

EXAM 1 - NUCE 2101

```
clear all
```

Dane Sabo

October 5, 2025

Problem 1

Part A

Usually, the in-hour equation looks like:

$$\rho = \omega\Lambda + \sum_{i=1}^6 \frac{\omega\beta_i}{\omega + \lambda_i}$$

but for us with 2 groups and $\Lambda = 50\text{microseconds}$, we have:

```
syms omega rho
sympref('FloatingPointOutput',1)
```

```
ans = logical
     1
```

```
Lambda = 5e-5; %s
beta = [0.00168272, 0.00476728];
lambda = [0.0254904, 0.268012]; %Hz
```

```
in_hour = rho == omega*Lambda + omega*beta(1)/(omega + lambda(1))...
           + omega*beta(2)/(omega + lambda(2))
```

```
in_hour =
```

$$\rho = 5.0000\text{e-}05\omega + \frac{0.0048\omega}{\omega + 0.2680} + \frac{0.0017\omega}{\omega + 0.0255}$$

Now we can solve for the largest roots for different values of ρ :

```
rho_beta = [-10, -0.25, -0.1, 0, 0.1, 0.25]*(sum(beta))
```

```
rho_beta = 1x6
    -0.0645    -0.0016    -0.0006         0     0.0006     0.0016
```

```

for i = rho_beta
    in_hour_eq = subs(in_hour, rho, i); % Substitute rho with current value of i
    soln = solve(in_hour_eq, omega);
    largest_root = max(double(soln)); % Convert symbolic solution to double and find the largest root
    fprintf('Largest root for rho = %.4f is: %.5f\n', i/sum(beta), largest_root);
end

```

```

Largest root for rho = -10.0000 is: -0.02484
Largest root for rho = -0.2500 is: -0.01156
Largest root for rho = -0.1000 is: -0.00613
Largest root for rho = 0.0000 is: 0.00000
Largest root for rho = 0.1000 is: 0.00998
Largest root for rho = 0.2500 is: 0.03847

```

Compared to the class solution for the six group in-hour equation on page 15 of Fundamental Kinetics Ideas V17 (FKI), we achieve an interesting result.

1. For $\rho = -10$, two groups significantly underpredicts the largest root (roughly twice the magnitude!).
2. For $\rho = -0.25$, two groups underpredicts the largest root.
3. For $\rho = -0.1$, two groups underpredicts the largest root.
4. For $\rho = 0$, two groups matches the six-group solution exactly (both predict a marginally stable system with $\omega = 0$).
5. For $\rho = 0.1$, two groups is close, but still underpredicts the largest root.
6. For $\rho = 0.25$, two groups underpredicts the largest root.

It appears that our solution always underpredicts the six-group equation, and suffers more error further away from $\rho = 0$.

Part B

$$A = \begin{bmatrix} \frac{\rho - (\beta_1 + \beta_2)}{\Lambda} & \lambda_1 & \lambda_2 \\ \beta_1/\Lambda & -\lambda_1 & 0 \\ \beta_2/\Lambda & 0 & -\lambda_2 \end{bmatrix}$$

$$X_0 = \frac{n_0}{\Lambda} \begin{bmatrix} \Lambda \\ \beta_1/\lambda_1 \\ \beta_2/\lambda_2 \end{bmatrix}$$

$$S = \begin{bmatrix} S_0 \\ 0 \\ 0 \end{bmatrix}$$

$$\text{with } X = \begin{bmatrix} n(t) \\ C_1(t) \\ C_2(t) \end{bmatrix}$$

These results are easily obtained by modifying the matrices on page 17 of the Fundamental Kinetics Ideas V17 handout. For our case, we can trim the matrices down assuming that $\beta_3, \dots, \beta_6$ and $\lambda_3, \dots, \lambda_6$ are all zero, and then remove the padded rows and columns of zeros to obtain our result.

Part C

Our system in part B can be used to create the system of differential equations:

$$\frac{dX}{dt} = AX + S$$

This is a simple first order system, and as such has standard solution:

$$X(t) = e^{At}X_0 + e^{At} \int_0^t e^{-A\tau} S(\tau) d\tau$$

as found on page 17 of FKI.

Part D

```
A = [(0.25*sum(beta) - sum(beta))/Lambda, lambda(1), lambda(2);  
      beta(1)/Lambda, -lambda(1), 0;  
      beta(2)/Lambda, 0, -lambda(2)]
```

```
A = 3x3  
-96.7500    0.0255    0.2680  
 33.6544   -0.0255         0  
 95.3456         0   -0.2680
```

```
%The modal matrix is P, while the eigenvalues are the diagonal entries  
%of D.  
[P, D] = eig(A)
```

```
P = 3x3  
-0.6915    0.0016    0.0009  
 0.2399    0.8608   -0.9104  
 0.6814    0.5090    0.4138
```

```
D = 3x3  
-97.0230         0         0  
         0    0.0385         0  
         0         0   -0.0590
```

```
% Calculate the matrix exponential of A multiplied by tau  
%Here is e^A*tau  
tau = 1e-4; %s  
part_d_matrix = P*expm(D*tau)*inv(P)
```

```
part_d_matrix = 3x3  
 0.9904    0.0000    0.0000  
 0.0033    1.0000    0.0000  
 0.0095    0.0000    1.0000
```

```
%expm is used because technically D is a matrix here.
%exp(trace(D) * tau) is equivalent
proof = P*exp(trace(D) * tau)*inv(P)
```

```
proof = 3x3
    0.9903   -0.0000   -0.0000
         0    0.9903    0.0000
    0.0000         0    0.9903
```

