NUCL 511 HMWK 6

Alex Hagen

3/6/14

1 Write a program to solve the in-hour equation. Using this program and the data below, determine the sever roots for the step reactivity insertions of 1.2\$, 0.8\$, and −3.0\$.

$$\Lambda = 4.41190 \times 10^{-7}\ s$$

Group	β_k	$\lambda_k\ \left(s^{-1}\right)$
1	7.87173×10^{-5}	1.29660×10^{-2}
2	7.09826×10^{-4}	3.12874×10^{-2}
3	6.10649×10^{-4}	1.34616×10^{-1}
4	1.20866×10^{-3}	3.44560×10^{-1}
5	5.47426×10^{-4}	1.38307×10^{0}
6	1.65755×10^{-4}	3.76334×10^{0}

The inhour equation is given as

$$\rho = \alpha\Lambda + \sum_{k=1}^{6} \frac{\beta_k \alpha}{\alpha + \lambda_k}$$

with ρ is the reactivity, α is the inverse period, β_k is the delayed neutron fraction of group k, and λ_k is the decay constant of group k [2, p. 101]. The asymptotic solutions of this equation are important, and Figure 1 shows the equation, plotted. Notice that there are 6 distinct asymptotic branches of the equation.

To solve the inhour equation, a polynomial root finding numerical method must be used. Newton's method is often used for these cases, but as stated by Bradie [1, p. 101],

> "...for roots of multiplicity greater than one... ...the rate of convergence in these cases for Newton's method will be $O\left((1 - 1/m)^n\right)$. Note that for a root of multiplicity greater than two, this rate of convergence is slower than that of the bisection method."

So, for this case, the bisection method will be used in each asymptotic region. The bisection method involves setting an interval between which the root will occur. The interval is halved, and the half of the interval in which the root will occur then becomes the new interval. Convergence is then obtained by comparing the root estimated at the moment to the last iterations. When the root of the current iteration is within a certain amount, ε, of the old iteration, then the iteration is completed and root is found. Source Code 1 shows the MATLAB function written to perform the bisection method on the inhour equation, and Table 1 shows the results for the cases requested.

2 Inhour Equation

Table 1: Solutions of the Inhour Equation for Differing Reactivities

ρ	**1**	**2**	**3**	**4**	**5**	**6**
1.2\$	–	-2.2965	-0.6024	-0.0375	-0.0313	–
0.8\$	–	-1.3831	-0.8493	-0.1696	-0.0313	1.5274
−3.0\$	–	-3.7180	-1.3271	-0.1270	-0.0291	–

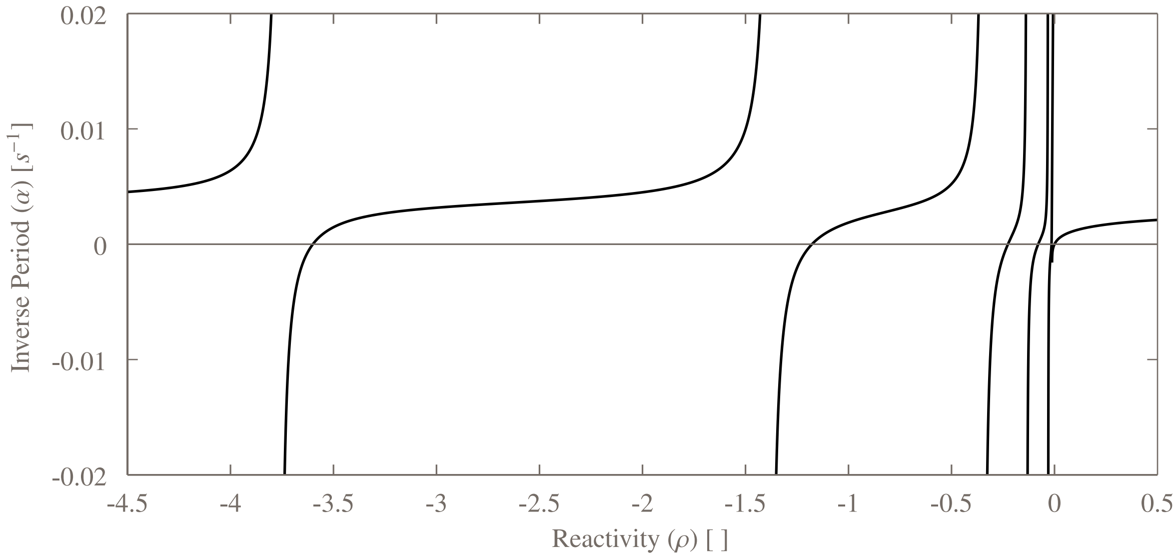

Figure 1: Inhour Equation Asymptotic Solutions Plotted

(1) Find the stable and prompt-period branches for ^{235}U as fuel and $\Lambda = 10^{-4}$, 10^{-5}, and 4×10^{-7} s (data given in Lecture note 2). The prompt-period and stable branches correspond to the branches with

$$\rho = \alpha\Lambda + \frac{\beta_6 \alpha}{\alpha + \lambda_6}$$

and

$$\rho = \alpha\Lambda + \frac{\beta_1 \alpha}{\alpha + \lambda_1}$$

(2) Find $\rho(\alpha)$ in the one-delay-group approximation with $\lambda = \overline{\lambda}$. For the one delay group approximation, we must use $\beta = \sum \beta_k$, and $\lambda = \overline{\lambda}$, which leads to the equation

$$\rho = \alpha\Lambda + \frac{\beta\alpha}{\alpha + \overline{\lambda}}$$

(3) Plot both results outside of the range of the singularities, with a shadowed area indicating the singularities. Figure 2 shows this plot.

