ECE431s
Optoelectronics project

Name	ID
Momen Khaled Moselhi	2001874
Seif Ashraf Mohamed Abd-El-Salam	2001549
Mohab Tarek Samir	2001364
Abdallah Karim Motwea	2000993
Youssef Mansour Abdelhamed	2001422

-DC Characteristics:

Code:

Functions:

1.

```
function [dNdt, dSdt] = compute_derivatives(y, J, Ntr, vg)
    tp = 1.6e-12;
    q = 1.602e-19;
    d = 0.2e-4;
    G = 0.3;
    Se = 3e-17;
    a = 2.5e-16;
    b = 1e-6;
    A = 1e+8;
    B = 1e-10;
    C = 3e-29;

    N = y(1);
    S = y(2);

    % Differential equations
    dNdt = J / (q * d) - N .* (A + B .* N + C .* (N .* 2)) - (G * vg * a .* (N - Ntr) .* S) ./ (1 + Se .* S);
    dSdt = (G * vg * a .* (N - Ntr) .* S) ./ (1 + Se .* S) + (G * b .* N) .* (A + B .* N + C .* (N .* 2)) - S / tp;
end
```

2.

```
function [J, Ntr, vg] = compute_intermediates(t, y, it, Temp, i)
    w = 2e-4;
    L = 250e-4;
    c = 3e+10;
    ng = 4;
    j = i / (w * L);
    J = interp1(it, j, t);
    Ntr = (1e+18) .* exp((Temp - 300) / 50); % Ntr at T = 300 K
    vg = c / ng;
end
```

3.

```
function Diff_equ_Solve = Diff_equ(t, y, it, Temp, i)
    [J, Ntr, vg] = compute_intermediates(t, y, it, Temp, i);
    [dNdt, dSdt] = compute_derivatives(y, J, Ntr, vg);
    Diff_equ_Solve = [dNdt; dSdt];
end
```

4.

```
% Define rate equations function
function dydt = rate_eqs(t, y, I)
    q = 1.6e-19;          % Electron charge (C)
    Gamma = 0.3;          % Confinement factor
    A_nr = 1e8;           % Nonradiative recombination rate (s^-1)
    B = 1e-10;            % Radiative recombination coefficient (cm^3/s)
    C = 3e-29;            % Auger recombination coefficient (cm^6/s)
    a = 2.5e-16;          % Gain coefficient (cm^2)
    n0 = 1e18;            % Carrier density at transparency (cm^-3)
    tau_p = 1.6e-12;      % Photon lifetime (s)
    beta_sp = 1e-5;       % Spontaneous emission factor
    vg = 7.5e9;           % Group velocity (cm/s)
    L = 250e-4;           % Cavity length (cm)
    w = 2e-4;             % Active region width (cm)
    d = 0.2e-4;           % Active region thickness (cm)
    V = L * w * d;        % Active region volume (cm^3)

    N = y(1); % Carrier density
    S = y(2); % Photon density

    % Rate equations
    dNdt = (I / (q * V)) - A_nr * N - B * N^2 - C * N^3 - Gamma * vg * a * (N - n0) * S;
    dSdt = Gamma * vg * a * (N - n0) * S - S / tau_p + beta_sp * B * N^2;

    dydt = [dNdt; dSdt];
end
```

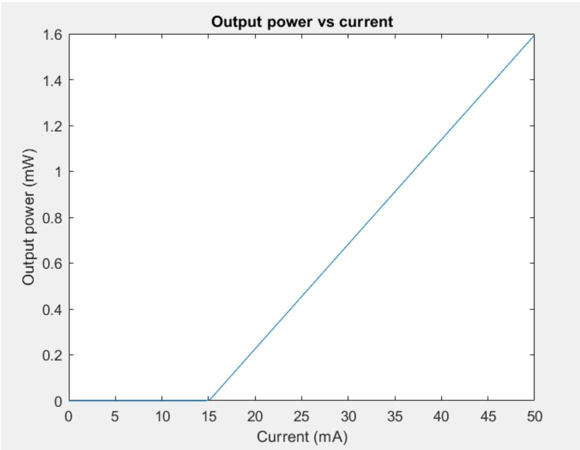
Main:

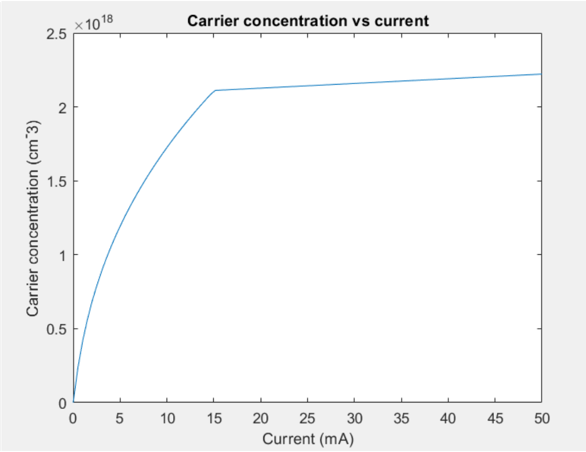
```matlab
close all
clc
clear

% Initial
p = 1000;
t_t = linspace(0, 20e-9, p);

% DC 0 to 50mA
Ivaln = 100;
I = linspace(0, 50e-3, Ivaln);
Carrier_density = zeros(1, Ivaln);
Photon_density = zeros(1, Ivaln);

for i = 1:Ivaln
    [~, y] = ode23s(@(t, y) Diff_equ(t, y, t_t, 300, I(i) .* ones(1, p)), t_t, [0 0]);
    N = y(:, 1);
    Carrier_density(i) = N(end);
    S = y(:, 2);
    Photon_density(i) = S(end);
end

figure;
plot(1e+3 .* I, Carrier_density);
grid;
title('Carrier density Vs Current');
xlabel('I (mA)');
ylabel('Carrier density');

figure;
plot(1e+3 .* I, Photon_density);
grid on;
title('Photon density Vs Current');
xlabel('I (mA)');
ylabel('Photon density');

% Transient 0 to 50mA
Ivaln = 6;
I = linspace(0, 50e-3, Ivaln);
Carrier_density = zeros(Ivaln, p);
Photon_density = zeros(Ivaln, p);
```