Part E

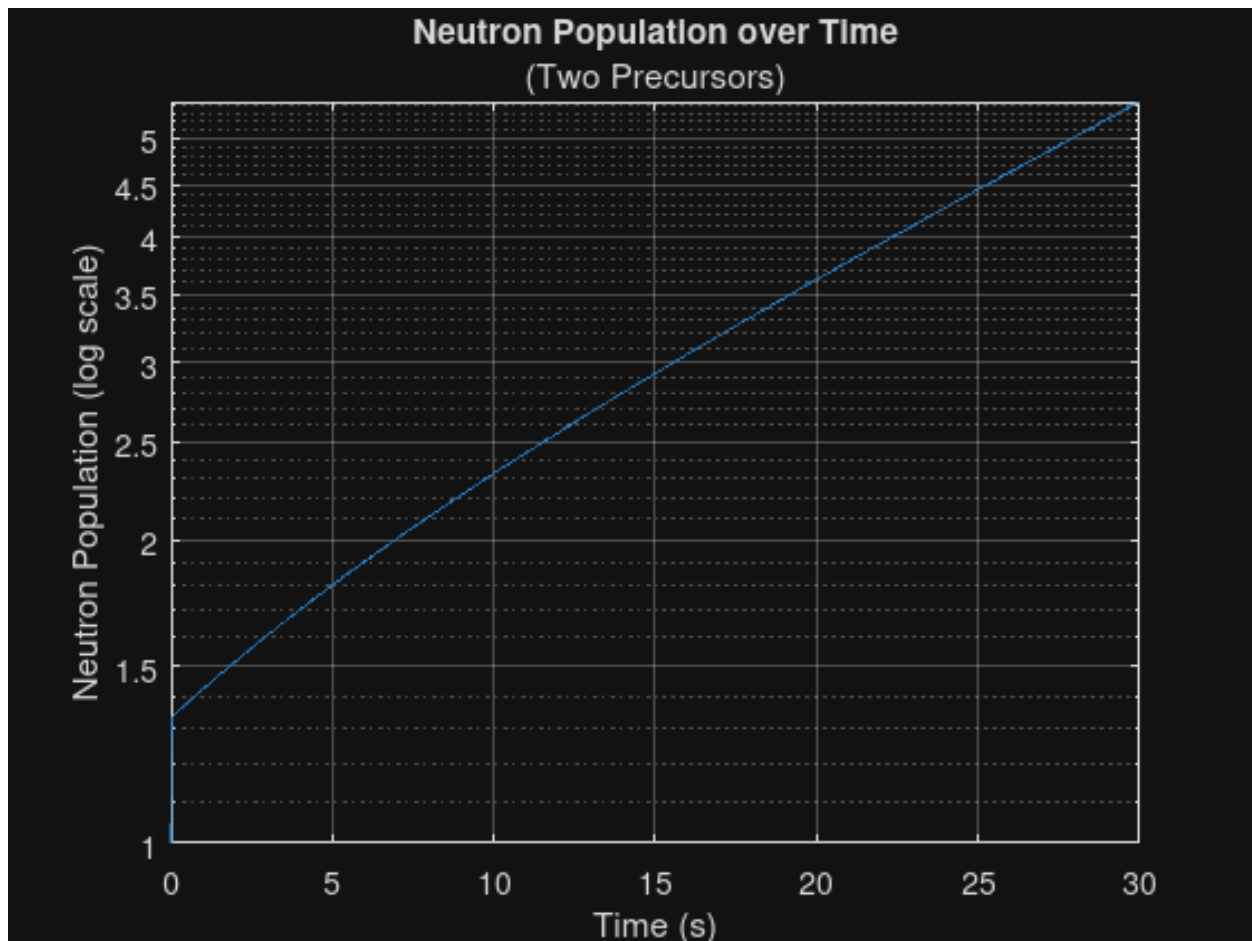
```
%Initial conditions
n_0 = 1; %this is an assumption
X_0 = n_0/Lambda*[Lambda; beta(1)/lambda(1); beta(2)/lambda(2)];
S = [0;0;0];

% Define the time span for the ODE solution
tspan = [0, 30]; % seconds

% Define the ODE function
part_e = @(t, X) A * X + S;

% Solve the ODE using ode45
[t_e, X] = ode45(part_e, tspan, X_0);

% Plot the first state of X (neutron population) over time
figure;
semilogy(t_e, X(:, 1)/n_0);
xlabel('Time (s)');
ylabel('Neutron Population (log scale)');
title('Neutron Population over Time');
subtitle('(Two Precursors)');
grid on;
```



Part F

```
lambda_eff = 0.1 %Hz
```

```
lambda_eff = 0.1000
```

```
rho = 0.25*sum(beta)
```

```
rho = 0.0016
```

The prompt jump makes an assumption about the value of the precursor, and assumes that prompt neutrons immediately equalize. Using the formula given in the exam:

```
prompt_jump = n_0 * sum(beta)/(sum(beta)-rho)
```

```
prompt_jump = 1.3333
```

And then we use this value as our initial condition for the neutron population.

