

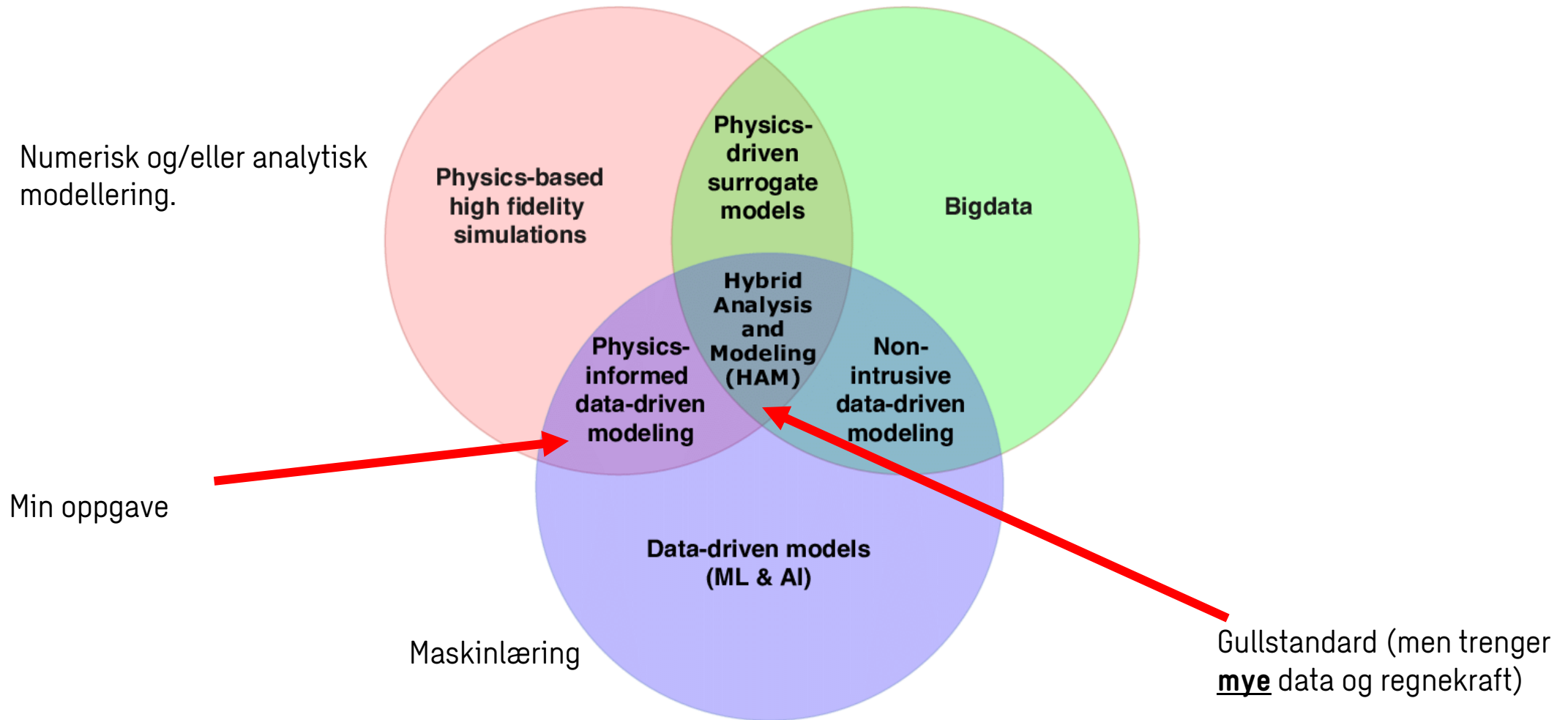
# Neurale nettverk som numerisk verktøy

Jakob Muan Haga  
14.09.2023

# Innhold

- Introduksjon
- 1D konsollidering
  - Prediksjon
  - Interpolasjon
  - Ekstrapolasjon
  - Tilbakeregning
  - Sanntid bayesian updating
  - Diskusjon resultat
- 2D koblet strømning-deformasjon
  - Prediksjon
  - Diskusjon resultat
- Sluttkommentar