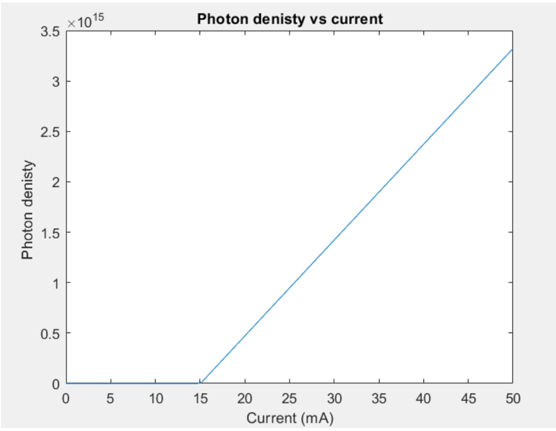
```
for i = 1:Ivaln
    [~, y] = ode23s(@(t, y) Diff_equ(t, y, t_t, 300, I(i) .* ones(1, p)), t_t, [0 0]);
    N = y(:, 1);
    Carrier_density(i, :) = N;
    S = y(:, 2);
    Photon_density(i, :) = S;
end

for i = 1:Ivaln
    figure;
    plot(t_t .* 1e9, Carrier_density(i, :));
    title([num2str(10 * (i - 1)), ' mA']);
    xlabel('Time (ns)');
    ylabel('Carrier density');
    grid on;
end

for i = 1:Ivaln
    figure;
    plot(t_t .* 1e9, Photon_density(i, :));
    title([num2str(10 * (i - 1)), ' mA']);
    xlabel('Time (ns)');
    ylabel('Photon density');
    grid on;
end
```

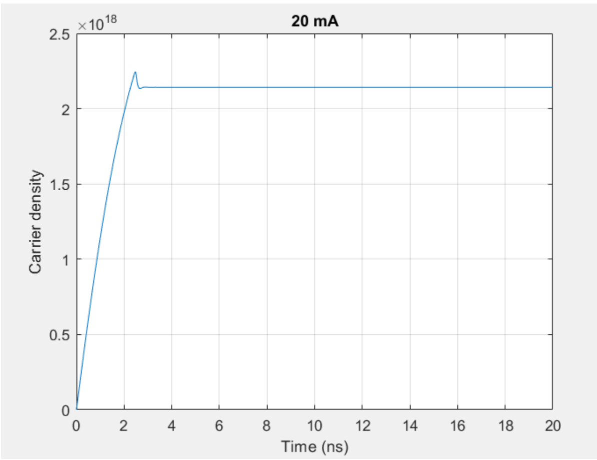
```
% DC Current Sweep
I_dc = linspace(0, 50e-3, 100); % Linearly increasing current (A)
P_out = zeros(size(I_dc));      % Output power array

% Solve rate equations for each DC current
for i = 1:length(I_dc)
    [~, Y] = ode23s(@(t, y) rate_eqs(t, y, I_dc(i)), [0, 10e-6], [0, 0]);
    S = Y(end, 2); % Steady-state photon density
    P_out(i) = h * nu * Gamma * S; % Optical power (W)
end

% Plot Output Power vs DC Current
figure;
plot(I_dc * 1e3, P_out * 1e3, 'LineWidth', 1.5); % Convert A to mA and W to mW
xlabel('DC Current (mA)');
ylabel('Output Power (mW)');
title('Output Power vs. DC Current');
grid on;

disp('Output Power vs. DC Current plotted.');
```

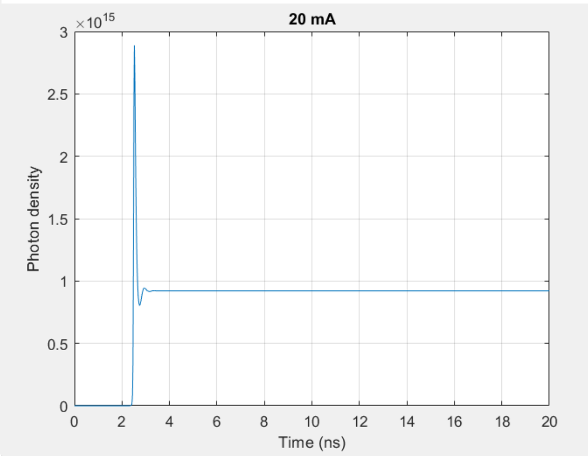
Plots:











Comment

Carrier density increases approximately linearly by increasing the pumping current till reaching the threshold current then saturates at Nth = 2.1 $*10^{18}\ cm^{-3}$

Photon density is zero till reaching the threshold current then increases linearly

