Code manual

Montag, 14. Juni 2021 15:15

LIBRARIES

```
import numpy as np
import multiprocessing as mp
from multiprocessing import Queue, cpu_count
import matplotlib.pyplot as plt
import time
import os
```

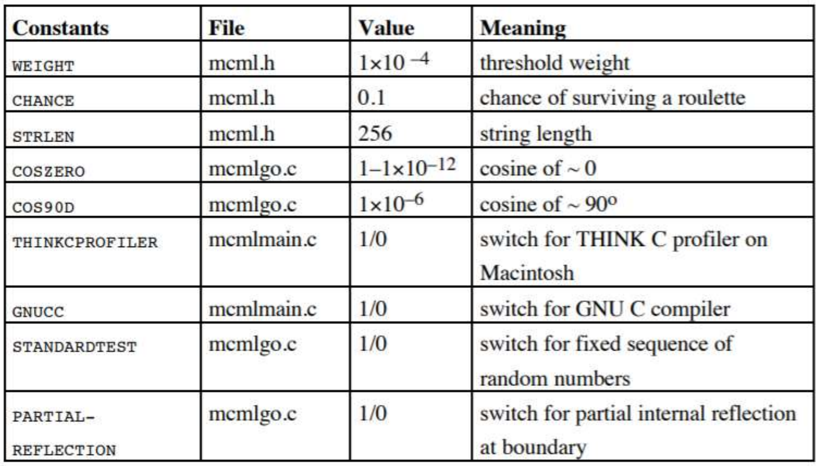
Simulation files :

Parameter	meaning	symbol	unit	Data from	Data to	To know
wavelength	all wavelength data for this program					
Model_input	some parameters of the simulated tissue model					
Mua_water	the optical absorption properties (μ<sub>a</sub> cm<sup>-1</sup>) of water	μ_a	Cm^{-1}	0.0002668	0.043376	
Mua_oxy	the optical absorption properties (μ<sub>a</sub> cm<sup>-1</sup>) of oxygenated whole blood	μ_a	Cm^{-1}			150 g hemoglobin/liter or $2.33x10^{-3}$ M From <https://omlc.org/software/mc/mcxyz/index.html>
Mua_melanin	the optical absorption properties (μ<sub>a</sub> cm<sup>-1</sup>) ofinterior of typical cutaneous melanosome</br>	μ_a	Cm^{-1}			

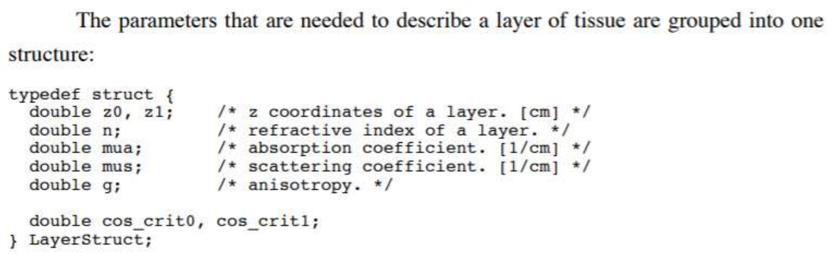
Initial parameters:

```
WEIGHT :
WEIGHT = 1e-4       # critical weight for roulette
CHANCE = 0.1                    # Chance of roulette survival
PARTIALREFLECTION = 0    # 1=split photon, 0=statistical reflection.
COSZERO = 1.0 - 1.0e-12     # cosine of about 1e-6 rad
COS90D = 1.0e-6     # cosine of about 1.57 - 1e-6 rad
path = os.path.dirname(__file__)
```

Table 5.1. Important constants in the program mcml.



Constants	File	Value	Meaning
WEIGHT	mcml.h	1×10^{-4}	threshold weight
CHANCE	mcml.h	0.1	chance of surviving a roulette
STRLEN	mcml.h	256	string length
COSZERO	mcmlgo.c	$1-1\times10^{-12}$	cosine of ~ 0
COS90D	mcmlgo.c	1×10^{-6}	cosine of ~ 90°
THINKCPROFILER	mcmlmain.c	1/0	switch for THINK C profiler on Macintosh
GNUCC	mcmlmain.c	1/0	switch for GNU C compiler
STANDARDTEST	mcmlgo.c	1/0	switch for fixed sequence of random numbers
PARTIAL-REFLECTION	mcmlgo.c	1/0	switch for partial internal reflection at boundary



