

Tutorial on Monte Carlo simulation of photon transport in biological tissues [Invited]

STEVEN L. JACQUES* 

Bioengineering, University of Washington, 3720 15th Ave NE, Seattle, WA 98195-5061, USA

**stevjacq@uw.edu*

Abstract: A tutorial introduction to Monte Carlo (MC) simulation of light propagation in biological tissues. MC statistical sampling is introduced, the basic design of a MC program is explained, and examples of application in biomedicine are presented.

© 2023 Optica Publishing Group under the terms of the [Optica Open Access Publishing Agreement](#)

1. Introduction

The penetration of light into and within a biological tissue is the key to many diagnostic and therapeutic uses of lasers and other light sources in medicine and biology. The dose of light experienced by a target structure within a tissue, eg., a blood vessel or a tumor, guides the irradiance and exposure time required for a particular photochemical, photothermal, or photomechanical effect. Understanding how photons propagate within and escape from a tissue is necessary for proper interpretation of optical diagnostic measurements. The field of biomedical optics and biophotonics is currently very actively using computer simulations of light transport to guide clinical protocols and develop optical tools for medicine and biology.

Monte Carlo (MC) simulations of light transport in media with absorption and scattering properties has long been an important tool in many fields. The MC method is a statistical sampling of probability density functions, and an early application was study of the movement of neutrons in fissile materials for nuclear reactions; see [1] for a history of early MC development. MC simulations have been widely used in astrophysical, atmospheric, oceanic, and now biomedical applications. This paper does not attempt a review of the field. Rather, this tutorial introduces MC to newcomers to the field. At its heart, MC is quite simple and demystifying the computation is a goal of the paper.

There have been many MC programs written by many investigators. Some recent examples are *mcml* [2], *MCmatlab* [3], and *mcx* [4]. Others have created websites that allow users to run simulations on-line, such as the Biophotonics Initiative [5], *mcxspace* [6], Cloud based Monte Carlo [7], and Multi-Scattering [8].

The paper begins with an introduction to Monte Carlo sampling, then discusses the basic structure of an MC program, and concludes with three example problems: (1) How does skin architecture affect skin color? (2) What exposure time is needed for photobiomodulation of the knee? (3) Will a pulsed laser treatment reach the bulb of a hair follicle?

2. Monte Carlo sampling

2.1. Concept

Monte Carlo sampling is the random sampling of a probability density function, $p(x)$, to pick a value x . This section illustrates MC sampling using a bucket of marbles, then presents the general approach toward finding an analytic expression that uses a random number (RND) to pick a value of x . Two examples are given: (1) picking a stepsize for a photon to propagate before being scattered, and (2) picking a launch position for launching each photon as a large number of photons are delivered to a tissue as a collimated Gaussian beam.

Bucket of marbles

Consider a bucket that you are going to fill with a N marbles, for example $N = 1000$. Divide the range of x into discrete steps of x , for example 100 steps of size $dx = 0.01$ between 0 and 1. Calculate the corresponding discrete values of $p(x)$ that correspond to each discrete value of x , where $p(x)$ is a normalized probability density function. For each x value, write the value of x on a set of $Np(x)dx$ marbles and place these marbles in the bucket. For example, if $x = 0.2$ and $p(x) = 5e^{-5x}$, $x = 1.8394$, then $Np(x)dx$ equals $\text{round}((1000)(1.8394)(0.01)) = 18$ marbles, on which you write the value 0.2 and place in the bucket. After doing this for all your steps of x , you now have a bucket filled with 1000 labeled marbles, but more are labeled with small values of x and fewer are labeled with large values of x , since $p(x)$ is an exponential decay function.

Now proceed to sample $p(x)$ by picking a marble from the bucket. The value on the marble is your x_1 . Put the marble back into the bucket, shake the bucket, and draw a second marble, yielding a new x_1 . Repeat. This process of sampling marbles in the bucket is a simple version of the Monte Carlo method. Figure 1 illustrates the concept of the bucket filled with marbles.

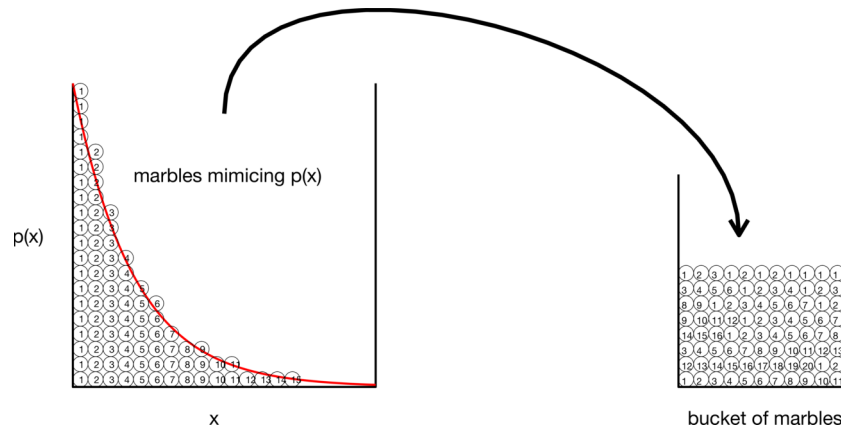


Fig. 1. Monte Carlo sampling. On the left, marbles distributed as $p(x)$ are labeled by their x value. On the right, the labeled marbles have filled a bucket. Randomly drawing a marble from the bucket yields a marble labeled by x .

Sampling diagram

Now consider using the Monte Carlo method to sample a generic probability density function $p(x)$, normalized such that

$$\int_a^b p(x)dx = 1 \quad (1)$$

The corresponding probability distribution function is $F(x)$:

$$F(x) = \int_a^x p(x')dx' \quad (2)$$

Now consider a random number generator that generates a random number RND . The $p(RND)$ equals a constant 1 over the interval 0 to 1, since any value of RND has an equal chance. Consequently, $F(RND) = RND$, which is a linear increase from 0 to 1 with a slope of 1. To sample the $p(x)$, equate the probability distribution functions, $F(RND) = F(x)$, in other words, let $F(x) = RND$. Then solve for x to yield a new equation $x = \text{function}(RND)$. Now, the sampling of $p(x)$ uses a randomly generated RND_1 to produce a value x_1 . This is a computational approach toward Monte Carlo sampling.

This strategy is illustrated in Fig. 2, using $p(s) = \mu_s e^{-\mu_s s}$, which is the probability density function for the step size (s) a photon will take before being scattered. The $p(s)$ is properly normalized (integral of $p(s)$ from 0 to ∞ equals unity). The function $F(s)$ is

$$F(s) = \int_0^s p(s') ds' \quad (3)$$

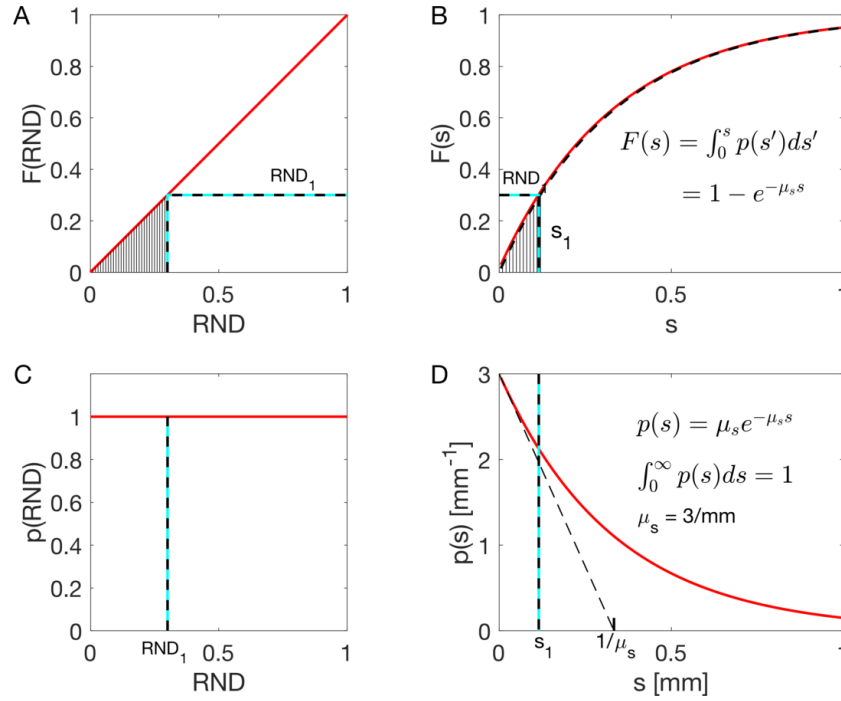


Fig. 2. Monte Carlo sampling of $p(s)$ using a random number RND . A choice of RND_1 connects to a value s_1 by equating the $F(r)$ to RND . The hatched areas under the curves $p(RND)$ and $p(s)$ are equal.

