INFORME DE LA PRÁCTICA 4

Diego Ismael Antolinos García
Andrés Ruz Nieto

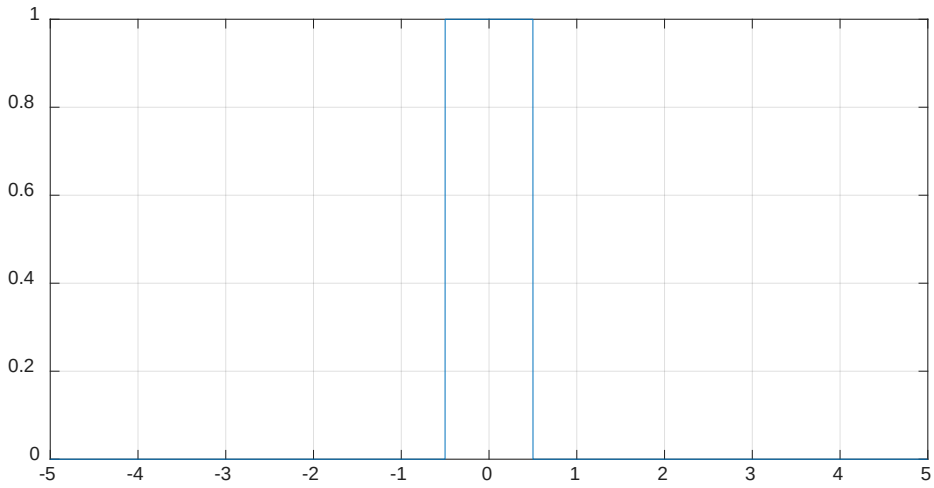
1. La transformada de Fourier

a. Problemas

i. Generar en MATLAB las siguientes señales aperiódicas

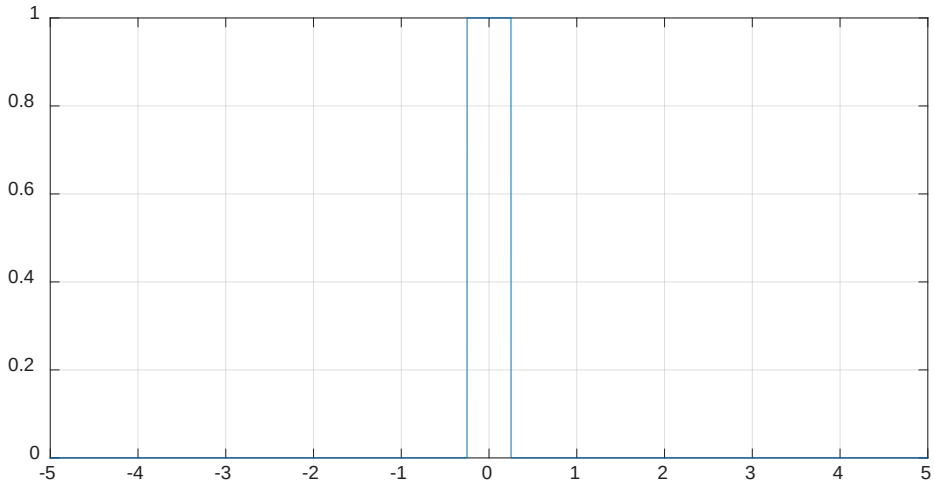
1. Señal a

```
dt=0.001; T=10;
t=-T/2:dt:T/2-dt;
x=zeros(1, length(t));
x(find(abs(t)<=0.5))=1;
figure, plot(t,x);
grid on
```



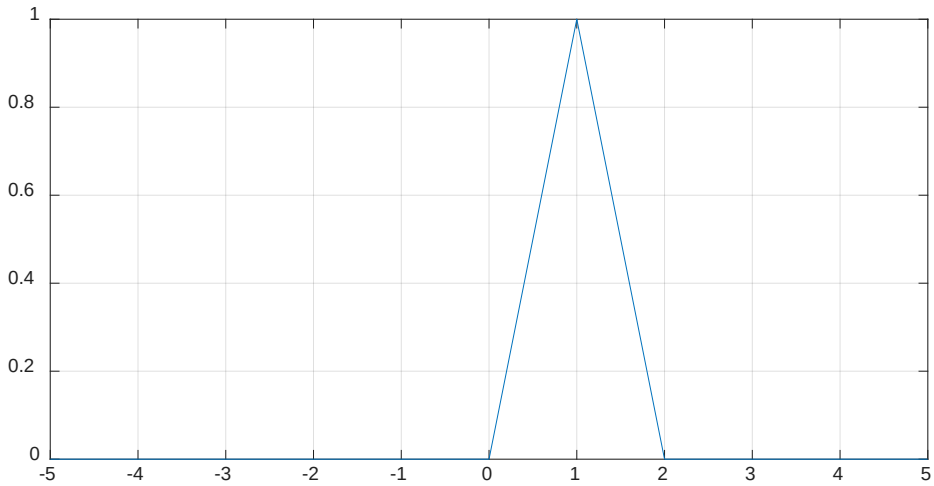
2. Señal b

```
dt=0.001; T=10;
t=-T/2:dt:T/2-dt;
x=zeros(1, length(t));
x(find(abs(t)<=0.25))=1;
figure, plot(t,x);
grid on
```



