NE-860: Numerical Transport

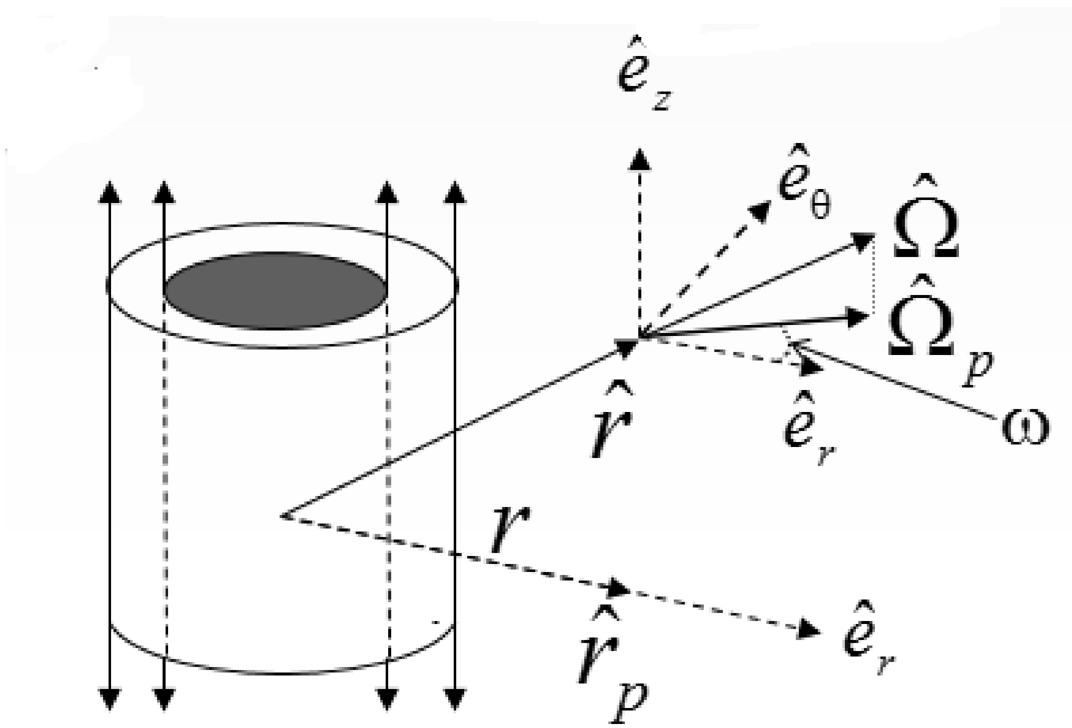
Homework 3

Date: 3/1/2017

Solutions by Michael Pfeifer

1) Derive the streaming term for the neutron transport equation in 1-D cylindrical geometry. Include diagrams to help explain your approach; You might try TikZ for these diagrams, though you are not required to do so.

