PHYSOR 2004 -The Physics of Fuel Cycles and Advanced Nuclear Systems: Global Developments
Chicago, Illinois, April 25-29, 2004, on CD-ROM, American Nuclear Society, Lagrange Park, IL. (2004)

AGENT Code: Open-Architecture Analysis and Configuration of Research Reactors – *Neutron Transport Modeling with Numerical Examples*

Tatjana Jevremovic, Hyun Chul Lee, Kevin Retzke, Yefei Peng, and Mathieu Hursin
Purdue University, School of Nuclear Engineering, 400 Central Drive, West Lafayette, IN 47907, USA

Two papers presented in this Conference focus on our recent development of the advanced computational environment for the analysis and configuration of University Research and Training Reactors (URTRs). This computational environment is intending to assist researchers and educators with tools for an open-architecture neutronic analysis and configuration of the URTRs (to optimize experiments, fuel locations for flux shaping, detector selection and configuration). Such computational environment does not currently exist. The method of characteristics based computer code, AGENT, will revolutionize the way in which URTR research is planned, deployed, and analyzed through the so called "*virtual reactor environment*." In this paper we describe the robustness, efficiency and accuracy of the AGENT method illustrated by two representative numerical examples.

KEYWORDS: neutron transport, method of characteristics, research reactors, geometry representation, R-functions

1. Introduction

It is the desire of researchers and educators throughout the world to have the ability to quickly and easily model University Research and Training Reactors (URTRs), and to be able to perform "virtual" experiments and demonstrations without the hassles of working with complex code systems. Two papers [1] presented at this Conference focus on the recent development of an advanced computational environment intending to provide researchers and educators with tools for an open-architecture neutronic analysis and configuration of URTRs. This environment will allow users to optimize experiments, test fuel configurations, and provide "virtual" demonstrations. A computational environment such as this does not currently exist. The centerpiece of this environment is the method of characteristics based computer code, Arbitrary Geometry Neutron Transport (AGENT), which will revolutionize the way URTR research is planned, deployed, and analyzed through the so-called "*virtual reactor environment*."

2. Neutron Transport Modeling

AGENT solves the integral transport equation using the Method of Characteristics (MOC) in a user-defined number of energy groups in a geometry represented via the flexible theory of R-functions [1, 2]. The combination permits accurate, efficient, and fast particle transport modeling in complex and heterogeneous geometrical domains. The MOC allows a full treatment of highly heterogeneous systems with a large number of energy groups and detailed neutron path map (satisfactory number of directions, and fine spatial resolution).

2.1 Geometrical Representation

A general and efficient way of representing complex three-dimensional geometries is a problem that spans many disciplines, from CAD to computer graphics to particle transport. The R-function modeler used to represent complex domains through the combination of simple primitives into a single analytical equation allows great flexibility in hierarchical organization of the arbitrary geometry. The modeler is as general as the typical Monte Carlo approach, but much simpler and importantly faster. The R-function method makes no essential difference between elementary domains (primitive shapes such as boxes and cylinders) and complex domains (those generated by the combination of elementary ones), as both are expressed as continuous analytic functions.

The steps in generating a complex domain in AGENT are simple: the user specifies the primitive objects that will form the domain and Boolean operations to combine these primitives into the complex domain. When later referencing the complex domain, the code does not need to know what primitive objects form the domain (except in the special ray tracing case described below). The nature of R-functions is that the domain function representing a given geometry will evaluate positive for a point within the domain, negative for a point outside the domain, and zero for a point on the domain boundary. Therefore, to test if a point is within a certain object, the code merely has to evaluate the function at that point.

