Engineering Electromagnetic Theory

Laboratory 3

Dec 7, 2022

Wang Zhuoyang
12112907

Contents

A Introduction

This is the report for Engineering Electromagnetic Theory Lab 3.

This time we will illustrate the distribution of the spatial magnetic field produced by a pair of current loops and learn an intuitive way to calculate the magnetic field distribution and plot the relevant graphs using MATLAB.

B Case 1: Magnetic Field Distribution of Helmholtz Coils

In this case, we build up a scene where two current loop with the same radius $a = 2$ (m), current $I = 500$ (A) and current direction are placed at $(0, 0, 1)$ (m) and $(0, 0, -1)$ (m) respectively (Figure 1).

We are going to analyze the magnetic field intensity distribution (directions and magnitudes) and the magnetic line distribution.

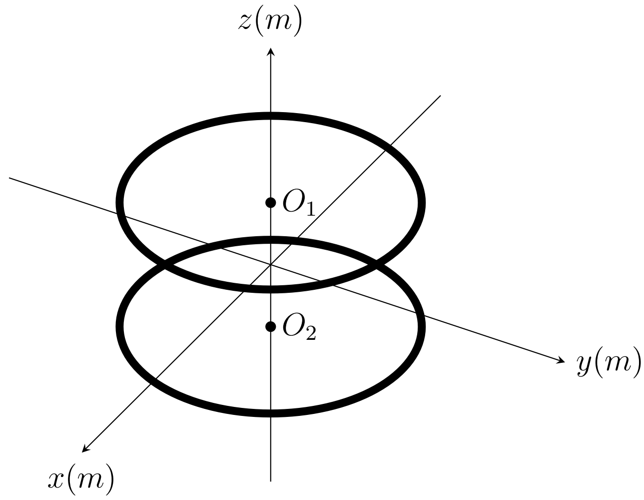

Figure 1: The sketch of the model in Case 1

B.1 Declarations

The first step is to declare the parameters to be used for calculation and display.

It is worthwhile noticing that the parameters defined below may be modified to cater to the following questions. For example, the first two figures need lower *sampling_density* to make the elements clearer, while the last figure needs higher *sampling_density* to compensate for the errors the function *streamline* brings, and reasonable lower *segment_number* to speed up.

```
%% Initialization
clear;
clc;

%% Declarations
% Basic Parameters
a = 2;    % m
I = 500; % A
d = 2;    % m
% Scene
segment_number = 30;
segment_length = 2 * pi * a / segment_number;
angles = linspace(0, 2 * pi, segment_number);
% Viewport
sampling_density = 200;
length_y = 6;
length_z = 3;
sampling_number_y = sampling_density * length_y + 1;
sampling_number_z = sampling_density * length_z + 1;
range_y = linspace(-length_y / 2, length_y / 2, sampling_number_y);
range_z = linspace(-length_z / 2, length_z / 2, sampling_number_z);
```

