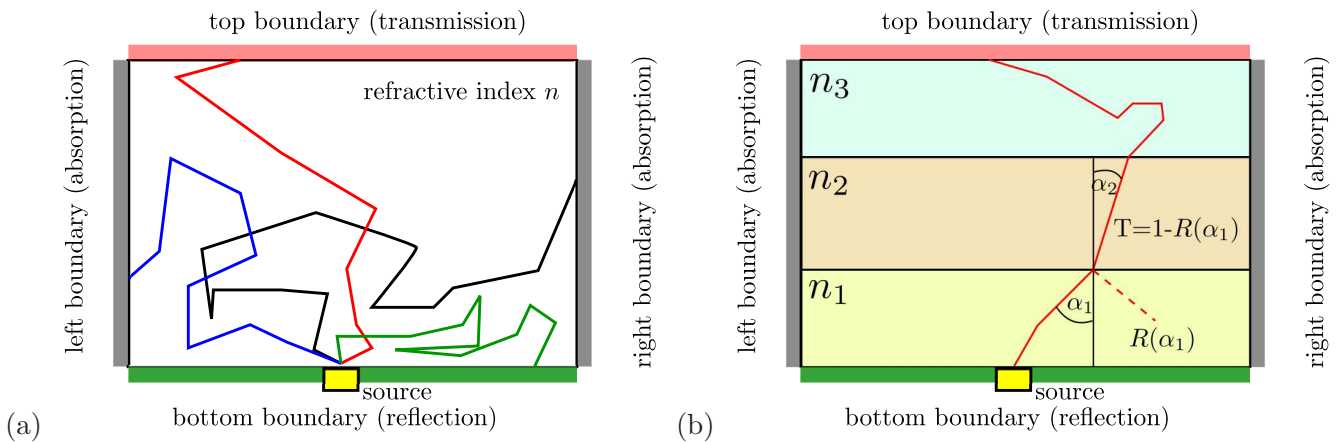


# Monte Carlo: symulacja rozpraszania fotonów w ośrodku częściowo przezroczystym

1 czerwca 2023

## 1 Wstęp

### 1.1 Rozpraszanie fotonów w układzie jedno- i wielowarstwowym.



Rysunek 1: (a) Schemat rozpraszania fotonów w ośrodku o współczynniku załamania  $n$ . Źródło fotonów umieszczone jest na dole (żółty). (b) Schemat rozpraszania fotonów w układzie składającym się z trzech warstw o współczynnikach załamania  $n_1$ ,  $n_2$  i  $n_3$ .

Na rysunku 1(a) pokazany jest schemat rozpraszania fotonów w ośrodku o współczynniku załamania  $n$ . Fotony generowane w źródle (kolor żółty) umieszczonym na dolnym brzegu próbki kierowane są do obszaru czynnego, gdzie podlegają: (a) rozpraszaniu i (b) absorpcji. Pierwszy mechanizm prowadzi do zmiany kierunku fotonu, natomiast drugi do absorpcji, oba mechanizmy mają naturę stochastyczną. Foton po wielokrotnym rozproszeniu może powrócić do dolnego brzegu dając wkład do dyfuzyjnego współczynnika odbicia (reflection), może dotrzeć do górnej granicy dając wkład do współczynnika transmisji (transmission) lub może zostać zaabsorbowany w obszarze czynnym lub na prawym bądź lewym brzegu (absorption). W symulacji MC zbieramy informacje o punktach absorpcji, ponieważ tam deponowana jest energia co może być też głównym celem eksperymentu fizycznego.

Badany układ może mieć budowę warstwową, wówczas foton przechodząc przez granicę ośrodków o różnych współczynnikach załamania podlega prawom optyki geometrycznej - czyli konieczne staje się określenie kąta padania, kąta odbicia, współczynnika odbicia i współczynnika transmisji.

