

CO32: Fourier Optics

F. I. Giasemis
Keble College, University of Oxford

October 2016

1 Abstract

The aim of this practical is to investigate the *Discrete Fourier Transform* (DFT). We applied the formulae related to the Discrete Fourier Transform and in this way we calculated various diffraction patterns using MatLab, comparing each result with the previous.

2 Introduction

The discrete Fourier transform (DFT) converts a finite sequence of equally-spaced samples of a function into an equivalent-length sequence of equally-spaced samples of the discrete-time Fourier transform (DTFT), which is a complex-valued function of frequency. The interval at which the DTFT is sampled is the reciprocal of the duration of the input sequence. An inverse DFT is a Fourier series, using the DTFT samples as coefficients of complex sinusoids at the corresponding DTFT frequencies. It has the same sample-values as the original input sequence. The DFT is the most important discrete transform, used to perform Fourier analysis in many practical applications.^{[1][2]}

3 MatLab code

The MatLab scripts we used for our calculations are the following.

```
1 %CO32
2 %Fourier Optics
3 %Discrete Fourier Transform (DFT)
4 %Fotios Ioannis Giasemis
5 %Part 1
6
7 clear all;
8
9 disp('1 or 2? ');
10 A=input('ANS=');
11 clc
12 N=250;
13 If=zeros(1,500);
14 Rf=zeros(1,500);
15
16 for i=245:255;
17     Rf(1,i)=Rf(1,i)+1;
18 end
19
20 S1=0; S2=0;
21
22
23 for i=(-N):(N-1)
```

```

24     s1=0; s2=0;
25     for j=(-N):(N-1)
26         s1=s1+Rf(1,j+N+1)*cos(-pi*j*i/N)-If(1,j+N+1)*sin(-pi*j*i/N);
27         s2=s2+Rf(1,j+N+1)*sin(-pi*j*i/N)-If(1,j+N+1)*cos(-pi*j*i/N);
28     end
29     RF(1,i+N+1)=1/(2*N)*s1;
30     IF(1,i+N+1)=1/(2*N)*s2;
31
32 end
33
34 x=1:500; amp=sqrt(RF.^2+IF.^2);
35
36 if A==2
37     figure
38     subplot(1,2,1)
39     plot(x,Rf)
40     subplot(1,2,2)
41     plot(x,amp)
42 elseif A==1
43     plot(x,amp)
44 end

```

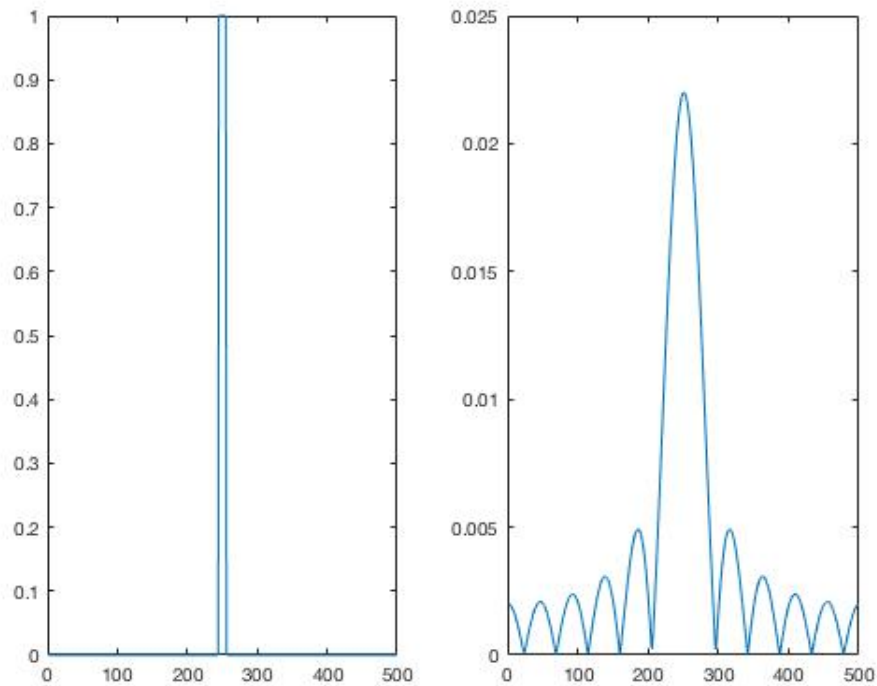


Figure 1: The real part and the amplitude of the transform

```

1 %CO32
2 %Fourier Optics
3 %Discrete Fourier Transform (DFT)
4 %Fotios Ioannis Giasemis
5 %Part 2