This modeler permits automatic generation of *flat-flux* zones that are required for accurate MOC calculations. Spatial discretization of the domain (and boundary conditions) is an underlying principle of most modern methods, such as finite element, finite difference, or MOC. Spatial discretization of static complex geometrical domains is often a tedious task and source of error, while for a time varying geometry it represents insurmountable difficulties. By contrast, the R-function method discretizes not the domain but the underlying spatial domain function. This allows for separation of geometric information from numerical procedures and results in solution techniques independent of numerical solutions, [1]. To illustrate this we use the simplest geometry unit of a typical reactor lattices, a square unit cell consisting of fuel pin and moderator region. "Figure 1" shows a two dimensional moderator region of the unit cell domain and corresponding function that vanishes on the boundary of the domain. This example illustrates how constructive solid geometry representation of any geometric domain can be translated into corresponding implicit function, [1]. It also suggests that the function can be used to generate any type of mesh subdivision and be used to evaluate the domain surfaces. This is underlying mathematical principle of the geometry description and ray tracing techniques developed in the AGENT code system.

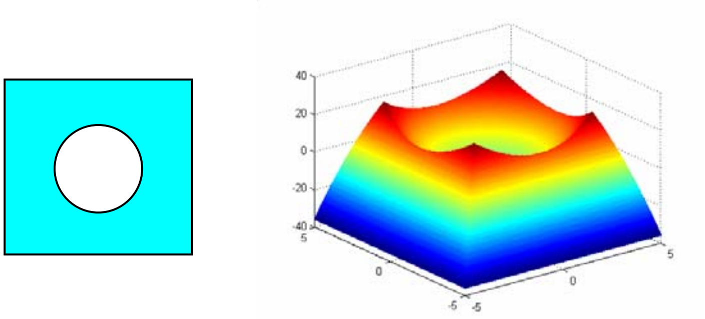


Fig. 1: Two dimensional moderator region of the reactor unit cell and domain function that vanishes on its boundary

2.2 Ray Tracing Technique

As with geometrical representation, ray tracing is a necessary function that must be performed in many disciplines. The R-function method provides two elegant solutions to the ray tracing problem, both employed in AGENT. The first method involves taking discrete steps along the ray, testing the domain functions of the objects of interest at each point. As mentioned above, if the point is outside the object, the function will evaluate negative, while if the point is within the object, the function will evaluate positive. Therefore, if the signs of the function at two consecutive points are opposite, a boundary has been crossed. Then it is simply a matter of “homing in” on the boundary, i.e. the point along the ray where the function evaluates within a certain tolerance of zero. However, this method can be time-consuming, and is error-prone in certain geometries that contain very small objects. It is conceivable that an object could be missed due to the step size being larger than the overall size of the object. The solution to this would be to reduce the step size, which would greatly add to the computational time required. In cases such as this it is desirable to use the second ray tracing method. As the analytic equation representing each primitive object is known, it is therefore possible to derive an analytic solution for the ray-object intersection(s). However, except in the simplest of geometries only a fraction of these intersections are along the complex domain boundary. It is therefore necessary to test each point of intersection with the complex domain function, with any point evaluating to other than zero (within tolerance) being omitted. This method provides a high degree of accuracy in determining the object boundaries; however, it requires that the ray tracing module have information on the primitive domains making up the complex object, which removes the separation required for a truly geometry-independent system. A preliminary comparison of these two ray tracing methods shows that the geometry dependant method may be as, if not more, efficient than the independent method. The method most appropriate for a given problem is very dependant upon the problem specifics.

2.3 Computational MOC

In its stand-alone form AGENT code solves eigenvalue equations for the eigenvalue, volumetric flux and reaction rates. The multigroup energy cross sections are obtained from the front-end code routines. At the moment, the cross sections are generated by the lattice code HELIOS v-1.8, [3]. The HELIOS output is then processed to be used directly by AGENT. The number of energy groups involved in an AGENT calculation is not limited. The spatial discretization module generates characteristic particles’ paths in a set of selected number of angles. Isotropic neutron propagation along the paths for each set of discrete angles starts at the most outer boundary and reflects back into the system following the user selected boundary conditions. The iterative process consists of outer ("fission source") iterations and inner ("scattering source") iterations with the spectrum rebalancing and coarse mesh finite difference (CMFD) acceleration [4-6]. After every MOC sweep, the entire problem domain is homogenized and the homogenized point problem is used to calculate the group scalar flux. The ratio of the group scalar flux from the homogenized point problem to the space-integrated group scalar flux from the MOC calculation is multiplied to the zone wise group scalar flux. The rebalanced zone wise group scalar fluxes are used to calculate the zone wise source terms for next iteration. In CMFD acceleration technique, all pin cells are homogenized and a set of current-corrected, pin-cell based, coarse mesh, finite-difference-like diffusion equations (CMFD equations) are constructed, in which the net currents of MOC calculation at the interfaces of the pin cells are preserved with the help of the current correction factor. The ratios of the group scalar flux for pin cells from the CMFD equation to the space-integrated group scalar flux for pin cells from the MOC calculation are multiplied to the zone wise group scalar flux. The rebalanced zone wise group scalar fluxes are used to calculate the zone wise source terms for next iteration. “Figure 2” illustrates the process followed by the CMFD solver in AGENT.

