Development of MoCha-Foam: a new Method of Characteristics Neutron Transport Solver for OpenFOAM

Paul Cosgrove, Eugene Shwageraus

Department of Engineering, University of Cambridge, Trumpington St., Cambridge, UK, CB2 1PZ. pmc55@cam.ac.uk

Abstract – *MoCha-Foam is a new Method of Characteristics solver developed for the open-source multi-physics platform OpenFOAM. MoCha-Foam is capable of solving criticality problems with isotropic or anisotropic scattering on unstructured meshes of arbitrary heterogeneity within a rectangular region subject to reflective or vacuum boundaries. This paper includes a description of the code and successfully compares its performance against a number of benchmark problems of varying complexity.*

I. INTRODUCTION

The Method of Characteristics (MOC) is a technique used to solve partial differential equations which has long been applied to the numerical solution of the neutron transport equation [1]. Although initially unpopular due to limits on computational power, the MOC has since become a mainstay of lattice physics codes. Examples include CASMO [2], APOLLO [3], and DRAGON [4]. Among numerical transport methods, the MOC is particularly advantageous due to possessing a very general geometric capability, its ease of acceleration and parallelization, and being both simple in concept and in implementation. Advancing the MOC in a number of areas is one of the most prominent topics in transport methods – recent work includes attempted extensions to 3D and time-dependent problems [5][6].

OpenFOAM is an open-source C++ library. At inception it was primarily a platform for solving problems in computational fluid dynamics but has since evolved to handle other problems in continuum mechanics. More generally, OpenFOAM has proved a convenient platform on which to deal with the solution of partial differential equations due to its versatility and wide library of numerical methods [7].

Given that modern reactor physics problems often require a multi-physics approach, OpenFOAM's ability to straightforwardly tailor and couple multiple solvers and an open-source license have resulted in its growing popularity in nuclear applications. This has culminated in the formation of a Nuclear Special Interest Group (SIG) for collaboration on OpenFOAM. One notable showcase of OpenFOAM's multi-physics capabilities in this nuclear domain was the development of GeN-Foam, a coupled thermal-hydraulics, solid mechanics, and neutron diffusion solver for transient analysis of fast reactors [8]. Ultimately, this work on OpenFOAM is carried out with the aspiration of developing a new multi-physics platform capable of supplementing legacy codes with faster, parallel, and implicitly coupled tools in order to tackle modern challenges in reactor design.

The present work was begun recognizing the SIG's lack of an MOC solver; as MOC is a modern standard for deterministic transport methods, it was decided that this would prove a valuable contribution to expanding OpenFOAM as a nuclear platform with the potential to verify diffusion calculations and eventually couple with other solvers in the future. Hence, this work consists of the development and verification of an MOC solver on OpenFOAM, one which is capable of handling arbitrary heterogeneity on unstructured meshes within a rectangular boundary. Furthermore, this implementation has been extended to handle anisotropic scattering of arbitrary order. This paper briefly describes the implementation of MoCha-Foam and proceeds to demonstrate its capabilities.

II. DESCRIPTION OF THE ACTUAL WORK

Nomenclature

ψ = angular neutron flux
ϕ = scalar neutron flux
Q = angular neutron source
s = local characteristic length
δA= characteristic track spacing
Δ = change in angular neutron flux along characteristic
m = discrete direction index
k = characteristic track index
i = FSR index
A = FSR area
ω = quadrature weight for given direction

The Method of Characteristics

The basis of the MOC consists of transforming the global frame of reference of the multi-group neutron transport equation to one of neutrons streaming in discrete directions. The simplest implementations of MOC also assume that neutrons sources are discretized into a number of uniform regions and are isotropic. This reduces the neutron transport equation to:

$$\frac{d}{ds}\psi_{m,k,i}^{g}(s) + \Sigma_{Tr,i}^{g}\psi_{m,k,i}^{g}(s) = Q_i^g \tag{1}$$

Here the isotropic source is taken to be:

$$Q_i^g = \frac{1}{4\pi}\left(\sum_{g'} \Sigma_{s,i}^{g' \to g} \phi_i^{g'} + \frac{\chi^g}{k_{eff}} \sum_{g'} \nu\Sigma_{f,i}^{g'} \phi_i^{g'}\right) \tag{2}$$

Note, that the total cross-section normally present in the transport equation is replaced by a transport-corrected cross-section to account for the assumed isotropy. These assumptions allow for an analytical solution of the characteristic equation across a flat source region (FSR) along a characteristic line giving:

$$\psi_{m,k,i}^g(s) = \psi_{m,k,i}^g(0) e^{-\Sigma_{Tr,i}^g s} + \frac{Q_i^g}{\Sigma_{Tr,i}^g}\left(1 - e^{-\Sigma_{Tr,i}^g s}\right) \tag{3}$$

