THE CONNECTION BETWEEN FOURIER TRANSFORM & FOURIER SERIES

A SEARCH FOR DEEPER UNDERSTANDING

START WITH THIS SLIDE, FROM OWO HANDOUT

Example Fourier Series

H140(081019)

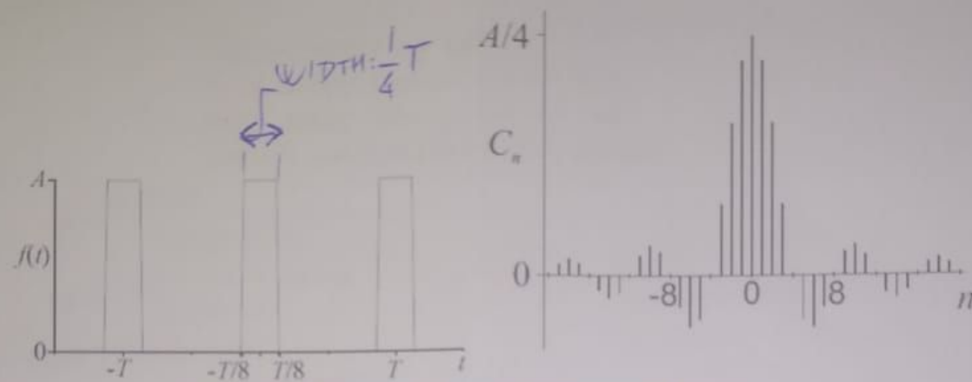


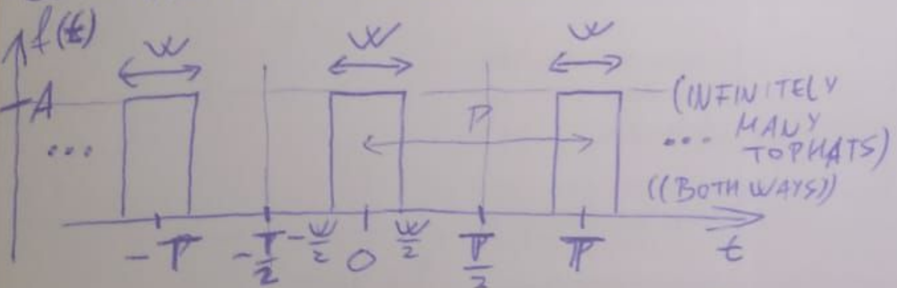
Figure 75: Fourier coefficients for a periodic function.

This function repeats with period $T = 2\pi/\omega_0$, and is non-zero for $-T/8 < t < T/8$. Its Fourier coefficients are

$$C_n = \frac{1}{T}\int_{-T/8}^{T/8} A e^{-in\omega_0 t}\,dt = \frac{A}{T}\left[\frac{e^{-in\omega_0 t}}{-in\omega_0}\right]_{-T/8}^{T/8} = \frac{A}{\pi n}\sin(n\pi/4)$$

The coefficient is zero whenever n is a multiple of 4.

LETS RECREATE THIS RESULT FOR A MORE GENERAL SQUARE-WAVE SIGNAL.



$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\frac{2\pi n}{T}t} = \sum_{n=-\infty}^{\infty} c_n e^{in\omega_0 t} \quad \left(\omega_0 = \frac{2\pi}{T}\right)$$

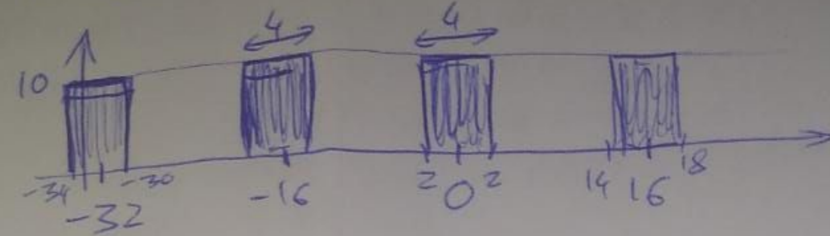
$$c_n = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-i\frac{2\pi}{T}nt}\,dt = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} f(t) e^{-i\omega_0 n t}\,dt$$

$$= \frac{1}{T}\int_{-\frac{W}{2}}^{\frac{W}{2}} A e^{-in\omega_0 t}\,dt = \frac{A}{T}\left[\frac{e^{-in\omega_0 t}}{-in\omega_0}\right]_{-\frac{W}{2}}^{\frac{W}{2}} = \frac{A}{T}\frac{1}{-in\omega_0}(-2i)\sin\left(\frac{n\omega_0 W}{2}\right)$$

$$= \frac{A}{T}\frac{1}{-in\frac{2\pi}{T}}(-2i)\sin\left(\frac{n\omega_0 W}{2}\right) = \frac{A}{n\pi}\sin\left(\frac{n\frac{2\pi}{T}W}{2}\right) = \frac{A}{n\pi}\sin\left(\frac{n\pi W}{T}\right)$$

