NUCL 511 Nuclear Reactor Theory and Kinetics

Homework #6

Due March 6

1. Write a program to solve the in-hour equation. Using this program and the data below, determine the seven roots for the step reactivity insertions of 1.2\$, 0.8\$, and -3.0\$. (50 points)

Λ (s)	4.41190E-07	
Group	β_k	λ_k (s^{-1})
1	7.87173E-05	1.29660E-02
2	7.09826E-04	3.12874E-02
3	6.10649E-04	1.34616E-01
4	1.20866E-03	3.44560E-01
5	5.47426E-04	1.38307E+00
6	1.65755E-04	3.76334E+00

Answer) A FORTRAN program to calculate the reactivity for given prompt inverse period or to determine the prompt inverse periods for given reactivity values by solving the in-hour equation is given in the attachment. Using this program, the prompt inverse periods can be obtained as

Reactivity	1.2\$	0.8\$	-3.0\$
1	1.50835E+03	1.52741E+00	-1.28578E-02
2	-1.31906E-02	-1.32827E-02	-2.91765E-02
3	-3.75329E-02	-4.10217E-02	-1.27030E-01
4	-1.58061E-01	-1.69638E-01	-3.12379E-01
5	-6.02382E-01	-8.49255E-01	-1.32714E+00
6	-2.29652E+00	-3.26116E+00	-3.71800E+00
7	-5.41798E+00	-1.50835E+03	-3.01099E+04

2. In-hour equation (10 points)

   a. 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 the lecture note 2).

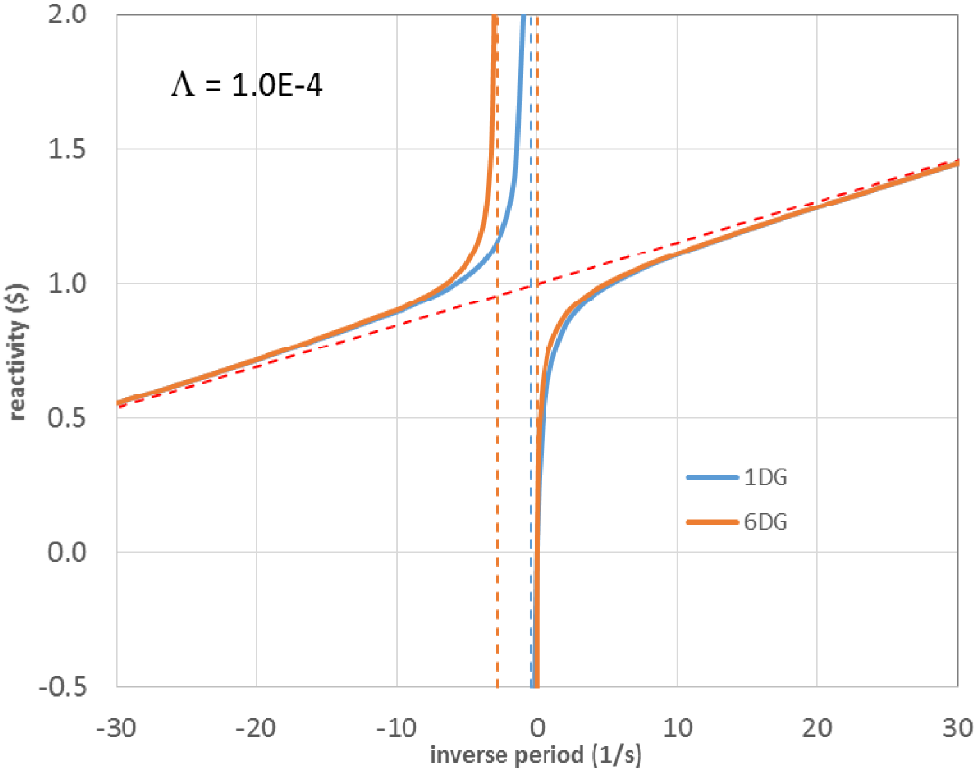
   Answer) The stable branch is the one for $\alpha > -\lambda_1 = -0.0133\ \mathrm{s}^{-1}$ and the prompt period branch is the one for $\alpha < -\lambda_6 = -2.853\ \mathrm{s}^{-1}$.

