Lab 4 Fourier Transform

Objective

- Learn the Fourier transform of continuous-time signal with MATLAB.
- Analyze the signals with Fourier Transform.
- Analyze the LTI system with system models.

Content

For a periodic signal, as $T \to \infty$, it becomes an aperiodic signal. Its spectrum changes from the discrete spectrum to a continuous spectrum. At the same time the amplitude of the spectrum becomes smaller, but the relative difference still exists, as shown in Figure 1.

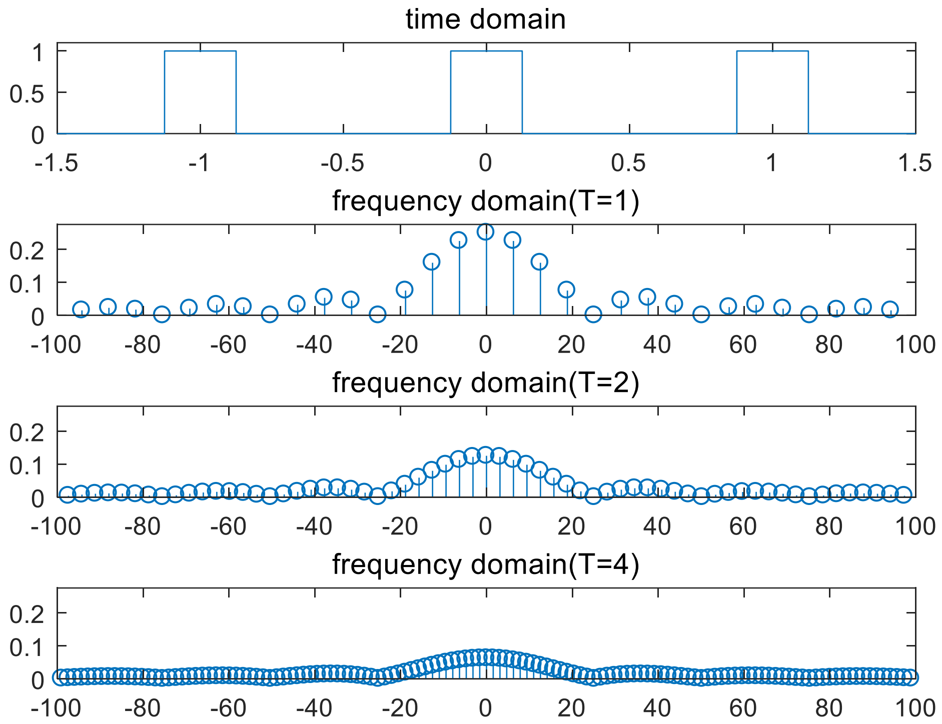


Figure 1 Influence of T

The spectrogram comparison of the periodic signal and the aperiodic signal is shown in Figure 2.

