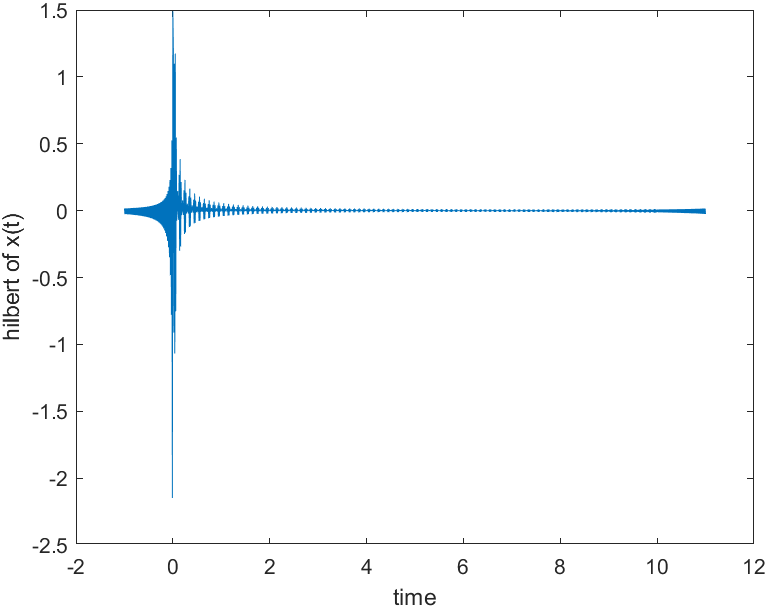
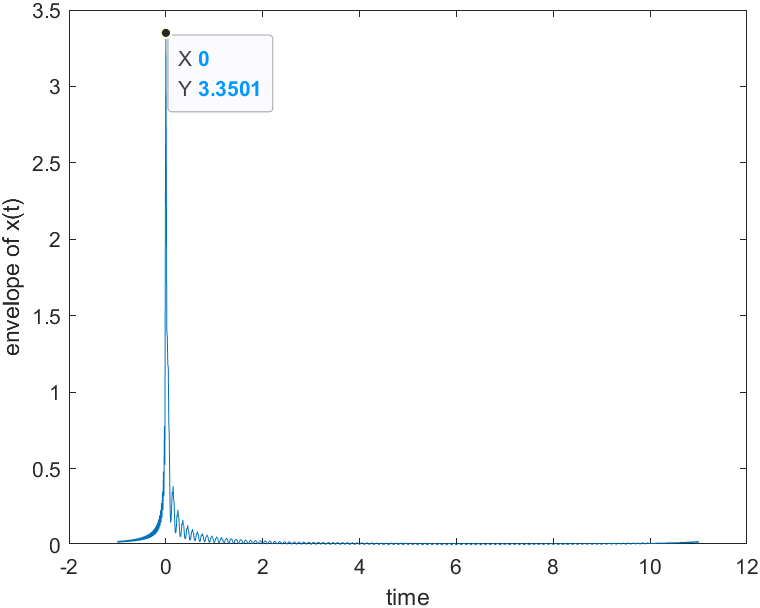
1. **(a)**

對原始訊號x(t)做matlab内建的hilbert(x)，就可以得到x(t)的analytic signal，接著取analytic signal的虛部，就是原始訊號x(t)的hilbert transform，結果輸出圖如下：



**(b)**

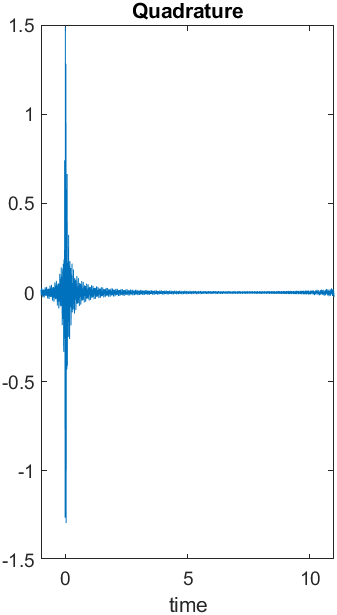
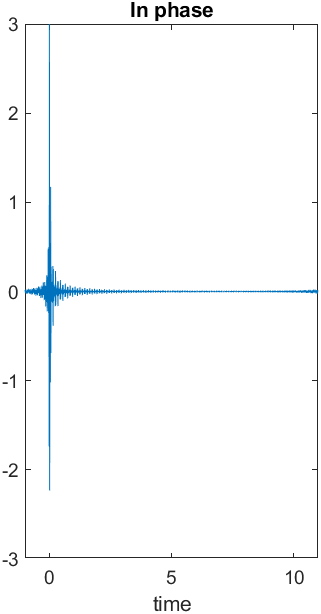
繼上一題得到analytic signal後，取其絕對值便可得到原始訊號x(t)的envelope，結果輸出圖如下：