FOR $n=0$, $c_n = \frac{WA}{T}$ (BY OBSERVING THIS INTEGRAL)

LET'S FIND FOURIER SERIES FOR THIS SIGNAL:



AMPLITUDE

PERIOD

WIDTH

WE CALCULATE C_nS BETWEEN C_{-20} & C_{20} (BOTH INCLUSIVE)

```
import numpy as np
from numpy import sin as sin
from numpy import cos as cos
import matplotlib.pyplot as plt
```

```
A=10
pi = np.pi
T=16
w=4
nrangelim=20
n = np.arange(-nrangelim,nrangelim+1)
```

```
Cn=[]
for eachn in n:
    if eachn != 0:
        Cn.append(A/(pi*eachn)*sin(eachn * pi * w/ T))
    else:
        Cn.append(w*A/T)
```

$C_n = \frac{1}{n\pi} \sin\left(\frac{n \pi w}{T}\right)$

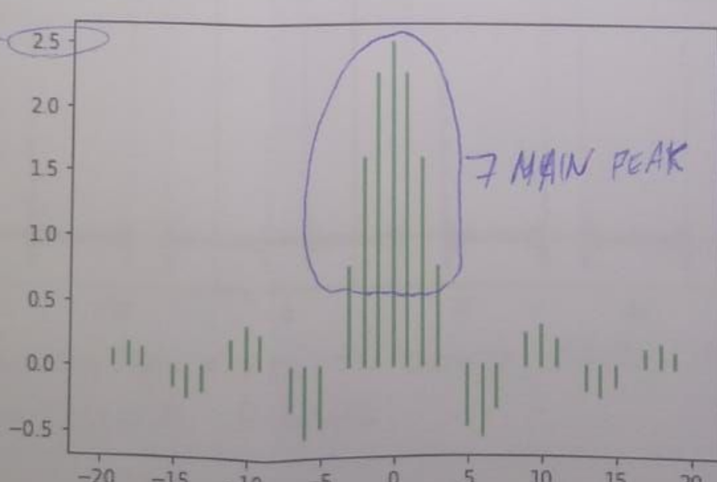
($n=0$ IS AN EXCEPTION, CALCULATED BEFORE.)

```
def ourplotis(n,Cn):
    for each in zip(n,Cn):
        plt.plot([each[0],each[0]],[0,each[1]],c='g')
```

```
ourplotis(n,Cn)
```

THIS IS $\frac{A}{4}$.

NICE AGREEMENT WITH LECTURE SLIDE



LETS RECREATE THE SIGNAL FROM ITS FOURIER SERIES.
SAY, WE ARE INTERESTED IN $f(t)$ FROM $t=-3.5T$
TO $t=3.5T$.
(T WAS 16 IN OUR CASE.)

WE WANT $f(t)$ IN EVERY 0.01 UNIT IN THAT INTERVAL.

```
def f(fromwhicht, towhicht, stepsize, T, Cn, n):

    tvalues=[]
    tvalues.append(fromwhicht)
    for stepcount in range(int((towhicht-fromwhicht)/stepsize)):
        tvalues.append(fromwhicht+(stepcount+1)*stepsize)
```

CREATING DISCRETE t VALUES WHERE $f(t)$ IS GOING TO BE CALCULATED.

RECALL: $f(t)=\sum_{n=-\infty}^{\infty} C_n \exp\left(i\frac{2\pi}{T}nt\right)=\sum_{n=-\infty}^{\infty} C_n\left[\cos\left(\frac{2\pi nt}{T}\right)+i\sin\left(\frac{2\pi nt}{T}\right)\right]$

```
    ft=[]
    for eacht in tvalues:
        value_of_f_at_eacht = sum([Cn[index]*cos(2*pi*eachn*eacht/T)
                                   for index, eachn in enumerate(n)])

        ft.append(value_of_f_at_eacht)

    return tvalues, ft
```