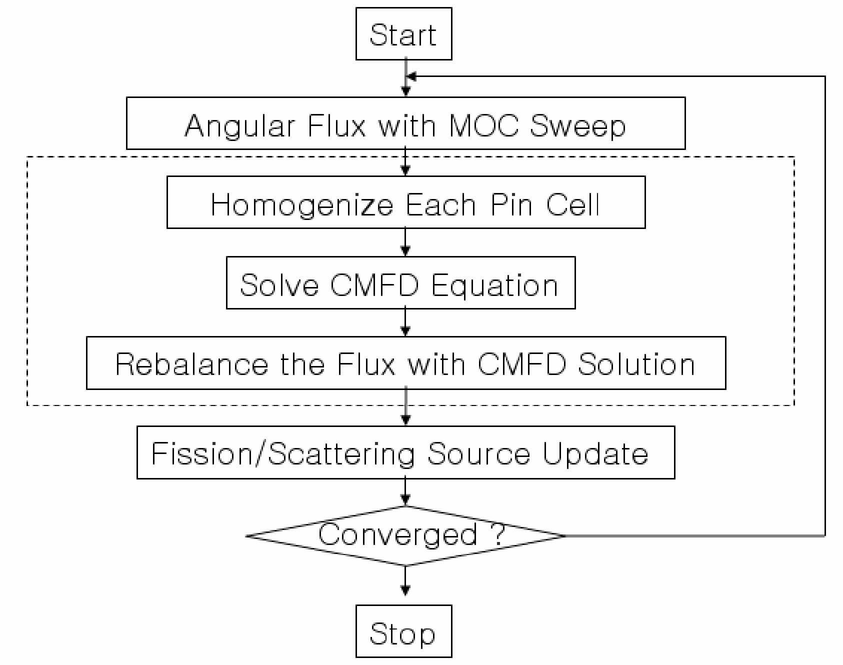

Start
Angular Flux with MOC Sweep
Homogenize Each Pin Cell
Solve CMFD Equation
Rebalance the Flux with CMFD Solution
Fission/Scattering Source Update
Converged ?
Stop

Fig. 2: Scheme of the CMFD acceleration technique.

One test has been performed to compare the CMFD method to the spectrum rebalancing technique alone. The problem was derived from a 17×17 PWR assembly with 268 fuel pins and 21 guide tubes. Numerical test shows that the CMFD technique is about 8 times faster than the spectrum rebalancing one, as shown in Table 1.

