Computation of IAEA-3D PWR Benchmark with MATLAB PDE Toolbox

Abiodun Ajirotutu, Xue Yang

Texas A&M University-Kingsville: 700 University Blvd, Kingsville, TX 78363, xue.yang@tamuk.edu

INTRODUCTION

In designing PWR cores, some parameters such as multiplication factor(k_{eff}), neutron flux distribution and assembly power must be calculated in order to make some design justification. To compute these parameters, various computational tools with different mathematical approach have been used. To validate the effectiveness of these tools, the IAEA 3-D PWR core result from VENTURE Code block is a crucial benchmark for making judgment.

VENTURE code block is a computational tool designed for solving multi- neutron-energy group and multi-dimensional neutronics problems. It applies Finite Difference Diffusion or a simple P1 theory approximation to neutron transport. With VENTURE code, IAEA-3D PWR core problem was solved to determine k_{eff}, neutron flux distribution and power density. This result is now one of the references for comparing.

In this publication, MATLAB PDE Toolbox is being investigated for computing of three-dimensional neutronics problems. PDE Toolbox is a partial differential equation solver based on the Finite Element Method [3], it uses continuous functions that are linear on each triangle of the mesh (piecewise linearity). It has functions for solving structural mechanics, heat transfer, and general PDEs. It is also applied in linear static analysis, modeling of structural dynamics and vibration. Other applications of PDE Toolbox are for solving standard problems such as diffusion, electrostatics as well as custom PDEs and eigenvalue problems.

This work validates that using MATLAB PDE Toolbox for computing neutronics problems is effective since its k_{eff} and the power density results for IAEA 3-D core compared fairly well with that from VENTURE Code.

DESCRIPTION OF IAEA BENCHMARK

The IAEA 3-D is a light water reactor (LWR) with a core configuration of 177 fuel assemblies, 9 fully rodded assemblies, 4 partially rodded fuel assemblies, and 64 reflector surrounding the core. The height of the core is 380 cm, which includes a 340 cm high fuel assembly along with a 20 cm thick axial reflector at the top and bottom as shown in Fig.1 and Fig.2. Besides, Table 1 provides the lists the materials and their cross sections. [2]

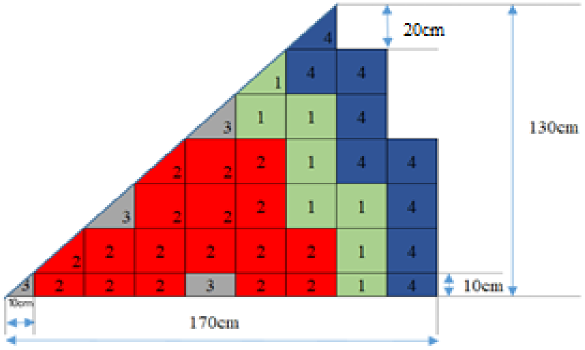

Fig. 1. Horizontal cross-section of the IAEA 3-D benchmark.

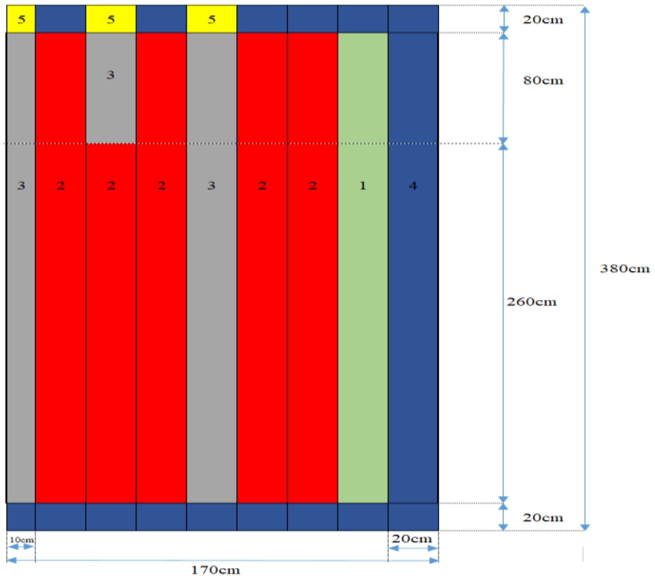

Fig. 2. Vertical cross-section IAEA 3-D benchmark.

