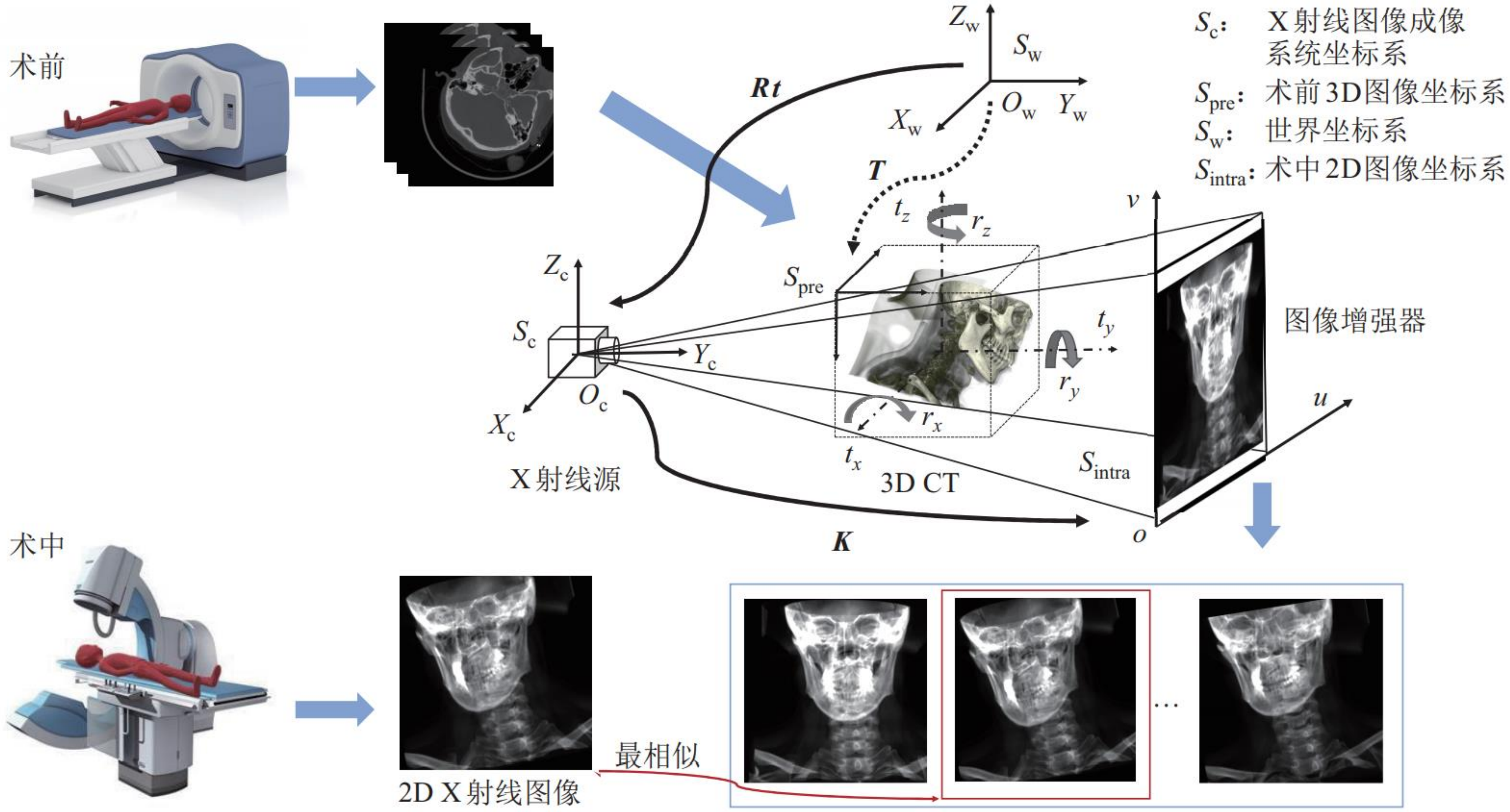
第9章　傅里叶分析

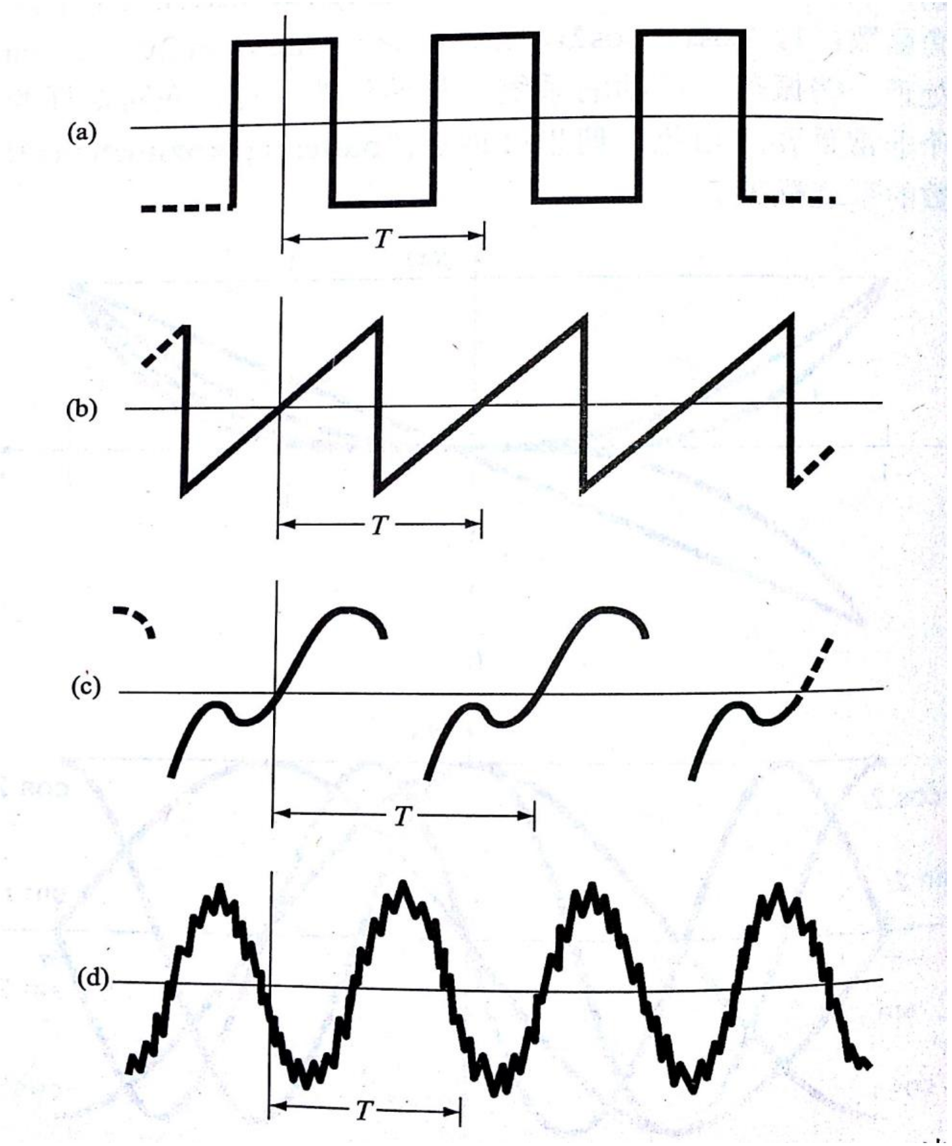
苏 芮
srhello@zju.edu.cn
开物苑4-202

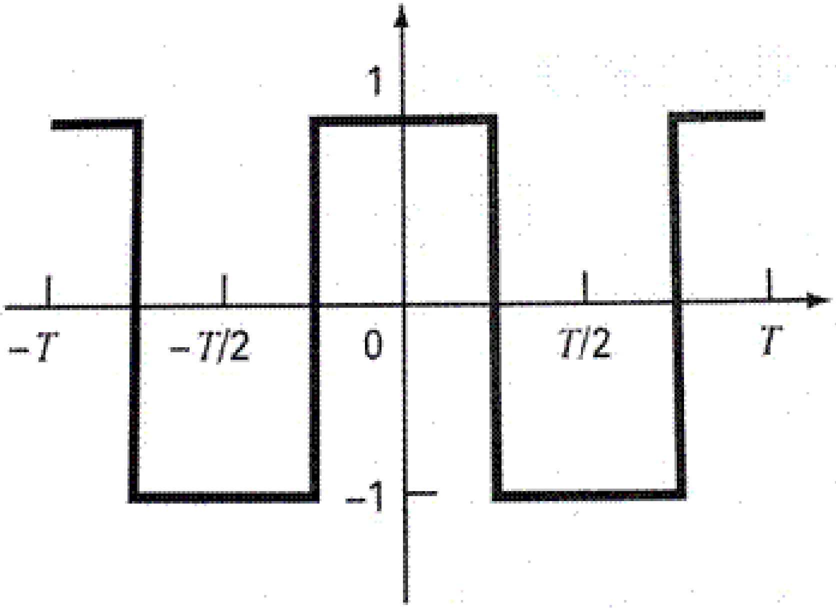


黑色实线箭头是术前已校正的. $\boldsymbol{Rt}$ 为 S_w 和 S_c 之间的刚性变换. $\boldsymbol{K}$ 为投影变换.
$\boldsymbol{T}$ 是待求的 S_w 和 S_c 之间的变换，它由 6 个参数 $r_x, r_y, r_z, t_x, t_y, t_z$ 定义. 不同的 $\boldsymbol{T}$ 可生成不同的 DRR，如图蓝色框所示.