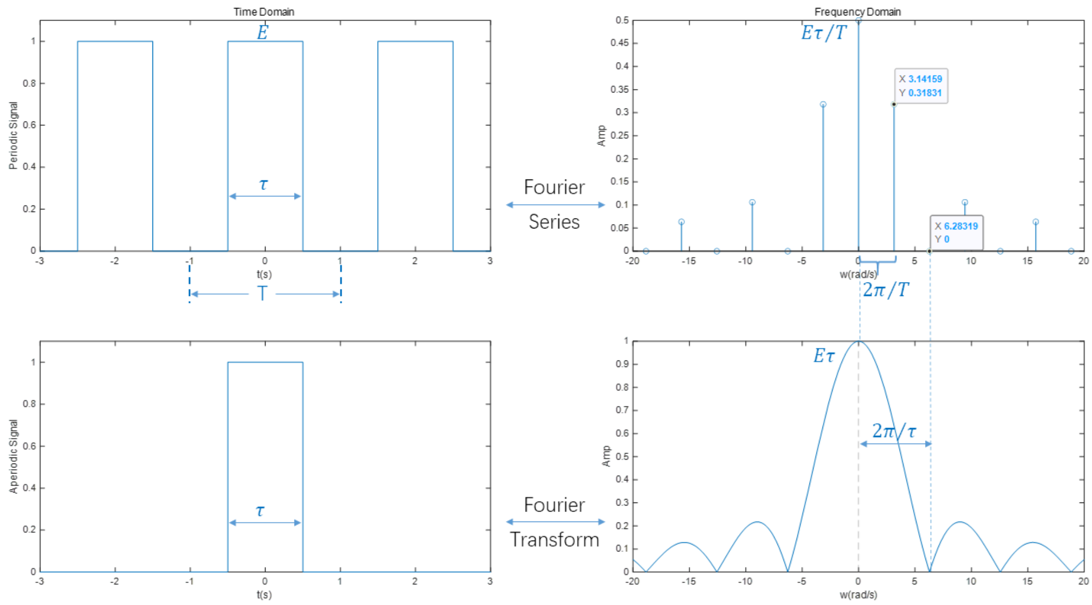


Figure 2 Fourier Series and Fourier Transform

Fourier Transform can analyze both periodic and aperiodic signals, while the signals should satisfy certain conditions.

Signal Analysis

Fourier Transform and Inverse Fourier Transform with Symbolic Method

The Fourier transform of f(t) is defined as:

$$\mathrm{F}(j\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t}\,dt$$

And the inverse Fourier transform is defined as:

$$\mathrm{f(t)} = \frac{1}{2\pi}\int_{-\infty}^{+\infty} F(j\omega)e^{j\omega t}\,\mathrm{d}\omega$$

For signals that can be expressed by an expression, use the function **fourier** and **ifourier** to do the Fourier transfer and inverse Fourier transfer with MATLAB. Both **fourier** and **ifourier** are symbolic methods. The format of the two functions is listed in Table 1.

Table 1 Format of fourier and ifourier

Format	Description
F=fourier(f)	Do Fourier transform of function f with variable x. The result F is the function with variable w.
F=fourier(f,v)	Do Fourier transform of function f with variable x. The result F is the function with variable v.
F=fourier(f,u,v)	Do Fourier transform of function f with variable u. The result F is the function with variable v.

f=ifourier(F)	Do inverse Fourier transform of function F with variable w. The result f is the function with variable x.
f = ifourier(F,u)	Do inverse Fourier transform of function F with variable w. The result f is the function with variable u.
f = ifourier(F,v,u)	Do inverse Fourier transform of function F with variable v. The result f is the function with variable u.

Use function **abs** and **angle** to find out the amplitude and phase of the signal in frequency domain.
Example: Do the Fourier transfer of the single-sided exponential signal.

$$f(t) = \begin{cases} e^{-2t}, t > 0 \\ 0, t < 0 \end{cases}$$

```
clear;clf;
syms t
a = 2;
f = exp(-(a*t))*heaviside(t);
subplot(2,2,[1 3]);fplot(f,[0 3]);grid on;
xlabel('t(s)');ylabel('f(t)');title('time domain')
F = fourier(f);
subplot(2,2,2);fplot(abs(F),[-10 10]);grid on;
xlabel('w(rad)');ylabel('Amplitude');title('Amp-Freq')
subplot(2,2,4);fplot(angle(F),[-10 10]);grid on;
xlabel('w(rad)');ylabel('Phase(radian)');title('Phase-Freq')
```

Example: Do the inverse Fourier transfer of $F(j\omega) = \frac{1}{1+\omega^2}$.

```
syms t w
ifourier(1/(1+w^2),t);
```

Fourier Transform with Numeric Method

Fourier Transform with Function fft

Function **fft** is used to do Fourier transform in the numeric method. Format **fft(x)** computes the discrete Fourier transform of x using a fast Fourier transform algorithm. Format **fft(x,n)** returns the n-point DFT.
Example: x=cos(2*pi*10.*t)+2sin(2*pi*15.*t)+3cos(2*pi*20.*t)). Observe the signal in frequency domain.

