

FINITE-DIFFERENCE TIME-DOMAIN (FDTD) METHOD

Simulation of the 1d time-dependent Schrodinger equation ('real-time, real-space' method)

$$i\hbar \frac{\partial \Psi(x, t)}{\partial t} = -\frac{\hbar^2}{2m} \Delta \Psi(x, t) + U(x) \Psi(x, t)$$

$$\Psi(x, t) = \Psi(n \cdot \Delta x, m \cdot \Delta t) = \Psi^m(n)$$

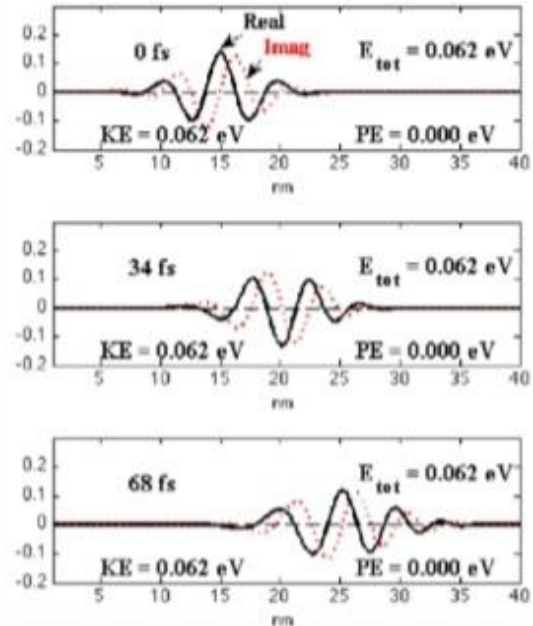
Leapfrogging technique между Re и Im членами:

$$\begin{aligned} \Psi_{Real}^{m+1}(n) &= \Psi_{Real}^m(n) \\ &- \frac{\hbar}{2m} \frac{\Delta t}{(\Delta x)^2} \left[\Psi_{Im}^{m+\frac{1}{2}}(n-1) - 2\Psi_{Im}^{m+\frac{1}{2}}(n) \right. \\ &\quad \left. + \Psi_{Im}^{m+\frac{1}{2}}(n+1) \right] + \frac{\Delta t}{\hbar} U(n) \Psi_{Im}^{m+\frac{1}{2}}(n); \\ \Psi_{Im}^{m+\frac{3}{2}}(n) &= \Psi_{Im}^{m+\frac{1}{2}}(n) \\ &+ \frac{\hbar}{2m} \frac{\Delta t}{(\Delta x)^2} \left[\Psi_{Real}^{m+1}(n-1) - 2\Psi_{Real}^{m+1}(n) \right. \\ &\quad \left. + \Psi_{Real}^{m+1}(n+1) \right] - \frac{\Delta t}{\hbar} U(n) \Psi_{Real}^{m+1}(n); \end{aligned}$$

$$\leq 0,15 \Leftrightarrow \text{Если } \Delta x = 0,1 \text{ нм} \Rightarrow \Delta t = 0,02e-15 = 0,02 \text{ фс.}$$

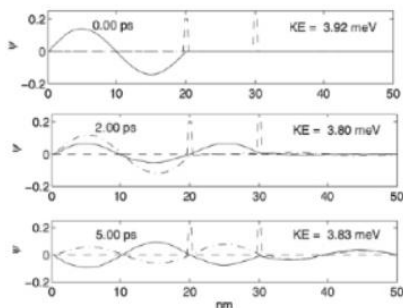
Упражнение

Моделирование движения частицы в свободном пространстве в положительном направлении оси x. Время: m = [1 1700 3400]



