
Table of Contents

Ntambaazi Tonny, CIMET: Report Lab 2	1
The point propagation function ppf	1
Reference image	1
Object image	7
Phase difference between Object and Reference	12
Obtaining the value of h from the phase difference formula	14

Ntambaazi Tonny, CIMET: Report Lab 2

```
% This report and code shows the application of the point propagation
% function and
% also show the distribution of h
% As in first report, it uses the filter function for masking the
% images
% It also uses the unwrap_Gold2_1_modified.m file for unwrapping the
% phase

clear all; close all; clc;

Reference = imread('Reference.tif');
Object    = imread('Object.tif');
radius = 20;
dim = 2;
DeltaX = 4.4 * (10^-6);
lamda = 632 * (10^-9);
dimChange = dim * (10^-3);
No = 1.59;
Nb = 1.52;
```

The point propagation function ppf

```
frequency = (1/(2*DeltaX));

for i = 1:100
    m = (i*(6.28/lamda)*sqrt(1-(frequency^2))*dimChange);
    ppf =+ m;
end

% pp = sum((1:100)*((6.28/lamda)*sqrt(1-(frequency^2))*dimChange));
% pp1 = (100)*((6.28/lamda)*sqrt(1-(frequency^2))*dimChange);
```

Reference image

```
Ref_FT = fft2(Reference);
```

```

Ref_FT = fftshift(Ref_FT);