```

```

6
7  clear all;
8
9  N=250;
10 If=zeros(1,500);
11 Rf=zeros(1,500);
12
13 for i=(245-10):(255-10);
14     Rf(1,i)=Rf(1,i)+1;
15 end
16
17 S1=0; S2=0;
18
19
20 for i=(-N):(N-1)
21     s1=0; s2=0;
22     for j=(-N):(N-1)
23         s1=s1+Rf(1,j+N+1)*cos(-pi*j*i/N)-If(1,j+N+1)*sin(-pi*j*i/N);
24         s2=s2+Rf(1,j+N+1)*sin(-pi*j*i/N)-If(1,j+N+1)*cos(-pi*j*i/N);
25     end
26     RF(1,i+N+1)=1/(2*N)*s1;
27     IF(1,i+N+1)=1/(2*N)*s2;
28
29 end
30
31 x=1:500; amp=sqrt(RF.^2+IF.^2);
32 figure
33 subplot(1,2,1)
34 plot(x,Rf)
35 subplot(1,2,2)
36 plot(x,amp)

```

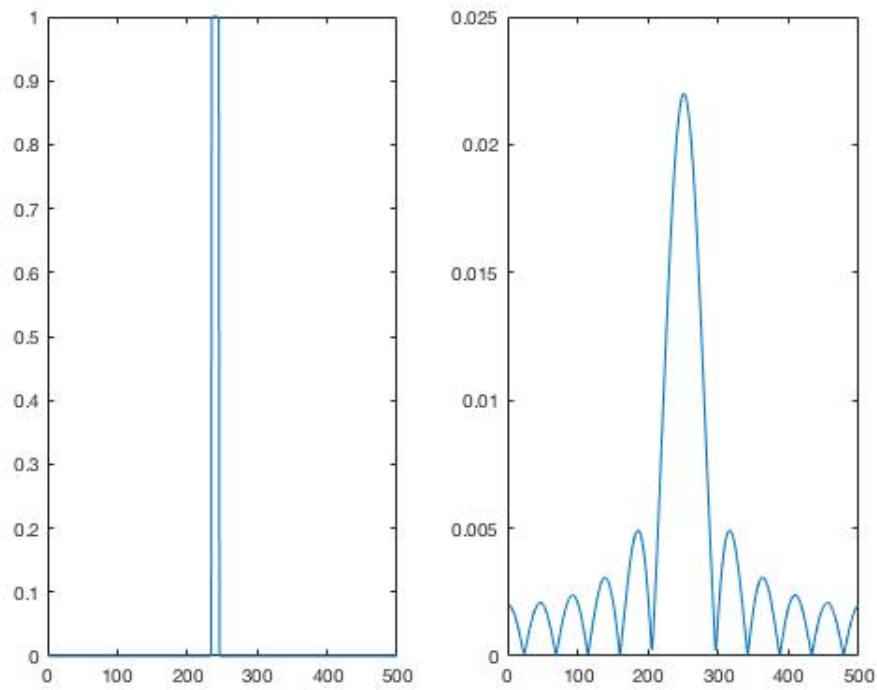


Figure 2: The real part and the amplitude of the transform

```

1 %CO32
2 %Fourier Optics
3 %Discrete Fourier Transform (DFT)
4 %Fotios Ioannis Giasemis
5 %Part 3
6
7 clear all;
8
9 N=250;
10 If=zeros(1,500);
11 Rf=zeros(1,500);
12
13 for i=240:260;
14     Rf(1,i)=Rf(1,i)+1;
15 end
16
17 S1=0; S2=0;
18
19
20 for i=(-N):(N-1)
21     s1=0; s2=0;
22     for j=(-N):(N-1)
23         s1=s1+Rf(1,j+N+1)*cos(-pi*j*i/N)-If(1,j+N+1)*sin(-pi*j*i/N);
24         s2=s2+Rf(1,j+N+1)*sin(-pi*j*i/N)-If(1,j+N+1)*cos(-pi*j*i/N);
25     end
26     RF(1,i+N+1)=1/(2*N)*s1;
27     IF(1,i+N+1)=1/(2*N)*s2;
28