METALL-OXID-SEMICONDUCTOR FIELD-EFFECT TRANSISTOR

2. Инициализация частицы во 2-ом собственном состоянии левой ямы

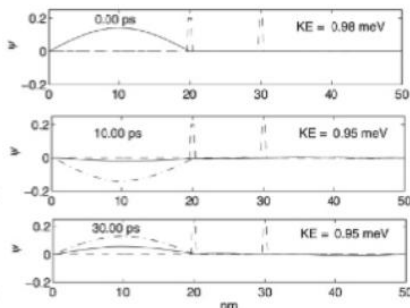


Строка 5: NN = 250;
Строка 13: dx = .2e-9;
Строка 14: dt = 1e-16;
Строка 22: V(0.4*NN-1:0.4*NN+1)=0.2*ev2J;
V(0.6*NN-1:0.6*NN+1)=0.2*ev2J; plot(V)

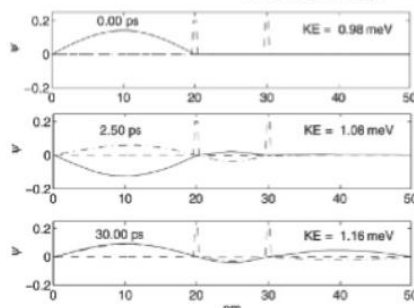
Строка 31: lambda=100;
pr(2:0.4*NN-1)=sin(2*pi*(2:0.4*NN-1)/lambda);
ptot = sum(prl.*2.*pim.*2);
Строки 32-36: ЗАКОММЕНТИРОВАНО
Строка 94: TT = text(5,15,sprintf('%7.2f ps',T*dt*1e12));
Строка 96: TT = text(5,-15,sprintf('KE = %5.2f meV', KE*1e3));

Временные точки: m = 1 10 000 50 000
0 ps 2 ps 5 ps
How many time steps: 1 20 000 30 000

3. Инициализация частицы в основном состоянии левой ямы:



4. Приложен потенциал 3 мэВ к средней яме (энергия основного состояния = 3.75-3=0.75 мэВ):

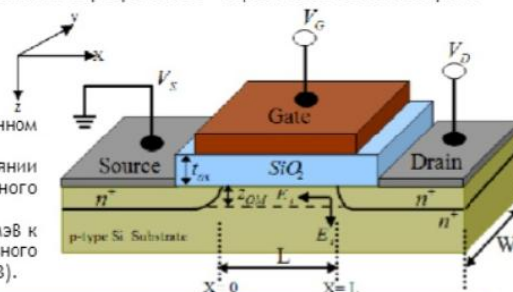


Размер MOSFETs требует квантово-механические модели!

1. Пусть V аппроксимирован тремя ПЯ. Барьеры 0.2 эВ > первых собственных энергий.
 $dx = 0.2$ нм, $NN=250$,
 $dt = 0.10$ фс = $1e-16$ с,

