Point Reactor Kinetics

For reactors, we are primarily concerned with the movement and reaction rates of neutrons around the reactor. This can get really really complicated.

One simplification we can make to help us understand reactor behaviour is to assume that for time dependent behaviour, the neutron population can be treated as a spatial average, or a point (Lamarsh et al., 2001). When using point reactor kinetics, we are often interested in short term transients. Ie to say, these are transients of about 2 minutes or less, sometimes too fast for human reaction time to handle. For this, we look look at chapter 9 of Bell and Glasstone (Bell & Glasstone, 1970).

Neutron Balance Equation

Let's start with a formula of neutron balance:

$$\text{rate of change of neutron population} = \text{gain} - \text{loss}$$

Let's call the neutron population n, so the rate of change is $\frac{\partial n}{\partial t}$. Now, we assume that neutron population doesn't change with space, angle or energy, so we can write the total derviative:

$$\frac{dn}{dt} = \text{gain} - \text{loss}$$

What then are the gain terms? These are fission sources. Loss terms can be resonance absorptions, leakage or other factors. These factors are neatly summed up in the six factor formula described previously.

$$k_{\text{eff}} = P_{\text{TNL}} P_{\text{FNL}} \eta \varepsilon f p$$

Now, k_{eff} also means how many neutrons in are in the next generation, compared to how many neutrons are in the current generation. Therefore, k_{eff} can also be interpreted as the ratio of gain to loss of neutrons:

$$k_{\text{eff}} = \frac{\text{gain}}{\text{loss}}$$

$$\frac{dn}{d\ \text{generation}} = \text{loss}\ \left(\frac{\text{gain}}{\text{loss}} - 1\right)$$

$$\frac{dn}{d\ \text{generation}} = \text{loss}\ \left(k_{\text{eff}} - 1\right)$$

What then is the loss term? It is simply the current neutron population.

$$\frac{dn}{d\ \text{generation}} = n(k_{\text{eff}} - 1)$$

Neutron Lifetime

Now, k_{eff} does tell us about the neutron population between successive generations of neutrons, but how then must we get a time dependent equation?

$$\frac{dn}{dt}\frac{dt}{d\ \text{generation}} = n(k_{\text{eff}} - 1)$$

$\frac{dt}{d\ \text{generation}}$ is what we can call generation time. Now, neutrons are not all created equal. Some emerge about less than 1 nanosecond after fission. To be precise, a prompt neutron is typically emitted 10^{-14} seconds after fission (Mihalczo, 2004). Virtually instantaneous for all intents and purposes. In

contrast, delayed neutrons can emerge typically microseconds or whole minutes after fission (Bell & Glasstone, 1970). These delayed neutrons emerge from the fission fragments after fission. These are known as precursors.

As you'd guess, the number of different types of isotopes emitting these delayed neutrons are too numerous to count. To sort this out, the standard practice is to band these precursors into groups with similar decay constants. It is customary to have six (Bell & Glasstone, 1970) or eight groups of such precursors.

Now, to make life easier for ourselves, let's only consider prompt neutrons. The prompt neutron lifetime is sometimes denoted l_p (Lamarsh et al., 2001). To simplify things further, let's consider an infinite reactor where we don't have to worry about leakage, it is customary to denote this l_∞.

Prompt Critical Infinite Reactor

For an infinite reactor with no leakage:

$$\frac{dn}{dt}\frac{dt}{d\text{ generation}} = n(k_\infty - 1)$$

$$\frac{dn}{dt} l_\infty = n(k_\infty - 1)$$

$$\frac{dn}{dt} = \frac{k_\infty - 1}{l_\infty} n$$

For k_∞, there is zero probability of leakage:

$$k_\infty = \eta \varepsilon f p$$

All well and good. Now, what is l_∞? It is the average time a neutron survives in the reactor before being absorbed.

The mean lifetime is determined in turn by the mean path before absorption:

$$l_{\text{abs}} = \frac{1}{\Sigma_{\text{abs}}}$$

(derive this expression yea?)

