

Построить график решения двух точечной краевой задачи при $L = 10$ нм.

Фото задания:

ЧИСЛЕННОЕ РЕШЕНИЕ УРАВНЕНИЯ ШРЕДИНГЕРА

1. $\Psi(\vec{r}, t) \rightarrow \{\psi(t)\};$
2. $\hat{H} \rightarrow [H];$
3. $i\hbar \frac{\partial \Psi(\vec{r}, t)}{\partial t} = \hat{H} \Psi(\vec{r}, t) \rightarrow i\hbar \frac{\partial \{\psi(t)\}}{\partial t} = [H]\{\psi(t)\}$

$$\frac{d^2 \psi_j}{dx^2} = \frac{\psi_{j-1} - 2\psi_j + \psi_{j+1}}{(\Delta x)^2};$$

$$\frac{d^2 \psi_j}{dx^2} = \frac{-\psi_{j-2} + 16\psi_{j-1} - 30\psi_j + 16\psi_{j+1} - \psi_{j+2}}{12(\Delta x)^2};$$

$$[H] = \begin{bmatrix} 2t_0 + U_1 & -t_0 & 0 & \dots \\ -t_0 & 2t_0 + U_2 & -t_0 & \dots \\ 0 & -t_0 & 2t_0 + U_3 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix};$$

$\{\psi(t)\} = ?$

$$[H]\{\psi_n\} = E_n\{\psi_n\};$$

$$\{\psi(t)\} = \sum_n C_n \{\psi_n\} \exp\left(-i \frac{E_n}{\hbar} t\right);$$

$$\frac{\partial U}{\partial t} = 0 \Rightarrow \hat{S} = e^{-\frac{i}{\hbar} [H] t} \Rightarrow |\psi(t)\rangle = \hat{S} |\psi(0)\rangle.$$

$\psi \sim \sin(k_n x):$

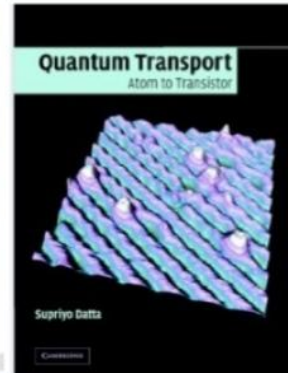
$$E_n = t_0 (k_n \Delta x)^2 \quad \text{и} \quad E_n = 2t_0 (1 - \cos(k_n \Delta x)).$$

$$\int_{-\infty}^{\infty} |\Psi_n(x)|^2 dx = 1 \quad \text{и} \quad \sum_{j=1}^{Np} |\psi_n(x_j)|^2 = 1.$$

Справочно: $(\sin kx)' = k \cos kx$
 $\sin(\alpha \pm \beta) = \sin \alpha \cos \beta \pm \sin \beta \cos \alpha$
 $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$

Литература по теме

1. А.А.Амосов и др. **Вычислительные методы:** Учебное пособие. - СПб.: Лань, 2014. (Глава 15. Решение двухточечных краевых задач. стр. 555-578)
2. Спец. практикум: разностные схемы
3. Datta Supriyo. **Quantum Transport. Atom to Transistor.** Cambridge University press. 2005. (2.2 Method of finite differences. P.38-40)



Теория:

Двухточечная краевая задача - это задача отыскания решения обыкновенного дифференциального уравнения или системы обыкновенных дифференциальных уравнений на отрезке $a \leq x \leq b$, в которой дополнительные условия на решение налагаются в двух точках a и b - "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. $\Psi(r, t) \rightarrow \{\psi(t)\}$
2. $\hat{H} \rightarrow [H]$;
3. $i\hbar \frac{\partial}{\partial t} \Psi(r, t) = \hat{H} \Psi(r, t) \rightarrow i\hbar \frac{\partial}{\partial t} \{\psi(t)\} = [H] \{\psi(t)\}$

Из

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

Получим:

$$\frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{\Delta x^2} = \frac{d^2}{dx^2} \psi_i \Rightarrow \text{это трехточечная аппроксимация оператора 2-ой производной, где}$$

$$\Delta x = \frac{L}{N_p + 1}$$

Подставим его:

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

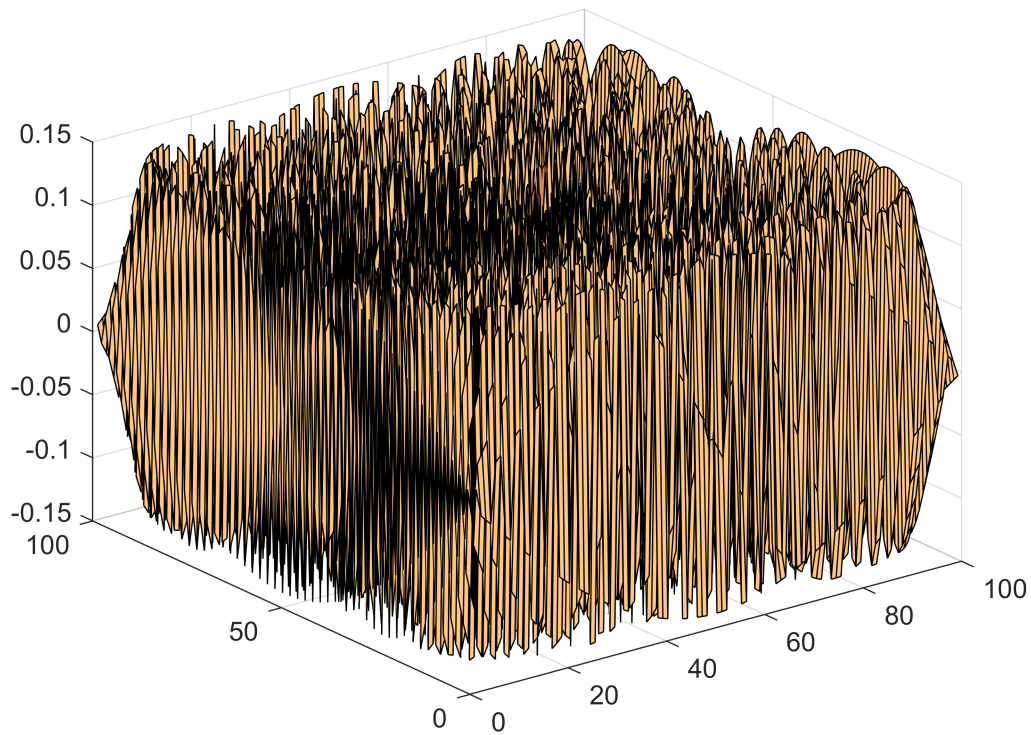
Матрицей этого уравнения будет

$$[H] = \begin{bmatrix} 2x_0 + U_1 & -x_0 & 0 & \dots \\ -x_0 & 2x_0 + U_2 & -x_0 & \dots \\ 0 & -x_0 & 2x_0 + U_3 & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}, \text{ где } x_0 \text{ это константа, равная } x_0 = \frac{\hbar^2}{2m} * \frac{1}{\Delta x}, U_i - \text{ матрица}$$

увеличивающая высоту, в нашей задаче = 0, по этому при вычислении учтываются не будет

Код Matlab:

```
cla reset;
L = 10;
Np = 100;
[X, Y] = matrixSchrodinger(L, Np); %*(см. ниже)
surf(X, Y);
colormap("copper");
```



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

```
ans = datetime
      17-Oct-2021 11:58:53
```

*Функция создания графика:

```
function [X, Y] = matrixSchrodinger(L, Np)
load('constans.mat','hbar',"m0");
x0 = hbar^2 / (2 * m0 * ((L * 1e-9) / (Np + 1)));
H = zeros(Np); % заполним матрицу нулями
for r = 1:Np
    for c = 1:Np
        if r == 1
            if r == c
                H(r, c) = -2 * x0;
                H(r, c + 1) = x0;
            end
        end
        if and(r ~= 1, r ~= Np )
            if r == c
                H(r, c) = -2 * x0;
                H(r, c - 1) = x0;
                H(r, c + 1) = x0;
            end
        end
        if r == Np
```

```
        if r == c
            H(r, c) = -2 * x0;
            H(r, c - 1) = x0;
        end
    end
end
[X, Y] = eig(H); % вернем диагональную матрицу
end
```