(4) Discuss the comparison. The one delay group approximation approaches the values for the prompt and stable periods, but has only one singularity. It also provides values that would be wildly incorrect in the region of singularities. For very large negative or large positive periods, the one delay group approximation is a good approximation, but anywhere near the region of singularities, it would give bad results.

3 Find $\rho(\alpha)$ for $\alpha < 0.1$ s^{-1} (for the same three Λ values as problem 2a). Plot and discuss the comparison of the results with the approximate formula given in the text for very small α values. Extend the comparison to negative α. The inhour equation with very samll α can be estimated by neglecting α in the denominator of the inhour equation [2, p. 106].

$$\rho = \alpha\Lambda + \alpha \sum \frac{\beta_k}{\lambda_k}$$

Figure 3 shows this plot. The approximation follows the general trend of the six delay group solution, but does not take into account any of the singularities. In any of the regions away from the singularities, this approximation would be accurate, but near singularities, this approximation would give incorrect results.

and

Source Code 1 Inhour Solver using Bisection Method

```
function [alpha]=inhour(rho,Lambda,betak,lambdak)
    % make a contianer for the solution
    alpha=zeros(1,6);
    % Set the lower and upper bounds for each asymptotic region
    boundlow=[-500 -2.853 -0.8495 -0.3028 -0.0327 0.0133];
    boundhigh=[-2.854 -0.8496 -0.3029 -0.0328 0.0132 500];
    % for each asymptotic solution
    for i=1:6
        low=boundlow(i);
        high=boundhigh(i);
        vallow=reactivityeqn(low,Lambda,betak,lambdak)-rho;
        valhigh=reactivityeqn(high,Lambda,betak,lambdak)-rho;
        oldalpha=low;
        epsilon=1;
        % until error between iterations is less than 1E-6
        while epsilon > 1E-6
            % find the middle coordinate and its value
            mid=(low+high)/2;
            valmid=reactivityeqn(mid,Lambda,betak,lambdak)-rho;
            % case that we've accidentally landed on a root, make root
            % equal to that value
            if (abs(vallow)<1E-6)
                alpha(i)=low; break;
            end
            if (abs(valhigh)<1E-6)
                alpha(i)=high; break;
            end
            if (abs(valmid)<1E-6)
                alpha(i)=mid; break;
            end
            % case that the root crosses in between low and mid, switch
            % high down to mid
            if (vallow/abs(vallow) ~= valmid/abs(valmid))
                high=mid;   valhigh=valmid;
            % case that the root crosses in between high and mid, switch
            % low up to mid
            elseif (valhigh/abs(valhigh) ~= valmid/abs(valmid))
                low=mid; vallow=valmid;
            % case that the root doesn't cross the interval, make the
            % current alpha equal to NaN
            elseif ((valhigh/abs(valhigh) == valmid/abs(valmid)) && ...
                    (valhigh/abs(valhigh) == vallow/abs(vallow)))
                alpha(i)=NaN; break;
            end
            % determine convergence
            alpha(i)=mid;
            epsilon=abs(alpha(i)-oldalpha);
            oldalpha=alpha(i);
        end
    end
end
% anonymous function to determine the roots
function [rho]=reactivityeqn(alpha,Lambda,betak,lambdak)
    rho = alpha * Lambda + sum(betak*alpha./(alpha+lambdak));
end
```