Choosing a random number value RND_1 , equate RND_1 and $F(s)$ and solve for s , which specifies s_1 , the sampled value of s :

$$RND_1 = F(s) = 1 - e^{-\mu_s s} \quad (4)$$

$$s_1 = \frac{-\ln(RND_1)}{\mu_s}$$

A second example is shown in Fig. 3, illustrating the sampling of a 2D Gaussian beam with a $1/e$ radius of w . This example is useful for launching a photon into tissue using a collimated Gaussian beam. The goal is to sample the distribution for radial position (r) of photon in the 2D Gaussian beam. Therefore, the rings of integration ($2\pi r dr$) must be included and the probability properly normalized: $p(r) = e^{-(r/w)^2} \frac{2\pi r}{\pi w^2}$. The integral of $p(r)$ is $F(r) = 1 - e^{-(\frac{r}{w})^2}$. Solving for a particular r_1 using a particular RND_1 ,

$$r_1 = w \sqrt{-\ln(RND_1)} \quad (5)$$

This sampling method to specify the launch position for each photon can be applied to a variety of beam shapes amenable to Eq. (2). For beams not amenable to Eq. (2), a lookup table can be used, analogous to the bucket of marbles.

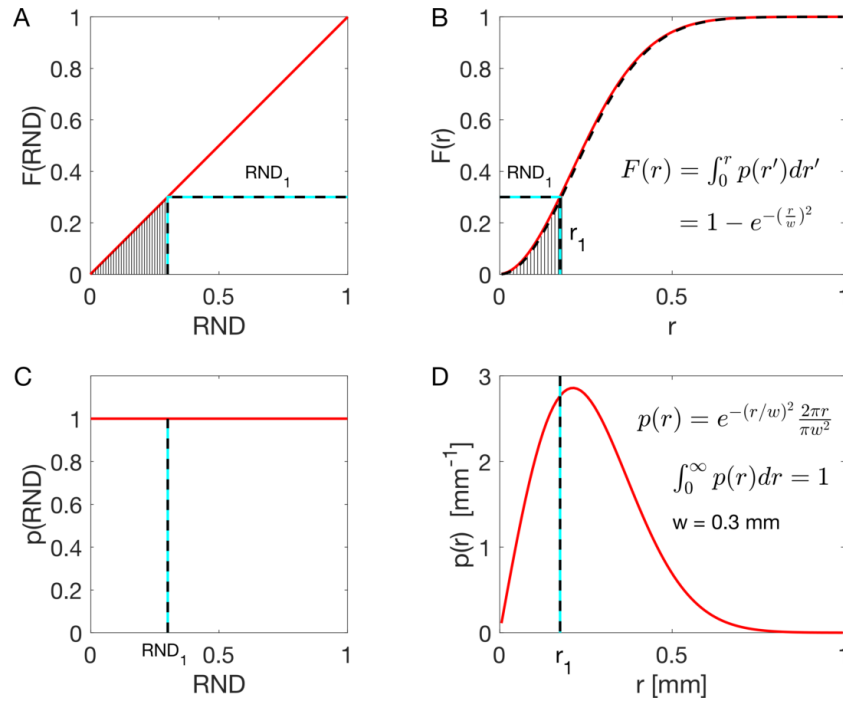


Fig. 3. Monte Carlo sampling to specify the radial position (r) for launching a photon, such that a population of launched photons would mimic a collimated Gaussian beam with a $1/e$ radius of $w = 3$ mm.

3. Monte Carlo program

This section discusses the basis elements of a MC simulation. The program has three input files: (1) a tissue volume, which specifies the 3D distribution of tissue types (epidermis, dermis, blood, etc.), (2) a list specifying the optical properties of each tissue type (absorption coefficient, scattering coefficient, etc.), (3) a list of run parameters (number of photons to launch, voxel sizes, beam size, etc.) that control the simulation.

3.1. Tissue volume

Consider a tissue volume modeled as a cubic array, $T(x,y,z)$, where each voxel is assigned an integer value that specifies a particular tissue type. A complex tissue, such as skin with its several layers (epidermis, papillary dermis, vascular plexus, reticular dermis) along with an embedded tumor and a large blood vessel would have 6 tissue types. The $T(x,y,z)$ voxels would be filled with integers between 1 to 6 to create the tissue structure. The following example shows MATLAB code for generating a two-layer skin tissue (epidermis, dermis) with a $200\text{-}\mu\text{m}$ -dia. blood vessel (as shown in Fig. 4). There is specular reflectance of the incident beam at the air/tissue interface ($r_{sp} = (\frac{1.4-1.0}{1.4+1.0})^2 = 0.0278$, $n_{air} = 1.0$, $n_{tissue} = 1.4$). There is also total internal reflectance when photons strike the air/tissue boundary obliquely as they attempt to escape the tissue ($r_i \approx 0.53$).

There are a variety of ways to generate a tissue volume $T(x,y,z)$ for input to a MC simulation, this is just one example.

```

Ny=400; Nx=Ny; Nz=Ny;
dx = 0.010; % mm, 10-um bin size.
dz = dx;
x = ([1:Nx]' -Nx/2)*dx;
z = [1:Nz]'*dx;
T = double(zeros(Nx,Ny,Nz)); % number of voxels in tissue cube
T = T + 4; % fill background with skin (4 = dermis)
zsurf = 1; % position of air/skin surface
for iz=1:Nz % for every depth z(iz)
    % air
    if iz<=round(zsurf/dz) % dz = voxel size
        T(:, :, iz) = 1; % 1 = air
    end
    % epidermis (100 um thick)
    if iz>round(zsurf/dz) & iz<=round((zsurf+0.100)/dz)
        T(:, :, iz) = 5; % 5 = epidermis
    end
    % blood vessel @ xc, zc, radius, oriented along y axis
    xc = 0; % [mm], center of blood vessel
    zc = Nz/2*dz; % [mm], center of blood vessel
    vesselradius = 0.5; % blood vessel radius [mm]
    for ix=1:Nx
        xd = x(ix) - xc; % vessel, x distance from vessel center
        zd = z(iz) - zc; % vessel, z distance from vessel center
        r = sqrt(xd^2 + zd^2); % r from vessel center
        if (r<=vesselradius) % if r is within vessel
            T(ix, :, iz) = 3; % 3 = blood. Extend pattern over all y.
        end
    end %ix
end % iz

```

3.2. Tissue optical properties

For each tissue type, the optical properties can be generated as a function of wavelength by specifying tissue parameters:

1. B = blood volume fraction ($B=1$ indicates blood at 150 g/l hemoglobin, or 2.3 mM)
2. S = oxygen saturation of hemoglobin in mixed arteriovenous vasculature
3. W = water content as volume fraction
4. M = melanosome volume fraction (eg., $M \approx 0.07$ for tanned Caucasian)
5. F = volume fraction of fat
6. a = scattering strength ($a = \mu_s(1 - g_1)$ at reference wavelength, eg., $\lambda_{ref} = 500$ nm)
7. f_{ray} = fraction of Rayleigh scatter at λ_{ref}
8. f_{mie} = fraction of Mie scatter at λ_{ref} ($f_{mie} = 1 - f_{ray}$)
9. b_{mie} = scattering power for Mie scatter

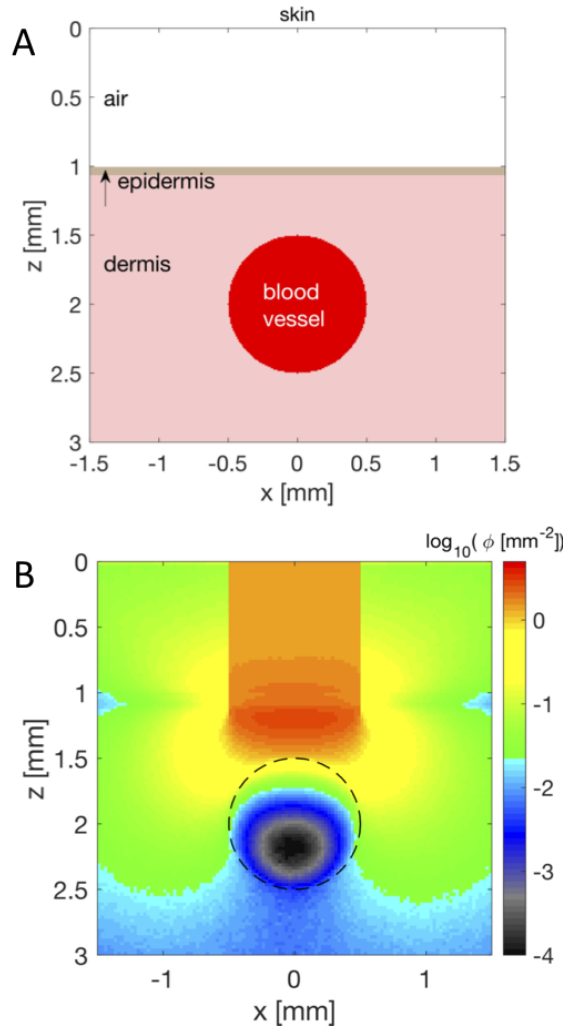


Fig. 4. Example of skin tissue with a large blood vessel. The air/skin interface is at $z = 1$ mm. (A) Tissue structure showing $T(z,x)@y=0$. There is a 100- μm -thick epidermis and a 1-mm-dia. blood vessel centered at 1 mm below the skin surface. (B) Relative fluence, ϕ [mm^{-2}] in response to a 1-mm-diameter 500-nm-wavelength beam of light.

10. g_1 = anisotropy of scatter ($g_1 = \langle \cos(\theta) \rangle$, θ is polar angle of photon deflection)

The above tissue parameters are used to calculate the optical properties for a tissue type:

$$\begin{aligned} \mu_a &= BS\mu_{a.oxy} + B(1 - S)\mu_{a.deoxy} + W\mu_{a.water} + F\mu_{a.fat} + M\mu_{a.mel} \\ \mu'_s &= a \left(f_{ray} \left(\frac{\lambda}{\lambda_{ref}} \right)^{-4} + (1 - f_{ray}) \left(\frac{\lambda}{\lambda_{ref}} \right)^{-b_{mie}} \right) \\ \mu_s &= \frac{\mu'_s}{1 - g_1} \end{aligned} \quad (6)$$

The parameter μ_a is the absorption coefficient. The parameter $\mu'_s = \mu_s(1 - g_1)$ is the reduced scattering coefficient and is used since the literature on tissue optical properties has routinely measured and reported μ'_s . However, MC simulations use $\mu_s = \frac{\mu'_s}{1 - g_1}$. Alternatively, it is common

to use $\mu'_s = a(\frac{\lambda}{\lambda_{ref}})^{-b}$, which works well in the visible wavelength range. The value of anisotropy g_1 is sparsely specified in the literature for various tissues, and a value of $g_1 = 0.90$ is commonly used. However, g_1 does vary with wavelength and tissue type, and is a parameter deserving more study.

A two-term scattering function can allow separate control of the forward and backward contributions to the overall scattering function [9]. The forward component dominates the penetration of light into a tissue and subsequently the diffuse reflectance due to multiply scattered photons. The backward component dominates the early scatter, perhaps the first 10 scattering events, and the subdiffuse reflectance for the superficial layer of a tissue. The average optical scattering of a tissue may be described by a Plum and Pudding model [10], in which a nucleus (plum) sits within a background of sub-micron scattering structures (pudding).

To illustrate how to use the above tissue parameters, the following excerpt from a MATLAB program `makeTissueList.m` shows the calculation for epidermis, having requested the optical properties for $\lambda = 460$ nm. A matrix of absorption spectra, `MU (nmLIB, 1:5)` has been loaded, in which 5 columns specify the absorption coefficient of 5 tissue parameters versus wavelength (`nmLIB = 300-1000` nm): $\mu_{a,ox}$ for oxygenated hemoglobin in blood, $\mu_{a,deoxy}$ for deoxygenated hemoglobin in blood, $\mu_{a,water}$ for water, $\mu_{a,meI}$ for the interior of a cutaneous melanosome, $\mu_{a,fat}$ for fat. Interpolation yields 5 values for these 5 parameters at a requested wavelength. In this excerpt, a blood-free epidermis with a skin type II is specified. The oxygen saturation S is ignored because there is no blood.

```
nm = 460; % requested wavelength
MU2 = interp1(nmLIB,MU,nm); % horizontal vector
j=5; % pointer to tissue type
tissue(j).name = 'epidermis';
B = 0; % blood volume fraction
S = 0; % saturation of hemoglobin
W = 0.75; % water volume fraction
M = 0.03; % melanosome volume fraction
F = 0; % fat volume fraction
a = 2; % mm-1, scattering strength = reduced scattering coeff. at lambda_ref=500nm
fray = 0.0; % fraction of reduced scattering at 500nm due to Rayleigh scattering
% 1-fray = fraction of reduced scattering at 500nm due to Mie scattering
bmie = 1.0; % Mie scattering power
g1 = 0.90; % anisotropy of scatter
musp = a*(fray*(nm/500).^4 + (1-fray)*(nm/500).^bmie); % reduced scattering
X = [B*S B*(1-S) W M F]'; % vertical vector
tissue(j).mua = MU2*X; % matrix multiplication --> absorption coeff.
tissue(j).mus = a/(1-g1); % scattering coeff.
tissue(j).g1 = g1; % anisotropy of scatter
```

The values of μ_a (`tissue(j).mua`), μ_s (`tissue(j).mus`), and g_1 (`tissue(j).g1`) are then passed to the MC program as part of the input file.

3.3. Launching photons

This section discusses launching photons. Each photon, sometimes called a “photon bundle”, is assigned a weight $W = 1.0$ at launch, and this W will slowly decrease as absorption occurs with every incremental step of the photon (discussed later in the DROP section). A large number of photons ($N_{photons}$) is launched, where the number depends on the output desired. Only a few thousand may be needed to generate total diffuse reflectance (R_d [dimensionless]). Perhaps 10^5 may be needed to generate the point spread function ($R(r)$ [mm^{-1}]). A few million may be needed to generate the 3D fluence distribution ($\phi(x, y, z)$ [mm^{-2}]). In all cases, launching more photons reduces the variance in output.

