# 上机实习\_4

## 任务一

1. 两个离散信号x(n)=n(-100<=n<150)，y(n) =n-10（-200<=n<100），绘制出信号

代码：

X = np.arange(-100, 151, 1)  
Y = np.arange(-200, 101, 1)  
Xn = X  
Yn = Y - 10  
plt.scatter(X,Xn,s=10)  
plt.xlim(-20, 20)  
plt.ylim(-20, 20)  
plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.title(**u'x(x)'**)  
plt.show()  
  
  
plt.scatter(Y,Yn,s=10)  
plt.xlim(-20, 20)  
plt.ylim(-30, 20)  
plt.title(**u'y(n)'**)  
plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.show()

运行结果：

|  |  |
| --- | --- |
| 图 1 x(n)=n (-100<=n<150) | 图 2 y(n) =n-10 (-200<=n<100) |

1. 用MATLAB的褶积程序计算x(n)\* y(n)和y(n)\* x(n) 的结果，并绘制出褶积结果；

代码：

Xc = np.arange(0, 551, 1)  
Cn = scipy.signal.convolve(Xn, Yn)  
plt.scatter(Xc,Cn)  
plt.title(**u'x(x)\*y(n)'**)  
plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.show()

运行结果：

|  |  |
| --- | --- |
| 图 3 x(n)\*y(n) | 图 4 y(n)\*x(n) |

1. 分别用MATLAB的相关程序计算两个信号的互相关函数r12和r21，并绘出互相关函数

代码：

n = np.arange(0, 251)  
fs = 1000   
t = n/fs   
acf1 = np.correlate(Xn, Xn, mode=**'full'**)  
N = len(Xn)  
acf1 = acf1[N-1:]  
acf1 = acf1 / np.arange(N, 0, -1)  
acf1 = acf1 / acf1[0]  
plt.plot(t, acf1)  
plt.title(**u'自相关函数 x(n)'**)  
plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.show()

运行结果：

|  |  |
| --- | --- |
| 图 5 互相关函数r12 | 图 6 互相关函数r21 |

1. 分析比较计算的结果

分析结果：

离散褶积具有可交换性，交换顺序后结果未发生改变。

互相关运算一般不服从交换律，互相关与褶积意义不同，但互相关可以用卷机表示

## 二、 任务二

1. 设计一个周期方波波信号，长度为6个周期

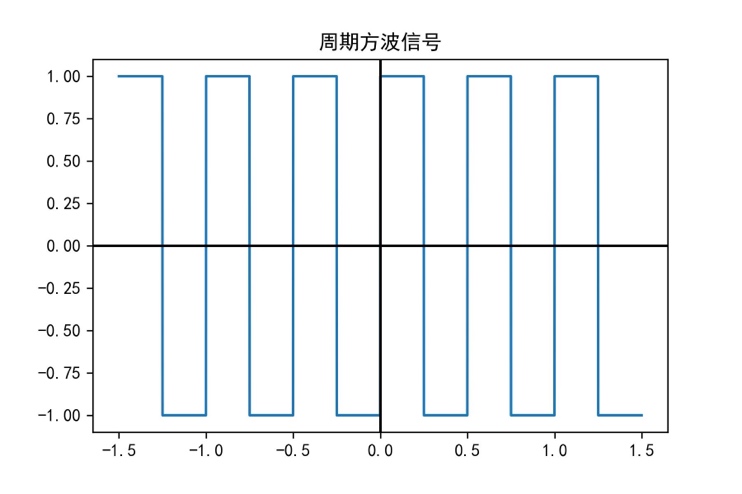


图 7 周期方波信号

1. 加入一定比例的随机噪声n(t0)，为x(t)

代码：

n = np.arange(-1500, 1500)  
fs = 1000   
f0 = 10  
t = n/fs   
sq = signal.square(2 \* np.pi \* 2 \* t)   
xt = sq + 0.01\*np.random.randn(3000)*#xt为添加过噪声的函数*plt.plot(t, xt)  
plt.title(**u'周期方波信号+噪声 x(t)'**)  
plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.show()

