Fundamental Kinetics Ideas

William N. Locke, Adjunct Professor
University of Pittsburgh, ME 2101
September 26, 2025
Revision 17

Introduction

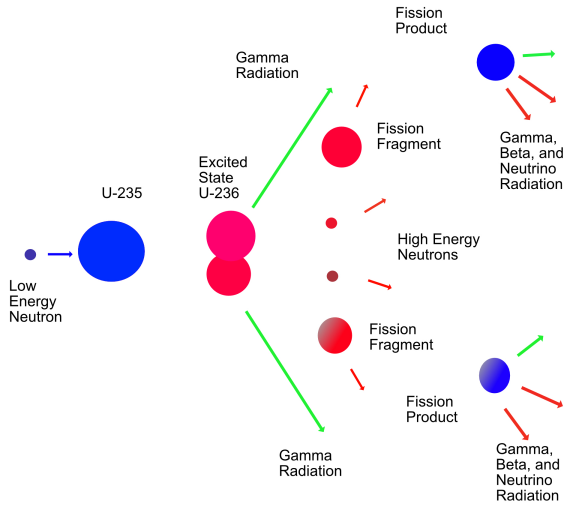
This document is provided to serve as class notes for a course in nuclear reactor kinetics. The purpose of this paper is to start with a point kinetics model and consider:

1. Review the fission process and other fundamentals.
2. The reactor model with only prompt neutrons.
3. Development of the reactor kinetics equations.
4. Traditional solution methods for the kinetics equations.
5. Matrix methods for solution of the kinetics equations.
6. The prompt jump assumption and related solutions of the kinetics equation.
7. The one delayed group assumption and the consequences of this assumption.
8. Ramp additions of reactivity.
9. Startup rate and related equations.
10. Source and reactivity transfer functions.

A point kinetics model is based on assuming that the differential equations associated with a reactor are separable in space and time. That is, the solutions for fission rate, flux, and other commodities may be treated as a product of a function of time multiplied by a function of space. Real reactors will not have this behavior, but in most cases the choice is a good approximation. In this course, our interest is the time behavior of reactors.

1. Review The Fission Process

The Fission Process



- The Low Energy Neutron is absorbed by the U-235 nucleus; energy is released as this happens. The energy is initially in the form of nuclear vibration. When the nucleus takes a dumbbell shape, the nuclear forces cannot prevent separation due to the strong electrostatic forces repelling the fission fragments.
- The fission fragments lose energy by emitting neutrons and gamma radiation.
- This process takes about 10^{-12} sec.
- The fission fragments repel each other departing with nearly 165 million electon-volts of kinetic energy.
- Prompt Neutrons and Gammas are produced directly from the fission. These neutrons have energies averaging nearly 2 MEV. On the average, about 2.5 prompt neutrons are born directly in fission.

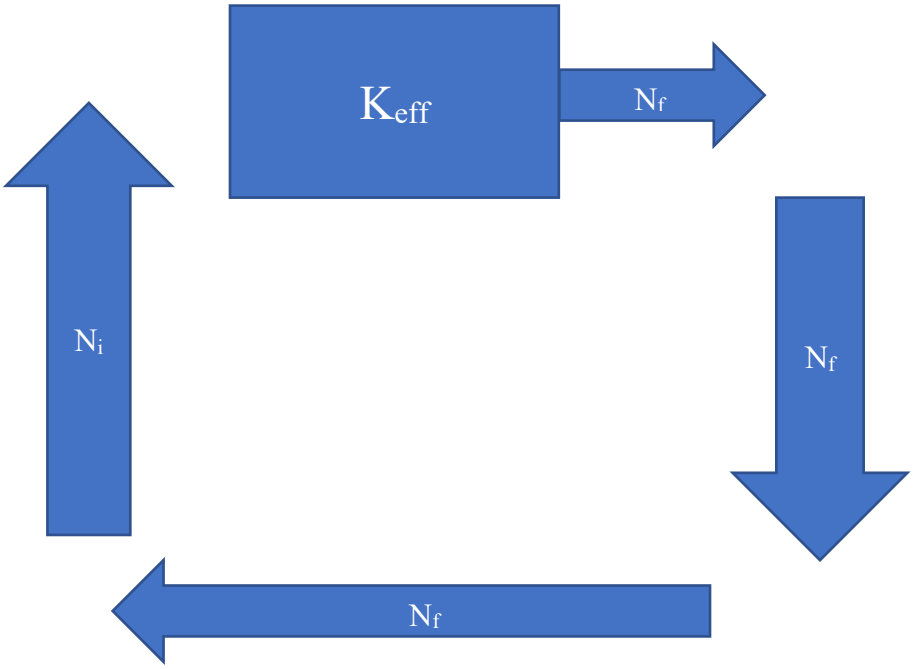
- The kinetic energy of the fission fragments is spent within the metal or ceramic matrix of the fuel. This is the primary source of heat in the reactor.
- The total energy released in a fission is approximately 200 MEV.
- Once the fission fragments lose their internal excitation energy, they become known as fission products.
- The fission products are neutron rich, and they undergo beta minus decay. This reduces the number of neutrons by one while increasing the number of protons. These decays have half-lives ranging from fractions of a second, to years.
- The beta decay of the fission products releases both beta and gamma radiation. This produces “decay heat”, this decay heat represents nearly 7% of the total reactor heat.
- The neutrons that have a high probability of causing fission have low energy, on the order of 0.025 EV. The neutrons born in fission need to be slowed down in order to cause a subsequent fission with high probability.
- Note an EV is an “electron-volt”, the energy gained by an electron falling through a potential difference of one volt. An MEV is a million electron-volts.

Neutron Lifecycle

Neutrons are produced in an operating reactor primarily from initial decay of excited state fission fragments (prompt neutrons). These neutrons have energies on the order of an MEV. There are also neutrons that come from the decay of fission products much later. And some neutrons are derived from other sources. These generally have energies on the order of tenths of an MEV. If there is to be a large chance of these neutrons causing fission, they need to be slowed down to energies that are hundredths an electron volt (EV). So, the kinetic energy of the neutrons needs to be reduced by a factor of roughly 1.0×10^{-8}. This is accomplished using collisions with a material (such as water) which contains a significant amount of hydrogen. At the same time, there is a chance that neutrons will be adsorbed in materials within the reactor that do not yield a fission. The neutrons may also leak out of the reactor. We will study these phenomena in detail in a future lesson. For now, define Keff, the number of neutrons that follow a cycle divided by the number of neutrons that start the cycle. We apply the term "cycle" here loosely as if the system grouped neutrons and passed them through the process sequentially. This idea is not what happens, but the thought is useful for understanding the behavior of a reactor.

2. Prompt Neutron Only Model (As if there were no delayed neutrons)

Examine the dynamic behavior of a point reactor with only prompt neutrons:



$$N_f - N_i = K_{eff}N_i - N_i$$

$$\frac{N_f - N_i}{\Delta t} = \frac{\left(1 - \frac{1}{K_{eff}}\right)K_{eff}N_i}{\Delta t}$$

For small Δt.

$$\frac{dN(t)}{dt} = \frac{\rho N(t)}{\Lambda}$$

Where generation time, $\Lambda = \frac{\Delta t}{K_{eff}}$ and $\rho = 1 - 1/K_{eff}$

If the reactivity is not a function of time the resulting neutron population is:

$$N(t) = N_0 e^{\frac{\rho t}{\Lambda}}$$

A typical value for the generation time is $5x10^{-5}$sec. And a reasonable reactivity would be $250x10^{-5}$. (This number would commonly be written as 250 pcm.) After one second this would lead to a large value of N(t).

$$N(1\text{sec}) = N_0 e^{250*1/5} = N_0 e^{50} = N_0 * 5.2x10^{21}$$

This is not acceptable if 250 pcm is a reasonable reactivity. The reactivity value is typical so a reactor without delayed neutrons would not be controllable.

3. Delayed Neutron Impact

It is instructive to consider an estimate where we simply modify the generation time by including the impact of the delayed neutron groups. This does not lead to correct dynamic behavior, but it demonstrates a key point related to the effective decay constant. If a radioactive element has a decay constant, λ ,then its mean expected life is $1/\lambda$. We think of the known precursors as existing within groups with similar decay constants.

Group	Half-life (sec)	λ_i	% Fission	β_i
1	55.72	0.0124	2.25	0.000215
2	22.72	0.0305	21.8605	0.001424
3	6.22	0.111	19.6899	0.001274
4	2.3	0.301	39.5349	0.002568
5	0.610	1.14	11.472	0.000748
6	0.230	3.01	4.1860	0.000273

The β_i in this table are the fraction of all neutrons born in fission that are born into the i'th group. The adjusted generation time weighted by the β_i could be:

$\Lambda'=$	The fraction of the neutrons that are born prompt times Λ	The fraction of neutrons born into the i'th group times the mean decay time of the group plus Λ

$$\Lambda' = (1-\beta)\Lambda + \sum_{i=1}^{i=6}\beta_i\left(\frac{1}{\lambda_i}+\Lambda\right) = \Lambda + \sum_{i=1}^{i=6}\beta_i/\lambda_i = 0.837sec$$

Here we have assumed that a delayed neutron, once born, will have the same chance as a prompt neutron of slowing down and causing a new fission. This is not exactly accurate as we will find later. Note: β is the sum of the β_i.

The interesting result here is that while the delayed neutrons are a small fraction of the total neutrons born in fission, they represent a major influence on the overall time constant in the system because of their relatively long lifetimes.

So, with 250 pcm (1 pcm = $1.0\text{x}10^{-5}$) of reactivity we have:

$$N(1\text{sec}) = N_0 e^{250x10^{-5}*1/0.837} = N_0 e^{0.0299} = 1.0303\, N_0$$

Without developing equations which demonstrate the dynamic behavior of the reactor we already have a feel for why delayed neutrons are so important in making reactor control possible. Their relatively long lifetimes have a significant impact.

We shall now develop a more detailed view of the kinetics of a reactor.

The Effective Delayed Neutron Fraction

The values of the delayed neutron fraction tabulated above are the fraction of the neutrons born in thermal fission that are born delayed. These are a property of the fuel itself and are fixed. The parameter that matters in a thermal reactor is the fraction of thermal neutrons that were born delayed. This means that not only do the neutrons need to be created but they also need to be slowed down to thermal energies as do the prompt neutrons. This introduces a complication.

The prompt neutrons are born with a mean energy of about 2 MEV. The delayed neutrons are born at many different energies, largely an order of magnitude less than the prompt neutron birth energy. This means that delayed neutrons are more likely to successfully thermalize than prompt neutrons. This effect will cause an increase in the effective value of this fraction. These new fractions are denoted as $\bar{\beta}_i$ and $\bar{\beta}$.

For a geometrically small reactor this value tends to be large as the leakage effects will accentuate the difference in the slowing down probability. The effect is much less in a large reactor.

Low enrichment cores will also have fuel conversion from U^{238} to Pu^{239}. Over time the fuel concentration shifts towards Pu^{239} while not eliminating U^{235} completely. Pu^{239} has a β of approximately 200×10^{-5} as compared to 640×10^{-5} for U^{235}. This tends to drive the $\bar{\beta}$ for the system down.

A typical large thermal reactor could have a $\bar{\beta}$ ranging from 700×10^{-5} at beginning of life down to about 500×10^{-5} at the end of life. (Data for AP1000, T.M. Sembiring *et al* 2018 *J. Phys.: Conf. Ser.* **962** 012030)

This paper is available at:
https://iopscience.iop.org/article/10.1088/1742-6596/962/1/012030/pdf

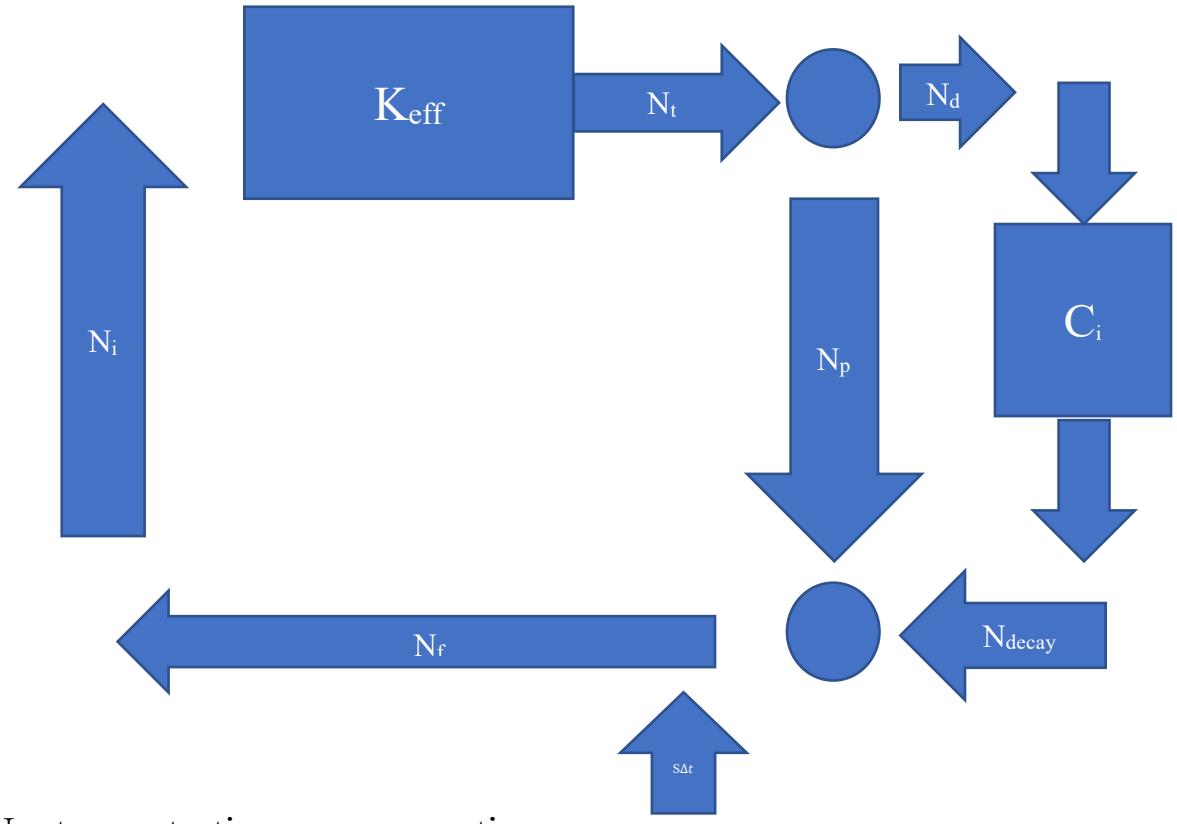
This document uses β_i and β throughout rather than the adjusted values. The kinetics transients and other calculations are done using these natural fuel values for U^{235}. For the cases where MATLAB programs are provided it would be a small matter to adjust these values if needed.

In the process of defining the internal kinetics of a reactor we define an entity called a delayed neutron precursor. This is a fission product which decays at some point releasing a delayed neutron. Real isotopes tend to release delayed neutrons with some probability. We make a distinction here. Precursors are a population that do release a delayed neutron so the probability that a delayed neutron is released from the decay of a precursor is one. We use the symbol "C_i" to represent the concentration of precursors in the i'th group. And we use the symbol, "C", to represent the total concentration of precursors.

We will now build a "cycle model" to provide an intuitive derivation of the reactor kinetics equations. We start with only the concept of K_{eff}, delayed neutrons with β and β_i and we think of about the numbers of neutrons as we go around a cycle.

In the future we will look at a much more first principles-based version of this derivation. The interesting thing is that we will find that our simple derivation produces the exact result, not a conceptual approximation.

1. Development of the Reactor Kinetics Equations



N_i = Neutrons starting one generation
N_t = Fictitious number of neutrons after K_{eff}
N_d = Number of neutrons that will be born delayed = $\beta K_{eff} Ni$
S = Number of neutrons from other sources in Δt.
$N_p = (1 - \beta) K_{eff} N_i$
$N_{decay} = \sum_{i=1}^{6} \lambda_i C_i \Delta t$
$N_f = (1 - \beta) K_{eff} N_i + \sum_{i=1}^{6} \lambda_i C_i \Delta t + S\Delta t$
The change in neutron population in one generation is:

$$N_f - N_i = (1 - \beta) K_{eff} N_i + \sum_{i=1}^{6} \lambda_i C_i \Delta t + S\Delta t - N_i$$

So

$$\frac{N_f - N_i}{\Delta t} = (1 - \beta)K_{eff}N_i + \sum_{i=1}^{6} \lambda_i C_i \Delta t + S - N_i$$

Or

$$\frac{N_f - N_i}{\Delta t} = \frac{\left(1 - \frac{1}{K_{eff}} - \beta\right)K_{eff}N_i}{\Delta t} + \sum_{i=1}^{6} \lambda_i C_i + S$$

This becomes for small delta t.

$$\frac{dN(t)}{dt} = \frac{(\rho - \beta)N(t)}{\Lambda} + \sum_{i=1}^{6} \lambda_i C_i + \mathrm{S}$$

Where $\Lambda = \frac{\Delta t}{K_{eff}}$

Similarly

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

Observations:

- These equations represent a group of seven differential equations. If the reactivity is piecewise constant they may be treated as linear equations and solved in closed form.
- The quantity $\rho = 1 - \frac{1}{K_{eff}}$ is the reactivity. It may be arbitrarily negative. It must be maintained less than β for the reactor to be controllable as may be seen by examining the first reactor kinetics equation.

- When the reactivity is not constant, these equations are nonlinear and are often solved using approximation techniques.

    - The Prompt Jump Assumption. (Good assumption): For small reactivity, the quantity $\Lambda \frac{dN(t)}{dt}$ is considered small enough to ignore. This removes much of the computational difficulty (stiffness) in the equations resulting in a reduction of the first kinetics equation to:

      $$N(t) = \Lambda(\sum_{i=1}^{6} \lambda_i C_i + S)/(\beta - \rho)$$

      This reduces the kinetics problem to solving the six precursor differential equations.

      As long as the reactivity is controlled to be much less than β this is adequate to develop accurate results.

      A stiff problem is one where the derivative can have large erroneous values when the unknown itself is small.

    - The one delayed group assumption (Very poor assumption): The six groups of precursors are treated as one group with a single decay constant. The sum is reduced to $\lambda_{eff} C$. This approach is used primarily in teaching environments. It is not helpful in understanding the precise values of the related parameters. In any case this assumption reduces the kinetics problem to a single differential equation coupled with the prompt jump approximation algebraic fission rate equation.

2. Traditional Solution Methods for the Reactor Kinetics Equations

Traditional Approach (Without Prompt Jump or one delayed group assumption)

Assume reactivity is not a function of time (Linear Case) and there are no external sources of neutrons. For now, set S = 0.

$$\frac{dn(t)}{dt} = \frac{n(t)(\rho - \beta)}{\Lambda} + \sum_{i=1}^{6} \lambda_i C_i(t)$$

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

This amounts to seven differential equations in seven function unknowns. The equations have constant coefficients and are known to have solutions which will be linear sums of terms.

$$n(t) = \sum_{j=0}^{6} a_j e^{\omega_j t}$$

$$C_i = \sum_{j=0}^{6} b_{ij} e^{\omega_j t}$$

The seven $\{\omega_{j=0:6}\}$ are the roots of the characteristic equation:

Let $n = ae^{\omega t}$ and $C_i(t) = b_i e^{\omega t}$. Recognizing that the derivative of $e^{\omega t}$ is just $\omega e^{\omega t}$ we see that all the terms with $e^{\omega t}$ will drop out of the equation leading to

$$\omega a = \frac{a(\rho - \beta)}{\Lambda} + \sum_{i=1}^{6} \lambda_i b_i$$

$$\omega b_i = \frac{n(t)\beta_i}{\Lambda} - \lambda_i b_i$$

Solving the second of these equations for b_i and substituting into the first gives the following form with "a" canceled.

$$\omega = \frac{(\rho - \beta)}{\Lambda} + \frac{\beta_i \lambda_i}{\Lambda(\omega + \lambda_i)}$$

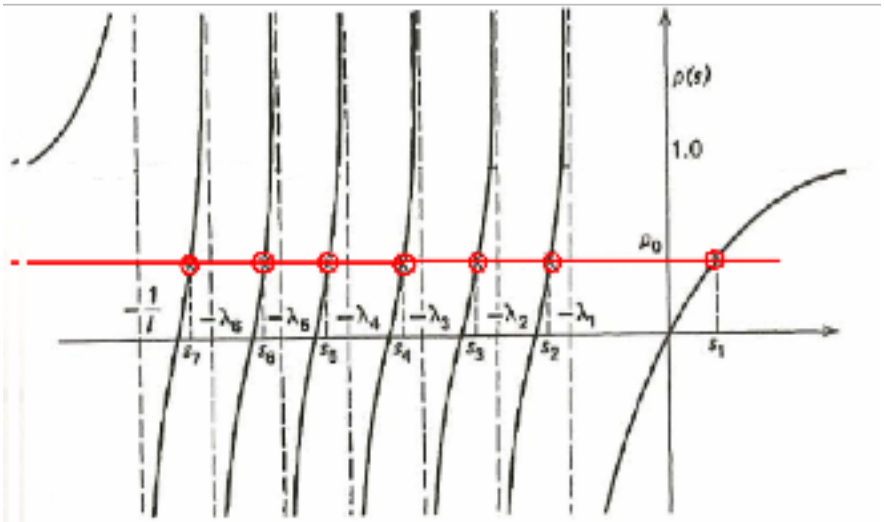
Now solving for the reactivity, we get:

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

This is known as the in-hour equation (inverse-hour), and it has the following properties:

1. The reactivity is limited by the physics of the system. For example, there are about 2.5 neutrons created per fission and a fuel absorption has only approximately 5/6 chance of creating a fission. So K_{eff} could be no larger than $\frac{2.5*5}{6}$. This alone would bound the reactivity at 0.52. Realistically other absorption and leakage would further limit the reactivity. As we have seen, safe reactor operation must limit reactivity to well below β.
2. There are seven real roots for ω and they vary over a large range.
3. If $\rho > 0$ the equation will have one positive root, ω_0. All others are non-zero and negative. For large positive reactivities much greater than β the asymptotic value of $\omega_0 = \rho/\Lambda$.
4. If $\rho < 0$ the equation will have seven negative roots.

