

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет имени Н. Э.  
Баумана (национальный исследовательский университет)»  
(МГТУ им. Н. Э. Баумана)**

**Отчёт: «Метод стрельбы для решения стационарного уравнения  
Шрёдингера»**

**По курсу:  
«Информационное обеспечение разработок в области нанотехнологий»**

Выполнил: Шарафутдинов Э.М.

Группа: РЛ6-61Б

Подпись:

Дата:

Проверила: Ветрова Н.А.

Подпись:

Дата:

**Москва, 2023 г**

## Оглавление

<b>1 Теория</b>	<b>3</b>
<b>1.1 Метод стрельбы</b>	<b>3</b>
<b>1.2 Пример</b>	<b>4</b>
<b>1.3 Метод Рунге-Кутты</b>	<b>4</b>
<b>2 Алгоритм</b>	<b>8</b>
<b>3 Требования к входным данным</b>	<b>10</b>
<b>4 Реализация метода на языке MatLab</b>	<b>11</b>
<b>5 Сравнение метода стрельбы с другими методами решения стационарного уравнения Шрёдингера</b>	<b>16</b>
<b>5.1 Сравнение с аналитическим методом</b>	<b>16</b>
<b>5.2 Сравнение с методом конечных разностей</b>	<b>22</b>
<b>6 Возможная оптимизация реализации метода</b>	<b>31</b>
<b>Список литературы</b>	<b>36</b>

# 1 Теория

## 1.1 Метод стрельбы

Методы стрельбы разработаны с целью преобразования краевых задач ОДУ в эквивалентные начальные задачи. В задачах с начальным значением мы можем начать с начального значения и продвигаться вперед, чтобы получить решение. Но этот метод не работает для краевых задач, потому что не хватает начальных условий для решения ОДУ.

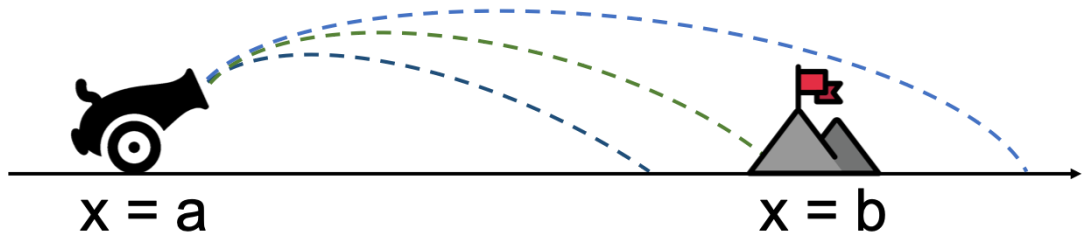


Рисунок 1 - метод стрельбы

Название метода стрельбы происходит от аналогии со стрельбой по мишеням: как показано на рисунке выше, мы стреляем в цель и наблюдаем, где она попадает в цель, на основе ошибок мы можем скорректировать нашу цель и снова стрелять в надежде, что попадём близко к цели. Из аналогии видно, что метод стрельбы является итерационным методом.

Метод стрельбы является удобным инструментом для численного решения одномерных обыкновенных дифференциальных уравнений и получения необходимых собственных значений. Он может быть использован для вычисления собственных значений стационарного одномерного уравнения Шрёдингера с граничными условиями.

## 1.2 Пример

ОДУ второго порядка при заданных  $f(a)=fa$  и  $f(b)=fb$ .

$$F\left(x, f(x), \frac{df(x)}{dx}\right) = \frac{d^2 f(x)}{dx^2}$$

Шаг 1: Мы начинаем весь процесс с угадывания  $f'(a)=\alpha$  вместе с  $f(a)=fa$ , мы превращаем вышеуказанную задачу в задачу с начальным значением с двумя условиями, все для значения  $x=a$ . Это шаг прицеливания.

Шаг 2: Мы можем использовать метод Рунге-Кутты, чтобы найти  $f(b)=fb$ . Это этап стрельбы.

Шаг 3: Теперь мы сравниваем значение  $fb$  с  $fb$ , обычно наше начальное предположение не является правильным, и  $fb \neq fb$ , но мы хотим, чтобы  $fb - fb = 0$ , поэтому мы корректируем наши начальные предположения и повторяем. Это шаг итерации.

## 1.3 Метод Рунге-Кутты

Метод Рунге-Кутты используют для расчета стандартных моделей достаточно часто, так как при небольшом объеме вычислений он обладает точностью метода  $O^4(h)$ .

Для построения разностной схемы интегрирования воспользуемся разложением функции

$$\frac{dy}{dx} = f(x, y(x)), \quad 0 < x \leq T, \quad y(0) = y_0 \quad (1)$$

в ряд Тейлора:

$$y(x_{k+1}) = y(x_k) + y'(x_k)h + y''(x_k)\frac{h^2}{2} + \dots$$

Заменим вторую производную в этом разложении выражением

$$y''(x_k) = (y'(x_k))' = f'(x_k, y(x_k)) \approx \frac{f(\tilde{x}, \tilde{y}) - f(x_k, y(x_k))}{\Delta x}$$

где

$$\tilde{x} = x_k + \Delta x, \tilde{y} = y(x_k + \Delta x)$$

Причем  $\Delta x$  подбирается из условия достижения наибольшей точности записанного выражения. Для дальнейших выкладок произведем замену величины «у с тильдой» разложением в ряд Тейлора:

$$\tilde{y} = y(x_k + \Delta x) = y(x_k) + y'(x_k)\Delta x + \dots$$

Для исходного уравнения (1) построим вычислительную схему:

$$y_{k+1} = y_k + f(x_k, y_k) \cdot h + \frac{h^2}{2 \Delta x} \cdot (f(x_k + \Delta x, y_k + y'_k \Delta x) - f(x_k, y_k))$$

которую преобразуем к виду:

$$\begin{aligned} y_{k+1} &= y_k + h \cdot \left[ \left( 1 - \frac{h}{2 \Delta x} \right) \cdot f(x_k, y_k) + \frac{h}{2 \Delta x} f(x_k + \Delta x, y_k + y'_k \Delta x) \right] = \\ &= y_k + h \cdot \left[ \left( 1 - \frac{h}{2 \Delta x} \right) \cdot f(x_k, y_k) + \frac{h}{2 \Delta x} f\left(x_k + \frac{\Delta x}{h} h, y_k + f(x_k, y_k) \frac{\Delta x}{h} h\right) \right] \end{aligned}$$

