Annals of Nuclear Energy 112 (2018) 693–714

Contents lists available at ScienceDirect

Annals of Nuclear Energy

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

ARRC: A random ray neutron transport code for nuclear reactor simulation

John R. Tramm [a,*], Kord S. Smith [a], Benoit Forget [a], Andrew R. Siegel [b]

[a] Massachusetts Institute of Technology, Department of Nuclear Science & Engineering, 77 Massachusetts Avenue, 24-107, Cambridge, MA 02139, United States
[b] Argonne National Laboratory, Mathematics and Computer Science Department, 9700 S Cass Ave, Argonne, IL 60439, United States

ARTICLE INFO

Article history:
Received 26 June 2017
Received in revised form 3 October 2017
Accepted 6 October 2017
Available online 11 November 2017

Keywords:
The Random Ray Method
Method of characteristics
Neutron transport
Reactor simulation
High performance computing

ABSTRACT

A massively parallel implementation of a recently developed technique for numerically integrating the transport equation, The Random Ray Method (TRRM) (Tramm et al., 2017), is applied to several large reactor benchmark problems. The implementation, which is part of a new development called The Advanced Random Ray Code (ARRC), is one of the first parallel implementations of TRRM. Our goal is to better understand the accuracy and performance characteristics of TRRM on massive scale problems, and to provide community software that facilitates further algorithmic development and potentially its application to a broader class of problems. Key features of ARRC include extreme memory efficiency, domain decomposition, a task based parallel structure, and the ability to efficiently utilize Single Instruction Multiple Data (SIMD) vector units. These attributes lead to efficient performance on modern high performance computer (HPC) architectures, enabling the detailed simulation of reactor cores in three dimensions.

1. Introduction

The high fidelity simulation of a full core nuclear reactor is a computational challenge well suited to the Exascale era of supercomputing due to the extremely high number of core-hours required to fully resolve a neutron transport problem (Smith, 2003). A variety of transport methods are in theory capable of resolving such problems, though not all are well suited for continued increases in performance on next generation computing architectures.

We recently reported on a new, highly efficient transport method called The Random Ray Method (TRRM) (Tramm et al., 2017). TRRM can be viewed as a hybrid between Monte Carlo (MC) and Method of Characteristic (MOC) methods. It shares the basic computational structure of a multigroup MOC algorithm but with a stochastic rather than deterministic method of discretizing the spatial and angular dependencies of the Boltzmann Neutron Transport Equation. TRRM allows for full 3D geometric flexibility similar to MC methods, requiring no assumptions of axial homogeneity in the reactor geometry to maintain computational efficiency. It also provides extreme reductions in memory usage compared to traditional MOC methods, facilitating its use for full core, high fidelity 3D reactor simulation. TRRM has also been shown to reduce algorithmic complexity compared to traditional MOC methods, allowing a sparser computational grid while maintaining accuracy. Initial work on TRRM demonstrated the new method using a test application on several benchmark problems, all 2D and 3D pin cell problems or 2D assembly scale problems of reduced complexity (Tramm et al., 2017). A complete TRRM implementation has not been reported on, and no large 3D problems have been demonstrated.

As a next step, The Advanced Random Ray Code (ARRC) has been developed as a more robust, complete implementation of TRRM targeting high fidelity full core nuclear reactor problems on next generation supercomputing systems. It is implemented in the C programming language for performance and portability on high performance computing (HPC) architectures. Three-way hybrid parallelism is achieved by using compiler directives for SIMD vectorization, OpenMP for shared memory threading, and MPI for distributed memory domain decomposition. Additionally, an improved convergence scheme using Shannon Entropy was developed and implemented in ARRC to offer more reliable stationary source detection.

The aim of this study is to analyze the accuracy and performance of ARRC to gauge its utility for large scale 3D problems. To this end, several 2D and 3D benchmark problems are executed on a variety of HPC architectures. Several challenges and their solu-

---

* Corresponding author.
E-mail addresses: jtramm@mit.edu (J.R. Tramm), kord@mit.edu (K.S. Smith), bforget@mit.edu (B. Forget), siegela@mcs.anl.gov (A.R. Siegel).

https://doi.org/10.1016/j.anucene.2017.10.015
0306-4549/

tions relating to the usage of TRRM are also presented, including difficulties introduced by the stochastic convergence process.

2. Methods

2.1. Neutron transport

Reactor simulation aims to calculate specific properties of the core both to make revisions to the initial reactor design and to better predict its operational behavior. Two of the most important phenomena are: (1) the eigenvalue, or criticality, of the reactor, and (2) its spatial power distribution. The eigenvalue, referred to as "k-effective", determines the ratio of neutron populations between successive generations within the reactor. It therefore determines the balance between neutron production and loss. The power distribution within the reactor governs the thermal-hydraulic design considerations as well as the rate of burn-up of the nuclear fuel. The eigenvalue and power distribution are computed by numerically estimating the solution of the Boltzmann Neutron Transport Equation (Duderstadt and Hamilton, 1976, pp. 111–145; Hebert, 2009, pp. 67–186).

