
```

%preparing workspace
clc
clear
close all

%constants
k=1.38e-23;
hbar=1.0546e-34;
m0=9.1e-31;
e=1.6e-19;
T=300;

%initial params
a=3e-9;           %width of well
b=2e-9;           %width of barrier
U0=1;             %height of barrier, 1 eV
m1=0.067*m0;      %eff. mass in GaAs, kg
m2=(0.067+0.083*0.3)*m0+m1;%eff. mass in AlGaAs(0.3), kg

Nen=50;
V=linspace(0,5,Nen);
V3=0;

%defining structure
j=2;              %amount of barriers
dx=2e-10;         %grid step
L=j*b+(j+1)*a;    %total length
Np=floor(L/dx);    %amount of grid cells
koef=hbar^2/(2*m1*(dx^2))/e;
x=(0:Np-1)*dx;    %x-vector

One=linspace(0,1,Np); %empty vector
n=1e21;
mu=hbar^2/(2*m1)*(3*pi^2*n)^(2/3)/e*0+0.3; %chem potential
% mu=mu*(1-pi^2/12*(k*300/e/mu)^2)

%correting hamiltonian corresponding to the heterstructure
E=eye(Np)*(2);
E=E+diag(ones(1,Np-1)*(-1),-1);
E=E+diag(ones(1,Np-1)*(-1),1);
E=E*koef;
for t=0:(2*j-1)
    if(mod(t,2)==0)
        E(:,round(Np*(floor(t/2)*(b+a)+a)/L):round(Np*(floor(t/2)+1)*(b+a)/L))=E(:,round(Np*(floor(t/2)*(b+a)+a)/L):round(Np*(floor(t/2)+1)*(b+a)/L))/(m1/m2);
    end
end
U=zeros(1,Np);
for t=0:(2*j-1)
    if(mod(t,2)==0)

```

```

        U(round(Np*(floor(t/2)*(b+a)+a)/L):round(Np*(floor(t/2)+1)*(b+a)/
L))=U0;
    end
end
H=E+diag(U);

Umax=1.1;
Ener=linspace(0,Umax,Nen);

D=zeros(1,Nen);
J=zeros(1,length(V));
JJ=zeros(1,length(V));

for klo=1:length(V)

    %correcting hamiltonian
    U1=-V(klo)*One;
    Ham=H+diag(U1);

    %Green's transmittion coef-t
    S1=zeros(Np);S2=S1;
    for i=1:Nen
        En=Ener(i);
        S1(1,1)=-koef*exp(1i*sqrt(2*m1*(En-U1(1))*e)/hbar*dx);
        S2(Np,Np)=-koef*exp(1i*sqrt(2*m1*(En-U1(end))*e)/hbar*dx);

        G1=1i*(S1-S1');
        G2=1i*(S2-S2');

        Gr=(eye(Np)*(En)-Ham-S1-S2);
        D(i)=real(trace(G1/Gr*G2/Gr'));
    end

    %transmission coefficient w/o votalge
    if(klo==1)
        Emax=Ener(islocalmax(D));
        D1=D;
        if (isempty(D1))
            Emax=0;
            D1=0;
        end
    end

    %fermi functions for source and drain
    f1=@(Ex,Ey,Ez)1./(1+exp((Ex+Ey+Ez-mu)/(k*T/e)));
    f2=@(Ex,Ey,Ez)1./(1+exp((Ex+Ey+Ez-mu+V(klo))/(k*T/e)));

    dE=Ener(2)-Ener(1); %energy step

    %attempt to integrate numerically
    %    Uup=10; Steps=105;
    %    ex=linspace(0,Uup,Steps);
    %    ey=linspace(0,Uup,Steps);
    %    [EX,EY,EZ]=meshgrid(ex,ey,Ener);