Solution: Let us consider the cylindrical case



From this we may write the components to our angular flux

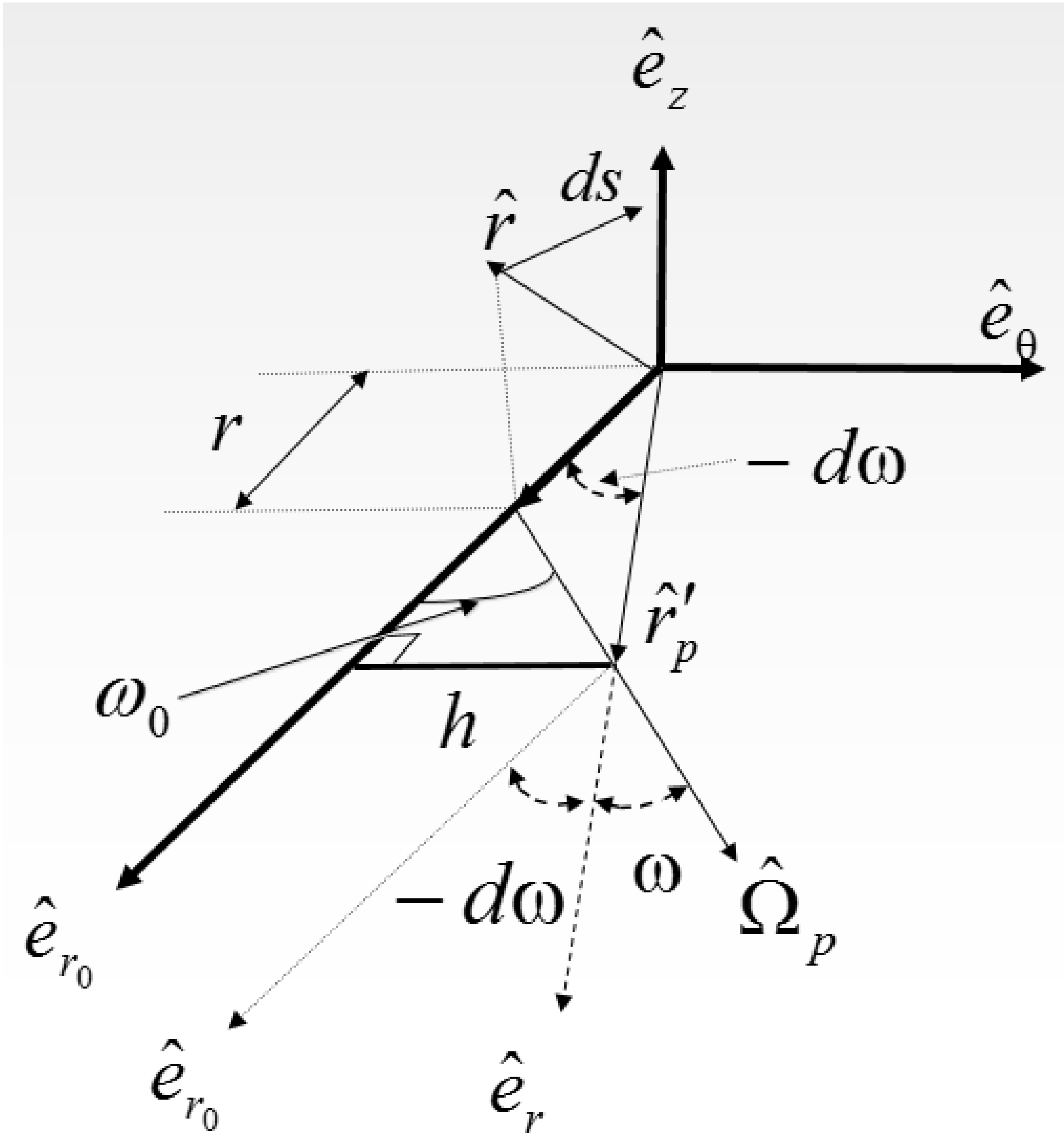
$$\frac{d\psi(s)}{ds} = \frac{d\psi}{dr}\frac{dr}{ds} + \frac{d\psi}{d\theta}\frac{d\theta}{ds} + \frac{d\psi}{dz}\frac{dz}{ds} + \frac{d\psi}{d\omega}\frac{d\omega}{ds} + \frac{d\psi}{d\eta}\frac{d\eta}{ds} + \frac{d\psi}{d\xi}\frac{d\xi}{ds} + \frac{d\psi}{dE}\frac{dE}{ds}$$

Simplify to include only the radial variables

$$\frac{d\psi(s)}{ds} = \frac{d\psi}{dr}\frac{dr}{ds} + \frac{d\psi}{d\omega}\frac{d\omega}{ds} + \frac{d\psi}{d\xi}\frac{d\xi}{ds}$$

Let μ to represent $\frac{dr}{ds}$ and write xi in terms of omega

$$\frac{d\psi(s)}{ds} = \mu\frac{d\phi}{dr} + \left[\frac{d\omega}{ds}\right]\frac{d\phi}{d\omega}$$



we may substitute the relation $-d\omega \cdot r = \eta \cdot ds$ to get the final answer

$$\boxed{\frac{d\psi(s)}{ds} = \mu \frac{d\phi}{dr} + \frac{\eta}{r}\frac{d\phi}{d\omega}}$$

2) Show that the first-flight kernel for a sperical shell is given by

$$k_{sph}(r,r') = \frac{r'}{2r} ki_1(\Sigma_t|r-r'| - \Sigma_t|r+r'|)$$

see Eq.(6.46) in the lecture notes (Integral Form, Exercises for a definition of ki_1.

