TD Monte Carlo with MCX

Arthur Gautheron - arthur.gautheron@univ-st-etienne.fr

Table of contents

Prerequisite:

- Have a PC with Matlab installed
- Have a stable internet connection
- Basic programming knowledge on Matlab At the end

of this machine-based TD, you will be able to:

- Correctly select the parameters of a Monte Carlo simulation
- Perform a simple simulation (layer model) with MCX Cloud software
- Compare the results from an MCX simulation with those from the diffusion approximation

1. Getting started with the MCX software

a. Presentation

Monte Carlo (MC) methods are a set of statistically based computational algorithms that are particularly well suited for simulations of complex systems. Unlike most model-based techniques that generate solutions by solving a set of differential equations, Monte Carlo methods generate *solutions by estimating the probability distribution after running a large number of independent random trials. Since the Monte Carlo method does not require an explicit analytical model*

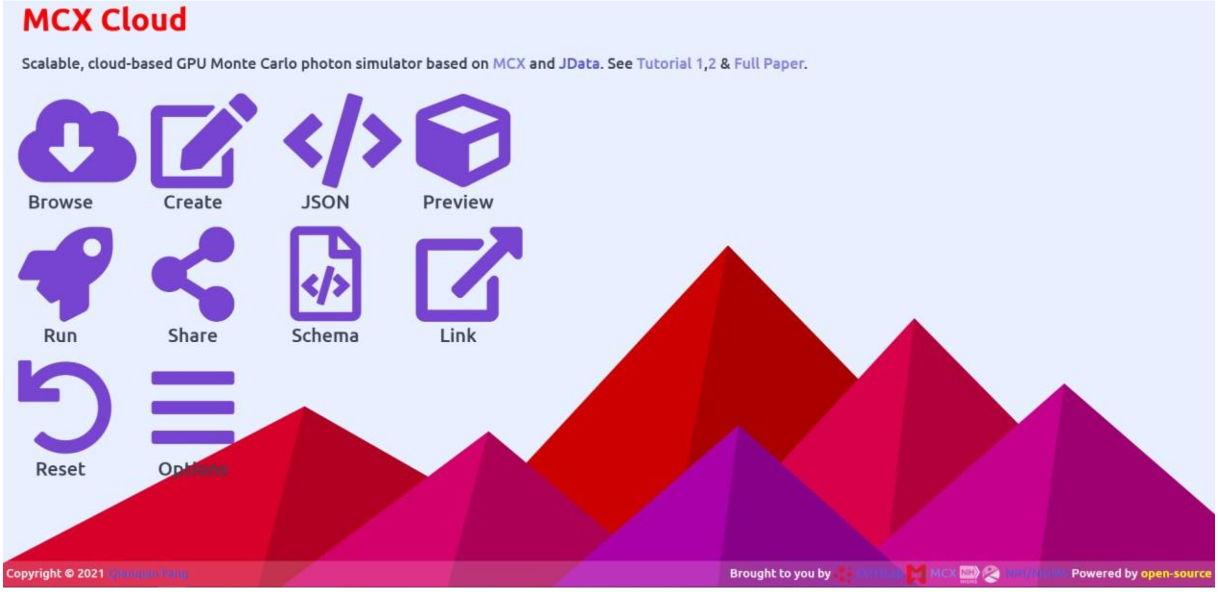
> *of the underlying system, it has proven to be a popular choice for exploring particle system physics, financial simulations and computer visualization.*
>
> *Modeling photon migration in turbid media is one such area where MC has proven to be a valuable tool, particularly in bio-optical imaging applications such as functional brain imaging and small animal imaging. Because of its generality and ability to handle arbitrary media, including low scattering, MC has been used as a reference in these cases, where efficient and accurate methods are needed to simulate photon propagation in heterogeneous human tissues. Although the general analytical model, i.e., the radiative transfer equation (RTE), exists, its solution for arbitrarily complex media is not trivial.*

Translated from Q. Fang and D. A. Boas, "Monte Carlo Simulation of Photon Migration in 3D Turbid Media Accelerated by Graphics Processing Units," Opt. Express **17**, 20178-20190 (2009)[1]

For this tutorial, we will use a Monte Carlo simulation software called "Monte Carlo eXtreme"[1]. Q. Fang and his team have developed and maintain this GPU-accelerated photon transport simulator, which is freely available online at http://mcx.space/. Several distributions of this software exist and allow to run the calculations on CPU, or CUDA compatible GPU and interface with Matlab or Python programming software. In this tutorial, so that everyone can have access to the same resources, we will use the "cloud" version of MCX.

b. Cloud

The cloud version of MCX is available at http://mcx.space/cloud/ .



MCX Cloud

Scalable, cloud-based GPU Monte Carlo photon simulator based on MCX and JData. See Tutorial 1,2 & Full Paper.

Copyright © 2021 Qianqian Fang

Brought to you by COTILab MCX NIH/NIGMS Powered by open-source

Connect to MCX Cloud.

