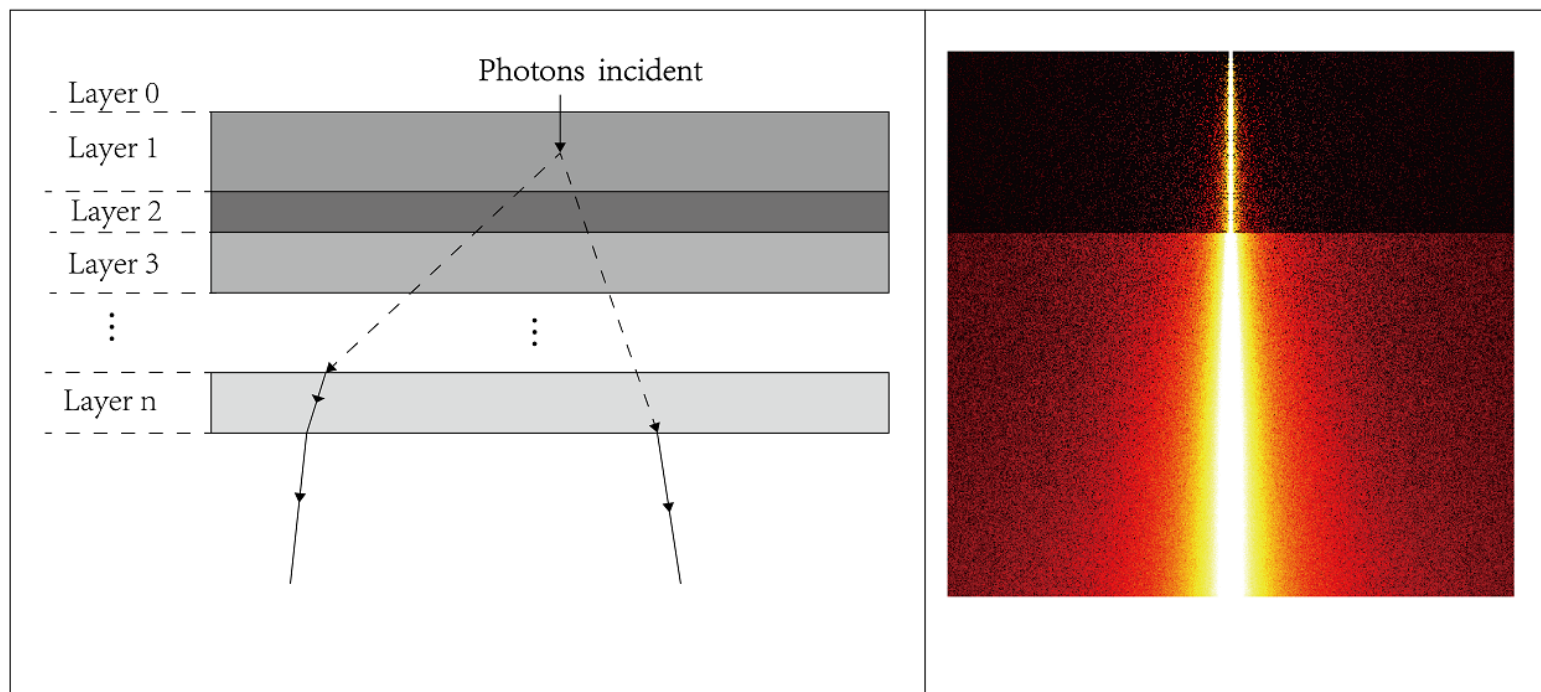


Monte Carlo Simulation in MATLAB



多层生物组织透射成像的蒙特卡洛仿真

Matlab 程序使用参考

作者：吴天翔

组织：BMO-Q's Lab

最后更新：2021 年 7 月 5 日

特别声明

此程序用于仿真组织内部发光（如荧光等），经过多层组织透射后成像的过程。1995年汪立宏老师编写了 C 语言的光子在多层组织中的蒙特卡洛仿真及反卷积的程序，其应用非常广泛。

为了能够快速修改、利用代码来达成各种其他需求，现在用 Matlab 编写了所有仿真的正向函数，并增加了透镜成像的过程。

Matlab 作为脚本语言，它的优势在于即时运行，调试方便。当有其他的应用或场景需求时，可以便捷的修改和调试。如果您只需要了解光子经多层组织的扩散情况，不需要其他的应用场景（例如成像），可以使用汪立宏老师用 C 语言编译的 exe 文件来提升运行速度。

代码总不是完美的。如果您在运行过程中出现代码报错，或遇到其他问题，可以通过邮箱联系我：wtx@zju.edu.cn。

吴天翔

2021 年 06 月 28 日

目录

1	程序的基本使用说明	2
1.1	程序输入	2
1.2	程序输出	2
1.3	文件结构	3
2	脚本文件与程序结构	5
2.1	输入参数文件: Sample_Input.m	5
2.2	数据初始化文件: MCMLini.m	5
2.3	程序运行入口: MCMLMAIN.m	8
3	正向蒙特卡洛的主函数文件	11
3.1	光子行进相关算法函数	11
3.2	不涉及算法的辅助函数	16
3.3	数据记录函数	17
3.4	数据处理相关函数	18
A	参考数据	21
A.1	散射和吸收系数数据	21

我们要解决什么问题

本文描述的蒙特卡洛仿真处理的是光子入射到多层组织中的传输过程。特别地，主要考虑光子经过界面反射、组织散射、组织吸收等过程，最终透射离开组织层，经过成像系统到达探测器过程。

仿真中，每层组织无限宽，并由下列参数描述：厚度 d 、折射率 n 、吸收系数 μ_a 、散射系数 μ_s 和各项异性因子 g 。此外，还需要给出顶部环境（例如空气）和底部环境介质（成像空间）的折射率。实际情况中组织并不可能是无限宽的，但只要它远大于光子的分布范围，这样的近似就是有效的。

在笛卡尔坐标系中将组织按一定精度栅格化，模拟每个光子在组织中传播，并记录三维栅格中的光子通量、吸收、散射和透射等信息，用来分析宏观的光束在组织内的散射和吸收情况。

