NUEN 629
Homework 3
Paul Mendoza

Starting with the time-dependent multigroup diffusion equation:

$$\frac{1}{\bar{v}_g}\delta_t\phi_g - \nabla\cdot \mathrm{D_g}\nabla\phi_g + \Sigma_{rg}\phi_g = \sum_{g'=1,g'\neq g}^{G}\Sigma_{s,g'\to g}\,\phi_{g'} + \chi_\mathrm{g}\sum_{g'=1}^{G}\bar{\nu}\Sigma_{fg'}\phi_{g'} + Q_g$$

Grouping and defining the terms first (with multiple regions and forbearing the time term):

Fission Source Time Inscattering

$$Q^{*\ell,n}_{g,ijk} = \chi_\mathrm{g}\sum_{g'=1}^{G}\bar{\nu}\Sigma_{fg',ijk}\phi^{\ell,n}_{g',ijk} + Q_{g,ijk} + \frac{1}{v_g\Delta t}\phi^{\ell,n}_{g,ijk} + \sum_{g'=1,g'\neq g}^{G}\Sigma_{s,g'\to g}\,\phi^{\ell,n}_{g',ijk}$$

n indices added for time (would need to add to previous equations)

Absorb Outscatter Time

$$\Sigma^{*}_{rg,ijk} = \Sigma_{ag,ijk} + \sum_{g=1,g\neq g'}^{G}\Sigma_{s,g\to g'}\ (outscattering) + \frac{1}{v_g\Delta t}$$

For a 1 group problem the fission term should be coupled with the removal term (no summation just $\bar{\nu}\Sigma_{fg,ijk}$)

This simplifies our problem to:

$$-\nabla\cdot \mathrm{D_g}\nabla\phi^{\ell+1}_g + \Sigma^{*}_{rg,ijk}\phi^{\ell+1}_{g,ijk} = Q^{*\ell,n}_{g,ijk}$$

This turns into a matrix solve:

$$A\phi^{\ell+1}_{g,ijk} = Q^{*\ell}_{g,ijk}\left(\phi^{\ell}_{g,ijk}\right)$$

Finite Difference:

Discrete equations for a zone away from the boundary (2-D):

$$-\frac{\widehat{D}_{g,\left(i+\frac{1}{2}\right)jk}\left(\phi_{g,(i+1)jk}^{\ell+1}-\phi_{g,ijk}^{\ell+1}\right)-\widehat{D}_{g,\left(i-\frac{1}{2}\right)jk}\left(\phi_{g,ijk}^{\ell+1}-\phi_{g,(i-1)jk}^{\ell+1}\right)}{h_x^2}$$
$$-\frac{\widehat{D}_{g,i\left(j+\frac{1}{2}\right)k}\left(\phi_{g,i(j+1)k}^{\ell+1}-\phi_{g,ijk}^{\ell+1}\right)-\widehat{D}_{g,i\left(j-\frac{1}{2}\right)k}\left(\phi_{g,ijk}^{\ell+1}-\phi_{g,i(j-1)k}^{\ell+1}\right)}{h_y^2}+\Sigma_{rg,ijk}^{*}\phi_{g,ijk}^{\ell+1}$$
$$={Q^{*}}_{g,ijk}^{\ell}$$

Where:

$$\widehat{D}_{g,\left(i+\frac{1}{2}\right)jk}=\frac{2D_{g,ijk}D_{g,(i+1),jk}}{D_{g,ijk}+D_{g,(i+1),jk}}$$

For left boundary:

$$-\frac{\widehat{D}_{g,\left(\frac{3}{2}\right)jk}\left(\phi_{g,(2)jk}^{\ell+1}-\phi_{g,1jk}^{\ell+1}\right)-h_xJ_{g,x}(0,y_j,z_k)}{h_x^2}$$
$$-\frac{\widehat{D}_{g,1\left(j+\frac{1}{2}\right)k}\left(\phi_{g,1(j+1)k}^{\ell+1}-\phi_{g,1jk}^{\ell+1}\right)-\widehat{D}_{g,1\left(j-\frac{1}{2}\right)k}\left(\phi_{g,1jk}^{\ell+1}-\phi_{g,1(j-1)k}^{\ell+1}\right)}{h_y^2}+\Sigma_{rg,1jk}^{*}\phi_{g,1jk}^{\ell+1}$$
$$={Q^{*}}_{g,1jk}^{\ell}$$

Where:

$$J_{g,x}(0,y_j,z_k)=-\frac{D_{g,ijk}}{B}\left(C-\frac{A}{2}\left(3\phi_{1jk}-\phi_{2jk}\right)\right)$$

If B =0

$$J_{g,x}(0,y_j,z_k)=-\frac{D_{g,ijk}\left(\phi_{g,1jk}-\frac{C}{A}\right)}{\frac{h_x}{2}}$$

Some algebra is needed to figure what goes where in A, but it is the hope of the student that Dr. McClarren's code already helps with this.

In short the finite difference code settled upon looks like:

```
import numpy as np
import scipy as sp
import scipy.sparse as sparse
import scipy.sparse.linalg as splinalg
import time
import sys

tart_time = time.time()

#Functions
def coordLookup_l(i, j, k, I, J):
def coordLookup_ijk(l, I, J):
def diffusion_steady_fixed_source(I,J,K,Nx,Ny,Nz,hx,hy,hz,ihx2,ihy2,ihz2,BCs,D,Sigma,Q,L,…):
def lattice2G():
def inner_iteration(G,I,J,K,Nx,Ny,Nz,hx,hy,hz,ihx2,ihy2,ihz2,BCGs,Sigmarg,Sigmasgg,D,Q,L,…):
def kproblem_mg_diffusion(I,J,K,G,Nx,Ny,Nz,hx,hy,hz,ihx2,ihy2,ihz2,BCGs,Sigmarg,Sigmasgg,…):
def Power_find(I,J,K,Power):
def plotting(phig,Powermap):

#Define the Problem
Power,hx,hy,hz,ihx2,ihy2,ihz2,x,y,z,Sigmarg,Sigmasgg,nuSigmafg,nug,chig,D,Q,L,I,J,K,Nx,Ny,Nz,G,BC
Gs = lattice2G()
#k Convergence
k,iterations,phig =
kproblem_mg_diffusion(I,J,K,G,Nx,Ny,Nz,hx,hy,hz,ihx2,ihy2,ihz2,BCGs,Sigmarg,Sigmasgg,nuSigmafg,ch
ig,D,L,…)
#Determine Assembly Power Profile
Powermap=Power_find(I,J,K,Power)
#End the Run Time
print("--- %s ---" % (time.time() - start_time))
#Make the plots
plotting(phig,Powermap)
```

