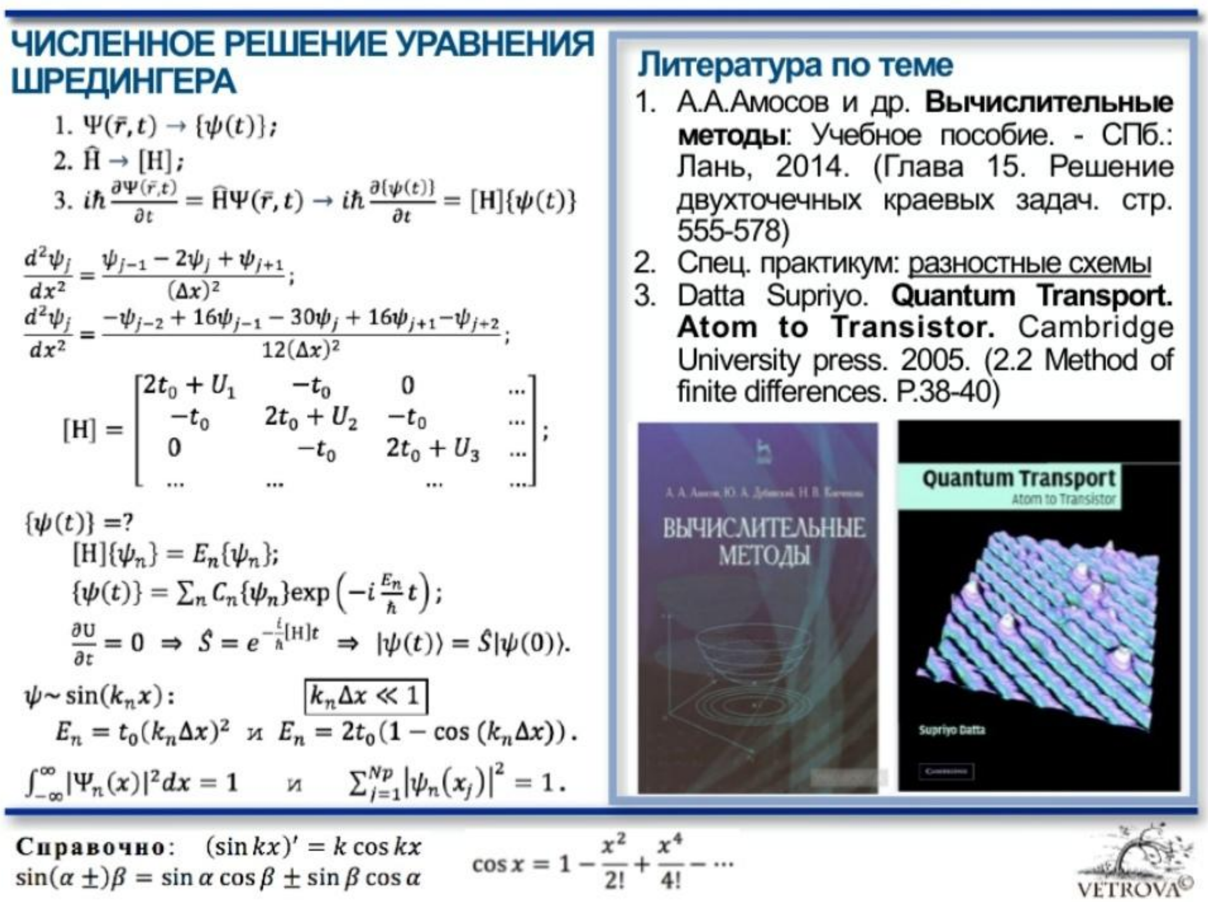
Филимонов Степан РЛ6-31

Лабараторная номер 5

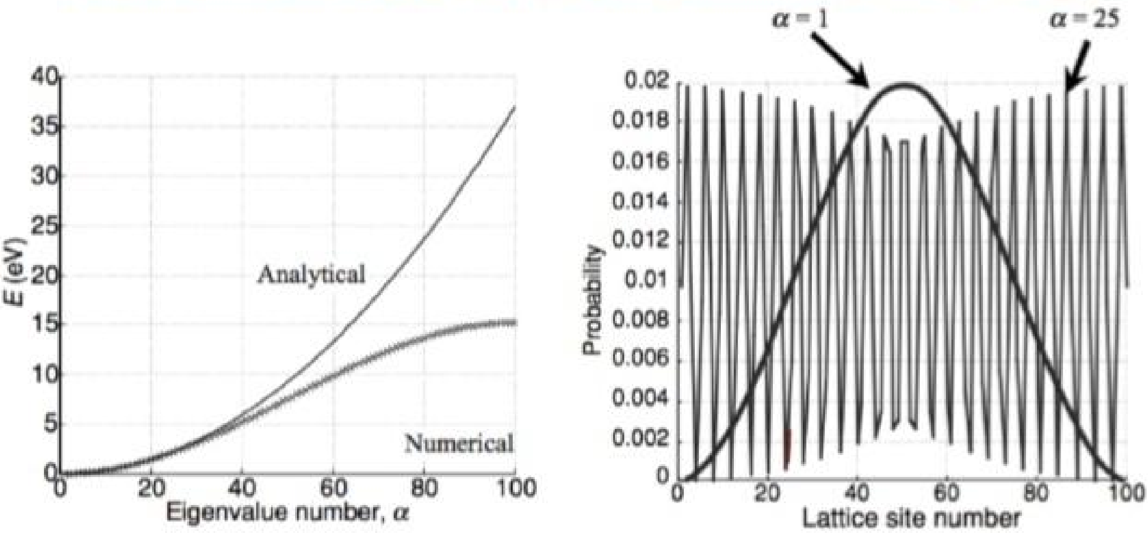
Задача:

Необходимо отрисовать для электрона локализованного в одномерной бесконечно глубокой яме 10 нм график пси функции от координаты (для основного состояния и 24-го возбужденного). Сравнить графики полученный на основе аналитического решения уравнения Шрёдингера с результатами работы встроенной MATLAB-функции eig.

Фото задания:



Частица массы $m0$ в ПЯ L = 101 А (дискретизация: Np = 100, Δx=1A).



	E1	E2	E3	E25
Ан, эВ	0.003 7	0.014 8	0.033 2	2.306 9
ΔE, мэВ	ΔE12= 11	ΔE23=19	...	kT=26мэВ

t0 = 3.815 эВ

Теория:

Квантовая яма с бесконечными стенками — область пространства размером порядка длины волны де Бройля рассматриваемой частицы (хотя бы в одном направлении), вне которой потенциальная энергия *U* бесконечна. Иногда данную область называют «ящиком»

Двухточечная краевая задача - это задача отыскания решения обыкновенного дифференциального уравнения или системы обыкновенных дифференциальных уравнений на отрезке $a \le x \le б$, в которой дополнительные условия на решение налагаются в двух точках а и б - "краях" отрезка (отсюда - и название задачи).

Решение:

Численное решение уравнения Шредингера(стационарного)

$$-\frac{\hbar}{2m}\Delta\psi(x) + U(x)\psi(x) = E\psi(x)$$

1. $\Psi(\bar{r}, t) \to \{\psi(t)\}$
2. $\hat{\mathrm{H}} \to [\mathrm{H}]$;

3. $i\hbar\frac{\partial}{\partial t}\Psi(\bar{r},t) = \widehat{\mathrm{H}}\Psi(\bar{r},t) \to i\hbar\frac{\partial}{\partial t}\{\psi(t)\} = [\mathrm{H}]\{\psi(t)\}$