5. The most negative eigenvalue is asymptotic to $\frac{-1}{\Lambda K_{eff}}$. This is on the order of -1.0x10^5 sec^{-1}
6. In the positive and negative directions, each eigenvalue is asymptotic to one of the $\{\lambda_i\}$ except at the ends.



A graphical determination of the roots to the inhour equation

Taken from J. Duderstadt & L. Hamilton, "Nuclear Reactor Analysis", p. 245

The reference is available at: http://milproj.dc.umich.edu/pdfs/books/1976_Nuclear%20Reactor%20Analysis.pdf

Here are some solutions for ω assuming $\Lambda = 50\mu sec$

ρ/β	0.0	0.1	0.25	1.0	4.0
ω_0	-0.0000	0.0102	0.0395	6.6280	1161.0
ω_1	-0.0144	-0.0138	-0.0134	-0.0128	-0.012504
ω_2	0.0682	-0.0628	-0.0548	-0.0380	-0.031206
ω_3	-0.1950	-0.1877	-0.1764	-0.1376	-0.11367
ω_4	-1.0203	-1.0094	-0.9889	-0.6921	-0.31443
ω_5	-2.8992	-2.8880	-2.8664	-2.2738	-1.1507
ω_6	-129.4078	-116.5533	-97.2944	-8.0785	-3.0272

ρ/β	-0.1	-0.25	-1.0	-10
ω_0	-0.0058468	-0.0097112	-0.011986	-0.012423
ω_1	-0.01532	-0.017606	-0.024912	-0.029871
ω_2	-0.073233	-0.079534	-0.095291	-0.10928
ω_3	-0.20185	-0.21115	-0.24199	-0.29042
ω_4	-1.0295	-1.0409	-1.0743	-1.1243
ω_5	-2.9085	-2.92	-2.9531	-3.0017
ω_6	-142.27	-161.58	-258.2	-1419

Prompt Criticality $\rho/\beta = 1.0$

This has been the traditional approach towards understanding reactor kinetics. The method requires a constant reactivity so it may only be used in a region where fission rate does not affect reactivity.

What remains is to determine the 49 coefficients for the exponential terms in each of the sums.

$$n(t) = \sum_{j=0}^{6} a_j e^{\omega_j t}$$

$$C_i = \sum_{j=0}^{6} b_{ij} e^{\omega_j t}$$

This done by matching the initial conditions on n and C_i as well as their derivatives and is a great deal of work.

Next, we will look at another approach that lends itself more readily to modern tools such as MATLAB, Python, Mathematica, and so on. The new method also requires a constant reactivity for now.

3. Matrix Method for solving the Reactor Kinetics Equations

Modern computer programs make manipulation of matrixes and vectors easy. This includes the computation of eigenvalues and eigenvectors related to matrixes. This means that a simple method is available for solving the kinetics equations which allows computation in only a few lines of programming. For example, MATLAB can be used to perform this quickly.

The seven reactor kinetics equations may be written as:

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

Where:

$$X(t) = \begin{bmatrix} n(t) \\ C_1(t) \\ C_2(t) \\ C_3(t) \\ C_4(t) \\ C_5(t) \\ C_6(t) \end{bmatrix} \quad A = \begin{bmatrix} \frac{\rho-\beta}{\Lambda} & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 \\ \beta_1/\Lambda & -\lambda_1 & 0 & 0 & 0 & 0 & 0 \\ \beta_2/\Lambda & 0 & -\lambda_2 & 0 & 0 & 0 & 0 \\ \beta_3/\Lambda & 0 & 0 & -\lambda_3 & 0 & 0 & 0 \\ \beta_4/\Lambda & 0 & 0 & 0 & -\lambda_4 & 0 & 0 \\ \beta_5/\Lambda & 0 & 0 & 0 & 0 & -\lambda_5 & 0 \\ \beta_6/\Lambda & 0 & 0 & 0 & 0 & 0 & -\lambda_6 \end{bmatrix} \quad S = \begin{bmatrix} \mathrm{So} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad Xo = \frac{n_0}{\Lambda} \begin{bmatrix} \Lambda \\ \beta_1/\lambda_1 \\ \beta_2/\lambda_2 \\ \beta_3/\lambda_3 \\ \beta_4/\lambda_4 \\ \beta_5/\lambda_5 \\ \beta_6/\lambda_6 \end{bmatrix}$$

So is the source rate and n_0 is the initial fission rate.

Assuming the reactivity is constant this equation has a solution using the integrating factor: $e^{-\mathrm{At}}$

$$X(t) = e^{\mathrm{At}} Xo + e^{\mathrm{At}} \int_0^t e^{-At'} \mathrm{S(t)}\, \mathrm{d}t'$$

and

$$e^{\mathrm{At}} = \sum_{i=0}^{\infty} (1/i!)(\mathrm{At})^i$$

We will assume the source term S is constant which results in X(t) being:

$$X(t) = e^{\mathrm{At}} Xo + A^{-1} e^{\mathrm{At}} (1 - e^{-\mathrm{At}}) S = e^{\mathrm{At}} X(0) + A^{-1} (e^{\mathrm{At}} - I) S$$

The obstacle here is the computation of the matrix exponential. This problem can be greatly simplified using diagonalization of A as follows. Define the eigenvalues and eigenvectors of A as follows:

$$\mathrm{Det}(A - \omega I) = 0 \qquad \mathrm{A}\vec{e}_{0..6}^{7x1} = \omega_{0..6} \vec{e}_{0..6}^{7x1}$$

The matrix is 7X7 and there will be seven solutions for omega. There will be seven eigenvectors E. Formally the solutions for omega could be real or complex where the complex roots would appear in congregate pairs. As it happens, the roots to this problem all are real. Likewise, the eigenvectors are real.

The key idea here is that a matrix M formed with its columns being the eigenvectors may be used to form a diagonal form of the A matrix.

$$M^{7x7} = [\vec{e}_0^{7x1} \quad \vec{e}_1^{7x1} \quad \vec{e}_2^{7x1} \quad \vec{e}_3^{7x1} \quad \vec{e}_4^{7x1} \quad \vec{e}_5^{7x1} \quad \vec{e}_6^{7x1}]$$

$$D = M^{-1}\mathrm{AM}$$

and

$$D = \begin{bmatrix} \omega_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \omega_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \omega_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \omega_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \omega_6 \end{bmatrix}$$

The following relationships also hold

$$A = \mathrm{MDM}^{-1} \qquad A^{-1} = \mathrm{MD}^{-1}M^{-1}$$

We can write

$$e^{Dt} = \sum_{i=0}^{\infty}(1/i!)(M^{-1}AMt)^i = \begin{bmatrix} e^{\omega_0 t} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{\omega_1 t} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{\omega_2 t} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{\omega_3 t} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{\omega_4 t} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{\omega_5 t} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{\omega_6 t} \end{bmatrix}$$

So, the solutions to the kinetics equations:

$$X(t) = e^{\mathrm{At}}Xo + A^{-1}e^{\mathrm{At}}(1 - e^{-\mathrm{At}})S = e^{\mathrm{At}}X(0) + A^{-1}(e^{\mathrm{At}} - I)S$$

May be rewritten as follows:

$$X(t) = \mathrm{Me}^{\mathrm{Dt}}M^{-1}Xo + \mathrm{MD}^{-1}(e^{\mathrm{Dt}} - I)M^{-1}S$$

Which may be directly computed once the eigenvectors and eigenvalues are known.

Once we have defined A, X, and S, MATLAB computes D and the M matrixes with one command. The e^{Dt} matrix, are computed. Finally, X for any value of t can be found. This gives us the fission rate and all precursor concentrations for each value of time.

Note that A, M, M^{-1}, D, and D^{-1} are constant for all steps in the iteration and need only be computed once. Further note that D^{-1} may be computed by inverting the diagonal elements of D. Further, products such as MD^{-1} and $M^{-1}S$ need only be computed once.

Software tools such as MATLAB use various mathematical methods, such as the Pade approximation, to compute the matrix exponential.

This avoids the need to perform the diagonalization explained in this document. Hence the matrix nay be found simply as ExpAt = expm(A*t). We will use the diagonalization method because it is instructive.

Linear System Stability

A linear system defined by

$$\frac{dX^{7x1}}{dt} = A^{7x7}X^{7x1} + B^{7x1}$$

is stable if all the eigenvalues of A have negative real parts. This implies that for any finite B there will be a steady state value of X given by

$$X^{7x1}{}_{ss} = -(A^{7x7})^{-1}B^{7x1}$$

For example, for B equal to zero the steady state value will be zero.

Further we know that the condition for a matrix to be invertible (nonsingular) is that none of its eigenvalues be zero. This results from the fact that the determinate is the product of the eigenvalues of the matrix.

Using this idea, we can see that the stability of a linear system is not a function of the vector X. If the system is stable for one solution it is stable for all solutions. Indeed, this fact will lead us to realize that from a practical standpoint, few systems are linear in the extreme. For example, if we design an amplifier to be linear, the linearity will only apply over a range of inputs and outputs.

There are many ways that one might define a nonlinear system and discuss its solution. The following form is useful in our nuclear reactor context:

$$\frac{dX^{nx1}}{dt} = A^{nxn}(X^{n+1}_{history})X^{nx1} + B^{nx1}$$

Here the system A, X, and B is expanded to include parameters beyond the neutron density (Fission rate) and the precursor concentrations. They will include elements related to the coolant temperature, and any other features which impact the system reactivity. A becomes a function of X and its entire history.

The linear nuclear kinetics problem is stable whenever the reactivity is less than zero. It is not stable if the reactivity is greater than or equal to zero. (Note that stability in this formal sense is not the same idea as controllability. Also, we use the term controllability to mean that we may control the reactor which is different from the formal definition of this word used in system control theory.)

We will discuss definitions of controllable reactors later in the course.

We will discuss the nonlinear example shown above at length later in the course. When reactor fission rate impacts the temperature and hence the reactivity we have this situation. All this will be covered later in the course.

Example Transients

The pages that follow contain several examples transients. These plots are the log base ten of the fission rate (proportional to the log of the neutron density); and the log base ten if the normalized precursor concentrations.

We take advantage of the fact that A is constant and that we will use a constant time step in the iteration. Starting with the general solution above:

$$X(t) = \mathrm{M}e^{\mathrm{D}t}M^{-1}Xo + \mathrm{M}\mathrm{D}^{-1}(e^{\mathrm{D}t} - I)M^{-1}S$$

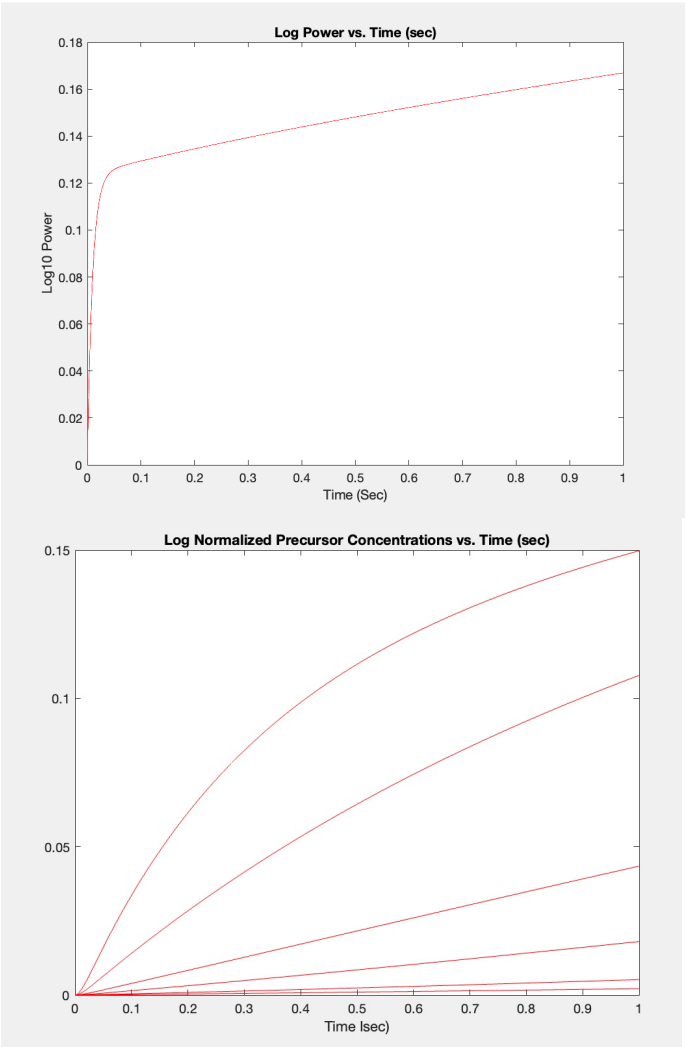
Define a time step length as $\tau, e^{A\tau} = \mathrm{M}e^{\mathrm{D}\tau}M^{-1}$. Then for subsequent steps we only need to multiply by $e^{A\tau}$ for the next time step.

Start with G = I. Also precompute B = $\mathrm{A}^{-1}S$. Then the iteration becomes:

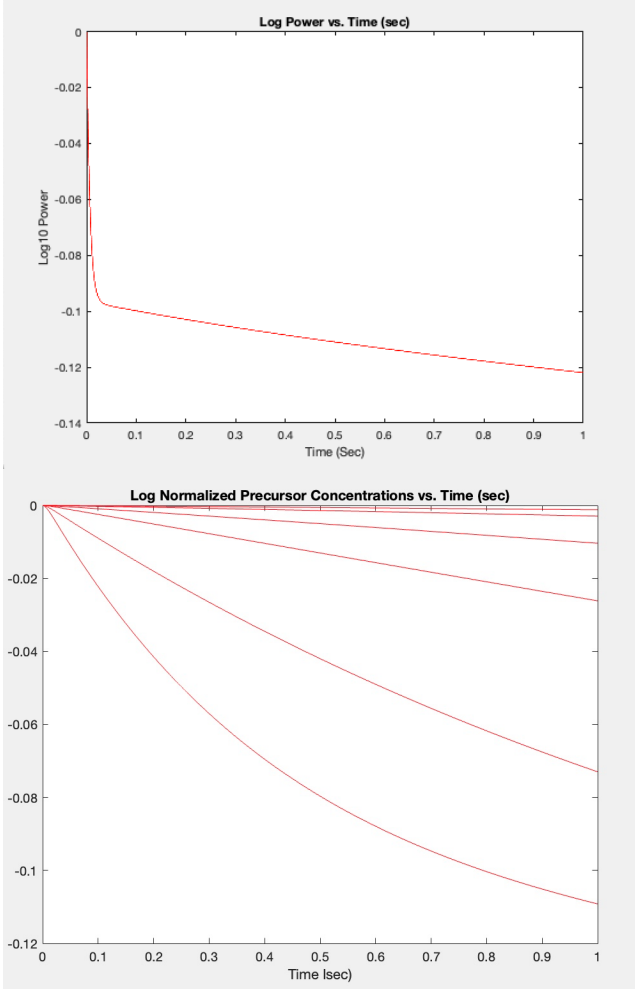
For Step = 0 to Number if Iterations
    G = $e^{A\tau}G$
    $X(Step + 1) = \mathrm{G}Xo + (\mathrm{G} - I)\mathrm{B}$
End

This greatly reduces the computation. We are using MATLAB and it requires array indexes to be greater than zero, so we need to modify the above steps to account for this limitation, but otherwise, what we have here is all that we need.

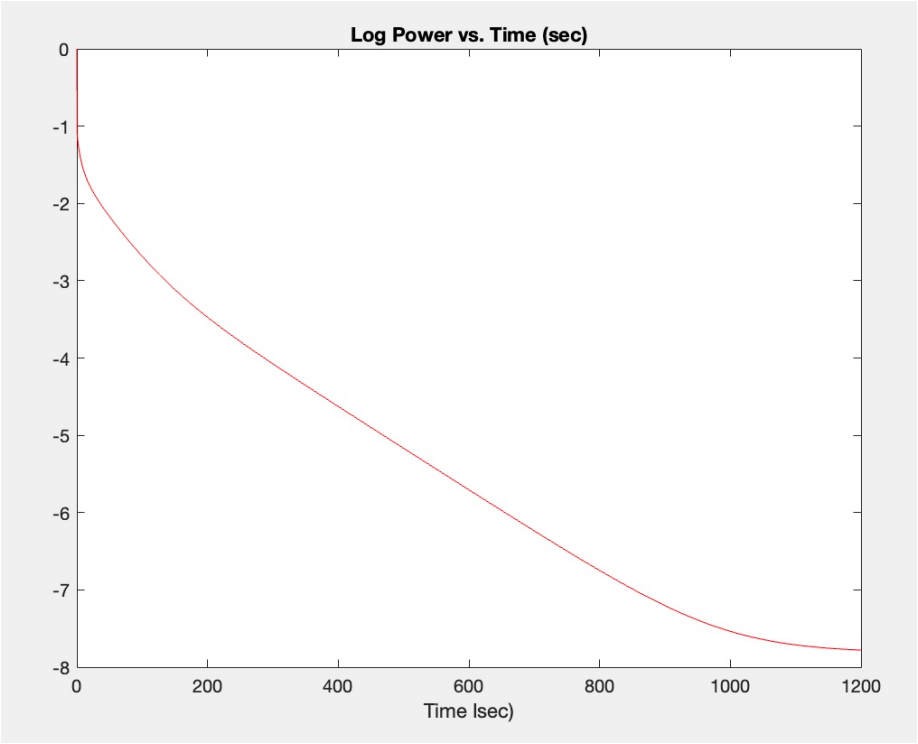
Linear System Examples: Reactivity = 0.25β for 1 sec



Examples: Reactivity = -0.25β for 1 sec

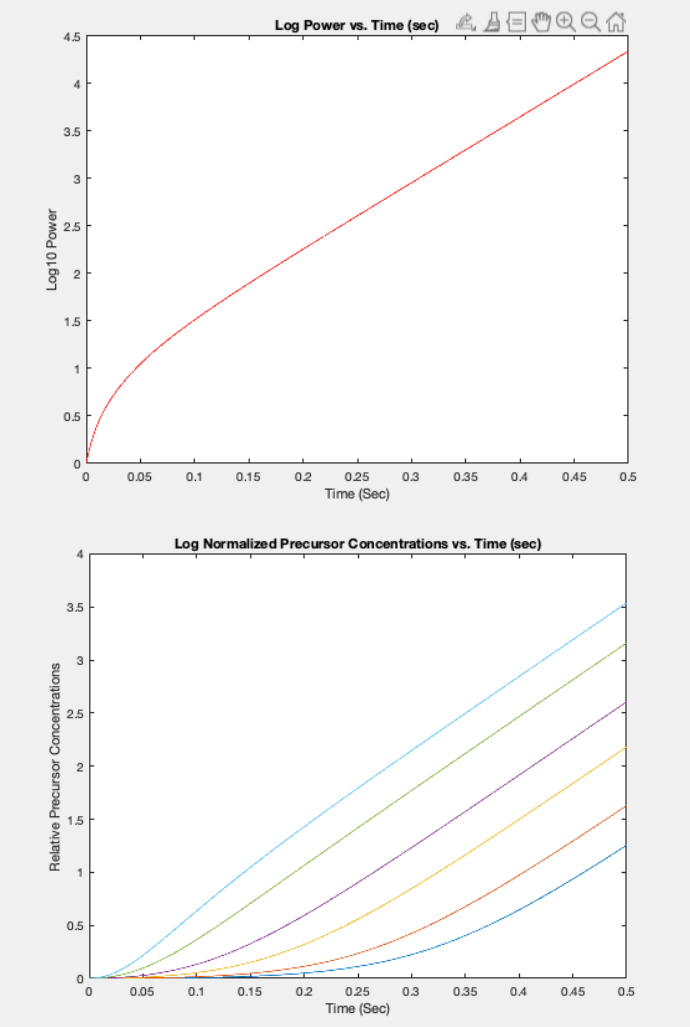


Examples: Reactivity = -10β for 20 minutes with a So $= 2.0\text{x}10^{-5}$



Note that the fission rate here is scaled to an initial value of one source the source rate is on that basis. The fission rate levels off here due to these source neutrons. We will look more at this later.