同时，使用一个理想薄透镜，对焦组织内的光源点平面，以 1 倍的放大率进行成像，模拟荧光的发射、穿透组织并经过成像系统到达探测器的实际过程。

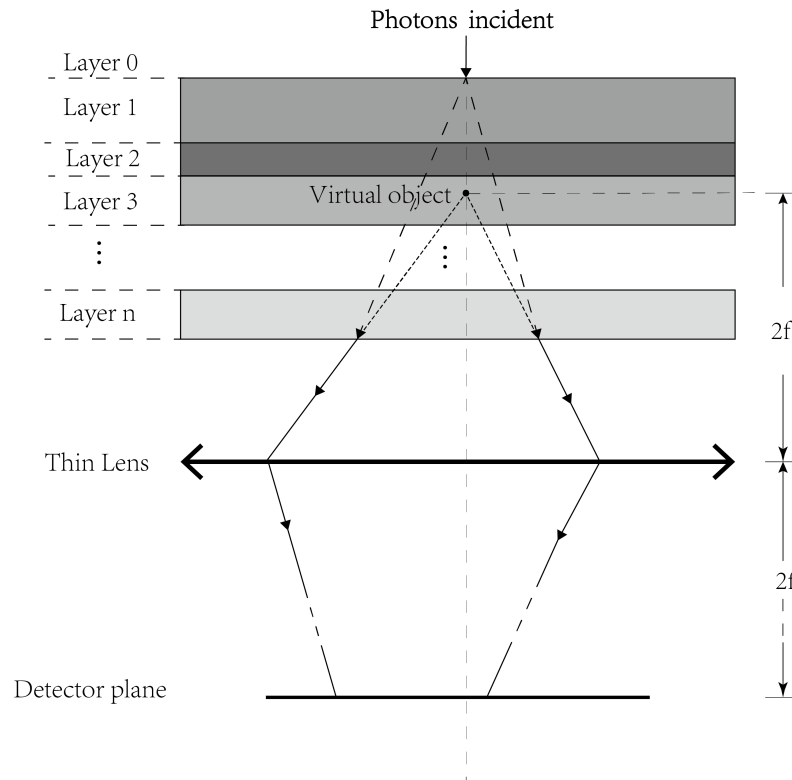


图 1: 多层组织传播、成像的流程

第一章 程序的基本使用说明

如果您仅仅利用此代码来进行仿真，只需要参考这一章节，了解程序的输入输出即可，程序详细的运行过程可以忽略。

请注意！默认情况下，将仿真无限窄的光子束垂直入射到多层组织上。

快速跳转

- 程序输入：如何修改组织参数
- 文件结构：怎样运行仿真
- 程序输出：如何查看仿真结果

1.1 程序输入

文件Sample_Input.m定义了仿真所用的初始化数据，用户可以在这个文件中修改组织参数、仿真的光子数等自定义参数。运行主程序前，请仔细检查输入参数的数量是否对应正确，有关输入参数的含义，详见表 1.1。

表 1.1: 输入参数详细信息

变量名	意义	单位	元素个数	输入示例 ¹
np	仿真的光子数量	个	1	np=1e5
nl	组织层数	层	1	nl=2
n	组织折射率	-	nl+2 ²	n=[1,1.3,1.3,1]
mua	组织吸收系数	cm ⁻¹	nl	mua=[1.1,1.2]
mus	约化散射系数	cm ⁻¹	nl	mus=[1,10]
g	平均散射余弦	-	nl	g=[0.8,0.9]
d	组织层厚度 ³	cm	nl	d=[0.1,0.2]
nz	z 轴栅格化精度	-	1	nz=200
nr	中心距 r 栅格化精度	-	1	nr=200
na	角度 a 栅格化精度	-	1	na=100
rlim	组织表面半径 ⁴	cm	1	rlim=2

注 对Sample_Input.m文件修改完成后，保存即可，不需要运行此 m 文件。

1.2 程序输出

仿真全部运行完成后，在 MATLAB 工作区将会出现一个名为 Output 的结构体变量，所有的输出结果都会保存在此结构体中。

¹以 2 层组织为例。

²折射率变量包含组织层和上下两侧介质的折射率，组织上侧和下侧的折射率分别是 n 的第一个和最后一个参数。所以一共有 nl+2 个元素。

³注意！仿真时间与组织厚度和组织参数有关，如果组织厚度过大且散射系数很大，散射次数过多可能会导致结果存在偏差，这是由于数据计算中的截断误差的大量累计导致的。

⁴注意！此参数只限制画图时考虑的组织范围，实际仿真中，组织是无限大的。

对于特定的应用场景来说，不是所有的输出参数都有意义。因此在数据分析前，您可以先从 `Output` 中复制出需要的输出参数，但请注意尽量不要直接对输出的结构体变量进行编辑。或者，您可以将输出的结果 `Output` 复制一份，再进行操作。

分析多层组织的透射成像应用中，可用的部分输出参数的意义见表 1.2。若要查看全部输出参数的详细信息，请参见表 2.3 on page 9。

表 1.2: 部分输出参数详细信息

变量名	意义	单位	元素个数	备注
<code>Output.A_rz</code>	2 维吸收概率分布	$1/cm^3$	<code>nr×nz</code>	距离 <code>r</code> 和深度 <code>z</code>
<code>Output.A_l</code>	每层组织的吸收概率	-	<code>nl</code>	
<code>Output.A</code>	总吸收率	-	1	
<code>Output.Tt_xy</code>	2 维透射分布	-	<code>nr×nr</code>	横纵坐标: <code>x-y</code>
<code>Output.Tt_r</code>	1 维透射分布	$1/cm^2$	<code>nr</code>	沿距离 <code>r</code> 的分布
<code>Output.Tt</code>	总透射率	-	1	
<code>Output.Im</code>	2 维透射分布	-	<code>512×512</code>	探测面强度分布 ⁵
<code>Output.E_rz</code>	2 维散射能量分布	-	<code>nr×nz</code>	距离 <code>r</code> 和深度 <code>z</code>



笔记 在透射成像的仿真中，似乎只需要关注以下变量：

1. `Output.Tt_xy`, 组织最后一层底部透射的光子强度分布。
2. `Output.Tt_r`, 透射能量沿远离入射点距离 `r` 的分布。
3. `Output.Im`, 经过成像系统后到达探测器的光子的强度分布，成像系统以 1 倍放大率成像。
4. `Output.E_rz`, 沿着组织 `z` 轴和距离 `r` 的散射能量分布，是输出参数中最接近人眼看到的光线真实分布的参量。