3. Señal c

```
dt=0.001; T=10;
t=-T/2:dt:T/2-dt;
x=zeros(1, length(t));
x(find (t<=1 & t>0))=t(find (t<=1 & t>0));
x(find (t<=2 & t>1))=2-t(find (t<=2 & t>1));
figure, plot(t,x);
grid on
```



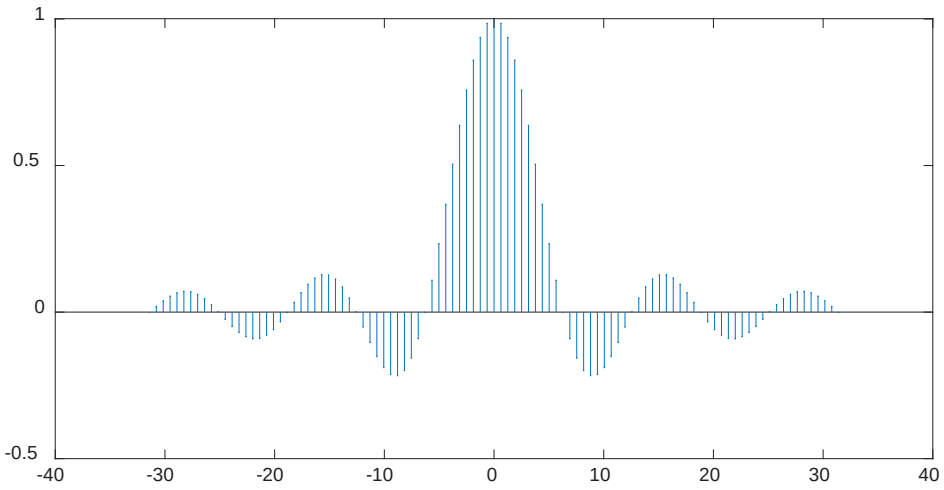
Debemos destacar que hemos utilizado un "periodo ficticio" (T=10), con el objetivo de delimitar el eje temporal t, y poder representarlas.

ii. Asumiendo que las señales anteriores son periódicas

1. Calcular su DSF mediante ak=sfourier(x,T,N,dt), y normalizar por T(ak=T*ak).Dado que cada armónico está asociado a la frecuencia k2π/T, representar las series de Fourier normalizadas (mediante stem(w,ak,'.')) en el eje de frecuencias. Crear previamente el vector de frecuencias w=2*π*k/T.

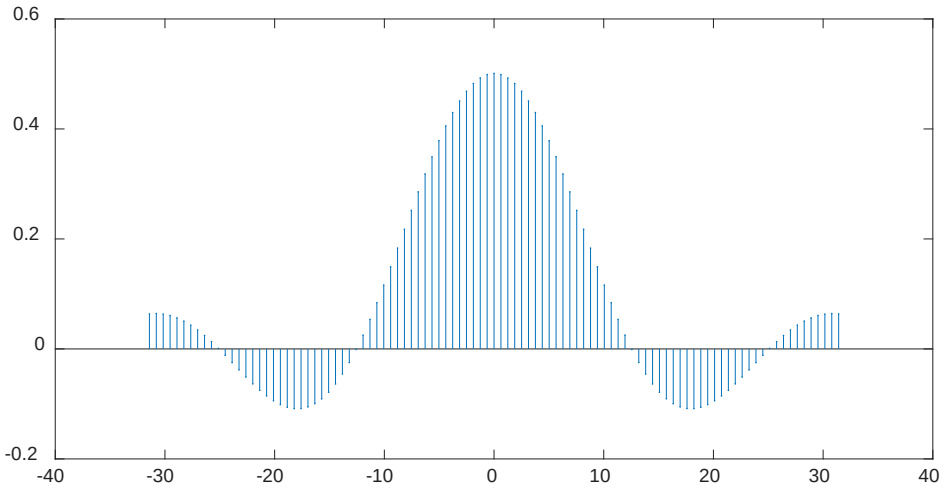
   a. Señal a

```
N=50;
k=-N:N;
w=2*pi*k/T;
ak1=sfourier(x,T,N,dt)
ak1=T*ak1;
figure, stem(w,ak1,'.');
```