B.2 Magnetic Field Intensity Vector Distribution

B.2.1 MATLAB Code

Here is the code. For clarity we use the complete vectors rather than the components in the calculation. Thus we can use *cross* to calculate the cross product instead of applying the formula manually.

```
%% Magnetic Field Intensity Vector Distribution
Hy = zeros(sampling_number_y, sampling_number_z);
Hz = zeros(sampling_number_y, sampling_number_z);
% Iterate the mesh points
for it_y = 1 : sampling_number_y
    for it_z = 1 : sampling_number_z
```

```
        % Obtain real position of the mesh point
        P = [0, ...
            (it_y - 1) / sampling_density - length_y / 2, ...
            (it_z - 1) / sampling_density - length_z / 2];
        % Iterate the loops
        for S_z = [-d / 2, d / 2]
            % Iterate the segments
            for S_angle = angles(1 : segment_number)
                % Obtain the position of current segment
                S = [a * cos(S_angle), a * sin(S_angle), S_z];
                % Obtain the displacement
                R = P - S;
                % Obtain the differential length
                dL = [-segment_length * sin(S_angle), segment_length * cos(S_angle), 0];
                % Apply Biot{Savart Law
                dH = cross(I .* dL, R) ./ (4 .* pi .* norm(R) .^ 3);
                % Accumulate
                Hy(it_y, it_z) = Hy(it_y, it_z) + dH(2);
                Hz(it_y, it_z) = Hz(it_y, it_z) + dH(3);
            end
        end
    end
end

% Plot the figure
figure(1);
grid on, axis equal, hold on;
[mesh_y, mesh_z] = meshgrid(range_y, range_z);
quiver(mesh_y, mesh_z, Hy', Hz');
plot(a, d / 2, 'ro', -a, d / 2, 'bo', a, -d / 2, 'ro', -a, -d / 2, 'bo');
plot([a, -a], [d / 2, d / 2], 'b-', [-a, -a], [d / 2, -d / 2], 'g--', [-a, a], [-d / 2, -d / 2], 'b-', [a, a], [-d / 2, d / 2], 'g--')
axis([-length_y / 2, length_y / 2, -length_z / 2, length_z / 2]);
set(gcf, 'Position', [50, 50, 900, 600]);
title(["Magnetic Field Intensity Vector Distribution", "(Wang Zhuoyang, 12112907)"]);
xlabel("y (m)"), ylabel("z (m)");
saveas(1, "../fig/1_1_" + sampling_density + ".png");
```

B.2.2 Result and Analysis

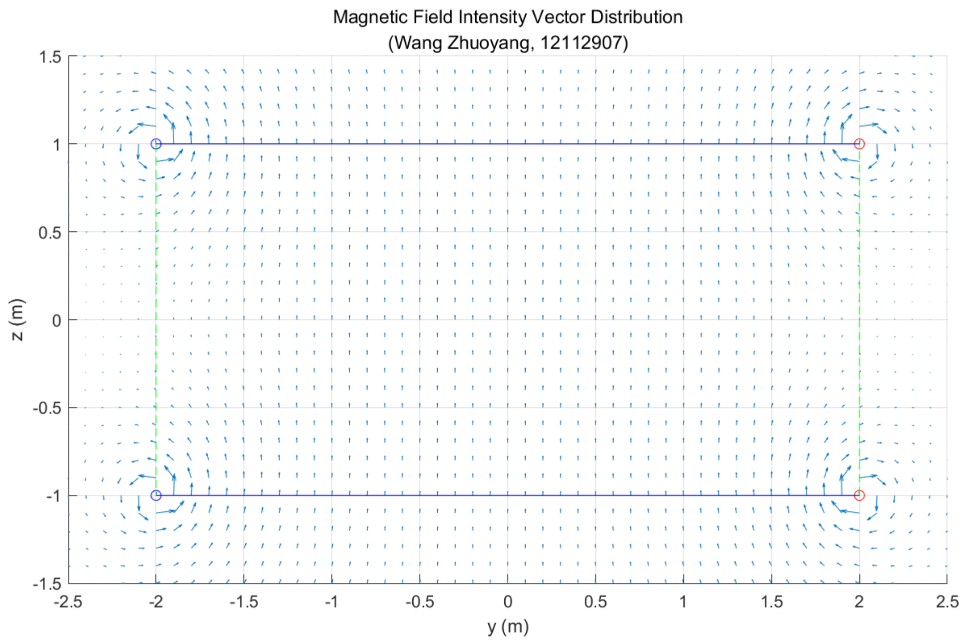

Figure 2: Magnetic field intensity vector distribution for Case 1 (yOz plane section)

From Figure 2, we can see that the magnetic field strength is larger near the cross section of the current ring and more evenly distributed inside the two isotropic current rings.

