Analysis of Reactor Static Characteristics

2008 년 2 학기

Project 1: Collision Probability Code

Due Oct. 14, 2008

The objective of this project is to write and examine a collision probability code applicable for 1-D cylindrical geometry problems. The input data that specify the problem consist of the following:

nr,ngauss : number of rings, number of Gauss points
sigt(1:nr) : total (transport corrected) cross section for each region
sigs(1:nr) :scattering (transport corrected and self scattering) cross section for each region
q(1:nr) :source at each region
alb,jext :albedo and external incoming current

The code should have modules for 1) an input processing, 2) memory allocation and variable initialization, 3) collision probability kernel calculation. 4) a direct linear system solution based on LU factorization, 5) flux and current calculation, and 6) edit generation. Among those, the collision probability kernel and the direct linear solver should be made as efficiently as possible because these modules are executed repeatedly in an integral transport theory based lattice physics code. The performance of your code will be compared with the reference code which was written in FORTRAN for 1 million times repeated calculation with different cross section perturbation for each time. Use double precision arithmetic for all floating point operations.

Part 1. Code Module Preparation

1. Write a collision probability routine that generates the collision probability kernel, *p(i,j)* given the cylindrical geometry and cross section data. Include the first flight blackness (γ_i) as another output of the routine. You need to be very careful in order not to waste computation or memory in this routine. Use the number of Gauss points specified in the input. The 3-rd order Bickeley function needed for the collision probability function is attached at the end. The Gauss-Jacobi quadrature is also given.

2. Write a LU factorization routine which incorporates partial pivoting so that the resulting L, U factors are for the permuted matrix, namely: $PA = LU$. Note that the permutation matrix is for row exchange which corresponds to changing the order of equations. Then

also write the forward and backward substitution module that uses the L and U matrices and the row exchange information vector to solve a linear system given a right hand side vector.

3. Write a routine that determines the flux factors due to the source and source current, respectively. This will involve the solution of (n+1) linear systems.

4. Write a routine that determines the flux at each region and currents at the external boundary. Generate the edit shown at the end.

Part 2. Performance Examination

1. Solve the two problems attached. The result should be the same as the reference output.

2. Try the normal Gauss quadrature instead of the Gauss-Jacobi quadrature and examine the difference in the accuracy of the solution.

3. Implement a small driver that performs the same calculation 100,000 times. Incorporate random perturbation in the given removal cross section. Your code will be examined on a common platform for comparison with other people's code.

Part 3. Physics Examination

1. Try different absorption cross section to simulate various levels of resonance. Then plot the flux shapes to compare different extents of spatial self shielding. Determine the cross sections on your own according to your own logic.

