

Obrada informacija: Prva laboratorijska vježba

Rijeke imaju veliki gospodarski značaj, od termoelektrana i nuklearnih elektrana kojima su rijeke nužne za rad, javne vodoopskrbe do poljoprivrede i transporta. Premda ljudi danas uvelike upravljaju vodotocima, uslijed vremenskih prilika ipak može doći do poplava, erozija tla i sličnih prirodnih nepogoda. Kako bismo se mogli na vrijeme pripremiti za potencijalne probleme, potrebno je istražiti ponašanje rijeka na temelju višegodišnjih praćenja vodostaja. U ovoj laboratorijskoj vježbi ćete analizirati vodostaje rijeke Save kroz period od 25 godina.

Unutar komprimiranog direktorija u kojem se nalazi ova bilježnica, nalazi se i .mat datoteka s podacima o vodostaju rijeke Save u periodu od 1.1.1982. do 31.12.2007. Svaki podatak označava jedan dan mjerenja vodostaja.

Vježba se izvodi u Pythonu/Google Colabu, a sve naredbe potrebne za provedbu vježbe te njihova objašnjenja dana su u predavanju. Ova laboratorijska vježba nosi 3 boda.

Kad ste gotovi s vježbom, na *Moodle* postavite .pdf izvješće s vježbe (.pdf izvješće možete generirati s naredbom File-Print-Location: PDF).

1. Učitajte biblioteke: NumPy, matplotlib.pyplot, scipy.fft, signal iz scipy te pywt.

```
# Ovo je mjesto na kojem možete izvoditi svoj kod.  
import numpy as np #  
import matplotlib.pyplot as plt #  
from scipy.fft import fft #  
from scipy import signal #  
import pywt #
```

2. Učitajte podatke o vodostaju iz .mat datoteke koja se nalazi unutar istog komprimiranog direktorija. Učitani podaci su spremljeni u rječnik. Izdvojite ključ i pripadne podatke o vodostajima te po potrebi smanjite dimenziju.

Savjet: .mat datoteku učitajte sa svog Google Drivea (unutar mape Colab Notebooks na svom Google Driveu postavite .mat datoteku). Ostali savjeti nalaze se u sljedećem programskom odsječku.

```
# Ovo je mjesto na kojem možete izvoditi svoj kod.  
  
from google.colab import drive  
drive.mount('/content/drive')
```

```
# Nakon što pokrenete ovaj odsječak koda, pojavit će se link u prozoru ispod. Kliknite na :
# dobiti kod koji trebate kopirati u izlaz ispod ovog prozora (u prozor "Enter your author:
# pojavit će se poruka "Mounted at /content/drive".
# Ako unutar takvog Colab Notebook direktorija imate spremljenu datoteku "OIkobas.mat", mo
from scipy import io
kobas_iz_matlaba=io.loadmat("drive/My Drive/Colab Notebooks/3_OIkobas.mat")
print(kobas_iz_matlaba)
```

```
Mounted at /content/drive
{'__header__': b'MATLAB 5.0 MAT-file, Platform: MACI64, Created on: Sat Oct 10 13:46:
    [674],
    [685],
    ...,
    [242],
    [230],
    [228]], dtype=uint16)}
```

3. Ispišite: broj dana za koje postoje podaci o vodostajima, srednju vrijednost vodostaja, standardnu devijaciju, minimalnu vrijednost, maksimalnu vrijednost te median.