运行结果：

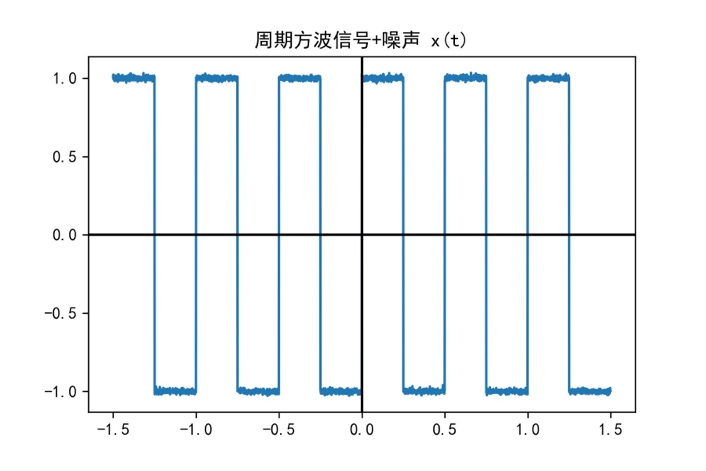


图 8 加入噪声 x(t)

1. 计算信号的自相关函数rxx

代码：

acf1 = np.correlate(xt, xt, mode=**'full'**)  
N = 3000  
acf1 = acf1[N-1:]  
acf1 = acf1 / np.arange(N, 0, -1)  
acf1 = acf1 / acf1[0]  
plt.plot(t, acf1)  
plt.title(**u'自相关函数'**)  
plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.show()

运行结果：

|  |
| --- |
|  |
| 图 9 自相关函数 |
|  |

1. 分别绘出信号及其自相关函数
2. 分析比较计算的结果

连续性相关函数为偶函数。

周期函数的自相关函数是具有与原函数相同周期的函数。

## 三、任务三

1. 设计一个[-100ms,+100ms]的子波

代码：

n = np.arange(-1, 1, 0.001)  
fs = 10  
f0 = 10  
t = n/fs   
st = np.exp(-(np.pi \* f0 \* t)\*\*2) \* (1 - 2 \* (np.pi \* f0 \* t)\*\*2)   
plt.plot(t, st)  
plt.title(**u's(t)'**)  
plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.xlim(-0.12,0.3)  
plt.ylim(-0.5,1.2)  
plt.show()

运行结果：

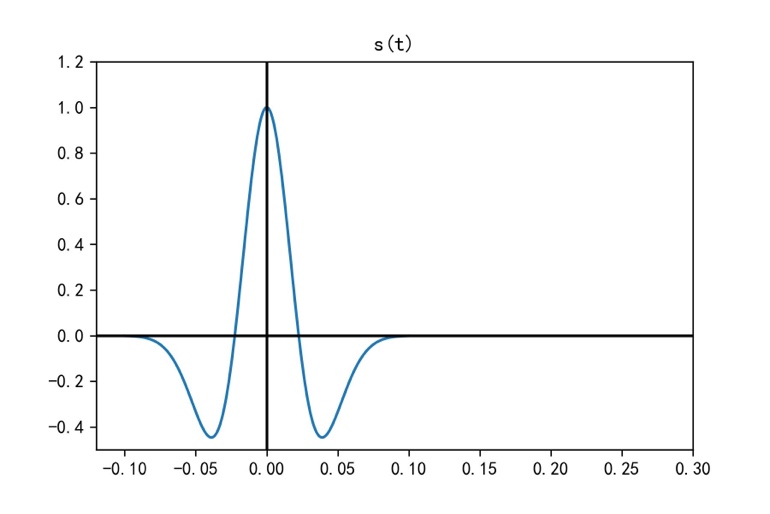


图 10 s(t)

1. 将信号s(t)向右时移t0>100ms,为s(t-t0)