The reference show that identical current rings are often called **Helmholtz Coils** when their spacing is equal to the radius. It is characterized by a very uniform distribution of the magnetic field in the trapped space, which is confirmed by the above results.

B.3 Magnetic Field Intensity Magnitude Distribution

B.3.1 MATLAB Code

In order to observe the intensity of the magnetic field more clearly, we plot the distribution of the magnetic field intensity magnitude near the cross section of the current rings. And here is the code.

```
%% Magnetic Field Intensity Magnitude Distribution
H_norm = sqrt(Hy .^ 2 + Hz .^ 2);

% Plot the figure
figure(2);
grid on, axis equal, hold on;
[mesh_y, mesh_z] = meshgrid(range_y, range_z);
mesh(mesh_y, mesh_z, H_norm');
axis([-length_y / 2, length_y / 2, -length_z / 2, length_z / 2, 0, 500]);
set(gcf, 'Position', [50, 50, 400, 700]);
view([3, 3, 200]);
title(["Magnetic Field Intensity Magnitude Distribution", "(Wang Zhuoyang, 12112907)"]);
xlabel("y (m)"), ylabel("z (m)"), zlabel("|H| (A/m)");
saveas(2, "../fig/1_2_" + sampling_density + ".png");
```

B.3.2 Result and Analysis

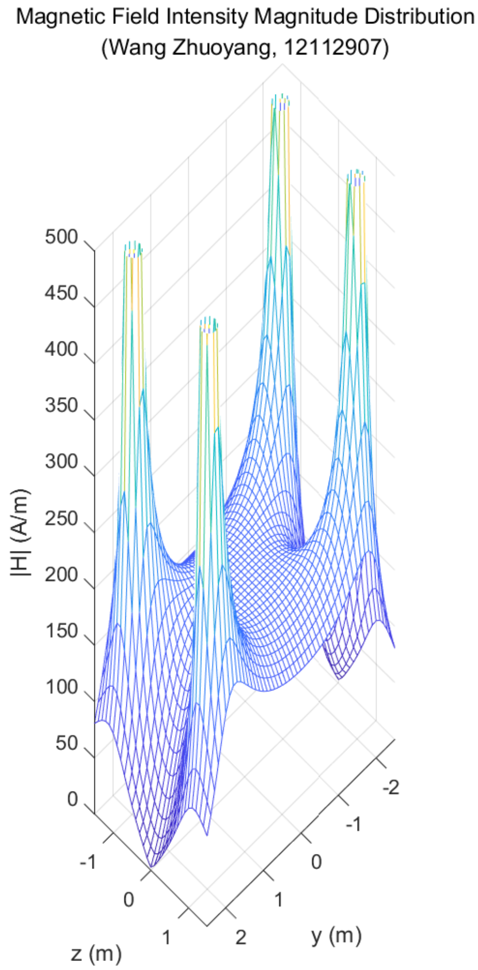

Figure 3: Magnetic field intensity magnitude distribution for Case 1 (yOz plane section)

From Figure 3, the distribution of the magnetic field strength can be seen more clearly, roughly in line with the inverse relationship near the current.

B.4 Magnetic Line Distribution

B.4.1 MATLAB Code

To visualize the trend of the magnetic field flow direction, we plot the distribution of the magnetic lines of force.

We use the function *streamline* to draw the magnetic force line, the key point of using this function is the selection of the starting point of the magnetic force line and the calculation of the gradient field. We have already obtained the gradient field, so now we need to find a good way to pick the starting point.

The basic requirement for magnetic lines is that **the sparsity of the lines is positively related to the magnetic field strength at the field point**. So we can sample the intensity of the magnetic field on a path and obtain a discrete function similar to the probability density (PDF) after normalization. The distribution function (CDF, denoted as F) is obtained by accumulating (integrating) the probability density functions (PDF).

