

Christophe Blomsen  
[chriskbl@student.matnat.uio.no](mailto:chriskbl@student.matnat.uio.no)

2. mai 2020

## Innhold

a)	1
b)	1
c)	2
d)	3
e)	3
f)	4
g)	5
1 <a href="#">Appendiks</a>	8

## Figurer

1 <a href="#">Graf til oppgave b</a>	7
2 <a href="#">Graf til oppgave c</a>	8
3 <a href="#">Graf til oppgave d</a>	9
4 <a href="#">Graf til oppgave e</a>	10

## Kode

1 <a href="#">Kode oppgave a)</a>	1
2 <a href="#">Kode oppgave b</a>	2
3 <a href="#">Kode til oppgave c</a>	2
4 <a href="#">Oppgave d</a>	3

a)

All kode vil ligge i samme python fil så de relevante kode snuttene er tilgjengelig på de tilsvarende deloppgavene. Hele filen kan finnes i Appendikset.

Kode 1: Kode oppgave a)

```
1 import scipy.io as sio
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 data = sio.loadmat('data.mat')
6 x = data.get('x')
7 y = data.get('y')
8 u = data.get('u')
9 v = data.get('v')
10 xit = data.get('xit')
11 yit = data.get('yit')
12
13 print(f"x shape er {np.shape(x)}")
14 print(f"y shape er {np.shape(y)}")
15 print(f"u shape er {np.shape(u)}")
16 print(f"v shape er {np.shape(v)}")
17 print(f"xit shape er {np.shape(xit)}")
18 print(f"yit shape er {np.shape(yit)}")
19
20 print(x)
21 print(y)
```

Utskrift til terminalen blir

```
x shape er (201, 194)
y shape er (201, 194)
u shape er (201, 194)
v shape er (201, 194)
xit shape er (1, 194)
yit shape er (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 retninger. Samt at  $y$ -koordinatene spenner ut hele diameteren til røret.

b)

Observerer fra presentasjonen av eksperimentet at det er væskefase i den nedre halvdelen av røret og gassfase i den andre halvdelen.

Kode 2: Kode oppgave b

```
1 velocity = np.sqrt(u**2 + v**2)
2
3 plt.subplot(2, 1, 1)
4 plt.plot(xit, yit, "k*")
5 water_bender = plt.contourf(x, y, velocity, np.linspace(0, 500, 100))
6 plt.colorbar(water_bender)
7
8 plt.subplot(2, 1, 2)
9 plt.plot(xit, yit, "k*")
10 air_bender = plt.contourf(x, y, velocity, np.linspace(1000, 5000, 100))
11 plt.colorbar(air_bender)
12
13 plt.savefig("oppgave_b.png")
14 plt.show()
```

Det produserer følgende [plot](#)

c)

Velger å bruke vært femte element i pilplottet

Kode 3: Kode til oppgave c

```
1 def rectangle(x1, x2, y1, y2):
2     position1 = (x[x2, x1], y[x2, y1])
3     position2 = (x[y2, y1], y[y2, y1])
4
5     # Bottom
6     plt.plot([position1[0], position2[0]], [position1[1], position1[1]], "r")
7
8     # Right
9     plt.plot([position2[0], position2[0]], [position1[1], position2[1]], "g")
10
11     # Top
12     plt.plot([position1[0], position2[0]], [position2[1], position2[1]], "b")
13
14     # Left
15     plt.plot([position1[0], position1[0]], [position1[1], position2[1]], "k")
16
17
18 def draw_rectangles():
19     rectangle1_values = [34, 159, 69, 169]
20     rectangle(rectangle1_values[0], rectangle1_values[1],
21               rectangle1_values[2], rectangle1_values[3])
22
23     rectangle2_values = [34, 84, 69, 100]
24     rectangle(rectangle2_values[0], rectangle2_values[1],
25               rectangle2_values[2], rectangle2_values[3])
26
27     rectangle3_values = [34, 49, 69, 59]
28     rectangle(rectangle3_values[0], rectangle3_values[1],
29               rectangle3_values[2], rectangle3_values[3])
30
31 draw_rectangles()
32 plt.plot(xit, yit, "k*")
33 num_skip = 5
```