```

```

29 end
30 x=1:500; amp=sqrt(RF.^2+IF.^2);
31
32 plot(x, amp)

```

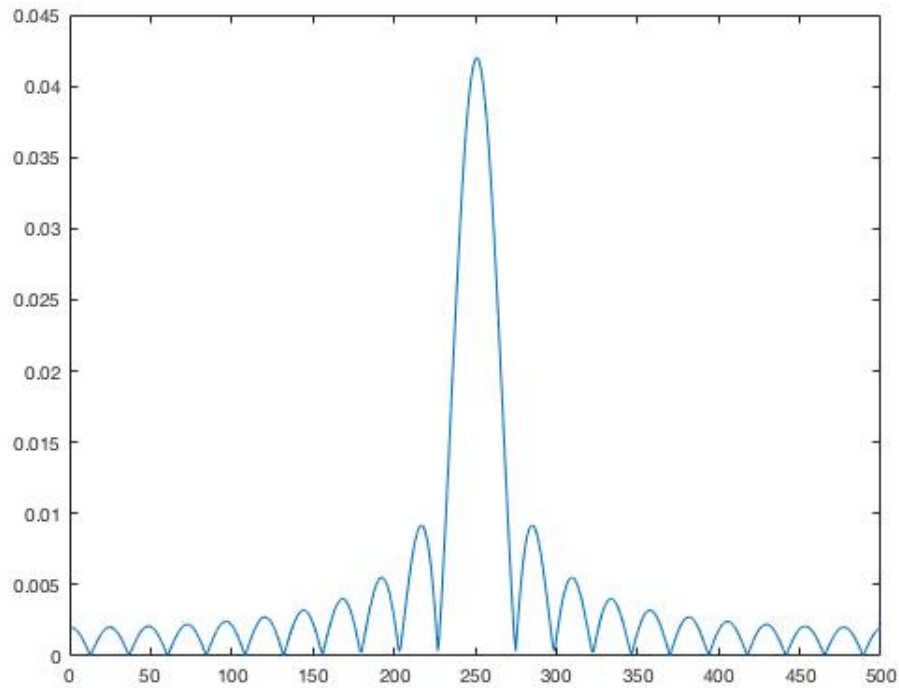


Figure 3: The amplitude of the transform

```

1 %CO32
2 %Fourier Optics
3 %Discrete Fourier Transform (DFT)
4 %Fotios Ioannis Giasemis
5 %Part 4
6
7 clear all;
8
9 N=250;
10 If=zeros(1,500);
11 Rf=zeros(1,500);
12
13 for i=240:260;
14     Rf(1,i)=Rf(1,i)+1/2;
15 end
16
17 S1=0; S2=0;
18
19
20 for i=(-N):(N-1)
21     s1=0; s2=0;
22     for j=(-N):(N-1)

```

```

23     s1=s1+Rf(1,j+N+1)*cos(-pi*j*i/N)-If(1,j+N+1)*sin(-pi*j*i/N);
24     s2=s2+Rf(1,j+N+1)*sin(-pi*j*i/N)-If(1,j+N+1)*cos(-pi*j*i/N);
25     end
26     RF(1,i+N+1)=1/(2*N)*s1;
27     IF(1,i+N+1)=1/(2*N)*s2;
28
29 end
30
31 x=1:500;amp=sqrt(RF.^2+IF.^2);
32 plot(x,amp)