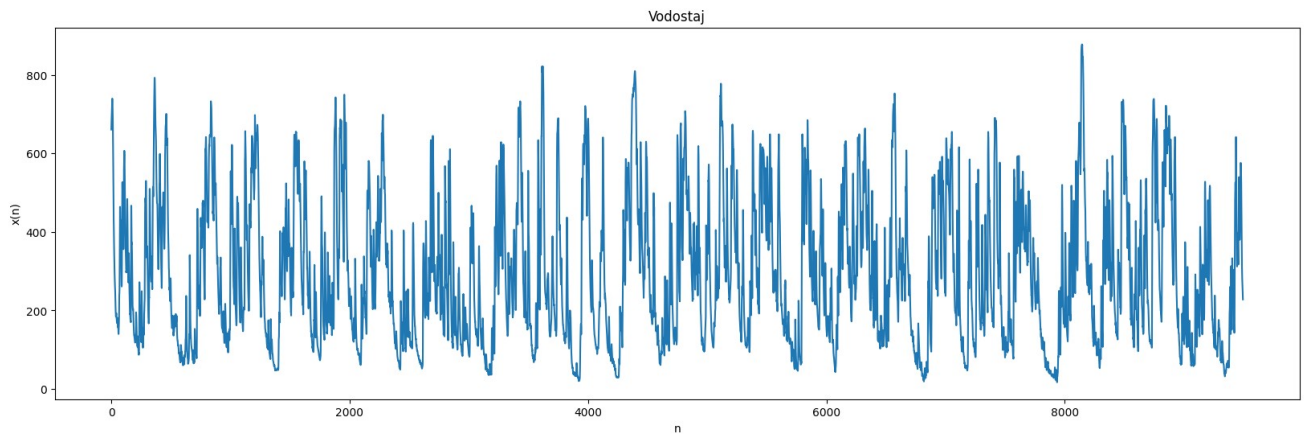
```
# Ovo je mjesto na kojem možete izvoditi svoj kod.
kobas = kobas_iz_matlaba['kobas']
kobas = np.squeeze(kobas)
#kobas = kobas[0].toList()
print(kobas.size)
print(np.mean(kobas))
print(np.std(kobas))
print(np.min(kobas))
print(np.max(kobas))
print(np.median(kobas))
```

```
9496
294.25758213984835
183.09167165945507
17
878
255.0
```

4. Nacrtajte vodostaj. Obilježite x i y os te naslov slike.

```
plt.figure(figsize=(20,6))
#plt.subplot(1,1,1)
#plt.stem(kobas)
plt.title('Vodostaj')
plt.xlabel('n')
plt.ylabel('x(n)')
#plt.xticks(np.arange(0, len(kobas)))
```

```
time = np.arange(0, len(kobas))  
plt.plot(time, kobas)  
plt.show()
```