Next, by interpolation we can obtain the inverse function of the CDF, denoted as F^{-1}. Based on the knowledge of probability theory we know that by applying F^{-1} to a set of sample points satisfying a unitary uniform distribution, we can obtain a set of sample points that match the distribution of the original random variable:

$$u \sim U(0,1),\ \ F(F_X^{-1}(u)) = F_X(x)$$

So we end up with a set of starting points on a particular path whose sparsity matches the magnetic field strength distribution. Then we can now use the function *streamline* in the appropriate direction for them.

By the way, as mentioned before, in this case we need higher *sampling_density* to compensate for the errors the function *streamline* brings, and reasonable lower *segment_number* to speed up.

Here is the code.

```
%% Magnetic Line Distribution
% Sample the magnitude of H
H_samples_range_y = [-a - 0.1, a + 0.1];
H_samples_z = d / 2.5;
H_samples_index_y = floor((H_samples_range_y(1) + length_y / 2) * sampling_density) : floor((H_samples_range_y(2) + length_y / 2) * sampling_density);
H_samples_index_z = floor((length_z / 2 + H_samples_z) * sampling_density);
H_samples = H_norm(H_samples_index_y, H_samples_index_z);

% Calculate PDF & CDF
H_pdf = H_samples ./ sum(H_samples);
H_cdf = zeros(1, length(H_pdf));
H_cdf(1) = H_pdf(1);
for it = 2 : length(H_pdf)
    H_cdf(it) = H_cdf(it - 1) + H_pdf(it);
end

% Sample basing on PDF & CDF
line_number = 64;
line_start_y = zeros(1, line_number);
line_start_z = zeros(1, line_number);
uniform_samples = linspace(0.01, 0.978, line_number);
for it_u = 1 : line_number
    u = uniform_samples(it_u);
    if u < H_cdf(1)
        left = 0;
        right = H_samples_range_y(1);
    else
        for it_s = 1 : (length(H_cdf) - 1)
            if u >= H_cdf(it_s) && u < H_cdf(it_s + 1)
                delta = (H_samples_range_y(2) - H_samples_range_y(1)) / length(H_cdf);
                left = H_samples_range_y(1) + delta * (it_s - 1);
                right = H_samples_range_y(1) + delta * it_s;
            end
        end
    end
    line_start_y(it_u) = (left + right) / 2;
    line_start_z(it_u) = H_samples_z;
end
```

```

% Plot the figure
figure(3);
grid on, axis equal, hold on;
[mesh_y, mesh_z] = meshgrid(range_y, range_z);
fig_sl = streamline(mesh_y, mesh_z, -Hy', -Hz', line_start_y, line_start_z);
set(fig_sl, "lineWidth", 0.4, "color", [0.9, 0.4, 0.8]);
fig_sl = streamline(mesh_y, mesh_z, Hy', Hz', line_start_y, line_start_z);
set(fig_sl, "lineWidth", 0.4, "color", [0.9, 0.4, 0.8]);
fig_sl = streamline(mesh_y, mesh_z, Hy', Hz', line_start_y, -line_start_z);
set(fig_sl, "lineWidth", 0.4, "color", [0.9, 0.4, 0.8]);
plot(a, d / 2, 'ro', -a, d / 2, 'bo', a, -d / 2, 'ro', -a, -d / 2, 'bo');
plot([a, -a], [d / 2, d / 2], 'b-', [-a, -a], [d / 2, -d / 2], 'g--', [-a, a],
↪ [-d / 2, -d / 2], 'b-', [a, a], [-d / 2, d / 2], 'g--')
axis([-length_y / 2, length_y / 2, -length_z / 2, length_z / 2]);
set(gcf, 'Position', [50, 50, 900, 600]);
title(["Magnetic Line Distribution", "(Wang Zhuoyang, 12112907)"]);
xlabel("y (m)"), ylabel("z (m)");
saveas(3, "../fig/1_3_" + sampling_density + ".png");
```