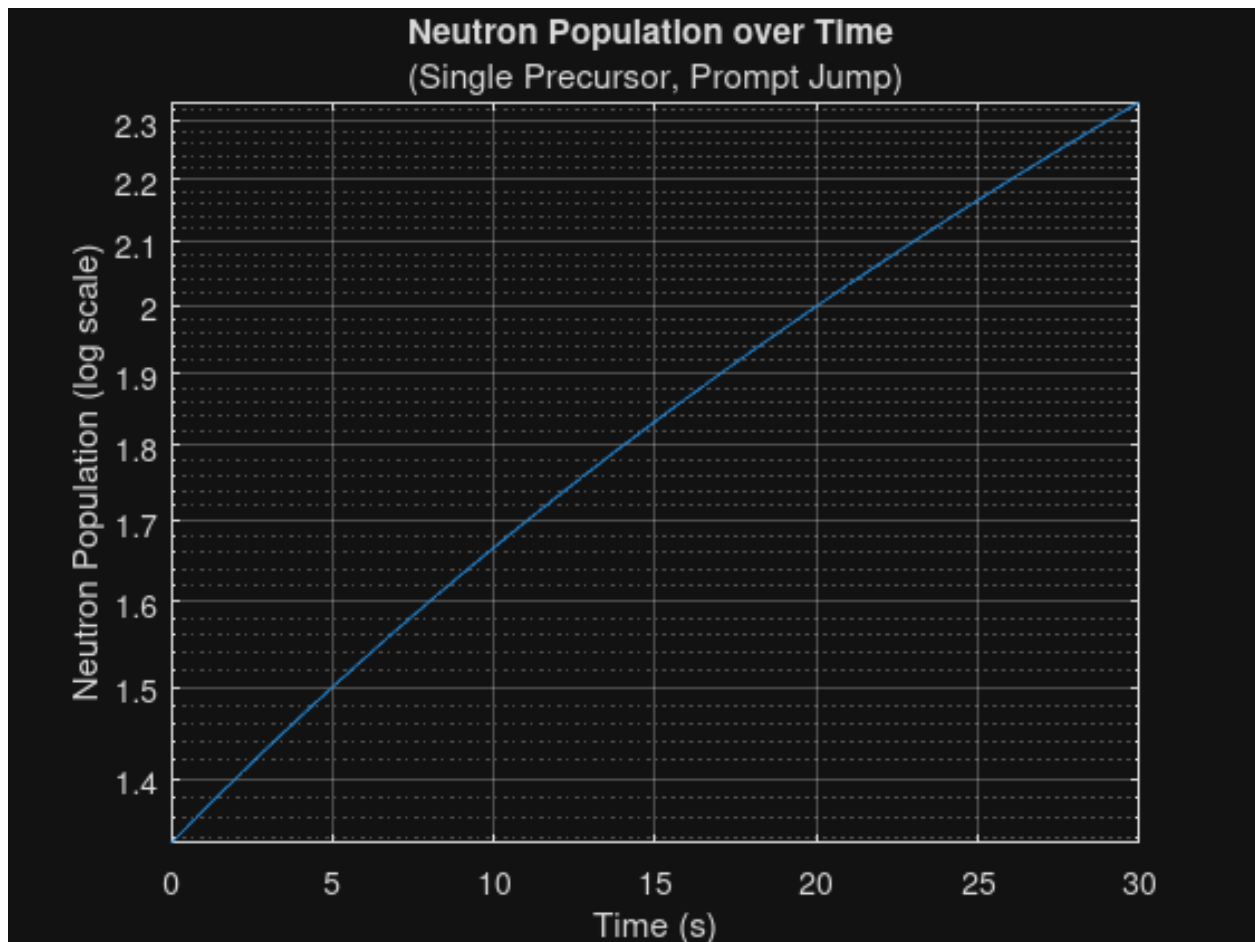
```

% Define the ODE function
part_f = @(t, X) lambda_eff*rho/(sum(beta) - rho);

% Solve the ODE using ode45
[t_f, n] = ode45(part_f, tspan, prompt_jump);

% Plot the first state of X (neutron population) over time
figure;
semilogy(t_f, n/n_0);
xlabel('Time (s)');
ylabel('Neutron Population (log scale)');
title('Neutron Population over Time ');
subtitle('(Single Precursor, Prompt Jump)');
grid on;

