Problem Set 3

22.S904 Nuclear Reactor Kinetics

Due: 3 October 2012

Bryan Herman

Diffusion Code

A general three-dimensional second order finite volume code was written to solve the neutron diffusion equation for both steady state and transient applications. **The source code can be reviewed at:**

http://github.com/bhermanmit/Kinetics/tree/master/HW3/src.

Derivation of Transient Diffusion Equations

We can begin the derivation with the multigroup form of the neutron diffusion equation and precursor balance equation:

$$\frac{1}{v_g(\vec{r})}\frac{\partial}{\partial t}\phi_g(\vec{r},t) = \nabla\cdot D_g(\vec{r},t)\nabla\phi_g(\vec{r},t) - \Sigma_{tg}(\vec{r},t)\phi_g(\vec{r},t) + \sum_h \Sigma_{s,h\to g}(\vec{r},t)\phi_h(\vec{r},t)$$
$$+ [1-\beta(\vec{r})]\frac{\chi_g^p(\vec{r})}{k_{crit}}\sum_h \nu\Sigma_{sh}(\vec{r},t)\phi_h(\vec{r},t) + \sum_i \chi_{ig}^d \lambda_i C_i(\vec{r},t) \qquad g = 1,...,G$$

$$\frac{\partial}{\partial t}C_i(\vec{r},t) = -\lambda_i C_i(\vec{r},t) + \frac{\beta_i(\vec{r})}{k_{crit}}\sum_h \nu\Sigma_{sh}(\vec{r},t)\phi_h(\vec{r},t) \qquad i = 1,...,I.$$

In these equations we assume that velocity and other kinetics parameters are not strongly varying with time. In general, these parameters could be interpolated on from resulting lattice calculations. We can now discretize this set of equations in time using implicit Euler. The time step will be denoted as n. The form of the equation is now

$$\frac{1}{v_g\left(\vec{r}\right)dt^n}\left[\phi_g^{n+1}\left(\vec{r}\right)-\phi_g^n\left(\vec{r}\right)\right]=\nabla\cdot D_g^{n+1}\left(\vec{r}\right)\nabla\phi_g^{n+1}\left(\vec{r}\right)-\Sigma_{tg}^{n+1}\left(\vec{r}\right)\phi_g^{n+1}\left(\vec{r}\right)+\sum_h\Sigma_{s,h\rightarrow g}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)$$
$$+\left[1-\beta\left(\vec{r}\right)\right]\frac{\chi_g^p\left(\vec{r}\right)}{k_{crit}}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)+\sum_i\chi_{ig}^d\lambda_iC_i^{n+1}\left(\vec{r}\right)$$

$$\frac{1}{dt^n}\left[C_i^{n+1}\left(\vec{r}\right)-C_i^n\left(\vec{r}\right)\right]=-\lambda_iC_i^{n+1}\left(\vec{r}\right)+\frac{\beta_i\left(\vec{r}\right)}{k_{crit}}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right).$$

We move all the $n+1$ terms to the left hand side and everything else to the left,

$$-\nabla\cdot D_g^{n+1}\left(\vec{r}\right)\nabla\phi_g^{n+1}\left(\vec{r}\right)+\left[\Sigma_{tg}^{n+1}\left(\vec{r}\right)+\frac{1}{v_g\left(\vec{r}\right)dt^n}\right]\phi_g^{n+1}\left(\vec{r}\right)-\sum_h\Sigma_{s,h\rightarrow g}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)$$
$$-\left[1-\beta\left(\vec{r}\right)\right]\frac{\chi_g^p\left(\vec{r}\right)}{k_{crit}}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)-\sum_i\chi_{ig}^d\lambda_iC_i^{n+1}\left(\vec{r}\right)=\frac{1}{v_g\left(\vec{r}\right)dt^n}\phi_g^n\left(\vec{r}\right)$$

$$C_i^{n+1}\left(\vec{r}\right)\left(1+\lambda_idt^n\right)-\frac{\beta_i\left(\vec{r}\right)dt^n}{k_{crit}}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)=C_i^n\left(\vec{r}\right).$$