B.4.2 Result and Analysis

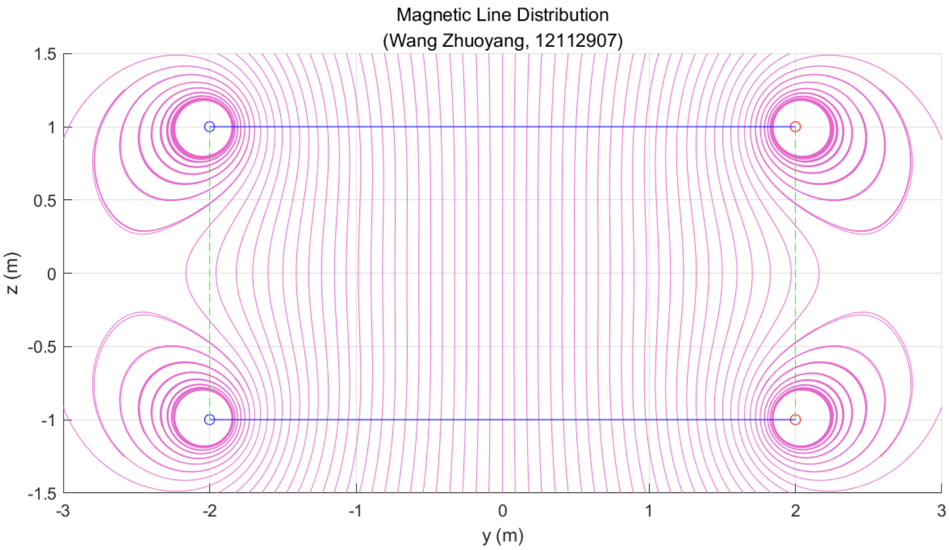

Figure 4: Magnetic line distribution for Case 1 (yOz plane section)

The result in Figure 4 are as expected. And it can be observed that the spatial magnetic field distribution inside the current ring is very uniform.

C Case 2: Magnetic Field Distribution of Current Loops With Opposite Current Direction

We change the current in the same direction in Case 1 to reverse to get the scenario in Case 2.

C.1 Declarations

Similarly, the first step is to declare the parameters to be used for calculation and display.

```
%% Initialization
clear;
clc;

%% Declarations
% Basic Parameters
a = 2;    % m
I = 500; % A
d = 2;    % m
% Scene
segment_number = 30;
segment_length = 2 * pi * a / segment_number;
angles = linspace(0, 2 * pi, segment_number);
% Viewport
sampling_density = 200;
length_y = 6;
length_z = 3;
sampling_number_y = sampling_density * length_y + 1;
sampling_number_z = sampling_density * length_z + 1;
range_y = linspace(-length_y / 2, length_y / 2, sampling_number_y);
range_z = linspace(-length_z / 2, length_z / 2, sampling_number_z);
```

