

Fizyka cząstek elementarnych

warsztaty popularnonaukowe

Spotkanie 2

Fizyka cząstek elementarnych i komputerowe
modelowanie procesów rozpraszania
ćwiczenia

Rafał Staszewski
Maciej Trzebiński

Instytut Fizyki Jądrowej
Polskiej Akademii Nauk

Zamiast wstępu

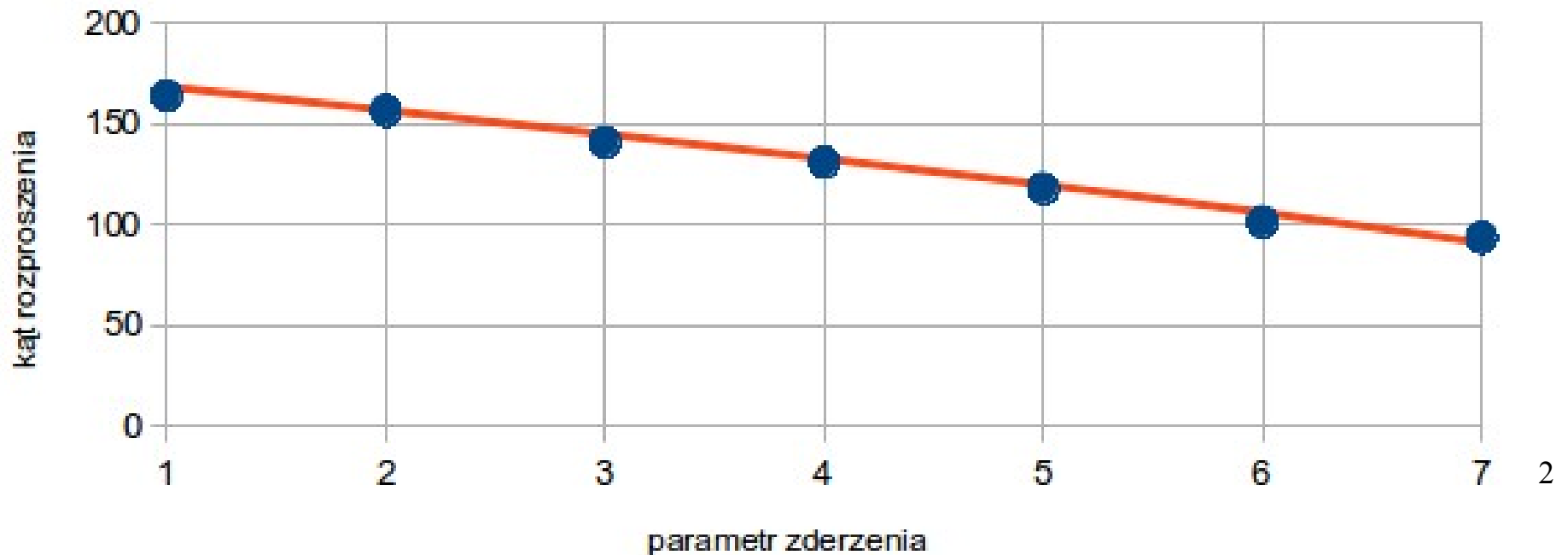
Spotkanie 1 - dyskusja n/t pomiaru zależności kąta rozpraszania od parametru zderzenia:

- wyprowadzenie zależności teoretycznej,
- przedstawienie danych doświadczalnych oraz krzywej teoretycznej na wykresie,
- uwzględnienie błędów systematycznych.

Zależność kąta rozpraszania od parametru zderzenia

dla małej kulki ($d \ll D$)

— f_i _theor_D ● f_i _srednie



Witaj świecie!

polecenia **cout** oraz **endl** nie są znane kompilatorowi – w celu ich użycia należy dołączyć odpowiednią bibliotekę (w tym przypadku **iostream**)

w celu uniknięcia pisania **std::cout** oraz **std::endl** można użyć przestrzeni nazw

główna funkcja programu w tym przypadku nie przyjmuje argumentów ()

wypisanie na ekran wiadomości **Hello world!**

```
1 #include <iostream>
```

```
3 using namespace std;
```

```
5 int main(){
```

```
7     cout << "Hello world!" << endl;
```

```
9     return 0;
```

```
10 }
```

funkcja typu **int** – musi więc zwracać wartość **całkowitą**:

Inne typy funkcji:

bool – zwraca wartość **true/false**

double – zwraca wartość **rzeczywistą**

void – **nie zwraca** żadnej wartości

Zadanie 1

Do programu **dodać funkcję** *void funkcja1 (int liczba_gwiazdek)*, która **zwórci linię z żadaną liczbą gwiazdek**.

