Bangladesh University of Engineering and Technology (BUET)

Assignment

A Self-Consistent Schrödinger-Poisson Solver Using MATLAB

Course Code: EEE 6512
Course Title: Nanoscale Device Modeling and Simulation Techniques

Submitted to:
Dr. Md. Kawsar Alam
Professor
Department of EEE, BUET

Submitted by:
Mirza Rabiul Hasan
Id: 0423062321
Dept: EEE

Assignment 2- Self Consistent Solver

Instructions:

Develop A Self-Consistent Schrödinger-Poisson Solver Using MATLAB. Things to be included in the report:

- Description and Work Flow
- Potential Profile
- Electric field profile
- Charge Profile
- Band Profile
- C-V characteristics
- Code as Appendix

You can choose any arbitrary device of your choice (e.g. SGMOS, DGMOS, FinFET etc.). Make sure the C-V profile is known for that device (from books or papers) so that you can compare.

Description and Work Flow: In this self-consistent Schrödinger poison solver a single gate MOSFET has been used. Potential, Electric field, Charge, Band profile and C-V characteristics were found almost same to the analytical solution. Working steps are given below.

- Start with Zero charge profile for n(z)
- Solve poison equation
- Find Potential profile
- Then Electric field profile by differentiating potential
- Get band diagram by inversing potential profile
- Solve Schrödinger equation by inserting data from band profile
- Get wave function and energy
- Now calculate n(z) using wave function and energy
- Insert this n(z) value in poison again
- Observe difference between old and new voltage
- Use convergence criteria by setting tolerance
- If solution converge then do it for another voltage
- Note all inversion charge for different applied voltage
- Finally find C-V curve by differentiating inversion charge with respect to applied voltage

Potential Profile:

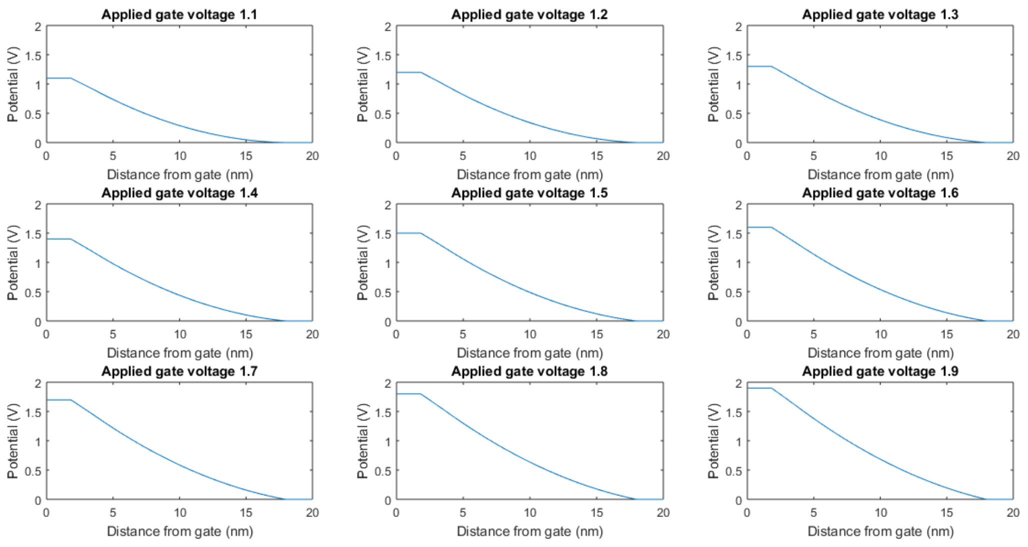

Fig 1: Potential Profile

Electric field profile:

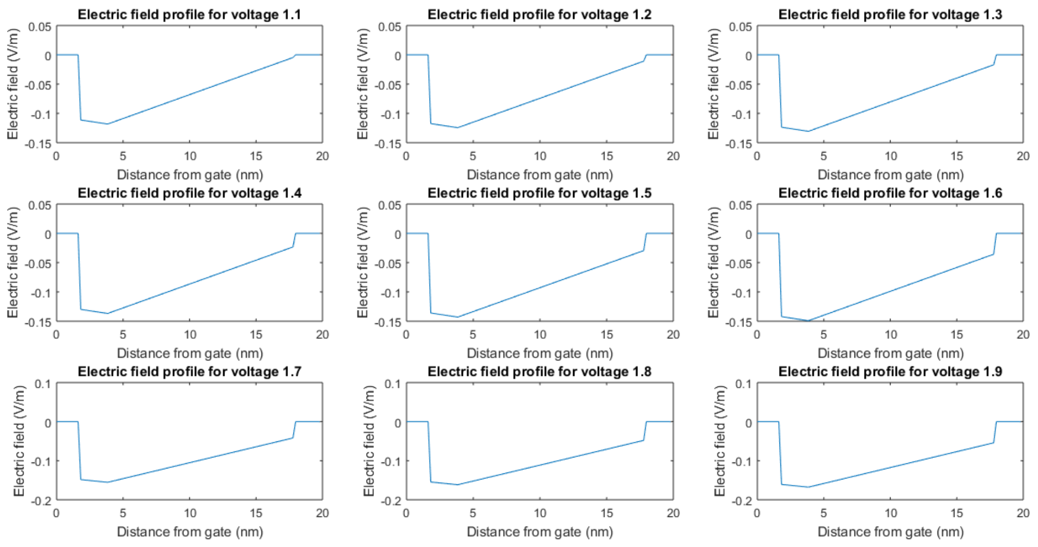

Fig 2: Electric field profile

Band Profile:

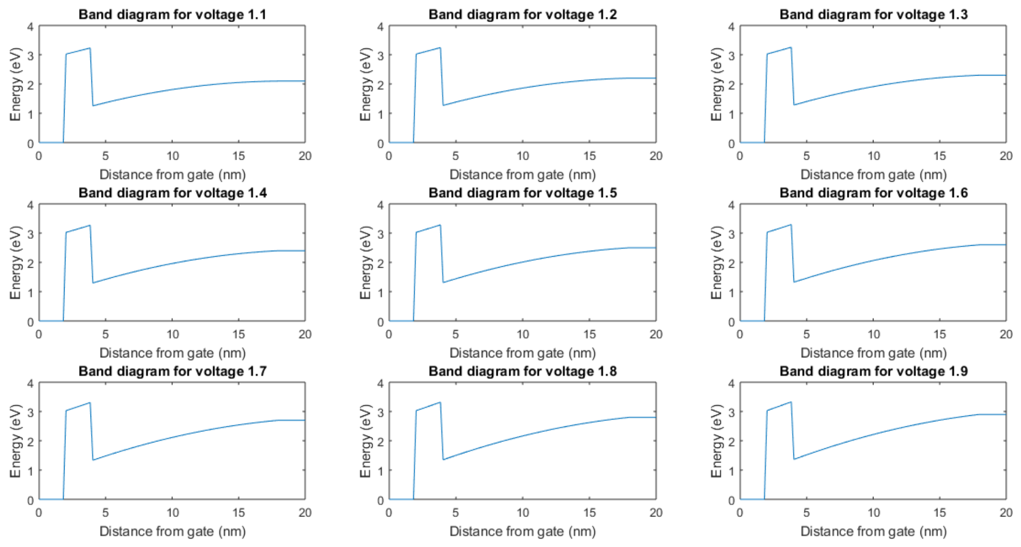

Fig 3: Band Profile

Wave Function:

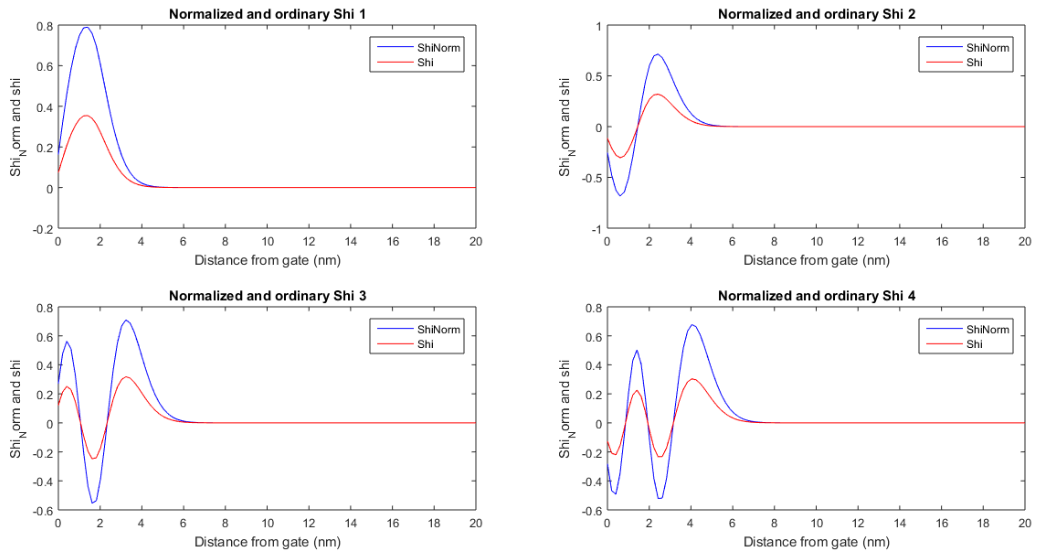

Fig 4: Wave Function

Charge Profile:

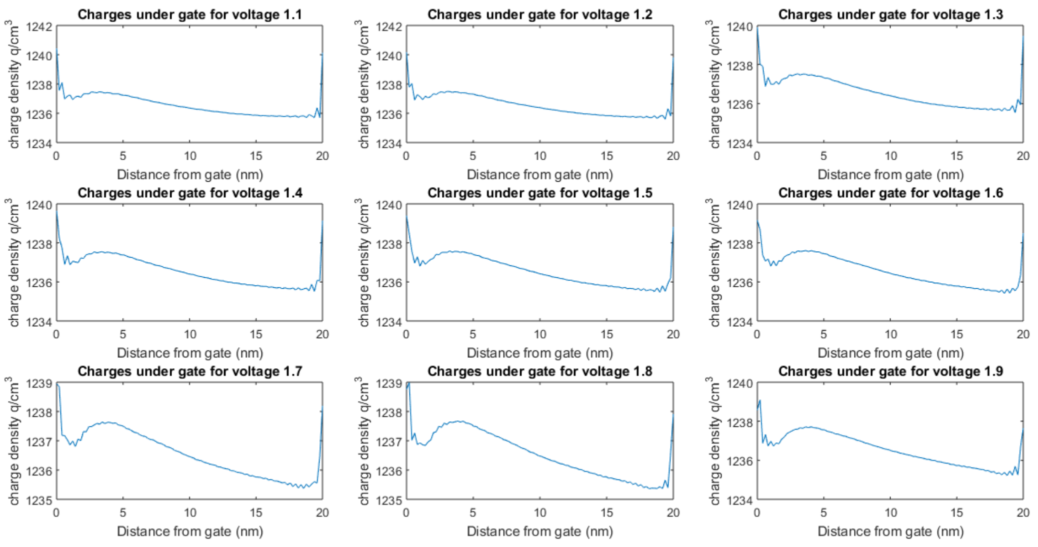

Fig 5: Charge Profile

C-V characteristics:

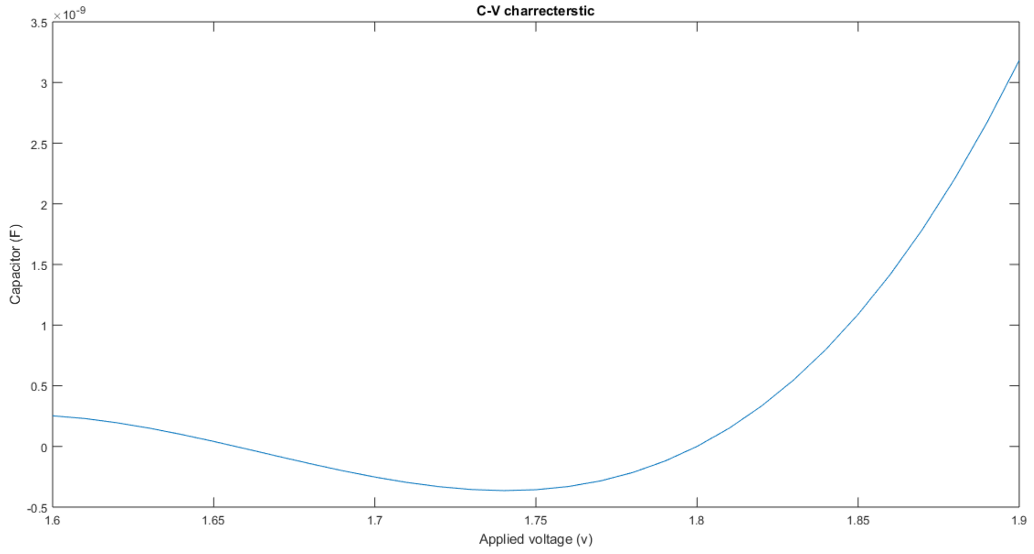

Fig 6: C-V characteristics

Appendix:

```
clear all
close all
clc

% define constant
q=-1.6e-19; % electron charge
ro_ox=q*1e12*(1e-12); % oxide charge density in um^-3
ro_s=q*1e16*(1e-12); % si charge density

ep0=8.854e-12; % permitivity of free space
ep_ox=3.9*ep0; % oxide permitivity
ep_si= 11.68*ep0; % Si permitivity
reduced_h=1.054*10^-34; % reduced plank constant
m=.25*9.11*10^-31; % electon mass

% nvi and mdi are the ith valley degeneracy and the ith density-of-states effective mass in Si.
nv = 6;
md = 0.26*9.1*10e-31;
Ef= .38*1.6e-19; % fermi energy level
kT= 1.38e-23*300;

x= linspace(0,20,100); % dimention from get to bottom in um
dx=x(2)-x(1);
a=eye(100,100); % initilize 1st matrix for poison

v = zeros(100,1); % initialize voltage distribution matrix
c=zeros(100,1); % initilize 2nd matrix for poison eqn
a1=eye(100,100); % initilize Hamiltonian matrix for Schrodinger eqn
e_field = zeros(100,1); % Electric field matrix initialization
nz1=zeros(100,1); % charge distribution matrix
nz_f=zeros(100,9); % charge distribution matrix for different voltage
N = zeros([100 100]); % N/ is the carrier concentration in the jth subband of the ith valley.
count=0; % for iteration cheaking in convergence
v_app=linspace(1.1,1.9,9); % applied voltage from 1.1v to 1.9v
n_inv=zeros(1,9);  % invertion charge for applied voltage

for j=1:1:length(v_app) % main loop for apppling different voltage
    rms_v1=0; % voltage initialization for convergence cheaking
    rms_v2=.1;
    while abs(rms_v2-rms_v1)>0.0001 % tollerence cheaking
        rms_v1=rms_v2;

      for i=10:88 % poison 1st matrix
        a(i+1,[i i+1 i+2])=(dx*dx)*[1 -2 1];
      end

        c(1:10)=v_app(j); % initial value applied in metal
        c(11:20)=-(ro_ox-(q*nz1(11:20)))/ep_ox; % oxide region
        c(21:99)=-(ro_s-(q*nz1(21:99)))/ep_si; % semiconductor region
        c(100)=0; % final value for single gate mos
        v=(inv(a)*c); % voltage calculation

        rms_v2=(trapz(x,(v.^2)/20).^.5); % rms voltage for cheaking convergence

        mE = -v(1:10)+v_app(j); % energy band diagram keeping metal as referance 0ev
        oxE = -v(11:20)+3+v_app(j); % oxide band diagram
        siE = -v(21:100)+1+v_app(j); % si band diagram
        E= [mE;oxE;siE]; % combined band diagram

        a1(1,1)=-2+v(1); % initialization for hamiltonian matrix