```



Part G

I've rewritten SimplePower.m using ode45 for clarity. This is the six group equivalent solution, compared to the other two solutions I made previously.

```

% Physical parameters for 6-group model
beta_6 = [0.00021, 0.00141, 0.00127, 0.00255, 0.00074, 0.00027];
% delayed neutron fractions
lambda_6 = [0.01246403, 0.03052863, 0.11141479, 0.30130435, 1.13606557, 3.01304348];
% decay constants [1/sec]

% Build system matrix A_6
A_6 = zeros(7, 7);
A_6(1, 1) = (rho - sum(beta_6)) / Lambda;
A_6(1, 2:7) = lambda_6;
for i = 1:6
    A_6(i+1, 1) = beta_6(i) / Lambda;
    A_6(i+1, i+1) = -lambda_6(i);
end

% Initial conditions (steady state)
X_0_6 = n_0 / Lambda * [Lambda; beta_6(1)/lambda_6(1);
beta_6(2)/lambda_6(2);
beta_6(3)/lambda_6(3);
beta_6(4)/lambda_6(4);
beta_6(5)/lambda_6(5);
beta_6(6)/lambda_6(6)];

% Source term (zero for this problem)
S_6 = zeros(7, 1);

% Define the ODE function
part_g = @(t, X) A_6 * X + S_6;

% Solve the ODE using ode45
[t_6, X_6] = ode45(part_g, tspan, X_0_6);
% Combined plot of all three models
figure;
hold on;

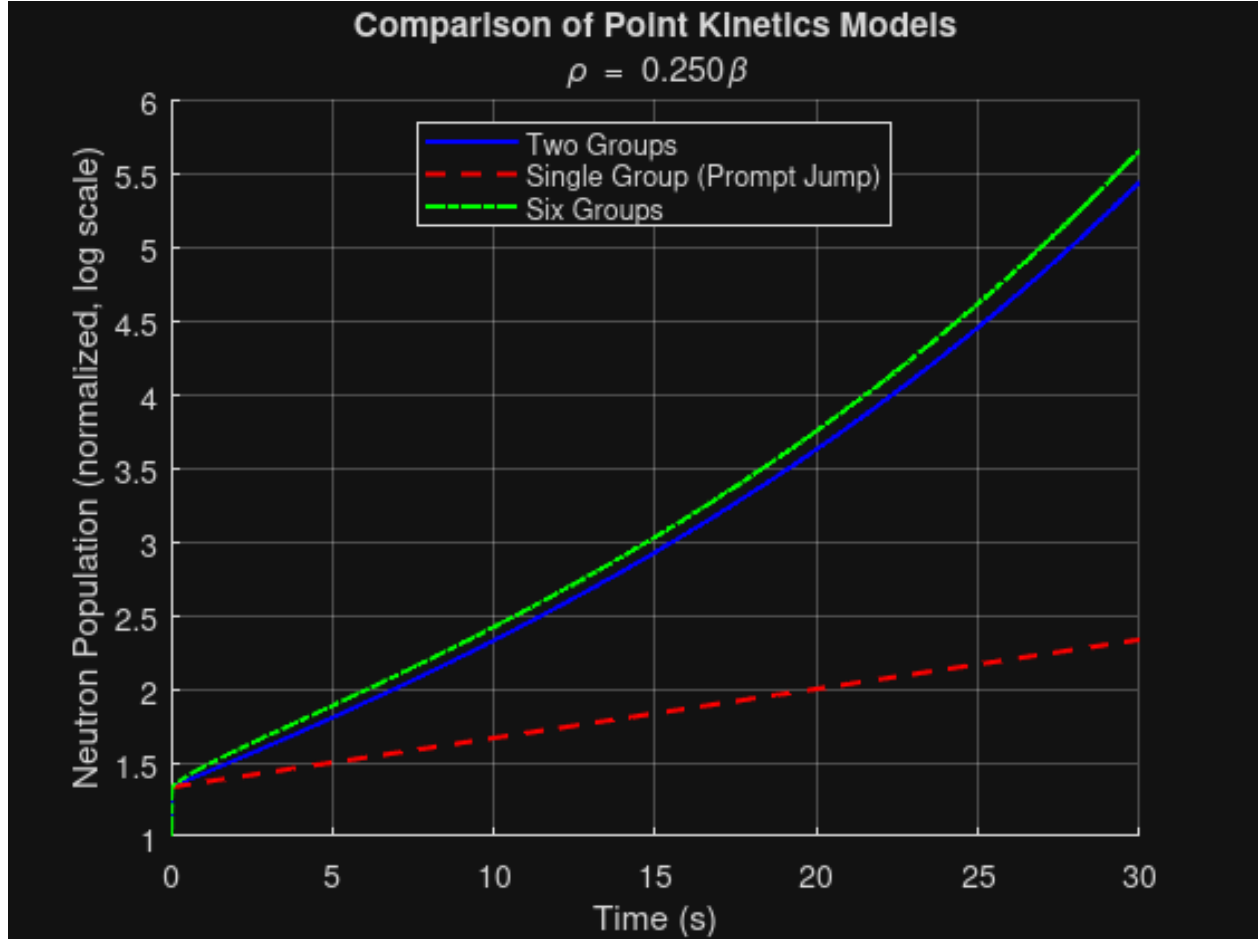
% Two-group model (from Part E)
semilogy(t_e, X(:, 1)/n_0, 'b-', 'LineWidth', 1.5, 'DisplayName', 'Two Groups');

% Single group prompt jump (from Part F)
semilogy(t_f, n/n_0, 'r--', 'LineWidth', 1.5, 'DisplayName', ...
'Single Group (Prompt Jump)');

% Six-group model (from Part G)
semilogy(t_6, X_6(:, 1)/n_0, 'g-.', 'LineWidth', 1.5, 'DisplayName', 'Six Groups');

hold off;
xlabel('Time (s)');
ylabel('Neutron Population (normalized, log scale)');
title('Comparison of Point Kinetics Models');
subtitle(sprintf('\rho = %.3f\beta', rho/sum(beta)));
legend('Location', 'best');
grid on;