代码：  
st0 = st  
tx = np.arange(0.5, 2.5, 0.001)/10  
plt.plot(tx, st0)  
plt.title(**u'时移 s(t-t0)'**)  
plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.xlim(-0.12,0.3)  
plt.ylim(-0.5,1.2)  
plt.show()

运行结果：

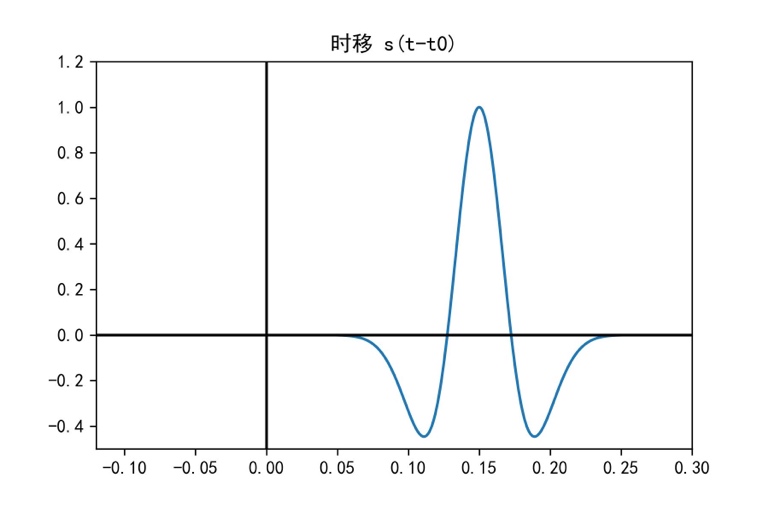


图 11 s(t-t0)

1. 给信号s(t-t0)加入一定比例的随机噪声n(t0)，为x(t)

代码：

xt = st0 + 0.02\*np.random.randn(len(tx))  
plt.plot(tx, xt)  
plt.title(**u'加噪声 x(t)'**)  
plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.xlim(-0.12,0.3)  
plt.ylim(-0.5,1.2)  
plt.show()

运行结果：

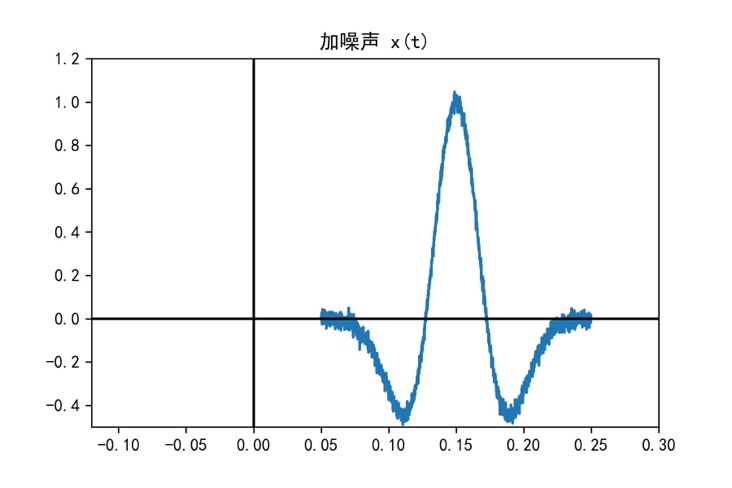


图 12 x(t)

1. 计算信号x(t)和s(t)的互相关函数rxs

代码：

acf1 = np.correlate(xt, st, mode=**'full'**)  
N = 2000  
acf1 = acf1[N-1:]  
acf1 = acf1 / np.arange(N, 0, -1)  
acf1 = acf1 / acf1[0]  
plt.plot(t, acf1)  
plt.title(**u'x(t) s(t) 互相关函数'**)  
*# plt.xlim(-0.6,0.4)*plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.show()  
*#x(t) s(t)*