The parameters that are needed to describe a layer of tissue are grouped into one structure:

```
typedef struct {
  double z0, z1;     /* z coordinates of a layer. [cm] */
  double n;          /* refractive index of a layer. */
  double mua;        /* absorption coefficient. [1/cm] */
  double mus;        /* scattering coefficient. [1/cm] */
  double g;          /* anisotropy. */

  double cos_crit0, cos_crit1;
} LayerStruct;
```

For both air and water, a moderate scattering coefficient is specified so that the photon steps through the tissue, but the value of g is set to 1.0 so there is not photon deflection. Also a very very low absorption coefficient is specified (eg., μ_a = 0.0001 cm^{-1} for air, or the absorption of water at the chosen wavelength). Hence, the photon will step through air or water, and deposit a very small amount of photon weight in the voxels, thereby specifying its pathlength spent in the air or water. But the energy lost in the air is negligible, so it does not significantly influence the distribution of light. The absorption in water depends on the chosen wavelength. At the end of the program (discussed in **How to use mcxy.c**, below), the fluence rate ϕ is calculated by dividing the deposited energy by the very small μ_a which recovers the ϕ in the air and water.

MEDIUM CLASSE

Medium :

n - refractive index
mua - absorption coefficient. [1/cm]
mus - scattering coefficient. [1/cm]
g - anisotropy

MEDIUMS and default parameters

All made from :

- oxygenated whole blood (150 g hemoglobin/liter or 2.33x10^{-3} M)
- deoxygenated whole blood
- water
- interior of typical cutaneous melanosome
- fat

Medium	**Biological name**	**Refractive index n**	**Attenuation coeffieitn mua**	**Scattering coefficient mus**	**g**	**Cb**	**Cwater**	**Cmel**	**thickness**
Air	Air	1	0		1				
Tissue 1	epidermis	1.5	((0.1-(0.3e-4*wavelength))+1.25 *Rayleigh)*(1-Cwater)+Cwater*water	1000	0.86				20e-4
Tissue 2	dermis	1.34	(Cmel*melanin+(1-Cmel)*Rayleigh)*(1-Cwater)+Cwater*water	450.0	0.8				80e-4
Tissue 3		1.4	((1-S)*gamma*Cb*deoxy)+(S*gamma*Cb*oxy)+((1-gamma*Cb)*Cwater*water)+((1-gamma*Cb)*(1-Cwater)*Rayleigh)	300	0.9	0.04	0.5		150e-4
Tissue 4		1.39	((1-S)*gamma*Cb*deoxy)+(S*gamma*Cb*oxy)+((1-gamma*Cb)*Cwater*water)+((1-gamma*Cb)*(1-Cwater)*Rayleigh)	350	0.95				80e-4
Tissue 5		1.4	((1-S)*gamma*Cb*deoxy)+(S*gamma*Cb*oxy)+((1-gamma*Cb)*Cwater*water)+((1-gamma*Cb)*(1-Cwater)*Rayleigh)	250	0.8				1500e-4

Tissue 6		1.38	((1-S)*gamma*Cb*deoxy)+(S*gamma*Cb*oxy)+((1-gamma*Cb)*Cwater*water)+((1-gamma*Cb)*(1-Cwater)*Rayleigh)	300	0.95				80e-4
Tissue 7		1.44	((1-S)*gamma*Cb*deoxy)+(S*gamma*Cb*oxy)+((1-gamma*Cb)*Cwater*water)+((1-gamma*Cb)*(1-Cwater)*Rayleigh)	50	0.75				6000e-4

- B = blood volume fraction
- S = oxygen saturation of hemoglobin
- W = water volume fraction
- M = melanosome volume fraction
- F = fat volume fraction
- $\mu_{s.500nm}'$ = reduced scattering coeff. at 500 nm [cm^{-1}]
- f = fraction of scattering at 500 nm due to Rayleigh scattering
- 1-f = fraction of scattering at 500 nm due to Mie scattering
- b_{mie} = scattering power for Mie scattering

$$\mu_s' = \mu_{s.500nm}' \left(f_{Rayleigh}(\lambda/500nm)^{-4} + f_{Mie}(\lambda/500nm)^{-bMie}\right)$$

- $\mu_{s.500nm}'$ = reduced scattering coeff. at 500 nm
- $f_{Rayleigh}$ = fraction of Rayleigh scattering at 500 nm
- f_{Mie} = fraction of Mie scattering at 500 nm
- λ = wavelength [nm]
- b_{Mie} = scattering power of Mie scattering