And the time is length divided by neutron speed:

$$l_\infty = \frac{l_{\text{abs}}}{v_n} = \frac{1}{\Sigma_{\text{abs}} v_n}$$

So we then get our expression for prompt neutrons in the infinite reactor.

Prompt Critical PRKE with leakage

However, real reactors have leakage. For this, we can assume that when a neutron is born, it can choose to leak or not leak in an instantaneous manner. After which, it is then subject to absorption when travelling through the reactor. It's kind of simplistic, seeing how neutrons do have to travel a bit before leaking out. So in reality, rather than neutrons leaking instantaneously after being born, they do travel a bit.

Nevertheless, neutrons that do leak never get absorbed anyway. This means we can automatically discount them from the next generation of neutrons. Mathematically, we do not have to account for their travel time in the mean neutron lifetime. Neutrons just either leak, or they don't.

$$l = P_{\text{NL}} l_\infty = P_{\text{NL}} \frac{l_{\text{abs}}}{v_n} = P_{\text{NL}} \frac{1}{\Sigma_{\text{abs}} v_n}$$

Note that here, we don't distinguish between energy groups as this is a common simplification in Point Reactor Kinetics Equations (PRKE). Now, rather than k_∞, we use k_{eff}. Our prompt PRKE reactor equation becomes:

$$\frac{dn}{dt} = \frac{k_{\text{eff}} - 1}{l} n$$

Suppose for a thermal reactor v_n is 2200 m/s. A one group thermal absorption cross section for water is about 0.0197 cm^{-1} (Lamarsh et al., 2001). Of course, we should be using uranium cross sections in a reactor, but let's just pump out some numbers for now.

What is the generation time?

$$l_\infty = \frac{1}{0.0197 \text{cm}^{-1} * 2200 \frac{m}{s}}$$

$$l_\infty = \frac{1}{0.0197 \text{cm}^{-1} \frac{100 \text{ cm}}{1 m} * 2200 \frac{m}{s}}$$

$$l_\infty = \frac{1}{1.97 m^{-1} * 2200 \frac{m}{s}}$$

$$l_\infty = \frac{1}{4334} s$$

$$l_\infty = 0.00023073 s$$

This is really short! About 2.31×10^{-4} s. Try this with actual U-235 or Pu-239 cross sections and densities. In pure metal reactors, the neutrons are not moderated, meaning they travel a lot faster (1 MeV of energy or so!) l_∞ is on the order of 10^{-7} s. Is there enough time for the operator to react? Let's see whether you can replicate the results.

Delayed Critical PRKE Equations

For a reactor like this, it is effectively functioning like a bomb (but not quite) because the neutrons are multiplying so fast, and you cannot control this reactor.

Thankfully, not all neutrons are born promptly, some are delayed. Some are born seconds or minutes after the fission event. Hence, the multiplication time between generations is now in minutes rather than microseconds. This reactor is controllable.

Again, there are so many nuclide's releasing neutrons with different release rates that it is hard to put it all into equation form. To help us deal with this mess, we traditionally use six to eight groups for delayed neutrons, but in most nuclear engineering courses, we like to use one group because it is less mathematically complex.

Let's modify our prompt neutron equations where we account for delayed neutron fraction β. Where $0 < \beta < 1$.

The prompt source becomes

$$\frac{dn}{dt} = \frac{k_{\text{eff}}(1 - \beta) - 1}{l} n$$

$$\frac{dn}{dt} = \frac{k_{\text{eff}} - \beta k_{\text{eff}} - 1}{l} n$$

$$\frac{dn}{dt} = \frac{k_{\text{eff}} - 1 - \beta k_{\text{eff}}}{l} n$$

$$\frac{dn}{dt} = \frac{k_{\text{eff}} - 1}{l} n - \frac{\beta k_{\text{eff}}}{l} n$$

So we have reduced contribution of prompt neutrons.

