Solution of the Point Kinetics Equations

Introduction

In general, analytical solution of the point kinetics equations is not easy -- recall that we have a coupled set of seven ODEs. Thus, in most cases, these equations are evaluated for a given k(t) or ρ(t) using numerical methods (such as Matlab's **ode15s** solver, for example). However, for a few specific cases, the analytical solution schemes are manageable -- and the resultant solutions give considerable insight into the general behavior of the time dependent neutron balance in real systems. Thus, the focus of this section of notes is to give an overview of some typical analytical solutions with an emphasis on the interpretation of the analytical expressions for the time dependent amplitude of the neutron flux under various conditions. This will allow us to introduce some common terminology related to reactor kinetics, and to gain a good understanding of the expected behavior in several common situations. In addition, we also overview the operation of the **kinetics_gui** code which allows easy visualization of many of the key concepts discussed within this set of Lecture Notes.

Step Change in Reactivity

In particular, one common situation that is relatively easy to solve analytically involves a step change in reactivity in a critical reactor operating at low power (the "low power" specification is needed so that **reactivity feedback effects are negligible**). If we focus on the generation time formulation of point kinetics (see Refs. 1 and 2 for details) with no external source, we have

$$\frac{dP}{dt} = \frac{(\rho - \beta)}{\Lambda} P + \sum_i \lambda_i c_i \tag{1}$$

$$\frac{dc_i}{dt} = \frac{\beta_i}{\Lambda} P - \lambda_i c_i \qquad \text{for } i = 1, 2, \cdots 6 \tag{2}$$

where $c_i(t)$ refers to the normalized precursor densities, P(t) is the reactor power level (which is simply proportional to the amplitude of the neutron flux), Λ, β_i, and λ_i are the so-called effective kinetics parameters for the system of interest, and ρ(t) is the reactivity. In most applications of eqns. (1) and (2), the kinetics parameters are assumed to be constant, the reactivity is the driving force for the transient, and P(t) and $c_i(t)$ are the dependent variables that vary with time due to some changing ρ(t). However, for a step change in reactivity, **ρ(t) = ρ = constant**, and the above equations become a system of seven linear constant coefficient ODEs -- and this falls into a class of problems that we know how to handle analytically (from your sophomore-level Differential Equations class).

The standard approach for solving linear time-invariant systems is to assume a solution of the form of a simple exponential. Following this technique, we assume that

$$P(t) = A_o e^{\omega t} \qquad \text{and} \qquad c_i(t) = A_i e^{\omega t} \tag{3}$$

Now, we substitute these assumed solutions into the precursor balance equation, to obtain

$$A_i \omega e^{\omega t} = \frac{\beta_i}{\Lambda} A_o e^{\omega t} - \lambda_i A_i e^{\omega t}$$

or $$A_i\left(\omega+\lambda_i\right)=\frac{\beta_i}{\Lambda}A_o$$

and $$A_i=\frac{\beta_i/\Lambda}{\omega+\lambda_i}A_o \tag{4}$$

Now, putting the assumed solutions, along with eqn. (4), into the equation for P(t) gives

$$A_o\omega e^{\omega t}=\left(\frac{\rho-\beta}{\Lambda}\right)A_o e^{\omega t}+\sum_i\lambda_i A_i e^{\omega t}=\left(\frac{\rho-\beta}{\Lambda}\right)A_o e^{\omega t}+\sum_i\lambda_i\frac{\beta_i/\Lambda}{\omega+\lambda_i}A_o e^{\omega t}$$

Cancelling the common $A_o e^{\omega t}$ factor in each term and multiplication by Λ gives

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

and solving for ρ gives

$$\rho=\Lambda\omega+\beta-\sum_i\frac{\beta_i\lambda_i}{\omega+\lambda_i} \tag{5}$$

To put this expression into standard form, note that $\beta=\sum\beta_i$. Now, using this equality, eqn. (5) becomes

$$\rho=\Lambda\omega+\sum_i\left(\beta_i-\frac{\beta_i\lambda_i}{\omega+\lambda_i}\right)=\Lambda\omega+\sum_i\left(\frac{\beta_i\omega+\beta_i\lambda_i-\beta_i\lambda_i}{\omega+\lambda_i}\right)$$

or $$\rho=\Lambda\omega+\sum_i\frac{\beta_i\omega}{\omega+\lambda_i} \tag{6}$$

This equation is the standard form of the so-called **reactivity equation** obtained from the generation time formulation of point kinetics. From a pure mathematical viewpoint, it is simply the **characteristic equation** associated with the original seven linear constant coefficient ODEs -- and, for a given value of reactivity, the **roots** of this equation give the values of ω that satisfy the original form of the assumed solution. Also, since we have seven coupled first-order ODEs (giving an overall 7th order system), we should expect seven roots (or seven values of ω) that will satisfy eqn. (6). Assuming that each ω_j for j = 1, 2, ··· 7 is distinct, the linear superposition principle allows us to write the general solution for P(t) as a linear combination of the linearly independent individual solutions. Thus, for example, we can write a general solution for the time-dependent power level as

$$P(t)=\sum_{j=1}^{7}A_j e^{\omega_j t}=A_1e^{\omega_1 t}+A_2e^{\omega_2 t}+\quad\cdots\quad+A_7e^{\omega_7 t} \tag{7}$$

where the ω_j are the roots of eqn. (6) and the A_j are the seven arbitrary coefficients needed for the general solution of a 7th order initial value problem (IVP). To obtain a unique solution (i.e. determine explicit values for the A_j coefficients), we simply need to specify seven initial conditions (ICs) on the original balance equations given in eqns. (1) and (2).

Although the above paragraph establishes a solid mathematical foundation for our solution methodology and the analytical form for the time-dependent power level (or neutron amplitude), it really has not shed a lot of insight into the actual behavior of P(t). To do this, we must first try to get a better understanding of the reactivity equation and the value of its roots.

