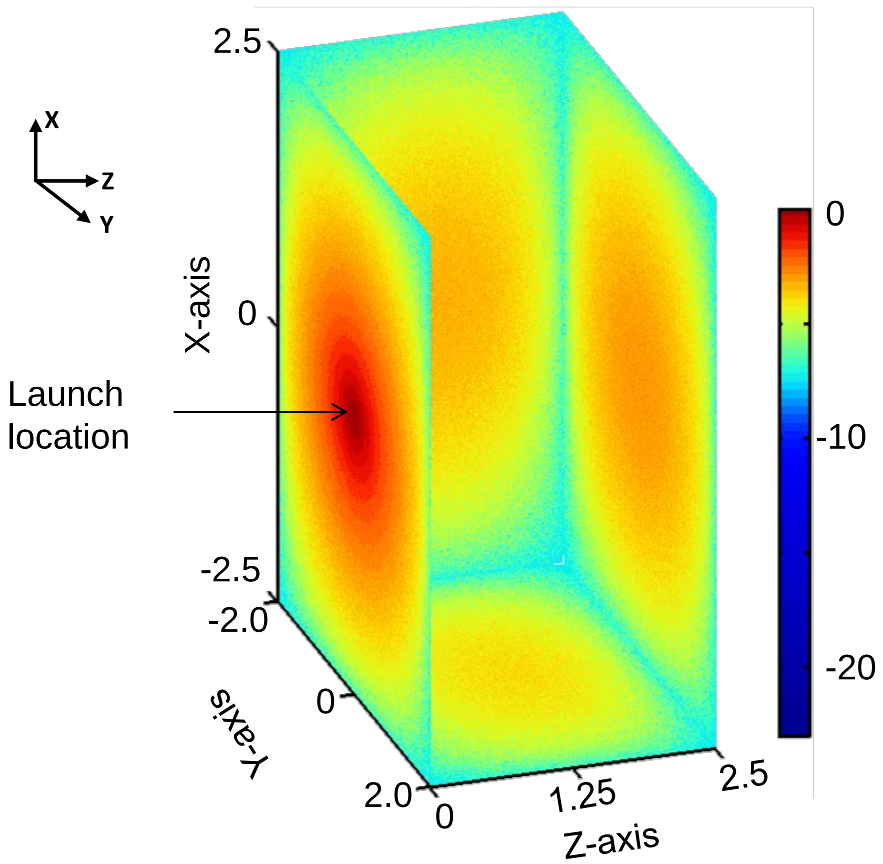
Raman Monte Carlo Simulation for Light Transport in Cuboid in Standard C

Vijitha Periyasamy, B.E
Manojit Pramanik, Ph.D.
Biomedical Imaging Laboratory
School of Chemical and Biomedical Engineering,
Nanyang Technological University, Singapore 637459



Overview

Monte Carlo Modeling of steady-state Light Transport in Multilayered Tissues (MCML) by Dr. Lihong Wang and Dr. Steven Jacques is a tool used to simulate the light transport in biological tissues. It can provide diffuse reflectance, diffuse transmittance and absorbance in a multi-layered tissue structure. It can be used to obtain the absorbance of light inside tissue as well, which is useful in many applications such as photoacoustic imaging, photodynamic therapy, diffuse optical tomography, Raman imaging etc. The simulation is statistical and hence requires large number of photons to be tracked to reduce variation. Since photon is treated as a particle, the phase and polarization is ignored. The aim of MCML is to look at radiant energy distribution in a turbid medium where phase and polarization is quickly randomized and hardly have any role to play in the optical output. Currently MCML simulations exist for infant brain, surficial blood vessels, etc. Modifications of basic MCML are time-resolved MCML, GPU speeded-up MCML and Mesh-based MCML. We have worked on a modified MCML to incorporate spherical, cylindrical, ellipsoidal, or cuboidal inclusion inside the turbid medium to mimic sentinel lymph node (SLN) structure, tumor, cells, blood clots, blood vessels, etc.

