

Breakdown kode øving 12

Simen Hustad

November 10, 2021

Dette dokumentet har til hensikt å bryte ned de individuelle kodesnuttene som kreves for hver oppgave på dataøvingen. All kode som står bak `#` er kommentarer i koden

Oppgave 1

a)

Oppgitt r_1 og r_2 er feil, skal være:

$$r_1 = r + (1, 0)$$

$$r_2 = r - (1, 0)$$

b)

For å kunne plotte et vektorfelt må vi først vite hvordan å plotte ved bruk av numpy og matplotlib.pyplot. Det første vi må gjøre er å importere pakkene:

```
import numpy as np
import matplotlib.pyplot as plt
```

Videre trenger vi å definere et koordinatsystem med punkter for å danne figuren. Ved å bruke np.linspace kan vi danne en liste med et bestemt antall punkter, hvor mellomrommet mellom hvert punkt er like stort. Vi setter deretter en liste med punkter langs x -aksen sammen med en liste med punkter i y -aksen for å danne et koordinatsystem.

```
xpoints = np.linspace(-10, 10, 20)
ypoints = np.linspace(-10, 10, 20)
xp, yp = np.meshgrid(xpoints, ypoints)
```

Syntax:

```
np.linspace(minste verdi, største verdi, antall punkter)
np.meshgrid(akse 1, akse 2)
```

Merk: `np.meshgrid` danner et koordinatsystem mellom de bestemte rammene `xpoints` og `ypoints`

Koordinatsystemet vi har nå inneholder ingen informasjon eller verdier. Vi trenger å parametrisere funksjonen vår slik at vi kan legge til ønskede verdier for hvert punkt x og y . Numpy er ganske smart når det kommer til å jobbe med lister, slik at vi kan sette inn parametrisering direkte. Vi parametriserer $r = (x, y)$, og setter da inn i uttrykket for $v(r)$:

```
L = 3
M = 1
r = np.array([xp, yp]) #r = (x, y)
r1 = [r[0] + 1, r[1]] #r1 = r + (1, 0) = (x + 1, y)
r2 = [r[0] - 1, r[1]] #r2 = r - (1, 0) = (x - 1, y)
r1Len = np.sqrt(r1[0]**2 + r1[1]**2) #|r1|
r2Len = np.sqrt(r2[0]**2 + r2[1]**2) #|r2|
r1Hatt = r1/r1Len # r1Hatt = r1/|r1|
r2Hatt = r2/r2Len # r2Hatt = r2/|r1|
F = L/r1Len*r1Hatt - M/r2Len*r2Hatt #Setter inn i v
u = F[0] # F rste koordinat til vektorfeltet
v = F[1] # Andre koordinat til vektorfeltet
```

Merk: Vi bruker `np.array()` og ikke vanlige lister siden numpy har god integrasjon av den typen lister.

Det vi nå står igjen med er u og v som er koordinatene til vektorfeltet basert på koordinatsystemet vi lagde. Vi trenger nå å lage et plote element fra `matplotlib` og legge til verdiene vi har funnet:

```
fig, ax = plt.subplots() #Lager figur element
```

```

ax.quiver(xp, yp, u, v) #Lager vektorpiler
ax.grid() #Skrur p et grid i figuren
plt.show() #Viser figuren vi lagde

```

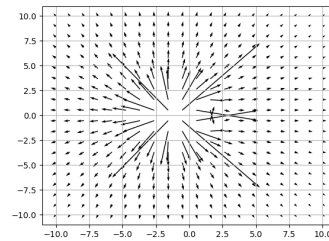
Koden og plottet du sitter igjen med burde se noe slik ut:

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 xpoints = np.linspace(-10, 10, 20)
5 ypoints = np.linspace(-10, 10, 20)
6 xp, yp = np.meshgrid(xpoints, ypoints)
7
8 L = 3
9 M = 1
10 r = np.array([xp, yp]) # r = (x, y)
11 r1 = [r[0] + 1, r[1]] # r1 = r + (1, 0) = (x + 1, y)
12 r2 = [r[0] - 1, r[1]] # r2 = r - (1, 0) = (x - 1, y)
13 r1len = np.sqrt(r1[0]**2 + r1[1]**2) # |r1|
14 r2len = np.sqrt(r2[0]**2 + r2[1]**2) # |r2|
15 r1hatt = r1/r1len # r1hatt = r1/|r1|
16 r2hatt = r2/r2len # r2hatt = r2/|r2|
17 F = L/r1len*r1hatt - M/r2len*r2hatt # Setter inn i v
18 u = F[0] # Første koordinat til vektorfeltet
19 v = F[1] # Andre koordinat til vektorfeltet
20
21 fig, ax = plt.subplots() #Lager figur element
22 ax.quiver(xp, yp, u, v) #Lager vektorpiler
23 ax.grid() #Skrur på et grid i figuren
24 plt.show() #Viser figuren vi lagde

```

(a) Kodesnutt



(b) Figur

Figure 1: Plotting av vektorfelt

c)