# Modeling tissue optics using Monte Carlo modeling: a tutorial - art. no. 68540T

**Article** *in* Proceedings of SPIE - The International Society for Optical Engineering · March 2008

DOI: 10.1117/12.776997



Steven L Jacques
Oregon Health and Science University

Invited Paper

# Modeling tissue optics using Monte Carlo modeling: a tutorial

Steven L. Jacques[a,b]
[a]Biomedical Engineering, Oregon Health & Science Univ., 3303 SW Bond St., Portland, OR, 97239
[b]Dermatology, Oregon Health & Science University, 3303 SW Bond St., Portland, OR, USA 97239

## ABSTRACT

A brief introduction to time-resolved and steady-state Monte Carlo modeling is presented. The methods of time-resolved and steady-state Monte Carlo simulations of photon trajectories within a tissue are presented. An example simulation using Monte Carlo Multi-Layered (`mcml`) demonstrates the spatial distributions of power deposition and fluence rate within a layered tissue with different optical properties (absorption, reduced scattering) in each layer.

**Keywords:** Monte Carlo, photon migration, optical transport diagnostic imaging, dosimetry

## 1. INTRODUCTION

Monte Carlo simulations show the expected movement of individual photons, which are treated as particles of light that behave according to certain probability density functions for their movement. Figure 1 shows 25 photons propagating into a tissue. Some photons terminate due to absorption. Many others escape the tissue as observable reflectance.

Monte Carlo simulations provide a flexible approach toward light transport that yields maps of the light distributions in tissues induced by a pulsed or continuous laser or an alternative light source. This report discusses the methods of time-resolved and steady-state Monte Carlo simulations of photon trajectories within a tissue. An example simulation using Monte Carlo Multi-Layered (`mcml`) demonstrates the spatial distributions of power deposition and fluence rate within a layered tissue with different optical properties (absorption, reduced scattering) in each layer.

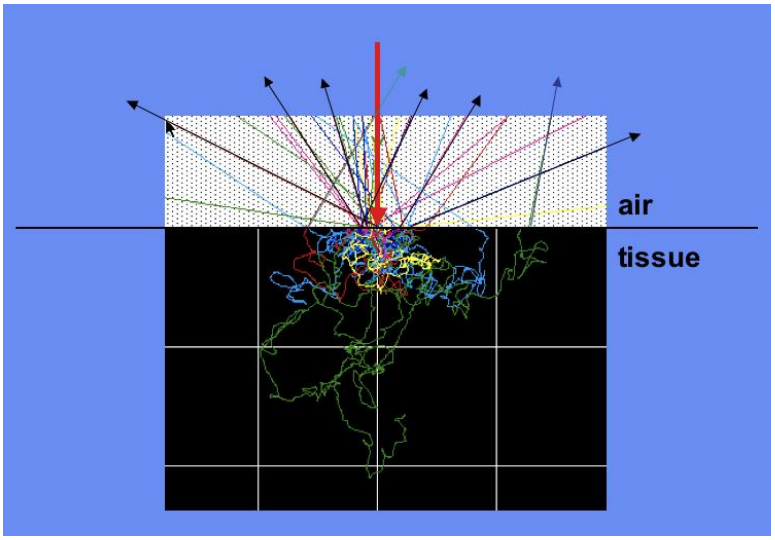


Fig. 1. Monte Carlo simulation of 25 red (630 nm) photons propagating in esophageal tissue after launching as a collimated beam directed perpendicular to the tissue surface at the origin (red arrow).. The grid shows 5–mm spacings. Some photons terminate due to absorption Many photons escape as observable reflectance.

Optical Interactions with Tissue and Cells XIX, edited by Steven L. Jacques, William P. Roach, Robert J. Thomas,
Proc. of SPIE Vol. 6854, 68540T, (2008) · 1605-7422/08/$18 · doi: 10.1117/12.776997

## 2. MONTE CARLO SAMPLING OF A PROBABILITY DENSITY FUNCTION

Monte Carlo is a generic method for sampling a probability density function, p(x). The p(x) could be the probability density function for step size of the photon (s), for angle of deflection (π), or for azimuthal angle of scatter (π).