Here, MCML has been programmed to handle Raman scattering. In physical world, at every Rayleigh scattering site 10^{-6} part of the photon is Raman scattered. Since tracking of parts of photon is time consuming, in this stochastic modeling Raman scattering is done at a probability of 10^{-6} and the photon is tagged as Raman photons whose exit from the medium is recorded. Another modification to Raman MC is that the multilayered planar geometry is converted to multilayered cuboid. The example used here is an explosive concealed in a plastic container. So we have two layers container and material. In the output file, total number of Raman photons generated and the photons exiting the planes recorded by the code are listed. Planes recorded are xy in z^0, xy in z^d, xz at y^+ and yz at x^+. As photons are launched at origin, which is z^0 plane, reflected photons are captured in this plane. z^d plane is the plane parallel to z^0 plane which is transmittance plane. z^d is the depth of the cuboid (distance between reflectance and transmittance plane). Due to the cylindrical symmetry in the experimental set-up xy^+ and xy^- are symmetric and zy^+ and zy^- planes are symmetric. Hence only the positive planes are recorded.

Table of Contents

1. Introduction

Here we assume that the user is well versed with the original MCML code. If not, we recommend them to first go through the original MCML code and relevant support document to understand how the Monte Carlo modelling is done for light transport in multi-layered tissues. The original MCML codes with support files are available on Dr. Lihong Wang's website (http://oilab.seas.wustl.edu/mc.html).

The three dimensions [height (X), width (Y) and depth (Z)] of the desired cuboid are given as input, along with the Raman probability for each layer which is usually 1E-6. Figure 1 demonstrates the problem geometry discussed in this manual. The number of Raman photons and the distribution of the escaping Raman photons from both the layers across the 4 planes are given in the output file. In literature, there are instances where the path taken by the Raman photons are exactly same as the Rayleigh photons. There are also instances when a Raman photon undergoes isotropic scattering once they are generated. To enable the code to handle these two scenarios, in the input file a flag is allotted to be set 1 for isotropic scattering and 0 to follow Rayleigh path. Also the thickness of the container is given as input. A pointed source of photons is launched at the origin.

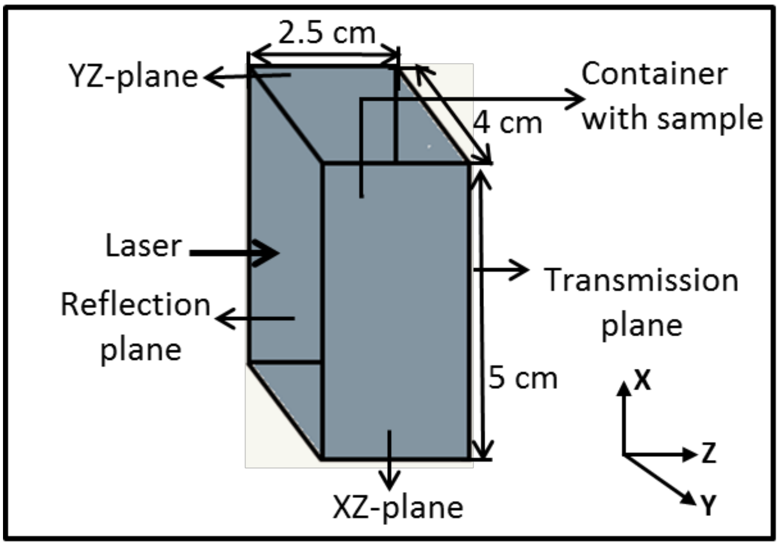


Figure 1: Schematic representation of the cuboid in Cartesian co-ordinate system.

