

## Fourier Transformu ve DFT

Unlu matematikci Fourier sunu kesfetmisti; periyodik olan bir fonksiyon  $F(x)$  sinus ve cosinus terimlerinin toplami olarak temsil edilebilir.

$$F(x) = a_0 + \sum_{n=1}^{\infty} a_n \cos nx + \sum_{n=1}^{\infty} b_n \sin nx$$

Bu fonksiyonda a ve b sayi degerlerinin bilinmesi gerekmektedir. Onlari nasil buluruz?

$a_k$  degerlerini bulmak icin iki tarafi  $\cos kx$  ile carpip  $\int_{-\pi}^{\pi}$  ile integralini alirsak,

$$\begin{aligned} \int_{-\pi}^{\pi} F(x) \cos kx \, dx &= \int_{-\pi}^{\pi} a_0 \cos kx \, dx + \sum_{n=1}^{\infty} \int_{-\pi}^{\pi} a_n \cos nx \cos kx \, dx + \\ &\quad \sum_{n=1}^{\infty} \int_{-\pi}^{\pi} a_n \sin nx \cos kx \, dx \end{aligned}$$

Esitligin sag tarafinda birinci terim  $\cos(kx)$ ,  $\sin(kx)$ 'e donusur. Fakat sinus fonksiyonu  $\pi$  ve  $-\pi$  noktalarinda (ya da onlari  $k$  ile carpilmis  $2\pi$ ,  $-2\pi$ , vs. gibi katlarinda) sifir degerine sahip oldugu icin, bu terim tamamen sifir olacaktir, formulden atilabilir.

Ikinci terimde  $\cos(nx) \cos(kx)$ 'in ustteki gibi integrali eger  $k$  ve  $n$  esit degilse, sifirdir. Sadece  $n$  ve  $k$  esit ise  $a_k(\cos kx)^2$  degeri elde edilir.  $(\cos kx)^2$ 'in ise ustteki sekilde integrali  $\pi$  sonucunu verir. Yani ikinci terimde olan, o sonsuza kadar giden koca toplam icinde sadece tek bir terim sag kalabilir.

Ucuncu terimde  $\sin nx \cos kx$  carpiminin integrali her zaman sifir degerini dondurur. Bu terim de formulden atilir. Geri kalanlari tekrar duzenlersek,

$$a_k = \int_{-\pi}^{\pi} F(x) \cos kx \, dx$$

sonucunu elde ederiz.  $b_k$  icin benzer islemler, ama bu sefer  $\sin kx$  ile carpilarak yapilrsa ve sonuc asagi yukari ayni.

$$b_k = \int_{-\pi}^{\pi} F(x) \sin kx \, dx$$

$a_0$  icin ise,  $\cos kx$  ya da  $\sin kx$  ile carpmaya gerek yok. Sadece iki tarafin integralini almak yeterli,  $a_0$ 'i istedigimiz icin  $n = 0$  demektir, o zaman  $\sin$  ve  $\cos$  iceren hicbir terime ihtiyac yoktur.

$$\begin{aligned}
\int_{-\pi}^{\pi} F(x)dx &= \int_{-\pi}^{\pi} a_0 dx \\
&= a_0 x \Big|_{-\pi}^{\pi} \\
&= a_0(\pi - (-\pi)) \\
&= 2\pi a_0
\end{aligned}$$

Yani

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(x)dx$$

Kompleks Sayıları Kullanmak

$a_0$ ,  $a_n$  ve  $b_n$  yerine tek bir  $c_n$  turu sayı kullanmak istersek, kompleks sayı sistemine geçmek lazım. O zaman ilk  $F(x)$  formülünü de dönüştürmemiz lazım.

Trigonometrik fonksiyonlarda bilinen iki eşitlik şöyledir:

$$\cos(\theta) = \frac{e^{i\theta} + e^{-i\theta}}{2}$$

$$\sin(\theta) = \frac{e^{i\theta} - e^{-i\theta}}{2i}$$

Bu formülde  $i$  değeri hayali sayı olarak bilinen  $\sqrt{-1}$  değeridir.

