22.212 MOC Homework

Isaac Meyer
Professor Forget

October 19, 2018

Overview

A 2D pseudo-random ray method of characteristics code ("MIMOSA") utilizing constructive solid geometry is developed in python and some performance metrics are analyzed. The code for this project is available at https://github.com/icmeyer/mimosa. The guideline for the physics protion of the code is mostly guided by John Tramm's work on true random ray method of characteristics [1] and OpenMOC [2].

Method of Characteristics

The method of characteristics relies on tracking the angular flux of a problem and making a contribution to flat source regions (FSRs). We can calculate this change in ψ with an entry point s and exit s' by

$$\Psi_{kg}\left(s'\right) = \Psi_{k,g}\left(s\right) e^{-\tau_{kig}} + \frac{Q_{ig}}{\Sigma_{ig}^{T}}\left(1 - e^{-\tau_{kig}}\right) \text{ where } \tau_{kig} = \Sigma_{ig}^{T}(s' - s)$$

1 Code Description

Traditional method of characteristics codes solve the eigenvalue problem by using a predetermined set of tracks, and performing power iterations with updates to the source by calculating the contributions of each track. This requires a careful placing of the rays and keeping track of a quadrature for adding track contributions in each FSR. True random ray never uses the same ray twice and instead produces new initial points and angles for each step of the power iteration. In this method of "pseudo random ray", an initial set of rays is created using the random method but reused for each iteration. The basic algorithms for the solution are laid out below. In the code itself the algorithms are generally contained as such:

Table 1: Code Structure

Algorithm	Corresponding File
1	main.py
2	ray.py
3	physics.py
4	physics.py

Algorithm 1 Power Iteration Driver

Lay tracks using Algorithm 2
$k \leftarrow$ guess
$\phi \leftarrow$ guess
while not converged **do**
    $q \leftarrow$ Algorithm 3
    update k
    $\phi \leftarrow 0$
    Transport Sweep using Algorithm 4
    for region i **do**
        for group g **do**
            $\phi_{i,g} \leftarrow \frac{\phi_{i,g}^{\text{transport}}}{\Sigma_{t,i,g} V_i D_{i,\text{tracks}}} + 4\pi Q_{r,e}$ ▷ Where $D_{i,\text{tracks}}$ is the total active track length over the whole problem

Algorithm 2 Track Laying

for ray **do**
    $r \leftarrow (\xi_1, \xi_2)$
    $\theta \leftarrow (1 - 2\xi_3)\frac{\pi}{2}$
    $\varphi \leftarrow \xi_4 2\pi$
    while $d_{tot} <$ cutoff **do** ▷ We have the answer if r is 0
        find nearest surface
        calculate distance d
        d_{tot} += d
        if $d_{tot} > d_{active}$ **then**
            store segment with d and region information

Algorithm 3 Q Calculation

for region i **do**
    for group g **do**
        $q_i \leftarrow \frac{1}{4\pi\Sigma_{t,i,g}} \left[S_{i,g} + \frac{1}{k} F_{i,g}\right]$ ▷ Where S and F represent the scatter and fission sources respectively

Algorithm 4 Transport Sweep

Calculate initial source q
for ray r **do**
    $\psi_{init} \leftarrow q_i$ ▷ i denotes the region
    for segment s **do**
        for group g **do**
            $\tau \leftarrow \Sigma_{tig}, d_s$
            $\Delta\psi \leftarrow (\psi_g - q_{ig}) \cdot (1 - e^{-\tau})$
            if $d_{tot} > d_{active}$ **then**
                ϕ_{ig} += $4\pi\Delta\psi$
            ψ −= $\Delta\psi$

Application

Cross sections for a 3.2% enriched uranium pincell with $R = 0.39218$ cm and pitch= $1.26cm$ were generated using OpenMC with reflective boundary conditions. Both 2- and 10-group cross sections were generated with the following group structure.

Table 2: Energy Groups

Groups	Bin Boundaries (eV)
2	$0.0, 1000, 20e6$
10	$0.0, 0.058, 0.14, 0.28, 0.625, 4.0, 10.0, 40.0, 5530.0, 821e3, 20e6$

Two problems were run: a pincell with two regions and a 3-by-3 lattice of equivalent enrichment pincells with 4 spatial fuel regions. Visualizations of the MOC geometry are generated by plotting individual segments with the color of the region in which they are contained. Figure 1 shows a good examples of the randomly laid tracks. Plots more representative of the full track lay down are also shown.