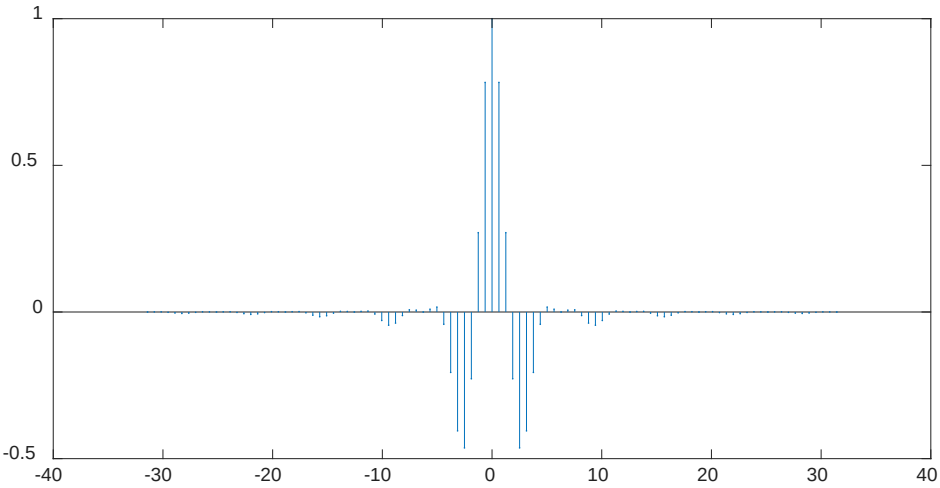
b. Señal b

```
N=50;
k=-N:N;
w=2*pi*k/T;
ak1=sfourier(x,T,N,dt)
ak1=T*ak1;
figure, stem(w,ak1,'.');
```



c. Señal c

```
N=50;
k=-N:N;
w=2*pi*k/T;
ak1=sfourier(x,T,N,dt)
ak1=T*ak1;
figure, stem(w,ak1,'.');
```



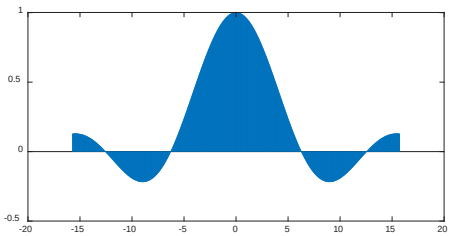
2. Aumentar la longitud de la señal

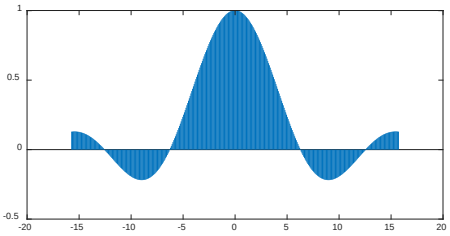
```
x=[zeros(1,Ti) x zeros(1,Ti)];
figure, plot(Ti,x);
```

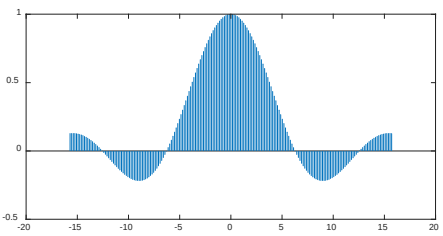
3. Calcular las nuevas series de Fourier normalizado

a. Señal a

```
N=50;
for i=1:3
T=2*T;
N=2*N;
k=-N:N;
w=2*pi*k/T;
Ti = floor(length(x)/2);
x=[zeros(1,Ti) x zeros(1,Ti)];
ak=sfourier(x,T,N,dt);
ak=ak*T;
figure;stem(w,ak,'.');
end
```