Now, we solve the precursor equation for $C_i^{n+1}\left(\vec{r}\right)$ and plug it into the neutron diffusion equation,

$$C_i^{n+1}\left(\vec{r}\right)=\frac{1}{1+\lambda_idt^n}C_i^n\left(\vec{r}\right)+\frac{\beta_i\left(\vec{r}\right)dt^n}{k_{crit}\left(1+\lambda_idt^n\right)}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right).$$

$$-\nabla\cdot D_g^{n+1}\left(\vec{r}\right)\nabla\phi_g^{n+1}\left(\vec{r}\right)+\left[\Sigma_{tg}^{n+1}\left(\vec{r}\right)+\frac{1}{v_g\left(\vec{r}\right)dt^n}\right]\phi_g^{n+1}\left(\vec{r}\right)-\sum_h\Sigma_{s,h\rightarrow g}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)$$
$$-\left[1-\beta\left(\vec{r}\right)\right]\frac{\chi_g^p\left(\vec{r}\right)}{k_{crit}}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)$$
$$-\sum_i\chi_{ig}^d\lambda_i\left[\frac{1}{1+\lambda_idt^n}C_i^n\left(\vec{r}\right)+\frac{\beta_i\left(\vec{r}\right)dt^n}{k_{crit}\left(1+\lambda_idt^n\right)}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)\right]=\frac{1}{v_g\left(\vec{r}\right)dt^n}\phi_g^n\left(\vec{r}\right).$$

We can now collect terms and rearrange again,

$$C_i^{n+1}\left(\vec{r}\right)=\frac{1}{1+\lambda_idt^n}C_i^n\left(\vec{r}\right)+\frac{\beta_i\left(\vec{r}\right)dt^n}{k_{crit}\left(1+\lambda_idt^n\right)}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right).$$

$$-\nabla\cdot D_g^{n+1}\left(\vec{r}\right)\nabla\phi_g^{n+1}\left(\vec{r}\right)+\left[\Sigma_{tg}^{n+1}\left(\vec{r}\right)+\frac{1}{v_g\left(\vec{r}\right)dt^n}\right]\phi_g^{n+1}\left(\vec{r}\right)-\sum_h\Sigma_{s,h\rightarrow g}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)$$
$$-\left\{\left[1-\beta\left(\vec{r}\right)\right]\frac{\chi_g^p\left(\vec{r}\right)}{k_{crit}}+\sum_i\frac{\chi_{ig}^d\lambda_i\beta_i\left(\vec{r}\right)dt^n}{k_{crit}\left(1+\lambda_i dt^n\right)}\right\}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)$$
$$=\frac{1}{v_g\left(\vec{r}\right)dt^n}\phi_g^n\left(\vec{r}\right)+\sum_i\frac{\chi_{ig}^d\lambda_i}{1+\lambda_i dt^n}C_i^n\left(\vec{r}\right).$$

We make one last approximation and that is the delay spectrum is independent of the percursor group. This is the way we have implemented it in the code is that the user may supply one set of delayed spectrum information. This will not affect two group calculations as the spectrum is all produced in group 1. The final equations to be solved are

$$-\nabla\cdot D_g^{n+1}\left(\vec{r}\right)\nabla\phi_g^{n+1}\left(\vec{r}\right)+\left[\Sigma_{tg}^{n+1}\left(\vec{r}\right)+\frac{1}{v_g\left(\vec{r}\right)dt^n}\right]\phi_g^{n+1}\left(\vec{r}\right)-\sum_h\Sigma_{s,h\rightarrow g}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)$$
$$-\left\{\left[1-\beta\left(\vec{r}\right)\right]\frac{\chi_g^p\left(\vec{r}\right)}{k_{crit}}+\frac{\chi_g^d}{k_{crit}}\sum_i\frac{\lambda_i\beta_i\left(\vec{r}\right)dt^n}{\left(1+\lambda_i dt^n\right)}\right\}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right)$$
$$=\frac{1}{v_g\left(\vec{r}\right)dt^n}\phi_g^n\left(\vec{r}\right)+\sum_i\frac{\chi_{ig}^d\lambda_i}{1+\lambda_i dt^n}C_i^n\left(\vec{r}\right)$$

$$C_i^{n+1}\left(\vec{r}\right)=\frac{1}{1+\lambda_i dt^n}C_i^n\left(\vec{r}\right)+\frac{\beta_i\left(\vec{r}\right)dt^n}{k_{crit}\left(1+\lambda_i dt^n\right)}\sum_h\nu\Sigma_{sh}^{n+1}\left(\vec{r}\right)\phi_h^{n+1}\left(\vec{r}\right).$$

The first equation looks very similar to the steady state fixed source form of the neutron diffusion equation. Here, we have an extra term on the total cross section and nu-fission cross section. The algorithm to perform the transient analysis is shown below.