图 1　2D/3D 图像配准的原理图及其几何设置

8.1 连续函数与傅里叶级数概念





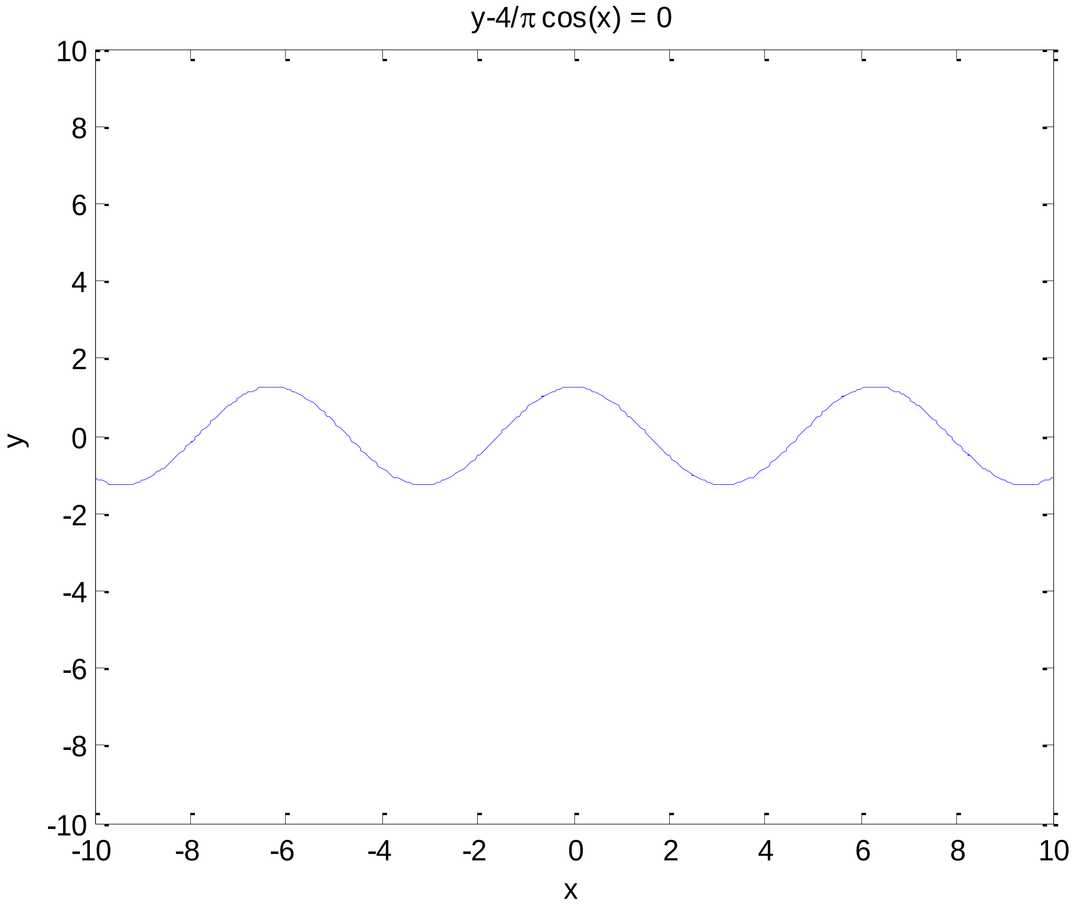
傅里叶级数：

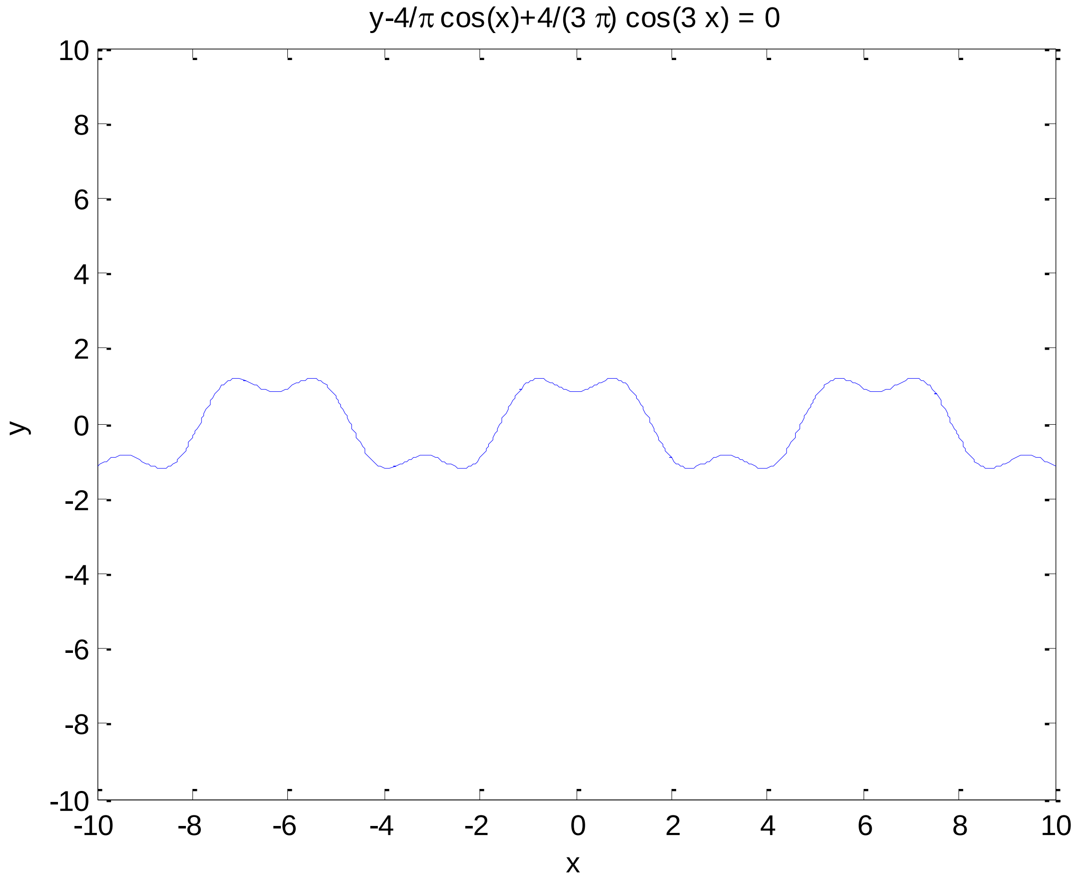
$$f(t)=\frac{4}{\pi}\cos(\omega_0 t)-\frac{4}{3\pi}\cos(3\omega_0 t)+\frac{4}{5\pi}\cos(5\omega_0 t)-\frac{4}{5\pi}\cos(5\omega_0 t)+\cdots$$

假设角速度为1

ezplot('y-4/pi*cos(x)',[-10,10])