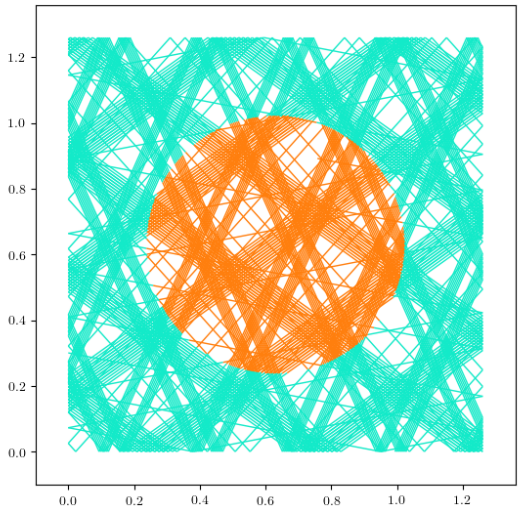

Figure 1: Pincell, 4 Rays at 300 cm

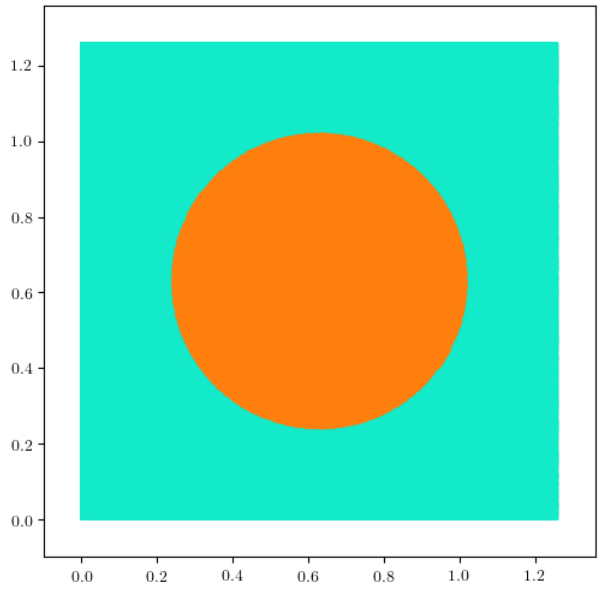

Figure 2: Pincell, 1000 Rays at 300 cm

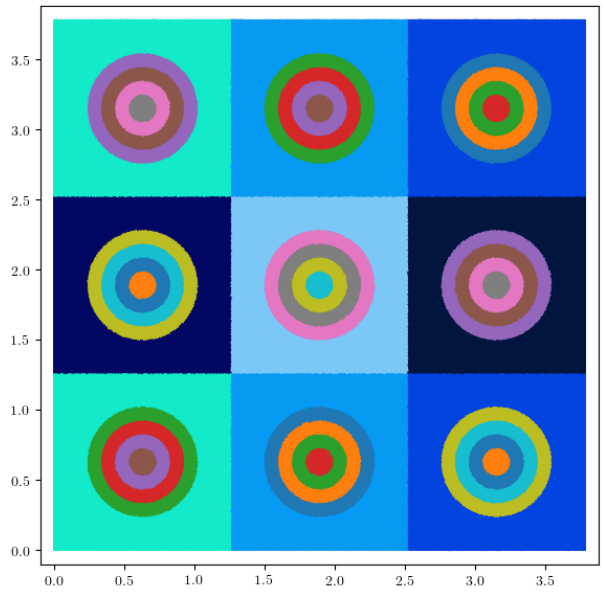

Figure 3: 3-by-3 with refinement, 1000 Rays at 300 cm

1.1 Numerical Results

The simulation convergence criterion was a change in the absolute value of less than $1e-5$. Simulations were run with a total of 1000 rays, ray length of 300 cm and a deadzone of 50 cm. Performance was basedon the wall-clock time of the entire simulation. Results are shown in Table 3.

