

# Datorlab. 1: Meteorologi

## FFM234 Vektorfält och klassisk fysik

Jonatan Haraldsson      Oscar Lindberg

Chalmers tekniska högskola  
September 2021

## Inledning

Meteorologi är ett vetenskapligt fält där olika aspekter av vädret studeras, såsom temperatur, nederbörd, lufttryck och vind. Stora mängder meteorologiska data samlas kontinuerligt in över hela världen för att analyseras, främst i syfte att bygga matematiska modeller för att förutsäga hur väderförhållandena kommer att utvecklas. Ett centralt verktyg vid dessa analyser är fältteori. Till exempel kan nablaoperatoren ( $\nabla$ ) vara användbar i studiet av divergens och virvlar.

I denna rapport analyseras meteorologiska data från ett område i Nordatlanten, Norska havet och Nordsjön, som enligt figur 1 omfattar Norges västkust, norra England, Skottland, Nordirland och Island. Väderdatan kommer från Meteorologisk institutt, Blindern, Oslo och uppmättes tisdagen den 19:e februari 2002, kl. 18:00 UTC av de många väderstationer som är verksamma i området. I datan ingår mätningar av lufttrycket vid havsnivån och vindvektorer på isobarytan 850 hPa, motsvarande ca 1000–1500 m över havet. Datat är indelat i ett ruttnät med  $\Delta x = \Delta y = 55$  km, där  $x$  är öst-västlig riktning och  $y$  är nord-sydlig riktning.

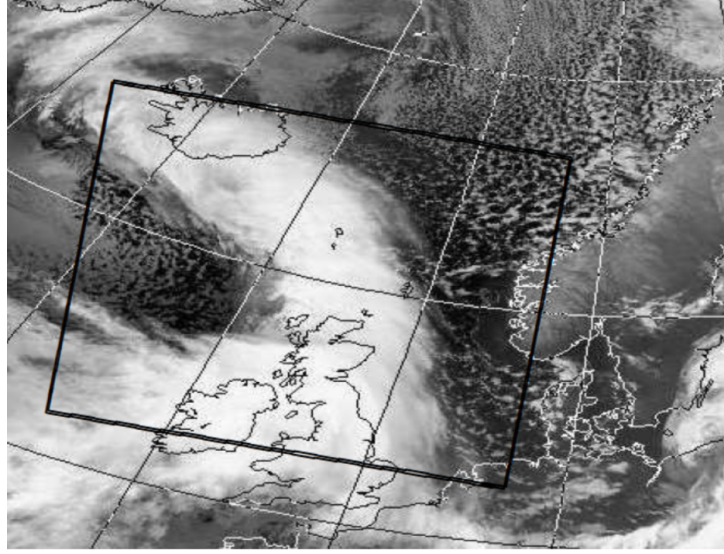
## Metod

I datan finns vindens hastighetskomponenter tillgängliga som två matriser,  $u$  för  $\hat{x}$  och  $v$  för  $\hat{y}$ , och lufttrycket som matrisen  $P$ . Samtliga matriser är av storlek  $30 \times 29$ . Väderdatan analyserades i `Python` med hjälp av biblioteken `numpy` och `matplotlib`. Källkoden presenteras i bilagan.

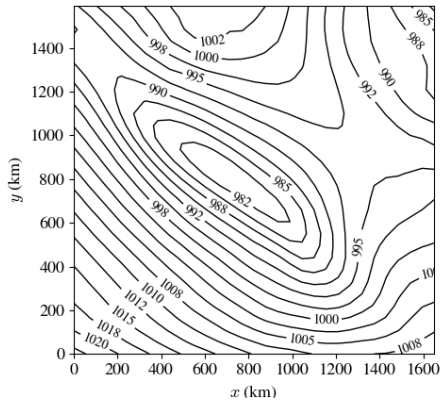
### Tryck- och vindfält

Inledningsvis skapades en "meshgrid" av de två koordinatlistorna  $x$  och  $y$ , där  $x$  består av 31 punkter och  $y$  av 30 punkter med steglängden  $\Delta x = \Delta y = 55$  km i respektive riktning. Isobarer för lufttrycket visualiserades med hjälp av funktionen `contour`, medan vindfältet visualiserades genom `streamplot`. Den maximala vindhastigheten beräknades enligt  $\max_i \sqrt{u_i^2 + v_i^2}$ , via funktionen `max`, där  $u$  och  $v$  är listor med hastighetskomponenter i  $x$ - respektive  $y$ -led.