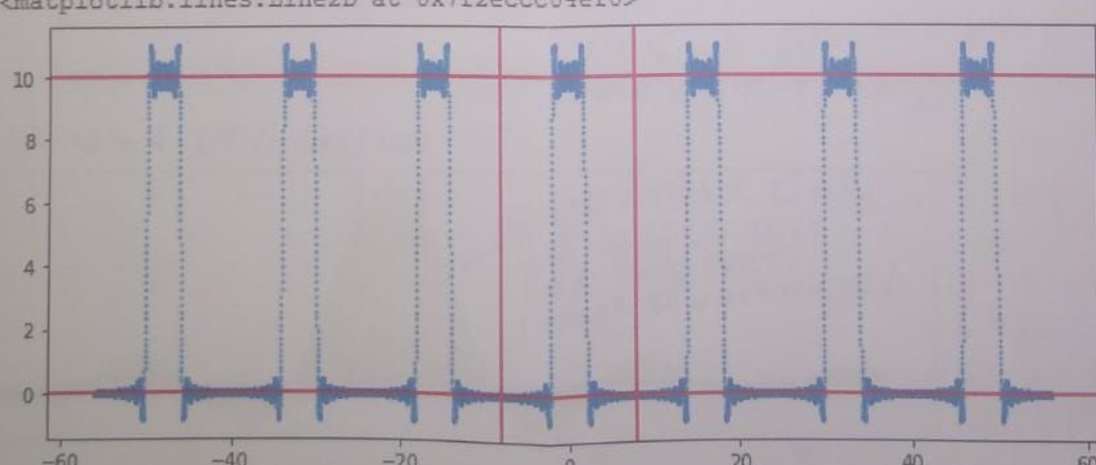
OUR SIGNAL IS EVEN, SO WE DO NOT EXPECT ANY SIN COMPONENTS IN $f(t)$. THAT'S WHY IT'S ONLY COS HERE.

```
tvalues, ft = f(-3.5*T,3.5*T,0.01, T, Cn, n)
```

```
plt.figure(figsize=[10,4])
plt.scatter(tvalues,ft,s=1)
plt.axvline(x=-T/2,c='r')
plt.axvline(x=T/2,c='r')
plt.axhline(y=0,c='r')
plt.axhline(y=A,c='r')
```

FOR CHECKING, CHANGE THIS COS TO SIN & PLOT AGAIN THE SAME THING WE ARE PLOTTING NOW (IE NOTHING CHANGES, JUST COS ⇒ SIN). THE RESULT IS A FLAT LINE.

```
<matplotlib.lines.Line2D at 0x7f2eccc04ef0>
```

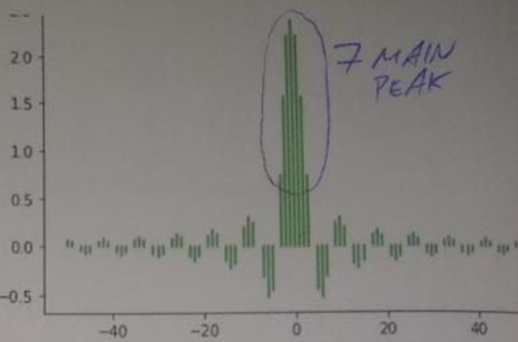


THIS SEEMS TO BE A PRETTY GOOD REPLICA OF THE ORIGINAL SIGNAL.

LET'S USE MORE C_nS NOW!

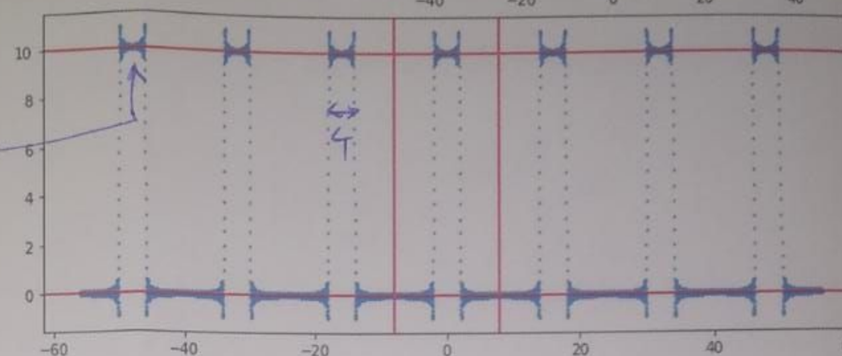
THE SAME PLOTS WITH MORE C_mS:
($A=10$, $T=16$, $W=4$, NRANGELIM$=50$)

NOTE THAT THIS IS NOT ANY WAY MORE SINC FUNCTION LIKE THAN THE PLOT TWO PAGES AGO.