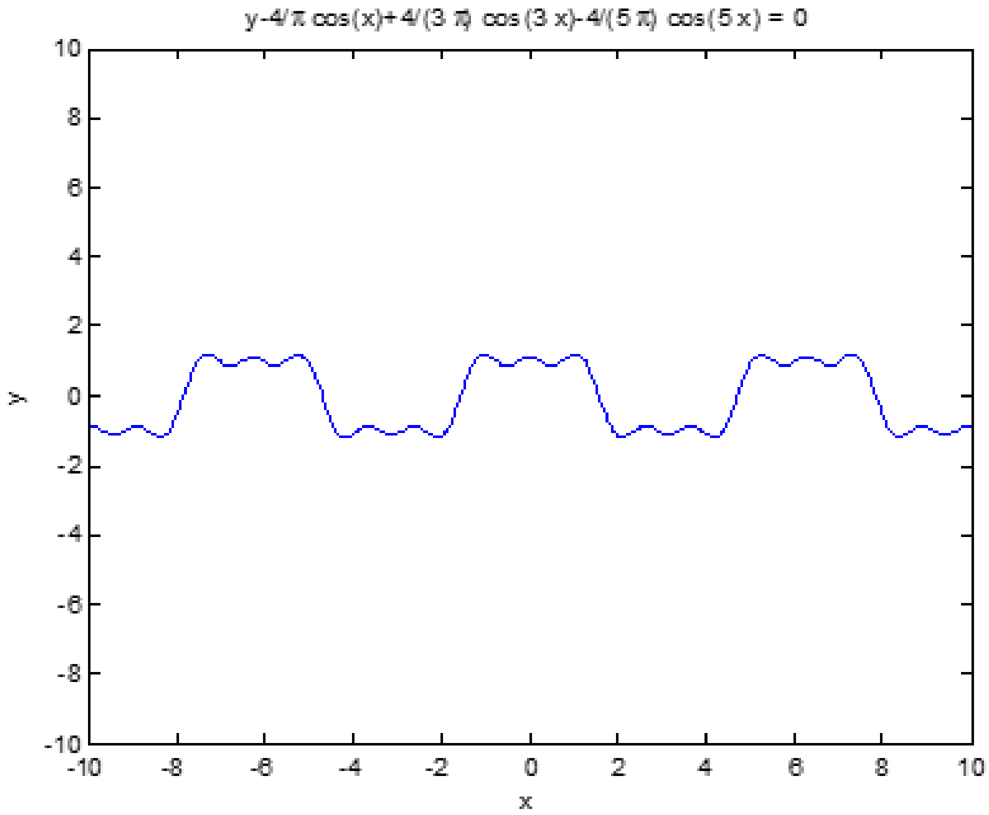
ezplot('y-4/pi*cos(x)+4/(3*pi)*cos(3*x)',[-10,10])

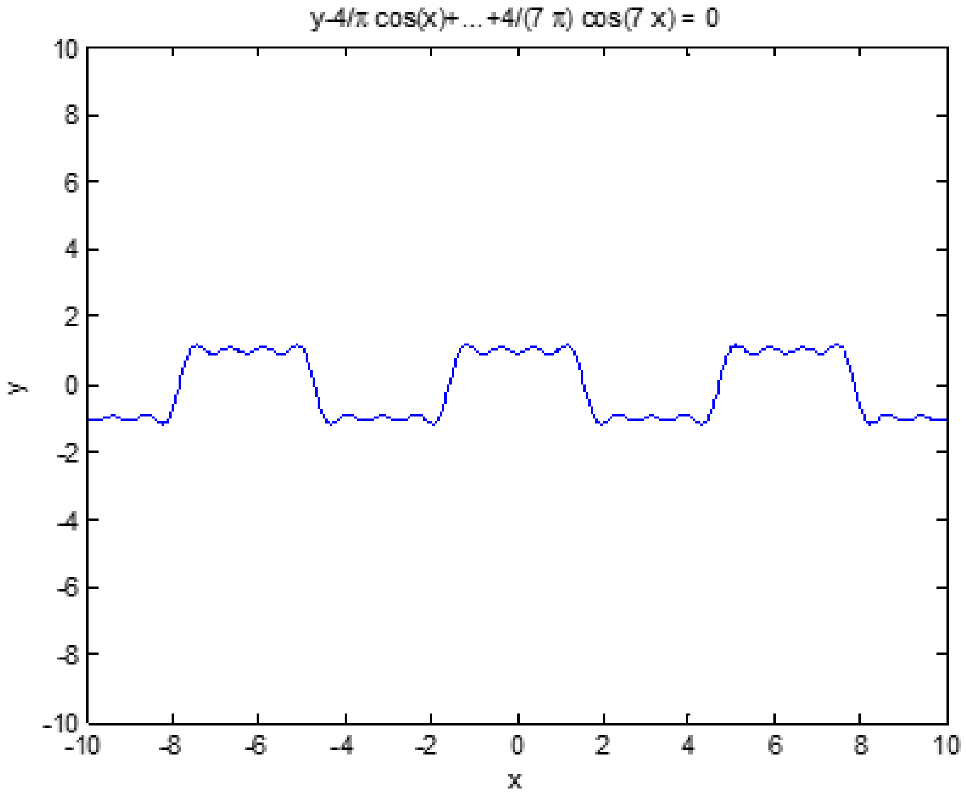




```
ezplot('y-4/pi*cos(x)+4/(3*pi)*cos(3*x)-4/(5*pi)*cos(5*x)',[-10,10])
```

```
ezplot('y-4/pi*cos(x)+4/(3*pi)*cos(3*x)-4/(5*pi)*cos(5*x)+4/(7*pi)*cos(7*x)',[-10,10])
```





8. 1 连续函数与傅里叶级数概念

#傅里叶变换的基本思想

傅里叶级数是研究周期函数频谱的有力工具，但是有许多波形并不是很有规律的重复，例如闪电只发生一次，但是它会对很宽频率范围内的无线电接收设备产生干扰，这意味着这种非重复信号具有连续频谱。要实现从周期函数到非周期函数的过渡，只要让周期函数的周期T趋近于无穷即可，也就是说T为无穷时，函数将永远不会重复，就变成了非周期函数。

8. 1 连续函数与傅里叶级数概念

傅里叶变换公式如下：

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

通过傅里叶变换，以时间 t 积分(-∞，∞)的时域函数变到了以角频率ω 。$F(\varpi)$为复函数，有振幅谱和相位谱。

$$F(\omega)=\left|F(\omega)\right|\cdot e^{j\varphi(\omega)}$$

傅里叶逆变换公式如下：

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

8. 1 连续函数与傅里叶级数概念

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

$$=\frac{1}{2\pi}\int_{-\infty}^{\infty}|F(\omega)|\cdot\cos[\omega t+\varphi(\omega)]d\omega+\frac{\mathrm{j}}{2\pi}\int_{-\infty}^{\infty}|F(\omega)|\cdot\sin[\omega t+\varphi(\omega)]d\omega$$ 如果 **f(t)**为实函数，去除虚部，则上式变成：

$$f(\mathrm{t})=\frac{1}{2\pi}\int_{-\infty}^{\infty}|F(\omega)|\cdot\cos[\omega t+\varphi(\omega)]d\omega=\frac{1}{\pi}\int_{0}^{\infty}|F(\omega)|\cdot\cos[\omega t+\varphi(\omega)]d\omega$$