Examples: Reactivity = 1.1β for 0.5 sec



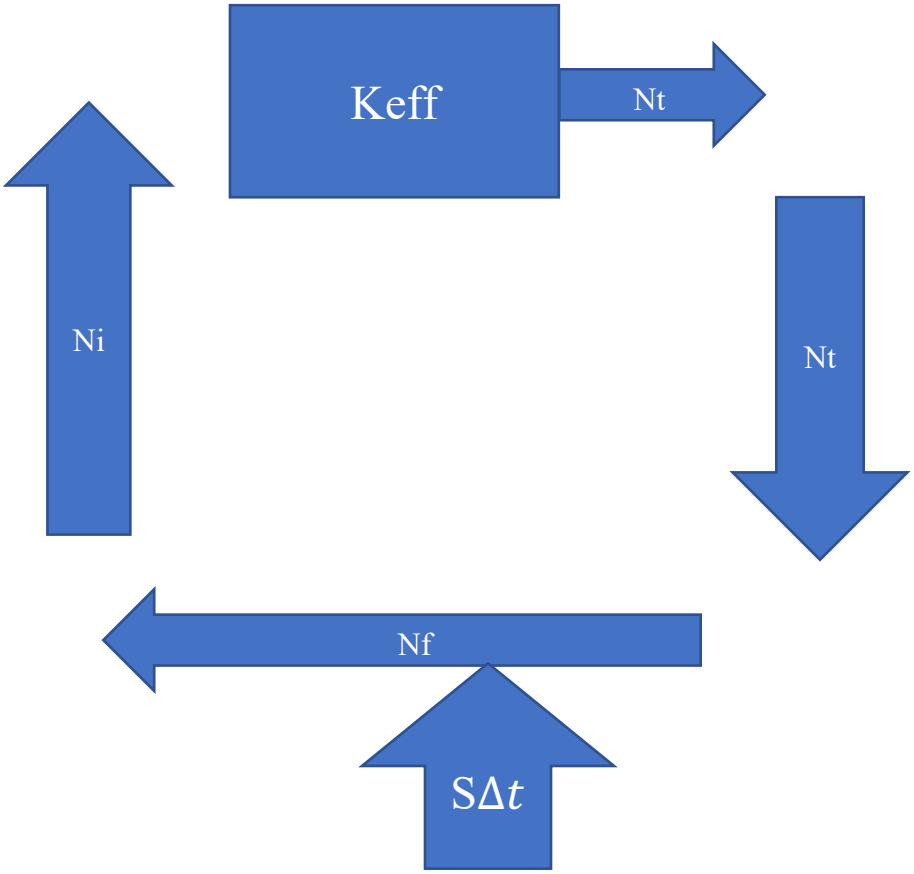
MATLAB Code to Create Transients

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SimplePower.m
% W.N. Locke
% May 2025
% Step response to a step at time 0. Initial fission rate normalized to 1.
% Example call: SimplePower(0.25,1)
% This will plot the 1 second transient with reactivity equal
% to 0.25*Beta
function SimplePower(ReactivityFractionOfBeta,TimeInterval)
    So = 2.0e-5; % Source Rate  normalized for a unit initial fission rate.
    S = [So;0;0;0;0;0;0];
    Tau = 1.0e-4; % Time step duration in seconds.
    T_hist = 0:Tau:TimeInterval;
    HistoryLength = length(T_hist);
    X_hist = zeros(7,HistoryLength);

    Betas = ...
        [0.00021;0.00141;0.00127;0.00255;0.00074;0.00027];
    BetaTotal = sum(Betas);
    Lambdas = ...
       [0.01246403;0.03052863;0.11141479;0.30130435;1.13606557;3.01304348];
    GenerationTime = 5.0e-5;
    Xo = [1;Betas./(Lambdas*GenerationTime)];
    function A = Amatrix(ReactivityFraction)
        A = zeros(7,7);
        A(1,1) = ...  % The three dots extends the line.
        BetaTotal*(ReactivityFraction-1)/GenerationTime;
        A(2:7,1) = Betas/GenerationTime; % Fills first column 2:7
        A(1,2:7) = Lambdas'; % Fills first row 2:7
        A = A + diag([0;-Lambdas]); % Fills diagonal after (1,1)
    end

    A = Amatrix(ReactivityFractionOfBeta);
    [M, D] = eig(A);
    ExpDTau = diag(exp(diag(D)*Tau),0);
    ExpATau = M*ExpDTau/M;   % '/' Right multiplies by the inverse of M.
    AinvS = A\S;     % '\' Left multiplies by the inverse of A.
    X_hist(:,1) = Xo(); % Copies vector Xo into the first col of X_hist.
    I = eye(7);  % This is a 7x7 unit matrix.
    G = I;
    for Step = 1:HistoryLength-1
        G = ExpATau*G;
        X_hist(:,Step+1)= G*Xo+(G-I)*AinvS;
    end
    figure
    plot(T_hist,log10(X_hist(1,:)),'r')
    title('Log Fission rate vs. Time (sec)');
    xlabel("Time Isec)")
    figure
    plot(T_hist,log10(X_hist(2:7,:)./Xo(2:7)),'r')
    title('Log Normalized Precursor Concentrations vs. Time (sec)');
    xlabel("Time (sec)")
end
```

Subcritical Multiplication



Consider a shutdown reactor. Subcritical multiplication is the process whereby source neutrons make up for the losses in the fission chain. When the reactor is shutdown the Keff is < 1 so neutrons are lost on each trip around the loop. The number can be made constant with an injection of neutrons from non-fission sources.

The delayed neutron part of the cycle is not presented here because we are in a virtual steady state. The steady state condition is:
$N_{Steady\ State} = K_{eff} N_{Steady\ State} + S\Delta t$
The above equation may be rearranged as

$$N_{Steady\ State} = \frac{S\Delta t}{1 - K_{eff}}$$

Now once again

$$\Lambda = \frac{\Delta t}{K_{eff}}$$

So

$$N_{Steady\ State} = \frac{-S\Lambda}{1 - 1/K_{eff}} = \frac{-S\Lambda}{\rho} = \frac{S\Lambda}{|\rho|}$$

The addition of a source impacts the first reactor kinetics equation as follows:

$$\frac{dn(t)}{dt} = \frac{n(t) * (\rho - \beta)}{\Lambda} + \sum_{i=1}^{6} \lambda_i C_i(t) + S$$

Here the value of S is in units of neutron density rate of change due to sources. If there were no n(t) nor precursors at t= 0, S is the rate at which fission rate would start to increase.

Where do source neutrons come from?

For a core that has been operated, the primary source of neutrons is the reaction $H_1^2 + \gamma_0^0 \rightarrow H_1^1 + n_0^1$. This gamma must have at least 2.23 MEV. These high energy gammas come from a relatively small number of fission product decay chains. There is a rapid reduction in this source over the first day after shutdown. Following that the level will reduce with a 12.8-day half-life due to Ba^{140} to La^{140} fission product decay. Following this, after several months, decay is controlled by other isotopes with half-lives of approximately one year.

In many reactors other sources of neutrons are installed to provide a visible count rate on the Source Range Instruments.

There also natural sources of neutrons including cosmic rays, spontaneous fission, and certain internal core (α, n) reactions. The last of these will vary based on the specifics of the fuel being used.

4. The Prompt Jump Assumption and Related Solutions of the Kinetics Equation

The reactor kinetics differential equations bring significant difficulty. This difficulty stems from the range of the eigenvalues of the A matrix. If the reactivity is 0.1β, we have the following eigenvalue set. These all have units of inverse time in seconds.

-116.5533	-2.8879	-1.0094	-187.7195e-3
-62.7713e-3	10.1868 e-3	-13.8297e-3	

The ultimate use of these numbers is to be multiplied by time and placed in exponentials. The first number here is a problem. Its related term decays rapidly, which can lead to computational issues. It develops that this problem traces to the use of the generation time in our differential equations. The step size needed to integrate it is much less than would be required for the other eigenvalues.

$$\frac{dN(t)}{dt} = \frac{(\rho-\beta)N(t)}{\Lambda} + \sum_{i=1}^{6} \lambda_i C_i + \mathrm{S}$$

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

One solution is to approximate these equations by replacing the first with another equation with $\Lambda \frac{dN(t)}{dt}$ set to zero. Doing so introduces error, but it will develop that the error is small so long as we do not have rapid jumps in reactivity and that the reactivity is well below β. We will handle jumps in reactivity as a discontinuity in the fission rate (recognizing that this is non-physical). This approximation is called the "Prompt Jump Assumption".

The first equation then becomes:

$$N(t) = \Lambda(\sum_{i=1}^{6} \lambda_i C_i + S)/(\beta - \rho)$$

If we pose an examination of a sudden step change in reactivity. We realize that the number of precursors is continuous. The fission rate would be continuous if we did not make the prompt jump assumption. However, fission rate becomes discontinuous.

$$-\frac{(\rho(0_-) - \beta)N(0_-)}{\Lambda} = \sum_{i=1}^{6} \lambda_i C_i(0_-) + S$$

$$-\frac{(\rho(0_+) - \beta)N(0_+)}{\Lambda} = \sum_{i=1}^{6} \lambda_i C_i(0_+) + S$$

The two right sides are the same because C's and S are continuous so

$$N(0_+) = N(0_-)\left[\frac{\beta - \rho(0_-)}{\beta - \rho(0_+)}\right]$$

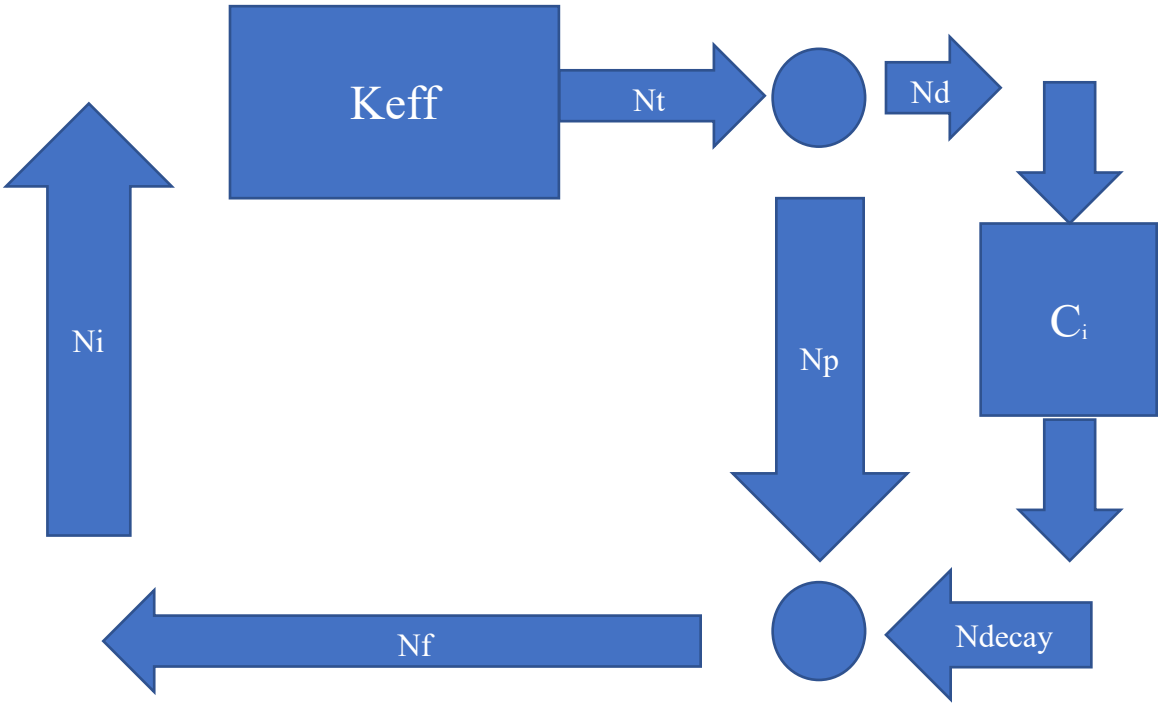
Or for an initially critical reactor with a step of reactivity this becomes the following:

$$N(0_+) = N(0_-)\left[\frac{\beta}{\beta - \rho}\right]$$

If we return to the plots that we showed above for fission rate transients. Step changes in reactivity both up and down produced a rapid change in the fission rate followed by a slow rise or fall depending on the sign of the reactivity change. What we have done with this assumption is to close the time for that initial rise or fall, down to zero.

Understanding the Prompt Jump

1. When K_{eff} is changed the inner prompt cycle N starts to rise.
2. The number of neutrons coming from the decay of precursors is not yet changing. The number of neutrons being lost from the cycle Nd is going up as the inner loop number is going up.
3. A quasi-steady state happens when the losses from the loop due to the precursor production is equal to the gains coming across K_{eff}. After that fission rate only rises as the precursor decay rate increases.



5. Computational Consequence of the Prompt Jump Approximation

We set the left-hand size of the of the following equal to zero and solve for n(t):

$$\frac{dn(t)}{dt} = \frac{n(t) * (\rho - \beta)}{\Lambda} + \sum_{i=1}^{6} \lambda_i C_i(t) + S$$

We start with the fission rate and full precursor equations:

$$N(t) = \Lambda(\sum_{i=1}^{6} \lambda_i C_i + S)/(\beta - \rho) \qquad \frac{dC_i(t)}{dt} = \frac{N(t) * \beta_i}{\Lambda} - \lambda_i C_i(t)$$

We can write the precursor differential equations in matrix form imbedding the fission rate equation directly as the production term involving N(t).

$$\frac{\mathrm{dC}^{6x1}}{\mathrm{dt}} = Apj^{6x6}\mathrm{C}^{6x1} + \mathrm{B}^{6x1}$$

$$C(t) = \begin{vmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{vmatrix} \qquad Co(0) = \begin{vmatrix} \beta_1/\lambda_1 \\ \beta_2/\lambda_2 \\ \beta_3/\lambda_3 \\ \beta_4/\lambda_4 \\ \beta_5/\lambda_5 \\ \beta_6/\lambda_6 \end{vmatrix} \frac{N_{S/D}}{\Lambda} \qquad B = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \end{bmatrix} \left(\frac{S}{\beta - \rho}\right)$$

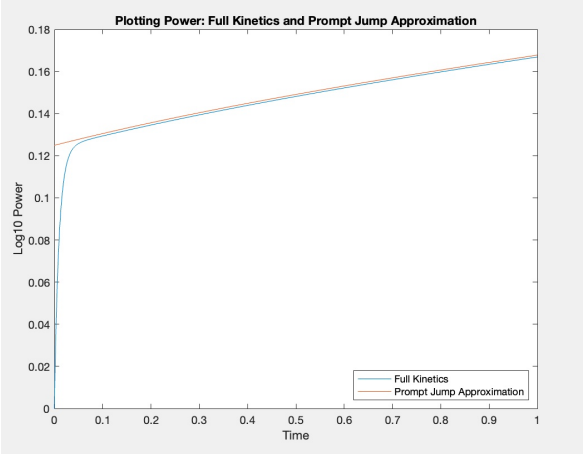
$$Apj = \left(\frac{1}{\beta - \rho}\right) \begin{vmatrix} \beta_1\lambda_1 - \lambda_1(\beta - \rho) & \beta_1\lambda_2 & \beta_1\lambda_3 & \beta_1\lambda_4 & \beta_1\lambda_5 & \beta_1\lambda_6 \\ \beta_2\lambda_1 & \beta_2\lambda_2 - \lambda_2(\beta - \rho) & \beta_2\lambda_3 & \beta_2\lambda_4 & \beta_2\lambda_5 & \beta_2\lambda_6 \\ \beta_3\lambda_1 & \beta_3\lambda_2 & \beta_3\lambda_{3-\lambda_3(\beta-\rho)} & \beta_3\lambda_4 & \beta_3\lambda_5 & \beta_3\lambda_6 \\ \beta_4\lambda_1 & \beta_4\lambda_2 & \beta_4\lambda_3 & \beta_4\lambda_4 - \lambda_4(\beta - \rho) & \beta_4\lambda_5 & \beta_4\lambda_6 \\ \beta_5\lambda_1 & \beta_5\lambda_2 & \beta_5\lambda_3 & \beta_5\lambda_4 & \beta_5\lambda_5 - \lambda_5(\beta - \rho) & \beta_5\lambda_6 \\ \beta_6\lambda_1 & \beta_6\lambda_2 & \beta_6\lambda_3 & \beta_6\lambda_4 & \beta_6\lambda_5 & \beta_6\lambda_6 - \lambda_6(\beta - \rho) \end{vmatrix}$$

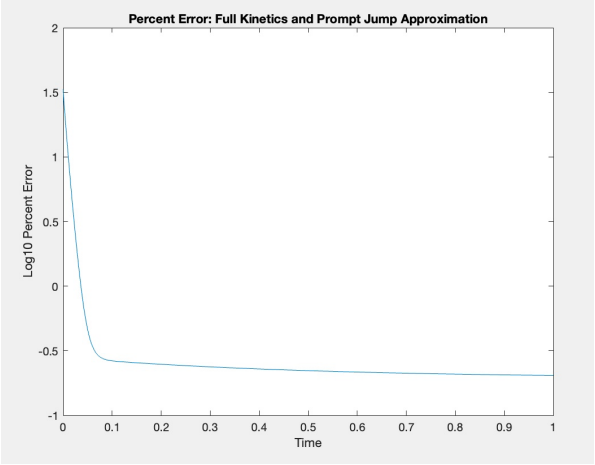
As before, assuming the reactivity is constant, this equation has a solution:

$$C(t) = e^{\mathrm{Apjt}}C(0) + e^{\mathrm{Apjt}}\int_0^t e^{-Apjt'}\,\mathrm{B}\,\mathrm{d}t'$$

Note: The shutdown equilibrium condition is that $\frac{\mathrm{dC}}{\mathrm{dt}}$ is zero. In that case $C_i = \beta_i N_{S/D}/\Lambda\lambda_i$. Substituting, into $0 = Apj\mathrm{C} + \mathrm{B}$ yields the expected shutdown equilibrium equation: $N_{S/D} = -\frac{S\Lambda}{\rho}$. From a computational perspective this is superior to the full seven-dimensional system. The eigenvalues of the P matrix do not exhibit the wide variation found in the full kinetics equation solution. For a critical reactor the full kinetics eigenvalues range from zero to -129.4 sec^{-1}. For this case, the values range from zero to -2.9 sec^{-1}.

Comparison of the full kinetics solution with the prompt jump approximation solution. Reactivity = 0.25β for 1 sec.





Comparison of Full Kinetics solution and Prompt Jump Approximation following a step insertion of Reactivity.

Reactivity (fraction of β)	Percent Error 1 Second	Percent Error 20 Seconds
0.01	0.00435691	0.00438015
0.05	0.0238797	0.0256779
0.1	0.0538971	0.0634204
0.2	0.140564	0.20239
0.3	0.285914	0.519591
0.4	0.544648	1.28996
0.5	1.04626	3.33663
0.6	2.15068	9.63026
0.7	5.13245	35.6158
0.8	16.8672	298.031

As expected, the prompt jump assumption solution falls apart as the reactivity approaches the value of β . In the region less than 40% of β the prompt jump approximation is strong. Real operating reactors usually limit reactivity to less than this value. 40% of β would yield a steady state SUR of about 3 DPM. 20% of β yields about 1 DPM.

The computation approach for the prompt jump assumption cases uses the same method as was used for the full kinetics approach. This takes full advantage of the non-changing value of the system matrix to allow its computation only once. Once again, we define a matrix G which is recursively modified for each time step.

Computation for a Reactivity Step using the Prompt Jump Assumption

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SimplePowerPJ.m
% W.N. Locke
% May 2025
% Step response to a step at time 0. Initial fission rate normalized to 1.
% Example call: SimplePower(0.25,1)
% This will plot the 1 second transient with reactivity equal
% to 0.25*Beta
function SimplePowerPJ(ReactivityFractionOfBeta,TimeInterval)
    So = 2.0e-5; % Source Rate  normalized for a unit initial fission rate.
    Tau = 1.0e-4; % Time step duration in seconds.
    T_hist = 0:Tau:TimeInterval;
    HistoryLength = length(T_hist);
    C_hist = zeros(6,HistoryLength);
    N_hist = zeros(1,HistoryLength);
    N_hist(1) = 1;
    Betas = ...
        [0.00021;0.00141;0.00127;0.00255;0.00074;0.00027];
    BetaTotal = sum(Betas);
    Lambdas = ...
       [0.01246403;0.03052863;0.11141479;0.30130435;1.13606557;3.01304348];
    GenerationTime = 5.0e-5;
    Co = [Betas./(Lambdas*GenerationTime)];
    function APJ = AmatrixPJ(ReactivityFraction)
            APJ = Betas*Lambdas';
            D = diag(Lambdas)*BetaTotal*(1-ReactivityFraction);
            APJ = (APJ - D)/(BetaTotal*(1-ReactivityFraction));
    end
    Apj = AmatrixPJ(ReactivityFractionOfBeta);
    [M, D] = eig(Apj);
    ExpDTau = diag(exp(diag(D)*Tau),0);
    ExpATau = M*ExpDTau/M;   % '/' Right multiplies by the inverse of M.
    rf = 1/(BetaTotal*(1-ReactivityFractionOfBeta));
    B = Betas*So*rf;
    ApjinvB = Apj\B;
    C_hist(:,1) = Co(); % Copies vector Co into the first col of X_hist.
    I = eye(6);  % This is a 6x6 unit matrix.
    G = I;
    for Step = 1:HistoryLength-1
        G = ExpATau*G;
        C_hist(:,Step+1)= G*Co+(G-I)*ApjinvB;
        N_hist(Step+1) = ...
            GenerationTime*rf*(dot(Lambdas,C_hist(:,Step+1))+So);
    end
    figure
    plot(T_hist,log10(N_hist(1,:)),'r')
    title('Log Power vs. Time (sec)');
    xlabel("Time Isec)")
    figure
    plot(T_hist,log10(C_hist(1:6,:)./Co(:)),'r')
    title('Log Normalized Precursor Concentrations vs. Time (sec)');
    xlabel("Time (sec)")
end
```

Ramp Reactivity Additions

So far, our study of reactor kinetics has assumed a linear time-invariant system, allowing us to use the integrating factor in solving the kinetics equations. We treated A as constant, moving it in and out of derivatives or integrals. However, when reactivity changes, A varies over time. This complicates our approach. We can still write the point reactor kinetics equations, but now A and other variables depend on both time and the solution X or C.

$$\frac{\mathrm{dX}^{7x1}}{\mathrm{dt}} = \mathrm{A}^{7x7}\mathrm{X}^{7x1} + S^{7x1} \qquad \frac{\mathrm{dC}^{6x1}}{\mathrm{dt}} = Apj^{6x6}\mathrm{C}^{6x1} + \mathrm{B}^{6x1}$$

Prior to the days when computer and computational capabilities became ubiquitous people put significant effort into solving the case where $\rho(t)$ was a simple linear function of time. The solutions involved esoteric tabulated functions and brought little practical help.

For example, one case is a ramp with only one delayed neutron group being considered. Further the reactivity ramp rate,γ, is constrained to be exactly $\lambda\beta$. This yields the following.

$$\frac{n(t)}{n_0} = \frac{\beta^2}{\gamma\Lambda}\left\{e^{-\lambda t} - \frac{\beta-\gamma\Lambda-\gamma\mathrm{t}}{\beta}\exp\left(\frac{\gamma}{2\Lambda}t^2 - \frac{\beta}{\Lambda}t\right)X\left[1 + \beta\left(\frac{\pi}{2\gamma\Lambda}\right)^{1/2}\exp\left(\frac{(\beta-\lambda\Lambda)^2}{2\gamma\Lambda}\right)\left(\mathrm{erf}\left(\frac{\beta-\gamma\Lambda}{\sqrt{2\gamma\Lambda}}\right) - \mathrm{erf}\left(\frac{\beta-\gamma\Lambda-\gamma\mathrm{t}}{\sqrt{2\gamma\Lambda}}\right)\right)\right]\right\}$$

Here the erf function is defined as:

$$\text{erf}(x) = \int_0^x e^{-\xi^2} d\xi$$

Reference: Dynamics of Nuclear Reactors, David L. Hetrick, University of Chicago Press, 1971.