imagesc(log(abs(Ref_FT)));

% Applying mask to the image using the filter function
Filt_Ref = Filter(Ref_FT, radius );
figure; imagesc(log(abs(Filt_Ref))); title('Reference filtered
image');

% shifting the frequency to the center of spectrum
shift_Ref = fftshift(Filt_Ref, dim);

% Applying the propagation function
shift_Ref = shift_Ref* ppf;

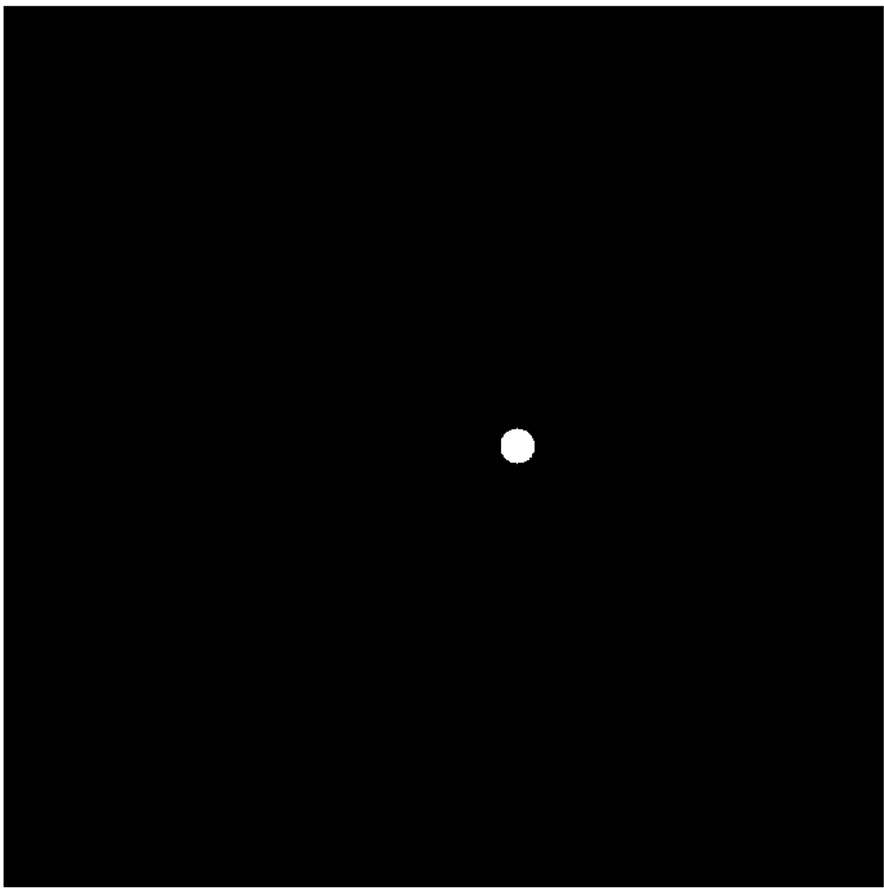
figure; imagesc(log(abs(shift_Ref))); title('Result of the propagation
function');

% Applying the inverse Fourir transform
Ref_mask = ifft2(shift_Ref);
figure; imshow(Ref_mask); title ('Reference IFT');

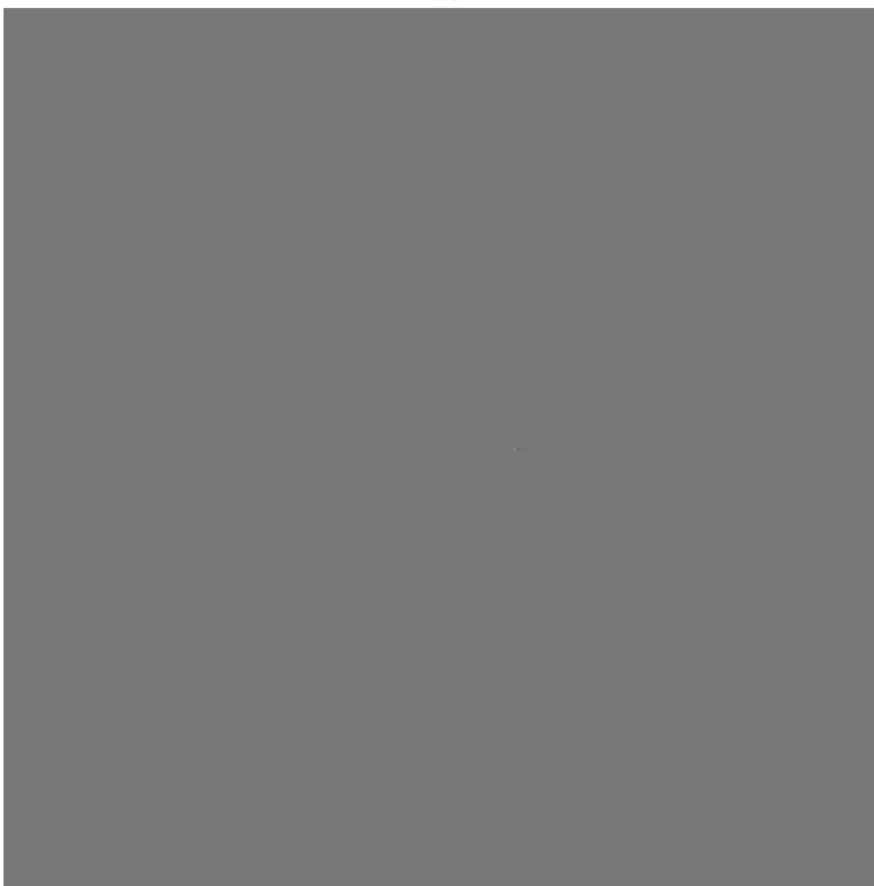
% The reference Phase
Ref_phase = angle(Ref_mask);
figure; imshow(Ref_phase); title('Reference phase');

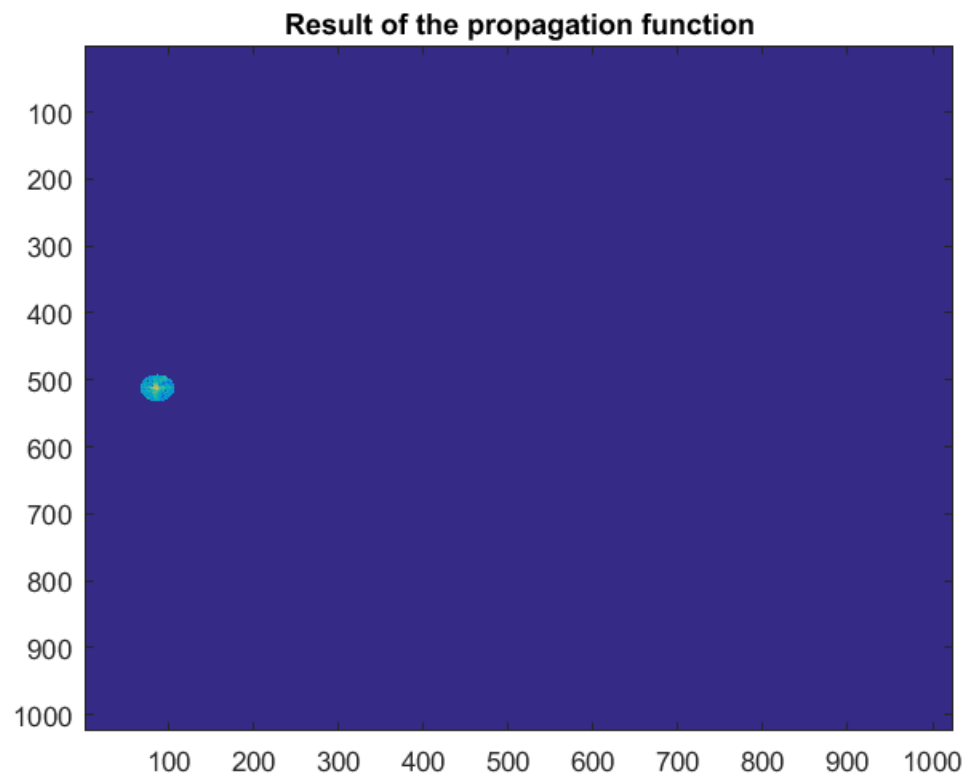
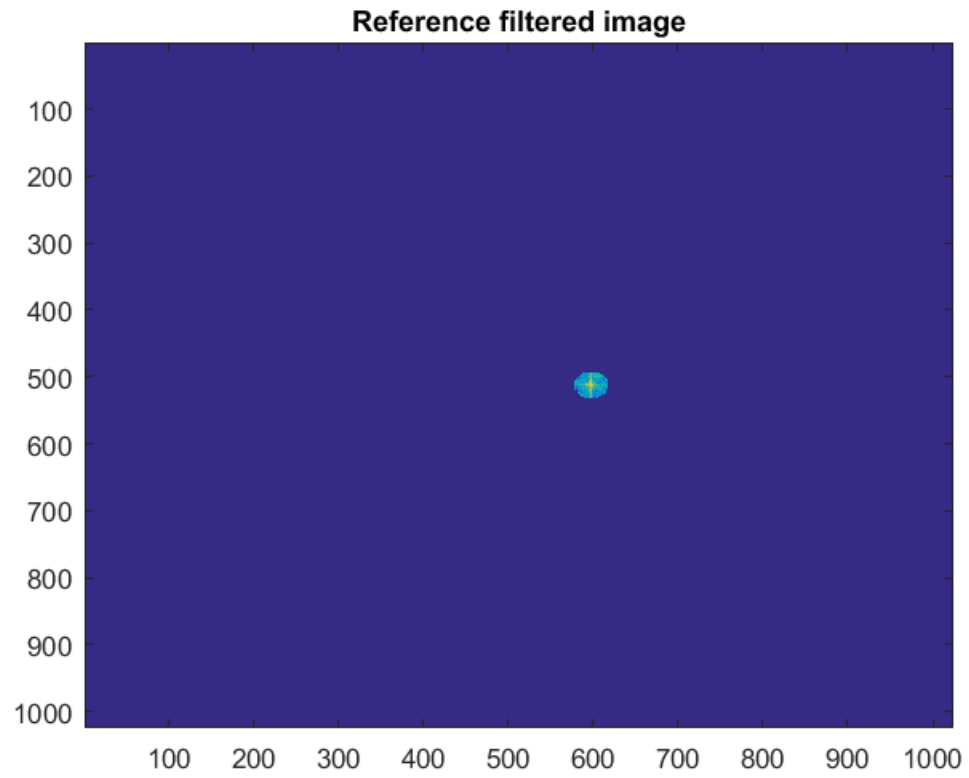
Warning: Image is too big to fit on screen; displaying at 67%
Warning: Displaying real part of complex input.
Warning: Image is too big to fit on screen; displaying at 67%
Warning: Displaying real part of complex input.
Warning: Image is too big to fit on screen; displaying at 67%
Warning: Image is too big to fit on screen; displaying at 67%