Thus the average angular flux streaming along a given ray traversing an FSR is given by:

$$\bar{\psi}_{m,k,i}^g = \frac{1}{s_{m,k,i}} \int_0^{s_{m,k,i}} \psi_{m,k,i}^g(s) ds = \frac{Q_i^g}{\Sigma_{Tr,i}^g} + \frac{\Delta_{m,k,i}^g}{\Sigma_{Tr,i}^g s_{m,k,i}} \tag{4}$$

The region average angular flux can then be computed by weighting each ray in a given direction across the FSR with respect to the area swept out by the ray using a user specified ray separation:

$$\bar{\psi}_{m,i}^g = \frac{\sum_k \bar{\psi}_{m,k,i}^g s_{m,k,i} \delta A_k}{\sum_k s_{m,k,i} \delta A_k} \tag{5}$$

Finally, the scalar flux may be calculated by weighting each average angular flux with respect to an angular quadrature dictated by the implementation:

$$\phi_i^g = \sum_m \bar{\psi}_{m,i}^g \, \omega_m \tag{6}$$

Hence the expression for the scalar flux in an FSR is:

$$\phi_i^g = \frac{4\pi Q_i^g}{\Sigma_{Tr,i}^g} + \frac{1}{\Sigma_{Tr,i}^g A_i} \sum_m \omega_m \delta A_m \sum_k \Delta_{m,k,i}^g \tag{7}$$

By specifying the angular flux at the intersection of a characteristic ray with the problem boundary, the above equations allow for a simple algorithm consisting of a series of nested for loops during which the change in angular flux across each FSR is computed and the scalar flux is incremented. Once converged, this flux is used to calculate the source terms and k_{eff}. The new source is then used to recalculate the scalar flux and the procedure repeats until sources and k_{eff} are converged.

Description of the implementation

The code developed traces rays across a geometry given a number of azimuthal angles and ray spacing by using cyclic ray tracing which ensures that each ray – after being reflected from the problem boundaries – ultimately returns to the point from which it originated. This guarantees that reflective and periodic boundary conditions can be straightforwardly implemented at the cost of restricting the generality of the boundary to be rectangular. However, within the boundaries, the problem may have arbitrary heterogeneity due to OpenFOAM's unstructured mesh capabilities.

The azimuthal and polar quadrature set have been decoupled in the implementation: the azimuthal set is a standard for cyclic ray tracing, modified from the equal weight azimuthal quadrature set to account for the adjustment to azimuthal angles and ray spacing required by the ray tracing routine. The polar quadrature uses either the Tabuchi-Yamamoto quadrature set [9] or Gauss-Legendre set [10] with the former option as the default.

As the code does not generate its own cross-section data, this must be input separately in a ‘nuclearData’ file.

The code is much simplified by OpenFOAM’s ‘field’ class which allows algebra to be straightforwardly carried out among the different field objects while simplifying bookkeeping by automatically associating each value of a field to an element in the mesh. These fields have been used for each scalar flux energy group, each fission/scattering/total source energy group, and likewise for the relevant cross-sections. Pre-existing features of OpenFOAM such as this have proved very advantageous in the quick development of MoCha-Foam. Similarly, significant time was saved due to OpenFOAM’s inclusion of a versatile post-processing package, ParaView [11].

Efforts have been made to relax the assumptions of isotropic scattering in this code. This was done following the flat-source version of the anisotropic scattering implementation given by Ferrer & Rhodes for CASMO-5 [12]. Here the transport sweep is used to construct not only the scalar flux but also its angular moments using the real spherical harmonics. This implementation was chosen with a view towards incorporating linear source capabilities in the code as well in the near future.

The transport sweep employed in the code will assume either isotropic or anisotropic sources, depending on the information provided in nuclearData. The only notable difference from simple transport sweep implementations is the recalculation of the scattering source at the end of each iteration. This increases the runtime of each transport sweep but proves very advantageous in speeding up spectral convergence and hence significantly decreasing the number of outer iterations required.

A notable omission from the code at present is the lack of spatial acceleration methods – the future addition of a Coarse Mesh Rebalance or other alternative will prove crucial if the solver is to succeed in tackling large problems with high dominance ratios. As a consequence, this imposes a practical limit on the size of the problems which can be tackled in the following analysis to the scale of a super-cell or small assembly, rather than multi-assembly or full core problems which are now standard benchmarks for deterministic codes. The speed of the code is also hampered by the particularly slow ray-tracing method currently used in the implementation – for larger problems with a relatively fine mesh, this tends to demand significantly more time than the total duration of the transport sweep calculation.

III. RESULTS

1. One-dimensional benchmarks

The first problems tackled by MoCha-Foam are infinite homogeneous benchmarks with descriptions taken from Sood et. al. [13]. Due to the extreme simplicity of these problems, no attempt is made to examine sensitivity to changes in ray spacing etc. The benchmarks and their analytical k_{inf} values are given in Table 1. Each problem is described by a code of the type M-N-S-G where M gives the material (e.g., PUa for a set of plutonium cross-sections), N is the number of energy groups, S is the order of scattering anisotropy (e,g, 0 for isotropic) and G is the problem geometry – here either IN for infinite or SL for slab.