This tells the story. We are given the simplest of changing reactivities and we are forced into depilating assumptions which will make our numbers too far in error to be of value. One can certainly not place this limit on the reactivity addition rate. It is also clear that the vast manipulation needed to arrive at this result will not enhance our understanding of the physics present. We also understand that computation with the one delayed group model is just wrong.

The solution to this dilemma is to turn to numerical approximation in solving the reactor kinetics equations. The full 7x7 formulation of the problem is difficult because of the wide variations of the eigenvalues of the A matrix. We will use the prompt jump approximation 6x6 representation of the system. In so doing, we require the reactivity to be maintained well below β.

In this section we introduce the "startup rate". This is the number of decades of fission rate change per minute (DPM). This quantity will be discussed at length in the next section. We define it as:

$$SUR = 26.06(dpm - \text{sec})\frac{dP/dt}{P}$$

We will now look at the entire computation associated with this transient using numerical methods with MATLAB. One could use either Python or Julia with similar effort. The rod motion starts at "StartTime", and has a duration, "PullInterval". The rod speed is defined as follows:

$$RodSpeed = \frac{FinalReactivityFractionfraction\ of\ Beta}{PullInterval}$$

With this the reactivity can be found as follows:

```
function Rho = Reactivity(t)
        if t < StartTime
            Rho = 0.0;
        elseif t < PullInterval+StartTime
            Rho = RodSpeed*(t-StartTime);
        else
            Rho = RodSpeed*PullInterval;
        end
end
```

We will find a history of the precursor concentrations and from this we can compute fission rate, startup rate, and the effective precursor decay constant. The effective decay constant is a weight average time constant defined as follows:

$$\lambda_{eff}(t) = \frac{\sum_{i=1}^{6} \lambda_i C_i(t)}{\sum_{i=1}^{6} C_i\,(t)}$$

The prompt jump approximation Apj matrix is developed as follows.

```
function APJ = AmatrixPJ(ReactivityFraction)
        APJ = Betas*Lambdas';
        D = diag(Lambdas)*BetaTotal*(1-ReactivityFraction);
        APJ = (APJ - D)/(BetaTotal*(1-ReactivityFraction));
End
```

"diag" creates a diagonal matrix from the Lambdas vector. Betas*Lambdas$^{\text{Transpose}}$ creates the outer product of these two vectors resulting in a matrix $\beta_i\lambda_j$, "i" is the row and "j" is the column. The " ' " causes a transpose.

The B(t) vector is formed by the following:

```
function Bvector = B(Rho)
```

```
        Bvector = Betas*So/(BetaTotal*(1-Rho));
end
```

The source rate is $S_0 = 1.0e-8$.

With these devices the ode target function is:

```
function dCdt = TargetFunction(t,C)
        Rho = Reactivity(t);
        A = AmatrixPJ(Rho);
        dCdt = A*C+B(Rho);
end
```

The initial state of the precursors vector is Co

```
Co = Po*Betas./(Lambdas*GenerationTime);
```

Given this the precursor differential equations are solved in three lines:

```
span = [0,TotalTime];
opts = odeset(RelTol=1e-9,AbsTol=1e-10);
[T_hist,C_hist] = ode23t(@TargetFunction,span,Co,opts);
```

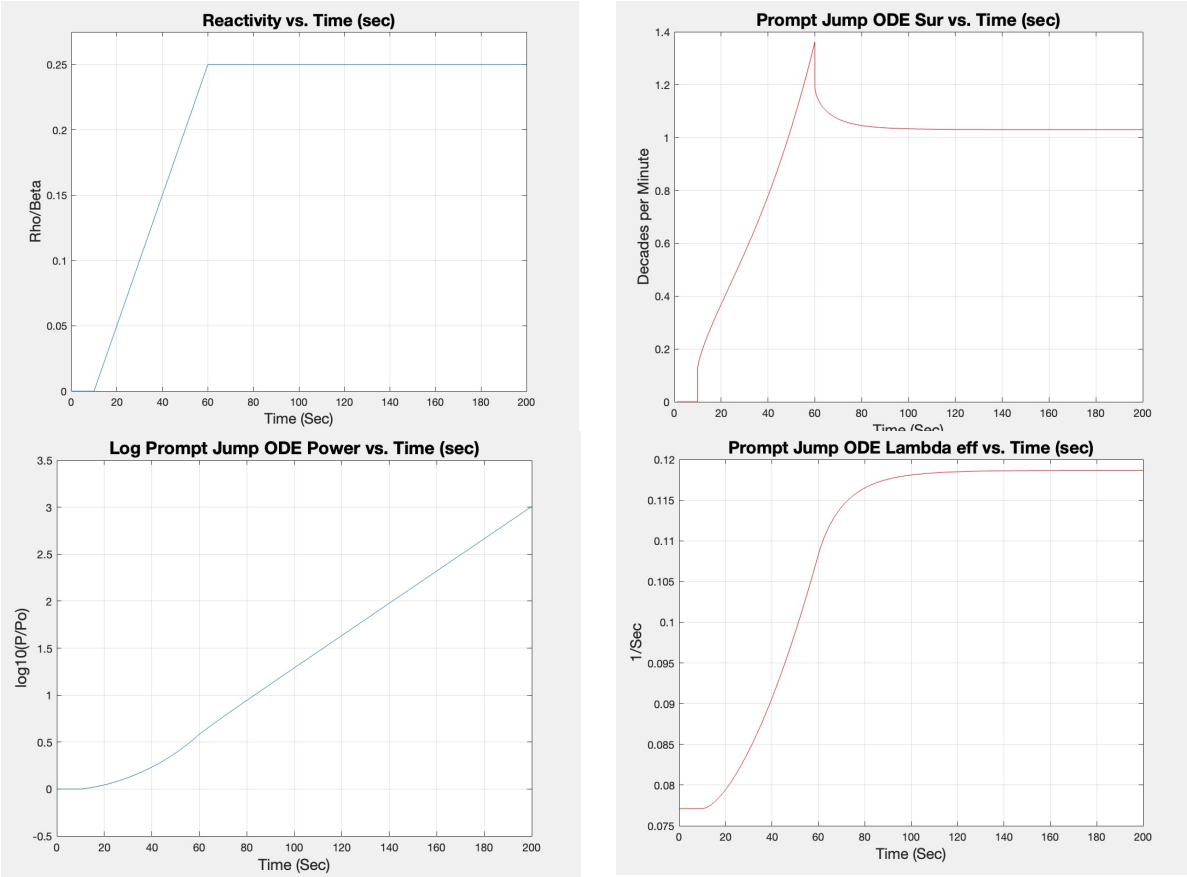
Once the history of precursor concentrations has been developed the fission rate, λ_{eff}, and startup rate are computed directly.

```
for k=1:Num
     Rho = Reactivity(T_hist(k));
     C = C_hist(k,:);
     P_hist(t) = GenerationTime*dot(Lambdas,C)/(BetaTotal*(1-Rho));
     LambdaEff_hist(k) = dot(C,Lambdas)/sum(C);
end
% This implements SUR = 26.06 dpm-sec (dP/dt)/P
SUR_hist = 26.06*diff(P_hist)./(diff(T_hist).*P_hist(2:end));
```

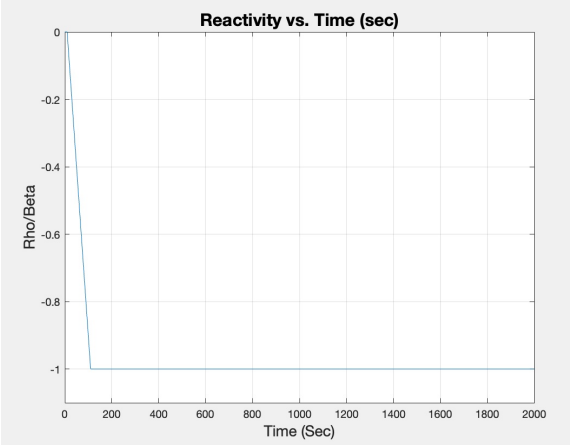
Note that the startup rate history is being computed as a vector calculation. The diff() function takes the difference between adjacent values. And the "./" operator causes each element of the numerator

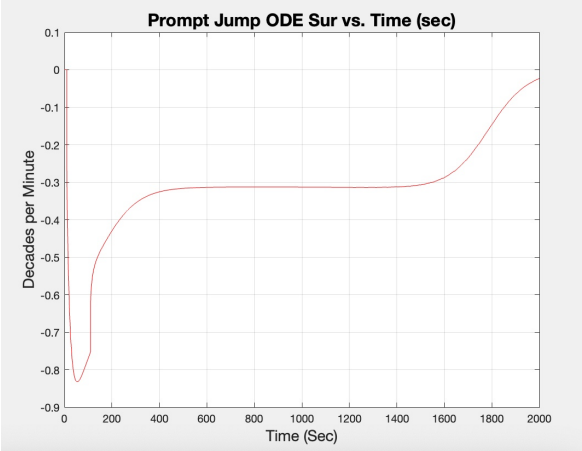
vector to be divided by each element of the denominator vector. Likewise, the ".*" operator indicates element by element multiplication. The diff function produces a vector of length one less than the length of its argument.

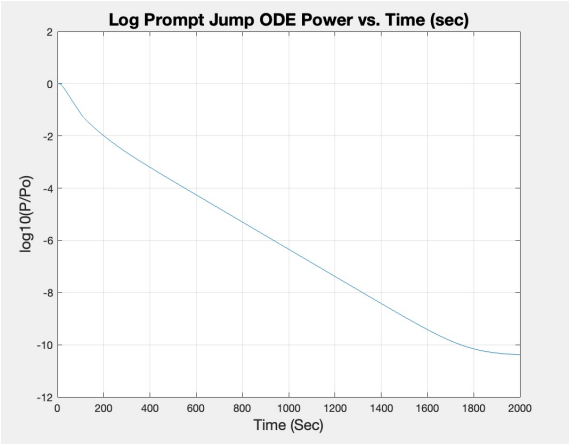
Rho = 0.25, Initial Fission rate = 1.0

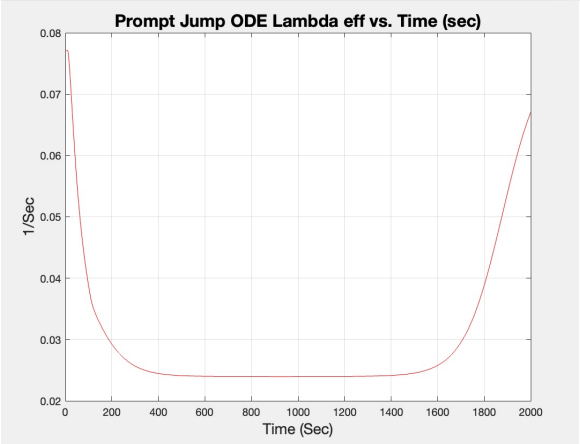


Rho = -1, Initial Fission rate = 1.0

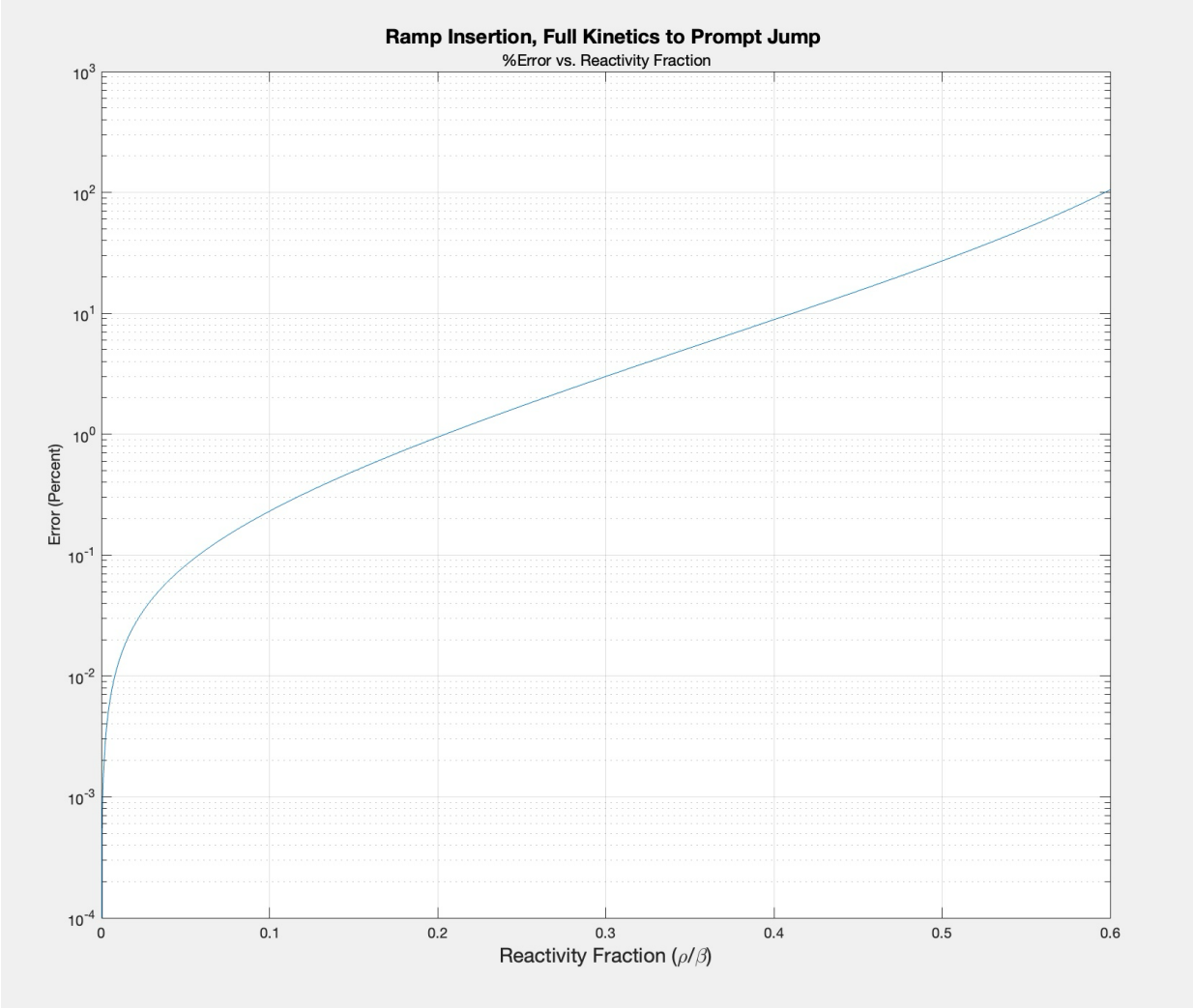








We have used the prompt jump assumption for these calculations. The following graph demonstrates the errors that result from this approach compared to a full kinetics calculation. The plot shows the percent error in the final fission rate reached after a 200 second transient using a ten second start time, a ten second rod pull, followed by a 180 second wait. The reactivity at each point is what remained after each rod pull. A total of 1000 transients were used to create this plot.



Full MATLAB Program for Computing a Ramp Reactivity Insertion

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ReactivityRampODEPJ.m
% W.N. Locke
% April 17, 2025
% Step response to a step at time 0.
% Example call: RampFission RateSUR(
%   FractionOfBeta, ...
%   Start Time,...
%   PullInterval,...
%   TotalTime
%   );
%
% The first will plot the 100 second transient with reactivity equal
% to 0.25*Beta. The second will plot a shutdown
%
function ReactivityRampODEPJ()

    TReactivityRampODEPJ(0.25,10,50,200)
   % TReactivityRampODEPJ(-5,10,100,2000)
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Global Generation Time
function TReactivityRampODEPJ(...
    FinalReactivityFractionOfBeta,...
    StartTime,...
    PullInterval,...
    TotalTime ...
    )
    Lambdas = ...

[0.01246403;0.03052863;0.11141479;0.30130435;1.13606557;3.01304348];
    Betas =[0.00021;0.00141;0.00127;0.00255;0.00074;0.00027];
    BetaTotal = sum(Betas);
    GenerationTime = 5.0e-5;
    So = 1e-8;

    RodSpeed = FinalReactivityFractionOfBeta/PullInterval;

    no = 1;

    function Rho = Reactivity(t)
        if t < StartTime
            Rho = 0.0;
        elseif t < PullInterval+StartTime
            Rho = RodSpeed*(t-StartTime);
        else
```

```
            Rho = RodSpeed*PullInterval;
        end
    end

    function Bvector = B(Rho)
        Bvector = Betas*So/(BetaTotal*(1-Rho));
    end

    function APJ = AmatrixPJ(ReactivityFraction)
        APJ = Betas*Lambdas';
        D = diag(Lambdas)*BetaTotal*(1-ReactivityFraction);
        APJ = (APJ - D)/(BetaTotal*(1-ReactivityFraction));
    end
    function n = FissionRate(C,Rho)
        n = GenerationTime*dot(Lambdas,C)/(BetaTotal*(1-Rho));
    end
    function dCdt = TargetFunction(t,C)
        Rho = Reactivity(t);
        A = AmatrixPJ(Rho);
        dCdt = A*C+B(Rho);
    end

     Co = no*Betas./(Lambdas*GenerationTime);
     span = [0,TotalTime];
     opts = odeset(RelTol=1e-9,AbsTol=1e-10);
     [T_hist,C_hist] = ode23t(@TargetFunction,span,Co,opts);
     Num = length(T_hist);
     LambdaEff_hist=zeros(1,Num);
     n_hist = zeros(Num,1);
     Rho_hist= zeros(Num,1);
     for k=1:Num
         C = C_hist(k,:);
         LambdaEff_hist(k) = dot(C,Lambdas)/sum(C);
         Rho_hist(k) = Reactivity(T_hist(k));
         n_hist(k) = FissionRate(C,Rho_hist(k));
     end
     SUR_hist = 26.06*diff(n_hist)./(diff(T_hist).*n_hist(2:end));

PlotKineticsData(Num,T_hist,n_hist,LambdaEff_hist,SUR_hist,Rho_hist)
end

function PlotKineticsData(...
    Num,T_hist,...
    n_hist,...
    LambdaEff_hist,...
    SUR_hist,...
    Rho_hist...
    )
        figure;
        plot(T_hist,Rho_hist);
        title('Reactivity vs. Time (sec)','FontSize',16);
        ylim([1.1*min(Rho_hist),1.1*max(Rho_hist)])
```