Step by step this code does the following:

1. Sets up the geometry
    a. The geometry setup is read from a geometry file, which provides the assembly type at each i,j,k location of the problem. This file is generated from an excel file which produces the geometry for a given number of nodes in the problem.
    b. The assembly numbers are read from a power file, which provides the assembly number at each i,j,k location. This file is generated from an excel file which determines which locations are of each assembly for a given number of nodes in the problem.
    c. Subsequently other variables (cross section variables, spacing, length…) are produced either by loops or are read or calculated.
2. Runs through the kproblem diffusion solver
    a. This diffusion solver will start with a random flux for domain if no previous runs for the problem were completed. Otherwise, the program will search for old flux configurations and interpolate flux values for the current geometry and start with that for the initial guess
    b. The diffusion solver runs through an iterative loop to converge the flux and determine k. This loop:
        i. Determines the fission source term

$$Q^{*\ell,n}_{g,ijk} = \chi_g \sum_{g'=1}^{G} \bar{\nu}\Sigma_{fg',ijk}\phi^{\ell,n}_{g',ijk}$$

ii. Passes this term, along with the flux to an inner convergence loop which iteratively solves for the flux group by group by iteratively updating the scattering source and solving the system.

$$A\phi_{g,ijk}^{\ell+1} = Q^{*\ell}{}_{g,ijk}(\phi_{g,ijk}^{\ell})$$

Where:

$$A = -\nabla \cdot \mathrm{D_g}\nabla + \Sigma_{rg,ijk}^{*}$$

$$Q^{*\ell,n}{}_{g,ijk} = \left[\chi_{\mathrm{g}} \sum_{g'=1}^{G} \bar{\nu}\Sigma_{fg',ijk}\phi_{g',ijk}^{\ell}\right]_{constant} + \left[\sum_{g'=1,g'\neq g}^{G} \Sigma_{s,g'\to g}\,\phi_{g',ijk}^{\ell+i}\right]_{updated}$$

And $-\nabla \cdot \mathrm{D_g}\nabla$ is given at the top of page two.

iii. Once a new $\phi_{g,ijk}^{\ell+1}$ is determined, a new k is calculated by

$$k_{new} = \frac{\left\|\phi_{g,ijk}^{\ell+1}\right\|}{\left\|\phi_{g,ijk}^{\ell}\right\|}$$

iv. If this new multiplication factor is within 1e-5 of the old one, the iteration stops.

c. Once the flux has been determined the power in each assembly is determined by the equation

$$P_{assy} = \sum_{g=1}^{2}\sum_{n}^{N} dxdydz\ 200\ \phi_{ijkg}\nu\Sigma_{fg}$$

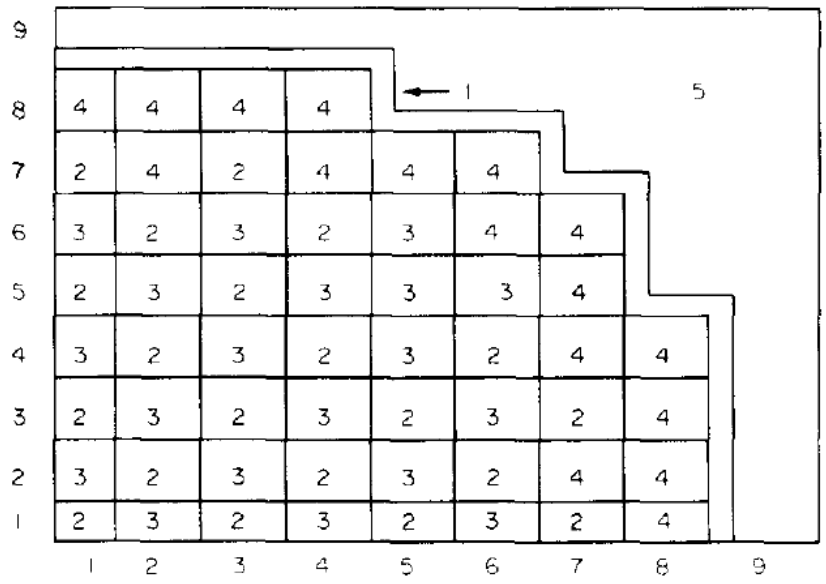
This loop looks like:

```
Powermat=np.zeros(int(Power[1:,:].max()))
  for assy in range(int(Power[1:,:].max())):
    for k in range(K):
      for j in range(J):
        for i in range(I):
          if(assy+1==int(Power[J-j,i])):
            for g in range(G):
              if(assy==0):
               Powermat[assy] +=4*hx*hy*hz*200*phig[i,j,k,g]*nuSigmafg[i,j,k,g]
              elif(i<iswitch):
               Powermat[assy] +=2*hx*hy*hz*200*phig[i,j,k,g]*nuSigmafg[i,j,k,g]
              elif(j<jswitch):
               Powermat[assy] +=2*hx*hy*hz*200*phig[i,j,k,g]*nuSigmafg[i,j,k,g]
              else:
               Powermat[assy] +=hx*hy*hz*200*phig[i,j,k,g]*nuSigmafg[i,j,k,g]
```

The assembly in the middle is multiplied by 4 because the code only has ¼ of the assembly shown, the assemblies on the edge are multiplied by 2 because the code has only ½ of the assembly shown.