Table 1. Infinite, homogeneous benchmark problems and their solution using MoCha-Foam

Benchmark	Reference k_{inf}	Calculated k_{inf}	Error (%)
PUa-1-0-IN	2.612903	2.612902	3.8e-5
PUb-1-0-IN	2.290323	2.290321	8.7e-5
Ua-1-0-IN	2.250000	2.249998	8.9e-5
Ub-1-0-IN	2.330917	2.330916	4.3e-5
Ue-1-0-IN	2.1806667	2.180665	7.8e-5
UD_2O-1-0-IN	1.133333	1.133332	8.8e-5
UAl-2-0-IN	2.661745	2.661737	3.0e-4
URR-3-0-IN	1.600000	1.599999	6.3e-5
URR-6-0-IN	1.600000	1.599999	6.3e-5

To demonstrate the anisotropic scattering capability of the code, 1D critical slab problems with vacuum boundaries were attempted for problems from Sood et. al. containing anisotropic scattering with critical thickness specified for each. Although k_{inf} remains the same for problems with or without anisotropic scattering, k_{eff} for a slab will differ. For the two plutonium benchmarks, this anisotropy is quadratic while the remainder of benchmarks have linear anisotropy. The results from these experiments are given in Table 2.

Table 2. Homogeneous benchmark problems in critical slab geometry featuring anisotropic scattering

Benchmark	Reference k_{eff}	Calculated k_{eff}	Error (%)
PUa-1-2-SL	1.000000	0.999658	0.034
PUb-1-2-SL	1.000000	0.999727	0.027
URR-2-1-SL	1.000000	0.999492	0.051
UD_2O-2-1-SL	1.000000	0.999974	0.003

2. Two-dimensional isotropic benchmarks

Following this, the solver was used to tackle a series of 2D problems as presented by [14], [15], [16] and [17]. For each of these problems the results from the MOC solver will be compared against results found in the references in terms of k_{eff} and – where given – flux distribution. Visualizations of neutron and fission distributions and profiles can also be easily obtained using ParaView, OpenFOAM’s default visualization tool [11]. Aside from the first of these problems, all meshes used were unstructured meshes produced using Gmsh which is a free-to-use (not open-source) mesh generator with which OpenFOAM can interface conveniently [18].

The first problem is a 2-group, 2-region BWR cell in which the fuel pins are homogenized into a fuel material and surrounded by moderator. The geometry is show in Figure 1 and has a width and length of 8.9cm. This problem is used to examine the sensitivity of the solver to variations in the number of azimuthal angles, average ray spacing, and mesh size. Due to the use of the Tabuchi-Yamamoto polar quadrature generally requiring only 3 polar angles for acceptable accuracy, the effect of varying polar angles was not examined. The mesh used for angular and ray spacing investigations was created using OpenFOAM’s blockMesh utility and is composed of rectangular elements of size ~0.25cm x 0.25cm. When not being varied, 64 azimuthal angles were used with a ray density of approximately 5 rays intersecting each mesh element for each angle. The results of the sensitivity experiments are show in Tables 3, 4 and 5.

Table 3. Sensitivity of k_{inf} of the homogeneous BWR assembly to number of azimuthal angles

Azimuthal angles	Calculated k	Ray Tracing Time (s)	Calculation Time (s)
24	1.212674	8	6
36	1.212615	12	10
48	1.212600	16	14
64	1.212590	21	18
72	1.212588	25	19
128	1.212581	43	35

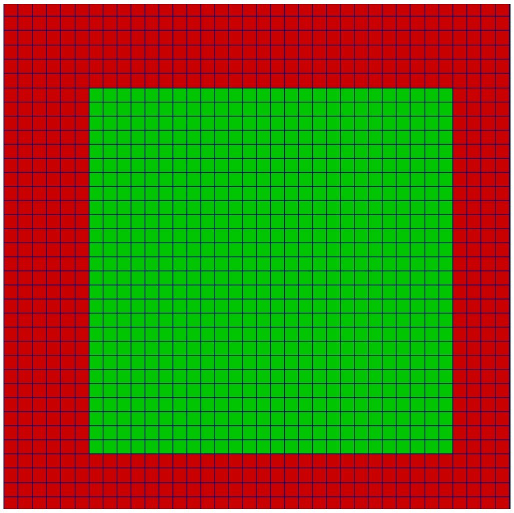

Figure 1: Mesh used for the homogeneous BWR assembly benchmark

Table 4. Sensitivity of k_{inf} of the homogeneous BWR assembly to space between parallel rays

Ray spacing (cm)	Calculated k	Ray Tracing Time (s)	Calculation Time (s)
0.5	1.206497	2	2
0.1	1.212603	10	9
0.05	1.212590	22	17
0.01	1.212588	100	91

Table 5. Sensitivity of k_{inf} of the homogeneous BWR assembly to mesh element size