A useful exercise is to run a set of MC simulations, each with its random number generator initiated with a unique seed. Consequently, the outputs of each simulation will be unique. Then the variance of the outputs can be calculated, which offers an estimate of the precision of the result.

Launch position, x,y,z

Each photon is launched at some position x,y,z. Consider how to launch photons at the tissue surface (z=0) as a collimated Gaussian beam with a $1/e$ radius w . MC sampling specifies the radial position r of each photon that is launched, then the x,y positions are calculated:

$$\begin{aligned} r &= w\sqrt{-\ln(RND)} \\ \beta &= 2\pi RND \\ x &= r \cos(\beta) \\ y &= r \sin(\beta) \end{aligned} \quad (7)$$

Each photon is launched with a “weight” $W=1$. During propagation the photon’s W will drop as photon energy is deposited along the photon’s path, which is discussed later (see Drop).

Photon trajectory, ux,uy,uz

The MC program launches a photon at a position x,y,z with a trajectory vector of unit length specified by ux,uy,uz, where ux is the projection of the vector onto the x-axis, uy projects onto the y-axis, and uz projects onto the z-axis, and therefore $\sqrt{ux^2 + uy^2 + uz^2} = 1$. For example, a photon launched at x=0, y=0, z=0 at an angle of 22.5 degrees ($\pi/8$ radians) off the normal to the surface and parallel to the x-z plane would have a trajectory specified by $ux = \cos(\pi/8)$, $uy = 0$, $uz = \sin(\pi/8)$. Figure 5 illustrates this example.

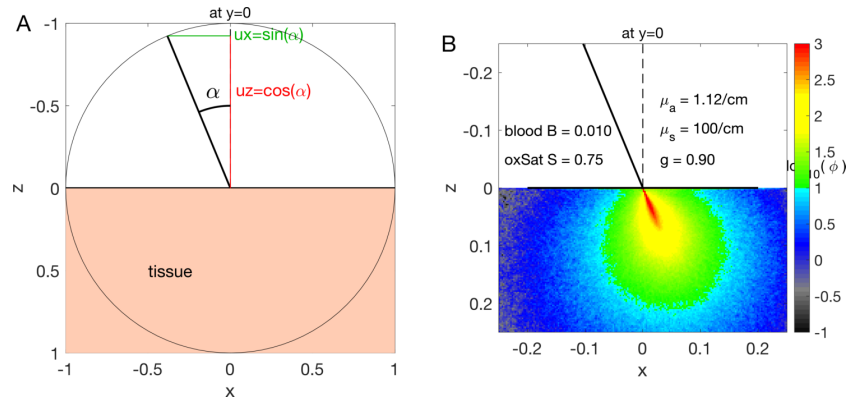


Fig. 5. Launching photons (A) A photon is launched at 22.5 degrees ($\pi/8$ radians) off the normal to the surface at x=0,y=0, parallel to the x-z plane (i.e., an oblique pencil beam). The trajectory is $ux=\sin(\pi/8)$, $uy=0$, and $uz=\cos(\pi/8)$. (B) MC simulation shows the launched photons penetrating a soft tissue (wavelength = 500 nm, $\mu_a = 0.112 \text{ cm}^{-1}$, $\mu_s = 100 \text{ cm}^{-1}$, $g_1=0.90$, for the case of 1% blood volume fraction (B) and 75% oxygen saturation of hemoglobin in the mixed arterio-venous vasculature.)

3.4. Propagating photons

A generic flow diagram for propagating photons in a MC simulation is shown in Fig. 6. The following subsections describe the steps in this flow diagram.

The transport of unpolarized light (a population of photons with randomized orientations of their electric fields) is considered. The basic steps are hop, drop, spin, and check, as in Fig. 6.

Hop

The HOP step samples a step size (s) for the photon before it encounters its next scattering event,

$$s = \frac{-\ln(RND)}{\mu_s} \quad (8)$$

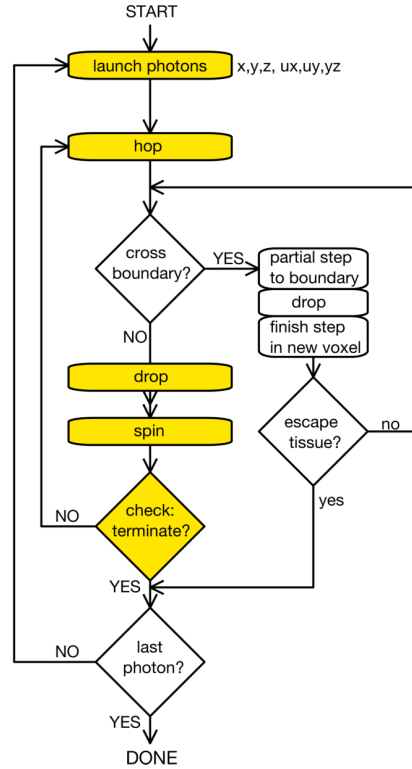


Fig. 6. Flow chart for a simple Monte Carlo simulation. Refraction of transmission if photon crosses boundary between voxels, or trajectory change if reflected, based of Fresnel transmission/reflection.

This step size is weighted exponentially, and is sampled as illustrated in Fig. 2. Alternatively, the program mcml.c uses the expression,

$$s = \frac{-\log(RND)}{\mu_t} \quad (9)$$

where $\mu_t = \mu_a + \mu_s$. The step s is between “interactions”, at which a fraction of the photon weight, $albedo = \frac{\mu_s}{\mu_t}$, survives and a fraction $1 - albedo$ is absorbed. The two methods give the same results when many photons are launched. However, Eq. (8) is perhaps more appropriate when considering the first few scattering events in a tissue, such as the study of photon scatter in a superficial epithelium. This tutorial uses the first method (Eq. (8)).

The position of a photon is updated by the step size s , where the movement in each x,y,z direction is scaled by the trajectory vectors ux, uy, uz :

$$\begin{aligned} x &= x + su_x \\ y &= y + su_y \\ z &= z + su_z \end{aligned} \quad (10)$$

where u_x , u_y , u_z are the projections of the photon trajectory onto the x, y, and z axes.

Crossed a boundary?

The voxel in which the photon resides is specified by the ix, iy, iz pointers to the voxel, which are obtained by taking the integer value of each x,y,z position. For example in MATLAB notation, the function $ix = \text{floor}(x/dx)$ where dx is the voxel size converts $x_1 < x < x_2$ to ix_1 such that $x(ix_1) = x_1$. In other words it does not “round up” to ix_2 when ix is closer to ix_2 . In C code, $ix = (\text{int})((x - 0.5)/dx)$ accomplishes the same function. Similarly, iy and iz are determined. If the voxel pointers before and after taking step s are the same, the photon is still in the same voxel. But if they differ, then the photon is crossing a voxel boundary.

If the photon is attempting to cross a boundary, then the step s must be divided into a step s_1 to the voxel boundary, drop some photon weight in the current voxel (see Drop below), then take the remainder of the step, s_2 , in the next voxel, governed by the optical properties of that voxel. The photon step is split based on its dimensionless value $\mu_{s1}s$ where subscript 1 indicates the current voxel. The step along x to the voxel boundary at x_2 is $s_1 = ix_2 dx - x$ with a dimensionless value $\mu_{s1}s_1$, where μ_{s1} is the scattering within the current voxel. Similarly, the steps along y and z are calculated. The minimum s_1 along the x,y,z directions becomes the step size to the boundary. The photon hops to the boundary with step s_1 , and drops some weight in the current voxel (see Drop section). The position of the photon is advanced by a negligible amount (eg., 10^{-9} mm) to cross the boundary. Then it continues with its remaining weight in the new voxel with a step s_2 based on the remaining dimensionless step, $\mu_{s1}(s - s_1)$, as in the following equations,

$$\begin{aligned} s_1 &= ix dx - x \\ s_2 &= \frac{\mu_{s1}(s - s_1)}{\mu_{s2}} \end{aligned} \quad (11)$$

Transmit or reflect?