```



Part H

We can create a two-group prompt jump assumption model by adapting the six group prompt jump model on page 34 of FKI:

$$N(t) = \frac{\Lambda \left(\sum_{i=1}^2 \lambda_i C_i + S \right)}{\beta - \rho}$$

where each precursor is a first order differential equation:

$$\frac{dC_i(t)}{dt} = \frac{N(t)\beta_i}{\Lambda} - \lambda_i C_i(t)$$

We have two precursors, so we then can build a matrix form of the two equations together:

$$\begin{bmatrix} \dot{C}_1 \\ \dot{C}_2 \end{bmatrix} = \begin{bmatrix} \beta_1 \lambda_1 - \lambda_1(\beta - \rho) & \beta_1 \lambda_2 \\ \beta_2 \lambda_1 & \beta_2 \lambda_2 - \lambda_2(\beta - \rho) \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} + \left(\frac{S}{\beta - \rho} \right) \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$$

where the initial condition for each precursor is:

with initial condition

$$\vec{C}_0 = \frac{N_{S/D}}{\Lambda} \begin{bmatrix} \beta_1/\lambda_1 \\ \beta_2/\lambda_2 \end{bmatrix}$$

where $N_{S/D}$ is derived from the shutdown equilibrium.

Problem 2

clear all

Part A

Let's assume for this problem, the 'sudden insertion' of reactivity is an instant addition. In that case, the prompt neutrons will quickly increase the total neutron population in the first few milliseconds. After they very quickly stabilize, the prompt neutrons will play a back-seat role to the delayed neutrons. A fraction of these neutrons and fission reactions are becoming precursors, which will release delayed neutrons on a much slower timescale. These delayed neutrons will continue to increase power, as the reactor is delayed critical with this reactivity addition.

Part B

The equation becomes

$$\frac{N(0_+)}{N(0_-)} = \frac{\beta - \rho(0_-)}{\beta - \rho(0_+)}$$

Usually we just assume that $\rho(0_-) = 0$, but with sources involved to maintain prompt neutron steady state, $\rho(0_-)$ is actually negative. This is supported by page 32 of FKI.

Part C

For the initial jump power, we can use the prompt jump assumption from part B:

```
P_0_left = 1.0e-8; %assume neutron count and power count are proportional
P_0_right = P_0_left *(1-(-10))/(1-(-9)) %beta factors out and cancels
```

```
P_0_right = 1.1000e-08
```

This is the initial jump power.

To calculate the steady state value, we recognize that ρ is still strongly subcritical. For that reason, the steady state value is eventually going to return to a shutdown equilibrium power. We can use the shutdown equilibrium equation on page 35 of FKI to find the steady state power:

$$P_{S/D} \propto -\frac{S\Lambda}{\rho}$$

We assume that the source and prompt neutron period are constant, and thus can be solved for.

$$S\Lambda \propto -1.0 \times 10^{-8} \times -10\beta$$

and then plug this value back into the equation but with using $\rho = -9\beta$:

$$P_{S/D}(0_+) = \frac{-(1.0 \times 10^{-8} \times 10\beta)}{-9\beta}$$

$$P_{S/D}(0_+) = 1.111 \times 10^{-8}$$

This is the final steady state power. Very slightly more than the prompt jump ($\sim 1\%$).

Part D

The length of time for the transient to settle is dependent on the precursor pole locations. In this problem, we are not given values for λ , but I can make an educated guess. Using the table on page 15 of FKI, we can extract the slowest period for the in-hour equation where $\rho/\beta = -10$. This is $\omega_0 = -0.012423$. This corresponds to a half-life $t_{1/2} = \frac{0.693}{|-0.012423|} = 55.7836[\text{s}]$.

For five half lives, this value is scaled by 5:

$$t_{ss} = 5t_{1/2} = 278.9181[\text{s}].$$

Problem 3

clear all

Part A

Subcritical multiplication is the process where neutron sources play a significant role in the fission chain. This is usually for low power or shutdown reactors, as the reactivity is so low the steady state power of the reactor is a function of the source activities.