# Hybrid modellering

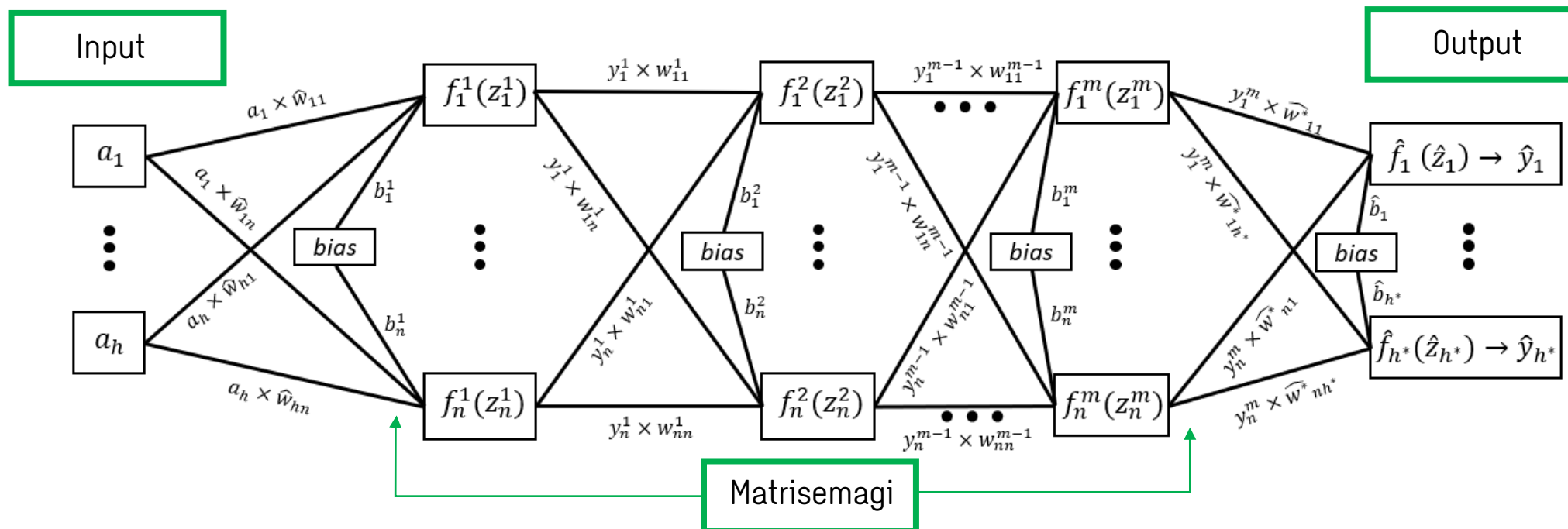


# Nevrale nettverk

Ekte vare:



Kunstig:



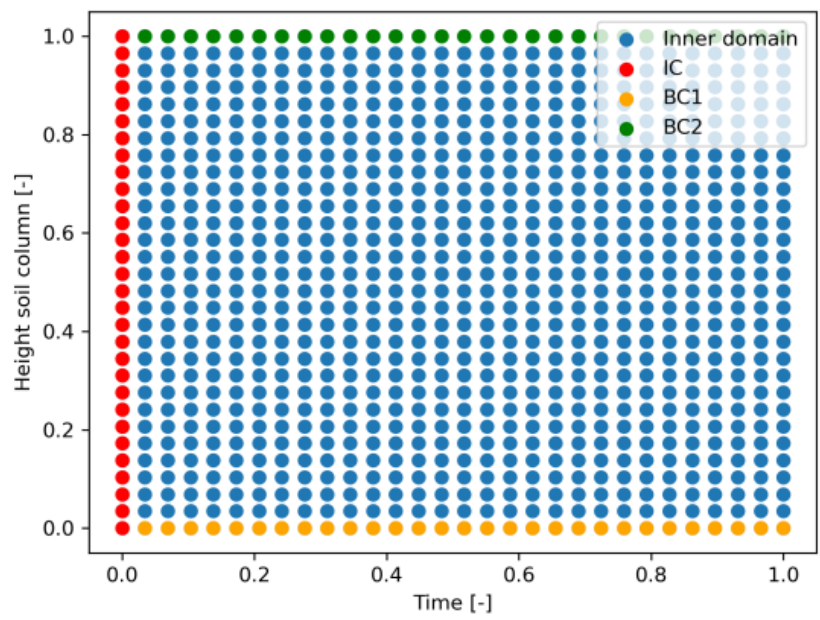
# Opplæring i 1D konsollidering

- Modell lærer ved:

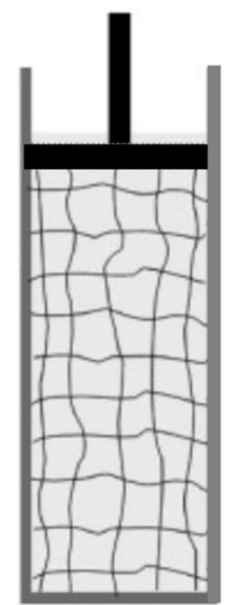
Prediksjon  $\rightarrow$  Beregne feil  $\rightarrow$  Oppdatere parametere



$$\mathcal{L}_{total} = \mathcal{L}_{pde} + \mathcal{L}_{ic} + \mathcal{L}_{bc} + \cancel{\mathcal{L}_{data}}$$



Terzhagi problem



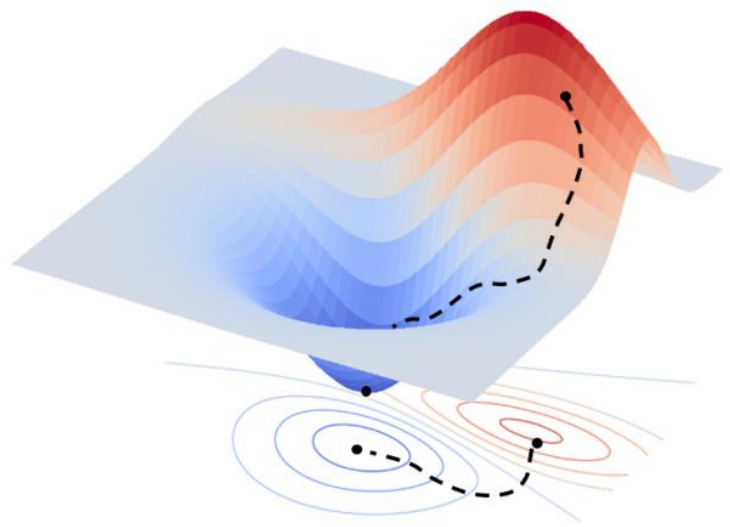
$$c_v \frac{\partial^2 u}{\partial z^2} = \frac{\partial u}{\partial t}$$