The first Problem is described below:

The Zion PWR Benchmark (K.S. Smith, NSE, 1986), is a 2-D, two-group, reactor benchmark calculation where a quarter reactor is specified by the following figure:



Composition	Group, g	D_g	Σ_{ag}	$\nu\Sigma_{fg}$	$\Sigma_{gg'}$
1	1	1.02130	0.00322	0.0	0.0
	2	0.33548	0.14596	0.0	0.0
2	1	1.47160	0.00855	0.00536	0.01742
	2	0.37335	0.06669	0.10433	0.0
3	1	1.41920	0.00882	0.00601	0.01694
	2	0.37370	0.07606	0.12472	0.0
4	1	1.42650	0.00902	0.00653	0.01658
	2	0.37424	0.08359	0.14120	0.0
5	1	1.45540	0.00047	0.0	0.02903
	2	0.28994	0.00949	0.0	0.0

$X_1 = 1.0$, $X_2 = 0.0$.
Assembly pitch: 21.608 cm. Baffle thickness: 2.8575 cm.
Boundary condtions: reflective: left, bottom, zero flux: top, right.
$\Sigma_{tr,g} = 1/3D_g$ for isotropic scattering, transport problem.

Figure 1: Zion PWR benchmark from K.S. Smith, NSE, 1986.

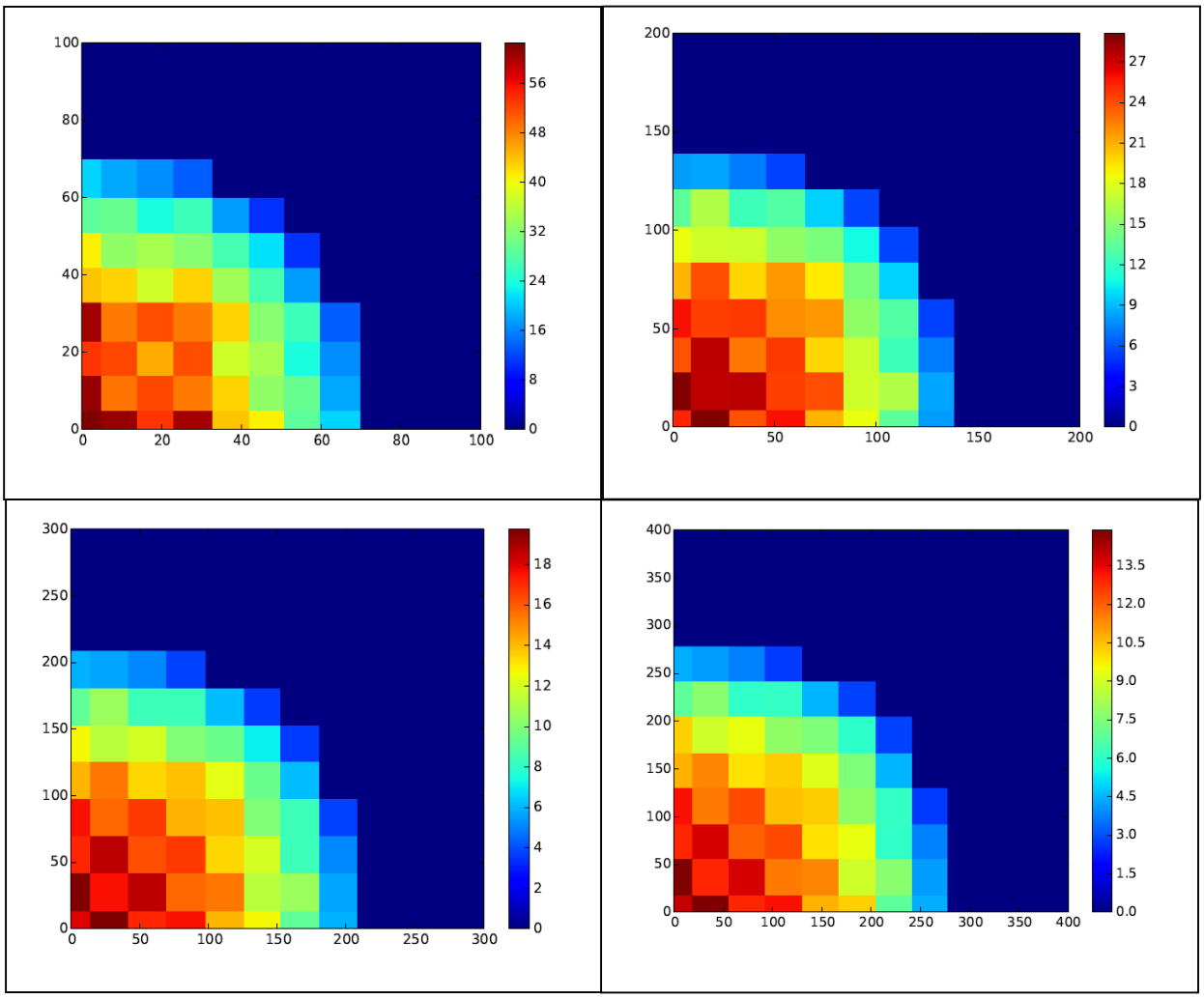
A quarter of the reactor is a square that is divided in the figure into a 9 x 9 grid of 21.608 x 21.608 cm squares (i.e., the quarter reactor size is 194.472 x 194.472 cm). At the midline there are 8 assemblies in the x and y directions. In the problem specification there is only downscattering and what we have called Σ_{rg} is called Σ_{ag}. Also, note that all fission neutrons are born fast.

1. Solve this problem using the finite difference method.
   a. Find k_{eff} and the fission power in each assembly.

Number of Nodes	dx (cm)	k effective	time (s)	k iterations	t per iteration
100	2.32	1.27507	53.9	7	8
200	1.16	1.2749	204.97	6	34
300	0.7747	1.27484	587.62	7	84
400	0.58104	1.27482	193.44	2	97

The time includes setting up the geometry but does not include producing the graphs. These times would be longer if old solutions were not used for the more refined mesh, so probably are not indicative of the efficiency of the algorithm.

The power in each assembly, as determined by the above algorithm, is shown below with the different mesh resolutions:



With each refinement of the mesh, specific assemblies with higher fissile material in the center of the core are highlighted as producing more power. The central pin, although seeing the highest fluxes does not produce the most power because it has the lowest enrichment of the assemblies.

