Christophe Blomsen
chriskbl@student.matnat.uio.no

23. april 2020

Innhold

Figurer

Kode

a)

Lorem ipsum

```python
import scipy.io as sio
import numpy as np
import matplotlib.pyplot as plt

data = sio.loadmat('data.mat')
x = data.get('x')
y = data.get('y')
u = data.get('u')
v = data.get('v')
xit = data.get('xit')
yit = data.get('yit')

print(np.shape(x))
print(np.shape(y))
print(np.shape(u))
print(np.shape(v))
print(np.shape(xit))
print(np.shape(yit))

print(x)
print(y)
```

Utskrift til terminalen blir

```
x shape is (201, 194)
y shape is (201, 194)
u shape is (201, 194)
v shape is (201, 194)
xit shape is (1, 194)
yit shape is (1, 194)
[[ 0. 0.5 1. ... 95.5 96. 96.5]
[ 0. 0.5 1. ... 95.5 96. 96.5]
[ 0. 0.5 1. ... 95.5 96. 96.5]
...
[ 0. 0.5 1. ... 95.5 96. 96.5]
[ 0. 0.5 1. ... 95.5 96. 96.5]
[ 0. 0.5 1. ... 95.5 96. 96.5]]
[[-50. -50. -50. ... -50. -50. -50. ]
[-49.5 -49.5 -49.5 ... -49.5 -49.5 -49.5]
[-49. -49. -49. ... -49. -49. -49. ]
...
[ 49. 49. 49. ... 49. 49. 49. ]
[ 49.5 49.5 49.5 ... 49.5 49.5 49.5]
[ 50. 50. 50. ... 50. 50. 50. ]]
```

Ser da at griddet i xy-planet har et regulært intervall på 0.5 mm i begge rettninger

b)

Lorem ipsum

```
velocity = np.sqrt(u**2 + v**2)

plt.subplot(2, 1, 1)
plt.plot(xit, yit, "k*")
water_bender = plt.contourf(x, y, velocity, np.linspace(0, 500, 100))
plt.colorbar(water_bender)

plt.subplot(2, 1, 2)
plt.plot(xit, yit, "k*")
air_bender = plt.contourf(x, y, velocity, np.linspace(1000, 5000, 100))
plt.colorbar(air_bender)

plt.savefig("oppgave_b.png")
plt.show()
```

c)

Lorem ipsum

```
def rectangle(x1, x2, y1, y2):
    position1 = (x[x2, x1], y[x2, y1])
    position2 = (x[y2, y1], y[y2, y1])

    # Bottom
    plt.plot([position1[0], position2[0]], [position1[1], position1[1]], "r")

    # Right
    plt.plot([position2[0], position2[0]], [position1[1], position2[1]], "g")

    # Top
    plt.plot([position1[0], position2[0]], [position2[1], position2[1]], "b")

    # Left
    plt.plot([position1[0], position1[0]], [position1[1], position2[1]], "k")


def draw_rectangles():
    rectangle1_values = [35, 160, 70, 170]
    rectangle(rectangle1_values[0], rectangle1_values[1],
              rectangle1_values[2], rectangle1_values[3])

    rectangle2_values = [35, 85, 70, 100]
    rectangle(rectangle2_values[0], rectangle2_values[1],
              rectangle2_values[2], rectangle2_values[3])

    rectangle3_values = [35, 50, 70, 60]
    rectangle(rectangle3_values[0], rectangle3_values[1],
              rectangle3_values[2], rectangle3_values[3])


plt.plot(xit, yit, "k*")
num_skip = 5
plt.quiver(x[::num_skip, ::num_skip], y[::num_skip, ::num_skip],
           u[::num_skip, ::num_skip], v[::num_skip, ::num_skip])

plt.title("Oppgave c)")
plt.xlabel("x")
plt.ylabel("y")

plt.savefig("oppgave_c.png")
```

c)

