CODES FOR CHAPTER 21

mocacyl.f, rnarray.f
Program mocacyl is a Monte Carlo routine for a nongray, nonisothermal, isotropically scattering medium confined inside a two-dimensional, axisymmetric cylindrical enclosure bounded by gray, diffusely emitting and reflecting walls. Temperature and radiative properties are assumed known everywhere inside the enclosure and along the walls. Requires use of program rnarray to set up random number relationships (locations and wavenumbers of emission vs. random numbers). Calculates local radiative fluxes to the walls q_w^R. mocacyl.f is supplied in two slightly different versions: mocacyl_std.f uses standard Monte Carlo for absorption, i.e., an absorption length is picked from equation (21.17) and the bundle is traced until this point is reached (and its energy is deposited at that point) or to a point on a wall, where it is absorbed (i.e., picking a random number $R_\alpha < \alpha$), whichever comes first. In the other version, mocacyl_ep.f, the energy partitioning scheme of Sections 8.7 and 21.7 is employed, i.e., the bundle's energy is gradually attenuated by absorption along its path, and by the fraction α, whenever the bundle hits (and is reflected from) a wall, until the bundles energy becomes negligible. This method is somewhat more expensive per bundle, but should in many instances give converged results with a lot less bundles. At the present time only mocacyl_std.f also calculates the internal radiative source $-\nabla \cdot \mathbf{q}^R$ in addition to wall fluxes.
The package consists of the following files:

- the main programs mocacyl_std.f and mocacyl_ep.f ,
- the program preparing random number relationships for medium emission, rnarray.f,
- file mocasub.f, which contains simple versions of subroutine PROPS and function ABSCO, as well as a poor man's random number generator called RNUM, all of which the user can (and should) replace,
- a file splines.f for monotonic splines,used by both mocacyl.f and rnarray.f, and
- sample output files datlam.dat and results.dat.

Program **rnarray**
This program prepares random number relationships for photon emission locations within the cylindrical medium, using equations (21.9) and (21.11).
Input:

NRP	=	Number of radial nodes for medium emission random number relationships
NZP	=	Number of axial nodes for medium emission random number relationships
NNP	=	Number of random numbers for medium emission relationships
RL	=	Radius of cylinder, (cm)
ZL	=	Length of cylinder, (cm)
AN	=	Refractive index of medium (AN=1.0 for gases)
STN	=	Refractive index of soot
STK	=	Absorptive index of soot
IGRAY	=	Gray/nongray medium switch: IGRAY=0 nongray, IGRAY=1 gray (ignoring contribution from gases; absorption coefficient = PAC)

LU = Logical unit number for output: LU=6 sends output to screen, other (legal) values send output to file datlam.dat

Output:

File datlam.dat contains random number relationships generated by RNARRAY:

ETOTAL = Total energy emitted (per unit time) by entire volume, in W

PLMCL(I),I=1,NZP = Planck-mean absorption coefficient along centerline, in cm^{-1}

RRA(J),J=1,NNP = Emission radial location as f(random#), in cm

ZR(K,J),K=1,NRP,J=1,NNP = Emission axial location as $f(r,$ random#), in cm

WVE2(K,I,J),K=1,NRP,I=1,NZP,J=1,NRNP1 = Emission wavelength as $f(r,z,$ random#) (IGRAY=0 only), in μm

These arrays are used by mocacyl.f to determine emission location and wavelength, using single (r), double (z), and triple (λ) linear interpolation between tabulated values.

Note: this program requires two user-supplied subroutines, SUBROUTINE PROPS and FUNCTION ABSCO. Subroutine PROPS(R,Z,T,SVF,PCO2,PH2O,PAC,PSC), upon inputting radial position R (in cm) and axial position Z (in cm), must return local values of T (temperature in K), SVF (soot volume fraction, –), PCO2 (partial pressure of CO_2, in bar), PH2O (partial pressure of H_2O, in bar), PAC (nonsoot particle background absorption coefficient, in cm^{-1}), and PSC (nonsoot particle background scattering coefficient, in cm^{-1}). As provided here, the subroutine produces a uniform field, i.e., SVF=0., T=1000., PH2O=.1, PCO2=.1, PAC=.01, PSC=0.

