Finite Difference based 1D Nonlinerar Possion Equation Solver for pn-Junctions

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Name                    : pn_poisson_calc.mlx
% Author                  : Patrick Gsoels (11771814)
% Contact                 : patrick.gsoels@student.tugraz.at
% Creation Date           : 16.07.2022
% Version                 : 1.1
% Description             : This live script computes the charge carrier
%                           concentrations, charge-density, electric field,
%                           elctric potential and band-structure of arbitrary
%                           pn-junction doping profiles.
% Language                : MATLAB R2020b
% Category                : live script
% To Do                   : -
% --------------------------------------------------------------------------
% 1:01: initial version [PG]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
```

Table of Contents

Poisson Equation for Electrostatics
Newton-Raphson Method
Finite Difference Formulation
Discretization of the 1D Poisson Equation
Declaration of Constants
Grid Specifications
Doping Profile
Newton-Raphson Parameters
Implementation of a Newton-Raphson Assisted Finite Difference 1D Poisson Solver
Plotting the Results
    Doping Profile
    Charge Carrier Concentrations
    Charge Density
    Internal Electric Field
    Build-in Potential
    Band Diagram
References

Poisson Equation for Electrostatics

In electrodynamics Gauss's law states the relation between the electric displacement field $\mathbf{D}$ and the free charge density ρ within a volume. The differential form of Gauss's law can be written as

$$\nabla \cdot \mathbf{D} = \rho. \quad (1)$$

For a homogenious medium the following relation between electric displacement field $\mathbf{D}$ and electric field $\mathbf{E}$ holds:

$$\mathbf{D} = \varepsilon \cdot \mathbf{E} \quad (2)$$

In electrostatics we do not have any time-dependent fields. Hence, Faraday's law of induction will simplify to the following expression:

$$\nabla \times \mathbf{E} = \mathbf{0} \quad (3)$$

The simplified version of Faraday's law of induction states that the electric field $\mathbf{E}$ is free of curl. Thus, the electric field $\mathbf{E}$ is a conservative field for which one can define a potential.

$$\mathbf{E} = -\nabla V \quad (4)$$

Inserting Equation (2) and (4) into Equation (1) will give us the Poisson equation for electrostatics.

$$\nabla \cdot (\varepsilon \cdot \nabla V) = -\rho \quad (5)$$

For a constant dielectric media Equation (5) simplifies to

$$\nabla^2 V(x, y, z) = -\frac{\rho(x, y, z)}{\varepsilon}. \quad (6)$$

pn-junctions do typically have doping profiles which intentionally vary along the latteral axis which is perpendicular to the junction interface. Thus, the Poisson equation from (6) reduces to one spacial coordinate.

$$\nabla^2 V(x) = \frac{\mathrm{d}^2 V}{\mathrm{d}x^2} = -\frac{\rho(x)}{\varepsilon}. \quad (7)$$

Due to the interconnection of differently doped materials, charge carriers diffuse accros the pn-junction. At the same time an internal electric field establishes due to the movement of charge carriers. These facts lead to the necessity of introducing a potential-dependent charge density ρ, which reformulates the Poisson equation to

$$\nabla^2 V(x) = \frac{\mathrm{d}^2 V}{\mathrm{d}x^2} = -\frac{\rho(V(x))}{\varepsilon}. \quad (8)$$

According to [1], assuming the dopants to be totally ionised, the charge density inside a semiconductor is given by

$$\rho(x) = q \cdot \left[p(V(x)) - n(V(x)) + N_D(x) - N_A(x)\right]. \quad (9)$$

According to [2], the electron $n(x)$ and hole density $p(x)$ can be formulated as

$$n(x) = n_i \cdot \exp\left(\frac{E_{Fn} - E_i}{k_B \cdot T}\right) = n_i \cdot \exp\left(\frac{q \cdot V(x)}{k_B \cdot T}\right) \quad (10)$$

and

$$p(x) = n_i \cdot \exp\left(\frac{E_i - E_{Fp}}{k_B \cdot T}\right) = n_i \cdot \exp\left(\frac{-q \cdot V(x)}{k_B \cdot T}\right). \quad (11)$$

Inserting Equation (10) and (11) into (9) will provide us with the full formulation of the potential-dependent charge density ρ.

$$\rho(x) = q \cdot \left[n_i \cdot \exp\left(\frac{-q \cdot V(x)}{k_B \cdot T}\right) - n_i \cdot \exp\left(\frac{q \cdot V(x)}{k_B \cdot T}\right) + N_D(x) - N_A(x)\right] \quad (12)$$

with n_i being the intrinsic carrier concentration of the semiconductor, which is defined as

$$n_i^2 = n \cdot p = N_c(T_{300K}) \cdot \left(\frac{T}{T_{300K}}\right)^{\frac{3}{2}} \cdot N_v(T_{300K}) \cdot \left(\frac{T}{T_{300K}}\right)^{\frac{3}{2}} \cdot \exp\left(\frac{-E_g}{k_B \cdot T}\right). \quad (13)$$

Bear in mind that the formulars for n, p and n_i are derived by using the Boltzmann approximation instead of the real fermi-dirac distribution. Hence, the condition $|E - E_F| > 3 \cdot k_B \cdot T$ has to hold.

