Application of MATLAB PDE Toolbox for the IAEA-3D PWR Benchmark

Abiodun Ajirotutu, Xue Yang

Texas A&M University-Kingsville: 700 University Blvd, Kingsville, TX 78363, xue.yang@tamuk.edu
https://dx.doi.org/10.13182/T122-32626

INTRODUCTION

The last few decades have witnessed significant progress in the area of modeling Pressurized Water Reactor (PWR) cores. In designing PWR cores, some parameters such as multiplication factor (k_{eff}) and assembly power must be computed in order to make certain design decisions. To compute these parameters, various computational tools with different mathematical approaches have been used. Some of these computational tools include VENTURE Code block, VANCER, COMSOL Multiphysics and CITVAP [1, 2, 3, 4]. To validate the accuracy and effectiveness of untested computing tools, the IAEA 3-D PWR core result from VENTURE Code block is considered as one of the crucial benchmarks to match.

VENTURE code block is a computational code designed for solving multi- neutron-energy group and multi-dimensional neutronics problems. It applies Finite Difference Diffusion or a simple P1 theory approximation to evaluate neutron transport equations [2]. With VENTURE code, IAEA 3-D PWR core's neutron diffusion equations were solved to determine k_{eff}, neutron flux distribution and power density [1]. These results are referred to validate the accuracy and effectiveness of the new computing methods.

In this manuscript, MATLAB PDE Toolbox is investigated for computing three-dimensional neutronics problems. The PDE Toolbox is a partial differential equation solver that employs the Finite Element Method and uses piecewise linear continuous functions on each mesh. The function can also solve structural mechanics, heat transfer, and general PDE problems. Other applications of PDE Toolbox are in solving standard problems such as diffusion, electrostatics as well as customized PDEs and eigenvalue problems [5].

Since MATLAB is a user friendly, popular and easily accessible software, it offers a great advantage to nuclear scholars and professionals to reliably compute the neutron transport equations. This paper studies the accuracy and effectiveness of MATLAB PDE Toolbox for computing neutronics problems. The k_{eff} and the power distribution results for IAEA 3-D obtained from MATLAB reasonably fitted with that from VENTURE Code.

DESCRIPTION OF IAEA BENCHMARK

The IAEA 3-D core 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 assemblies surrounding the core [1]. 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. The IAEA 3-D core contains three different fuel types. There are four control rods that can be partially or fully inserted in each of the assemblies with type-3 fuel has shown in the Fig. 1. Table I also provides a summary of the materials for the core and their cross sections. The colors and numbers correspond to different materials given in the Table I.

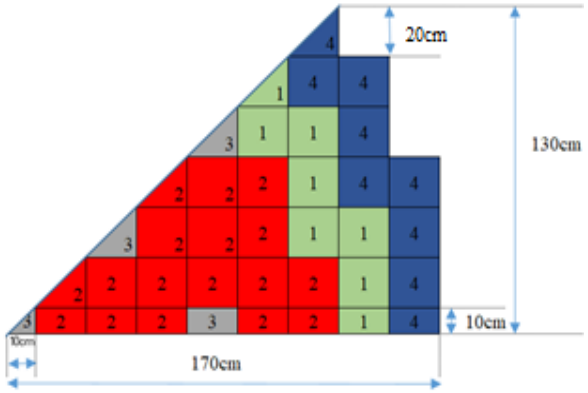

Fig. 1. Horizontal cross-section of the IAEA 3-D benchmark.