Algorithm 1 Transient Analysis

1: Solve steady state eigenvalue
2: Compute steady state precursor concentrations
3: Begin time-step loop
4: **for** $i = 1, 2, 3, ...n_timesteps$ **do**
5: Compute modified cross sections
6: Compute kinetics modifications to removal and fission
7: Build matrix
8: Build right hand side source
9: Solve for end of time step fluxes
10: Compute end of time step precursors
11: Compute core power
12: **end for**

To solve for the end of time step fluxes we implement PETSc GMRES solver with ILU level 5 preconditioning.

Part A

What is the static rod worth (in fraction of beta)?

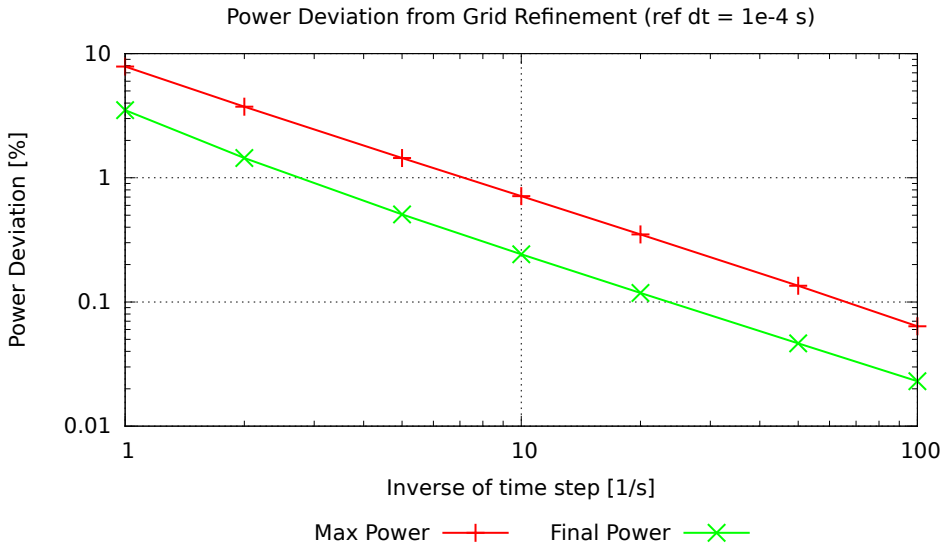
$$\Delta\rho = \frac{k_{unrod} - k_{rod}}{k_{unrod}} = \frac{1.367999 - 1.364613}{1.367999} = 0.37231\beta.$$

What is the dominance ratio of the static rod-inserted base case?

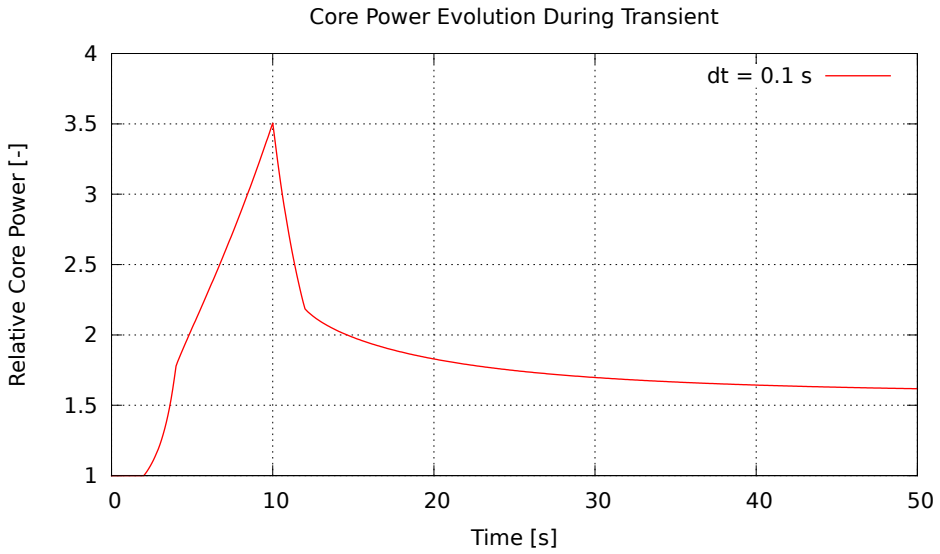
$$DR = 0.99127$$

What uniform time step size is required to converge the peak core power and final core power to 1%?

