

## משימת סיום מבוא לאופטיקה מודרנית ואלקטרואופטיקה, חלק 2

מגיש: דניאל ברוקר-315015594

### שאלה 3:

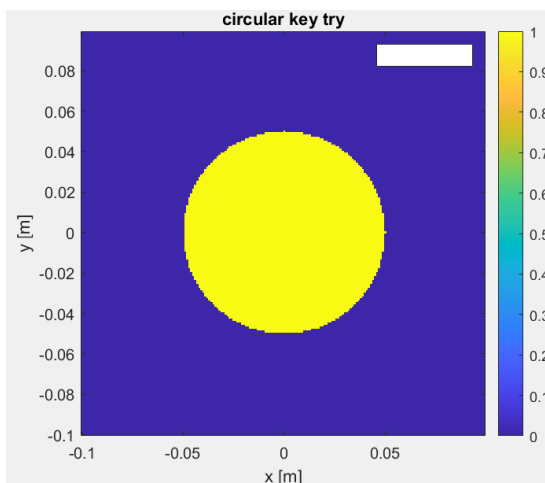
(1)

א.

הפונקציה  $\text{circ}$  תקבל את 3 הערכים שנתונים בסעיף ב'  $L$  (טווח ראייה),  $R$  (רדיוס של המעגל)  $N$ , (מספר הדגימות בכל מימד).

$\text{circ}$  בעצם יוצרת ומחזירה מטריצה ובודקת מרחק מהמרכז לכל נקודה במטריצה, אם המרחק הזה  $r$  קטן מהרדיוס  $R$  שהיא מקבלת אז הערך במטריצה במיקום הזה מקבל 1 ואם הנקודה נמצאת מחוץ למעגל  $(R < r)$  אז הערך יהיה 0.

ב.ג.

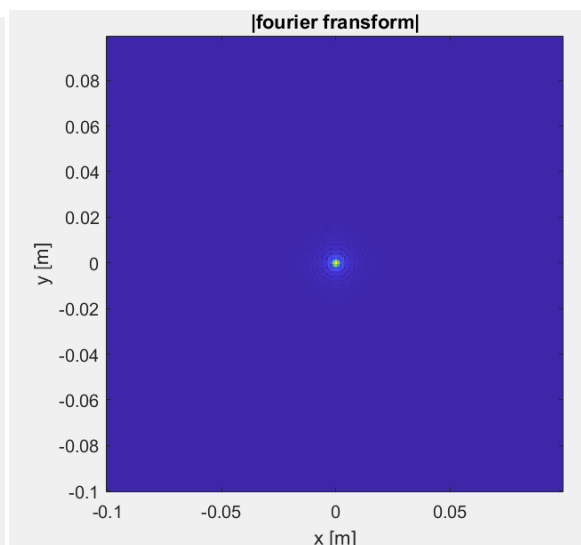
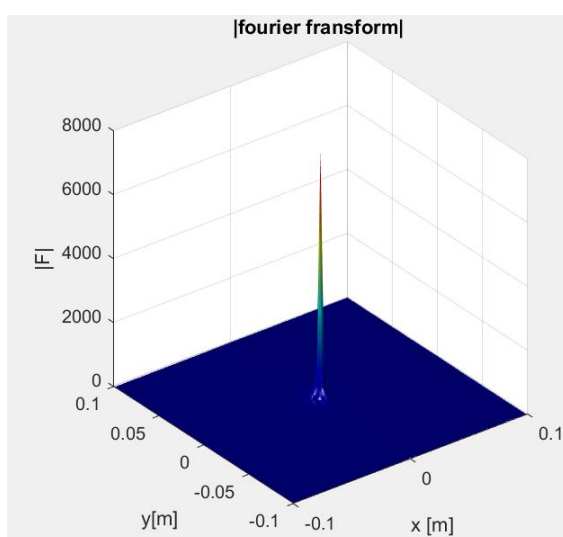


$$R = (\text{mod}(594,5) + 1) \cdot 10 - 2 = 0.05[m]$$

$$L = 0.2[m]$$

$$N = 200$$

ד.



(2)

$$\lambda = \text{round}\left(400 + \frac{594}{999} \cdot 900\right) [\text{nm}] = 935 [\text{nm}]$$

.א.