How do we add the delayed neutron concentrations? Suppose each fission results in some delayed precursors. The concentrations of which we shall denote as C. This is determined by the fission rate:

$$\text{fission neutron production rate} = \frac{k_{\text{eff}}}{l} n$$

$$\text{precursor production rate} = \beta * \text{fission neutron production rate} = \beta \frac{k_{\text{eff}}}{l} n$$

If we have the decay rate of the precursors, λC, where λ is a decay constant, we can write balance equations for the precursors.

$$\frac{dC}{dt} = \beta \frac{k_{\text{eff}}}{l} n - \lambda C$$

The neutron production rate, assuming one precursor decay results in one neutron, is:

$$\lambda C$$

These two equations must be solved simultaneously in order to obtain the neutron population over time:

$$\frac{dn}{dt} = \frac{k_{\text{eff}} - 1}{l} n - \frac{\beta k_{\text{eff}}}{l} n + \lambda C$$

$$\frac{dC}{dt} = \beta \frac{k_{\text{eff}}}{l} n - \lambda C$$

Now, we see the term $\frac{l}{k_{\text{eff}}}$ appear multiple times here. This is important because l represents time to loss, whereas k_{eff} is the ratio of fission production to loss. Hence, $\frac{l}{k_{\text{eff}}}$ represents the time the neutron is born until the next generation born. This is called mean generation time Λ.

$$\Lambda = \frac{l}{k_{\text{eff}}}$$

Though TBH, it's nice to reduce the number of terms here:

$$\frac{dn}{dt} = \frac{k_{\text{eff}} - 1}{l} n - \frac{\beta}{\Lambda} n + \lambda C$$

$$\frac{dC}{dt} = \beta \frac{1}{\Lambda} n - \lambda C$$

We can introduce that into our prompt term as well:

$$\frac{dn}{dt} = \frac{k_{\text{eff}} - 1}{k_{\text{eff}}} \frac{k_{\text{eff}}}{l} n - \frac{\beta}{\Lambda} n + \lambda C$$

$$\frac{dn}{dt} = \frac{k_{\text{eff}} - 1}{k_{\text{eff}}} \frac{1}{\Lambda} n - \frac{\beta}{\Lambda} n + \lambda C$$

$$\frac{dC}{dt} = \frac{\beta}{\Lambda} n - \lambda C$$

We have this leftover term $\frac{k_{\text{eff}}-1}{k_{\text{eff}}}$. This is called reactivity (ρ):

$$\rho(t) \equiv \frac{k_{\text{eff}} - 1}{k_{\text{eff}}}$$

$$\frac{dn}{dt} = \frac{\rho}{\Lambda} n - \frac{\beta}{\Lambda} n + \lambda C$$

$$\frac{dC}{dt} = \frac{\beta}{\Lambda} n - \lambda C$$

With some tidying up, we get the PRKE with delayed neutrons for one precursor group:

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \lambda C$$

$$\frac{dC}{dt} = \frac{\beta}{\Lambda} n - \lambda C$$

For multiple precursor groups, eg. 6 or 8, we simply introduce the delayed precursor equations for each group (denoted j) like so:

$$\frac{dC_j}{dt} = \frac{\beta_j}{\Lambda} n - \lambda_j C_j$$

The total delayed fraction for N precursor groups is:

$$\beta = \sum_{j=1}^{N} \beta_j$$

The total delayed neutron source will then be the summation of the contributions from each group:

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_{j=1}^{N} \lambda_j C_j$$

Therefore, the delayed neutron precursor group for N equations is:

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_{j=1}^{N} \lambda_j C_j$$

$$\frac{dC_j}{dt} = \frac{\beta_j}{\Lambda} n - \lambda_j C_j$$

Now, if the fuel heats up, or if the coolant heats up, there will be a significant impact on neutron cross sections which could impact reactivity. These are known as reactor feedback mechanisms. These delayed neutron equations do not take into account the heating and cooling of the reactor. Hence, they are often called zero power PRKE equations.