To answer the question we ran an array of time step sizes. A plot is shown below that shows this convergence. The plot shows the percent error from a reference time step size of 1e-4 seconds. From the plot, convergence is met when the time step is 0.1s. We also see that the max power takes a smaller time step to converge than the final power for the given convergence criterion. We also see a pretty linear convergence rate on a log-log plot. This would indicate a first order time discretization scheme.

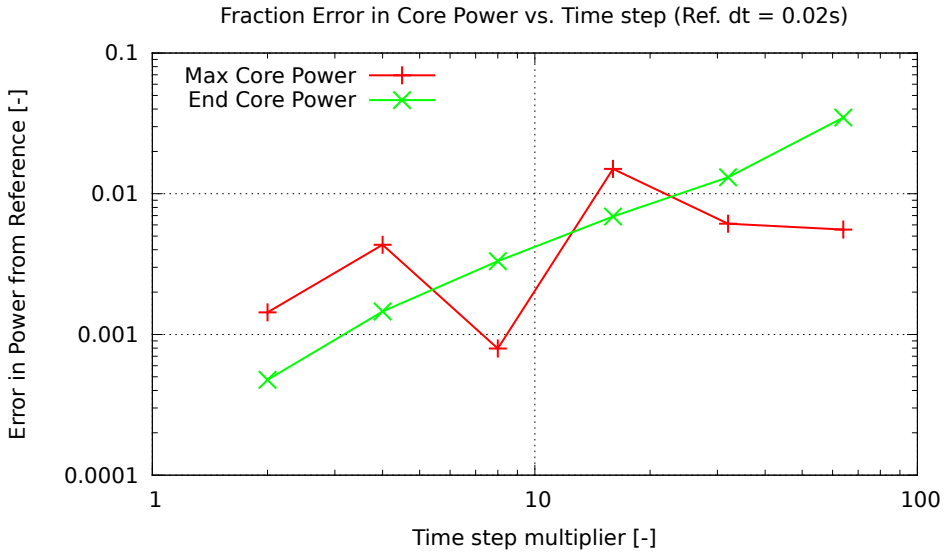


Plot normalized core power vs. time for the converged time-step.



Plot fractional error in peak and final core powers vs. time step when the converged time step is multiplied by 1, 2, 4, 8, 16, 32, and 64.

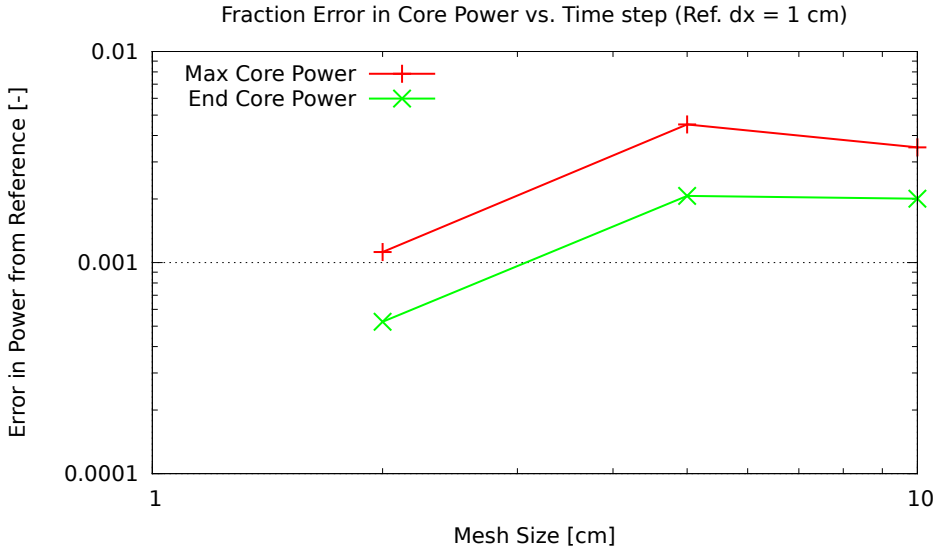
Since our converged time step was 0.1 seconds, multiplying it by 64 would not yield enough time steps to run the problem. Therefore, we just choose a time step of 0.02 seconds. The resulting plot is shown below.



The max power line is very choppy which indicates are time step is not small enough to reach asymptotic convergence rate. The final core power line shows a smoother convergence rate which is about first order.

Plot fractional error in peak and final core powers vs. spatial mesh (using converged time step) for mesh of 1.0, 2.0, 5.0, and 10.0 cm.