Not explicitly shown in Fig. 6 is a decision on whether to reflect off the boundary being crossed or to transmit across the boundary. This decision occurs in the `cross boundary` step of Fig. 6. If the refractive indices of the current and next voxel, n_1 and n_2 , are different, then a subroutine is called, which is based on Fresnel reflectance, $[r \ ca_2] = RFresnel(n_1, n_2, ca_1, ca_2)$ where r is the Fresnel reflectance of the boundary, n_1 and n_2 are the refractive indices of the current and new voxel, ca_1 is the cosine of the angle of incidence onto the boundary, and ca_2 is the cosine of the angle of transmittance. The subroutine returns r and ca_2 . A random number RND is then generated, and if $RND \leq r$ then the photon reflects off the boundary. Otherwise, it transmits into the new voxel at the angle ca_2 . This handling of the boundary works well when regions of different refractive indices are aligned with the xy, xz, or yz planes. However, a tissue structure with curved regions of different refractive indices requires guidance by the curved surface to refract correctly. An example of such guidance is demonstrated in the MC program reported by Tran and Jacques [11].

Escaping?

When crossing a boundary, a transmitted photon may be crossing a surface boundary (eg., tissue/air) and escaping the tissue. In this case, after the partial step to the boundary and the calculation of the escaping trajectory angle, the photon weight can be assigned to increment an element of an array of escaping flux $J(x,y,z)$ at some x,y,z position, at some angles of escape $J(\alpha, \beta)$, or a combined $J(x,y,z, \alpha, \beta)$. If the photon is not escaping, it returns to the step “Cross boundary?” and continues propagating.

Drop

The DROP step deposits some of the photon weight (W) into an array element associated with the current voxel, $A(x, y, z)$, then decreases the surviving weight of the photon,

$$\begin{aligned} T &= e^{-s_1 \mu_a(x, y, z)} \\ A(x, y, z) &\leftarrow A(x, y, z) + W(1 - T) \\ W &\leftarrow WT \end{aligned} \quad (12)$$

where s_1 is the step length taken within the current voxel, $\mu_a(x, y, z)$ is the absorption coefficient of the current voxel, and W is the current weight of the photon. Each photon contributes to the accumulation of the spatial distribution of deposited photon weight, $A(x, y, z)$ [dimensionless], which records the fraction of the total energy of all delivered photons that was deposited in each x, y, z voxel. This $A(x, y, z)$ is later converted to relative fluence rate (see OUTPUT module).

Spin

The SPIN step changes the trajectory of the photon based on a chosen scattering function, i.e., it updates the parameters ux, uy, uz . Figure 7 shows the calculations within the SPIN module. Each use of RND is a freshly generated random number.

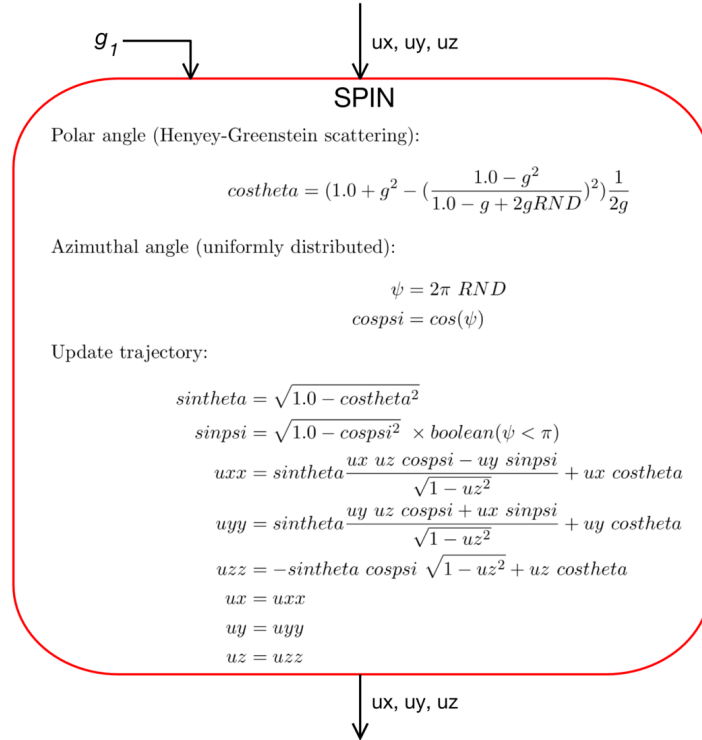


Fig. 7. The SPIN module changes the trajectory of the photon, specified by (ux, uy, uz) .

This example uses the Henye-Green scattering function, controlled by the anisotropy g_1 for the tissue in the current voxel. First, the polar deflection angle, θ , is sampled, yielding the parameter costheta . Second, the azimuthal angle of scatter, ψ , is sampled, yielding cospsi , as if it were randomly distributed between 0 and 2π , since the scattering structures are randomly oriented. This is the usual assumption but in some cases, for example striated muscle, there is an orientation to the scattering structures and the azimuthal angle may have to be

sampled appropriately. Third, the $\cos\theta$ and $\cos\psi$ along with the current trajectory (u_x, u_y, u_z) are used to create a new trajectory (u_{xx}, u_{yy}, u_{zz}) , which is then assigned to u_x, u_y, u_z . Hence, the photon trajectory has been adjusted by a single scattering event.

Check

A final step in this iterative hop, drop, spin, check process is the Check module to properly terminate a photon. Since the photon weight W continually decreases as the photon propagates, at some point the W is so low that continuing to propagate the photon adds little to the final solution. However, one cannot simply terminate the photon because the residual W will be lost, violating conservation of energy and resulting in an erroneous result. So the photon is statistically terminated using the so-called “Roulette Method”. When W drops below a threshold value, a random number (RND) is generated, and if RND is less than a value chance ($0 < \text{chance} < 1$), then W is increased by $\frac{1}{\text{chance}}$. Otherwise, the photon is terminated.

```
chance = 0.10
threshold = 0.001
if W <= threshold
    if RND < chance, W = W/chance
    else W=0 (kill photon)
else do_nothing (photon continues)
```

The choice of $\text{threshold} = 0.001$ means that 99.9% of a photon’s weight has been deposited before considering to extend the photon’s life by increasing its weight. The choice of $\text{chance} = 0.1$ means that the photon terminates 90% of the time and 10% of the time the photon’s W is increased 10-fold to allow the photon to continue propagating. Statistically this procedure conserves energy ($\text{Escape} + \text{Absorption} = 1$). The conservation of energy is the key reason for the roulette method. When a photon’s lifetime is extended, it may escape as reflectance or transmission, or it may be absorbed. The expanded spatial extent of photon movement afforded by this lifetime extension of 0.01% of the photons balances the loss of photon movement for 99.99% of the photons.

Once a photon is terminated, another photon with a fresh $W = 1.0$ is launched. Photons are launched until the requested number of photons is reached, which terminates the launching of photons. The simulation is now complete, and the program then proceeds to the OUTPUT module to generate a report of results.

3.5. Output

After N_{photons} photons have been launched, a final step of scaling is needed to generate the output for the user.

3.5.1. Absorbed photon fraction, $A(x,y,z)$

The absorbed fraction $A(x,y,z)$ holds the total weight deposited by all the photons in the voxel at x,y,z . This weight is then normalized by the number of photons launched and by the volume of the voxel to yield a density of photon deposition, $\frac{A [\text{counts}]}{N_{\text{photons}} V_{\text{voxel}}} \rightarrow A [\text{cm}^{-3}]$. For an input of 1 J, A indicates J/cm^3 . For an input of 1 W, A indicates W/cm^3 .