התנאי למרחק התצפית לפי תרגול 10 יהיה  $z \gg \frac{\pi \cdot d^2}{4 \cdot \lambda}$

$$z \gg \frac{\pi \cdot (2 \cdot R)^2}{4 \cdot \lambda} = \frac{\pi \cdot (R)^2}{\lambda} = \frac{\pi \cdot (0.05)^2}{935 \cdot 10^{-9}} = 8399.98$$

ניקח  $z_0$  גדולה פי 5  $Z_0 = 41999.9$

.ב.

$$E(x, y, z) = \frac{e^{ikz} e^{\frac{ik}{2z} \cdot (x^2 + y^2)}}{\lambda z i} * \mathcal{F}\{E(x', y', 0)\}$$

$$= \frac{(e^{ikz} e^{\frac{ik}{2z} \cdot (x^2 + y^2)})}{\lambda z i} * R^2 * \text{jinc}(R * \text{sqrt}(f_x^2 + f_y^2))$$

נציב:  $f_y = \frac{y}{\lambda z}, f_x = \frac{x}{\lambda z}$

$$= \frac{(e^{ikz} e^{\frac{ik}{2z} \cdot (x^2 + y^2)})}{\lambda z i} * R^2 * \text{jinc}\left(R * \text{sqrt}\left(\frac{x^2 + y^2}{\lambda^2 * z^2}\right)\right)$$

נזכור כי העוצמה  $I \sim |E|^2$

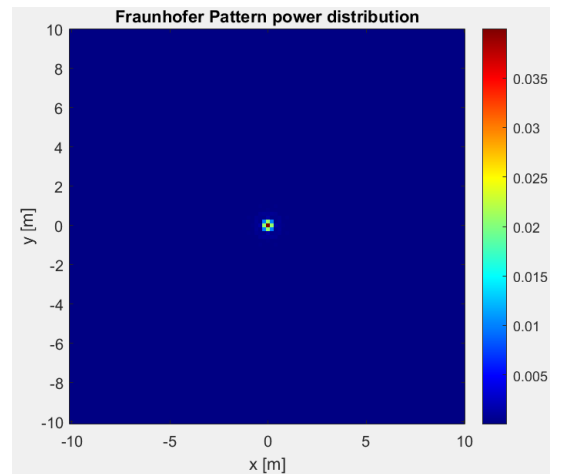
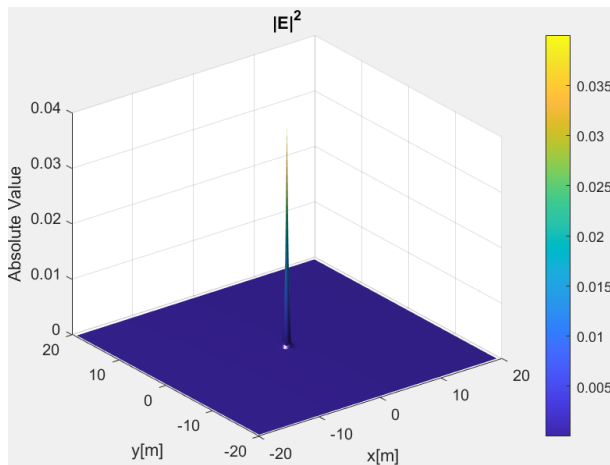
$$|E(x, y, z_0)|^2 = \left| \frac{(e^{ikz_0} e^{\frac{ik}{2z_0} \cdot (x^2 + y^2)})}{\lambda z_0 * i} \right|^2 * R^4 * \left| \text{jinc}\left(R * \text{sqrt}\left(\frac{x^2 + y^2}{\lambda^2 * z_0^2}\right)\right) \right|^2$$

$$|E(x, y, z_0)|^2 = \frac{e^{2ik\left(\frac{x^2 + y^2}{2z_0} + z_0\right)}}{\lambda^2 z_0^2} * R^4 * \left| \text{jinc}\left(R * \text{sqrt}\left(\frac{x^2 + y^2}{\lambda^2 * z_0^2}\right)\right) \right|^2$$