Why we want to operate in a Delayed Critical Regime

Criticality Accidents: Slotin Incident
See reference (Stratton & Smith, 1989) page 75 to 76. Sphere went prompt critical, no time for experimenter (Louis Slotin) to react. 10< cents above prompt criticality (Oettingen, 2018). How do you calculate a cent or dollar?

$$\text{reactivity (dollars)} = \frac{\rho}{\beta}$$

If $\rho > \beta$, we get prompt critical ($1). GG, no time to react. Here is a "Fat man and little boy" movie clip.

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_{j=1}^{N} \lambda_j C_j$$

$$\frac{dC_j}{dt} = \frac{\beta_j}{\Lambda} n - \lambda_j C_j$$

In this case $\rho > \beta$. If prompt critical, what happens? No time to react. Louis Slotin died 9 days after the accident due to acute radiation poisoning (Oettingen, 2018; Stratton & Smith, 1989).

Documentation

Feedback mechanisms for PRKE

For a typical PRKE equation, shown here:

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_{j=1}^{N} \lambda_j C_j$$

$$\frac{dC_j}{dt} = \frac{\beta_j}{\Lambda} n - \lambda_j C_j$$

We do not usually consider other feedback effects such as fuel temperature or other effects. However, these are important to consider in actual reactor kinetics.

Control Rods
The first of these things are control rods. Control rods will definitely impact reactivity of the reactor at any time. The rod worth (which is the control rod's impact on reactivity) can be expressed as a function of length inserted in the reactor (Lamarsh et al., 2001):

$$\rho_\omega(x) = \rho_\omega(H)\left[\frac{x}{H} - \frac{1}{2\pi}\sin\left(\frac{2\pi x}{H}\right)\right]$$

This is for a cylindrical reactor of height H where x is the length of which the rod is inserted into the core. $\rho_\omega(H)$ is the maximum rod worth within the core when the rod is fully inserted. A rough estimate of control rod worth $\rho_\omega(H)$ can be 0.07 (Lamarsh et al., 2001).

Fuel Temperature Feedback (Doppler Effect)
When U-238 heats up, the resonances change in shape because of the doppler effect. As a result, resonance escape probability p decreases with increasing fuel temperature (as long as it contains U-238).

For this, we define α_T as the temperature coefficient of reactivity, defined as (Lamarsh et al., 2001):

$$\alpha_T = \frac{d\rho}{dT}$$

if we substitute, $\rho = \frac{k-1}{k}$:

$$\alpha_T = \frac{1}{k^2}\frac{dk}{dT}$$

At $k \approx 1$:

$$\alpha_T \approx \frac{1}{k}\frac{dk}{dT}$$

This is useful because in the six factor formula,

$$k = \eta f p \varepsilon P_{\text{TNL}} P_{\text{FNL}}$$

As mentioned, resonance escape probability p is the temperature dependent term.

$$\ln k = \ln(\eta f \varepsilon P_{\text{TNL}} P_{\text{FNL}}) + \ln p$$

For temperature changes,

$$\frac{\partial}{\partial T}\ln k = \frac{\partial}{\partial T}\ln(\eta f \varepsilon P_{\text{TNL}} P_{\text{FNL}}) + \frac{\partial}{\partial T}\ln p$$

We assume only p depends on temperature T.

$$\frac{\partial}{\partial T}\ln k = \frac{\partial}{\partial T}\ln p$$

$$\frac{1}{k}\frac{\partial k}{\partial T} = \frac{\partial}{\partial T}\ln p$$

Using total derviatives

$$\frac{1}{k}\frac{dk}{dT} = \frac{d}{dT}\ln p$$

$$\frac{1}{k}\frac{dk}{dT} = \frac{1}{p}\frac{dp}{dT}$$

$$\alpha \approx \frac{1}{p}\frac{dp}{dT}$$

How does resonance escape probability depend on temperature such that we can differentiate it?