Lorem ipsum

```
dudx = np.gradient(u, axis=0)
dvdy = np.gradient(v, axis=1)

divergence = dudx + dvdy
print(f"The divergence is {divergence}")

plt.contourf(x, y, divergence)
plt.colorbar()
plt.title("Oppgave d)")
plt.savefig("oppgave_d.png")
```

e)

Lorem ipsum

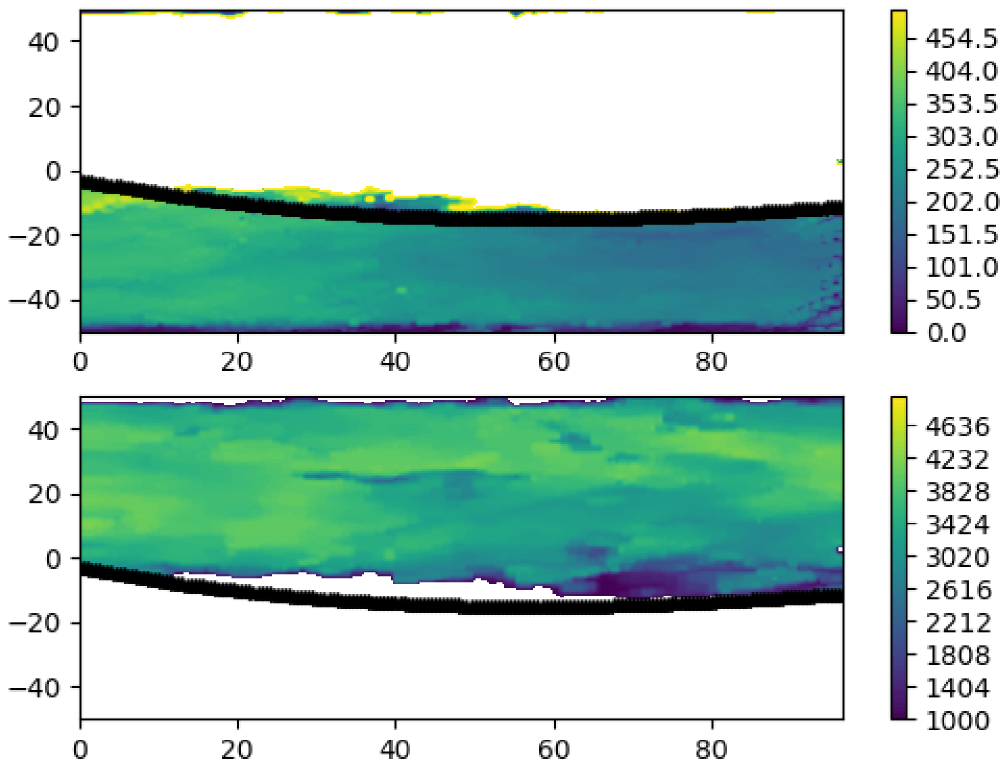
f)

Lorem ipsum

g)