Przy użyciu tej funkcji należy napisać następujący tekst na ekranie, a następnie zapisać go do pliku:

```
*  
**  
***  
****  
*****  
*****
```

Plik powinien mieć rozszerzenie **.txt**
Funkcja powinna wykorzystywać pętlę **for**.

Zadanie 1 - rozwiązanie

```
1 #include <iostream>
2 #include <fstream>
3
4 using namespace std;
5
6 ofstream plik_wyjsciowy;
7
8 void funkcja1(int liczba_gwiazdek){
9     for (int b=0; b<liczba_gwiazdek; b++)
10    {
11        plik_wyjsciowy << "*";
12    }
13    plik_wyjsciowy << endl;
14 }
15
16 int main(){
17
18     cout << "Witaj swiecie!" << endl;
19
20     plik_wyjsciowy.open("petla_for.txt");
21     for (int a=1; a<7; a++)
22     {
23         funkcja1(a); //uruchomienie funkcji funkcja1
24     }
25     plik_wyjsciowy.close();
26
27     return 0;
28 }
```

należy dołączyć bibliotekę obsługującą zapis do pliku (**fstream**)

tworzenie zmiennej typu **ofstream**

deklaracja funkcji (typ **void**)

zapisanie symbolu * do pliku

zapisanie znaku końca linii pliku

otwarcie pliku o nazwie *petla_for.txt*

zamknięcie pliku

Zadanie 2

Do programu **dodać funkcję** *rozpraszanie_na_kulce*, która zwróci wartość kąta rozproszenia dla danego parametru zderzenia.

Następnie utworzyć plik zawierający wartości kąta rozpraszania dla parametrów zderzenia od 0 do 1 z krokiem 0.05.

Format pliku:

wartosc_parametru_b *znak_tabulacji* *wartosc_kata* *znak_konca_linii*

Rozmiar tarczy $R = 1$.

Rozmiar pocisku $r = 0$.

Dane z pliku skopiować do arkusza kalkulacyjnego i narysować wykres zależności kąta rozproszenia od parametru zderzenia.

Zadanie 2 - rozwiązanie

```
3 #include <math.h>
```


biblioteka z funkcjami matematycznymi (w szczególności **asin**)

```
70 double rozpraszanie_na_kulce(double b, double R, double r){
71     double fi;
72     fi = 180. - 360.*asin(b / (R + r) )/3.1415;
73     return fi;
74 }
```

funkcja:

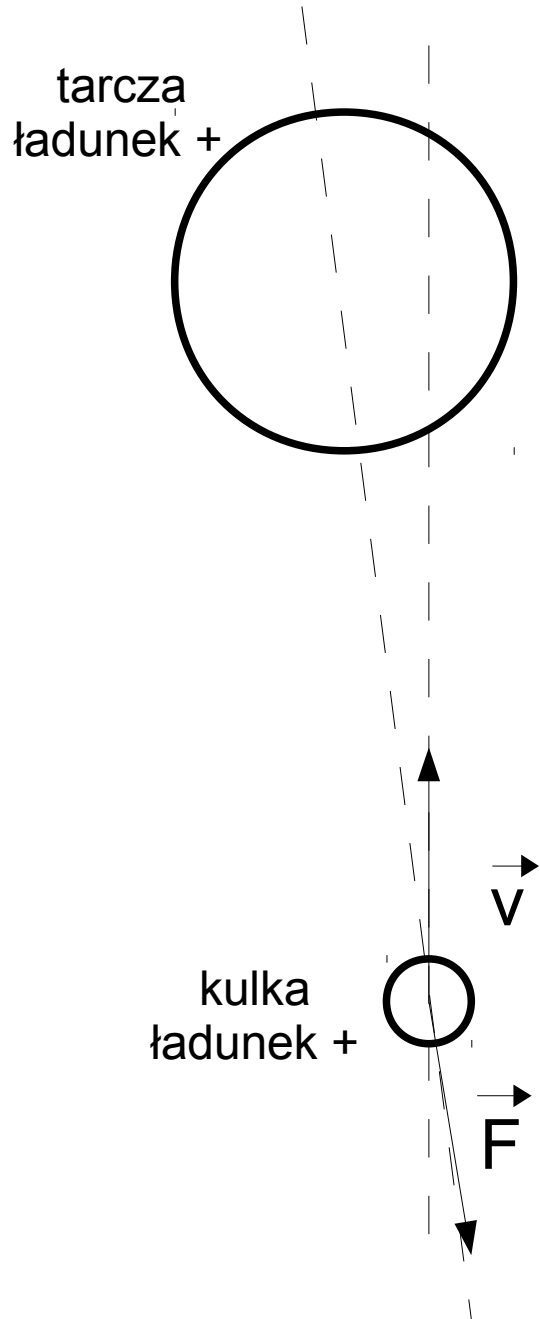
- przyjmuje trzy parametry typu double
- zwraca jeden typu double

```
90 int main(){
91
92     cout << "Witaj swiecie!" << endl;
93
94     petla_for();
95
96     ofstream plik_wyjsciowy_kulka;
97     plik_wyjsciowy_kulka.open("rozpraszanie_na_kulce.txt");
98     for (double b=0.; b<1.; b+=0.05)
99     {
100         plik_wyjsciowy_kulka << b << "\t" << rozpraszanie_na_kulce(b, 1. , 0.) << endl;
101     }
102     plik_wyjsciowy_kulka.close();
103
104     return 0;
105 }
```