运行结果：

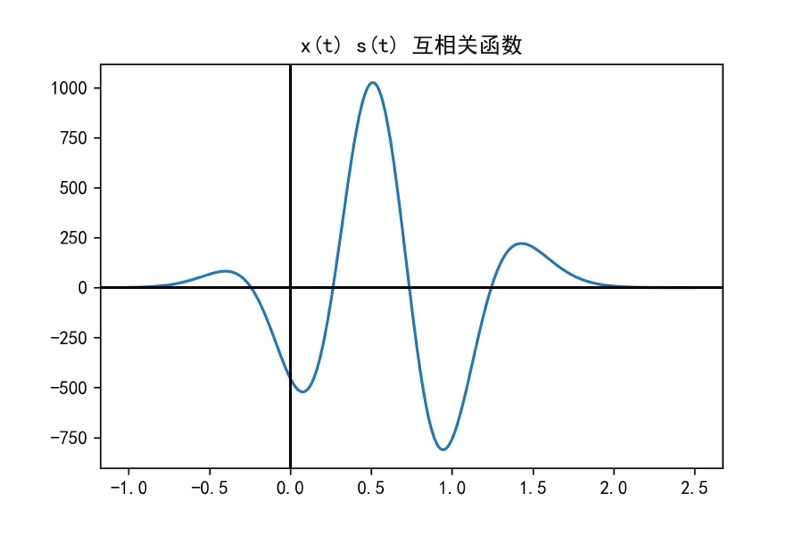


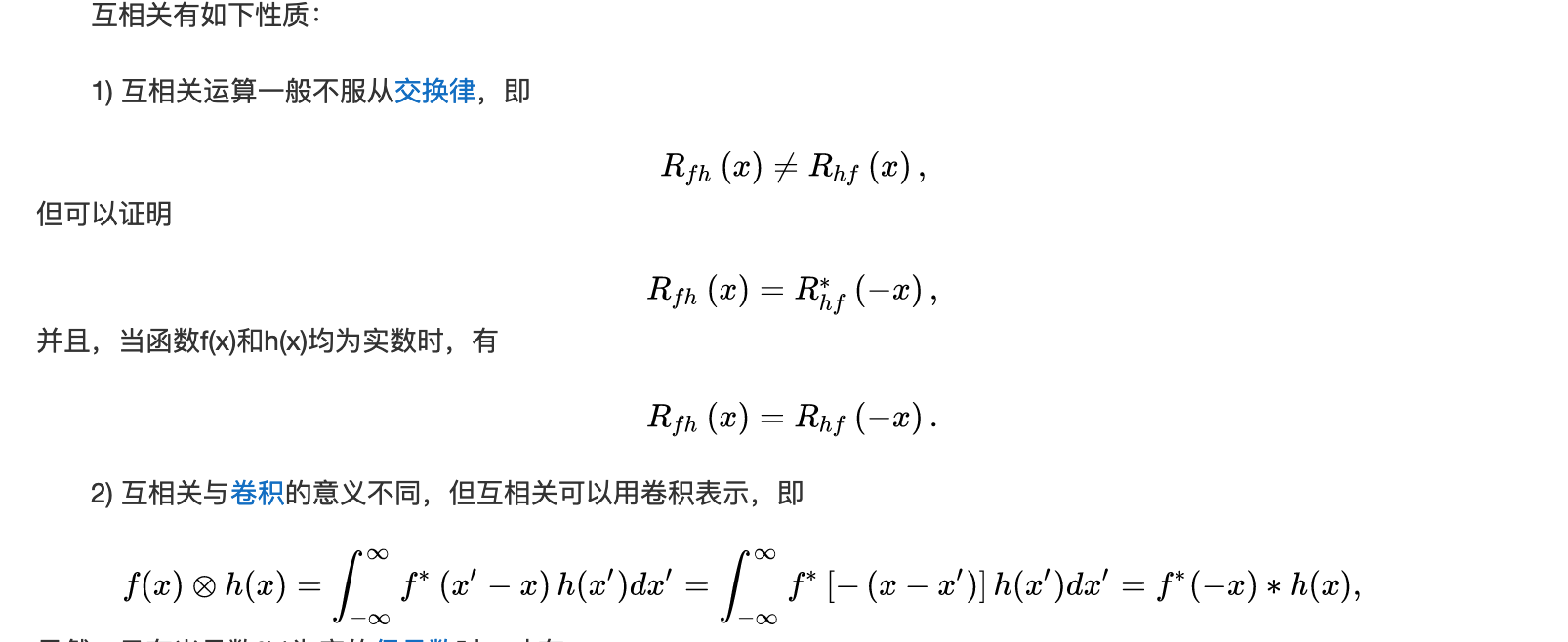
图 13 rxs

1. 分别绘出各个信号及其互相关函数

运行结果：

|  |  |
| --- | --- |
| 图 14 x(t) s(t) | 图 15 s(t) x(t) |
| 图 16 s(t-t0) x(t) | 图 17 x(t) s(t-t0) |
| 图 18 s(t) s(t-t0) | 图 19 s(t-t0) s(t) |

1. 分析比较计算结果



## 四、任务四

1. 学生用MATLAB（或其他语言）编写程序实现离散傅里叶变换DFT(最好是子程序）

代码：

*# DTF变换***for** k **in** range(N):  
 **for** n **in** range(150):  
 X[k] = X[k] + x[n]\*np.exp(-1j\*n\*k/N\*2\*np.pi)

1. 用编写的DFT程序计算序列x(n)的频谱X(k)，其中，x(n)=6n2+30(0≤n<150)，绘出信号和频谱图

代码：

N = 150  
x = [6\*i\*\*2+30 **for** i **in** range(150)]  
X = np.zeros(N) + np.zeros(N)\*1j *# 频域频谱*  
   
fig = plt.figure(figsize=(6, 6))  
n1 = np.arange(0, 150, 1)  
plt.plot(n1, x)  
plt.title(**u'x(n)'**)  
plt.axvline(x=0, color=**'k'**)  
plt.axhline(y=0, color=**'k'**)  
plt.show()

