more realistic and detailed. As the computer facilities and capabilities are developed tremendously in decades, geometry modeling capabilities grow from thousands in the 80s to millions now [Bro96]. A text-based geometry description is a tedious, time-consuming and error-prone process, especially for complex geometries of reactors [Wil08]. We have developed visual modeling tool JLAMT for the field application of large-scale model. It relies on the UG CAD kernel, provides functions to build a computing model with geometry representation and physical data, and transforms it to a GDML format file that Geant4 and JMCT can accept.



a)Different cells b) One assembly c) The core of the reactor

Figure 1. Components repeated in a reactor core

A device as complex as a full-core reactor, requires tens of thousands of primitives in CSG. It is a huge pressure for memory and display (may cause lag phase when operating interactively). Even when modifying the whole model, it is hard to observe and pick up. And the more primitives used, the more time-consuming for correctness detection. In this paper, we present methods and strategies to handle and display huge amount of primitives, and build the computing model efficiently. And we discuss JMCT successfully simulated the full-core pin-by-pin problem of Daya Bay Nuclear Power Station Reactor, and calculated the k_{eff} and local tallies with the sophisticated model built by JLAMT

2. Efficient modeling strategy

2.1 Hierarchy Tree and Implicit modeling

The computing model for Monte Carlo simulation includes geometry shape, physical material, and spatial relationship. And a solid may contain a smaller solid in different materials, they have mother-daughter relationship. As a concept by Geant4, a solid with material information are logical volume. A copy of logical volume with position and rotation inside the mother volume creates physical volume [Ago03]. The supported code JMCT uses a GDML format file as model input. GDML (Geometry Description Mark-up Language) is extended from XML with many aspects of a geometry, including material properties and assemblies[Chy06].

The largest volume is called world, and all other volumes are the daughters of the world. For the spatial relation, we can build up a hierarchy tree, with the root of world. In the nuclear devices, many components are repeated with the same geometry shape and material, shown as in Figure 1. One assembly is composed by hundreds of cells, and the core is filled with hundreds of assemblies. As complex as a nuclear reactor core, a device is formed by millions of volumes. We use the implicit modeling strategy to relieve the heavy load for modeling and display. Shown in Figure2, it is a hierarchy tree of physical volumes. Volume A and volume A' are exactly the same, except their positions in space. It means that volume A and A' have the same geometry shape, the same material, and their daughters and descendants are same in shape, material and relative positions with their mothers. When creating the whole model, we create volumes of the sub-tree of volume A, and only draw the solid of volume A' without any details of descendants on the screen. All information of A' is copied from the original volume A. Therefore， the implicit modeling saves the memory and time to model and display.

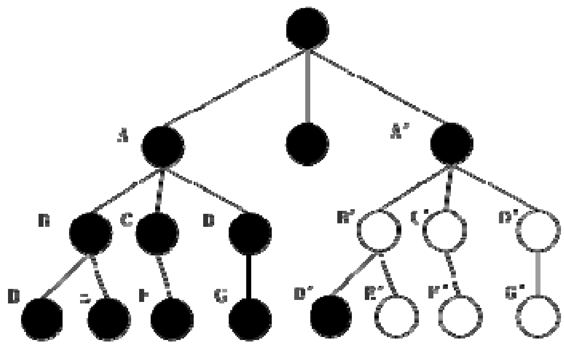


Figure2. The hierarchy tree of physical volumes.

2.2 Overlap checking

Two volumes have mother-daughter relationship if one is fully contained within the other. However, if a volume extends out of the boundaries of its mother volume, defined as overlapping. And if two volumes with no relation intersect, they are also being defined as overlapping, shown in Figure 3.