在一次仿真中，以上四个参数的图像可见图 1.1 on the next page。

1.3 文件结构

在执行仿真时，仅需运行主文件 `MCMLMAIN.m` 即可。 仿真程序共包含四个文件，在运行并按照以下顺序依次执行：

1. `MCMLMAIN.m`, 主文件, 程序入口
2. `Sample_Input.m`, 读取输入参数
3. `MCMLini.m`, 数据初始化
4. `MCMLGO.m`, 光子运动与记录

注 运行主文件后，对输出数据的任何处理（成像、对比度调整、缩放等）请：

- 在 Matlab 中新建脚本文件处理；
- 在工作区打开要处理的变量，右键保存数据，使用其他处理软件进行处理。

⁵这是经过成像透镜后，到达探测器面并被接收的光子的强度分布。

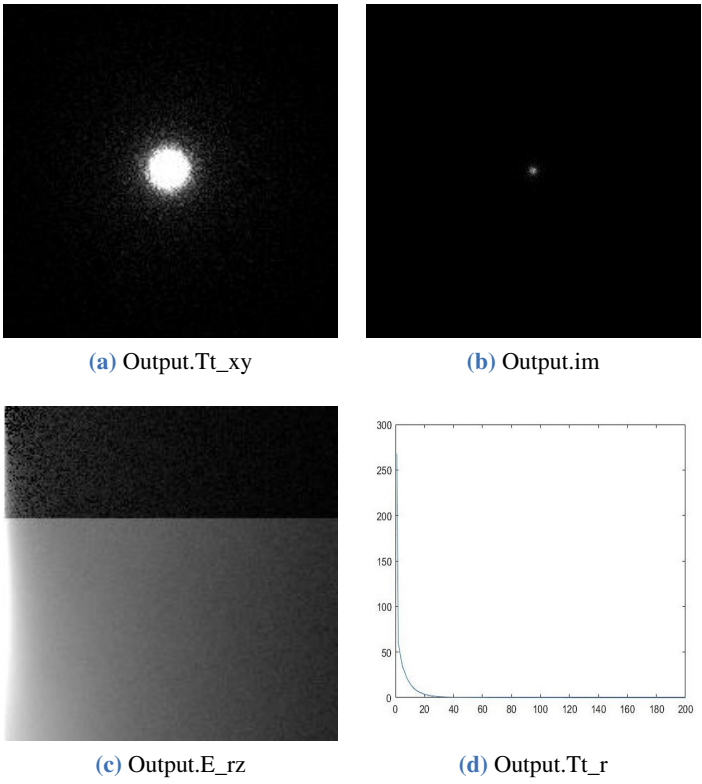


图 1.1: 部分输出参数的图像

第二章 脚本文件与程序结构

此章节将会涉及详细的文件和变量的分析，在了解变量意义之后，您可以按需求直接在函数文件中增加语句、数据处理，或修改之前没有提及的变量，例如成像透镜的位置、焦距、探测器的分辨率和像元大小、光源汇聚或发散等。

2.1 输入参数文件：Sample_Input.m

输入参数文件 `Sample_Input.m` 是一个 Matlab 脚本文件。

它将用户定义的输入参数（组织的散射吸收系数、层数和厚度、栅格化精度等）导入 Matlab 工作区。输入参数的详细信息见表 1.1 on page 2。

2.2 数据初始化文件：MCMLini.m

初始化文件 `MCMLini.m` 是一个 Matlab 脚本文件。修改光源类型及其参数、透镜参数、探测器参数时，必须在此脚本中进行。

它将工作区的用户自定义参数打包成结构体变量，便于在仿真函数中互相传递。在对数据进行打包前，它将检查输入参数的合理性。同时，为提高运算速度，它会对输出数据、光子参数、成像透镜等参数进行运算，并为这些变量并预分配内存。

初始化后的工作区变量的意义可以参考表 2.1。

表 2.1: 经初始化的参数信息

变量名	变量类型	意义
Input	struct	仿真的输入参数
Output	struct	仿真的输出参数
Photon	struct	运动光子的参数
Layer	struct	组织层的参数
InputCheck	logical	输入参数检查（正常为 1）

2.2.1 输入参数结构体：Input

此结构体包含仿真除运动光子和组织层详细信息以外其他的所有信息，它包含的参数有：组织大小、栅格化精度、透镜和探测器信息、仿真的总光子数量。

结构体 `Input` 的详细参数可以见表 2.2 on the following page，它包含了用户在文件 `Sample_Input.m` 中输入的所有参数。



笔记 如果您想修改 `Sample_Input.m` 中的参数，请尽量在 `Sample_Input.m` 文件中进行。

请注意！如果您在初始化文件 `MCMLini.m` 中直接以数值方式进行赋值，在 `Sample_Input.m` 中的修改 将会失效。

表 2.2: 输入参数 Input 详细信息

变量名	意义	单位	备注
Input.Photon_num	仿真的光子数量	个	
Input.Layer_num	组织层数	层	
Input.LayerUp_n	组织上侧折射率	-	默认值 1
Input.LayerDown_n	组织下侧折射率	-	默认值 1
Input.wth	轮盘赌权重阈值	-	默认值 0.001
Input.dz	z 轴栅格大小	cm	
Input.dr	半径 r 栅格大小	cm	
Input.da	立体角 a 栅格大小	sr	
Input.nz	z 轴栅格化数目	-	
Input.nr	中心距 r 栅格化数目	-	
Input.na	立体角 a 栅格化数目	-	
Input.SourceType	光源类型	-	1-细光束 2-汇聚光 3-发散光
Input.BeamR	入射光束宽度	cm	光源汇聚时才会出现
Input.BeamDepth	光源汇聚深度	cm	光源汇聚时才会出现
Input.BeamAngle	光源发散半角	rad	光源发散时才会出现
Input.Lens_f	成像透镜焦距	cm	默认值 4
Input.Lens_D	成像透镜直径	cm	默认与组织大小相同
Input.Lens_n	成像透镜折射率	-	默认值 1.5
Input.Lens_z	透镜 z 轴位置	cm	程序自动对焦
Input.Detector_z	探测器 z 轴位置	cm	默认距透镜 2f
Input.Detector_x	探测器 x 方向大小	cm	默认与组织大小相同
Input.Detector_y	探测器 y 方向大小	cm	默认与组织大小相同
Input.Detector_xnum	探测器 x 方向像素数	-	默认值 512
Input.Detector_ynum	探测器 y 方向像素数	-	默认值 512
Input.Detector_dx	探测器 x 方向像元大小	cm	
Input.Detector_dy	探测器 y 方向像元大小	cm	