Table I. Two- Group Cross Sections for IAEA Core

Material	Material	D_g	Σ_g^a	$\nu\Sigma_g^f$	$\Sigma_{2\text{-}1}$
1	Fuel 1	1.500	0.010	0.000	0.000
		0.400	0.080	0.135	0.020
2	Fuel 2	1.500	0.000	0.000	0.000
		0.400	0.085	0.135	0.020
3	Fuel 1 + Rod	1.500	0.010	0.000	0.000
		0.400	0.130	0.135	0.020
4	Reflector	2.000	0.000	0.000	0.000
		0.300	0.010	0.000	0.040
5	Reflector + Rod	2.000	0.000	0.000	0.000
		0.300	0.055	0.000	0.400

SOLUTION METHOD

The equation 1 and 2 below are the two-group diffusion equations for the fast and thermal reaction to be solved in this exercise.

$$-\nabla D_1 \nabla \Phi_1 + (\boldsymbol{\Sigma_1^a} + \boldsymbol{\Sigma_{1\to2}^s})\Phi_1 = \frac{\nu\boldsymbol{\Sigma_1^f}\Phi_1 + \nu\boldsymbol{\Sigma_2^f}\Phi_2}{k} \qquad (1)$$

$$-\nabla D_2 \nabla \Phi_2 + \boldsymbol{\Sigma_2^a}\Phi_2 \; = \; \boldsymbol{\Sigma_{1\to2}^s}\Phi_1 \qquad (2)$$

To solve these equations with MATLAB PDE Toolbox, their variables must first be redefined to suit the PDE Toolbox Generic Systems equation format shown in equation 3 and Table II.

$$-\nabla(c\,\nabla u) + au = \lambda du \qquad (3)$$

Table II. Defined variables:

MATLAB PDE Toolbox	*c*	*u*	*a*	*λ*	*d*
2-Group Neutron Equation	*D*	*Φ*	$\boldsymbol{\Sigma_g^a}$	*k*	$\boldsymbol{\Sigma_g^f}$

The procedure for solving the *k*, flux *Φ*, and power distribution includes two parts, which are the pre-processing and post-processing steps.

Pre-processing

For the first part, the geometry of the 1/8th IAEA 3-D core was developed using SOLIDWORKS CAD software and the model is shown in Fig. 3. The generated model was saved in the stereolithographic (STL) file because PDE Toolbox can only read STL format [3].

MATLAB algorithm, consisting of a main script and some assisting functions, was developed to provide PDE Toolbox with specific parameters required for solving the two-group neutron diffusion equation. These parameters include the number of PDEs to be solved, the file name of the geometry to be imported, the cross sections of the materials, coordinates of each lattice, boundary conditions to be applied, coefficients of the PDEs, plot command and total number fuel lattice required to solve for the k_{eff} and normalized assembly power.

The main script imports the model into MATLAB interface and provides all parameters required to solve the problem. Additionally, this script commands other functions to perform their designated tasks. One of the functions creates a table containing the coordinates of the lattice. These coordinates subsequently assist other functions to specify the cross sections for each material in their corresponding subdomain (Fig. 1 and Fig. 2). The material cross sections [1] represent the coefficients of the PDEs to be solved. The last function computes the normalized assembly power using Gauss Quadrature numerical integration method. This is function is further discussed in the post-processing

It is essential to specify certain boundary conditions. For the 1/8th IAEA 3-D, there are two different conditions across the boundaries: the vacuum and the reflective conditions. The vacuum boundary condition applies to the external surfaces. The neutron current on outside boundaries meet the condition of equation 6 below, which is a simplification of equation 4 and 5. Hence, the external surfaces represented in Fig. 3 tagged F3, F4, F5, F6, F7, F8 and F9 have their boundary conditions set to Neumann and their parameters defined according to equation 6.

$$\frac{1}{\phi}\frac{\partial\phi}{\partial n} = \frac{-1}{d} \qquad (4)$$

$$d \; = \; 0.7104(3D) \qquad (5)$$

$$D\frac{\partial\Phi}{\partial n} \; = 0.46922\Phi \qquad (6)$$

$$J = -D\frac{\partial\phi}{\partial n} = 0 \qquad (7)$$

On the other hand, a mirror boundary condition was applied to the inner surfaces. The condition across these surfaces is specified based on the assumption that neutron current is zero at these surfaces, as represented in equation 7. Thus, the inner surfaces tagged F1 and F2 boundary conditions were set to Neumann and their parameter stated according to equation 7. Afterward, the model was discretized with a tetrahedral mesh size of 5cm and computed for the solution of $k_{\text{-}eff}$ and the group 1 and 2 neutron fluxes plot.