C.2 Magnetic Field Intensity Vector Distribution

C.2.1 MATLAB Code

The process is exactly the same as Case 1 and will not be repeated.

```
%% Magnetic Field Intensity Vector Distribution
Hy = zeros(sampling_number_y, sampling_number_z);
Hz = zeros(sampling_number_y, sampling_number_z);
% Iterate the mesh points
for it_y = 1 : sampling_number_y
    for it_z = 1 : sampling_number_z
        % Obtain real position of the mesh point
        P = [0, ...
            (it_y - 1) / sampling_density - length_y / 2, ...
            (it_z - 1) / sampling_density - length_z / 2];
        % Iterate the loops
        for S_z = [-d / 2, d / 2]
            % Iterate the segments
            for S_angle = angles(1 : segment_number)
                % Obtain the position of current segment
                S = [a * cos(S_angle), a * sin(S_angle), S_z];
                % Obtain the displacement
                R = P - S;
                % Obtain the differential length
                dL = [-segment_length * sin(S_angle), segment_length *
↪   cos(S_angle), 0] * sign(S_z);
                % Apply Biot{Savart Law
                dH = cross(I .* dL, R) ./ (4 .* pi .* norm(R) .^ 3);
                % Accumulate
                Hy(it_y, it_z) = Hy(it_y, it_z) + dH(2);
                Hz(it_y, it_z) = Hz(it_y, it_z) + dH(3);
            end
        end
    end
end

% Plot the figure
figure(1);
grid on, axis equal, hold on;
```

```
[mesh_y, mesh_z] = meshgrid(range_y, range_z);
quiver(mesh_y, mesh_z, Hy', Hz');
plot(a, d / 2, 'ro', -a, d / 2, 'bo', a, -d / 2, 'ro', -a, -d / 2, 'bo');
plot([a, -a], [d / 2, d / 2], 'b-', [-a, -a], [d / 2, -d / 2], 'g--', [-a, a],
↪  [-d / 2, -d / 2], 'b-', [a, a], [-d / 2, d / 2], 'g--')
axis([-length_y / 2, length_y / 2, -length_z / 2, length_z / 2]);
set(gcf, 'Position', [50, 50, 900, 600]);
title(["Magnetic Field Intensity Vector Distribution", "(Wang Zhuoyang,
↪  12112907)"]);
xlabel("y (m)"), ylabel("z (m)");
saveas(1, "../fig/2_1_" + sampling_density + ".png");
```

C.2.2 Result and Analysis

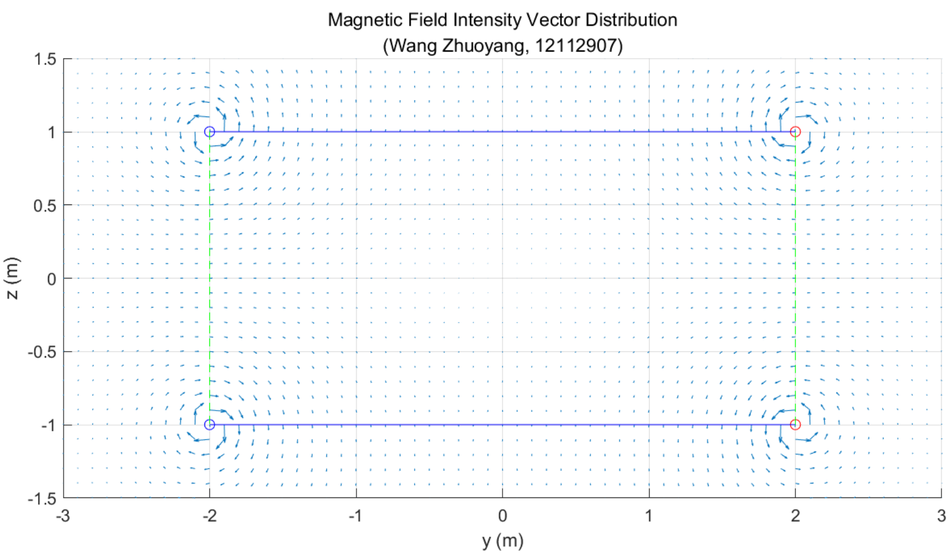

Figure 5: Magnetic field intensity vector distribution for Case 2 (yOz plane section)

In Figure 5 we can observe that the magnetic field distribution is changed compared with Case 1, and the internal distribution is no longer uniform.