$F(x)$  formülünü üstteki trigonometrik eşitliklere göre dönüştürelim.

$$\begin{aligned}
F(x) &= \dots + \sum_{n=1}^{\infty} a_n \cos nx + \sum_{n=1}^{\infty} b_n \sin nx \\
&= \dots + \sum_{n=1}^{\infty} a_n \frac{e^{inx} + e^{-inx}}{2} + \sum_{n=1}^{\infty} b_n \frac{e^{inx} - e^{-inx}}{2i} \\
&= \dots + \sum_{n=1}^{\infty} \frac{a_n e^{inx}}{2} + \frac{a_n e^{-inx}}{2} + \frac{b_n e^{inx}}{2i} - \frac{b_n e^{-inx}}{2i} \\
&= \dots + \sum_{n=1}^{\infty} \frac{a_n e^{inx}}{2} + \frac{a_n e^{-inx}}{2} - \frac{i b_n e^{inx}}{2} + \frac{b_n e^{-inx}}{2}
\end{aligned}$$

Benzer terimleri, yani  $e^{inx}$  ve  $e^{-inx}$  terimlerini beraber yazalım:

$$= .. + \sum_{n=1}^{\infty} \frac{a_n - b_n i}{2} e^{inx} + \frac{a_n + b_n i}{2} e^{-inx}$$

Bolumde olan  $2i$  icindeki  $i$  nasil yukari cikabildi? Bu durum, hayali sayilarin bir ozelligiyle alakali:  $1/i = -i$ . Boylece ikinci ve dorduncu terimdeki arti ve eksi isaretleri degismis oldu. Daha kısa yazmak icin

$$c_{-n} = \frac{a_n + b_n}{2}$$

$$c_n = \frac{a_n - b_n}{2}$$

olarak temsil edersek

$$F(x) = \sum_{n=-\infty}^{\infty} c_n e^{inx}$$

Ustteki  $c_{-n}$  ve  $c_n$  kullanimi bize ek bir avantaj sagliyor:  $-inx$  ibaresindeki eksi degeri de cekip cikartabiliyoruz, eksi deger  $i$ 'den alinip  $n$ 'ye veriliyor yani, ve eksilik toplamdaki alt sinir olarak tanimlaniyor. Nasil olsa final formilde  $i$  ve  $n$  carpildigi icin sonuc degismiyecek, ve tek bir terim kullanabilecegiz. Ek olarak  $a_0$  ise  $c_0$  haline geldi.

Ustteki entegralli teknigin benzerini  $c_n$  icin de kullanabiliriz. Esitligin sag tarafindaki kisim ustteki formilde  $\Sigma$  toplamının acilmis halini kullanalim, ve iki tarafı da  $e^{-ikx}$  ile carpalım, sonra  $-\pi$  ve  $\pi$  arasında integralini alalım:

$$\int_{-\pi}^{\pi} F(x) e^{-ikx} dx = \int_{-\pi}^{\pi} c_0 e^{-ikx} dx +$$

$$\int_{-\pi}^{\pi} c_1 e^{ix} e^{-ikx} dx + ... +$$

$$\int_{-\pi}^{\pi} c_k e^{ikx} e^{-ikx} dx + ...$$

Toplamdaki tum terimleri gostermedik, onemli olan kisim zaten k'inci terim, yani  $e^{-ikx}$  ile carpilan  $e^{ikx}$  ifadesi. Bu carpım basit bir cebirsel islemlerle  $e^{-ikx} e^{ikx} = e^{-ikx+ikx} = e^0 = 1$ , yani bir degerine esit. Diger tum terimler eger integrali hesaplarsak gorebilecegimiz gibi sifira esit. Bir degerinin  $-\pi$  ve  $\pi$  arasında integrali  $2\pi$ . Geri kalanlar

O zaman

$$2\pi c_k = \int_{-\pi}^{\pi} F(x) e^{-ikx} dx$$

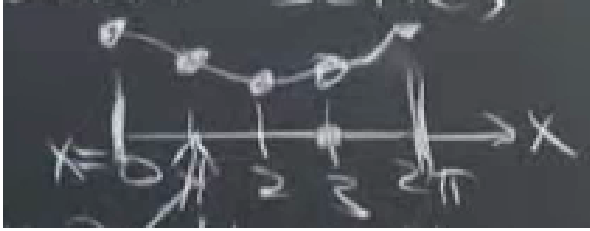
$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(x) e^{-ikx} dx$$

Sifira esitligin nasıl olduğunu cebirsel olarak gösterelim. Entegrali alalım,

$$\begin{aligned} & \int_{-\pi}^{\pi} e^{inx} e^{-ikx} dx \\ &= \int_{-\pi}^{\pi} e^{i(n-k)x} dx \\ &= \frac{e^{i(n-k)x}}{i(n-k)} \Big|_{-\pi}^{\pi} \\ &= e^{in\pi} e^{-ik\pi} - e^{-in\pi} e^{ik\pi} = 0 \end{aligned}$$

