Nuclear Engineering and Technology 48 (2016) 635—641

Available online at ScienceDirect

Nuclear Engineering and Technology

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

Original Article

Domain Decomposition Strategy for Pin-wise Full-Core Monte Carlo Depletion Calculation with the Reactor Monte Carlo Code

Jingang Liang [a,*], Kan Wang [a], Yishu Qiu [a], Xiaoming Chai [b], and Shenglong Qiang [b]

[a] Department of Engineering Physics, LiuQing Building, Tsinghua University, Beijing, 100084, China
[b] Science and Technology on Reactor System Design Technology Laboratory, Nuclear Power Institute of China, Changshun Ave 328, Shuangliu, Chengdu 610041, China

ARTICLE INFO

Article history:
Received 19 October 2015
Received in revised form
14 December 2015
Accepted 19 January 2016
Available online 9 February 2016

Keywords:
Domain Decomposition
Full-Core Depletion
Monte Carlo
Reactor Monte Carlo
RPHA15

ABSTRACT

Because of prohibitive data storage requirements in large-scale simulations, the memory problem is an obstacle for Monte Carlo (MC) codes in accomplishing pin-wise three-dimensional (3D) full-core calculations, particularly for whole-core depletion analyses. Various kinds of data are evaluated and quantificational total memory requirements are analyzed based on the Reactor Monte Carlo (RMC) code, showing that tally data, material data, and isotope densities in depletion are three major parts of memory storage. The domain decomposition method is investigated as a means of saving memory, by dividing spatial geometry into domains that are simulated separately by parallel processors. For the validity of particle tracking during transport simulations, particles need to be communicated between domains. In consideration of efficiency, an asynchronous particle communication algorithm is designed and implemented. Furthermore, we couple the domain decomposition method with MC burnup process, under a strategy of utilizing consistent domain partition in both transport and depletion modules. A numerical test of 3D full-core burnup calculations is carried out, indicating that the RMC code, with the domain decomposition method, is capable of pin-wise full-core burnup calculations with millions of depletion regions.

1. Introduction

With the higher requirements for the safety and economy of nuclear reactors, as well as the developments of new types of nuclear systems, traditional methods and tools for reactor analysis are being challenged. The Monte Carlo (MC) method is becoming an important area of research for the next generation of methods for reactor physics calculations. With the development of parallel computing technology, the expectations are rising to see the MC method being truly applied in nuclear reactor engineering design practices [1]. However, a prohibitive amount of data is required for storage in large-

* Corresponding author.
E-mail address: liangjg2008@gmail.com (J. Liang).
http://dx.doi.org/10.1016/j.net.2016.01.015

scale calculations in MC codes. Such excessive memory demands turn into a key obstacle for the application of MC method in accomplishing pin-wise three-dimensional (3D) full-core calculations. In particular, for whole-core burnup calculations with millions of burnup regions, the data storage reaches up to hundreds of gigabytes or even terabytes, which far exceed the capacity of current computers.

Data decomposition [2,3] and domain decomposition [4,5] are two feasible ways to solve the memory problem of MC. Through the former method, specific types of data are decomposed and distributed on different processors and parallel communications are called for data operation in neutron-simulating processes. For the latter method, the idea is to divide the problem model into smaller geometry domains, which are assigned to different processors. The domain-related data are, meanwhile, decomposed. The particles are communicated between processors in the domain decomposition method as the tracks of particles are cut into pieces.

In previous studies, tally data decomposition (TDD) algorithms [6] have been designed and implemented based on the Reactor Monte Carlo (RMC) code [7]. Thereafter, a combination of TDD and depletion isotope data decomposition [8] is utilized to alleviate the memory problem, enabling simple 3D whole-core MC burnup calculations with hundreds of thousands of depletion regions [9]. However, the memory problem still exists for larger scale or fine 3D whole-core burnup, because the material data cannot be decomposed in the TDD method. In this paper, the domain decomposition method is investigated to solve the memory problem thoroughly. Through this work, fine 3D whole-core burnup calculations with millions of depletion region are achieved.