```
        grid on;
        xlabel("Time (Sec)",'FontSize',14);
        ylabel("Rho/Beta",'FontSize',14);
        figure;
        plot(T_hist,log10(n_hist));
        title(...
          'Log Prompt Jump ODE Fission Rate vs. Time
(sec)','FontSize'...
          ,16);
        grid on;
        xlabel("Time (Sec)",'FontSize',14);
        ylabel("log10(P/Po)",'FontSize',14);
        figure
        plot(T_hist,LambdaEff_hist,'r');
        title('Prompt Jump ODE Lambda eff vs. Time
(sec)','FontSize',16);
        xlabel("Time (Sec)",'FontSize',14)
        ylabel("1/Sec",'FontSize',14)
        grid on;
        figure
        plot(T_hist(2:end),SUR_hist,'r');
        title('Prompt Jump ODE Sur vs. Time (sec)','FontSize',16);
        xlabel("Time (Sec)",'FontSize',14)
        ylabel("Decades per Minute",'FontSize',14)
        grid on;
        fprintf("Number of Iterations: %d\n",Num)
        fprintf("Final ODE Lambda: %f\n",LambdaEff_hist(Num))
        fprintf("Final ODE SUR: %f\n",SUR_hist(end))
        fprintf("Final Log10 ODE Fission Rate: %g\n",log10(n_hist(end)))
end
```

The concept of startup rate and the related equations

Early in reactor development it became evident that both the protection equipment and operators need information related to the rate at which fission rate changes. In simple form the kinetics equations have exponential solutions. This led people to think of a "reactor period", the time it takes for fission rate to change by a factor of "e". This measure can however be confusing because a steady state period is infinite. The next step is to consider an inverse period $1/\tau$, so fission rate is changing as $n(t) = n(0)e^{t/\tau}$. And we could display $1/\tau$. Beyond this however, the industry generally objected to being asked to think in fission rates of "e". The choice was to change the equation to a base ten and to convert the resulting rate expression to units of per minute rather than per second. τ itself is in units of seconds so the conversion is as follows $n(t) = n(0)10^{\int_0^t SUR(t')_{[Decades/min]}dt'^{[min]}}$. Using this idea $SUR = log_{10}(e) * \frac{60\left[\frac{sec}{min}\right]}{\tau^{[sec]}} = 26.06$ [DPM-sec]/ $\tau^{[sec]}$. Further, using a simple idea related to a decay equation the period may be defined as $\tau \equiv \frac{n(t)}{\dot{n}(t)}$ With this definition the

$$SUR = 26.06\ [DPM - sec] * \frac{\dot{n}(t)}{n(t)}.$$

Alternately, this is also used in the following form within protection and control equipment:

$$SUR = 26.06\ [DPM - sec]\ \frac{d}{dt}(\ln n(t))$$

The equation $SUR = 26.06\ [DPM - sec]\frac{\dot{n}(t)}{n(t)}$ may be directly used with a stream of digital data representing n(t). For example, in simplest form this could be: $SUR = 26.06\ [DPM - sec](\frac{n(t+\Delta t)-n(t)}{\Delta t * n(t+\Delta t)})$. With real plant data this method could present problems due to electrical noise. The best approach would be to apply digital filtering to the samples prior

to computing the difference. Digital sampling theory could also be used to combine the difference with the filtering.

The SUR equation

We proceed by forming an expression for

$$SUR = 26.06\ [DPM - sec] * \frac{\dot{n}(t)}{n(t)}$$

This SUR equation is a tool used throughout the nuclear industry as a training aid. The equation is usually derived making the following assumptions:

- Point Kinetics is adequate to represent the reactor.
- The prompt jump assumption is used $\Lambda \frac{dN}{dt} \approx 0$ and the reactivity is significantly less than β.
- The delayed neutron precursors are placed into a single group with one effective decay constant.
- Sources denoted by S are constant in time.

The first two assumptions are acceptable from the standpoint of normal operations. The third is not, it leads to significant error in the numbers computed by the resulting equation.

Define: $\lambda_{eff} = \sum_{i=1}^{6} \lambda_i C_i(t) \,/\, \sum_{i=1}^{6} C_i(t)$. We plotted this value in our ramp transient example plots above. Also recall that $\beta = \sum_{i=1}^{6} \beta_i$.

We will proceed using the first two assumptions and the one delayed group assumption. However, we will consider that the group decay effective λ_{eff} has a time derivative. This leads to an interesting correction to the SUR equation.

The original point kinetics equations are as follows:

$$\frac{dn(t)}{dt} = \frac{n(t)(\rho - \beta)}{\Lambda} + \sum_{i=1}^{6} \lambda_i C_i(t) + S$$

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

The one delayed group approximations and prompt jump assumptions convert these as follows using the dot notation for the time derivative.

$$n(t) = \frac{\lambda_{eff} C(t)\Lambda + S\Lambda}{\beta - \rho} \qquad C(t) = \sum_{i=1}^{6} C_i(t)$$

$$\dot{C}(t) = \frac{n(t)\beta}{\Lambda} - \lambda_{eff} C(t) \qquad \lambda_{eff} C(t) = \sum_{i=1}^{6} \lambda_i C_i(t)$$

Rearrange the first of these by clearing the denominator to the left-hand side and perform an implicit derivative, we obtain.

$$\dot{n}(t)(\beta - \rho) - n(t)\dot{\rho} = \dot{\lambda_{eff}} C(t)\Lambda + \lambda_{eff} \dot{C(t)}\Lambda$$

Now observe the following rearrangements of the above equations

$$\lambda_{eff} C(t)\Lambda = n(t)(\beta - \rho) - S\Lambda$$

$$\dot{C(t)}\Lambda = n(t)\beta - \lambda_{eff} C(t)\Lambda = n(t)\rho + S\Lambda$$

Substituting these equations in the implicit derivative we have the following

$$\dot{n}(t)(\beta-\rho)-\ n(t)\dot{\rho} = \frac{\dot{\lambda_{eff}}}{\lambda_{eff}}[n(t)(\beta-\rho)-\ S\Lambda]+\lambda_{eff}(n(t)\rho+\ S\Lambda)$$

So, we have an expression for $\dot{n}(t)/\ n(t)$ which is $1/\tau$.

$$\frac{\dot{n}(t)}{n(t)}=\frac{\dot{\rho}+\frac{\dot{\lambda_{eff}}}{\lambda_{eff}}[(\beta-\rho)-\frac{S\Lambda}{n(t)}]+\lambda_{eff}(\rho+\frac{S\Lambda}{n(t)})}{(\beta-\rho)}$$

Case	SUR Equation
$S \neq 0$ $\dot{\lambda_{eff}} \neq 0$	$SUR = 26.06\ [dpm-sec]\frac{\dot{\rho}+\frac{\dot{\lambda_{eff}}}{\lambda_{eff}}[(\beta-\rho)-\frac{S\Lambda}{n(t)}]+\lambda_{eff}(\rho+\frac{S\Lambda}{n(t)})}{(\beta-\rho)}$
$S = 0$ $\dot{\lambda_{eff}} \neq 0$	$SUR = 26.06\ [dpm-sec]\left[\frac{\dot{\rho}+\lambda_{eff}\rho}{(\beta-\rho)}+\frac{\dot{\lambda_{eff}}}{\lambda_{eff}}\right]$
$S \neq 0$ $\dot{\lambda_{eff}} = 0$	$SUR = 26.06\ [dpm-sec]\frac{\dot{\rho}+\lambda_{eff}(\rho+\frac{S\Lambda}{n(t)})}{(\beta-\rho)}$
$S = 0$ $\dot{\lambda_{eff}} = 0$	$SUR = 26.06\ [dpm-sec]\frac{\dot{\rho}+\lambda_{eff}\rho}{(\beta-\rho)}$

The last of these is what is commonly known as the startup rate equation.

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

The results of this equation may be effectively corrected with the adjustment provided by adding $\frac{\dot{\lambda_{eff}}}{\lambda_{eff}}$ so long as the reactivity is much less than β.

Aside – SUR for a Sudden Insertion of Reactivity

If we do not set $\Lambda \frac{dn(t)}{dt} = 0$

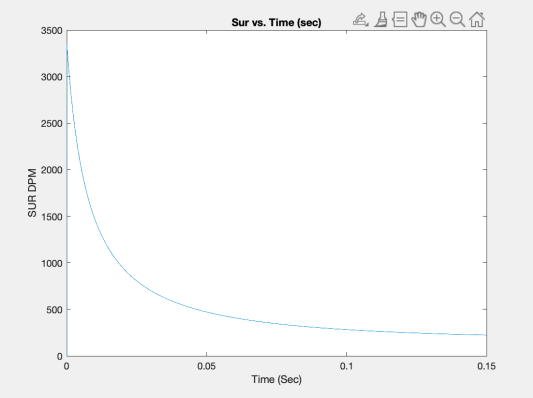
$$\frac{dn(t)}{dt} = \frac{n(t) * (\rho - \beta)}{\Lambda} + \sum_{i=1}^{6} \lambda_i C_i(t)$$

However, we can easily compute the initial SUR of a prompt critical reactor.

Consider a steady state reactor such that $\lambda_{eff} C(0) = \beta n(0)/\Lambda$. Now assume we step a reactivity just equal to β. Prior to the precursor concentrations changing the first reactor kinetics becomes:

$\frac{dn(t)}{dt} = \rho\, n(0)/\Lambda$ which yields a $SUR = 26.06\, \frac{dn(t)}{dt}/n(t) = 26.06\, \rho\, /\Lambda$.

If $\Lambda = 50\ microseconds$ this will yield 3336 DPM.
This is a six group full kinetics sur solution for a step of constant reactivity equal to β.



Typical Example Problem using the SUR equation

Using this equation suppose we start from steady state and pull rods for 15 seconds. The final reactivity is 0.25β. Plot the SUR transient and fission rate. Also assume $\beta = 640pcm \; and \; \lambda_{eff} = 0.1\frac{1}{sec}$.

While pulling $\dot{\rho} = \frac{0.25\beta}{15\text{sec}} = (5/300)\beta$

Before Rods Move	SUR=26.06 $[dpm - sec]\frac{0+(0.1\ 1/\text{sec})0}{(1-0)}$	0.000 DPM
Rods Start Moving	SUR=26.06 $[dpm - sec]\frac{5/300\ /sec+(0.1\ 1/\text{sec})0}{(1-0)}$	0.43 DPM
Rods Finish Just before stopping	SUR=26.06 $[dpm - sec]\frac{5/300\ /sec+(0.1\ 1/\text{sec})0.25}{(1-0.25)}$	1.45 DPM
After Rods Stop	SUR=26.06 $[dpm - sec]\frac{0.0\ /sec+(0.1\ 1/\text{sec})0.25}{(1-0.25)}$	0.87 DPM

The following plots demonstrate this calculation compared with a six delayed neutron group calculation. The fission rate is plotted. The kinetics equation results computed, and the SUR equation results are plotted. And the computed $\lambda_{eff} \; and \; \frac{\dot{\lambda}_{eff}}{\lambda_{eff}}$ are plotted. Finally, the reactivity transient is plotted.

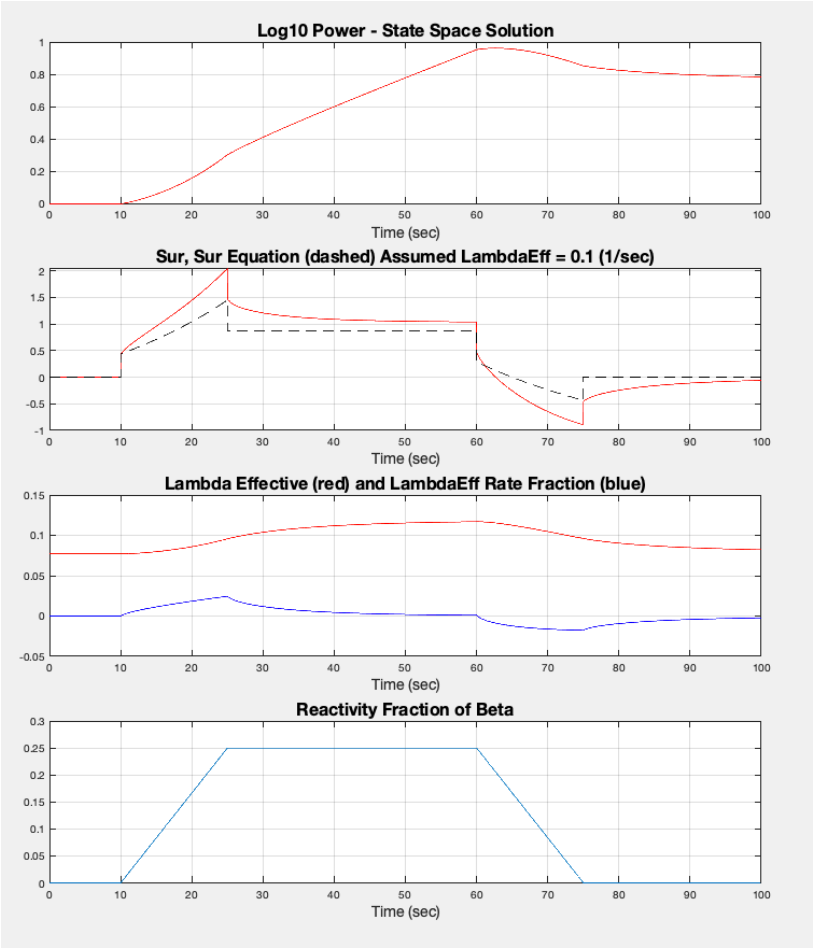
Plotting Power For a Varying Startup Rate

We often plot the $\log10(n(t)/n(0))$ along with the startup rate plots. With a varying SUR(t) the power is given by the following:

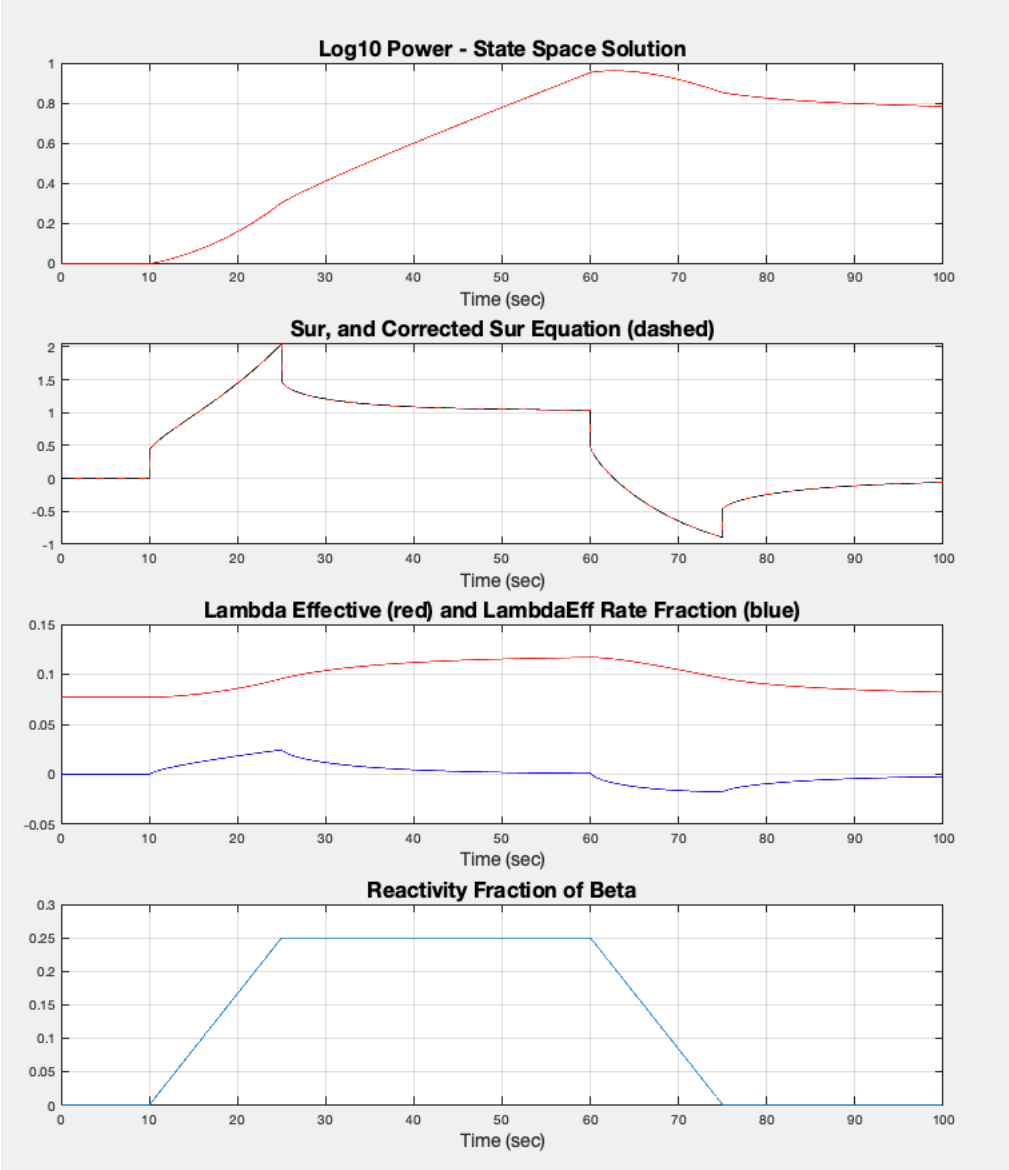
$$n(t) = n(0)10^{\int_0^t SUR(t')_{[\text{Decades/min}]}dt'^{[\text{min}]}}$$

So, when plotting $\log10(n(t)/n(0))$ we are simply plotting $\int_0^t SUR(t')dt'$. This is the area under the startup rate curve. Take care here with the units of time. SUR is given in per minute. So, if we want the left-hand side of this equation in seconds, we will need to make a time conversion as follows.

$$n(t) = n(0)\ 10^{\int_0^t SUR(t')_{[\text{Decades/min}]}} \frac{dt'^{[\text{sec}]}}{60\ [\text{sec/min}]}$$



There are significant differences between the computed SUR and the values coming from the SUR equation. If we replot this using the equation which is corrected by the term $\frac{\dot{\lambda_{eff}}}{\lambda_{eff}}$ we get the following:



Clearly the problem here is that it is not easy to know the value of the ratio $\frac{\dot{\lambda_{eff}}}{\lambda_{eff}}$ unless we are doing a full six group calculation as we are here. Therefore, people use the simplified equation to get an idea of the nature of the transient even while the numbers are incorrect.

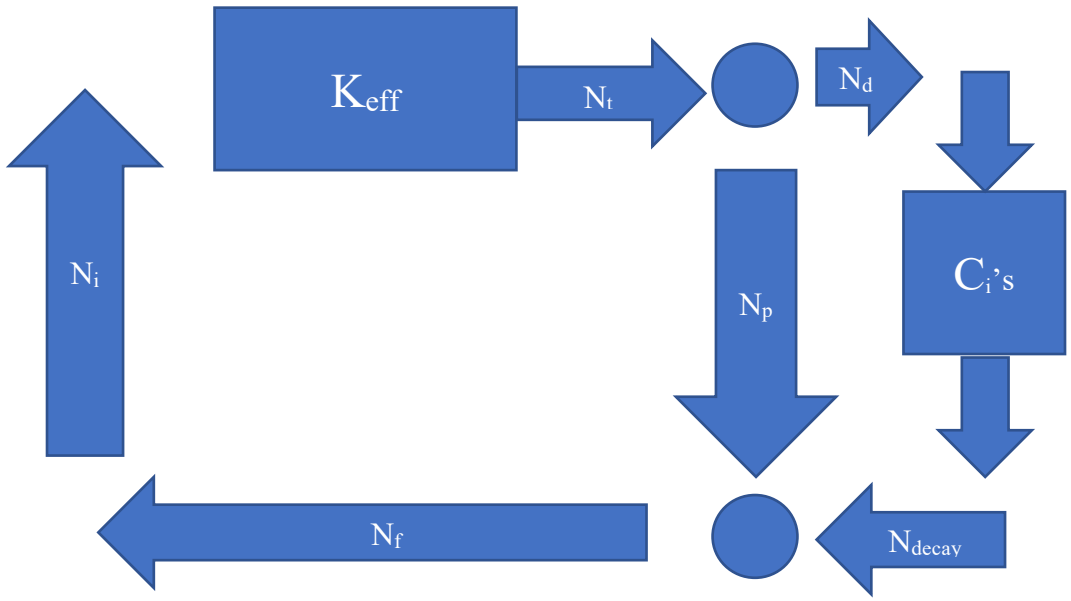
Power Turning

Examine the second part of the transient shown in the previous two figures. The fission rate turns (the SUR is zero) well before the reactivity is back to zero. The numerator of the SUR equation is $\dot{\rho} + \lambda_{eff}\rho$. The fission rate turns when this sum is zero. The negative $\dot{\rho}$ is forcing this to happen.

There are two contributions to the rate of change of fission rate, one is the rate of change of the prompt cycle neutrons, the other is the rate of change of the precursor concentrations. Ignoring S and $\dot{\lambda_{eff}}$

$$\dot{n}(t) = \frac{n(t)\dot{\rho}}{(\beta - \rho)} + \frac{\lambda_{eff}\dot{C(t)}\Lambda}{(\beta - \rho)}$$

Because we found $\dot{C(t)}\Lambda = n(t)\rho$, we know that a positive reactivity will always indicate that the precursor concentrations are going up. But the negative $\dot{\rho}$ will mean the prompt cycle is lowering. Fission rate turns when the two effects sum to zero.