Output power is zero till reaching the threshold current then increases linearly

$I_{th}=15\ mA$

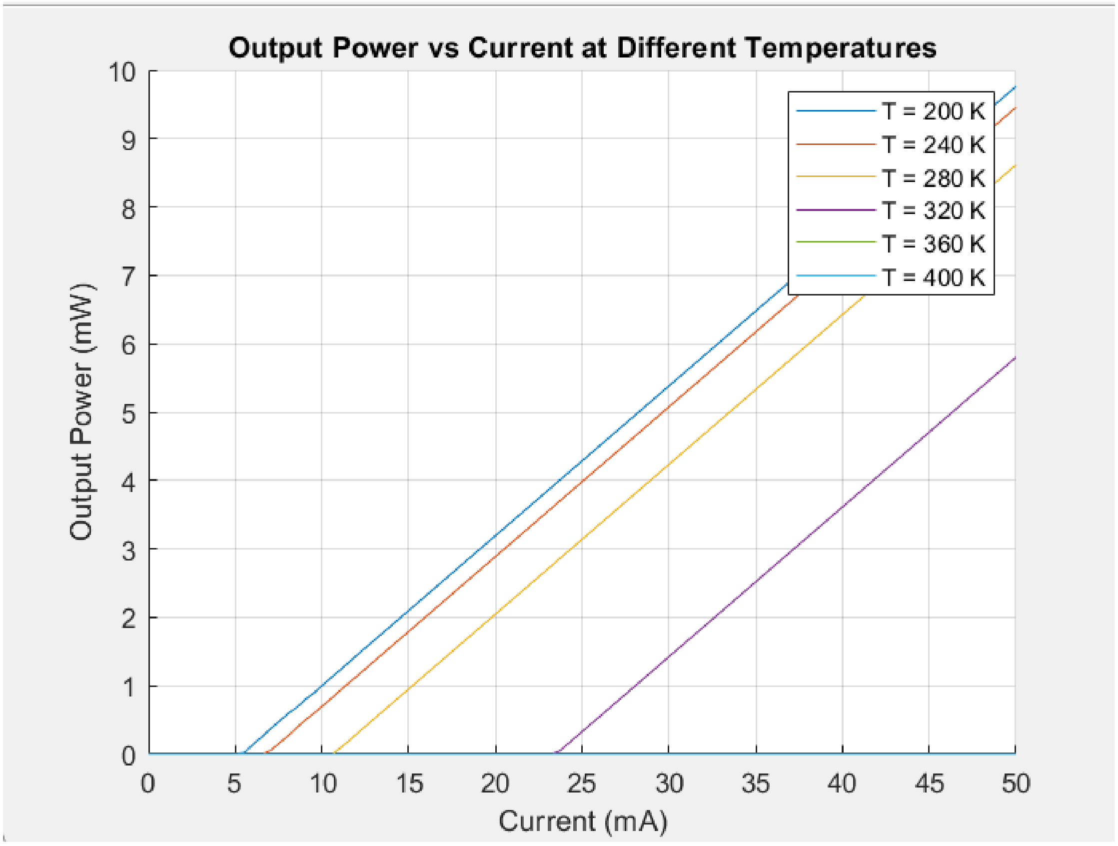
Output results is same as expected from analytical solution

-DC Characteristics:

Code:

```
function dydt = odeOT1(t,y,it,i,T)
    T0=50;
    N = y(1);
    S = y(2);
    A=1e+8;
    B=1e-10;
    C=3e-29;

    i = i * exp(T / T0); % Threshold current
    Ntr = 1e+18 * exp((T - 300) / T0); % Transparency carrier density
    G = 0.3* B*((3*10^8)/4)*(N - Ntr); % Adjust gain with N_tr
    w = 2e-4;
    L = 250e-4;
    j = i/(w*L);
    J = interp1(it, j, t);

    tp = 1.6e-12;
    q = 1.602e-19;
    d = 20e-6;

    c = 3e10;
    ng = 4;
    vg = c/ng;
    Se = 3e-17;
    a =2.5e-16;
    b=1e-6;

    dNdt = J/(q*d) - N.*(A + B.*N + C.*(N.^2)) - (G*vg*a.*(N - Ntr).*S)./(1+Se.*S);
    dSdt = (G*vg*a.*(N-Ntr).*S)./(1+Se.*S) + (G*b.*N).*(A + B.*N + C.*(N.^2)) - S/tp;
    dydt = [dNdt; dSdt];
    end

for j = 1:TempValn
    for i = 1:Ivaln-1
        [~, y] = ode23s(@(t, y) odeOT1(t, y, tt, Ival(i) .* ones(1, p), TempVal(j)), tt, ic);

        N = y(:, 1);
        NT(j, i) = N(end);

        S = y(:, 2);
        ST(j, i) = S(end);
    end

    % Calculate output power
    P_out = ST(j, :) * 1.6e-19 * 0.3 * 1e-12;

    % Plot the data
    plot(1e+3 .* Ival, P_out);

    % Add the label for the current temperature
    legendLabels{j} = ['Temp ' num2str(TempVal(j))];
end

% Add the legend after the loop
legend(legendLabels, 'Location', 'best');

hold off;
```

Plots:



- Comment

Output power diminishes at higher temperatures for the same current This reduction in output power is a direct consequence of lower photon density and reduced overall efficiency of the system at elevated temperatures.
Threshold current varies exponentially with temperature, so as temperature increase, Ith increases $I_{th} = I_o\, e\text{^}(T/T_o)$
Threshold Carrier density increases with increasing temperature as:

$$\mathbf{N_{th}} = N_{tr} + \frac{1}{v_g \Gamma \gamma_m \tau_{ph}}$$

$N_{tr} = N_{tro}\, e\text{^}(T-300)/50$

-AC Analysis

Code:

```
%% constants
e_charge = 1.6e-19;
cavity_length = 250e-6;                                        %(m)
active_width = 2e-6;                                           %(m)
active_thickness = 0.2e-6;                                     %(m)
active_volume = cavity_length * active_width * active_thickness;   %(m^3)
confinement_factor = 0.3;
nonradiative_rate = 1e8;                                       %(s^-1)
Bsp = 1e-16;                                                   %Radiative recombination coefficient (m^3/s)
auger_coeff = 3e-41;                                           %(m^6/s)
gain_coeff = 2.5e-20;                                          %(m^2)
Ntr = 1e24;                                                    %Carrier density at transparency (m^-3)
carrier_lifetime = 1.6e-12;                                    %(s)
spontaneous_emission_factor = 1e-5;
ng = 4;                                                        %Group index
group_velocity = 3e8 / ng;                                     %(m/s)
h = 6.63e-34;                                                  %Planck's constant (J·s)
log_term = log((ng + 1)^2 / (ng - 1)^2);
optical_frequency = 2.3e14;                                    %(Hz)

frequency_range = logspace(8, 10, 100);
time_span = linspace(0, 10e-9, 1000);
currents = [0.75e-3 1e-3 1.25e-3 1.5e-3];
photon_density_ac = zeros(100, 10);
```

```
%% Define rate equations
rate_equations = @(t, y, current) [
(current / (e_charge * active_volume)) - nonradiative_rate * y(1) - Bsp * y(1)^2 - auger_coeff * y(1)^3 - (confinement_factor * group_velocity * gain_coeff * (y(1) - Ntr) * y(2));
(confinement_factor * group_velocity * gain_coeff * (y(1) - Ntr) * y(2)) - (y(2) / carrier_lifetime) + spontaneous_emission_factor * Bsp * y(1)^2
];
```

```
%% Solve equationsns for different AC signal amplitudes
initial_current = 35e-3;

figure;
for i = 1:length(currents)
    input_current = currents(i);
    for j = 1:length(frequency_range)
        frequency = frequency_range(j);
        ac_signal = initial_current + input_current * cos(2 * pi * frequency * time_span);

        % Solve the rate equations with the AC signal
        [~, y] = ode23s(@(t, y) rate_equations(t, y, ac_signal(round(t * 1e9) + 1)), time_span, [0, 0]);

        % Calculate AC component of photon density
        photon_density_dc(j, i) = mean(abs(y((end / 2 : end), 2)));
        photon_density_ac(j, i) = max(abs(y(:, 2))) - photon_density_dc(j, i);
    end

    % Plot
    subplot(2, 2, i)
    semilogx(frequency_range, 20 * log10(photon_density_ac(:, i) / photon_density_ac(1, i)), 'LineWidth', 2);
    xlabel('Frequency (Hz)');
    ylabel('Magnitude (dB)');
    title(['Frequency Response for I = ', num2str(input_current * 10^3), ' mA']);
    ylim([-0.5 0.2]);
    grid on;
end
```

```matlab
%% DC Current and Power Calculation
dc_current_values = [20e-3 24e-3 25e-3 30e-3];
output_power = zeros(size(dc_current_values));      % Output power array
resonant_freq = zeros(size(dc_current_values));
photon_density_dc_signal = zeros(size(frequency_range));

figure;
for i = 1:length(dc_current_values)
    input_current = 1e-3;
    dc_current = dc_current_values(i);
    for j = 1:length(frequency_range)
        frequency = frequency_range(j);
        ac_signal = dc_current + input_current * cos(2 * pi * frequency * time_span);
        % Solve the rate equations with the AC signal
        [~, y] = ode23s(@(t, y) rate_equations(t, y, ac_signal(round(t * 1e9) + 1)), time_span, [0, 0]);

        % Calculate AC component of photon density
        photon_density_dc_signal(j, i) = mean(abs(y((end / 2 : end), 2)));
        photon_density_ac(j, i) = max(abs(y(:, 2))) - photon_density_dc_signal(j, i);
    end

    [max_photon_density_ac, max_idx] = max(photon_density_ac(:,i));
    output_power(i) = photon_density_dc_signal(max_idx, i) * active_width * active_thickness * group_velocity * log_term * h * optical_frequency * 0.5;