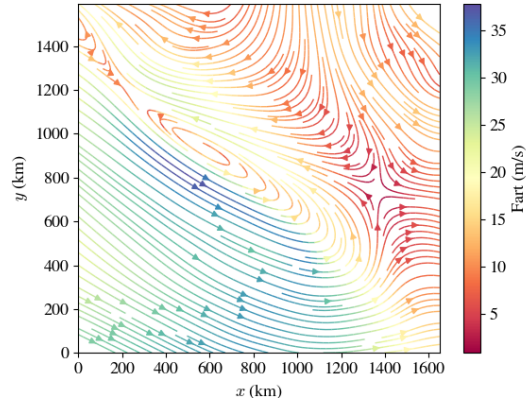




Figur 1: Satellitbild från 16:13 UTC den 19:e februari 2002 från ett område i Nordatlanten, Norska havet och Nordsjön. Notera att bilden är tagen ett par timmar innan väderdatan samlades in.



Figur 2: Lufttryck med isobarer i hPa.



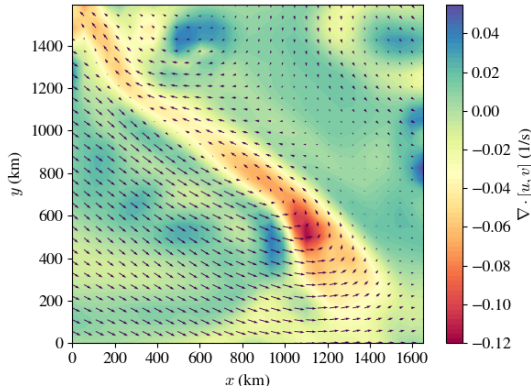
Figur 3: Vindfält.

## Divergens och rotation

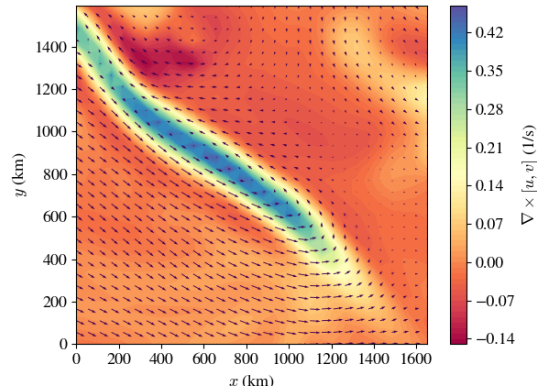
Divergens och rotation av vindfälten erhöles med hjälp av funktionen **gradient**, vilken numeriskt beräknar gradienten med centraldifferens. I ändpunkterna används framåt- respektive bakåtdifferens. Steglängden, med vilken **gradient** approximerar derivatan, sattes till 55 för att matcha  $\Delta x = \Delta y = 55$  km. Funktionen **gradient** applicerades sedan på de två hastighetskomponenterna  $u$  och  $v$ . Sammantaget erhöles två listor  $\vec{\nabla}u = [\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}]$  och  $\vec{\nabla}v = [\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}]$ .

Från  $\vec{\nabla}u$  och  $\vec{\nabla}v$  beräknades divergens enligt  $\vec{\nabla} \cdot [u, v] = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$  och rotation enligt  $\vec{\nabla} \times [u, v] = (\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y})\hat{z}$ . Då de uppmätta vindhastigheterna endast bestod av  $x$ - och  $y$ -komponenter gör





Figur 4: Divergens och vindfält.



Figur 5: Rotation och vindfält.

det att rotationsfältet enbart har en komponent i  $z$ -riktning och därav kan betraktas som ett skalärt fält. Divergens och rotation visualiseras med nivåkurvor i figur 4 och 5 tillsammans med vindens fältlinjer. I koden användes funktionen `contourf` för nivåkurvorna och `quiver` för fältet.

## Resultat och diskussion

I figur 2 syns ett lågtryck i området  $x \in (600, 1000)$  och  $y \in (600, 1000)$  km, vilket på satellitbilden – figur 1 – motsvarar det mest molniga området. I figur 3 syns även att vindarna söder om lågtrycksområdet i allmänhet är kraftigare än norr om det samt att vindhastigheten är större där isobarerna är tätare. Vidare tenderar vindens fältlinjer i figur 3 att vara parallella med isobarerna i figur 2.