Consider a p(x) where x is some variable that lies between a and b (see Fig. 2). The integration of p(x) from x = a to x = $x_1$ yields the probability distribution function $F(x_1)$.

On the left side of Fig. 2, the function p(RND) = 1 over the range x = 0 to 1. Hence, the $F(RND_1) = RND_1$.

The Monte Carlo method consists of equating $F(RND_1) = F(x_1)$, and rearranging the equation to solve for $x_1$ in terms of $RND_1$. Then the computer can pick $RND_1$, where $0 < RND_1 \leq 1$, and specify $x_1$.

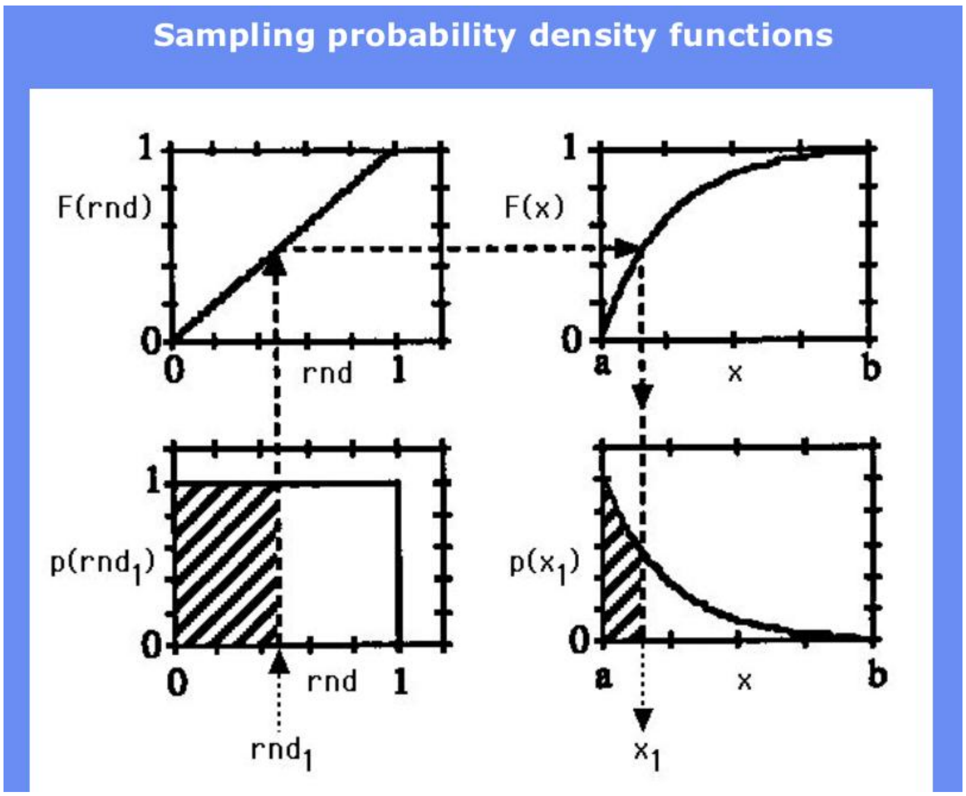


Fig. 2. Monte Carlo sampling of probability density function p(x). A choice of $rnd_1$ is mapped via $F(x_1) = rnd_1$ to predict $x_1$.