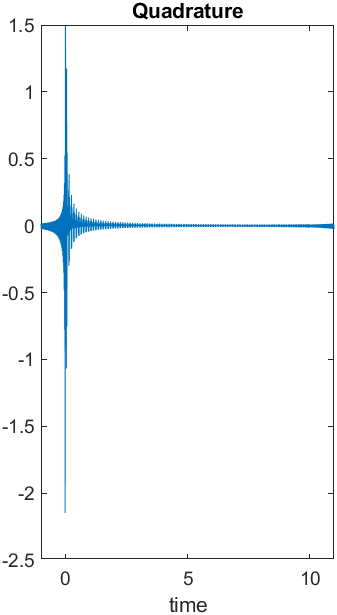
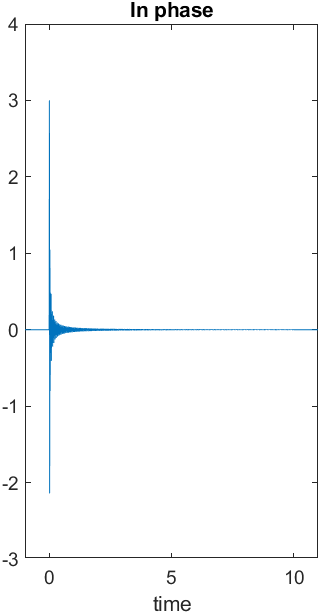
觀察原始訊號x(t)的peak，以及hilbert transfer的peak，取其平方相加再開根號，發現計算結果的envelope peak值與輸出結果一致。

**(c)**

接著我們將analytic signal與 相乘得到lowpass equivalent signal，然後分別取其實部與虛部，就可以得到lowpass equivalent signal的in-phase component 和 quadrature component了，輸出圖如下：



與原始訊號x(t)的in-phase component 和 quadrature component比較：



程式碼如下：

clear**;** clc**;** close all**;**

fs **=** 100**;**

ts **=** 1**/**fs**;**

t **=** **-**1**:**ts**:**11**;**

x **=** **(**2**\***sinc**(**20**\***t**).\***cos**(**2**\***pi**\***50**\***t**)** **+** sinc**(**20**\***t**).^**2.**\***cos**(**2**\***pi**\***130**\***t**)).\*(**t**>=**0**).\*(**t**<=**10**);**

z **=** hilbert**(**x**);**

figure

plot**(**t**,**real**(**z**));** xlabel**(**'time'**);** ylabel**(**'hilbert of x(t)'**);**

% (a)

figure

plot**(**t**,**imag**(**z**));** xlabel**(**'time'**);** ylabel**(**'hilbert of x(t)'**);**

% (b)

figure

plot**(**t**,**abs**(**z**));** xlabel**(**'time'**);** ylabel**(**'envelope of x(t)'**);**

% (c)

f0 **=** 85**;**

t2 **=** 0**:**ts**:(**length**(**x**)-**1**)\***ts**;**

x\_low **=** z**.\***exp**(-**1i**\***2**\***pi**\***f0**\***t2**);**

figure**;**

subplot**(**1**,**2**,**1**);** plot**(**t**,**real**(**x\_low**));** title**(**'In phase'**);** xlabel**(**'time'**);** % In phase component

subplot**(**1**,**2**,**2**);** plot**(**t**,**imag**(**x\_low**));** title**(**'Quadrature'**);** xlabel**(**'time'**);** % Quadrature component

figure**;**

subplot**(**1**,**2**,**1**);** plot**(**t**,**real**(**z**));** title**(**'In phase'**);** xlabel**(**'time'**);** % In phase component

subplot**(**1**,**2**,**2**);** plot**(**t**,**imag**(**z**));** title**(**'Quadrature'**);** xlabel**(**'time'**);** % Quadrature component