变量 2.1. Input.SourceType

变量Input.SourceType用于指定输入光源的性质。

当Input.SourceType=1时, 光源以无限窄光束垂直入射, 入射坐标 (0, 0, 0)。

当Input.SourceType=2时, 光源以汇聚光入射至组织中, 入射宽度和聚焦深度分别由变量Input.BeamR和变量Input.BeamDepth指定, 如图 2.1(b) 中所示。

当Input.SourceType=3时, 光源以发散光入射至组织中, 发散半角由变量Input.BeamAngle指定。如图 2.1(a) 中所示。

如果您修改了光源类型, 程序会使透镜自动对焦至光源的聚焦点所在平面。如果您需要透镜对焦到其他平面, 请手动修改透镜位置。

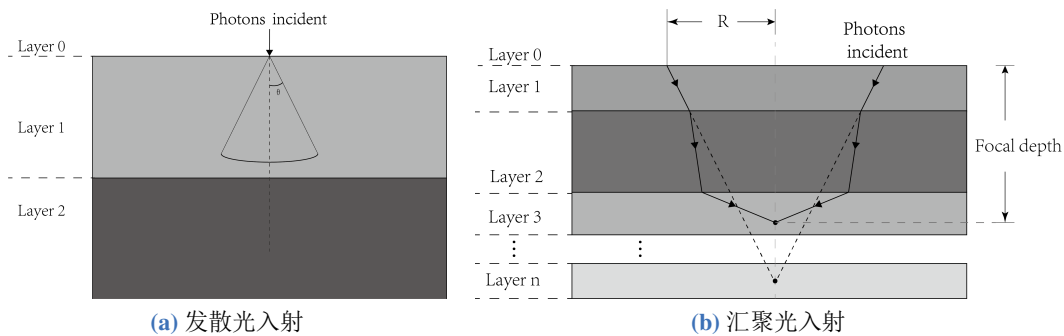


图 2.1: 光源类型的参数

变量 2.2. Input.Lens_f

变量Input.Lens_f 用于指定成像透镜的焦距, 单位是 cm。

成像透镜的焦距是用户设定的, 在程序中默认设定为 4cm。程序会根据组织的厚度、层数、折射率等信息, 按照弹道光传播对透镜进行移动, 始终保持透镜对焦在光子入射点对应的虚拟物点, 对焦情况参见图 2.2。

当您尝试修改探测器和透镜信息时, 请确认数据是否符合以下要求:

- 透镜焦距大于 0;
- 透镜 z 轴位置必须大于组织总厚度; 由于透镜对焦是程序自动完成的, 如果仍然出现透镜位置在组织内的情况, 请尝试增加透镜的焦距。
- 清晰成像时, 探测器应当位于透镜下方 2 倍焦距处;



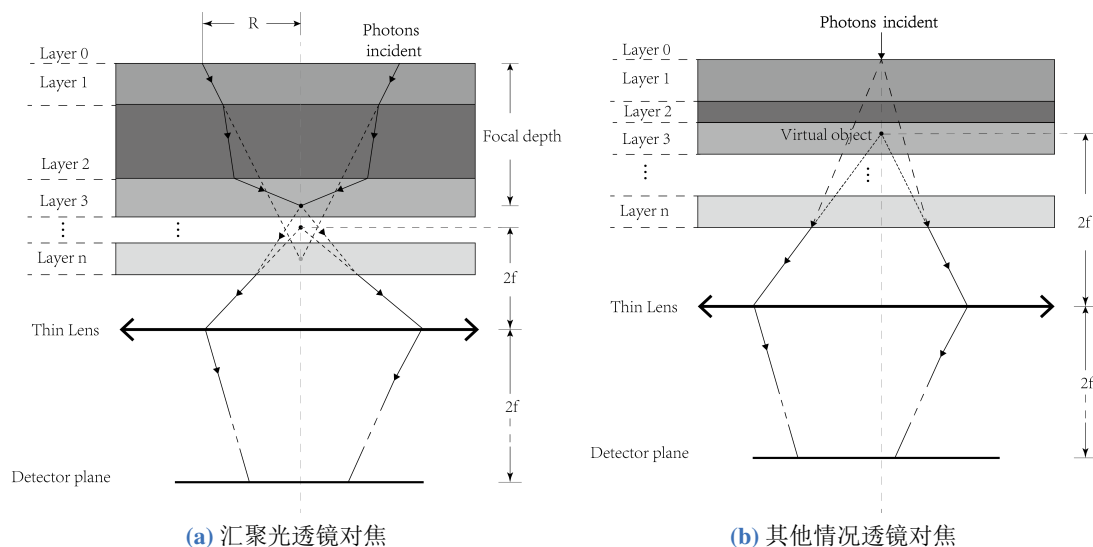


图 2.2: 透镜的自动对焦

2.2.2 输出参数结构体：Output

输出结构体 **Output** 包含了以距离 r 、深度 z 、立体角 a 和笛卡尔坐标 xy 为维度的透射、反射和散射数据。输出参数的详细含义见表 2.3 on the next page。

2.2.3 光子结构体：Photon

光子结构体 **Photon** 包含了光子的位置、方向余弦、权重、移动步长等信息，详细的参数见表 2.4 on the following page。

光子结构体是仿真中的重要结构体，它的参数每一次运动后都会改变。所有光子都将共用这段内存空间，因此在每个光子发射前，需要经过初始化。此初始化由函数 **LaunchPhoton** 进行，详情参见函数 3.1 on page 11。



笔记 光子穿过边界时，运动会拆分为两步。先移动至边界，考虑折射或反射，再移动第二步。变量 `Photon.sleft` 用来表示第二步的步长。

2.2.4 组织层结构体：Layer

生物组织层结构体 **Layer** 包含了所有的组织信息，同时包含在层间全反射的临界角余弦值。详细的参数见表 2.5 on the following page。