```

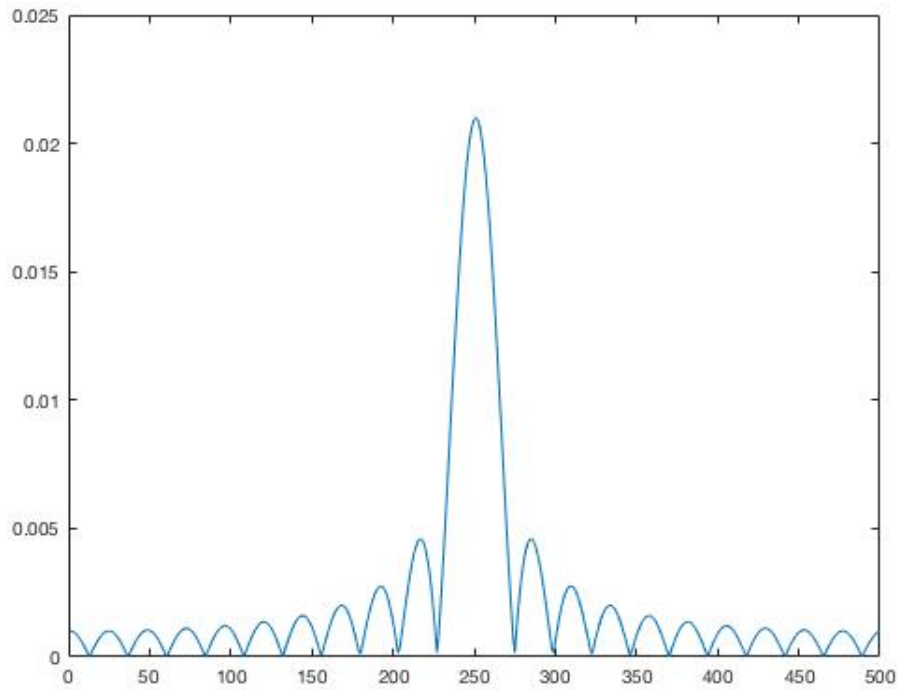


Figure 4: The amplitude of the transform

```

1  %CO32
2  %Fourier Optics
3  %Discrete Fourier Transform (DFT)
4  %Fotios Ioannis Giasemis
5  %Part 5
6
7  clear all;
8
9  N=250;
10 If=zeros(1,500);
11 Rf=zeros(1,500);
12
13 for i=240:260;
14     Rf(1,i)=Rf(1,i)+1;
15 end
16

```

```

17 s1=0; s2=0;
18
19 C=conv(Rf,Rf); C(1,1000)=0;
20
21 N=500;
22 for k=(-N):(N-1)
23     s1=0;
24
25     for j=(-N):(N-1)
26         s1=s1+C(1,j+N+1)*exp(-pi*i*j*k/N);
27     end
28
29     FC(1,k+N+1)=1/(2*N)*s1;
30
31 end
32
33 x=1:1000;
34 figure
35 plot(x,FC);