To gain this insight, let's try to actually visualize the root finding problem implied by the reactivity equation given in eqn. (6). To do this, we can think of the right hand side (RHS) of eqn. (6) as some function of ω, say $f(\omega)$, and simply plot $f(\omega)$ vs. ω for a wide range of ω. Then, if we superimpose the LHS of eqn. (6) [i.e. $\rho(\omega)$ = constant] on the plot, the intersections of the two curves give the desired roots, ω_j, of the reactivity equation. Although in theory this is quite straightforward, the generation of a single plot that shows the desired visualization is somewhat impractical because of the wide range of ω values that must be addressed. Thus, to see this relationship quantitatively, multiple plots covering different ranges and scales for the ω variable are often given.

This capability has been implemented within a Matlab code and a typical set of profiles are given in Figs. 1-3 for three different ranges of ω (here $\Lambda = 0.1$ ms and $\beta = 0.0065$). In these plots there are seven individual red curve segments for $f(\omega)$ separated by six vertical blue dashed lines that represent the asymptotes of $f(\omega)$ as $\omega \rightarrow -\lambda_i$ [note that $f(\omega) \rightarrow \pm\infty$ as $\omega \rightarrow -\lambda_i$]. In addition to the red $f(\omega)$ curve, we have also plotted the $\rho(\omega)$ = constant lines (dashed green lines) for the case where $\rho = \pm 0.25$ dollars of reactivity (note that $\rho = \beta$ is \$1 of positive reactivity). The seven intersections of the solid red $f(\omega)$ curves and the dashed green ρ = constant lines represent the seven specific roots of the reactivity equation (characteristic equation) for the given parameters for this example problem ($\Lambda = 0.1$ ms, $\beta = 0.0065$, and $\rho = \pm$ \$0.25).

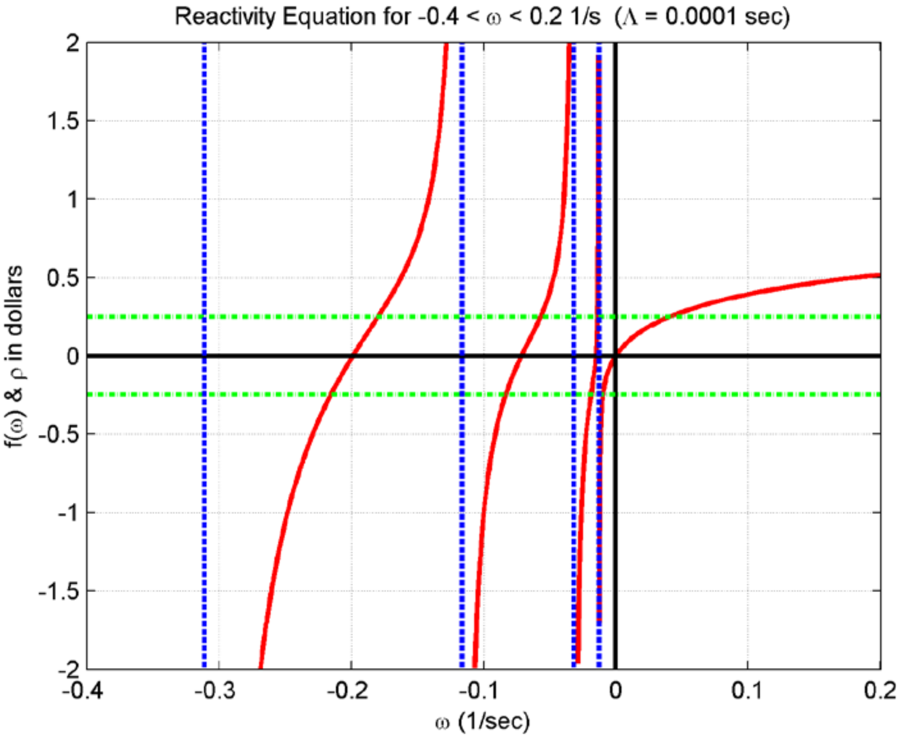


Fig. 1 Reactivity equation for -0.4 < ω < 0.2 1/s.

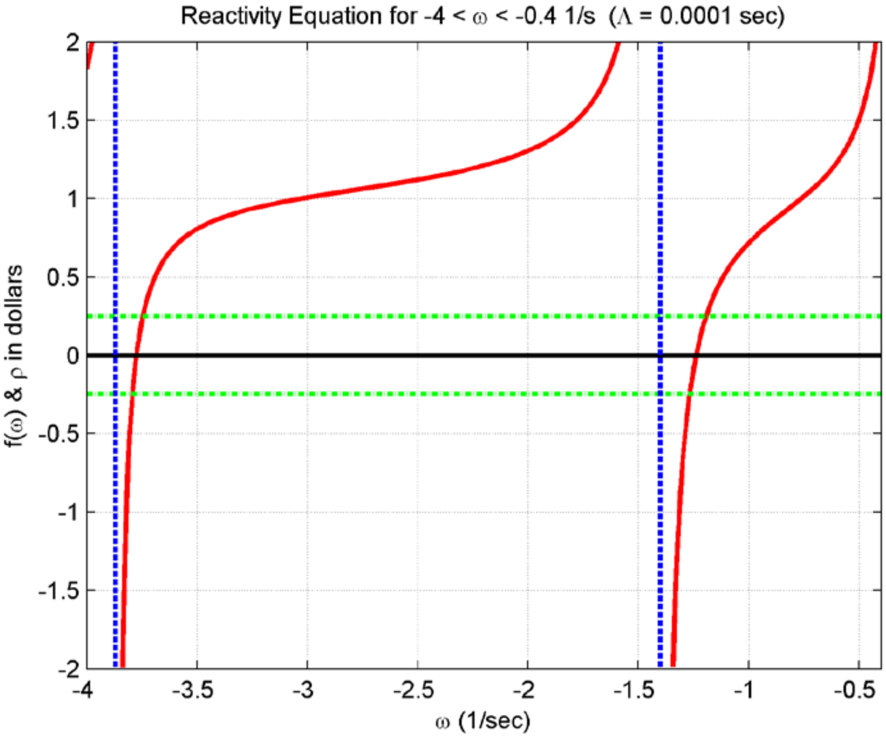


Fig. 2   Reactivity equation for -4 < ω < -0.4 1/s.