Subcritical multiplication is also present when a reactor core that has been operating is shut down. Certain fission decay chains have products that create additional neutrons in different timescales. First, hydrogen interacting with gamma rays produces neutrons for about the first day, then heavier elements dominate source production thereafter.

Part B

Assuming the precursors are in steady-state, 10% of the neutrons in the fission chain must come from sources to maintain steady state.

Problem 4

clear all

Part A

$$SUR = 26.06[\text{dpm-sec}] \frac{\dot{\rho} + \lambda_{eff}\rho}{(\beta - \rho)}$$

First, I'm going to find the values for SUR at certain points relating to the rod movement:

```
SUR = @(rho_dot, lambda_eff, rho, beta) 26.06*(rho_dot + lambda_eff*rho)/(beta-rho);
```

```
%before rod pull
```

```
rho_dot = 0;
```

```
lambda_eff = 0.1; %Hz
```

```
rho = 0;
```

```
beta = 0.00645; %assumed
```

```
SUR(rho_dot, lambda_eff, rho, beta)
```

```
ans = 0
```

```
%now rod pull starts
```

```
rho_dot = 0.00160/15;
```

```
SUR(rho_dot, lambda_eff, rho, beta)
```

```
ans = 0.4310
```

```
%end of rod pull
```

```
rho = 0.00160;
```

```
SUR(rho_dot, lambda_eff, rho, beta)
```

```
ans = 1.4329
```

```
%rods not moving, just chillin'
```

```
rho_dot = 0;
```

```
SUR(rho_dot, lambda_eff, rho, beta)
```

```
ans = 0.8597
```

```
%start putting rods back in
```

```
rho_dot = -0.00160/15;
```

```
SUR(rho_dot, lambda_eff, rho, beta)
```

```
ans = 0.2866
```

```
%end of rod push
```

```
rho = 0;
```

```
SUR(rho_dot, lambda_eff, rho, beta)
```

```
ans = -0.4310
```

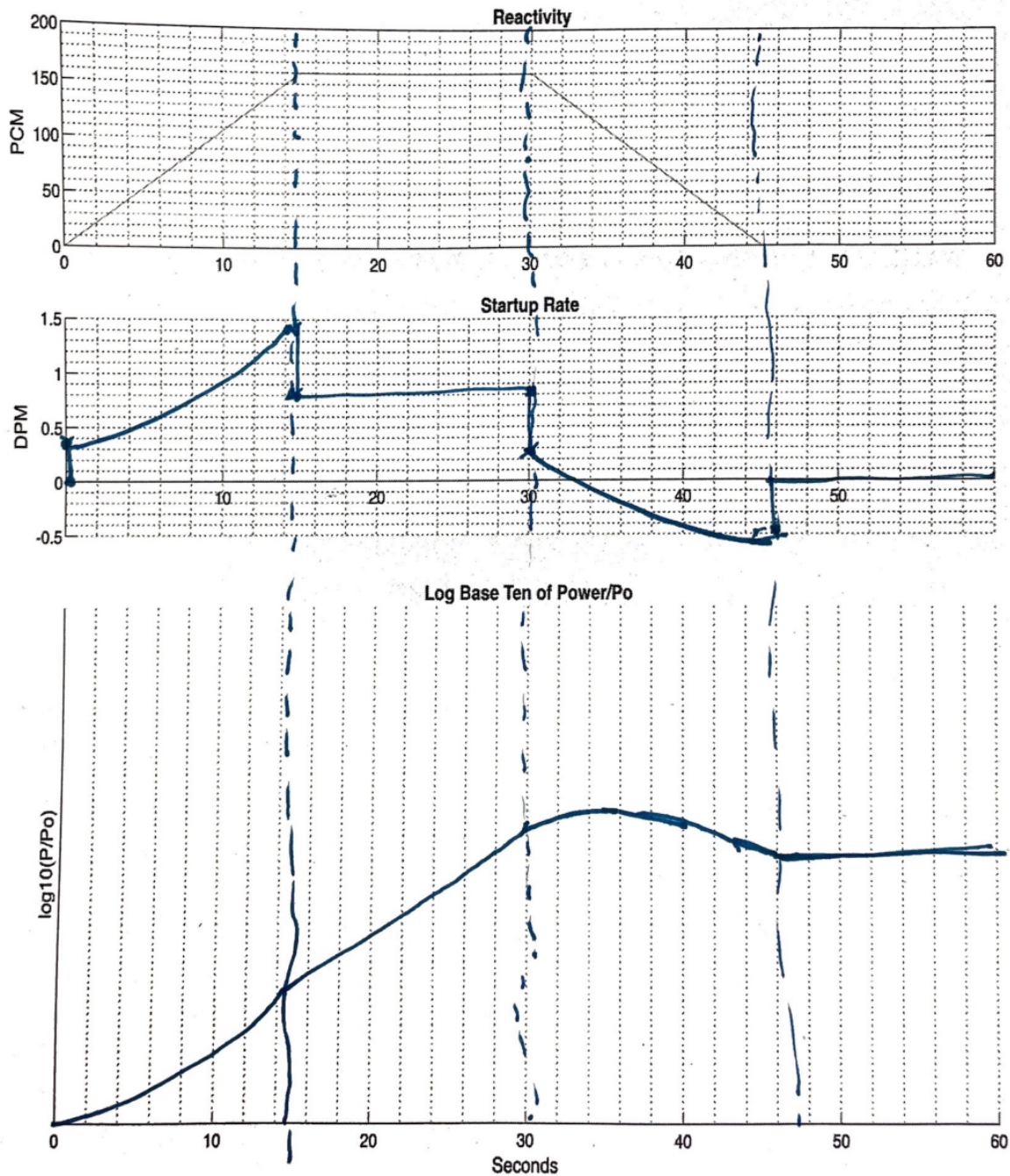
```
%rods back in to zero
```

```
rho_dot = 0;
```

```
SUR(rho_dot, lambda_eff, rho, beta)
```

```
ans = 0
```