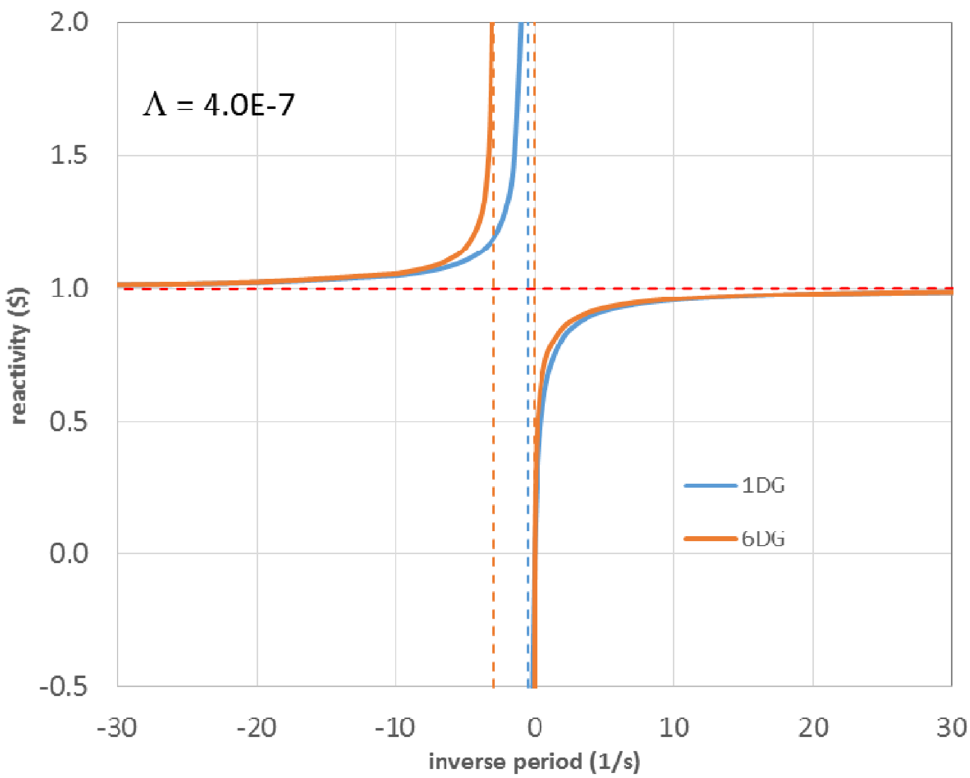
   b. Find $\rho(\alpha)$ in the one-delay-group approximation with $\lambda = \bar{\lambda}$.

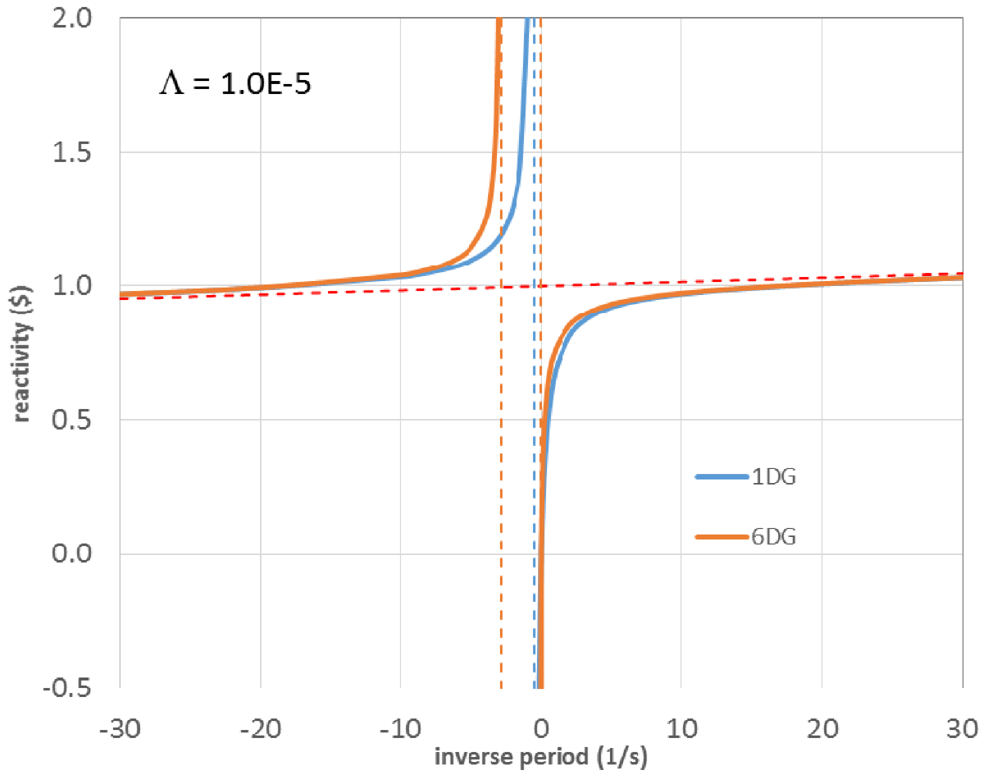
   Answer) Using the data in the lecture note 2, $\bar{\lambda} = \sum_k \beta_k \lambda_k / \beta = 0.4688\ \mathrm{s}^{-1}$. Thus $\rho(\alpha)$ is given by