Введем следующие обозначения:

$$\alpha = \frac{h}{2 \Delta x}, \quad \beta = 1 - \frac{h}{2 \Delta x}, \quad \gamma = \frac{\Delta x}{h}, \quad \delta = f(x_k, y_k) \frac{\Delta x}{h}$$

Эти обозначения позволяют записать предыдущее выражение в форме:

$$y_{k+1} = y_k + h \cdot [\beta \cdot f(x_k, y_k) + \alpha \cdot f(x_k + \gamma \cdot h, y_k + \delta \cdot h)]$$

Очевидно, что все введенные коэффициенты зависят от величины  $\Delta x$  и могут быть определены через коэффициент  $\alpha$ , который в этом случае играет роль параметра:

$$\beta = 1 - \alpha, \quad \gamma = \frac{1}{2\alpha}, \quad \delta = f(x_k, y_k) \frac{2}{\alpha}$$

Окончательно схема Рунге-Кутты принимает вид:

$$y_{k+1} = y_k + h \cdot \left[ (1 - \alpha) \cdot f(x_k, y_k) + \alpha \cdot f\left(x_k + \frac{h}{2\alpha}, y_k + f(x_k, y_k) \frac{h}{2\alpha}\right) \right]$$

Та же схема в форме разностного аналога уравнения (1):

$$\frac{y_{k+1} - y_k}{h} = (1 - \alpha) \cdot f(x_k, y_k) + \alpha \cdot f\left(x_k + \frac{h}{2\alpha}, y_k + f(x_k, y_k) \frac{h}{2\alpha}\right)$$

При  $\alpha = 0$  получаем как частный случай уже известную схему Эйлера:

$$y_{k+1} = y_k + h \cdot f(x_k, y_k)$$

При  $\alpha = 1$ :

$$y_{k+1} = y_k + h \cdot f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2} \cdot f(x_k, y_k)\right)$$

При  $\alpha = 1$  проведение расчетов на очередном шаге интегрирования можно рассматривать как последовательность нижеследующих операций.

1. Вычисляется выражение, представляющее собой полушаг интегрирования по схеме Эйлера, то есть определяется приближенное значение искомой функции в точке  $x_k + h/2$ :

$$y_{k+1/2} = y_k + \frac{h}{2} \cdot f(x_k, y_k)$$

2. Для той же промежуточной точки находится приближенное значение производной:

$$y'_{k+1/2} = f\left(x_k + \frac{h}{2}, y_{k+1/2}\right)$$

3. Определяется уточненное значение функции в конечной точке всего шага, причем по схеме Эйлера с вычисленным на предыдущем шаге значением производной:

$$y_{k+1} = y_k + h \cdot y'_{k+1/2}$$

## 2 Алгоритм

Функция, вычисляющая  $\psi$  методом стрельбы, принимает на вход длину потенциальной ямы  $L$ , нм, предполагаемую энергию частицы  $E$ , эВ, потенциал  $U$ , эВ, шаг продвижения  $dx$ , в нм и допуск точности решения  $tolerance$ . Функция возвращает вектор  $x_r$  со значениями координатной сетки в яме в нм, вектор значений  $psip \cdot nm^{-0.5}$  в узлах координатной сетки, а также вектор  $E$  в эВ, содержащий аппроксимированную энергию, при которой выполняется второе граничное условие.

Алгоритм:

1. Задаётся величина  $dE$  в эВ, на которую будет увеличиваться входная энергия в случае, если не будет выполняться приближённое равенство волновой функции нулю на второй границе ямы, рассчитывается вектор координат  $x_p$ .
2. Задаётся граничное условие на первой границе:  $\psi(0) = 0$ .
3. Предположение о том, что волновая функция не равна нулю на второй границе:  $psifinal = 1$ .
4. Инициализация первого цикла. Условие выхода -  $psifinal \approx 0$ . Задаётся нулевой вектор  $psip$ , который будут хранить значения  $\psi$ , а также следствие из первого граничного условия:  $\frac{d\psi(0)}{dx} = 1$ . В случае невыполнения второго граничного условия каждую итерацию происходит увеличение входной энергии  $E$  на  $dE$ , и обнуление вектора  $psip$ .
5. Инициализация второго цикла (вложен в первый). Условие выхода -  $x = L$ . Рассчитывается  $\psi$ :



1) Из уравнения Шрёдингера: 
$$\ddot{\psi} = -\frac{2 * m}{\hbar^2} * E * \psi$$

2) По определению производной 
$$\dot{\psi} = \frac{\psi(x + \Delta x) - \psi(x)}{\Delta x}$$
 . Откуда 
$$\psi(x + \Delta x) = \psi(x) + \dot{\psi} * \Delta x$$
 .

3) 
$$\dot{\psi}(x) = \frac{\psi(x + \Delta x) - \psi(x)}{\Delta x}$$
 . Откуда 
$$\psi(x + \Delta x) = \psi(x) + \dot{\psi} * \Delta x$$
 .

Полученное значение записывается в вектор `psip` .

6. После работы второго цикла последнее значение  $\psi$  записывается в `psifinal` . Проверка условия завершения первого цикла. В случае выполнения условия происходит выход из первого цикла, нормировка  $\psi$  и выход из функции. В случае невыполнения - новая итерация.

### 3 Требования к входным данным

1. Потенциал  $U$  должен быть конечным и непрерывным в интересующей нас области.
2. Начальное предположение для энергии  $E$  должно быть достаточно близко к фактическому собственному значению и быть меньше него, чтобы метод поиска корней мог эффективно сходиться. Чтобы найти правильный энергетический диапазон, может потребоваться несколько догадок.
3. Шаг продвижения по координатной сетке  $dx$  должен быть достаточно мал. Слишком большой шаг приведет к неточным решениям, не удовлетворяющим граничным условиям.
4. Допуск для нахождения корня  $tolerance$  должен быть достаточно малым для желаемой точности собственного значения энергии. Однако более жёсткие допуски означают большее количество итераций для сходимости.