Table 1. Comparison of Acceleration Techniques

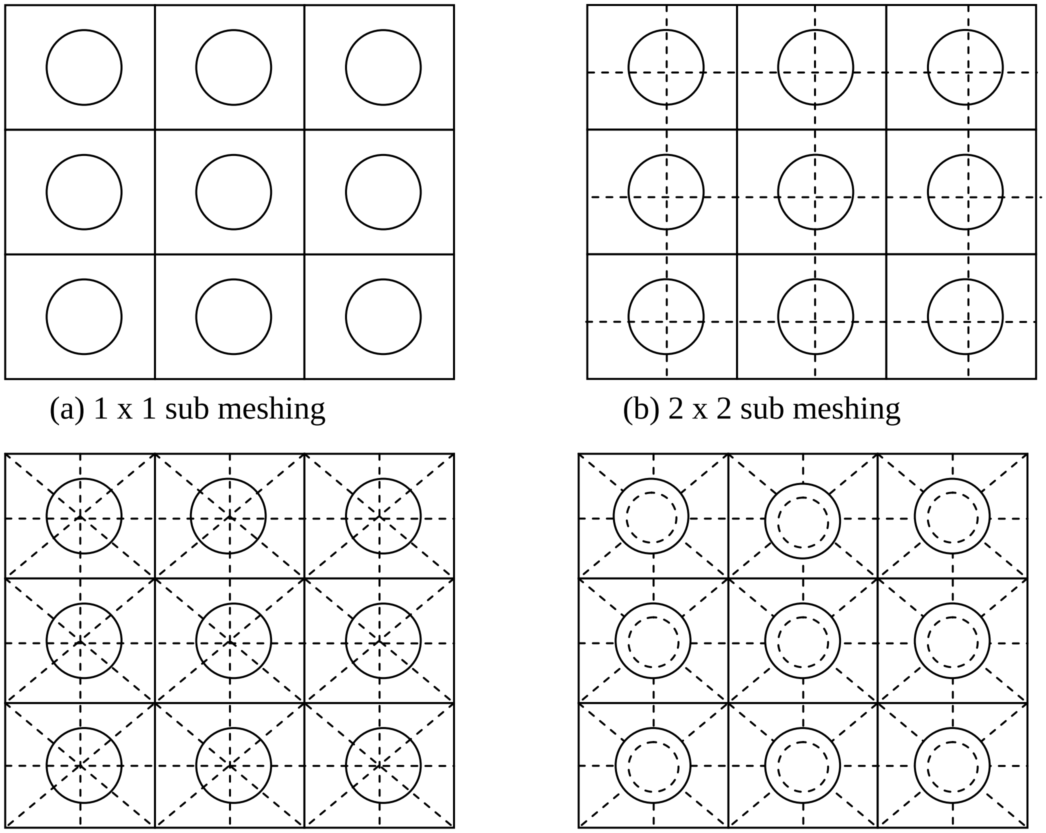
Method	**Iterations**
Spectrum rebalancing	125
CMFD	16

3. Numerical Examples

The AGENT code is optimized for robustness, simplicity, accuracy, and efficiency. It supports a full treatment of neutron transport in highly heterogeneous geometries such as research reactors. The robustness of the geometrical module is assured through the sequential generation of the layers of geometry and automatic submeshing based on the R-function solid modeler. The simplicity of geometry description and selection of parameters for accurate treatment of neutron propagation is achieved through the hierarchical organization of simple primitives into complex domains. The accuracy is comparable to Monte Carlo codes and is obtained by following neutron propagation through real geometrical domains. The efficiency is maintained through a set of acceleration techniques introduced in all important calculation levels. The following two examples are selected to show the accuracy and efficiency of the code, but also the flexibility in selecting subdivision patterns independent on the geometry type. First benchmark problem is a simple 3x3 array of fuel pins arranged in square lattice and benchmarked against the MOC DRAGON code, [7]. The second example is Purdue University research reactor assembly that is benchmarked against MCNP.

3.1 3x3 Assembly Unit

This simple test is used to analyze the speed and accuracy of fine mesh used in AGENT code combined with the fine spatial discretization, and to compare with DRAGON code. “Figure 3” shows three different types of sub-meshing used in AGENT. DRAGON code does not support octant sub-meshing. The k-inf and CPU times are listed in Table 2. It can be seen that the AGENT code is more superior in submeshing flexibility showing shorter running times for higher zone numer and increased complexity of the domain.



(a) 1 x 1 sub meshing

(b) 2 x 2 sub meshing

(c) Octant sub meshing

(d) Octant with subdivided pins

Fig. 3 Different sub meshing in AGENT code

3.2 PUR-1 Assembly Unit

The second benchmark example is one assembly of the PUR-1 (Purdue University Reactor 1), shown in “Figure 4.” The assembly has been modeled with both AGENT and MCNP code. As can be seen, the assembly, while relatively simple, does contain many small zones, representing thus a challenge for majority of computational models.