运行结果：

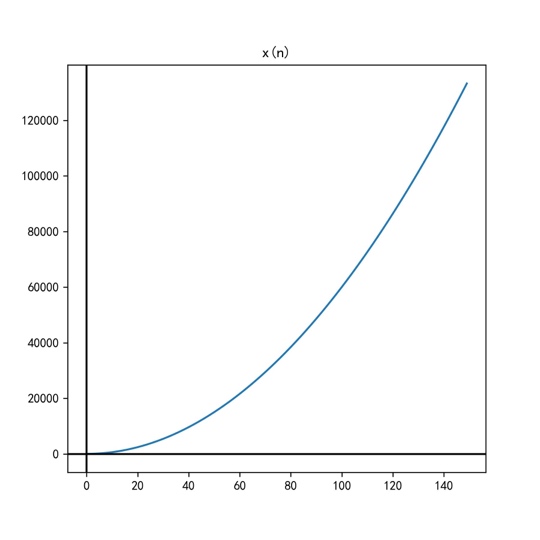


图 20 x(n)信号

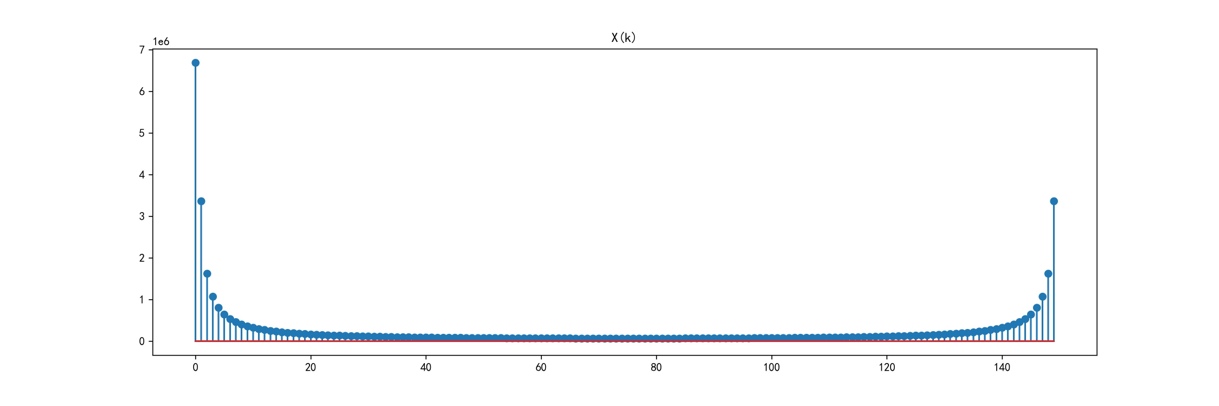


图 21 x(k)频谱

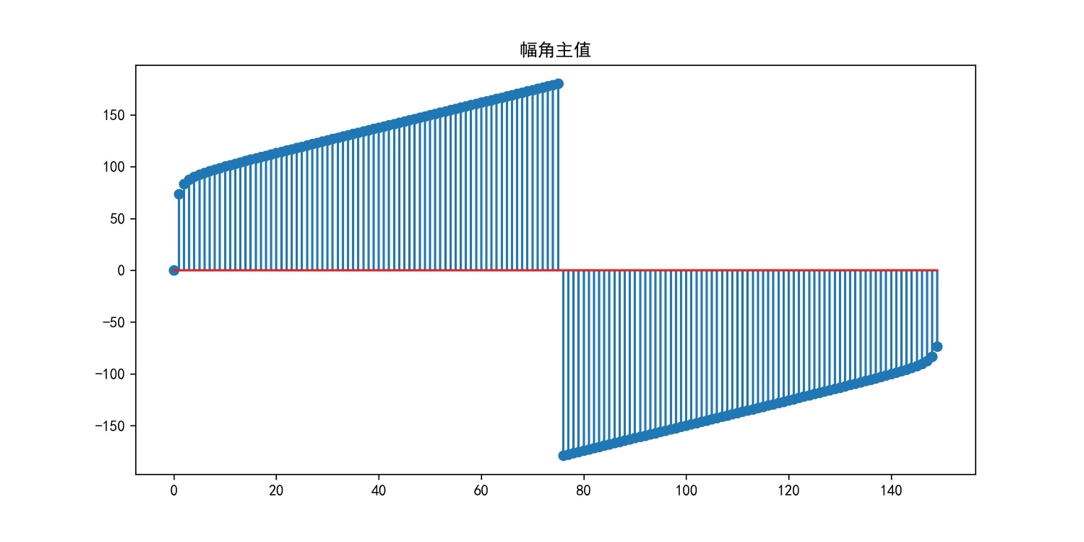


图 22 幅角主值

1. 学生用MATLAB（或其他语言）编写程序实现离散傅里叶逆变换IDFT(最好是子程序）

代码：

*# IDFT*x\_p = np.zeros(len(x), dtype=complex)  
**for** n **in** range(len(x)):  
 **for** k **in** range(N):  
 x\_p[n] = x\_p[n] + 1/N\*X[k]\*np.exp(1j\*n\*k/N\*2\*np.pi)