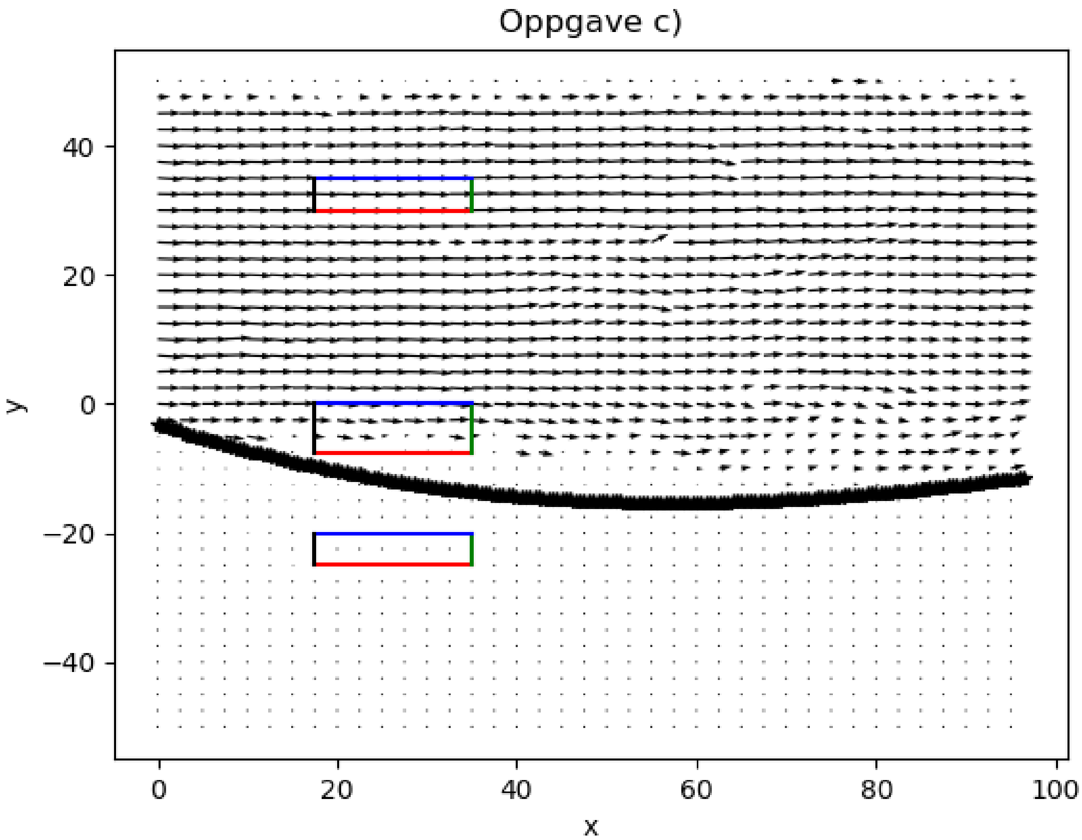
Lorem ipsum

Figur 1: ¡++¿

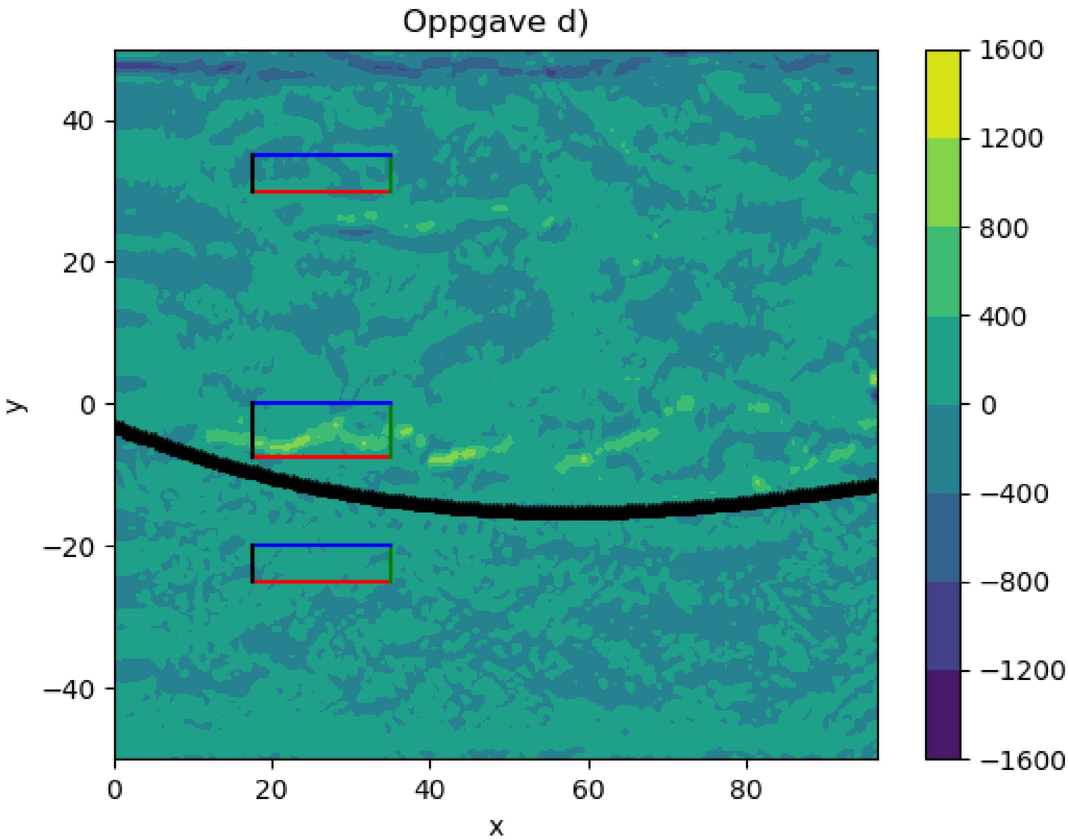


40
20
0
−20
−40
0
20
40
60
80
454.5
404.0
353.5
303.0
252.5
202.0
151.5
101.0
50.5
0.0
4636
4232
3828
3424
3020
2616
2212
1808
1404
1000

Figur 2: Graf til ¡++¿