Mesh edge length (cm)	Calculated k	Ray Tracing Time (s)	Calculation Time (s)
0.5	1.213528	2	5
0.25	1.212590	22	17
0.125	1.212394	292	65
0.0625	1.212339	5543	286

A comparison of the relative integral flux distribution found by the code in each group and material region can also be made with that found by other codes and is shown in Table 6. There is good agreement between MoCha-Foam and the transmission probability code TPTRI but differences are present in the moderator region when compared against the spherical harmonics code TEPFEM and integral transport code SURCU. It is suspected that this is due to both TPTRI and MoCha-Foam assuming that scattering in the moderator is isotropic (due to lack of higher order scattering cross-sections in the data provided in [15]) whereas SURCU and TEPFEM account for the strong anisotropy of scattering on hydrogen. Note that all fluxes have been normalized to the fast flux in the fuel.

Table 6. Flux distribution in the homogeneous BWR problem according to different codes

Code	Group 1		Group 2		k eigenvalue
	Fuel	Mod.	Fuel	Mod.	
SURCU	1.0	0.9269	0.3527	0.4514	1.2127
TEPFEM	1.0	0.9207	0.3563	0.4536	1.2136
TPTRI	1.0	0.8643	0.3525	0.4203	1.2128
MoCha-Foam	1.0	0.8638	0.3522	0.4216	1.212393

The second test is a one-group problem featuring an array of six rectangles of fuel, each surrounded by moderator used to test the code’s vacuum boundary capability. Each fuel element is 18cm × 1cm while the entire geometry is 20cm × 20cm. The geometry is show in Figure 2.

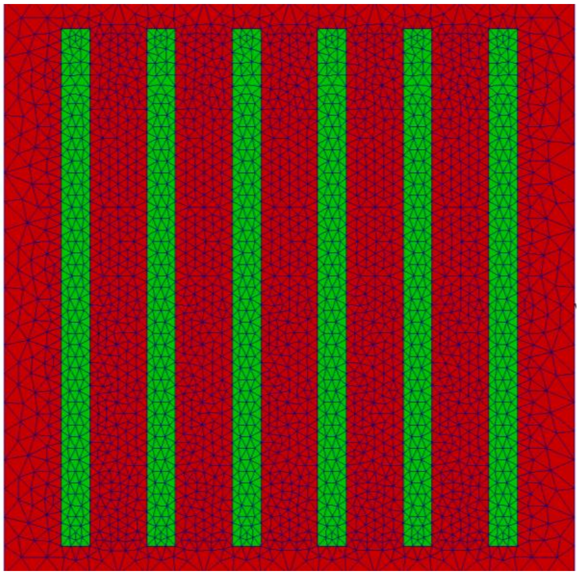

Figure 2. Mesh used for the mono-group benchmark

The flux distribution obtained is compared against other solvers in Table 7. As in Table 6, there is better agreement shown in relative flux distribution between MoCha-Foam and the isotropic code TPTRI than with TEPFEM.

Table 7. Relative flux distribution for the one-group benchmark as calculated by different solvers

Code	Φ_F / Φ_M
TPTRI	0.5831
TEPFEM	0.5761
MoCha-Foam	0.5841

The third problem, referred to as an LWR with burnable poison, is a variant of the first with one homogenized fuel pin replaced by a pin containing gadolinium. The geometry of this problem is depicted in Figure 3.

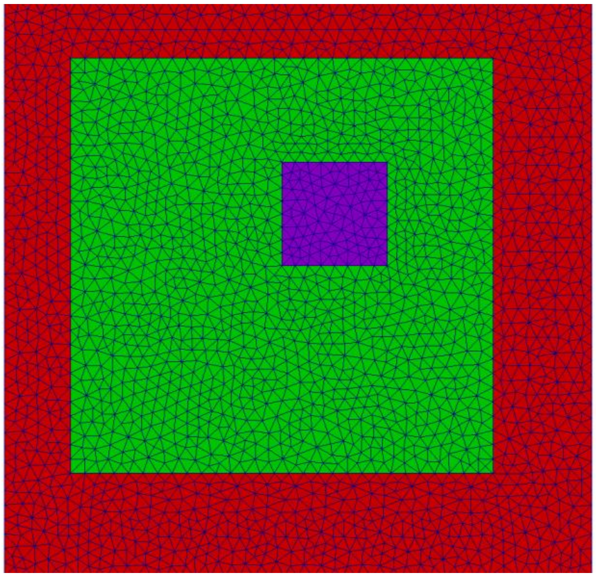

Figure 3. Mesh used for the homogeneous LWR with burnable poison benchmark

Finally, the last problem is a heterogeneous 4×4 BWR supercell of cladded fuel pins with two containing Gadolinium and is used to assess MoCha-Foam's heterogeneous geometry capabilities. The geometry is shown in Figure 4. Colours in the geometry correspond to neutronically identical pins which are identified to compare the power generated by each in Table 8. Pin 6 contains gadolinium.