Table 2 k-inf and CPU time comparison of DRAGON and AGENT

# of azimuthal angles	Ray Separation (cm)	Mesh 1×1		Mesh 2×2		Mesh octant	Mesh Octant w/ Subdivided pins
		AGENT	DRAGON	AGENT	DRAGON	AGENT	AGENT
20	0.2	1.403261	1.403263	1.403261	1.403264	1.403261	1.403261
	0.1	1.403261	1.403263	1.403261	1.403264	1.403261	1.403261
	0.05	1.403261	1.403263	1.403261	1.403264	1.403261	1.403261

# of azimuthal angles	Ray Separation (cm)	Mesh 1×1		Mesh 2×2		Mesh Octant	Mesh Octant w/ Subdivided pins
		AGENT	DRAGON	AGENT	DRAGON	AGENT	AGENT
20	0.2	26s	14.1s	31.5s	44.1s	31.9s	35.8s
	0.1	26.1s	27.1s	31.0s	1m 54.9s	32.0s	37.4s
	0.05	27.3s	72.5s	32.3s	3m 43.9s	33.4s	40.3s

With the number of azimuthal angles set to 20 and a ray separation of 0.01 cm, AGENT gives a k-infinite of 1.11110 (CPU time 17s), compared to MNCP value of 1.11229 with a standard deviation of 0.00024 (CPU time 58 minutes). The relative error in AGENTs k-infinite is only 0.1%, with the CPU time 200 times shorter compared to MCNP calculation.

For further comparison, we examine the thermal neutron flux and absorption rate (0.057eV < E < 0.25 eV) at the cross-section area indicated by the dashed line in “Figure 3.” “Figure 5a” shows the flux as determined by AGENT, and “Figure 5b” shows the flux obtained by MCNP. In both cases, the flux tallies have been normalized such that the integral over the geometry domain is unity. The maximum discrepancy comparing to MCNP values in thermal neutron flux is below 2%, while in fast energy region is below 1%. “Figure 6a and 6b” show the absorption rates as determined by AGENT and MCNP respectively. As can be seen, the results of AGENT compare favorably with those obtained using the Monte Carlo method.

4. Conclusion

The current state of development of codes that address the flexible neutron transport modeling in research reactor core geometries is extremely limited. The common method for accurate modeling of particle interactions is a Monte Carlo technique. While it provides great geometrical flexibility and high calculational accuracy, it suffers from long computational times, cumbersome modeling requirements, tedious and error-prone process of generating input and knowledgeable evaluation of numerically produced statistical errors.