Из

$$\begin{cases} \psi_{i+1} = \psi_i + \Delta x\frac{d}{\mathrm{d}x}\psi_i + \frac{\Delta x^2}{2!}\frac{d^2}{\mathrm{d}x^2}\psi_i + \frac{\Delta x^3}{3!}\frac{d^3}{\mathrm{d}x^3}\psi_i + \frac{\Delta x^4}{4!}\frac{d^4}{\mathrm{d}x^4}\psi_i + \dots \\ \psi_{i-1} = \psi_i - \Delta x\frac{d}{\mathrm{d}x}\psi_i + \frac{\Delta x^2}{2!}\frac{d^2}{\mathrm{d}x^2}\psi_i - \frac{\Delta x^3}{3!}\frac{d^3}{\mathrm{d}x^3}\psi_i + \frac{\Delta x^4}{4!}\frac{d^4}{\mathrm{d}x^4}\psi_i + \dots \end{cases}$$

Получим:

$\frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{\Delta x^2} = \frac{d^2}{\mathrm{d}x^2}\psi_i \Longrightarrow$ это трехточечная апроксимация оператора 2-ой производной , где

$\mathrm{dx} = \frac{L}{N_p + 1}$

Подставим его:

$-\frac{\hbar}{2m}\left[\frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{\Delta x^2}\right] + U_i\psi_i = E_i\psi_i$

Матрицей этого уравнения будет

$[H] = \begin{bmatrix} 2x_0 + U_1 & -x_0 & 0 & \dots \\ -x_0 & 2x_0 + U_2 & -x_0 & \dots \\ 0 & -x_0 & 2x_0 + U_3 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$, где x_0это константанта, равная $x_0 = \frac{\hbar^2}{2m} * \frac{1}{\mathrm{dx}}, U_i$ - матрица увеличивающая высоту, в нашей задаче = 0, по этому при вычислении учитывотся не будет

Аналитическое решение $\Upsilon = \sin\left(\pi\frac{\mathrm{condition}}{L}x\right)$, следует из $k = \frac{\mathrm{condition} * \pi}{L}$ и $\psi(x) = \mathrm{A}\sin(\mathrm{kx})\mathrm{dx}$, в нашем случае $A = \sqrt{\frac{2}{L}}$.

Энергию состояния, описываемого волновой функцией $\psi_n = C_n\sin\left(n\frac{\pi x}{L}\right)$, получим подставив $k_n = n\frac{\pi}{L}$ в $E = \frac{\hbar}{2m}k^2$:

$E_n = \frac{n^2\hbar^2\pi^2}{2mL^2}$

,где *n* - номер орбитали, $\hbar$ - постоянная Планка-Дирка, *m* - масса частицы (по условию $m = meff * m_0$, *L* - ширина потенциальной ямы(нм).

$\Delta E = E_{n+1} - E_n$.

$E_n = \frac{n^2\hbar^2\pi^2}{2mL^2}$; $\mathrm{kL} = \pi n$(плотность состояния) из этого следует, что $\Delta E_{n,n+1} = \frac{\hbar^2\pi^2}{2mL^2}((n+1)^2 - n^2) = \mathrm{kT}$,

$((n+1)^2 - n^2)$ = 3, т.к. n = 1. Тогда

$$L = \hbar\pi\sqrt{\frac{3}{2m * k * T}}$$

Код Matlab:

```
cla reset;
hold on;
L = 101 * 1e-10;
[Phk, En, x0] = matrixSchrodinger(L); %*(см. ниже)
```