The web interface is composed of several parts. Not all of them will be useful for the tutorial, but here is an exhaustive presentation:

- The **Browse** tab allows you to search for a simulation model already made. For example, if you type colin27 in the search bar, the result that appears is a simulation of photon propagation on the brain atlas "Colin 27".
- The **Create** tab will be very useful for this tutorial. It allows you to define a large amount of simulation parameters (although you could define even more simulation parameters if you download the version of MCX to use locally on your PC). We will go into the details of the contents of this tab later in the topic.

- The **JSON** tab allows you to enter the parameters of the simulation from a specific .json file. This mode The "expert" parameter input will not be used during the TD, but we will be able to download the automatically generated parameter file via the "Download input" button at the bottom of the page.

- The **Preview** tab allows you to have a 3D pre-rendering of the simulation volume, and the position of the sources and detectors. Note that it is in this tab that *after the* simulation, the data can be downloaded.

- The **Run** tab allows you to launch the simulation on the calculation server hosted in the United States. To do so, you will have to enter some information that will remain confidential.

- The **Share** tab allows you to share a configuration that will then be accessible to all via the **Browse** tab.

- The **Schema** tab allows you to modify the simulation volume from a specific .json file. This mode The "expert" input of the volume configuration to be simulated will not be used during the TD.

- The **Link** tab allows you to share a specific simulation configuration by link.

- The **Reset** tab allows you to restore the default settings.

- The **Options** tab allows you to change the way the simulation parameters are entered.

Now that the introductions are done, it's time to do some preliminary checks to complete the rest of the TD.

1. Check in the MCX Cloud **Options** tab that you are in "Show optional properties (with checkbox)" mode
2. In a new browser tab, go to the following address:
   https://github.com/fangq/jsonlab/archive/refs/heads/master.zip
3. In your personal space, create a new folder "TD_MonteCarlo
4. Unzip the downloaded archive into the "TD_MonteCarlo" folder. If everything goes well, you should see a new folder called "jsonlab".

c. Processing the results

To process the results of the simulations performed on MCX Cloud you will need to use Matlab. Thus :

1. Launch Matlab
2. Make sure that your Current Folder is "TD_MonteCarlo".
3. Create a main.m file in the "TD_MonteCarlo" folder
4. In the beginning of the file start with :
   a. Clear workspace : `clear all`
   b. Close all graphics windows : `close all`
   c. Clear the command window: `clc`
   d. Add the path to the folder : `addpath('./jsonlab')`
5. For your information, in the rest of the tutorial, you can use the following command to load the file myfile.json : `params=loadjson('./myfile.json')`

Now that you have all the elements in hand, let's study some of the parameters of the Monte Carlo simulation.

2. Simple simulation (Layer)

In MCX Cloud, go to the **Create** tab. You will notice the presence of 5 sub-tabs:

Shapes, **Session**, **Forward**, **Optode**, **Domain.**

a. Domain

The **Domain** sub-tab allows you to describe the size of the enclosing volume, i.e. the rectangular parallelepiped.

By default, the volume is 60 pixels x 60 pixels x 60 pixels = 216000 voxels, you may have to change these values. Note that for the Cloud version, the size is limited to 300x300x300 pixels.

Even though we are doing simulations, they are related to physical reality; thus it is necessary to define the size of a voxel side (in mm). To do this, check the "Voxel Size" box; the default value is that each voxel is 1mm^3 .

For all simulations, I encourage you to leave the origin at [0,0,0].

You will notice at the bottom of this tab, the presence of a "Media" section, which we will discuss in part 2.c.

b. Shapes

Now let's go to the **Shapes** tab. In this tab you can define different simple volume shapes. This is how you can reproduce the geometry of your volume.

You have some examples in figures 1 & 2.

In our case, we will simply represent a homogeneous environment. To do this, you just have to leave the default grid, adjusting its size to the size of the domain you have previously defined.

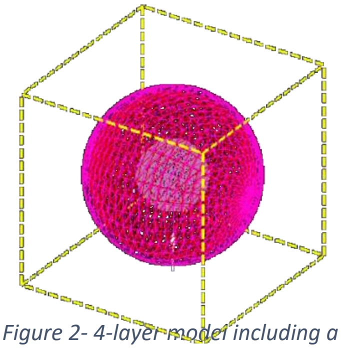


Figure 2- 4-layer model including a small sphere (sphshells)

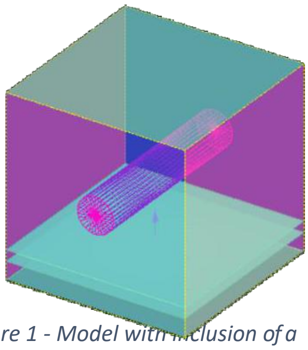


Figure 1 - Model with inclusion of a cylindrical vessel (skinvessel)

I draw your attention to the "Tag" field: It allows you to associate the optical properties for the defined shape. Here, we associates to the grid the optical properties 1.

To understand a little better how to declare several forms, I encourage you to look at the composition of the tab **Shapes** in the two examples shown in figures 1 & 2 (which you can find via the **Browse** tab).

c. Media

Now let's go back to the **Domain** tab. In the lower part, go to the "Media" part. The table in this part allows you to enter the optical properties of each medium. Each row allows to describe the four optical properties (absorption, scattering, anisotropy and optical index) of a medium.