With the help of some trigonometric identity ($e^x - e^{-x} = 2 \cdot \sinh(x)$), Equation (12) can be reformulated as

$$\rho(x) = q \cdot \left[N_D(x) - N_A(x) - 2 \cdot n_i \cdot \sinh\left(\frac{q \cdot V(x)}{k_B \cdot T}\right)\right]. \quad (14)$$

Since our one-dimensional Poisson equation is non-linear, one will need at some point the partial derivative of the charge density ρ with respect to the electric potential V.

$$\frac{\partial \rho}{\partial V} = -\frac{2 \cdot q^2 \cdot n_i}{k_B \cdot T} \cdot \cosh\left(\frac{q \cdot V(x)}{k_B \cdot T}\right) \quad (15)$$

Newton-Raphson Method

The Newton-Raphson method is mathematically the most preferred of the several known methods for the solution of systems of nonlinear equations. It is a successive approximation technique based on an initial estimate of the solution and the Taylor series expansion [1]. Consider the set of equations

$$\mathbf{f}(\mathbf{V}) = \begin{bmatrix} f_1(\mathbf{V}) \\ \vdots \\ f_N(\mathbf{V}) \end{bmatrix} = \mathbf{0} \quad (16)$$

A Taylor approximation of (16), which is terminated after the linear term, leads to

$$\mathbf{f}(\mathbf{V}^{(k+1)}) \approx \mathbf{f}(\mathbf{V}^{(k)}) + \mathcal{J}(\mathbf{V}^{(k)}) \cdot (\mathbf{V}^{(k+1)} - \mathbf{V}^{(k)}) = \mathbf{f}(\mathbf{V}^{(k)}) + \mathcal{J}(\mathbf{V}^{(k)}) \cdot \Delta\mathbf{V}^{(k)} \overset{!}{=} \mathbf{0}. \quad (17)$$

$$\mathbf{f}(\mathbf{V}^{(k+1)}) \approx \mathbf{f}(\mathbf{V}^{(k)}) + \mathcal{J}(\mathbf{V}^{(k)}) \cdot \Delta\mathbf{V}^{(k)} = \begin{bmatrix} f_1(\mathbf{V}^{(k)}) + \sum_{j=1}^{N} \left.\frac{\partial f_1}{\partial V_j^{(k)}}\right|_{\mathbf{V}^{(k)}} \cdot \Delta V_j^{(k)} \\ \vdots \\ f_N(\mathbf{V}^{(k)}) + \sum_{j=1}^{N} \left.\frac{\partial f_N}{\partial V_j^{(k)}}\right|_{\mathbf{V}^{(k)}} \cdot \Delta V_j^{(k)} \end{bmatrix} \overset{!}{=} \mathbf{0}. \quad (18)$$

with

$$\mathcal{J}(\mathbf{V}^{(k)}) = \begin{bmatrix} \left.\frac{\partial \mathbf{f}(\mathbf{V})}{\partial V_1}\right|_{\mathbf{V}=\mathbf{V}^{(k)}} & \cdots & \left.\frac{\partial \mathbf{f}(\mathbf{V})}{\partial V_N}\right|_{\mathbf{V}=\mathbf{V}^{(k)}} \end{bmatrix} = \begin{bmatrix} \nabla^T f_1(\mathbf{V})\big|_{\mathbf{V}=\mathbf{V}^{(k)}} \\ \vdots \\ \nabla^T f_M(\mathbf{V})\big|_{\mathbf{V}=\mathbf{V}^{(k)}} \end{bmatrix} = \begin{bmatrix} \left.\frac{\partial f_1(\mathbf{V})}{\partial V_1}\right|_{\mathbf{V}=\mathbf{V}^{(k)}} & \cdots & \left.\frac{\partial f_1(\mathbf{V})}{\partial V_N}\right|_{\mathbf{V}=\mathbf{V}^{(k)}} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial f_M(\mathbf{V})}{\partial V_1}\right|_{\mathbf{V}=\mathbf{V}^{(k)}} & \cdots & \left.\frac{\partial f_M(\mathbf{V})}{\partial V_N}\right|_{\mathbf{V}=\mathbf{V}^{(k)}} \end{bmatrix} \quad (19)$$

being the Jaccobian.

After rearranging (17), one gets the expression for the Newton-Raphson step

$$\Delta\mathbf{V}^{(k)} = -\mathcal{J}(\mathbf{V}^{(k)}) \cdot \mathbf{f}(\mathbf{V}^{(k)}) \quad (20)$$

and the $(k+1)$-th iterate

$$\mathbf{V}^{(k+1)} = \mathbf{V}^{(k)} + \Delta\mathbf{V}^{(k)}. \quad (21)$$

Finite Difference Formulation

The finite difference formulation can be again derived from the Taylor approximation of $V(x)$ as depicted below.

$$V(x_j+\Delta x_j) = V(x_j) + \frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \cdot \Delta x_j + \frac{1}{2!}\cdot\frac{\mathrm{d}^2V(x)}{\mathrm{d}x^2}\bigg|_{x=x_j} \cdot \Delta x_j^2 + \frac{1}{3!}\cdot\frac{\mathrm{d}^3V(x)}{\mathrm{d}x^3}\bigg|_{x=x_j} \cdot \Delta x_j^3 + \ldots = V(x_j) + \frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \cdot \Delta x_j + \mathcal{O}\left(\Delta x_j^2\right) \approx V(x_j) + \frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \cdot \Delta x_j \tag{22}$$

$$\frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \approx V_j^{(FD)'} = \frac{V(x_j+\Delta x_j) - V(x_j)}{\Delta x_j} = \frac{V_{j+1}-V_j}{\Delta x_j} \text{ [forward difference]} \tag{23}$$

$$V(x_j-\Delta x_j) = V(x_j) - \frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \cdot \Delta x_j + \frac{1}{2!}\cdot\frac{\mathrm{d}^2V(x)}{\mathrm{d}x^2}\bigg|_{x=x_j} \cdot \Delta x_j^2 - \frac{1}{3!}\cdot\frac{\mathrm{d}^3V(x)}{\mathrm{d}x^3}\bigg|_{x=x_j} \cdot \Delta x_j^3 + \ldots = V(x_j) - \frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \cdot \Delta x_j + \mathcal{O}\left(\Delta x_j^2\right) \approx V(x_j) + \frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \cdot \Delta x_j \tag{24}$$

$$\frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \approx V_j^{(BD)'} = \frac{V(x_j) - V(x_j-\Delta x_j)}{\Delta x_j} = \frac{V_j-V_{j-1}}{\Delta x_j} \text{ [backward difference]} \tag{25}$$