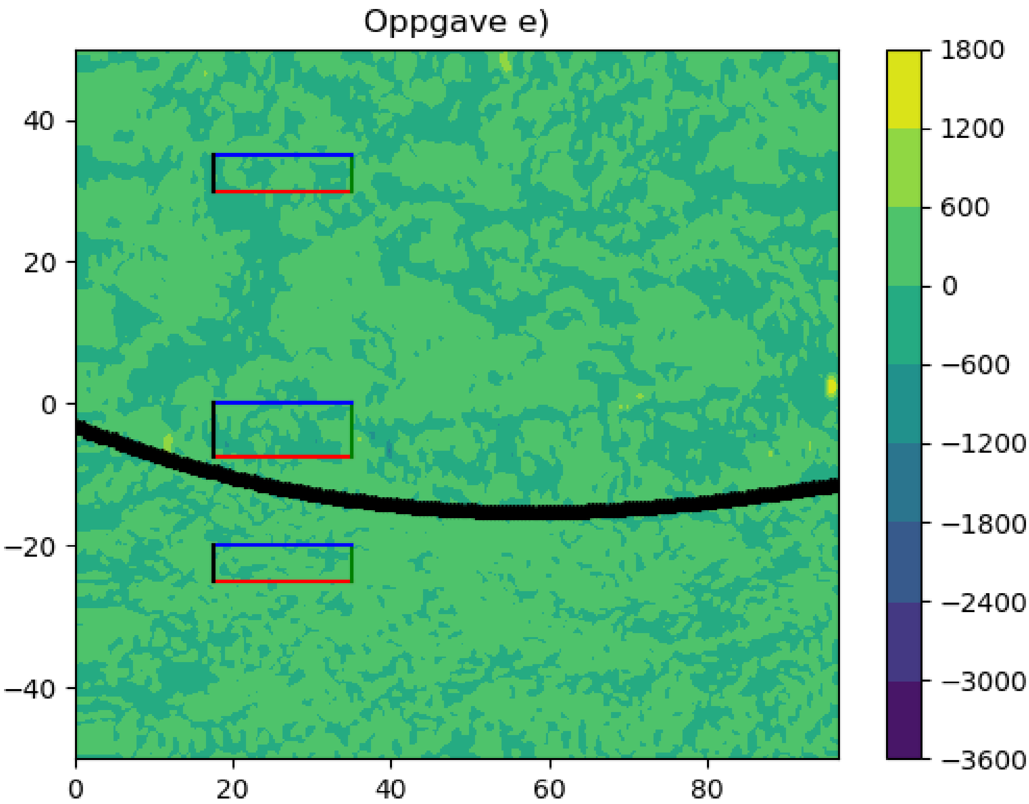


Figur 3: ¡++¿



Oppgave d)
40
20
0
−20
−40
y
0
20
40
60
80
x
1600
1200
800
400
0
−400
−800
−1200
−1600

Figur 4: ¡++¿


Oppgave e)
40
20
0
−20
−40
0
20
40
60
80
1800
1200
600
0
−600
−1200
−1800
−2400
−3000
−3600