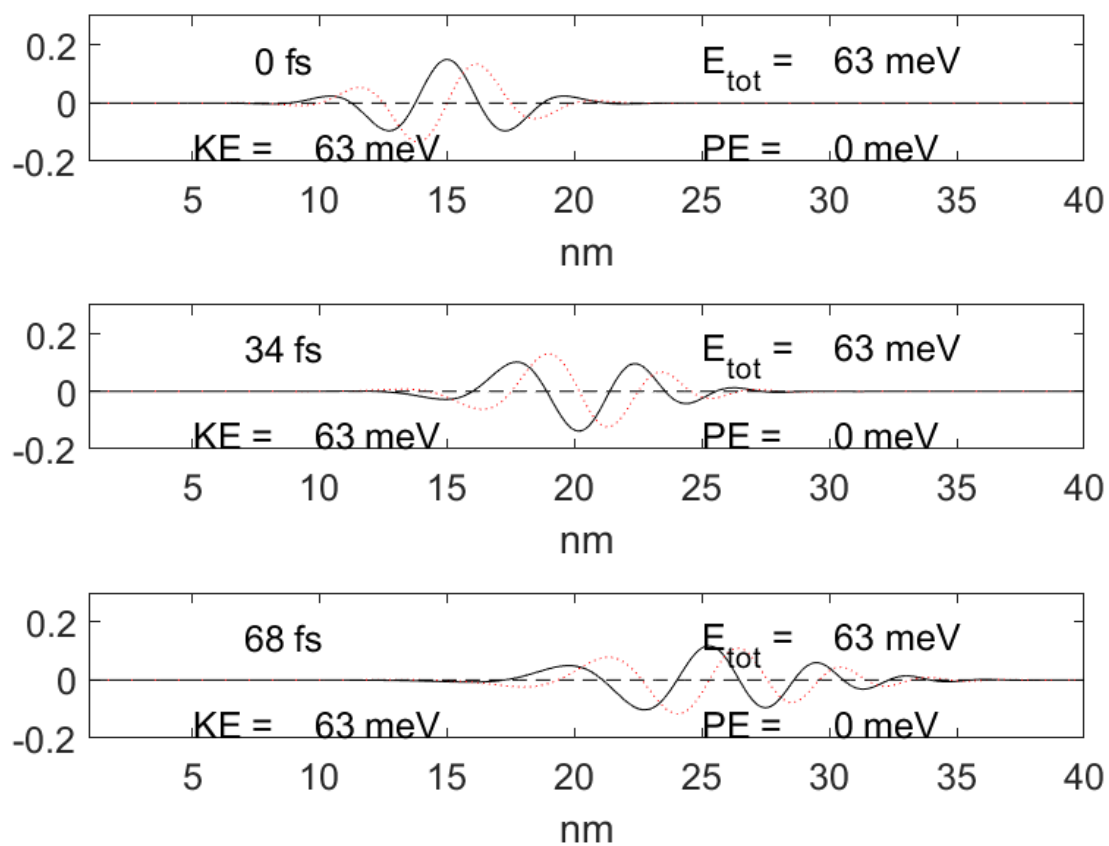
Какие частицы пройдут от истока к стоку?

1. частицы во втором собственном состоянии левой ямы;
2. частицы в основном состоянии левой ямы без приложенного напряжения на затворе;
3. частицы в основном состоянии средней ямы без приложенного напряжения на затворе;
4. если приложен потенциал 3 мэВ к средней яме (энергия основного состояния = $3.75 - 3 = 0.75$ мэВ).



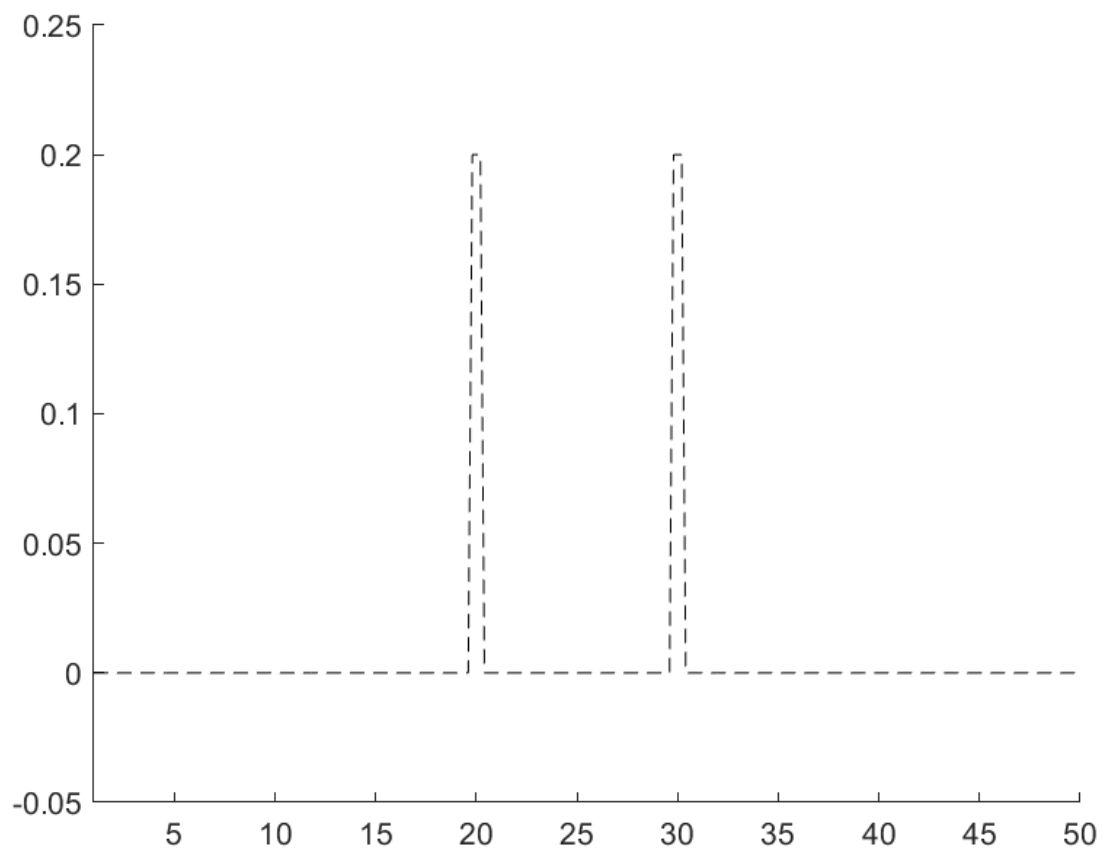
Решение:

Задание 1:

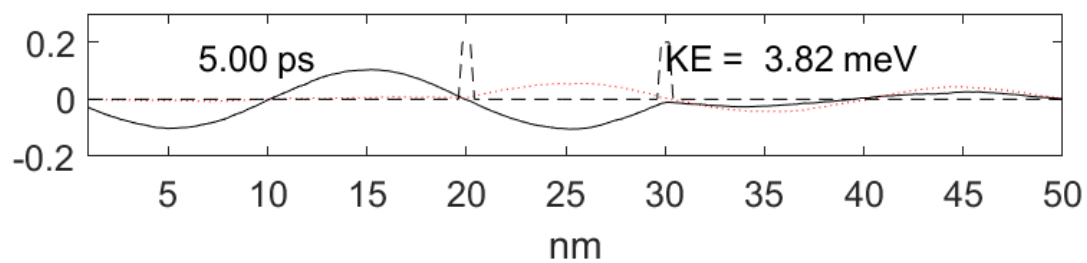
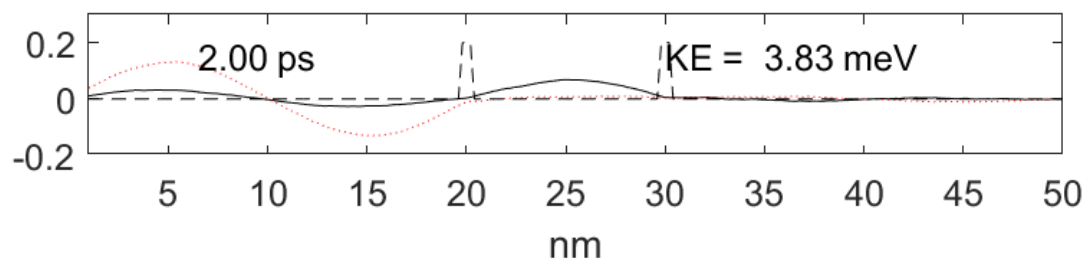
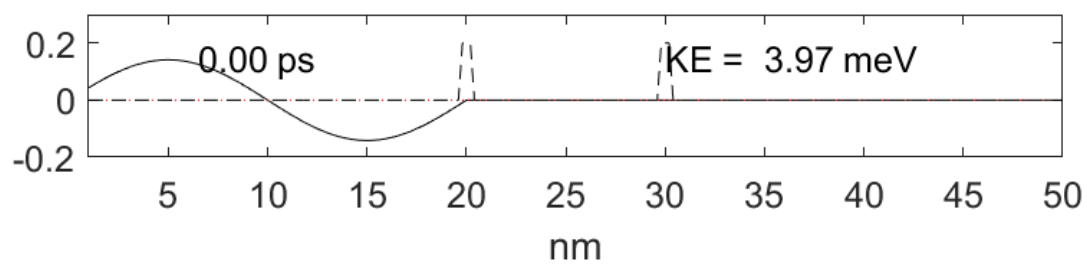