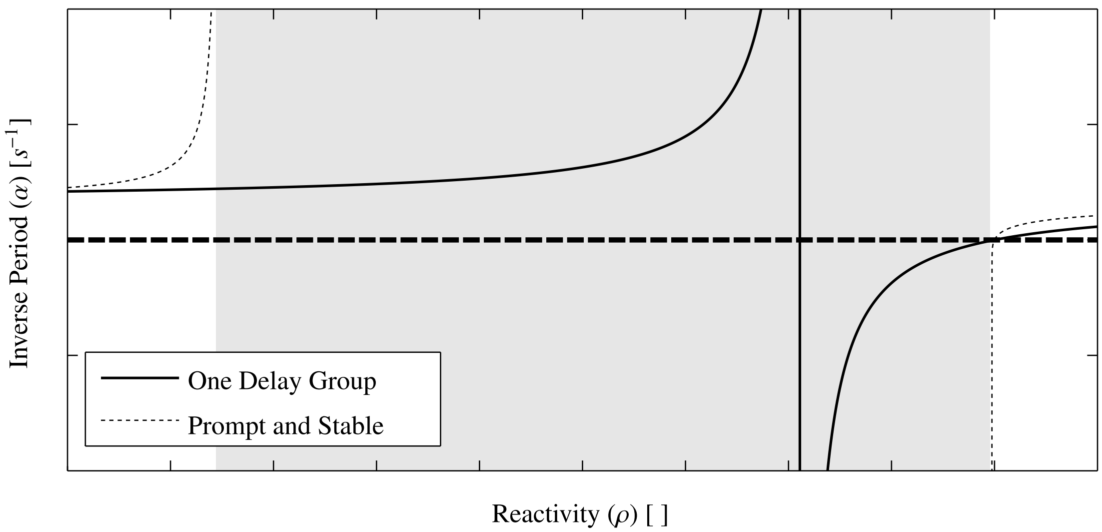

Figure 2: One Delay Group Approximation of Inhour Equation Versus Inhour Solutions for Prompt and Stable Periods

References

[1] Brian Bradie. *A Friendly Introduction to Numerical Analysis*. Pearson Prentice Hall, Upper Saddle River, New Jersey, 2006.

[2] K Ott and R Neuhold. *Introductory Nuclear Reactor Dynamics*. American Nuclear Society, La Grange Park, Illinois, 1985.

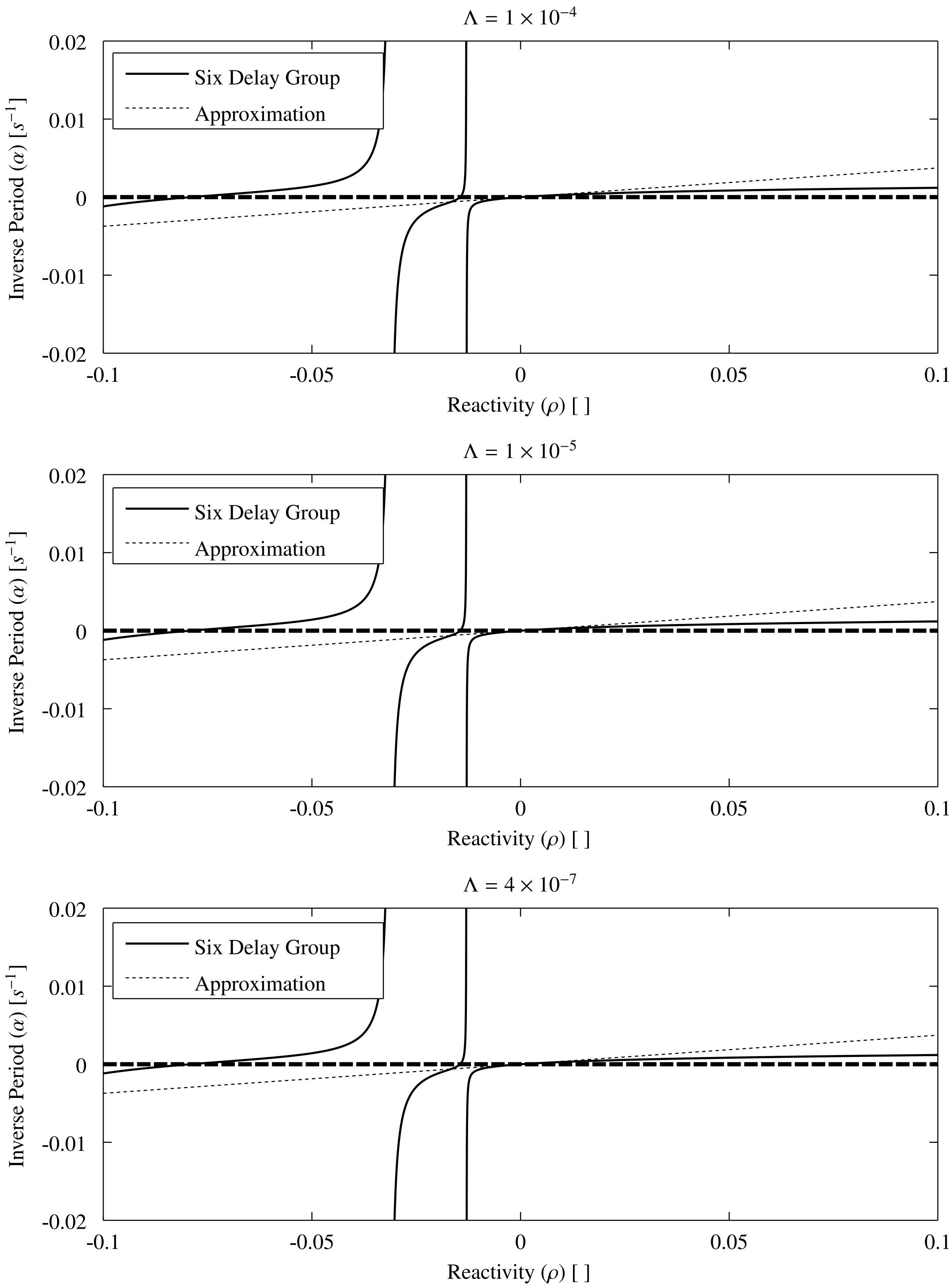

Figure 3: Approximation and Six Delay Group Solution for Very Small Periods