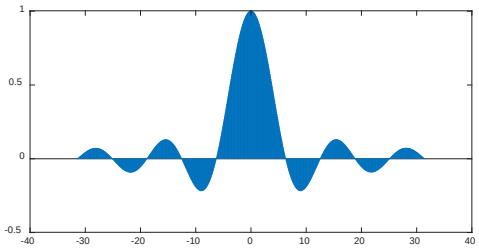
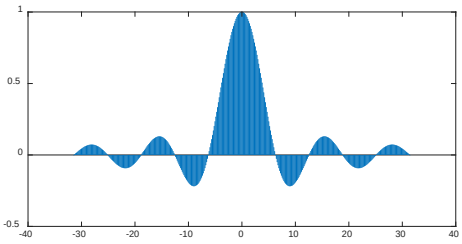
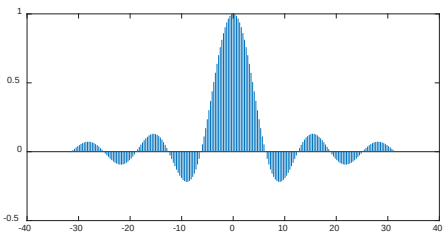






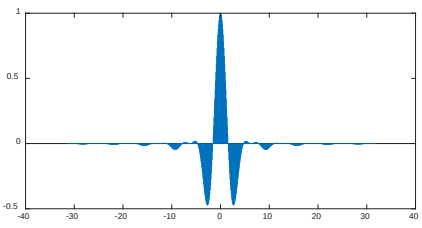
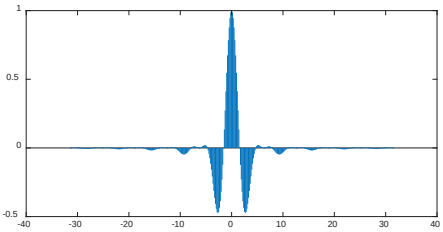
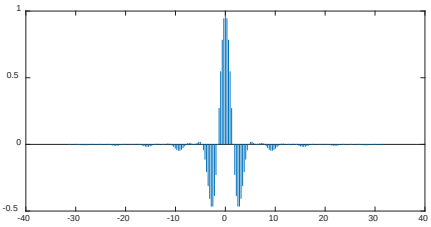
b. Señal b

```
N=50;
for i=1:3
T=2*T;
N=2*N;
k=-N:N;
w=2*pi*k/T;
Ti = floor(length(y)/2);
y=[zeros(1,Ti) y zeros(1,Ti)];
ak=sfourier(y,T,N,dt);
ak=ak*T;
figure;stem(w,ak,'.');
end
```

Señal c

```
N=50;
for i=1:3
T=2*T;
N=2*N;
k=-N:N;
w=2*pi*k/T;
Ti = floor(length(z)/2);
z=[zeros(1,Ti) z zeros(1,Ti)];
ak=sfourier(z,T,N,dt);
ak=ak*T;
figure;stem(w,ak,'.');
end
```

Vemos que al aumentar el periodo la envolvente de las señales no cambia y el número de coeficientes ak aumenta conforme se aumente el periodo. Cuando el periodo sea infinito los ak “formarán” una señal continua (que se corresponde con la transformada de Fourier).

4. Escribe un programa que calcule la TF

```
function [X,w]=tfourier(x,t,dw,wmax)
dt=0.001;
w=-wmax:dw:wmax-dw;
X = zeros(1,length(w));
for i=1:length(w)
```

```
    X(i)=sum(x.*exp(-j.*w(i).*t)).*dt;
end
```

5. Escribe un programa que calcule la TF-1

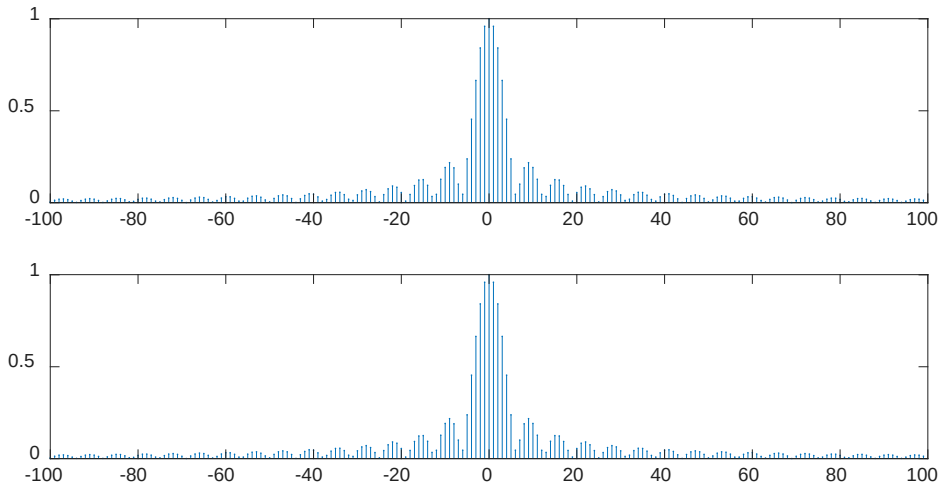
```
function [x,t] = itfourier (X,w,dt,tmax)
[X1,w]= tfourier(conj(X),w,dt,tmax);
x=conj(X1)./(2*pi);
end
```