```

```

%
%      f=f1(EX,EY,EZ)-f2(EX,EY,EZ);
%
%      %F= squeeze(trapz(ey,trapz(ex,f,1),2));

%integrating functionally
q1=@(x)integral(@(y)(f1(x,y,Ener)-f2(x,y,Ener)),0,Inf,'ArrayValued',true);
F=integral(q1,0,Inf,'ArrayValued',true)';

%      %fermi functions for source and drain
%      f1=1./(1+exp(-(mu-Ener)./(k*T/e)));
%      f2=1./(1+exp(-(mu+V(klo)-Ener)./(k*T/e)));
%

%J by Green
J(klo)=e/hbar*2/(2*pi)^3*trapz(D.*(F'))*dE*e;

%J by Tsu
S=@(Ez)log((1+exp((mu-Ez)/(k*T/e)))/(1+exp((mu-Ez-V(klo))/(k*T/e))));
S=2*pi*m1*k*T/hbar^2*S(Ener)*e*10^-0.75;
JJ(klo)=e/hbar*2/(2*pi)^3*trapz(S.*D)*dE*e;

if(klo==50)
    h=figure('Units','normalized','OuterPosition',[1 0 0.5 1]);
    subplot(2,1,1)
    plot(Ener, log10(S),'--k','LineWidth',2);
    hold on;
    plot(Ener, log10(F))
    legend('S','int')
    grid on
    xlabel('Ez, ##')
    ylabel('#-### #####-##, lg')

    subplot(2,1,2)
    plot(Ener, (S),'--k','LineWidth',2);
    hold on;
    plot(Ener, (F))
%      plot(Ener, 1./(S').*(F));
%      plot(Ener, (S')-(F));
    legend('S','int')%,'/','-')
    grid on
    xlabel('Ez, ##')
    ylabel('#-### #####-##, #/#')
end
end

%microcurrent
J=J*1e6;
JJ=JJ*1e6;

f=figure('Units','normalized','OuterPosition',[0.05 0.05 0.5 0.5]);
subplot(1,3,2)
plot(D1,Ener)
ylabel('E,##')