C.3 Magnetic Field Intensity Magnitude Distribution

C.3.1 MATLAB Code

The process is exactly the same as Case 1 and will not be repeated.

```matlab
%% Magnetic Field Intensity Magnitude Distribution
H_norm = sqrt(Hy .^ 2 + Hz .^ 2);

% Plot the figure
figure(2);
grid on, axis equal, hold on;
[mesh_y, mesh_z] = meshgrid(range_y, range_z);
mesh(mesh_y, mesh_z, H_norm');
axis([-length_y / 2, length_y / 2, -length_z / 2, length_z / 2, 0, 500]);
set(gcf, 'Position', [50, 50, 400, 700]);
view([3, 3, 200]);
title(["Magnetic Field Intensity Magnitude Distribution", "(Wang Zhuoyang,
↪  12112907)"]);
xlabel("y (m)"), ylabel("z (m)"), zlabel("|H| (A/m)");
saveas(2, "../fig/2_2_" + sampling_density + ".png");
```

C.3.2 Result and Analysis

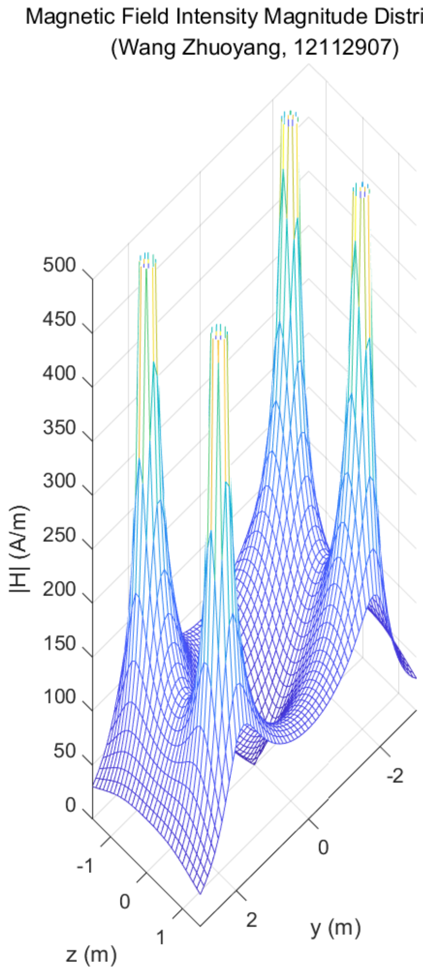

Figure 6: Magnetic field intensity magnitude distribution for Case 2 (yOz plane section)

The results in Figure 6 match the expectations.

C.4 Magnetic Line Distribution

C.4.1 MATLAB Code

The process is exactly the same as Case 1 and will not be repeated.

```
%% Magnetic Line Distribution
% Sample the magnitude of H
H_samples_range_y = [-a, a];
H_samples_z = d / 2;
H_samples_index_y = floor((H_samples_range_y(1) + length_y / 2) * sampling_density) : floor((H_samples_range_y(2) + length_y / 2) * sampling_density);
H_samples_index_z = floor((length_z / 2 + H_samples_z) * sampling_density);
H_samples = H_norm(H_samples_index_y, H_samples_index_z);

% Calculate PDF & CDF
H_pdf = H_samples ./ sum(H_samples);
H_cdf = zeros(1, length(H_pdf));
H_cdf(1) = H_pdf(1);
for it = 2 : length(H_pdf)
    H_cdf(it) = H_cdf(it - 1) + H_pdf(it);
end

% Sample basing on PDF & CDF
line_number = 28;
line_start_y = zeros(1, line_number);
line_start_z = zeros(1, line_number);
uniform_samples = linspace(0.012, 0.99, line_number);
for it_u = 1 : line_number
    u = uniform_samples(it_u);
    if u < H_cdf(1)
        left = 0;
        right = H_samples_range_y(1);
    else
        for it_s = 1 : (length(H_cdf) - 1)
            if u >= H_cdf(it_s) && u < H_cdf(it_s + 1)
                delta = (H_samples_range_y(2) - H_samples_range_y(1)) / length(H_cdf);
                left = H_samples_range_y(1) + delta * (it_s - 1);
```

```
                right = H_samples_range_y(1) + delta * it_s;
            end
        end
    end
    line_start_y(it_u) = (left + right) / 2;
    line_start_z(it_u) = H_samples_z;
end

% Plot the figure
figure(3);
grid on, axis equal, hold on;
[mesh_y, mesh_z] = meshgrid(range_y, range_z);
fig_sl = streamline(mesh_y, mesh_z, -Hy', -Hz', line_start_y, line_start_z);
set(fig_sl, "lineWidth", 0.4, "color", [0.9, 0.4, 0.8]);
fig_sl = streamline(mesh_y, mesh_z, Hy', Hz', line_start_y, line_start_z);
set(fig_sl, "lineWidth", 0.4, "color", [0.9, 0.4, 0.8]);
fig_sl = streamline(mesh_y, mesh_z, -Hy', -Hz', line_start_y, -line_start_z);
set(fig_sl, "lineWidth", 0.4, "color", [0.9, 0.4, 0.8]);
fig_sl = streamline(mesh_y, mesh_z, Hy', Hz', line_start_y, -line_start_z);
set(fig_sl, "lineWidth", 0.4, "color", [0.9, 0.4, 0.8]);
plot(a, d / 2, 'ro', -a, d / 2, 'bo', a, -d / 2, 'ro', -a, -d / 2, 'bo');
plot([a, -a], [d / 2, d / 2], 'b-', [-a, -a], [d / 2, -d / 2], 'g--', [-a, a],
↪ [-d / 2, -d / 2], 'b-', [a, a], [-d / 2, d / 2], 'g--')
axis([-length_y / 2, length_y / 2, -length_z / 2, length_z / 2]);
set(gcf, 'Position', [50, 50, 900, 600]);
title(["Magnetic Line Distribution", "(Wang Zhuoyang, 12112907)"]);
xlabel("y (m)"), ylabel("z (m)");
saveas(3, "../fig/2_3_" + sampling_density + ".png");
```