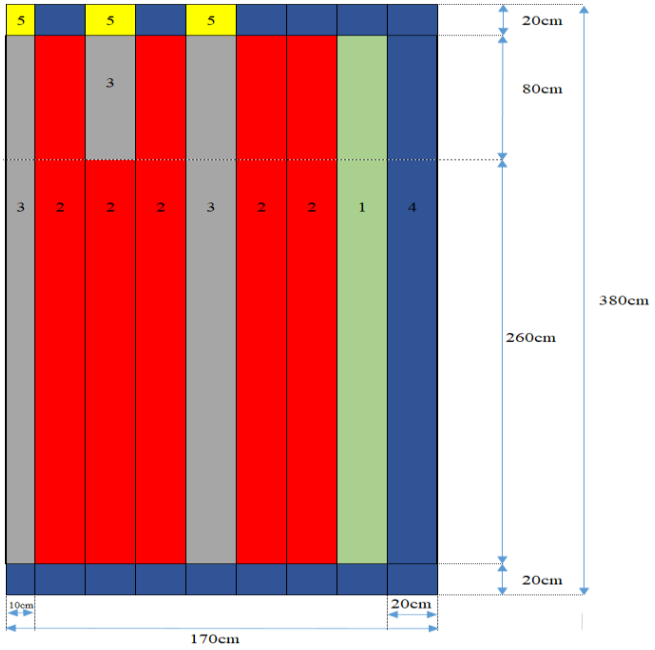

Fig. 2. Vertical cross-section IAEA 3-D benchmark [1].

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

To solve the two-group neutron diffusion equations (equation 1 and 2) with MATLAB PDE Toolbox, the parameters are redefined as per the PDE Toolbox Generic Systems equation notation (equation 3). For example, ∇c corresponds to ∇D, ∇u corresponds to $\nabla \Phi$, etc. as shown in Table II. Note that the MATLAB notation for diffusion coefficients, neutron flux and absorption coefficients are all in matrices form.

$$-\nabla D_1 \nabla \Phi_1 + (\Sigma_1^a + \Sigma_{1\to 2}^s)\Phi_1 = \frac{\nu\Sigma_1^f \Phi_1 + \nu\Sigma_2^f \Phi_2}{k} \quad (1)$$

$$-\nabla D_2 \nabla \Phi_2 + \Sigma_2^a \Phi_2 = \Sigma_{1\to 2}^s \Phi_1 \quad (2)$$

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

Table II. Correspondence between MATLAB PDE Toolbox and Neutron Diffusion Notation:

MATLAB PDE Toolbox	∇c	∇u	a	λ	D
2-Group Neutron Equation	∇D	$\nabla \Phi$	Σ_g^a	$1/k$	Σ_g^f

PDE Toolbox generally follows three major steps viz., the pre-processing, the solving, and the post-processing for solving the system of equations.

Pre-processing

To begin with, the geometry of the 1/8 IAEA 3-D core is modelled using SOLIDWORKS CAD software as shown in Fig. 3. The model is exported in the stereolithographic file format (.stl) and then imported in the MATLAB PDE Toolbox [5].

A program is developed to use the MATLAB PDE Toolbox for solving the two-group neutron diffusion equations. The program uses different algorithms for performing different unique task.

The first algorithm imports the model to MATLAB interface and provides all parameters, such as the number of PDEs, dimensions of each lattices, etc., required to solve the problem. Additionally, this algorithm (also referred to as script) interacts with some supplementary algorithms (called functions) to perform some designated tasks. One of the supplementary algorithms creates a table of coordinates of the lattices and with the coordinates, each material's data are specified to their corresponding lattice domain in the model as provided in Fig. 1 and Fig. 2.

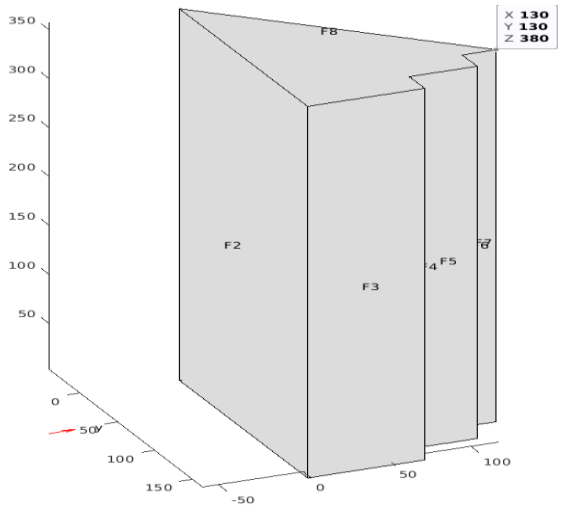

Fig. 3. 1/8 IAEA 3-D model (unit: cm).

In this model, there are two different boundary conditions: the vacuum and the reflective boundary conditions. The vacuum boundary condition is applied to the external surfaces only, which implies that there is no incoming neutron current on the external boundaries. Hence, the external surfaces represented in Fig. 3 with tags F3, F4, F5, F6, F7, F8 and F9 are set to the Neumann boundary condition. Their parameters are defined according to equation 6. Similar to the diffusion equations, the boundary condition notations in equation 8 are redefined in MATLAB according to Table III.