```

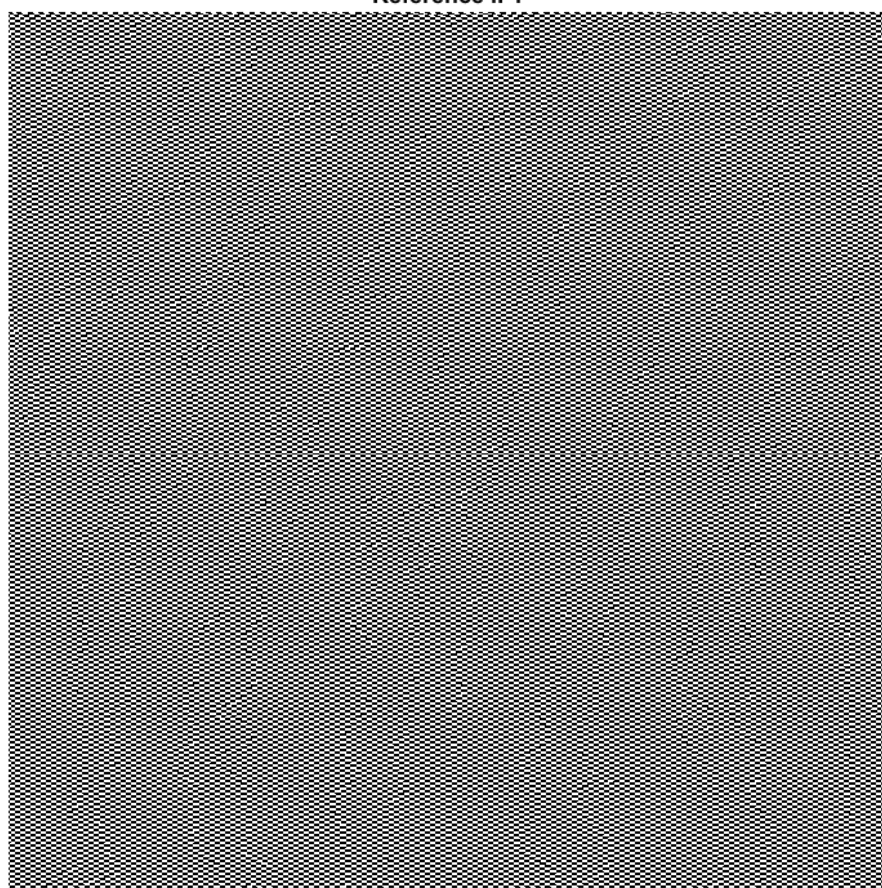


mask

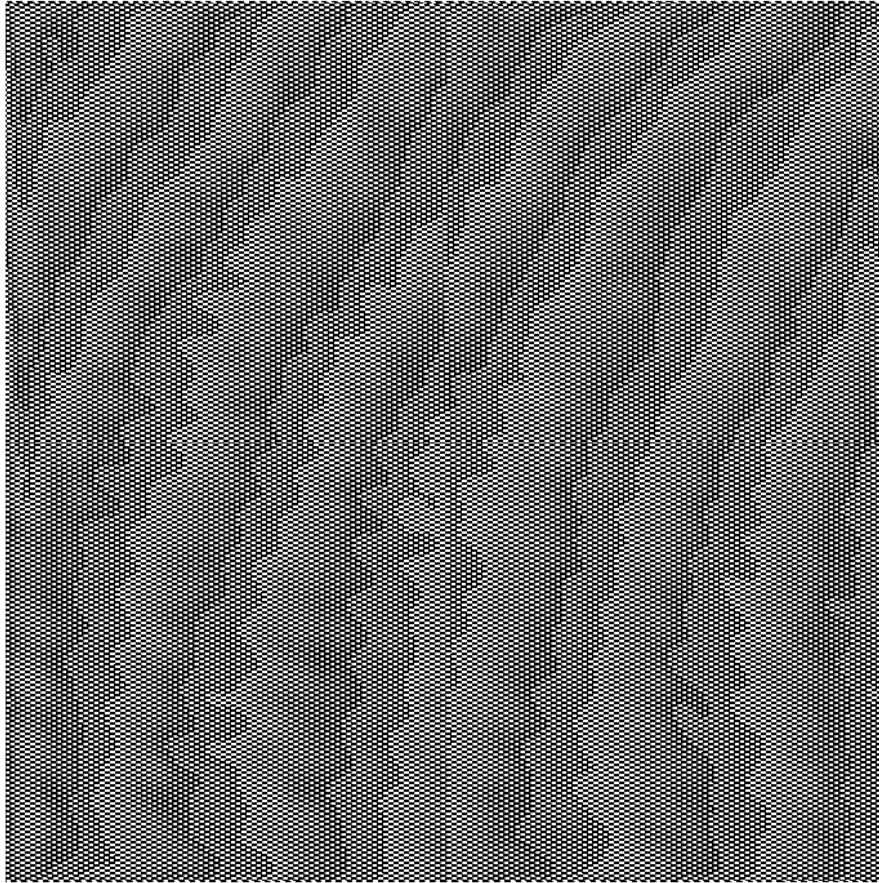




Reference IFT



Reference phase



Object image

```
Obj_FT = fft2(Object);
Obj_FT = fftshift(Obj_FT, 10);
imagesc(log(abs(Obj_FT)));

% Applying mask to the image using the filter function
Filt_Object = Filter(Obj_FT, radius );
figure; imagesc(log(abs(Filt_Object))); title ('Object filtered
Image');

% Shifting the frequency to center of spectrum