    resonant_freq(i) = frequency_range(max_idx);

    subplot(2, 2, i);
    % Plot frequency response
    semilogx(frequency_range, 20 * log10(photon_density_ac(:, i) / photon_density_ac(1, i)), 'LineWidth', 2, 'Color', [0, 0.4470, 0.7410]);
    xlabel('Frequency (Hz)', 'FontWeight', 'bold');
    ylabel('Magnitude (dB)', 'FontWeight', 'bold');
    title(['Frequency Response for I = ', num2str(dc_current * 10^3), ' mA']);
    grid on;
end
```

```matlab
% Constants and parameters
electron_charge = 1.6e-19;  % Electron charge (C)
length_cm = 250e-4;         % Length (cm)
width_cm = 2e-4;            % Width (cm)
thickness_cm = 0.2e-4;      % Thickness (cm)
volume_cm3 = length_cm * width_cm * thickness_cm;  % Volume (cm^3)
confinement_factor = 0.3;   % Confinement factor
recomb_nonrad = 1e8;        % Non-radiative recombination coefficient
recomb_rad = 1e-10;         % Radiative recombination coefficient
recomb_auger = 3e-29;       % Auger recombination coefficient
gain_coeff = 2.5e-16;       % Gain coefficient
transparency_density = 1e18; % Transparency carrier density
carrier_lifetime = 2.2e-9;  % Carrier lifetime (s)
photon_lifetime = 1.6e-12;  % Photon lifetime (s)
spont_emission = 1e-5;      % Spontaneous emission factor
group_velocity = 7.5e9;     % Group velocity (cm/s)
chirp_coeff = 5;            % Chirping coefficient
gain_term = 5.62e1 * 5;     % Gain term (1/cm)
dc_current = 35e-3;         % DC current (A)
ac_current = 5e-3;          % AC current amplitude (10 mA)
modulation_frequency = 1e9; % Modulation frequency (1 GHz)
```

```matlab
% Simulation time span extended to capture more modulation cycles
time_span = linspace(0, 20e-9, 5000);  % Time span for the simulation (s)

% Modulation current as a function of time
current_modulated = @(t) dc_current + ac_current * sin(2 * pi * modulation_frequency * t);

% Define rate equations for carrier density (N), photon density (S), and phase (phi)
rate_equations = @(t, state) [
    % Carrier density (modulated)
    (current_modulated(t) / (electron_charge * volume_cm3)) ...
    - recomb_nonrad * state(1) - recomb_rad * state(1)^2 - recomb_auger * state(1)^3 ...
    - (confinement_factor * group_velocity * gain_coeff * (state(1) - transparency_density) * state(2));

    % Photon density
    (confinement_factor * group_velocity * gain_coeff * (state(1) - transparency_density) * state(2)) ...
    - (state(2) / photon_lifetime) + spont_emission * recomb_rad * state(1)^2;

    % Phase
    0.5 * chirp_coeff * gain_term * volume_cm3 * (state(1) - state(4));

    % Carrier density (unmodulated)
    (dc_current / (electron_charge * volume_cm3)) ...
    - recomb_nonrad * state(4) - recomb_rad * state(4)^2 - recomb_auger * state(4)^3 ...
    - (confinement_factor * group_velocity * gain_coeff * (state(1) - transparency_density) * state(2))
];
```

```matlab
% Initial conditions for [N_mod, S_mod, phi_mod, N_unmod]
initial_conditions = [0, 0, 0, 0];

% Solve the system of ODEs using ode45
[~, states] = ode45(@(t, state) rate_equations(t, state), time_span, initial_conditions);

% Extract modulated and unmodulated carrier densities and photon density and phase
carrier_density_mod = states(:, 1);      % Modulated carrier density
photon_density_mod = states(:, 2);      % Modulated photon density
phase_mod = states(:, 3);               % Modulated phase
carrier_density_unmod = states(:, 4);  % Unmodulated carrier density

% Compute the electric field (E) as E(t) = sqrt(S) * exp(j * phi(t))
electric_field = sqrt(photon_density_mod) .* exp(1j * phase_mod);

% Normalize the electric field to avoid numerical issues
electric_field_norm = electric_field / max(abs(electric_field));

% Compute the FFT of the normalized electric field
fft_electric_field = fft(electric_field_norm(length(electric_field)/2:end));

% Shift the FFT to center the zero frequency
fft_shifted = fftshift(fft_electric_field);

% Frequency axis for FFT (positive and negative frequencies)
frequency_axis = (-length(fft_electric_field)/2:length(fft_electric_field)/2-1) ...
    * (1 / (time_span(2) - time_span(1))) / length(fft_electric_field);
```

```matlab
% Plot the FFT of the normalized electric field in the frequency domain
figure;
plot(frequency_axis / 1e9, abs(fft_shifted), 'LineWidth', 1.5);  % Plot in GHz
xlim([-10 10]);  % Limit frequency axis from -10 GHz to +10 GHz
xlabel('Frequency (GHz)');
ylabel('Magnitude (Normalized Electric Field FFT)');
title('FFT of Normalized Electric Field (Frequency Domain)');
grid on;
```