6. Comprueba las siguientes propiedades

Previamente hemos definido las señales y calculado sus transformadas para que sea más fácil luego trabajar con ellas. Además, también indicamos dw y wmáx:

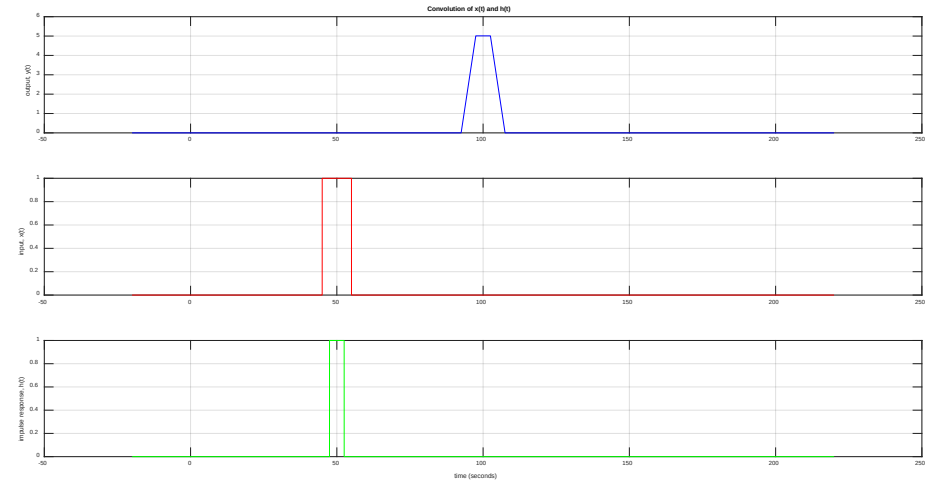
a. Linealidad

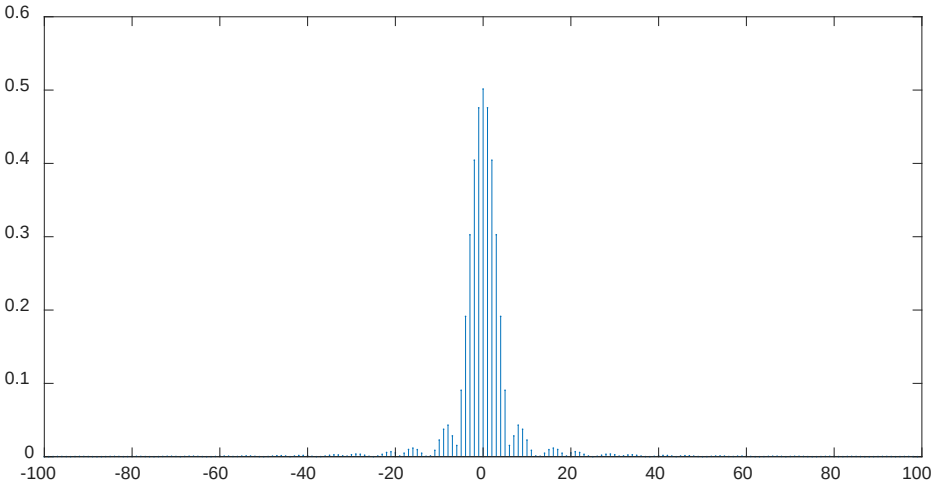
```
%Definimos parametros
T=10;
dt=0.001;
t=-T/2:dt:T/2-dt;
dw=1;
wmax=100;
%Señal a
xa=zeros(1,length(t));
ti=find(abs(t)<=0.5); xa(ti)=1;
%Señal b
xb=zeros(1,length(t));
ti=find(abs(t)<=0.25); xa(ti)=1;
%Suma de transformadas
[XA,w]=tfourier(xa,t,dw,wmax);
[XB,w]=tfourier(xb,t,dw,wmax);
L=zeros(1,(length(XA)+length(XB)));
L=XA+XB;
subplot(2,1,1);
stem(w,abs(L),'.');
%Transformada de la suma
suma=zeros(1,(length(xa)+length(xb)));
suma=xa+xb;
[SUMA,w]=tfourier(suma,t,dw,wmax);
subplot(2,1,2);
stem(w,abs(SUMA),'.');
```