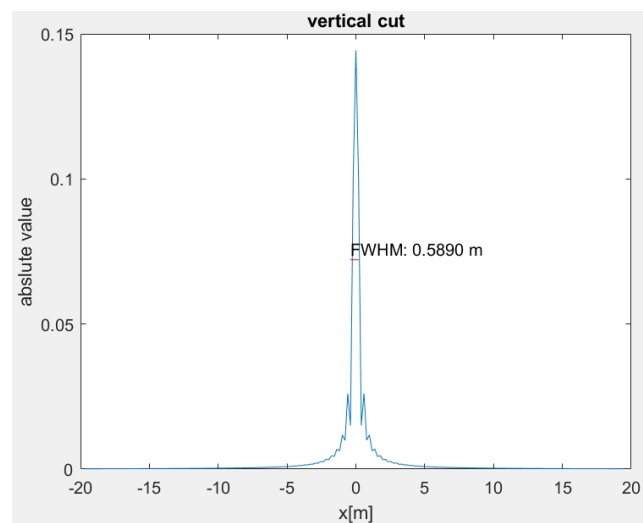
$$k = \frac{2 * \pi i}{\lambda}$$

$$I \sim |E(x, y, z_0)|^2 = \frac{1}{\lambda^2 z_0^2} * R^4 * \left| \text{jinc}\left(R * \text{sqrt}\left(\frac{x^2 + y^2}{\lambda^2 * z_0^2}\right)\right) \right|^2$$

ג.



ד.



FWHM משמעותו "Full Width at Half Maximum". כלומר מודדים את רוחב האות בחצי גובה המקסימום שלו בציר Z

$Z_0$  - ככול ש  $Z_0$  יגדל כך גם FWHM יגדל - אני משער שממרחק גדול יותר הפיזור של האור יגדל וזה יוריד את הרזולוציה וזה ישפיע על פיזור העוצמה של האור.

R - ככול ש R יגדל כך גם FWHM יקטן- דומה והפוך ל L מפתח גדול יותר מאפשר לקליטה של יותר אור שאמורה לשפר את הרזולוציה ולהוביל לפיזור צר יותר של עוצמת האור.

ג - ככול ש  $\lambda$  יגדל כך גם FWHM יגדל - אני מעריך שזה נובע מזה שאורכי גל ארוכים יותר נוטים להתעקם ביחס לאורכי גל קצרים יותר. העקיפה המוגברת יכולה להוביל לפיזור רחב יותר של עוצמת האור, מה שמוריד את הרזולוציה ומוביל ל-FWHM גדול יותר.

-כל ההשערות מגובות עם ניסיונות של שינוי הערכים בקוד וקבלת התוצאות האלו.

(3)

א.

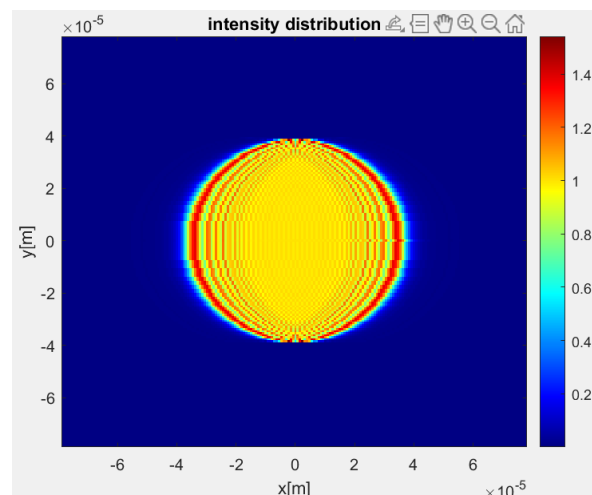
fftshift - ממרכזת את התמרת הפורייה סביב התדירות 0, תעזור לנו בסידור התצוגה אחרי שימוש בfft במקרה של מטריצות כמו אצלנו היא תחליף בין הרביעים.

ifftshift - מבצעת פעולה הפוכה ל fftshift ומחזירה את רכיב התדר האפס למיקומו המקורי.

ב.

מצורף בנספח קוד

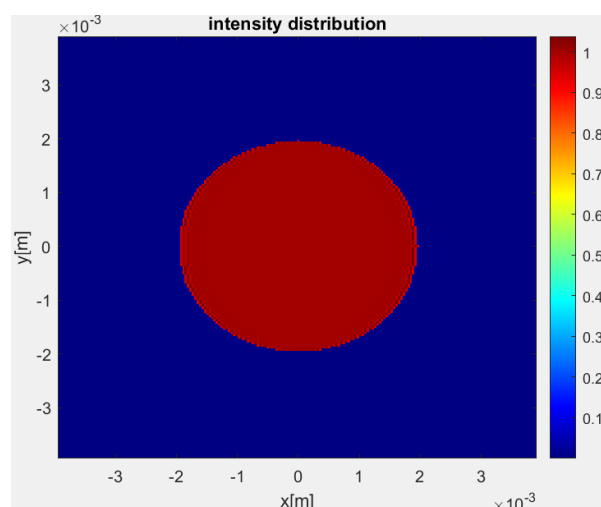
ג.



