24.536 Reactor Experiments and 407.403 Advanced Nuclear Lab

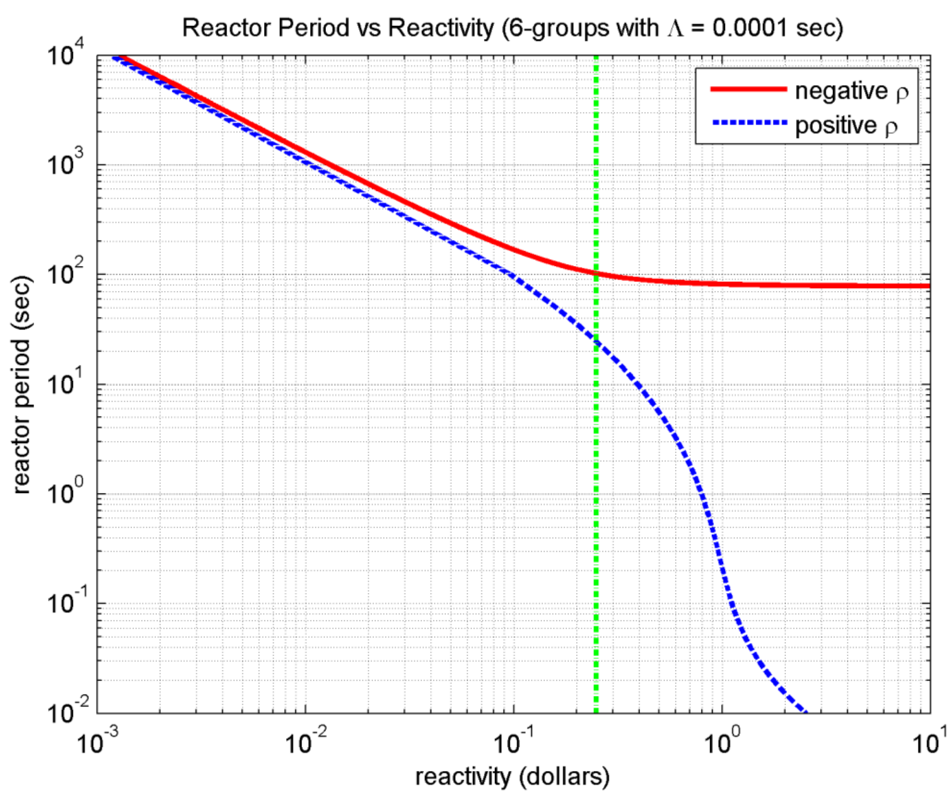
HW #6: "Reactivity Measurements" Pre-lab Exercises -- Solutions

Problem 1: In class, it was stated that, "for large reactivity changes, the **Stable Period Method** breaks down for both positive and negative ρ." Clearly explain this statement in some detail.

Directly from the Lecture Notes:

Finally, we should emphasize that, for large reactivity changes, the stable period method breaks down for both positive and negative ρ -- for quite different reasons. For large positive ρ, the reactor period becomes too small such that P(t) increases too rapidly, quickly causing an unsafe reactor condition that clearly must be avoided in all cases. In addition, even for moderate $+\rho$, the reactor power often approaches a level where we can no longer assume that the feedbacks are negligible. Thus, positive reactivity changes beyond about 0.10 dollars simply cannot be measured with the stable period method (actually there is no simple dynamic method that can be used to measure large positive changes in reactivity).

For large negative ρ, the situation is quite different. Here the reactor power is decreasing, so safety is not the concern. However, as discussed in Ref. 3 and as seen in the plot below, τ vs. ρ approaches a constant for large negative ρ values -- thus, beyond about -0.10 to -0.20 dollars, it is simply not possible to relate a unique combination of τ and ρ via eqn. (8).



Problem 2: Summarize briefly the region of applicability of the four methods discussed in the Lecture Notes -- that is, in what situation is each method applied?