```

34 plt.quiver(x[::num_skip, ::num_skip], y[::num_skip, ::num_skip],
35            u[::num_skip, ::num_skip], v[::num_skip, ::num_skip])
36
37 plt.title("Oppgave c)")
38 plt.xlabel("x")
39 plt.ylabel("y")
40
41 plt.savefig("oppgave_c.png")

```

Denne kodesnutten produserer følgende [plot](#)

d)

I numpy pakken til python så finnes det flere finne funksjoner, i denne oppgaven så blir numpy.gradient funksjonen brukt. Det den gjør er å regne ut gradienten til arrays. Hvis man også bruker keyword argumentet axis så kan man velge hvilken av komponente du vil ha. I kodesnutten under er da dette oppnådd med at vi i *dudx* kun trekker ut *x*-komponente fra gradienten til *u* tilsvarende for *dvdy*.

Kode 4: Oppgave d

```

1 dudx = np.gradient(u, 0.5, axis=0)
2 dvdy = np.gradient(v, 0.5, axis=1)
3
4 divergence = dudx + dvdy
5 print(f"Divergensen er {divergence}")
6
7 plt.contourf(x, y, divergence)
8 plt.colorbar()
9 plt.title("Oppgave d)")
10 plt.savefig("oppgave_d.png")

```

Denne kodesnutten produserer da følgende [plot](#).

Divergensen til  $u\mathbf{i} + v\mathbf{j} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$ , så det mangler *w* komponenten for at det skal være likt med  $\nabla \cdot \mathbf{v}$

Siden gassen og væska er inkompressible så betyr det at  $\nabla \cdot \mathbf{v} = 0$ . Dette medfører

$$\nabla \cdot \mathbf{v} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$$

$$\frac{\partial w}{\partial z} = -\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)$$

Da vil hastighetskomponenten

$$w = \int \left( \frac{\partial v}{\partial y} + \frac{\partial u}{\partial x} \right) dz$$

Dette integralet kan ikke være 0 derfor må divergensen til  $\mathbf{v}$  være forskjellig fra divergensen til  $u\mathbf{i} + v\mathbf{j}$

e)

numpy.gradient blir brukt på samme måte i denne oppgaven for å finne virvlingen.

```

1 dudy = np.gradient(u, 0.5, axis=0)
2 dvdx = np.gradient(v, 0.5, axis=1)
3
4 curl_v = dvdx - dudy
5
6 curl_plot = plt.contourf(x, y, curl_v)
7 plt.streamplot(x, y, u, v, color="orange")
8 plt.colorbar()
9
10 plt.title("Oppgave e)")
11
12 plt.savefig("oppgave-e.png")
13 plt.show()

```

Konturplottet av denne virvlingskomponenten kan finnes [her](#). Observerer at strømmingen skaper sirkulasjon mellom gass- og væskefasen, spesielt rundt det miderste rektangelet. Ser også at strømmingen trekkes til veggen på grunn av friksjon, og at det er større friksjon i væskefasen.

f)