One gets the central difference formulation for the first order derivative if one subtracts Equation (24) from (22) like shown below.

$$V(x_j+\Delta x_j) - V(x_j-\Delta x_j) = 2\cdot\frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \cdot \Delta x_j + 2\cdot\mathcal{O}\left(\Delta x_j^3\right) \approx 2\cdot\frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \cdot \Delta x_j \tag{26}$$

$$\frac{\mathrm{d}V(x)}{\mathrm{d}x}\bigg|_{x=x_j} \approx V_j^{(CD)'} = \frac{V(x_j+\Delta x_j) - V(x_j-\Delta x_j)}{2\Delta x_j} = \frac{V_{j+1}-V_{j-1}}{2\Delta x_j} \text{ [central difference]} \tag{27}$$

The second order derivative can be approximated by inserting Equation (23) and (25) into the expression depicted below.

$$\frac{\mathrm{d}^2V(x)}{\mathrm{d}x^2}\bigg|_{x=x_k} \approx V_j{}'' = \frac{V_j^{(FD)'} - V_j^{(BD)'}}{\Delta x_j} = \frac{V_{j+1} - 2\cdot V_j + V_{j-1}}{\Delta x_j^2} \tag{28}$$

Discretization of the 1D Poisson Equation

With the aid of the finite difference formulation one can approximate Equation (8), which results in

$$\frac{V_{j+1} - 2\cdot V_j + V_{j-1}}{\Delta x_j^2} = -\frac{\rho(V_j)}{\varepsilon},\ j = 1,\ldots,N. \tag{29}$$

After rearranging Equation (29) one gets the non-linear algebraic difference formulation of the Poisson Equation.

$$f_i(\mathbf{V}) = \frac{V_{j+1} - 2\cdot V_j + V_{j-1}}{\Delta x_j^2} + \frac{\rho(V_j)}{\varepsilon} = 0,\ j = 1,\ldots,N. \tag{30}$$

By using the Newton-Raphson technique for solving non-linear systems of equations, the non-linear algebraic difference equation (30) formulates to

$$f_i(\mathbf{V}^{(k+1)}) \approx \left[f_i(\mathbf{V}^{(k)})\right] + \left\{\sum_{j=1}^{N}\frac{\partial f_i}{\partial V_j}\bigg|_{\mathbf{V}=\mathbf{V}^{(k)}} \cdot \Delta V_j^{(k)}\right\} = \left[\frac{V_{j+1}^{(k)} - 2\cdot V_j^{(k)} + V_{j-1}^{(k)}}{\Delta x_j^2} + \frac{\rho\left(V_j^{(k)}\right)}{\varepsilon}\right] + \left\{\frac{\Delta V_{j+1}^{(k)} - 2\cdot \Delta V_j^{(k)} + \Delta V_{j-1}^{(k)}}{\Delta x_j^2} + \frac{\partial\rho}{\partial V}\bigg|_{V=V_j^{(k)}} \cdot \Delta V_j^{(k)}\right\} = 0. \tag{31}$$

The known parts of Equation (31) will define our right hand-side and the unknown parts will define the left-hand side of the system of equations which needs to be solved for $\Delta V_j^{(k)}$.

$$-\left\{\frac{\Delta V_{j+1}^{(k)} - 2\cdot \Delta V_j^{(k)} + \Delta V_{j-1}^{(k)}}{\Delta x_j^2} + \frac{\partial\rho}{\partial V}\bigg|_{V=V_j^{(k)}} \cdot \Delta V_j^{(k)}\right\} = \left[\frac{V_{j+1}^{(k)} - 2\cdot V_j^{(k)} + V_{j-1}^{(k)}}{\Delta x_j^2} + \frac{\rho\left(V_j^{(k)}\right)}{\varepsilon}\right] \tag{32}$$

By defining the right-hand side of Equation (32) as

$$R_j^{(k)} = \frac{V_{j+1}^{(k)} - 2\cdot V_j^{(k)} + V_{j-1}^{(k)}}{\Delta x_j^2} + \frac{\rho\left(V_j^{(k)}\right)}{\varepsilon} \tag{33}$$

and the left-hand side as

$$M_{ji}^{(k)}\cdot\Delta V_j^{(k)} = \begin{cases} \left(\dfrac{2}{\Delta x_j^2} - \dfrac{\partial\rho}{\partial V}\bigg|_{V=V_j^{(k)}}\right)\cdot\Delta V_j^{(k)} & ,\, i = j \\ -\dfrac{1}{\Delta x_j^2}\cdot\Delta V_j^{(k)} & ,\, i = |j\pm1| \\ 0 & ,\,\text{else} \end{cases}. \tag{34}$$

With Equation (33) and (34) one can easily write Equation (32) in matrix form

$$\begin{bmatrix} M_{1,1}^{(k)} & M_{1,2}^{(k)} & 0 & & & & \\ M_{2,1}^{(k)} & M_{2,2}^{(k)} & M_{2,3}^{(k)} & 0 & & & \\ 0 & M_{3,2}^{(k)} & M_{3,3}^{(k)} & M_{3,4}^{(k)} & 0 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 0 & M_{(N-2),(N-3)}^{(k)} & M_{(N-2),(N-2)}^{(k)} & M_{(N-2),(N-1)}^{(k)} & 0 \\ & & & 0 & M_{(N-1),(N-2)}^{(k)} & M_{(N-1),(N-1)}^{(k)} & M_{(N-1),N}^{(k)} \\ & & & & 0 & M_{N,(N-1)}^{(k)} & M_{N,N}^{(k)} \end{bmatrix} \cdot \begin{bmatrix} \Delta V_1^{(k)} \\ \Delta V_2^{(k)} \\ \Delta V_3^{(k)} \\ \vdots \\ \Delta V_{N-2}^{(k)} \\ \Delta V_{N-1}^{(k)} \\ \Delta V_N^{(k)} \end{bmatrix} = \begin{bmatrix} R_1^{(k)} \\ R_2^{(k)} \\ R_3^{(k)} \\ \vdots \\ R_{N-2}^{(k)} \\ R_{N-1}^{(k)} \\ R_N^{(k)} \end{bmatrix} \tag{35}$$

with $\mathbf{V}^{(0)} = \mathbf{V}_{init}$, $V_0^{(k)} = V_{init}(x_0)$, $V_{N+1}^{(k)} = V_{init}(x_{N+1})$, $\Delta V_0^{(k)} = \Delta V_{N+1}^{(k)} = 0$ and

$$V_{init}(x_j) = \begin{cases} \mathrm{sign}\left(N_D^{(j)} - N_A^{(j)}\right) \cdot \frac{k_B \cdot T}{q} \cdot \ln\left(\frac{\left|N_D^{(j)} - N_A^{(j)}\right|}{n_i}\right) & , \left|N_D^{(j)} - N_A^{(j)}\right| \geq n_i \\ 0 & , \text{else} \end{cases}. \quad (36)$$

Finally, one has to solve Equation (35) for the Newton-Raphson step $\Delta \mathbf{V}^{(k)}$ in order to generate the next iterate of the potential $\mathbf{V}^{(k+1)}$ as defined by Equation (21).

$$\mathbf{V}^{(k+1)} = \mathbf{V}^{(k)} + \Delta \mathbf{V}^{(k)} \quad (37)$$

Declaration of Constants