Function ABSCO(SVF,PCO2,PH2O,PAC,W,T), upon inputting SVF (soot volume fraction, –), PCO2 (partial pressure of CO_2, in bar), PH2O (partial pressure of H_2O, in bar), PAC (nonsoot particle background absorption coefficient, in cm^{-1}), W (wavelength in μm), and T (temperature in K), must return ABSCO, the absorption coefficient of the medium (in cm^{-1}). As provided, function ABSCO calculates the gas absorption coefficient from the wide-band model [with an approximate evaluation of $\alpha(T)/\alpha_0$ in equation (11.144)] assuming strong overlap ($\beta \to \infty$), and the soot absorption coefficient is calculated from equation (12.123).

Function ABSCO should return ABSCO=PAC if W < 0 (gray medium). Both, PROPS and ABSCO, must contain the common statement line COMMON RL,ZL,AN,STN,STK,NRR,NZL,NRN.

Program **mocacyl**

Program mocacyl requires the following input:

NRP = Number of radial nodes for medium emission random number relationships

NZP = Number of axial nodes for medium emission random number relationships

NNP = Number of random numbers for medium emission relationships

NR = Number of radial nodes for surface flux calculations

NZ = Number of axial nodes for surface flux calculations

T3(NZ) = Temperature of liner wall ($r = R$), (K)

EPS(3) = Surface emittances: EPS(1)=inlet, EPS(2)=exit, EPS(3)=liner

RL = Radius of cylinder, (cm)

ZL = Length of cylinder, (cm)

AN = Refractive index of medium (AN=1.0 for gases)

STN = Refractive index of soot

STK = Absorptive index of soot

NTOTAL = Total number of photon bundles emitted from medium (number of bundles for surface emission are chosen automatically as function of NTOTAL)

IGRAY = Gray/nongray medium switch: IGRAY=0 nongray, IGRAY=1 gray (ignoring contribution from gases and soot; absorption coefficient = PAC)

IWALL = Wall emission switch: IWALL=0 only considers medium emission; IWALL=1 also considers surface emission

LU = Logical unit number for output: LU=6 sends output to screen, other (legal) values send output to file results.dat

File datlam.dat Contains random number relationships generated by rnarray

Note: The program **does not check** for consistency of datlam.dat, i.e., whether identical input values have been chosen in both rnarray and mocacyl!

Output:
Upon output relevant input data are displayed, as well as
QW(1,1..NR) = Axial radiative heat flux for NR radial nodes at inlet (W/cm^2)
QW(2,1..NR) = Axial radiative heat flux for NR radial nodes at exit (W/cm^2)
QW(3,1..NZ) = Radial radiative heat flux for NZ axial nodes at liner (W/cm^2)
Note: QW > 0 implies that the flux goes into wall, while for QW < 0 the flux is out of the wall.

Example:
We will use routines PROPS and ABSCO as provided, and also the input data as stated in mocacyl (and similar in rnarray):

```
 C C GENERAL DATA C C T3=liner temperature (K),
EPS=emittances(inlet,exit,liner), C RL=radius (cm), ZL=length (cm),
AN=refractive index of medium (-), C STN= soot refractive index, STK= soot
absorptive index, C NRR=# radial nodes, NZL=# axial nodes, NRN=# wavelength
nodes C
       DATA T3/7*1000./
       DATA EPS/1.,1.,.5/
       RL=10.
       ZL=10.
       TW0=1000.
       TWL=0.
       AN=1.
       STN=1.89
       STK=0.92
       NTOTAL=500000
       IGRAY=0
       IWALL=1
       LU=7
```