## 4 Реализация метода на языке MatLab

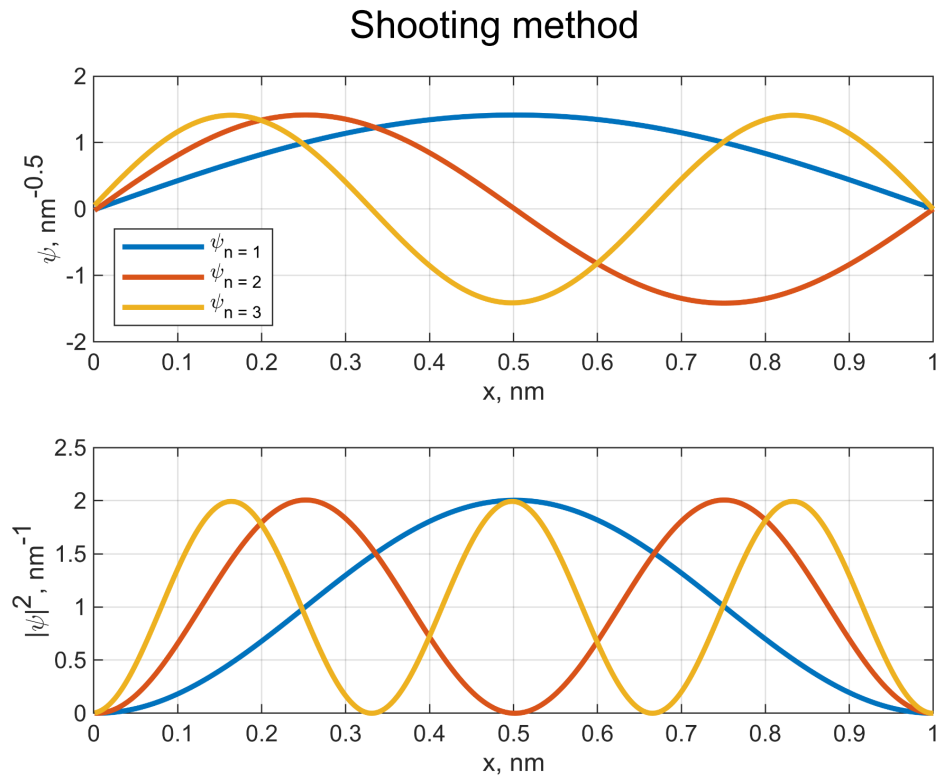


Рисунок 2 - вид волновой функции в бесконечно глубокой потенциальной яме, полученной методом стрельбы

Листинг 1 - метод стрельбы для решения стационарного уравнения Шрёдингера для электрона в 1d-PW на языке MatLab

```
clc, clear, close all
datetime('now')
```

```
ans = datetime
```

```
18-Jun-2023 16:55:05
```

```
%Planck's constant in J * sec
hbar = 1.0546e-34;

%Mass of electron in kg
m0 = 9.1094e-31;

%Conversion constant from joules to electronvolts
J2eV = 6.2415e18;
```

```

% Pit length in nm
L = 1;

%Potential in eV
U = 0;

%Principal quantum numbers
n = [1, 2, 3];

%Energies of the first three energy levels
E_an = pi ^ 2 * n .^ 2 * hbar ^ 2 / (2 * m0 * (L * 1e-9) ^2) * J2eV;

```

Shooting method

```

%Energies close to the analytically calculated energies of the first three energy levels
E_apr = [0.99 * E_an(1), 0.99 * E_an(2), 0.99 * E_an(3)]; %eV

%Dependence of the wave function on the coordinate
subplot(2, 1, 1);

%First energy level
[xp, psi_sh_1, E_sh_1] = shooting(E_apr(1), U, L, 0.001, 0.001);
plot(xp, psi_sh_1, 'LineWidth', 2);
hold on;

%Second energy level
[xp, psi_sh_2, E_sh_2] = shooting(E_apr(2), U, L, 0.001, 0.001);
plot(xp, psi_sh_2, 'LineWidth', 2);

%Third energy level
[xp, psi_sh_3, E_sh_3] = shooting(E_apr(3), U, L, 0.001, 0.001);
plot(xp, psi_sh_3, 'LineWidth', 2);

```

```

%Graphics customization
legend('\psi_{n = 1}', '\psi_{n = 2}', '\psi_{n = 3}', 'Location', 'southwest');
grid on;
xlabel('x, nm');
ylabel('\psi, nm^{-0.5}');
xlim([0, L]);

%Dependence of the squared modulus of the wave function on the coordinate
subplot(2, 1, 2);

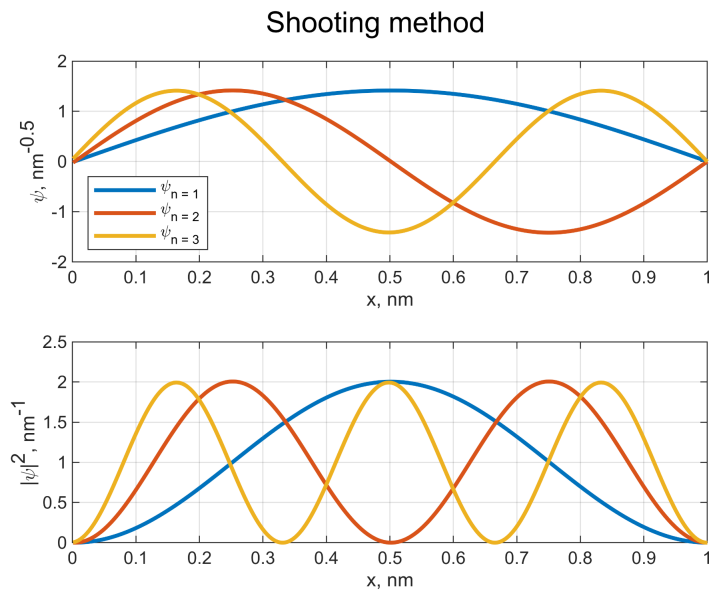
%First energy level
plot(xp, abs(psi_sh_1) .^2, 'LineWidth', 2);
hold on;

%Second energy level
plot(xp, abs(psi_sh_2) .^2, 'LineWidth', 2);

%Third energy level
plot(xp, abs(psi_sh_3) .^2, 'LineWidth', 2);

%Graphics customization
grid on;
xlabel('x, nm');
ylabel('|\psi|^2, nm^{-1}');
xlim([0, L]);
sgtitle('Shooting method');

```



```
function [xp, psip, E] = shooting(E, U, L, dx, tolerance)
    %hbar * c [eV * nm]
    hbc = 1.0546e-34 * 6.2415e18 * 3e17;
    %mass of electron in mc^2
    m = 510998.95;
    %Coefficient in the Schrodinger equation
    k = 2 * m / hbc ^2;

    %Increasing Energy to Satisfy the Second Boundary Condition
    dE = 0.001; %eV

    %Vector for storing the coordinate
    xp = dx : dx : L;

    %First boundary condition
    psi = 0;
    %The last point of the wave function should be approximately equal to zero
    %We assume that it is not equal to 0
    psifinal = 1;

    %Loop until psifinal is close to zero
    while abs(psifinal) > tolerance
        %First boundary condition
        dpsi = 1;

        %Vector for storing value of the wave function
        psip = zeros(1, numel(xp));

        %Loop until we reach the end of the pit
        for i = 1 : L / dx
```

```

    %Schrodinger equation
    ddpsi = -k * psi * E;

    %Representing the derivative as a limit
    dpsi = dpsi + ddpsi * dx;
    psi = psi + dpsi * dx;

    %Filling vector
    psip(i) = psi;
end
%Assigning the last value of the wave function to psifinal
psifinal = psi;

%Increasing the energy to the desired value only
% if the second boundary condition is not met
if (abs(psifinal) > tolerance)
    E = E + dE;
end
end

%Normalization
area = sum(psip .^ 2) * dx;

psip = psip / sqrt(area);
end

```

## 5 Сравнение метода стрельбы с другими методами решения стационарного уравнения Шрёдингера

### 5.1 Сравнение с аналитическим методом

Метод сводится к аналитическому решению дифференциального уравнения второго порядка с дополнительными условиями.