笔记 此结构体总会带有组织下侧的折射率信息。因此即使您定义了 n 层组织，结构体 **Layer** 仍会包含 $n+1$ 层的信息。

2.3 程序运行入口：MCMLMAIN.m

运行文件 **MCMLMAIN.m** 时，将载入输入数据，对数据做初始化，如果输入正确，则开始仿真。

表 2.3: 输出参数 Output 详细信息

变量名	意义	单位	元素个数	备注
Output.Rsp	-	-	-	保留
Output.Rd_ra	2 维反射分布	$1/(cm^2 \cdot sr)$	$nr \times na$	距离 r 和立体角 a
Output.Rd_r	1 维反射概率分布	$1/cm^2$	nr	沿距离 r 的分布
Output.Rd_a	1 维反射分布	$1/sr$	na	沿立体角 a 的分布
Output.Rd	总反射率	-	1	
Output.A_rz	2 维吸收概率分布	$1/cm^3$	$nr \times nz$	距离 r 和深度 z
Output.A_z	1 维吸收概率密度	$1/cm$	nz	沿距离 r 的分布
Output.A_l	每层组织的吸收概率	-	nl	
Output.A	总吸收率	-	1	
Output.Tt_ra	2 维透射分布	$1/(cm^2 \cdot sr)$	$nr \times na$	距离 r 和立体角 a
Output.Tt_xy	2 维透射分布	-	$nr \times nr$	横纵坐标: x - y
Output.Tt_r	1 维透射分布	$1/cm^2$	nr	沿距离 r 的分布
Output.Tt_a	1 维透射分布	$1/sr$	na	沿立体角 a 的分布
Output.Tt	总透射率	-	1	
Output.Im	2 维透射分布	-	512×512	探测面强度分布
Output.E_rz	2 维散射能量分布	-	$nr \times nz$	距离 r 和深度 z

表 2.4: 光子参数 Photon 详细信息

变量名	意义	单位	备注
Photon.x	光子的笛卡尔坐标 (x)	cm	
Photon.y	光子的笛卡尔坐标 (y)	cm	
Photon.z	光子的笛卡尔坐标 (z)	cm	
Photon.mux	x 方向余弦	-	
Photon.muy	y 方向余弦	-	
Photon.muz	z 方向余弦	-	
Photon.w	光子权重	-	
Photon.dead	光子是否死亡	-	1: 死亡
Photon.layer	当前位于第几层组织	-	
Photon.s	下一步移动步长	cm	
Photon.sleft	碰壁后的剩余步长	cm	

表 2.5: 组织层参数 Layer 详细信息

变量名	意义	单位	备注
Layer.n	折射率	-	包含组织下层的折射率
Layer.mua	组织吸收系数	cm^{-1}	
Layer.mus	约化散射系数	cm^{-1}	
Layer.g	平均散射余弦	-	
Layer.z0	层起始坐标 (z 轴)	cm	
Layer.z1	层结束坐标 (z 轴)	cm	
Layer.cos_crit0	上界面全反射临界角余弦	-	不存在全反射时, 值为 0
Layer.cos_crit1	下界面全反射临界角余弦	-	

```
run Sample_Input.m % Load input data
run MCMLini.m % Initialization data
if InputCheck
    [Photon,Input,Output] = MCMLGO(Photon,Layer,Input,Output);
end
```

第三章 正向蒙特卡洛的主函数文件

文件 MCMLGO.m 是标准 Matlab 函数文件，它包含蒙特卡洛仿真的主要函数。这一章节将详细介绍文件中的函数及其结构。

请注意，文中的代码可能与实际的仿真代码不同。为了更好的说明函数作用，文中出现的代码可能做了些许改动。

快速跳转

☐ MCMLGO

☐ 文件结构：怎样运行仿真

☐ 程序输出：如何查看仿真结果

3.1 光子行进相关算法函数

函数 3.1. LaunchPhoton

每一个光子出发前，由 LaunchPhoton 函数执行初始化。因此，调整光源的汇聚或发散的具体参数，也经过此函数进行。如果您没有对光源做任何修改，它将按照下面的代码对光子进行初始化，使光源为无限窄的细光束垂直入射。



```
function Photon = LaunchPhoton(Photon,Layer,Input)
Photon.w = 1 ;
Photon.dead = 0;
Photon.layer = 1;% in tissue
Photon.x = 0;
Photon.y = 0;
Photon.z = 0; %Cartesian coordinates.[mm]
Photon.mux =0;
Photon.muy =0;
Photon.muz =1;%directional cosines of a photon.
end
```

函数 3.2. ConvergBeam

ConvergBeam 使用输入参数创建一个汇聚光的光子，其位置在给定的光束宽度内随机取得，光子方向指向焦点。



```
function [Photon] = ConvergBeam(R,Depth,Photon)

r = rand()*R;
x = 2*r * rand()-r; %-r,r
temp = sqrt(r^2-x^2);
```

```

y = 2*temp * rand() - temp; % -temp,temp

s = sqrt(x^2 + y^2 + Depth^2);
Photon.x = x;
Photon.y = y;
Photon.z = 0;
Photon.mux = -x/s;
Photon.muy = -y/s;
Photon.muz = Depth/s;
end

```

函数 3.3. RandomDirect

RandomDirect 使用输入参数创建一个发散光的光子，光子位于 (0, 0, 0)，在给定的发射角内以随机角度内发射。



```

function [Photon] = RandomDirect(Photon,Input)
%Setting light angle with angle as half angle
    angle = Input.BeamAngle;
    phi1 = ( 2 * angle * rand - angle);%(-angle,angle)
    phi2 = 2 * pi * rand;
    cos_phi1 = cos(phi1);
    sin_phi1 = sin(phi1);
    cos_phi2 = cos(phi2);
    sin_phi2 = sin(phi2);

    Photon.mux = sin_phi1 * sin_phi2;
    Photon.muy = sin_phi1 * cos_phi2;
    Photon.muz = cos_phi1;

    Photon.x = 0;
    Photon.y = 0;
    Photon.z = 0;
end

```

函数 3.4. SpinTheta

SpinTheta 将计算一个新的偏折 θ 角的余弦值。如果平均散射余弦为 0，则计算公式为： $\cos(\theta) = 2 \times \text{rand}() - 1$ 。如果 g 不是 0，则根据 Henyey-Greenstein 函数计算余弦值。



```

function cost = SpinTheta(g)
if g == 0
    cost = 2*rand()-1;