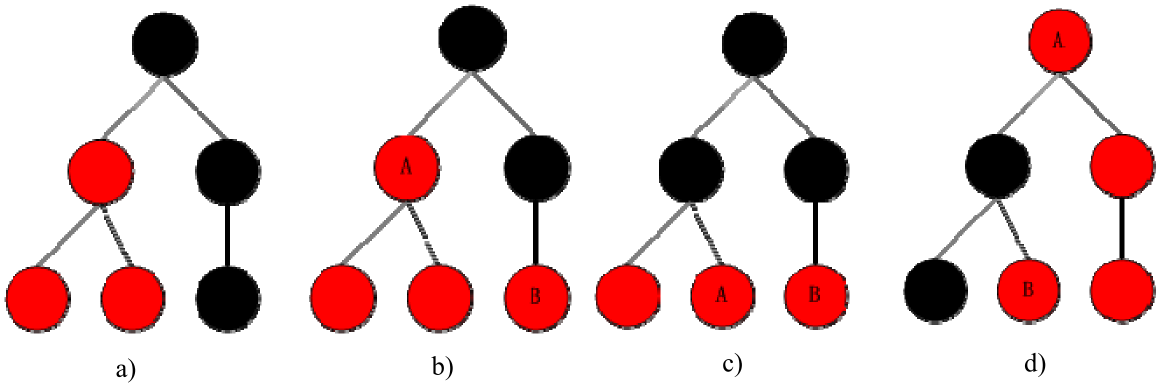


Figure 5. Four situations of volumes occurredin layer-based modeling

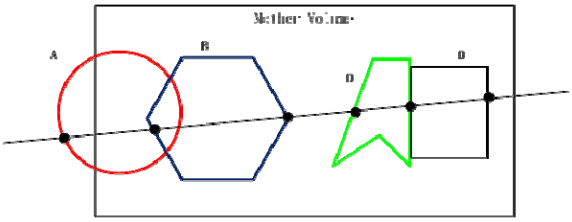


Figure 3.Different cases of volumes overlappingeach other

Detecting overlaps between volumes is bounded by the complexity of the solid model description. Unlike usual detection of CAD, the intersection between two volumes should be checked again if they are mother and daughter. The straightforward detecting algorithm is described by pseudo code below:

```
Function overlapDetect (HierarchyTree node)
    for each child of node
        CAD detect with pair of (child, node)
        if child intersects node
            check intersection with node
            if intersection is beyond the boundary of node
                add ovlapping data to record
          end if
          else add ovlapping data to record
          end if
          overlapDetect(child);
    end for
    for child1 and child2 in node
        CAD detect with pair of (child1, child2)
        if child1 intersects child2
            add overlapping data to record
        end if
    end for
```

The function overlapDetect () is a detecting method of particle transport using one thousand particles to collide with the volume.

Since the volume A' with implicit modeling has the same structure of children as the original volume A, if the original volume does not overlap with its children, A' is correct and vice versa. Therefore, creating only the solid of A' doesn't affect the result of overlap detection.

2.3 Layer-based modeling

A geometry model for complex is composed by tens of thousands volumes. It has huge pressure to display and may cause lag phase when operating interactively. Even when modifying the whole model, it is hard to observe and pick up. The major technique for managing the complex information contained in larger drawing and CAD models is layering [Howard07]. We use layer-based modeling method to narrow down the amount of volumes on the screen.

The main idea is that we create segments of the whole device in separate layers, and display the segments in one particular layer once. We can only create, edit or modify the volumes in the layer which is set to be the working layer. The other volumes not in the working layer are invisible by default. Since all layers share one coordinate system, we combine all segments intuitively as the layers are transparent, and form the whole computing model. The figure4 illustrates the hierarchy tree of the combined model.

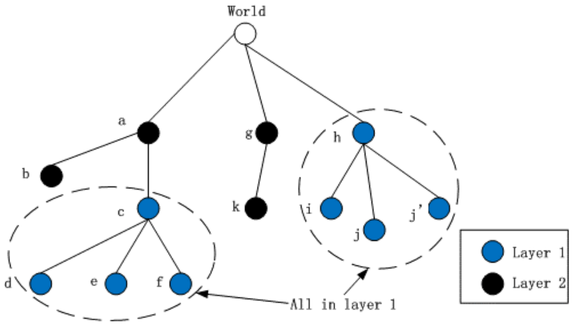


Figure 4. Relationship between hierarchy tree and volumes in layers

The strategy to create volumes in a layer depends on modelers. They can create assemble units separately or volumes with the same material in one layer. We also provide a function that volumes in a layer can be exported as a sub-model for partial simulation. The hierarchy tree of the whole model (complete tree) is separated and a hierarchy tree is built in one layer. There are several situations for the hierarchy tree shown in figure 5, volumes in red are in the same layer. The best one is the case a) where the hierarchy tree of one layer is a sub-tree of the complete tree; cases b) and c) are similar, where the mothers of volumes A and B are not the same, thus we give a virtual mother "world" to them; in case d), volumes A and B in different layers seem independent, but A and B overlap in space, which will cause some problems. Therefore, when doing the layer-based modeling, we check automatically with the rules as follows (taking the schematic in Figure 4 as example):

- The mother-daughter relationship is allowed in one layer (e.g., c and d);
- Volumes in different branches can be in one layer (e.g., c and h);
- Two volumes cannot be in one layer if there are other volumes, which do not belong to the same layer, between (e.g., a and f);
- The volume built with implicit modeling must be in the layer with the original volume (j and j').