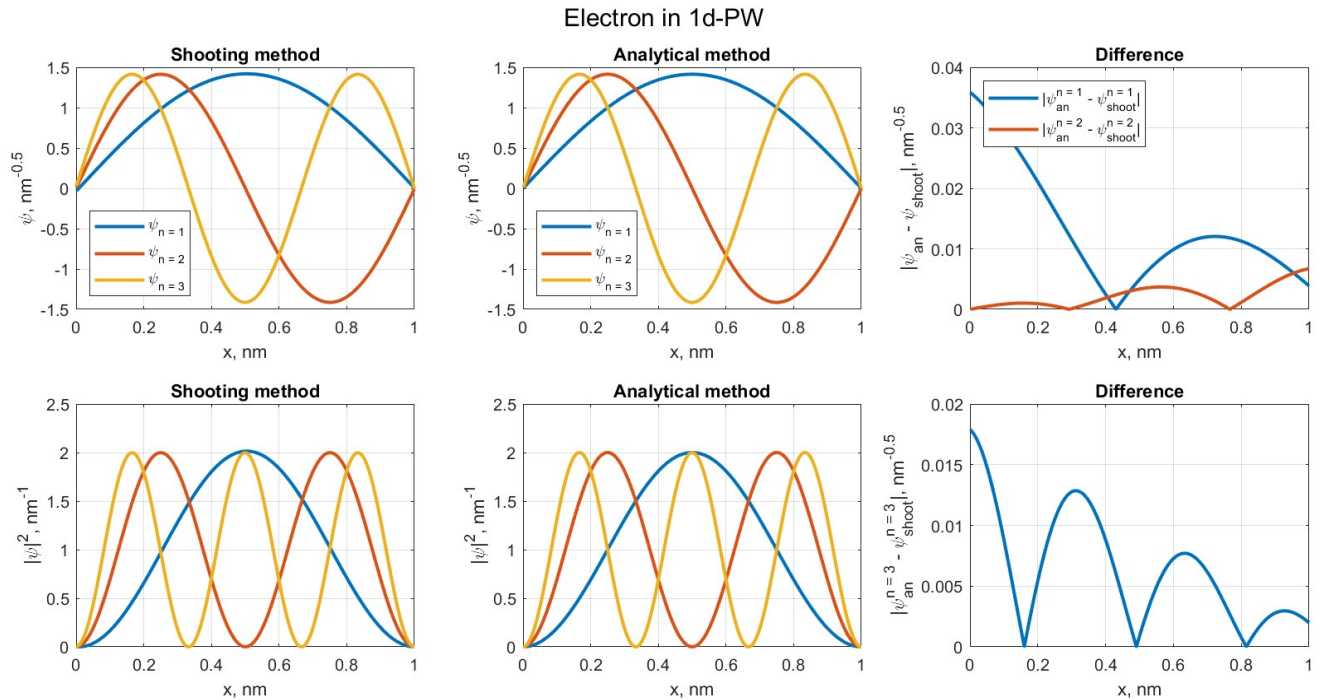


Рисунок 3 - сравнение результатов метода стрельбы и аналитического метода

Листинг 2 - сравнение результатов метода стрельбы с аналитическим методом

```
clc, clear, close all
datetime('now')
```

```
ans = datetime
```

```
18-Jun-2023 16:55:05
```

```
%Planck's constant in J * sec
hbar = 1.0546e-34;
```



```

%Mass of electron in kg
m0 = 9.1094e-31;

%Conversion constant from joules to electronvolts
J2eV = 6.2415e18;

% Pit length in nm
L = 1;

%Potential in eV
U = 0;

%Cell array for energy comparison
C = cell(3, 4);
C(1, 1) = {'n'};
C(2, 1) = {'E_shoot, eV'};
C(3, 1) = {'E_an, eV'};
C(1, 2) = {1};
C(1, 3) = {2};
C(1, 4) = {3};

```

Shooting method

```

%Energies close to the analytically calculated energies of the first three energy
levels
E_apr = [0.37, 1.504, 3.38]; %eV

%Dependence of the wave function on the coordinate
figure(1);
subplot(2, 3, 1);

%First energy level
[xp, psi_sh_1, E_sh_1] = shooting(E_apr(1), U, L, 0.001, 0.001);
plot(xp, psi_sh_1, 'LineWidth', 2);
hold on;

```

```

%Second energy level
[xp, psi_sh_2, E_sh_2] = shooting(E_apr(2), U, L, 0.001, 0.001);
plot(xp, psi_sh_2, 'LineWidth', 2);

%Third energy level
[xp, psi_sh_3, E_sh_3] = shooting(E_apr(3), U, L, 0.001, 0.001);
plot(xp, psi_sh_3, 'LineWidth', 2);

%Filling cell array with energies calculated by shooting method
C(2, 3) = num2cell(E_sh_2);
C(2, 2) = num2cell(E_sh_1);
C(2, 4) = num2cell(E_sh_3);

%Graphics customization
legend('\psi_{n = 1}', '\psi_{n = 2}', '\psi_{n = 3}', 'Location', 'southwest');
title('Shooting method');
grid on;
xlabel('x, nm');
ylabel('\psi, nm^{-0.5}');
xlim([0, L]);

%Dependence of the squared modulus of the wave function on the coordinate
subplot(2, 3, 4);

%First energy level
plot(xp, abs(psi_sh_1) .^2, 'LineWidth', 2);
hold on;

%Second energy level
plot(xp, abs(psi_sh_2) .^2, 'LineWidth', 2);