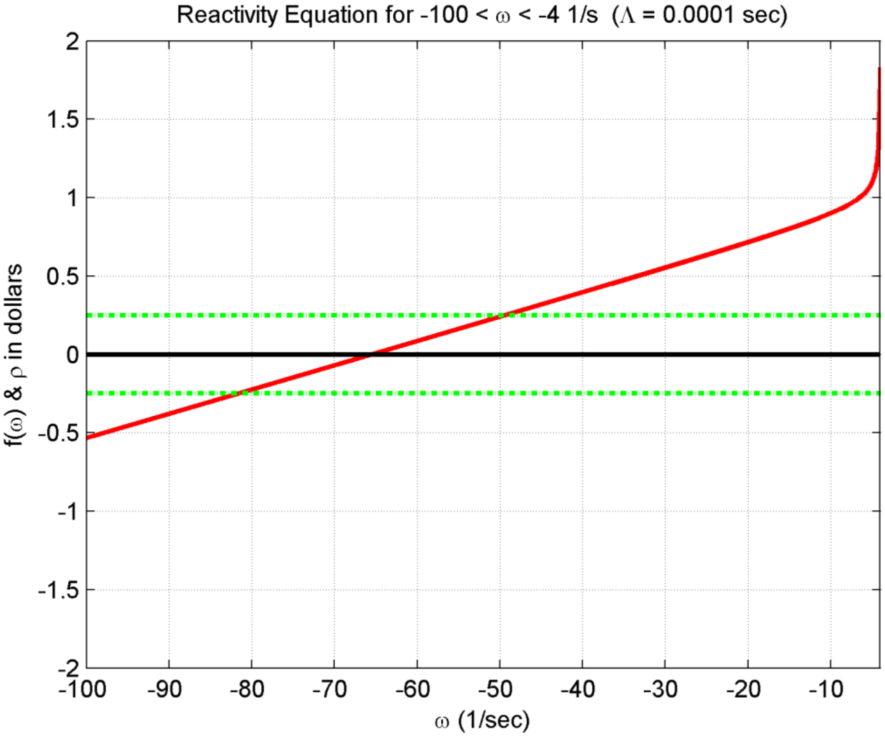


Fig. 3   Reactivity equation for -100 < ω < -4 1/s.

Although the numerical values of all the roots, ω_j, are needed for quantitative analysis, here we are only interested in a qualitative description of the transient behavior of the neutron flux magnitude. In particular, the key observations from Figs. 1-3 can be summarized as follows:

positive ρ: reactivity equation has **one positive root** and **six negative roots**

negative ρ: reactivity equation has **seven negative roots**

If we order the roots ω_j from most positive to most negative, then, after a relatively short transient time, the last six terms in eqn. (7) decay away (because $\omega_2, \omega_3, \cdots, \omega_7 < 0$ for both positive and negative reactivity), leaving only the term containing ω_1, or

$$P(t) = P_1 e^{\omega_1 t} = P_1 e^{\pm t/\tau} \qquad \text{(after some transient time)} \qquad (8)$$

where $\tau = 1/|\omega_1|$ is called the **reactor period** and P_1 is the power level after the short transient period. If ρ is positive, $\omega_1 > 0$, and the reactor period is positive -- so the power level and neutron population grow indefinitely as $e^{t/\tau}$ (remember that we are assuming no feedbacks at this point). And, for negative reactivity, just the opposite occurs. That is, for $\omega_1 < 0$, the reactor period is negative -- so the power level and neutron population decrease indefinitely towards zero with the form $e^{-t/\tau}$. Thus, after a short transient time, the dominate behavior of P(t) is simply associated with the most positive root of the reactivity equation and it is represented as a simple growing or decaying exponential as shown in eqn. (8). This shows that the time behavior after the early fast transient is simply characterized by the reactor period τ -- which is indeed a very important quantity in reactor operations!

Since $\tau = 1/|\omega_1|$ is so important, we can solve eqn. (6) for this quantity for a range of reactivity values (both positive and negative) and plot the summary results. This has also been done within a Matlab code and the results for the case where $\Lambda = 0.1$ ms and $\beta = 0.0065$ are given in Fig. 4. This figure can be thought of as a “design curve” or an “operational aid” depending on the intended use. For example, if we wanted to know the reactor period associated with a reactivity change of $\rho = \pm \$0.25$, we can simply extract this information from Fig. 4 (the green dashed line has been included on the plot to highlight this case). For this specific case, $\rho = +25$ cents gives $\tau \approx 24.4$ sec and $\rho = -25$ cents gives $\tau \approx -102$ sec. From a different perspective, one might, for example, add a small amount of positive reactivity into a critical system by pulling out a control rod a small distance and, by observing the power vs. time after the short transient period, actually measure the reactor period, τ. If, in a specific case, a positive period of about 100 seconds is obtained, then Fig. 4 shows that the reactivity worth of the rod movement corresponds to about +0.095 dollars or +9.5 cents. In short, a reactor period vs. reactivity curve can be a very useful tool for any system!