From literature, we are given (Lamarsh et al., 2001):

$$p = \exp\left[-\frac{N_F V_F I}{\zeta_M \Sigma_{sM} V_M}\right]$$

N_F is atom density of fuel, V_F and V_M are the volume of fuel and moderator in a unit cell respectively. Σ_{sM} is scattering cross section of moderator and ζ_M is a constant for the moderator.

I is the portion which is temperature dependent. It can be determined by the formula:

$$I(T) = I(300K)\left[1 + \beta_I\left(\sqrt{T} - \sqrt{300K}\right)\right]$$

$$\beta_I = A' + \frac{C'}{a\rho}$$

For a fuel rod using β_I, a is rod radius in cm and ρ is fuel density in $\frac{g}{\text{cm}^3}$

Constants for β_I are given in table 7.4 in lamarsh's textbook (Lamarsh et al., 2001).

After some differentiation, (Lamarsh et al., 2001)

$$\alpha_{\text{prompt}} = \frac{N_F V_F I(300K)}{\zeta_M \Sigma_{sM} V_M} \frac{\beta_I}{2\sqrt{T}}$$

or

$$\alpha_{\text{prompt}} = \frac{\beta_I}{2\sqrt{T}} \ln\left[\frac{1}{p(300K)}\right]$$

Implicit Time Integration Schemes for Zero Power PRKE equation

For a typical PRKE equation:

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_{j=1}^{N} \lambda_j C_j$$

$$\frac{dC_j}{dt} = \frac{\beta_j}{\Lambda} n - \lambda_j C_j$$

There are several ways to tackle this equation. One common one for prompt reactivity feedback is the prompt jump approximation. In this case, $\rho < \beta$ and $\frac{dn}{dt} = 0$. This for simulation in delayed criticality cases. It is also less computationally expensive and longer timesteps can be taken.

However, if we want to simulate even prompt reactivity feedback, then we can simulate these numerically using the Euler integration scheme. We just need to be wary of timestep. Moreover, if prompt supercriticality is reached, there is no way to control the reaction. So some other feedback mechanism must be put into the fuel to simulate the self-regulating effect of negative fuel temperature feedback.

$$\frac{dn}{dt} = \frac{\rho - \beta}{\Lambda} n + \sum_{j=1}^{N} \lambda_j C_j$$

Should we discretise, and use explicit Euler scheme:

$$\frac{n^{t+\Delta t} - n^t}{\Delta t} = \frac{\rho - \beta}{\Lambda} n^t + \sum_{j=1}^{N} \lambda_j C_j^t$$

The implicit Euler scheme is:

$$\frac{n^{t+\Delta t} - n^t}{\Delta t} = \frac{\rho - \beta}{\Lambda} n^{t+\Delta t} + \sum_{j=1}^{N} \lambda_j C_j^{t+\Delta t}$$

$$\frac{n^{t+\Delta t} - n^t}{\Delta t} = \frac{\rho - \beta}{\Lambda} n^{t+\Delta t} + \sum_{j=1}^{N} \lambda_j C_j^{t+\Delta t}$$

The delayed neutron precursor equations would be for solid fuelled reactors:

$$\frac{C_j^{t+\Delta t} - C_j^t}{\Delta t} = \frac{\beta_j}{\Lambda} n^{t+\Delta t} - \lambda_j C_j^{t+\Delta t}$$

These would give us an upper triangular or lower triangular matrix system of equations to solve at every timestep. For a small system of equations (six to eight equations at most), I suppose the solving speed should be reasonably fast.

For numerical stability, the time step Δt should be on the same order of magnitude as Λ. This is so that we don't have a stiff system of equations.

$$\frac{\Delta t}{\Lambda} \sim \mathcal{O}(1)$$

Of course, implicit schemes have a better stability than explicit schemes, so we may be able to use larger timesteps than the above constraints.