```

```

%Third energy level
plot(xp, abs(psi_sh_3) .^2, 'LineWidth', 2);

%Graphics customization
title('Shooting method');
grid on;
xlabel('x, nm');
ylabel('|psi|^2, nm^{-1}');
xlim([0, L]);

```

Analytical method

```

%Principal quantum numbers
n = [1, 2, 3];

%Energies of the first three energy levels
E_an = pi ^ 2 * n .^ 2 * hbar ^ 2 / (2 * m0 * (L * 1e-9) ^2) * J2eV;

%Filling cell array with analytically calculated energies
C(3, 2) = num2cell(E_an(1));
C(3, 3) = num2cell(E_an(2));
C(3, 4) = num2cell(E_an(3));

%Dependence of the wave function on the coordinate
subplot(2, 3, 2);

%First energy level
psi_an_1 = sqrt(2 / L) * sin(pi * n(1) * xp / L);
plot(xp, psi_an_1, 'LineWidth', 2);
hold on;

%Second energy level
psi_an_2 = sqrt(2 / L) * sin(pi * n(2) * xp / L);

```

```

plot(xp, psi_an_2, 'LineWidth', 2);

%Third energy level
psi_an_3 = sqrt(2 / L) * sin(pi * n(3) * xp / L);
plot(xp, psi_an_3, 'LineWidth', 2);

%Graphics customization
legend('\psi_{n = 1}', '\psi_{n = 2}', '\psi_{n = 3}', 'Location', 'southwest');
title('Analytical method');
grid on;
xlabel('x, nm');
ylabel('\psi, nm^{-0.5}');
xlim([0, L]);

%Dependence of the squared modulus of the wave function on the coordinate
subplot(2, 3, 5);

%First energy level
plot(xp, abs(psi_an_1) .^2, 'LineWidth', 2);
hold on;

%Second energy level
plot(xp, abs(psi_an_2) .^2, 'LineWidth', 2);

%Third energy level
plot(xp, abs(psi_an_3) .^2, 'LineWidth', 2);

%Graphics customization
title('Analytical method');
grid on;
xlabel('x, nm');
ylabel('|\psi|^2, nm^{-1}');

```

```
xlim([0, L]);
```

Difference

```
%Abs difference between psi_sh and psi_an
subplot(2, 3, 3)

%psi_sh_1 and psi_an_1
plot(xp, abs(psi_an_1 - psi_sh_1), 'LineWidth', 2);
hold on;
%psi_sh_2 and psi_an_2
plot(xp, abs(psi_an_2 - psi_sh_2), 'LineWidth', 2);

%Graphics customization
title('Difference');
legend('\psi_{an}^{n = 1} - \psi_{shoot}^{n = 1}|', ...
       '\psi_{an}^{n = 2} - \psi_{shoot}^{n = 2}|', ...
       'Location', 'northwest');
grid on;
xlabel('x, nm');
ylabel('|\psi_{an} - \psi_{shoot}|, nm^{-0.5}');
xlim([0, L]);
sgtitle('Electron in 1d-PW');

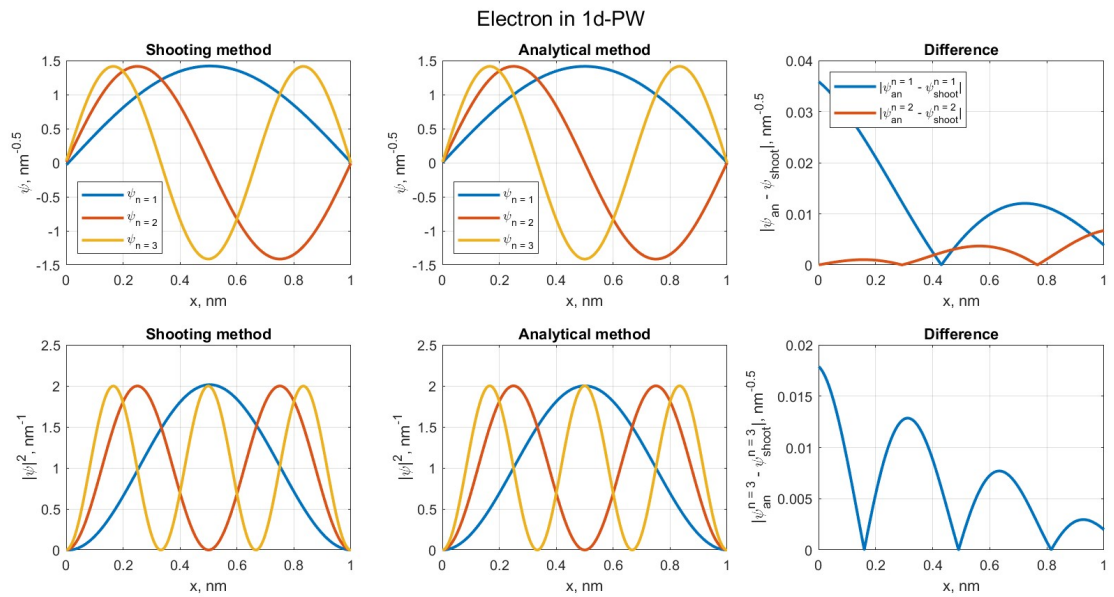
subplot(2, 3, 6)
%psi_sh_3 and psi_an_3
plot(xp, abs(psi_an_3 - psi_sh_3), 'LineWidth', 2);
hold on;

%Graphics customization
title('Difference');
grid on;
xlabel('x, nm');
```

```

ylabel('| \psi_{an}^{n = 3} - \psi_{shoot}^{n = 3}|, nm^{-0.5}');
xlim([0, L]);
sgtitle('Electron in 1d-PW');

```



```
%Outputting filled cell array
```

```
C
```

```
C = 3x4 cell
```

	1	2	3	4
1	'n'	1	2	3
2	'E_shoot, eV'	0.382	1.504	3.381
3	'E_an, eV'	0.376	1.5042	3.3844

## 5.2 Сравнение с методом конечных разностей

Этот метод аппроксимирует дифференциальное уравнение, заменяя производные в уравнении соответствующими приближениями разностных частных. Интервал разбивается на  $N$  интервалов с точками сетки  $a = x_1, \dots, x_N, x_{N+1} = b$ , равноотстоящими через  $h = (b - a) / N$ .