$$\rho(\alpha) = \Lambda\alpha + \beta - \frac{\bar{\lambda}\beta}{\alpha + \bar{\lambda}} \tag{1}$$

c. Plot both results outside of the range of the singularities, with a shadowed area indicating the singularities.







d. Discussion the comparison.

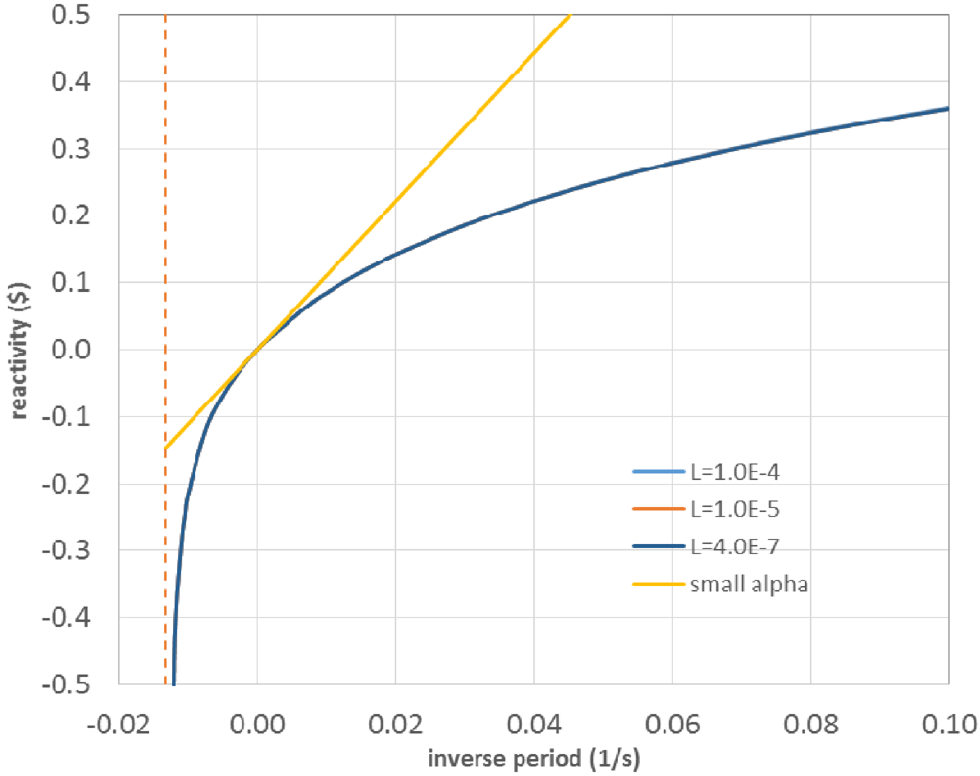
Answer) The prompt inverse period of the one-delay-group approximation is less negative than that the six-delay-group model. Thus, the one-delay-group model yields a slower decay of prompt neutrons than the six-group model. One the other hand, the one-group approximation yields a large inverse period for the stable branch, resulting in a more rapid increase of power amplitude for a positive reactivity insertion or a slower decrease for a negative reactivity insertion.