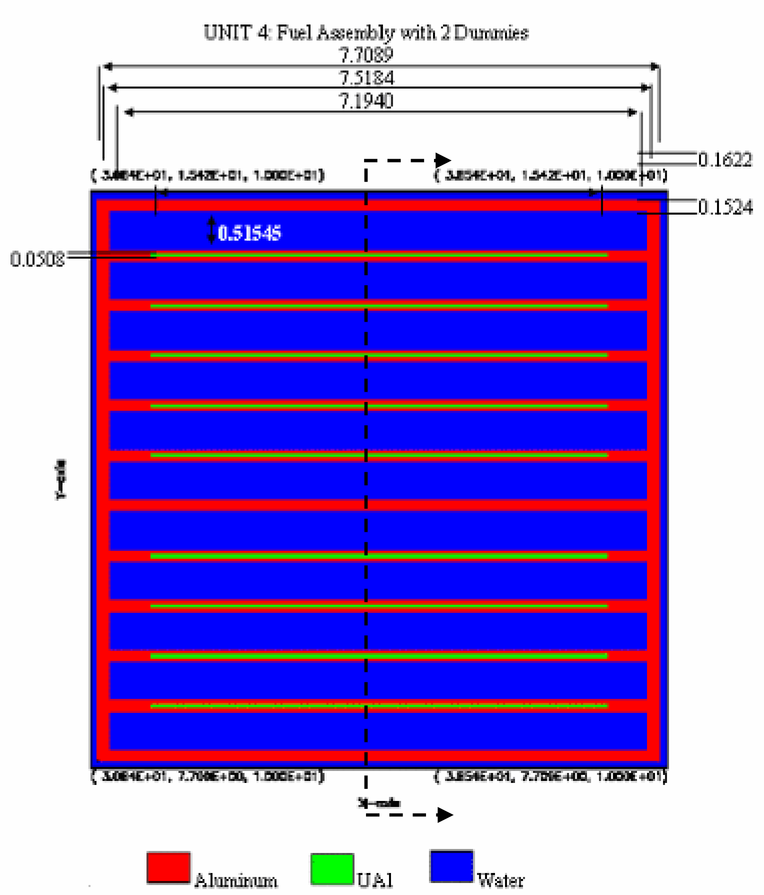

UNIT 4: Fuel Assembly with 2 Dummies
7.7089
7.5184
7.1940
0.1622
(3.084E+01, 1.542E+01, 1.000E+01)
(3.854E+01, 1.542E+01, 1.000E+01)
0.1524
0.51545
0.0508
Y-axis
(3.084E+01, 7.700E+00, 1.000E+01)
(3.854E+01, 7.700E+00, 1.000E+01)
X-axis
Aluminum
UA1
Water

Fig 4 Two dimensional cross section of the PUR-1 unit 4

Highly accurate modeling of neutron trajectories within heterogeneous systems and within reasonable computation runtimes is possible using the MOC. The neutron random walks of Monte Carlo are replaced with a large number of parallel paths in a number of discrete directions in MOC to calculate flux and source data. We combine the MOC with the powerful mathematical tool of R-function theory to achieve the flexibility and accuracy of Monte Carlo codes, yet retain the speed and simplicity of deterministic codes. The AGENT code system will comprise a "*virtual reactor*" where users can configure core fuel arrangements, reactor experiments, and advanced reactor configurations. This will permit the community of URTR operators and experimenters to run a virtual reactor and monitor reactor behavior and performance. It will also support distance educational and outreach programs where remote access to the code suite will allow educators to perform virtual reactor demonstrations.

Acknowledgements

Research supported by Grant Number 2402-PU-DOE-4423 under the Innovations in Nuclear Infrastructure and Education (INIE) Program of the US Department of Energy.

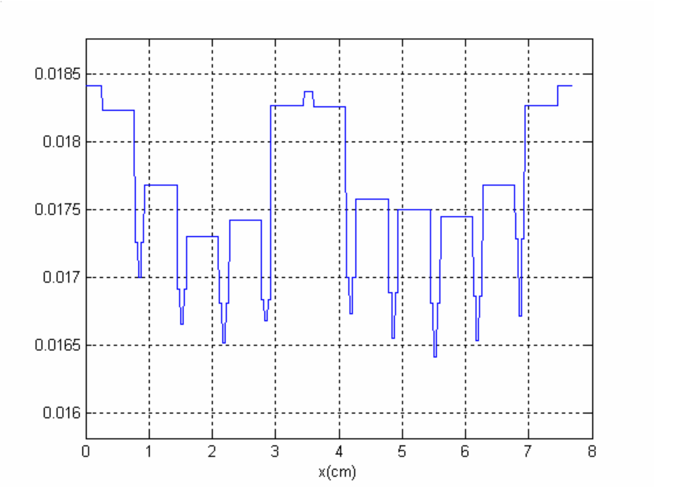


Fig. 5a. AGENT thermal neutron flux (group 6)

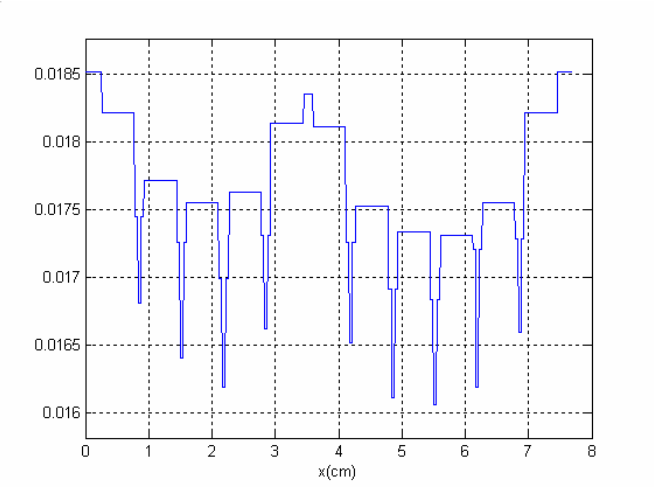


Fig. 5b. MCNP thermal neutron flux (group 6)