这两个公式称为傅里叶变化对

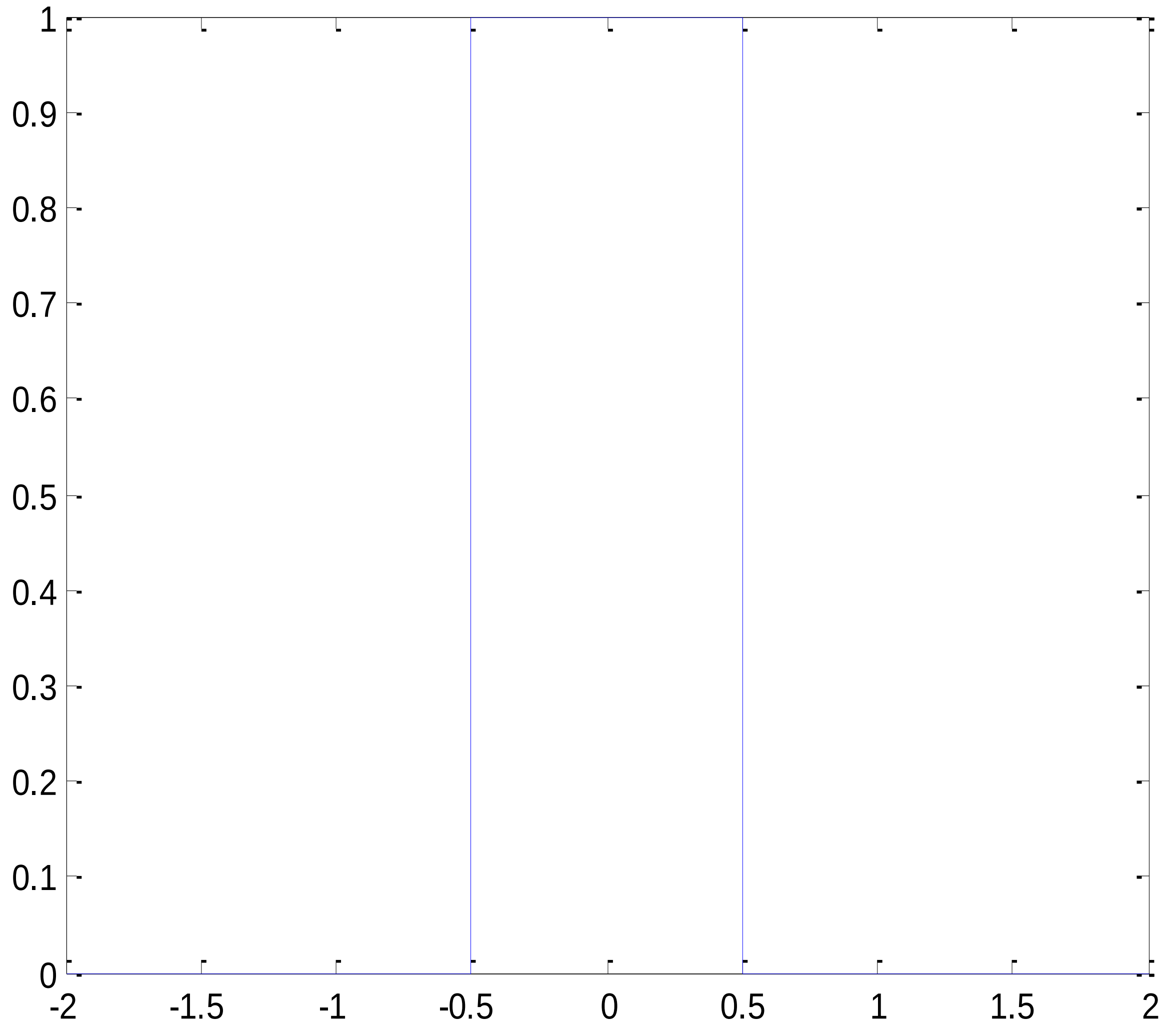
$$F(\varpi)=F[f(t)]$$

$$f(t)=F^{-1}[F(\omega)]$$

8.1 连续函数与傅里叶级数概念

举例说明一个门函数（矩形脉冲函数）的傅里叶变换：

$$g_\tau(t)=\begin{cases}1, & |t|<\dfrac{\tau}{2}\\[2mm] 0, & |t|>\dfrac{\tau}{2}\end{cases}$$



8.1 连续函数与傅里叶级数概念

进行傅里叶变换：

$$F(\omega)=\int_{-\infty}^{\infty} f(t)\cdot e^{-j\omega t}dt=\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} e^{-j\omega t}dt=\frac{1}{-j\omega}\left(e^{-j\frac{\omega\tau}{2}}-e^{\frac{\omega\tau}{2}}\right)$$

$$=\frac{2\sin\frac{\omega\tau}{2}}{\omega}=\tau\frac{\sin\frac{\omega\tau}{2}}{\frac{\omega\tau}{2}}$$

得到与角速度为横坐标，

$$F(\omega)=\int_{-\infty}^{\infty} f(t)\cdot e^{-j\omega t}dt=\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} e^{-j\omega t}dt=\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}}(\cos\omega t+i\sin\omega t)dt$$

$$=\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} \cos\omega\, tdt+\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} i\sin\omega\, tdt$$

可以看出，对于所有的频率都有

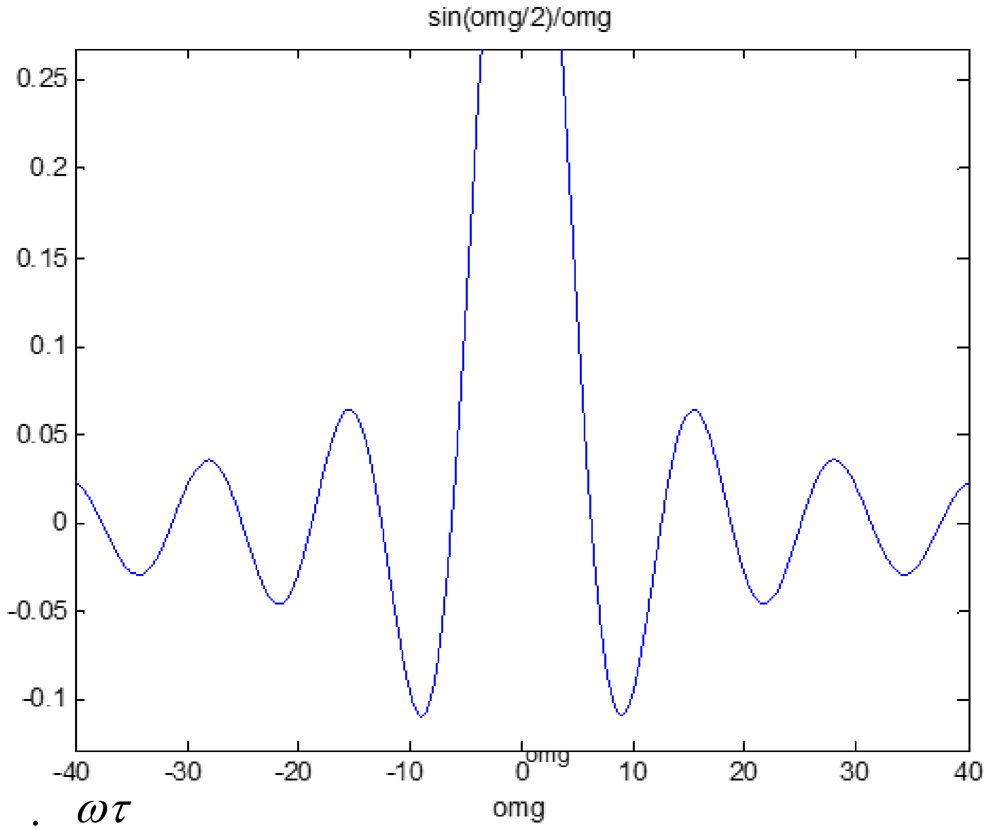
$$\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} i\sin\omega tdt=0$$

$$\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}}\cos\omega tdt=\frac{1}{\omega}\int_{-\frac{\tau}{2}}^{\frac{\tau}{2}}\cos\omega td(\omega t)=\frac{1}{\omega}(\sin\omega t)\Big|_{-\frac{\tau}{2}}^{\frac{\tau}{2}}=\frac{1}{\omega}\left(\sin\omega\frac{\tau}{2}+\sin\omega\frac{\tau}{2}\right)=\frac{2}{\omega}\sin\frac{\omega\tau}{2}$$