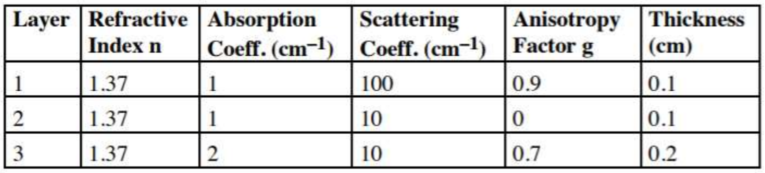
The melanin content is described by the volume fraction (M_f) of a standard cutaneous melanosome in the epidermis. The absorption coefficient of the interior of a melanosome was estimated by the threshold pulsed laser radiant exposure required to explode a cutaneous melanosome:

From <https://omlc.org/news/feb15/generic_optics/index.html>

$$\mu_{a.melanosome} = (679\ cm^{-1})\left(\frac{\lambda}{500nm}\right)^{-3.33}$$

stratum corneum boundary
With a starting photon weight W of 1, every time a photon packet interacts with a bin it loses part of its weight and then gets
scattered in a direction determined by the anisotropy factor and scattering coefficient. At the end of the simulation, all diffusely scattered photons locating at the incident side (z<0) were added up to give a diffuse reflectance spectrum. </br>

Table 6.6. The optical properties of the three-layer tissue.



Layer	Refractive Index n	Absorption Coeff. (cm^{-1})	Scattering Coeff. (cm^{-1})	Anisotropy Factor g	Thickness (cm)
1	1.37	1	100	0.9	0.1
2	1.37	1	10	0	0.1
3	1.37	2	10	0.7	0.2

MODEL SIMULATION CLASS

Wth - play roulette if photon weight < Wth
dz - z grid separation [cm]
dr - r grid separation [cm]
da - alpha grid separation [radian]
nz - array range 0..nz-1
nr - array range 0..nr-1
na - array range 0..na-1
layerObj - medium layer structure class instance

FUNCTIONS TO EXPLAIN

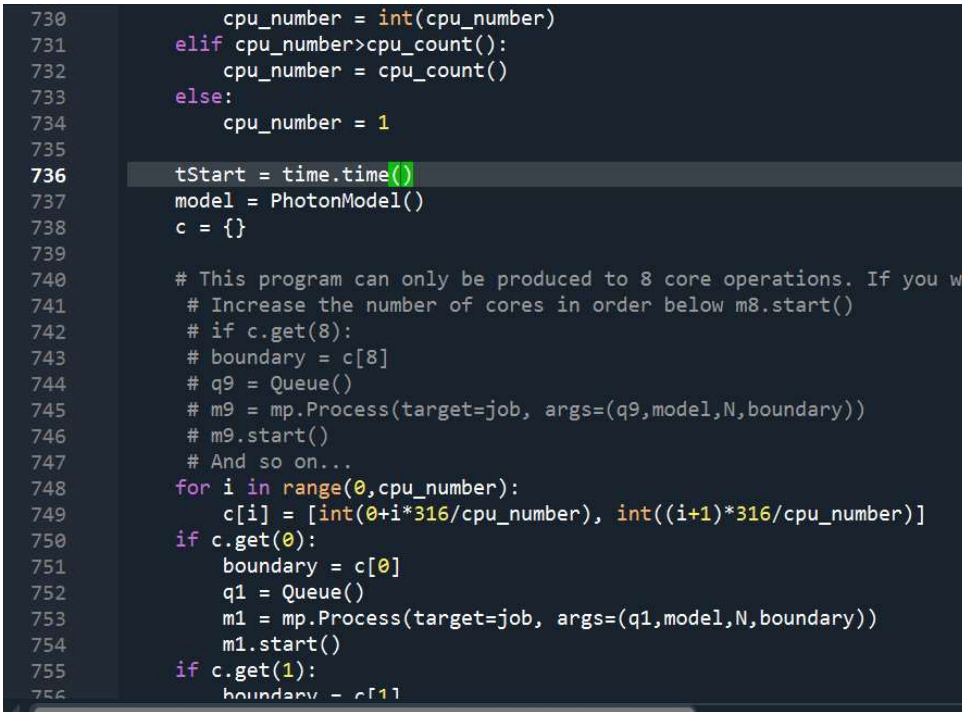
np.genfromtxt :

np.power

MIGHT BE NEEDED MODIFICATION

Line 740:

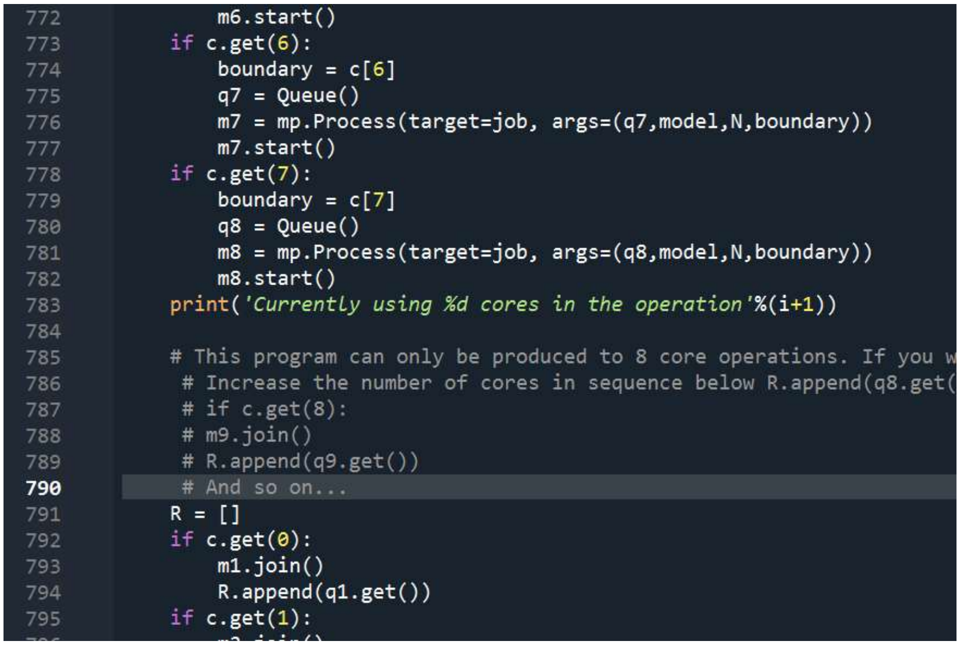
```
# This program can only be produced to 8 core operations. If you want to expand more core operations, please add the following code such as:
    # Increase the number of cores in order below m8.start()
    # if c.get(8):
    # boundary = c[8]
    # q9 = Queue()
    # m9 = mp.Process(target=job, args=(q9,model,N,boundary))
    # m9.start()
    # And so on...
```



```
730             cpu_number = int(cpu_number)
731         elif cpu_number>cpu_count():
732             cpu_number = cpu_count()
733         else:
734             cpu_number = 1
735 
736         tStart = time.time()
737         model = PhotonModel()
738         c = {}
739 
740         # This program can only be produced to 8 core operations. If you w
741          # Increase the number of cores in order below m8.start()
742          # if c.get(8):
743          # boundary = c[8]
744          # q9 = Queue()
745          # m9 = mp.Process(target=job, args=(q9,model,N,boundary))
746          # m9.start()
747          # And so on...
748         for i in range(0,cpu_number):
749             c[i] = [int(0+i*316/cpu_number), int((i+1)*316/cpu_number)]
750         if c.get(0):
751             boundary = c[0]
752             q1 = Queue()
753             m1 = mp.Process(target=job, args=(q1,model,N,boundary))
754             m1.start()
755         if c.get(1):
756             boundary = c[1]
```

