

Politechnika Krakowska  
Im. Tadeusza Kościuszki  
Wydział Fizyki, Matematyki i Informatyki  
Kierunek Fizyka Techniczna



IGOR SKIBA

**Symulacja Propagowania Neutronów Przez Materię Metodą  
Monte Carlo**

Praca Inżynierska

Promotor:  
Prof. dr hab. Tadeusz Lesiak

Kraków, Styczeń 2016



*Chciałbym wyrazić najszczerze podziękowania  
mojemu promotorowi prof. Tadeuszowi Lesiakowi  
za jego czas, cierpliwość i życzliwość podczas  
przygotowywania tej pracy inżynierskiej.*  
*Autor*

## **STRESZCZENIE**

Celem pracy jest opracowanie oraz stworzenie dwóch symulacji propagacji neutronów przez dowolne medium. Praca zawiera podstawowe informacje dotyczące metody Monte Carlo oraz rozważania teoretyczne niezbędne do wykonania symulacji.

## Spis Treści

1. Wprowadzenie .....	7
1.1. Temat i cel pracy .....	7
1.1.2 Zakres pracy .....	7
<b>I. Część studialna</b>	
2. Metoda Monte Carlo .....	8
2.1. Definicja .....	8
2.2. Historia .....	8
2.3. Zastosowanie .....	9
3. Transmisja neutronów przez nieskończoną płytę .....	10
3.1. Wprowadzenie .....	10
3.2. Opis symulacji .....	10
4. Ucieczka neutronów z nieskończonej płyty .....	13
4.1. Wprowadzenie .....	13
4.2. Opis symulacji .....	13
<b>II. Część projektowa</b>	
5. Transmisja neutronów przez nieskończoną płytę .....	14
5.1. Algorytm .....	14
5.2. Opis działania programu .....	15
5.3. Rezultaty symulacji .....	17
6. Ucieczka neutronów z nieskończonej płyty .....	20
6.1. Procedura .....	20
6.2. Opis działania programu .....	20
6.3. Rezultaty symulacji .....	22
7. Kody źródłowe .....	24
7.1. Transmisja neutronów przez nieskończoną płytę .....	24
7.2. Ucieczka neutronów z nieskończonej płyty .....	26
8. Wnioski i podsumowanie .....	28
9. Bibliografia .....	29



# **1. WPROWADZENIE**

## **1.1. TEMAT I CEL PRACY**

Praca ma na celu zapoznanie się z metodą Monte Carlo oraz wdrożenie jej do stworzenia dwóch symulacji propagacji neutronów przez materię. Pierwsza symulacja dotyczy przypadku kiedy neutron zderza się z płytą, a w drugiej symulacji kiedy neutron powstaje wewnątrz płyty. Praca ma również na celu nabycie praktycznych umiejętności w programowaniu komputerowym. Język C++ został wybrany jako język programowania obu symulacji.

## **1.2. ZAKRES PRACY**

Zakres pracy został podzielony na część studialną i część projektową. Część studialna zawiera podstawowe informacje o metodzie Monte Carlo oraz teoretyczne informacje potrzebne do stworzenia symulacji. W części projektowej zostały zawarte algorytmy oraz procedury, które zostały użyte do implementacji kodu. Następnie wygenerowano wykresy i na ich podstawie wyciągnięto wnioski dotyczące natury symulowanych zjawisk.

# I. CZĘŚĆ STUDIALNA

## 2. METODA MONTE CARLO

### 2.1. DEFINICJA

Metoda Monte Carlo umożliwia rozwiązywanie problemów obliczeniowych poprzez generowanie odpowiedniej ilości liczb losowych. Metody tej używa się najczęściej do numerycznych rozwiązań problemów, które są zbyt skomplikowane do rozwiązania analitycznego.

### 2.2. HISTORIA

Metoda Monte Carlo została po raz pierwszy zastosowana do obliczeń na dużą skalę przez polskiego matematyka Stanisława Marcina Ulmana w późnych latach czterdziestych dwudziestego wieku, podczas prac nad bronią jądrową (Project Manhattan) w Los Alamos National Laboratory. Pod kierownictwem profesora Johna von Neumanna, Ulam zaprojektował symulację z wykorzystaniem liczb losowych, służącą do badania propagacji neutronów w różnych ośrodkach. Warto zatem podkreślić, iż pierwsze zastosowanie metody Monte Carlo dotyczyło tematyki dyskutowanej w tej pracy inżynierskiej. Do obliczeń wykorzystano komputer ENIAC (pierwszy elektroniczny, numeryczny integrator i komputer – do 1975 roku uważany za najszybszy komputer na świecie).

Jako że działalność potrzebowała specjalnego kryptonimu ze względu na jej sekretną treść, Ulam nadał jej nazwę Monte Carlo. Odnosiła się ona do kasyna Monte Carlo w Monako gdzie wujek Ulmana często uprawiał hazard. Wynalezioną metodę oraz zaproponowaną nazwę łączyła jedna cecha – liczby losowe.



Rys 2.1. Stanisław Ulam. [M1]

Wczesne symulacje Monte Carlo były bardzo ograniczone ze względu na słabe możliwości obliczeniowe dostępne w ówczesnym okresie. Jednak nie stanęło to na przeszkodzie do stworzenia pierwszej broni jądrowej. W latach pięćdziesiątych symulacje Monte Carlo zostały użyte do stworzenia bomby jądrowej i stawały się coraz bardziej popularne



w wielu dziedzinach nauki. Z biegiem czasu świat komputeryzował się i metoda Monte Carlo znalazła coraz większe zastosowanie w najróżniejszych dziedzinach życia.

### 2.3. ZASTOSOWANIE

Współcześnie, metoda Monte Carlo znalazła zastosowanie w niemal każdej dziedzinie nauki, niektóre z nich to:

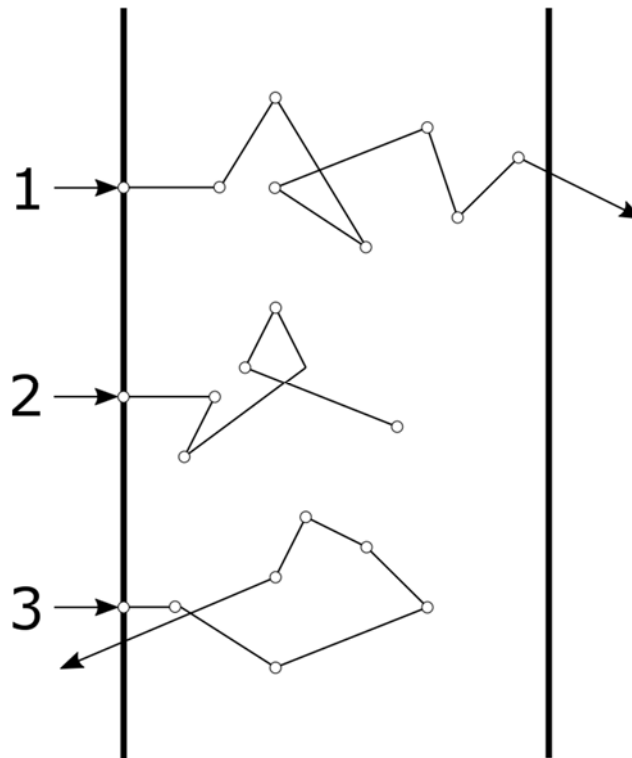
- Nauki fizyczne – w fizyce komputerowej, chemii fizycznej oraz pokrewnych dziedzinach. Dzięki tej metodzie można zaprojektować m.in. osłony termiczne i osłony aerodynamiczne oraz można jej użyć podczas modelowania transportu promieniowania.
- Inżynieria – jest jednym z najpowszechniejszych zastosowań użycia metody Monte Carlo, np. w mikroelektronice (analiza układów scalonych), dynamice płynów (analiza dynamiki gazów rozrzedzonych) oraz automatyce i robotyce (pozycjonowanie robotów).
- Biologia obliczeniowa – metodę Monte Carlo można użyć do studiowania biologicznych układów takich jak membrany, genomy albo proteiny. Dzięki takim symulacjom, można obserwować środowisko poszczególnych molekuł w celu zbadania czy zachodzą określone reakcje chemiczne.
- Grafika komputerowa – w tak zwanym „Path Tracing” która tworzy fotorealistyczne obrazy obiektów 3-wymiarowych. Renderowanie to jest uważane za jedno z najbardziej dokładnych współcześnie.
- Sztuczna inteligencja w grach – „Monte-Carlo tree search” jest zastosowaniem omawianej metody do tworzenia sztucznej inteligencji w grach komputerowych. Kładzie nacisk na analizie najbardziej prawdopodobnych ruchów. Przykłady gier to np. „Pac-Man” i „Poker”.
- Wizualizacje - do tworzenia fotorealistycznych zdjęć oraz obiektów 3D w grach komputerowych, architekturze i budownictwie oraz podczas tworzenia efektów specjalnych w filmach.
- Poszukiwanie i ratunek „Search and Rescue” – program komputerowy „SAROPS – Search and Rescue Optimal Planning System” jest używany do określenia prawdopodobieństwa lokalizacji statków które zaginęły jak również do optymalizacji środków ratowniczych.
- Finanse - do szacowania inwestycji w projektach, ustalania cen opcji oraz analizy ryzyka finansowego.