$$\Delta w_{ij}^r = -\eta \frac{\partial E}{\partial w_{ij}^r} \quad (53)$$

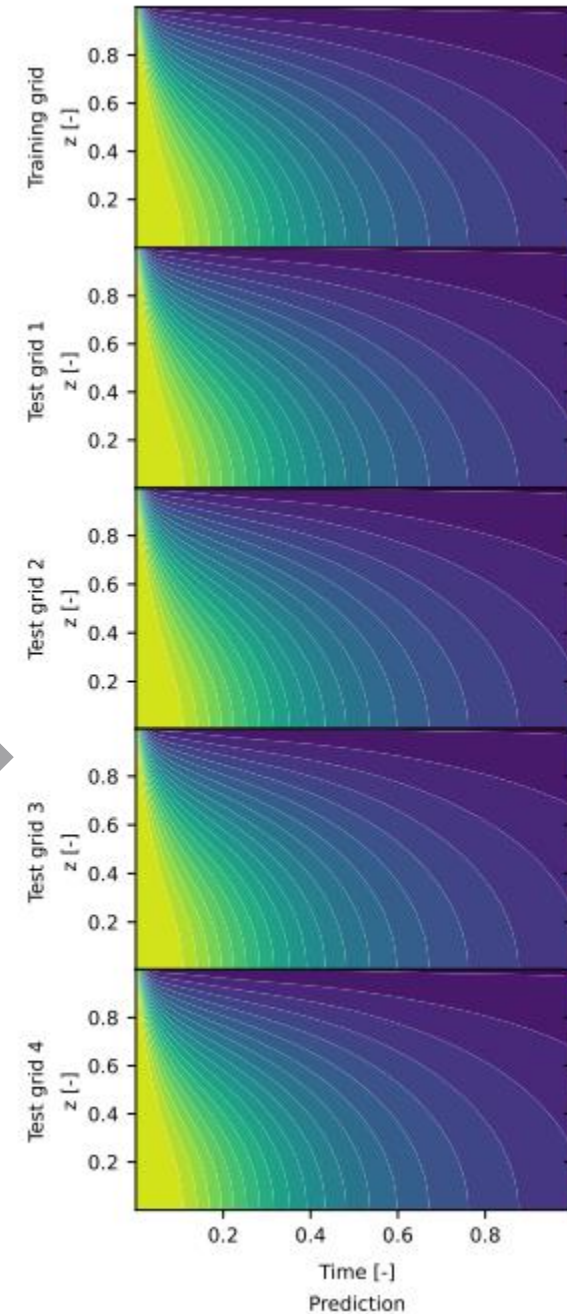
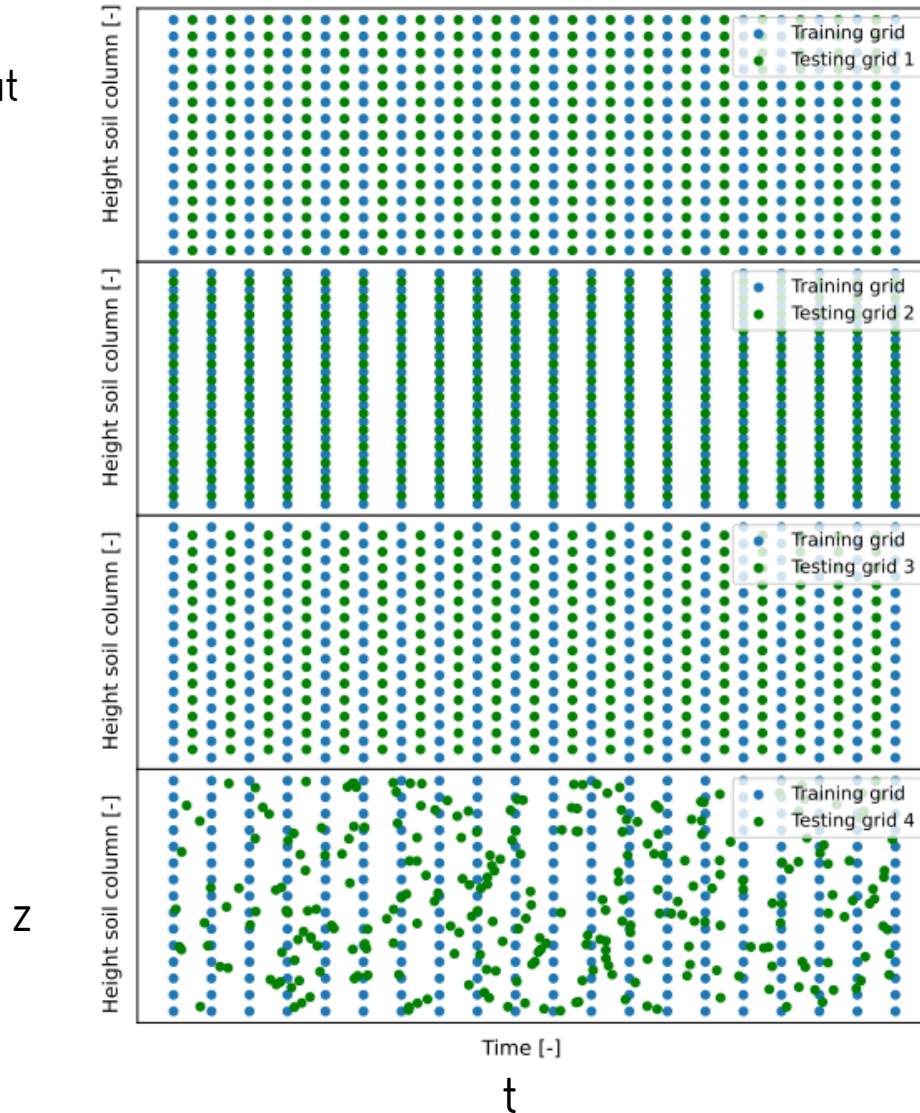
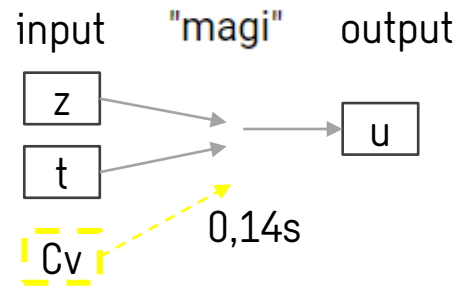
$$w_{ij,(Updated)}^r = w_{ij,(Old)}^r + \Delta w_{ij}^r \quad (54)$$