        a1(1,2)=1;
        a1(100,100)=-2+v(100);
        a1(100,99)=1;
        for i=1:98
            a1(i+1,[i i+1 i+2])=[1 -2+v(i+1) 1];
        end

        a1=(reduced_h^2./(2*m*dx.^2)).*a1;
        [shi,Eg]=eig(a1); % finding wave function and energy

        for i=1:100
            shi_norm_coff(i) = (trapz(x,(shi(:,i)).^2)).^.5;
            shi_norm(:,i)=shi(:,i)/shi_norm_coff(i); % normalized wave function
        end

        for i=1:100
            N(i,:)=((nv*md*kT)*log(1+exp((Ef-Eg(i,:))/kT)))/(pi*reduced_h*reduced_h); % Nij matrix
        end

        nz=(shi_norm.*shi_norm).*N; % crarge density distribuation matrix
        nz1 = (1e-17)*sum(nz,2);
        count=count+1;
    end

% voltage profile
figure(1);
subplot(3,3,j);
plot(x,v);
axis([0 20 1.1 1.9 ])
title(['Applied gate voltage ' num2str(v_app(j))]);
xlabel('Gate to bottom direction');
ylabel('Voltage');


nz_f(:,j)=nz1; % charge distribution for different applied voltage
n_inv(j)=q*trapz(x,(nz_f(:,j))); % inversion region total charge

for i=1:99
    e_field(i)=(v(i+1)-v(i))/dx; % Electric field calculation
end

% Band Structure
figure(2);
subplot(3,3,j);
plot(x,E)
title(['Band diagram for voltage ' num2str(v_app(j))]);
```

```
xlabel('Gate to bottom direction');
ylabel('Energy');

% Electric field profile
figure(3);
subplot(3,3,j);
plot(x(1:99),e_field(1:99));
title(['Electric field profile for voltage ' num2str(v_app(j))]);
xlabel('Gate to bottom direction');
ylabel('Electric field');

% Charge profile
figure(4);
subplot(3,3,j);
plot(x, nz1);
title(['Charges under gate for voltage ' num2str(v_app(j))]);
xlabel('Gate to bottom axis');
ylabel('charge density');

k=1;
for i=97:100
figure(4+j)
subplot(2,2,5-k)
plot(x,(-shi_norm(:,i)),'b')
hold on
plot(x,(-shi(:,i)),'r')
legend('ShiNorm','Shi')
title(['Normalized and ordinary Shi ' num2str(5-k)]);
xlabel('Gate to bottom direction');
ylabel('ShiNorm and shi');
hold off
k=k+1;
end
end

% Calculate capacitance from charge and voltage dQ/dV
cap = zeros(1,length(n_inv)-1);
for i=1:length(n_inv)-1
    cap(i)=(n_inv(i+1)-n_inv(i))/(v_app(2)-v_app(1));
end

% C-V profile
figure(14);
v_int = 1.6:.01:1.9; % add more point by interpolating
cap_new=(1.6e19)*interp1(v_app(1:8),cap,v_int,'spline');
plot(v_int,cap_new);
title('C-V charrecterstic');
xlabel('Applied voltage');
ylabel('Capacitane')
```