```
% temperature
% (temperature-dependent band-gap and intrinsic carrier concentration is not implemented)
T = 300;                 % in degree K

% permittivity of vacuum
eps_0 = 8.854187813e-12; % in As/Vm

% relative permittivity
eps_r = 12;

% absolute permittivity
epsilon = eps_0 * eps_r; % in As/Vm

% Boltzmann constant
kB = 1.380658e-23;       % in J/K

% elementary charge
q = 1.60217733e-19;      % in As

% intrinsic carrier concentration of SI
ni = 1.45e16;            % in 1/m^3

% band gap energy of SI
Eg = 1.12;               % in eV
```

Grid Specifications

```
% lower grid bound
xmin = str2double("-4e-6"); % in m

% upper grid bound
xmax = str2double("4e-6"); % in m

% number of grid points
N = round(str2double("1000"));
```

Doping Profile

```
doping_profile = "abrupt_pnp";
```

Newton-Raphson Parameters

```
maxIter = round(str2double("1000"));
alpha = 1e-6;
```

Implementation of a Newton-Raphson Assisted Finite Difference 1D Poisson Solver

```
% grid points (sampling)
x = linspace(xmin,xmax,N).';

% step-size
dx = x(2)-x(1);

% grid extension (needed for central differences)
xg = [ min(x)-dx;x;max(x)+dx ];

% load doping profile
[NA,ND] = doping_profiles(doping_profile,x);

% initializing electron concentration
n = ND;

% initializing hole concentration
p = NA;

% initializing electric potential
V = zeros(size(xg));
Vinit = band_initialization(NA,ND,ni,q,kB,T);
```

```
V(2:N+1) = Vinit;
V(1) = V(2);       % boundary
V(end) = V(end-1); % boundary