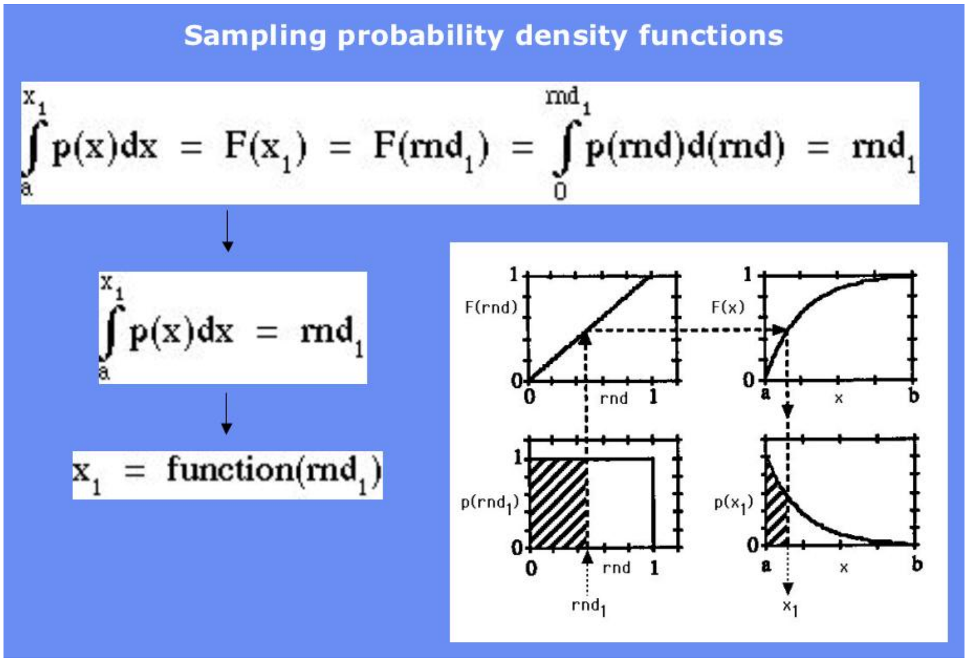


Fig. 3. Monte Carlo sampling of probability density function `p(x)`. A choice of $rnd_1$ is mapped via $F(x_1) = rnd_1$ to predict $x_1$.

## 3. STEADY-STATE MONTE CARLO

A simple steady-state Monte Carlo program (`mc321.c`) is available at

`http://omlc.ogi.edu/classroom/ece532/class4/ssmc/mc321.c`

`http://omlc.ogi.edu/software/mc/mc321.c`

This program demonstrates the propagation of photons by steps of stepsize s between locations of interaction between photon and the tissue by either absorption or scattering. The stepsize is

$$s = \frac{-\ln(RND)}{\mu_s + \mu_a} \tag{1}$$

The steady-state propagation rule is summarized in Fig. 4. The photon is launched with an initial weight W = 1. After the photon steps to the next *interaction* point, a fraction `(1—a)` of the weight is deposited in the local bin, `A(ir,iz) = (1- a)W` or `A(ir,iz)` or `A(iz)` or whatever binning scheme is chosen. The photon retains a fraction `a` of the weight,, `W = aW`. The binning can follow any desired format, for example:

| | | |
|---|---|---|
| `A(ix,iy,iz)` | `3D Cartesian grid` | `V(ix,iy,iz) = dx dy dz` |
| `A(ir)` | `1D spherically symmetric grid` | $V(ir) = 4\pi r^2 dr$ |
| `A(ir,iz)` | `2D cylindrically symmetric grid` | $V(ir) = 2\pi r dr dz$ |
| `A(iz)` | `1D planar grid` | $V(ir) = (1\ cm^2)dz$ |

where `a` is the `albedo` of the tissue:

$$a = \frac{\mu_s}{\mu_s + \mu_a} \tag{2}$$

After propagating many photons, $N_{photons}$, typically $10^4$ to $10^7$ photons, the array A(ir,iz) is filled with photon weight. Each element in the array is normalized to yield the *fraction of total delivered photon weight deposited per unit volume* based on the photon weight that was deposited in that element:

$$\mathtt{A(ir,iz)} = \frac{\mathtt{A(ir,iz)}}{\mathtt{N_{photons}V(ir,iz)}} \tag{3}$$

The final value of A(ir,iz) is in units of [1/cm$^3$]. The value of V(ir,iz) depends on the geometry, as described above for the various geometries of recording.

