```
close all
clear
clc

fig=figure('Name','Method of variations','NumberTitle','off');
fig.Units='normalized';
fig.Position=[1 0 1 1];


hbar=1.0546e-34;
m0=9.1e-31;
e=1.6e-19;

%forming a task
L=1e-8;                         %width of structure
Np=1000;                        %amount of steps
x=linspace(-L/2,L/2, Np);       %creating a 'x-axis'
dx=x(2)-x(1);                   %definig a primive step
koef=-hbar^2/(2*m0*12*(dx^2));  %definig a coefficient for analytical
 solving

%defining type of force field
type=5;
switch type
    case 2
        U=e/2-abs(x/(L/2)*e/2);
    case 1
        U=abs(x/(L/2)*e/2);
    case 3
        U=e/2*(1+sin(pi*10*x/L));
    case 4
        U=e/2*(exp(-x.^2/(L/10)^2));
    case 5
        U=heaviside(x)*e/100;
end

%numerical solution for hamiltonian
%defining secind devirative
E=eye(Np)*(-30);
E=E+diag(ones(1,Np-1)*16,-1);
E=E+diag(ones(1,Np-1)*16,1);
E=E+diag(ones(1,Np-2)*(-1),-2);
E=E+diag(ones(1,Np-2)*(-1),2);

%Hamiltonian
H=E*koef+diag(U);

%finding eigenvalues and eigenvectors
[P,Ei]=eig(H);
Ei=diag(Ei);

%normalization eigvectors
```

```
P=P*sqrt(1/dx);

%visualizating analytical main state
subplot(2,3,1)
hold off
plot(x*1e9, U/e, '--k', 'LineWidth', 1)
hold on;
Amp=P(islocalmax(abs(P(:,1))),1); %defining amplitude for scaling a
 wave-function
Amp=max(Amp)/(Ei(1))*e;
for i=1:1
    plot(x*1e9, Ei(i)/e+P(:,i)/Amp,'-');
    plot(x*1e9, Ei(i)*ones(1,Np)/e,'--r');
    text(-4.5,Ei(i)/e,sprintf('$E_%i = %2.2f meV$',[i Ei(i)/
e*1000]),...
        'Interpreter','latex','FontSize',14,...
        'HorizontalAlignment','left','VerticalAlignment','bottom')
end
xlabel('$x,nm$', 'Interpreter', 'latex');
ylabel('$E,eV$', 'Interpreter', 'latex');
text((sum(xlim) + diff(xlim))/2+.07*(diff(xlim)-
sum(xlim))/2,0.5*sum(ylim),...
    '${\Psi\over \Psi_{1MAX}}+E
$','Interpreter','latex','Rotation',-90,...

  'HorizontalAlignment','center','VerticalAlignment','baseline','FontSize',16);
title('$Numreical$ $\Psi,E$', 'Interpreter', 'latex');
E1=Ei(1);
grid on;

%solving with variation method
%for choosen Gaboure variation function we have three variables, that
 should be varied
%first one variable is sigma
lmbd=2*L;
mu=0;
sg=linspace(0.01*L,2*L,Np);
w=real(exp(-(x-mu).^2./(2*sg'.^2)+1i*(2*pi*(x-mu)/lmbd)));
w=w./sqrt(sum(w.*w,2)*dx);

%solving for E_med of that function
dE=zeros(Np);
for i=1:Np
    G=[0 0 w(i,:) 0 0];
    for n=3:Np-2
        Dif2=-G(n-2)+16*G(n-1)-30*G(n)+16*G(n+1)-G(n+2);
        dE(i,n)=conj(G(n))*(koef*Dif2+U(n).*G(n));
    end
end
E=real(sum(dE,2)*dx);

%choosing minimal value of energy and corresponding sigma
[~,n]=min(E);
sigma=sg(n);
```

```
Es=E(n);

%visualizating dependence of energy by sigma
subplot(2,3,4)
hold off
plot(sg*1e9,E/e,'HandleVisibility','off')
hold on
plot(sigma*1e9,Es/e,'*')
xlabel('$\sigma,nm$', 'Interpreter', 'latex');