2. Input template

```
####
# Template of input files for Raman Monte Carlo simulation for Cuboid
(RMCC).
# Anything in a line after "#" is ignored as comments.
# Space lines are also ignored.
# Lengths are in cm, mua and mus are in 1/cm.
####

1.0                                 # file version
1                                   # number of runs

### Specify data for run 1
temp_out.mco     A            # output filename, ASCII/Binary
40000                         # No. of photons
0.005 20E-4                   # dz, dr
500   1    1                  # No. of dz, dr & da.
0.005  0.005                  # dx, dy
1000  800                     # No. fo dx & dy
1                             # Flag for isotropic scattering.
                              # 1 for iso-scattering.

5 4 2.5 0.1      # Dim. of cuboid; x,y,z and thickness of container.

# n   mua   mus   g     RP          # One line for each layer
1.0                                 # n for medium above.
1.0   0     250  0.90   1E-6        # Container
1.0   0     500  0.90   1E-6        # Material
1.0                                 # n for medium below.
```

3. Changes in code files

Cuboid dimension, thickness of container and flag for isotropic scattering are read from the input file. Photon structure when launched is initialized with a pointer to indicate if the photon is within the container or material. Boundary distance is checked for the planes of cuboid. When the photon is in container, boundaries for material and air are checked. When the photon is in material, boundary conditions are checked only for container walls. A random number is generated to convert the Rayleigh photon to Raman photon. Photon is tagged with the layer in which it is converted to Raman if the conversion condition is satisfied. If the photon is just converted to Raman and if the flag is 1, isotropic scattering is done. Tagged Raman photons are recorded in their respective planes when they leave the problem geometry.