To wrap up our formal discussion of eqns. (6) and (7), it makes sense to show the typical P(t) behavior for a specific change in both positive and negative reactivity. This was done in the same Matlab code as noted above, but the actual solution was generated via numerical solution rather than analytically as described above. The numerical approach was selected for **two important reasons,** as follows:

1. **The numerical solution is much easier to obtain.** The point of the above discussion for the analytical solution was so we could get a good understanding of the expected behavior and to introduce some important terminology associated with reactor kinetics. However, actually

computing accurate values for all seven roots of the reactivity equations (via seven calls to Matlab's **fzero** routine, for example), and then setting up the appropriate equations and solving for the seven coefficients for each transient situation of interest is not really easy to implement. It certainly can be done for the case where ρ = constant but, even for this simple situation, the numerical solution of the seven coupled ODEs using a readily available ODE solver is a much easier path to follow. Thus, this was the technique chosen here.

2. **The numerical solution allows the treatment of feedback effects into the model, since we are not restricted to the case where ρ = constant.** Recall that the above development of the analytical solution is based on the fact that ρ = constant. In practice, however, the material properties are all functions of temperature and, as the neutron flux and power level change, so do the material temperatures within the system. Thus, the unbounded exponential increase predicted for a positive reactivity insertion is not realistic -- instead, the inherent negative feedbacks within the system tend to stabilize the behavior at some point.

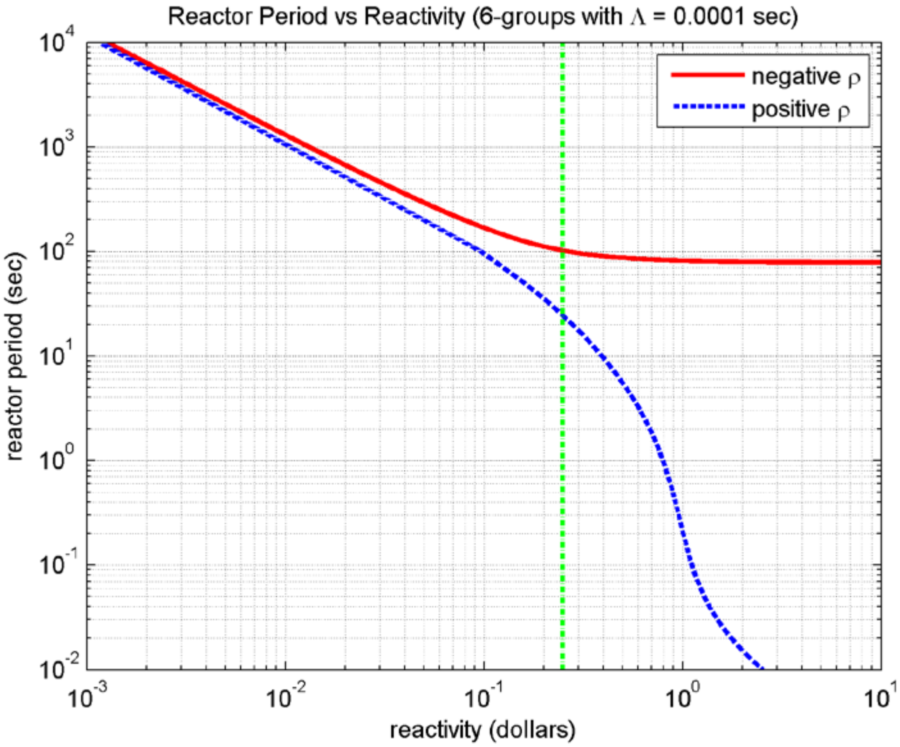


Fig. 4 Reactor period vs. reactivity for both positive and negative changes in ρ/β.

To illustrate the stabilizing effect associated with negative feedback, let's define a generic power feedback coefficient as

$$\alpha_p = \frac{\partial \rho}{\partial P} = \frac{1}{k^2}\frac{\partial k}{\partial P} \approx \frac{1}{k}\frac{\partial k}{\partial P} \tag{9}$$

With this definition, we see that the actual reactivity that enters into the point kinetics equation is a combination of the externally applied reactivity, ρ_{ext} (e.g., due to a change in control rod position) and the feedback reactivity, ρ_f, which is inherently time-dependent due to changes that are not under operator control, or

$$\rho(t) = \rho_{ext} + \rho_f(t) = \rho_{ext} + \alpha_p\left(P(t) - P_o\right) \qquad (10)$$

Note also that the system is inherently unstable and it will quickly destroy itself if the feedback coefficient is positive. This is easily apparent from eqn. (10) since, with a positive value of α_p, an increase in reactivity leads to an increase in power, which, in turn, leads to a further increase in reactivity, which gives another increase in power, and so on -- which leads to a runaway system. If, on the other hand, α_p is negative, then an increase in power reduces ρ, which decreases P, which increases ρ, etc. until some new equilibrium steady state condition is realized. Thus, this situation is clearly the only reasonable option, and all operating reactors are required to have a negative feedback coefficient under all possible hot zero or full power conditions.

Also it should be emphasized that, by definition, criticality is achieved when $\rho = 0$. Thus, we see that the new steady state power level associated with the negative feedback case will be reached when the feedback reactivity exactly cancels the applied external reactivity, or

$$\alpha_p\left(P_{new} - P_o\right) = -\rho_{ext} \qquad \text{or} \qquad P_{new} = P_o - \frac{\rho_{ext}}{\alpha_p} \qquad (11)$$

Well, with this brief background, we can now actually simulate and compare the dynamics of a system with and without feedback. For the feedback-free case, we simply set $\alpha_p = 0$, and for the simulation with inherent feedbacks, we set the power feedback coefficient to its appropriate value for the system of interest (for the current example with feedbacks, $\alpha_p = -2\times10^{-4}$ $\Delta k/k$ per unit ΔP). The results of the Matlab simulation for the two cases with and without feedback are shown in Figs. 5 and 6 for the insertion and removal of reactivity, respectively.