This models an isothermal, cylindrical, nongray medium with temperatures of 1000 K for medium, inlet and liner, only the exit being cold at 0 K. For the simulation we will use 500,000 bundles for medium emission, and corresponding numbers for surface emission (such that all bundles carry roughly identical energies). Executing rnarray produces the required data file datlam.dat and, running mocacyl, the results are contained in results.dat as:

```
          GENERAL DATA
          ************

               BURNER RADIUS                  10.00 CM
               BURNER LENGTH                  10.00 CM
               EMITTANCES:             INLET:  1.00
                                        EXIT:  1.00
                                       LINER:  0.50
               REFRACTIVE INDEX                1.00
               NUMBER OF BUNDLES: MEDIUM: 500000
                                  INLET: 474561
                                   EXIT:      1
                                  LINER: 474562



     WALL TEMPERATURES (DEG.K)

       TW0   TWL  T3:I= 1     2     3     4     5     6     7
      1000.    0.     1000. 1000. 1000. 1000. 1000. 1000. 1000.



     PROPERTY VALUES ALONG CENTER LINE:
```

```
      I     Z     T      FR    PCO2    PH2O    K-PL   ABSC   SCAT
           CM   DEG.K    %     ATM     ATM     CM-1   CM-1   CM-1

      1    0.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000
      2    0.5  1000.  0.00  0.100  0.100  0.032  0.010  0.000
      3    1.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000
      4    1.5  1000.  0.00  0.100  0.100  0.032  0.010  0.000
      5    2.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000
      6    2.5  1000.  0.00  0.100  0.100  0.032  0.010  0.000
      7    3.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000
      8    3.5  1000.  0.00  0.100  0.100  0.032  0.010  0.000
      9    4.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     10    4.5  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     11    5.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     12    5.5  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     13    6.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     14    6.5  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     15    7.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     16    7.5  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     17    8.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     18    8.5  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     19    9.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     20    9.5  1000.  0.00  0.100  0.100  0.032  0.010  0.000
     21   10.0  1000.  0.00  0.100  0.100  0.032  0.010  0.000


  RADIAL HEAT FLUXES AT NODES I (W/SQCM)

      I     Z   QWALL
           CM   W/SQCM

      1    0.7   -0.4
      2    2.1   -0.4
      3    3.6   -0.5
      4    5.0   -0.7
      5    6.4   -0.8
      6    7.9   -1.0
      7    9.3   -1.2


  AXIAL HEAT FLUXES AT NODES J (W/SQCM)

J       1     2     3     4     5
Q0   -2.4  -2.3  -2.1  -1.9  -1.6
QL    5.9   5.8   5.8   5.6   5.4
```

Note that the fluxes for the three hot walls are slightly negative (surfaces are losing heat, while the cold exit surface experiences strong positive heat fluxes. Also note that the code does not provide error estimates, i.e., it should be run for different values of `NTOTAL` to get an idea of variances.

`FwdMCcs.f90`

Program `FwdMCcs` is a standard forward Monte Carlo code for a narrow collimated, cylindrical beam (centered at $x = y = 0$) penetrating through a nonabsorbing, isotropically scattering slab, calculating the flux onto a small, directionally-selective detector located at $x_0 < x < x_0 + dx$, $0 < y < dy$, $z = L$, as shown in Fig. 21-6 and described in Example 21.3. (`FwdMCck1` and `FwdMCck2` are forward Monte Carlo codes for the same problem, but also allow for absorption in the medium; `FwdMCck1` uses standard ray tracing, while `FwdMCck2` uses energy partitioning; see Example 21.4.)

Input:

`L`	=	thickness of layer, L (m);
`sig`	=	scattering coefficient of medium, σ_s (m^{-1});
`QT`	=	total energy contained in collimated beam, (W);
`R`	=	radius of collimated beam, R (m);
`x0`	=	displacement of left end of detector from center of beam, x_0 (m);
`dx`	=	width of detector in x-direction, (m);
`dy`	=	width of detector in y-direction, (m);
`thd`	=	opening angle of detector, $\theta_{\max}$ (degrees);
`N`	=	minimum number of photon bundles to be traced for each "sample;"
`numsmpl`	=	number of numerical "samples" collected for the determination of a variance;
`stddevmax`	=	maximum relative variance allowed for the calculation of $q_{\rm det}$, the flux hitting detector, $= \sigma_m/q_{\rm det}$ from equation (8.8).