```

1 def line_integral(x1, y1, x2, y2):
2     side1 = 0
3     side2 = 0
4     side3 = 0
5     side4 = 0
6     dx = 0.5
7     dy = 0.5
8     for k in u[y1, x1:x2+1]:
9         side1 += k*dx
10    for k in v[x2, y1:y2+1]:
11        side2 += k*dy
12    for k in u[y2, x1:x2+1]:
13        side3 -= k*dx
14    for k in v[x1, y1:y2+1]:
15        side4 -= k*dy
16    summation = side1 + side2 + side3 + side4
17    return summation, side1, side2, side3, side4
18
19 def surface_integral(x1, y1, x2, y2):
20     integral = 0
21     dx = 0.5
22     dy = 0.5
23     for i in range(x1, x2+1):
24         for j in range(y1, y2+1):
25             integral += curl_v[j, i]*dx*dy
26     return integral
27
28 line_integral_1, side1_1, side2_1, side3_1, side4_1 = line_integral(34, 159, 69, 169)
29 line_integral_2, side1_2, side2_2, side3_2, side4_2 = line_integral(34, 84, 69, 99)
30 line_integral_3, side1_3, side2_3, side3_3, side4_3 = line_integral(34, 49, 69, 59)
31
32 integral_1 = surface_integral(34, 159, 69, 169)
33 integral_2 = surface_integral(34, 84, 69, 99)
34 integral_3 = surface_integral(34, 49, 69, 59)
35
36 print(f"Sirkulasjonen til rektangel 1 ble {line_integral_1}")
37 print(f"Sirkulasjonen til rektangel 2 ble {line_integral_2}")
38 print(f"Sirkulasjonen til rektangel 3 ble {line_integral_3}")
39
40 print(f"""Rektangel 1 har side verdier: Side 1={side1_1},
41       Side 2={side2_1}, Side 3={side3_1}, Side 4={side4_1}""")
42 print(f"""Rektangel 2 har side verdier: Side 1={side1_2},
43       Side 2={side2_2}, Side 3={side3_2}, Side 4={side4_2}""")
44 print(f"""Rektangel 3 har side verdier: Side 1={side1_3},
45       Side 2={side2_3}, Side 3={side3_3}, Side 4={side4_3}""")

```

Sirkulasjonen til rektangel 1 ble 1796.013421486005  
 Sirkulasjonen til rektangel 2 ble -60206.56400779366  
 Sirkulasjonen til rektangel 3 ble -143.18708039114364  
 Rektangel 1 har side verdier: Side 1=70100.52387861427,  
 Side 2=-100.99982042140701, Side 3=-68332.85609978675, Side 4=129.34546307988458  
 Rektangel 2 har side verdier: Side 1=198.47559740489203,  
 Side 2=919.0821556116496, Side 3=-61243.46477849595, Side 4=-80.65698231424645  
 Rektangel 3 har side verdier: Side 1=5133.347850903836,  
 Side 2=175.1650519009061, Side 3=-5410.039721925995, Side 4=-41.660261269891

Som nevnt i oppgave e så ser vi nå med hjelp av kurveintegral at sirkulasjon er størst i det midterste rektangelet og minst i det nederste.

g)

```

1 def gauss(x1, y1, x2, y2):
2     side1 = 0
3     side2 = 0
4     side3 = 0
5     side4 = 0
6     dx = 0.5
7     dy = 0.5
8     dz = 1
9
10    for k in v[y1, x1:x2+1]:
11        side1 -= k*dx*dz
12
13    for k in u[x2, y1:y2+1]:
14        side2 += k*dy*dz
15
16    for k in v[y2, x1:x2+1]:
17        side3 += k*dx*dz
18
19    for k in u[x1, y1:y2+1]:
20        side4 -= k*dy*dz
21
22    sumation = side1 + side2 + side3 + side4
23    return sumation, side1, side2, side3, side4
24
25 gauss_1, gauss_side1_1, gauss_side2_1, gauss_side3_1, gauss_side4_1 = gauss(34, 159, ↵
69, 169)
26 gauss_2, gauss_side1_2, gauss_side2_2, gauss_side3_2, gauss_side4_2 = gauss(34, 84, ↵
69, 99)
27 gauss_3, gauss_side1_3, gauss_side2_3, gauss_side3_3, gauss_side4_3 = gauss(34, 49, ↵
69, 59)
28
29 print(f"Fluksen til kurveintegral 1 ble {gauss_1}")
30 print(f"Fluksen til kurveintegral 2 ble {gauss_2}")
31 print(f"Fluksen til kurveintegral 3 ble {gauss_3}")
32
33 print(f"""Rektangel 1 har side verdier: Side 1={gauss_side1_1},
34         Side 2={gauss_side2_1}, Side 3={gauss_side3_1}, Side 4={gauss_side4_1}""")
35 print(f"""Rektangel 2 har side verdier: Side 1={gauss_side1_2},
36         Side 2={gauss_side2_2}, Side 3={gauss_side3_2}, Side 4={gauss_side4_2}""")
37 print(f"""Rektangel 3 har side verdier: Side 1={gauss_side1_3},
38         Side 2={gauss_side2_3}, Side 3={gauss_side3_3}, Side 4={gauss_side4_3}""")