In particular, Fig. 5 shows the expected unbounded exponential increase in the flux or power level for the case of a positive reactivity insertion with no feedback (as predicted in the above analytical treatment). However, for the case where the inherent negative feedback is treated, the power level rises less rapidly and it eventually levels off at a new steady state power that is roughly 9 times the initial value [note that $P_o - \rho_{ext}/\alpha_p = 1 + (0.25)(0.0065)/(0.0002) = 9.125$].

For the case where $\rho_{ext} = -25$ cents, both simulations (with and without feedback) lead to a decreasing power level as shown in Fig. 6. In the case with inherent feedbacks, the exponential decrease is reduced slightly, but not enough to keep the reactor from complete shutdown. This is true because the positive reactivity due to the power feedback is not sufficient to overcome the original negative external reactivity added to the system. Thus, both cases lead to complete shutdown (for the simulation parameters used here).

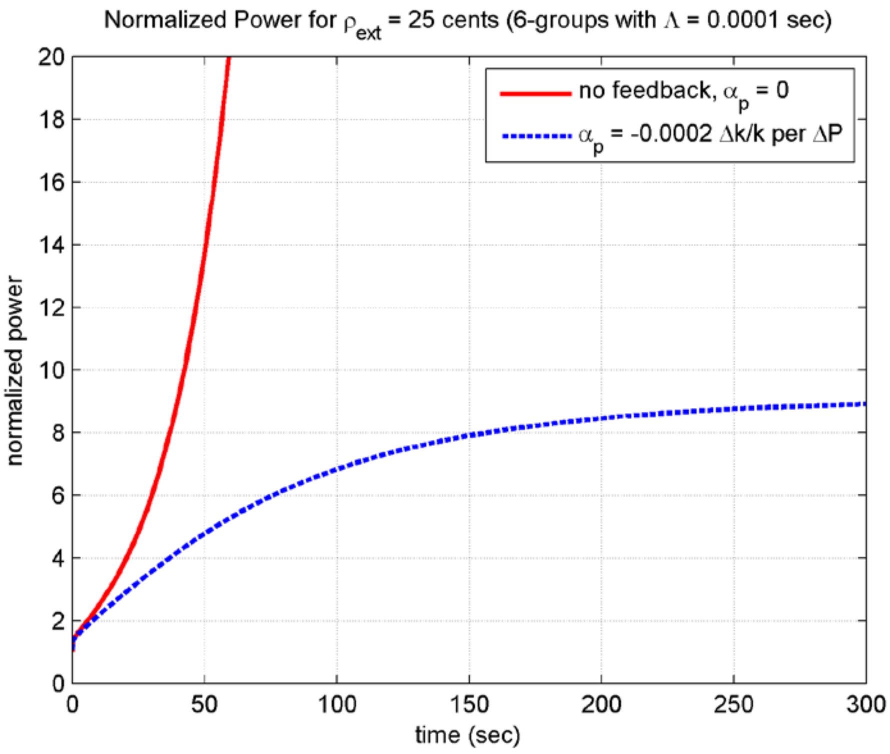


Fig. 5 $P(t)/P_o$ for ρ_{ext} = +25 cents.

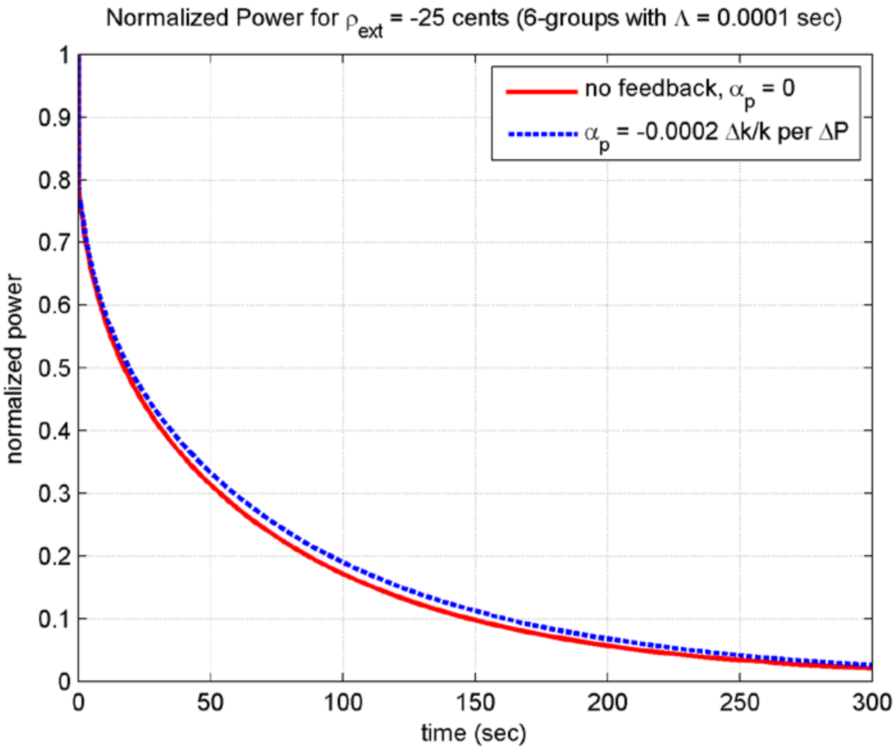


Fig. 6 $P(t)/P_o$ for ρ_{ext} = -25 cents.

The Prompt Jump/Drop

If we expand upon the first few seconds of the transient profiles for the normalized power shown in Figs. 5 and 6, we see a nearly instantaneous rise or fall in the normalized power immediately after the step change in reactivity is made. This prompt jump and drop are highlighted for the case of a 25 cent change in both positive (jump) and negative (drop) reactivity in Figs. 7 and 8, respectively. The rapid change seen here is due to the most negative root, ω_7, of the reactivity equation -- that is, the $A_7 e^{\omega_7 t}$ term in eqn. (7). Because ω_7 has such a large negative value, this term goes to zero very quickly after initiation of the transient.

