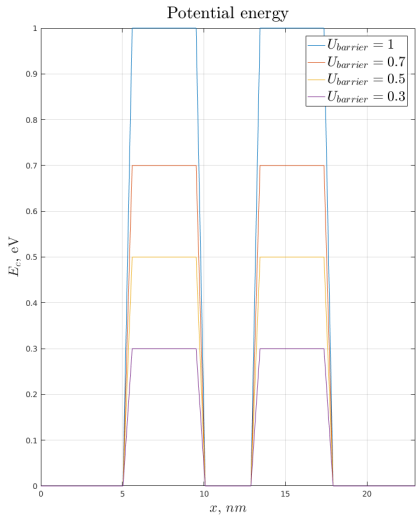
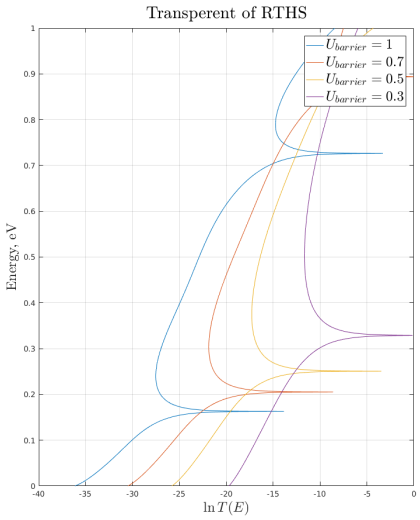
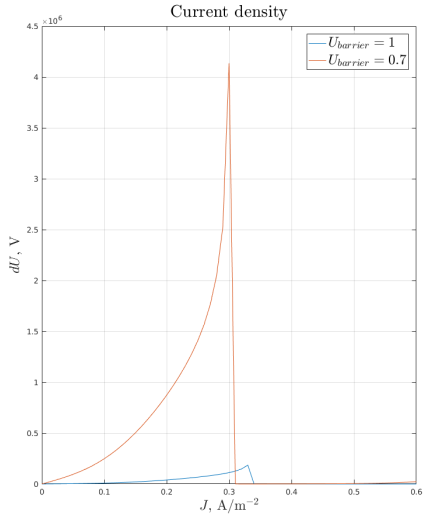
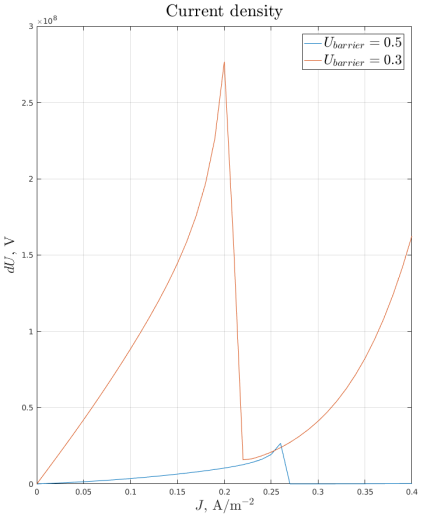
```
%Img 3.2, 3.3

clear; clc;

e = 1.6e-19; eVtoJ = e; JtoEv = e^(-1);
me = 9.11*1e-31; nm = 1e-9;

% Do it smooth
dis = 1;
% atoms' radius
dx = 0.56; %nm
dx = dx/dis;

% Count layers
a = 10; % monolayers
b = 8;
c = 6;

a = a*dis;
b = b*dis;
c = c*dis;

sizeHS = a + b + c + b + a;

% Fermi Energy
EFermi = 1.51*1e-20; % J

% Applyied voltage
dU = 0:0.01:0.6;

% Ec
Ec = [...
 zeros(1, a), 1*ones(1, b), zeros(1, c), 1*ones(1, b), zeros(1, a);
 zeros(1, a), 0.7*ones(1, b), zeros(1, c), 0.7*ones(1, b), zeros(1,
 a);
 zeros(1, a), 0.5*ones(1, b), zeros(1, c), 0.5*ones(1, b), zeros(1,
 a);
 zeros(1, a), 0.3*ones(1, b), zeros(1, c), 0.3*ones(1, b), zeros(1,
 a)...
];

% meff
meff = [...
 0.067*ones(1, a), 0.15*ones(1, b), 0.067*ones(1, c), 0.15*ones(1, b),
 0.067*ones(1, a);
 0.067*ones(1, a), 0.15*ones(1, b), 0.067*ones(1, c), 0.15*ones(1, b),
 0.067*ones(1, a);
 0.067*ones(1, a), 0.15*ones(1, b), 0.067*ones(1, c), 0.15*ones(1, b),
 0.067*ones(1, a);
 0.067*ones(1, a), 0.15*ones(1, b), 0.067*ones(1, c), 0.15*ones(1, b),
 0.067*ones(1, a)...
];
```

```
numPoint = 5000;
for j = 1 : length(Ec(:, 1))
 Tr(j, :) = getTransperent(...
  dx*nm, ...
  meff(j, :)*me, ...
  Ec(j, :)*eVtoJ,...
  numPoint...
 );

 J(j, :) = getJ(dx*nm, ...
  meff(j, :)*me, ...
  Ec(j, :)*eVtoJ, ...
  dU*eVtoJ, ...
  EFermi...
 );
end

showResult(dx*nm, sizeHS, Ec, J, dU, Tr);  %Img 3.2, 3.3
```









Published with MATLAB® R2016b