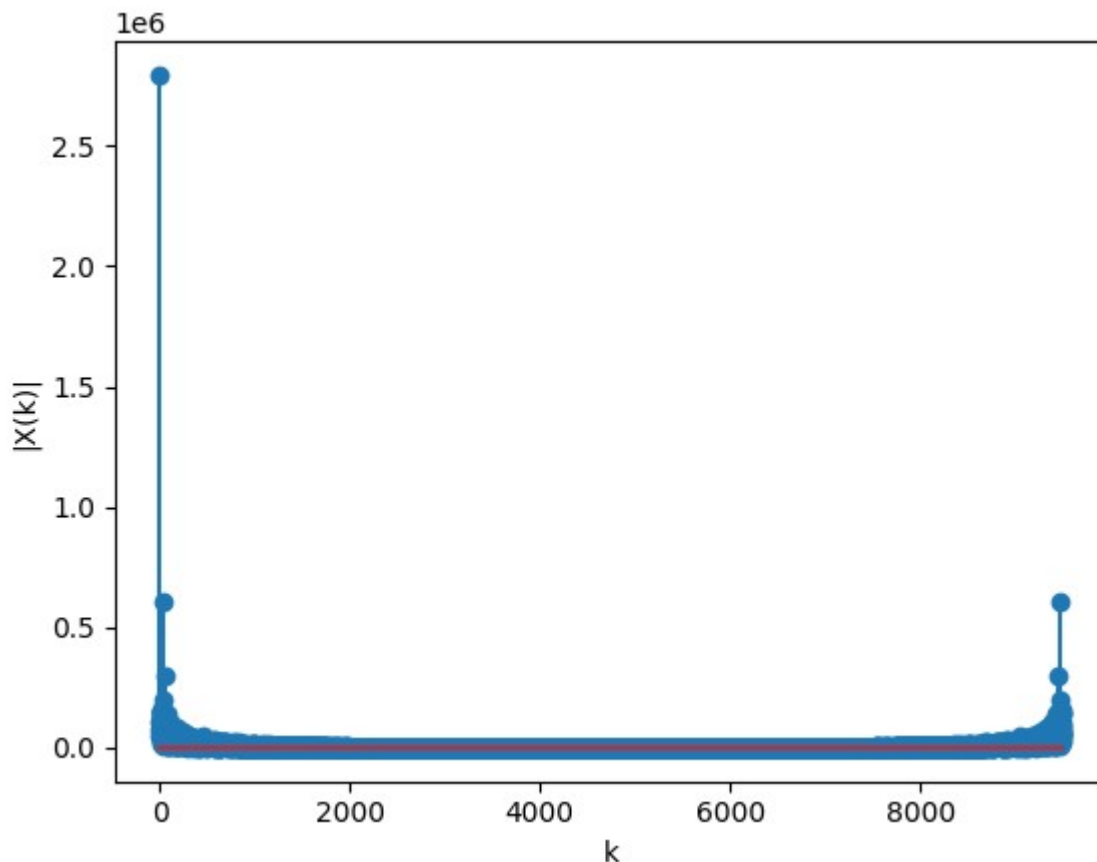


5. Izračunajte diskretnu Fourierovu transformaciju zadanog vodostaja. Ispišite dobivene vrijednosti. Nacrtajte apsolutnu vrijednost dobivenih koeficijenata.

```
# Ovo je mjesto na kojem možete izvoditi svoj kod.  
#x = np.array(kobas)  
#print(x)  
X = fft(kobas)  
print(X)  
  
plt.figure(1)  
plt.subplot(1,1,1)  
plt.stem(np.abs(X))  
#plt.plot(X,absX)  
plt.xlabel('k')  
plt.ylabel('|X(k)|')
```

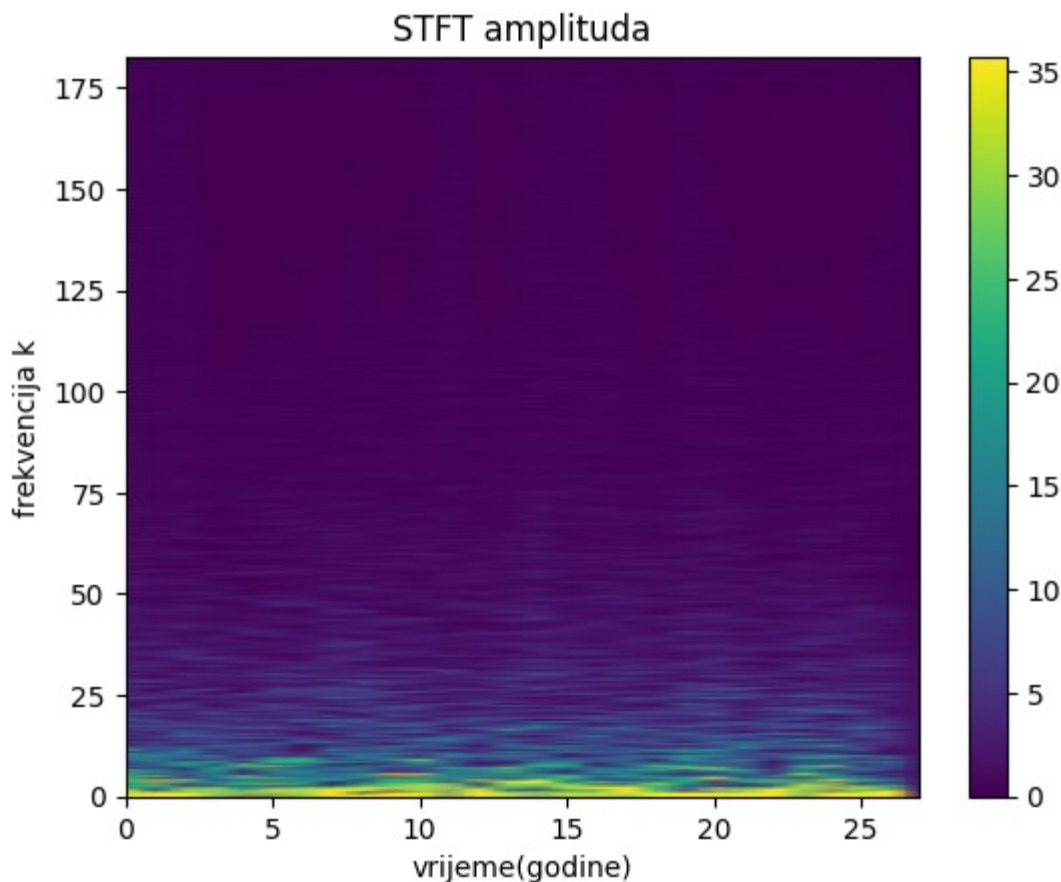
```
#plt.yticks(np.arange(0,kobas.size,step=1))
plt.show()
```

```
[2794270.          -0.j          -9312.52756202+49784.33987081j
 80617.82235263-75905.53826228j ...  -75144.8689581 -18406.78122837j
 80617.82235263+75905.53826228j   -9312.52756202-49784.33987081j]
```