Plots:

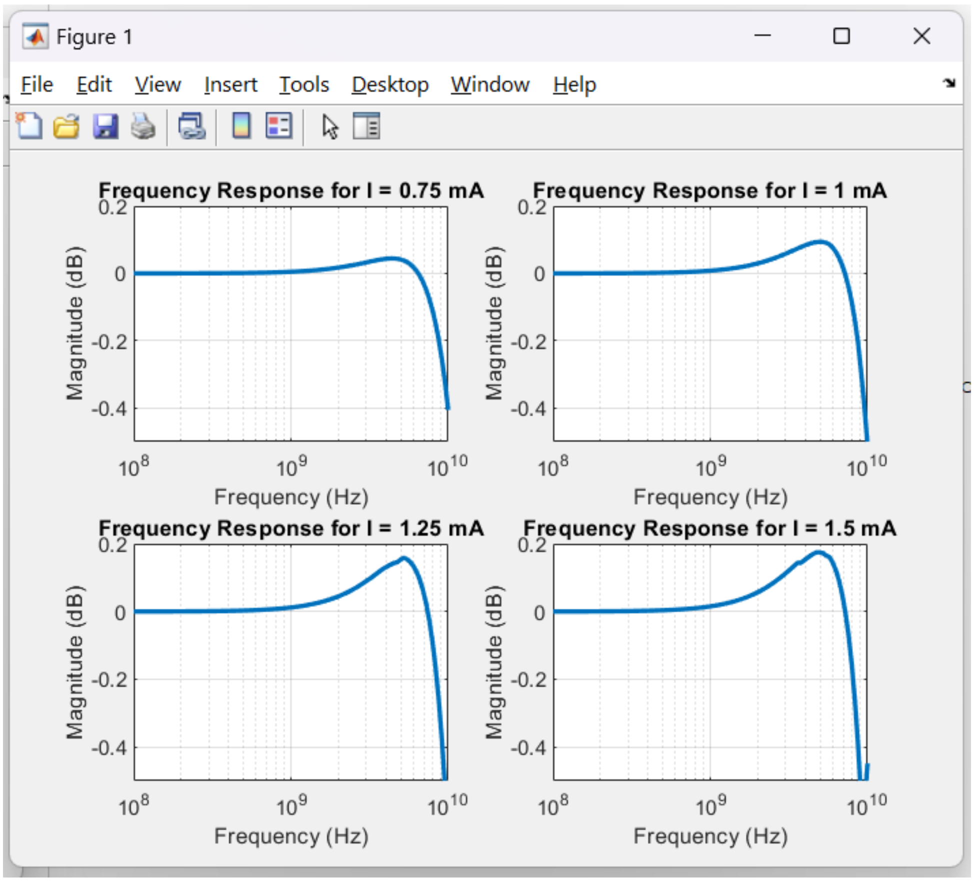


Figure 1
File Edit View Insert Tools Desktop Window Help
Frequency Response for I = 0.75 mA
Frequency Response for I = 1 mA
Frequency Response for I = 1.25 mA
Frequency Response for I = 1.5 mA
Magnitude (dB)
Frequency (Hz)
0.2
0
-0.2
-0.4
10^8
10^9
10^{10}

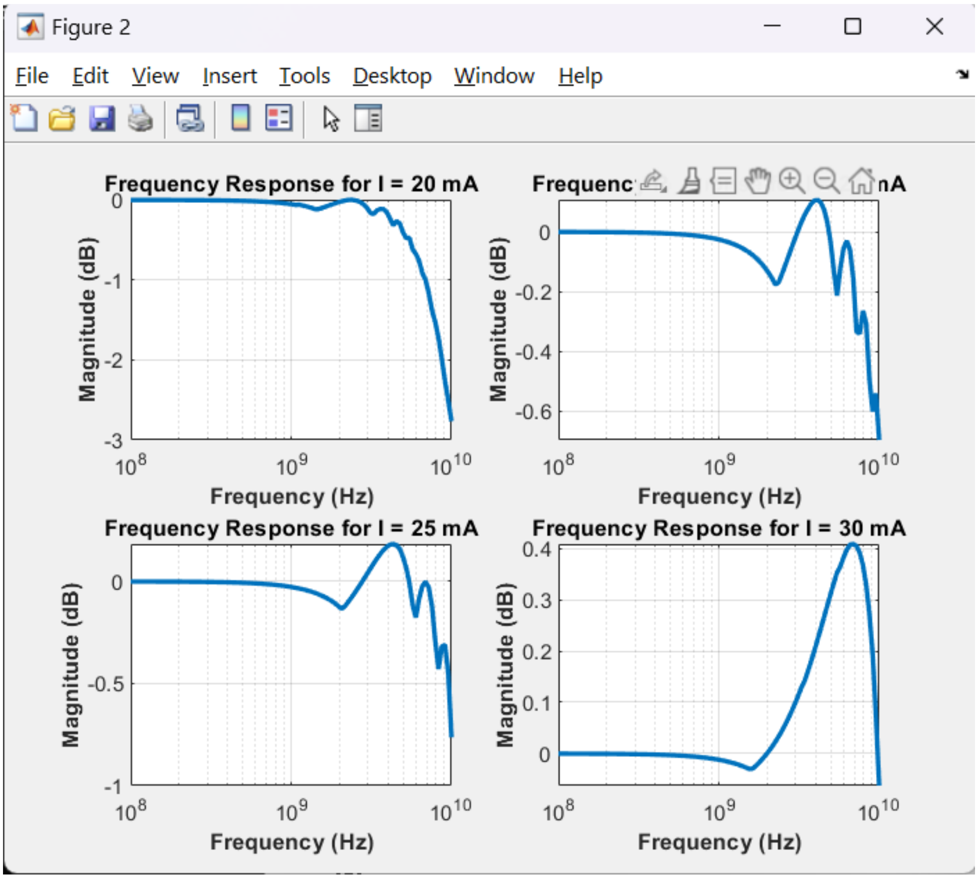


Figure 2
File Edit View Insert Tools Desktop Window Help
Frequency Response for I = 20 mA
Frequenc
nA
Frequency Response for I = 25 mA
Frequency Response for I = 30 mA
Magnitude (dB)
Frequency (Hz)
0
-1
-2
-3
-0.2
-0.4
-0.6
-0.5
0.4
0.3
0.2
0.1
10^8
10^9
10^{10}