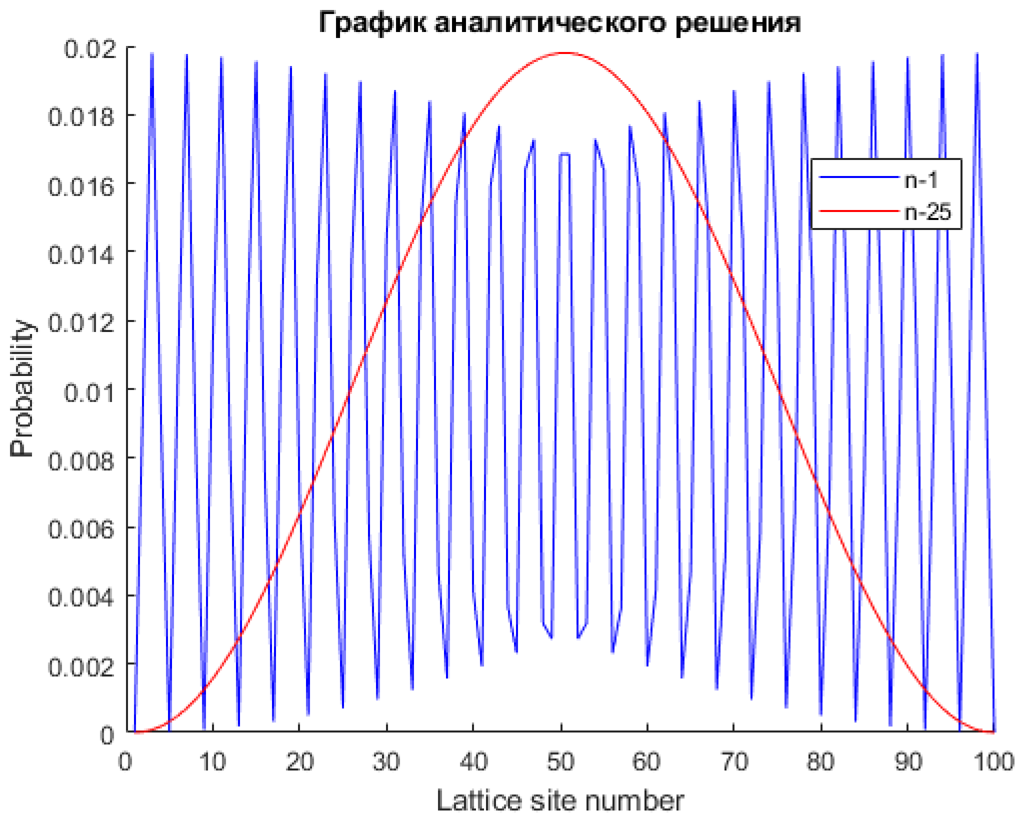
- **Задание 1:** получение нормы энергии

```
normal_energy = norma_energy_get(Phk)
```

```
normal_energy = 100×1
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
      ⋮
```

- **Задание 2:** построение графиков аналитического решения и результата вычесления встроенной MATLAB-функции eig

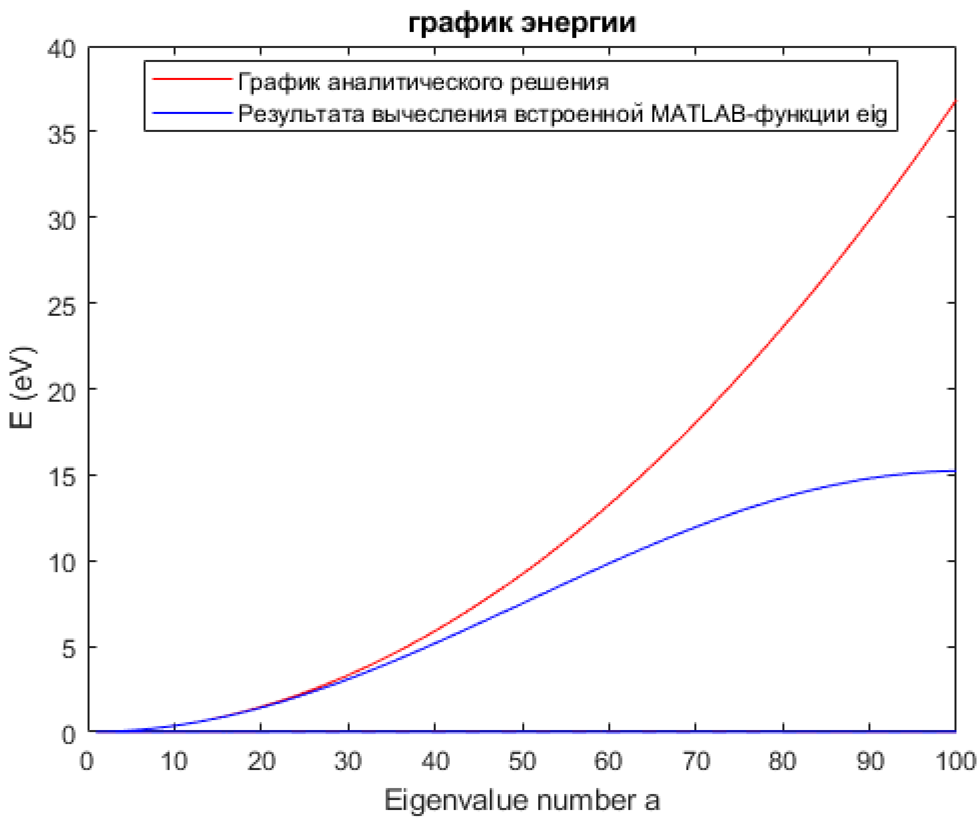
```
Xn = 1:100;
plot(Xn, Ytheoretically(25, L),"Color",'blue');
plot(Xn, Ytheoretically(1, L),"Color",'red'); %*(см. ниже)
title('График аналитического решения');
legend('n-1','n-25','Location',"best");
xlabel('Lattice site number');
ylabel('Probability');
hold off;
```




```
plot(Xn, abs(Phk(1:end, 1)).^2,"Color",'red');
hold on;
plot(Xn, abs(Phk(1:end, 25)).^2,"Color",'blue');
title('График результата вычесления встроенной MATLAB-функции eig');
legend('n-1','n-25','Location',"best");
xlabel('Lattice site number');
ylabel('Probability');
```