Again directly from the Lecture Notes:

Asymptotic Period Technique -- used to measure small positive and negative reactivity changes from critical

Rod Drop Method -- used to measure large negative reactivity changes from critical

Source Jerk Method -- used to measure the subcriticality level, ρ_o, in a subcritical system

Subcritical Multiplication Factor Approach -- used to measure positive and negative reactivity changes within a subcritical system (ρ_o must be known)

Problem 3: The **Rod Drop** and **Source Jerk Methods** require numerical integration of the measured P(t) data. This exercise wants you to become familiar with the ***trapz*** routine in Matlab for performing the needed integrations. In particular, let's focus on a simple function that decays exponentially to zero as $t \rightarrow \infty$ -- say, for example, $f(t) = e^{-at}$, where a is a positive constant.

a. Integrate this function analytically from 0 to ∞ -- that is, what is the exact value of I, where

$$I = \int_0^\infty f(t)dt \qquad \text{for} \qquad f(t) = e^{-at}$$

for two different values of the constant a, say $a = a_1 = 0.5\ s^{-1}$ and $a = a_2 = 2.0\ s^{-1}$.

b. Now write a Matlab code to numerically integrate this function over the range $0 \le t \le 30$ seconds using the ***trapz*** function for the two different constants given above (note that the contribution to the integral beyond 30 seconds is negligible in both cases). Since ***trapz*** requires discrete vectors for the function, f(t), and independent variable, t, let's study the accuracy of the numerical integration versus the selected step size, Δt. In particular, let Δt take on the following four different values, **dt = [10 1 0.1 0.01]** seconds. Create discrete vectors for **t** and **f** and use ***trapz*** to perform the desired integral, I, for the four different step sizes and the two different values of the decay factor, a, and compare your results to the exact analytical solutions. Do things behave as expected? Explain your observations here…

Note that the sampling time for the data acquisition system in the UMLRR is 1 second. What does this say about our ability to measure fast negative transients within the UMLRR?

See code and printed results below…

```
      clear all, close all
%
      a = input('Entry desired value for the decay constant, a: ');
      if a < 0, a = -a;  end          % make sure constant is positive
      ft = @(t) exp(-a*t);            % function of interest
%
      dt = [10 1 0.1 0.01];  ftime = 30;
      Iexact = -(exp(-a*ftime)-1)/a;  % exact integral
      for j = 1:length(dt)
        t = 0:dt(j):ftime;   f = ft(t);
        I = trapz(t,f);                 % estimate of integral
        err = 100*(I-Iexact)/Iexact;    % % error
        fprintf(' For dt = %6.3f, I = %8.5f     %%error = %6.2f \n',dt(j),I,err);
      end
      fprintf('\n The exact value of I should be %8.5f !!!\n', Iexact);
```

```
Entry desired value for the decay constant, a: 0.5
 For dt = 10.000, I =  5.06783     %error = 153.39
 For dt =  1.000, I =  2.04149     %error =   2.07
 For dt =  0.100, I =  2.00042     %error =   0.02
 For dt =  0.010, I =  2.00000     %error =   0.00

 The exact value of I should be  2.00000 !!!
```

```
>> trapz_demo
Entry desired value for the decay constant, a: 2.0
 For dt = 10.000, I =  5.00000     %error = 900.00
 For dt =  1.000, I =  0.65652     %error =  31.30
 For dt =  0.100, I =  0.50167     %error =   0.33
 For dt =  0.010, I =  0.50002     %error =   0.00

 The exact value of I should be  0.50000 !!!
```

Because of the fixed sampling time of 1 sec that is used in the UMLRR data acquisition system, the very large and fast prompt drop associated with a large reactivity insertion (or the rapid removal of the source at subcritical) tends to give somewhat larger prediction errors. Although a smaller Δt between sampled P(t) data would lead to better results, a smaller sampling interval is not really practical for use within the real system -- we are stuck with the 1 second sampling time…

Problem 4: Here we want to illustrate the use of the **Stable Period Method**.