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

$$D\frac{\partial \phi}{\partial n} = 0 \quad (7)$$

$$c\nabla u + qu = g \quad (8)$$

Table III. Correspondence between PDE Toolbox Vacuum Boundary Condition and Neutron Diffusion Boundary Condition Notation.

PDE Toolbox	c	∇u	q	u	g
Neutron diffusion	D	$\frac{\partial \Phi}{\partial n}$	0.4692	Φ	0.0

For the inner surfaces, the reflective boundary condition is applied, which implies that the net neutron current is zero at the surfaces. The inner surfaces tagged as F1 and F2 are set to the Neumann boundary condition and their parameters defined according to equation 7 [6]. The PDE notation are redefined according to Table IV.

After the boundary condition, the model is discretized with a tetrahedral mesh with the maximum cell size of 5cm.

Table IV. Correspondence between PDE Toolbox Reflective Boundary Condition and Neutron Diffusion Boundary Condition Notation.

PDE Toolbox	c	∇u	q	U	G
Neutron diffusion	D	$\frac{\partial \Phi}{\partial n}$	0.0	Φ	0.0

Solving

The solving phase is commenced with a mesh cell of maximum size 8cm. The computation is also repeated for maximum size of 7cm and finally 5cm. The result appeared to converge at maximum cell size of 5cm. With a total mess cells of 3,017,700, it takes MATLAB a run time of 645 minutes for the computation to be fully carried out.

Post-Processing

In this phase, one of the developed supplementary algorithms in the program computes the normalized assembly power of the model. The algorithm uses Gauss Quadrature numerical integration method to compute the fission reaction rate in each of the square lattices. In the 2-D plane, each square lattice is divided into eight triangles. The fission rate across each triangle is interpolated by taking seven integration points with the 2-D Gauss quadrature method [7]. The normalized assembly power generated for the mid-plane (Z =190cm) is shown in Fig. 8.

Finally, nineteen different vertical planes were selected based on the 1-D Gauss Quadrature integration method, and their corresponding power distribution for each plane were interpolated.

RESULTS

PDE Toolbox successfully computed the k_{eff} and normalized assembly power of the IAEA 3-D PWR problem. The group-1 and group-2 2-D flux plots are shown in Fig. 4 and Fig. 5 while the 3-D plots are in Fig 6 and Fig. 7. The result obtained for mesh cell size of 5cm fairly related well with that from VENTURE code. The computed k_{eff} is 1.02916 which is about 13pcm compared to the VENTURE CODE result.

Furthermore, comparison of normalized assembly power calculated from MATLAB and VENTURE code is shown in Fig. 8. The maximum percentage error between the results is 3.6% and the estimated root mean square error (RMS) is about 0.0106. Fig. 9 graphically illustrates the bar plot comparing MATLAB and VENTURE results for each lattice.

It is worth noting that the VENTURE code result referred is the “extrapolated result” [1] and also that the reflector column (column 8) in Fig. 8 was intentionally omitted since the values are all zeros. Table V. provides details on the run time and processor used for running MATLAB program.

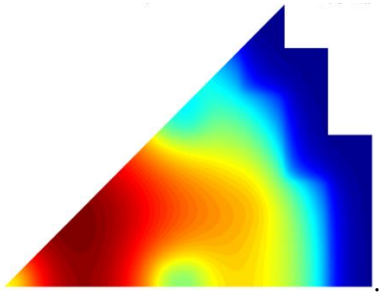

Fig.4 2-D view of group 1 flux at mid-plane (Z = 190cm)

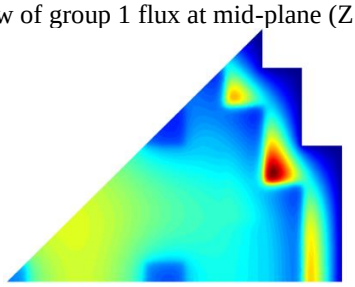

Fig.5 2-D view of group 2 flux at mid-plane (Z = 190cm)

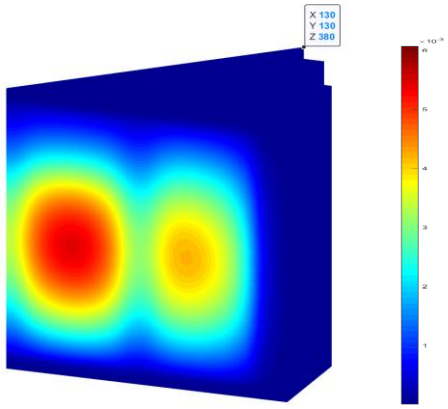

Fig. 6. Group 1 flux 3-D view.