### 3. TRANSMISJA NEUTRONÓW PRZEZ NIESKOŃCZONĄ PŁYTĘ

#### 3.1. WPROWADZENIE

Rozważamy strumień neutronów o energii  $E_0$ , skierowany na jednorodną nieskończoną płytę o grubości  $0 \leq x \leq h$ . Kąt zderzenia neutronów o płytę jest równy  $90^\circ$  (względem powierzchni płyty). W wyniku zderzenia neutronów z atomami płyty, neutrony mogą zostać elastycznie rozproszone albo zaabsorbowane. Dla uproszczenia zakłada się, że energia neutronu nie zmienia się podczas rozproszenia oraz że wszystkie kierunki rozpraszania się neutronów są równo prawdopodobne. Powstają zatem trzy możliwości oddziaływania neutronów w płycie:

1. Transmisja,
2. Absorpcja,
3. Odbicie.



Rys. 3.1. Ścieżki neutronów przy kontakcie z płytą.

#### 3.2. OPIS SYMULACJI

Interakcja neutronów z materią jest scharakteryzowana przez dwie stałe  $\Sigma_a$  i  $\Sigma_s$ .  $\Sigma_a$  jest to przekrój absorpcji natomiast  $\Sigma_s$  jest przekrojem rozproszenia. Po zsumowaniu tych dwóch stałych otrzymujemy przekrój całkowity  $\Sigma$ .

$$\Sigma = \Sigma_a + \Sigma_s.$$

Tym samym  $\frac{\Sigma_a}{\Sigma}$  jest prawdopodobieństwem absorpcji natomiast  $\frac{\Sigma_s}{\Sigma}$  jest prawdopodobieństwem rozproszenia.

Typowa odległość  $\lambda$  pomiędzy dwoma kolejnymi kolizjami neutronów nazywana jest średnią drogą swobodną (free path length). Jest to losowa zmienna która może przyjąć dowolną pozytywną wartość z prawdopodobieństwem, zgodnie z rozkładem:

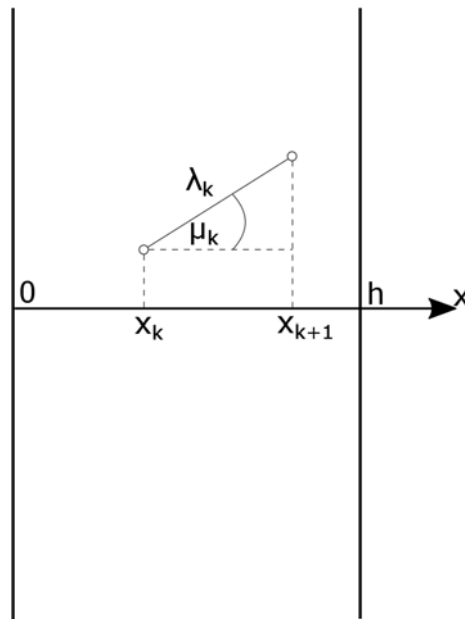
$$p(x) = \Sigma e^{-x\Sigma}, \quad 0 < x < \infty.$$

Założmy, że neutron został poddany  $k$ -temu rozproszeniu wewnątrz płyty w punkcie o współrzędnej  $x_k$ , po czym zaczął poruszać się w kierunku  $\mu_k$ . Średnia droga swobodna jest wtedy obliczona ze wzoru:

$$\lambda_k = -\frac{\ln(\gamma)}{\Sigma},$$

gdzie  $\gamma$  oznacza zmienną losową o rozkładzie jednorodnym. Odcięta odpowiadająca następnemu zdarzeniu wyraża się wówczas wzorem:

$$x_{k+1} = x_k + \lambda_k \mu_k.$$



Rys. 3.2. Schemat kolizji neutronu wewnątrz materii.

Neutron, który dokonał transmisji spełnia następujący warunek:

$$x_{k+1} \geq h.$$

Jeżeli neutron nie spełnia tego warunku, sprawdza się warunek na odbicie:

$$x_{k+1} \leq 0.$$

Jeżeli ten warunek również nie zostanie spełniony, oznacza to że neutron został poddany  $(k + 1)$  kolizji w płycie i sprawdza się wówczas warunek na absorpcję:

$$\gamma < \frac{\Sigma_a}{\Sigma}.$$

Jeżeli ten warunek nie jest spełniony, zakładamy że neutron został poddany rozproszeniu w punkcie odciętej  $x_{k+1}$ .

Następnie znajdujemy nowy kierunek ruchu zgodnie z formułą:

$$\mu_{k+1} = 2\gamma - 1,$$

i powtarzamy cały przebieg procesu jeszcze raz (z nową wartością  $\gamma$ ).

Tym samym w pojedynczej iteracji programu konieczne jest wygenerowanie trzech liczb losowych  $\gamma$  z rozkładu jednorodnego:

- pierwsza wartość do określenia  $\lambda_k = -\frac{\ln(\gamma)}{\Sigma}$ ,
- druga wartość do sprawdzenia warunku na absorpcję  $\gamma < \frac{\Sigma_a}{\Sigma}$ ,
- trzecia wartość do wyznaczenia nowego kierunku neutronu  $\mu_{k+1} = 2\gamma - 1$ .

Początkowe współrzędne każdej trajektorii to:

$$x_0 = 0, \mu_0 = 1.$$

W wyniku symulacji z zastosowaniem  $N$  trajektorii, znajdujemy  $N^+$  neutronów poddanych transmisji,  $N^-$  odbitych i  $N^0$  zaabsorwowanych. Aby obliczyć prawdopodobieństwo tych zdarzeń należy podzielić otrzymane liczby neutronów przez całkowitą ilość neutronów.

$$\begin{aligned} p^+ &\approx \frac{N^+}{N} & - & \text{prawdopodobieństwo transmisji,} \\ p^- &\approx \frac{N^-}{N} & - & \text{prawdopodobieństwo odbicia,} \\ p^0 &\approx \frac{N^0}{N} & - & \text{prawdopodobieństwo absorpcji.} \end{aligned}$$

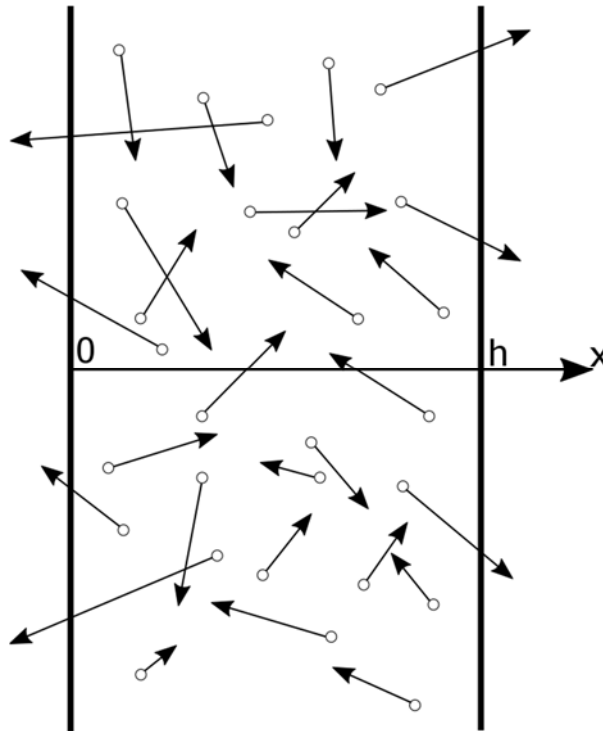
## 4. UCIECZKA NEUTRONÓW Z NIESKOŃCZONEJ PŁYTY

### 4.1. WPROWADZENIE

Celem symulacji jest określenie liczby neutronów, które albo opuszczają medium albo są przez nie absorbowane. Zakładamy, że neutron powstaje wewnątrz medium.