The absolute value on each mesh is arbitrary and in order to relate to power the sum total of the flux and percent of fission energy deposited in the reactor should be considered. Along this line of thought, the sum flux for the 300 and 400 cases are provided for the thermal and fast groups.

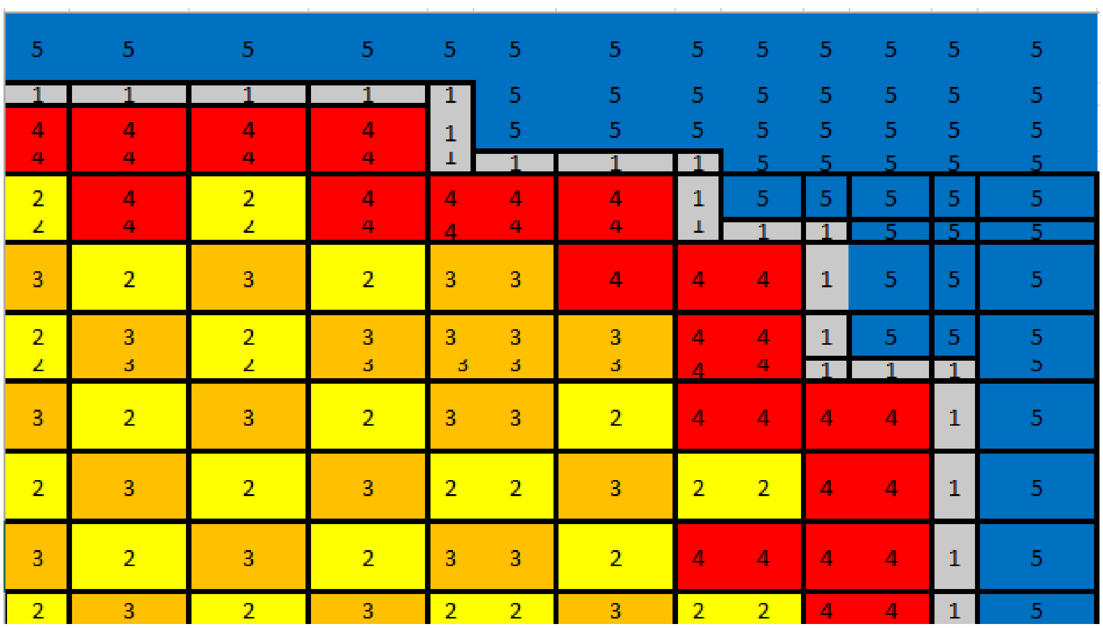
Number of Nodes	Flux Fast Group Sum	Flux 1 Group Sum	Total
300	139	34.9	173.9
400	185	46.1	231.1

Please note that the fast flux group sum is larger than the thermal group sum. There should be an explanation for this.

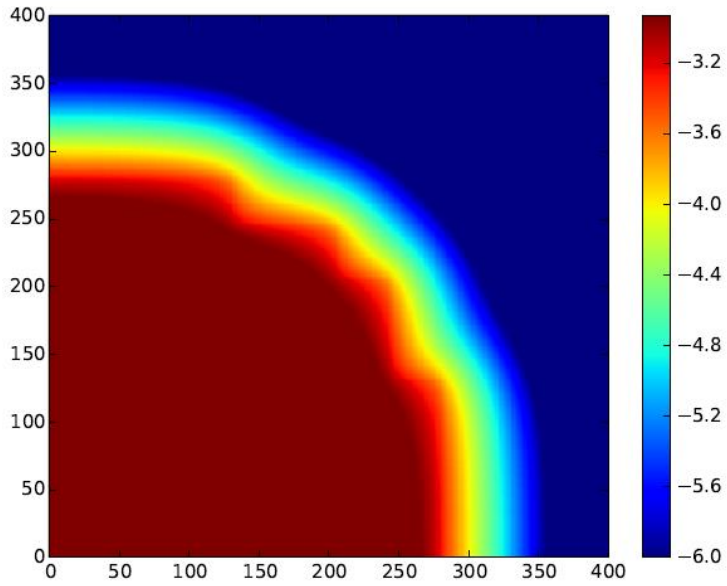
The mesh resolution should be around 0.58104 to resolve the multiplication factor to 1 pcm.

Although the problem does not ask for the following plots, they are pretty and provided below:

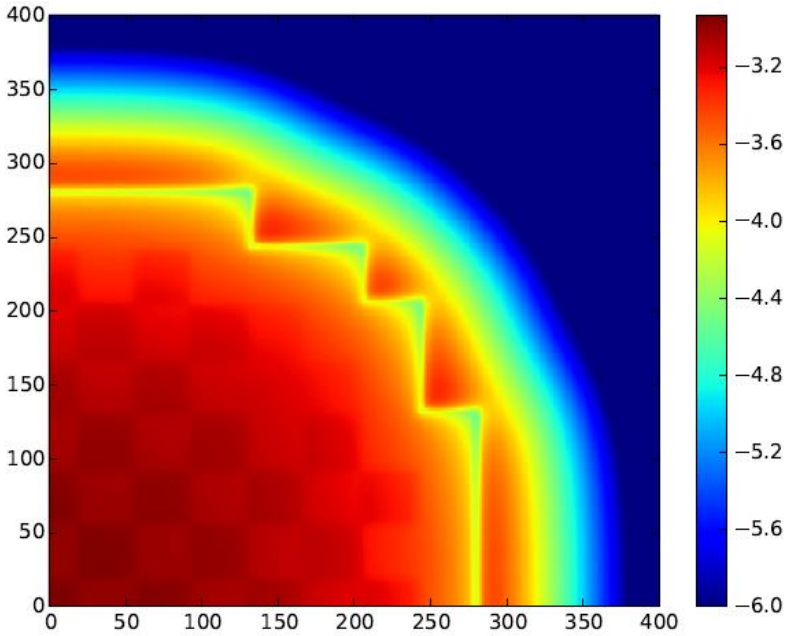
First a picture of the geometry as set up in excel:



The fast flux did not change much with increasing resolution:



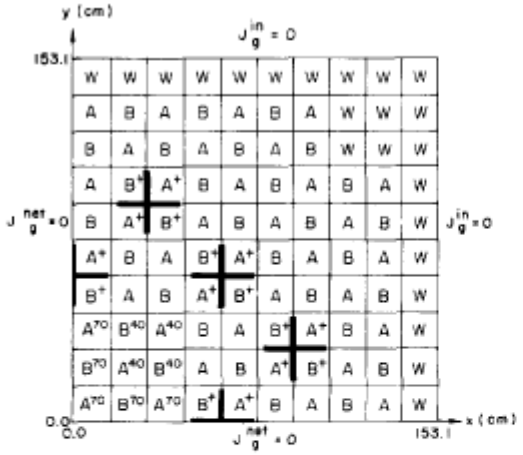
Neither did the thermal flux:



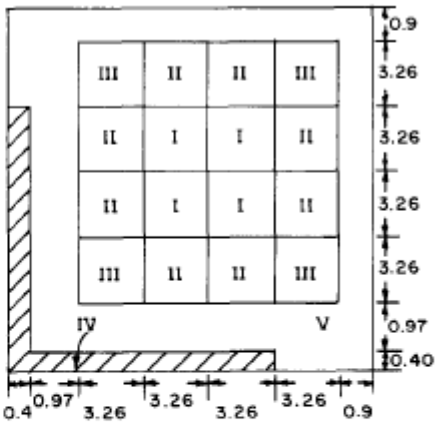
The second problem is described below:

The HAFAS BWR Benchmark (K.S. Smith, NSE, 1986) is a 2-D, two-group, reactor benchmark calculation with pin-level homogenization where a quarter reactor is specified by different assemblies as whown in the following figure.

Quadrant of the two-dimensional Reactor



Fuel Assembly Description



Composition-to-Zone Assignments

Zone	Assembly Type A	A^{40}	A^{70}	A^+	B	B^{40}	B^{70}	B^+	W
I	1	5	9	1	2	6	10	2	15
II	2	6	10	2	3	7	11	3	15
III	3	7	11	3	4	8	12	4	15
IV	13	13	13	14	13	13	13	14	15
V	13	13	13	13	13	13	13	13	15

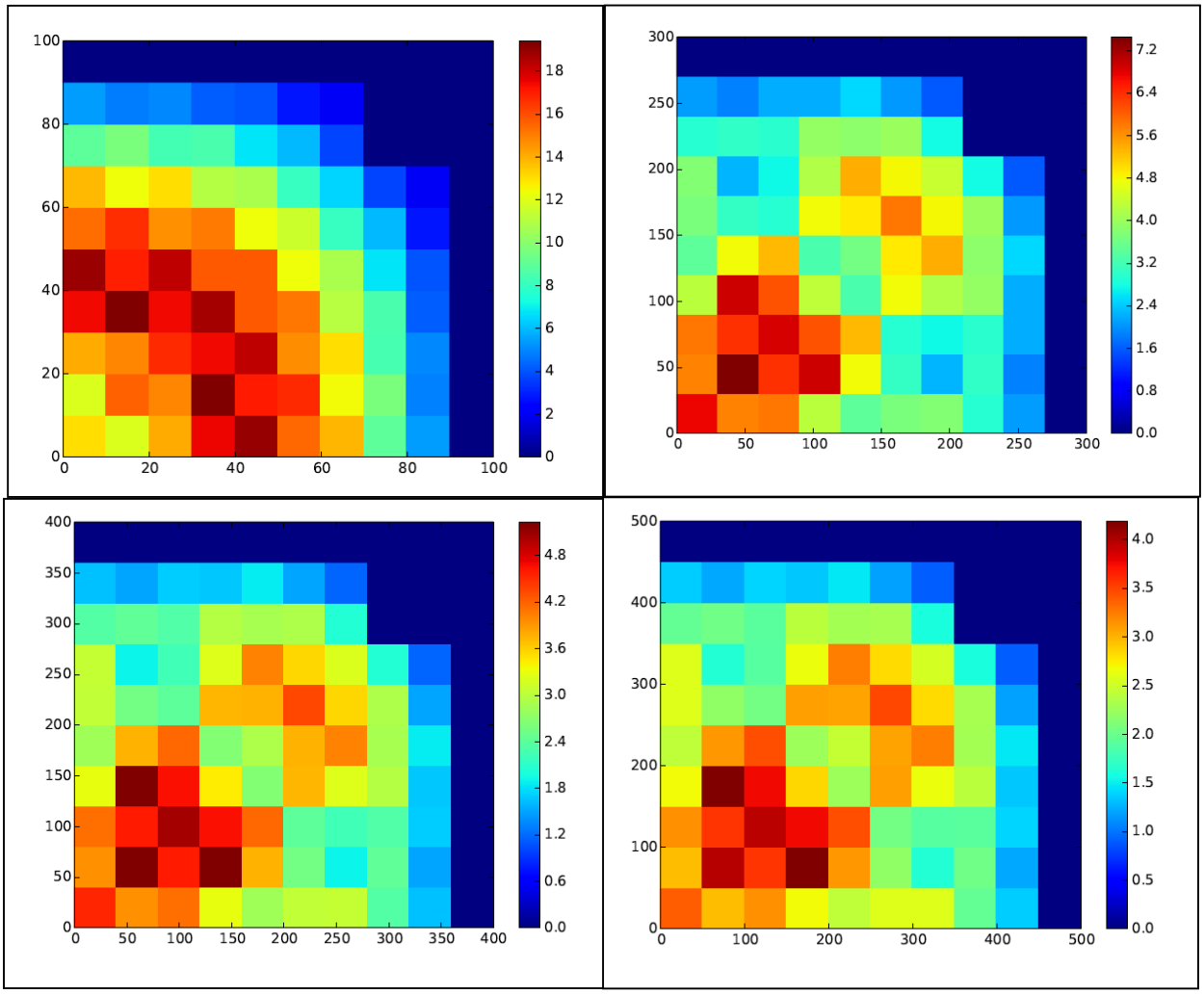
2. Solve this problem using the finite difference method.
   a. Find k_{eff} and the fission power in each assembly.

