

Extraövningar i PYTHON

1. Rita en graf av funktionen $f(x) = 2 \sin(x) + 3 \sin(2x)$ på intervallet $-\pi \leq x \leq \pi$ i en figur. Rita grafen $g(x) = 8 \sin(x) + 2 \sin(2x)$ i samma figur. Använd samma x -värden.

2. Skapa följande fält i NumPy:

- (a) Ett fält som innehåller 100 värden mellan 0 och π jämnt fördelade över intervallet
- (b) Ett fält som innehåller de jämna talen 2, 4, 6, ..., 100
- (c) Ett fält som innehåller de udda talen 99, 97, 95, ..., 1
- (d) Ett fält som innehåller talen 1, -3, 5, -7, ..., -95, 97

3. Låt

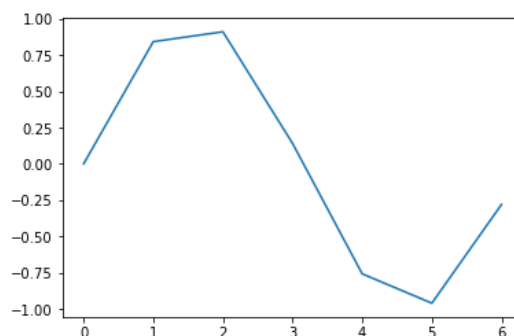
```
import numpy as np
T=np.array([31,28,31,30,31,30,31,31,30,31,30,31])
```

- (a) Beräkna medelvärdet av elementen i T. Använd kommandona `np.sum` och `np.size` eller kommandot `np.mean`.
 - (b) Beräkna differenserna av elementen, dvs $T[1]-T[0]$, $T[2]-T[1]$... med kommandot `np.diff`.
 - (c) Använd `np.max` och `np.min` för att bestämma största och minsta värde i T.
 - (d) Bilda ett fält som innehåller indexen för de element som har värdet 31 i T.
4. Man kan rita en triangel genom att placera tre punkter på randen till en cirkel. Sekvensen nedan ritar en triangel, utöka sekvensen så att den även ritar en cirkel som omsluter triangeln.

```
import numpy as np
import matplotlib.pyplot as plt
t=np.linspace(0,2*np.pi,4);
x=np.cos(t); y=np.sin(t)
plt.plot(x,y)
plt.axis('equal')
```

5. Beräkna summan $1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{100}$ genom att bilda ett fält som innehåller de hundra termerna i summan och sedan summera elementen i fältet med `sum`.
6. Exemplet nedan ritar en graf för $\sin(x)$ på intervallet $0 \leq x \leq 6$. Grafen består av sju punkter och sex räta linjer (delsträckor) som förbinder dem.

```
import numpy as np
import matplotlib.pyplot as plt
x=np.arange(0,7,1)
y=np.sin(x)
plt.plot(x,y)
```



- (a) Använd Pythagoras sats och beräkna längderna av alla delsträckorna. Längden av den första delsträckan ges av

```
np.sqrt((x[1]-x[0])**2+(y[1]-y[0])**2)
```

Använd förslagsvis kommandot `np.diff`.

- (b) Summera längderna av delsträckorna för att få längden på hela grafen. Använd `np.sum`.
 (c) Ändra antalet delsträckor till 100 och beräkna en approximation till längden av grafen för $\sin(x)$ på intervallet $0 \leq x \leq 6$ (dvs. approximera grafens båglängd).

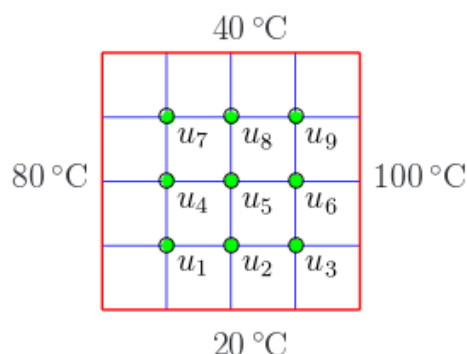
7. När man multiplicerar två matriser med varandra multiplicerar man ihop rader med kolumner. Rader i den vänstra matrisen multipliceras med kolumner i den högra

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

I PYTHON kan man använda kommandot `np.matmul` från Numpy för att multiplicera matriser. Skriv in matriserna i exemplet ovan, låt dem vara `np.array`, och beräkna produkten med `matmul`. Testa även operatoren `*` och beskriv hur operationerna fungerar.

8. Beräkna temperaturen på en stålplatta där plattans kanter hålls vid temperaturer enligt figuren



Antag att temperaturen i en nodpunkt är medelvärdet av temperaturerna i de närmsta nodpunkterna i väster, öster, söder och norr. Låt u_1, u_2, \dots, u_9 beteckna temperaturerna i de olika nodpunkterna. Sätt upp de ekvationer som ger temperaturen i de olika nodpunkterna. Skriv det linjära ekvationssystemet på matrisform $\mathbf{A}\mathbf{u} = \mathbf{b}$ och lös detta i PYTHON.

Lösningsförslag till många av extraövningarna

1.

```
from numpy import linspace
from matplotlib.pyplot import plot
from math import sin, pi

x=linspace(-pi,pi)
f=2*sin(x)+3*sin(2*x)
g=8*sin(x)+2*sin(2*x)
plot(x,f)
plot(x,g)
```
2.

```
import numpy as np
a=np.linspace(0,np.pi,100)
b=np.arange(2,101,2)
c=np.arange(99,0,-2)

n=49
i=np.ones(n)          # fält med 49 ettor
i[np.arange(1,n,2)]=-1 # Låt vartannat element vara -1
d=np.arange(1,98,2)*i  # multiplicera vartannat element i [1, 3, 5, ...] med -1
```
3.

```
import numpy as np
T=np.array([31,28,31,30,31,30,31,31,30,31,30,31])
# (a)
a=np.sum(T)/np.size(T)
a=np.mean(T)

# (b)
b=np.diff(T)

# (c)
storst=np.max(T)
minst=np.min(T)

# (d)
plats=np.arange(1,13,1);
svar=plats[T==31]
```
- 4.
5.

```
import numpy as np
n=100
x=1/np.arange(1,101,1)
svar=np.sum(x)
```
6.

```
import numpy as np
x=np.arange(0,7,1)
```

```
y=np.sin(x)
# (a) Längderna av de 6 delsträckorna
L=np.sqrt(np.diff(x)**2+np.diff(y)**2)
# (b) Längden av hela sträckan
b=np.sum(L)
# (c) Fler delsträckor
x=np.linspace(0,6,100)
y=np.sin(x)
L=np.sqrt(np.diff(x)**2+np.diff(y)**2)
c=np.sum(L)
```

7.

8.