3. Find $\rho(\alpha)$ for $\alpha < 0.1\,/\mathrm{s}$ for the same three Λ values as problem 1a). Plot and discuss the comparison of the results with the approximate formula given in the test for very small α values. Extend the discussion to negative α. (10 points)

Answer) For very small α values, the in-hour equation can be approximated as

$$\begin{aligned}\rho(\alpha) &= \Lambda\alpha + \beta - \sum_k \frac{\lambda_k \beta_k}{\alpha + \lambda_k} = \Lambda\alpha + \beta - \sum_k \frac{\beta_k}{1+\alpha/\lambda_k} \approx \Lambda\alpha + \beta - \sum_k \beta_k \left(1 - \frac{\alpha}{\lambda_k}\right) \\ &= \Lambda\alpha + \sum_k \beta_k \frac{\alpha}{\lambda_k} = \alpha\left(\Lambda + \frac{\beta}{\bar{\lambda}_{in}}\right) = \alpha(\Lambda + 0.0720)\end{aligned} \tag{2}$$

This approximation is valid only when $\alpha << 0.0132$ since $\lambda_1 = 0.0132$. The first order Taylor expansion used in Eq. (2) is valid even for negative α as far as $|\alpha| << \lambda_1$. So the approximate formula for very small α values works well in the region where $|\rho| < 0.1\$$.


0.5
0.4
0.3
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
reactivity ($)
-0.02
0.00
0.02
0.04
0.06
0.08
0.10
inverse period (1/s)
L=1.0E-4
L=1.0E-5
L=4.0E-7
small alpha