There are a wide variety of common transport simulation methods and accompanying applications, such as the method of discrete ordinances (PROTEUS-SN Shermon et al., 2014), Rattlesnake (Idaho National Laboratory: Rattlesnake, 2017), PARTISN (Alcouffe et al., 2005), Monte Carlo (MCNP Briesmeister, 1986, OpenMC Romano and Forget, 2013, RMC Wang et al., 2013, Serpent Leppänen et al., 2015, TRIPOLI Nimal and Vergnaud, 1990), and the Method of Characteristics (OpenMOC Boyd et al., 2014, MPACT Kochunas et al., 2013, CASMO-5 Rhodes et al., 2006), among many others. The long list is due to each method or implementation often carrying its own set of strengths and weaknesses, with various applications often excelling in different problem areas and use cases. While there are many applications already, there is still ample opportunity for new methods and applications to be useful particularly in the 3D high fidelity full core reactor simulation space. This is evidenced by the significant amount of recent research being performed in the field of high fidelity 3D MOC methods (Yamamoto et al., 2017; Zheng et al., 2017; Sanchez, 2012; Sciannandrone and Santandrea, 2013).

2.2. The Random Ray Method

The Random Ray Method (TRRM) is a recently developed hybrid neutron transport method (Tramm et al., 2017) based on the Method of Characteristics (MOC) (Prabha and Marleau, 2013; Prabha et al., 2014; Sanchez, 2012; Eklund et al., 2015). MOC solves a partial differential equation (PDE) by defining characteristic lines (or curves) along which the PDE is reduced to an ordinary differential equation (ODE). By solving the ODE along a set of discrete characteristic lines (sometimes in conjunction with ray tracing) and iterating on the initial conditions for the lines, a solution to the governing PDE can be numerically estimated. Traditional MOC applications use a cyclical deterministic quadrature (Grassi, 2007; Sciannandrone and Santandrea, 2013; Kochunas et al., 2007; Kochunas, 2013; Boyd, 2014) composed of many tracks to cover the phase space of the system evenly. In this way, the eigenvalue and distribution of neutrons throughout a reactor can be numerically estimated.

Unlike traditional deterministic MOC methods, TRRM does not use a cyclical deterministic quadrature. Rather, characteristic lines (also known as rays) sampled from a uniform random distribution in space and angle are followed through the system until termination criteria are met, in effect forming a stochastic integration quadrature. This cycle of sampling rays randomly and following them through the domain until they are terminated is repeated in an iterative manner, updating the scalar flux in each region of the computation after each iteration. Usage of the new method can allow for increases in accuracy due to accumulation of effective resolution over successive iterations, while also reducing computational storage requirements by removing the need to store deterministic quadrature data and the even more costly starting condition data (i.e., angular flux). While a small bias is introduced into the computation due to approximation of each ray's starting conditions, this bias can be mitigated through usage of a "dead zone" as discussed in Sections 4 and 2.4. Compared to other traditional multigroup deterministic methods, TRRM can:

- easily handle arbitrary 3D geometries (Tramm et al., 2017)
- provide extreme improvements in memory efficiency (Tramm et al., 2017)
- allow for a continuous angular treatment of the angular flux in the reactor, allowing for usage of continuous scattering kernels (Tramm et al., 2017)
- result in significant reductions in algorithmic complexity on some simulation problems (Tramm et al., 2017)
- map very well onto next generation supercomputing architectures due to its task based structure and ability to vectorize (Tramm et al., 2016)

A thorough description of the method and its mathematical derivation is given by Tramm et al. (2017). ARRC implements 3D, flat source TRRM in the form of a high performance and massively parallel application for use on modern supercomputing architectures.

2.3. Geometry

Fundamental to many fields of computational science is the ability to represent the geometrical structure of an object in a manner that a computer can understand. One high accuracy method that is often used in neutron transport codes is constructive solid geometry (CSG). In CSG, an object is represented exactly by definition of second order surfaces such as cylinders, spheres, and planes. Complex objects can be made by defining spaces that form the intersection, union, or complement of multiple different second order surfaces.

Many computational methods also perform "ray tracing", wherein a virtual ray is followed through a CSG geometry to determine where it intersects, reflects, and how far it travels through each region. Traditionally, ray tracing for deterministic methods to generate a deterministic quadrature is performed once during initialization of the program and stored in memory as a table for the duration of the computation. As TRRM uses a new stochastic quadrature every iteration, the same rays will never be used twice so it is not useful to precompute and store ray tracing data. As such, TRRM method uses fully on-the-fly ray tracing, which adds to the floating point computational cost of a simulation but also reduces the memory footprint and bandwidth requirements by not storing and frequently accessing a large "tracking file". The improved cache performance and lower bandwidth requirements resulting from on-the-fly ray tracing in TRRM have also been shown to outweigh the increased floating point costs, resulting in better overall performance compared to traditional deterministic MOC "precompute and store" methods as measured by the amount of time spent per angular flux integration (Tramm et al., 2017). While such performance results may vary between codes due to different ray tracing and CSG implementations as well as differing code optimization levels, it is clear at least that on-the-fly ray tracing is not prohibitively expensive. One caveat is that on-the-fly ray tracing does become relatively more computationally expensive when