通过 Matlab 画出曲线：

```
syms omg;
y=sin(omg/2)/omg;
ezplot(y,-20,20);
```

（这里门函数的τ=1，可以看出最丰富的频谱在于-2π和2π间，为正值时，相位为0，为负值时，相位为π或者-π）



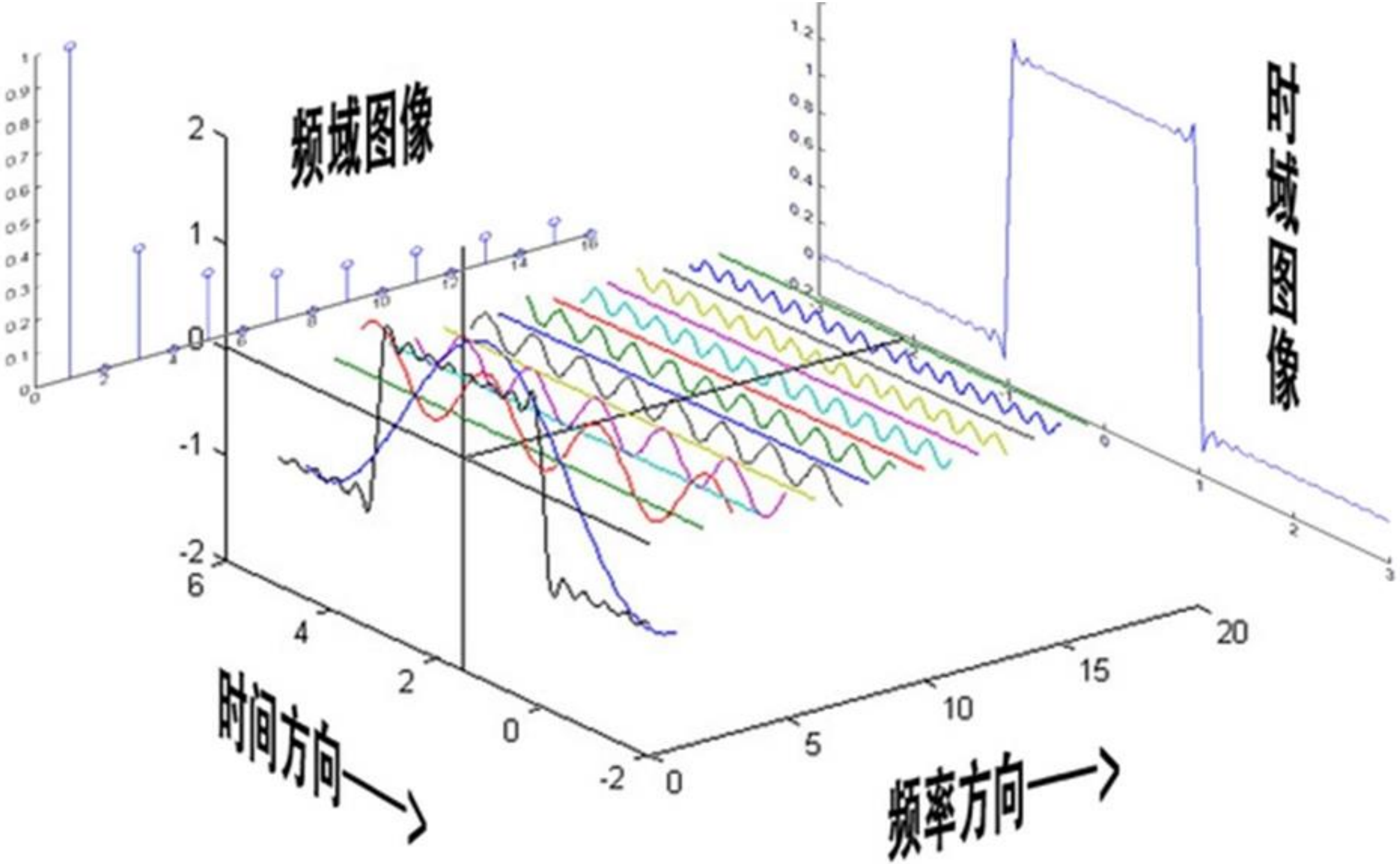
8.1 连续函数与傅里叶级数概念

需要注意的是傅里叶变换出来后是有实部和虚部的区别的，而出现的复数的原因就是存在相位差，和电学上的复数的存在是一样的，如果电流电压采用复函数表示的话，傅里叶变换也可以变换出复函数来，如果 **f(t)**是实函数，则变换出实函数来。

傅里叶变换能将满足一定条件的某个函数表示成三角函数（正弦和/或余弦函数）或者它们的积分的线性组合。在不同的研究领域，傅里叶变换具有多种不同的变体形式，如连续傅里叶变换和离散傅里叶变换。

傅里叶变换的实质是将一个信号分离为无穷多多正弦/复指数信号的加成，也就是说，把信号变成正弦信号相加的形式——既然是无穷多个信号相加，那对于非周期信号来说，每个信号的加权有密度上的差别，傅里叶变换之后，横坐标即为分离出的正弦信号的频率，纵坐标对应的是加权密度。对于周期信号来说，因为确实可以提取出某些频率的正弦波成分，所以其加权不为零——在幅度谱上，表现为无限大—但这些无限大显然是有区别的，所以我们用冲激函数表示

8.1 连续函数与傅里叶级数概念



8.1 连续函数与傅里叶级数概念

#MATLAB的Sound命令

SOUND Play vector as sound.

SOUND(Y,FS) sends the signal in vector Y (with sample frequency FS) out to the speaker on platforms that support sound. Values in Y are assumed to be in the range -1.0 <= y <= 1.0. Values outside that range are clipped. Stereo sounds are played, on platforms that support it, when Y is an N-by-2 matrix.

SOUND(Y) plays the sound at the default sample rate of 8192 Hz.

SOUND(Y,FS,BITS) plays the sound using BITS bits/sample if possible. Most platforms support BITS=8 or 16.

Example:

```
load handel
sound(y,Fs)
```

当我们将采样频率Fs设定为Fs=4000,可以发现声音低沉了许多。

```
Fs=4000
sound(y,Fs)
```

当将采样频率Fs设定为Fs=10000,音调提高了许多，女声

```
Fs=10000
sound(y,Fs)
```

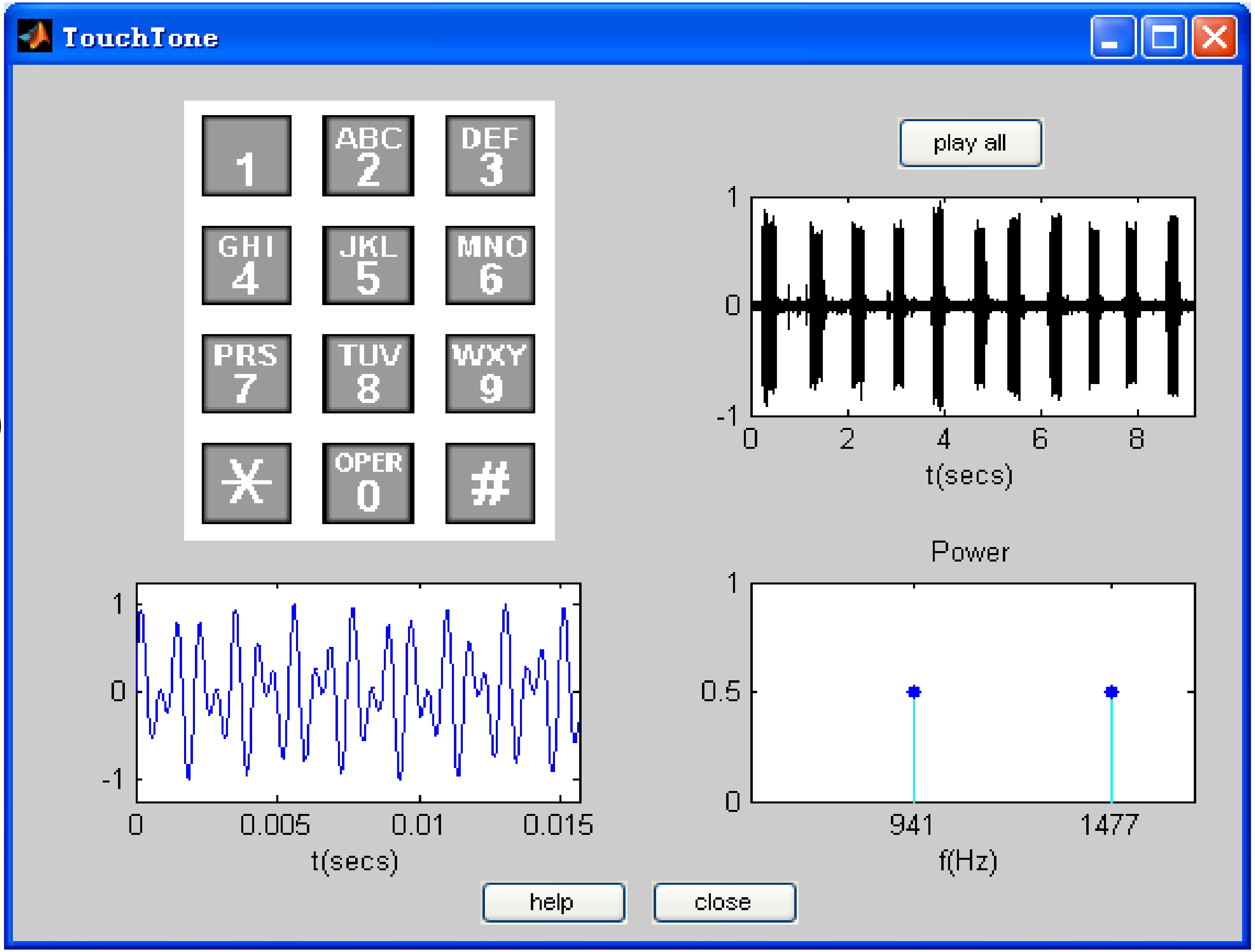
当将采样频率Fs设定为Fs=12000,音调提高了许多，童声

```
Fs=13000
sound(y,Fs)
```