zgodnie z przyjętymi założeniami parametr b zmienia się od 0 do 1 co 0.05, natomiast parametry R oraz r są stałe

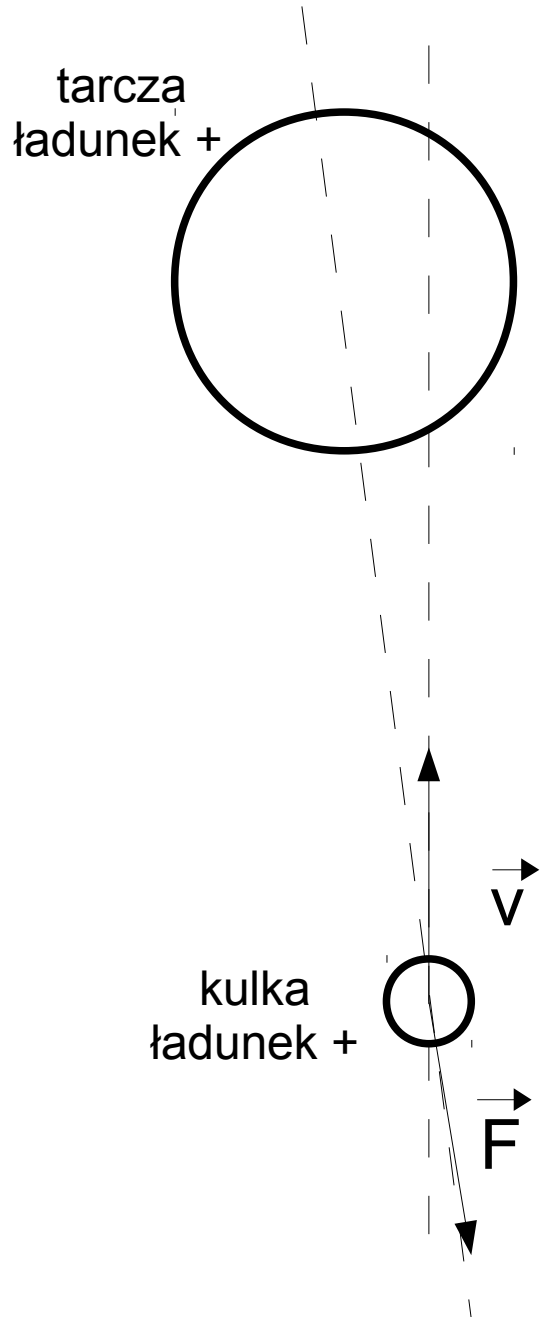
Rozpraszanie dwóch ładunków



Na kulkę działa siła \vec{F} .

Jakie będzie położenie kulki po czasie t ?

Rozpraszanie dwóch ładunków



Na kulkę działa siła \vec{F} .