shift_Obj = fftshift(Filt_Object, dim);

% Applying the propagation function
shift_Obj = shift_Obj* ppf;

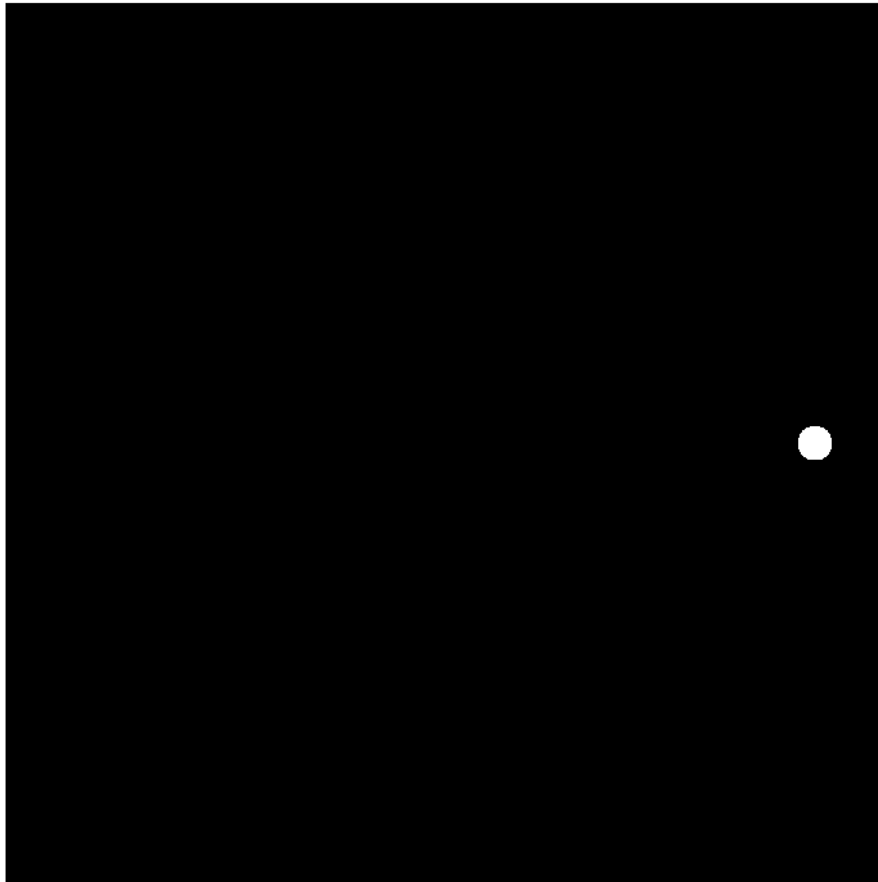
figure; imagesc(log(abs(shift_Obj))); title('Result of the propagation
function');
```

```
% Applying the inverse Fourir transform

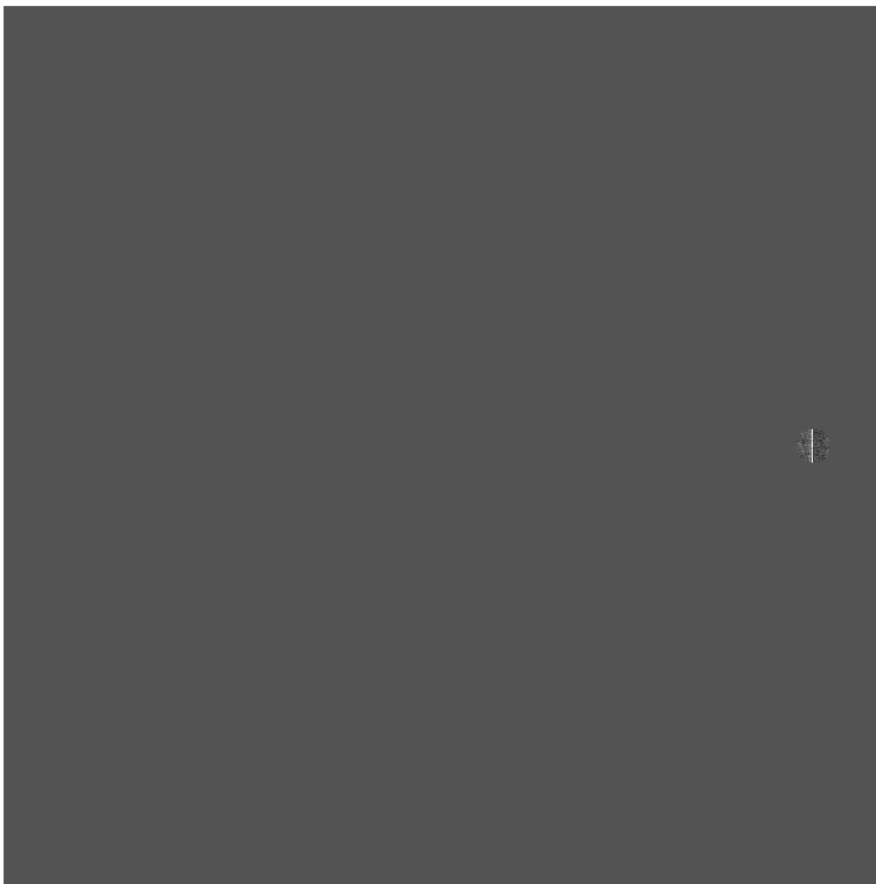
Obj_mask = ifft2(shift_Obj);
figure; imshow(Obj_mask); title ('Object IFT');

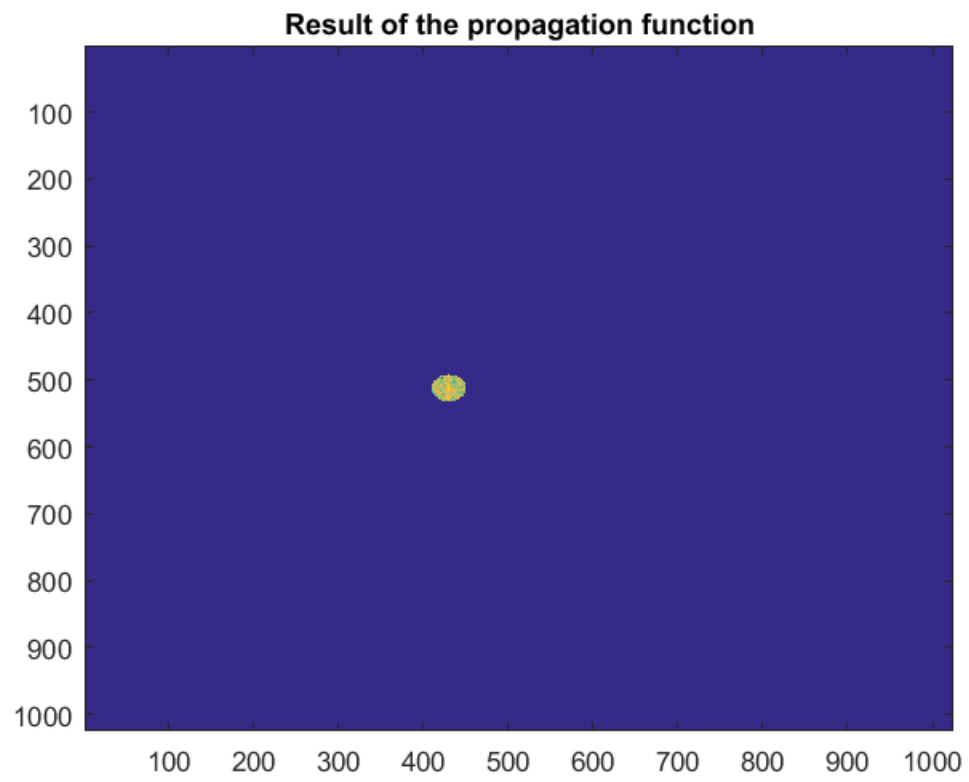
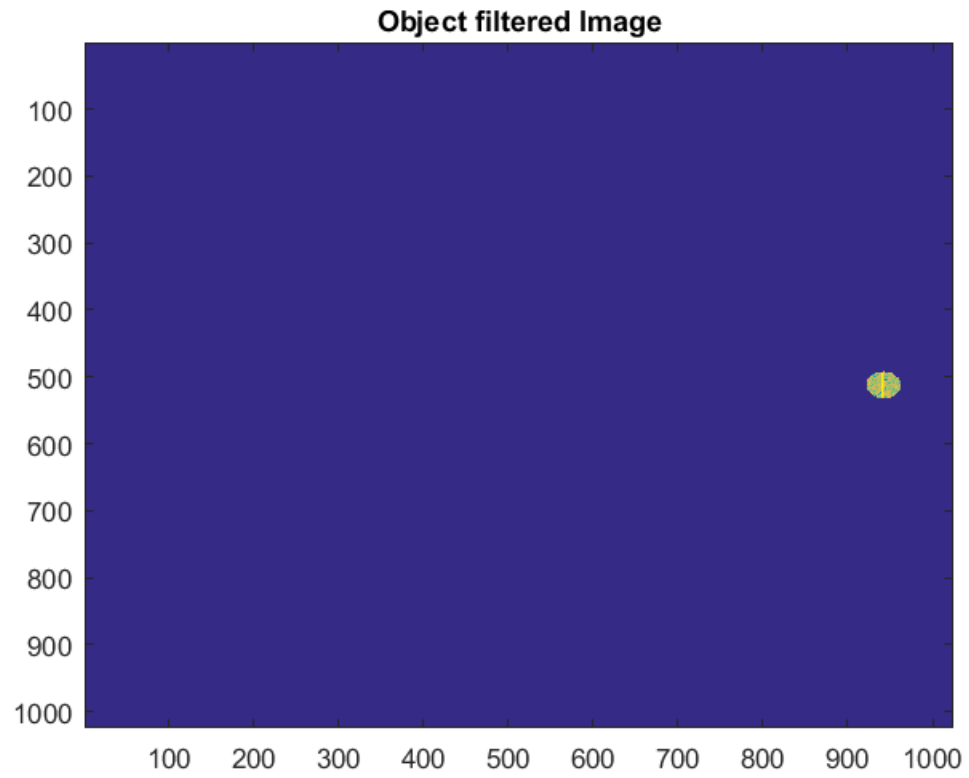
% Getting the phase angle
Obj_phase = angle (Obj_mask);
figure; imshow(Obj_phase); title('object phase');

Warning: Image is too big to fit on screen; displaying at 67%
Warning: Displaying real part of complex input.
Warning: Image is too big to fit on screen; displaying at 67%
Warning: Displaying real part of complex input.
Warning: Image is too big to fit on screen; displaying at 67%
Warning: Image is too big to fit on screen; displaying at 67%
```