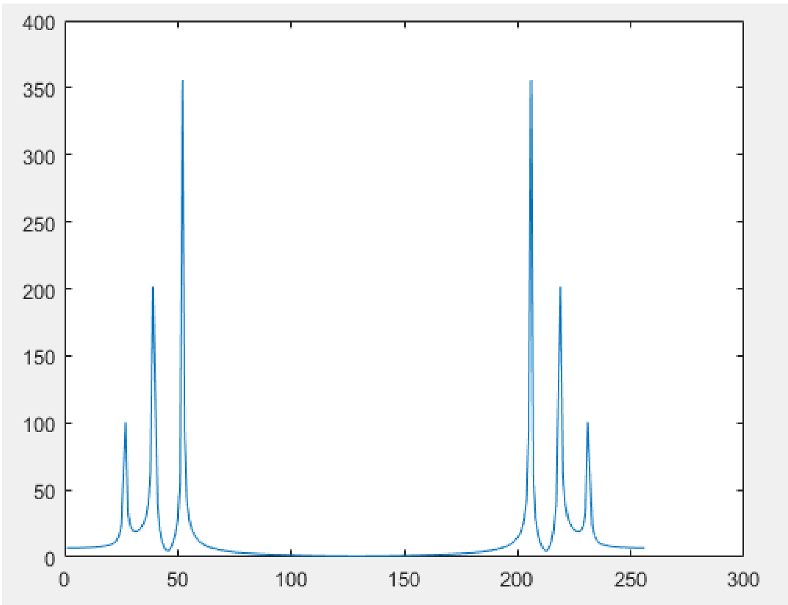
```
clear;clf;
N = 256;
dt = 0.01; Fs = 1/dt;
t = [0:N-1]*dt;
x = 1*cos(2*pi*10.*t) + 2*sin(2*pi*15.*t) + 3*cos(2*pi*20.*t);
X = fft(x);
plot(abs(X))
```

1. Figure 3 (a) shows the graphics of X, and (b) shows the value. From (b), it is easy to find that

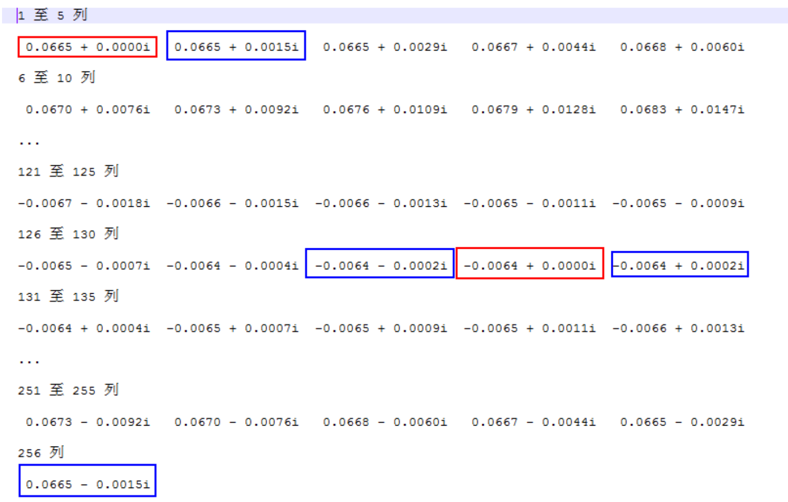
the 1th and the 129th items are real numbers, and the items at the sides of 129th are conjugated. This is because MATLAB present frequency components from f=0 to f=Fs (from DC to the sampling frequency). However, we are only interested in the range from 0 to Fs/2. So only the 1th to 128th items (N/2) are valuable.

2. The corresponding frequency to the X(n) is calculated by $f(n) = \frac{n-1}{N} Fs$, $\mathrm{n} = 1,2,\ldots,\mathrm{N}$, where $\frac{Fs}{N}$ is the frequency resolution.

3. As mentioned in 1 and 2, the effective observation items are from 1th to $\frac{N}{2}$ th, so the corresponding frequency is from 0 to $\frac{Fs}{2}$. And this is the basis for choosing a suitable Fs. According to the characters of function **fft**, it is recommended that N prefers 2^n.



(a)



```
1 至 5 列
0.0665 + 0.0000i   0.0665 + 0.0015i   0.0665 + 0.0029i   0.0667 + 0.0044i   0.0668 + 0.0060i
6 至 10 列
0.0670 + 0.0076i   0.0673 + 0.0092i   0.0676 + 0.0109i   0.0679 + 0.0128i   0.0683 + 0.0147i
...
121 至 125 列
-0.0067 - 0.0018i  -0.0066 - 0.0015i  -0.0066 - 0.0013i  -0.0065 - 0.0011i  -0.0065 - 0.0009i
126 至 130 列
-0.0065 - 0.0007i  -0.0064 - 0.0004i  -0.0064 - 0.0002i  -0.0064 + 0.0000i  -0.0064 + 0.0002i
131 至 135 列
-0.0064 + 0.0004i  -0.0065 + 0.0007i  -0.0065 + 0.0009i  -0.0065 + 0.0011i  -0.0066 + 0.0013i
...
251 至 255 列
0.0673 - 0.0092i   0.0670 - 0.0076i   0.0668 - 0.0060i   0.0667 - 0.0044i   0.0665 - 0.0029i
256 列
0.0665 - 0.0015i
```

(b)

Figure 3 fft Result

4. The amplitude information of the frequency domain signal can be obtained by the function **abs()**. In order to get correct amplitude value, for a continuous periodic signal, all the elements need to be divided by N; for a continuous aperiodic signal, all the elements need to be multiplied by dt. This is caused by the algorithm of **fft**. Then all the elements, except the 1st one (the DC component), need to be multiplied by 2 to transfer the energy at the right side of the 129th element to the left side.