The values for input parameters are assigned in sequence near the top of the program. As distributed, a 1 m thick layer with a scattering coefficient of $\sigma_s = 1\text{m}^{-1}$ is modeled, for a 10 cm radius beam of 100 W strength. The (rather large) detector is 10 cm × 10 cm displaced by 20 cm from the center of the beam, and has an acceptance angle of 10°. `numsmpl`= 10 numerical samples will be taken, initially each containing `N`= 100,000 bundles, to be increased (if necessary) until the relative variance falls below `stddevmax`= 0.05 or 5%.

```
    open(unit=8,file='fwdmccs.dat',status='unknown')
    write(8,1)
    write(*,1)
  1 format('no. of bundles     q_det     variance    rel.var.(%)')
!
    L=1.        ! m
    sig=1.      ! 1/m
    QT=100.     ! W
    R=.1        ! m
    x0=.2       ! m
    dx=0.10     ! m
    dy=0.10     ! m
    thd=10.     ! deg
    N=100000
    numsmpl=10
    stddevmax=0.05
!
```

The program consists of two major parts. The first is a double loop over `numsmpl*N` photon bundles, tracing their paths, as described in Example 21.3. In the second part an average value for the detector irradiation is determined, as well as its relative standard deviation, based on the `numsampl` data points. If the relative standard deviation is too large (`stddev>stddevmax`) the `numsampl` samples of $q_{\rm det}$ (based on N bundles) are combined into ½×`numsampl` samples (with $2N$ bundles each), the number of bundles is doubled to $2N$, and an additional ½×`numsampl` samples are obtained (with $2N$ bundles each). Thus, after going through the bundle-tracing part one more time, we have again `numsmpl` samples, but each based on twice as many photon bundles. This procedure is repeated until the convergence criteria are met.

For the given case that leads to the following output, stored in `fwdmccs.dat`,

```
no. of bundles     q_det     variance    rel.var.(%)
       1000000  0.3200E-02  0.4899E-03      15.31
       2000000  0.3500E-02  0.3944E-03      11.27
       4000000  0.3100E-02  0.3055E-03       9.86
       8000000  0.2963E-02  0.1468E-03       4.95
```

i.e., for this large detector 8,000,000 photon bundles are needed to attain a relative variance of less than 5%. If a smaller detector was chosen, the necessary number of bundles would be roughly inversely proportional to the detector area!

FwdMCck1.f90

Program FwdMCck1.f90 is identical to FwdMCcs.f90, except that the medium also absorbs radiation (besides isotropically scattering it). Therefore, the input is:

Input:

L	=	thickness of layer, L (m);
sig	=	scattering coefficient of medium, σ_s (m^{-1});
kap	=	absorption coefficient of medium, κ (m^{-1});
QT	=	total energy contained in collimated beam, (W);
R	=	radius of collimated beam, R (m);
x0	=	displacement of left end of detector from center of beam, x_0 (m);
dx	=	width of detector in x-direction, (m);
dy	=	width of detector in y-direction, (m);
thd	=	opening angle of detector, $\theta_{\max}$ (degrees);
N	=	minimum number of photon bundles to be traced for each "sample;"
numsmpl	=	number of numerical "samples" collected for the determination of a variance;
stddevmax	=	maximum relative variance allowed for the calculation of $q_{\rm det}$, the flux hitting detector, $= \sigma_m/q_{\rm det}$ from equation (8.8).