Problem 4



Part B

I would expect power to reach its maximum value at about 32 ± 1 seconds from the beginning of the operation. That is when SUR becomes negative.

Part C

We need to find $\rho_{\dot{}}$ that makes SUR 0 at the beginning of rod push.

$$SUR = 26.06[\text{dpm-sec}] \frac{\dot{\rho} + \lambda_{eff}\rho}{(\beta - \rho)}$$

$$0 = 26.06[\text{dpm-sec}] \frac{\dot{\rho} + \lambda_{eff}\rho}{(\beta - \rho)}$$

$$0 = \dot{\rho} + \lambda_{eff}\rho$$

$$\dot{\rho} = -\lambda_{eff}\rho$$

```
rho = 0.0016;
rho_dot = -lambda_eff * rho
```

```
rho_dot = -1.6000e-04
```

We would need a rate of 1.6 pcm/s reactivity removal.

Check:

```
SUR(rho_dot, lambda_eff, rho, beta)
```

```
ans = 0
```

Part D

This use of the start up rate equation has significant numerical errors that limit its usefulness in reactor operations, limiting the basic SUR equation to only be used for getting a feel for transient behavior, not for exact operations. The big error is the assumption that λ_{eff} isn't changing with time. This can be corrected by using the additional term $\frac{\dot{\lambda}_{eff}}{\lambda}$ as shown on page 51 of FKI, but doing this requires solving for all six precursors continuously. Using modern computational methods, this is not difficult, and achievable with real-time simulation.

Problem 5

λ_{eff} changes significantly with the rate of power change because the dominant precursor effect is dependent on that rate of power change and the duration of the change in power. When power is rising, the simplification of λ_{eff} must either neglect the behavior of long time constant precursors or short time constant precursors. If the rate of power change is fast, λ_{eff} also needs to be fast to model the dominating effect of the fast responding precursors. If the rate of power change is slow, λ_{eff} must be slow to accurately capture the dominating behavior of the slow precursors. Generally, λ_{eff} must keep up with the fast precursors on power increase. On power decrease, the slower precursors can drag power into the mix for longer, requiring a slower λ_{eff} .

This is further complicated by the duration of the power change. If the duration of a power change is long enough, a fast λ_{eff} is not going to correctly model the contribution of large time constant precursors (and vice versa for the short duration power change case). This is true for both increases and decreases in power.

Basically, there are too many exponential decays and growths to be modelled accurately in all cases by one λ_{eff} .

Problem 6

The adjoint flux ϕ^* represents the importance of a neutron at a given location and energy for contributing to the reactor's power generation or other quantities of interest (such as reactivity or detector response). It is highest in locations like the core center, where a neutron at the right spot is most likely to cause fission events that contribute to the global reactor behavior, and lowest near the edges of the reactor, where neutron escape becomes more likely. It's a way to account for the fact that a constant amount of flux everywhere does not create a constant power everywhere.

This is super useful if there's mixed fuel rods in the reactor. The adjoint flux is lower around more spent fuel rods compared to fresh rods.

Problem 7

The equation developed in class is:

$$\begin{aligned} 1. & \frac{d}{dt} \left[\frac{\phi(\vec{r}, E, t)}{v} \right] = \\ 2. & \nabla \cdot D(\vec{r}, E) \nabla \phi(\vec{r}, E, t) \\ 3. & -\Sigma_t(\vec{r}, E, t) \phi(\vec{r}, E, t) \\ 4. & + \int_0^\infty \Sigma_s(\vec{r}, E' \rightarrow E, t) \phi(\vec{r}, E', t) dE' \\ 5. & + q(\vec{r}, E, t) \\ 6. & + \nu(1 - \beta) \chi_p(E) \int_0^\infty \Sigma_f(\vec{r}, E', t) \phi(\vec{r}, E', t) dE' \\ 7. & + \sum_{i=1}^6 \chi_{di}(E) \lambda_i C_i(\vec{r}, t) \end{aligned}$$