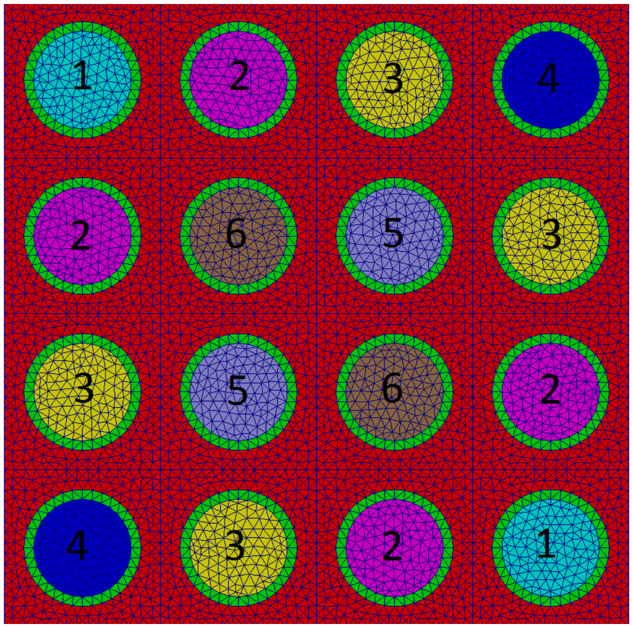

Figure 4. Mesh used for the heterogeneous BWR benchmark with numbered pin cells

In terms of the fission rate distributions MoCha-Foam shows excellent agreement with DRAGON and MOCUM.

Table 8. Normalized pin power distributions in the heterogeneous BWR benchmark

Pin	DRAGON	MOCUM	MoCha-Foam	% Error vs. DRAGON
1	6.9391e-2	6.9476e-2	6.9441e-2	0.07
2	6.6246e-2	6.6268e-2	6.6196e-2	0.08
3	6.9398e-2	6.9441e-2	6.9409e-2	0.02
4	7.2549e-2	7.2626e-2	7.2584e-2	0.05
5	6.2429e-2	6.2322e-2	6.2321e-2	0.17
6	2.4344e-2	2.4265e-2	2.4443e-2	0.41

For each of these benchmark problems, reference k eigenvalues as produced by other codes are shown in Table 9. There is no firm agreement between previous codes on the reference values for each of these benchmarks but MoCha-Foam falls either within the range of values provided or reasonably close outside. Hence, MoCha-Foam is successful at solving isotropic 2D neutron transport problems.

Table 9. Comparison of k eigenvalues produced by MoCha-Foam against previous benchmark solutions

Benchmark	Code	Reference k	MoCha-Foam k	Max % Error
Homogeneous BWR assembly	SURCU	1.2127	1.21234	0.10
	TEPFEM	1.2136		
	TPTRI	1.2128		
	BOXER3	1.2127		
One-group eigenvalue problem	SP3	0.798617	0.802211	0.49
	TEPFEM	0.803068		
	TPTRI	0.806123		
	BOXER3	0.80147		
LWR assembly with poison	SURCU	0.8805	0.886423	0.67
	TPTRI	0.8828		
	BOXER3	0.884846		
Heterogeneous BWR	DRAGON	0.986561	0.989983	0.35
	MOCUM	0.987785		
	BOXER3	0.9876		
	Unnamed[17]	0.989683		

3. Two-dimensional anisotropic benchmarks

Having succeeded in demonstrating the solver's capabilities for isotropic problems in 2D, the solver's ability to handle problems with anisotropy is examined. The first of these problems is taken from Postma & Vujic [19] – a 2-group pin cell problem featuring linear anisotropy. The problem in question is a circular fuel pellet of radius 0.41cm, surrounded by cladding with thickness 0.06cm in a light water unit cell with pitch 1.2657cm. The cross-sections for this problem are provided in the reference and the k_{inf} values obtained by both Postma & Vujic's CHAR-A as well as TIBERE-2 are given. Each of these k_{inf} values and that obtained by MoCha-Foam are given in Table 10.

Table 10. Comparison of k value produced by MoCha-Foam on the linearly anisotropic benchmark against other anisotropic solvers

Code	TIBERE-2	CHAR-A	MoCha-Foam
k_{inf}	1.06496	1.06403	1.064451
% Error	0.048%	-0.040%	-

In order to examine the code's ability to handle anisotropic scattering of arbitrary order, a relatively anisotropic benchmark problem has been devised. This problem consists of a quarter of a hypothetical reactor assembly containing central UO_2 pins (5% enrichment), guide tubes, peripheral MOX pins and a water gap of half-thickness 1.26cm. Each pin and guide tube has inner radius 4.095mm, a clad thickness of 0.655mm, and a pitch of 1.26cm. All boundaries were set as reflective. The geometry is illustrated in Figure 5.

To obtain cross-section information, the problem description was fed into SERPENT [20] and the relevant cross-sections extracted using a MATLAB script. SERPENT was able to provide both transport-corrected and standard cross-sections with information for up to P_7 scattering moment. However, to prevent the need for excessive statistics, only scattering up to P_4 is considered in this paper.