We still use a time step of 0.02s for this convergence study. The resulting plot is shown below.



We can't make any conclusions about the convergence rate in space the indicated mesh sizes are too coarse still.

Part B

What is the static rod worth (in fraction of beta)?

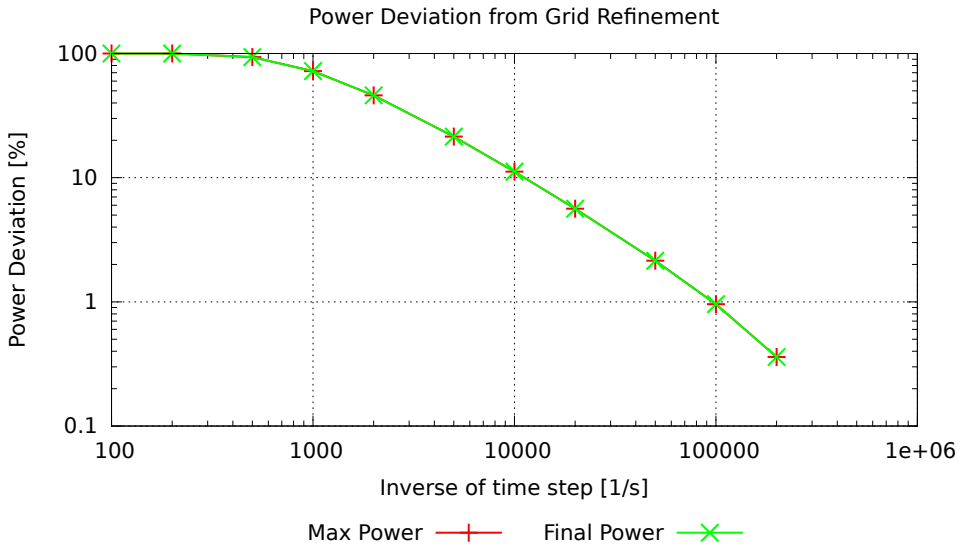
$$\Delta\rho = \frac{k_{unrod} - k_{rod}}{k_{unrod}} = \frac{1.367999 - 1.356805}{1.367999} = 1.23088\beta.$$

Therefore, the core will go prompt super critical when we run this case.
What is the dominance ratio of the static rod-inserted base case?

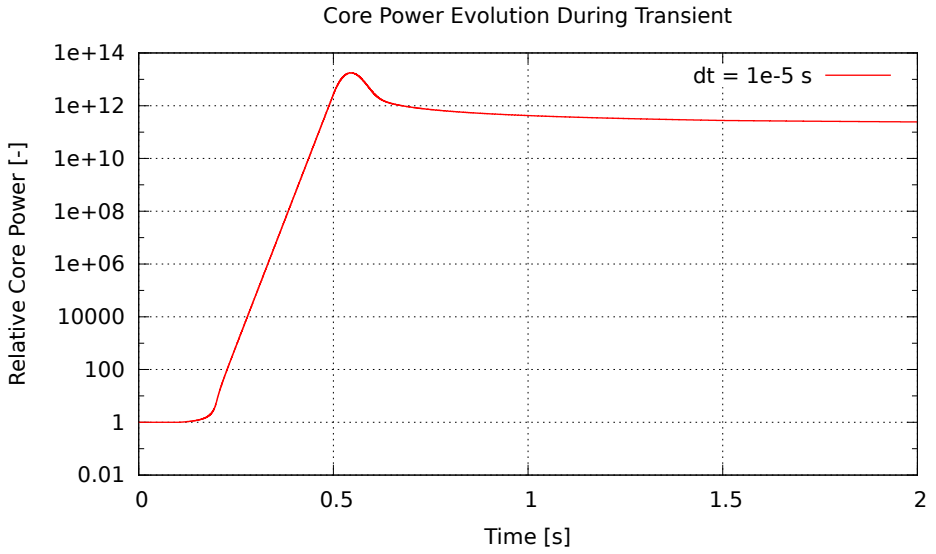
$$DR = 0.9992$$

We see that the dominance ratio is higher than the previous case. Because the rod is worth more, it decouples the core even more. Thus, the dominance ratio increases.
What uniform time step size is required to converge the peak core power and final core power to 1%?

To answer the question we ran an array of time step sizes. A plot is shown below that shows this convergence. The plot shows the percent error from successive time step sizes. From the plot, convergence is met when the time step is 1e-5 seconds. We also see that the max power takes a similar time step to converge as the final power. We also see a pretty linear convergence rate on a log-log plot when it reaches asymptotic convergence. This would indicate a first order time discretization scheme.