For implicit schemes can then arrange the equations:

$$n^{t+\Delta t} - n^t = \Delta t \frac{\rho - \beta}{\Lambda} n^{t+\Delta t} + \sum_{j=1}^{N} \Delta t \lambda_j C_j^{t+\Delta t}$$

$$n^{t+\Delta t} - \Delta t \frac{\rho - \beta}{\Lambda} n^{t+\Delta t} - \sum_{j=1}^{N} \Delta t \lambda_j C_j^{t+\Delta t} = n^t$$

$$n^{t+\Delta t} \left[1 - \Delta t \frac{\rho - \beta}{\Lambda} \right] - \sum_{j=1}^{N} \Delta t \lambda_j C_j^{t+\Delta t} = n^t$$

Likewise for the delayed precursors:

$$C_j^{t+\Delta t} - \Delta t \frac{\beta_j}{\Lambda} n^{t+\Delta t} + \Delta t \lambda_j C_j^{t+\Delta t} = C_j^t$$

$$C_j^{t+\Delta t} \left[1 + \Delta t \lambda_j \right] - \Delta t \frac{\beta_j}{\Lambda} n^{t+\Delta t} = C_j^t$$

Hence, the discretised PRKE in implicit Euler scheme becomes:

$$n^{t+\Delta t} \left[1 - \Delta t \frac{\rho - \beta}{\Lambda} \right] - \sum_{j=1}^{N} \Delta t \lambda_j C_j^{t+\Delta t} = n^t$$

$$C_j^{t+\Delta t} \left[1 + \Delta t \lambda_j \right] - \Delta t \frac{\beta_j}{\Lambda} n^{t+\Delta t} = C_j^t$$

We can start from a neutron population of 1, the initial condition, and zero precursor concentration at the initial condition. We then adjust reactivity. The neutron population and precursor concentration should change over time. Alternatively, we should simulate a background neutron source S_b which is realistic in real reactors:

$$n^{t+\Delta t} \left[1 - \Delta t \frac{\rho - \beta}{\Lambda} \right] - \sum_{j=1}^{N} \Delta t \lambda_j C_j^{t+\Delta t} = n^t + S_b \Delta t$$

$$C_j^{t+\Delta t} \left[1 + \Delta t \lambda_j \right] - \Delta t \frac{\beta_j}{\Lambda} n^{t+\Delta t} = C_j^t$$

This will allow the neutron population never to fall to zero. The other feedback mechanisms impacting reactivity can then be simulated in a decoupled fashion with these point kinetics equations. These can be control rod worth or even fuel temperature feedback.

Now, I also foresee that fuel temperature feedback is the only means by which the prompt reactivity feedback can be stabilised. Therefore, the fuel temperature feedback needs to be time integrated using a similar time scale. Or it should be tightly coupled in the implicit integration scheme. The power production rate is:

$$P(t) = n(t)v_n\Sigma_f * \text{energy per fission}$$

For fuel temperature, the energy balance for a convection cooled reactor can be written as:

$$mc_p\frac{\partial T_{\text{fuel}}}{\partial t} = P(t) - hA_s(T_{\text{fuel}} - T_{\text{surr}})$$

$$mc_p\frac{\partial T_{\text{fuel}}}{\partial t} = n(t)v_n\Sigma_f * \text{energy per fission} - hA_s(T_{\text{fuel}} - T_{\text{surr}})$$

Where T_{fuel} is the fuel temperature and T_{surr} is the ambient temperature or coolant temperature. The fuel temperature is in turn important for determining the fuel temperature feedback coefficient. Unfortunately, the fuel temperature feedback mechanism is quite non-linear. So we will have to explicitly integrate this system of equations. This makes is quite pointless to form a system of matrices in the first place. Nevertheless, prompt reactivity insertion aside, it is still quite a good exercise to do.