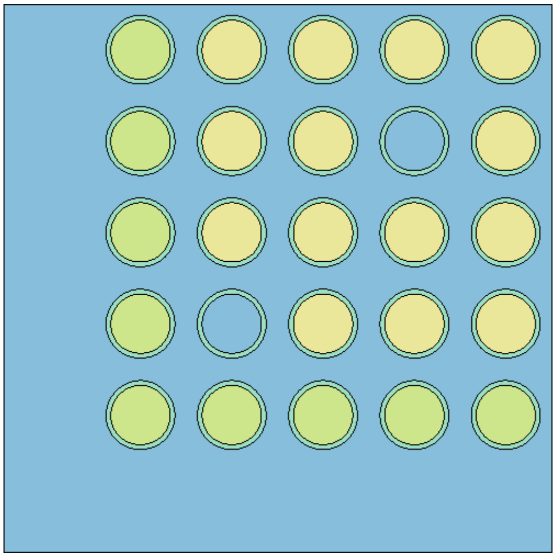

Figure 5. Geometry of benchmark assembly. Green pins contain MOX, yellow contain UO_2, and blue are guide tubes containing moderator

As the Monte Carlo solution to the problem may differ from the deterministic solution due to differences in the solution method, handling of neutron energies, and geometric fidelity, it was desired to also compare MoCha-Foam against another deterministic solver. Hence, the transport-corrected cross-sections produced by SERPENT were fed into the isotropic scattering MOC code OpenMOC [21]. The finely divided flat source regions used in the OpenMOC solution are shown in Figure 6 along with the mesh used by MoCha-Foam. Both OpenMOC and MoCha-Foam used 3-point Tabuchi-Yamamoto polar quadrature, 72 azimuthal angles and an average ray spacing of 0.01cm in all solutions.

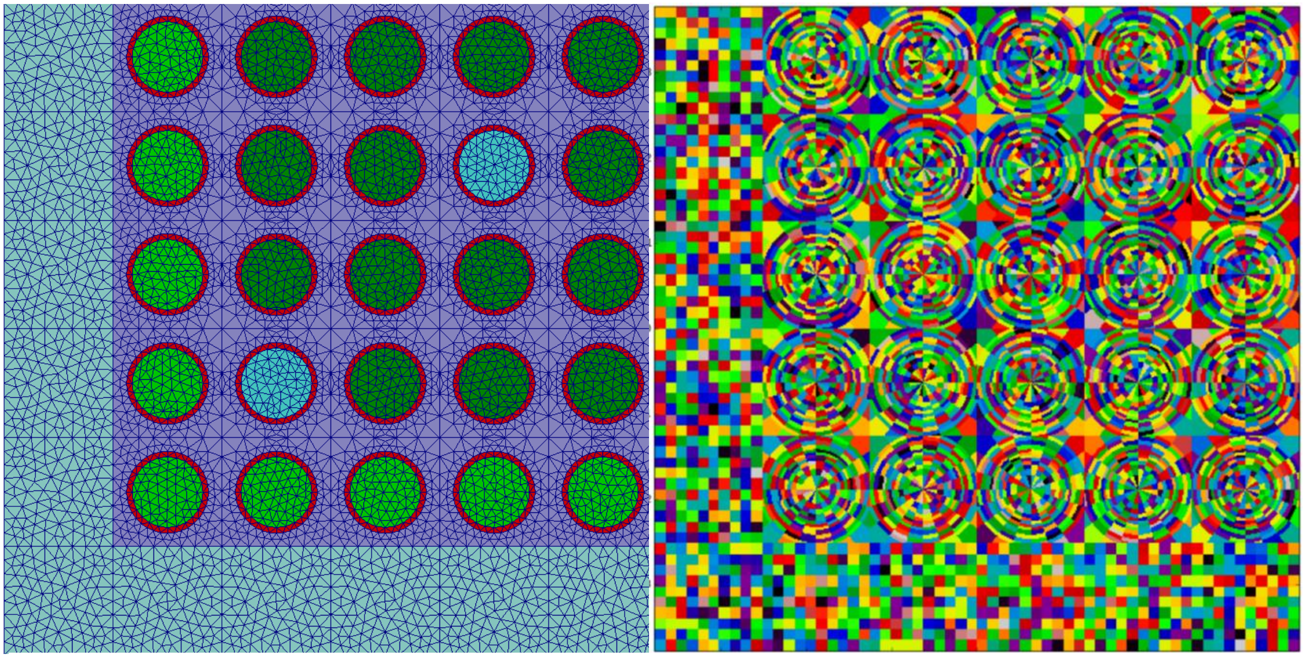

Figure 6. Flat Source Regions/Mesh cells used by MoCha-Foam (left) and OpenMOC (right) for the quarter assembly benchmark