6. Izračunajte Fourierovu transformaciju na vremenskom otvoru zadanog vodostaja koristeći pravokutni otvor širine 2 godine. Prikažite rezultat pomoću pcolormesh. Odgovorite: koje frekvencije su vidljive u vodostaju? Koje godine se javljaju?

```
# Ovo je mjesto na kojem možete izvoditi svoj kod.
fs = 365
window = 'boxcar'
f,t,Zxx = signal.stft(kobas,fs>window,nperseg=365*2) #slajd 73
#plt.plot()
plt.pcolormesh(t,f,np.abs(Zxx),vmin=0,vmax=np.abs(Zxx.max())/10,shading='gouraud')
plt.colorbar()
plt.title('STFT amplituda')
plt.ylabel('frekvencija k')
plt.xlabel('vrijeme(godine)')
plt.show()
```



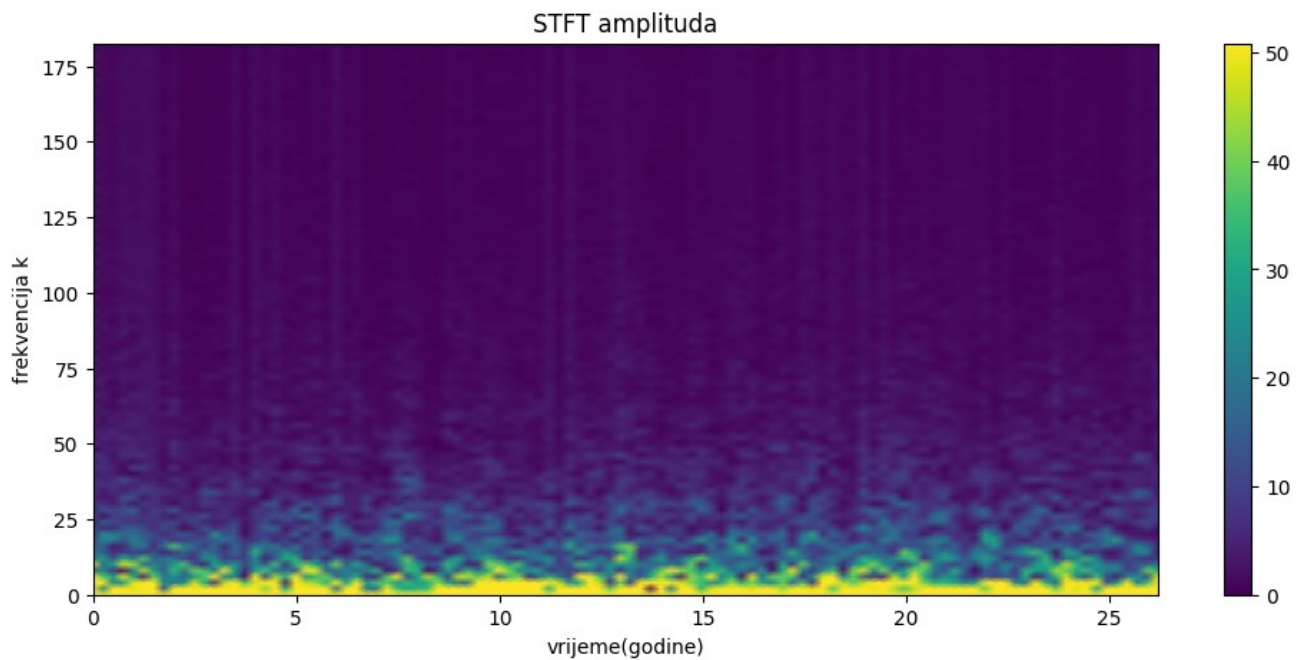
Vide se veće amplitude za mali k (do otprilike 4). Možemo zaključiti da kako se k povećava sve je veći razmak između susjednih amplituda. Za npr. $k = 1$ vidimo uglavnom jaču prisutnost frekvencija a kad je $k = 2$ vidimo razmake između. Malo su očitije godine: 7,10,22,25