$$\Delta b_j^r = -\eta \frac{\partial E}{\partial b_j^r} \quad (55)$$

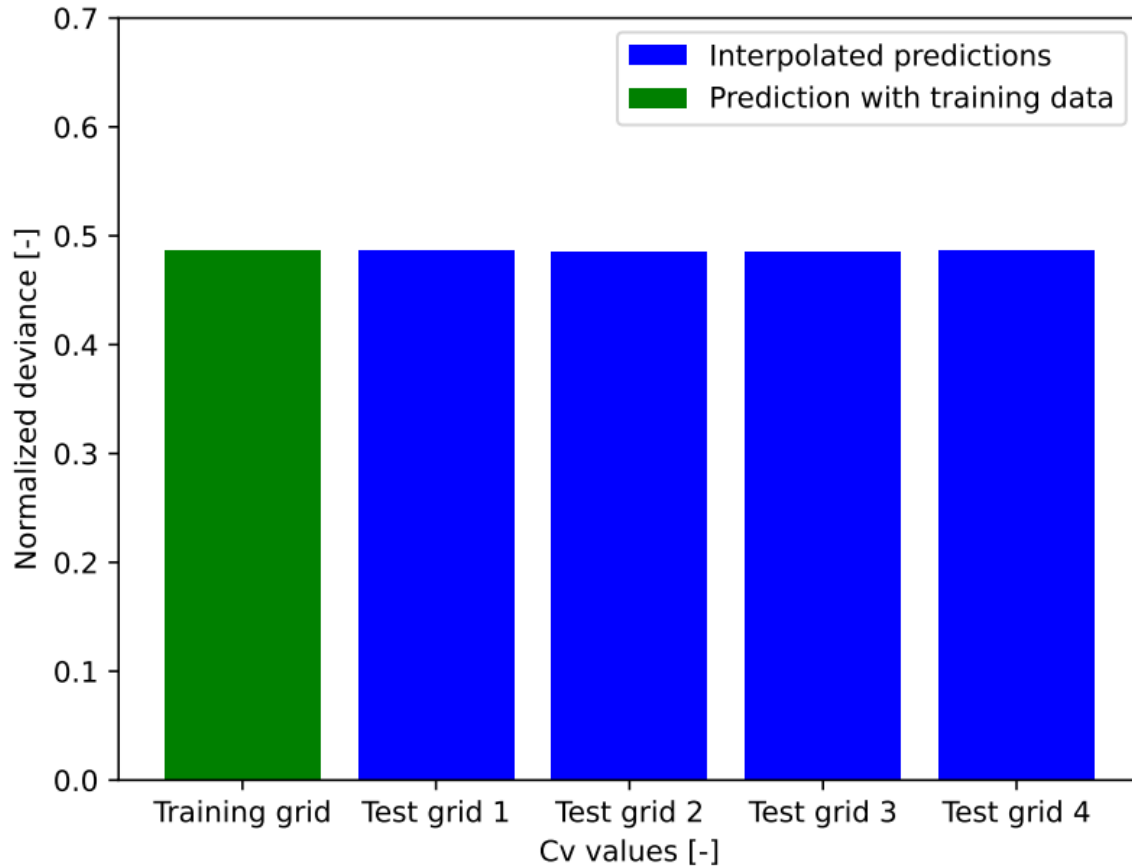
$$b_{j,(Updated)}^r = b_{j,(Old)}^r + \Delta b_j^r \quad (56)$$