3.5.2. Photon energy concentration or fluence, $\phi(x,y,z)$

The Monte Carlo simulation has recorded and scaled $A(x,y,z)$ as $[\text{cm}^{-3}]$, which is related to the fluence $\phi(x,y,z) [\text{cm}^{-2}]$ at that voxel by the absorption coefficient $[\text{cm}^{-1}]$ of that voxel,

$$\phi(x,y,z) = \frac{A(x,y,z)}{\mu_a(x,y,z)} \quad (13)$$

If the incidence is 1 J of energy, as in a pulsed laser, the fluence ϕ has units of $[J/cm^2]$, which might be called the radiant exposure experienced by the voxel. If the incidence is 1 W of power, the fluence rate ϕ is expressed as $[W/cm^2]$, which might be called the irradiance experienced by the voxel. The fluence rate ϕ in $[W/cm^2]$ is proportional to the instantaneous energy density in the voxel, $C [J/cm^3]$: $\phi = cC$, where $c [cm/s]$ is the speed of light in the medium ($c = \frac{c_0}{n}$. c_0 is speed *in vacuo*, n is refractive index).

3.5.3. Escaping flux, $J(x,y,z, \alpha, \beta)$

The light that escapes the tissue was recorded as photon weight W escaping at some x,y,z position and angles α, β , $J(x,y,z, \alpha, \beta)$ [dimensionless weight]. To illustrate, consider the total escape at all angles from one voxel at the front surface of a tissue slab where light was delivered, $J(x,y)$ at $z=0$. This escaping weight is then normalized by the number of photons launched and the surface area ($dxdy$) of the voxel from which the photons escaped: $J(x,y) \leftarrow \frac{J(x,y)}{N_{photons} dxdy}$ with units

of $[cm^{-2}]$, which is a flux density. Such escape from the irradiated surface may be called local reflectance, $R(x,y)$, and the total diffuse reflectance is $R_d = \sum_{x,y} R(x,y) dxdy$. Escape from the rear surface of a tissue slab may be called local transmittance and the total diffuse transmission similarly calculated. For tissue geometries other than slabs, total reflectance and transmittance are not always the best descriptors, and the local flux density of escape may be more appropriate.

It is also possible to record the escaping photon weight as a function of the polar (α) and azimuthal (β) angles of escape into the medium outside the tissue. Usually, the azimuthal variation of escape is uniform since scatters are randomly oriented, so only the polar variation is recorded. For example, record $J(\alpha)$ where α is divided into n increments from 0 to $\pi/2$, i.e., reflectance orthogonal to tissue surface (0°) to the extreme of broad scatter (90°). Then, the cumulative sum of $J(\alpha)$ from 0 up to some $\alpha_{collection}$ yields the escape that would be collected by the numerical aperture of a detector centered on the z -axis at $x,y=0,0$ ($NA = \sin(\alpha_{collection})$). Interpolation of this cumulative sum can yield the collection of any particular detector (eg., optical fiber, camera).

In summary, the MC simulation can yield a deposition, a fluence, or an escaping flux expressed as a fraction of the delivered power [W] or energy [J]. Considering effects of light on the tissue, the deposited energy, J/cm^3 per J delivered = cm^{-2} , causes heating or photochemistry. Considering diagnostics, the escaping flux, W per W delivered = reflectance (not including specular reflectance), or escaping flux density, W/cm^2 per W delivered = local reflectance, provide information about the tissue. Deposited energy within the tissue will affect the escaping flux, allowing the diagnostic detection of local absorbing structures like a blood vessel or tumor.

3.6. Nuances in MC code

There are nuances when setting up a MC simulation. For example, the lateral boundaries at $\pm x$ and $\pm y$ may allow photons to escape the simulation volume and never return, which drops the fluence near these boundaries. Setting the boundaries to act like a mirror can prevent such loss, but now there are virtual images of the source and tissue structure appearing outside the volume and too much light may return from the boundary. Either a compromise between these two boundary conditions is needed, or photons are allowed to propagate outside the simulation volume and later return into the volume. Another nuance is to be sure that the boundaries between different tissue types are aligned with the bins of the simulation. For example, if the boundary between two tissue types occurs within a voxel, rather than at the front or rear surface of the voxel, then photons in that voxel will sometimes act as if in the first tissue type and sometimes as if in the second tissue type, resulting in odd results at that voxel. The possible nuances depend on the particular implementation of a MC simulation.

3.7. Polarized light and coherence

The character of a photon can be characterized by the orientation and phase of its oscillating electric field. A population of photons can be described by a polarization state, the Stokes vector = [I Q U V]. Ramella-Roman et al. [12] describe a MC simulation of polarized light propagation, which was recently implemented in a GPU-accelerated simulation [13].

The propagation of coherence by a MC simulation tracks the orientation and phase of a photon as it propagates. At a detector or region of interest, multiple photons add via vector addition to yield the net intensity, orientation, and phase. Fischer et al. [14] describe a MC simulation that predicts the spatial coherence of light propagation. Shen et al. [15] describe propagation of coherence in random medium. Sawicki et al. [16] describe a MC simulation of coherent backscatter from a tissue.

4. Example uses

4.1. How does skin architecture affect skin color?

The light reflected from the skin over the 380-720 nm range of wavelengths yields the perceived color of the skin. A set of Monte Carlo simulations can specify a diffuse reflectance spectrum $R_d(\lambda)$ over this range of wavelengths (λ), which can be converted into a triplet of red, green, and blue values (R,G,B) which combine to yield a perceived color.

The CIE standard observer functions, $\bar{x}$, $\bar{y}$, $\bar{z}$, multiply the reflectance spectrum, the illumination spectrum ($I_{illum}(\lambda)$), and the daylight spectrum ($I_{d65}(\lambda)$), followed by integration to yield an X,Y,Z triplet of values.

$$\begin{aligned} X &= \int_{380nm}^{720nm} \bar{x} R_d I_{illum} I_{d65} d\lambda \\ Y &= \int_{380nm}^{720nm} \bar{y} R_d I_{illum} I_{d65} d\lambda \\ Z &= \int_{380nm}^{720nm} \bar{z} R_d I_{illum} I_{d65} d\lambda \end{aligned} \quad (14)$$

The next step uses a transformation matrix M to transform the triplet X,Y,Z into the triplet R,G,B (an XYZ $\rightarrow$ sRGB transformation). This M is based on the daylight illumination (I_{d65} , ~6500 color temperature).

$$[R \ G \ B]^T = M \times [X \ Y \ Z]^T \quad (15)$$

where superscript T indicates the transpose to a vertical vector, and M is the transformation matrix,

$$M = \begin{bmatrix} 3.2404542 & -1.5371385 & -0.4985314 \\ -0.9692660 & 1.8760108 & 0.0415560 \\ 0.0556434 & -0.2040259 & 1.0572252 \end{bmatrix} \quad (16)$$

The resulting R,G,B triplet combines to yield the skin color. The wavelength dependence of the I_{d65} and the M matrix cancel each other. If $I_{illum} = 1.0$ and $R = 1.0$ for all wavelengths, the R,G,B triplet yields a white color on a computer screen. If a particular $I_{illum}(\lambda)$ is used in Eq. (14), then the displayed color will properly portray the perceived color under that particular illumination.

Consider a four-layer skin architecture: (1) a pigmented epidermis with low melanin to allow better display of the skin color, (2) a superficial papillary dermal layer with low blood content (just thin capillaries), (3) a 50- μ m-thick vascular plexus with 5% blood content serving as the reservoir for supplying the capillaries, and (4) a deeper reticular dermis with a few larger blood vessels feeding/draining the vascular plexus. Figure 8 shows the skin model and the resulting

colors. When the plexus is close to the skin surface, the skin color is slightly orange. As the plexus is moved further from the surface, the skin turns slightly pink. Why?