Nota bene : the first line of the table corresponds to the description of the surrounding medium " background " which is often air ($_{\mu a}$= 0 mm^{-1} , $_{\mu s}$= 0 mm^{-1} , g=1, n=1). Thus, to describe the optical properties of the volume of Tag "1", it will be necessary to edit the second line of the table.

d. Optode

The **Optode** tab allows you to indicate the type of illumination source to be used, and to specify certain hyper parameters. In our case, we will mainly use the directional point source "pencil". It will be necessary to specify a position (X,Y,Z) and a direction.

For information, it is also possible to create a detector. It is in this detector that the parameters of the photons will be recorded.

e. Forward

Because MCX is a GPU-accelerated Monte Carlo algorithm for modeling time-resolved photon migration in arbitrary 3D turbid media, a time range for the Starting Time and Ending Time simulation must be defined as well as the Time gate width time step discretization for this simulation. The **Forward** tab allows you to enter these values in seconds. You must ensure that the total time of the simulation is sufficient for the simulation to be valid.

f. Session

The **Session** tab allows you to enter the simulation and saving parameters, in particular the number of photons, the elements to be taken into account (reflection at the interfaces, specular reflection, etc.) as well as the type of information to be saved (fluence, detected photons, type of saving file, type of data).

g. Read more

There are also a multitude of other types of configurations which are listed here:
http://mcx.space/wiki/index.cgi?Doc/MCXLAB

3. Recovering the skin's PSF

First, we recall that the point spread function (PSF) is the response of an imaging system to a point source.

In this part, we seek to find by Monte Carlo simulation, the PSF of the skin at 555 nm (given by [2]). For this, we consider that the skin is an isotropic medium *g=0*, optical index *n=1*.45, absorption coefficient $_{\mu a=0}$.03374 mm^{-1} , and scattering coefficient $_{\mu s=4}$.66811 mm^{-1} .

a. Create a result processing code

i. Enter the various parameters of the statement in the MCX Cloud software.

ii. On the **Session** tab**:**

1. set Save Fluence to true,
2. set Do normalization to true,
3. set Do reflection to true,
4. set Do Spectular reflection to true,
5. set Output file format to mc2,
6. set Output data type to f to save the fluency.

iii. Start the simulation via the **Run** tab

iv. Download the result file via the **Download output** button in the **Preview** tab

v. Load the downloaded .json file in matlab via the function mentioned previously in the TD.

b. Study of the influence of the number of photons

We now consider a simulation time large enough for the above condition to be correct.

- Perform 10 simulations for the number of photons in the table.

Number of Photons	100	1000	100000	1000000	100000000

- Study the standard deviation and the mean of the fluence over the 10 simulations of the same case.
- Plot the evolution of the standard deviation and the mean according to the different cases in the table.
- What do you conclude from this?

c. Study of the influence of the simulation time

Perform the simulations for each of the simulation times in the table below.

Simulation time T (sec)	1e-12	1e-11	1e-10	1e-9	1e-8	1e-7

- What do you observe regarding fluency? Why is this?
- What is the smallest value of simulation time to consider? Verify that it is of the order of magnitude of *T ≈ d/c*, where *c=3*.10^{8} m/s and *d* is the greatest distance of the volume (expressed in m).

d. Study of the influence of the size of the volume

Keeping the number of photons and the simulation time correct, perform the simulations for each of the simulation times in the table below.

Carry out the simulations with the optical properties of the statement as well as the number of photons and time of

correct simulation for each of the following volumes:

- ○ 300x300x60 mm^3
- ○ 150x150x30 mm^3
- ○ 75x75x15 mm^3
- ○ 30x30x6 mm^3

Nota Bene : The number of voxels must be integer (what is a half voxel ?) ; it will be relevant to change the millimeter size of a voxel between the different simulations.

- What do you notice about the last simulation?
- What relationship can you then deduce between the domain size and the mean free path $_{lt} = 1/(_{\mu a} + _{\mu s})$?

e. Impact of anisotropy

Perform a simulation for each of the anisotropy parameter values in the table below.

Anisotrpy parameter (g)	0.0	0.5	0.8

- Observe the volume of data in log10 scale using the relevant function for this
- What do you notice? Confirm your hunch by plotting the profile against the depth
- Now carry out a simulation for *g* = -0.8. What do you notice? Based on the graph of the Henyey-Greenstein phase function (which you will take care to find on the internet), what can you conclude about the MCX software ?

References

[1] Q. Fang and D. A. Boas, "Monte Carlo Simulation of Photon Migration in 3D Turbid Media Accelerated by Graphics Processing Units," *Opt Express*, vol. 17, n° 22, pp. 20178-20190, Oct. 2009, doi: 10.1364/OE.17.020178.

[2] S. L. Jacques, "Optical properties of biological tissues: a review," *Phys. Med. Biol.* vol. 58, n° 11, p. R37-R61, May 2013, doi: 10.1088/0031-9155/58/11/R37.