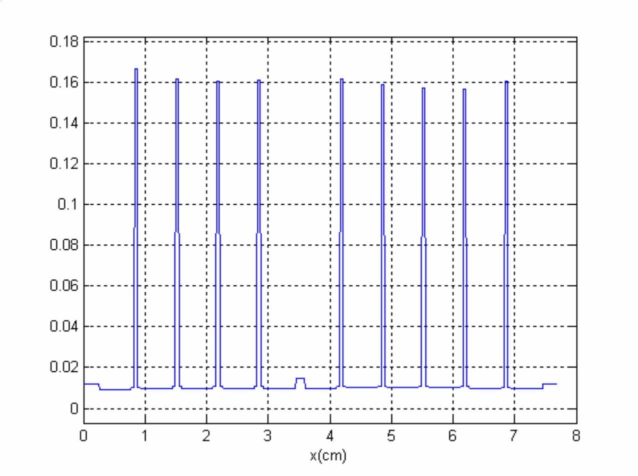


Fig. 6a. AGENT absorption rate (group 6)

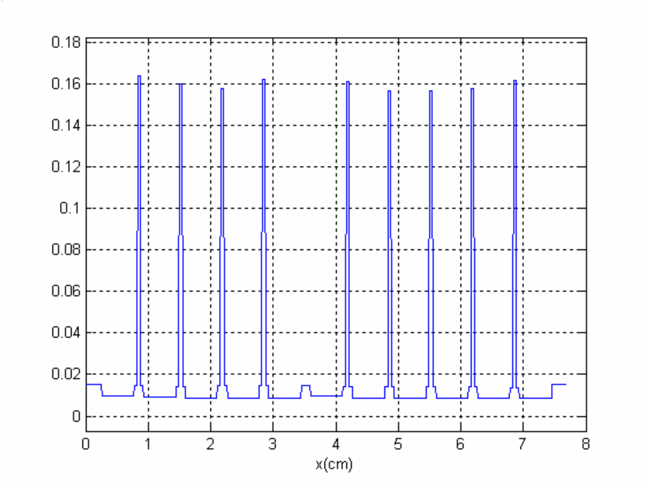


Fig. 6b. MCNP absorption rate (group 6)

References

1) D. ANDRITSOS and T. JEVREMOVIC, “AGENT Code: Open-Architecture Analysis and Configuration of Research Reactors – Graphic Tools”, This Conference

2) T. JEVREMOVIC, T. ITOH, and Y. INABA, “ANEMONA : Multiassembly Neutron Transport Modeling”, Ann. Nucl. Energy, 29, Pg. 2105-2125, (2002)

3) Scandpower, I., *HELIOS Documentation, FMS-The Scandpower Fuel Management System*. 199

4) G. S. LEE, N. J. Cho, and S. G. Hong, “Acceleration and Parallelization of the Method of Characteristics for Lattice and Whole Core Heterogeneous Calculation,” Proc. Int. Mtg. Advances in Reactor Physics and Mathematics and Computations into the new Millennium, Pittsburgh, Pennsylvania, May 7-12, 2000, CD-ROM, American Nucl. Soc. (2000)

5) J. Y. CHO, H. G. JOO, K. S. KIM, S. Q. ZEE, “Cell Based CMFD Formulation for Acceleration of Whole-Core Method of Characteristics Calculations,” J. Kor. Nucl. Soc., 35, pp. 250-258 (2002)

6) H. G. JOO, J. Y. CHO, H. Y. KIM, S. Q. ZEE, M. H. CHANG, “Dynamic Implementation of the Equivalence Theory in the Heterogeneous Whole Code Transport Calculation,” PHYSOR 2002, Seoul Korea, October 7-10, 2002, CD-ROM, American Nucl. Soc. (2002)

7) G. MARLEAU, A.H.a.R.R., *A User Guide for DRAGON*. 2000, Institut de genie nucleaire