8. 1 连续函数与傅里叶级数概念

#电话频率的分析

```
 % DTMF, Dual tone multi-frequencies, Hz
Fr=[697 770  852 941];
Fs=[1209 1336 1477];
 % Time (seconds)
   Fs = 32768;
   t = 0:1/Fs:0.25;
  % Plot component frequencies
 ax = get(gcf,'userdata');
   set(gcf,'currentaxes',ax.power)
   plot([f;f],[0*p;p],'c-',f,p,'b.','markersize',16)
   axis([500 1700 0 1])
   set(gca,'xtick',[fr(k) fc(j)])
   xlabel('f(Hz)')
   title('Power')
  cp = get(gca,'currentpoint');
   k = min(max(ceil(cp(1,2)/50),1),4);
   j = min(max(ceil(cp(1,1)/50),1),3);
   f = [fr(k) fc(j)];
   p = [1/2 1/2];
 % Superimpose two sinusoidal tones
   y1 = sin(2*pi*fr(k)*t);
   y2 = sin(2*pi*fc(j)*t);
   y = (y1 + y2)/2;
```



8.2离散信号的傅里叶变换

傅里叶变换是把各种形式的信号用正弦信号表示，非正弦信号进行傅里叶变换，会得到与原信号频率不同的成分——都是原信号频率的整数倍。这些高频信号是用来修饰频率与原信号相同的正弦信号，使之趋近于原信号的。所以说，频谱上频率最低的一个峰（往往是幅度上最高的），就是原信号频率。傅里叶变换把信号由时域转为频域，因此把不同频率的信号在时域上拼接起来进行傅里叶变换是没有意义的—实际情况下，我们隔一段时间采集一次信号进行变换，才能体现出信号在频域上随时间的变化。

另外，拉普拉斯变换的公式是：

$$F(s)=\int_0^{\infty} f(t)\cdot e^{-st}dt$$

#离散信号的傅里叶变换

$$Y_k=\sum_{j=0}^{n-1}\omega^{jk}y_j$$

ω是一个单位1的n次单位根：

$$\omega=e^{-2\pi i/n}$$

傅里叶变换可以表示成矩阵向量乘：

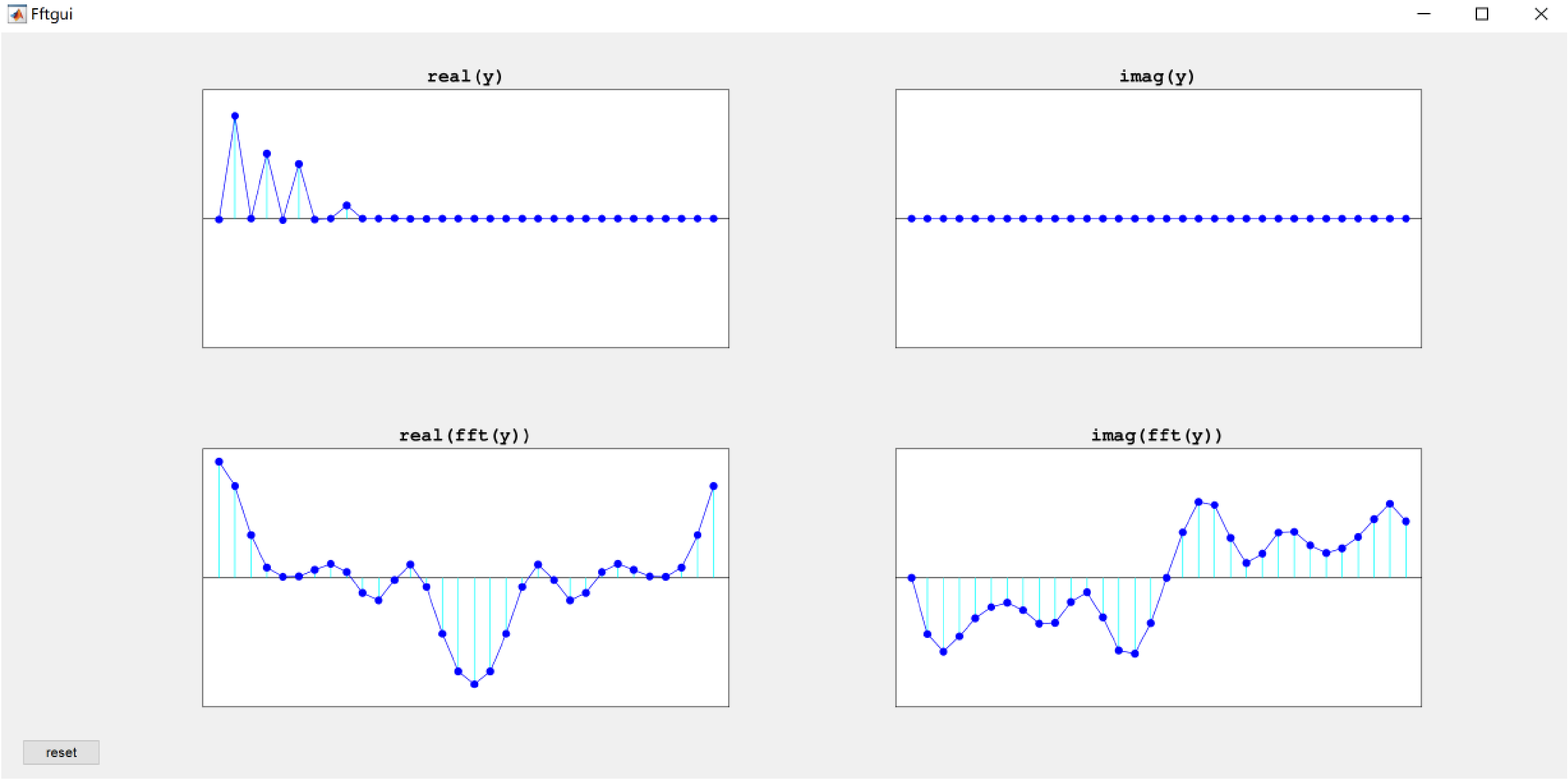
$$Y=Fy$$

矩阵F中的各元素为：

$$f_{k,j}=\omega^{jk}$$

8.3 fftgui

#fftgui-Matlab的图形化命令
fftgui是直接对于离散的采样数据数组y进行傅里叶分析，y数组默认的横坐标就是时间，分析得到四个图
real(y)：这是y数组的实数分量，如果f(t)本身就是实数，那就是y数组本身点的描绘
imag(y)：这是y数组的虚数分量，如果f(t)本身是实数，那就是虚数分量都为0
real(fft(y):y数组经过傅里叶变换后的实数分量
imag(fft(y)):y数组经过傅里叶变换后的虚数分量