## 1.2 Algorytm MC rozpraszania fotonów

Procesy rozpraszania i absorpcji to procesy losowe więc metoda MC nadaje się idealnie do symulacji tego typu eksperymentów. W wyniku absorpcji foton znika, więc po zaistnieniu takiego procesu w symulacji konieczne stałoby się wygenerowanie wielu nowych pojedynczych fotonów po to aby uzyskać odpowiednio dużą (statystycznie) liczbę przypadków, gdy fotony docierają do różnych oddalonych punktów brzegowych układu. Taki algorytm byłby mało wydajny, więc postępuje się inaczej. Zamiast pojedynczych fotonów, symuluje się trajektorię dla wiązki fotonów. Na starcie nadaje się wiązce wagę  $w = 1$  i w trakcie kolejnych procesów absorpcji waga ulega zmniejszeniu  $w = w - \Delta w$ , aż osiągnie zanedbywalnie małą wartość (np.  $w < 10^{-4}$ ). Symulacja pojedynczej wiązki wraz z punktami, gdzie doszło do absorpcji statystycznie odpowiada symulacji wielu pojedynczych fotonów - mamy więc zysk wydajnościowy.

### 1. Swobodna propagacja wiązki.

Określamy przewidywaną długość ścieżki, po której będzie się poruszać wiązka do kolejnego aktu rozproszenia/absorpcji. Określamy całkowity współczynnik oddziaływania z ośrodkiem

$$\mu_{tot} = \mu_a + \mu_s \quad (1)$$

i prawdopodobieństwo, że na drodze o długości  $s$  nie dojdzie do rozproszenia

$$P\{x < s\} = e^{-\mu_{tot}s} \quad (2)$$

Ponieważ określa ono dystrybucję rozkładu eksponencjalnego  $F(s) = P\{x < s\}$  wobec czego możemy określić losową długość drogi swobodnej wiązki

$$s = -\frac{\ln(U_1)}{\mu_{tot}}, \quad U_1 \sim U(0, 1) \quad (3)$$

### 2. Etap rozpraszania/absorpcji.

Po zakończeniu etapu swobodnej propagacji wiązki dochodzi do rozproszenia. Część wiązki jest absorbowana ( $\Delta w$ ) a pozostała część zmienia kierunek dalszej propagacji. Absorbowaną frakcję wiązki wyliczamy na podstawie relacji pomiędzy współczynnikami absorpcji i rozpraszania

$$\Delta w = \frac{\mu_a}{\mu_a + \mu_s} \quad (4)$$

i usuwamy tę część z wiązki

$$w \leftarrow w - \Delta w \quad (5)$$

Depozycję części wiązki fotonowej (energii) rejestrujemy zazwyczaj w tablicy, aby później móc określić które obszary najsilniej absorbują energię, np. w przypadku 2D

$$\vec{r} = [x, y] \quad (6)$$

$$i = \text{round}\left(\frac{x}{\Delta x}\right) \quad (7)$$

$$j = \text{round}\left(\frac{y}{\Delta z}\right) \quad (8)$$

$$\text{absorption}[i][j] += \text{absorption}[i][j] + \Delta w \quad (9)$$

Zmianę kierunku wykonujemy losując kąt rozproszenia na podstawie współczynnika anizotropii ( $g$ ) materiału. Rozkład prawdopodobieństwa dla kosinusa kąta rozproszenia względem kierunku pierwotnego

$$p(\cos \theta) = \frac{1 - g^2}{2(1 + g^2 - 2g \cos \theta)^{3/2}} \quad (10)$$

pozwała określić jego wartość losową

$$\cos \theta = \begin{cases} \frac{1}{2g} \left[ 1 + g^2 - \left( \frac{1-g^2}{1-g+2gU_1} \right)^2 \right]^{1/2}, & U_1 \sim U(0,1) \iff g \neq 0 \\ 2U_1 - 1, & U_1 \sim U(0,1) \iff g = 0 \end{cases} \quad (11)$$

To jest zmiana kąta biegunowego, natomiast rozkład kąta azymutalnego ( $\phi$ ) jest izotropowy  $\phi \in (0, 2\pi]$  (w przypadku 3D).

### 3. Rozpraszanie na granicy dwóch ośrodków.

Jeśli wiązka pada na granicę dwóch ośrodków należy sprawdzić zestaw warunków

- Całkowite wewnętrzne odbicie

Jeśli

$$n_i > n_s \quad \wedge \quad \alpha_i \geq \alpha_i^{critical} = \arcsin \frac{n_s}{n_i} \quad (12)$$

gdzie:  $n_i$  współczynnik załamania w obszarze w którym znajduje się foton,  $n_s$  to współczynnik załamania obszaru do którego kieruje się foton, to wiązka całkowicie się odbija i pozostaje w obszarze  $n_i$ .