Post-Processing

In this phase, a MATLAB function was developed to handle the computation of the normalized assembly power. The algorithm, written based on Gauss Quadrature numerical integration method, computes fission reaction rate in each square lattice. Each square was divided into eight triangles, and the fission rate across each triangle were interpolated by taking seven integration points according to 2-D Gauss quadrature method [5].

Finally, nineteen different vertical planes were selected based on the 1-D Gauss Quadrature integration method, and their corresponding power distribution results were interpolated to generate the 3-D normalized assembly power.

RESULTS

The result of the k_{eff} and the normalized assembly power and their comparisons with VENTURE code are shown in Table III. The data in color red are negative % error and others in black are the positive ones. Also, shown in Fig. 5 and Fig. 6 are the group-1 and group-2 of the 3-D neutron fluxes.

The k_{eff} computed from the PDE toolbox is 1.02916, which is a deviation of 40 pcm compared to the reference. For the normalized assembly power, Table III and Fig.4 provides the % error relative to the reference (“Extrapolated” results tabulated in Table 3 [1]). The square lattice numbering in Fig.

4 was adopted from the reference [1]. Lastly, the Root Mean Square (RMS) calculated for the normalized assembly power is 0.01063.

The MATLAB algorithm was ran on a “Dual Intel E5-2699 v3 (18 cores *2, 36 threads * 2) with 384GB memory, 2 TB +6 TB Hard drive, Dual NVidia Tesla P110”. A total of 3,017,700 meshes was generated and the runtime of the solution was 38,726 seconds.

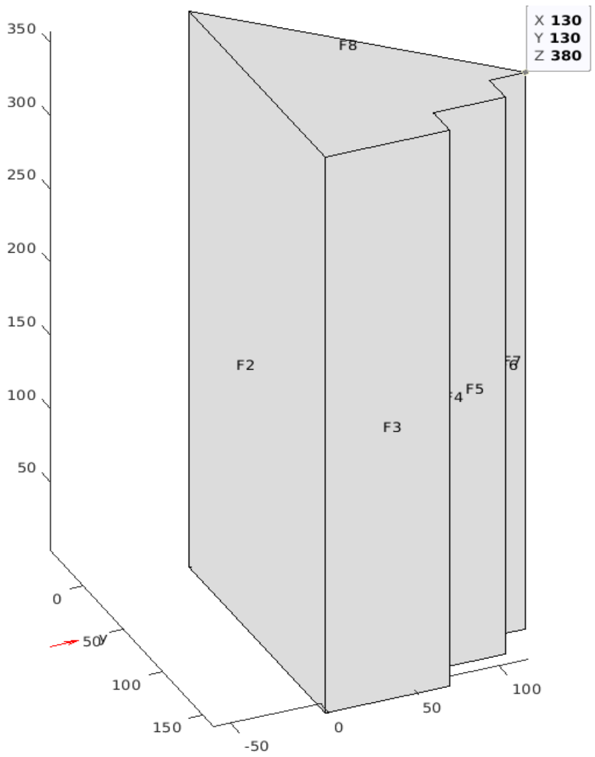

Fig. 3. 1/8 IAEA 3-D model [7].

Table III. Comparison of MATLAB assembly power with VENTURE [1, 6].