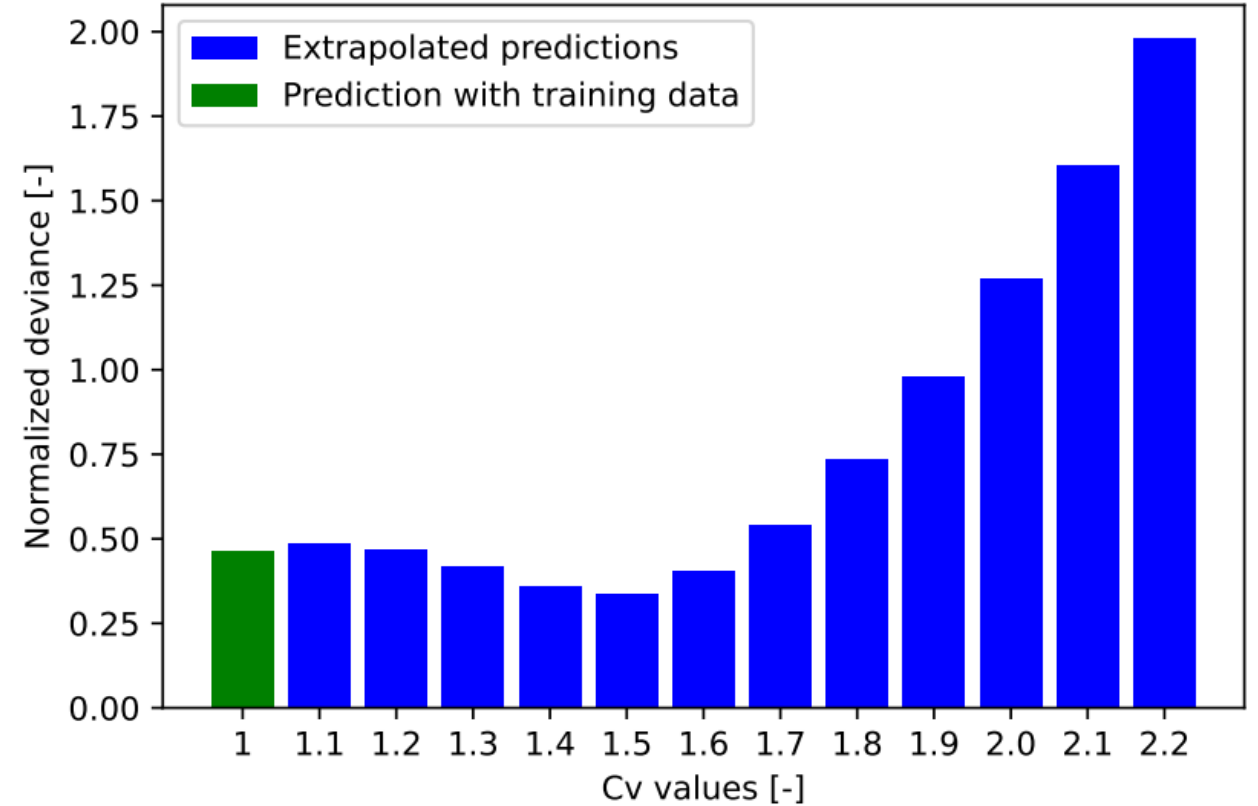
# Modell i bruk



# Kommentar til resultat

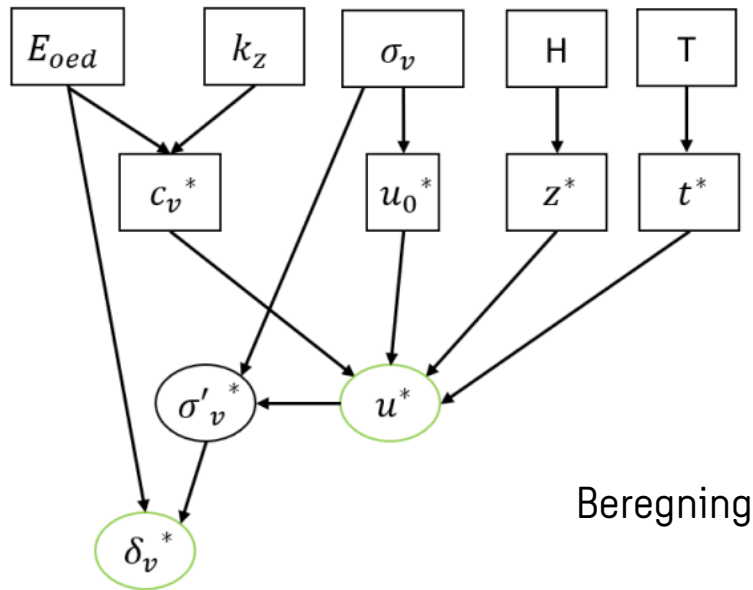


Resultat fra prediksjon av ulike testmesh (interpolering).