- Odbicie/przejsie przez interfejs

Jeśli wykluczeniu całkowitego odbicia rozważamy możliwość transmisji/częściowego odbicia na interfejsie dwóch materiałów. Znając kąt padania wiązki na interfejs ( $\alpha_i$  liczony względem normalnej do interfejsu) z prawa Snella wyznaczamy kąt wiązki przechodzącej ( $\alpha_s$ )

$$n_i \sin \alpha_i = n_s \sin \alpha_s \quad (13)$$

Jednak w trakcie tego przejścia część wiązki padającej zostaje odbita, wartość współczynnika odbicia wiązki można wyznaczyć z równania Fresnela (tu wzór jest średnią dla dwóch polaryzacji wiązki - wiązka niespolaryzowana jest mieszaniną fotonów o różnych polaryzacjach)

$$R(\alpha_i) = \frac{1}{2} \left[ \frac{\sin^2(\alpha_i - \alpha_s)}{\sin^2(\alpha_i + \alpha_s)} + \frac{\tan^2(\alpha_i - \alpha_s)}{\tan^2(\alpha_i + \alpha_s)} \right] \quad (14)$$

Ponieważ wartość współczynnika odbicia zwiera się w  $R(\alpha) \in [0, 1]$  więc w symulacji MC możliwość przejścia/odbicia obsługujemy w sposób losowy tj. pozwalając całej wiązce przejść przez interfejs lub całkowicie się odbić (po wykonaniu symulacji wielu takich wiązek wyniki się uśrednią do wartości zbliżonych do oczekiwanych).

$$U_1 \sim U(0,1) \quad \longrightarrow \quad \begin{cases} U_1 \leq R(\alpha_i) & \text{- odbicie całej wiązki} \\ U_1 > R(\alpha_i) & \text{- cała wiązka przechodzi} \end{cases} \quad (15)$$

- Kasowanie wiązki - schemat ruletki.

Jeśli wiązka będzie się rozpraszać, ale nie natrafi na brzeg (gdzie jest absorbowana w całości) to podczas kolejnych rozprożeń jej waga ulega pomniejszeniu, ale zawsze będzie  $w > 0$ . Jednak ewolucja wiązek o wagach np.  $w < 10^{-4}$  nie ma sensu, ponieważ nic nie wnoszą one

do rozwiązania bo są zbyt małe, a tylko tracimy czas zezwalając na ich dalszą propagację. Problem ten można rozwiązać statystycznie. Jeśli waga spadnie do wartości  $w < 10^{-4}$  to dokonujemy kasowania lub wzmocnienia wiązki określając wcześniej prawdopodobieństwo wzmocnienia ( $p_{amp}$ )

$$w < 10^{-4} ? \quad \rightarrow \quad \begin{cases} p_{amp} = 0.1 \\ U_1 \sim U(0, 1) \end{cases} \quad \rightarrow \quad \begin{cases} U_1 \leq p_{amp} & \rightarrow w \leftarrow w \frac{1}{p_{amp}} \text{ wzmocnienie} \\ U_1 > p_{amp} & \rightarrow w = 0 \text{ kasowanie} \end{cases} \quad (16)$$

## 2 Implementacja algorytmu MC - klasa PHOTON\_DIFFUSION\_2D

Mimo iż algorytm MC rozpraszania jest dość prosty, jego implementacja jest czasochłonna, dlatego na laboratorium wykorzystamy klasę **PHOTON\_DIFFUSION\_2D** zawartą w pliku **photon\_diffusion.cpp**

Klasa zawiera zmienne

- integer **write\_all\_paths** = 0/1 (default=0):  
1-zapis wszystkich ścieżek do pliku "write\_all\_paths.dat",  
0-brak zapisu
- integer **write\_source\_detection\_paths** = 0/1 (default=0):  
1-zapis do pliku "source\_detection\_paths.dat" ścieżek kończących się w obszarze detekcji,  
0-brak zapisu
- integer **nlayers** - liczba warstw (default=0, max=19)
- integer **nx, ny** - zakres liczby węzłów w kierunku "x" i "y" w tablicy **absorption[0:nx][0:ny]** (default: nx=0,ny=0)
- double **xmax** - rozmiar obszaru badanego w kierunku "x" (default=0)
- double **ymax** - rozmiar obszaru badanego w kierunku "y" (default=0) jest wyliczana w programie na podstawie grubości warstw
- double **dx,dy** - odległość między węzłami w tablicy **absorption** wyznaczone w programie ( $dx = xmax/nx$ ,  $dy = ymax/ny$ )
- double **p\_min** - parametr określający wzmocnienie wagi wiązki fotonowej w algorytmie ruletki, gdy  $w < w_{min}$  (default=0.1)
- double **w\_min** - wartość progowa wagi wiązki **w** (default w\_min=0.0001)
- double **x\_source** - położenie środka źródła światła na dolnym brzegu
- double **dx\_source** - szerokość źródła światła
- double **x\_detect** - położenie detektora na tym samym brzegu co źródło
- double **dx\_detect** - szerokość detektora
- double **rx0,ry0** - początkowy kierunek (wersor) wiązki (default = [rx,ry]=[1,0])