K-eff:			Ref:		0.5970		
Ref:	1.0296		Matlab		0.6032		
Matlab:	1.0292		% error		1.0385		
% error	0.0418			0.4760	0.7000	0.6110	
				0.4932	0.7040	0.6121	
				3.6134	0.5714	0.1800	
			1.1780	0.9720	0.9230	0.8660	
			1.1756	0.9741	0.9269	0.8669	
			0.2037	0.2160	0.4225	0.1039	
		1.3680	1.3110	1.1810	1.0890	1.0000	0.7110
		1.3625	1.3041	1.1795	1.0909	0.9991	0.7057
		0.4020	0.5263	0.1270	0.1745	0.0900	0.7454
	1.3970	1.4320	1.2910	1.0720	1.0550	0.9760	0.7570
	1.3840	1.4198	1.2837	1.0714	1.0566	0.9776	0.7547
	0.9306	0.8520	0.5655	0.0560	0.1517	0.1639	0.3038
0.7290	1.2810	1.4220	1.1930	0.6100	0.9530	0.9590	0.7770
0.7411	1.2701	1.4110	1.1887	0.6290	0.9560	0.9617	0.7758
1.6598	0.8509	0.7736	0.3604	3.1148	0.3148	0.2815	0.1544

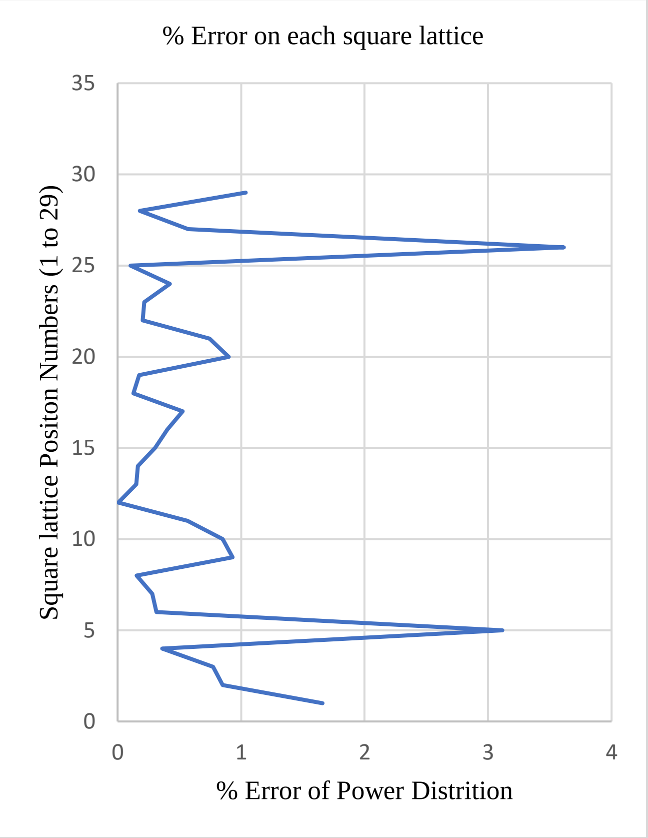

Fig.4 Percentage Error on Each Square Lattice

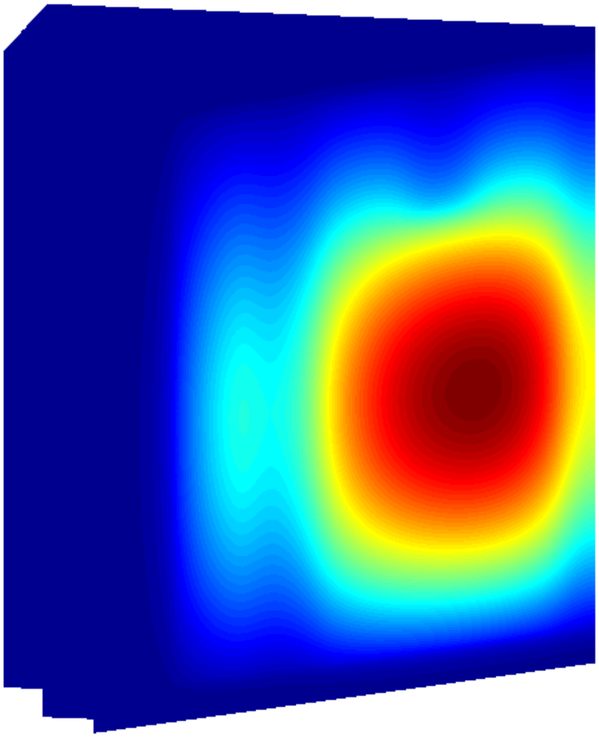

Fig. 5. Group 1 flux 3-D view.