```

```

else
    temp = (1-g*g)/(1-g+2*g*rand());
    cost = (1+g*g - temp*temp)/(2*g);
    if cost < -1
        cost = -1;
    elseif cost > 1
        cost = 1;
    end
end
end
end

```

函数 3.5. Spin

Spin 在垂直于传播方向的平面内旋转光子的前进方向，这个方向角 $\varphi \in [0, 2\pi)$ 。值得注意的是，计算角度的过程中， $\sin(\theta)$ 始终大于 0，因为 $\theta \in (0, \pi)$ 。因此可以用 $\sqrt{1 - \cos^2(\theta)}$ 来计算 $\sin(\theta)$ ，这将比使用 $\sin()$ 函数运行的速度更快。



```
function Photon = Spin(g,Photon)
```

函数 3.6. Hop

Hop 将光子在当前层中移动。



```

function Photon = Hop(Photon)
s = Photon.s;
Photon.x = Photon.x + s* Photon.mux;
Photon.y = Photon.y + s* Photon.muy;
Photon.z = Photon.z + s* Photon.muz;
end

```

函数 3.7. StepSizeInGlass

StepSizeInGlass 计算玻璃中光子的行进步长。如果 z 方向余弦为 0，函数返回 0。请务必在调用前确保光子的 z 方向余弦不为 0。



```

function Photon = StepSizeInGlass(Photon,Layer)
muz = Photon.muz;%dl_b :step size to boundary
if Photon.layer ==0 %Tianxiang 21/06/30 photon in layer 0.
    if muz <=0
        Photon.dead = 1;
    elseif muz >0
        dl_b = (Layer(1).z0 - Photon.z)/muz;
    end
else
    if muz >0

```



```

        dl_b = (Layer(Photon.layer).z1 - Photon.z)/muz;
    elseif muz < 0
        dl_b = (Layer(Photon.layer).z0 - Photon.z)/muz;
    else
        dl_b = 0;
    end
end
Photon.s = dl_b;
end

```

函数 3.8. StepSizeInTissue

StepSizeInTissue 为光子计算一个在组织内传播的步长。如果光子的碰壁剩余步长为 0，则计算一个新的步长，否则直接取剩余步长为下次移动的步长。



```

function Photon=StepSizeInTissue(Photon,Layer)
mua = Layer(Photon.layer).mua;
mus = Layer(Photon.layer).mus;

if Photon.sleft == 0 % make a new step
    Photon.s = -log(rand)/(mua+mus);
else %take the leftover.
    Photon.s = Photon.sleft/(mua+mus);
    Photon.sleft = 0;
end
end

```

函数 3.9. HitBoundary

HitBoundary 检查光子的下一步移动是否会碰撞组织的层边界，将要撞击边界返回 1，否则返回 0。



```

function [HitOrNot,Photon] = HitBoundary(Photon,Layer)

```

函数 3.10. Drop

Drop 使组织中的光子权重下降。此时的光子必须是存活状态，即 Photon.dead 为 0。

权重下降为 $dw = w \times \frac{\mu_a}{\mu_a + \mu'_s}$ 。

权重的下降值 dw 将被累加记录在吸收矩阵 Output.A_rz 中，而下降后的权重值将被累加记录在散射矩阵 Output.E_rz 中。



```

function [Photon,Output] = Drop(Photon,Layer,Input,Output)

```

函数 3.11. Roulette

Roulette 对低于阈值权重的光子执行轮盘赌。



```
function Photon = Roulette(Photon)
```

函数 3.12. RFresnel

RFresnel 计算菲涅尔反射率。调用函数前，必须确认入射角余弦值为正。



```
function [r,ca2] = RFresnel(n1,n2,ca1)
```

函数 3.13. CrossUpOrNot

CrossUpOrNot 决定光子能否从一个组织层的上表面透射入前一个组织层。如果光子已经位于第一层组织，则判断光子能否从组织上表面逃逸出去。



```
function [Photon,Output] = CrossUpOrNot(Photon,Input,Output,Layer)
```

函数 3.14. CrossDnOrNot

CrossDnOrNot 决定光子能否从一个组织层的下表面透射入更深的组织层。如果光子已经位于最后一层，则以菲涅尔反射率为概率决定光子是否从组织中逃逸。



```
function [Photon,Output] = CrossDnOrNot(Photon,Input,Output,Layer)
```

函数 3.15. HopInGlass

HopInGlass 在玻璃层中移动光子。玻璃层是指散射和吸收系数都为 0 的组织层。如果光子是平行于层传播，将被放弃追踪，当然这种情况极少。



```
function [Photon,Output] = HopInGlass(Photon,Input,Output,Layer)
if Photon.muz ==0
    Photon.dead = 1;
else
    Photon = StepSizeInGlass(Photon,Layer);
    Photon = Hop(Photon);
    [Photon,Output]=CrossOrNot(Photon,Input,Output,Layer);
end
end
```

函数 3.16. HopDropSpinInTissue

HopDropSpinInTissue 计算一个步长，移动光子并降低其权重值，并重新计算下一步的传播方向。如果某次计算出的步长足够让光子撞击边界，这一步长就会被分为两段步长：

1. 移动光子至边界，这个过程没有吸收和散射；

2. 在边界确定光子是透射或反射;
3. 使用剩余的步长在透射层 (透射) 或当前层 (反射) 传播;
4. 如果剩余步长仍然足够撞击边界, 则重复上述步骤。



```
function [Photon,Output]= HopDropSpinInTissue(Photon,Input,Output,Layer)
Photon=StepSizeInTissue(Photon,Layer);
[HitOrNot,Photon] = HitBoundary(Photon,Layer);
if HitOrNot
    Photon = Hop(Photon); % move to boundary plane.
    [Photon,Output]=CrossOrNot(Photon,Input,Output,Layer);
else
    Photon = Hop(Photon);
    [Photon,Output] = Drop(Photon,Layer,Input,Output);
    Photon = Spin(Layer(Photon.layer).g,Photon);
end
end
```

3.2 不涉及算法的辅助函数

函数 3.17. MCMLGO

MCMLGO 是蒙特卡洛仿真的入口, 它执行仿真并正确处理数据。



```
function [Photon,Input,Output] = MCMLGO(Photon,Layer,Input,Output)

[Photon,Output] = DoOneRun(Photon,Input,Output,Layer);
[Input,Output] = SumScaleResult(Layer,Input,Output);
end
```