Using the **bw_display** GUI with the most recent blade worth data, estimate the reactivity change associated with the following two cases:

1. RegBlade moves from 10″ withdrawn to 13.0″ out
2. RegBlade moves from 10″ withdrawn to 5.5″ out

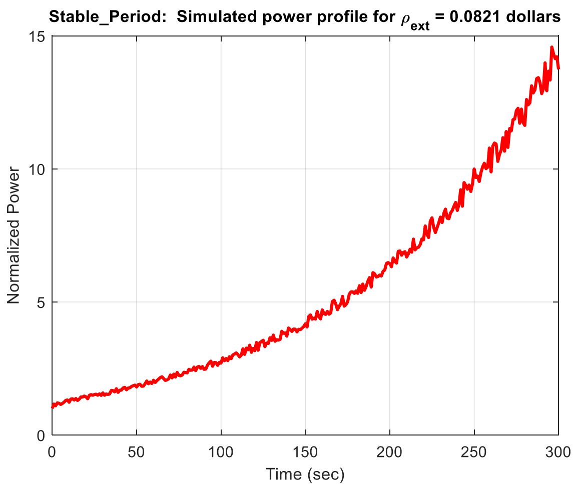
Now, use the ***rho_stable_period.m*** routine to simulate these reactivity perturbations in the UMLRR assuming that the system is initially critical, and use the simulated P(t) data within the **Stable Period Method** to "measure" the reactivity inserted, using the following parameters:

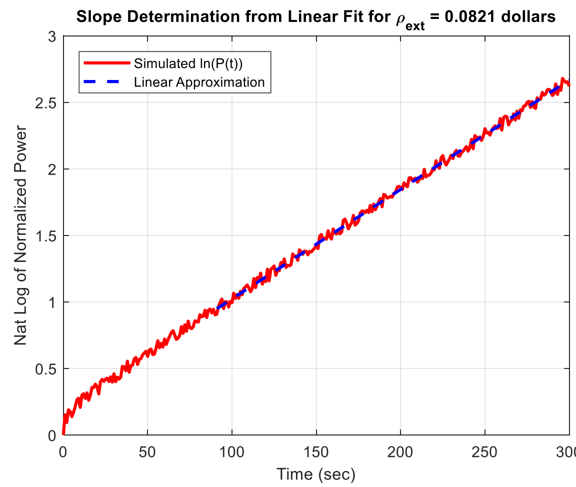
noise level: 5% short transient time: 90 seconds

Show and explain your results…

From bw_display GUI:

RegBlade 10″ → 13.0″ gives ρ = 0.064 %Δk/k = 0.00064 Δk/k = 0.0821 dollars

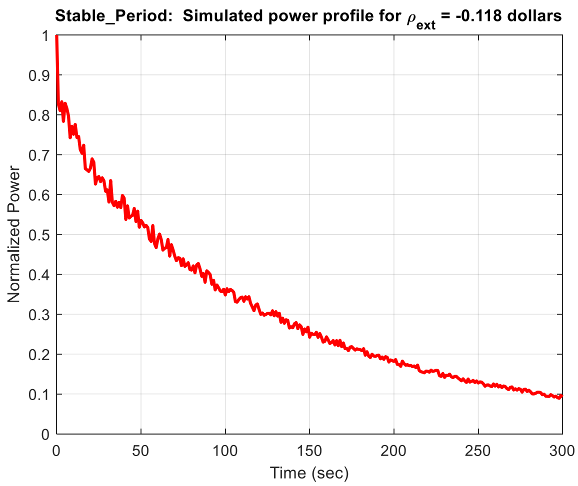


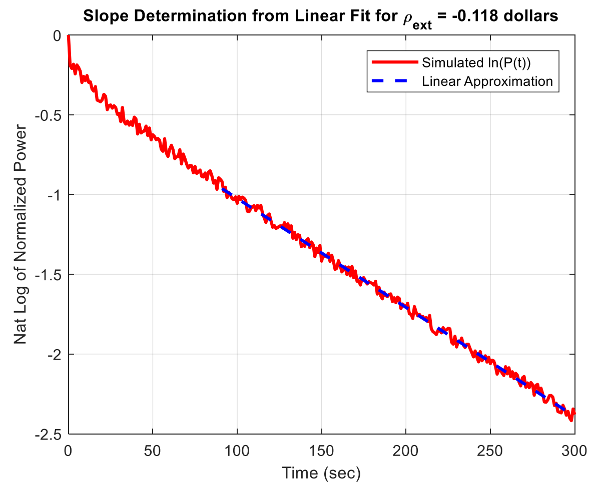



```
>> rho_stable_period

     Stable Period Method for Small Reactivity Change

  Enter small value of inserted reactivity in dollars: 0.0821

      Note: Actual rho used was  0.0821 dollars

      Measured stable period (sec):      121.64
      Measured rho (dollars):            0.0824
      Error in Predicted rho (%):          0.41
```

From bw_display GUI:

RegBlade 10″ → 5.5″ gives ρ = -0.092 %Δk/k = -0.00092 Δk/k = -0.118 dollars





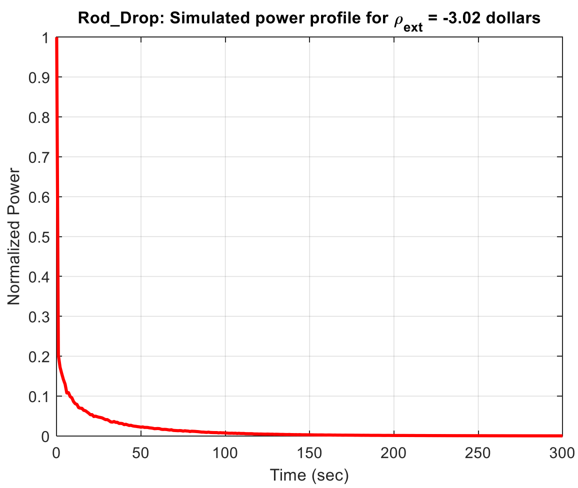

```
>> rho_stable_period

     Stable Period Method for Small Reactivity Change

  Enter small value of inserted reactivity in dollars: -0.118

      Note: Actual rho used was -0.1180 dollars

      Measured stable period (sec):     -146.41
      Measured rho (dollars):           -0.1241
      Error in Predicted rho (%):          5.19
```

Problem 5: Here we want to illustrate the use of the **Rod Drop Method**.

For a noise level of 5%, assume that **Control Blade 1** drops instantaneously from a position of 17.5″ withdrawn within a steady-state critical system. Simulate this event using the Matlab ***rho_rod_drop.m*** code and compare the "measured" worth relative to the actual reactivity used to initiate the transient. Show your results and briefly explain what was done here. Do your results make sense? Again use the **bw_display** GUI, as needed.

From bw_display GUI (with data from Jan. 2018):

Blade #1 17.5″ → 0″ gives ρ = -2.358 %Δk/k = -0.02358 Δk/k = -3.02 dollars




```
>> rho_rod_drop

   Rod Drop Method for Large Negative Reactivity
Change:

      Enter value of inserted reactivity in
dollars: -3.02

      Actual reactivity used ($):    -3.020
      Measured reactivity worth ($): -3.183
      Error in measured result (%):   5.41
```

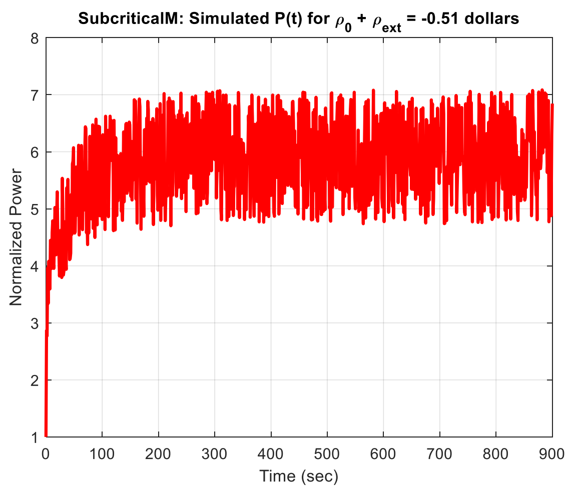
Problem 6: Here we illustrate the use of the **Subcritical Multiplication Factor Method**.