How much reactivity is in the core at the time that fission rate turned using the SUR equation?

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

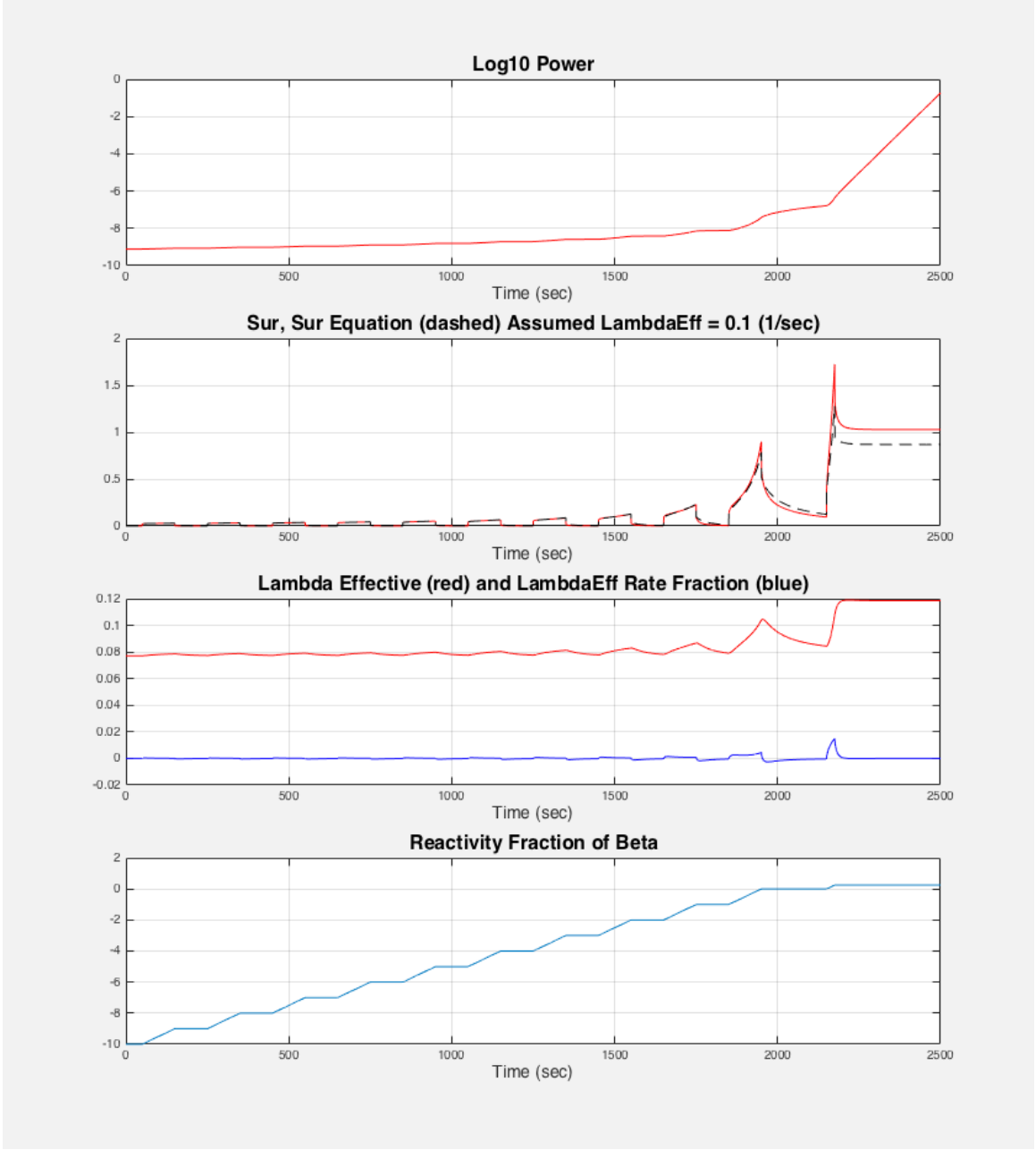
$$\rho = -\frac{\dot{\rho}}{\lambda_{eff}} = -(-\frac{5}{300\ \text{sec}})\beta/(0.1\frac{1}{\text{sec}}) = 1/6\ \beta$$

SUR Equation with Source Neutrons

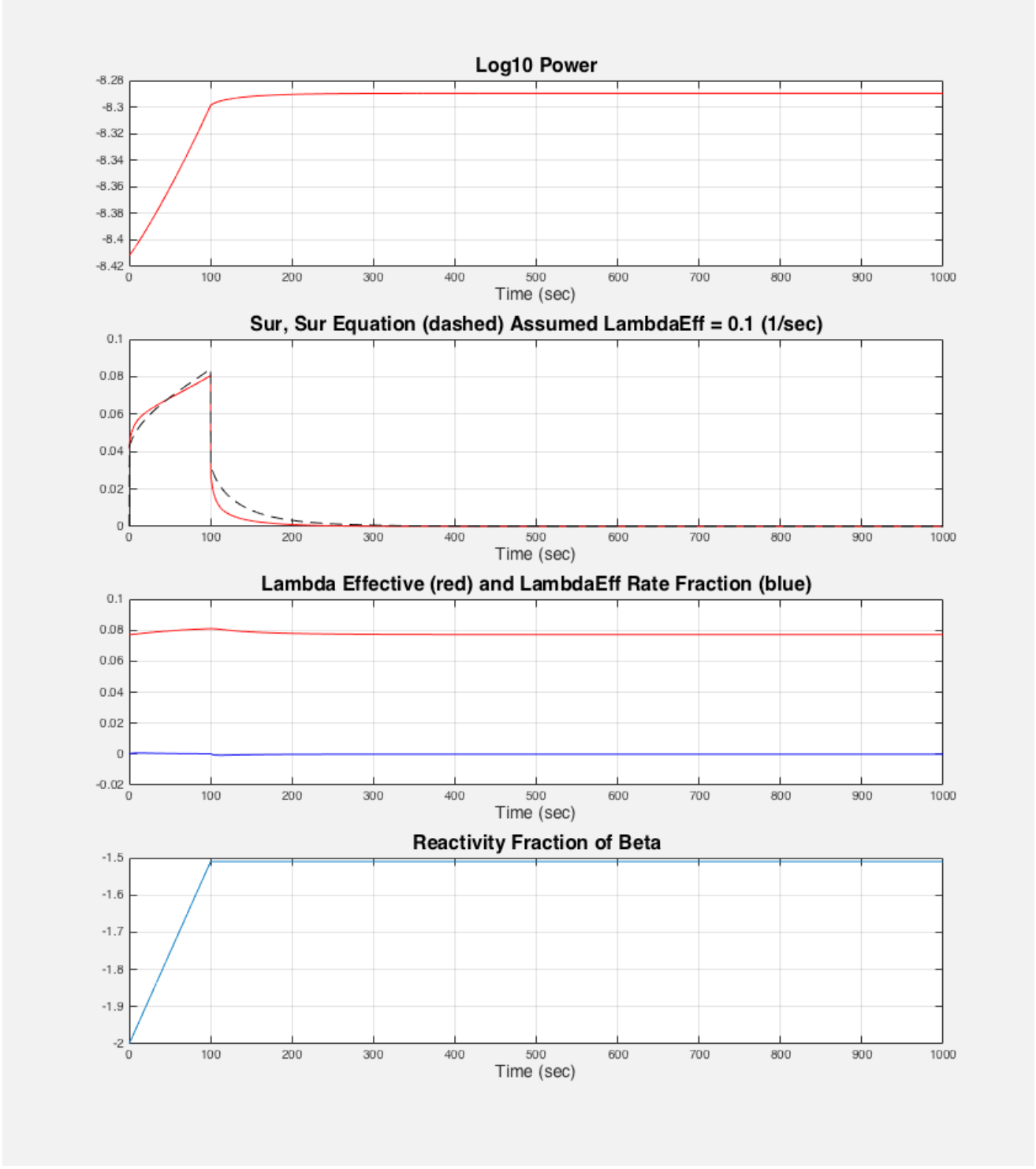
$$SUR = 26.06\ [dpm - sec]\frac{\dot{\rho} + \lambda_{eff}(\rho + \frac{S\Lambda}{n(t)})}{(\beta - \rho)}$$

Notice that this form gives the same equation as we derived for the steady state fission rate in a shutdown reactor with sources present. The steady state fission rate is inversely proportional to reactivity and directly proportional to $S\Lambda$.

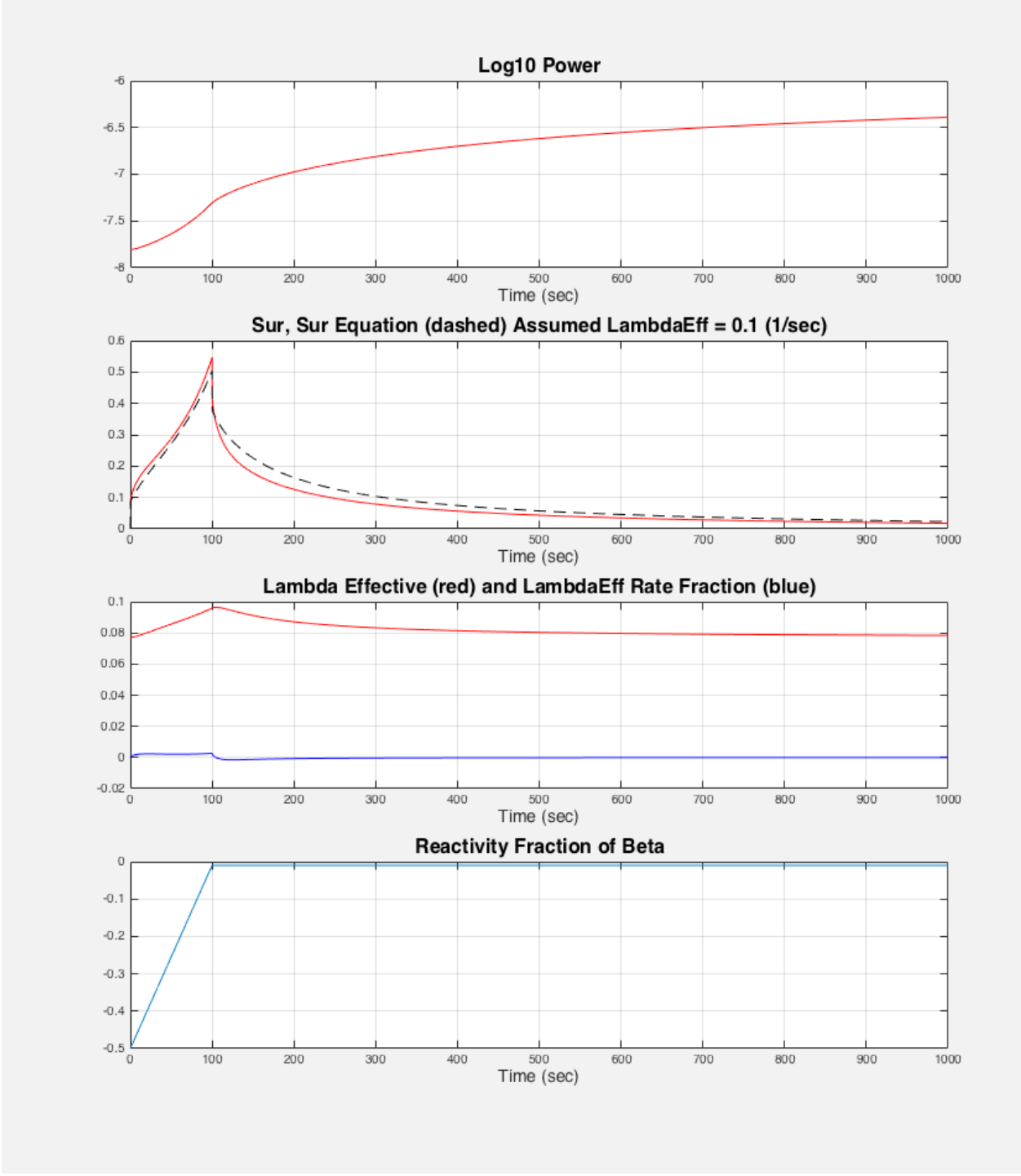
Additional Examples of Kinetics transients with fission rate below the point of adding heat.



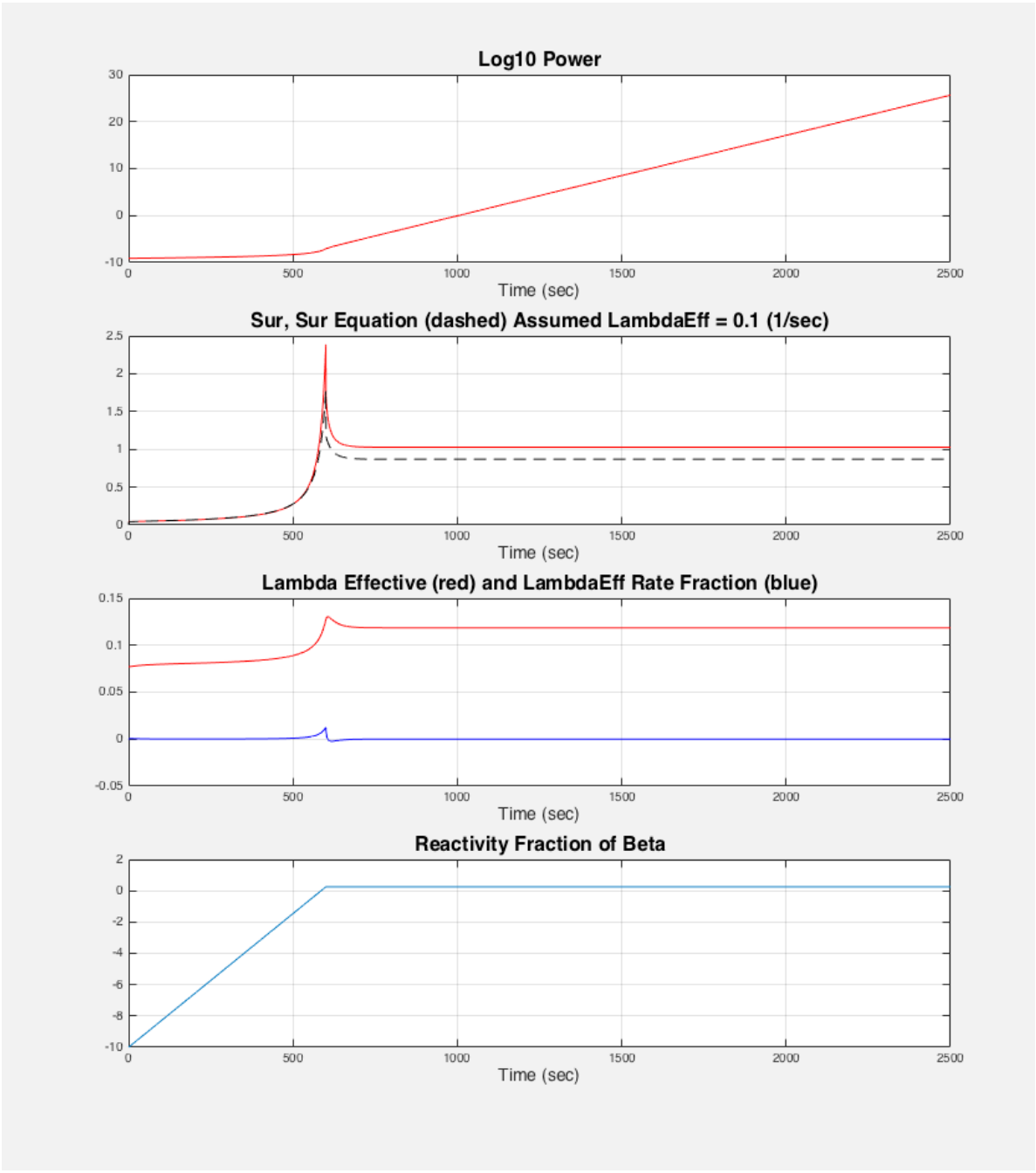
This is a reactor startup as a sequence of rod withdrawals.



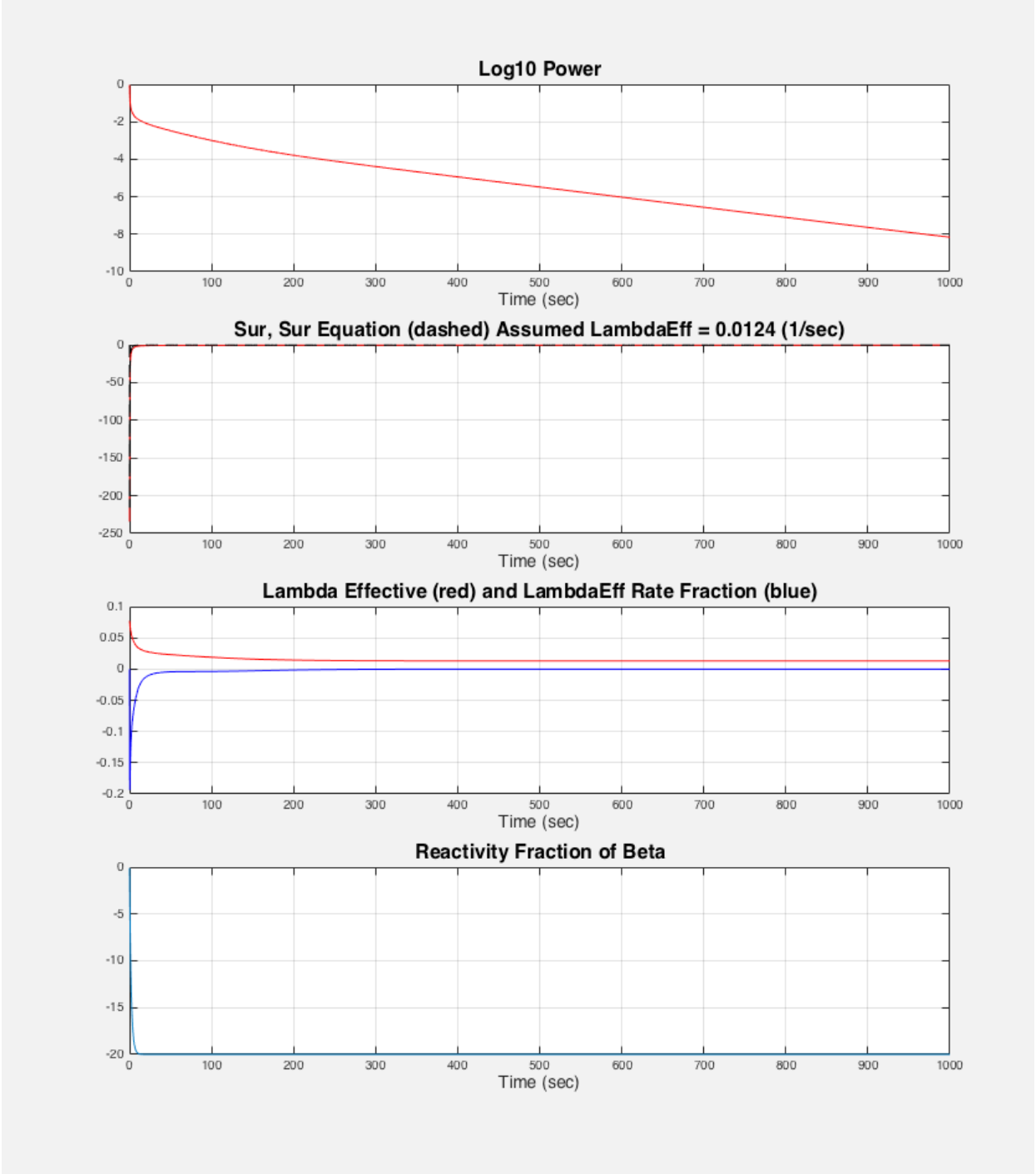
This is a close-up of a rod withdrawal when there remains a great deal of negative reactivity in the plant. In this case the fission rate rises only a small amount, and the SUR is small and quickly damped.



This is a close-up of a rod withdrawal when there remains little negative reactivity in the plant. In this case the fission rate rises much more, and the SUR is larger and slowly damped.



This is a reactor startup performed with a singular rod pull from -10 beta to 0.25 beta. Discuss this transient. Why would it trouble you? The final SUR here is again 1 DPM.



This is a reactor trip from critical. Notice that the log of fission rate becomes a straight line. What is the SUR? Notice the sudden drop in fission rate at time zero: $P_{0+} = P_{0-}\left(\frac{\beta}{\beta-\rho}\right) = \frac{1}{21}P_{0-} = 0.048P_{0-}$ and $\text{Log}_{10}(0.048) = -1.32$

Variation in the Effective Decay Constant

Find λ_{eff} and $C_i(t)$ in terms of the other quantities if fission rate is on a stable period τ, $N(t) = N_0 e^{t/\tau}$. Here the period is taken as 26.06 [dpm-sec]/SUR.

$$\frac{\mathrm{d}C_i}{\mathrm{d}t} = \frac{\beta_i}{\Lambda} n(t) - \lambda_i C_i$$

bring the term involving the C_i to the left-hand side of the equation and multiply both sides of the equation by $e^{\lambda_i t}$. This makes the left-hand side a total derivative:

$$\frac{d}{\mathrm{d}t}(e^{\lambda_i t} C_i) = \frac{\beta_i}{\Lambda} e^{\lambda_i t + t/\tau}$$

Integrating this from (0,t) yields:

$$C_i(t) = C_i(0) e^{-\lambda_i t} + e^{-\lambda_i t} \int_0^t \frac{\beta_i}{\Lambda} e^{\lambda_i t' + t'/\tau} \mathrm{d}t'$$

This becomes:

$$C_i(t) = C_i(0) e^{-\lambda_i t} + \frac{\beta_i}{\Lambda(\lambda_i + 1/\tau)} (e^{t/\tau} - e^{-\lambda_i t})$$

But with a stable positive period, all the $e^{-\lambda_i t}$ terms would be small as compared to $e^{t/\tau}$:

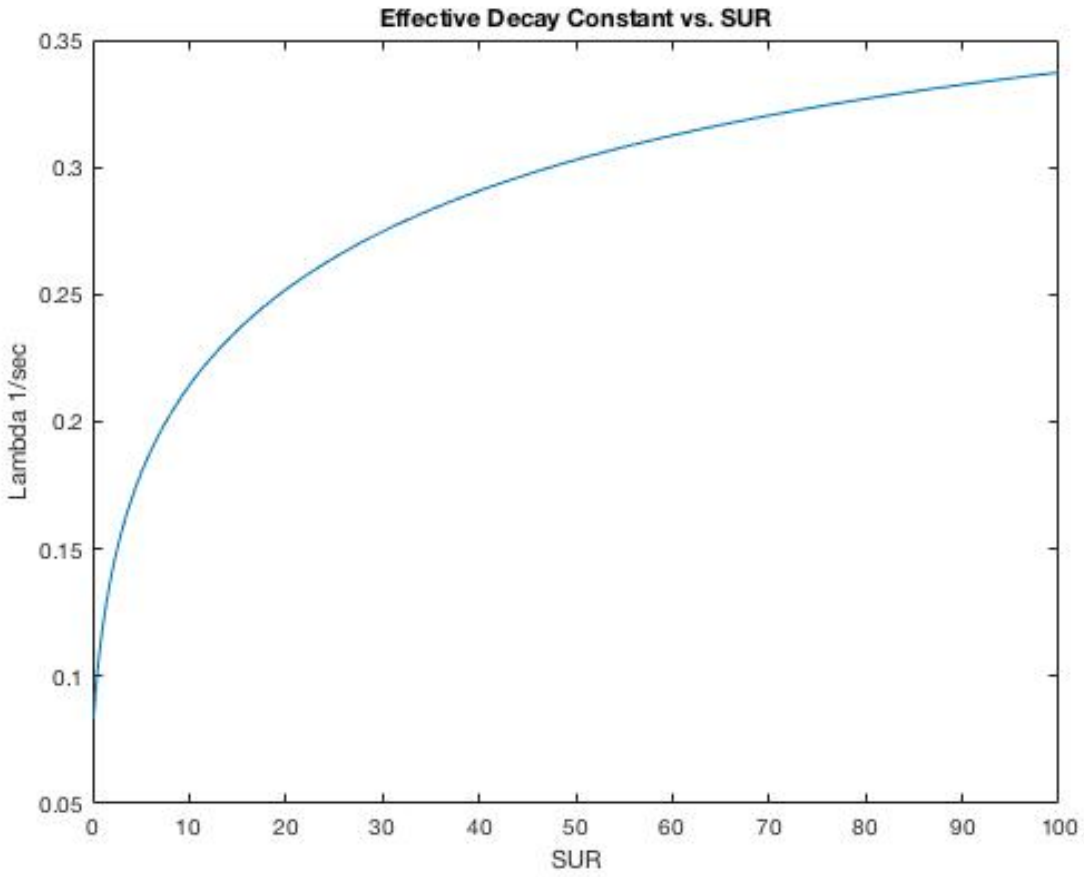
$$C_i(t) = \frac{\beta_i}{\Lambda(\lambda_i + 1/\tau)} e^{t/\tau}$$

Now with

$$\lambda_{\text{eff}} = \frac{\sum_1^6 \lambda_i C_i(t)}{\sum_1^6 C_i(t)}$$

We have the following result for the effective delayed neutron fraction in terms of the stable positive period:

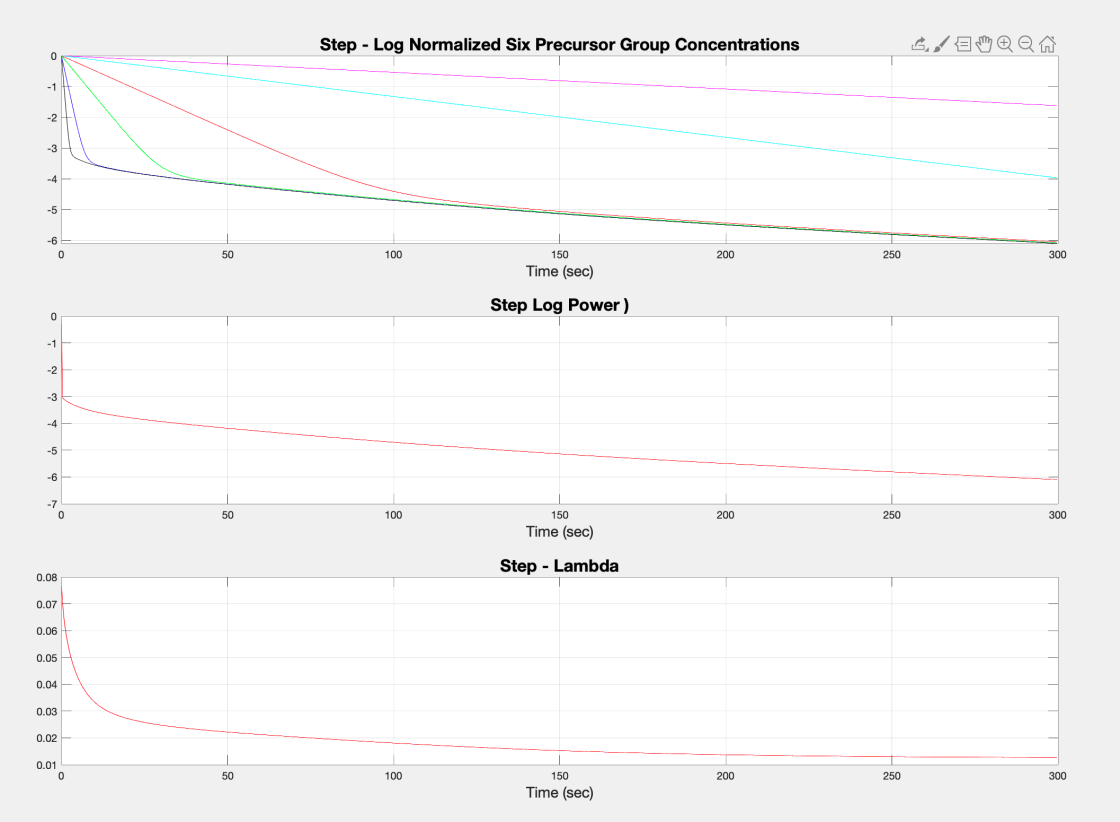
$$\lambda_{\text{eff}} = \frac{\sum_1^6 \lambda_i \frac{\beta_i}{(\lambda_i \tau + 1)}}{\sum_1^6 \frac{\beta_i}{(\lambda_i \tau + 1)})} \qquad \tau = \frac{26.06[dpm - sec]}{SUR}$$



Values Computed Assuming a Stable Period		
SUR (dpm)	**Effective Decay Constant 1/sec**	**Approximate Reactivity as a Fraction of $\overline{\beta}$**
0	0.0771	0
0.1	0.0833	0.03
0.5	0.1016	0.13
1	0.1178	0.22
2	0.1405	0.34
5	0.1795	0.52
10	0.2142	0.65
100	0.3373	0.95
170	0.3578	1.0 (Prompt Critical)

Generation Time Assumed to be 5×10^{-5} sec.

Effective decay constant with negative reactivity following a prompt insertion of reactivity. Insertion of -1000 β.



The final value of the effective decay constant is a function of the stepped in reactivity

Conclusion

While use of an effective decay constant of unchanging value during transients may have some heuristic value in giving people a view of how startup rate behaves, it is misleading and will lead to significantly wrong answers. Kinetics analysis really requires a full solution of the associated coupled differential equations.

Transfer Functions

We looked at transfer functions in our review of Laplace transforms in a previous lesson. Recall that transfer functions only apply in a system which initially is in a zero state, and it is unchanging other than the state vector. This will work for us if we consider a system of constant reactivity but not if the reactivity is changing in time. Here we will look at two cases where we can meet these requirements. And then we will create a system with varying reactivity but approximating the system by making small variations.

Source Transfer Function

Consider a sample of fissile material which contains fuel and moderator but is not large enough to be a critical mass. It will likely have a fission rate due to neutrons from outside sources, but that total fission rate will be assumed to be below anything significant. We now consider the effect of bring a neutron source into proximity with our sample.

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

$$X(t) = \begin{bmatrix} n(t) \\ C_1(t) \\ C_2(t) \\ C_3(t) \\ C_4(t) \\ C_5(t) \\ C_6(t) \end{bmatrix} \quad A = \begin{bmatrix} \frac{\rho - \beta}{\Lambda} & \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 \\ \beta_1/\Lambda & -\lambda_1 & 0 & 0 & 0 & 0 & 0 \\ \beta_2/\Lambda & 0 & -\lambda_2 & 0 & 0 & 0 & 0 \\ \beta_3/\Lambda & 0 & 0 & -\lambda_3 & 0 & 0 & 0 \\ \beta_4/\Lambda & 0 & 0 & 0 & -\lambda_4 & 0 & 0 \\ \beta_5/\Lambda & 0 & 0 & 0 & 0 & -\lambda_5 & 0 \\ \beta_6/\Lambda & 0 & 0 & 0 & 0 & 0 & -\lambda_6 \end{bmatrix} \quad S = \begin{bmatrix} \mathrm{SourceRate} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Take the Laplace transform of this equation and solve for X

$$X = [\,sI - A]^{-1}S$$

Our transfer function is $G = \ [\,sI - A]^{-1}$ and X(0) = 0.

This is all fine and well, but how do we compute this thing?

Let's back up to the original equation and stop before we form the inverse:

$$[sI - A]X = S$$

Now as earlier we form a diagonalized system using the eigenvalues and the modal matrix, D and M.

We know that $D = \ M^{-1}AM, A = \ MDM^{-1}, sI = \ M^{-1}sIM$

$$D = \begin{bmatrix} \mathrm{d1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & d6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & d7 \end{bmatrix}$$

Left multiply by M^{-1} and inject the identity before X vector.

$$M^{-1}[sI - A]MM^{-1}X = M^{-1}S$$

This is

$$[sI - D]M^{-1}X = M^{-1}S$$

And

$$[sI - D] = \begin{bmatrix} \mathrm{s-d1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & s-d2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & s-d3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s-d4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & s-d5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & s-d6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s-d7 \end{bmatrix}$$

The inverse of a diagonal matrix is a matrix which inverts each diagonal element so:

$$[sI - D]^{-1} = \begin{bmatrix} \frac{1}{\mathrm{s-d1}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\mathrm{s-d2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\mathrm{s-d3}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\mathrm{s-d4}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\mathrm{s-d5}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\mathrm{s-d6}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\mathrm{s-d7}} \end{bmatrix}$$

So now looking at:

$$[sI - D]M^{-1}X = M^{-1}S$$

$$M^{-1}X = [sI - D]^{-1}M^{-1}S$$

Now multiply by M

$$\mathrm{X} = \mathrm{M}[sI - D]^{-1} M^{-1} S$$

So, we have done it, our transfer function is:

$$X = G * S \text{ and } G = \mathrm{M}[sI - D]^{-1} M^{-1}$$

This G is easy to calculate with modern tools.
The inverse Laplace transform, $\mathcal{L}^{-1}$, of this is

$$\mathcal{L}^{-1}\,\mathrm{M}[sI - D]^{-1}{}_{M^{-1}}$$

$$= \mathrm{M}\begin{bmatrix} e^{d1t} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & e^{d2t} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e^{d3t} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{d4t} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{d5t} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{d6t} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e^{d7t} \end{bmatrix} M^{-1}$$

This is the time domain impulse response of the system. That is, this is the behavior we would get with short pulse of neutrons with an integral of 1 were to hit our system.

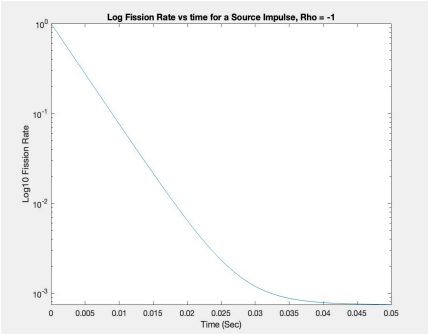
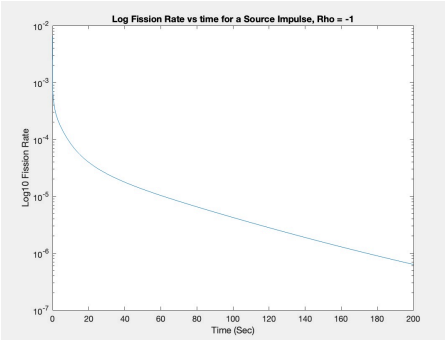
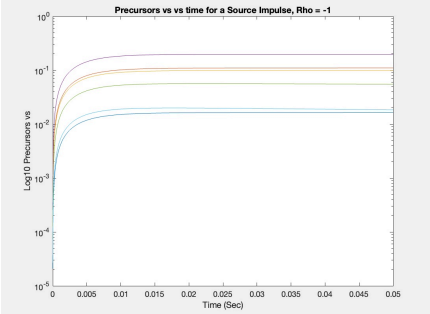
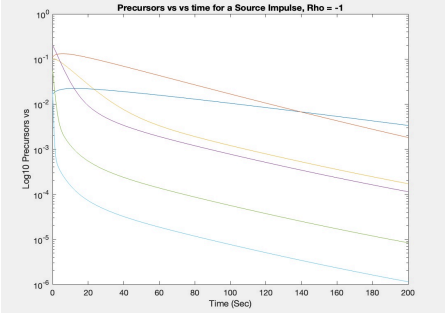
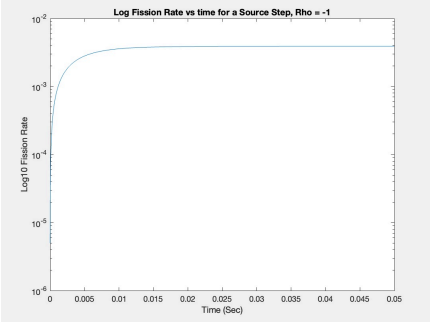
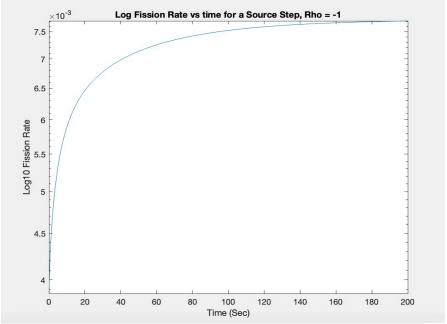
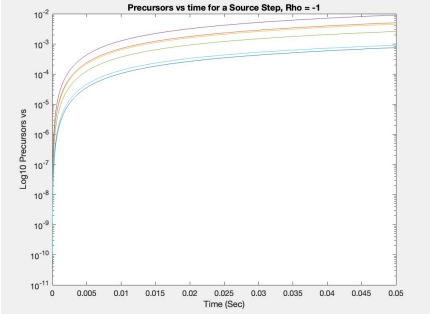
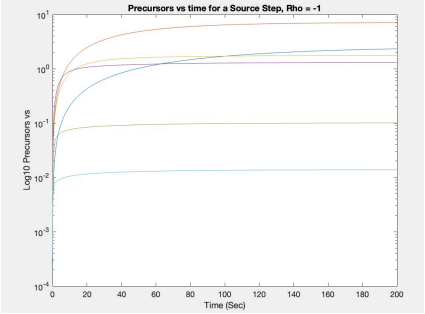
To get the result for a step source we would need to multiply by So/s and transform the result:

$$\mathcal{L}^{-1}\,\mathrm{M}[s(sI - D)]^{-1}{}_{M^{-1}So}$$

$$X(t) = M \begin{bmatrix} \frac{(1-e^{d1t})}{-d1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{(1-e^{d2t})}{-d2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{(1-e^{d3t})}{-d3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-e^{d4t})}{-d4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-e^{d5t})}{-d5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-e^{d6t})}{-d6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{(1-e^{d7t})}{-d7} \end{bmatrix} M^{-1}$$

This is not hard. First find the eigenvalues and eigenvectors of A, form the diagonal matrix show above with the inverted terms for a given value of s. Then multiply from the left by M and right by M inverse.

Multiplication of this result by the source vector will result in only the first column of G being used. This is because only the first element of S is non-zero. The first result will be the fission rate, the following six elements will represent the precursor concentrations.

	Short Term (0.05 Seconds)	Long Term (200 Seconds)
Impulse Fission Rate		
Impulse Precursors		
Step Fission Rate		
Step Precursors		

We see a prompt jump behavior in both the impulse and step response curves. This prompt jump is due to the sudden introduction of source neutrons rather than a change in reactivity as we usually discuss. Before the precursors start to respond in either case, the prompt jump will be $\frac{So\Lambda}{\beta-\rho}$. For the step response case the final equilibrium fission rate will be $\frac{-So\Lambda}{\rho}$. In this the log base ten of these two values are 0.0039 and 0.0078. Based on a source rate So = 1.

MATLAB Script for Calculating Impulse or Step Response.