### 4.2. OPIS SYMULACJI

Celem symulacji jest określenie prawdopodobieństwa ucieczki  $N_{esc}$  cząsteczek powstałych w sposób przypadkowy, zgodnie z rozkładem jednorodnym, w jakimś czysto absorbującym ośrodku dla którego  $\Sigma = \Sigma_a$ .



Rys. 4.1. Ścieżki neutronów wewnątrz płyty.

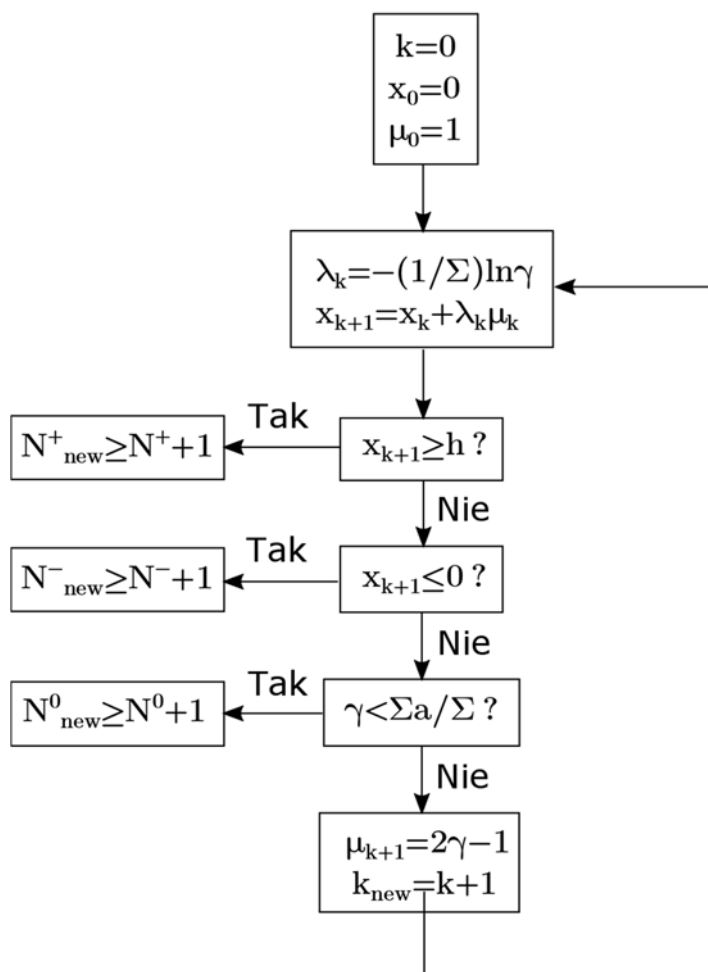
Rozważamy przypadek w którym cząstki są emitowane jednorodnie i izotropowo przez nieskończoną, absorbującą płytę o grubości  $h$ . Ponieważ płyta jest nieskończona w kierunku  $y$  i  $z$ , ucieczka z ośrodka możliwa jest tylko w kierunku  $x$ . Dla izotropowej emisji, cosinus kąta emisji  $\theta$  zgodny z dodatnią polarną osią  $x$  jest jednorodnie rozłożony z przedziału  $[-1,1]$ . Azymut kąta emisji nie wpływa na prawdopodobieństwo ucieczki, więc tylko kąt polarny  $\theta$  musi zostać zasymulowany. Wyżej opisaną symulację powtórzono  $N$  razy w celu określenia liczby cząstek które uciekną  $N_{esc}$ .

## II. CZĘŚĆ PROJEKTOWA

### 5. TRANSMISJA NEUTRONÓW PRZEZ NIESKOŃCZONĄ PŁYTĘ

Celem programu jest określenie liczby neutronów, które zostają poddane transmisji, odbiciu i absorpcji. Neutrony padają prostopadłe na płytę i mogą „wędrować” w jej wnętrzu.

#### 5.1. ALGORYTM



Rys. 5.1. Algorytm propagacji neutronów.

## 5.2. OPIS DZIAŁANIA PROGRAMU

Jako medium został wybrany najbardziej powszechny izotop żelaza Fe-56 (stanowi 91.754% wszystkich izotopów tego pierwiastka).

Tab. 5.1. Przekroje czynne na oddziaływania neutronów w wybranych ośrodkach. [M2]

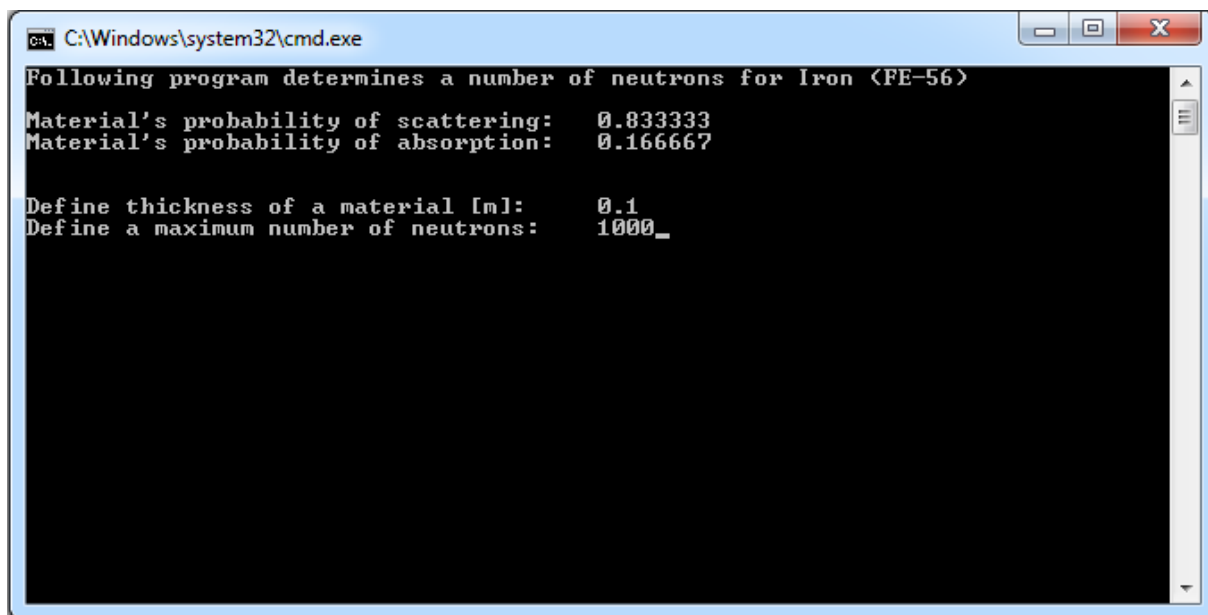
		Przekrój czynny neutronów termicznych (barn)			Przekrój czynny neutronów prędkich (barn)		
		Rozproszenie	Absorpcja	Rozszczepienie	Rozproszenie	Absorpcja	Rozszczepienie
Moderator	H-1	20	0.2	-	4	0.00004	-
	H-2	4	0.0003	-	3	0.000007	-
	C (nat)	5	0.002	-	2	0.00001	-
Materiały konstrukcyjne, inne	Zr-90	5	0.006	-	5	0.006	-
	<b>Fe-56</b>	<b>10</b>	<b>2</b>	-	20	0.003	-
	Cr-52	3	0.5	-	3	0.002	-
	Ni-58	20	3	-	3	0.008	-
	O-16	4	0.0001	-	3	0.00000003	-
Absorber	B-10	2	200	-	2	0.4	-
	Cd-113	100	30,000	-	4	0.05	-
	Xe-135	400,000	2,000,000	-	5	0.0008	-
	In-115	2	100	-	4	0.02	-
Paliwo	U-235	10	99	583	4	0.09	1
	U-238	9	2	0.00002	5	0.07	0.3
	Pu-239	8	269	748	5	0.05	2