2. Write a report that describes the method, the coding, and the analysis result in a self-contained form, meaning that the report can be understood later without resorting to the textbook or the class note.

```
      FUNCTION KI3(XX)
!
      use global
#include <global.h>
!
      real(NBF) KI3
C                                                     **********
C       /HEBE   , 02 Jan 1996 ----- By Rudi Stamm'ler       *       *
C                                                     **********
C   ACCURACY 1.0E-5, UP TO XX=9.0.
C ---------------------------------------------------------------------
C

      X   =ABS(XX)
      IF(X.GT.0.99999)  GO TO 16
      I  =INT(20.0*X+1.00001)
      GO TO (1,2,3,4,5,6,7,8,9,10,11,11,12,12,13,13,14,14,15,15),I
C                       ** RANGE 0.00-0.05 **
    1 KI3=(.7266088*X-.9990226)*X+.7853961
      RETURN
C                       ** RANGE 0.05-0.10 **
    2 KI3=(.6466375*X-.9912340)*X+.7852024
      RETURN
C                       ** RANGE 0.10-0.15 **
    3 KI3=(.5856605*X-.9791293)*X+.7845986
      RETURN
C                       ** RANGE 0.15-0.20 **
    4 KI3=(.5346648*X-.9638914)*X+.7834577
      RETURN
C                       ** RANGE 0.20-0.25 **
    5 KI3=(.4907827*X-.9463843)*X+.7817094
      RETURN
C                       ** RANGE 0.25-0.30 **
    6 KI3=(.4521752*X-.9271152)*X+.7793031
      RETURN
C                       ** RANGE 0.30-0.35 **
    7 KI3=(.4177388*X-.9064822)*X+.7762107
      RETURN
C                       ** RANGE 0.35-0.40 **
    8 KI3=(.3869945*X-.8849865)*X+.7724519
      RETURN
C                       ** RANGE 0.40-0.45 **
    9 KI3=(.3590753*X-.8626685)*X+.7679903
      RETURN
C                       ** RANGE 0.45-0.50 **
   10 KI3=(.3338676*X-.8400133)*X+.7628988
      RETURN
C                       ** RANGE 0.50-0.60 **
   11 KI3=(.2998569*X-.8054172)*X+.7540982
      RETURN
C                       ** RANGE 0.60-0.70 **
   12 KI3=(.2609154*X-.7587821)*X+.7401279
      RETURN
C                       ** RANGE 0.70-0.80 **
   13 KI3=(.2278226*X-.7125290)*X+.7239594
      RETURN
C                       ** RANGE 0.80-0.90 **
   14 KI3=(.1994999*X-.6672761)*X+.7058777
      RETURN
C                       ** RANGE 0.90-1.00 **
   15 KI3=(.1751248*X-.6234536)*X+.6861762
      RETURN
C
   16 IF(X.GT.9.0) GO TO 160
      I   =INT(2.5*(X-0.99998)) +1
      GO TO (17,18,19,20,21,22,23,24,25,26,
     &      27,27,28,28,29,29,30,30,31,31,31),I
  160 KI3=0.0
      RETURN
C                        ** RANGE 1.0-1.4 **
```

```
   17 KI3=((-.05337485*X+.3203223)*X-.7538355)*X+.7247294
      RETURN
C                       ** RANGE 1.4-1.8 **
   18 KI3=((-.03146833*X+.2295280)*X-.6279752)*X+.6663720
      RETURN
C                       ** RANGE 1.8-2.2 **
   19 KI3=((-.01906198*X+.1631667)*X-.5094124)*X+.5956163
      RETURN
C                       ** RANGE 2.2-2.6 **
   20 KI3=((-.01174752*X+.1152418)*X-.4046007)*X+.5191031
      RETURN
C                       ** RANGE 2.6-3.0 **
   21 KI3=((-.007328415*X+.08097913)*X-.3159648)*X+.4425954
      RETURN
C                       ** RANGE 3.0-3.4 **
   22 KI3=((-.004617254*X+.05669960)*X-.2434341)*X+.3703178
      RETURN
C                       ** RANGE 3.4-3.8 **
   23 KI3=(.007923547*X-.07158569)*X+.1684022
      RETURN
C                       ** RANGE 3.8-4.2 **
   24 KI3=(.005095111*X-.05016344)*X+.1278307
      RETURN
C                       ** RANGE 4.2-4.6 **
   25 KI3=(.003286040*X-.03501524)*X+.09611422
      RETURN
C                       ** RANGE 4.6-5.0 **
   26 KI3=(.002126242*X-.02437465)*X+.07170491
      RETURN
C                       ** RANGE 5.0-5.8 **
   27 KI3=(.001123687*X-.01425519)*X+.04616317
      RETURN
C                       ** RANGE 5.8-6.6 **
   28 KI3=(4.762937E-4*X-6.810124E-3)*X+.02475115
      RETURN
C                       ** RANGE 6.6-7.4 **
   29 KI3=(2.031843E-4*X-3.232035E-3)*X+.01302864
      RETURN
C                       ** RANGE 7.4-8.2 **
   30 KI3=(8.701440E-5*X-1.524126E-3)*X+6.749972E-3
      RETURN
C                       ** RANGE 8.2-9.0 **
   31 KI3=(3.742673E-5*X-7.157367E-4)*X+3.454768E-3
C
      RETURN
      END
```

Gauss Jacobi Quadrature

```
      parameter (ngjp1=6)
!
      real(NBF) KI3
C
      REAL(NBF) R(0:nrmax),SIG(nrmax),P(0:nrmax,0:nrmax),GAM(nrmax)
     +,        TAU(0:nrmax),RSQ(0:nrmax)
      REAL(NBF) GJC(ngjp1*ngjp1),GJP(ngjp1,ngjp1),GJW(ngjp1,ngjp1)
      equivalence (GJC(1),GJP(1,1),GJW(1,1))
C
C              Constants for GAUSS-JACOBI integration
C             ABRAMOWITZ and STEGUN, page 721, for K=I
C             X(I)=1.0-X(I)**2, W(I)=4.0*W(I), GJP(I,N)
C       W(I) are extra doubled for integration in two directions
C
      DATA        GJC /
     1  .0      , .55555556,.87393877,.95491150,.98046718,.99029084,
     2 2.0      , .0      ,.28606124,.65127016,.82660307,.90725799,
     3  .72783448,1.27216552,.0      ,.16932809,.47704397,.68412769,
     4  .27930792, .91696442,.80372766,.0      ,.11094751,.35681753,
     5  .12472388, .51939018,.81385828,.54202764,.0      ,.07803490,
     6  .06299166, .29563548,.58554794,.66869856,.38712636,.0      /
```

Analysis of Reactor Static Characteristics

2008 년 2 학기

Input set 1

```
3 1            !nr,ngauss
 0.5   0.6   1.0   !radius
 1.0   0.5   2.0   !sigt
 0.1   0.4   1.9   !sigs
 0.1   0.0   1.0   !qi
 0.    1.0        !albedo, jext
 0.1   0.0   1.0   !qi
 1.0   -1.2097     !albedo, jext
```

