Presentasjon av bacheloroppgaven

Are Frode Kvanum

April 28, 2021

Section 1

Using finite difference methods to numerically solve the Eady Model with Python

Section 2

Foreløpig utvikling

Oppsett av grid

$$\frac{\partial}{\partial t}\frac{\partial}{\partial z}\psi - \frac{\partial}{\partial x}\psi = 0 \quad z(0)$$
$$(\frac{\partial}{\partial t} + \frac{\partial}{\partial x})\frac{\partial}{\partial z}\psi - \frac{\partial}{\partial x}\psi = 0 \quad z(1)$$

- Arbeider med en spektral-modell
- Gridet er spesifisert som en kombinasjon av fase, bølgetall og en strømfunksjon
- Mye arbeid for å sette opp gridet, men med spektralmodell -> konvolusjon i frekvensdomenet. (Dette gjaldt visst ikke for flere lag, kommer tilbake det :))

```
def forward_euler(psit, alpha, m, dt):
    #                    konvolusjonsfilter
    return psit - ((1j*m*dt)/alpha)*(1.+alpha)*psit
```

Ettlagsmodell

- Modellerer topplaget av strømningen
- Løsningen utvikles i frekvensdomenet
- Enkel finite difference i en dimensjon

```
for k in range(int(nk/dt)):
    psit = solver(psit, self.alpha, self.m2, dt)
    frames.append(np.real(np.fft.ifft2(psit)))
```

Tolagsmodell

- Inkluderer koblingen mellom topplaget og bunnlaget
- Planar-waves løsning
- Manuell transformasjon ut av fourierdomenet tilbake til spektraldomenet

```
Q = np.exp(1j*m*xv)*np.exp(1j*n*yv)

psit_bot = A*np.exp(-alpha) + B
psit_top = A + B*np.exp(-alpha)

frames.append([np.real(psit_top*Q),
               np.real(psit_bot*Q)])
```

Utfordringer

- Opprinnelig skulle begge løsninger følge oppskriften til ettlagsmodellen
- Med flere lag ble det utfordrende å hente ut amplituden til begge lagene ut fra en initiell bølgeløsning
- Måtte tenke nytt, dermed manuell fouriertransformasjon som skissert i forrige slide

Veien videre

- n-lags modell
- Har jeg tid til å inkludere turbulens?
- Er egentlig resultatene riktig, oppfører modellen seg riktig?