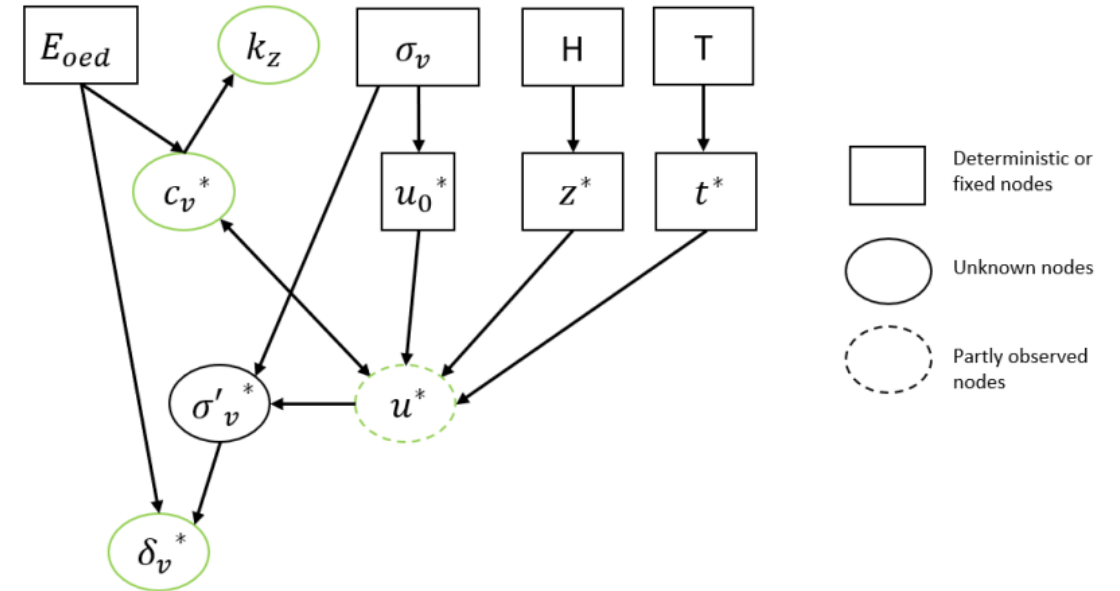
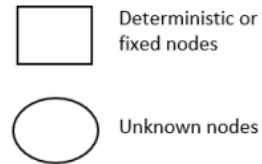


Resultat fra prediksjon av ulike test Cv-verdier (ekstrapolering).

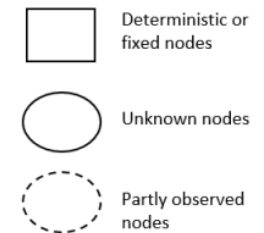
# Flytskjema 1D konsollidering



Beregning

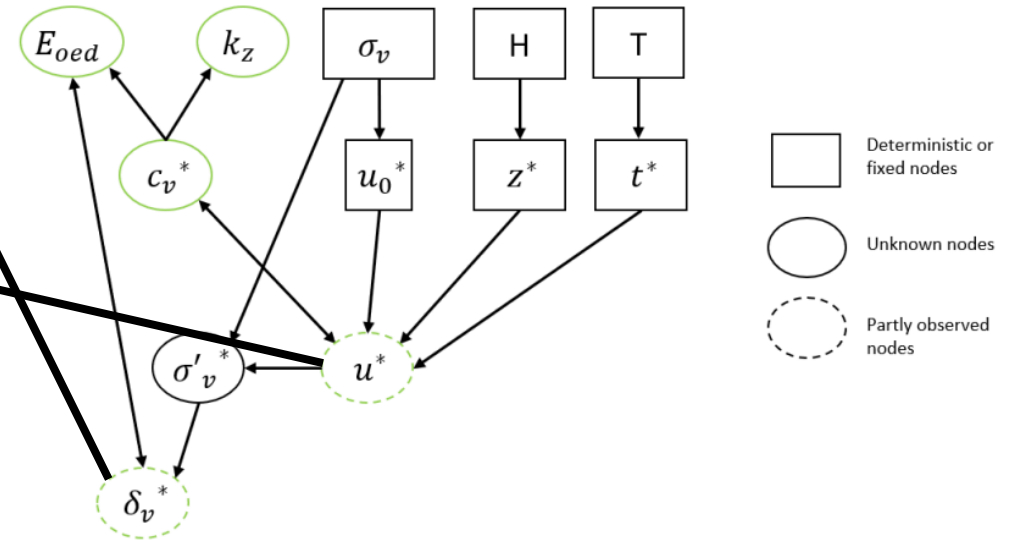
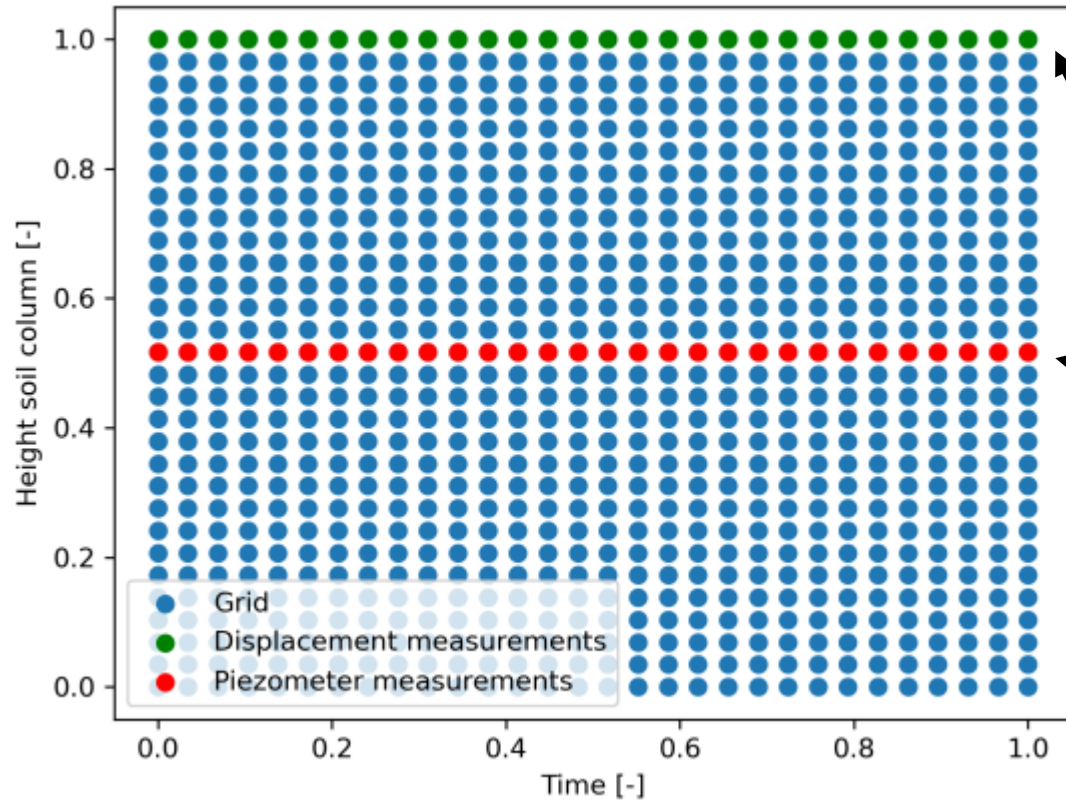