The relative fluence rate F(ir,iz) [1/cm$^2$] = [W/cm$^2$ per W delivered] is calculated:

$$\mathtt{F(ir,iz)} = \frac{\mathtt{A(ir,iz)}}{\mu_a\mathtt{(ir,iz)\ N_{photons}V(ir,iz)}} \tag{4}$$

where $\mu_a$(ir,iz) is the local absorption coefficient [cm$^{-1}$] in the (ir,iz) bin.

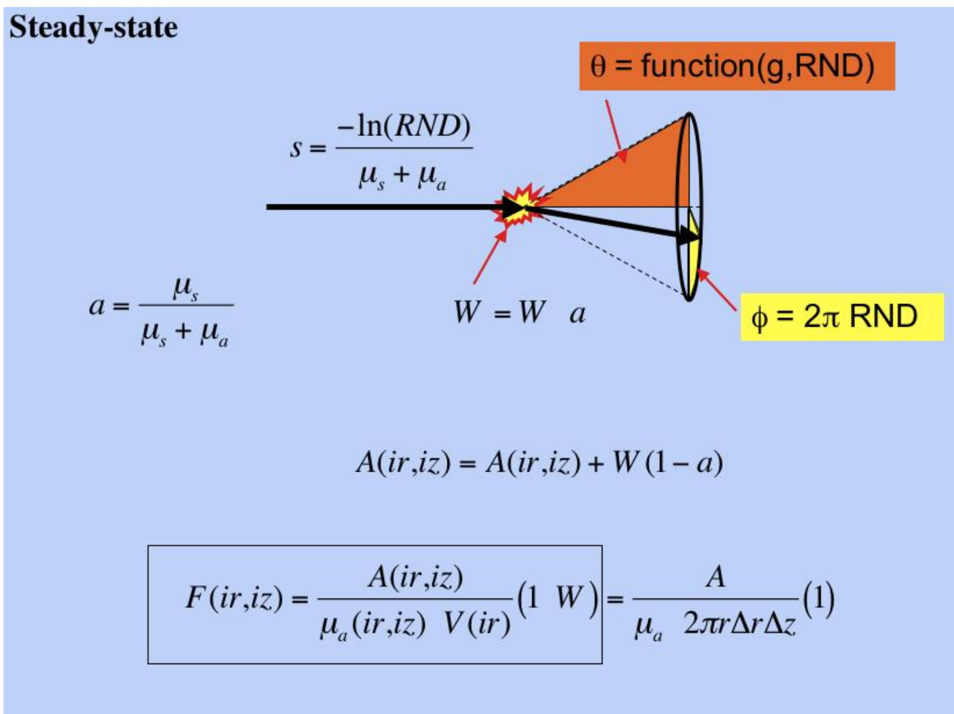


Fig. 4. Steady-state Monte Carlo.

## 4. TIME-RESOLVED MONTE CARLO

A simple time-resolved Monte Carlo program (`trmc.c`) is available at

`http://omlc.ogi.edu/classroom/ece532/class4/trmc/trmc.c`

`http://omlc.ogi.edu/software/mc/trmc.c`

This program demonstrates the propagation of photons by moving the photon until the photon reaches a desired time point, $t_{snapshop}$, for storing a snapshot of the photon distribution. The photon takes steps of stepsize s between locations of *scattering*. The stepsize is

$$s = \frac{-\ln(RND)}{\mu_s} \tag{5}$$

The time-resolved propagation rule is summarized in Fig. 5. The photon is launched with an initial weight W = 1, which never changes. The photon takes multiple steps until its next step will pass a total pathlength L = $ct_{snapshop}$. When that occurs, the photon takes a partial step to reach the total pathlength $ct_{snapshop}$. The local bin, C(ir,iz), is updated as `C(ir,iz,1) = C(ir,iz,1) + 1`. Then the photon completes its full step, and continues propagating until it reaches the next desired timepoint. At that point again a partial step is taken and the C(ir,iz,2) array is update. This repeats, creating a set of C(ir,iz,it) where it denotes one of the time points of interest.