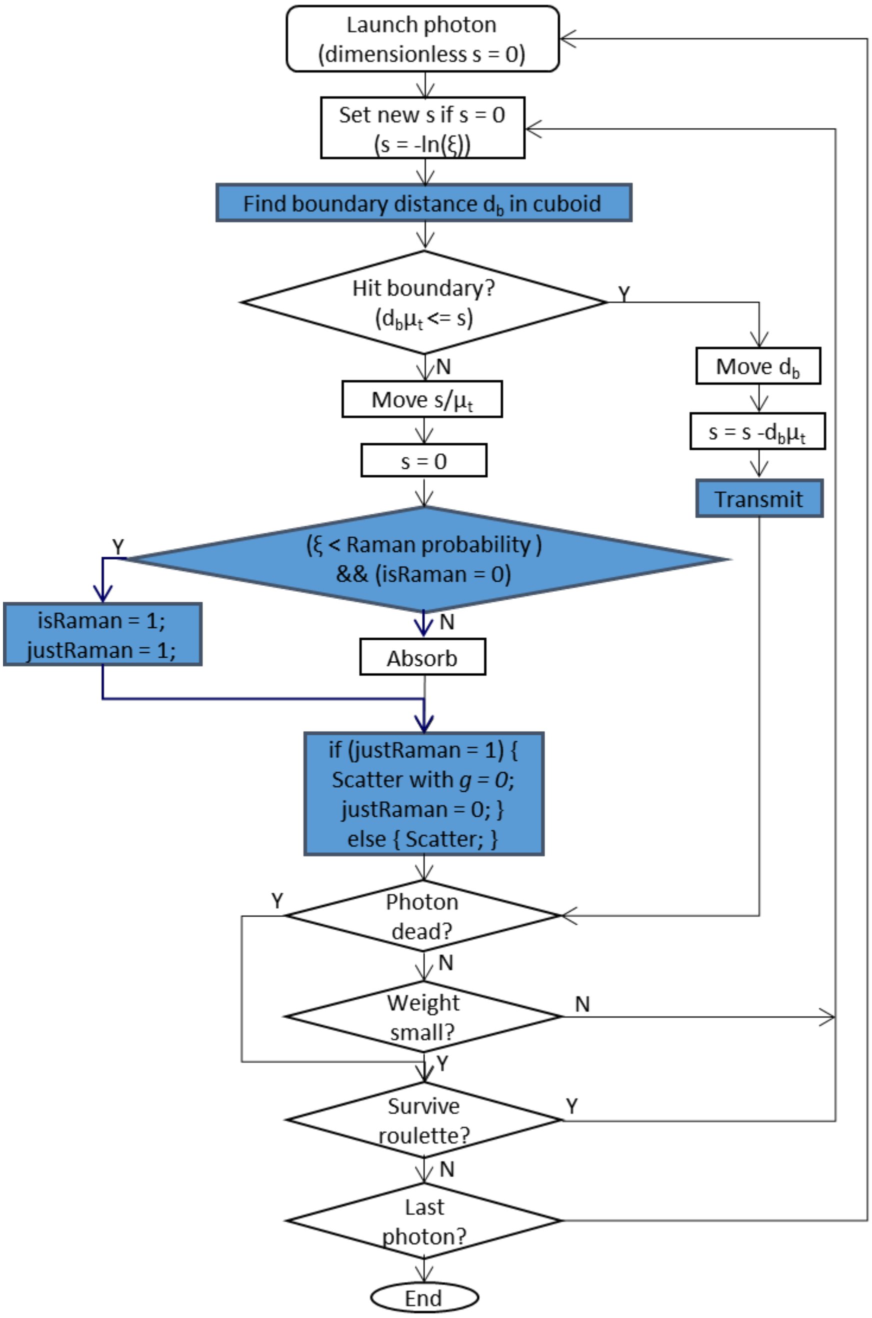


Figure 2: Flow chart for Monte Carlo simulation of Raman scattering for cuboid

a. Changes in MCML.H
    i. Photon structure for pointers - is the photon converted to Raman and is the photon in container or material.
    ii. To read object properties from input file in input structure.
    iii. Add variables for recording the tagged Raman photons escaping from 4 planes.

b. Changes in MCMLIO.C
    i. Function Readparm was updated to read object properties.
    ii. Record the photons from respective layers (2 layers) exiting at respective planes (4 planes).
    iii. Initialize 8 matrices to record the output.
    iv. Compute the x and y co-ordinates from the height and width given as input for the container and material boundary planes.

c. Changes in MCMLGO.C
    i. Function LaunchPhoton to initialize the Raman layer 0.
    ii. Functions checkOnPlane, hitBox, HitBoundaryBox, CrossOrNotBox are newly added to check the boundary conditions of the photon.
    iii. Functions RecordRCon_xy, RecordRCon_yz, RecordRCon_xz, RecordTCon_xy, RecordRMat_xy, RecordRMat_yz, RecordRMat_xz, and RecordTMat_xy are the functions added to record the 8 matrices.
    iv. Boundary conditions are rechecked in HopDropSpinInTissue function.
    v. checkForRaman is called after each Rayleigh scattering to see if the photon is to be converted to Raman.

None of the functions in file MCMLNR.C and MCMLMAIN.C were changed.