Tablice

- double **layers\_data**[0:20][0:20] - przechowuje informacje o parametrach każdej z warstw, parametry wpisujemy od wiersza 1 do **nlayers** (wiersze 0 i nlayers+1 to obszary pod i nad obiektem badanym - powietrze)  
zawartość kolumn:  
**layers\_data**[m][0]= $\mu_a^{(m)}$  - współczynnik absorpcji warstwy o indeksie m  
**layers\_data**[m][1]= $\mu_s^{(m)}$  - współczynnik rozpraszania warstwy o indeksie m  
**layers\_data**[m][2]= $d^{(m)}$  - grubość warstwy o indeksie m  
**layers\_data**[m][3]= $g^{(m)}$  - współczynnik anizotropii warstwy o indeksie m  
**layers\_data**[m][4]= $n^{(m)}$  - współczynnik załamania warstwy o indeksie m
- double **absorption**[0:nx][0:ny] - zawiera rozkład przestrzenny współczynnika absorpcji (odpowiednik ilości energii zdeponowanej przez wiązkę w danym punkcie)
- double **reflectance**[0:nx] - zawiera rozkład przestrzenny współczynnika odbicia na wejściu układu ( $y = 0$ )
- double **transmittance**[0:nx] - zawiera rozkład przestrzenny współczynnika transmisji na wyjściu układu ( $y = y_{max}$ )

Tablice absorpcji, odbicia i transmisji spełniają warunek unormowania

$$A = \sum_{i=0}^{n_x} \sum_{j=0}^{n_y} absorption[i][j]$$

$$R = \sum_{i=0}^{n_x} reflectance[i]$$

$$T = \sum_{i=0}^{n_x} transmittance[i]$$

$$\frac{A + R + T}{N} = 1$$

gdzie  $N$  to liczba wiązek fotonowych użytych w symulacji. Ze względu na zastosowanie algorytmu ruletki w celu podniesienia wydajności symulacji MC, suma może się nieznacznie różnić od 1.

Funkcje

- void **init**() - funkcję należy wywołać po zainicjowaniu wartości parametrów symulacji
- void **single\_path**() - funkcja wykonuje symulację dla pojedynczej wiązki, wyniki akumuluje w tablicach: absorption, reflectance, transmittance oraz może wygenerowaną ścieżkę dodać do plików plików "all\_paths.dat" i "source\_detection\_paths.dat"

Przykład wykonania symulacji

```
PHOTON_DIFFUSION_2D ob;
ob.xmax=0.2;
ob.x_source=0.1;
ob.dx_source=0.0;
ob.x_detect=0.15;
ob.dx_detect=0.01;
ob.nx=100;
```

```

ob.ny=100;

ob.rx0=0.0;
ob.ry0=1.0;

ob.nlayers=2;

ob.layers_data[1][0]=1.; //absorption
ob.layers_data[1][1]=10.; //scattering
ob.layers_data[1][2]=0.02; //width
ob.layers_data[1][3]=0.75; //g_anizo
ob.layers_data[1][4]=1.0; //n_refraction

ob.layers_data[2][0]=1.; //absorption
ob.layers_data[2][1]=10.; //scattering
ob.layers_data[2][2]=0.02; //width
ob.layers_data[2][3]=0.75; //g_anizo
ob.layers_data[2][4]=1.0; //n_refraction

ob.init();

int N=100000; - liczba wiązek fotonowych
ob.write_all_paths=0;
ob.write_source_detection_paths=0;

for(int k=0;k<N;k++){
    ob.single_path();
}