Dla żelaza, przekrój czynny absorpcji  $\Sigma_a$  wynosi 2 barny, przekrój rozproszenia  $\Sigma_s$  wynosi 10 barnów, natomiast przekrój całkowity  $\Sigma$  wynosi 12 barnów. Z tych informacji możemy obliczyć:

$$\text{Prawdopodobieństwo absorpcji Fe-56} \quad \frac{\Sigma_a}{\Sigma} = 0.1666667,$$

$$\text{Prawdopodobieństwo rozproszenia Fe-56} \quad \frac{\Sigma_s}{\Sigma} = 0.83333333.$$

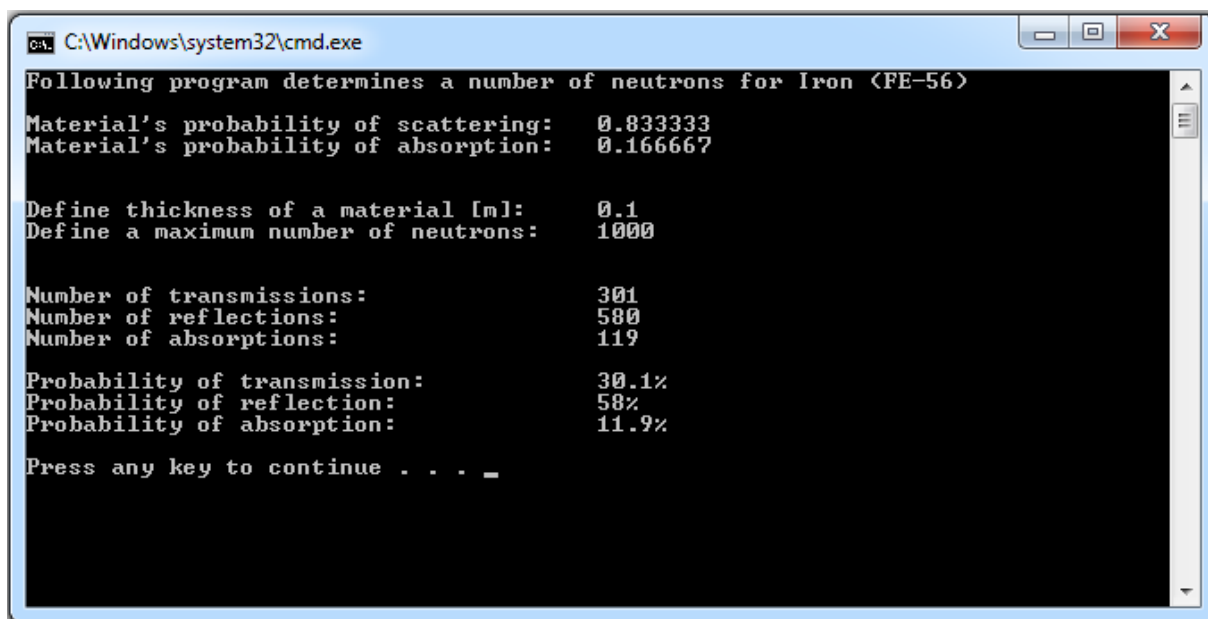
Program został stworzony tak, aby użytkownik mógł ręcznie podać liczbę neutronów  $N$  oraz grubość płyty  $h$ . Po wpisaniu tych dwóch danych, program oblicza liczbę neutronów które zostały poddane transmisji, odbiciu i absorpcji oraz prawdopodobieństwo transmisji, odbicia i absorpcji.



```
C:\Windows\system32\cmd.exe
Following program determines a number of neutrons for Iron <FE-56>
Material's probability of scattering: 0.833333
Material's probability of absorption: 0.166667

Define thickness of a material [m]: 0.1
Define a maximum number of neutrons: 1000_
```

Rys. 5.2. Wprowadzanie danych do programu.



```
C:\Windows\system32\cmd.exe
Following program determines a number of neutrons for Iron <FE-56>
Material's probability of scattering: 0.833333
Material's probability of absorption: 0.166667

Define thickness of a material [m]: 0.1
Define a maximum number of neutrons: 1000

Number of transmissions: 301
Number of reflections: 580
Number of absorptions: 119

Probability of transmission: 30.1%
Probability of reflection: 58%
Probability of absorption: 11.9%

Press any key to continue . . . _
```

Rys. 5.3. Wyniki otrzymane po wprowadzeniu danych.



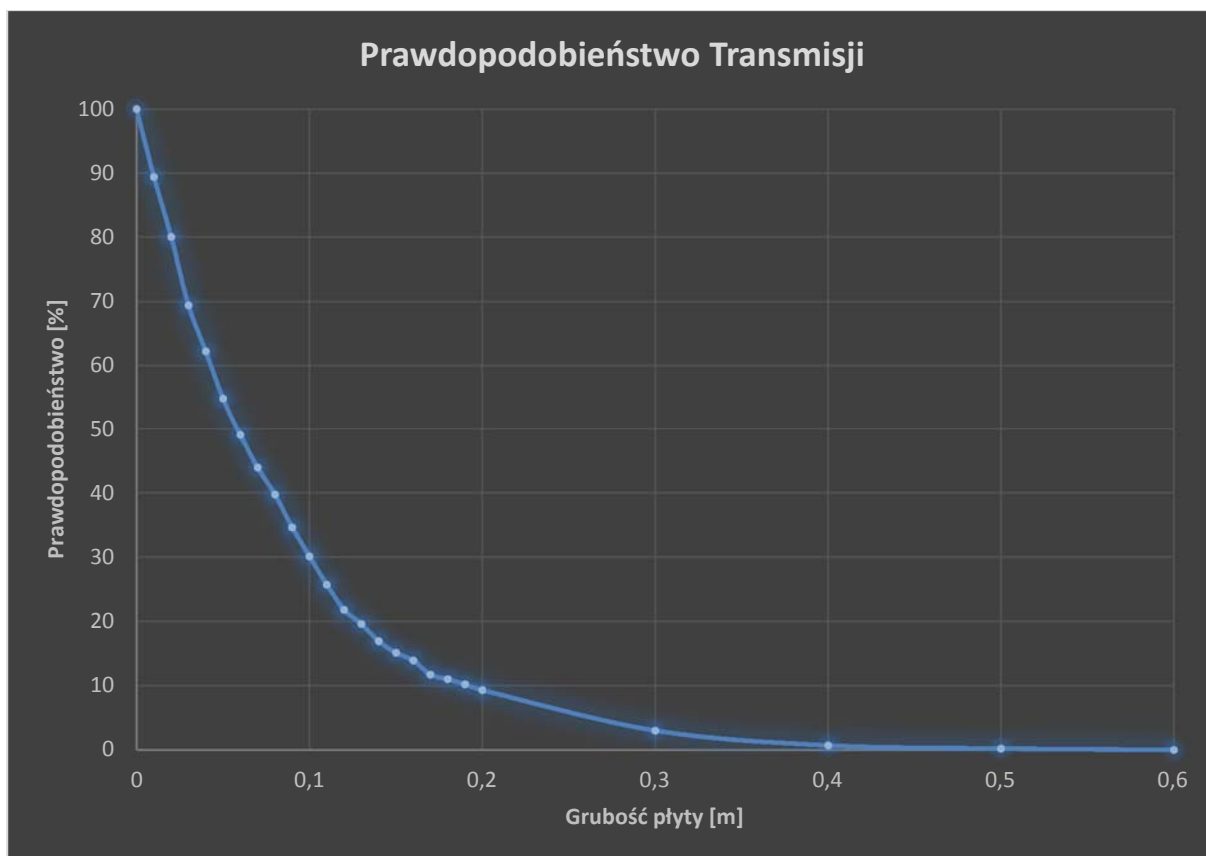
### 5.3. REZULTATY SYMULACJI

Symulacja została przeprowadzona na 1000 neutronów. Propagacja neutronów zmieniała się najbardziej w przedziale  $(0 - 0.2)m$ . Przy grubości  $0.6m$ , transmisja neutronów wynosiła 0%.