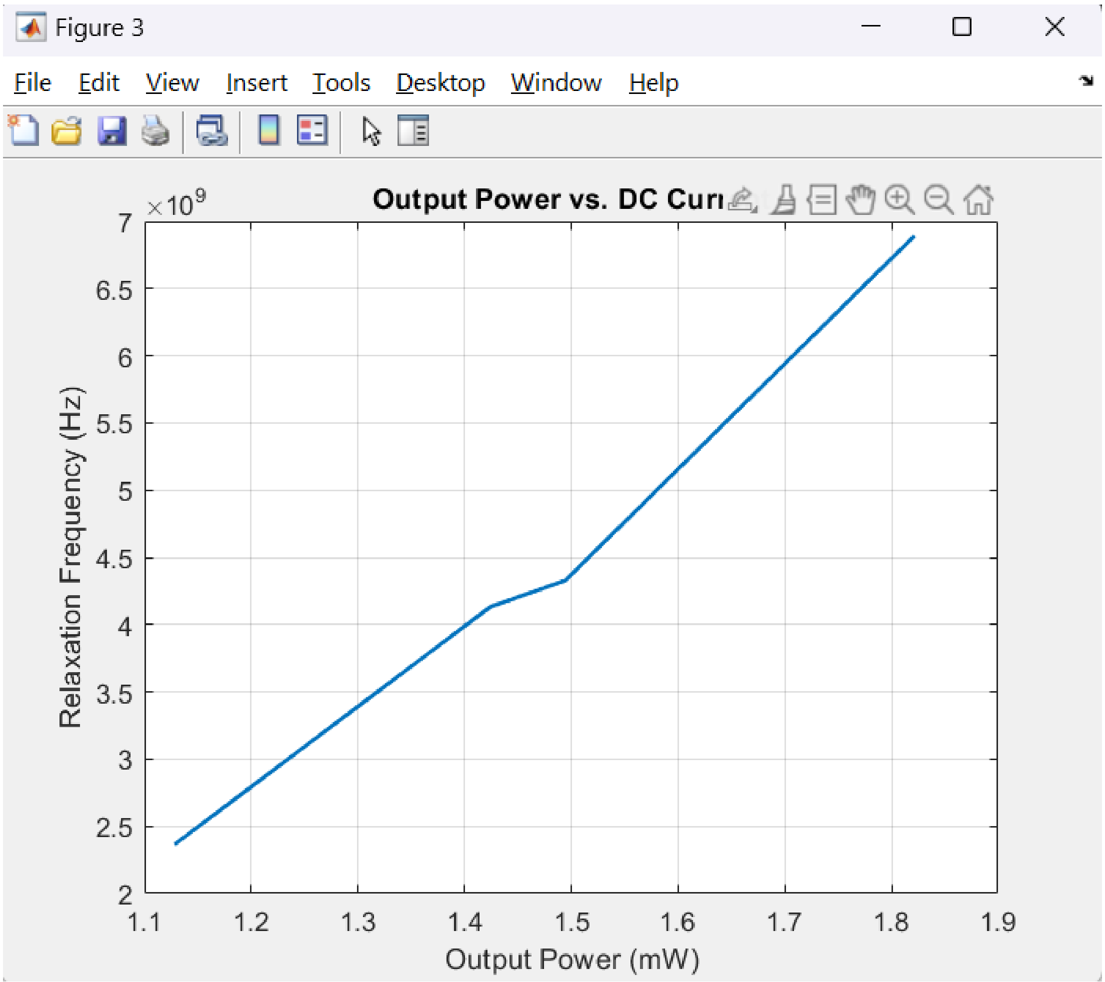
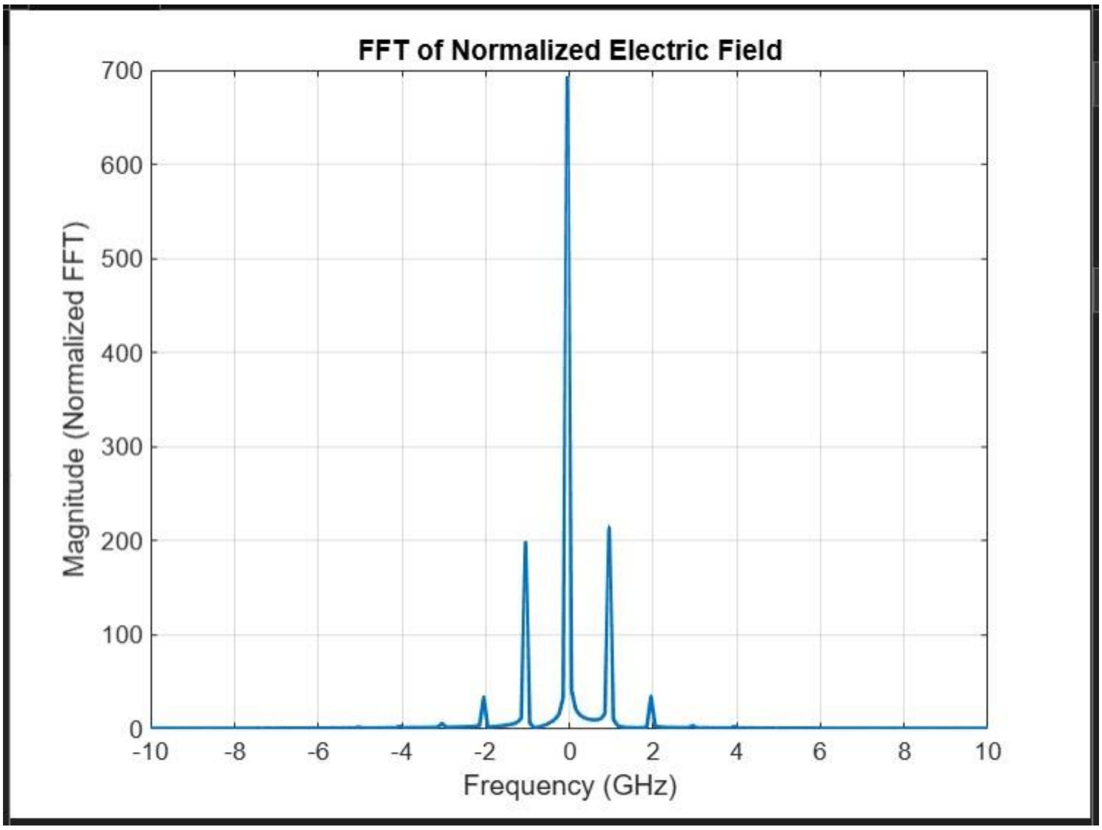