7. Upotrijebite drugu širinu otvora (po izboru) i ponovite prethodni zadatak. Ispišite koju širinu ste upotrijebili. Objasnite razlike u dobivenim slikama.

```
fs = 365
T = 365
window = 'boxcar'
f,t,Zxx = signal.stft(kobas,fs>window,nperseg=T/2)

print(T/2)
plt.figure(figsize=(12,5))
plt.pcolormesh(t,f,np.abs(Zxx),vmin=0,vmax=np.abs(Zxx.max())/10,shading='gouraud')
plt.colorbar()
plt.title('STFT amplituda')
plt.ylabel('frekvencija k')
plt.xlabel('vrijeme(godine)')
plt.show()
```

182.5



Razlika je u tome što je u drugoj slici korišten kraći vremenski interval (pola godine) pa dobivamo bolju rezoluciju za vrijeme, ali lošiju za frekvenciju k . Sada malo bolje vidimo koje godine se javljaju.

8. Ispišite sve obitelji kontinuiranih valića koje se nalaze u PyWavelets biblioteci. Na istoj slici, koristeći subplot naredbu, nacrtajte dvije valićne funkcije po izboru. U naslovu svake slike napišite o kojim valićima se radi.

```
print(pywt.wavelist(kind='continuous'))

wave1 = 'morl'
wave2 = 'mexh'

w = pywt.ContinuousWavelet(wave1)
psi, t = w.wavefun(level=10)

plt.figure(1)
plt.subplot(2,1,1)
plt.plot(t,psi)
plt.title('Morlet wavelet')
plt.xlabel('time')
```

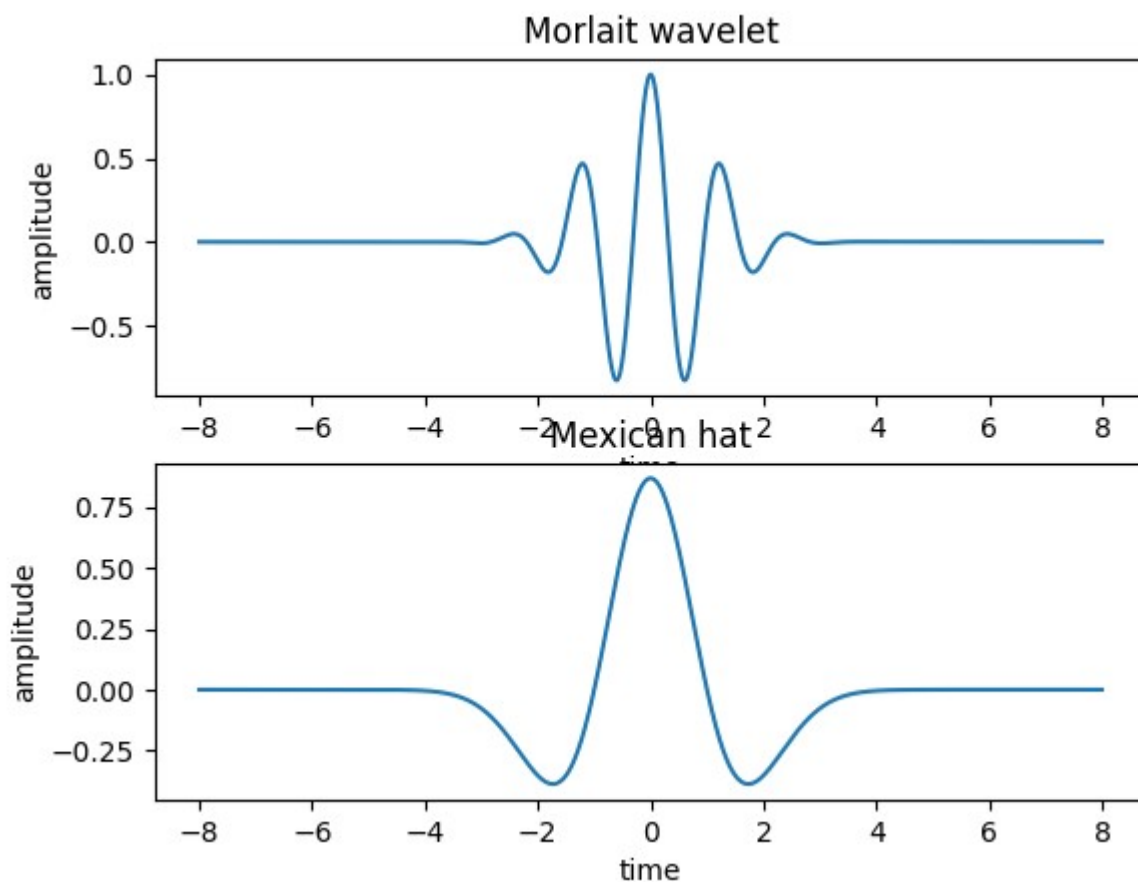
```
plt.ylabel('amplitude')

w = pywt.ContinuousWavelet(wave2)
psi, t = w.wavefun(level=10)

plt.subplot(2,1,2)
plt.plot(t,psi)
plt.title('Mexican hat')
plt.xlabel('time')
plt.ylabel('amplitude')

plt.show()
```