After propagating many photons, $N_{photons}$, the array C(ir,iz,it) is filled with photons. Each element in the array is normalized to yield the *fraction of total delivered photon weight deposited per unit volume* at that time point, based on the photon weight that was deposited in that element:

$$\mathtt{C(ir,iz,it)} = \frac{\mathtt{C(ir,iz,it)}}{\mathtt{N_{photons}V(ir,iz,it)}} \tag{6}$$

The final value of C(ir,iz,it), for each it$^{th}$ time point, is in units of [1/cm$^3$].

The relative fluence rate F(ir,iz) [1/cm$^2$] = [W/cm$^2$ per W delivered] is calculated:

$$\mathtt{F(ir,iz)} = c\,\frac{\mathtt{A(ir,iz)}}{\mathtt{N_{photons}V(ir,iz)}} \tag{7}$$

where $\mu_a$(ir,iz) is the local absorption coefficient [cm$^{-1}$] in the (ir,iz) bin.

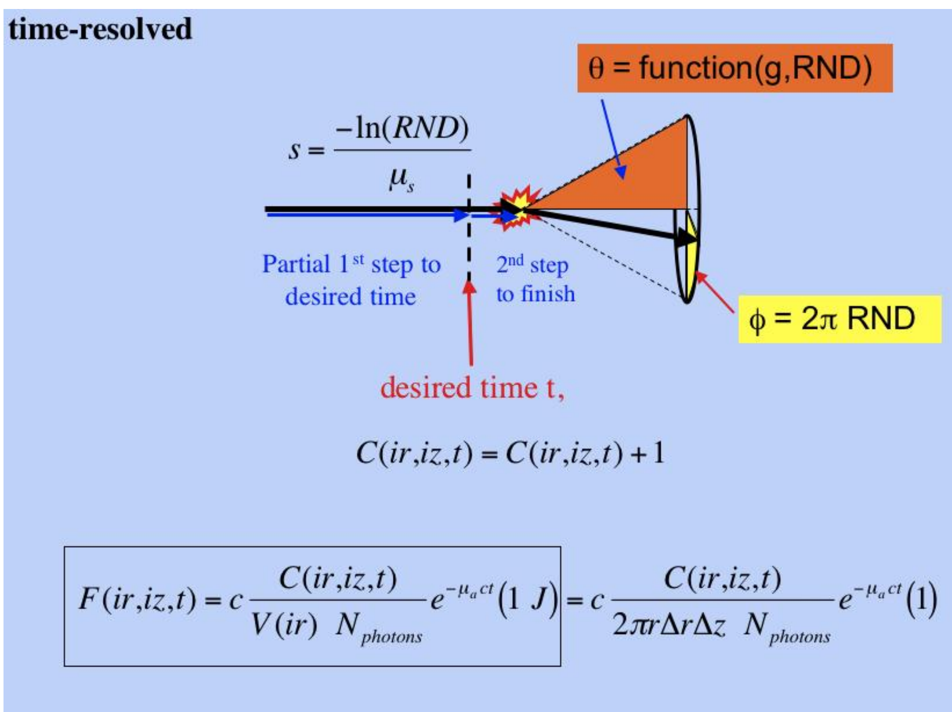


Fig. 5. Time-resolved Monte Carlo

## 5. MULTI-LAYERED MONTE CARLO (MCML)

A Monte Carlo program that handles multiple planar layers, each with its own optical properties and thickness, is available at

`http://omlc.ogi.edu/software/mc/index.html`

The program, written in ANSI Standard C, is run as a compiled program, and uses an input text file to specify the simulation and the output file name where results are recorded. An example input file is shown in Table 1.

