

MEK1100 - Oblig 2

Hans-Petter Harveg

April 2018

a)

I denne oppgaven har jeg lastet inn datafilen og skrevet testfunksjoner for de gitte oppgavene. Spesifikt har jeg skrevet følgende funksjoner

- `get_matrix_sizes()`: skriver ut dimensjonene på matrisen ved hjelp av funksjonen `shape()`.
- `test_pixel_spread()`: sjekker at bredden mellom pixlene er 0.5. Fordi vi har en matrise med x-verdier og en med y-verdier sjekker jeg begge matrisene. Funksjonen returnerer *False* dersom den finner en verdi $\Delta x \neq 0.5$.
- `test_y_range()`: sjekker at dataen spenner høyden på røret.

Koden i sin helhet ligger under [kildekode](#).

b)

I denne oppgaven har jeg laget to plot

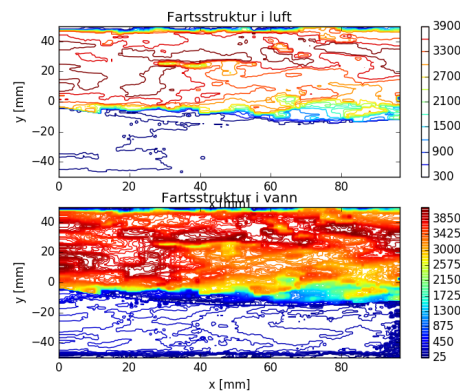


Figure 1: Fartstruktur i hennholdsvis luft og vann.

c)

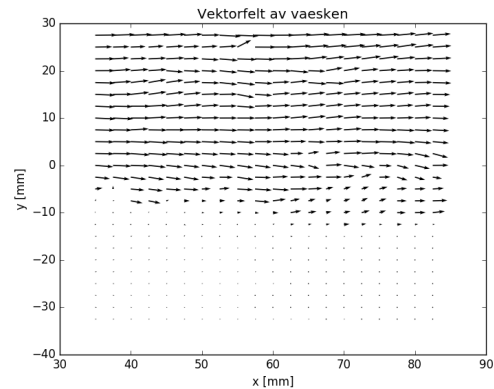


Figure 2: Contour for $N = 1/2$.

Kildekode

```

1 import scipy.io as sio
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 class DataSet():
6     def __init__(self, filename):
7         filedata = sio.loadmat(filename)
8         self.data = {"x":filedata.get("x"), "y":filedata.get("y"),
9 "u":filedata.get("u"), "v":filedata.get("v"), "xit":filedata.
10 get("xit"), "yit":filedata.get("yit")}
11
12     def get_data(self):
13         return (self.data["x"], self.data["y"], self.data["u"],
14 self.data["v"], self.data["xit"], self.data["yit"])
15
16     def get_matrices_sizes(self):
17         for key, value in self.data.items():
18             [a, b] = np.shape(value)
19             print "Size of [", key, "]:", b, a
20
21         return True
22
23     def test_pixel_spread(self):
24         [x, y, u, v, xit, yit] = self.get_data()
25
26         for iy in range(201):
27             prev_x = 0

```

```

28         prev_y = 0
29         for ix in range(1,192):
30             if x[iy][ix]-prev_x != 0.5:
31                 return False
32             prev_x = x[iy][ix]
33
34             if y[iy][ix]-prev_y != 0.5:
35                 return False
36             prev_y = y[iy][ix]
37
38         return True
39
40
41     def test_y_range(self):
42         yit = 0
43
44
45     def plot_contours(self):
46         [x, y, u, v, xit, yit] = self.get_data()
47         c = np.sqrt(u**2 + v**2)
48         plt.subplot(2,1,1)
49         plt.contour(x, y, c, 15)
50         plt.colorbar()
51         dataset.render_plot("Fartsstruktur i luft", "x [mm]", "y [
mm]")
52         plt.subplot(2,1,2)
53         plt.contour(x, y, c, 200)
54         plt.colorbar()
55         dataset.render_plot("Fartsstruktur i vann", "x [mm]", "y [
mm]")
56         plt.show()
57
58
59     def plot_quiver(self, x_0, x_1, y_0, y_1, step):
60         [x, y, u, v, xit, yit] = self.get_data()
61         plt.quiver(x[x_0:x_1:step, y_0:y_1:step], y[x_0:x_1:step,
y_0:y_1:step], u[x_0:x_1:step, y_0:y_1:step], v[x_0:x_1:step,
y_0:y_1:step])
62         dataset.render_plot("Vektorfelt av vaesken", "x [mm]", "y [
mm]")
63         plt.show()
64
65
66     def render_plot(self, title, xlabel, ylabel):
67         plt.title(title)
68         plt.xlabel(xlabel)
69         plt.ylabel(ylabel)
70
71
72     # Problem a)
73     dataset = DataSet("data.mat")
74     dataset.get_matrices_sizes()
75     dataset.test_pixel_spread()
76     dataset.test_y_range()
77
78     # Problem b)
79     #dataset.plot_contours()

```

```
80
81 # Problem c)
82 dataset.plot_quiver(35, 160, 70, 170, 5)
83 #dataset.plot_quiver(35, 85, 70, 100, 5)
84 #dataset.plot_quiver(35, 50, 70, 60, 5)
85
86
87 # Problem d)
88
89 # Problem e)
90
91 # Problem f)
92
93 # Problem g)
```