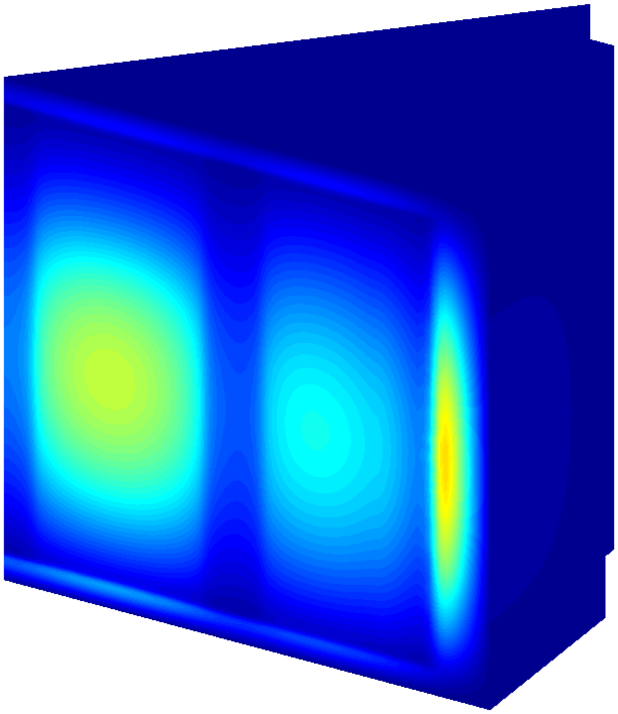

Fig. 6. Group 2 flux 3-D view.

The mesh sizes 8cm,7cm 5cm and 4.5cm were used for mess convergence test. The result converged and was very close to the reference at 5cm. At 4.5cm the duration was 4 times the duration for mesh size 5cm and with little or no change in the result.

CONCLUSION

The IAEA 3-D PWR core has been successfully modeled with MATLAB PDE Toolbox. The k_{eff} and the normalized assembly power were successfully estimated within a reasonable runtime and the results also satisfactorily match the references. The results validate that MATLAB PDE toolbox is effective and accurate in modeling PWR cores.

NOMENCLATURE

∇= gradient
D_g = diffusion coefficient of energy group g
$\Sigma_{a,g}$ = absorption cross section of energy group g
$\Sigma_{s,g\rightarrow G}$ = down scattering cross section: from group g to G
$\Sigma_{s,G\rightarrow g}$ = up scattering cross section from: normally zero
χ_g = fraction of newly borne neutrons in each energy group
k = multiplication factor
v = average number of neutrons released per fission reaction
Σ_{fg} = fission reaction for group g
Φ_g = neutron flux for energy group g
λ_{tr} =mean free path of neutron in air
d = extrapolation distance
$\partial\phi/\partial n$ = normal derivative
J =neutron current

ACKNOWLEDGMENT

This research is partially supported by Nuclear Power Institute (NPI) and Office of Naval Research (ONR), award No.: N00014-18-1-2732.

REFERENCES

1. Argonne National Laboratory "Argonne Code Center: Benchmark Problem Book", ANL-7416 Supplement 2. 9700 South Cass Avenue Argonne, Illinois 60439. American Nuclear Society (1977).
2. D.Vondy, VENTURE: A Code Block for Solving Multigroup Neutrnic Problems,ORNL-5062,OAK Ridge National Laboratory, Tennessee 37830 (1975)
3. MathWorks, "Partial Differential Equation Toolbox User's Guide", p.2.41-100, MathWorks Inc. Apple Hill Drive Natick, MA (2019).
4. X. Yang, "MATLAB PDE Toolbox for Neutron Diffusion Equation", ANS-2019, American Nuclear Society (2019).
5. J. Reddy, "An Introduction to the Finite Element Method" p.591-618, 3rd Edition,McGraw-Hill Education (2006).
6. A. Soliman "Validation of COMSOL Multiphysics for PWR Power Distribution via 3D", COMSOL Conference in Rotterdam (2017).
7. Dassault Systemes, "Introducing SolidWorks" p.1-28, Dassault Systemes SolidWorks Corporation, Wyman Street, Waltham, (2015).
8. R. Murray, "An Introduction to the Concepts, Systems, and Applications of Nuclear Processes, p.101-115", 8th Edition, Cambridge MA, (2020).