['cgau1', 'cgau2', 'cgau3', 'cgau4', 'cgau5', 'cgau6', 'cgau7', 'cgau8', 'cmor', 'fbs



9. Koristeći Morlet valić, odredite kontinuiranu valićnu transformaciju zadanog vodostaja. Nacrtajte apsolutnu vrijednost dobivenih koeficijenata. Odgovorite: koje frekvencije su vidljive u vodostaju? Koje godine se javljaju?

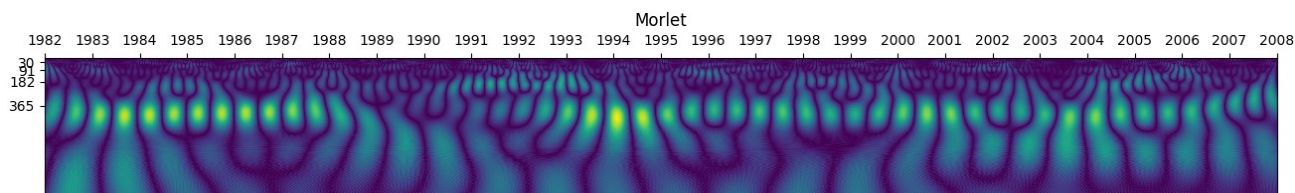
```
# Ovo je mjesto na kojem možete izvoditi svoj kod.
val = 'morl'
T = 365
w = pywt.ContinuousWavelet(val)
psi, t = w.wavefun(level=10)
step = 0.7
```

```

step = 0.1
skala = np.arange(0.1, 2*T, step)
coef, freqs = pywt.cwt(kobas, skala, val)

plt.matshow(abs(coef))
plt.xticks(np.arange(0, kobas.size, 365), np.arange(1982, 2009, 1))
plt.yticks([T//12, T//4, T//2, T])
plt.title('Morlet')
plt.show()

```



Najizraženije amplitude su za $k = 1$ za godine: 1982-1988, 1993-1995, 2000-2001, 2003-2005. Također su u manjoj mjeri vidljive amplitude za $k = 2$.

10. Odgovorite: objasnite razliku u dobivenim rezultatima STFT i CWT.

STFT: dijeli podatke na segmente i na svakom segmentu obavi FT. Rezolucija slike temelji se na širini prozora. Manja širina daje bolju sliku za vrijeme, ali lošiju za frekvenciju. Također, STFT se koristi za diskretnu analizu.

CWT: koristi se za kontinuiranu analizu. Ne vidi se podjela signala u pravokutnike. Umjesto otvora, CWT upari dani signal sa kontinuiranim valićem i rezultat je graf koji pokazuje jačinu pojedinih frekvencija za neki trenutak u vremenu. CWT je više intenzivan za računalu.