Assume that the UMLRR is subcritical by the amount of worth inserted by **Blade 1** in the previous problem (i.e. the worth of going from 17.5″ out to fully inserted). **Blade 1** is then pulled out quickly to 15″ withdrawn. Use the **bw_display** GUI and the ***rho_subcriticalM.m*** code to simulate and analyze this situation and to "measure" the worth associated with the movement of **Blade 1** from 0 to 15″ withdrawn. Again, show and explain your results. Assume a 20% noise level and that the ratio of count rates for the two steady-state subcritical configurations is the same as the power ratio in these states.

From bw_display GUI (with data from Jan. 2018):

Blade #1 0″ → 15″ gives ρ = 1.960 %Δk/k = 0.01960 Δk/k = 2.51 dollars

ρ_0 = -2.358 %Δk/k = -0.02358 Δk/k = -3.02 dollars (from previous problem)



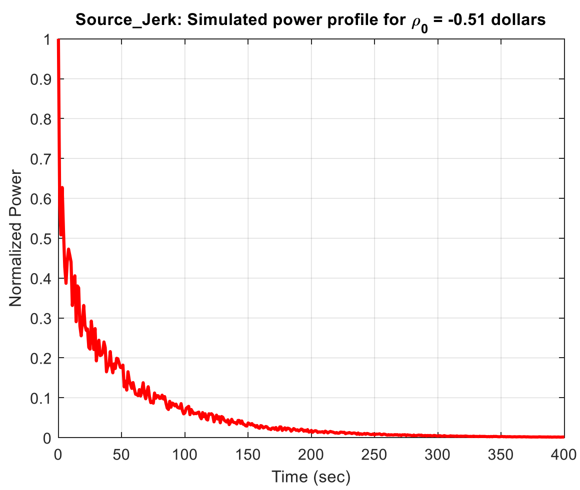

```
>> rho_subcriticalM

   Subcrit Mult Factor Method for Finding
Subcritical del_rho:

      Enter value of subcriticality level in
dollars: -3.02
      Enter the change in reactivity in dollars:
2.51

      Subcriticality level($):      -3.020
      Actual delrho used ($):        2.510
      Measured delrho ($):           2.513
      Error in measured delrho (%):  0.104
```

Problem 7: Here we want to illustrate the use of the **Source Jerk Method**.

Assume that the UMLRR is at steady state subcritical with **Blade 1** at 15″ withdrawn (that is, the endpoint of the previous problem simulation). Now the source is pulled out of the core very quickly. Use the ***rho_source_jerk.m*** code to simulate and analyze this situation and to "measure" the subcriticality level of the system with **Blade 1** at 15″ out. Again, show and explain your results and make the same assumptions as in the previous problem (i.e. a 20% noise level and that $C_1/C_0 = P_1/P_0$).

From Problems 5 and 6, we have

$$\boldsymbol{\rho_o = -2.358 + 1.960\ \%\Delta k/k = -0.398\ \%\Delta k/k = -0.00398\ \Delta k/k = -0.510\ \text{dollars}}$$




```
>> rho_source_jerk

   Source Jerk Method for Finding Subcriticality
Level:

      Enter value of subcriticality level in
dollars: -0.51

      Actual subcriticality ($):      -0.510
      Measured subcriticality ($):    -0.523
      Error in measured result (%):    2.60
```

Note: Problems 5-7 are related in that the end condition of one problem is the starting point of the next problem. In particular, the system is assumed to be just critical when **Blade 1** is dropped from 17.5″ to fully inserted (Problem 5). Then **Blade 1** is moved out to 15″ withdrawn (Problem 6). And, assuming steady-state at this condition, the source is pulled from the core (Problem 7). Understanding these relationships, makes these problem easier to interpret -- and this will be roughly the sequence of events that we perform in the live reactor lab during our upcoming Reactivity Measurements lab…