As distributed, a 1m thick layer with a scattering coefficient of $1m^{-1}$ and an absorption coefficient of $\kappa = 1m^{-1}$ is modeled, for a 10 cm radius beam of 100 W strength. The (rather large) detector is 10 cm × 10 cm displaced by 20 cm from the center of the beam, and has an acceptance angle of 10°. numsmpl= 10 numerical samples will be taken, initially each containing N = 100,000 bundles, to be increased (if necessary) until the relative variance falls below stddevmax = 0.05 or 5%:

```
    open(unit=8,file='fwdmck1.dat',status='unknown')
    write(8,1)
    write(*,1)
  1 format('no. of bundles     q_det     variance    rel.var.(%)')
!
    L=1.        ! m
    sig=1.      ! 1/m
    kap=1.      ! 1/m
    QT=100.     ! W
    R=.1        ! m
    x0=.2       ! m
    dx=0.10     ! m
    dy=0.10     ! m
    thd=10.     ! deg
    N=100000
    numsmpl=10
    stddevmax=0.05
!
```

For the given case that leads to the following output, stored in fwdmck1.dat:

```
no. of bundles     q_det     variance    rel.var.(%)
       1000000  0.1100E-02  0.2333E-03      21.21
       2000000  0.7500E-03  0.2007E-03      26.76
       4000000  0.6500E-03  0.1067E-03      16.42
       8000000  0.6875E-03  0.7512E-04      10.93
      16000000  0.6438E-03  0.4375E-04       6.80
      32000000  0.6781E-03  0.4396E-04       6.48
      64000000  0.6766E-03  0.3424E-04       5.06
     128000000  0.7102E-03  0.3011E-04       4.24
```

i.e., 128,000,000 photon bundles are required, or – making allowance for the slightly different variance – about 10 times as many as for the purely scattering medium. Clearly, with a minimum optical thickness

of $\sqrt{1^2 + 0.2^2} = 1.02$ many photon bundles, that would otherwise be scattered toward the detector, become absorbed first.

FwdMCck2.f90

Program FwdMCck2.f90 is identical to FwdMCck1.f90, except that energy partitioning is employed, i.e., photon bundles are emitted and have paths identical to the simulation in FwdMCcs.f90, but the bundles' strengths are attenuated exponentially along their way according to Beer's law. Input is identical to FwdMCck1.f90, as are the as-distributed input parameters. However, the output (stored in fwdmck2.dat) now looks like this:

```
no. of bundles     q_det     variance    rel.var.(%)
       1000000  0.9003E-03  0.1610E-03      17.89
       2000000  0.9382E-03  0.1109E-03      11.82
       4000000  0.8160E-03  0.8774E-04      10.75
       8000000  0.7927E-03  0.3710E-04       4.68
```

i.e., FwdMCck2.f90 converges at he same rate as the no-absorption case FwdMCcs.f90, demonstrating the power of the energy partitioning approach.

FwdMCps.f90

Program FwdMCps is a standard forward Monte Carlo code for radiative energy emitted by a point source penetrating through a nonabsorbing, isotropically scattering slab, calculating the flux onto a small, directionally-selective detector. It is a variation of FwdMCcs.f90, considering a purely scattering slab, but replacing the collimated beam by an internal point source at $x = 0$, $y = 0$, $z = z_{ps}$. Thus, the simulation is almost identical to that of FwdMCcs.f90, except that all photon bundles are now emitted from a single point, but into random directions. The input is also identical to FwdMCcs.f90, with R replaced by zps:

Input:

L	=	thickness of layer, L (m);
sig	=	scattering coefficient of medium, σ_s (m^{-1});
QT	=	total energy contained in point source, (W);
zps	=	z-coordinate of point source, (m);
x0	=	displacement of left end of detector from point source, x_0 (m);
dx	=	width of detector in x-direction, (m);
dy	=	width of detector in y-direction, (m);
thd	=	opening angle of detector, $\theta_{\max}$ (degrees);
N	=	minimum number of photon bundles to be traced for each "sample;"
numsmpl	=	number of numerical "samples" collected for the determination of a variance;
stddevmax	=	maximum relative variance allowed for the calculation of $q_{\det}$, the flux hitting detector, $= \sigma_m / q_{\det}$ from equation (8.8).