Part A

Here are descriptions of the physical significance of each term:

1. This is the time rate of change of the flux ϕ divided by the average neutrons speed v . This is analogous to the time rate of change of the total neutron population $\dot{n}(\vec{r}, E, t)$.
2. This is the neutron leakage term. This is a diffusion process that represents the physical movement of neutrons through space. Generally, it says neutrons are going to flow in the direction of the lowest concentration of neutrons.
3. This is the neutron removal rate term. This includes absorption and neutrons used in fission events.
4. This is the neutron in-scattering term. This describes the rate at which neutrons scatter to energy E from all other energies at position $\vec{r}$ at time t .
5. This is the external neutron source term.
6. This is the production rate of prompt neutrons integrated over all energies E , multiplied by a distribution of χ_p , which is the distribution of neutron energies after a fission event. The fraction $\nu(1 - \beta)$ selects only prompt neutrons. These are fuel fission neutrons.
7. This is the production rate from the delayed precursors, with six precursor groups described here, and multiplied by χ_{di} to represent the distribution of neutron energy from each precursor.

Part B

First we have the equation:

$$\phi(\vec{r}, E, t) = F(\vec{r})G(t)$$

with the condition that:

$$\nabla^2 F(\vec{r}) = -B^2 F(\vec{r})$$

We can expand this second equation by using the definition of Laplace operator:

$$\frac{d^2 F}{dx^2} + \frac{d^2 F}{dy^2} + \frac{d^2 F}{dz^2} + B^2 F = 0$$

This is the equation we need to solve for F in order to find the spatial part of the flux ϕ . To do so, we need to separate variables. Recognize that $F(\vec{r})$ can be rewritten as $X(x)Y(y)Z(z)$, where each term makes up the behavior for that one component of $\vec{r}$. Doing this, we can rewrite our diffusion equation:

$$YZ \frac{d^2 X}{dx^2} + XZ \frac{d^2 Y}{dy^2} + XY \frac{d^2 Z}{dz^2} + B^2 XYZ = 0$$

$$\frac{1}{X} \frac{d^2 X}{dx^2} + \frac{1}{Y} \frac{d^2 Y}{dy^2} + \frac{1}{Z} \frac{d^2 Z}{dz^2} + B^2 = 0$$

where $\boxed{B^2 = B_x^2 + B_y^2 + B_z^2}$ (geometric buckling).

which we can then break into components:

$$\frac{1}{X} \frac{d^2 X}{dx^2} + B_x^2 = 0, \quad \frac{1}{Y} \frac{d^2 Y}{dy^2} + B_y^2 = 0, \quad \frac{1}{Z} \frac{d^2 Z}{dz^2} + B_z^2 = 0,$$

which then become

$$\frac{d^2 X}{dx^2} + B_x^2 X = 0, \quad \frac{d^2 Y}{dy^2} + B_y^2 Y = 0, \quad \frac{d^2 Z}{dz^2} + B_z^2 Z = 0,$$

Now, we've got a bunch of 2nd order differential equations to solve, which have a standard solution. Because B^2 is always positive, we know the roots of the characteristic equation are always complex, and then that the solutions to these equations are going to have oscillatory solutions and the general form:

$$X = A \cos(B_x x) + C \sin(B_x x)$$

The other spacial coordinates are exactly the same, since we have a symmetric cube.

Our boundary conditions are also specified. We know just outside the boundary, there is zero flux. This means:

$$F(\hat{L}, \hat{L}, \hat{L}) = \vec{0}$$

We can also abuse another fact of symmetry: because our cube is symmetric, and our origin is centered in the cube, the sine part of our solution *must* be zero for our solution to be symmetric. So then, we know that:

$$X = A \cos(B_x x)$$

and with our boundary condition:

$$X\left(\frac{\hat{L}}{2}\right) = A \cos\left(B_x \frac{\hat{L}}{2}\right) = 0$$

A can be any scalar.

$$\cos\left(B_x \frac{\hat{L}}{2}\right) = 0$$

$B_x \frac{\hat{L}}{2} = \frac{n\pi}{2}$, $n = 1, 3, 5, \dots$ We'll just assume $n = 1$, and that we don't care about far outside of the cube.

Then, $\boxed{B_x = \frac{\pi}{\hat{L}}}$, and by symmetry, $\boxed{B_y = \frac{\pi}{\hat{L}}, \quad B_z = \frac{\pi}{\hat{L}}}$.

Thus our solutions for X , Y , and Z become:

$$X(x) = A \cos\left(\frac{\pi x}{\hat{L}}\right), \quad Y(y) = A \cos\left(\frac{\pi y}{\hat{L}}\right), \quad Z(z) = A \cos\left(\frac{\pi z}{\hat{L}}\right)$$

and putting things back together,

$$\boxed{F(x, y, z) = A^3 \cos\left(\frac{\pi x}{\hat{L}}\right) \cos\left(\frac{\pi y}{\hat{L}}\right) \cos\left(\frac{\pi z}{\hat{L}}\right)}$$