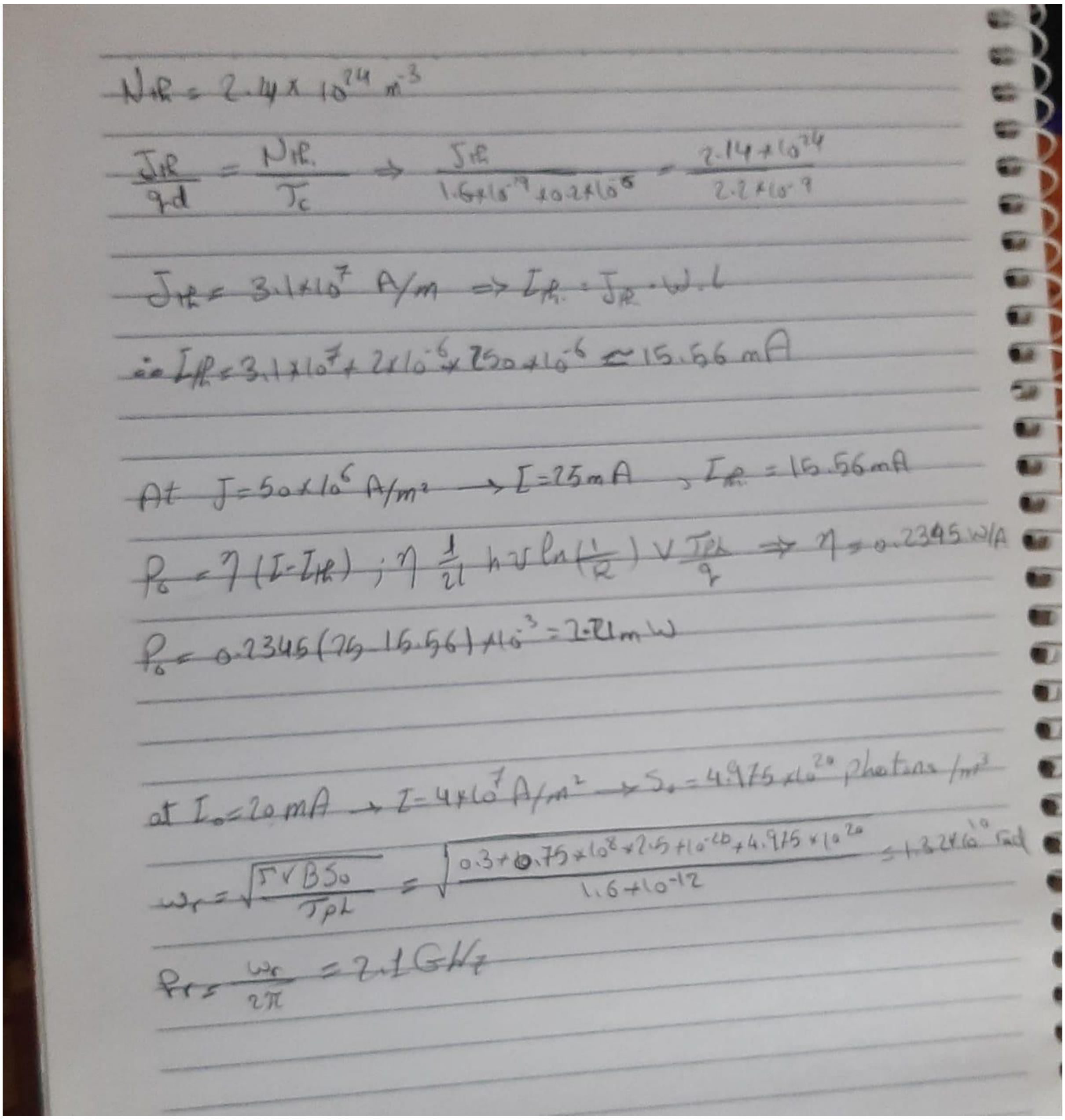
Figure 3
File Edit View Insert Tools Desktop Window Help
Output Power vs. DC Curr
×10^9
Relaxation Frequency (Hz)
Output Power (mW)


FFT of Normalized Electric Field
Magnitude (Normalized FFT)
Frequency (GHz)

Comment:

As the relaxation frequency increases the ripples in the frequency response increases.

- ## Hand Analysis



$N_{th} = 2.14 \times 10^{24}\ m^{-3}$

$\frac{J_{th}}{qd} = \frac{N_{th}}{\tau_c} \rightarrow \frac{J_{th}}{1.6\times10^{-19} \times 0.2\times10^{-6}} = \frac{2.14\times10^{24}}{2.2\times10^{-9}}$

$J_{th} = 3.1\times10^{7}\ A/m \Rightarrow I_{th} = J_{th} \cdot W \cdot L$

$\therefore I_{th} = 3.1\times10^{7} \times 2\times10^{-6} \times 250\times10^{-6} \simeq 15.56\ mA$

At $J = 50\times10^{6}\ A/m^2 \rightarrow I = 25\ mA$, $I_{th} = 15.56\ mA$

$P_o = \eta(I - I_{th})$; $\eta\ \frac{1}{2L} h\nu \ln\left(\frac{1}{R}\right) v \frac{\tau_{ph}}{q} \Rightarrow \eta = 0.2345\ W/A$

$P_o = 0.2345(25 - 15.56)\times10^{-3} = 2.21\ mW$

at $I_o = 20\ mA \rightarrow J = 4\times10^{7}\ A/m^2 \rightarrow S_o = 4.975\times10^{20}$ photons$/m^3$

$\omega_r = \sqrt{\frac{\Gamma v B S_o}{\tau_{ph}}} = \sqrt{\frac{0.3\times0.75\times10^{8}\times2.5\times10^{-20}\times4.975\times10^{20}}{1.6\times10^{-12}}} = 1.32\times10^{10}$ rad

$f_r = \frac{\omega_r}{2\pi} = 2.1\ GHz$