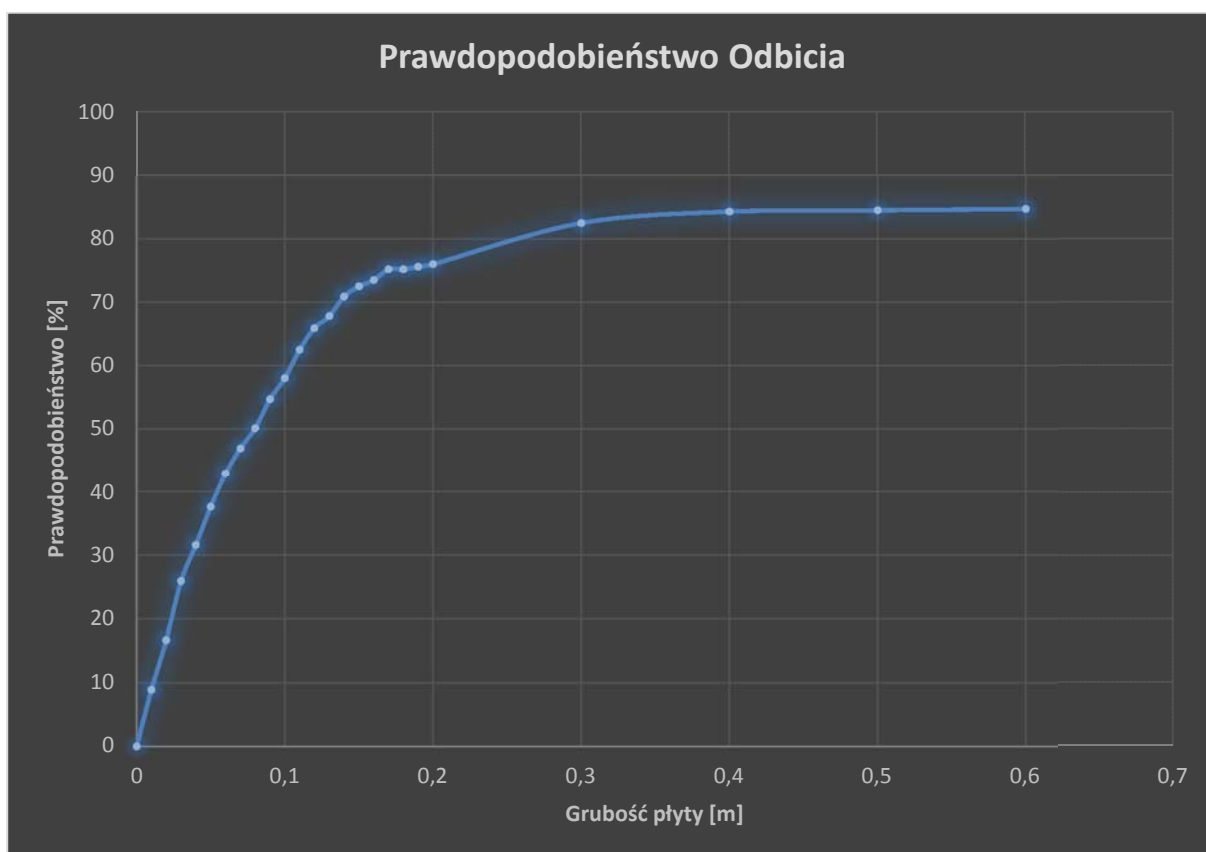
Tab nr 5.2. Prawdopodobieństwa transmisji, odbicia oraz absorpcji neutronów.

Grubość Płyty [m]	Prawdopodobieństwo [%]		
	Transmisji	Odbicia	Absorpcji
0	100	0	0
0,01	89,4	8,9	1,7
0,02	80	16,7	3,3
0,03	69,4	25,9	4,7
0,04	62,2	31,6	6,2
0,05	54,8	37,6	7,6
0,06	49	42,9	8,1
0,07	43,9	46,9	9,2
0,08	39,7	50,1	10,2
0,09	34,6	54,7	10,7
0,1	30,1	58	11,9
0,11	25,7	62,5	11,8
0,12	21,8	65,9	12,3
0,13	19,6	67,8	12,6
0,14	16,9	70,9	12,2
0,15	15,1	72,5	12,4
0,16	13,9	73,5	12,6
0,17	11,7	75,2	13,1
0,18	11	75,2	13,8
0,19	10,2	75,6	14,2
0,2	9,3	76	14,7
0,3	3	82,5	14,5
0,4	0,7	84,3	15
0,5	0,2	84,5	14,3
0,6	0	84,7	15,3

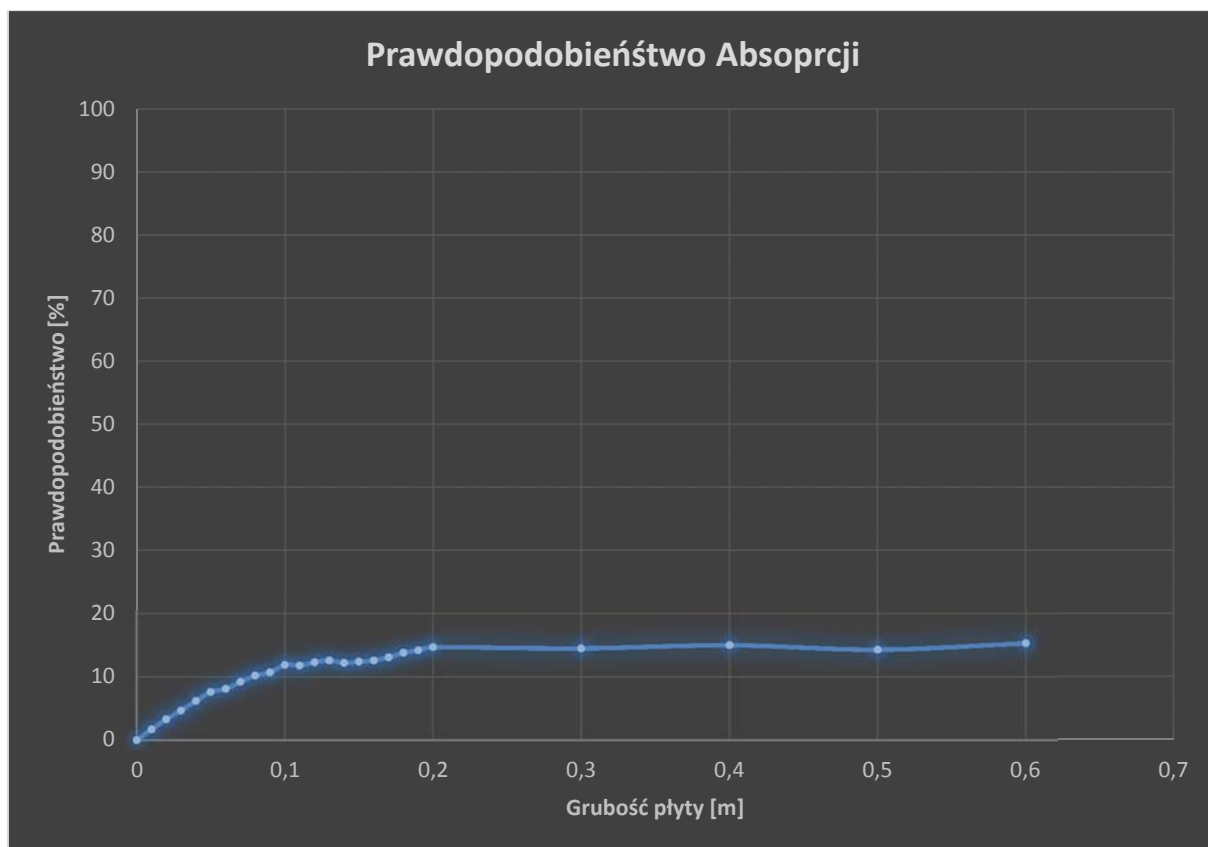
Na podstawie wyników przedstawionych w tabeli 5.2, skonstruowano wykresy prawdopodobieństwa transmisji, odbicia oraz absorpcji w zależności od grubości płyty (rysunki 5.3 – 5.5).