V_old = V;

% figure;
% plot(x,V(2:N+1))

% loop for Newton-Raphson Iterations
for k=1:maxIter

  % updating charge carrier concentrations
  n = ni*exp(q*V(2:N+1)/(kB*T));
  p = ni*exp(-q*V(2:N+1)/(kB*T));

  % charge density
  % (assuming that the dopants are totally ionized)
  rho = q*(p - n + ND - NA);

  % charge density change due to the internal potential
  drho_dV = -2*q^2*ni/(kB*T) * cosh(q*V(2:N+1)/(kB*T));

  % finite difference formulation for the potential
  d2V_dx2 = ( V(1:N,1) - 2*V(2:N+1,1) + V(3:N+2,1) )/(dx^2);

  % non-linear equation system
  Mjj_k = 2/(dx^2) - 1/epsilon*drho_dV;
  M_off = 1/(dx^2)*ones(N,1);
  M_k = spdiags([-M_off Mjj_k -M_off],-1:1,N,N);
  R_k = 1/epsilon*rho + d2V_dx2;

  % solving for potential changes
  delta_V = M_k\R_k;

  % Newton-Raphson Step (updating the potential)
  %V(2:N+1) = V(2:N+1)+(0.98^k)*delta_V;
  V(2:N+1) = V(2:N+1)+delta_V;

  % stopping criteria
  if( norm( (V - V_old), "inf" ) < alpha )
    break;
  end

  V_old = V;
end

% calculating the electric field
E = -diff(V(1:N+1))/dx;

% build in voltage
Vbi = abs(V(end)-V(1));

disp(['Number of Newton-Raphson Iterations: ',num2str(k)])
```

```
Number of Newton-Raphson Iterations: 6
```

```
disp(['build in voltage: Vbi = ',num2str(Vbi),'V'])
```

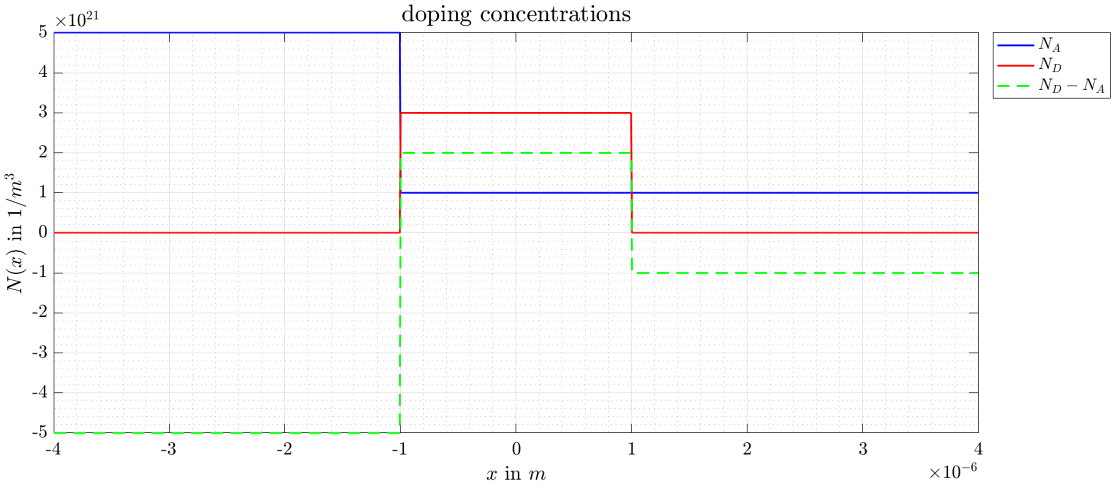
```
build in voltage: Vbi = 0.041607V
```

Plotting the Results

```
set(groot, ...
'DefaultTextInterpreter', 'LaTeX',...
'DefaultAxesTickLabelInterpreter', 'LaTeX',...
'DefaultLegendInterpreter', 'LaTeX',...
'DefaultAxesFontName', 'LaTeX',...
'DefaultFigureColor', 'w', ...
'DefaultAxesLineWidth', 0.5, ...
'DefaultAxesXColor', 'k', ...
'DefaultAxesYColor', 'k', ...
'DefaultAxesFontUnits', 'points', ...
'DefaultAxesFontSize', 15, ...
'DefaultAxesFontName', 'Helvetica', ...
'DefaultLineLineWidth', 1.5, ...
'DefaultTextFontUnits', 'Points', ...
'DefaultTextFontSize', 20, ...
'DefaultTextFontName', 'Helvetica', ...
'DefaultAxesBox', 'on', ...
'DefaultAxesTickLength', [0.01 0.01],...
'defaultAxesXGrid','on',...
'defaultAxesYGrid','on',...
'defaultfigureposition',[50 50 1080 400],...
'defaultAxesTitleFontSize',1.3);
```

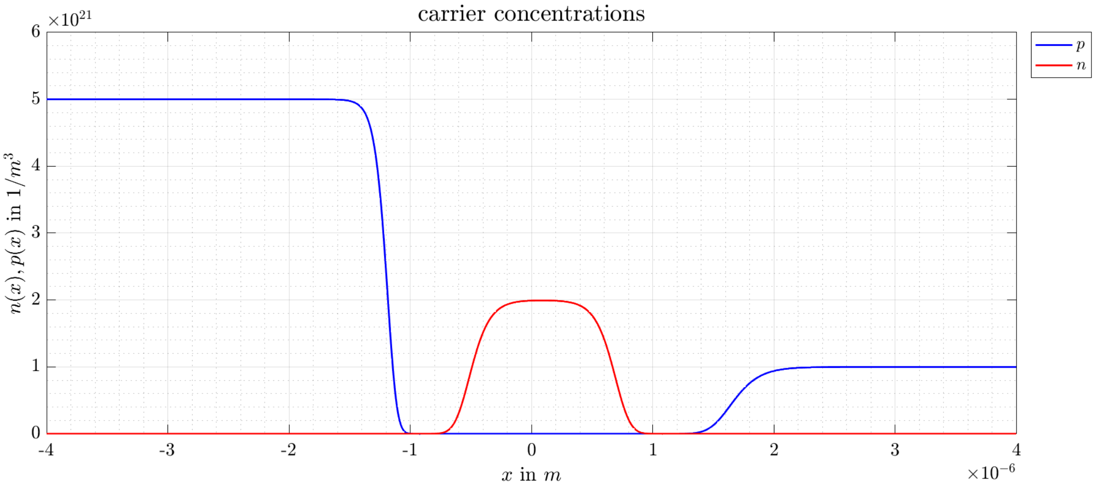
Doping Profile