Output for set 1

```
   Albedo =  0.0000E+00    Jext =   1.0000E+00
    Jnet =   1.2097E+00    jout =   2.2097E+00     jin =   1.0000E+00
     Region     Radius     SigRem    SigScat     Source    SRC Flux     J Flux  Total Flux    Removal
          1  5.0000E-01  9.0000E-01  1.0000E-01  1.0000E-01  5.2749E-01  3.2944E-01  8.5693E-01  6.0573E-01
          2  6.0000E-01  1.0000E-01  4.0000E-01  0.0000E+00  6.3783E-01  3.7834E-01  1.0162E+00  3.5116E-02
          3  1.0000E+00  1.0000E-01  1.9000E+00  1.0000E+00  7.0847E-01  4.7848E-01  1.1869E+00  2.3865E-01
    Rtot =  8.7949E-01 jnet   1.2097E+00 sum =  2.0892E+00 qtot   2.0892E+00
   Albedo =  1.0000E+00    Jext =  -1.2097E+00
    Jnet =   1.2097E+00    jout =   2.2096E+00     jin =   9.9990E-01
     Region     Radius     SigRem    SigScat     Source    SRC Flux     J Flux  Total Flux    Removal
          1  5.0000E-01  9.0000E-01  1.0000E-01  1.0000E-01  2.0217E+00 -1.1648E+00  8.5690E-01  6.0570E-01
          2  6.0000E-01  1.0000E-01  4.0000E-01  0.0000E+00  2.3538E+00 -1.3377E+00  1.0161E+00  3.5115E-02
          3  1.0000E+00  1.0000E-01  1.9000E+00  1.0000E+00  2.8786E+00 -1.6917E+00  1.1869E+00  2.3864E-01
    Rtot =  8.7946E-01 jnet   1.2097E+00 sum =  2.0892E+00 qtot   2.0892E+00
```

Input set 2

```
11 5           !nr,ngauss
 0.05    0.10    0.15    0.20    0.25    0.30    0.35    0.40    0.48    0.56    0.65    !radius
 2.450e+0 2.450e+0 2.450e+0 2.450e+0 2.450e+0 2.450e+0 2.450e+0 2.450e+0 5.000e-1 1.500e+0 1.500e+0 !sigt
 2.450e-1 2.450e-1 2.450e-1 2.450e-1 2.450e-1 2.450e-1 2.450e-1 2.450e-1 1.000e-1 1.330e+0 1.330e+0 !sigs
 1.000e+0 1.000e+0 1.000e+0 1.000e+0 1.000e+0 1.000e+0 1.000e+0 1.000e+0 1.000e+0 1.000e+0 1.000e+0 !q
 1.   0.0       !albedo, jext
```

Output for set 2

```
   Albedo =  1.0000E+00    Jext =   0.0000E+00
    Jnet =   0.0000E+00    jout =   1.4208E+00     jin =   1.4208E+00
     Region     Radius     SigRem    SigScat     Source    SRC Flux     J Flux  Total Flux    Removal
          1  5.0000E-02  2.2050E+00  2.4500E-01  1.0000E+00  7.9850E-01  0.0000E+00  7.9850E-01  1.3828E-02
          2  1.0000E-01  2.2050E+00  2.4500E-01  1.0000E+00  8.0561E-01  0.0000E+00  8.0561E-01  4.1854E-02
          3  1.5000E-01  2.2050E+00  2.4500E-01  1.0000E+00  8.2024E-01  0.0000E+00  8.2024E-01  7.1025E-02
          4  2.0000E-01  2.2050E+00  2.4500E-01  1.0000E+00  8.4366E-01  0.0000E+00  8.4366E-01  1.0227E-01
          5  2.5000E-01  2.2050E+00  2.4500E-01  1.0000E+00  8.7802E-01  0.0000E+00  8.7802E-01  1.3685E-01
          6  3.0000E-01  2.2050E+00  2.4500E-01  1.0000E+00  9.2729E-01  0.0000E+00  9.2729E-01  1.7665E-01
          7  3.5000E-01  2.2050E+00  2.4500E-01  1.0000E+00  9.9994E-01  0.0000E+00  9.9994E-01  2.2512E-01
          8  4.0000E-01  2.2050E+00  2.4500E-01  1.0000E+00  1.1232E+00  0.0000E+00  1.1232E+00  2.9177E-01
          9  4.8000E-01  4.0000E-01  1.0000E-01  1.0000E+00  1.3387E+00  0.0000E+00  1.3387E+00  1.1843E-01
         10  5.6000E-01  1.7000E-01  1.3300E+00  1.0000E+00  1.4405E+00  0.0000E+00  1.4405E+00  6.4009E-02
         11  6.5000E-01  1.7000E-01  1.3300E+00  1.0000E+00  1.4702E+00  0.0000E+00  1.4702E+00  8.5508E-02
    Rtot =  1.3273E+00 jnet   0.0000E+00 sum =  1.3273E+00 qtot   1.3273E+00
```