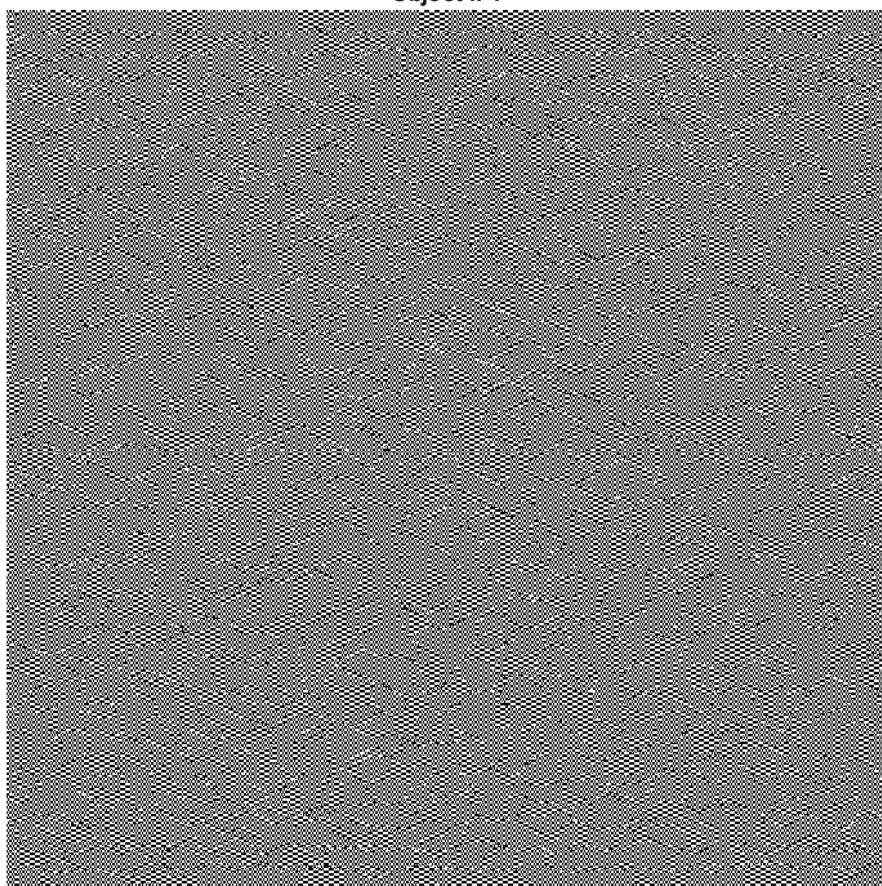


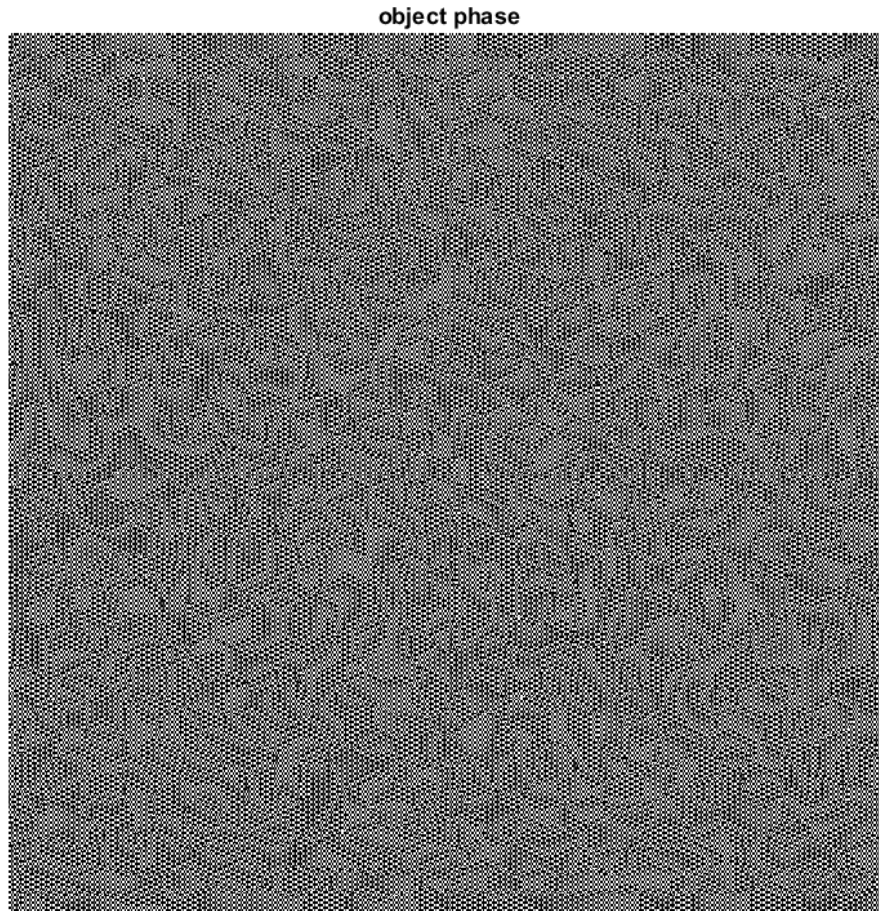
mask





Object IFT





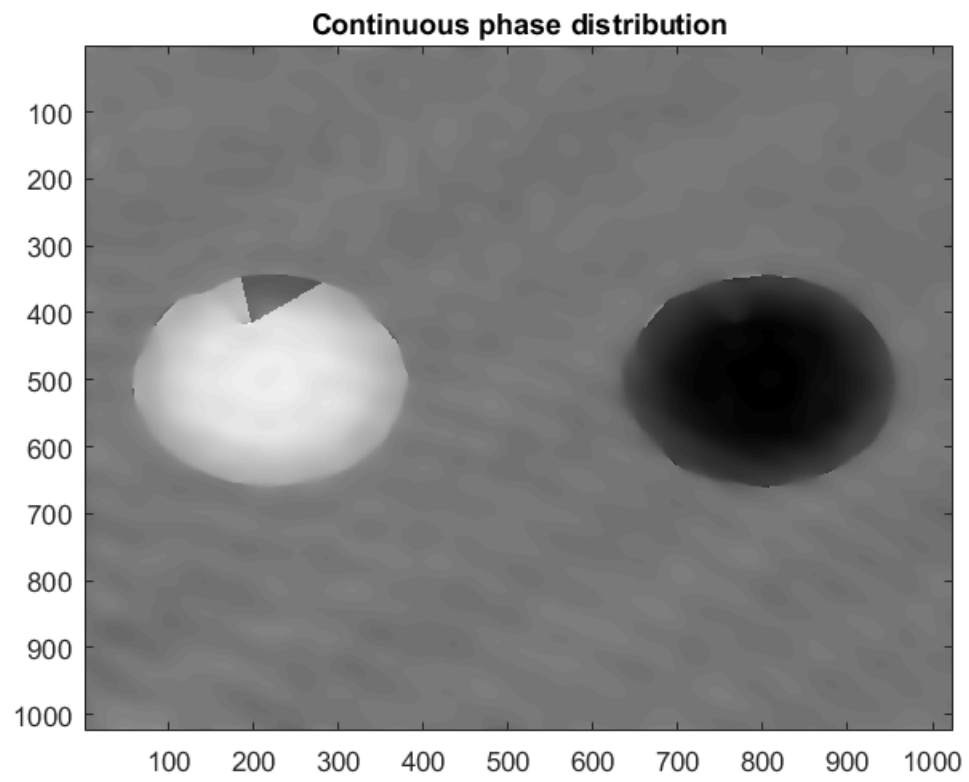
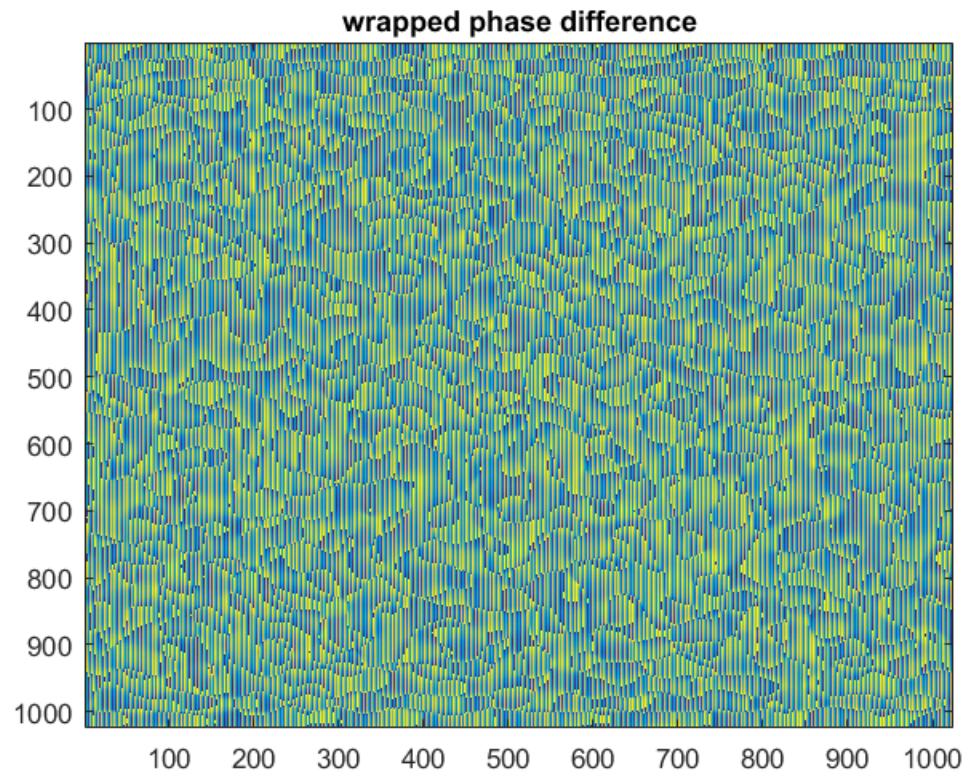
Phase difference between Object and Reference

```
phasediff = Obj_phase - Ref_phase;
phase_value = phasediff < 0;
phase = phasediff + (phase_value .* (2*3.14));