Number of Nodes	dx (cm)	k effective	time (s)	k iterations	t per iteration	t per single iteration
100	1.531	1.07645	237.4	36	6.58	8.52
200	0.7655	1.03306	3016.8	94	32.1	21.75
300	0.51033	1.03684	2744.5	32	85.77	54.40
400	0.3828	1.04213	7399.5	43	172.1	105.48
500	0.3062	1.05022	11236.1	39	288.1	187

The time includes setting up the geometry but does not include producing the graphs. These iterations were completed with an older version of the code that was slower (the inner iteration started with a

zero flux instead of a previous iterations flux). These times would be longer if old solutions were not used for the more refined mesh, so probably are not indicative of the efficiency of the algorithm.

The power in each assembly, as determined by the above algorithm (without multiplying by 2 or 4 for the central assemblies because a whole assembly is modeled), is shown below with the different mesh resolutions:



With each refinement of the mesh, specific assemblies with higher fissile material in the center of the core are highlighted as producing more power. The assemblies with control rods very clearly produced less power. There are two hearts in some of the pictures, if this answer is wrong, I do not want to be right.

The absolute value on each mesh is arbitrary and in order to relate to power the sum total of the flux and percent of fission energy deposited in the reactor should be considered. Along this line of thought, the sum flux for each cases are provided for the thermal and fast groups.

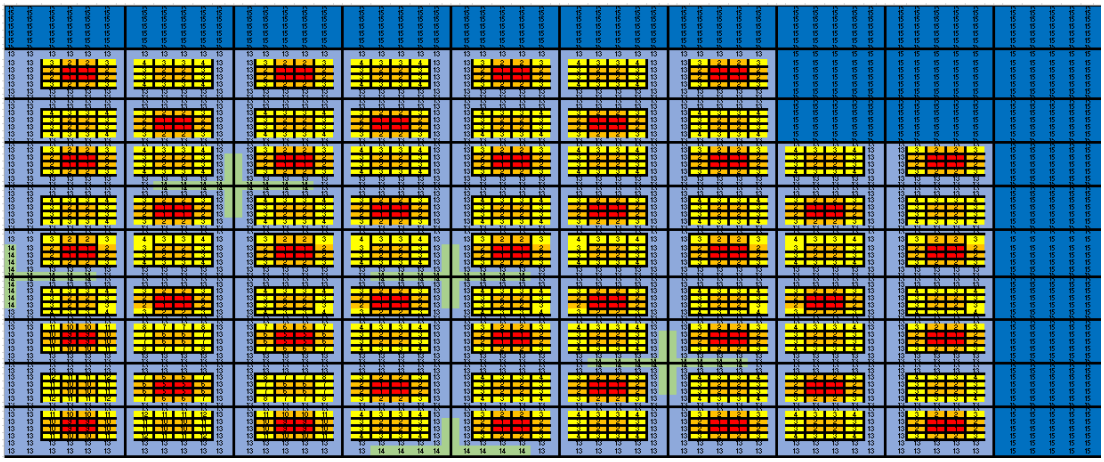
Number of Nodes	Group Thermal Sum	Group Fast Sum	Total
100	37	63	100
200	67.8	143	210.8
300	104	211	315
400	135	285	420
500	164	359	523

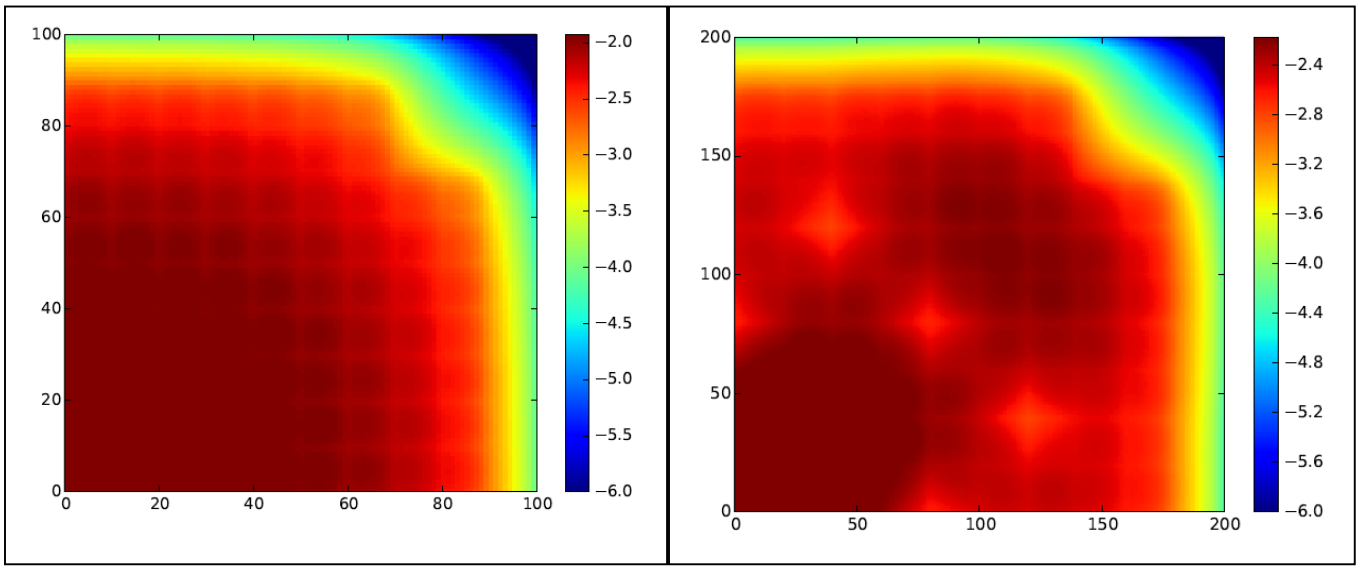
Please note that the fast flux group sum is larger than the thermal group sum. There should be an explanation for this.

The mesh resolution should be around 0.306 to resolve the multiplication factor to 1 pcm.