```
function [X,dt] = StepOrImpulseResponce(Rho,So,Tf,Type,PlotPoints)
    KC = KineticsConstants;
    A = KC.Amatrix(Rho);
    [M,d] = eig(A,"vector"); % M is the modal matrix, d is a vector
         % of the eigenvalues.
    dt = Tf/(PlotPoints-1);
    S = [So;0;0;0;0;0;0];
    One = ones(7,1);
    X = zeros(7,PlotPoints);
    for Step = 2:PlotPoints
        t = (Step-1)/(PlotPoints-1)*Tf;
        if Type == "Step"
            g = -(One-exp(d*t))./d; %Note this is a vector computation.
        else % Impulse Case
            g = exp(d*t); %Note this is a vector computation.

        end
        Gd = diag(g);
        G = M*Gd/M;
        X(:,Step) = G*S;
    end
end
```

Alternate Approach to the Source Transfer Function

We will now develop a source transfer function using a traditional algebraic approach which will be illustrative but more complex from a computational perspective.

$$\frac{dn(t)}{dt} = \frac{n(t)*(\rho-\beta)}{\Lambda} + \sum_{i=1}^{6} \lambda_i C_i(t) + \mathrm{S}$$

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

Now take the Laplace transform of both and solve for N(s) eliminating C(s), assume that n(t) and C(t) are both zero. Taking n(0) = 1.0:

$$N(s) = \frac{1}{s\Lambda + (\rho - \beta) - \sum_{i=1}^{6} \frac{\lambda_i \beta_i}{s + \lambda_i}} S(s)$$

This can be simplified a bit by moving the beta within the sum and eliminating this leads to:

$$N(s) = \frac{1}{s\Lambda + \sum_{i=1}^{6} \frac{s\beta_i}{s + \lambda_i} - \rho} S(s)$$

It would be best if this were in the form of a ratio of polynomials, so we now compute one polynomial and one vector of polynomials reducing the first by one factor

$$\psi = \prod_{i=1}^{6}(s + \lambda_i)$$

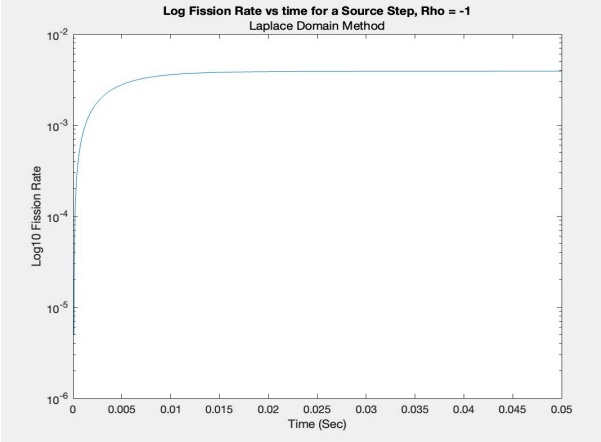
$$\varphi_{i=1:6} = \frac{\psi}{s + \lambda_i}$$

Then

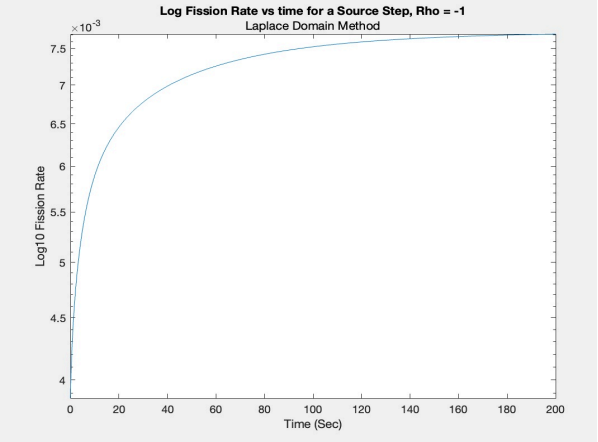
$$N(s) = \frac{\psi}{s(\psi\Lambda + \sum_{i=1}^{6}\beta_i\varphi_i) - \psi\rho}\, S(s)$$

Below are the results for a step. This is the same as we obtained using the matrix method above

Short Term
(0.05 Seconds)



Long Term
(200 Seconds)



The following table shows the MALAB script to create these plots using the Laplace domain method.

```
function PolynomialBasedTransferFunction()
    Rho = -1;
    [Numerator,Denominator]= FindNumeratorAndDenominator(-1);


    Tfinal = 0.05; % Time being observed seconds
    PlotingPoints = 10000;
    TimeArray = 1.0e-4:Tfinal/PlotingPoints:Tfinal;
    H = tf(Numerator,Denominator); %This prepares the transfer function.
    InputArray = ones(length(TimeArray),1);
    FissionRate = lsim(H,InputArray,TimeArray); %This does the simiulation.
    figure
    semilogy(TimeArray,FissionRate);
    title("Log Fission Rate vs time for a Source Step, Rho = "+ Rho);
    subtitle("Laplace Domain Method");
    xlabel("Time (Sec)");
    ylabel("Log10 Fission Rate");

end

function [Numerator,Denominator]= FindNumeratorAndDenominator(Rho)
    KC = KineticsConstants();

    Psi  = 1;
    for i = 1:6
        Psi = conv(Psi,[1 KC.L(i)]);
    end
    Phi = zeros(6,6);