ylabel('$E,eV$', 'Interpreter', 'latex');
title('$E(\sigma)$', 'Interpreter', 'latex');
grid on;
legend(['$\sigma=',num2str(round(sigma*1e9*100)/100),'nm
$'], 'Interpreter', 'latex');

%second one variable is lambda
LMBD=linspace(0.05*L,4*L,Np)';
w=real(exp(-(x-mu).^2./(2*sigma.^2)+1i*(2*pi*(x-mu)./LMBD)));
w=w./sqrt(sum(w.*w,2)*dx);

%solving for E_med of that function
dE=zeros(Np);
for i=1:Np
    G=[0 0 w(i,:) 0 0];
    for n=3:Np-2
        Dif2=-G(n-2)+16*G(n-1)-30*G(n)+16*G(n+1)-G(n+2);
        dE(i,n)=conj(G(n))*(koef*Dif2+U(n).*G(n));
    end
end
E=real(sum(dE,2)*dx);

%choosing minimal value of energy and corresponding lambda
[~,n]=min(E);
lmbd=LMBD(n);
El=E(n);

%visualizating dependence of energy by lambda
subplot(2,3,5)
hold off
plot(LMBD*1e9,E/e,'HandleVisibility','off')
hold on
plot(lmbd*1e9,El/e,'*')
xlabel('$\lambda,nm$', 'Interpreter', 'latex');
ylabel('$E,eV$', 'Interpreter', 'latex');
title('$E(\lambda)$', 'Interpreter', 'latex');
grid on;
legend(['$\lambda=',num2str(round(lmbd*1e9)/100),'nm
$'], 'Interpreter', 'latex');

%third one variable is mu
MU=linspace(-L/2,L/2,Np)';
w=real((exp(-(x-MU).^2./(2*sigma.^2)+1i*(2*pi*(x-MU)/lmbd))));
w=w./sqrt(sum(w.*w,2)*dx);
```

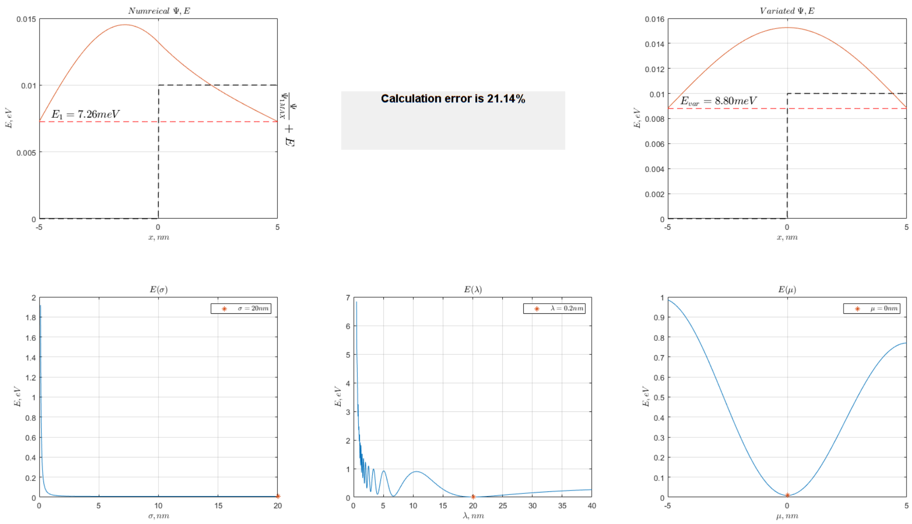
```
%solving for E_med of that function
dE=zeros(Np);
for i=1:Np
    G=[0 0 w(i,:) 0 0];
    for n=3:Np-2
        Dif2=-G(n-2)+16*G(n-1)-30*G(n)+16*G(n+1)-G(n+2);
        dE(i,n)=conj(G(n))*(koef*Dif2+U(n).*G(n));
    end
end
E=real(sum(dE,2)*dx);

%choosing minimal value of energy and corresponding mu
[~,n]=min(E);
mu=MU(n);
Em=E(n);
E2=Em;

%visualizating dependence of energy by mu
subplot(2,3,6)
hold off
plot(MU*1e9,E/e,'HandleVisibility','off')
hold on
plot(mu*1e9,Em/e,'*')
xlabel('$\mu,nm$', 'Interpreter', 'latex');
ylabel('$E,eV$', 'Interpreter', 'latex');
title('$E(\mu)$', 'Interpreter', 'latex');
grid on;
legend(['$\mu=',num2str(round(mu*1e9)/100),'nm
$'], 'Interpreter', 'latex');

%visualizating maximal optimal wave function for given variables
Psi=real((exp(-(x-mu).^2./(2*sigma.^2)+1i*(2*pi*(x-mu)/lmbd))));
Psi=Psi./sqrt(sum(Psi.*Psi,2)*dx);
subplot(2,3,3)
hold off
plot(x*1e9, U/e, '--k', 'LineWidth', 1)
hold on
plot(x*1e9,Em/e+Psi/Amp)
plot(x*1e9, Em/e*ones(1,1000),'--r');
text(-4.5,Em/e,sprintf('$E_{var} = %2.2f meV$',Em/
e*1000),'Interpreter','latex',...
'FontSize',14,'HorizontalAlignment','left','VerticalAlignment','bottom')
xlabel('$x,nm$', 'Interpreter', 'latex');
ylabel('$E,eV$', 'Interpreter', 'latex');
title('$Variated$ $\Psi,E$', 'Interpreter', 'latex');
grid on;

%verification of WF by comparison energies
Err = uicontrol('style','text');
txterr=sprintf('Calculation error is %2.2f%%',((E2-E1)/E1*100));
set(Err,'String',txterr,'FontSize',14,'FontWeight','bold');
Err.Units='normalized';
Err.Position=[0.4 0.7 0.2 0.1];
```



Published with MATLAB® R2020b