Let's write out the matrix for a four precursor group as an example:

$$n^{t+\Delta t}\left[1 - \Delta t\frac{\rho - \beta}{\Lambda}\right] - \sum_{j=1}^{N}\Delta t\lambda_j C_j^{t+\Delta t} = n^t + S_b\Delta t$$

$$n^{t+\Delta t}\left[1 - \Delta t\frac{\rho - \beta}{\Lambda}\right] - \Delta t\lambda_1 C_1^{t+\Delta t} - \Delta t\lambda_2 C_2^{t+\Delta t} - \Delta t\lambda_3 C_3^{t+\Delta t} - \Delta t\lambda_4 C_4^{t+\Delta t} = n^t + S_b\Delta t$$

The other four precursor equations are:

$$C_1^{t+\Delta t}[1 + \Delta t\lambda_1] - \Delta t\frac{\beta_1}{\Lambda}n^{t+\Delta t} = C_1^t$$

$$C_2^{t+\Delta t}[1 + \Delta t\lambda_2] - \Delta t\frac{\beta_2}{\Lambda}n^{t+\Delta t} = C_2^t$$

$$C_3^{t+\Delta t}[1 + \Delta t\lambda_3] - \Delta t\frac{\beta_3}{\Lambda}n^{t+\Delta t} = C_3^t$$

$$C_4^{t+\Delta t}[1 + \Delta t\lambda_4] - \Delta t\frac{\beta_4}{\Lambda}n^{t+\Delta t} = C_4^t$$

The matrix becomes:

$$\begin{pmatrix} \left[1-\Delta t\frac{\rho-\beta}{\Lambda}\right] & -\Delta t\lambda_1 & -\Delta t\lambda_2 & -\Delta t\lambda_3 & -\Delta t\lambda_4 \\ -\Delta t\frac{\beta_1}{\Lambda} & [1+\Delta t\lambda_1] & 0 & 0 & 0 \\ -\Delta t\frac{\beta_2}{\Lambda} & 0 & [1+\Delta t\lambda_2] & 0 & 0 \\ -\Delta t\frac{\beta_3}{\Lambda} & 0 & 0 & [1+\Delta t\lambda_3] & 0 \\ -\Delta t\frac{\beta_4}{\Lambda} & 0 & 0 & 0 & [1+\Delta t\lambda_4] \end{pmatrix} \begin{pmatrix} n^{t+\Delta t} \\ C_1^{t+\Delta t} \\ C_2^{t+\Delta t} \\ C_3^{t+\Delta t} \\ C_4^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} n^t + S_b\Delta t \\ C_1^t \\ C_2^t \\ C_3^t \\ C_4^t \end{pmatrix}$$

Okay, this doesn't seem like an upper triangular matrix, but it is rather sparse.

A six group system would be:

$$\begin{pmatrix} \left[1-\Delta t\frac{\rho-\beta}{\Lambda}\right] & -\Delta t\lambda_1 & -\Delta t\lambda_2 & -\Delta t\lambda_3 & -\Delta t\lambda_4 & -\Delta t\lambda_5 & -\Delta t\lambda_6 \\ -\Delta t\frac{\beta_1}{\Lambda} & [1+\Delta t\lambda_1] & 0 & 0 & 0 & 0 & 0 \\ -\Delta t\frac{\beta_2}{\Lambda} & 0 & [1+\Delta t\lambda_2] & 0 & 0 & 0 & 0 \\ -\Delta t\frac{\beta_3}{\Lambda} & 0 & 0 & [1+\Delta t\lambda_3] & 0 & 0 & 0 \\ -\Delta t\frac{\beta_4}{\Lambda} & 0 & 0 & 0 & [1+\Delta t\lambda_4] & 0 & 0 \\ -\Delta t\frac{\beta_5}{\Lambda} & 0 & 0 & 0 & 0 & [1+\Delta t\lambda_5] & 0 \\ -\Delta t\frac{\beta_6}{\Lambda} & 0 & 0 & 0 & 0 & 0 & [1+\Delta t\lambda_6] \end{pmatrix} \begin{pmatrix} n^{t+\Delta t} \\ C_1^{t+\Delta t} \\ C_2^{t+\Delta t} \\ C_3^{t+\Delta t} \\ C_4^{t+\Delta t} \\ C_5^{t+\Delta t} \\ C_6^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} n^t + S_b\Delta t \\ C_1^t \\ C_2^t \\ C_3^t \\ C_4^t \\ C_5^t \\ C_6^t \end{pmatrix}$$