```

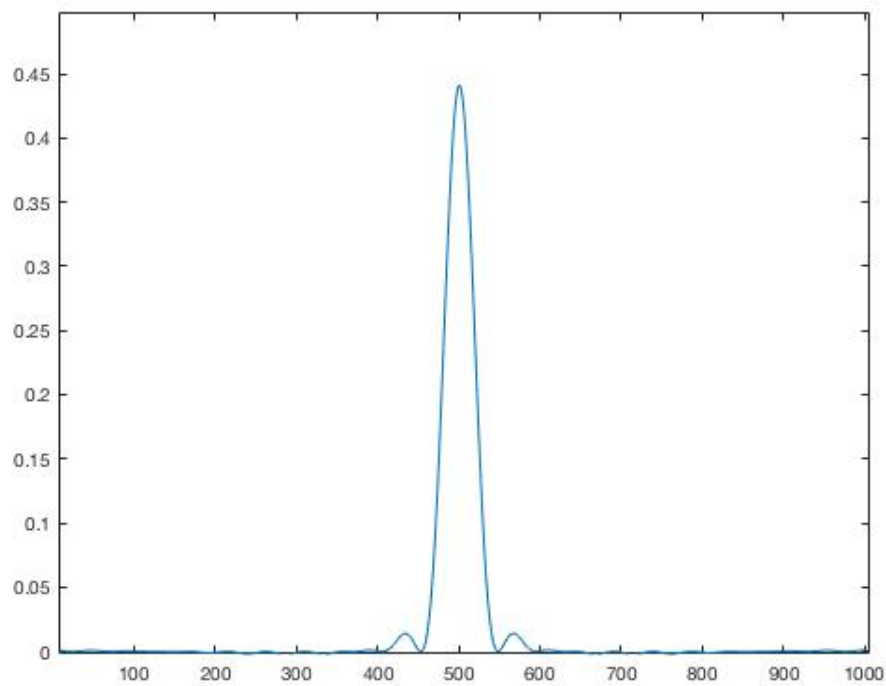


Figure 5: The amplitude of the transform of the convolution

If you do:

```

1 amp=amp^2;
2 plot(x,amp)

```

you can easily verify the Convolution theorem.

```

1 %CO32
2 %Fourier Optics

```

```

3  %Discrete Fourier Transform (DFT)
4  %Fotios Ioannis Giasemis
5  %Part 6
6
7  clear all;
8
9  clc
10 N=25; M=25;
11 If=zeros(2*N,2*M);
12 Rf=zeros(2*N,2*M);
13
14 Rf(25,25)=Rf(25,25)+1; Rf(25,26)=Rf(25,26)+1; Rf(25,24)=Rf(25,24)+1;
15 Rf(24,25)=Rf(24,25)+1; Rf(26,25)=Rf(26,25)+1;
16
17 for i=(-N):(N-1)
18     for j=(-M):(M-1)
19         s1=0; s2=0;
20         for k=(-N):(N-1)
21             for l=(-M):(M-1)
22                 s1=s1+Rf(k+N+1,l+N+1)*cos(-pi*k*i/N-pi*l*j/M)-If(k+N+1,l+N+1)*
23                     sin(-pi*k*i/N-pi*l*j/M);
24                 s2=s2+Rf(k+N+1,l+N+1)*sin(-pi*k*i/N-pi*l*j/M)+If(k+N+1,l+N+1)*
25                     cos(-pi*k*i/N-pi*l*j/M);
26                 RF(j+N+1,i+N+1)=1/(4*N*M)*s1;
27                 IF(j+N+1,i+N+1)=1/(4*N*M)*s2;
28             end
29         end
30     end
31 end
32
33 amp=sqrt(RF.^2+IF.^2);
34
35 figure
36     subplot(1,2,1)
37     surf(Rf)
38     subplot(1,2,2)
39     surf(amp)