NUCL 511 Nuclear Reactor Theory and Kinetics

Attachment: FORTRAN Program to Solve In-hour Equation

```
      program  inhour
!
!     Solve the inhour equation for a given reactivity
!
      double precision lambda(6),beta(6)
      double precision gent,rho,alpha,tbeta,amean,hmean
      double precision sum1,sum2,crho,temp
      integer iptype,ndg,k,n
!
!     read the generation time
!
      read(5,*) gent
!
!     read the number of delayed neutron groups
!
      read(5,*) ndg
!
!     read delayed neutron fractions and decay constants
!
      do k=1,ndg
         read(5,*) beta(k),lambda(k)
      enddo
!
!     total delayed neutron fraction, and arithematic and harmonic
!     means of decay constants
!
      tbeta=0.0
      do k=1,ndg
         tbeta=tbeta+beta(k)
      enddo
!
      sum1=0.0
      sum2=0.0
      do k=1,ndg
         sum1=sum1+beta(k)*lambda(k)
         sum2=sum2+beta(k)/lambda(k)
      enddo
      amean=sum1/tbeta
      hmean=tbeta/sum2
!
!     print input data
!
      write(6,1000) gent
 1000 format('Generation time (s) =',1pe12.5)
      write(6,1010) ndg
 1010 format('Number of delayed neutron groups =',i3)
      write(6,1020)
 1020 format('group',1x,'    beta    ',1x,'   lambda   ')
      do k=1,ndg
         write(6,1030) k,beta(k),lambda(k)
      enddo
```

```
1030 format(i3,2x,2(1x,1pe12.5))
     write(6,1040) tbeta
1040 format(' sum  ',1pe12.5)
     write(6,1050) amean
1050 format('Arithmetic average of decay constants (1/s) =',1pe12.5)
     write(6,1060) hmean
1060 format('Harmonic average of decay constants (1/s)   =',1pe12.5)
!
!    divide the generation time and delayed neutron fractions
!    by the total delayed neutron fraction
!
     gent=gent/tbeta
     do k=1,ndg
       beta(k)=beta(k)/tbeta
     enddo
!
!    read problem type and reactivity or inverse period
!      iptype - problem type
!         0 = determine the inverse periods (1/s) for a given
!            reactivity in $
!         1 = calculates the reactivity in $ for a given inverse
!            period (1/s)
!
     read(5,*) ncases
!
!    loop over cases
!
     do n=1,ncases
       read(5,*) iptype,temp
!
!      calculate inverse period or reactivities
!
       if (iptype.eq.0) then
         rho=temp
         call calpha(beta,lambda,gent,rho,ndg)
       else if (iptype.eq.1) then
         alpha=temp
         rho=crho(beta,lambda,gent,alpha,ndg)
!
         write(6,1070) alpha
         write(6,1080) rho
1070     format(/,'* Given inverse period (1/s) =',1pE12.5)
1080     format('  Calculated reactivity ($)  =',1pe12.5)
       endif
     enddo
!
     stop
     end
!
     subroutine calpha(beta,lambda,gent,rho,ndg)
!
!    calculate the inverse period for a given reactivity ($)
!
     double precision beta(6),lambda(6)
```

```
      double precision gent,rho,root1,root2,drho,large,small
      integer ndg,m,k
      integer count0,count1,crate,cmax
!
      data large/1.0d+15/,small/0.001/
!
      write(6,1010) rho
      write(6,1020)
 1010 format(/,'* Given reactivity ($)  =',1pe12.5)
 1020 format('         inverse     no. of    residual',/,&
             '   no.  period (1/s) iteration   reactivity')
 1030 format(i5,2x,1pe12.5,1x,0pi6,5x,1pe12.5)
!
      call system_clock(count0,crate,cmax)
!
      m=1
      if (rho.ge.1.0+small) then
        root1=(rho-1.0)/gent
      else if (rho.lt.1.0-small) then
        root1=-(1.0+beta(m)/(rho-1.0))*lambda(m)
      else
        root1=-0.5*lambda(m)+dsqrt(lambda(m)*beta(m)/gent)
      endif
      call findroot(beta,lambda,gent,rho,-lambda(m),large,&
                    root1,root2,drho,ndg,iter)
      write(6,1030) m,root1,iter,drho
!
      do m=2,ndg
        if (rho.ge.1.0+small) then
          root1=-(1.0+beta(m-1)/(rho-1.0))*lambda(m-1)
        else if (rho.lt.1.0-small) then
          root1=-(1.0+beta(m)/(rho-1.0))*lambda(m)
        else
          root1=-0.5*(lambda(m-1)+lambda(m))
        endif
        call findroot(beta,lambda,gent,rho,-lambda(m),-lambda(m-1),&
                      root1,root2,drho,ndg,iter)
        write(6,1030) m,root1,iter,drho
      enddo
!
      m=ndg+1
      if (rho.ge.1.0+small) then
        root1=-(1.0+beta(m-1)/(rho-1.0))*lambda(m-1)
      else if (rho.lt.1.0-small) then
        root1=(rho-1.0)/gent
      else
        root1=-0.5*lambda(m-1)-dsqrt(lambda(m-1)*beta(m-1)/gent)
      endif
      call findroot(beta,lambda,gent,rho,-large,-lambda(m-1),&
                    root1,root2,drho,ndg,iter)
      write(6,1030) m,root1,iter,drho
!
      call system_clock(count1,crate,cmax)
      write(6,1040) real(count1-count0)/real(crate)
```

```
 1040 format(/,'   elapsed time (sec) =',1pe12.5)
!
      return
      end
!
      subroutine findroot(beta,lambda,gent,rho,bl,bu,&
                      root1,root2,drho1,ndg,iter)
!
!     find a root for a brach of the inhour equation based on
!     1) the Newton-Rapson method, 2) the secant method, and
!     3) the bisection method
!
      double precision beta(6),lambda(6)
      double precision gent,rho,bl,bu,root1,root2,slope1
      double precision eps,small,drho1,drho2
      double precision troot,trho
      integer maxitr,k
!
      data eps/1.0d-15/,small/0.00001/,maxitr/50/
!
      iter=0
      drho1=crho(beta,lambda,gent,root1,ndg)-rho
!
      do while (dabs(drho1).gt.eps .and. iter.lt.maxitr)
        iter=iter+1
        slope1=gent
        do k=1,ndg
          slope1=slope1+beta(k)*lambda(k)/(root1+lambda(k))**2
        enddo
        troot=root1-drho1/slope1
        if (troot.lt.bl .or.troot.gt.bu) then
          if (iter.eq.1) then
            if (drho1.gt.0.0) then
              root2=bl+small
            else
              root2=bu-small
            endif
            drho2=crho(beta,lambda,gent,root2,ndg)-rho
          endif
          troot=root1-drho1*(root2-root1)/(drho2-drho1)
          if (troot.gt.bu) then
            troot=bu-small
          else if (troot.lt.bl) then
            troot=bl+small
          endif
        endif
        root2=root1
        drho2=drho1
        root1=troot
        drho1=crho(beta,lambda,gent,root1,ndg)-rho
      enddo
!
      return
      end
```

```
!
      function crho(beta,lambda,gent,alpha,ndg)
!
!     calculate the reactivity ($) for a given alpha
!
      double precision beta(6),lambda(6)
      double precision gent,alpha,crho
      integer ndg,k
!
      crho=gent*alpha+1.0
      do k=1,ndg
         crho=crho-lambda(k)*beta(k)/(alpha+lambda(k))
      enddo
!
      return
      end
```