Line 785:

```
 # This program can only be produced to 8 core operations. If you want to expand more core operations, please add the following code such as:
    # Increase the number of cores in sequence below R.append(q8.get())
    # if c.get(8):
    # m9.join()
    # R.append(q9.get())
    # And so on...
```



```
        m6.start()
    if c.get(6):
        boundary = c[6]
        q7 = Queue()
        m7 = mp.Process(target=job, args=(q7,model,N,boundary))
        m7.start()
    if c.get(7):
        boundary = c[7]
        q8 = Queue()
        m8 = mp.Process(target=job, args=(q8,model,N,boundary))
        m8.start()
    print('Currently using %d cores in the operation'%(i+1))

    # This program can only be produced to 8 core operations. If you w
     # Increase the number of cores in sequence below R.append(q8.get(
     # if c.get(8):
     # m9.join()
     # R.append(q9.get())
     # And so on...
    R = []
    if c.get(0):
        m1.join()
        R.append(q1.get())
    if c.get(1):
```

```
function tissue = makeTissueList(nm)
%function tissueProps = makeTissueList(nm)
%   Returns the tissue optical properties at the wavelength nm:
%       tissueProps = [mua; mus; g]';
%       global tissuenames(i).s
%   Uses
%       SpectralLIB.mat

%% Load spectral library
load spectralLIB.mat
%   muadeoxy      701x1            5608  double
%   muamel        701x1            5608  double
%   muaoxy        701x1            5608  double
%   muawater      701x1            5608  double
%   musp          701x1            5608  double
%   nmLIB         701x1            5608  double
MU(:,1) = interp1(nmLIB,muaoxy,nm);
MU(:,2) = interp1(nmLIB,muadeoxy,nm);
MU(:,3) = interp1(nmLIB,muawater,nm);
MU(:,4) = interp1(nmLIB,muamel,nm);
LOADED = 1;

%% Create tissueList

j=1;
tissue(j).name  = 'air';
tissue(j).mua   = 0.0001; % Negligible absorption yet still tracks,
tissue(j).mus   = 1.0;    % but take steps in air
tissue(j).g     = 1.0;    % and don't scatter.

j=2;
tissue(j).name  = 'water';
tissue(j).mua   = MU(3);
tissue(j).mus   = 10;   % Take steps in water,
tissue(j).g     = 1.0;  % but don't scatter.

j=3;
tissue(j).name  = 'blood';
B       = 1.00;
S       = 0.75;
W       = 0.95;
M       = 0;
musp500 = 10;
```

```
fray    = 0.0;
bmie    = 1.0;
gg      = 0.90;
musp = musp500*(fray*(nm/500).^-4 + (1-fray)*(nm/500).^-bmie);
X = [B*S B*(1-S) W M]';
tissue(j).mua = MU*X;
tissue(j).mus = musp/(1-gg);
tissue(j).g   = gg;

j = 4;
tissue(j).name = 'dermis';
B = 0.002;
S = 0.67;
W = 0.65;
M = 0;
musp500 = 42.4;
fray    = 0.62;
bmie    = 1.0;
gg      = 0.90;
musp = musp500*(fray*(nm/500).^-4 + (1-fray)*(nm/500).^-bmie);
X = [B*S B*(1-S) W M]';
tissue(j).mua = MU*X;
tissue(j).mus = musp/(1-gg);
tissue(j).g   = gg;

j=5;
tissue(j).name  = 'epidermis';
B = 0;
S = 0.75;
W = 0.75;
M = 0.03;
musp500 = 40;
fray    = 0.0;
bmie    = 1.0;
gg      = 0.90;
musp = musp500*(fray*(nm/500).^-4 + (1-fray)*(nm/500).^-bmie);
X = [B*S B*(1-S) W M]';
tissue(j).mua = MU*X;
tissue(j).mus = musp/(1-gg);
tissue(j).g   = gg;

j=6;
tissue(j).name  = 'skull';
B = 0.0005;
S = 0.75;
W = 0.35;
M = 0;
musp500 = 30;
fray    = 0.0;
bmie    = 1.0;
gg      = 0.90;
musp = musp500*(fray*(nm/500).^-4 + (1-fray)*(nm/500).^-bmie);
X = [B*S B*(1-S) W M]';
tissue(j).mua = MU*X;
tissue(j).mus = musp/(1-gg);
tissue(j).g   = gg;

j=7;
tissue(j).name = 'gray matter';
B = 0.01;
S = 0.75;
W = 0.75;
M = 0;
musp500 = 20;
fray    = 0.2;
```

```
bmie    = 1.0;
gg      = 0.90;
musp = musp500*(fray*(nm/500).^-4 + (1-fray)*(nm/500).^-bmie);
X = [B*S B*(1-S) W M]';
tissue(j).mua = MU*X;
tissue(j).mus = musp/(1-gg);
tissue(j).g   = gg;

j=8;
tissue(j).name  = 'white matter';
B = 0.01;
S = 0.75;
W = 0.75;
M = 0;
musp500 = 20;
fray    = 0.2;
bmie    = 1.0;
gg      = 0.90;
musp = musp500*(fray*(nm/500).^-4 + (1-fray)*(nm/500).^-bmie);
X = [B*S B*(1-S) W M]';
tissue(j).mua = MU*X;
tissue(j).mus = musp/(1-gg);
tissue(j).g   = gg;

j=9;
tissue(j).name  = 'standard tissue';
tissue(j).mua   = 1;
tissue(j).mus   = 100;
tissue(j).g     = 0.90;

disp(sprintf('---- tissueList ------ \tmua   \tmus  \tg  \tmusp'))
for i=1:length(tissue)
    disp(sprintf('%d\t%15s\t%0.4f\t%0.1f\t%0.3f\t%0.1f',...
        i,tissue(i).name, tissue(i).mua,tissue(i).mus,tissue(i).g,...
        tissue(i).mus*(1-tissue(i).g)))
end
disp(' ')
```