1. 利用DFT计算的频谱X(k)，用IDFT程序计算x(n)，绘制出相应结果

运行结果：

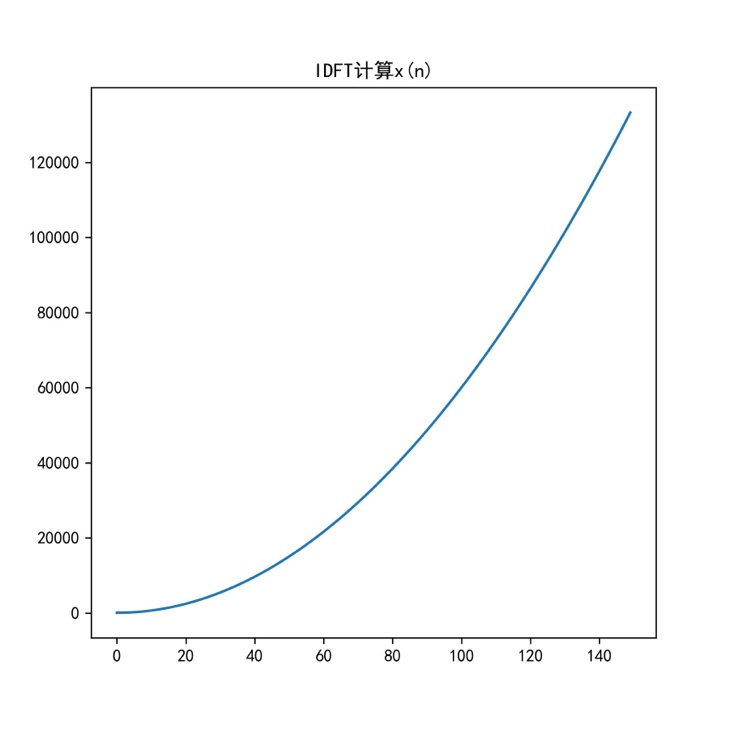


图 23 IDFT计算x(n)

1. 用MATLAB中自带的FFT命令验证编写程序的正确性

代码：

*# 用python自带的FFT验证***from** numpy **import** arange, sin, pi, cos  
**import** matplotlib.pyplot **as** plt  
**import** cmath  
**from** scipy.fftpack **import** fft,ifft  
yy = fft(x)  
yf = abs(yy)   
fig = plt.figure(figsize=(15, 5))  
plt.stem(n1, yf)  
plt.title(**u'FFT验证'**)  
plt.show()

运行结果：

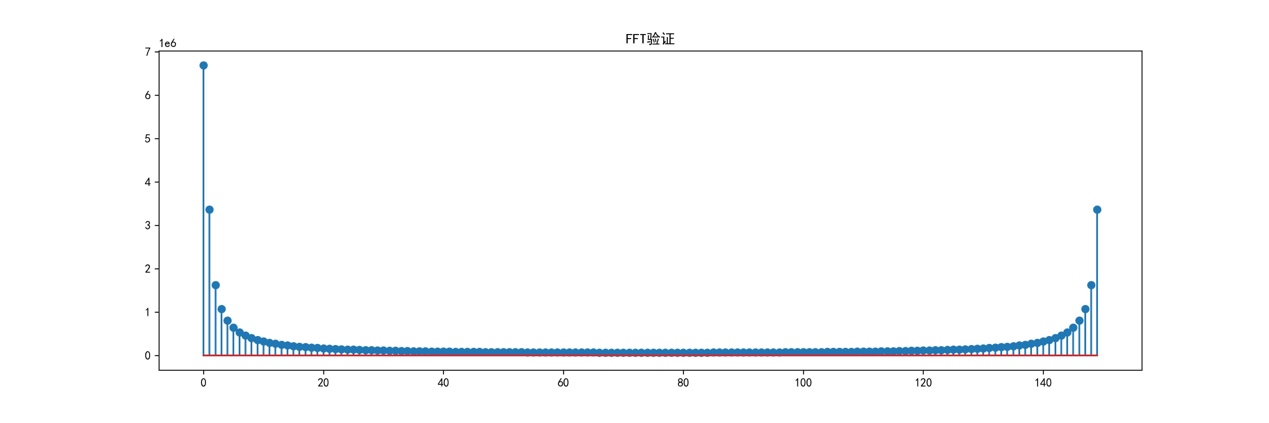


图 24 FFT验证

1. 分析结果

FFT验证下的幅角主值图与x(n)绘制出的幅角主值图像一致，即验证正确。