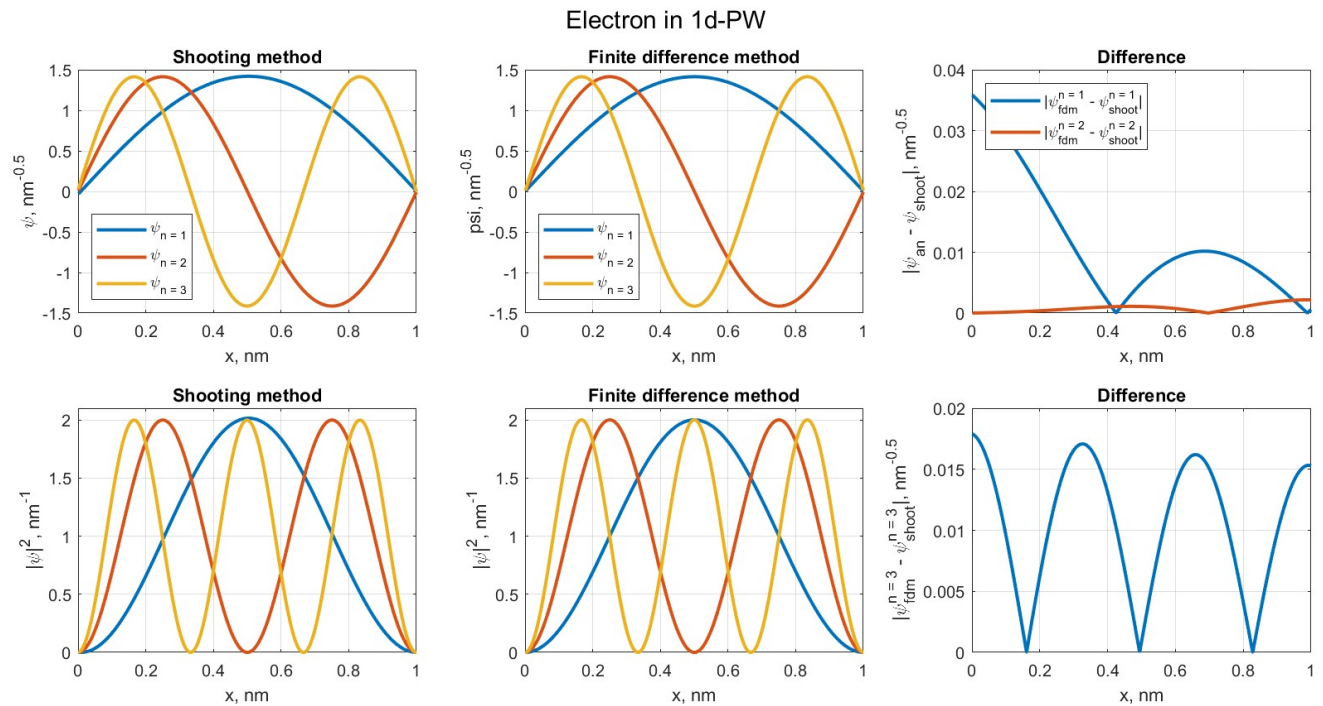


Рисунок 4 - сравнение результатов метода стрельбы и метода конечных разностей

Листинг 3 - сравнение результатов метода стрельбы с методом конечных разностей

```
clc, clear, close all
datetime('now')
```

```
ans = datetime
```

```
18-Jun-2023 16:55:05
```

```
%Planck's constant in J * sec
hbar = 1.0546e-34;
%Mass of electron in kg
```

```

m0 = 9.1094e-31;

%Conversion constant from joules to electronvolts
J2eV = 6.2415e18;

% Pit length in nm
L = 1;

%Potential in eV
U = 0;

%Cell array for energy comparison
C = cell(3, 4);
C(1, 1) = {'n'};
C(2, 1) = {'E_shoot, eV'};
C(3, 1) = {'E_an, eV'};
C(1, 2) = {1};
C(1, 3) = {2};
C(1, 4) = {3};

```

Shooting method

```

%Energies close to the analytically calculated energies of the first three energy
levels
E_apr = [0.37, 1.504, 3.38]; %eV

%Dependence of the wave function on the coordinate
figure(1);
subplot(2, 3, 1);

%First energy level
[xp, psi_sh_1, E_sh_1] = shooting(E_apr(1), U, L, 0.001, 0.001);
plot(xp, psi_sh_1, 'LineWidth', 2);
hold on;

```



```

%Second energy level
[xp, psi_sh_2, E_sh_2] = shooting(E_apr(2), U, L, 0.001, 0.001);
plot(xp, psi_sh_2, 'LineWidth', 2);

%Third energy level
[xp, psi_sh_3, E_sh_3] = shooting(E_apr(3), U, L, 0.001, 0.001);
plot(xp, psi_sh_3, 'LineWidth', 2);

%Filling cell array with energies calculated by shooting method
C(2, 3) = num2cell(E_sh_2);
C(2, 2) = num2cell(E_sh_1);
C(2, 4) = num2cell(E_sh_3);

%Graphics customization
legend('\psi_{n = 1}', '\psi_{n = 2}', '\psi_{n = 3}', 'Location', 'southwest');
title('Shooting method');
grid on;
xlabel('x, nm');
ylabel('\psi, nm^{-0.5}');
xlim([0, L]);

%Dependence of the squared modulus of the wave function on the coordinate
subplot(2, 3, 4);

%First energy level
plot(xp, abs(psi_sh_1) .^2, 'LineWidth', 2);
hold on;

%Second energy level
plot(xp, abs(psi_sh_2) .^2, 'LineWidth', 2);

%Third energy level

```

```

plot(xp, abs(psi_sh_3) .^2, 'LineWidth', 2);

%Graphics customization
title('Shooting method');
grid on;
xlabel('x, nm');
ylabel('| $\psi$ |^2, nm^{-1}');
xlim([0, L]);
ylim([0, 2.1]);

```

FDM

```

%Number of mesh nodes
N = 1000;

%Coefficient
t = hbar ^ 2 / ((L / (N + 1)) ^ 2 * 2 * m0);

%Compiling the Hamiltonian
v_1 = 2 * t * ones(1, N);
v_2 = -t * ones(1, N - 1);
H = diag(v_1) + diag(v_2, 1) + diag(v_2, -1);

%Finding eigenfunctions and eigenvalues of the Hamiltonian
[eigenvectors, eigenvalues] = eig(H);

%First energy level
psi_fdm_1 = -eigenvectors(:, 1)' / sqrt(L / (N + 1));
%Second energy level
psi_fdm_2 = -eigenvectors(:, 2)' / sqrt(L / (N + 1));
%Third energy level
psi_fdm_3 = -eigenvectors(:, 3)' / sqrt(L / (N + 1));