Solution: The Bickley-Naylor function of order n where $Ki_0(r) = k_0(r)$, the zeroth order modified Bessel function is given as.

$$Ki_n(x) = \int_0^{\pi/2} \cos^{n-1}(\theta) e^{-x/\cos(\theta)} d\theta = \int_0^{\infty} \frac{e^x \cosh(u)}{\cosh^n(u)}$$

If we consider a domain D to be a sphere of radius R. The solution of the integral equations only depends on r. To solve this problem we must first take into account the relations

$$dr'^3 = r'^2 dr' d\mu d\psi$$

$$\frac{(r-r')}{r-r'} \cdot \vec{n} = \cos\theta = \frac{r - r'\cos\theta}{\sqrt[2]{r^2 - 2'^2 - 2rr'\cos\theta}}$$

Obtain the first flight kernel for the scalar flux

$$\phi(r) = \int_0^R [Q(r') + \Sigma_s(r')\phi(r')] K^\phi(r,r') dr'$$

Introduce a new integration variable $\mu = cos\theta$ while integrating over the azimuth. We get K^ϕ in this case is equal to

$$K^\phi(r,r') = \frac{r'^2}{2} \int_{-1}^{1} d\mu \frac{exp(-\Sigma \sqrt[2]{r^2 + r'^2 - 2rr'\mu})}{(r^2 + r'^2 - 2rr'\mu)}$$

Take the variable $t = \sqrt[2]{r^2 + r'^2 - 2rr'cos\theta}$ to get

$$K^\phi(r,r') = \frac{r'}{2r} \int_{|r-r'|}^{|r+r'|} \frac{e^{-\Sigma t}}{t} dt$$

Which give us our answer

$$\boxed{K^\phi(r,r') = \frac{r'}{2r}\left[E_1(\Sigma|r-r'|) - E_1(\Sigma|r+r'|)\right]}$$

3) Implement CPM in slab geometry using any language of your choice. Your code should define a main function solve_cpm that can be used as follows (in python):

```
x = [0.0, 1.0, 2.0, 3.0]        # cell boundaries
SigmaT = [1.0, 1.5, 1.0]        # macroscopic total cross section
SigmaS = [1.0, 1.5, 1.0]        # macroscopic scattering cross section
S = [0.0, 1.0, 0.0]             # external source
phi = solve_cmp(x, SigmaT, SigmaS, S)
```

You should produce a helper function compute_pij as part of your code that produces the matrix of collision probabilities from region j to region i. It should be used as follows:

$$pij = compute\_pij(x, SigmaT, SigmaS, S)$$

Finally, your code should include a function for plotting the fulx as a continuous function of x. Because CPM leads to discrete, cell-averaged values, you'll need to figure out how to plot what looks something like a histogram.

Solution: Three functions were defined in python for use in the CPM method. The first function solved for the τ matrix as well as the probability matrix p_{ij}. The second function solved the CPM method by forming the coefficient matrix H that when multiplied by the column vector X gives the solution matrix S. The third function plotted the values and found a line of best fit using a second order polynomial. The values for the found $\tau, p_{ij}, H, X, andS$ are shown below. The final set of solutions are a list of the flux values ϕ in order for each cell. One important note is that our flux values are symmetric, which is expected for this case.

