Department of physics, NTNU
TFY4340 Mesoscopic Physics
Spring 2010

Exercise 5

Assistance: Tuesday February 23.

In the lectures, we showed that the conduction band edge $E_C(z)$, and hence the potential energy $V(z)$ ”seen” by the electrons, is to a good approximation linear near the AlGaAs–GaAs interface in the socalled High Electron Mobility Transistor (HEMT):

$$V(z) = Fz$$

for $z > 0$, taking $z = 0$ precisely at the interface. Here, the constant F has the dimension of a force, and numerically, it will be of the order of 10 meV/nm, i.e., 10^7 eV/m, or about 1 pN.

If we ignore the fact that the conduction band offset between AlGaAs and GaAs is finite (2-300 meV), and also ignore the fact that $V(z)$ will flatten out when coming some ten nanometers or so into GaAs, we may use the linear form for $V(z)$ for all positive z and set $V = \infty$ for $z < 0$. The problem can then be solved analytically, as showed in the lectures.

Here, we will use approximate methods to solve the problem: First, with a variational approach. Second, with a numerical approach.

a) Based on knowledge about the ground state wave function $\Phi_0(z)$, ”guess” the form

$$\Phi_0(z) = z\, e^{-\alpha z/2}$$

and show that a variational calculation yields the ground state energy

$$E_0 = \overset{\min}{\alpha}\ E(\alpha) \simeq 2.48\, \left(\hbar^2/2m^*\right)^{1/3}\, F^{2/3}.$$

b) Solve the Schrödinger equation numerically. In this case, you may just as well use a more realistic shape of the conduction band edge in the vicinity of the interface. Put $V = \infty$ at some distance $\pm Z$ ”sufficiently” far away from the interface.

Below, a Matlab program is presented, which solves the Schrödinger equation numerically for a 1D potential between Zmin and Zmax, for various potentials $V(z)$. It is based on discretizing the kinetic energy operator, and diagonalizing the resulting matrix. The number of rows and columns in this matrix equals the number of grid points.

```
% TFY4340 Mesoscopic physics, spring 2010. Exercise 5:
% One-dimensional "triangular" confining potential in GaAs
% at AlGaAs-GaAs interface, giving rise to a 2DEG.
% Solved by straightforward matrix diagonalization, using
% the Matlab function eig
%
% Let us use SI units until the matrix has been diagonalized:
hbar = 1.05*10^(-34);
% Effective mass near conduction band minimum in GaAs:
mass = 0.067*9.1*10^(-31);
% Slope of linear potential ("force"): 10 meV/nm = 1.6 pN
F = 1.6*10^(-12);
% Number of grid points: 2*N+1
N = 500;
% Use step value dz of 1 Angstrom = 10^(-10) m
% The range of the potential is then from -50 nm to + 50 nm
% (Outside this range, V is set to infinity, so the wavefunction is zero
% for i=0 and i=2N+2)
dz = 10^(-10);
for i = 1 : N
    z(i) = dz*(-N-1+i);
% Smooth potential, 270 meV for large negative z, 300 meV at interface (z=0)
    V(i) = 1.6*10^(-22)*300*(0.9 + 0.1*exp(-0.02*(N-i)));
% Try to multiply V(z<0) with e.g. 1000, and verify that the energy of the
% ground state is then slightly below 90 meV, as found analytically in the
% lectures.
end
for i = N+1 : 2*N+1
    z(i) = dz*(-N-1+i);
% Linear potential for positive z
    V(i) = F*dz*(-N-1+i);
end
%
% Comment: V(z) will in real life become flat at large positive z. However,
% the conduction band edge deep inside GaAs will be at about 600 or 700 meV,
% so we don't bother to build in the flattening of V(z) here.
%
% Diagonal elements of matrix
d = (1/dz^2)*hbar^2/mass + V;
% Off-diagonal elements
e = (-1/(2*dz^2))*hbar^2/mass;
% Setting up the tri-diagonal Hamiltonian matrix, first diagonal terms:
H = diag(d);
% Next, include off-diagonal elements:
```

```
H(2:(2*N+1),1:2*N) = diag(e*ones(1,2*N)) + H(2:(2*N+1),1:2*N);
H(1:2*N,2:2*N+1) = diag(e*ones(1,2*N)) + H(1:2*N,2:2*N+1);
% Diagonalizing the matrix solves the Schrdinger equation.
% The command [S,D] = eig(H) produces matrices of eigenvalues (D) and
% eigenvectors (S) of matrix H, so that H*S = S*D. Matrix D is the
% canonical form of H - a diagonal matrix with H's eigenvalues on
% the main diagonal. Matrix S is the modal matrix - its columns are
% the eigenvectors of H:
[S,D] = eig(H);
% An alternative in Matlab, which returns the 10 algebraically smallest
% eigenvalues of the real and symmetric matrix H:
% [S,D] = eigs(H,10,'sa');
% For descriptions, see:
% http://www.mathworks.com/access/helpdesk/help/techdoc/ref/eig.html
% http://www.mathworks.com/access/helpdesk/help/techdoc/ref/eigs.html
% We store the eigenvalues in the array "eigenvalues":
eigenvalues = diag(D);
% Print to screen the 5 lowest eigenvalues, in the unit meV:
[eigenvalues(1) eigenvalues(2) eigenvalues(3) ...
eigenvalues(4) eigenvalues(5)]/(1.6*10^(-22))
% General syntax for plotting wavefunction nr n:
% plot(z,(S(:,n)'));
% Syntax for plotting absolute square of wavefunction nr n:
% plot(z,(S(:,n)').^2);
% (use unit nm for z axis)
% Plot wavefunction for 5 lowest eigenvalues, including
% the potential V(z) (in units of 5000 meV, to be on the same
% scale as the wave functions)
figure;
plot(z*10^9,V/(5000*1.6*10^(-22)),z*10^9,(S(:,1)'),z*10^9,(S(:,2)'), ...
z*10^9,(S(:,3)'),z*10^9,(S(:,4)'),z*10^9,(S(:,5)'));
axis([-50 50 -0.15 0.15]);
title('First five eigenfunctions');
xlabel('z (nm)');
% print -dpng triangular.png   % will create png-file
% Make arrays for each of the 5 lowest eigenvalues (unit meV),
% simply to draw horizontal lines as function of z at these values:
e1=eigenvalues(1)*ones(1,2*N+1)/(1.6*10^(-22));
e2=eigenvalues(2)*ones(1,2*N+1)/(1.6*10^(-22));
e3=eigenvalues(3)*ones(1,2*N+1)/(1.6*10^(-22));
e4=eigenvalues(4)*ones(1,2*N+1)/(1.6*10^(-22));
e5=eigenvalues(5)*ones(1,2*N+1)/(1.6*10^(-22));
% Plot of V(z) and the positions of the 5 lowest eigenvalues:
figure;
plot(z*10^9,V/(1.6*10^(-22)),z*10^9,e1,z*10^9,e2,z*10^9,e3,z*10^9,e4,z*10^9,e5);
```

```
axis([-50 50 0 600])
title('Potential and 5 lowest eigenvalues');
xlabel('z (nm)');
ylabel('Energy (meV)');
%END OF PROGRAM
```