- **Задание 3:** построение энергитических графиков аналитического решения и результата вычесления встроенной MATLAB-функции eig

```
Et = Etheoretically(n, L); %*(см. ниже)
plot(n, Etheoretically(n, L),"Color",'red');
hold on;
plot(n, En,"Color",'blue');
title('график энергии')
legend('График аналитического решения','Результата вычесления встроенной MATLAB-функции eig',"
xlabel('Eigenvalue number a');
ylabel('E (eV)')
hold off;
```



- **Задание 4:** таблица

```
load("constans.mat","J2eV","kT");
ee = [Et(1) Et(2) Et(3) Et(25)]
```

```
ee = 1×4
    0.0037    0.0147    0.0332    2.3039
```

```
Et = Et.*1e3;
dE=[(Et(2)-Et(1)) (Et(3)-Et(2)) (Et(4)-Et(3)) kT*10^3]
```

```
dE = 1×4
   11.0586   18.4310   25.8035   25.8520
```

```
dE = int8(dE)
```

```
dE = 1×4 int8 row vector
   11   18   26   26
```

```
x0 = x0 * J2eV
```

```
x0 = 3.8100
```

- **Время:**

```
datetime('now')
```

```
ans = datetime
   02-Dec-2021 11:46:03
```

*Функция создания графика:

```
function yt = Ytheoretically(n, L)
 Np = 100;
 dx = L / (Np + 1);
 r = 2 / L;
 X = linspace(0, L, 100);
 yt = r * dx * abs(sin((pi * n / L)  * X)).^2;
end

function et = Etheoretically(n, L)
 load("constans.mat","m0","hbar","J2eV");
 et = (pi * hbar * n / L).^2/(2 * m0) * J2eV;
end

function [Y, En, x0] = matrixSchrodinger(L)
 load("constans.mat","m0","hbar","J2eV");
 Np = 100;
 dx = L / (Np + 1);
 x0 = (hbar ^ 2) / (2 * m0 * dx ^ 2);
 H = zeros(Np);
 H(1, 1) = 2 * x0;
 H(1, 2) = -x0;
 for r = 2:Np
   for c = 1:Np
     if and(and(r ~= 1, r ~= Np),r == c)
       H(r, c) = 2 * x0;
       H(r, c - 1) = -x0;
       H(r, c + 1) = -x0;
     end
   end
 end
 H(Np, Np) = 2 * x0;
 H(Np, Np - 1) = -x0;
 [Y, En] = eig(H); % вернем диагональную матрицу
 En = diag(En) * J2eV;
end

function ne = norma_energy_get(Phk)
 ne = 0;
 for c = 1:100
    ne = ne + abs(Phk(1:end, c)).^2;
 end
end
```