Tilbakeregning





# Data for tilbakeregning



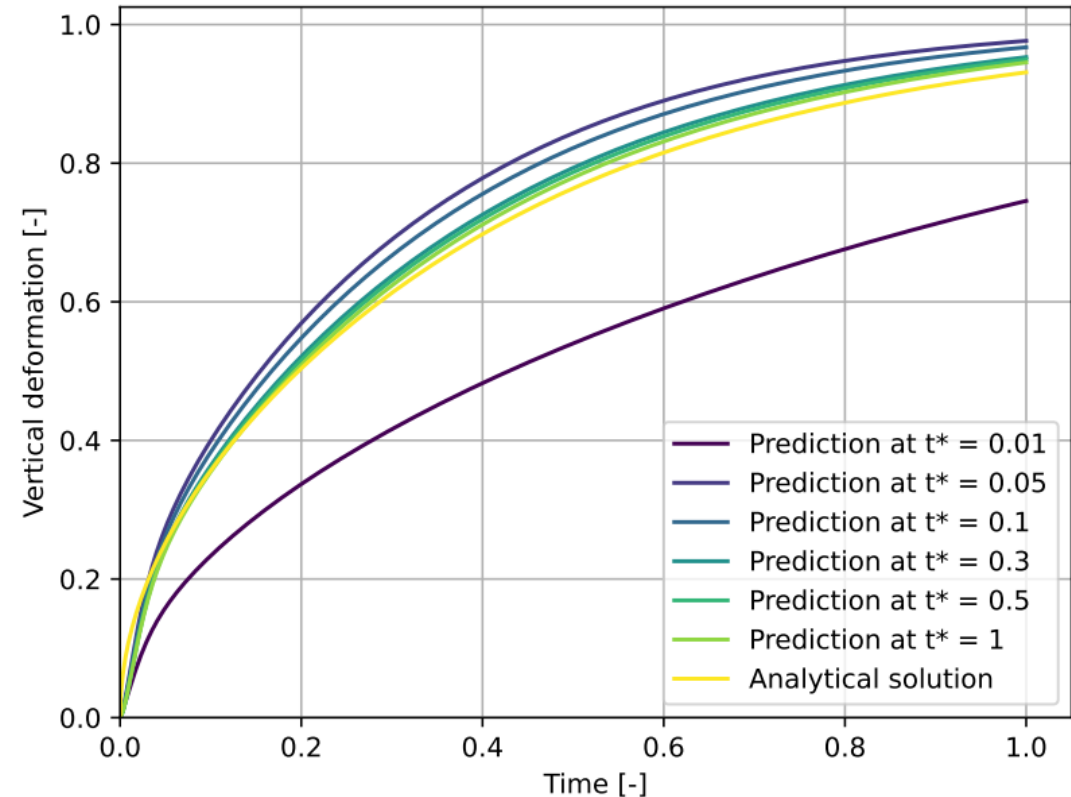
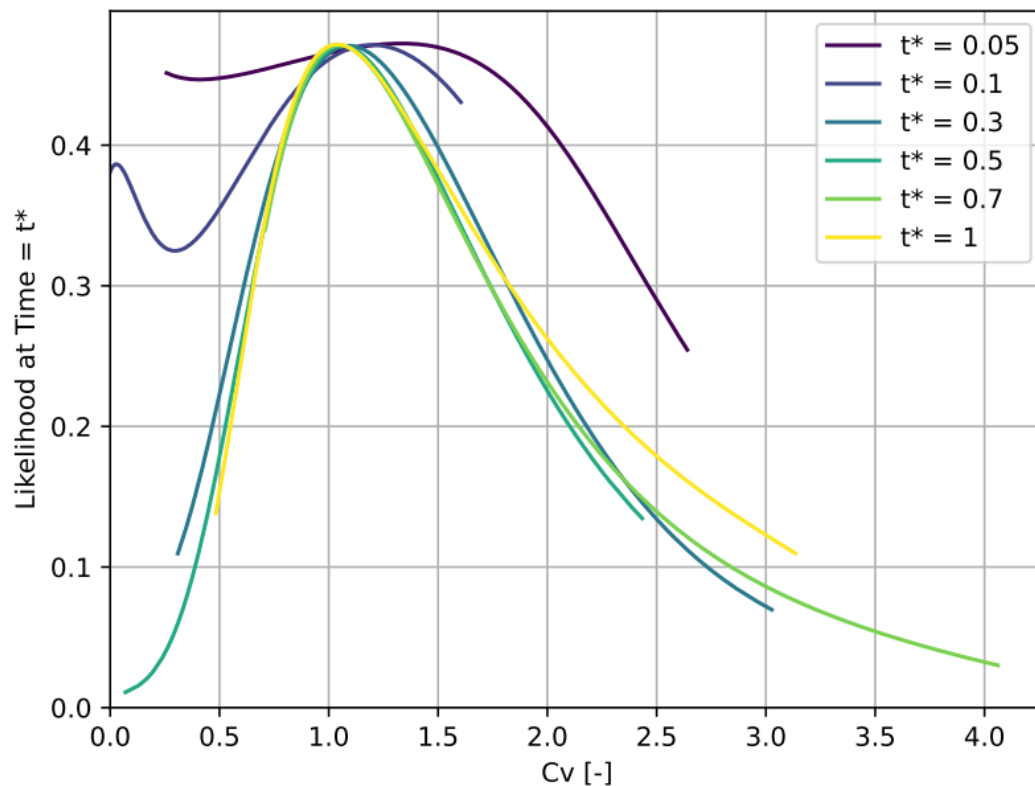
$$\mathcal{L}_{total} = \mathcal{L}_{pde} + \mathcal{L}_{ic} + \mathcal{L}_{bc} + \mathcal{L}_{data}$$