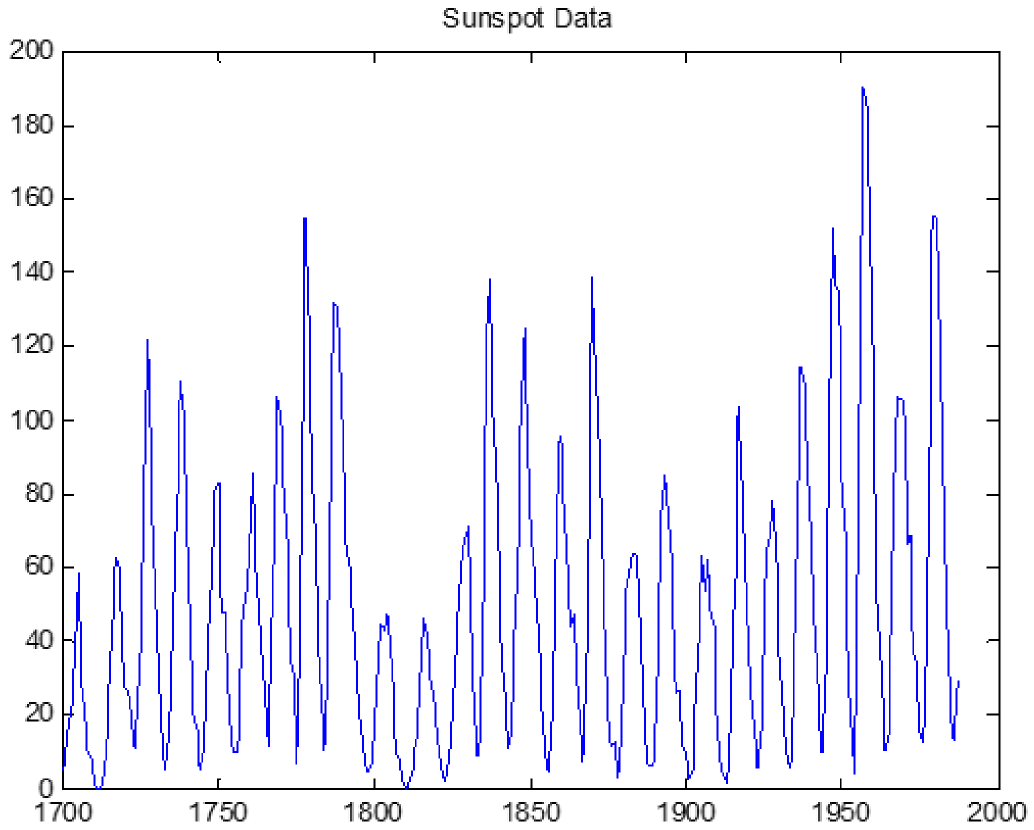


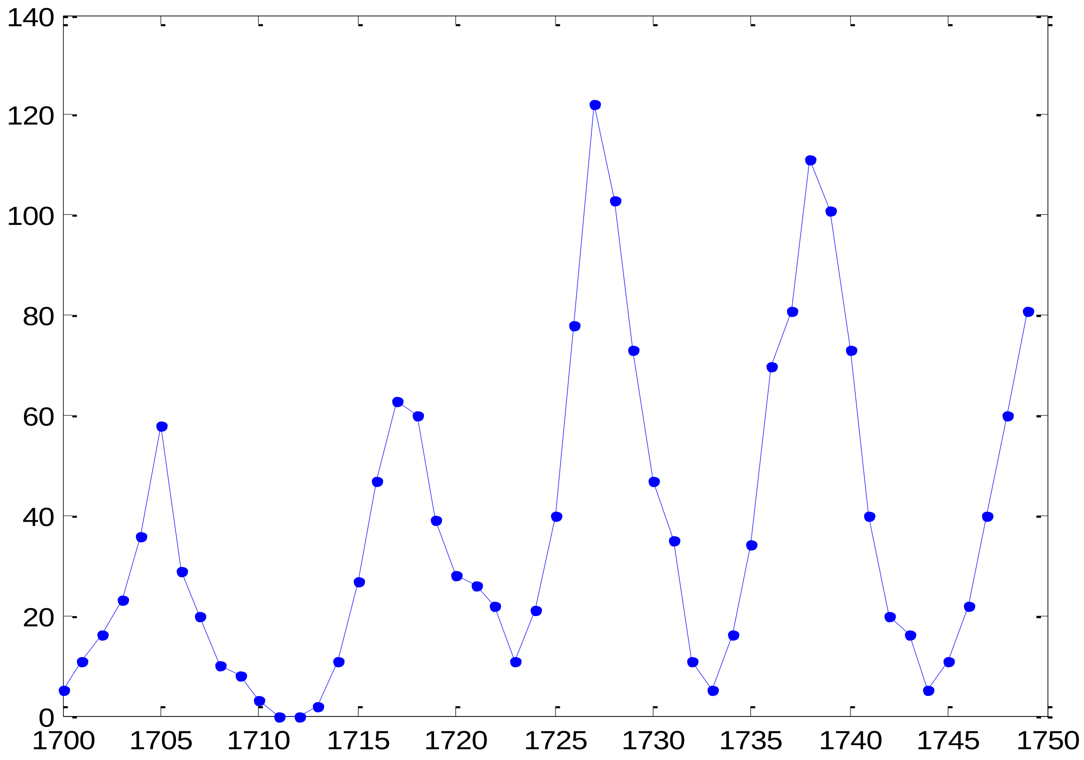
8.4 太阳黑子

```
#太阳黑子的分析程序
load sunspot.dat
year=sunspot(:,1);
relNums=sunspot(:,2);
plot(year,relNums)
title('Sunspot Data')
```

这是每年的太阳黑子数，采样频率是每年。可以放大了看：

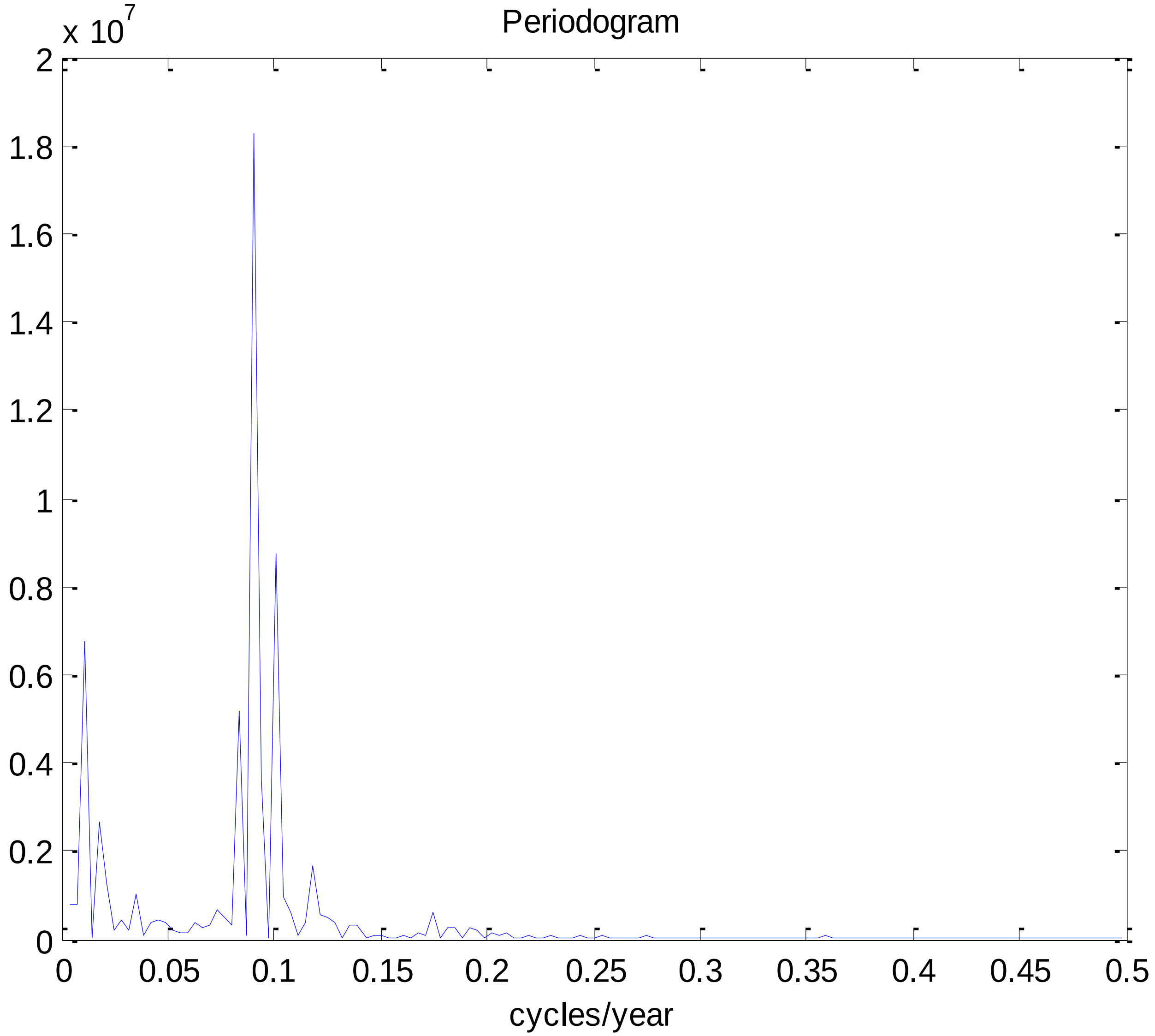
```
plot(year(1:50),relNums(1:50),'b.-');
```