1. **(a)**

將數量定爲2\*107個，並且將histogram bar的數量定為500根，使用rand函數產生2\*107 by 1的matrix並用histogram函數將random distribution畫出來。

程式碼如下：

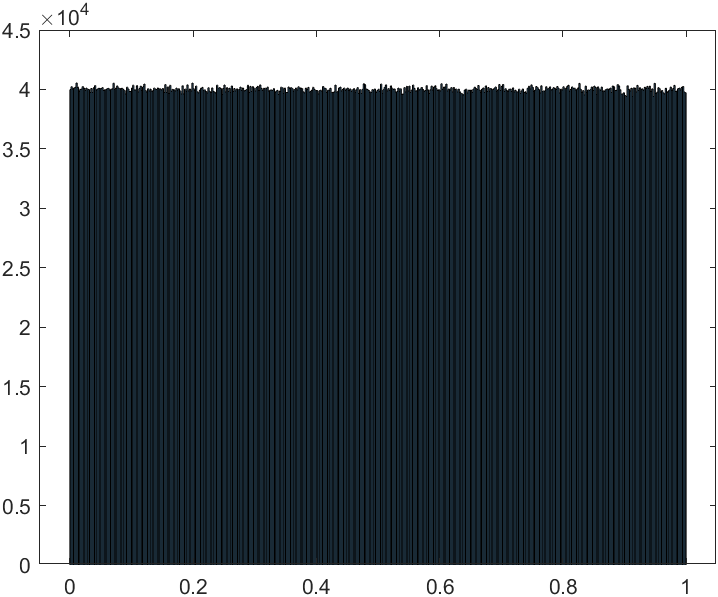
n **=** 2e7**;**

u **=** rand**(**n**,**1**);**

figure**;**

histogram**(**u**,**500**);**

histogram如下：



**(b)**

Pareto distribution的CDF為, 可以反推得:

其中U是uniform random variable，將xm = 2.5, α=1.25代入，並用histogram函數將Pareto distribution畫出來。