```
% plot chosen doping profile
figure, hold on, grid minor;
plot(x,NA,'b','DisplayName','$N_A$','LineWidth',1.5)
plot(x,ND,'r','DisplayName','$N_D$','LineWidth',1.5)
plot(x,(ND-NA),'g--','DisplayName','$N_D - N_A$','LineWidth',1.5)
title('doping concentrations')
xlabel('$x$ in $m$')
ylabel('$N(x)$ in $1$/$m^3$')
legend_obj = legend();
legend_obj.Location = 'northeastoutside';
hold off;
```



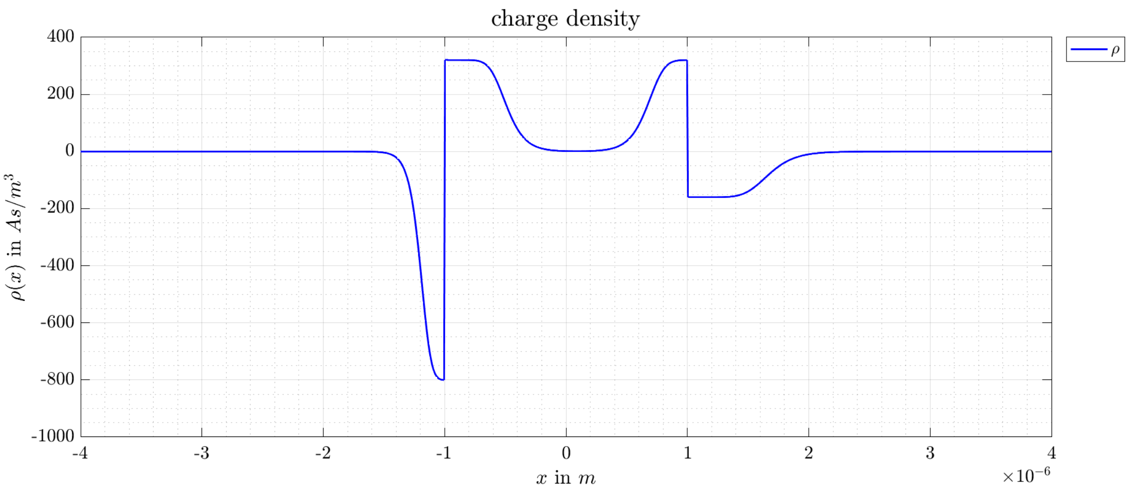
Charge Carrier Concentrations

```
figure, hold on, grid minor;
plot(x,p,'b','DisplayName','$p$','LineWidth',1.5)
plot(x,n,'r','DisplayName','$n$','LineWidth',1.5)
title('carrier concentrations')
xlabel('$x$ in $m$')
ylabel('$n(x),p(x)$ in $1$/$m^3$')
legend_obj = legend();
legend_obj.Location = 'northeastoutside';
hold off;
```



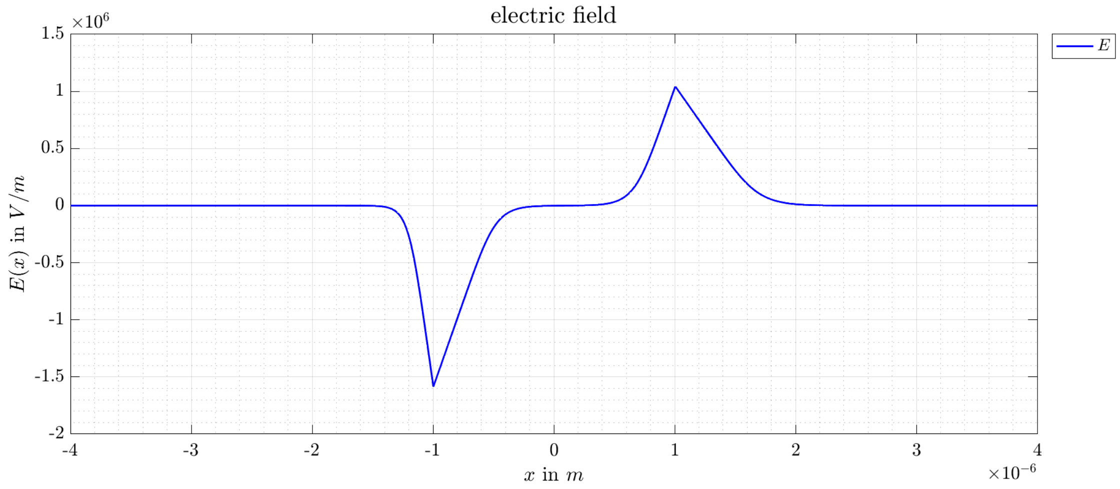
Charge Density