```

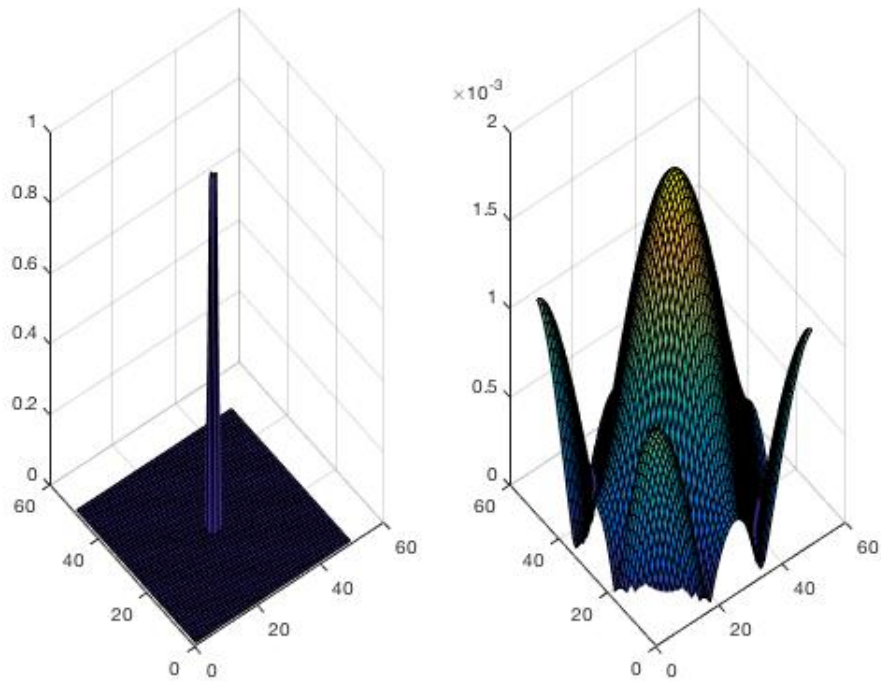



Figure 6: The real part and the amplitude of the transform

```

1  %CO32
2  %Fourier Optics
3  %Discrete Fourier Transform (DFT)
4  %Fotios Ioannis Giasemis
5  %Part 7
6
7  clear all;
8
9  clc
10 N=25; M=25;
11 If=zeros(2*N,2*M);
12 Rf=zeros(2*N,2*M);
13
14 for m=24:26
15     for n=15:35
16         Rf(m,n)=Rf(m,n)+1;
17     end
18 end
19
20 for i=(-N):(N-1)
21     for j=(-M):(M-1)
22         s1=0; s2=0;
23         for k=(-N):(N-1)
24             for l=(-M):(M-1)
25                 s1=s1+Rf(k+N+1,l+N+1)*cos(-pi*k*i/N-pi*l*j/M)-If(k+N+1,l+N+1)*
26                     sin(-pi*k*i/N-pi*l*j/M);
27                 s2=s2+Rf(k+N+1,l+N+1)*sin(-pi*k*i/N-pi*l*j/M)+If(k+N+1,l+N+1)*
28                     cos(-pi*k*i/N-pi*l*j/M);

```

```

27         RF(j+N+1,i+N+1)=1/(4*N*M)*s1;
28         IF(j+N+1,i+N+1)=1/(4*N*M)*s2;
29     end
30
31     end
32 end
33
34 end
35
36 amp=sqrt(RF.^2+IF.^2);
37
38 figure
39     subplot(1,2,1)
40     surf(Rf)
41     subplot(1,2,2)
42     surf(amp)

```

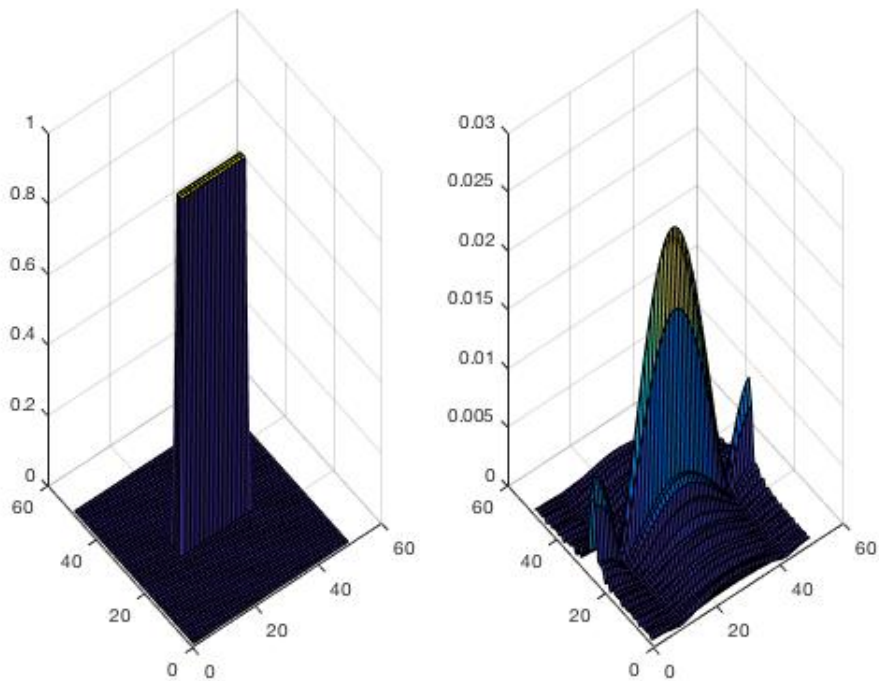


Figure 7: The real part and the amplitude of the transform

4 References

- [1] Strang, Gilbert (May/June 1994). "Wavelets". American Scientist. 82 (3): 253. Retrieved 8 October 2013.
- [2] https://en.wikipedia.org/wiki/Discrete_Fourier_transform