| | | | | | |
|---|---|---|---|---|---|
| 1.0 | | | # file version | | |
| 1 | | | # number of runs | | |
| | | | | | |
| example.mco | A | | # output filename, ASCII/Binary | | |
| 100000 | | | # No. of photons | | |
| 0.010 | 0.010 | | # dz, dr for OUTPUT | | |
| 100 | 100 | 1 | # No. of bins, Nz, Nr, Na for OUTPUT | | |
| 3 | | | # No. of layers | | |
| # n | mua | mus | g | d | # One line for each layer |
| 1.00 | | | # n for medium above. | | |
| 1.400 | 0.40 | 15 | 0.700 | 0.1000 | |
| 1.400 | 3.00 | 260 | 0.900 | 0.1000 | |
| 1.400 | 0.05 | 100 | 0.900 | 1.0000 | |
| 1.0 | | | # n for medium below. | | |

Table. 1: Example of an input file for MCML.

Figures 3-5 show the output from an MCML simulation using the input file in Table 1. The output was stored in a file called `example.mco` as an ASCII file. This file was in turn read by a MATLAB program `lookmcml.m` (and subroutine `getmcml.m`) that reads the output file and prepares the Figs. 3-5, using a colormap generated by `makec2f.m`. These programs are available at:

`http://omlc.ogi.edu/software/mc/lookmcml.m`

`http://omlc.ogi.edu/software/mc/getmcml.m`

`http://omlc.ogi.edu/software/mc/makec2f.m`

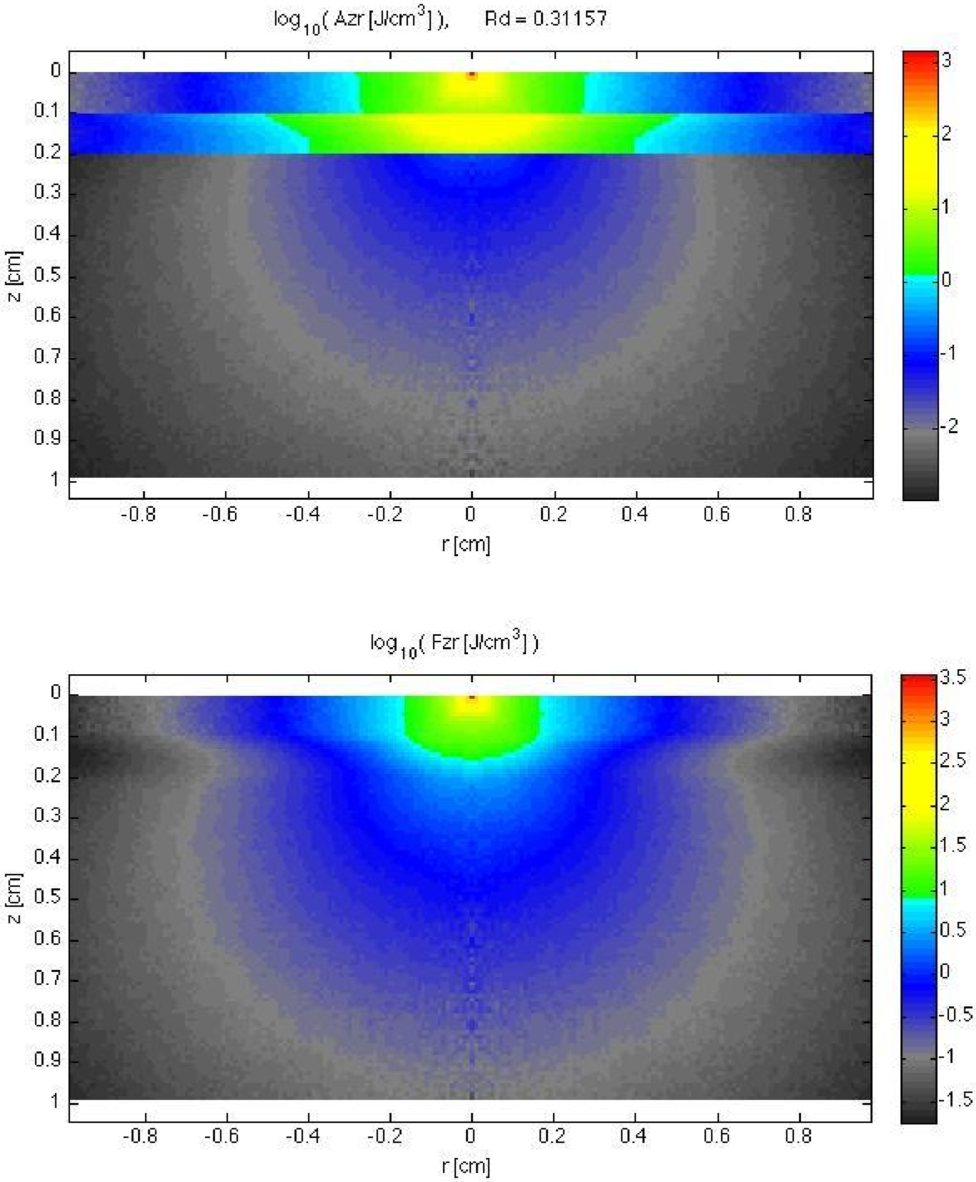


Fig. 3. Monte Carlo output from MCML. (TOP) The energy deposition A(z,r) [W/cm$^3$ per W delivered]. (BOTTOM) The fluence rate F(z,r) [W/cm$^2$ per W delivered]. The light is delivered as a point source of collimated light at r = 0, z = 0. The colorbar is a logscale. Note how A(z,r) is discontinuous at the boundaries between layers with different absorption coefficients, while F(z,r) is continuous.