Задание 2:

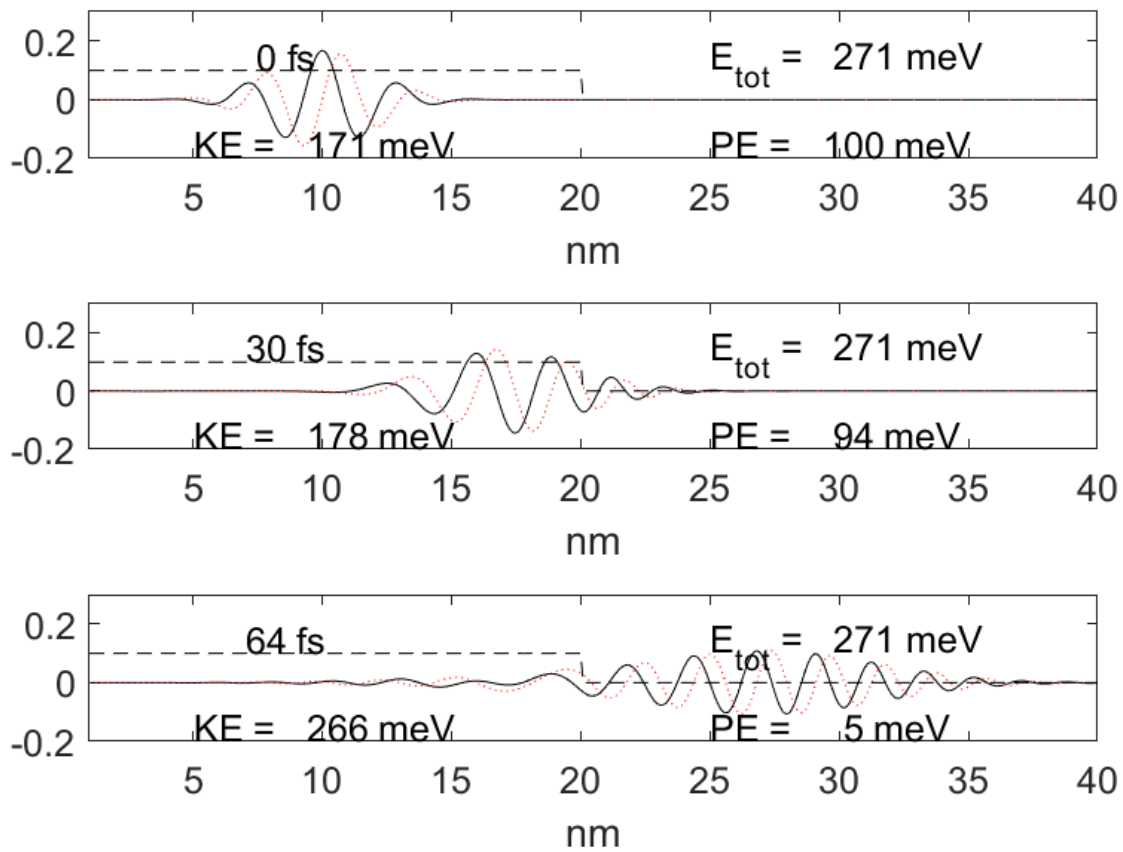
Затвор закрыт



Инициализация частицы во 2-ом собственном состоянии левой ямы



Тестовое задание:



Код Matlab:

```
cla reset;
hold on;
load("constans.mat");
```

• Константы и значения

```
i = sqrt(-1);
%Nr = 400; % Количество точек в проблемном пространстве Nr = 400;
Nr = 250;
T = 0;
Nstep=1;
time = [1 19999 30000]; % 1 1699 1700 % 1 19999 30000
meff = 1; % эффективная масс: Si is 1.08, Ge is 0.067, GaAs is 0.55
m = meff * m0; % Масса электрона
% dx = 0.1e-9; % Размер ячейки dx = 0.1e-9;
dx = 0.2e-9;
% dt = 2e-17; % Временные шаги dt = 0.2e-17;
dt = 1e-16;
```

$$ra = \frac{\hbar}{2m} * \frac{\Delta t}{(\Delta x)^2}$$

```
ra = (hbar / (2 * m)) * (dt / (dx)^2) % Коэффициент номер 1, должен быть меньше < 0.15
```

```
ra = 0.1447
```

ra < 0.15 -> возможна связь

```
DX = dx * 1e9; % Целое количество нм
```

```
XX = (1:Np) * DX; % Длина в нм для построения графика
```

```
Dsquared = (diag(ones(1,Np-1),1) - 2 * diag(ones(1,Np)) + diag(ones(1,Np-1),-1)); % диагональ
```

• Потенциал

```
V = zeros(1,Np)'; % 0..- => ' => 0|.. вертикальной сделает
```



```
nc_b1 = 2/5*Np;
```

```
nc_b2 = 3/5*Np;
```

```
% создание барьеров высотой 0.2эВ
```

```
V(nc_b1-1:nc_b1+1)=0.2*eV2J;
```

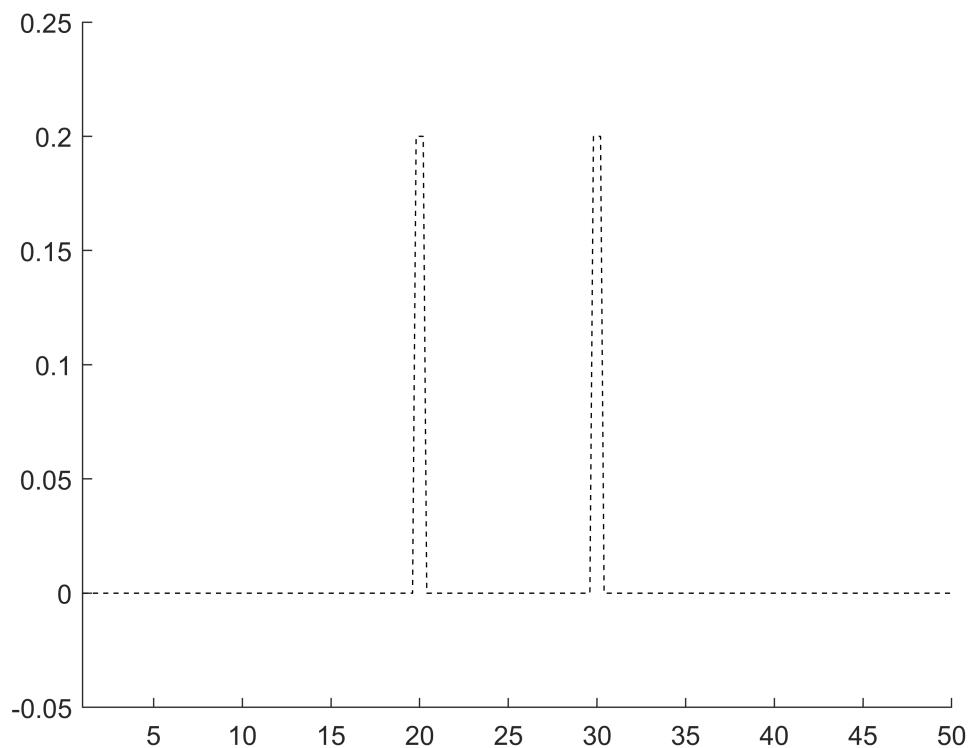
```
V(nc_b2-1:nc_b2+1)=0.2*eV2J;
```

```
% имитация напряжение на затворе
```

```
% V(nc_b1+2:nc_b2-2)=-0.003*eV2J;
```

```
axis([1 DX*Np -0.05 0.25])
```

```
plot(XX,J2eV*V, '--k')
```



```
% saveas(gcf, 'MOSFET_close.png')
```