b. Convolucion

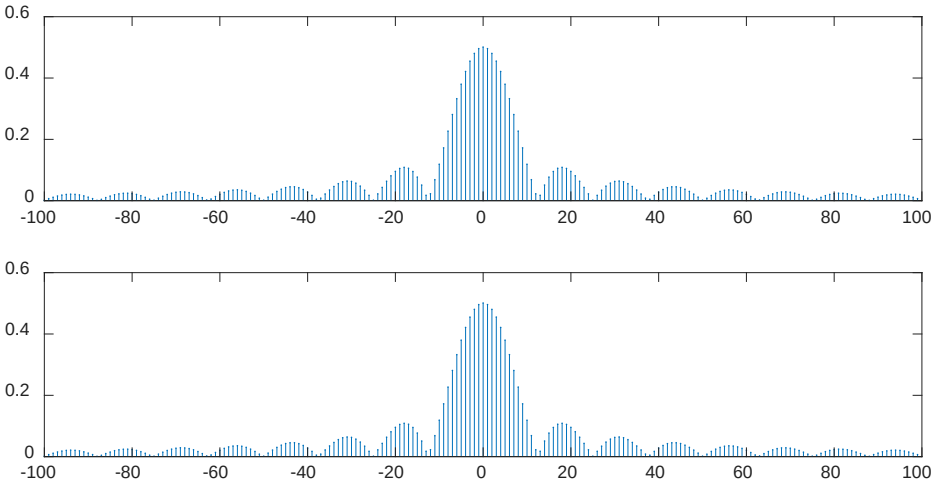
```
T=10;
%parámetros necesarios
dt=0.001;
t=-T/2:dt:T/2-dt;
dw=1;
wmax=100;
xa=zeros(1,length(t));
%generamos señal a
ti=find(abs(t)<=0.5);
xa(ti)=1;
xb=zeros(1,length(t));
%generamos señal b
ti=find(abs(t)<=0.25);
xb(ti)=1;
y=conv_continua(xa,0,xb,0,0.01);
%convolución
[XA,w]=tfourier(xa,t,dw,wmax);
[XB,w]=tfourier(xb,t,dw,wmax);
PROD=XA.*XB;
%Producto de transformadas
figure;
stem(w,abs(PROD),'.');
```





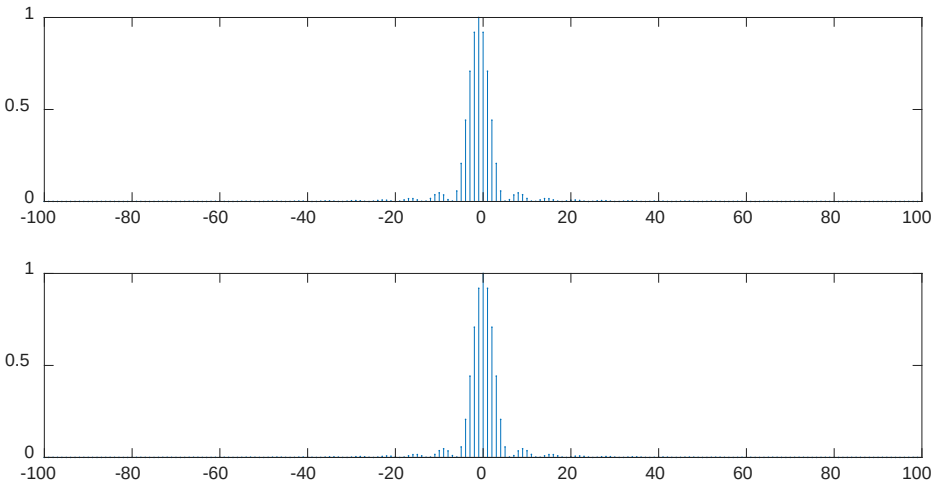
c. Desplazamiento temporal

```
T=10;
dt=0.001;
t=-T/2:dt:T/2-dt;
dw=1;
wmax=100;
j=sqrt(-1);
to=1;
xb=zeros(1,length(t));
ti=find(abs(t)<=0.25);
xb(ti)=1;
[B,w]=tfourier(xb,t,dw,wmax);
ND=exp(-j*w*to).*B;
subplot(2,1,1);
stem(w,abs(ND),'.');
d=desplazar(xb,to);
[D,w]=tfourier(d,t,dw,wmax);
subplot(2,1,2);
stem(w,abs(D),'.');
```