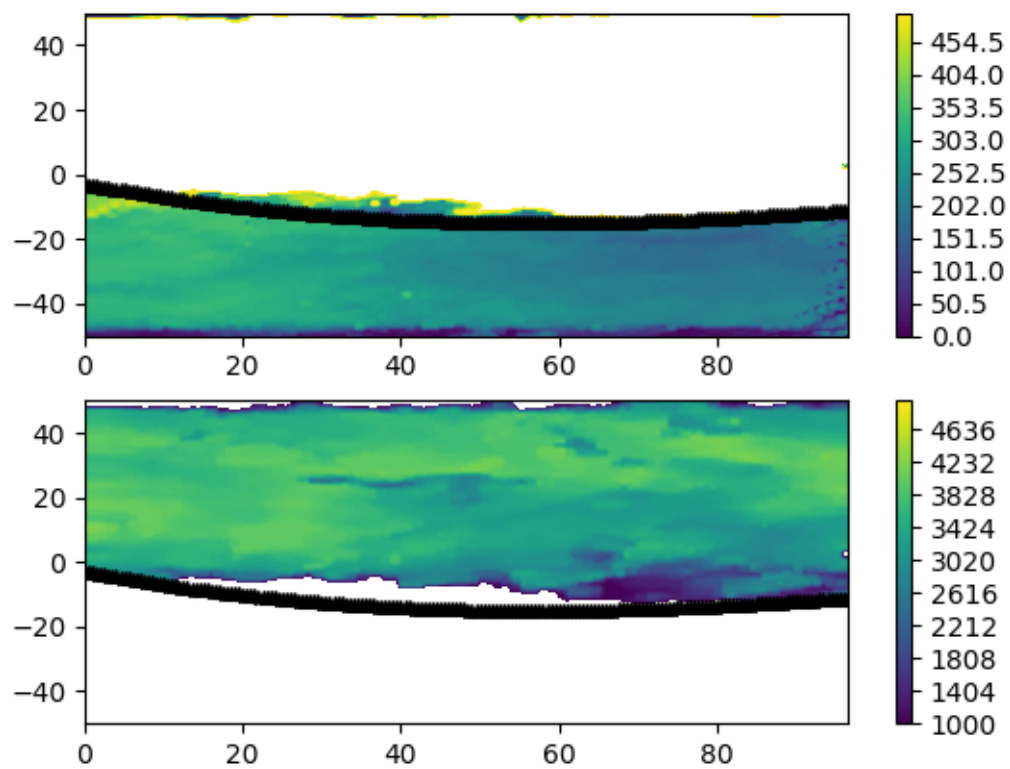
```

Fluksen til kurveintegral 1 ble 104.8526049082102  
Fluksen til kurveintegral 2 ble -6476.93918209796  
Fluksen til kurveintegral 3 ble -124.56866604496236  
Rektangel 1 har side verdier: Side 1=1556.8679439413959,  
Side 2=21664.567474322168, Side 3=-2059.6771847938708, Side 4=-21056.905628561482  
Rektangel 2 har side verdier: Side 1=-5187.564033067891,  
Side 2=14782.532896182345, Side 3=-4074.0522144394345, Side 4=-11997.85583077298  
Rektangel 3 har side verdier: Side 1=-195.5701479258336,  