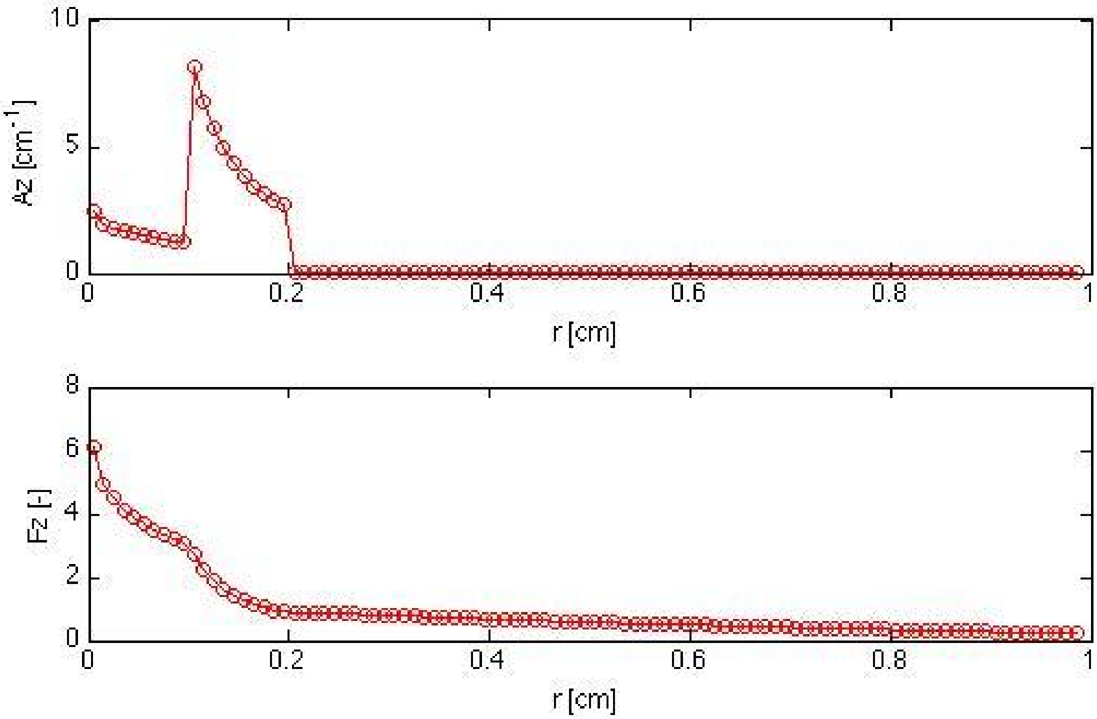


Fig. 4. Monte Carlo output of MCML showing the 1-D energy deposition A(z) [$cm^{-1}$] and fluence rate F(z) [dimensionless] in response to a broad beam of light [$W/cm^2$], obtained by integrating the A(z,r) and F(z,r) over all r at each depth z.