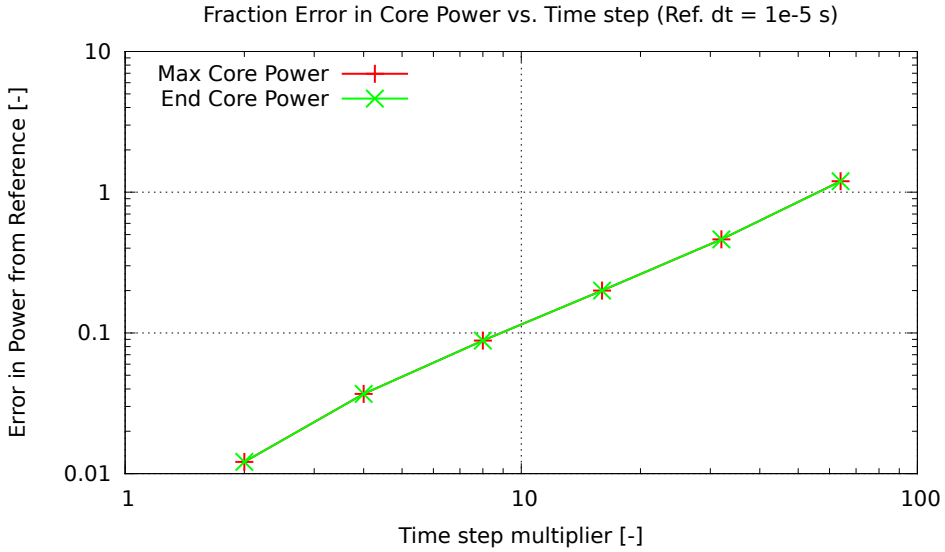
Plot normalized core power vs. time for the converged time-step.



We see that the core power increases enormously because it is super critical off of prompt neutrons only.

Plot fractional error in peak and final core powers vs. time step when the converged time step is multiplied by 1, 2, 4, 8, 16, 32, and 64.

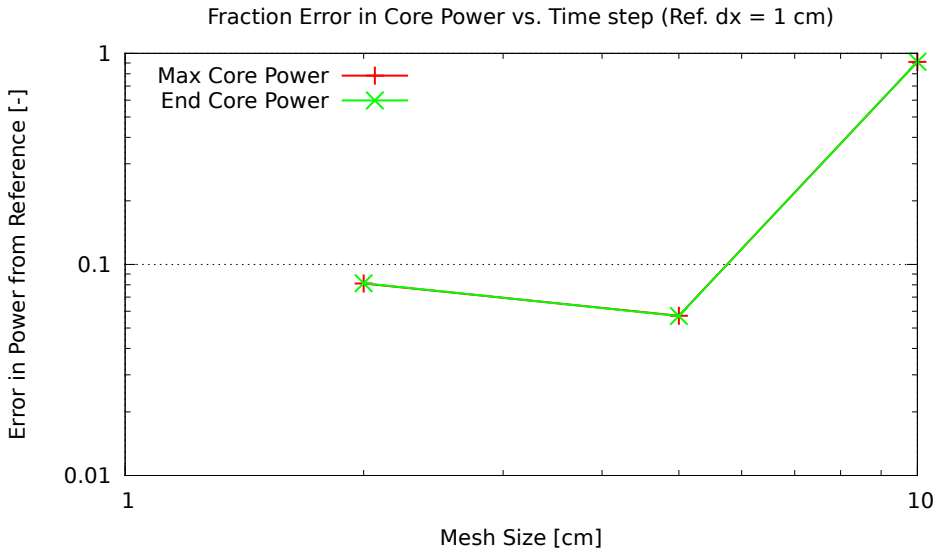
Since our converged time step was 0.2 seconds, multiplying it by 64 would not yield enough time steps to run the problem. Therefore, we just choose a time step of 0.02 seconds. The resulting plot is shown below.



Compared with Part A, this plot is very smooth and shows about a first order convergence rate.

Plot fractional error in peak and final core powers vs. spatial mesh (using converged time step) for mesh of 1.0, 2.0, 5.0, and 10.0 cm.

We still use a time step of 0.02s for this convergence study. The resulting plot is shown below.



We can't make any conclusions about the convergence rate in space the indicated mesh sizes are too coarse still.