As we know the amplitude and frequency of x is 1@10Hz, 2@15Hz, 3@20Hz. Let's change the code to:

```
clear;clf;
N = 256;
Fs = 100;
dt = 1/Fs;
df = Fs/N;
t = [0:N-1]*dt;
f = [0:N-1]*df;
x = 1*cos(2*pi*10.*t)+2*sin(2*pi*15.*t)+3*cos(2*pi*20.*t);
X = abs(fft(x))/N;
X = [X(1),2*X(2:N/2)];
f = f(1:N/2);
plot(f,X); xlabel('f(Hz)');ylabel('Amplitude')
```

The result is show in Figure 4. This is called unilateral amplitude spectrum, the frequency range of which is 0~∞Hz.

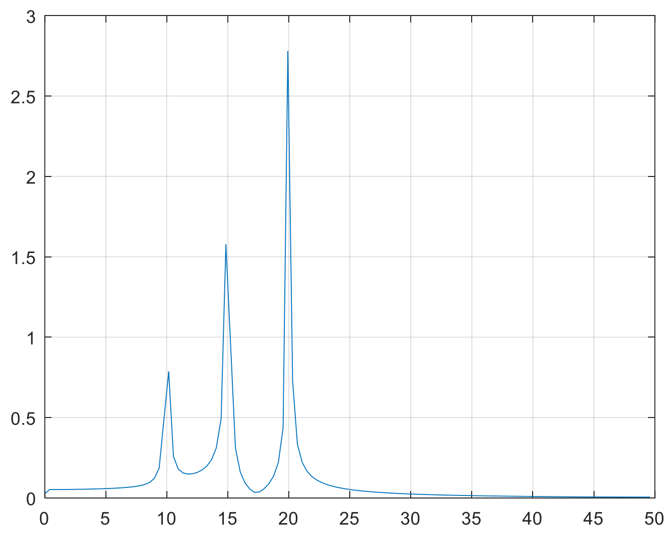


Figure 4 Adjustment

5. There is still some deviation of amplitude caused by the spectrum leak. To reduce the spectrum leak, the frequency of signal should be an integer multiple of the frequency resolution. That is $f = \frac{Fs}{N} * k$. Adjust Fs and N to make the it. Some adjustment of Fs and N is made in Table 2. Plot them to view the effect. The result is shown in Figure 5.

Table 2 Adjusting of Fs and N

k=f*N/Fs,N=256			
	1@10Hz	2@15Hz	3@20Hz
Fs=120	21.33	32	42.67
Fs=160	16	24	32
k=f*N/Fs,Fs=100			
	1@10Hz	2@15Hz	3@20Hz
N=260	26	39	52
N=250	25	37.5	50