- Инициализируем синусоидальную волну в гауссовой огибающей

$$P = e^{-\frac{(n - \text{исходное положение})^2}{2 \cdot \text{ширина импульса}^2}} * \left(\cos\left(2\pi \frac{n - \text{исходное положение}}{\text{длина волны}}\right) + i * \sin\left(2\pi \frac{n - \text{исходное положение}}{\text{длина волны}}\right) \right)$$

```
% ----- Задание 1 -----
% lambda = 50; % Длина волны импульса % для того чтобы была как в из усл. 50
% lamda_test = lambda * dx * 1e9
% sigma = 25; % Ширина импульса % для того чтобы была как в из усл. 25
% nc = 150; % для того чтобы была как в из усл. 150
% n=(1:Np)';
% Pulse = exp(-0.5*((n-nc)/sigma).^2).*(cos(2*pi*(n-nc)/lambda) + i*sin(2*pi*(n-nc)/lambda));
% PulseReal = real(Pulse); % Реальная часть импульса
% PulseImag = imag(Pulse); % Мнимая часть импульса
% ----- Задание 2 -----
lambda = 100;
PulseReal(2:nc_b1) = sin(2*pi*(2:nc_b1)/lambda);
PulseReal(nc_b1:Np) = 0;
PulseImag(1:Np) = 0;
% -----
PulseZ = PulseReal + i * PulseImag; % И мнимая и реальная часть импульса
```

- Нормируем её и проверяем

$$N = \frac{\vec{n}}{|\vec{n}|}$$

```
PulseNormal = sqrt(sum(PulseZ.*conj(PulseZ))); % Константа нормализации
PulseReal = PulseReal / PulseNormal;
PulseImag = PulseImag / PulseNormal;
PulseZ = PulseReal + i*PulseImag;
control = sqrt(sum(PulseZ.*conj(PulseZ)));
PulseNormal = sqrt(sum(PulseZ.*conj(PulseZ))) % Константа нормализации
```

```
PulseNormal = 1
```

• PDTD

```
for n_step = 1:3
```

• Главные вычисления

```
for mm=1:time(n_step)
    T = T + 1;
```

$$\Psi_{\text{Real}}(n) = \Psi_{\text{Real}}^m(n) - \frac{\hbar}{2m} * \frac{\Delta t}{(\Delta x)^2} * \Psi_{\text{Imag}}[(n-1) - 2 * (n) + (n+1)] + \frac{\Delta t}{\hbar} U(n) \Psi_{\text{Imag}}$$

```
PulseReal = PulseReal - ra*Dsqared*PulseImag + (dt/hbar)*V.*PulseImag;
```

$$\Psi_{\text{Imag}}(n) = \Psi_{\text{Imag}}^m(n) + \frac{\hbar}{2m} * \frac{\Delta t}{(\Delta x)^2} * \Psi_{\text{Real}}[(n-1) - 2 * (n) + (n+1)] - \frac{\Delta t}{\hbar} U(n) \Psi_{\text{Real}}$$

```
PulseImag = PulseImag + ra*Dsqared*PulseReal - (dt/hbar)*V.*PulseReal;
end
subplot(3,1,Nstep)
plot(XX,real(PulseReal),'k','Color','black')
hold on
plot(XX,real(PulseImag),':k','Color','red')
plot(XX,J2eV*V,'--k') % потенциальная энерги
hold off
axis([1 DX*Np -0.2 0.3])
xlabel('nm')
set(gca,'fontsize',12)
Nstep = Nstep + 1;
```