```
figure, hold on, grid minor;
plot(x,rho,'b','DisplayName','$\rho$','LineWidth',1.5)
title('charge density')
xlabel('$x$ in $m$')
ylabel('$\rho(x)$ in $As$/$m^3$')
legend_obj = legend();
legend_obj.Location = 'northeastoutside';
hold off;
```



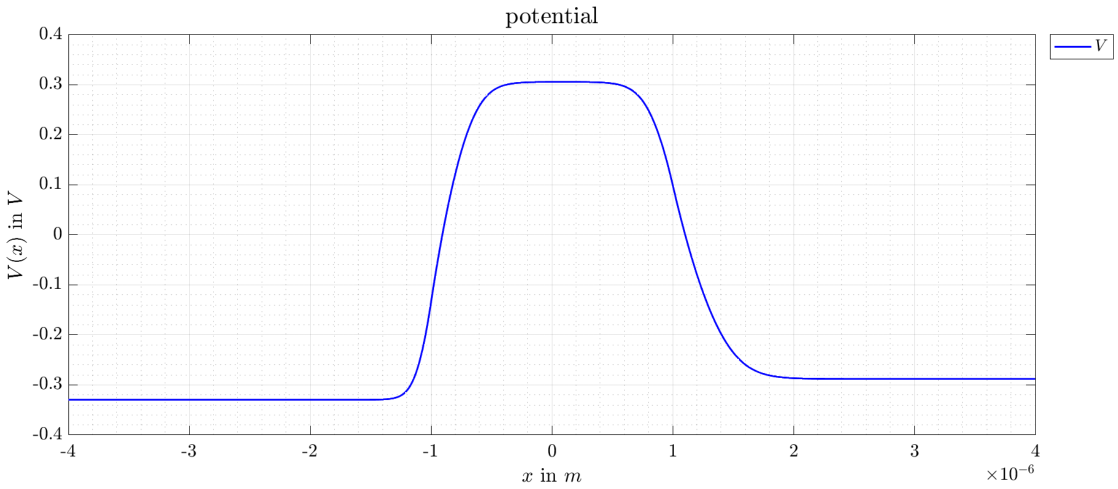
Internal Electric Field

```
figure, hold on, grid minor;
plot(x,E,'b','DisplayName','$E$','LineWidth',1.5)
title('electric field')
xlabel('$x$ in $m$')
ylabel('$E(x)$ in $V$/$m$')
legend_obj = legend();
legend_obj.Location = 'northeastoutside';
hold off;
```



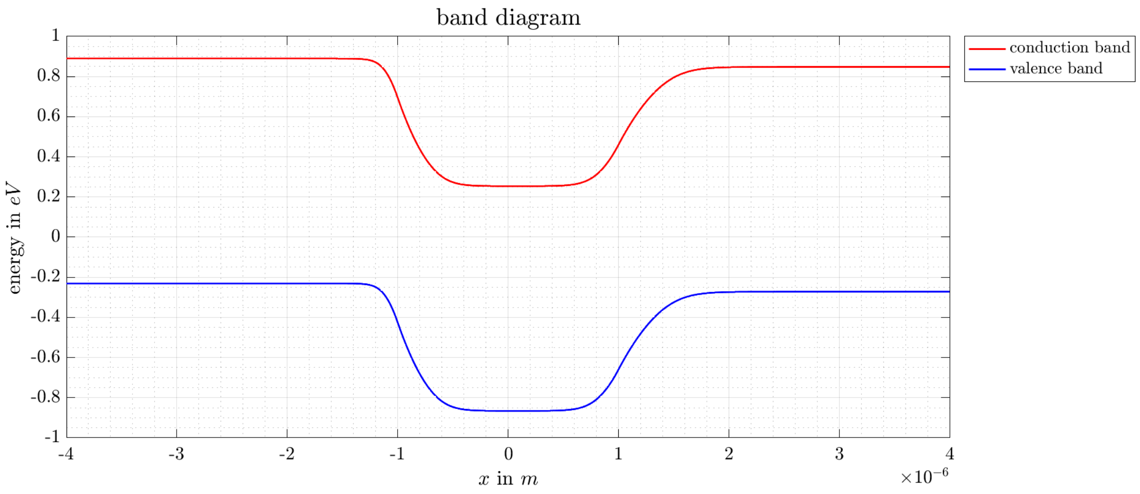
Build-in Potential