Table 3: Numerical Results

	Groups	k_{eff}	Performance (Seconds/Segment/Group)
OpenMC: pincell	n/a	1.28941 $\pm$ 0.00374	n/a
MIMOSA: pincell	2	1.28349	0.000868
MIMOSA: pincell	10	1.26921	0.000144
MIMOSA: 3-by-3	2	1.29553	0.00189
MIMOSA: 3-by-3	10	1.27125	0.000317

1.2 Behavior of convergence

These plots show the behavior of the eigenvalue over the power iterations.

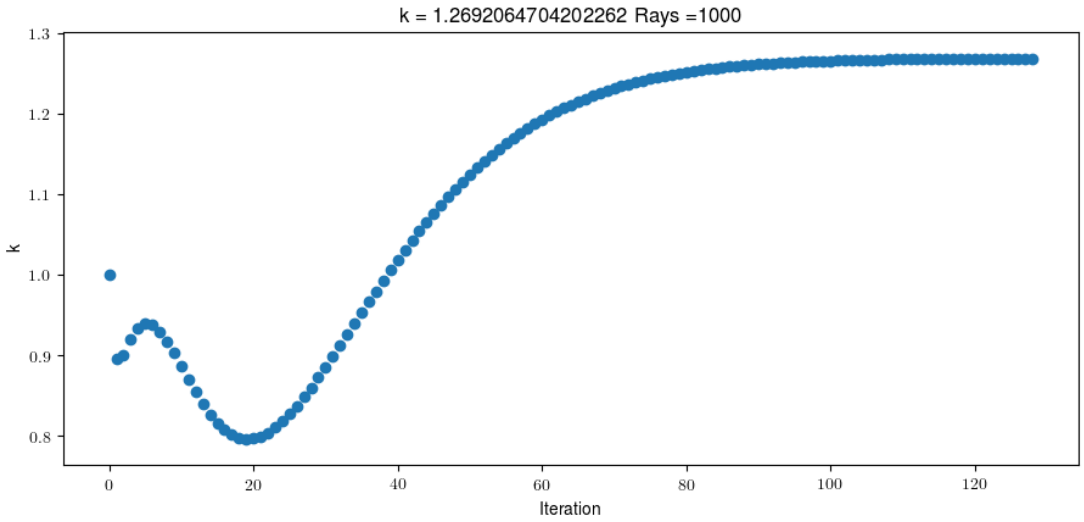

Figure 4: Pincell

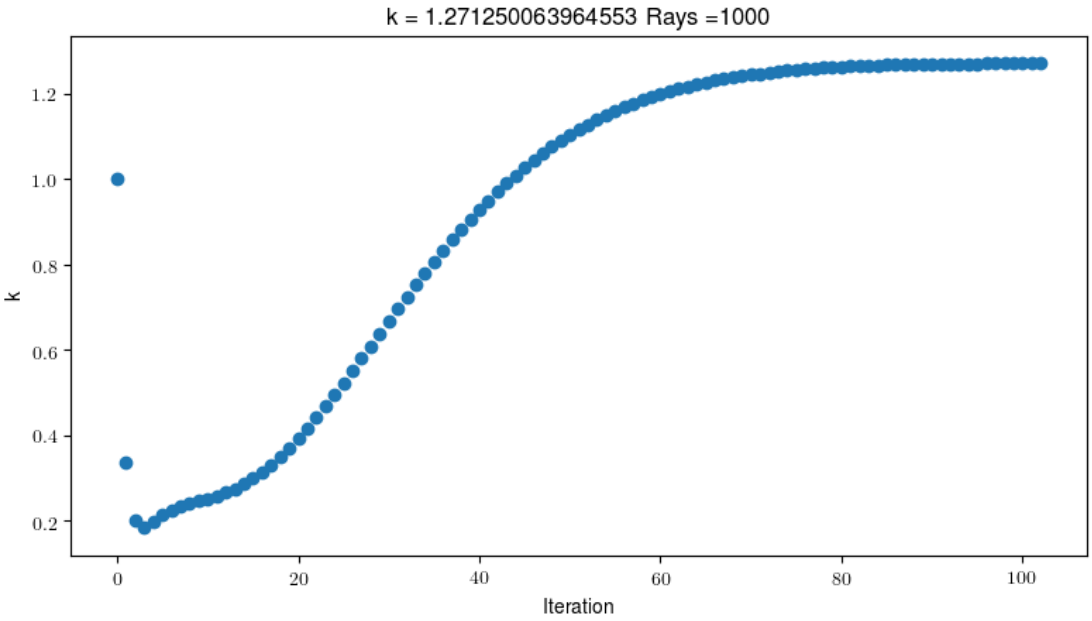

Figure 5: Pincell

1.3 Flux

The flux results follow what one might expect. The dips at the resonances are clear and