函数 3.18. CrossOrNot

CrossOrNot 光子撞击边界时, 根据光子的 z 轴方向余弦决定调用上行或下行穿越函数, 是一个辅助函数。



```
function [Photon,Output] = CrossOrNot(Photon,Input,Output,Layer)
if Photon.muz <0
    [Photon,Output]=CrossUpOrNot(Photon,Input,Output,Layer);
else
    [Photon,Output]=CrossDnOrNot(Photon,Input,Output,Layer);
end
end
```

函数 3.19. HopDropSpin

HopDropSpin 判断光子所处的层是组织还是玻璃，按情况分别调用对应的移动函数。如果光子权重太低，则执行轮盘赌。不涉及算法，是一个辅助函数。



```
function [Photon,Output]= HopDropSpin(Photon,Input,Output,Layer)

if Photon.layer==0 % Tianxiang 21/6/30 photon in layer 0
    [Photon,Output]=HopInGlass(Photon,Input,Output,Layer);
else
if (Layer(Photon.layer).mua==0 && Layer(Photon.layer).mus==0) % glass layer
    [Photon,Output]=HopInGlass(Photon,Input,Output,Layer);
else
    [Photon,Output]= HopDropSpinInTissue(Photon,Input,Output,Layer);
end
end
if Photon.w < Input.wth && Photon.dead ~=0
    Photon = Roulette(Photon);
end
end
```

函数 3.20. DoOneRun

DoOneRun 做一次独立的蒙特卡洛仿真。按照输入的光子数进行仿真，是不涉及算法的执行函数。



```
function [Photon,Output] = DoOneRun(Photon,Input,Output,Layer)
PhotonNum = Input.Photon_num;
while(PhotonNum >=0)
    PhotonNum = PhotonNum -1;
    Photon = LaunchPhoton(Photon,Layer,Input);
    while Photon.dead == 0
        [Photon,Output]= HopDropSpin(Photon,Input,Output,Layer);
    end
end
end
```

3.3 数据记录函数

函数 3.21. RecordR

RecordR 记录从组织最上层离开的光子，同时更新光子的权重。



```
function [Photon,Output] = RecordR(refl,Photon,Input,Output)
```

函数 3.22. RecordT

RecordT 记录从最下层组织透射的光子关于中心距 r 和立体角 a 的矩阵，同时更新光子的权重。



```
function [Photon,Output] = RecordT(refl,Photon,Input,Output)
```

函数 3.23. RecordXY

RecordXY 专门用来记录从最下层组织透射的光子在 xy 平面的分布，同时更新光子的权重。



```
function Output = RecordXY(refl,Photon,Input,Output)
```

3.4 数据处理相关函数

函数 3.24. Sum2DRd

Sum2DRd 从二维的反射数据中相加来获得一维和零维的数据。生成的数据包括:Output.Rd_a、Output.Rd。



```
function Output = Sum2DRd(Input,Output)
...
end
```

函数 3.25. IzToLayer

IzToLayer 由给定的 z 轴栅格数，计算此栅格位于组织的那一层，函数返回一个数字，代表输入栅格所在的组织层数。



```
function i = IzToLayer(Iz,Layer,Input)
```

函数 3.26. Sum2DA

Sum2DA 从二维的吸收数据中累加获得一维和零维的反射数据。生成的数据包括:Output.A_z、Output.A。



```
function Output = Sum2DA(Layer,Input,Output)
```

函数 3.27. Sum2DTt

Sum2DTt 从二维的透射数据中累加获得一维和零维的透射数据。生成的数据包括:Output.Tt_r、Output.Tt_a、Output.Tt。



```
function Output = Sum2DTt(Input,Output)
```

函数 3.28. ScaleRdTt

ScaleRdTt 正确地缩放透射矩阵中的数据。由于光子在入射时，即使光子在 x, y 方向都有相同且均匀的概率密度分布，但是对于 r 方向，由于下式：

$$r = \sqrt{x^2 + y^2}$$

导致 r 方向的概率密度分布不均，因此需要调整。



```
function [Output] = ScaleRdTt(Input,Output)
nr = Input.nr;
na = Input.na;
dr = Input.dr;
da = Input.da;
scale1 = 4 *pi *pi *dr *sin(da/2)*dr *Input.Photon_num;
%The factor (ir+0.5)*sin(2a) to be added.
for ir = 1:nr
    for ia = 1:na
        scale2 = 1.0/((ir-0.9)*sin(2.0*ia*da)*scale1);
        Output.Rd_ra(ir,ia) = Output.Rd_ra(ir,ia) *scale2;
        Output.Tt_ra(ir,ia) = Output.Tt_ra(ir,ia) *scale2;
    end
end
...
end
```

函数 3.29. ScaleA

ScaleA 正确缩放吸收相关的数据。有关缩放的原因，在函数 3.28 中有说明。



```
function [Output] = ScaleA(Input,Output)
```

函数 3.30. DeleteXY

DeleteXY 删除出界的数据。在仿真中，所有超出规定量程的光子，都将被记录在矩阵、数组的最后一行、列或元素中，在输出结果前将出界的信息删除。



```
%Tianxiang 21/07/01 Delete out-of-bounds data
function [Input,Output] = DeleteXY(Input,Output)
```

函数 3.31. SumScaleResult

SumScaleResult 在仿真结束后，计算一维和零维数据，并对所有结果执行缩放。



```
function [Input,Output] = SumScaleResult(Layer,Input,Output)
Output = Sum2DRd(Input,Output);
Output = Sum2DA(Layer,Input,Output);
```

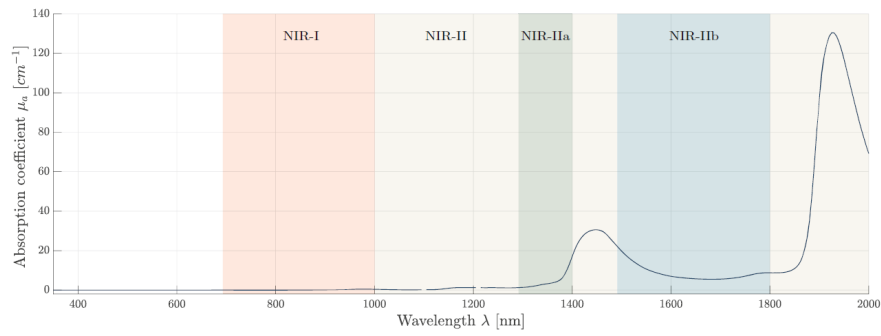
```
Output = Sum2DTt(Input,Output);  
  
Output = ScaleRdTt(Input,Output);  
Output = ScaleA(Input,Output);  
% Tianxiang 21/07/01  
[Input,Output] = DeleteXY(Input,Output);  
end
```