FWHM x-axis:  $7.2257 \times 10^{-5}$  [m]

FWHM y-axis:  $7.6969 \times 10^{-5}$  [m]

ד.

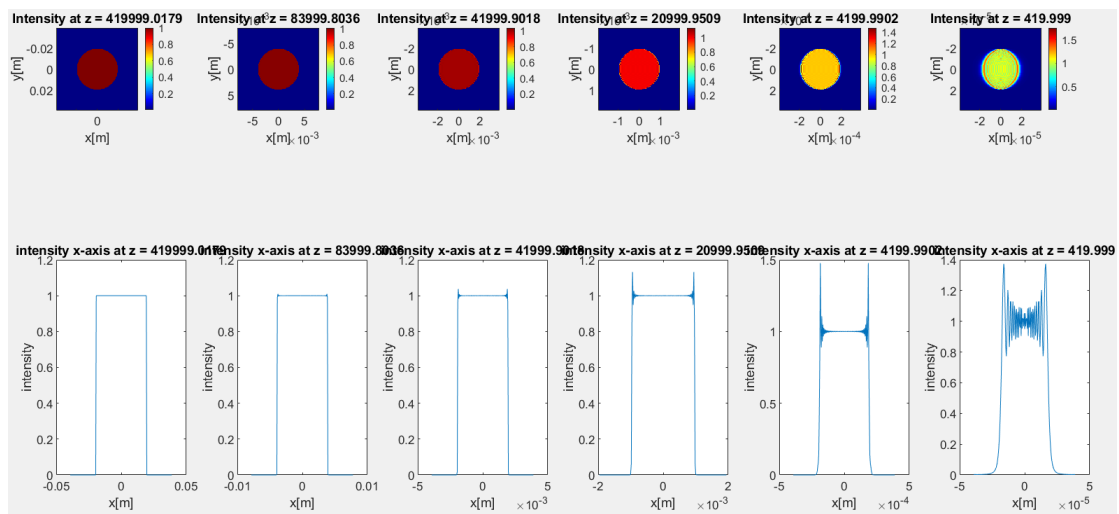
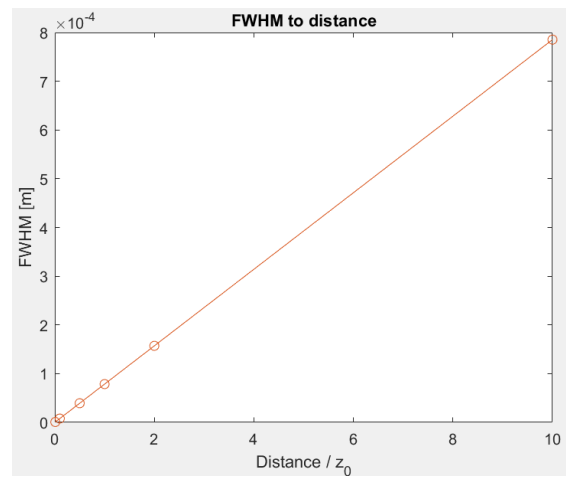


FWHM x-axis: 0.0038485 [m]

FWHM y-axis: 0.0038485 [m]

התקבל FWHM קטן בהרבה מאשר בסעיף ד בשאלה 2 וזה אומר שיש שיפור משמעותי. כלומר אפשר להגיד שזו שיטה עדיפה וקירוב מדוייק יותר.

ה.



## נספחים-קוד:

### פונקציות נוספות:

#### circ:

```
function circ_output = circ(L, N, R)
    delta_x=L/N;
    x = linspace(-L/2, L/2 - delta_x, N);
    [X, Y] = meshgrid(x, x);
    r = sqrt(X.^2 + Y.^2);
    circ_output = r < R;
end
```

#### propFresnel:

```
function [u2, x_prop] = propFresnel(u1, L, lambda, z)
    % propagation - according to Fresnel
    % assumes same x and y side lengths and
    % uniform sampling
    % u1 - source plane field
    % L - source plane side length (FOV)
    % lambda - wavelength
    % z - propagation distance
    % u2 - observation plane field
    % x_prop - axis in observation plane field

    i=sqrt(-1);
    N = size(u1, 1); %square grid
    delta_x = L / N;
    x = (-N/2 : N/2-1) * delta_x; % Source plane coordinates
    f_x = x / (lambda * z);

    U1 = F(u1);
    kernel = exp(i * pi * lambda * z * (f_x.^2));
    U2 = U1 .* kernel;
    u2 = iF(U2);

    x_prop = x * lambda * z;
end
```