```

```

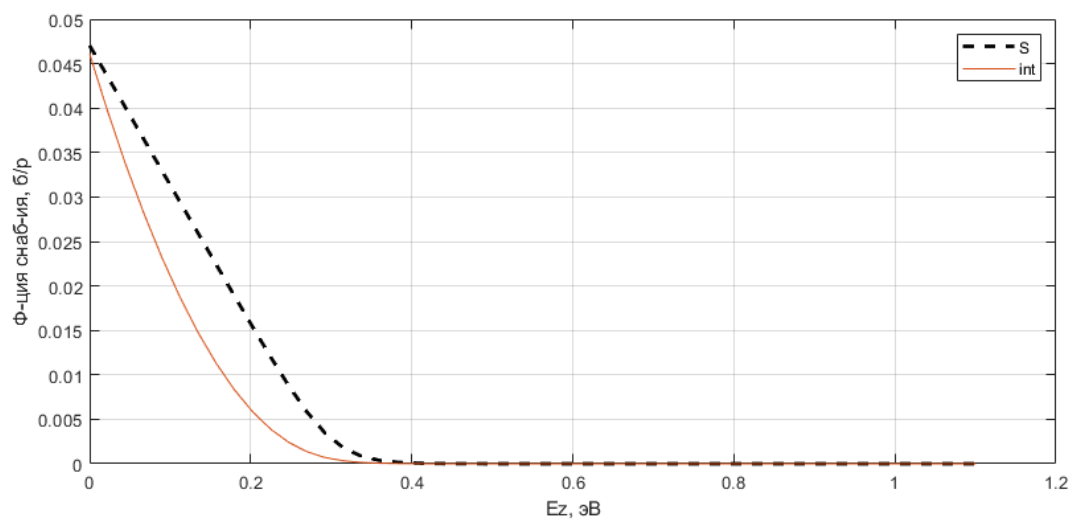
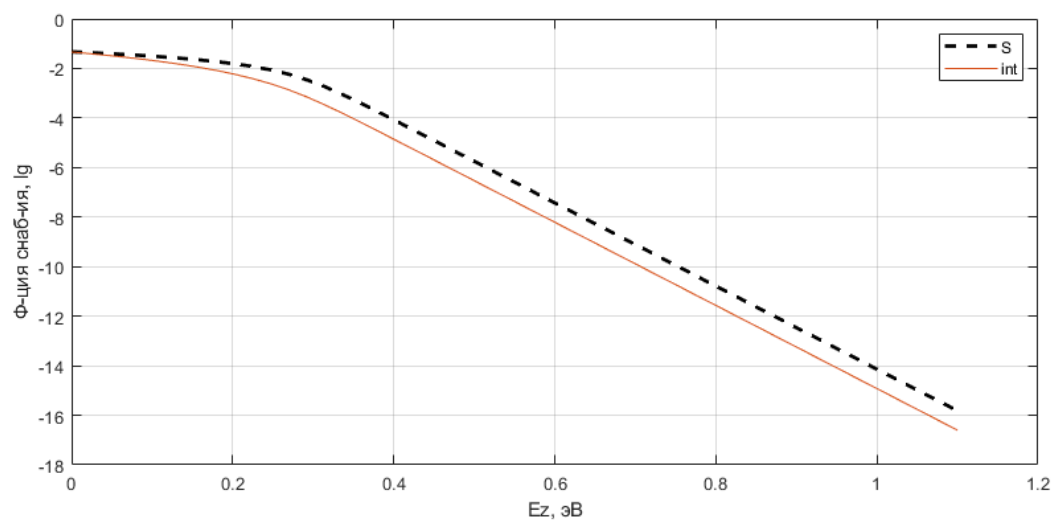
xlabel('D')
title('#####')
ylim([0, U0+0.1])
xlim([0 1])
grid on
hold on

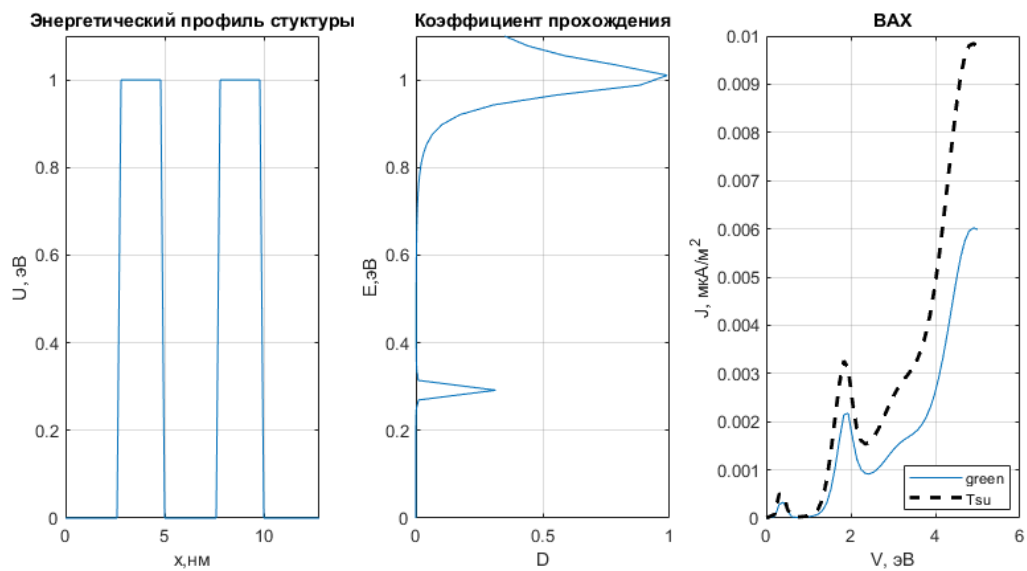
subplot(1,3,1)
plot(x*1e9,U)
hold on
ylim([-V3, U0+0.1-V3])
xlim([x(1), x(end)]*1e9)
xlabel('x,##')
ylabel('U, ##')
grid on
title('#####')

subplot(1,3,3)
hold off
plot(V,J)
hold on
plot(V,JJ,'--k','LineWidth',2)
ylabel('J, ###/#^2')
xlabel('V, ##')
title('###')
grid on
legend('green','Tsu','Location','best')

% exportgraphics(h,[num2str(number),'supply.jpg'])
% exportgraphics(f,[num2str(number),'vcc.jpg'])
% number=number+1

```





Published with MATLAB® R2021b