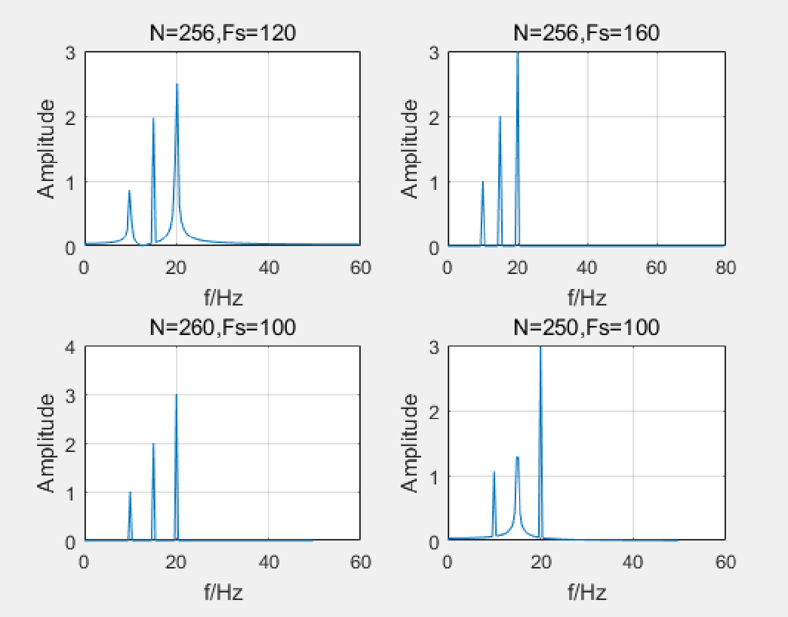
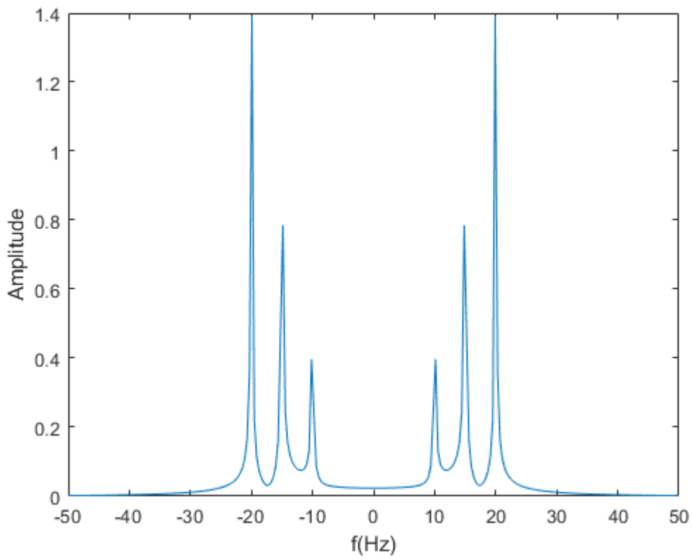


Figure 5 Adjust Fs and N

To obtain the bilateral amplitude spectrum of the signal, use function **fftshift** after **fft** to move the zero frequency component to the center of the spectrum. When doing so, there's no need to multiply the amplitude by 2. The frequency range should be -∞~∞Hz.

```
clear;clf;
N = 256;
Fs = 100;
dt = 1/Fs;
df = Fs/N;
t = [0:N-1]*dt;
f = [-N/2:N/2-1]*df;
x = 1*cos(2*pi*10.*t)+2*sin(2*pi*15.*t)+3*cos(2*pi*20.*t);
X = abs(fftshift(fft(x)))/N;
plot(f,X); xlabel('f(Hz)');ylabel('Amplitude')
```



About Using N and dt to Adjust the Amplitude

Function **fft** is an efficient DFT algorithm. It is an operation of discrete signals. The definition of DFT is:

$$X_{DFT}(k) = \sum_{n=0}^{N-1} x(n)e^{-jk\frac{2\pi}{N}n} = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k = 0, \cdots, N-1$$

When performing **fft** on a continuous signal, the continuous signal needs to be sampled first. For a continuous periodic signal x(t) with a period T_0, its Fourier series is like:

$$\mathrm{X(jk}w_0) = \frac{1}{T_0}\int_0^{T_0} x(t)e^{-jkw_0t}dt$$

After discretizing T_0 with N point, we define: $T_s = \frac{T_0}{N}$, $\mathrm{t} = \mathrm{k}T_s$, $dt = T_s$, $T_0 = \mathrm{N}T_s$. Thus

$$\mathrm{X(jk}w_0) = \frac{1}{T_0}\int_0^{T_0} x(t)e^{-jkw_0t}dt = \frac{T_s}{T_0}\sum_{n=0}^{N-1} x(nT_s)e^{-jk\frac{2\pi}{T_0}nT_s}$$

$$= \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-jk\frac{2\pi}{N}n} = \frac{1}{N}X_{DFT}(k) = X_T(k)$$

So when using function fft to find out the Fourier transform of a continuous periodic function, we need to divide Fn by N.