Side 2=1536.8217966413547, Side 3=284.9436464350764, Side 4=-1750.7639611955597

Anvender gauss sats så ser vi at fluksen i  $z$ -retning er like stor bare motsatt retning. Disse resultatene ser fornuftige ut siden det er side 2 og 4 som har størst verdi.

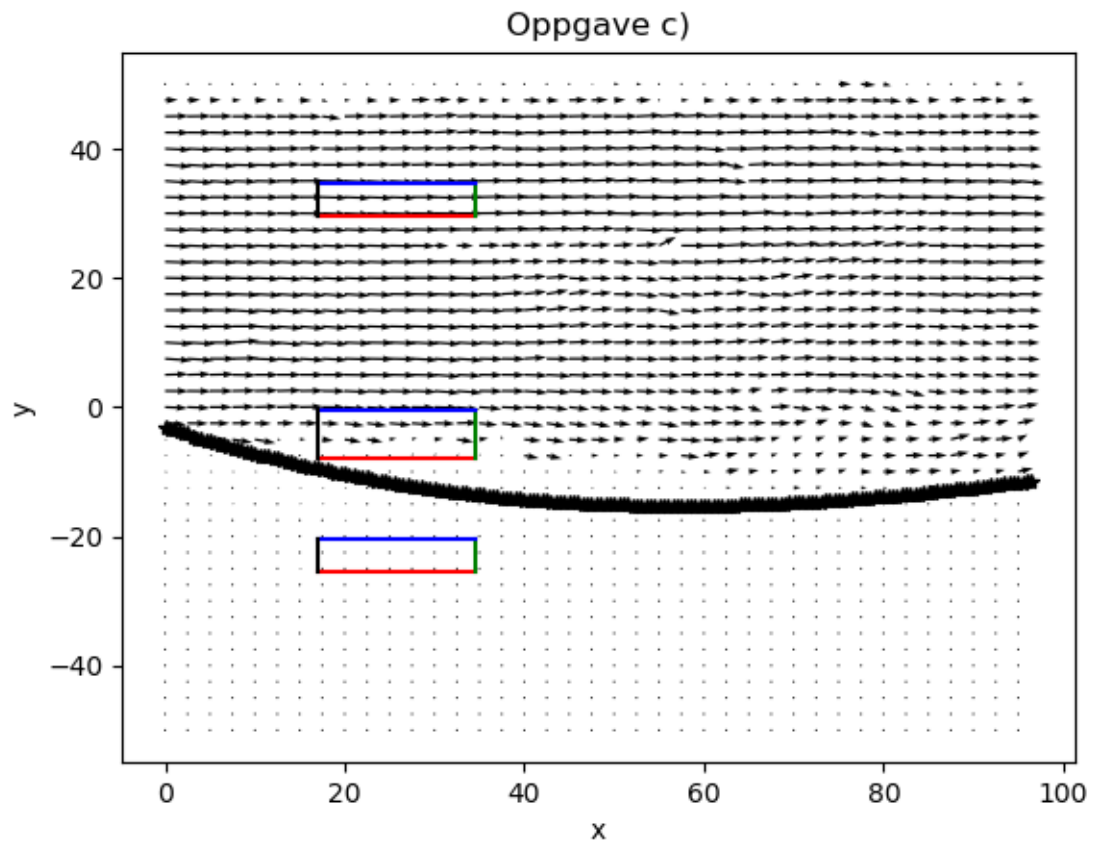
## Plots

Figur 1: Graf til oppgave [b](#)

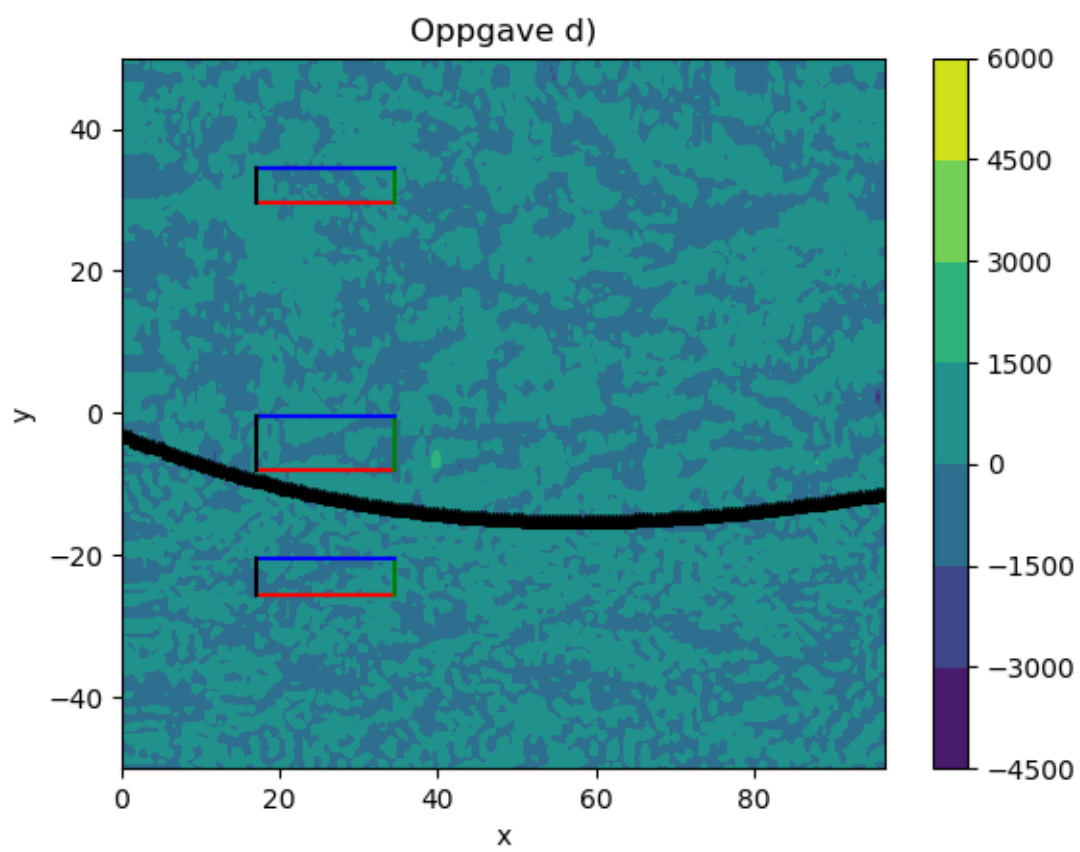




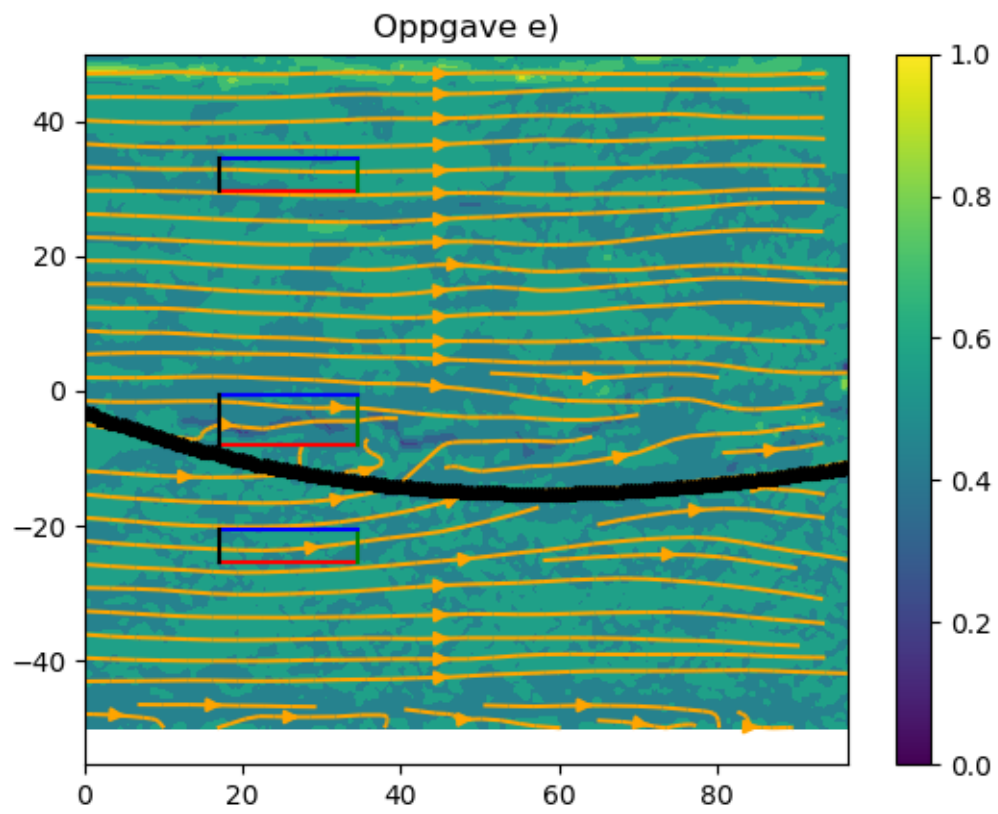
Figur 2: Graf til oppgave c



Figur 3: Graf til oppgave d



Figur 4: Graf til oppgave e



# 1 Appendiks