2.4 Overlap checking in layers

Volumes in one layer can be exported and be consider as an individual model. Overlap checking for this model is need. Before checking, we preprocess the hierarchy tree. We add a virtual mother Volume "world" to them, and detect whether the situation d) in figure5 is occurred. Then we get a new hierarchy tree of volumes for the exporting layer. With this hierarchy tree, overlapping caused by layer-based modeling is eliminated.

Layer-based modeling can also accelerate the overlap checking. If volumes on one layer are checked, they are marked. When we check the whole model, relations between the marked volumes will not be calculated. If a sub-tree is in the checked layer, the sub-tree can be removed from the whole hierarchy tree, except the root of the sub-tree.

We can also check volumes layer by layer first, then form a new hierarchy tree with roots of sub-trees in the layers. The speed for overlap checking relies on the separation of whole model deeply.

3. Experiment

The full-core reactor of Daya Bay power station is very complex, the core includes 157 fuel assemblies, each component has 17*17 fuel rods, each fuel rod is composed by 2 or 3 units and needs to be divided into 16 sections. There are also many parts outside the core, including detectors, irradiation surveillance capsules, reflecting board, concrete protecting walls and so on. Total number of volumes is 1.46 million of the whole reactor. The model of the full-core reactor is created by both implicit modeling and layer-based modeling methods in figure 6. a) is the assemble with two different cells; b) is the top view of the reactor core with implicit modeling; c) is the reactor core divided into 16 sections; d) is the core with further implicit modeling. The number of total volumes reduced to 6573 with implicit modeling method. It is less than 1% of the original number. We try to reduce it further by adding virtual mother in the core. And we export some parts of the reactor as partial models. The number of volumes and the time for overlap checking are in table 1.Since the model is for nuclear particle simulation, we check the overlap using the detecting method of particle transport without parallel optimization.

	Number of volumes	Overlap checking time
The whole reactor without any methods	1.46 million	3.57 hours
The whole reactor with implicit modeling	6573	48.5 min.
The whole reactor with further implicit modeling	4211	10 min.
The detector	460	5.5 min.
The core with further implicit modeling	1040	2 min.

Table 1.The number of volumes and the time for overlap checking for modeling

We export a GDML file of the whole reactor with implicit modeling. JMCT successfully simulated the full-core pin-by-pin problem of Daya Bay Nuclear

Power Station Reactor with the GDML file, and the visualized result is shown in figure 7.

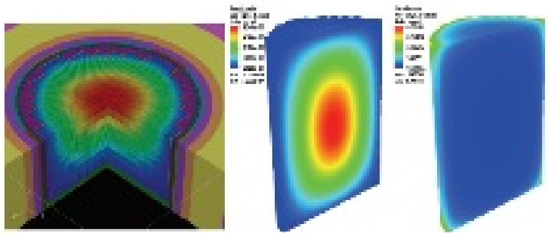


Figure 7. visualized result of the simulation

4. Conclusion

To observe the Monte Carlo simulation of complex devices precisely, such as a full-core reactor, the computing model needs to be realistic and detailed, which is represented by millions of volumes. In this paper, we present the efficient modeling methods, implicit modeling and layer-based modeling. The methods reduce the amount of volumes needed, and he burden on memory and the model displaying. We also discuss the effects for overlap checking that our methods bring to. Implicit modeling does not impact the result of overlap checking, but shortens the computing time. With the layer-based modeling, components of the device can be exported as an independent model.

5. REFERENCES

[Ago03] Agostinelli,S., et al., Geant4 - A Simulation Toolkit, Nuclear Instruments and Methods A 506 250-303, (2003).

[Bro96] Brown, F., Recent Advances and Future Prospects for Monte Carlo, Progress in NUCLEAR SCIENCE and TECHNOLOGY, Vol. 2, pp.1-4 (2011) T. F. Wiegand, Interactive Rendering of CSG Models, Computer Graphics Forum, 15(4):249-261, 1996.