while we do not expect a huge difference between moderator and fuel flux, we can see in Figure 6 that there is a slightly harder spectrum in the fuel as well as larger dips at the resonances.

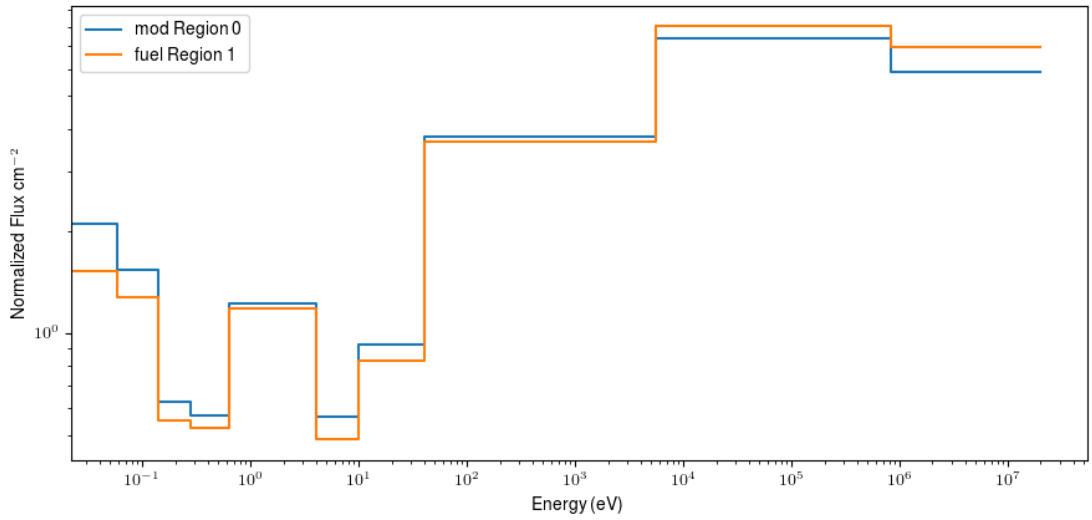

Figure 6: Pincell 10-group Flux

Sensitivity Analysis

Below are plots of the eigenvalue as a result of changing various parameters. Currently the code does not have the capability to lay a segment partially over a cell which presents a barrier to true sensitivity analysis because currently the tracklength and deadzone must be discretely varying parameters.