```
figure, hold on, grid minor;
plot(x,V(2:N+1),'b','DisplayName','$V$','LineWidth',1.5)
title('potential')
xlabel('$x$ in $m$')
ylabel('$V(x)$ in $V$')
legend_obj = legend();
legend_obj.Location = 'northeastoutside';
hold off;
```



Band Diagram

```
figure, hold on, grid minor;
plot(x,-V(2:N+1)+Eg/2,'r','DisplayName','conduction band','LineWidth',1.5)
plot(x,-V(2:N+1)-Eg/2,'b','DisplayName','valence band','LineWidth',1.5)
title('band diagram')
xlabel('$x$ in $m$')
ylabel('energy in $eV$')
legend_obj = legend();
legend_obj.Location = 'northeastoutside';
hold off;
```



```
function V = band_initialization(NA,ND,ni,q,kB,T)

  % if needed, add bias voltage to this function

  dN = ND - NA;
  V = zeros(size(dN));

  for i=1:length(V)
    if( abs(dN) >= ni )
      V(i) = sign(dN(i))*kB*T/q*log(abs(dN(i))/ni);
    end
  end
end

function [NA,ND] = doping_profiles(str,x)
  switch str
    case 'abrupt'
      NA = 1E14*ones(size(x));                                   % 1/cm^3
      NA = NA*1e6;                                               % 1/m^3
      ND = [ zeros(size(x(x<0))); ones(size(x(x>=0))) ] * 2E14; % 1/cm^3
      ND = ND*1e6;                                               % 1/m^3
    case 'linear'
      alpha  = 1e19*100e6;  % 1/m^4
      NA = zeros(size(x));
      ND = zeros(size(x));

      for i=1:length(x)
        if( x(i) <= 0 )
```

```
          NA(i) = -alpha*x(i);
        elseif( x(i) > 0  )
          ND(i) = alpha*x(i);
        end
      end
    case 'hyper_abrupt'
      NA=1E15*exp(0.5*(x/1e-6)).*heaviside(-x); % 1/cm^3
      NA = NA*1e6;                        % 1/m^3
      ND=1E15*exp(0.5*(-x/1e-6)).*heaviside(x); % 1/cm^3
      ND = ND*1e6;                        % 1/m^3
    case 'abrupt_pnp'
      NA = ( heaviside(-x-1e-6)*5E15 + heaviside(x+1e-6)*1E15 )*1e6; % 1/m^3
      ND = ( heaviside(x+1e-6)*3E15 - heaviside(x-1e-6)*3E15 )*1e6;  % 1/m^3
    case 'abrupt_npn'
      ND = ( heaviside(-x-1e-6)*5E15 + heaviside(x+1e-6)*1E15 )*1e6; % 1/m^3
      NA = ( heaviside(x+1e-6)*3E15 - heaviside(x-1e-6)*3E15 )*1e6;  % 1/m^3
    case 'abrupt_pnpn'
      NA = ( heaviside(-x-2e-6)*1E18 + ( heaviside(x-2e-6) - heaviside(x-3e-6) )*0.8E18 )*1e6; % 1/m^3
      ND = ( ( heaviside(x+2e-6) - heaviside(x-2e-6) )*5E17 + heaviside(x-3e-6)*1.2E18 )*1e6;  % 1/m^3
    case 'abrupt_npnp'
      ND = ( heaviside(-x-2e-6)*1E18 + ( heaviside(x-2e-6) - heaviside(x-3e-6) )*0.8E18 )*1e6; % 1/m^3
      NA = ( ( heaviside(x+2e-6) - heaviside(x-2e-6) )*5E17 + heaviside(x-3e-6)*1.2E18 )*1e6;  % 1/m^3
    case 'arbitrary'
      [NA,ND] = myProfile(x);
    otherwise
      error(['[ERROR]: ',str,' is not a valid doping profile!'])
  end
end

function [NA,ND] = myProfile(x)
  NA = 1E14*ones(size(x));       % 1/cm^3
  NA = NA*1e6;                   % 1/m^3

  ND = exp( -(x/1e-6).^2 )*1e15; % 1/cm^3
  ND = ND*1e6;                   % 1/m^3
end
```

References

- [1] Jabr, R.A. & Hamad, Mohammed & Mohanna, Yasser. (2007). Newton-Raphson Solution of Poisson's Equation in a Pn Diode. International Journal of Electrical Engineering Education. 44. 23-33. 10.7227/IJEEE.44.1.3. https://www.researchgate.net/publication/233646108_Newton-Raphson_Solution_of_Poisson%27s_Equation_in_a_Pn_Diode
- [2] SZE, S. M. (1985). *Semiconductor devices, physics and technology*. New York, Wiley.
- [3] F. R. Shapiro, "The numerical solution of Poisson's equation in a pn diode using a spreadsheet," in IEEE Transactions on Education, vol. 38, no. 4, pp. 380-384, Nov. 1995, doi: 10.1109/13.473161.
- [4] Chapra, Steven C, and Raymond P. Canale. *Numerical Methods for Engineers*. Boston: McGraw-Hill Higher Education, 2006. Print.
- [5] Lecture 9 (Computational Electromagnetics) - Finite Difference Method, Dr. Raymond Rumpf, https://www.youtube.com/watch?v=v-exTNOSG3g