Since the prompt jump/drop is an inherent feature of each transient, it would be convenient if we could get a quick and easy-to-use estimate of the magnitude associated with this phenomenon. In particular, since we have already argued that the $A_1 e^{\omega_1 t}$ term dominates the transient response after a short period (for the no feedback case), if we could determine the normalized power, P_1, just after the prompt jump/drop, we would have a simple way to estimate the complete power profile versus time,

$$P(t) = \left(\frac{P_1}{P_o}\right) P_o e^{\omega_1 t}$$

or

$$\frac{P(t)}{P_o} = \frac{P_1}{P_o} e^{\pm t/\tau} \qquad (12)$$

where P_1/P_o (or T_1/T_o if we use the normalized amplitude from Ref. 1 instead of the normalized power from Ref. 2) is the desired magnitude of the prompt jump ($P_1/P_o > 1$) or prompt drop ($P_1/P_o < 1$).

To derive an expression for P_1/P_o, let's go back and consider the generation time formulation of point kinetics as given in eqns. (1) and (2). Note that, since the phenomenon of interest here occurs very rapidly, we are only interested in the transient state over a period of 0.5 second or less. Over this short interval, it is very reasonable to assume that the precursor densities do not change significantly. Thus, over the time scale of interest for the prompt jump/drop, we have $dc_i/dt \approx 0$ and eqn. (2) gives

$$\lambda_i c_{io} = \frac{\beta_i}{\Lambda} P_o \qquad \text{or} \qquad \sum_i \lambda_i c_{io} = \frac{\beta}{\Lambda} P_o$$

where all the quantities are evaluated just prior to the reactivity change (i.e. at t = 0).

With this expression and the same assumption as above, the P(t) equation becomes

$$\frac{dP}{dt} = \frac{(\rho - \beta)}{\Lambda} P + \frac{\beta}{\Lambda} P_o$$

This is just a simple first order linear ODE that, when written in standard form, gives

$$\frac{dP}{dt} - \frac{(\rho - \beta)}{\Lambda} P = \frac{\beta}{\Lambda} P_o$$

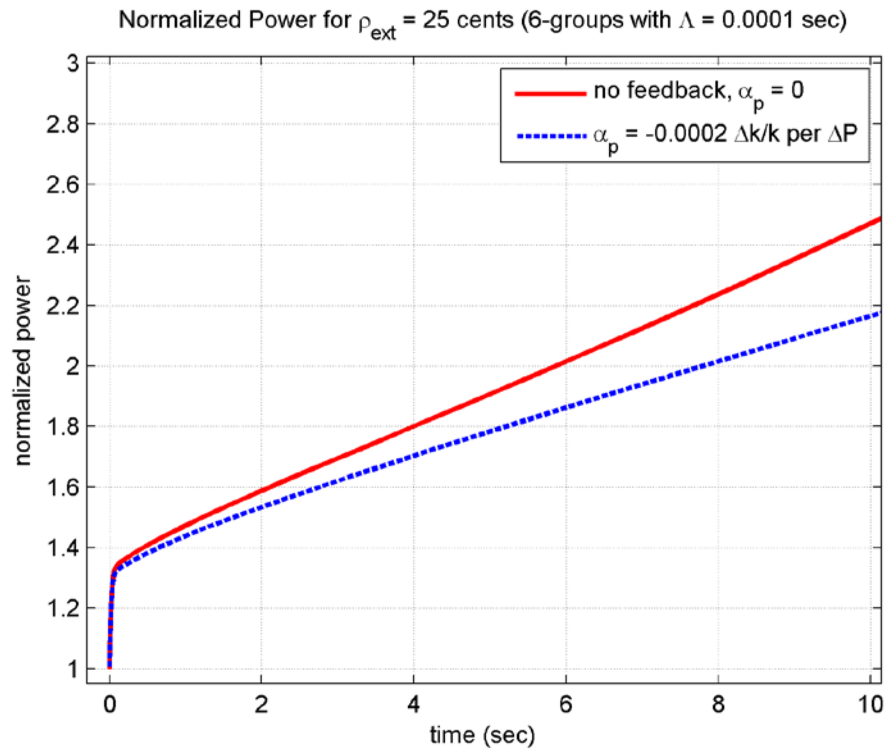


Fig. 7 P(t)/P$_o$ for ρ_{ext} = +25 cents (expanded view for 0 ≤ t ≤ 10 seconds).

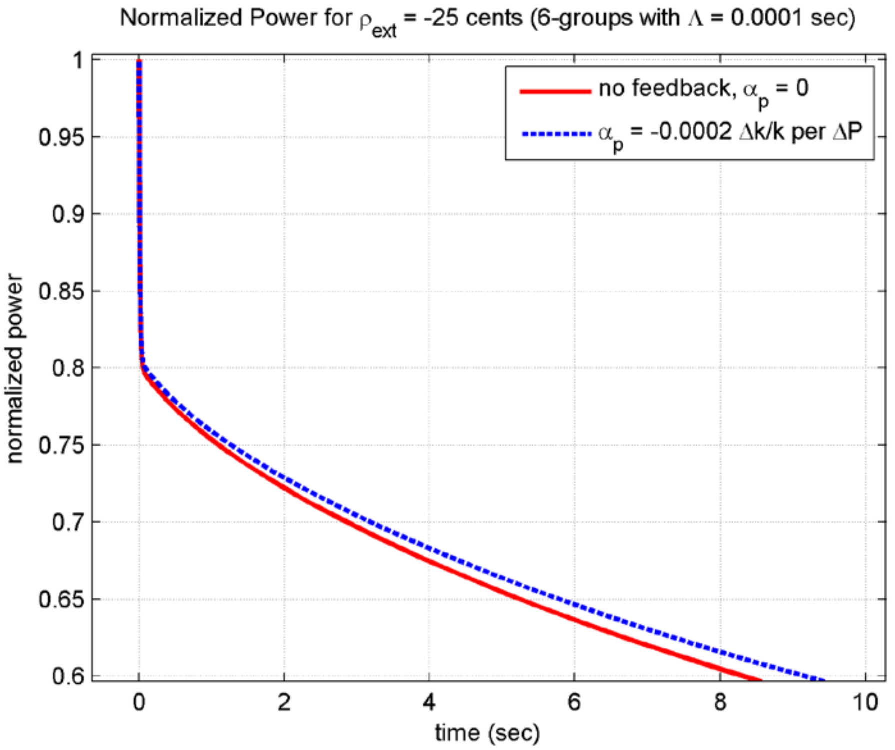


Fig. 8 P(t)/P$_o$ for ρ_{ext} = -25 cents (expanded view for 0 ≤ t ≤ 10 seconds).