$$A(z) = \sum_r \left( A(z,r) \;\; 2\sum r \sum r \right)$$

$$F(z) = \sum_r \left( F(z,r) \;\; 2\sum r \sum r \right)$$

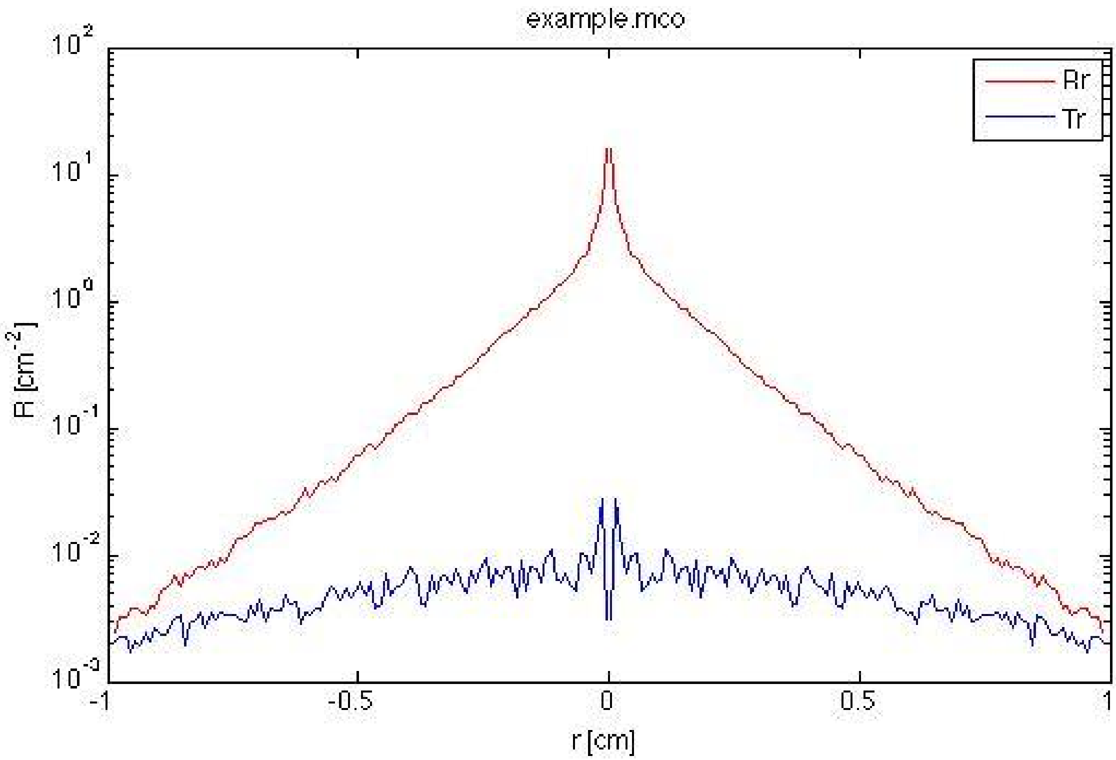


Fig. 5. Monte Carlo output from MCML, showing the spatially resolved reflectance (Rr [$cm^{-2}$]) and transmittance (Tr [$cm^{-2}$]) as a function of radial position r [cm].