The as-delivered case also is the same as for FwdMCcs.f90, with the 10 cm-radius beam replaced by a points source at zps = 0.5(m). Thus, the input section reads:

```
    open(unit=8,file='fwdmcps.dat',status='unknown')
    write(8,1)
    write(*,1)
  1 format('no. of bundles     q_det     variance    rel.var.(%)')
!
    L=1.        ! m
    sig=1.      ! 1/m
    QT=100.     ! W
    zps=.1      ! m
    x0=.2       ! m
    dx=0.10     ! m
    dy=0.10     ! m
    thd=10.     ! deg
    N=100000
    numsmpl=10
    stddevmax=0.05
```

```
!
```

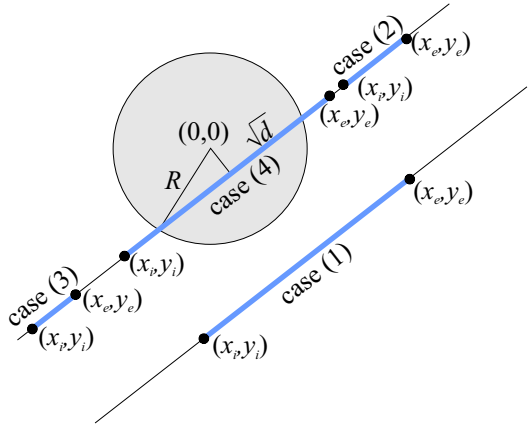
The resulting output, stored in `fwdmcps.dat`, is:

```
no. of bundles     q_det     variance    rel.var.(%)
       1000000  0.5000E-02  0.7303E-03      14.61
       2000000  0.4550E-02  0.4561E-03      10.02
       4000000  0.4800E-02  0.3958E-03       8.25
       8000000  0.4850E-02  0.3226E-03       6.65
      16000000  0.4850E-02  0.2143E-03       4.42
```

`RevMCcs.f90`

This program is a reverse or backward Monte Carlo implementation of the problem solved by `FwdMCcs.f90`, i.e., a narrow collimated beam hitting a purely scattering slab, and scattered toward a small, directionally-selective detector. Input and output are identical to `FwdMCcs`, except that the default minimum number of photon bundles is much smaller, here set to `N=1000`.

Again, the program consists of two parts, a double loop tracing over `numsmpl×N` photon bundles, and a module calculating detector power and its standard deviation. Tracing follows the rules of equation (21.84), and the detector flux from equation (21.79) (with $\epsilon'_\lambda = 1$ for the black detector). The coding is self-explanatory except when, between two scattering events, the bundle starts and ends outside the collimated beam column, $x^2 + y^2 = r^2 > R^2$. There are four different possibilities, as illustrated in the sketch below:



1. The path of the bundle misses the beam altogether,
2. the path intercepts the beam, but the bundle is moving away from the beam,
3. the path intercepts the beam and moves toward it, but is scattered anew before reaching it, and finally
4. the bundles path intersects the beam.

These four possibilities are tested by calculating the partial distances l_1 and l_2 (`lpart`) from the starting point (projected into the x-y-plane) (x_i, y_i) toward (x_e, y_e) (the final point) to where the path enters and exits the beam. Thus, using the x- and y-components of the vector equation

$$\mathbf{r}_i + l\hat{\mathbf{s}} = \mathbf{R}$$

leads to

$$x_i + ls_x = x_c,$$

$$y_i + ls_y = y_c,$$

where (x_c, y_c) is a point on the outer limit of the collimated beam, $r = R$. Squaring and adding the latter two equations gives

$$x_i^2 + y_i^2 + 2(x_i s_x + y_i s_y)l + (s_x^2 + s_y^2)l^2 = R^2$$

or

$$l_{1,2} = \frac{-b \pm \sqrt{d}}{c},$$

where

$$a = x_i^2 + y_i^2 - R^2$$
$$b = x_i s_x + y_i s_y$$
$$c = s_x^2 + s_y^2$$
$$d = b^2 - ac.$$