Rys. 5.3. Prawdopodobieństwo transmisji.



Rys. 5.4. Prawdopodobieństwo odbicia.



Rys. 5.5. Prawdopodobieństwo absorpcji.

## 6. UCIECZKA NEUTRONÓW Z NIESKOŃCZONEJ PŁYTY

W odróżnieniu od wcześniejszego programu, w tej symulacji neutrony powstają wewnątrz materii. Celem programu jest określenie liczby neutronów które wydostają się z płyty lub zostają przez nią zaabsorwowane.

### 6.1. PROCEDURA

1. Wybrany punkt emisyjny jednorodnie w  $(0, h)$  jako

$$x_i = h\gamma_i.$$

2. Wybrany kierunek emisyjny  $\mu = \cos\theta$  jako

$$\mu = 2\gamma_i - 1.$$

3. Obliczony odległość  $d$  do powierzchni

- 3.1 Jeżeli  $\mu_i > 0$  wtedy

$$d_i = \frac{h - x_i}{\mu_i}.$$

- 3.2 Jeżeli nie wtedy

$$d_i = \frac{x_i}{\mu_i}.$$

4. Wygenerowana losowo długość ścieżki  $\lambda$  dla pierwszej kolizji cząstki (absorpcji) w materiale z rozkładu  $f(\lambda) = \mu e^{-\mu\lambda}$  jako

$$\lambda_i = -\frac{\ln\gamma_i}{\Sigma}.$$

5. Warunek

- 5.1 Jeżeli  $\lambda_i < d_i$  wtedy cząstka jest zaabsorwowana i

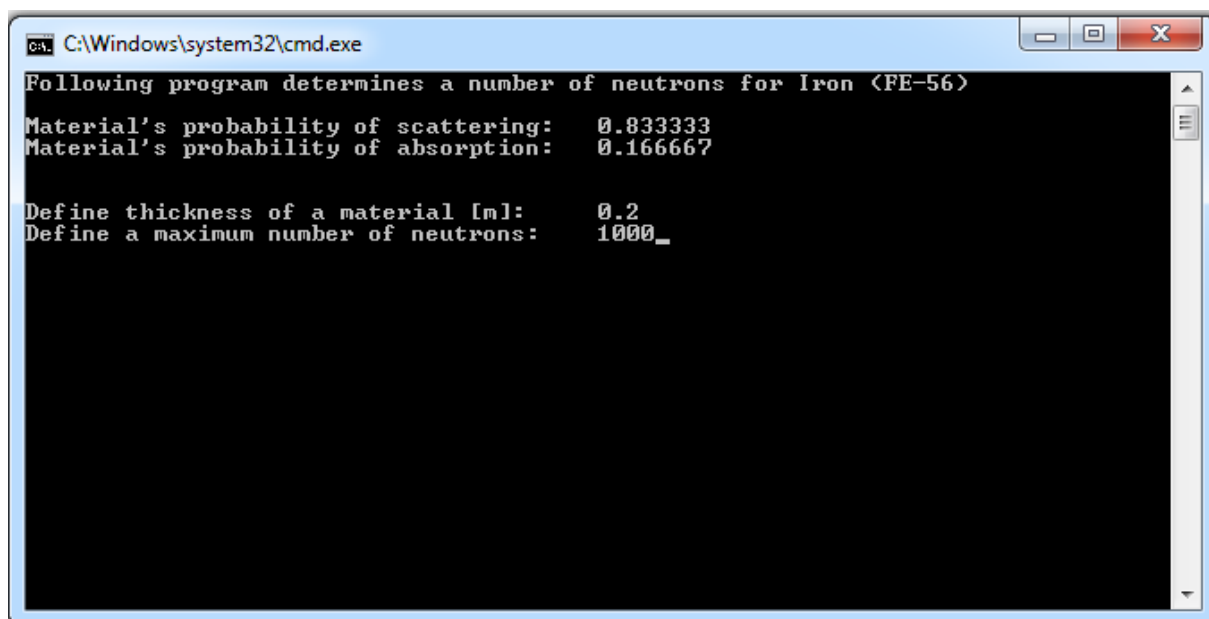
$$N_a = N_a + 1.$$

- 5.2 Jeżeli nie to cząstka ucieka i

$$N_e = N_e + 1.$$

### 6.2. OPIS DZIAŁANIA PROGRAMU

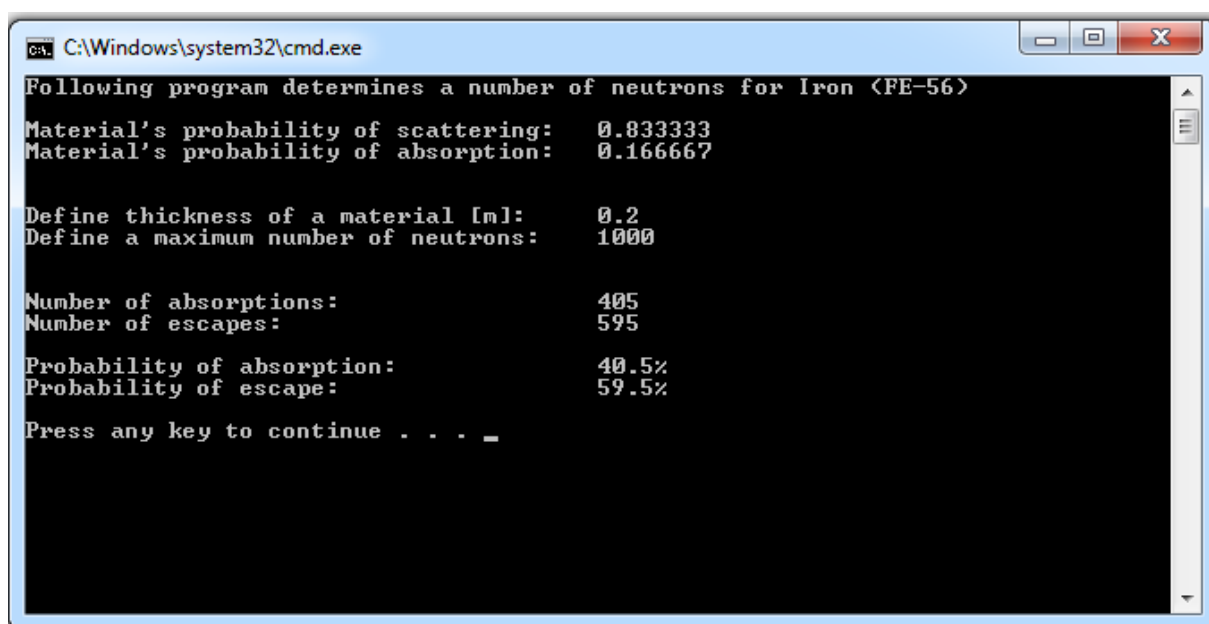
Tak jak we wcześniejszym programie, jako medium posłużył izotop żelaza Fe-56.



```
C:\Windows\system32\cmd.exe
Following program determines a number of neutrons for Iron <FE-56>
Material's probability of scattering: 0.833333
Material's probability of absorption: 0.166667

Define thickness of a material [m]: 0.2
Define a maximum number of neutrons: 1000_
```

Rys. 6.1. Wprowadzanie danych do programu.



```
C:\Windows\system32\cmd.exe
Following program determines a number of neutrons for Iron <FE-56>
Material's probability of scattering: 0.833333
Material's probability of absorption: 0.166667

Define thickness of a material [m]: 0.2
Define a maximum number of neutrons: 1000

Number of absorptions: 405
Number of escapes: 595

Probability of absorption: 40.5%
Probability of escape: 59.5%

Press any key to continue . . . _
```

Rys. 6.2. Wyniki otrzymane po wprowadzeniu danych.