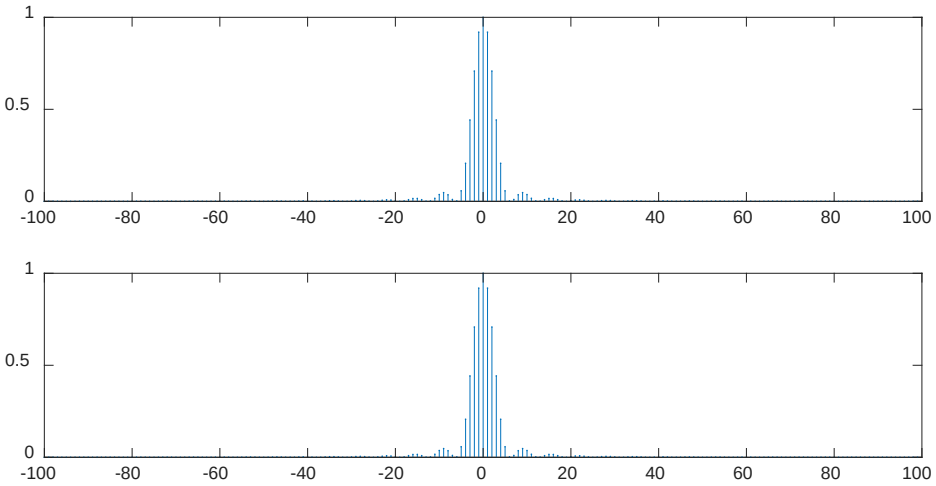
d. Inversion

```
T=10;
%parámetros necesarios
dt=0.001;
t=-T/2:dt:T/2-dt;
dw=1;
wmax=100;
xc=zeros(1,length(t));
%generamos la señal c
ti=find((t>0)&(t<=1));
xc(ti)=t(ti);
ti=find((t>1)&(t<=2));
xc(ti)=2-t(ti);
[I,w]=tfourier(xc,t,dw,wmax);
I1=invertir(I,t);
subplot(2,1,1);
stem(w,abs(I1),'.');
i1 = invertir(xc,t);
[I2,w]=tfourier(i1,t,dw,wmax);
subplot(2,1,2);
stem(w,abs(I2),'.');
```



e. Conjugacion

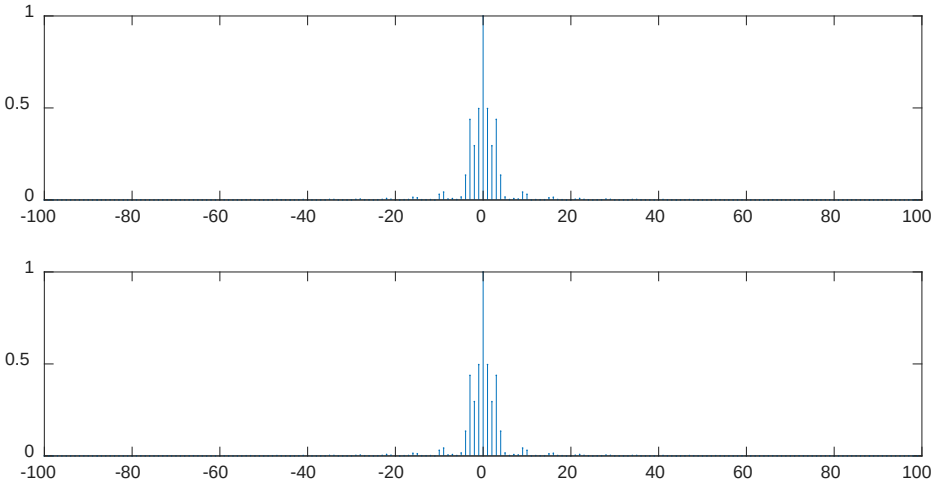
```
T=10;
%parámetros necesarios
dt=0.001;
t=-T/2:dt:T/2-dt;
dw=1;
wmax=100;
xc=zeros(1,length(t));
ti=find((t>0)&(t<=1));
xc(ti)=t(ti);
ti=find((t>1)&(t<=2));
xc(ti)=2-t(ti);
[XC,w]=tfourier(xc,t,dw,wmax);
%Transformamos y conjugamos
XCCONJ=conj(XC);
subplot(2,1,1);
stem(w,abs(XCCONJ),'.');
xc1=invertir(xc,t);
xc1conj=conj(xc1);
[XC1CONJ,w]=tfourier(xc1conj,t,dw,wmax);
subplot(2,1,2);
stem(w,abs(XC1CONJ),'.');
```