    for i = 1:6
        Phi(i,:) = polydiv(Psi,[1 KC.L(i)]);
    end

    D1 = KC.GenerationTime*[Psi,0];  %This is s*Psi times generation time.
    D2 = zeros(1,6);
    for i = 1:6
        D2 = D2 + KC.B(i).*Phi(i,:);
    end
    D2 = [0,D2,0]; %This is s times the sum times generation time.
    D3 = [0,-Rho*Psi*KC.BetaTotal];
    Denominator = D1+D2+D3;
    Numerator = KC.GenerationTime*Psi;
end
```

The Zero Fission Rate Six Group Reactivity Transfer Function Approximation

This development provides another look at a time varying reactivity and it results in methods to measure (or approximate) the parameter $\frac{\beta}{\Lambda}$. This is done by developing a frequency domain transfer function whose magnitude and phase is a function of a perturbation frequency. To make this measurement, a method for providing an oscillating small reactivity in a real reactor would be required. In any event we assume an initially steady state reactor.

We start with the kinetics equations

$$\frac{dn(t)}{dt} = \frac{n(t)(\rho - \beta)}{\Lambda} + \sum_{i=1}^{6} \lambda_i C_i(t)$$

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

Now replace the variables with perturbed values as follows:

$$\begin{aligned} n(t) &\rightarrow n_0 + \delta n(t) \\ C(t) &\rightarrow C_0 + \delta C(t) \\ \rho(t) &\rightarrow \rho_0 + \delta\rho(t) \end{aligned}$$

Ignore all terms which involve products of variations such as $\delta n(t)\, \delta\rho(t)$.

Also apply the steady state condition for all the terms in initial values:

$$0 = \frac{n_0(\rho_0 - \beta)}{\Lambda} + \sum_{i=1}^{6} \lambda_i C_{i_0}$$

$$0 = \frac{n_0 \beta_i}{\Lambda} - \lambda_i C_{i_0}$$

This results in:

$$\frac{d(\delta n(t))}{dt} = \frac{n_0}{\Lambda}\delta\rho(t) + \frac{\delta n(t)}{\Lambda}\left(\rho_0 - \beta\right) + \sum_{i=1}^{6} \lambda_i \delta C_i(\mathrm{t})$$

$$\frac{d\delta C_i(t)}{dt} = \frac{\delta n(t)\beta_i}{\Lambda} - \lambda_i \delta C_i(t)$$

Now take the Laplace transform of these equations and solve the system for the transfer function ($\mathrm{G(s)} = \frac{\delta n(s)}{\delta \rho(s)}$). Recall that the Laplace transform of $\frac{dY(t)}{dt}$ is $sY(s)$. Where s is the Laplace variable, and it becomes jω in the frequency domain. This results in the following:

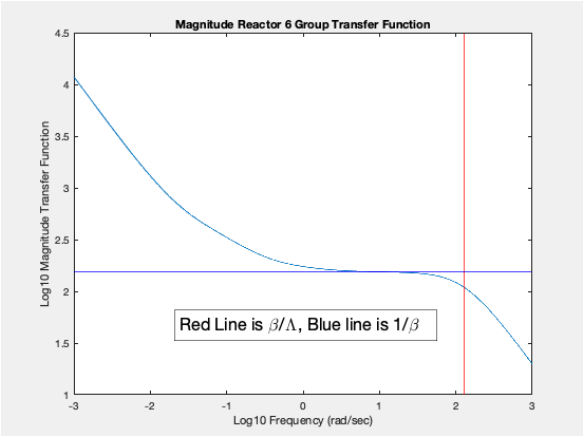
$$G(s) = \frac{n_0}{s(\Lambda + \sum_{i=1}^{6} \frac{\beta_i}{s + \lambda_i}) - \rho_0}$$

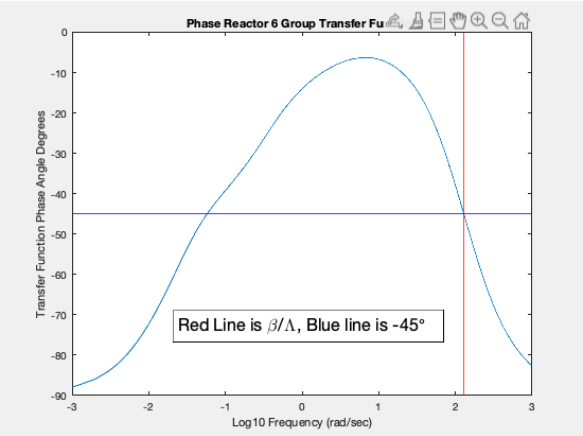
First, we solved for $\delta C_i(t)$ in the second equation and substituted into the first to develop the required ratio.

In the frequency domain this becomes

$$G(\mathrm{j}\omega) = \frac{n_0}{\mathrm{j}\omega(\Lambda + \sum_{i=1}^{6} \frac{\beta_i}{\mathrm{j}\omega + \lambda_i}) - \rho_0}$$

This equation may be used to observe several parameters in reactor testing. Plotting shows some interesting results.





So, if a reactor is presented with a small oscillating reactivity and the resultant fission rate oscillations magnitude and phase are recorded, these may be plotted as shown to estimate the parameters β and Λ. The point where the magnitude is $\frac{\beta}{\Lambda}$ occurs at a frequency of $\frac{129\,rad/sec}{2\pi} = 20.53Hz$.

To understand this approximation, assume $\rho_0 = 0$ and put the transfer function into a one delayed neutron group form:

$$G(\mathrm{j}\omega) = \frac{n_0}{\mathrm{j}\omega(\Lambda + \frac{\beta}{\mathrm{j}\omega + \lambda})} = \frac{n_0(j\omega + \lambda)}{(j\omega)(\Lambda(j\omega + \lambda) + \beta)}$$

$$G(\mathrm{j}\omega) = = \frac{n_0(j\omega + \lambda)}{(j\omega)(j\omega\Lambda + \lambda\Lambda + \beta)} \frac{(-j\omega\Lambda + \lambda\Lambda + \beta)}{(-j\omega\Lambda + \lambda\Lambda + \beta)}$$

$$G(\mathrm{j}\omega) = = \frac{n_0(j\omega + \lambda)(-j\omega\Lambda + \lambda\Lambda + \beta)}{(j\omega)((\omega\Lambda)^2 + (\lambda\Lambda + \beta)^2)}$$

The phase of this transfer function is 45° when the real and imaginary parts of the numerator are equal:

$$\omega\,\beta = \lambda^2\Lambda + \lambda\,\beta + \omega^2\Lambda$$

If the last term on the right-hand side dominates, we have:

$$\omega \cong \beta/\Lambda$$

By setting the real part and imaginary part of the numerator equal we would have a +45° angle. The factor of 1/j however represents a rotation of minus 90 degrees yielding the required -45° angle.

To be useful this transfer needs to be in a form where the roots can be readily computed. This implies that we need a ratio of polynomials.

To put the transfer function in this form we need to multiply numerator and denominator by the product:

$$\Psi = \prod_i (s + \lambda_i)$$

We also compute a matrix:

$$\phi_{i,:}^{6x5} = \Psi/(s + \lambda_i)$$

With this our equation becomes a ratio in polynomials:

$$G(\mathrm{j}\omega) = \frac{n_0\Psi}{\mathrm{s}(\Psi\Lambda + \sum_{i=1}^{6}\beta_i\phi_i(i,:)) - \Psi\rho_0}$$

The vector Ψ and the matrix ϕ both have constant sets of polynomial coefficients.

	S^6	S^5	S^4	S^3	S^2	S^1	S^0
Ψ	1	4.60482	5.36551	1.77599	0.183602	0.00553084	4.37243e-05]
$\phi(1,:)$	0	1	4.59236	5.30827	1.70983	0.162291	0.00350804
$\phi(2,:)$	0	1	4.57429	5.22586	1.61645	0.134254	0.00143224
$\phi(3,:)$	0	1	4.49341	4.86488	1.23397	0.0461195	0.000392446
$\phi(4,:)$	0	1	4.30352	4.06884	0.550034	0.0178747	0.000145117
$\phi(5,:)$	0	1	3.46876	1.42477	0.157357	0.00483454	3.84875e-05
$\phi(6,:)$	0	1	1.59178	0.569413	0.0603283	0.00183082	1.45117e-05

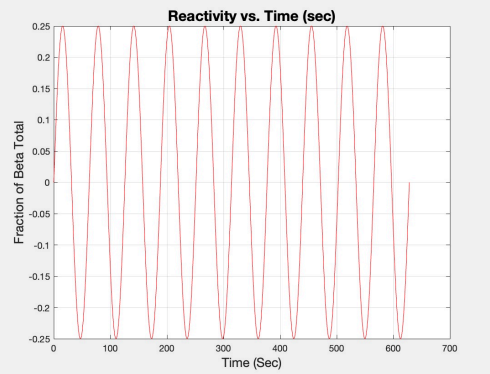
The material in this section has been presented largely for historical reasons. One does not do this testing of this sort on a fission rate reactor. Generally, these tests require a specially designed configuration. This sort of testing was done in the past using test reactors. None the less, it is worthwhile because it has introduced a method that will use later in analyzing the stability of closed loop reactor systems where the reactor heats fuel water and both impact reactivity.

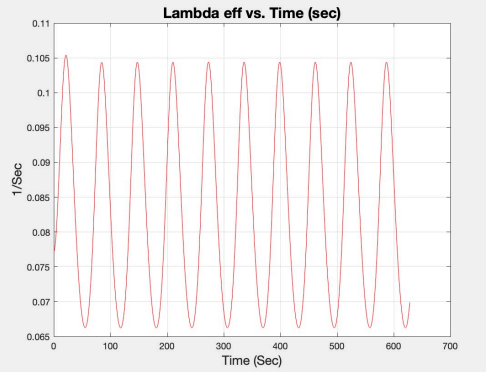
Oscillations of Significant Size

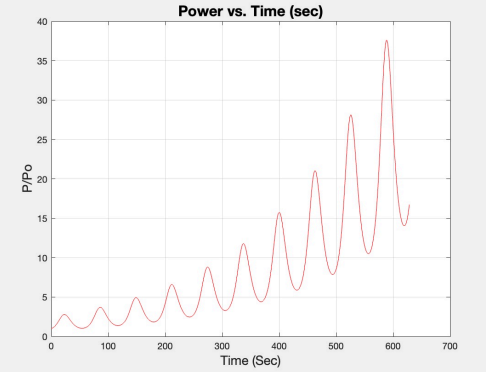
The small signal approximation leads to an output fission rate oscillation with a phase delay relative to the reactivity oscillation. This misses a real and interesting nonlinear aspect of a reactor's response to an oscillating reactivity. The actual output fission rate will oscillate but it will do so about a rising average.

What follows is an oscillation with a peak reactivity of 0.25β and a frequency of 0.016 Hz.

Reactor Response to a 0.25β Oscillation







Additional Topics:

1. SUR Equation Without the Prompt Jump Assumption and S=0

If we do not allow either the prompt jump assumption or the single delayed group approximation the startup equation takes the following form.

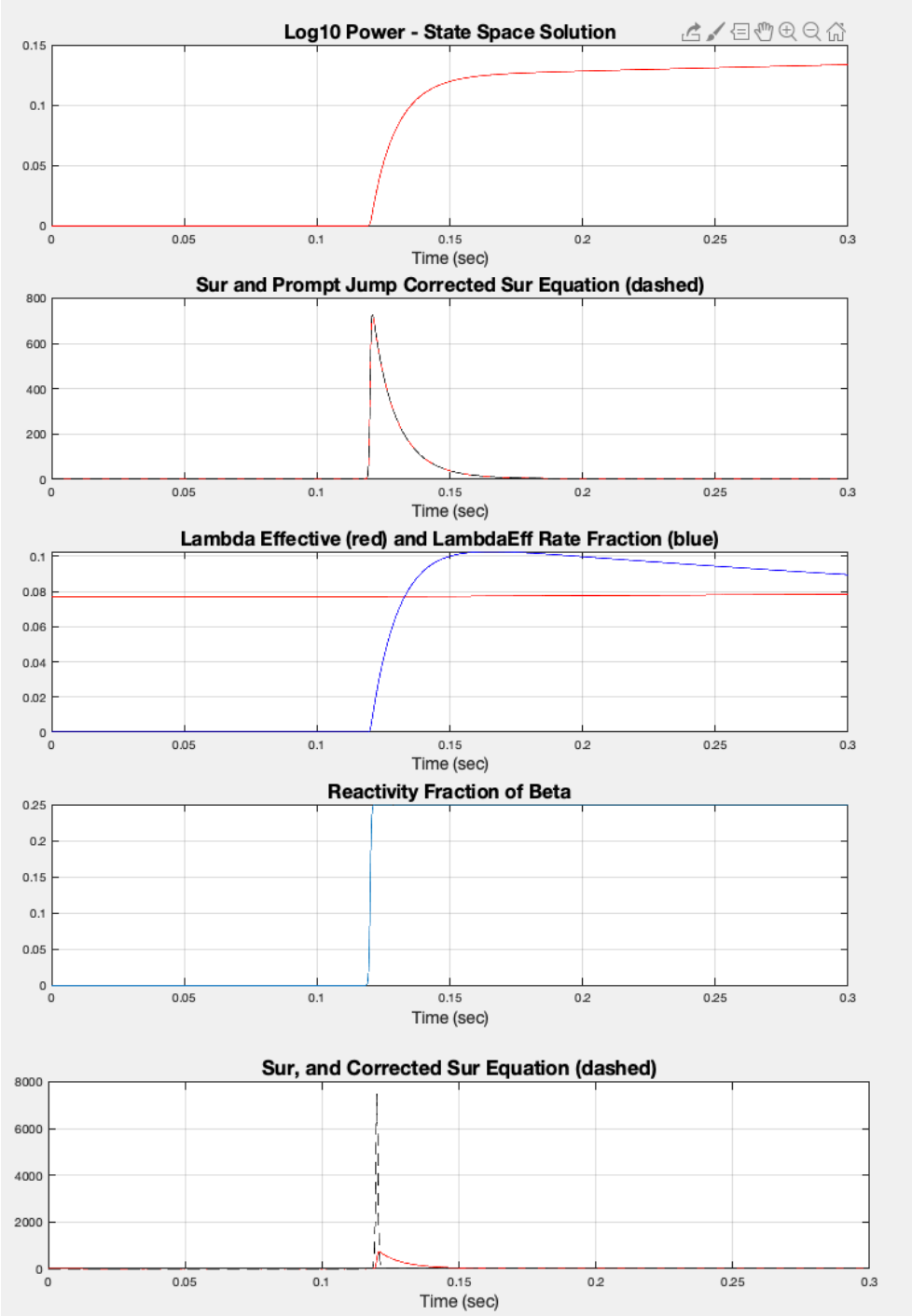
$$SUR = 26.06[dpm - sec]\frac{-\frac{\Lambda\ddot{n}}{n} + \dot{\rho} + \lambda_{eff}\rho + \frac{\dot{\lambda}_{eff}}{\lambda_{eff}}(\beta - \rho)}{\beta - \rho - \frac{\dot{\lambda}_{eff}}{\lambda_{eff}}\Lambda + \lambda_{eff}\Lambda}$$

This equation works well for demonstrating the SUR during a rapid change in reactivity but, near prompt criticality, it can suffer singularities and fail. The best approach is to calculate the SUR directly using the six-group reactor kinetics equation solutions as

$$SUR = 26.06[dpm - sec]\ \dot{n}/n$$

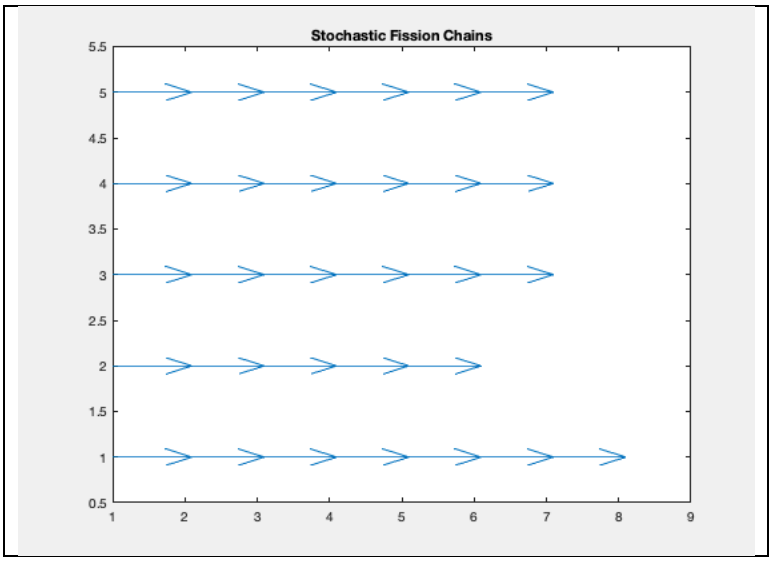
Even for the equation above we need to solve the six-group problem to obtain the second derivative of n, λ_{eff}, and the first derivative of λ_{eff}.

The plots below demonstrate a prompt jump (insertion with a tau of 0.5ms). First using the prompt jump corrected equation show in this section and last, using the version derived above for the variable lambda effective.



The first four of these plots are using the prompt jump corrected version of the SUR equation. The last plot is SUR for the same transient using the corrected lambda effective derivative method. One can see that the latter case significantly overestimates the SUR during the transient. In both cases, the six-group group solution is plotted in red.

2. <u>Another way to think about the reactor neutron multiplication process.</u>



Suppose we think of what is going on in a reactor as a set of chains of fissions with each chain initiated by the decay of a precursor or by introduction of a source neutron. The chance that one of these neutrons causes a subsequent fission is $P = K_{eff}(1-\beta)$. The expected value for the length of a chain is given by:

$$< k > = \frac{\sum_{k=1}^{\infty} kP^k}{\sum_{k=1}^{\infty} P^k} = \frac{\frac{P}{(1-P)^2}}{\frac{P}{(1-P)}} = \frac{1}{1-P}$$

This becomes:

$$< k > = \frac{1}{1-K_{eff}(1-\beta)} = \frac{1/K_{eff}}{\beta-\rho}$$

A critical reactor would have an expected chain length of $\frac{1}{\beta}$= 155 fissions.

Now suppose we consider the number of fissions caused by precursor decay or source neutron emission in a time Δt. We obtain a total number of chain creations as $\Delta t(\sum_{i=1}^{6} \lambda_i C_i + S)$. Putting this together with the expected length of each chain we get that the total fission rate as:

$$n = \frac{\frac{\Delta t}{K_{eff}}(\sum_{i=1}^{6} \lambda_i C_i + S)}{\beta - \rho}$$

And this is our prompt jump approximation fission rate expression:

$$n = \frac{\Lambda(\sum_{i=1}^{6} \lambda_i C_i + S)}{\beta - \rho}$$

When viewed this way, the prompt jump may be thought of as a sudden rise in the length of the chains.

Likewise, fission rate turning with a positive reactivity and a negative reactivity addition rate can be understood. The positive reactivity implies that the precursor concentrations are still rising. Hence the chain creation is also rising. The negative reactivity addition rate will imply that there is a shortening of the chains. Fission rate will turn when these two effects balance.

3. Perturbation approach to calculating a reactivity ramp calculation.

For small transients with a limited reactivity and for a limited time, we can represent the kinetics equations as:

$$\frac{d(\delta n(t))}{dt} = \frac{n_0}{\Lambda}\delta\rho(t) + \frac{\delta n(t)}{\Lambda}\left(\rho_0 - \beta\right) + \sum_{i=1}^{6}\lambda_i \delta C_i(\mathrm{t})$$

$$\frac{d\delta C_i(t)}{dt} = \frac{\delta n(t)\beta_i}{\Lambda} - \lambda_i \delta C_i(t)$$

We have assumed that any second order variation may be ignored.

In the same vein as we used in developing the prompt jump equation we can rewrite this as:

$$\frac{dC}{dt} = Apj(0) * C + \frac{n_0}{\Lambda}\delta\rho(t)\begin{bmatrix}\beta_1\\ \beta_2\\ \beta_3\\ \beta_4\\ \beta_5\\ \beta_6\end{bmatrix}$$

$Apj(0)$ is our 6x6 version of the prompt jump matrix computed with a zero reactivity. Define the vector of betas as $\vec{\beta}$. We can integrate this as before:

$$C(t) = e^{\mathrm{Apjt}}C(0) + \frac{n_0}{\Lambda}e^{\mathrm{Apjt}}\int_0^t e^{-Apjt'}\delta\rho(t)\,\mathrm{d}t'\vec{\beta}$$

Now remembering the basics, $e^{\mathrm{Apj\,t}}$ is a matrix. It can be put into diagonal form using the same M matrix that diagonalizes the Apj matrix.

This means that the task of performing the integration may be greatly simplified. We can look at the second term as follows.

$$C(t) = e^{\mathrm{Apjt}}C(0) + \frac{n_0}{\Lambda}Me^{\mathrm{Dpjt}}M^{-1}\int_0^t Me^{-Dpjt'}M^{-1}\delta\rho(t)\,\mathrm{d}t'\vec{\beta}$$

And because M is constant it may be moved outside of the integral.

$$C(t) = e^{\mathrm{Apjt}}C(0) + \frac{n_0}{\Lambda}Me^{\mathrm{Dpjt}}\int_0^t e^{-Dpjt'}\delta\rho(t')\,\mathrm{d}t'M^{-1}\vec{\beta}$$

So, during the rod pull $\delta\rho(t') = RS * t$ and after the rod pull it is constant, $\delta\rho(t') = Rho$

During the rod pull, the right-hand term becomes, for each eigenvalue, d_k:

$$RS(e^{d_k t_{pulling}} - d_k t_{pulling} - 1)/d_k^2$$

In the case that d_k is zero this becomes:

$$\frac{RS}{2}{t_{pulling}}^2$$

Once the rod motion stops this contribution to the total effect includes the integral taken to the limit of the rod pull multiplied by the decaying e^{Dpjt} . This results in the "past moving" part of the integral in that circumstance. $t_{current}$ here is measured from the time that the rods stopped moving.

$$RS(e^{d_k t_{current}} - \left(1 + d_k t_{pulled}\right)e^{d_k(t_{current} - t_{pulled})})/d_k^2$$

If d_k is zero, this reduces to $RS * {t_{pulled}}^2$.

Finally with the pulling stopped we need to add the contribution of the existing constant reactivity Rho:

$$Rho * (e^{d_k t_{current}} - 1)$$

If d_k is zero, this reduces to Rho*$t_{current}$.

So, we wind up with a resulting diagonal matrix $\widehat{D}(t)$ which is a function of time with its diagonal elements computed using either the first (moving) equation or the sum of the second two (past moving and stopped) equations.

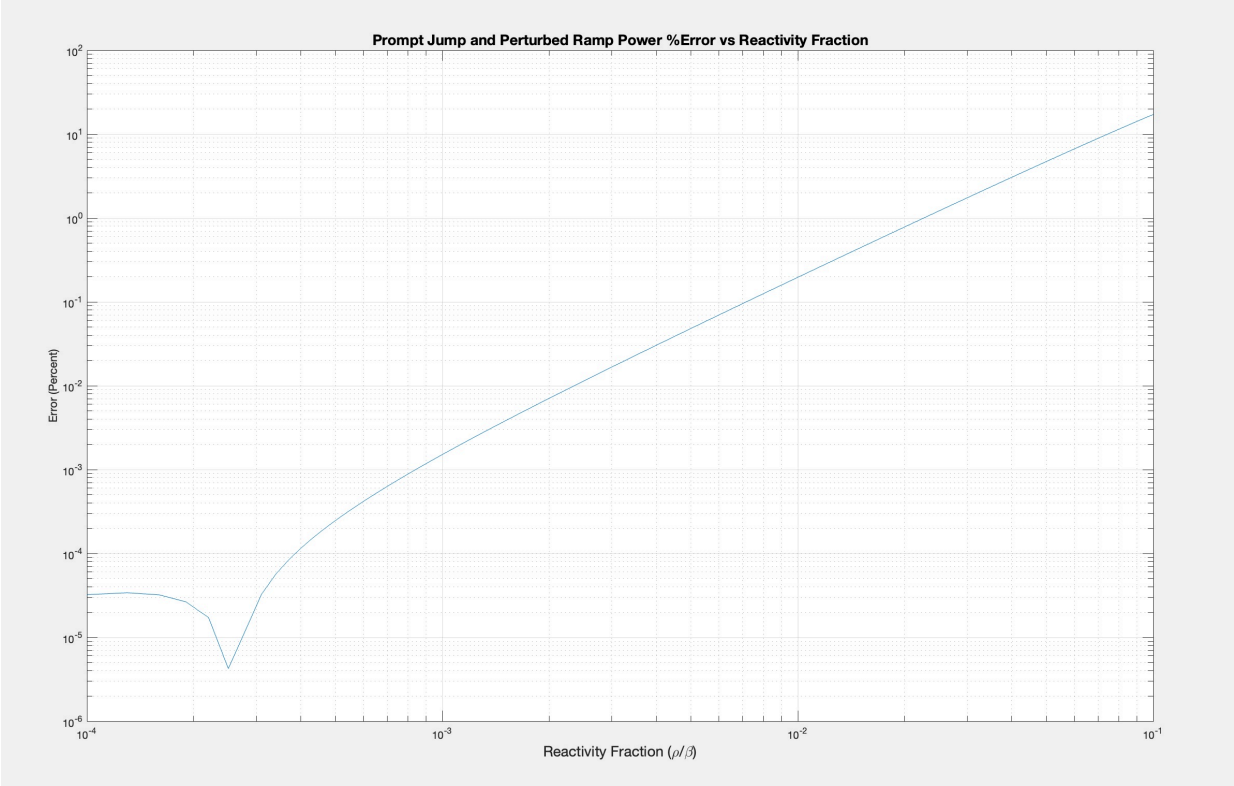
So, we can develop the final form of the term as follows

$$\frac{n_0}{\Lambda} M \widehat{D}(t) M^{-1} \vec{\beta}$$

Thus, with the limiting constraint requiring a small Rho and a short time we have the following ready to exactly compute:

$$C(t) = e^{\mathrm{Apjt}} C(0) + \frac{n_0}{\Lambda} M \widehat{D}(t) M^{-1} \vec{\beta}$$

While this is interesting, from a practical standpoint we can not so drastically limit the time of a transient or the amount of reactivity. The following graph shows the percent error in this expression relative to a numerically computed solution without the perturbation approximation. In each case we are allowing the transient to start after ten seconds, pull rods for ten seconds, and wait after the pull for 40 sec. We are measuring only the final fission rate after the transient.



This approach is interesting in that it can be directly computed but as may, be seen with only 0.1β of reactivity, the error is 17%. A typically 1DPM startup rate results from 0.25β.