```
1 import scipy.io as sio
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 #Oppgave a
6 data = sio.loadmat('data.mat')
7 x = data.get('x')
8 y = data.get('y')
9 u = data.get('u')
10 v = data.get('v')
11 xit = data.get('xit')
12 yit = data.get('yit')
13
14 print(f"x shape er {np.shape(x)}")
15 print(f"y shape er {np.shape(y)}")
16 print(f"u shape er {np.shape(u)}")
17 print(f"v shape er {np.shape(v)}")
18 print(f"xit shape er {np.shape(xit)}")
19 print(f"yit shape er {np.shape(yit)}")
20
21 print(x)
22 print(y)
23
24 #Oppgave b
25 velocity = np.sqrt(u**2 + v**2)
26
27 plt.subplot(2, 1, 1)
28 plt.plot(xit, yit, "k*")
29 water_bender = plt.contourf(x, y, velocity, np.linspace(0, 500, 100))
30 plt.colorbar(water_bender)
31
32 plt.subplot(2, 1, 2)
33 plt.plot(xit, yit, "k*")
34 air_bender = plt.contourf(x, y, velocity, np.linspace(1000, 5000, 100))
35 plt.colorbar(air_bender)
36
37 plt.savefig("oppgave_b.png")
38 plt.show()
39
40 #Oppgave c
41 def rectangle(x1, x2, y1, y2):
42     position1 = (x[x2, x1], y[x2, y1])
43     position2 = (x[y2, y1], y[y2, y1])
44
45     # Bottom
46     plt.plot([position1[0], position2[0]], [position1[1], position1[1]], "r")
47
48     # Right
49     plt.plot([position2[0], position2[0]], [position1[1], position2[1]], "g")
50
51     # Top
52     plt.plot([position1[0], position2[0]], [position2[1], position2[1]], "b")
53
54     # Left
55     plt.plot([position1[0], position1[0]], [position1[1], position2[1]], "k")
56
57 def draw_rectangles():
58     rectangle1_values = [34, 159, 69, 169]
59     rectangle(rectangle1_values[0], rectangle1_values[1],
60               rectangle1_values[2], rectangle1_values[3])
61
62     rectangle2_values = [34, 84, 69, 99]
63     rectangle(rectangle2_values[0], rectangle2_values[1],
64               rectangle2_values[2], rectangle2_values[3])
65
66     rectangle3_values = [34, 49, 69, 59]
67     rectangle(rectangle3_values[0], rectangle3_values[1],
68               rectangle3_values[2], rectangle3_values[3])
```