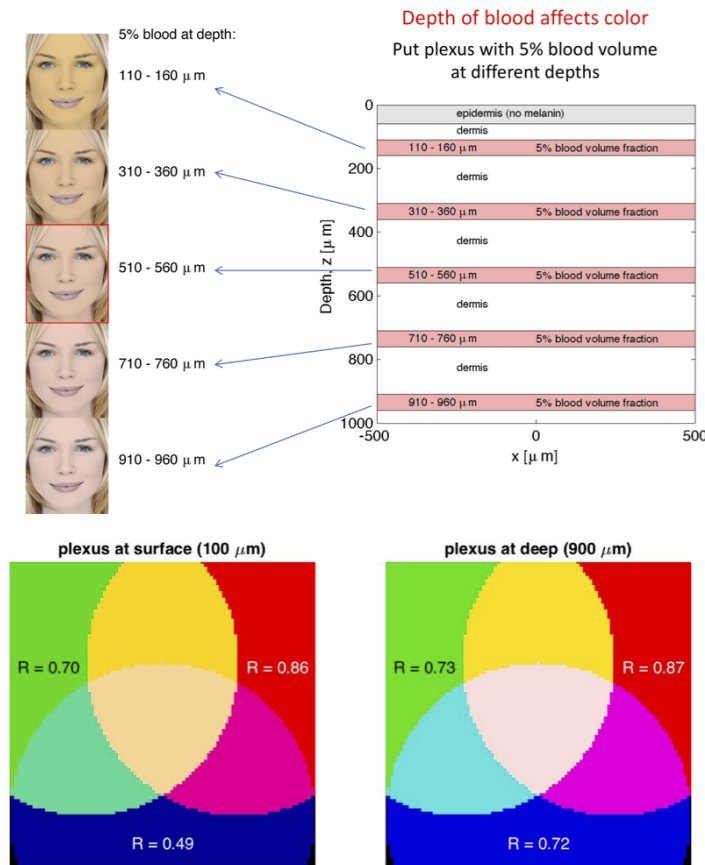


Fig. 8. As a 50- μm -thick vascular plexus with 5% blood content is moved further from the skin surface, the skin color changes from slightly orange to slightly pink. Above the plexus, the dermis is relatively blood free with only thin capillaries circulating red blood cells. Blue light is partially reflected from this superficial layer before it can reach the blood in the plexus and be absorbed, hence blue reflectance increases. This simple model only approximates true skin structure to illustrate the effect.

In the homogeneous skin model, the red, green, and blue light pass through a uniformly blood-perfused skin. The absorption of light by blood scales as blue > green > red, yielding an R,G,B triplet of [0.86 0.70 0.49], which is slightly orange. In the layered skin model, the red light and green light penetrate into the reticular dermis, then diffuse back out to escape at the surface, hence red and green light pass through the vascular plexus twice. Blue light, however, has difficulty penetrating into the plexus because it is strongly scattered by the relatively blood-free superficial layer. Consequently, blue light partially avoids absorption by blood and more blue is reflected. The R,G,B triplet is [0.87 0.73 0.72], which is slightly pink.

The same effect of architecture can be seen in pink gums, blue eyes, and blue monkey butts. The latter two examples feature a superficial collagenous layer overlaying a layer of melanin pigment. Blue light scatters and red and green penetrate to the melanin and are absorbed. In summary, tissue architecture affects the perceived color of a tissue.

4.2. Exposure time for photobiomodulation of the knee?

Photobiomodulation therapy (PBMT) is a phototherapy often using deep red light to elicit a biological effect on a tissue such as decreasing inflammation after tissue injury. The mechanism of PBMT is still a subject of investigation and several possible mechanisms are considered [17]. One mechanism involves interaction of treatment light with the cytochrome C pigment in the inner walls of the mitochondria, thereby affecting electron transport.

A typical dose of light for PBMT is $\sim 1 \text{ J/cm}^2$. Figure 9 shows the irradiation of a knee with 0.707-W collimated beam delivered uniformly over a 3-cm-dia. spot, which equals 0.100 W/cm^2 irradiance. This irradiance is commonly used in phototherapy to avoid overheating by allowing blood flow to cool the tissue. The figure indicates that fluence rate in the soft tissue between the patella and bone is about 0.01 W/cm^2 . Delivery of the beam for two minutes yields $(0.01 \text{ W/cm}^2)(60 \text{ s/min})(2 \text{ min}) = 1.2 \text{ J/cm}^2$ within the soft tissue. Therefore, a two-minute treatment may be approximately sufficient for treatment.

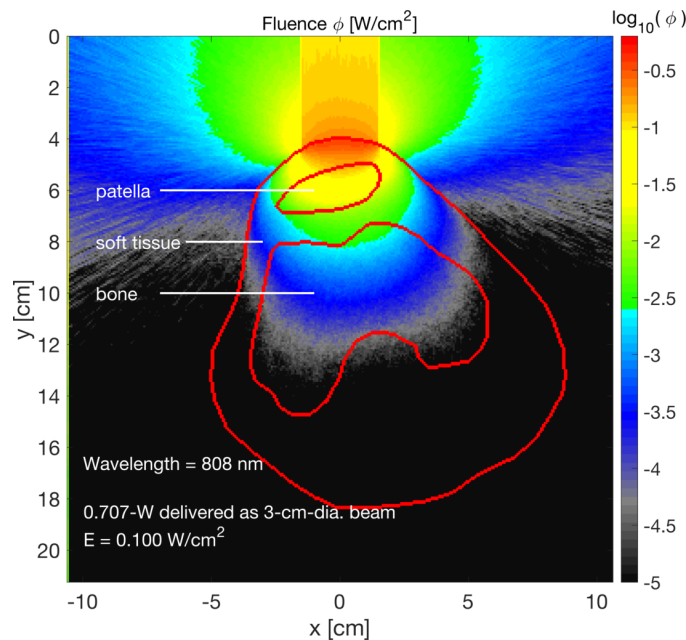


Fig. 9. Photobiomodulation therapy (PBMT) of the knee. The delivery of 0.707 W of treatment light at 808-nm wavelength over a 3-cm-dia. spot yields an irradiance $E = 0.100 \text{ W/cm}^2$, which penetrates into the knee to yield about 0.01 W/cm^2 fluence in the soft tissue between the patella and bone.

4.3. Will a pulsed laser treatment reach the bulb of a hair follicle?

Laser treatment sometimes employs thermal coagulation of tissue to achieve a therapeutic effect. A popular example is the use of a pulsed laser to cause thermal damage to the bulb of a hair follicle for the purpose of hair removal. Figure 10 illustrates the penetration of a 0.4-cm-dia. 1024-nm infrared pulsed laser down to the bulb. The focus does not reach the bulb due to light scattering by the tissue, however, diffusion allows light to reach the bulb. The energy deposition in the bulb reaches about 100 J/cm^3 per J delivered. Therefore, a 1.9-J 10-ms laser pulse (15 J/cm^2 exposure at skin surface) increases the 30°C skin temperature to $(1.9 \text{ J})(100 \text{ J/cm}^3/\text{J})/(4^\circ\text{C}\cdot\text{J/cm}^3) + 30^\circ\text{C} = 77^\circ\text{C}$, which is sufficient to thermally damage the bulb.