2. Memory evaluation of MC codes

For an in-depth knowledge of the memory problem of MC, it is necessary to classify data and analyze each data class quantitatively. Taking RMC as the reference, normally suitable to other MC codes, the data can be classified into six categories: geometry, material, nuclear data, particles, tallies, and burnup. The memory model can be constructed by going deep into each data type and evaluating their memory sizes in detail, as shown in Eqs. (1) and (2):

$$M = M_{geo} + M_{mat} + M_{cs} + M_{part} + M_{tally} + M_{burn} + M_{temp} \tag{1}$$

$$M \approx N_{cell}\overline{m}_{cell} + N_{mat} + \overline{m}_{mat} + N_{tot\_nuc}\overline{m}_{nuc\_cs} + N_{part}m_{part} + N_{tally}\overline{m}_{tally} + N_{burncell}m_{burncell} \tag{2}$$

where total memory usage of a code, M, is the sum of the memory of different data types, M_x, with x as the data category. For example, M_{geo} defines memory of geometry data. M_{temp} represents all other temporary and supporting data, which are generally negligible in the memory footprint. Furthermore, all data of concern have a vector structure, and their memory sizes are proportional to the amount of unit data. Total memory approximates into Eq. (2), where N_y and $m_y/\overline{m}_y$ are the number and unit or average storage size of specific data structure y, respectively.

Specifically, for the RMC code, the unit or average storage of each data type can be estimated. For example, data of one material contain names (12 bytes), ID (4 bytes), and atom/mass densities (16 bytes) of all nuclides in the material. For a depletion calculation, assuming there is an average of 150 nuclides/region, the memory storage of one material is about $32 \times 150 = 4.8 \times 10^3$ bytes. Particle data contain eight double-precision floating variables (3 for coordinates, 3 for direction, 1 for energy, and 1 for weight) to record particle state information, and therefore, its unit size is 64 bytes. Similarly, unit storage of tally, which is composed of statistics and filter data, is about 70 bytes. For burnup data, RMC accounts for 1500 nuclides in the depletion chain and the predictor–corrector method is used, and therefore, the unit datum is 3.6×10^4 bytes. Finally, Eq. (3) is obtained to describe the memory consumption in burnup simulations using RMC.

$$M_{RMC} \approx (100 \times N_{cell} + 4.8 \times 10^3 \times N_{mat} + 2 \times 10^6 \times N_{tot\_nuc} + 64 \times N_{part} + 70 \times N_{tally} + 3.6 \times 10^4 \times N_{burncell})\,\text{bytes} \tag{3}$$

Table 1 summarizes unit storage, scale of unit, and maximum storage of each data type. It can be seen that three types of data (i.e., tally data, burnup, and material) are the main sources of memory problems of MC codes.

The Hoogenboom–Martin whole core [10] was chosen as a case study of large-scale MC burnup calculations. There are a total of 241 assemblies and 63,624 fuel rods in the Hoogenboom–Martin core. In the modeling, each rod contains 24 burnup regions (12 axially by 2 radially) to perform the depletion calculation. Table 2 predicts the memory storage using the memory model.

3. Domain decomposition method

Different from the data decomposition method, spatial domain decomposition (SDD) divides spatial geometry into domains, which are simulated separately by parallel processors, and particles crossing domains are communicated for continuing tracking.

As indicated in Fig. 1, the main steps involved in implementing SDD in particle transport MC code include (1) dividing

Table 1 – Memory storage evaluation of data types.

Data types	Unit storage	Scale	Maximum storage
Geometry	100 bytes	10^0–10^6	0.1 GB
Material	4.8 KB	10^0–10^7	10 GB
Nuclear data[a]	2 MB	10^0–10^2	1 GB
Particle	64 bytes	10^4–10^6	0.1 GB
Tally	70 bytes	0–10^{10}	100 GB
Burnup	36 KB	0–10^7	100 GB

GB, gigabyte; KB, kilobyte; MB, megabyte.
[a] Assuming nuclides are in single temperature.

Table 2 – Memory storage of H–M whole-core burnup.

Data types	Memory storage
Material	7.3 GB
Nuclear data	400 MB
Particle	64 MB
Tally	64.1 GB
Burnup	55.0 GB
Total	126.9 GB

GB, gigabyte; MB, megabyte.