```

```

%Filling cell array with analytically calculated energies
C(3, 2) = num2cell(eigenvalues(1, 1) * J2eV * 1e9 ^ 2);
C(3, 3) = num2cell(eigenvalues(2, 2) * J2eV * 1e9 ^ 2);
C(3, 4) = num2cell(eigenvalues(3, 3) * J2eV * 1e9 ^ 2);

%Dependence of the wave function on the coordinate
subplot(2, 3, 2);

%First energy level
plot(xp, psi_fdm_1 , 'LineWidth', 2);
hold on;

%Second energy level
plot(xp, psi_fdm_2 , 'LineWidth', 2);

%Third energy level
plot(xp, psi_fdm_3 , 'LineWidth', 2);

%Graphics customization
title('Finite difference method');
legend('\psi_{n = 1}', '\psi_{n = 2}', '\psi_{n = 3}', 'Location', 'southwest');
grid on;
xlabel('x, nm');
ylabel('psi, nm^{-0.5}');
xlim([0, L]);

%Dependence of the squared modulus of the wave function on the coordinate
subplot(2, 3, 5);

%First energy level
plot(xp, abs(psi_fdm_1) .^2, 'LineWidth', 2);

```

```

hold on;

%Second energy level
plot(xp, abs(psi_fdm_2) .^2, 'LineWidth', 2);

%Third energy level
plot(xp, abs(psi_fdm_3) .^2, 'LineWidth', 2);

%Graphics customization
title('Finite difference method');
grid on;
xlabel('x, nm');
ylabel('| $\psi|^2$ , nm-1');
xlim([0, L]);
ylim([0, 2.1]);

```

Difference

```

%Abs difference between psi_sh and psi_an
subplot(2, 3, 3)

%psi_sh_1 and psi_fdm_1
plot(xp, abs(psi_fdm_1 - psi_sh_1), 'LineWidth', 2);
hold on;
%psi_sh_2 and psi_fdm_2
plot(xp, abs(psi_fdm_2 - psi_sh_2), 'LineWidth', 2);

%Graphics customization
title('Difference');
legend('| $\psi_{\text{fdm}}^{n=1}$  -  $\psi_{\text{shoot}}^{n=1}$ |', ...
      '| $\psi_{\text{fdm}}^{n=2}$  -  $\psi_{\text{shoot}}^{n=2}$ |', ...
      'Location', 'northwest');
grid on;

```

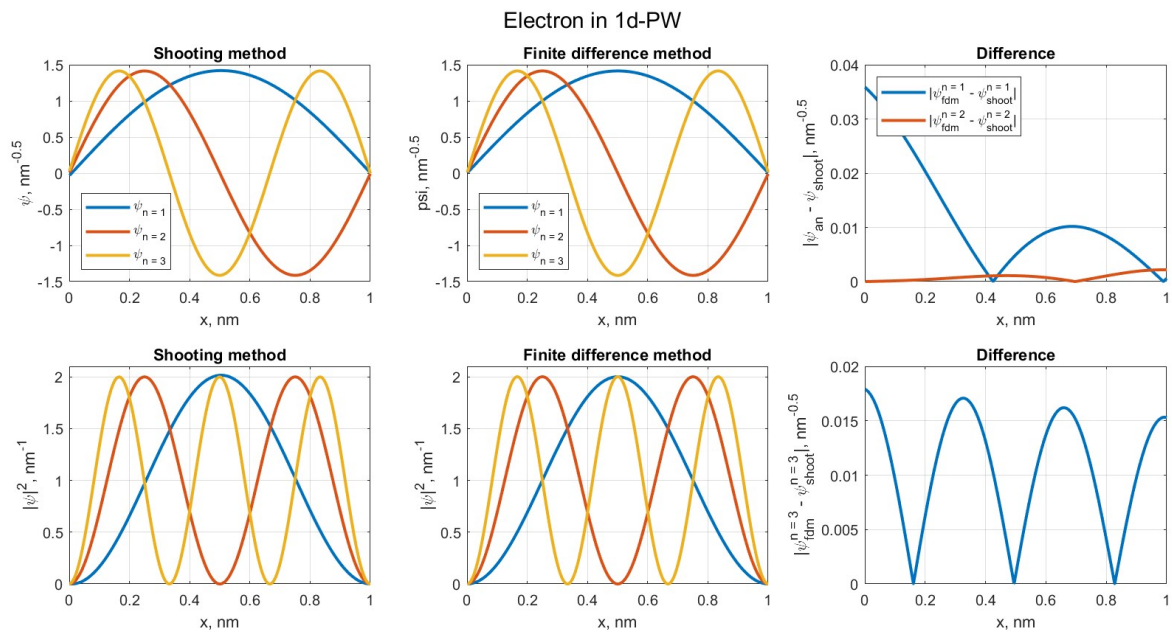
```

xlabel('x, nm');
ylabel('|ψan - ψshoot|, nm-0.5');
xlim([0, L]);
sgtitle('Electron in 1d-PW');

subplot(2, 3, 6)
%psi_sh_3 and psi_an_3
plot(xp, abs(psi_fdm_3 - psi_sh_3), 'LineWidth', 2);
hold on;

%Graphics customization
title('Difference');
grid on;
xlabel('x, nm');
ylabel('|ψfdmn = 3 - ψshootn = 3|, nm-0.5');
xlim([0, L]);
sgtitle('Electron in 1d-PW');

```



```

%Outputting filled cell array

```

C

C = 3×4 cell

	1	2	3	4
1	'n'	1	2	3
2	'E_shoot, eV'	0.382	1.504	3.381
3	'E_fdm, eV'	0.376	1.5042	3.3844

## 6 Возможная оптимизация реализации метода

1. Адаптивный размер шага  $dE$ : можно увеличивать / уменьшать значение  $dE$  в зависимости от того, на сколько  $ps_{final}$  отличается от допуска точности решения  $tolerance$ .

2. Ранняя остановка: если  $ps_{final}$  существенно не меняется от одной итерации к другой, то будет оптимальней раньше выйти из цикла.

Результат оптимизации на примере 5-го энергетического уровня:

Первоначальная функция: Elapsed time is 0.033320 seconds.

Оптимизированная функция: Elapsed time is 0.011614 seconds.