På Beaufortskalan motsvarar den maximalt uppmätta vindhastigheten, 37,9 m/s, kategori 13 av 17, vilket benämns som orkan. Wikipedias (*Beauforts skala*, hämtad 2021-09-10) beskrivning av kategori 12: *Stora föremål flyger i luften, fönster blåser in, byggnadsställningar rasar, båtar kastas upp på land, allmän ödeläggelse*. Stormen Gudrun hade enligt SMHI (*Stormar i Sverige*, hämtad 2021-09-10) en maximal byvind på 42 m/s.

Divergensen av vindfältet beskriver nettoflödet av luft ut från en punkt. I områden där divergensen, nettoflödet av luft, är positiv ökar vindhastigheten. I jämförelse med isobarerna i figur 2 sammanfaller områden med negativ divergens och lågtryck. Alltså tenderar områden med lågtryck att dra till sig luft från områden med högre tryck. Störst negativ divergens finns i området  $x \in (1000, 1100)$  och  $y \in (500, 600)$  km där isobarerna i figur 2 går från täta till glesa.

Positiv rotation svarar mot virvlar moturs i vindfältet, vilket visualiseras med det blåa området i figur 5. Det området sammanfaller väl med stormens öga på satellitbilden, figur 1. I området  $x \in (0, 600)$  och  $y \in (0, 600)$  km i figur 5 har vindfältet i stort sett oförändrad riktning, vilket istället ger en rotation nära noll. Att vindarna rör sig moturs kring ett lågtryck på norra halvklotet förklaras av Corioliskraften.

Enheterna för divergens och rotation visas i figur 4 och 5 och är 1/s. Det är naturligt då



divergens- och rotationsoperatorerna verkar som en typ av derivata på vindhastighetsfältet med avseende på längd. Divergens och rotation ger alltså en bild av hur mycket vindhastigheten förändras per längdenhet.



# Bilaga

## Källkod Python

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

# Read data
P = np.loadtxt('tryckfalt.dat')
u = np.loadtxt('vindfalt_u.dat')
v = np.loadtxt('vindfalt_v.dat')

# Meshgrid
x = np.linspace(0, 55*30, 31)
y = np.linspace(0, 55*29, 30)
X, Y = np.meshgrid(x, y)

# LaTeX font
plt.rcParams['mathtext.fontset'] = 'cm'
plt.rcParams['font.family'] = 'STIXGeneral'

# Wind speed
speed = np.sqrt(u*u + v*v)
print('Max wind speed:', speed.max())

### Air pressure isobars ###
CS = plt.contour(X, Y, P, 20, linewidths=1, colors='black',)
plt.clabel(CS, inline=True, fmt='%1.f', fontsize=10)
plt.xlabel('$x$ (km)', fontsize=13)
plt.ylabel('$y$ (km)', fontsize=13)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

### Wind velocity vector field ###
strm = plt.streamplot(X, Y, u, v, color=speed, cmap=plt.cm.Spectral, density=2,
                      linewidth=1)
cb = plt.colorbar()
cb.set_label('Fart (m/s)', fontsize=13)
cb.ax.tick_params(labelsize=13)
plt.xlabel('$x$ (km)', fontsize=13)
plt.ylabel('$y$ (km)', fontsize=13)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

# Divergence and curl
```



```

dx = 55
dy = 55
dudx = np.gradient(u, dx, axis=1, edge_order=2)
dudy = np.gradient(u, dy, axis=0, edge_order=2)
dvdx = np.gradient(v, dx, axis=1, edge_order=2)
dvdy = np.gradient(v, dy, axis=0, edge_order=2)
div = (dudx + dvdy)
curl = dvdx - dudy

### Divergence ###
CS = plt.contourf(X, Y, div, 75, cmap=plt.cm.Spectral)
cb = plt.colorbar(CS)
cb.set_label('$\\nabla \\cdot [u,v] $ (1/s)', fontsize=13)
cb.ax.tick_params(labelsize=13)
plt.xlabel('$x$ (km)', fontsize=13)
plt.ylabel('$y$ (km)', fontsize=13)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.quiver(X, Y, u, v, 15)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

### Curl ###
CS = plt.contourf(X, Y, curl, 74, cmap=plt.cm.Spectral)
cb = plt.colorbar(CS)
cb.set_label('$\\nabla \\times [u,v] $ (1/s)', fontsize=13)
cb.ax.tick_params(labelsize=13)
plt.xlabel('$x$ (km)', fontsize=13)
plt.ylabel('$y$ (km)', fontsize=13)
plt.xticks(fontsize=13)
plt.yticks(fontsize=13)
plt.quiver(X, Y, u, v, 15)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

```