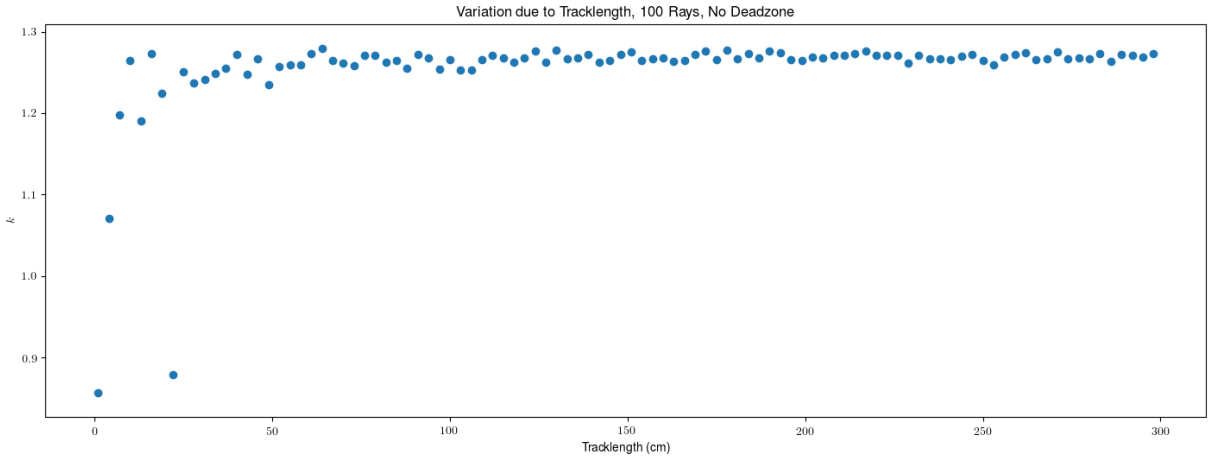

Figure 7: Tracklength

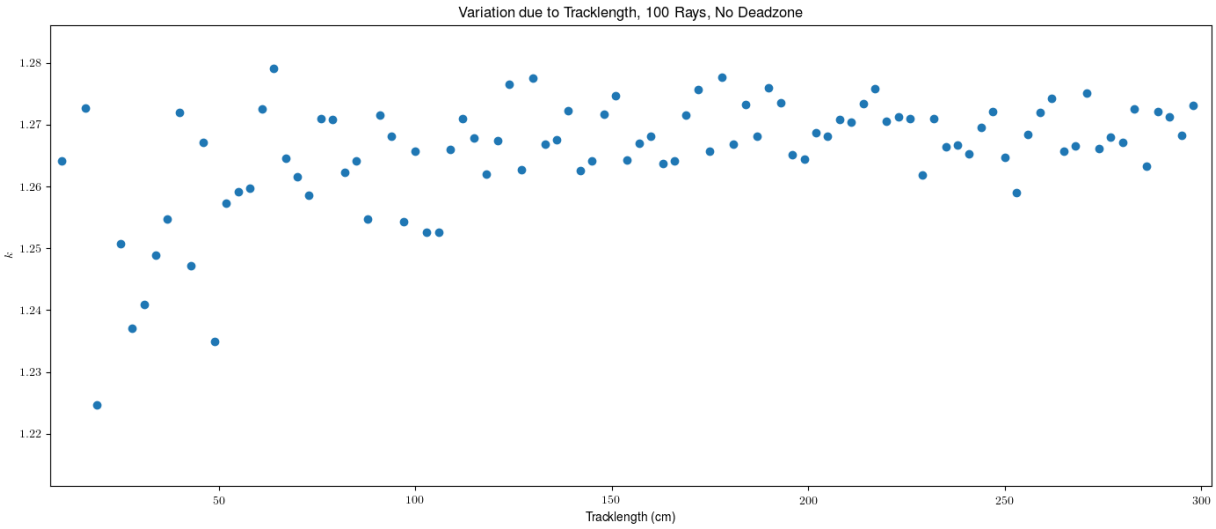

Figure 8: Tracklength

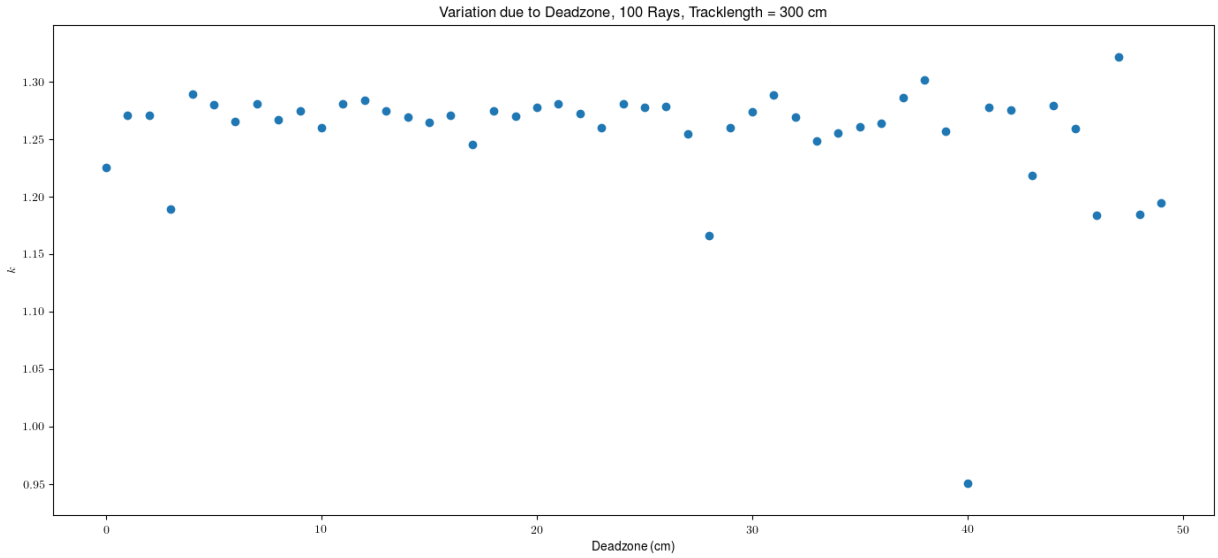

Figure 9: Deadzone

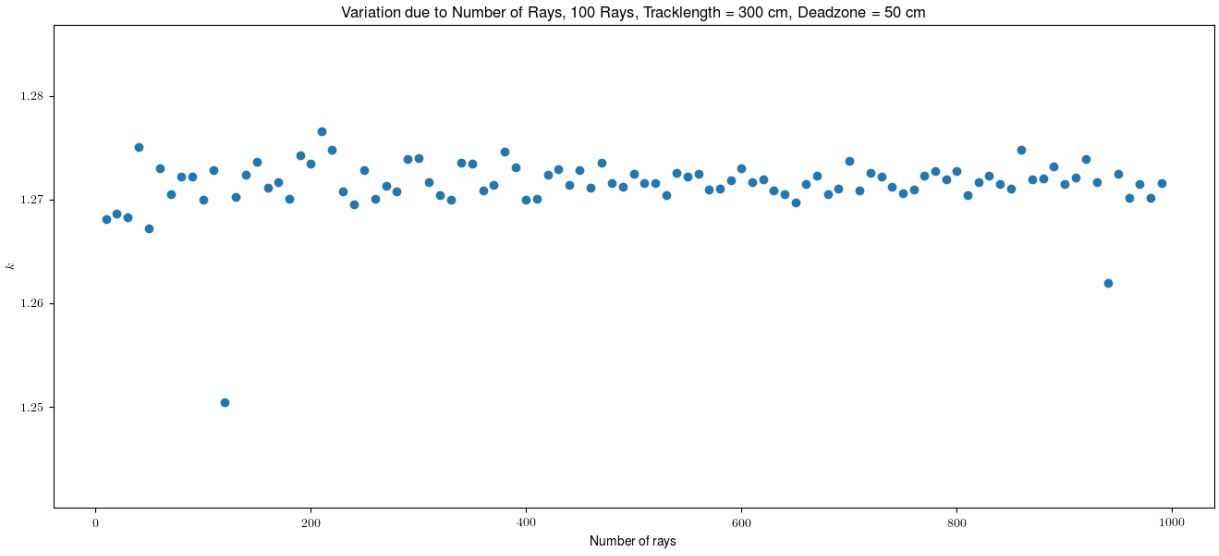

Figure 10: Number of Rays

At the codes current state this results are not quite expected. While changing the number of rays seems to generally converge the value, the sensitivity to dead zone and tracklength do not follow expected behavior. As stated before, this may be because currently the track placement only adds segments that fully traverse a region. This may introduce some bias in the sampling length of the material. Another issue may be that

100 rays is not enough to accurately capture the angullar dependence of the problem.

Note on performance

The main goal of this initial code was to produce a physical result. There are many aspects that could be improved in performance such as changing which operations need to be done inside of loops, more intelligent surfaces for ray tracing, and porting parts of the code to faster languages.

References

[1] J. R. Tramm, K. S. Smith, B. Forget, A. R. Siegel, The Random Ray Method for neutral particle transport, Journal of Computational Physics 342 (2017) 229–252. doi:10.1016/j.jcp.2017.04.038.

[2] W. Boyd, S. Shaner, L. Li, B. Forget, K. Smith, The openmoc method of characteristics neutral particle transport code, Annals of Nuclear Energy 68 (2014) 4352. doi:10.1016/j.anucene.2013.12.012.