The P_0 solution provided uses transport-corrected cross-sections while higher order solutions use the cross-sections without any correction. The calculated k values and flux distributions in each fuel pin for each scattering order are compared against both OpenMOC and Serpent. The k comparison is shown in Table 11 while the maximum and root mean square errors in fission source distribution are shown in Table 12. Note here that the run times from MoCha-Foam do not include ray tracing which takes significantly longer than the transport sweep typically. The run time for OpenMOC does include ray tracing, with the calculation performed using a single thread and no acceleration. The run time for SERPENT was on the order of 9 hours using 5 threads in order to obtain high accuracy in the scattering moment cross-sections. For practical applications, the ray tracing method used by MoCha-Foam must be optimized. Furthermore, attention must be paid to the anisotropic transport sweep algorithm which takes significantly longer than its isotropic counterpart – this section of the code is inefficient in both run time and memory usage and can likely be much improved with further efforts. As a comparison, see the relative run times and memory requirements reported by [12] for which the anisotropic solver does not take significantly longer than the isotropic in the flat source case.

Table 11. Comparison of k eigenvalue reported by SERPENT, OpenMOC, & MoCha-Foam using different scattering orders

Code	k_{inf}	Error vs. SERPENT (δpcm)	Run time (s)
SERPENT	1.41009±1	-	-
OpenMOC	1.410684	59	148
P_0	1.411936	185	411
P_1	1.410677	59	2320
P_2	1.410247	16	2698
P_3	1.410192	10	3137
P_4	1.409803	-29	3719

Regarding the difference in the k eigenvalue between OpenMOC and the P_0 solution, at present one speculates that the problem is due to the relative mesh fidelities: OpenMOC flat source regions possess higher order edges whereas OpenFOAM is restricted to using only elements with straight edges – this results in a loss of accuracy through possible loss of fissile volume and perturbing the fuel to moderator ratio of the problem. Further experiments have shown both k values to be fairly insensitive to changes in mesh size although these investigations were limited for OpenFOAM due to the large scaling of ray tracing time with decreases in mesh element size. A similar overestimation of the k eigenvalue is seen in the heterogeneous BWR benchmark above and was reported by another unstructured mesh MOC code [17].

Table 12. Relative Fission Source Errors relative to SERPENT reported by OpenMOC & MoCha-Foam using different scattering order

Code	Max Error %	RMS Error %
OpenMOC	5.13	1.63
P_0	4.28	1.52
P_1	2.36	1.08
P_2	2.56	1.15
P_3	2.54	1.14
P_4	2.54	1.14

Regarding fissions source distribution, both OpenMOC and the P_0 calculation agree in their disagreement – compared against each other, each pin agrees within 1% and most within 0.1%. However, both differ noticeably from SERPENT as can be seen in Figure 7. This appears in large part to be due to the presence of the water gap and resulting anisotropy to which the corner MOX pin is subjected. The inclusion of higher-order scattering terms corrects this error somewhat, also shown in Figure 7. On investigation, it was found that using SERPENT to generate cross-sections for the corner MOX pin and the other MOX pins separately improved this agreement by a 1-2%. It is suspected that generating unique cross-sections for each cell of the problem would improve the agreement further.

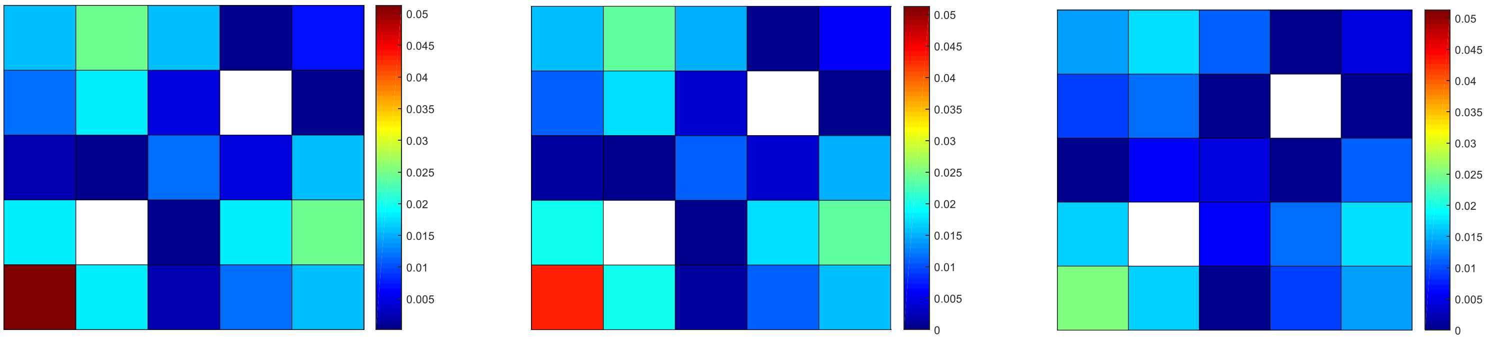

Figure 7. Fission Source Errors of OpenMOC (left), MoCha-Foam with P_0 scattering (middle), and MoCha-Foam with P_4 scattering (right) relative to SERPENT

IV. CONCLUSION

MoCha-Foam has demonstrated capability in 1D and 2D, isotropic and anisotropic problems through comparison against a number of benchmarks and other codes. This provides a valuable addition to OpenFOAM's repertoire in the nuclear domain. However, MoCha-Foam is still most certainly in the relatively early stages of its development and its capabilities cannot yet compare to those of other modern MOC codes, if only in terms of practical run times. Thus, further work on the project should continue with a focus on improving the speed of ray tracing and accelerating the flux solution. Additional attention must also be paid to optimizing the anisotropic transport sweep. Future code developments may also include the use of linear sources, parallelization of the transport sweep and coupling with thermal-hydraulics solvers.