DFT

Ayrık (discrete) olarak Fourier modellemesi yapmak istiyorsak, elimizde devamlı (continuous)  $f(x)$  fonksiyonu olmayacak, bir  $f(x)$  fonksiyonun belli noktalarındaki değerleri (olduğunu farzettığımız) verileri içeren bir \*vektor\* olacak. Bu vektörün  $N$  elemanı var diyelim. Fonksiyon periyodik olduğuna göre,  $x$  için  $2\pi$ 'i  $N$  esit parçaya böleriz (tahtadan alınan resim altta). Örneğimizde  $N=4$  olsun.



Ayrıca  $F(x)$  formülü biraz değişecek. Elimizde sonsuz tane nokta olmadığının göre

$$F(x) = \sum_{n=0}^N c_n e^{inx}$$

olması lazım. Şimdi, eğer bütün  $c_k$  değerlerini biliyor olsaydık, bu fonksiyon,  $x=0$  noktasında hangi değere sahip olurdu?

$$f(0) = c_0 + c_1 + c_2 + c_3 + c_4 = Y_0$$

Sonraki  $x$  degerleri  $2\pi/N$ ,  $4\pi/N$  icin asagidaki gibi devam edecegiz, ama ondan once bir  $w$  degiskeni tanimlayalim, bu degiskeni  $w = e^{2\pi i/N}$  olarak alalim. Boylece  $w^2$  dedigimizde ustel islemlerde carpim islemi toplama islemine donusecegi icin  $e^{4i\pi/N}$  degeri elde edilebilir,  $w^3$  ile  $e^{6i\pi/N}$  elde edilir, vs. Bu degerler bize lazim olacak degerler,  $w$  sayesinde formuller daha temiz olacak.  $F(2\pi/N)$  icindeki 3. terim ( $n = 2$ ) nedir?  $c_n e^{inx} = c_2 e^{2i2\pi/N} = c_2 e^{4i\pi/N} = c_2 w^2$ . O zaman

$$f(2\pi/N) = c_o + wc_1 + w^2c_2 + w^3c_3 = Y_1$$

Devam edelim:

$$f(4\pi/N) = c_o + w^2c_1 + w^4c_2 + w^6c_3 = Y_2$$

$$f(6\pi/N) = c_o + w^3c_1 + w^6c_2 + w^9c_3 = Y_3$$

Elimizdeki dort toplam islemine bakinca, bu toplamlar ve carpimların aslında lineer cebir uzerinden matrisler ile gosterilebildigini farkedebiliriz.

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & w & w^2 & w^3 \\ 1 & w^2 & w^4 & w^6 \\ 1 & w^3 & w^6 & w^9 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Her matris icin bir degisken kullanirsak

$$Y = WC$$

$F(x)$ 'ten (yani  $Y$ 'den)  $C$ 'ye gitmek istersek, elimizde  $Y_n$  degerleri var,  $w$  degerleri zaten sabittir,  $W$  bu sabit degere gore olusturulur, o zaman,  $c_n$  sayilarini nasil buluruz?

$$Y = WC$$

$$W^{-1}Y = W^{-1}WC$$

$$W^{-1}Y = C$$

Yani  $W$  matrisinin tersini (inverse) alip, onu  $Y$  ile carpinca elimize  $C$  degerleri gececek.

Gunes Benekleri

Guneste periyodik olarak olan benekler, asagi yukari 11 senede bir ortaya cikarlar. Bu benekler uzun suredir gozlenmekte ve olculmektedir, siddetlerine gore, `sunspots.dat` adli dosyada bulabiliriz. Benek verisindeki periyodik olus, Fourier transformu ile analiz etmek icin uygun. Altteki Python kodu  $w$ ,  $W$  gibi kavramlari kullanarak, ustteki carpimlarla  $C$  vektorunu bulacak. Bu vektor icindeki sayilar Fourier analizindeki belli frekanslara, harmoniklere tekabul ediyor olacaklar.