As for a continuous aperiodic signal x(t), we have $\mathrm{X(jw)} = \int_{-\infty}^{+\infty} x(t)e^{-jwt}dt$. Divide the signal according to the following Figure 6, we get

$$\mathrm{X(jw)} = \int_{-\infty}^{+\infty} x(t)e^{-jwt}dt = \sum_{n=0}^{N-1}\int_{nT_s}^{(n+1)T_s} x(t)e^{-jwt}dt$$

$$= \sum_{n=0}^{N-1} x(nT_s)e^{-jw\cdot nT_s} \cdot T_s \quad = \sum_{n=0}^{N-1} x(n)e^{-jk\cdot\frac{2\pi}{N}\cdot n} \cdot T_s = T_s \cdot X_{DFT}(k)$$

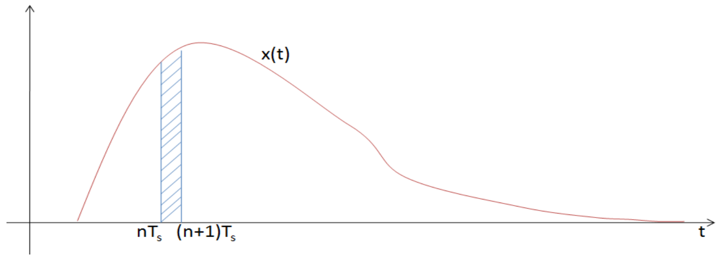


Figure 6 Divide the signal by Ts

So when using function fft to find out the Fourier transform of a continuous aperiodic function, we need to multiply Fn by dt.

To do the inverse Fourier transform use function **ifft**(X) or **ifft**(X,n).

Inverse Fourier Transform with Function ifft

After we use **fft** to convert the signal from time domain to frequency domain for processing, we can also use **ifft** to convert the processed frequency domain signal back to time domain.

It should be noted that when we use **ifft**, we need to restore the signal in the frequency domain to what it looked like when we just finished **fft**. That is, the frequency range covered by the signal is 0~Fs, not the unilateral spectrum 0~Fs/2 or the bilateral spectrum -Fs/2~Fs/2.

Observe the following example, when we draw bilateral spectrum we use Yshift, when we do ifft we use X=fft(x);

```
clear; clf;
N = 256;
Fs = 50;
df = Fs/N;
t = (0:N-1)/Fs;
x = 1*cos(2*pi*10.*t)+2*sin(2*pi*15.*t)+3*cos(2*pi*20.*t);
X = fft(x);

% when we draw bilateral spectrum we use Yshift
Yshift = fftshift(fft(x));  % to get bilateral spectrum
f = (-N/2:N/2-1)*df;
subplot(1,2,1);plot(f,abs(Yshift)/N);
xlabel("f(Hz)");ylabel("Amp");title("Amp-Freq");grid on

% when we do ifft, we use X
x1 = ifft(fftshift(Yshift));    % reshape the signal before ifft
subplot(1,2,2);plot(t,x1,t,x,'r--');xlim([0 1]);
xlabel("t(s)");ylabel("Amp");
title("frquency domain to time domain");
```

Direct Fourier Transform

In addition to do the Fourier transform with function **fft**, sometimes we can also use matrix operations to do the Fourier transform directly. For aperiodic signal,

$$\mathrm{F}(j\omega)=\int_{-\infty}^{+\infty} f(t)e^{-j\omega t}\,dt=\lim_{\tau\to 0}\sum_{n=-\infty}^{\infty} f(n\tau)e^{-jwn\tau}\cdot\tau$$

is true, if τ is small enough. Since the signal is time-limited, set N as the border of n. After discretizing ω, we have

$$F(\omega_k)=\tau\sum_{n=-N}^{n=N} f(n\tau)e^{-j\omega_k n\tau},-\mathrm{N}\le \mathrm{n}\le \mathrm{N},-\mathrm{M}\le \mathrm{k}\le \mathrm{M}$$

Which can be expressed in the matrix as follow:

$$\mathrm{F}(\mathrm{w}_k)=\begin{bmatrix}F(w_{-M})\\ \vdots\\ F(w_0)\\ \vdots\\ F(w_M)\end{bmatrix}'=\tau*\begin{bmatrix}f(-N\tau)\cdot e^{-jw_{-M}(-N\tau)}+\cdots+f(0\cdot\tau)\cdot e^{jw_{-M}(0\cdot\tau)}+\cdots+f(N\tau)\cdot e^{-jw_{-M}(N\tau)}\\ \vdots\\ f(-N\tau)\cdot e^{-jw_0(-N\tau)}+\cdots+f(0\cdot\tau)\cdot e^{jw_0(0\cdot\tau)}+\cdots+f(N\tau)\cdot e^{-jw_0(N\tau)}\\ \vdots\\ f(-N\tau)\cdot e^{-jw_M(-N\tau)}+\cdots+f(0\cdot\tau)\cdot e^{jw_M(0\cdot\tau)}+\cdots+f(N\tau)\cdot e^{-jw_M(N\tau)}\end{bmatrix}'$$