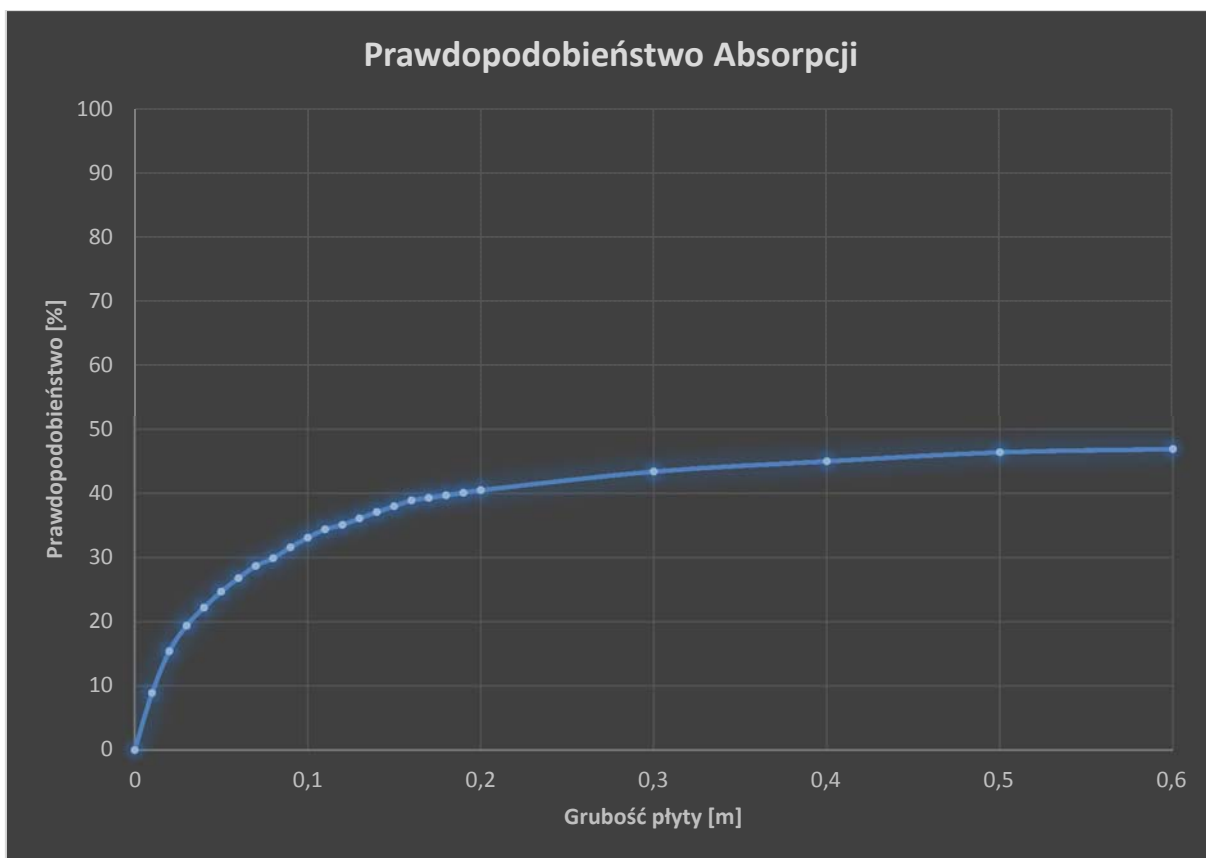
### 6.3. REZULTATY SYMULACJI

Jako, że w tej symulacji zastosowano ten sam ośrodek co w poprzedniej (Fe-56), można było się spodziewać, że podobnie jak we wcześniejszej symulacji, zmiany będą głównie zachodzić na grubości płyty 0m – 0.6m i rzeczywiście tak się stało. Powyżej grubości 0.6m prawdopodobieństwo absorpcji i wyjścia praktycznie się nie zmienia.

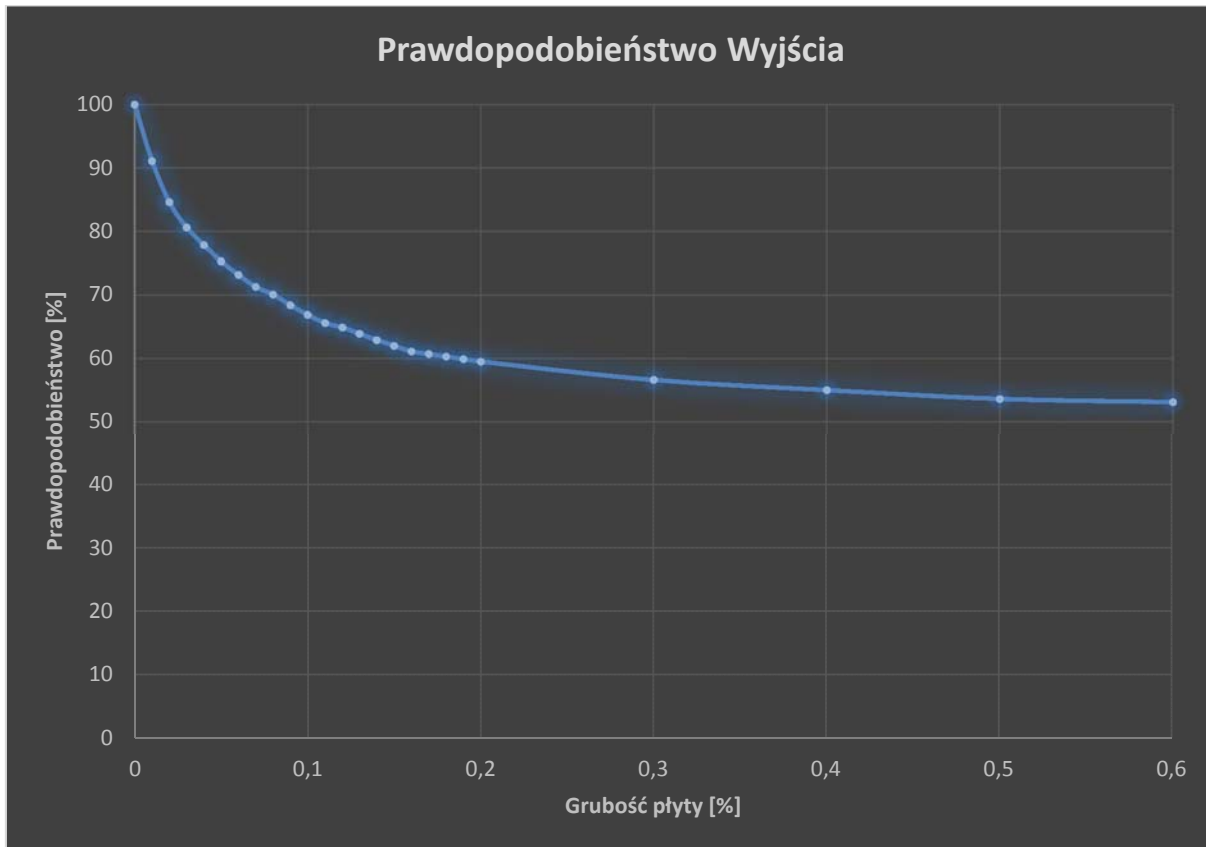
Tab nr 6.1. Prawdopodobieństwa ucieczki i absorpcji neutronów.