C.4.2 Result and Analysis

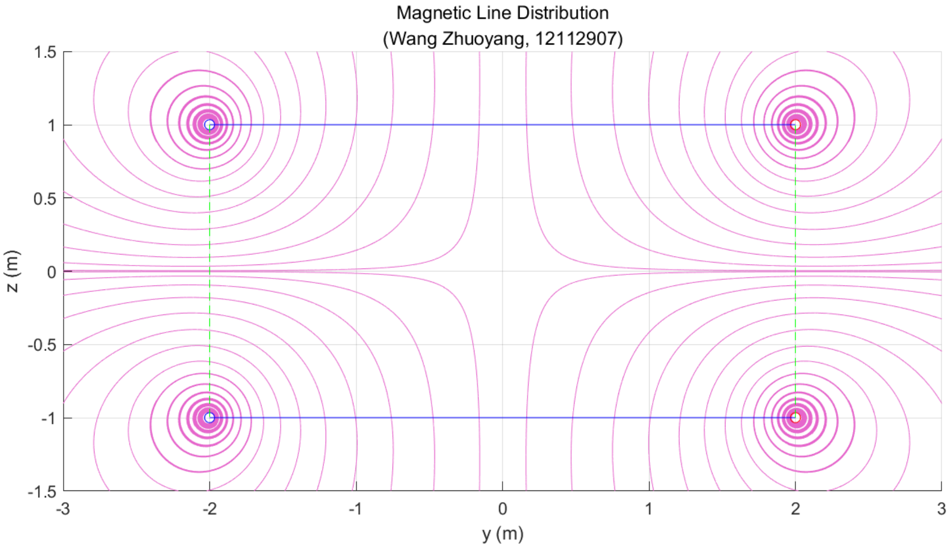

Figure 7: Magnetic line distribution for Case 2 (yOz plane section)

The results in Figure 7 match the expectations.

D Conclusion

From this experiment, we've learnt the way to plot the properties of the magnetic fields generated by current loop pairs.

And we deepened our understanding on the application of the Biot-Savart Law and other knowledge we learnt in class.

We also applied some knowledge from other disciplines to solve the problem and achieved better results, which was very rewarding.