with integrating factor

$$g(t) = e^{\int -\frac{(\rho-\beta)}{\Lambda}dt} = e^{-\frac{(\rho-\beta)}{\Lambda}t}$$

Now, multiplying by the integrating factor, gives

$$e^{-\frac{(\rho-\beta)}{\Lambda}t}\left(\frac{dP}{dt} - \frac{(\rho-\beta)}{\Lambda}P\right) = \frac{d}{dt}\left(e^{-\frac{(\rho-\beta)}{\Lambda}t}P(t)\right) = \frac{\beta}{\Lambda}P_o e^{-\frac{(\rho-\beta)}{\Lambda}t}$$

and multiplication by dt and integration give

$$e^{-\frac{(\rho-\beta)}{\Lambda}t}P(t) = \frac{\beta}{\Lambda}P_o\int e^{-\frac{(\rho-\beta)}{\Lambda}t}dt = \frac{\beta}{\beta-\rho}P_o e^{-\frac{(\rho-\beta)}{\Lambda}t} + C$$

or

$$P(t) = \frac{\beta}{\beta-\rho}P_o + Ce^{\frac{(\rho-\beta)}{\Lambda}t} \qquad (13)$$

where C is an arbitrary constant of integration.

Now, even without computing the constant C, we can argue that the exponential term containing C will vanish very quickly. Since $\rho < \beta$, the term $(\rho - \beta)/\Lambda$ is clearly negative. Also, since the generation time, Λ, is usually quite small, the coefficient in the exponent is usually fairly large, causing this term to decay very quickly.

As an example, let $\Lambda = 0.0001$ sec, $\beta = 0.0065$, and $\rho = 0.25\beta$. With these values, we have

$$\frac{(\rho-\beta)}{\Lambda} = \frac{(0.25-1)(0.0065)}{0.0001} = -48.75$$

and, in 0.20 sec, we have

$$e^{-48.75(0.20)} = 5.8\times10^{-5}$$

Thus, in about 0.2 seconds, this term is only about 0.006% of its original value.

Well, with the above argument, it is easy to see that a reasonable approximation for the prompt jump/drop is given by the first term in eqn. (13), or

$$\frac{P_1}{P_o} = \frac{\beta}{\beta-\rho} \qquad (14)$$

where P_1 is the power level just after the prompt jump/drop.

By way of example, for the simulation shown in Figs. 7 and 8, $\beta = 0.0065$ and $\rho = \pm0.25\beta$. Now, using the prompt jump/drop approximation in eqn. (14), we have

$$\frac{P_1}{P_o} = \frac{\beta}{\beta-\rho} = \frac{1}{1-0.25} = 1.33 \qquad \text{(for } \rho = +25 \text{ cents)}$$

and

$$\frac{P_1}{P_o} = \frac{\beta}{\beta - \rho} = \frac{1}{1 + 0.25} = 0.80 \qquad \text{(for } \rho = \text{-25 cents)}$$

and these values agree nearly exactly with the prompt jump and drop transients seen in Figs. 7 and 8.

Thus, for simple hand calculations, eqn. (12) and eqn. (14) combine to give a pretty good estimate of the power versus time behavior for a single step change in reactivity with no feedbacks.