4. Output template

```
InParm                          # Input parameters. cm is used.
temp_out.mco        A                     # output file name, ASCII.
40000                           # No. of photons
0.005     0.002                 # dz, dr [cm]
2         1         1           # No. of dz, dr, da.
0.005     0.005                 # dx, dy [cm]
2         2                     # No. of dx, dy.
1                               # Isotropic scatter of Raman.
5         4         2.5       0.1                 # lt, bt, ht, tn of box[cm]
#n        mua       mus       g         ramProb   # One line for each layer
1                                                 # n for medium above
1         0         250       0.9       1E-006              # layer 1
1         0         500       0.9       1E-006              # layer 2
1                                                 # n for medium below
Total Raman photons = 102
# Rd[x][y]. [1/(cm2sr)].
# Rd[0][0], [0][1],..[0][nx-1]
# Rd[1][0], [1][1],..[1][nx-1]
# ...
# Rd[ny-1][0], [ny-1][1],..[nx-1][ny-1]
Rd_xy
2         2                     #nx,ny
RdCon
 1.0000E-004
 0.0000E+000
 0.0000E+000
 0.0000E+000
RdMat
 1.3750E-003
 0.0000E+000
 0.0000E+000
 0.0000E+000
# Tt[x][y]. [1/(cm2sr)].
# Tt[0][0], [0][1],..[0][nx-1]
# Tt[1][0], [1][1],..[1][nx-1]
# ...
# Tt[ny-1][0], [ny-1][1],..[nx-1][ny-1]
Tt_xy
2         2                     #nx,ny
TtCon_xy
 0.0000E+000
 0.0000E+000
 0.0000E+000
 0.0000E+000
TtMat_xy
 4.0000E-004
 0.0000E+000
 0.0000E+000
 0.0000E+000
# Tt[y][z]. [1/(cm2sr)].
# Tt[0][0], [0][1],..[0][ny-1]
# Tt[1][0], [1][1],..[1][ny-1]
# ...
# Tt[nz-1][0], [nz-1][1],..[ny-1][nz-1]
Tt_yz
2         2                     #ny,nz
TtCon_yz
 0.0000E+000
 0.0000E+000
 0.0000E+000
 0.0000E+000
TtMat_yz
 0.0000E+000
 1.0000E-004
 0.0000E+000
 0.0000E+000
# Tt[x][z]. [1/(cm2sr)].
# Tt[0][0], [0][1],..[0][nz-1]
# Tt[1][0], [1][1],..[1][nz-1]
# ...
# Tt[nz-1][0], [nz-1][1],..[nx-1][nz-1]
Tt_xz
2         2                     #nx,nz
TtCon_xz
 0.0000E+000
 0.0000E+000
 0.0000E+000
 0.0000E+000
TtMat_xz
 0.0000E+000
 3.2500E-004
 0.0000E+000
 0.0000E+000
```

Along with the total number of Raman photons generated, the tagged Raman photons escaping from the simulation geometry is recorded. Table 1 summarizes the plane, the layer in which the photons are generated and the variable in which they are recorded. The number of grids in each of these matrices are re-emphasised before each of the matrices, for reference.

Variable name as seen in output file	**Photon generated in**	**Raman photon escaping from plane**
RdCon	Container	xy (z = 0)
RdMat	Material	xy (z = 0)
TtCon_xy	Container	xy (z = depth of cuboid)
TtMat_xy	Material	xy (z = depth of cuboid)
TtCon_yz	Container	yz (x = +(height of cuboid/2)
TtMat_yz	Material	yz (x = +(height of cuboid/2)
TtCon_xz	Container	xz (y = +(width of cuboid/2)
TtMat_xz	Material	xz (y = +(width of cuboid/2)

5. Bibliography

1. Algorithm for semi-infinite turbid medium by S. A. Prahl, M. Keijzer, S. L. Jacques, A. J. Welch, *SPIE Institute Series* Vol. IS 5 (1989), and by A.N. Witt, *The Astrophysical journal Supplement Series* **35**, 1-6 (1977).

2. Monte Carlo simulation of light transport in tissue for optimizing light delivery in photoacoustic imaging of the sentinel lymph node, V. Periyasamy and M. Pramanik, *Journal of Biomedical Optics* **18**(10), 106008 (2013).

3. Monte Carlo simulation of light transport in turbid medium with embedded object - spherical, cylindrical, ellipsoidal, or cuboidal object embedded within multilayered tissues, V. Periyasamy and M. Pramanik, *Journal of Biomedical Optics* **19**(4), 045003 (2014).

4. Experimentally validated Raman Monte Carlo simulation for a cuboid object to obtain Raman Spectroscopic signature for hidden material, V. Periyasamy, S. Sil, G. Dhal, F. Ariese, S. Umapathy, and M. Pramanik, *Journal of Raman Spectroscopy*, 2015