f. Real = Parte par

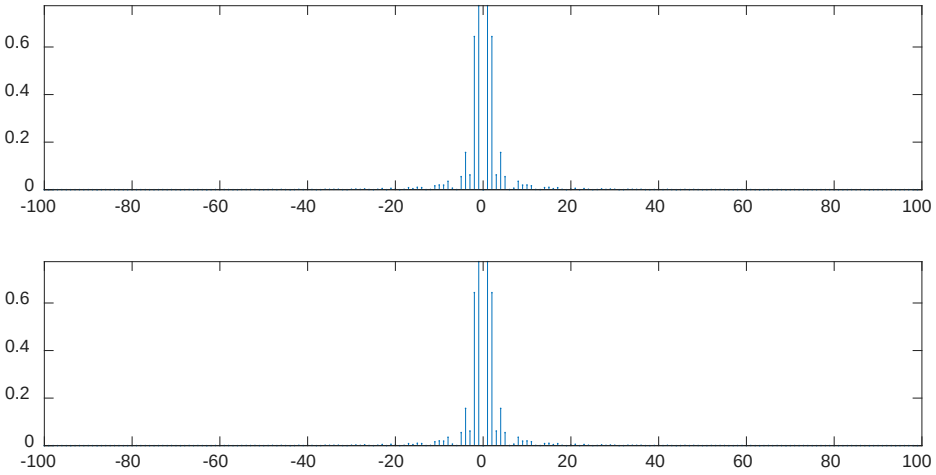
```
T=10;
%parámetros necesarios
dt=0.001;
t=-T/2:dt:T/2-dt;
dw=1;
wmax=100;
xc=zeros(1,length(t));
%generamos la señal c
    ti=find((t>0)&(t<=1));
    xc(ti)=t(ti);
    ti=find((t>1)&(t<=2));
    xc(ti)=2-t(ti);
    [XC,w]=tfourier(xc,t,dw,wmax);
    %Transformada y después parte real
    rXC=real(XC);
    subplot(2,1,1);
```

```
18.     stem(w,abs(rXC),'.');
19.     xc1=invertir(xc,t);
20.
21.     par=(xc+xc1)/2;
22.     [PAR,w]=tfourier(par,t,dw,wmax);
23.     subplot(2,1,2);
24.     stem(w,abs(PAR),'.');
```



a. Real = Parte imaginaria

```
25.     T=10;
26.     %parámetros necesarios
27.     dt=0.001;
28.     t=-T/2:dt:T/2-dt;
29.     dw=1;
30.     wmax=100;
31.     xc=zeros(1,length(t));
32.     ti=find((t>0)&(t<=1));
33.     xc(ti)=t(ti);
34.     ti=find((t>1)&(t<=2));
35.     xc(ti)=2-t(ti);
36.     [XC,w]=tfourier(xc,t,dw,wmax);
37.     %Transformada y después parte imaginaria
38.     imXC=imag(XC);
39.     subplot(2,1,1);
40.     stem(w,abs(imXC),'.');
41.     xc1=invertir(xc,t);
42.     %parte impar y después transformada
43.     impar=(xc-xc1)/2;
44.     [IMPAR,w]=tfourier(impar,t,dw,wmax);
45.     subplot(2,1,2);
46.     stem(w,abs(IMPAR),'.');
```



a. Diferenciacion

```
T=10;
%parámetros necesarios
dt=0.001;
t=-T/2:dt:T/2-dt;
dw=1;
wmax=100;
j=sqrt(-1);
xc=zeros(1,length(t));
%generamos la señal c
ti=find((t>0)&(t<=1));
xc(ti)=t(ti);
ti=find((t>1)&(t<=2));
xc(ti)=2-t(ti);
[XC,w]=tfourier(xc,t,dw,wmax);
%Transformada más multiplicacion jw
XC1=j.*w.*XC;
subplot(2,1,1);
stem(w,abs(XC1),'.');
dxc=gradient(xc,t);
%Transformada de la derivada
[DXC,w]=tfourier(dxc,t,dw,wmax);
subplot(2,1,2);
stem(w,abs(DXC),'.');
```