• Подписываем график

```
% ----- Проверяем нормализацию -----
PulseZ = PulseReal+i*PulseImag;
PulseZ'*PulseZ
PulseNormal = sqrt(sum(PulseZ.*conj(PulseZ)));
% ----- Вычислите ожидаемые значения -----
KE = J2eV * real(-(hbar/dx)^2/(2*m)*PulseZ'*(Dsqared*PulseZ))*1e3; % Кинетическая энергия
```

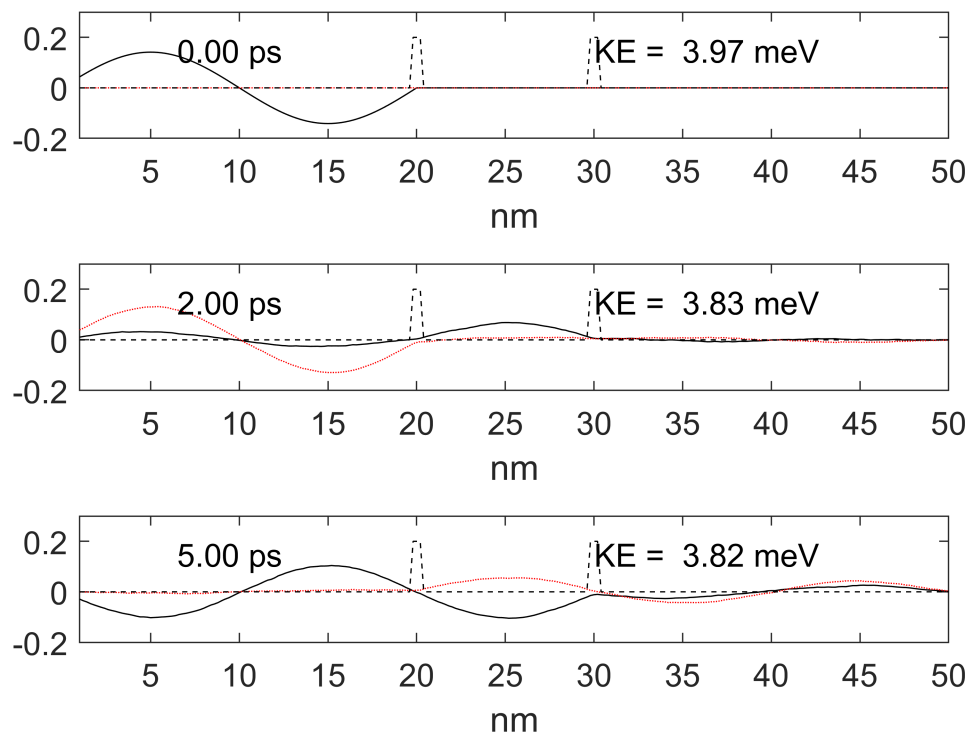
```
PE = J2eV * PulseZ' * (V.*PulseZ)*1e3; % Потенциальная энергия
TT = text(5,.15,sprintf('%7.2f ps',T*dt*1e12));
set(TT,'fontsize',12)
TT = text(30,.15,sprintf('KE = %5.2f meV',KE)); % 5,-.15,
set(TT,'fontsize',12)
```

```
% ----- Задание 1 -----
% TT = text(25,-.15,sprintf('PE = %5.0f meV',PE));
% set(TT,'fontsize',12)
% TT = text(25,.13,sprintf('E_t_o_t = %5.0f meV',KE+PE));
% set(TT,'fontsize',12)
```

• Сохраняем график

```
% saveas(gcf,'second_task.png') % This saves the picture to a file
end
```

```
ans = 1.0000
ans = 0.9999
ans = 1.0000
```



• Время:

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

```
ans = datetime
17-Dec-2021 11:11:29
```