figure; imagesc(phase); title('wrapped phase difference');

% Unwrapping the phase and displaying the final result
unwrap_Gold2_1_modified
figure;
imagesc(unphase)
title('Continuous phase distribution');
colormap(gray)

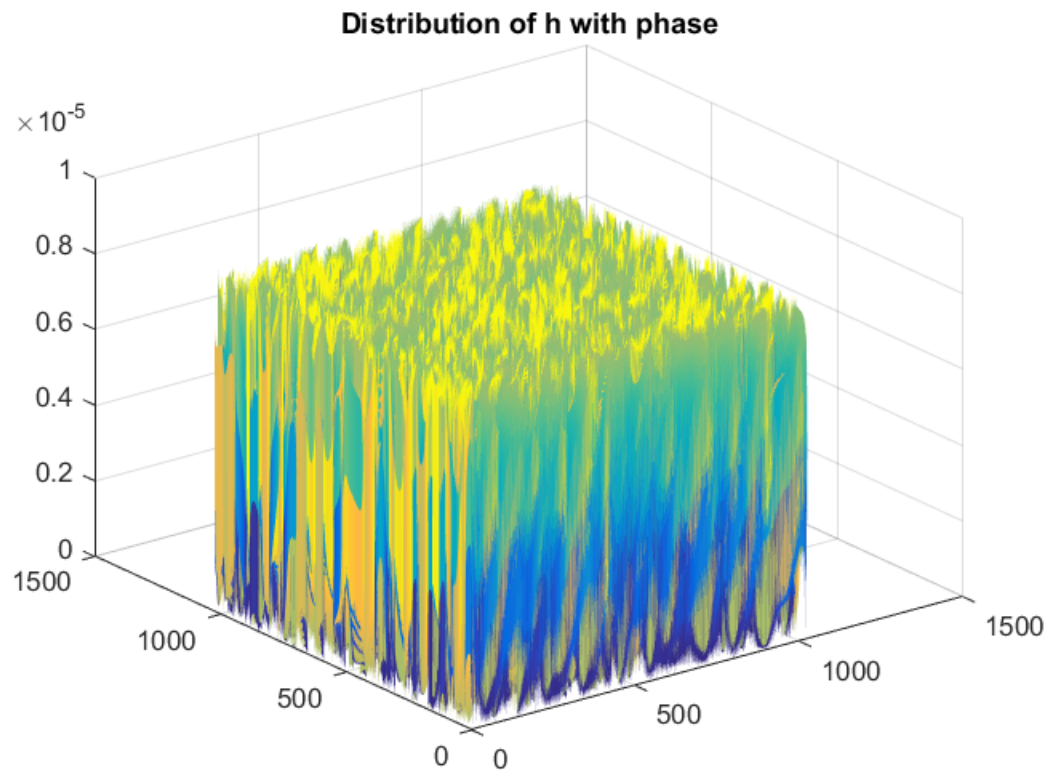
'unphs6' is not recognized as an internal or external command,
operable program or batch file.
```



Obtaining the value of h from the phase difference formula

```
% phase difference = (2*Pie/lamda)*(No-Nb)*h
%
h = phase/ ((2*3.14/lamda)*(No - Nb));

% plotting the value of h
figure; surf(h, 'EdgeColor','none'); title('Distribution of h with
phase');
```



Published with MATLAB® R2016a