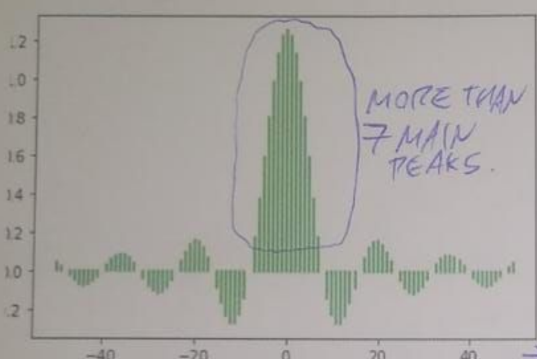


7 MAIN PEAK

INCREASING n, IE CALCULATING MORE C_mS DOES NOT GET US CLOSER TO SINC FUNCTION, BUT IT REDUCES THE ERROR IN THE RETRIEVED SIGNAL.



4



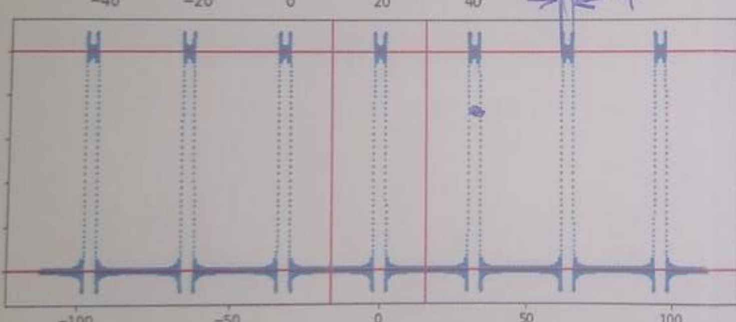
MORE THAN 7 MAIN PEAKS.

LET'S MAKE T BIGGER NOW! RESULTING PLOT

$A=10$
$T=32$
$W=4$
NRANGELIM$=50$ BUT: THIS IS MORE SINC LIKE, BECAUSE T INCREASED.

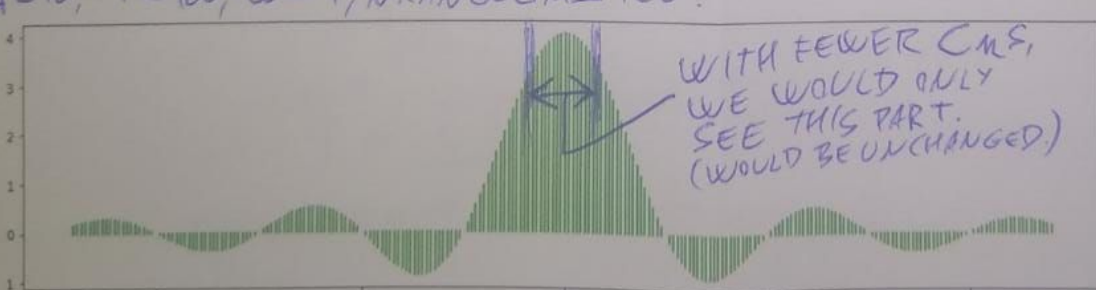
SAME NUMBER OF C_mS CALCULATED HERE & THERE

4



NOW LETS INCREASE THE NUMBER OF C_mS CALCULATED & ALSO T. WE GET THIS:

$A=10$, $T=100$, $W=4$, NRANGELIM$=120$.

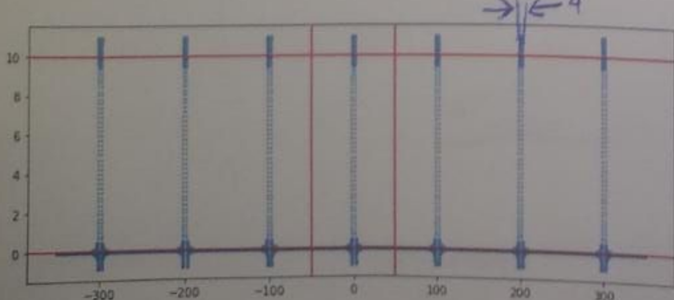


WITH FEWER C_mS, WE WOULD ONLY SEE THIS PART. (WOULD BE UNCHANGED.)

BY INCREASING T, WE MADE THE SHAPE MORE SINC-FUNCTION LIKE.

BY INCREASING THE NUMBER OF C_mS CALCULATED, WE OBTAINED A WIDER SAMPLE OF THE SINC-FUNCTION LIKE SHAPE.

4



WITH SMALLER NRANGELIM, THIS PLOT WOULD HAVE EXACTLY THE SAME SHAPE IN ITS CENTRE, BUT WE WOULD NOT SEE THE SIDES.