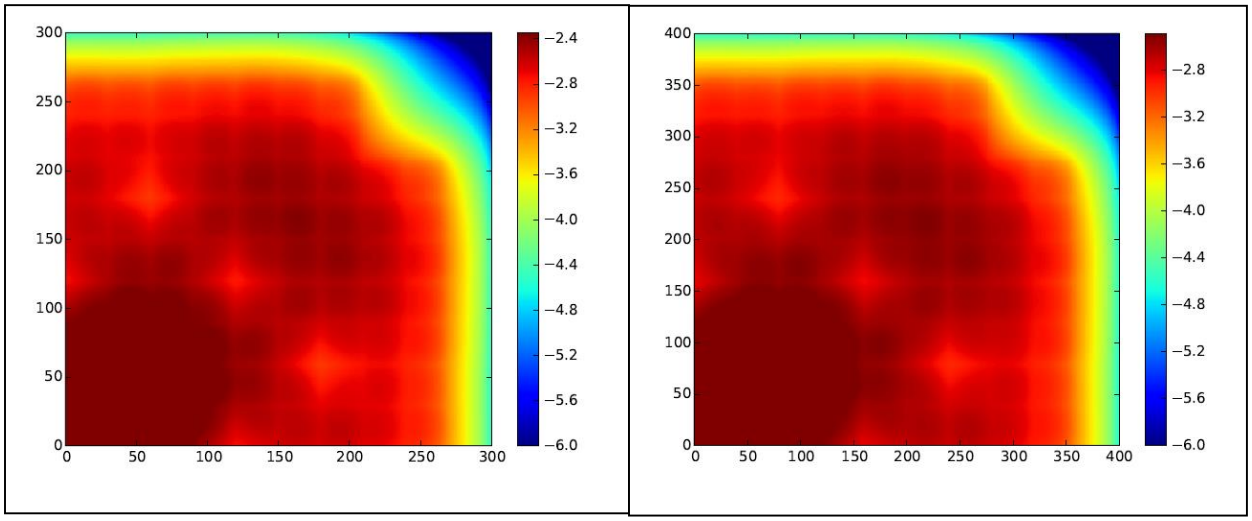
Although the problem does not ask for the following plots, they are pretty and provided below:

First a picture of the geometry for the problem (picture from excel):