```

### 3 Zadania do wykonania

Przyjmujemy podstawowe parametry symulacji, ewentualne zmiany poniżej będą dotyczyć jedynie wybranych parametrów i tylko je zmieniamy:

```

nlayers=3,
x_max=0.2,
x_source=0.1,
dx_source=0.0,
x_detect=0.15,
dx_detect=0.01,
nx=ny=100,
rx=0.0,
ry=1.0
N=200000 -liczba wiązek fotonowych

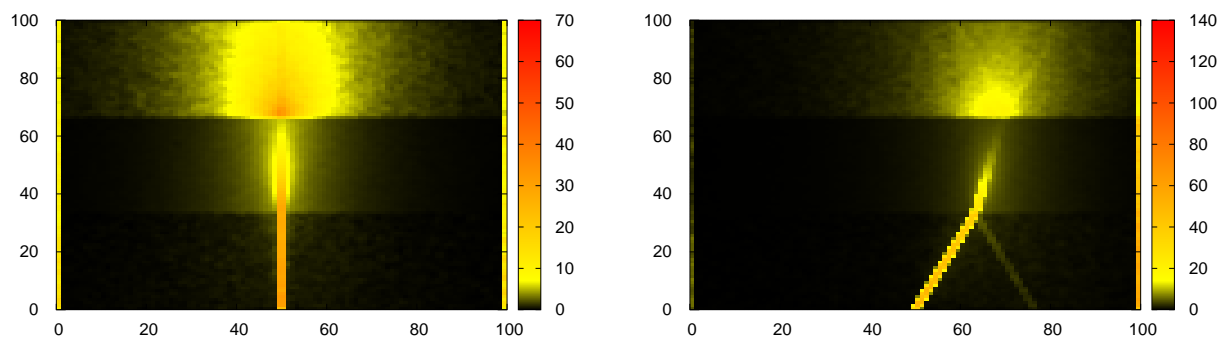
```

| $m$ | $\mu_a[1/cm]$ | $\mu_s[1/cm]$ | $d[cm]$ | $g[-]$ | $n[-]$ |
|-----|---------------|---------------|---------|--------|--------|
| 1   | 1             | 10            | 0.02    | 0.75   | 1.3    |
| 2   | 1             | 190           | 0.02    | 0.075  | 1.0    |
| 3   | 10            | 90            | 0.02    | 0.95   | 1.0    |

1. Napisać program wykorzystujący klasę **PHOTON\_DIFFUSION\_2\_D**
2. Przeprowadzić symulacje dla przypadku z wewnętrznym odbiciem wiązki
  - zmienionych parametrów:  $rx = 0.8$ ,  $ry = 0.6$ ,  $n^{(2)} = 1.5$
  - zmienionych parametrów:  $rx = 0.8$ ,  $ry = 0.6$ ,  $n^{(2)} = 2.5$
  - zmienionych parametrów:  $rx = 0.8$ ,  $ry = 0.6$ ,  $n^{(1)} = 1.0$ ,  $n^{(2)} = 1.5$
  - zmienionych parametrów:  $rx = 0.8$ ,  $ry = 0.6$ ,  $n^{(1)} = 1.0$ ,  $n^{(2)} = 1.5$ ,  $\mu_s^{(2)} = 10$
3. Przeprowadzić symulacje dla wiązki padającej prostopadle na obiekt
  - zestawu podstawowego parametrów (takich jak w tabelce)
  - zmienionych parametrów:  $n^{(1)} = 1.0$ ,  $\mu_a^{(2)} = 10$ ,  $\mu_s^{(2)} = 210$ ,  $n^{(2)} = 1.5$
  - zmienionych parametrów:  $n^{(1)} = 1.0$ ,  $\mu_a^{(2)} = 1$ ,  $\mu_s^{(2)} = 210$ ,  $n^{(2)} = 1.5$
  - zmienionych parametrów:  $n^{(1)} = 1.0$ ,  $\mu_a^{(2)} = 10$ ,  $\mu_s^{(2)} = 210$ ,  $n^{(2)} = 1.5$ ,  $g^{(2)} = 0.75$

Dla każdego przypadku narysować rozkłady przestrzenne absorpcji, określić zakumulowane wartości współczynników: absorpcji (A), odbicia (R) i transmisji (T). W raporcie przeprowadzić analizę uzyskanych wyników.

## 4 Przykładowe wyniki



Rysunek 2: Przykładowe wyniki propagacji wiązki fotonowej w materiale trzywarstwowym. Jasnożółty kolor na lewym i prawym brzegu oznacza obszar całkowitej absorpcji docierających tam fotonów.