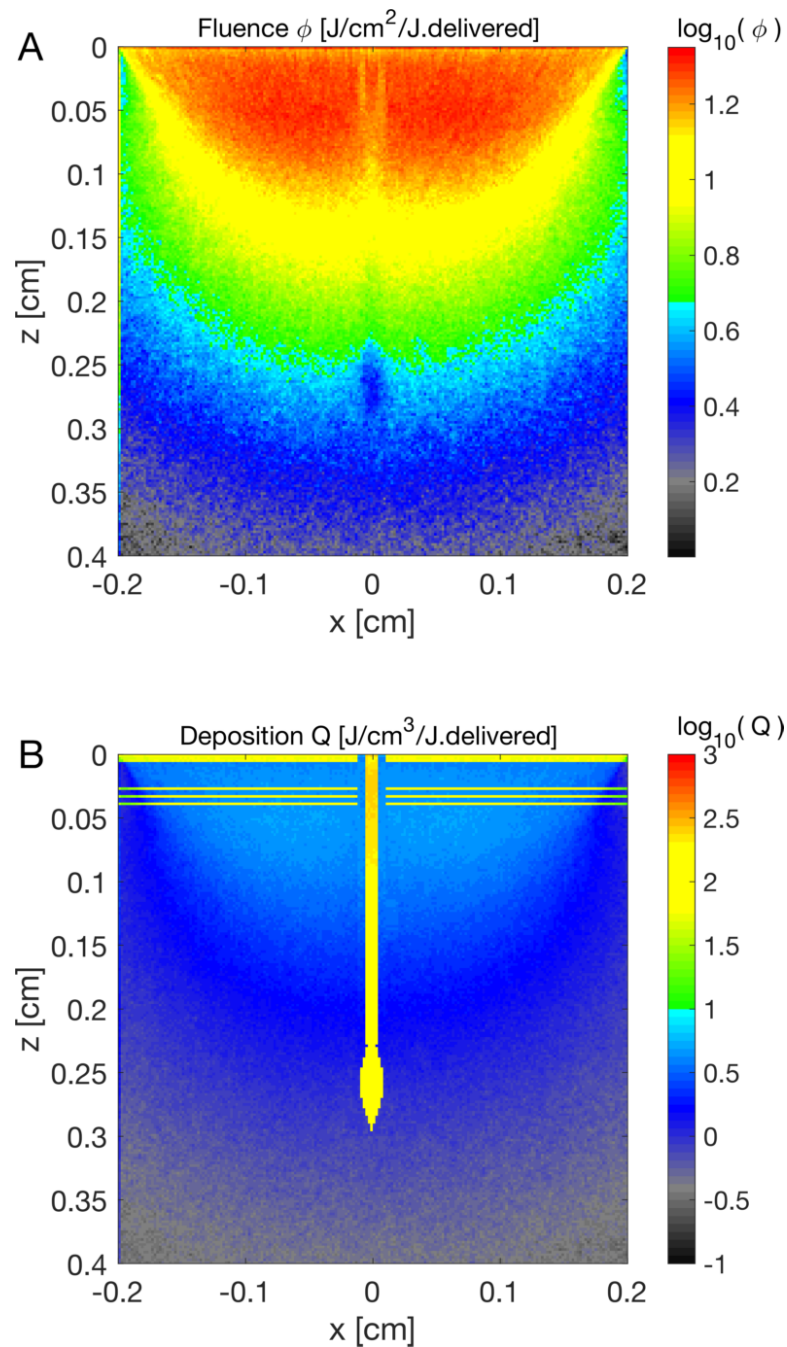


Fig. 10. (A) Penetration of a pulsed 1024-nm-wavelength 0.4-cm-dia. laser beam focused to the bulb of a hair follicle at a depth of 0.3 cm in the skin. The beam loses its focus within 1 cm and only diffuse light reaches the bulb. The depression of fluence along z at $x=0$ is due to absorption by the melanin pigment in the epithelium of the hair shaft. (B) The energy deposition in the bulb is sufficient to achieve thermal damage. The bulb has a 50% volume fraction of cutaneous melanosomes.

5. Conclusion

This tutorial has briefly discussed the typical design of a Monte Carlo (MC) simulation of light transport for the purposes of diagnostics and dosimetry during biomedical applications of light. There are now a number of MC simulations available on the internet for biomedical, oceanic, atmospheric, and astrophysical applications. The tutorials for using specific MC programs can instruct on how to use these programs.

Acknowledgments. The author's adventure with Monte Carlo simulations was shared with Scott Prah, Marleen Keijzer, Lihong Wang, Jessica Ramella-Roman, Ting Li, Yinchu Chen, and Anh Phong Tran, who all contributed to development of mcml.c and mcxyz.c.

Disclosures. The author declares no conflicts of interest.

Data availability. No data were generated or analyzed in the presented research.

References

1. N. Metropolis, "The beginning of the Monte Carlo method," *Los Alamos Science*, Special issue, 125–1330 (1987).
2. L. Wang, S. L. Jacques, and L. Zheng, "MCML - Monte Carlo modeling of light transport in multi-layered tissues," *Computer Methods and Programs in Biomedicine* **47**(2), 131–146 (1995).
3. D. Marti, R. N. Aasbjerg, P. E. Anersen, and A. K. Hansen, "MCmatlab: an open-source, user-friendly, MATLAB-integrated three-dimensional Monte Carlo light transport solver with heat diffusion and tissue damage," *J. Biomed. Opt.* **23**(12), 1 (2018).
4. Q. Fang and D. A. Boas, "Monte Carlo simulation of photon migration in 3D turbid media accelerated by graphics processing units," *Opt. Express* **17**(22), 20178 (2009).
5. "Virtual Photonics Technology Initiative," (2022) <https://virtualphotonics.org>.
6. "MCX, Monte Carlo eXtreme, a GPU-accelerated photon transport simulator," (2022), <http://mcx.space>.
7. "Cloud based Monte Carlo tool for photon transport," (2022), <http://www.lighttransport.net>.
8. J. Jönsson and E. Berrocal, "Multi-Scattering software: part I: online accelerated Monte Carlo simulation of light transport through scattering media," *Opt. Express* **28**(25), 37612–37638 (2020).
9. S. L. Jacques and N. J. McCormick, "Two-term scattering phase function for photon transport to model subdiffuse reflectance in superficial tissues," *Biomed. Opt. Express*, in press (2022).
10. M. Xu, "Plum pudding random medium model of biological tissue toward remote microscopy from spectroscopic light scattering," *Biomed. Opt. Express* **8**(6), 2879 (2017).
11. A. P. Tran and S. L. Jacques, "Modeling voxel-based Monte Carlo light transport with curved and oblique boundary surfaces," *J. Biomed. Opt.* **25**(02), 1 (2020).
12. J. C. Ramella-Roman, S. A. Prah, and S. L. Jacques, "Three Monte Carlo programs of polarized light transport into scattering media: part I," *Opt. Express* **13**(12), 4420–4438 (2005).
13. S. Yan, S. L. Jacques, J. C. Ramella-Roman, and Q. Fang, "Graphics-processing-unit-accelerated Monte Carlo simulation of polarized light in complex three-dimensional media," *J. Biomed. Opt.* **27**(08), 083015 (2022).
14. D. G. Fischer, S. A. Prah, and D. D. Duncan, "Monte Carlo modeling of spatial coherence: free-space diffraction," *J. Opt. Soc. Am. A* **25**(10), 2571–2581 (2008).
15. Z. Shen, S. Sukhov, and A. Dogariu, "Monte Carlo method to model optical coherence propagation in random media," *J. Opt. Soc. Am. A* **34**(12), 2189 (2017).
16. J. Sawicki, N. Kastor, and M. Xu, "Electric field Monte Carlo simulation of coherent backscattering of polarized light by a turbid medium containing Mie scatterers," *Opt. Express* **16**(8), 5728 (2008).
17. M. R. Hamblin, C. Farraresi, Y. Huang, L. Freitas de Freitas, and J. D. Carroll, *Low-level Light Therapy: Photo-biomodulation* (SPIE Press, 2018).