Data Py - inleveropdracht 1

Boris van Boxtel, Brechtje Poppen,
Floris Oostenbrug, Lotte Gritter

23 December 2022 - Week 51

Fourierreeksen

Testfuncties

Convergentie

(a). Vanaf nu is de variabelen f_0 gelijk aan 1. Gegeven is dat de coëfficiënten van de Fourierreeks van de driehoeksgolf gegeven worden door:

$$A_n = \frac{8}{\pi^2} \begin{cases} 0 & n \text{ even} \\ \frac{(-1)^{(n-1)/2}}{n^2} & n \text{ oneven} \end{cases} . \tag{1}$$

We berekenen de eerste n van deze coëfficiënten van de driehoeksgolf met de volgende python code:

```python
def gen_coefs_driehoeksgolf(n):
    coefs_list = np.array([])
    for i in range(n):
        if i % 2 == 0:
            coefs_list = np.append(coefs_list, 0)
        else:
            coefs_list = np.append(coefs_list, (-1)**( (i - 1) / 2 ) / i**2)
    return coefs_list * 8 / (np.pi**2)
```

Code blok 1: Code voor het genereren van de coëfficiënten bij verg. (1)

Ook is gegeven dat de coëfficiënten van de Fourierreeks van de zaagtandgolf gegeven worden door:

$$A_0 = 0 \quad A_{n>0} = -\frac{1}{\pi n}. \tag{2}$$

We berekenen de eerste n van de coëfficiënten van de zaagtandgolf met de volgende python code:

```python
def gen_coefs_zaagtandgolf(n):
    coefs_list = np.array([0])
    for i in range(1, n):
        coefs_list = np.append(coefs_list, -1/(np.pi * i))
    return coefs_list

```

Code blok 2: Code voor het genereren van de coëfficiënten bij verg. (2)

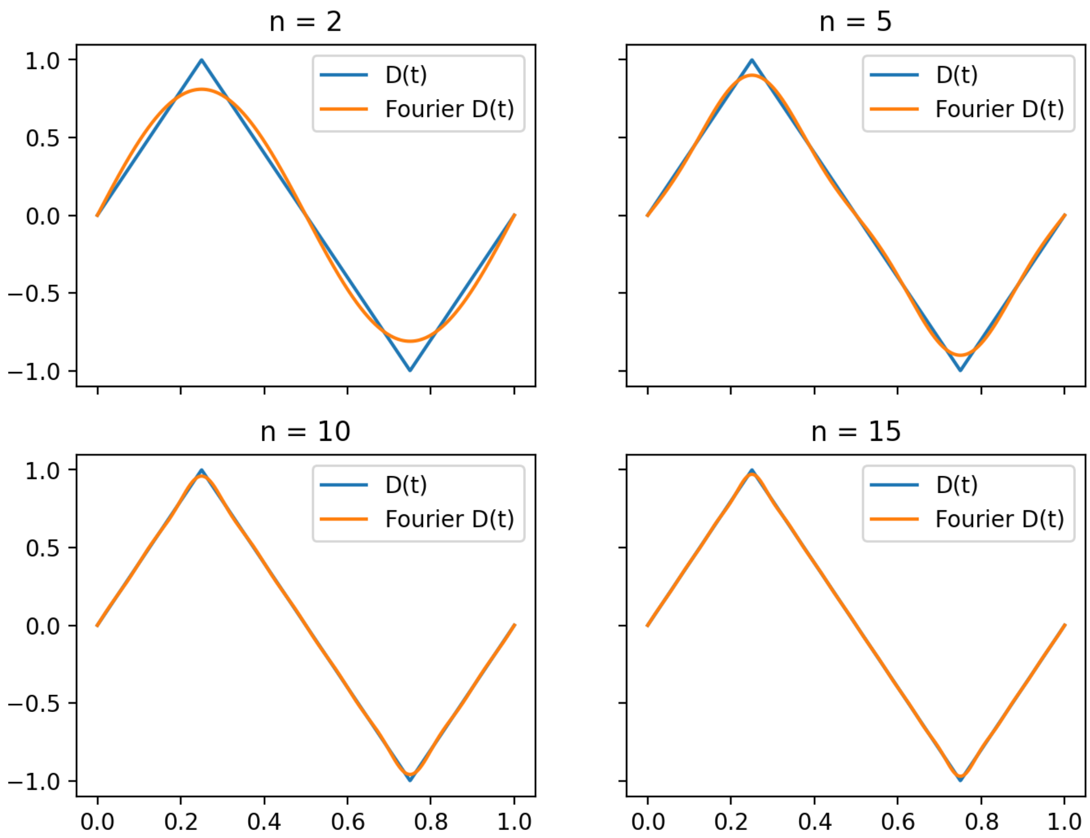
We kunnen met deze functies en de volgende code een illustratie maken van de Fourier transform van de driehoeksgolf:

```python
t = np.linspace(0,1,num=1000)  # Maakt een linspace aan.

fig, ax = plt.subplots(2, 2, sharex=True, sharey=True)  # Dit maakt de figuur aan.

ax[0,0].plot(t,  driehoek_golf(t, f_0=1), label = 'D(t)')  # Plot de orginele functie linksboven.
ax[0,0].plot(t, fourierreeks(t, gen_coefs_driehoeksgolf(2), 1), label = 'Fourier D(t)')      # Plot de fourier series op dezelfde plek.
ax[0,0].legend()  # Teken een legenda.
ax[0,0].set_title('n = 2')  # Zet een titel van de subfiguur.

ax[0,1].plot(t,  driehoek_golf(t, f_0=1), label = 'D(t)')
ax[0,1].plot(t, fourierreeks(t, gen_coefs_driehoeksgolf(5), 1), label = 'Fourier D(t)')
ax[0,1].legend()
ax[0,1].set_title('n = 5')

ax[1,0].plot(t,  driehoek_golf(t, f_0=1), label = 'D(t)')
ax[1,0].plot(t, fourierreeks(t, gen_coefs_driehoeksgolf(10), 1), label = 'Fourier D(t)')
ax[1,0].legend()
ax[1,0].set_title('n = 10')

ax[1,1].plot(t,  driehoek_golf(t, f_0=1), label = 'D(t)')
ax[1,1].plot(t, fourierreeks(t, gen_coefs_driehoeksgolf(15), 1), label = 'Fourier D(t)')
ax[1,1].legend()
ax[1,1].set_title('n = 15')
plt.gcf().set_size_inches(8, 6)  # De default grootte waarmee matplotlib een foto opslaat
```

```
# is aan de kleine kant, hiermee pas ik de grootte aan van het plaatje.
fig.savefig('../images/Fourier_driehoekgolf.png', bbox_inches='tight', dpi=200)
# En als laatste sla het plaatje op.
```