```

69
70
71 draw_rectangles()
72 plt.plot(xit, yit, "k*")
73 num_skip = 5
74 plt.quiver(x[::num_skip, ::num_skip], y[::num_skip, ::num_skip],
75            u[::num_skip, ::num_skip], v[::num_skip, ::num_skip])
76
77 plt.title("Oppgave c")
78 plt.xlabel("x")
79 plt.ylabel("y")
80
81 plt.savefig("oppgave-c.png")
82
83 #Oppgave d
84
85 dux = np.gradient(u, 0.5, axis=1)
86 dvy = np.gradient(v, 0.5, axis=0)
87
88 divergence = dux + dvy
89 print(f"Divergensen er {divergence}")
90
91 plt.contourf(x, y, divergence)
92 plt.colorbar()
93 plt.title("Oppgave d")
94 plt.savefig("oppgave-d.png")
95 plt.close()
96
97 #Oppgave e
98 dudy = np.gradient(u, 0.5, axis=0)
99 dvdx = np.gradient(v, 0.5, axis=1)
100
101 curl_v = dvdx - dudy
102
103 curl_plot = plt.contourf(x, y, curl_v)
104 plt.streamplot(x, y, u, v, color="orange")
105 plt.colorbar()
106
107 plt.title("Oppgave e")
108
109 plt.savefig("oppgave-e.png")
110 plt.show()
111
112 #Oppgave f
113
114 def line_integral(x1, y1, x2, y2):
115     side1 = 0
116     side2 = 0
117     side3 = 0
118     side4 = 0
119     dx = 0.5
120     dy = 0.5
121
122     for k in u[y1, x1:x2+1]:
123         side1 += k*dx
124
125     for k in v[x2, y1:y2+1]:
126         side2 += k*dy
127
128     for k in u[y2, x1:x2+1]:
129         side3 -= k*dx
130
131     for k in v[x1, y1:y2+1]:
132         side4 -= k*dy
133
134     sumation = side1 + side2 + side3 + side4
135     return sumation, side1, side2, side3, side4
136
137
138 def surface_integral(x1, y1, x2, y2):
139     integral = 0
140     dx = 0.5
141     dy = 0.5

```