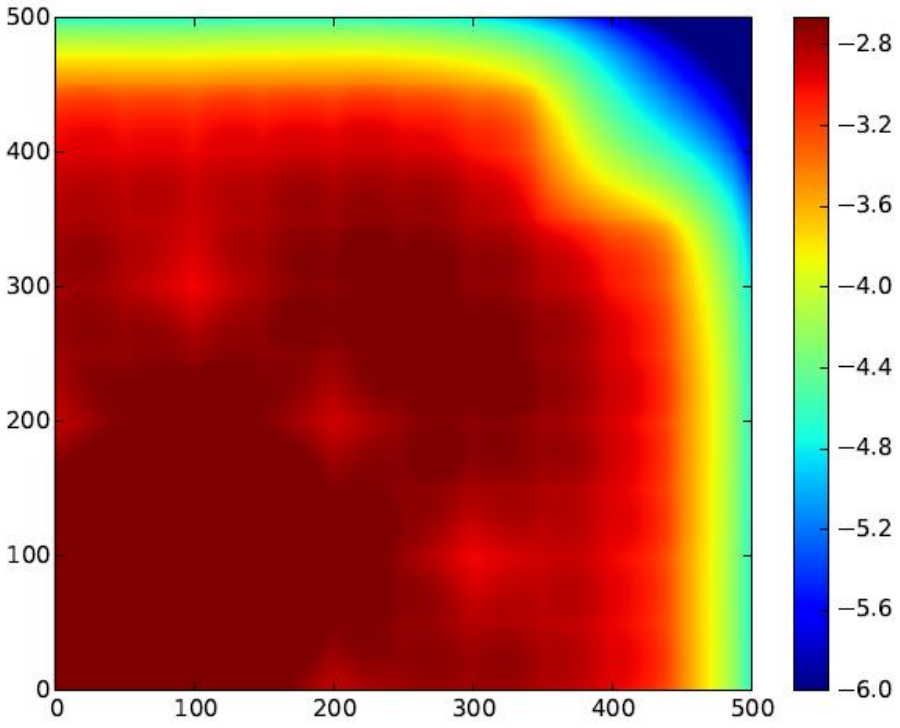




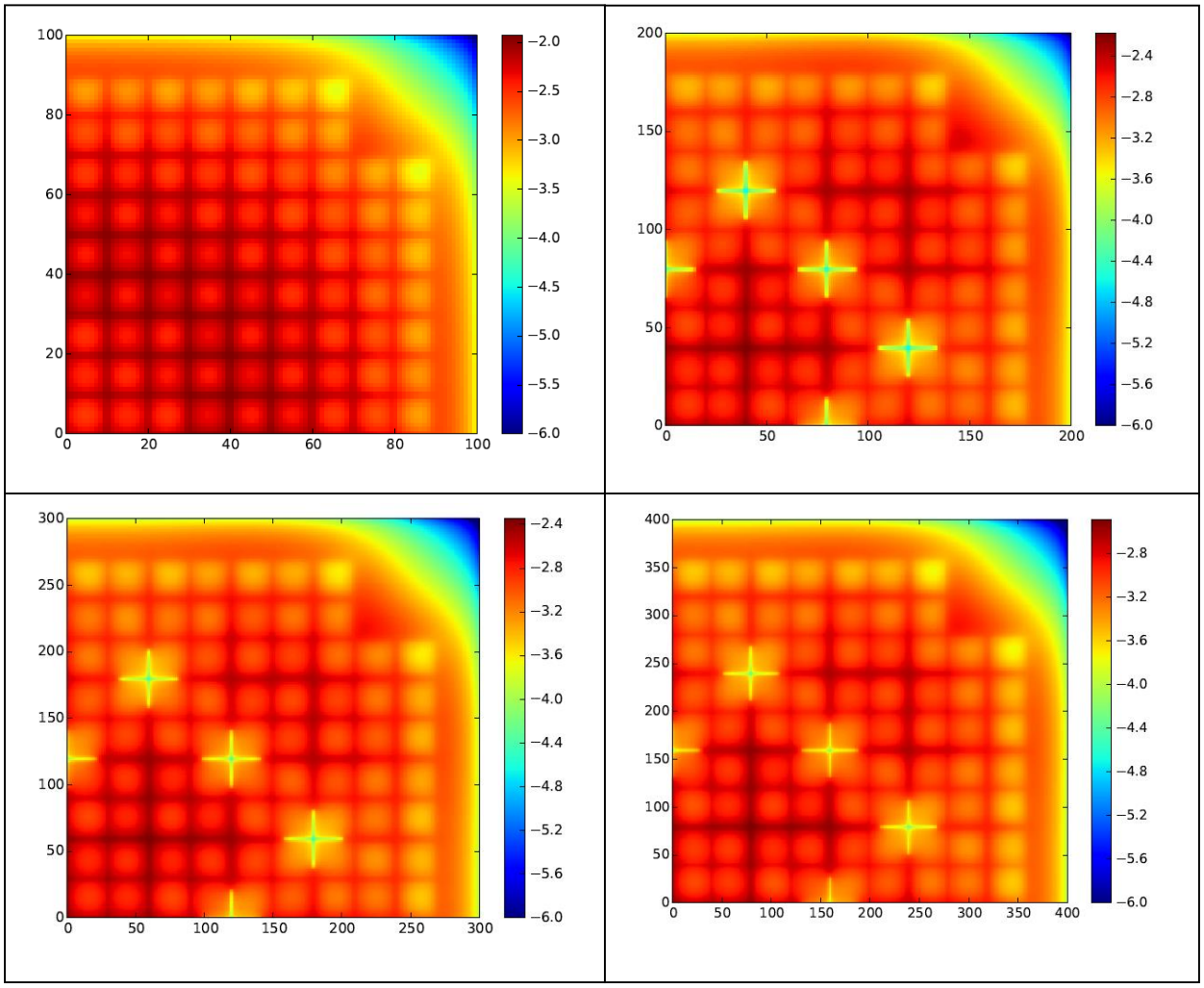
The fast flux did change with increasing resolution because the control rods were captured in different ways:

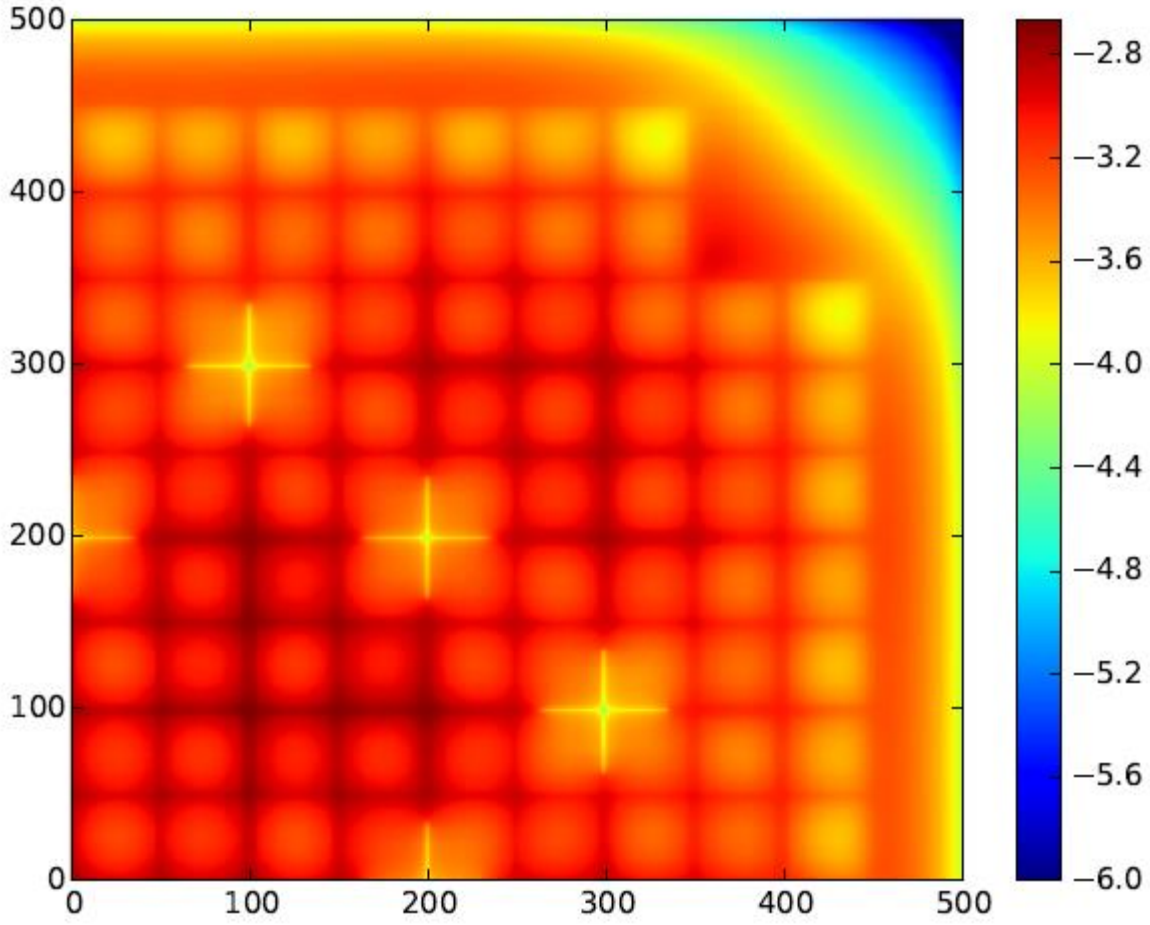


As the resolution increased different aspects of the fast flux were captured. There is a heart in there, showing my love for this homework assignment.



The thermal flux changed with resolution because the control rods were captures with increasing mesh.




500
400
300
200
100
0
0
100
200
300
400
500
−2.8
−3.2
−3.6
−4.0
−4.4
−4.8
−5.2
−5.6
−6.0