Grubość Płyty [m]	Prawdopodobieństwo %	
	Absorpcji	Wyjścia
0	0	100
0,01	8,9	91,1
0,02	15,4	84,6
0,03	19,4	80,6
0,04	22,2	77,8
0,05	24,7	75,3
0,06	26,8	73,2
0,07	28,7	71,3
0,08	29,9	70,1
0,09	31,6	68,4
0,1	33,1	66,9
0,11	34,4	65,6
0,12	35,1	64,9
0,13	36,1	63,9
0,14	37,1	62,9
0,15	38	62
0,16	38,9	61,1
0,17	39,3	60,7
0,18	39,7	60,3
0,19	40,1	59,9
0,2	40,5	59,5
0,3	43,4	56,6
0,4	45	55
0,5	46,4	53,6
0,6	46,9	53,1

Na podstawie tabeli 6.1 sporządzone wykresy prawdopodobieństwa absorpcji oraz wyjścia neutronów z medium w zależności od grubości płyty (rysunki 6.1 – 6.3).



Rys. 6.1. Prawdopodobieństwo absorpcji.



Rys. 6.2. Prawdopodobieństwo wyjścia.

## 7. KODY ŹRÓDŁOWE

### 7.1. TRANSMISJA NEUTRONÓW PRZEZ NIESKOŃCZONĄ PŁYTĘ

```
// Neutrony.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "random"
#include "iostream"
#include "cmath"

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    cout << "Following program determines a number of neutrons for Iron (FE-56)" << endl << endl;

    double Ea = 2;
    double Es = 10;
    double Et = Ea + Es;

    double p_of_s = Es / Et;
    double p_of_a = Ea / Et;

    cout << "Material's probability of scattering:    " << p_of_s << endl;
    cout << "Material's probability of absorption:    " << p_of_a << endl;
    cout << endl << endl;

    double h;
    cout << "Define thickness of a material [m]:    "; cin >> h;

    int N;
    cout << "Define a maximum number of neutrons:"; cin >> N;

    double N_t = 0;
    double N_r = 0;
    double N_a = 0;

    double mi = 1;

    for (int i = 0; i < N; i++){

        double x = 0;
        double *xk = new double[N];
        xk[0] = x;

        double mi = 1;
        double *mik = new double[N];
        mik[0] = mi;

        for (int k = 0; k < 1000; k++){
```



```

double gamma1 = ((double)rand() / (RAND_MAX));
double gamma2 = ((double)rand() / (RAND_MAX));
double gamma3 = ((double)rand() / (RAND_MAX));

double lambda = -(1.0 / Et) * log(gamma1);

xk[k + 1] = xk[k] + lambda*mik[k];

//-----CONDITIONS-----//
if (xk[k + 1] >= h){
    N_t = N_t + 1;
    break;
}

else if (xk[k + 1] <= 0){
    N_r = N_r + 1;
    break;
}

else if (gamma3 < p_of_a){
    N_a = N_a + 1;
    break;
}

mik[k] = 2.0 * gamma2 - 1.0;
//-----//
}

}

double P_t = N_t / N * 100;
double P_r = N_r / N * 100;
double P_a = N_a / N * 100;

//-----RESULTS-----//
cout << endl << endl;
cout << "Number of transmissions:      " << N_t << endl;
cout << "Number of reflections:          " << N_r << endl;
cout << "Number of absorptions:             " << N_a << endl;
cout << endl;
cout << "Probability of transmission:        " << P_t << "%" << endl;
cout << "Probability of reflection:          " << P_r << "%" << endl;
cout << "Probability of absorption:          " << P_a << "%" << endl;
cout << endl;
//-----//

system("pause");
return 0;
}

```

## 7.2. UCIECZKA NEUTRONÓW Z NIESKOŃCZONEJ PŁYTY

```
// Neutrony.cpp : Defines the entry point for the console application.
//

#include "stdafx.h"
#include "random"
#include "iostream"
#include "cmath"

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    cout << "Following program determines a number of neutrons for Iron (FE-56)" << endl << endl;

    double Ea = 2;
    double Es = 10;
    double Et = Ea + Es;

    double p_of_s = Es / Et;
    double p_of_a = Ea / Et;

    cout << "Material's probability of scattering:  " << p_of_s << endl;
    cout << "Material's probability of absorption:  " << p_of_a << endl;
    cout << endl << endl;

    double h;
    cout << "Define thickness of a material [m]:  "; cin >> h;

    int N;
    cout << "Define a maximum number of neutrons:      "; cin >> N;

    double N_a = 0;
    double N_e = 0;

    double d = 0;

    for (int i = 0; i < N; i++){

        double gamma1 = ((double)rand() / (RAND_MAX));
        double gamma2 = ((double)rand() / (RAND_MAX));
        double gamma3 = ((double)rand() / (RAND_MAX));

        double x = h*gamma1;

        double mi = 2.0 * gamma2 - 1.0;

        if (mi > 0){
            d = (h - x) / mi;
        }

        else {
            d = x / mi;
        }

        double lambda = -(1.0 / Et) * log(gamma3);
```

```

        if (lambda < d){
            N_a = N_a + 1;
        }

        else {
            N_e = N_e + 1;
        }

    }

    double P_a = N_a / N * 100;
    double P_e = N_e / N * 100;

    //-----RESULTS-----//
    cout << endl << endl;
    cout << "Number of absorptions:           " << N_a << endl;
    cout << "Number of escapes:                 " << N_e << endl;
    cout << endl;
    cout << "Probability of absorption:           " << P_a << "%" << endl;
    cout << "Probability of escape:                 " << P_e << "%" << endl;
    cout << endl;
    //-----//

    system("pause");
    return 0;
}

```

## 8. WNIOSKI I PODSUMOWANIE

W pracy zawarto i przedstawiono dwie uproszczone symulacje oddziaływania neutronów z materią. Uzyskane w oparciu o nie wyniki okazały się zgodne z przewidywaniami teoretycznymi dotyczącymi tych właśnie procesów.

W pierwszej symulacji – transmisji neutronów przez nieskończoną płytę, na podstawie otrzymanych wykresów można stwierdzić, że:

- Im większa grubość płyty, tym mniejsze prawdopodobieństwo transmisji neutronów,
- Im większa grubość płyty, tym większe prawdopodobieństwo odbicia neutronów,
- Im większa grubość płyty, tym większe prawdopodobieństwo absorpcji neutronów.

W drugiej symulacji – ucieczki neutronów przez nieskończoną płytę, na podstawie otrzymanych wykresów można stwierdzić, że:

- Im większa grubość płyty, tym większe prawdopodobieństwo absorpcji neutronów,
- Im większa grubość płyty, tym mniejsze prawdopodobieństwo wyjścia neutronów z danego medium.

Symulacje te można rozszerzać przez wprowadzanie wielu nowych parametrów albo zmianę geometrii ośrodka np. na walcową lub sferyczną.

## 9. BIBLIOGRAFIA

### Literatura:

- [1] Alireza Haghighat: Monte Carlo Methods for Particle Transport, CRC Press, 2015.
- [2] Ilya M. Sobol': A Primer for the Monte Carlo Method, CRC Press, 1994.
- [3] William L. Dunn, J. Kenneth Shultis: Exploring Monte Carlo Methods, Elsevier B.V., 2012.
- [4] Dariusz B. Tefelski: Symulacje Monte-Carlo transport neutronów. Wstęp do energetyki Jądrowej. Modelowanie Procesów Jądrowych, Wydział Fizyki Politechnika Warszawska, 2015.

### Strony internetowe:

- [1] [https://en.wikipedia.org/wiki/Monte\\_Carlo\\_method](https://en.wikipedia.org/wiki/Monte_Carlo_method)
- [2] [http://www.palisade.com/risk/monte\\_carlo\\_simulation.asp](http://www.palisade.com/risk/monte_carlo_simulation.asp)
- [3] <http://mathworld.wolfram.com/MonteCarloMethod.html>
- [4] <http://www.investopedia.com/terms/m/montecarlosimulation.asp>
- [5] <http://www.inference.phy.cam.ac.uk/mackay/erice.pdf>
- [6] <http://www.goldsim.com/Web/Introduction/Probabilistic/MonteCarlo/>

### Multimedia:

- [M1] [https://en.wikipedia.org/wiki/Stanislaw\\_Ulam](https://en.wikipedia.org/wiki/Stanislaw_Ulam)
- [M2] [https://en.wikipedia.org/wiki/Neutron\\_cross\\_section](https://en.wikipedia.org/wiki/Neutron_cross_section)