8. 4 太阳黑子

```
Y = fft(relNums);
Y(1)=[];
plot(Y,'ro')
title('Fourier Coefficients in the Complex Plane');
xlabel('Real Axis');
ylabel('Imaginary Axis');
n=length(Y);
power = abs(Y(1:floor(n/2))).^2;
nyquist = 1/2;
freq = (1:n/2)/(n/2)*nyquist;
plot(freq,power)
xlabel('cycles/year')
title('Periodogram')
```



可以看出，其频率小于**0.1**，大约**0.09**，周期即为**1/0.09=11**年

8.5 周期时间序列

由按键式电话电话产生的音调和 Wolfer 的太阳黑子索引是两个周期时间序列的例子。时间函数显示周期特性，至少相似。傅里叶分析使我们能够后计数值离散集的周期，该离散集以固定频率采集数据。下表显示了在这个分析里用到的各量之间的关系。

`y`	数据
`Fs`	采样点/单位时间
`n = length(y)`	采样点数目
`t = (0:n-1)/Fs`	总时间
`dt = 1/Fs`	时间增量
`Y = fft(y)`	有限傅里叶转换
`abs(Y)`	FFT值振幅
`abs(Y).^2`	幂
`f = (0:n-1)*(Fs/n)`	频率、周期/单位时间
`(n/2)*(Fs/n) = Fs/2`	奈奎斯特频率
`p = 1./f`	周期，单位时间/周期

周期图是 FFT 的振幅 abs(Y)或幂 abs(Y)，^2 与频率 f 的关系曲线图。只需画出一半，因为另一半图是前半图关于奈奎斯特频率的映象。

8.6 快速离散傅里叶变换

百万的一维的FFT和1000x1000的二维的傅里叶变换都十分常见，矩阵会大得惊人。计算量非常大。

快速傅里叶变换算法的关键是，单位1的$2n$次单位根的平方等于单位1的n次单位根，用复数表示：

$$\omega = \omega_n = e^{-2\pi i/n}$$

我们有

$$\omega_{2n}^2 = \omega_n$$

快速算法的推导始于离散傅里叶变换的定义：

$$Y_k = \sum_{j=0}^{n-1} \omega^{jk} y_j, \ k = 0, \cdots, n-1$$

设想n是偶数，而且$k \leqslant n/2-1$。将其按下标的奇偶性分开求和。

$$Y_k = \sum_{even\ j} \omega^{jk} y_j + \sum_{odd\ j} \omega^{jk} y_j = \sum_{j=0}^{n/2-1} \omega^{2jk} y_{2j} + \omega^k \sum_{j=0}^{n/2-1} \omega^{2jk} y_{2j+1}$$

方程右边的两个和是长度为$n/2$的FFT的分量，分别为y的奇、偶下标。为了得到长度为n的整个傅里叶变换，我们需要做两次长度为$n/2$的傅里叶变换，将其中一个结果乘上ω的幂，再将两部分结果相加，得到最终结果。

8.6 快速离散傅里叶变换

现在，如果 n 不仅是偶数，而且是 2 的幂，这个过程可以重复。一个长度为 n 的快速傅里叶变换可以表示为 2 个长度为 $n/2$ 的快速傅里叶变换，4 个长度为 $n/4$ 的快速傅里叶变换，接着 8 个长度为 $n/8$ 的快速傅里叶变换，然后类推直至变换为 n 个长度为 1 的傅里叶变换。长度为 1 的傅里叶变换就是它本身。如果 $n=2^p$，这个递归步数为 p 步。每一步时间复杂度为 $O(n)$，所以总的时间复杂度为

$$O(np)=O(n\ \log_2 n)$$

如果 n 不是 2 的幂，仍然可以将长度为 n 的傅里叶变换表示为几个长度短一些的傅里叶变换。一个长度为 100 的傅里叶变换可以写成两个长度为 50 的傅里叶变换，4 个长度为 25 的傅里叶变换。一个长度为 25 的傅里叶变换可以表示为 5 个长度为 5 的傅里叶变换。如果 n 不是一个素数，长度为 n 的傅里叶变换，可以表示为若干个其长度能被 n 整除的傅里叶变换。即使 n 是一个素数，傅里叶变换也可以嵌入到另一个其长度可以被因式分解的傅里叶变换中。在这里，我们不详细介绍这些算法。

8.7 ffttx

```
function y = ffttx(x)
%FFTTX  Textbook Fast Finite Fourier Transform.
%     FFTTX(X) computes the same finite Fourier transform as FFT(X).
%     The code uses a recursive divide and conquer algorithm for
%     even order and matrix-vector multiplication for odd order.
%     If length(X) is m*p where m is odd and p is a power of 2, the
%     computational complexity of this approach is O(m^2)*O(p*log2(p)).
%    Copyright 2014 Cleve Moler
%    Copyright 2014 The MathWorks, Inc.
x = x(:);
n = length(x);
omega = exp(-2*pi*i/n);
if rem(n,2) == 0
   % Recursive divide and conquer
   k = (0:n/2-1)';
   w = omega .^ k;
   u = ffttx(x(1:2:n-1));
   v = w.*ffttx(x(2:2:n));
   y = [u+v; u-v];
else
   % The Fourier matrix.
   j = 0:n-1;
   k = j';
   F = omega .^ (k*j);
   y = F*x;
end
```

8.8 傅里叶矩阵

Plot(fft(eye(n,n)))

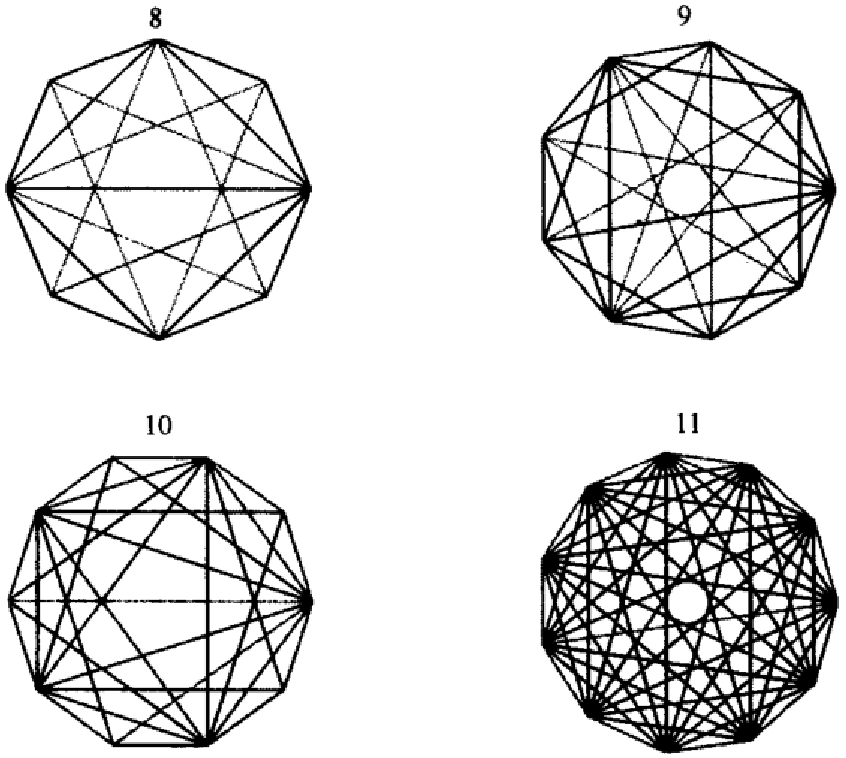


图 8-14　傅里叶矩阵图

连接 F 矩阵的每一列元素，产生一个 n 个点的子图。如果 n 是素数，则连接所有列的元素，会生成关于 n 个点的完全图。如果 n 不是素数，这个图的稀疏性与快速傅里叶变换有关。图 8-14 显示了 $n = 8, 9, 10$ 和 11 的图。因为 $n = 11$ 是素数，相应的图连接了所有点。但是其他 3 个值不是素数。图中的一些连接被省略了，这说明具有多个点的向量的离散傅里叶变换是可以使用快速算法的。

感谢聆听,欢迎讨论！