```

142     for i in range(x1, x2+1):
143         for j in range(y1, y2+1):
144             integral += curl_v[j, i]*dx*dy
145
146     return integral
147
148
149 line_integral_1, side1_1, side2_1, side3_1, side4_1 = line_integral(34, 159, 69, 169)
150 line_integral_2, side1_2, side2_2, side3_2, side4_2 = line_integral(34, 84, 69, 99)
151 line_integral_3, side1_3, side2_3, side3_3, side4_3 = line_integral(34, 49, 69, 59)
152
153 integral_1 = surface_integral(34, 159, 69, 169)
154 integral_2 = surface_integral(34, 84, 69, 99)
155 integral_3 = surface_integral(34, 49, 69, 159)
156
157 print(f"Sirkulasjonen til rektangel 1 ble {line_integral_1}")
158 print(f"Sirkulasjonen til rektangel 2 ble {line_integral_2}")
159 print(f"Sirkulasjonen til rektangel 3 ble {line_integral_3}")
160
161 print(f"""Rektangel 1 har side verdier: Side 1={side1_1},
162       Side 2={side2_1}, Side 3={side3_1}, Side 4={side4_1}""")
163 print(f"""Rektangel 2 har side verdier: Side 1={side1_2},
164       Side 2={side2_2}, Side 3={side3_2}, Side 4={side4_2}""")
165 print(f"""Rektangel 3 har side verdier: Side 1={side1_3},
166       Side 2={side2_3}, Side 3={side3_3}, Side 4={side4_3}""")
167
168 #Oppgave g
169 def gauss(x1, y1, x2, y2):
170     side1 = 0
171     side2 = 0
172     side3 = 0
173     side4 = 0
174     dx = 0.5
175     dy = 0.5
176
177     for k in v[y1, x1:x2+1]:
178         side1 -= k*dx
179
180     for k in u[y1:y2+1, x2]:
181         side2 += k*dy
182
183     for k in v[y2, x1:x2+1]:
184         side3 += k*dx
185
186     for k in u[y1:y2+1, x1]:
187         side4 -= k*dy
188
189     summation = side1 + side2 + side3 + side4
190     return summation, side1, side2, side3, side4
191
192
193 gauss_1, gauss_side1_1, gauss_side2_1, gauss_side3_1, gauss_side4_1 = gauss(34, 159, ↵
194     69, 169)
195 gauss_2, gauss_side1_2, gauss_side2_2, gauss_side3_2, gauss_side4_2 = gauss(34, 84, ↵
196     69, 99)
197 gauss_3, gauss_side1_3, gauss_side2_3, gauss_side3_3, gauss_side4_3 = gauss(34, 49, ↵
198     69, 59)
199
200 print(f"Fluksen til kurveintegral 1 ble {gauss_1}")
201 print(f"Fluksen til kurveintegral 2 ble {gauss_2}")
202 print(f"Fluksen til kurveintegral 3 ble {gauss_3}")
203
204 print(f"""Rektangel 1 har side verdier: Side 1={gauss_side1_1},
205       Side 2={gauss_side2_1}, Side 3={gauss_side3_1}, Side 4={gauss_side4_1}""")
206 print(f"""Rektangel 2 har side verdier: Side 1={gauss_side1_2},
207       Side 2={gauss_side2_2}, Side 3={gauss_side3_2}, Side 4={gauss_side4_2}""")
208 print(f"""Rektangel 3 har side verdier: Side 1={gauss_side1_3},
209       Side 2={gauss_side2_3}, Side 3={gauss_side3_3}, Side 4={gauss_side4_3}""")

```