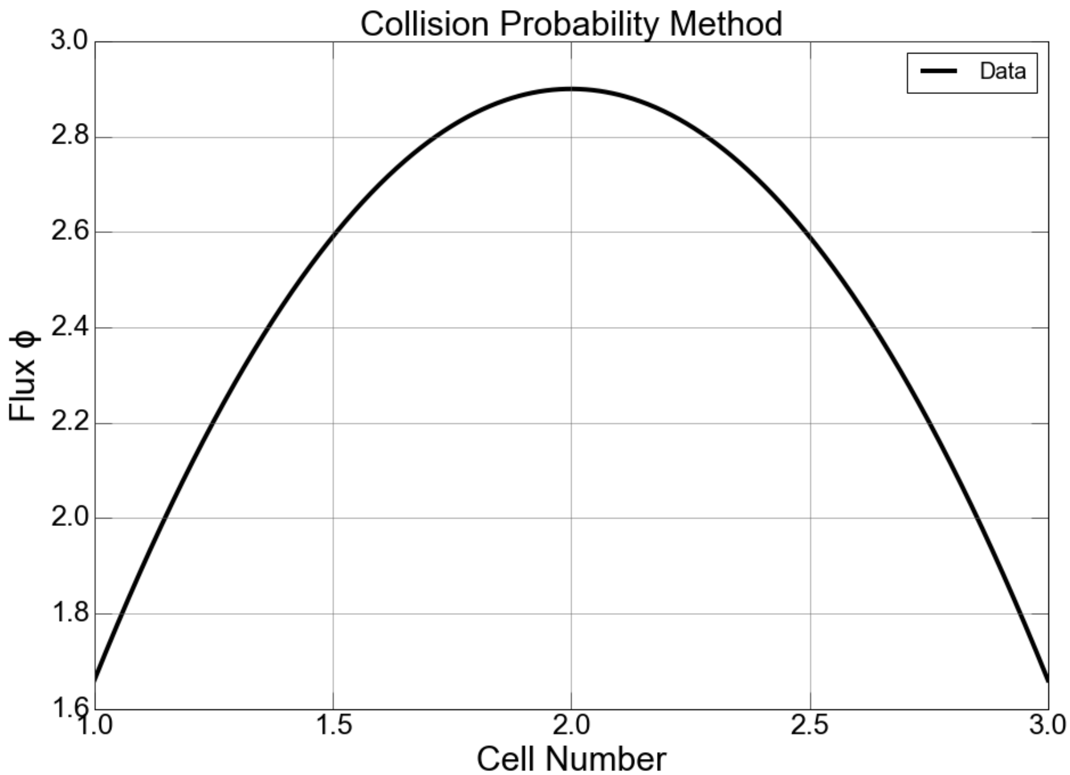
```
 This is the tau Matrix
[[ 0.   0.   1.5]
 [ 0.   0.   0. ]
 [ 1.5  0.   0. ]]


 This is the Pij Matrix
[[ 0.60969197  0.1166213   0.01454706]
 [ 0.17493196  0.70449299  0.17493196]
 [ 0.01454706  0.1166213   0.60969197]]


 This is the Solution Matrix
[[ 0.1166213 ]
 [ 0.70449299]
 [ 0.1166213 ]]


This is the X-Matrix
[[ 1.66036359]
 [ 4.34979295]
 [ 1.66036359]]


These are the flux values
[1.6603635887935277, 2.8998619671144925, 1.6603635887935275]
```



4) One you are sure your code for Problem 3 works, move to this problem, for which you are to derive and implement (1)"reflective" and (2) periodic boundary conditions. Your main code should be modified to have a left and right optional parameter were either can be set to vacuum or reflect. If one is set to periodic, then they both must be set to periodic.

Solution: The periodic and reflective conditions can be achieved by updating the probability matrix. For the white condition, we would define an albedo of 1 that would reflect all particles at the boundary. We then add our albedo probabilities to the previous probability and solve the CPM. For the periodic condition we would have an infinite series of a new probability term P_{lij} that would define our probability matrix for when $i \neq j$ as well as a new diagonal term that would be defined as the old term plus a infinite sum of P_{lij} from $-\infty$ to ∞ for $l \neq 0$.