$$=\tau*\begin{bmatrix}f(-N\tau) & \cdots & f(0\tau) & \cdots & f(N\tau)\end{bmatrix}*\begin{bmatrix}e^{-jw_{-M}(-N\tau)} & \cdots & e^{-jw_0(-N\tau)} & \cdots & e^{-jw_M(-N\tau)}\\ \vdots & & \vdots & & \vdots\\ e^{-jw_{-M}(0\bullet\tau)} & \cdots & e^{-jw_0(0\cdot\tau)} & \cdots & e^{-jw_M(0\cdot\tau)}\\ \vdots & & \vdots & & \vdots\\ e^{-jw_{-M}(N\tau)} & \cdots & e^{-jw_0(N\tau)} & \cdots & e^{-jw_M(N\tau)}\end{bmatrix}$$

$$=\tau*\begin{bmatrix}f(-N\tau) & \cdots & f(0\tau) & \cdots & f(N\tau)\end{bmatrix}*e^{-j\cdot\begin{bmatrix}-N\tau\\ \vdots\\ 0\cdot\tau\\ \vdots\\ N\tau\end{bmatrix}*\begin{bmatrix}w_{-M} & \cdots & w_0 & \cdots & w_M\end{bmatrix}}$$

Let ω be the range to be observed, then we have $\omega_k=\frac{\omega}{M}k$

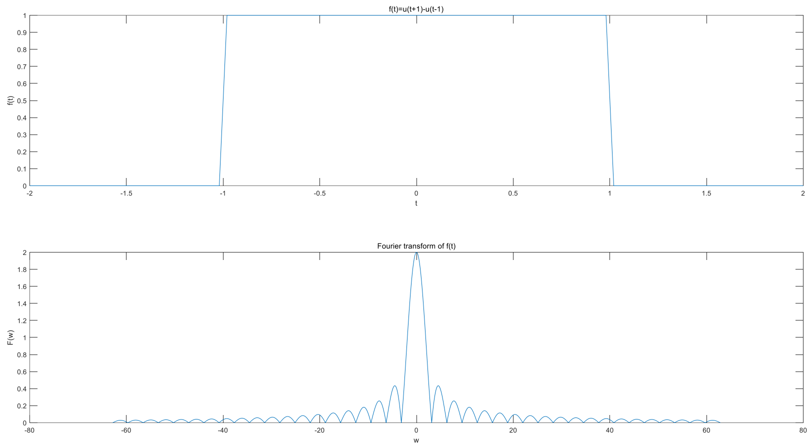
Example: Find out the Fourier transform of

$$f(t)=G(t)=\begin{cases}1,|t|\le 1\\ 0,|t|>1\end{cases}$$

```
clear; clf;
tao = 0.02;
t = -2:tao:2;
f = rectpuls(t,2);
M = 500; k=-M:M;
w=2*pi*10;     % 10 indicates the range of the frequency to observe
Wk=k*w/M;      % w1/M determines the resolution of the frequency
```

```
F = tao*f*exp(-1i*t'*Wk);
Fabs = abs(F);

subplot(2,1,1); plot(t,f);
xlabel('t(s)'); ylabel('f(t)'); title('f(t)=u(t+1)-u(t-1)');
subplot(2,1,2); plot(Wk,Fabs);
xlabel('w(rad/s)'); ylabel('F(w)'); title('Fourier transform of
f(t)');
```



For periodic signals, when using matrix operations to do the Fourier transform, the amplitude needs to be divided by the signal duration to obtain the correct value. Try to find out the amplitude-frequency diagram of signal: x=cos(2*pi*10.*t)+2sin(2*pi*15.*t)+3cos(2*pi*20.*t)).

Tips:

1. Functions used to read and play audio is listed in Table 3.

Table 3 Audio Function

format	description
[y,Fs]=audioread(FileName)	FileName:the audio to be read. y: sampled data. Fs: sample rate.
[y,Fs]=audioread(FileName,[Start,End])	Only the samples from Start to End is returned.
soundsc(y, Fs)	Play vector as sound.