程式碼如下：

x\_m **=** 2.5**;**

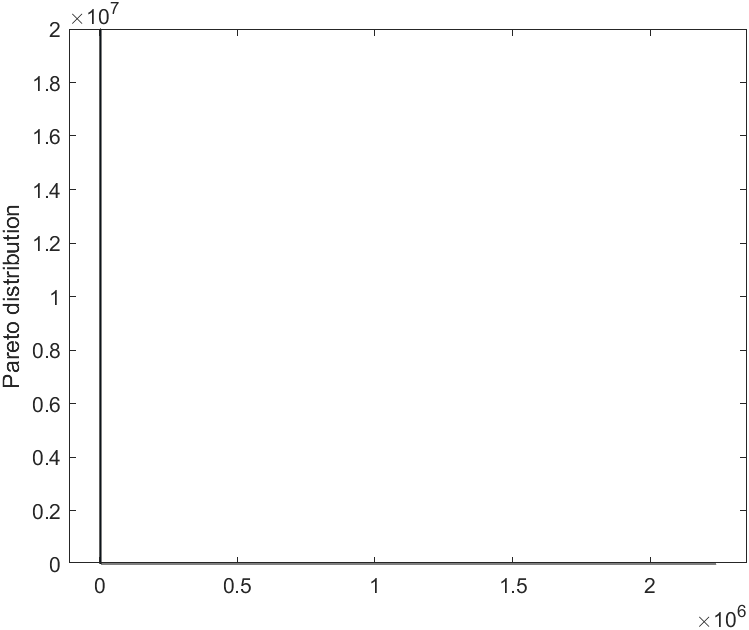
alpha **=** 1.25**;**

pareto **=** x\_m**./(**1**-**u**).^(**1**/**alpha**);**

figure**;**

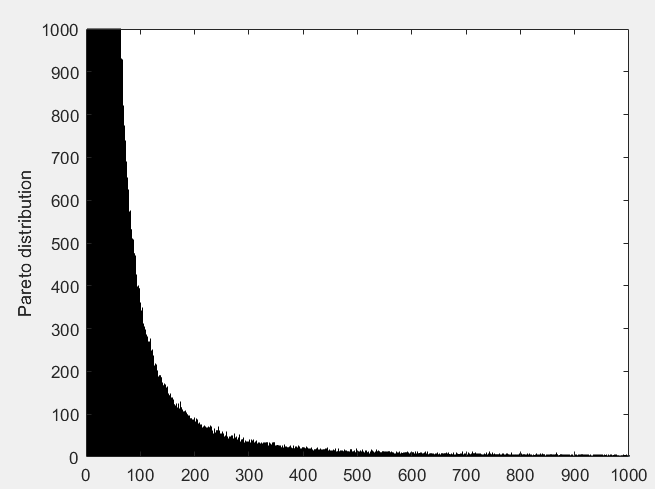
histogram**(**pareto**,**500**);**

histogram如下：

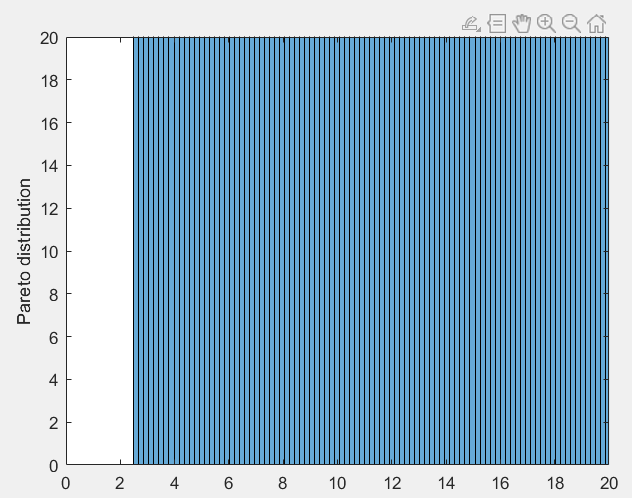


觀察發現幾乎所有值都落在0的那條bar上，推測是因爲數量太大了，並且xm也在同一支bar裏面。

如果將每一個outcome都輸出成一條bar，將範圍縮小，則較可以看見輪廓：



將範圍進一步縮小，可以發現小於xm就沒有分佈了，與預期相同。



**(c)**

先用randn函數得到兩個normal random variables x1，x2，接著算x1/ x2，程式碼如下：

x\_1 **=** randn**(**n**,**1**);**

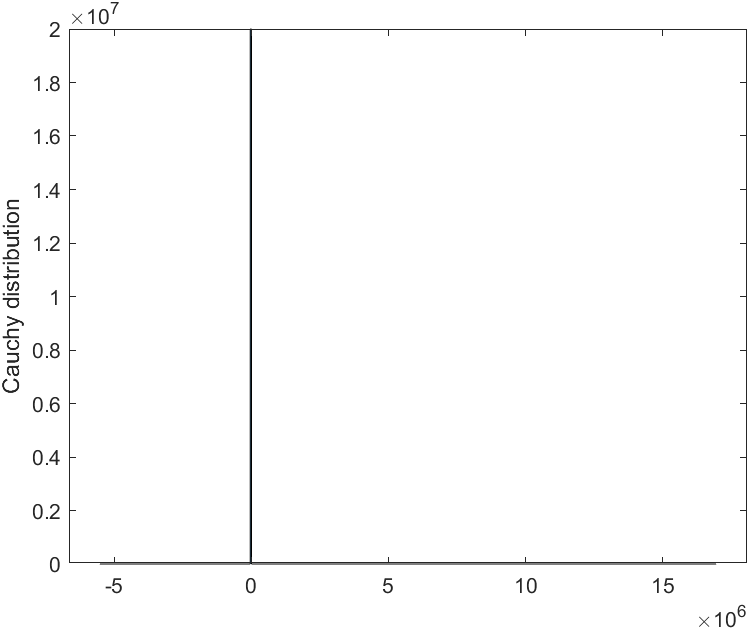
x\_2 **=** randn**(**n**,**1**);**

cauchy **=** x\_1**./**x\_2**;**

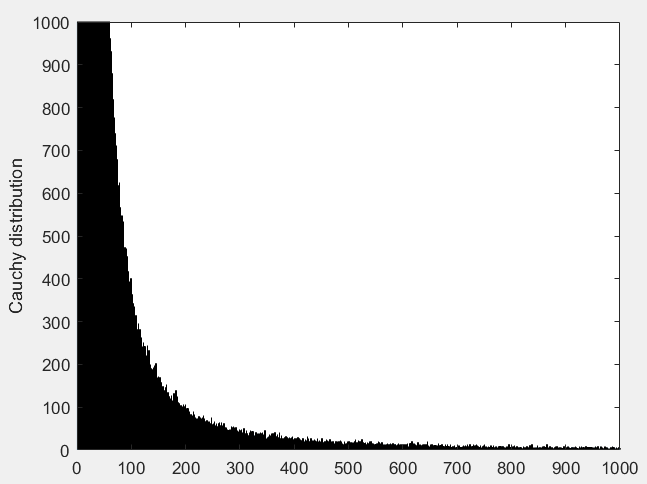
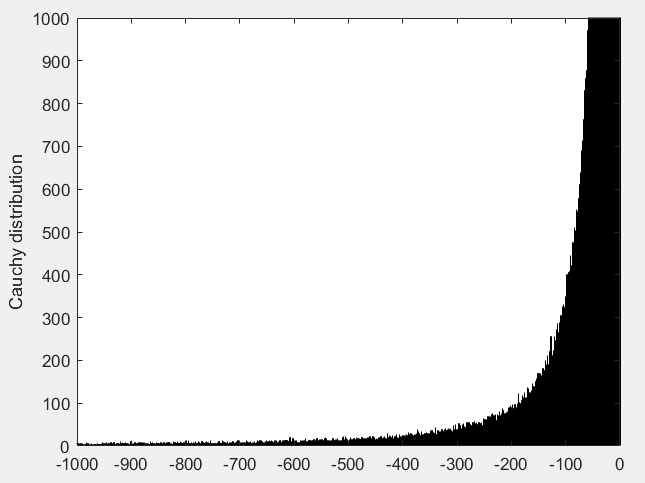
figure**;**