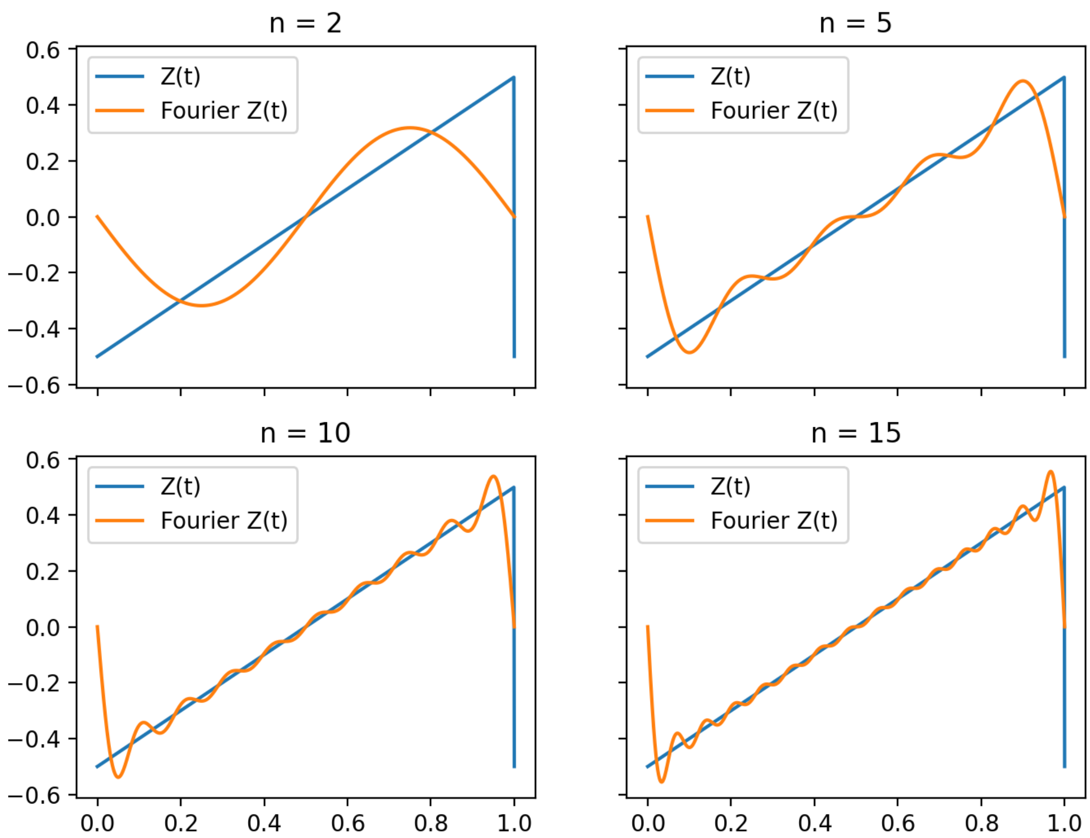
Code blok 3: Code voor het genereren van Figuur 1

Het plaatje dat hiermee wordt gegenereerd is het volgende:



Figuur 1: Illustratie van Fouriertransformatie driehoeksgolf

We kunnen met vergelijkbare code een plaatje maken voor de zaagtandgolf:



Figuur 2: Illustratie van Fouriertransformatie zaagtandgolf

(b). We gebruiken de volgende functie om het totale kwadratische verschil van twee array's te berekenen:

```
# Deze functie neemt twee arrays, en berekent de som van de
↪ kwadraten van de verschillen van elk element van de arrays.
def tot_kwadratisch_verschil(arr1, arr2):
    if arr1.size != arr2.size:
    ↪ # Hier wordt getest of de arrays dezelfde grootte hebben.
        print("Voor het kwadratisch verschil moeten de arrays
        ↪ dezelfde grootte hebben.")
        return

    return np.sum( (arr1 - arr2)**2 )
```

Code blok 4: Code voor het berekenen van het totale kwadratische verschil

We vinden het totale kwadratische verschil van bijvoorbeeld de driehoeksgolf met zijn Fourier transform met 2 coëfficiënten als volgt:

```
print( tot_kwadratisch_verschil(driehoek_golf(t, f_0=1),
↪ fourierreeks(t, gen_coefs_driehoeksgolf(2), 1)) )   # Bereken
↪ het kwadratisch verschil bij 2 Fourier coeffiecienten
```

Deze code geeft de volgende waarden voor de totale kwadratische verschillen voor een verschillend aantal Fourier coëfficiënten n:

Driehoeksgolf

n	totaal kwadratisch verschil
2	4.817
5	0.765
10	0.054
15	0.020

Zaagtandgolf

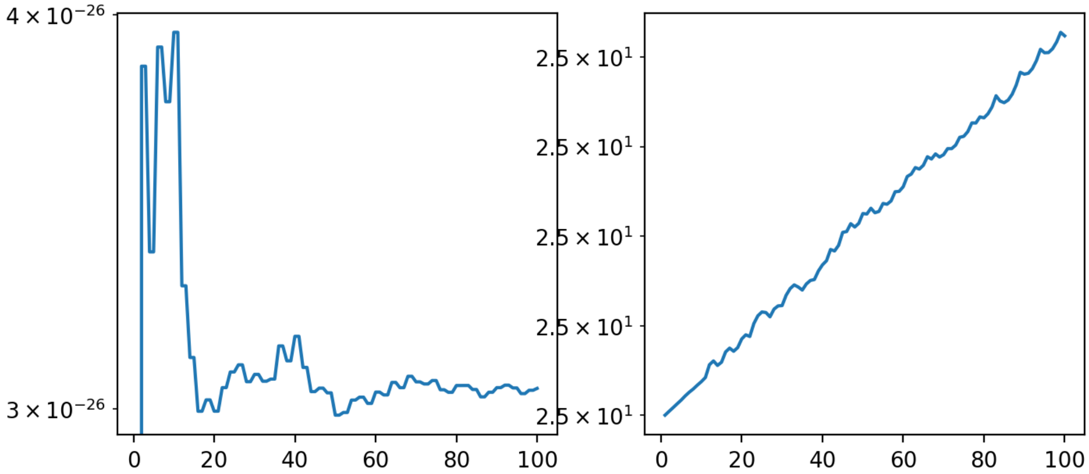
n	totaal kwadratisch verschil
2	32.89
5	11.45
10	5.576
15	3.744

We zien in de plots bij de driehoeksgolf dat deze bij hetzelfde aantal Fourier coëfficiënten ”dichter”bij de echte functie is dan de zaagtandgolf. Dit zien we vervolgens direct ook terug in de zojuist berekende kwadratische verschillen; deze zijn namelijk bij de zaagtandgolf een stuk groter.

(c). Tot slot kunnen we met de volgende code de totale kwadratische verschillen van zowel de driehoeksgolf als de zaagtandgolf weergeven. We kunnen hier helaas geen gebruik maken van numpy’s vermogen om met array’s te rekenen, dus we maken gebruik van een for loop.

```
n = np.linspace(1,100,num=100)  # Maakt een linspace aan.

fig, ax = plt.subplots(1, 2)
out_list_driehoek = np.array([])  # Initializeer de lijsten.
out_list_zaagtand = np.array([])
for i in n:  # Bereken de kwadratische verschillen
    out_list_driehoek = np.append(out_list_driehoek, tot_kwadratisch_verschil( driehoek_golf(n), fourierreeks( n, gen_coefs_driehoeksgolf(int(i)), 1 ) ))
    out_list_zaagtand = np.append(out_list_zaagtand, tot_kwadratisch_verschil( zaagtand_golf(n), fourierreeks( n, gen_coefs_zaagtandgolf(int(i)), 1 ) ))
ax[0].plot(n, out_list_driehoek)  # Plot het verschil tegen het aantal coefficienten
ax[0].set_yscale('log')  # Zet de yas op log schaal
ax[0].title.set_text('Kwadratisch verschil driehoeksgolf', loc='bottom')            # Zet de titel van de subplot
ax[1].plot(n, out_list_zaagtand)
ax[1].set_yscale('log')
ax[1].title.set_text('Kwadratisch verschil driehoeksgolf', loc='bottom')

plt.gcf().set_size_inches(8, 3.5)
fig.savefig('../images/tot_kwadratisch_verschil.png', bbox_inches='tight', dpi=200)         # Sla het bestand op
```

De figuur is het volgende:



Figuur 3: Kwadratisch verschil op de y-as tegen het aantal coëfficiënten op de x-as

Conclusies