Jakie będzie położenie kulki po czasie t ?

Początkowe położenie kulki: (x_0, y_0)

Początkowa prędkość kulki: $\vec{v} = (v_x, v_y)$

Równanie ruchu kulki:

$$\begin{cases} x(t+\Delta t) = \frac{1}{2} a_x \cdot \Delta t^2 + v_x \cdot \Delta t + x(t) \\ y(t+\Delta t) = \frac{1}{2} a_y \cdot \Delta t^2 + v_y \cdot \Delta t + y(t) \end{cases}$$

Przyśpieszenie wynosi: $\vec{a} = \vec{F} / m$ (gdzie m jest stałą)

Dla potencjału $1/r$ siła wynosi: $\vec{F} = C \vec{r} / r^3$ (gdzie C jest stałą)

$$\vec{a} = A \vec{r} / r^3,$$

czyli:

$$a_x = A \cdot x / r^3 \text{ oraz } a_y = A \cdot y / r^3$$

Prędkość wynosi:

$$\begin{cases} v_x = a_x \cdot \Delta t + v_{x0} \\ v_y = a_y \cdot \Delta t + v_{y0} \end{cases}$$

Zadanie 3

Do programu dodać funkcję *rozpraszanie_na_potencjale*, która zwróci wartość kąta rozproszenia dla danego parametru zderzenia.

Następnie utworzyć plik zawierający wartości kąta rozpraszania dla parametrów zderzenia od 0.05 do 1 z krokiem 0.05.

Format pliku:

wartosc_parametru_b *znak_tabulacji* *wartosc_kata* *znak_konca_linii*

Dane z pliku skopiować do arkusza kalkulacyjnego i narysować wykres zależności kąta rozproszenia od parametru zderzenia.

Wskazówki:

- dane zapisać w pliku tekstowym (*rozpraszanie_na_potencjale_histogram.txt*) a histogram narysować za pomocą arkusza kalkulacyjnego
- stałą A przyjąć równą 1
- wielkość kroku $t = 0.01$
- początkowe położenie w x jest równe parametrowi zderzenia
- początkowe położenie w y musi być daleko od tarczy (np. $y = -10$)
- początkowa prędkość w x $v_x = 0$
- początkowa prędkość w y $v_y = 10$

Zadanie 3 - rozwiązanie (funkcja)

```
61 double rozpraszanie_na_potencjale(double b){
62
63     const double A = 1.;
64     const double t = 0.01;
65     const double x0 = b;
66     const double y0 = -10.;
67     const double v_x0 = 0.;
68     const double v_y0 = 10.;
69
70     double r, x, y, v_x, v_y, a_x, a_y, fi;
71
72     x = x0;
73     y = y0;
74     v_x = v_x0;
75     v_y = v_y0;
76
77     while (x*x + y*y <= x0*x0 + y0*y0)
78     {
79         r = sqrt(x*x + y*y);
80
81         a_x = A * x / (r*r*r);
82         a_y = A * y / (r*r*r);
83
84         v_x = a_x * t + v_x;
85         v_y = a_y * t + v_y;
86
87         x = 0.5 * a_x * t * t + v_x * t + x;
88         y = 0.5 * a_y * t * t + v_y * t + y;
89     }
90
91     if (b>0.) fi = atan2(v_x, v_y);
92     else fi = atan2(-v_x, v_y);
93
94     return fi*180./3.14159265;
95 }
```

deklaracje stałych

dynamiczne ustawienie zakresu,
w którym program ma dokonywać
obliczeń

formuła na kąt rozpraszania

obliczenie przyśpieszenia

obliczenie prędkości

równania ruchu

formuła na kąt rozpraszania

przeliczenie na stopnie

Zadanie 3 - rozwiązanie (program główny)

```
79 ofstream plik_wyjsciowy_potencjal;  
80 plik_wyjsciowy_potencjal.open("rozpraszanie_na_potencjale.txt");  
81  
82 for (double a=0.05; a<1.; a+=0.05)  
83 {  
84     plik_wyjsciowy_potencjal << a << "\t" << rozpraszanie_na_potencjale(a) << endl;  
85 }  
86  
87 plik_wyjsciowy_potencjal.close();  
88
```

Zadanie 6 - przykładowa trajektoria

Trajektoria pocisku dla $A=1$, $b=0.001$, $y=-10$, $v_y=1$