histogram**(**cauchy**,**500**);** ylabel**(**'Cauchy distribution'**)**

histogram如下：



可以發現還是不易辨識，因此和上一題一樣將其放大：



這樣就比較符合預期。

**(d)**

同樣利用randn函數得到10 by 2的matrix，接著自己内積自己得Y=X12+X22+……+X102，以下是程式碼：

k **=** randn**(**n**,**10**);**

chi **=** zeros**(**n**,**1**);**

**for** i**=**1**:**length**(**k**)**

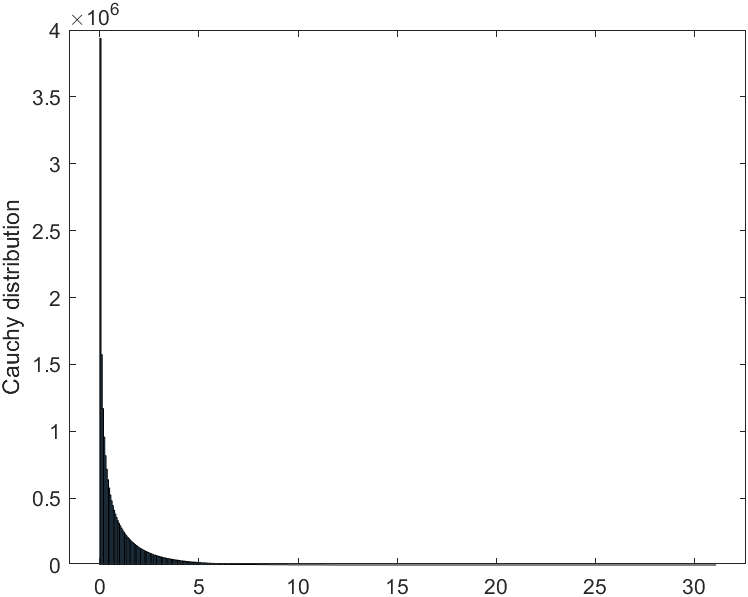
chi**(**i**)=**k**(**i**)\***k**(**i**)';**

**end**

figure**;**

histogram**(**chi**,**500**);** ylabel**(**'Cauchy distribution'**)**

接著輸出histogram：



輸出的結果基本與預期相同。

Histogram的橫軸是這個distribution的值，而縱軸是對應出現的數量，Histogram就是想要表達一組數值分佈的狀況；而bar的數量既不能太多也不能太少，前者會造成運算及輸出時間拉的較長，後者則會讓“解析度”太差，看不到更細緻的分佈。

既然是數值分佈的狀況，那將其除以數量的summation，也就是將其normalization，就得到這組數值的機率質量函數PMF，當bar足夠多，也能將其想像成PDF，將PMF與數值相乘並相加，就可以得到這組數據的expectation，因此histogram能大略看出random variable的統計分布。