Thus the above four scenarios correspond to

1. If $d < 0$ the roots are complex, i.e., there is no intersection,
2. If $l_1 < 0$ the bundle moves away from beam
3. If $l_1 < l_\sigma$ the bundle is scattered again before reaching the beam, and
4. otherwise both intercepts are calculated to determine $I_{\lambda n.}$

After the tracing of photon bundles is completed, average values and standard deviations are calculated as in the forward Monte Carlo codes, e.g., `FwdMCcs`. Using the as-supplied input (same as for `FwdMCcs`, but with `N=1000`) leads to the output, stored in `revmccs.dat`:

```
no. of bundles     q_det     variance    rel.var.(%)
         10000  0.2882E-02  0.1192E-03       4.14
```

Thus, a better variance is achieved with only 10,000 bundles, as opposed to the 8,000,000 bundles used in the forward simulation (and this ratio would become correspondingly more extreme for smaller detector areas and acceptance angles).

`RevMCck1.f90`, `RevMCck2.f90`

These programs are backward Monte Carlo Implementations of `FwdMCck1` and `FwdMCck2`, respectively, as also discussed in Example 21.4. They have identical inputs (except the much lower minimum number of photon bundles, here set to `N=1000` as default); and their outputs also follow the format of their counterparts. Ray tracing for `RevMCck1` is the same as for `RevMCcs`, except for the slight modification demanded by equation (21.85). If energy partitioning is used, attenuation along the entire path length of the photon bundle must be considered, as explained in the last equation of Example 21.4. For the as-supplied cases the output from `RevMCck1`, stored in `revmck1.dat`, is:

```
no. of bundles     q_det     variance    rel.var.(%)
         10000  0.7261E-03  0.5045E-04       6.95
         20000  0.7540E-03  0.3091E-04       4.10
```

i.e., 20,000 bundles are required (as opposed to 128,000,000 used by `FwdMCck1`). For `RevMCck2` the output, stored in `revmck2.dat`, is:

```
no. of bundles     q_det     variance    rel.var.(%)
         10000  0.7623E-03  0.3061E-04       4.02
```

i.e., 10,000 bundles are required (as opposed to 8,000,000 used in `FwdMCck2`, or 20,000 used in `RevMCck1`).

`RevMCps.f90`

This program is the backward Monte Carlo equivalent of `FwdMCps`, with identical input and output format (again, with the exception of a much smaller base line value for the number of bundles). In the backward

Monte Carlo simulation, the detector flux again consists of a direct and a scattered component. In the code it is assumed that the direct component is zero, this time because all direct radiation hits the detector at an angle larger than the acceptance angle (this could, of course, be easily changed). As for collimated irradiation backward Monte Carlo also becomes inefficient if the radiation source comes from a very small surface or volume. The trick is again to break up intensity into a direct component (intensity coming directly from the source without scattering or wall reflections), and a multiply-scattered and reflected part, as given by Modest [1] and briefly described here. Again, we let I_d satisfy the radiative transfer equation without the inscattering term, or,

$$\hat{\mathbf{s}} \cdot \nabla I_d(\mathbf{r}, \hat{\mathbf{s}}) = S_d(\mathbf{r}, \hat{\mathbf{s}}) - \beta(\mathbf{r}) I_d(\mathbf{r}, \hat{\mathbf{s}}),$$

which has the simple solution

$$I_d(\mathbf{r}, \hat{\mathbf{s}}) = \int S_d(\mathbf{r}', \hat{\mathbf{s}}) \exp\left[- \int_{\mathbf{r} \to \mathbf{r}'} (\kappa + \sigma_s)\, ds' \right] ds, \tag{CC-21-1}$$

where the main integral is along a straight path from the boundary of the medium to point $\mathbf{r}$ in the direction of $\hat{\mathbf{s}}$. For example, if there is only a simple point source at $\mathbf{r}_0$ with total strength Q_0, emitting isotropically across a tiny volume δV, equation (CC-21-1) becomes