附录 A 参考数据

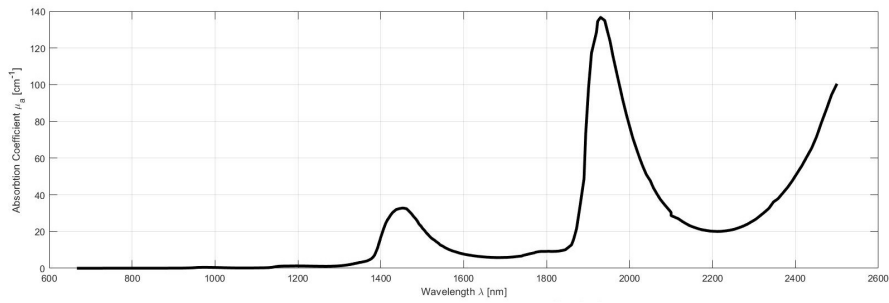
A.1 散射和吸收系数数据

A.1.1 水的吸收系数图

水在可见光几乎没有吸收。在近红外，吸收系数开始震荡增加，在 980, 1450, 1910 处有局部吸收峰。详见图 A.1。



(a) 400-2000 nm

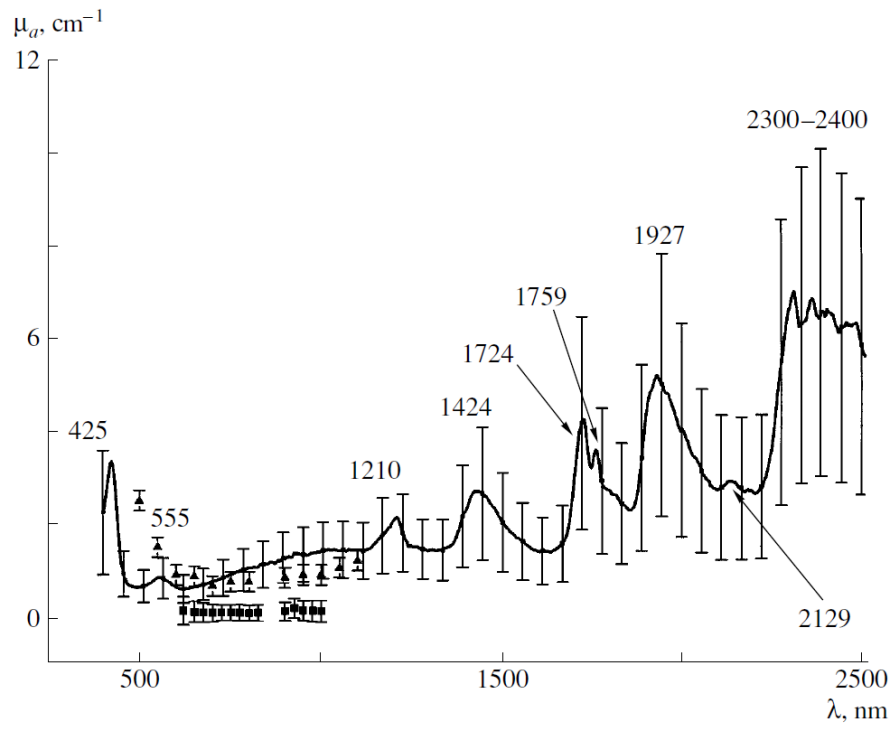


(b) 670-2500 nm

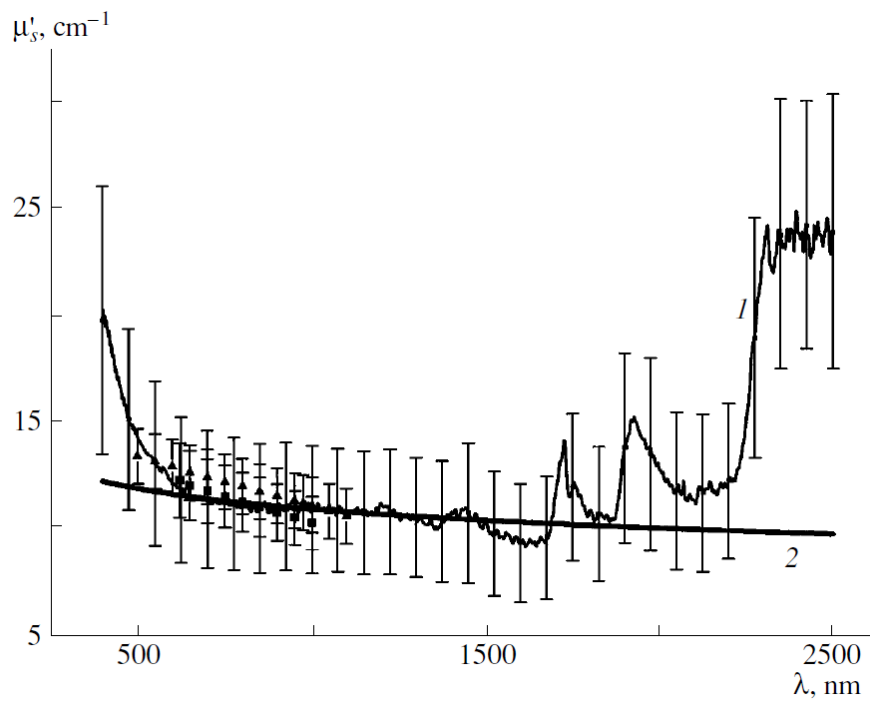
图 A.1: 水吸收图谱

A.1.2 油脂吸收与散射系数

油脂也是活体生物的重要组成部分，其光学参数对成像有较大影响。油脂的吸收和散射系数见图 A.2 on the following page。



(a) 吸收系数



(b) 散射系数

图 A.2: 油脂光学参数