## שאלה 3:

```
%Q3
%%
%3-1
disp('3-1:')
disp('3-1B+C:');
R = 0.05; %radius of the circle [m]
L = 0.2; %viewing range [m]
N = 200; %number of samples in each dimension
delta_x = L / N;

%x,y axis
x = linspace(-L/2, L/2 - delta_x, N);
y = linspace(-L/2, L/2 - delta_x, N);

%circular key sample
circ_output = circ(L, N, R);

%plot
figure;
imagesc(x, y , circ_output);
axis xy;
axis image;
xlabel('x [m]');
ylabel('y [m]');
title('circular key try');

%%
%3-1D
disp('3-1D:');

%frequency axes
f_x = linspace(-1/(2*delta_x), 1/(2*delta_x) - 1/L, N); %[cycles/m]
f_y = linspace(-1/(2*delta_x), 1/(2*delta_x) - 1/L, N); %[cycles/m]

%Fourier transform
F_result = fftshift(fft2(circ_output));

axes=-L/2:L/N:L/2-L/N;
%plot
figure;
imagesc(axes,axes,abs(F_result));
axis xy;
axis image;
xlabel('x [m]');
ylabel('y [m]');
title('|fourier fransform|');
axis xy;
axis image;

%plot
[a, b] = meshgrid(-L/2:L/N:L/2-L/N, -L/2:L/N:L/2-L/N);
figure;
```

```

surf(a,b, abs(F_result));
axis square;
colormap(jet);
xlabel('x [m]');
ylabel('y[m]');
zlabel('|F|');
title('|fourier fransform| ');
camlight left;
lighting phong;
shading interp;

%%
%3-2
disp('3-2:');
%3-2C
disp('3-2C:');

lambda = 935.*10.^(-9); %[m]
z0 = 41999.90179; %[m]
k = 2*pi / lambda; %[rad/m]
i = sqrt(-1);
x = f_x * lambda * z0;
y = f_y * lambda * z0;

[X, Y] = meshgrid(x, y);
jinc_output = jinc(R * sqrt((X.^2 + Y.^2) / (lambda^2 * z0^2)));

%I~|E|^2
e1=exp(i * k * z0) ./ (lambda * z0 * i);
e2=exp(i *k * (X.^2 + Y.^2) / (2 * z0));
E = e1*e2* R^2 .* jinc_output;

%plot intensity distribution
figure;
imagesc(x, y, abs(E).^2);%I~|E|^2
axis xy;
axis image;
colormap('jet');
colorbar;
xlabel('x[m]');
ylabel('y[m]');
title('Fraunhofer Pattern power distribution');

figure;
surf(x, y, abs(E).^2);
camlight left;
lighting phong;
shading interp
colorbar;
title('|E|^2');
xlabel('x[m]');
ylabel('y[m]');
zlabel('absolute value');

```



```

%%
%3-2D
disp('3-2D:');
mid_index = round(N / 2); %middle index
E_side_look =abs(E(:, mid_index));

%FWHM
half_peak = max(E_side_look) / 2; %value of half of the peak
above_half_peak = E_side_look > half_peak;%bigger valu from the half
FWHM_range = find(diff(above_half_peak) ~= 0);
FWHM = x(FWHM_range([1 end])); %FWHM size
fprintf('FWHM: %.4f to %.4f meters\n', FWHM(1), FWHM(2));

%plot
figure;
plot(x, E_side_look);
title('vertical cut');
xlabel('x[m]');
ylabel('abslute value');
hold on

plot(FWHM, [half_peak, half_peak], 'r');
text(FWHM(1), half_peak, sprintf('FWHM: %.4f m', FWHM(2) - FWHM(1)),
'VerticalAlignment', 'bottom');
hold off;

%%
%3-3
disp('3-3:');
%3-3B
disp('3-3B:');

% function [u2, x_prop] = propFresnel(u1, L, lambda, z)
%     % propagation - according to Fresnel
%     % assumes same x and y side lengths and
%     % uniform sampling
%     % u1 - source plane field
%     % L - source plane side length (FOV)
%     % lambda - wavelength
%     % z - propagation distance
%     % u2 - observation plane field
%     % x_prop - axis in observation plane field
%
%     % Define the grid and sampling parameters
%     N = size(u1, 1); % Assume square grid
%     delta_x = L / N; % Sampling interval
%     x = (-N/2 : N/2-1) * delta_x; % Source plane coordinates
%     f_x = x / (lambda * z); % Frequency axis in the source plane
%
%     % Fourier transform of the source plane field
%     U1 = F(u1);
%
%     % Fresnel kernel
%     kernel = exp(1i * pi * lambda * z * (f_x.^2));
%
%     % Propagation according to Fresnel
%     U2 = U1 .* kernel;
%
%     % Inverse Fourier transform to get the observation plane field