We could pivot it to ensure it is lower triangular:

$$\begin{pmatrix} -\Delta t\frac{\beta_1}{\Lambda} & [1+\Delta t\lambda_1] & 0 & 0 & 0 & 0 & 0 \\ -\Delta t\frac{\beta_2}{\Lambda} & 0 & [1+\Delta t\lambda_2] & 0 & 0 & 0 & 0 \\ -\Delta t\frac{\beta_3}{\Lambda} & 0 & 0 & [1+\Delta t\lambda_3] & 0 & 0 & 0 \\ -\Delta t\frac{\beta_4}{\Lambda} & 0 & 0 & 0 & [1+\Delta t\lambda_4] & 0 & 0 \\ -\Delta t\frac{\beta_5}{\Lambda} & 0 & 0 & 0 & 0 & [1+\Delta t\lambda_5] & 0 \\ -\Delta t\frac{\beta_6}{\Lambda} & 0 & 0 & 0 & 0 & 0 & [1+\Delta t\lambda_6] \\ \left[1-\Delta t\frac{\rho-\beta}{\Lambda}\right] & -\Delta t\lambda_1 & -\Delta t\lambda_2 & -\Delta t\lambda_3 & -\Delta t\lambda_4 & -\Delta t\lambda_5 & -\Delta t\lambda_6 \end{pmatrix} \begin{pmatrix} n^{t+\Delta t} \\ C_1^{t+\Delta t} \\ C_2^{t+\Delta t} \\ C_3^{t+\Delta t} \\ C_4^{t+\Delta t} \\ C_5^{t+\Delta t} \\ C_6^{t+\Delta t} \end{pmatrix} = \begin{pmatrix} C_1^t \\ C_2^t \\ C_3^t \\ C_4^t \\ C_5^t \\ C_6^t \\ n^t + S_b\Delta t \end{pmatrix}$$

While it is not strictly speaking lower triangular, this system should be easier to solve compared to a non-pivoted matrix. (as far as I know)

Thus, we could write some code in order to simulate this piece. Again, we need to ensure that the timestep used is short because we need to simulate non-linear coupled temperature feedback behaviour. Nevertheless, in terms of programming, we can solve a six group system first without the feedback mechanisms to start our simulator.

For Rust code, I can make a struct with four vectors, one on β_j, one on λ_j, one on the volumetric concentrations of precursors and neutrons at current timestep. Timestep Δt, Λ and ρ will then be given as input parameters. S_b can be set to a fixed value of 10 counts per second per m^3. We can then run some tests.

Explicit RKF45 Time Integration Scheme for PRKE

Now, neutron lifetime is typically 2.31×10^{-4} s or about 200 microseconds. A timestep of around 100-500 microseconds would be good. The upper limit is around 500 microseconds. Too high a timestep results in some numerical instability. However, with a 100 microsecond timestep, the program tends to freeze likely because of a mutex lock (I'm trying to run many things in parallel).

Most times, the solver runs within 1-5 microseconds. But when the matrices are suddenly perturbed, the solution algorithm takes longer to solve, sometimes in excess of 100 microseconds. With a 100 microseconds timestep, the program freezes.

Bibliography

Bell, G. I., & Glasstone, S. (1970). *Nuclear reactor theory*.

Lamarsh, J. R., Baratta, A. J., & others. (2001). *Introduction to nuclear engineering* (Vol. 3). Prentice hall Upper Saddle River, NJ.

Mihalczo, J. (2004). *Radiation detection from fission*. United States. Department of Energy.

Oettingen, M. (2018). Criticality analysis of the Louis Slotin accident. *Nuclear Engineering and Design*, *338*, 92–101.

Stratton, W. R., & Smith, D. R. (1989). *A review of criticality accidents*.