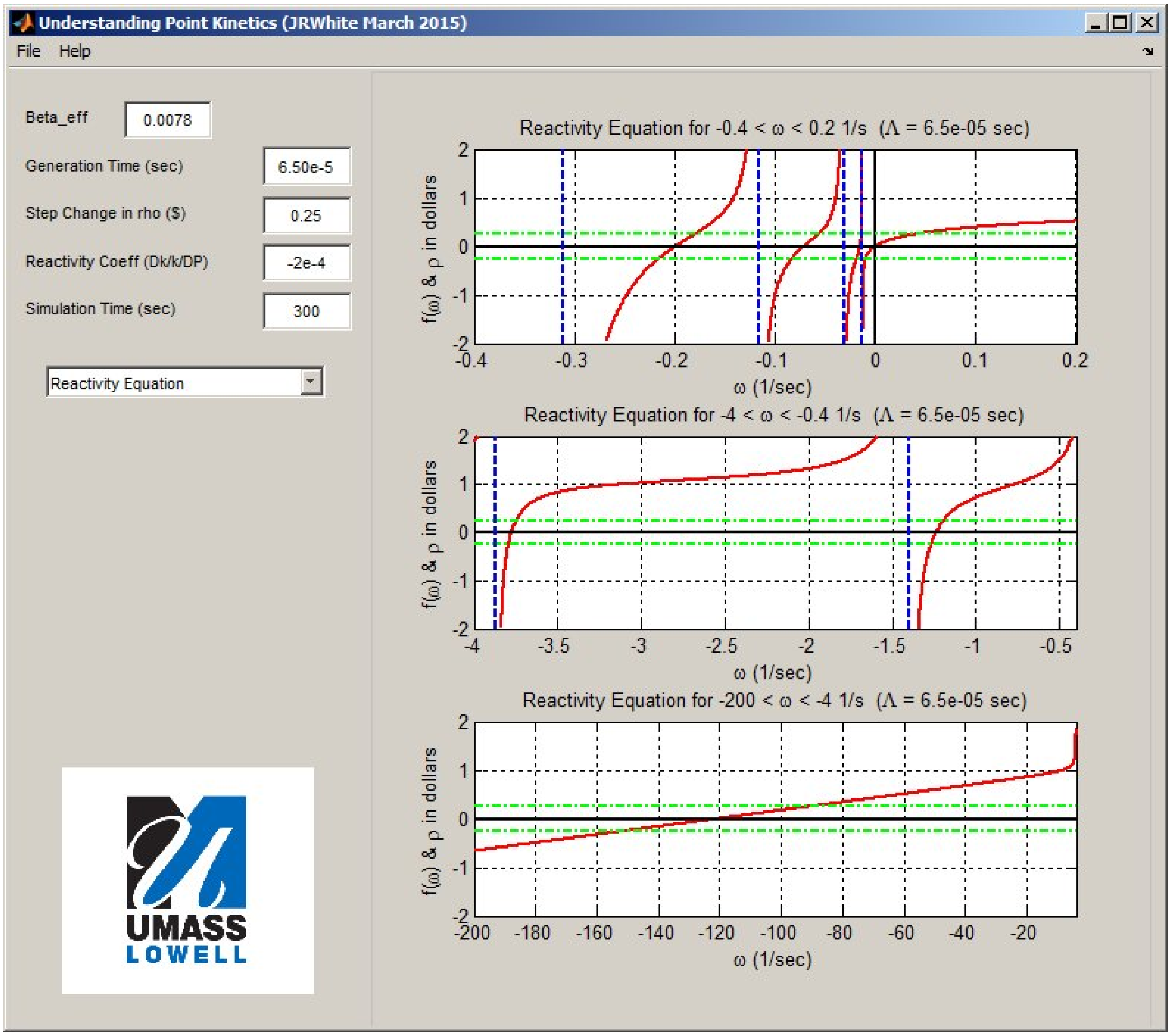
The kinetics_gui code

To complete this unit, we need to say a few words about the Matlab code that was used to generate the figures for use in this set of Lecture Notes. In particular, the Matlab code generated the plots associated with the visualization of the reactivity equation (Figs. 1 – 3), the reactor period vs. reactivity plot (Fig. 4), and the actual time-dependent normalized power profiles for both positive and negative step changes in reactivity (Figs. 5 – 8). The first two sets of plots were obtained by simply evaluating the reactivity equation [eqn. (6)] over appropriate ranges, and the normalized power profiles were generated by solving eqns. (1) and (2) numerically using Matlab's **ode15s** stiff ODE solver (stiff equations have widely varying time constants -- and the point kinetics equation is a perfect example of a very "stiff" system).

A graphical user interface was added to the computational part of the Matlab routines so that the student could easily run the Matlab program and readily change a variety of parameters. The final GUI version of the code is called **kinetics_gui**, and a view of the main interface is shown in Fig. 9. As apparent, the user can specify the effective delayed neutron fraction, β_{eff}, the prompt neutron generation time, Λ, the magnitude of the step change in reactivity, ρ, to use in the evaluations, the power reactivity coefficient, α_P, for the nonlinear simulations that include reactivity feedback, and the actual simulation time for the $P(t)/P_o$ computations. In addition, the user can select the plot type of interest, with the three plot choices as already indicated. The code is very straightforward and it is simple to use, but it gives a tremendous amount of insight into the workings of point kinetics. The student should exercise the code, varying a number of the kinetics parameters, as needed, to get a good understanding of the basics of point kinetics!!!

Summary

The goal here was to discuss the solution of the point kinetics equation and to give the student some real insight into the various behaviors that can be expected. In doing this, we have developed the so-called reactivity equation that is associated with a step change in reactivity, given a full discussion of its interpretation and use, simplified the estimation of the normalized power profiles using the concept of reactor period and the prompt jump/drop approximation, and actually solved the full generation time formulation of point kinetics for both the linear and nonlinear cases (i.e. no feedback and feedback cases). In addition, the **kinetics_gui** code was written so that each student can individually explore various scenarios in further detail. In all, this set of Lecture Notes tried to provide sufficient background in the theory and application of point kinetics so that the student gains a deeper understanding of its use in practical reactor operations scenarios -- hopefully these notes were successful in achieving this goal…



Beta_eff 0.0078

Generation Time (sec) 6.50e-5

Step Change in rho ($) 0.25

Reactivity Coeff (Dk/k/DP) -2e-4

Simulation Time (sec) 300

Reactivity Equation

Understanding Point Kinetics (JRWhite March 2015)
File
Help
Reactivity Equation for -0.4 < ω < 0.2 1/s (Λ = 6.5e-05 sec)
f(ω) & ρ in dollars
2
1
0
-1
-2
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
ω (1/sec)
Reactivity Equation for -4 < ω < -0.4 1/s (Λ = 6.5e-05 sec)
f(ω) & ρ in dollars
2
1
0
-1
-2
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
ω (1/sec)
Reactivity Equation for -200 < ω < -4 1/s (Λ = 6.5e-05 sec)
f(ω) & ρ in dollars
2
1
0
-1
-2
-200
-180
-160
-140
-120
-100
-80
-60
-40
-20
ω (1/sec)
UMASS
LOWELL

Fig. 9 Initial screen for the kinetics_gui program.

References

1. J. R. White, “One-Speed Point Kinetics Equations,” part of a series of Lecture Notes for the Nuclear Engineering Program at UMass-Lowell.
2. J. R. White, “Normalization of the Generation Time Formulation of the One-Speed Point Kinetics Equations,” part of a series of Lecture Notes for the Nuclear Engineering Program at UMass-Lowell.