```

```

%      u2 = iF(U2);
%
%      % Define axis in the observation plane field
%      x_prop = x * (lambda * z);
%
% end

%%
%3-3C
disp('3-3C:');

z1 = z0 ; %[m]
%circular key sample
u1 = circ(L, N, R);

%Fresnel pattern at z1
[u2, x_prop] = propFresnel(u1, L, lambda, z1);
%intensity distribution
intensity =abs(u2).^2;

figure;
imagesc(x_prop, x_prop, intensity);
axis xy;
colormap('jet');
colorbar;
xlabel('x[m]');
ylabel('y[m]');
title('intensity distribution');

figure;
surf(x_prop, x_prop, intensity);
camlight left;
lighting phong;
shading interp
colorbar;
title('|intensity|');
xlabel('x[m]');
ylabel('y[m]');
zlabel('absolute value');
camlight left;
lighting phong;
shading interp;

%FWHM of the intensity pattern
%x-direction
half_peak_X = max(intensity(:)) / 2;
X_range = find(intensity(size(intensity, 1) / 2, :) >= half_peak_X); %find
indices
FWHM_X = x_prop(X_range(end)) - x_prop(X_range(1));

%y-direction
half_peak_Y = max(intensity(:)) / 2;
X_range = find(intensity(:, size(intensity, 2) / 2) >= half_peak_Y); %find
indices
FWHM_Y = x_prop(X_range(end)) - x_prop(X_range(1));

disp(['FWHM x-axis: ', num2str(FWHM_X), ' [m]']);
disp(['FWHM y-axis: ', num2str(FWHM_Y), ' [m]']);

```

```

%%
%3-3D
disp('3-3D:');
%same as 3-3C just change z1 from 'z0/50' to 'z0'
%%
%3-3E
disp('3-3E:');

distances = [10*z0, 2*z0, z0, 0.5*z0, 0.1*z0, 0.01*z0];
FWHM_X_val = zeros(size(distances));
FWHM_Y_val = zeros(size(distances));

figure;
for i = 1:length(distances)

    [u2, x_prop] = propFresnel(u1, L, lambda, distances(i));

    intensity = abs(u2).^2;

    %FWHM
    max_intensity = max(intensity(:));
    half_peak_intensity = max_intensity / 2;

    %indices where intensity is closest to half max
    [~, idx] = min(abs(intensity(:) - half_peak_intensity));
    [y_indx, x_indx] = ind2sub(size(intensity), idx);

    %FWHM in y-axis
    top_indx = max(1, y_indx - 1);
    down_indx = min(N, y_indx + 1);
    FWHM_Y = abs(x_prop(down_indx) - x_prop(top_indx));
    %FWHM in x-axis
    left_indx = max(1, x_indx - 1);
    right_indx = min(N, x_indx + 1);
    FWHM_X = abs(x_prop(right_indx) - x_prop(left_indx));

    %save FWHM values
    FWHM_X_val(i) = FWHM_X;
    FWHM_Y_val(i) = FWHM_Y;

    subplot(2, 6, i);
    imagesc(x_prop, x_prop, intensity);
    axis square;
    colormap('jet');
    colorbar;
    xlabel('x[m]');
    ylabel('y[m]');
    title(['Intensity at z = ', num2str(distances(i))]);

    subplot(2, 6, 6+i);
    plot(x_prop, intensity(round(N/2), :));
    xlabel('x[m]');
    ylabel('intensity');
    title(['intensity x-axis at z = ', num2str(distances(i))]);
end

%plot FWHM to distance
figure;

```

```
normalized_distances = distances / z0;
plot(normalized_distances, FWHM_X_val, '-o', 'DisplayName', 'FWHM x');
hold on;
plot(normalized_distances, FWHM_Y_val, '-o', 'DisplayName', 'FWHM y');
xlabel('Distance / z_0[m]');
ylabel('FWHM [m]');
title('FWHM to distance');
legend;
```