REFERENCES

1. MJ. HALSALL, "CACTUS, a characteristics solution to the neutron transport equations in complicated geometries," Atomic Energy Establishment, Winfrith, United Kingdom (1980).
2. K. SMITH & J.D. RHODES, "CASMO-4 Characteristics Methods for Two-Dimensional PWR and BWR Core Calculations," *Transactions of the American Nuclear Society*, **83**, 294 (2000).
3. R. SANCHEZ, I. ZMIJAREVIC, M. COSTE-DELCLAUX, E. MASIELLO, S. SANTANDREA, E. MARTINOLLI, L. VILLATE, N. SCHWARTZ, & N. GULER, "APOLLO2 Year 2010" *Nuclear Engineering and Technology*, **42**(5), 474 (2010).
4. G. MARLEAU, A. HEBERT, & R. ROY, "A User Guide for DRAGON," Report IGE-174, Institut de genie nucléaire, École Polytechnique de Montréal, Montréal, Québec (2000).
5. G. GUNOW, S. SHANER, B. FORGET & K. SMITH, (2016). "Reducing 3D MOC Storage Requirements with Axial On-the-fly Ray Tracing," Proc. PHYSOR 2016, Sun Valley (2016).
6. A. TALAMO, "Numerical solution of the time dependent neutron transport equation by the method of the characteristics," *Journal of Computational Physics*, **240**, 248 (2013).
7. H. WELLER, G. TABOR, H. JASAK, & C. FUREBY, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in Physics*, **12**(6), 620 (1998).
8. C. FIORINA, I. CLIFFORD, M. AUFIERO, & K. MIKITYUK, "GeN-Foam: a novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors," *Nuclear Engineering and Design*, **294**, 24 (2015).
9. A. YAMAMOTO, M. TABUCHI, N. SUGIMURA, T. USHIO, & M. MORI, "Derivation of Optimum Polar Angle Quadrature Set for the Method of Characteristics Based on Approximation Error for the Bickley Function," *Journal of Nuclear Science and Technology*, **44**(2), 129 (2007).
10. E.E. LEWIS & W.F. MILLER, "Computational methods of neutron transport," Wiley, New York, ISBN 0-471-09245-2 (1984).
11. J. AHRENS, B. GEVECI, & C. LAW, *Visualisation Handbook, ParaView: An End-User Tool for Large Data Visualisation*, Elsevier, Burlington, MA (2005)
12. R. FERRER & J. RHODES III, "A Linear Source Approximation Scheme for the Method of Characteristics," *Nuclear Science and Engineering*, **182**, 151 (2016).
13. A. SOOD, R.A. FORSTER, & D.K. PARSONS, "Analytical Benchmark Test Set for Criticality Code Verification," *Progress in Nuclear Energy*, **42**(1), 55 (2003).
14. S. RAY, S.B. DEGWEKER, R. RAI, & K.P. SINGH, "A Collision Probability and MOC-Based Lattice and Burnup Code for Analysis of LWR Fuel Assemblies," *Nuclear Science and Engineering*, **184** (2016).
15.W. HONGCHUN, L. PINGPING, Z. YONGQIANG, & C. LIANGZHI, "Transmission probability method based on triangle meshes for solving unstructured geometry neutron transport problem," *Nuclear Engineering and Design*, **237**, 28 (2007).
16. X. YANG & N. SATVAT, "MOCUM: A two-dimensional method of characteristics code based on unstructured meshing for general geometries," *Annals of Nuclear Energy*, **46**, 20 (2012).
17. T. MAZUMDAR, "Solution of neutron transport equation by Method of Characteristics," *Annals of Nuclear Energy*, **77**, 522 (2015).
18. C. GEUZAINE & J. REMACLE, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, **79**(11), 1309 (2009).
19 T. POSTMA & J. VUJIC, "The Method of Characteristics in General Geometry with Anisotropic Scattering," Proc. M&C 1999, Madrid (1999).
20. J. LEPPANEN, "Development of a New Monte Carlo reactor physics code," PhD Thesis. Helsinki University of Technology (2007).
21. W. BOYD, S. SHANER, L. LI, B. FORGET, & K. SMITH, "The OpenMOC method of characteristics neutral particle transport code," *Annals of Nuclear Energy*, **68**, 43 (2014).