Листинг 4 - оптимизация функции Shooting

```
clc, clear, close all
datetime('now')

%Planck's constant in J * sec
hbar = 1.0546e-34;

%Mass of electron in kg
m0 = 9.1094e-31;

%Conversion constant from joules to electronvolts
J2eV = 6.2415e18;

% Pit length in nm
L = 1;

%Potential in eV
U = 0;

%Principal quantum number
n = 5;
```

```

%Energy of the fifth energy level
E_an = pi ^ 2 * n ^ 2 * hbar ^ 2 / (2 * m0 * (L * 1e-9) ^2) * J2eV;

%Energies close to the analytically calculated energy
E_apr = 0.8 * E_an; %eV

%Dependence of the wave function on the coordinate
subplot(2, 1, 1);

%Fifth energy level
tic
[xp, psi_sh_1, E_sh] = shooting(E_apr, U, L, 0.001, 0.0001);
toc
plot(xp, psi_sh_1, 'LineWidth', 2);

```

Elapsed time is 0.033320 seconds.

```

%Graphics customization
grid on;
xlabel('x, nm');
ylabel('\psi, nm^{-0.5}');
xlim([0, L]);

%Dependence of the wave function on the coordinate
subplot(2, 1, 2);

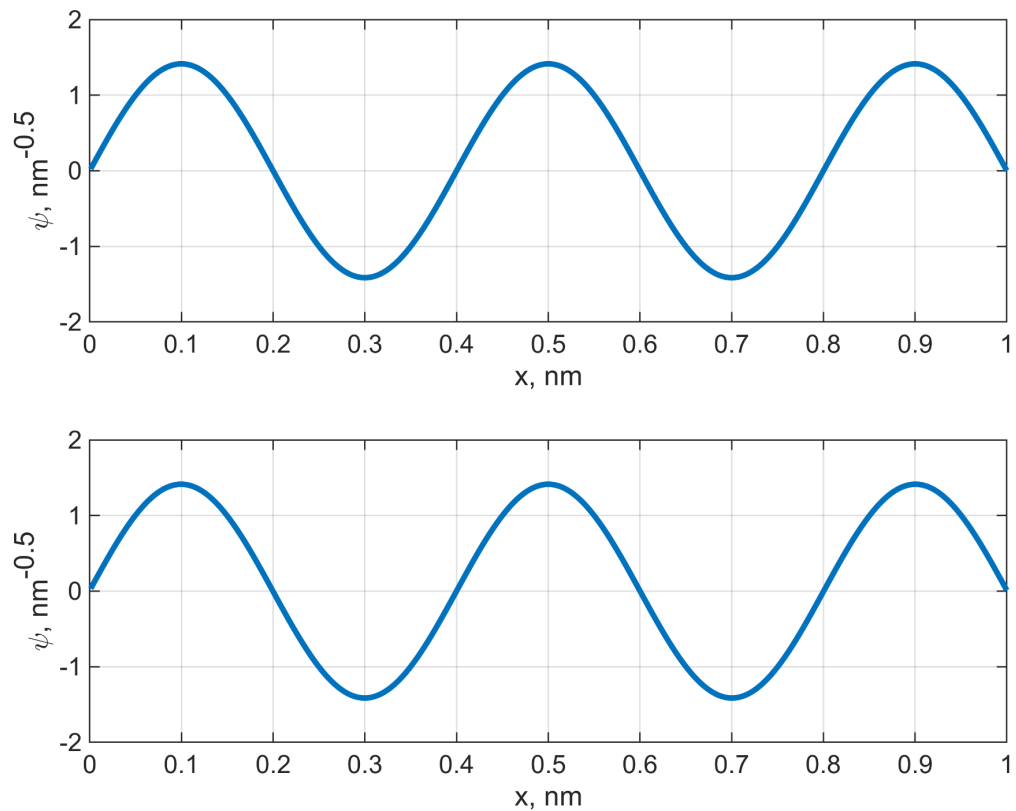
%Fifth energy level
tic
%Using modified function
[xp, psi_sh_1, E_sh] = shooting_modified(0.8 * E_apr, U, L, 0.001, 0.0001);
toc
plot(xp, psi_sh_1, 'LineWidth', 2);

```

Elapsed time is 0.011614 seconds.



```
%Graphics customization
grid on;
xlabel('x, nm');
ylabel('\psi, nm^{-0.5}');
xlim([0, L]);
```



```
function [xp, psip, E] = shooting_modified(E, U, L, dx, tolerance)
    %hbar * c [eV * nm]
    hbc = 1.0546e-34 * 6.2415e18 * 3e17;
    %mass of electron in mc^2
    m = 510998.95;
    %Coefficient in the Schrodinger equation
    k = 2 * m / hbc ^2;

    %Increasing Energy to Satisfy the Second Boundary Condition
```

```

dE = 0.001; %eV

%Vector for storing the coordinate
xp = dx : dx : L;

%First boundary condition
psi = 0;
%The last point of the wave function should be approximately equal to zero
%We assume that it is not equal to 0
psifinal = 1;

%Loop until psifinal is close to zero
while abs(psifinal) > tolerance
    %temporary variable to hold the old value of psifinal
    tmp = psifinal;

    %First boundary condition
    dpsi = 1;

    %Vector for storing value of the wave function
    psip = zeros(1, numel(xp));

    %Loop until we reach the end of the pit
    for i = 1 : L / dx
        %Schrodinger equation
        ddpsi = -k * psi * (E - U);

        %Representing the derivative as a limit
        dpsi = dpsi + ddpsi * dx;
        psi = psi + dpsi * dx;

        %Filling vector

```

```

        psip(i) = psi;
    end

    %Assigning the last value of the wave function to psifinal
    psifinal = psi;

    %If psifinal does not change significantly, then exit from the loop
    if (abs(psifinal - tmp)) < 1e-5
        break;
    end

    %Increasing the energy to the desired value only
    % if the second boundary condition is not met
    if (abs(psifinal) / tolerance >= 5)
        E = E + 4 * dE;
    elseif (abs(psifinal) / tolerance >= 2)
        E = E + 2 * dE;
    elseif (abs(psifinal) / tolerance >= 1.2)
        E = E + dE / 4;
    end
end

%Normalization
area = sum(psip .^ 2) * dx;

psip = psip / sqrt(area);
end

```

## Список литературы

1. Метод Рунге-Кутты URL: [Лекция 15. Метод Рунге-Кутты](#) (дата обращения 18 июня 2023).
2. Метод стрельбы URL: [The Shooting Methods](#) (дата обращения 18 июня 2023).
3. Luke Taylor. A comparison between the Shooting and Finite-Difference Method in solving a Nonlinear Boundary Value Problem found in the context of light propagation. URL: <https://arxiv.org/pdf/1709.04795.pdf> (дата обращения 18 июня 2023).
4. Marie Christine Ertl. Solving The Stationary One Dimensional Schrödinger Equation With The Shooting Method. URL: [Solving The Stationary One Dimensional Schrödinger Equation With The Shooting Method](#) (дата обращения 18 июня 2023).