$$I_d(\mathbf{r}, \hat{\mathbf{s}}) = \frac{Q_0}{4\pi |\mathbf{r}_0 - \mathbf{r}|^2} \exp\left[- \int_{\mathbf{r}_0 \to \mathbf{r}} (\kappa + \sigma_s)\, ds' \right] \delta(\hat{\mathbf{s}} - \hat{\mathbf{s}}_0), \tag{CC-21-2}$$

where $\hat{\mathbf{s}}$ is a unit vector pointing from $\mathbf{r}_0$ toward $\mathbf{r}$, and use has been made of the fact that

$$\delta V = \delta A\, \delta s = \delta \Omega_0 |\mathbf{r}_0 - \mathbf{r}|^2\, \delta s,$$

where $\delta\Omega_0$ is the solid angle, with which δV is seen from **r.** Equation (CC-21-2) can be used to calculate the direct contribution of Q_0 hitting a detector, and it can be used to determine the source term for the RTE of the scattered radiation as

$$\begin{aligned} S_1(\mathbf{r}, \hat{\mathbf{s}}) &= \frac{\sigma_s(\mathbf{r})}{4\pi} \int_{4\pi} I_d(\mathbf{r}, \hat{\mathbf{s}}') \Phi(\mathbf{r}, \hat{\mathbf{s}}', \hat{\mathbf{s}})\, d\Omega' \\ &= \frac{\sigma_s(\mathbf{r}) Q_0}{16\pi^2 |\mathbf{r}_0 - \mathbf{r}|^2} \exp\left[- \int_{\mathbf{r}_0 \to \mathbf{r}} (\kappa + \sigma_s) ds' \right] \Phi(\mathbf{r}, \hat{\mathbf{s}}_0, \hat{\mathbf{s}}). \end{aligned} \tag{CC-21-3}$$

The rest of the solution proceeds as before, with $I_n(\mathbf{r}_i, -\hat{\mathbf{s}}_i)$ found from equations (CC-21-3) and (21.82) as

$$I_n(\mathbf{r}_i, -\hat{\mathbf{s}}_i) = \frac{\sigma_s Q}{16\pi^2} \sum_j \int_{l_{\sigma,j}} \frac{e^{-\sigma_s |\mathbf{r}_0 - \mathbf{r}|}}{|\mathbf{r}_0 - \mathbf{r}|^2}\, dl', \tag{CC-21-4}$$

where the $l_{\sigma,j}$ are the straight paths the bundle travels between scattering events. Equation (CC-21-4) must be integrated numerically, and this can be done using a simple Newton-Cotes scheme. Alternatively, the integral can be obtained statistically from

$$I_n(\mathbf{r}_i, -\hat{\mathbf{s}}_i) = \frac{\sigma_s Q}{16\pi^2} \sum_j \frac{l_{\sigma,j}}{N_{\text{int}}} \sum_{k=1}^{N_{\text{int}}} \frac{e^{-\sigma_s |\mathbf{r}_0 - \mathbf{r}_k|}}{|\mathbf{r}_0 - \mathbf{r}_k|^2},$$

where the $\mathbf{r}_k$ are N_{int} random locations chosen uniformly along path $l_{\sigma,j}$. This was implemented in `RevMCps.f90`, choosing N_{int} (= `numint`) to be inversely proportional to the distance of the integration point from the source (or proportional to its relative importance). Results for detector flux for the as-supplied case (same as for `FwdMCps.f90`) are stored in `revmcps.dat` as:

```
no. of bundles     q_det     variance    rel.var.(%)
         10000  0.4614E-02  0.1057E-03       2.29
```

i.e., with only 10,000 bundles we achieved a much better variance then by using 16,000,000 bundles in `FwdMCps.f90`.

`RevMCps.f90`

The backward Monte Carlo equivalent of `FwdMCps`.

The documentation for this routine is not available.

References

1. Modest, M. F.: "Backward Monte Carlo simulations in radiative heat transfer," *ASME Journal of Heat Transfer*, vol. 125, no. 1, pp. 57–62, 2003.