[Chy06] R. Chytracek , McCormick, J., Pokorski, W., Santin, G., 2006. Geometry description markup language for physics simulation and analysis applications. IEEE Trans. Nucl. Sci. 53 (5), 2892–2896

[For 83] Forester, R., Godfrey, MCNP—a general Monte Carlo code for neutron and photon transport, in Methods and Applications in Neutronics, Photonics and Statistical Physics (R. Alcouffe, R. Dautray, A. Forster, G. Ledanois, and B. Mercier, eds.), pp. 33–47, New York: Springer-Verlag, 1983.

[Gui05] Guidera, S., Assessing New Applications for Solid Modeling (Csg) In Architectural Education Paper presented at 2005 Annual Conference, Portland, Oregon. https://peer.asee.org/14527

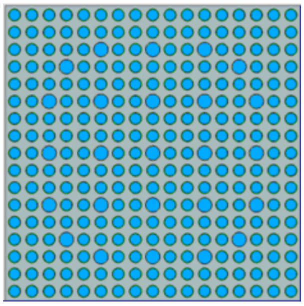
[How07] Howard, R., Bjork, B., Use of standards for CAD layers in building, Automation in Construction, vol. 16, no 3, 290 – 297, 2007

[Kep16] Kepisty, G., Cetnar, J., Stanisz, P., Parametric studies of the PWR fuel assembly modeling with Monte Carlo method, Annals of Nuclear Energy, Vol94,189-207,2016

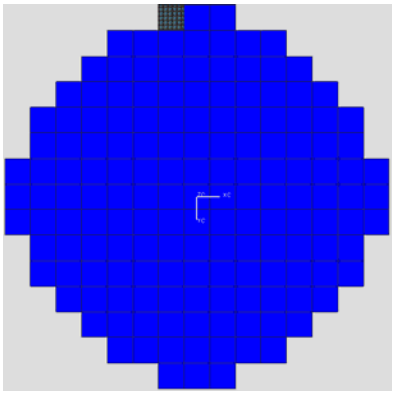
[Mar12] Martin, W., Challenges and prospects for whole-core Monte Carlo analysis, Nuclear Engineering and Technology, V44, no 2, 151-160, 2012

[Met87] Metropolis, M., the Beginning of the Monte Carlo method, Los Alamos Science, Special Issue 1987.

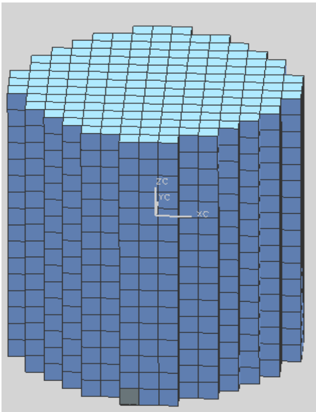
[Wil08] Wilson, P., Feder, R., et al., State-of-the-art 3-D radiation transport methods for fusion energy systems, Fusion Engineering and Design, Vol83, no 7, 824-833,2008



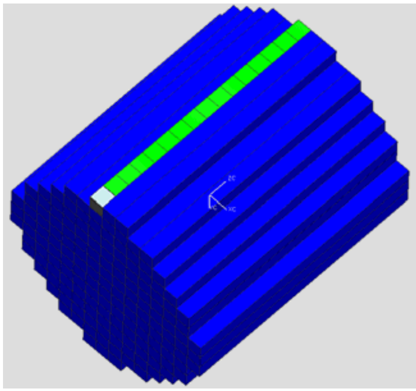
a) The assmbly



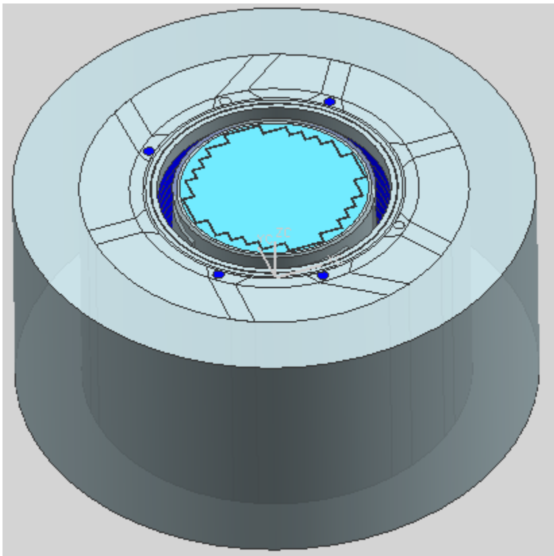
b) The top view of the reactor core



c) the reactor core divided in 16 sections



d) the reactor core with further implicit modeling



e) The complete pin-by-pin model for Daya

Figure 6. Daya Bay full-core pin-by-pin model