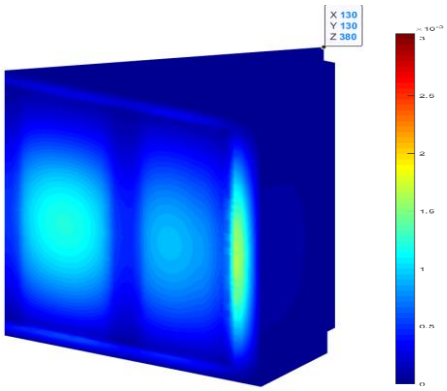

Fig. 7. Group 2 flux 3-D view.

Ref					0.5970		
Matlab					0.6032		
% error					1.0385		
k-eff:				0.4760	0.7000	0.6110	
Ref: 1.02903				0.4932	0.7040	0.6121	
Matlab: 1.02916				3.6134	0.5714	0.1800	
%Error: - 0.0130			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.8 Comparison of MATLAB assembly power with VENTURE [1, 6].

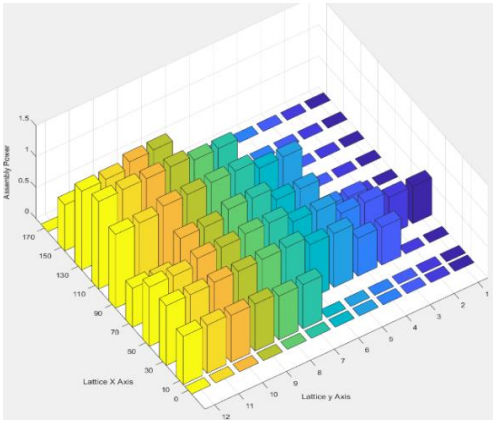

Fig.9 Plot comparing MATLAB and VENTURE for each lattice

Table V. Comparing Run Time and Processor Details used for VENTURE Code and for MATLAB PDE Toolbox

Description	VENTURE	MATLAB PDE
Mathematical Approach	Finite Difference Method	Finite Element Method
Mesh cells	102 x102 x19	10 x 301,770
Run Time(min)	360	650
Processor Model	IBM-360/91	Dual Intel E5-2699 v3 384GB RAM 2.30GHz
Date done	1975-76	2019 -2020

CONCLUSION

The k_{eff} and the normalized assembly power of the IAEA 3-D core were successfully estimated by MATLAB PDE Toolbox within a reasonable runtime and the results also match satisfactorily with the references. The results validate that MATLAB PDE Toolbox is accurate tool for modeling the PWR cores. MATLAB employs finite element methods for its modeling which is why it took longer than VENTURE Code that employs finite difference method. However, for complex geometry which may be difficult for the finite difference method to model, MATLAB PDE Tool will do a better job.

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
ν = average number of neutrons released per fission reaction
Σ_{fg} = fission reaction for group g
Φ_g = neutron flux for energy group g
λ = MATLAB notation for eigenvalue
c = MATLAB notation for diffusion
a = MATLAB notation for the coefficient of the flux
u = MATLAB notation for neutron flux
q = MATLAB boundary condition notation for coefficient of neutron flux
g = MATLAB boundary condition notation for a neutron current value

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. A. Soliman, "Validation of COMSOL Multiphysics for PWR Power Distribution via 3-D", COMSOL Conference in Rotterdam (2017).
4. H. Omara, "Validation of CITVAP Power Distribution via the IAEA 3-D PWR," Arab Journal of Nuclear Science and Applications (2015).
5. MathWorks, "Partial Differential Equation Toolbox User's Guide", p.2.41-100, MathWorks Inc. Apple Hill Drive Natick, MA (2019).
6. R. Murray, "An Introduction to the Concepts, Systems and Applications of Nuclear Processes, p.101-115", 8th Edition, Cambridge MA, (2020).
7. J. Reddy, "An Introduction to the Finite Element Method", p.591-618, 3rd Edition, McGraw-Hill Education (2006).