Bu  $C$  degerlerinden bazilari digerlerinden daha guclu bir etkidir, mesela 11 senelik periyot,  $C$  icinde daha guclu olarak cikacaktır, cikmalidir. Biz kabaca ilk ve son 30 sayi haricindekileri silerek onlara sifir degeri verdik, sonra bu yeni  $C$ 'yi kullanarak benek verisini tekrar "urettirdik". Sonuc onemli olan (ilk ve son 30 degerin temsil ettigi harmoniklerin onemli oldugunu varsayiyoruz) periyotlarin yeni bir toplami oldu. Her iki grafigi de ust uste cizdik ve cakisma oldugunu net bir sekilde gorebiliyoruz. Eger tum  $C$ 'leri kullansaydik, o zaman iki grafik daha benzer, hatta tipatip ayni cikacakti.

```
import scipy

tempdata = np.loadtxt("sunspots.dat")

year=tempdata[:,0]

Y=tempdata[:,1]

N = len(Y)

w = np.exp((2*np.pi*1j)/N)

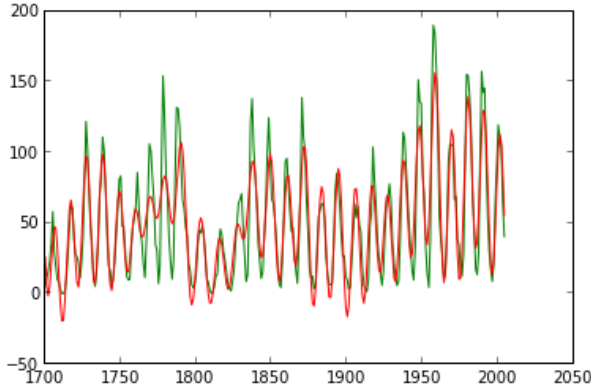
W = np.zeros((N,N), complex)
for i in range(N):
    for k in range(N):
        W[i,k] = w**(i*k)

C = np.dot(np.linalg.inv(W), Y)

C[30:-30] = 0.

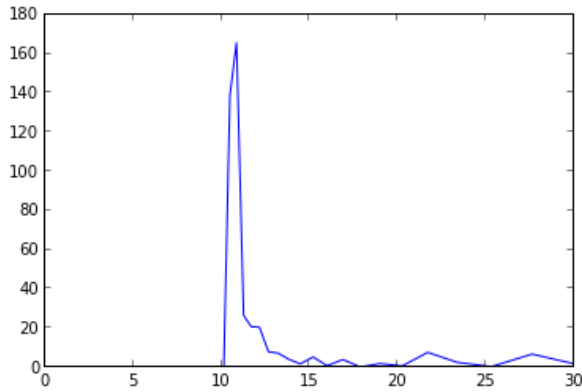
Y_new = scipy.real(np.dot(W, C))

plt.plot(year, Y, 'g')
plt.hold(True)
plt.plot(year, Y_new, 'r')
plt.savefig('fourier_1.png')
```



En yuksek periyotu gormek istersek, alttaki kodu kullanabiliriz [6].

```
Y = C
n=len(Y)
power = np.abs(Y[range(1,int(n/2))])**2
nyquist = 1./2
freq = np.array(map(float, np.array(arange(1,int(n/2)))) / (n/2)*nyquist)
period=1./freq;
plt.plot(period,power)
plt.xlim(0,30)
plt.savefig('fourier_02.png')
```



## FFT

Bitirmeden once FFT konusundan bahsedelim. **DFT** algoritmasi kodda goruldugu gibi bir  $W$  matrisi ortaya cikarir ve once tersini alma, sonra bu ters ile bir carpim islemi yaparak  $C$  sonucunu uretir.  $O$  notasyonunu kullanirsak DFT'nin karmasikligi  $O(N^2)$ 'dir. Bu iyi bir hizdir.

FFT algoritmasi ustteki carpimin bazi ozelliklerini kullanarak DFT'yi daha da hizlandirir ve  $O(\frac{1}{2}N\log_2 N)$  hizina getirir. FFT'den bu makalede bahsetmeyecegiz, aklimizda olsun, Scipy uzerinde fft cagrisi bu algoritmayi kullanir.

Kaynaklar

[1] Strang, G., OCW MIT Lecture #30, Computational Science and Engineering

[2] Strang, G., Computational Science and Engineering, sf. 340-370

[3] Chu, E., Discrete and Continuous Fourier Transforms

[4] Kammler, D., A First Course in Fourier Analysis

[5] Mattuck, A., OCW MIT Lecture #17-19, Differential Equations

[6] [www.mathworks.de/products/matlab/examples.html?file=/products/demos/shipping/matlab](http://www.mathworks.de/products/matlab/examples.html?file=/products/demos/shipping/matlab)