start

```
import numpy as np
import pyg as ahp
import pym as ahm

# material appli-
# cation class
class matl(object):
 N_a = 6.022E23;
 def __init__(self):
  n_dens = \
       rho*N_a/M
 def apply(self,\
       region):
  # apply to geo
  # region
```

$$D(E)\nabla^2\phi(\vec{r},E) - \Sigma_a(E)\phi(\vec{r},E) + s(\vec{r},E) = \frac{1}{v(E)}\frac{\partial\phi(\vec{r},E)}{\partial t}$$

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$

$$\nabla^2 = \frac{1}{r}\frac{\partial}{\partial r}r\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial\theta^2} + \frac{\partial^2}{\partial z^2}$$

Calculate cross secitons:

```
# cross-section
# calculation
class xs(object):
 def __init__(self):
  # get za of all
  # materials
  for matl in matls:
   za = matl.za;
   xs = get_endf(za)
   sigma_s = xs[1,:]
   sigma_a = xs[2,:]
   sigma_f = xs[3,:]
```

Generate banded matrices:

$$A = \begin{bmatrix} 1 & -2 & 0 & 0 & 4 \\ -2 & 1 & -2 & 0 & 0 \\ 0 & -2 & 1 & -2 & 0 \\ 0 & 0 & -2 & 1 & -2 \\ 4 & 0 & 0 & -2 & 1 \end{bmatrix}$$

$$\vec{b} = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_4 \end{bmatrix}^T$$

Invert matrix in place:

```
# matrix inversion
fct = \
 fastCopyAndTranspose
def inv():
 m, n = a.shape
 at = fct(a)
 ipiv = zeros(m, intc)
 info = dgetrf(m, n, at, \
  m, ipiv, 0)[info]
 lu = fct(at)
 p = list(range(m))
 for k in range(m-1):
  ell = ipiv[k] - 1 p[k],
  p[ell] = p[ell], p[k]
```

Solve equations by gaussian elimination:

$$A\vec{\phi} = \vec{b}$$

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{bmatrix} = [A]^{-1}\vec{b}$$

Calculate the total and spacewise residual:

$$R_n = \phi_{n,i}(\vec{r},E) - \phi_{n,i-1}(\vec{r},E)$$

$$R = \sum_{k=1}^{n}\phi_{k,i}(\vec{r},E) - \phi_{k,i-1}(\vec{r},E)$$

Is the residual small enough?

$$R \leq \varepsilon$$

Output to visualization