# Oppdatere prediksjon i sanntid

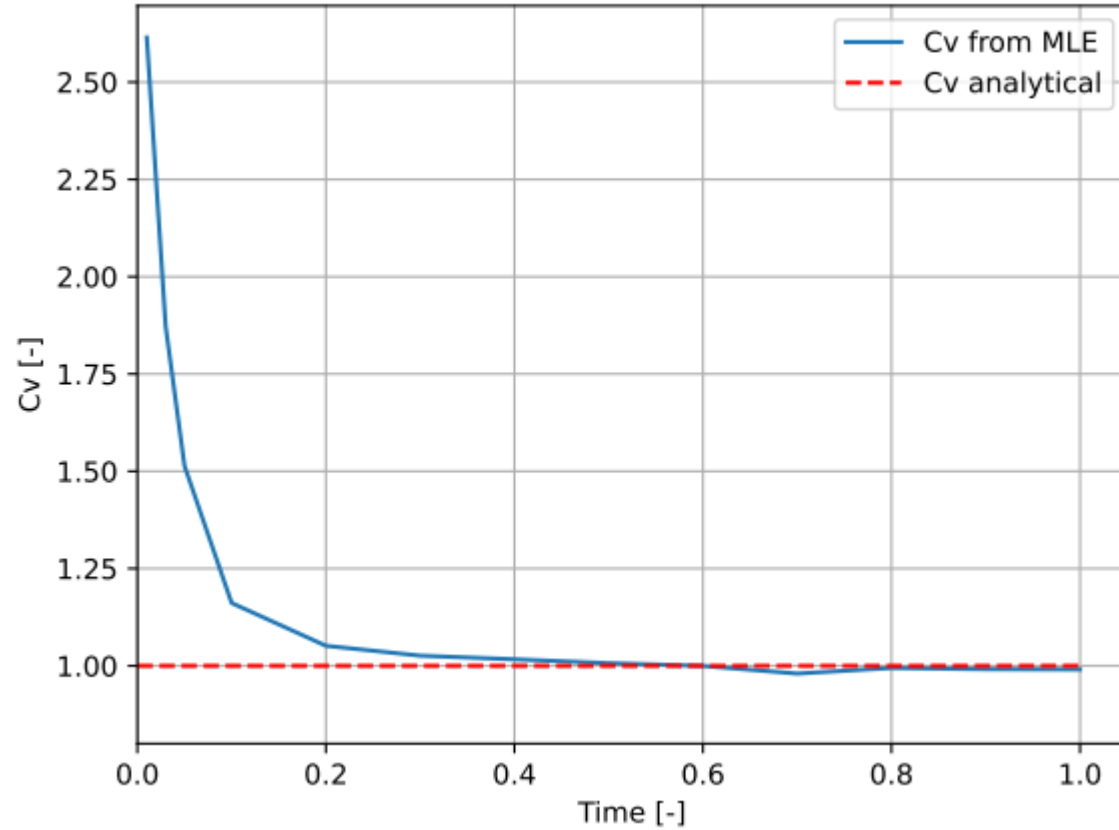
$$P(\mathbf{D}|C_v^*) = \prod_{i=1}^n \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2} \left(\frac{\text{pred}(t_i, C_v^*) - D_i}{\sigma}\right)^2\right\}$$

Sampling:

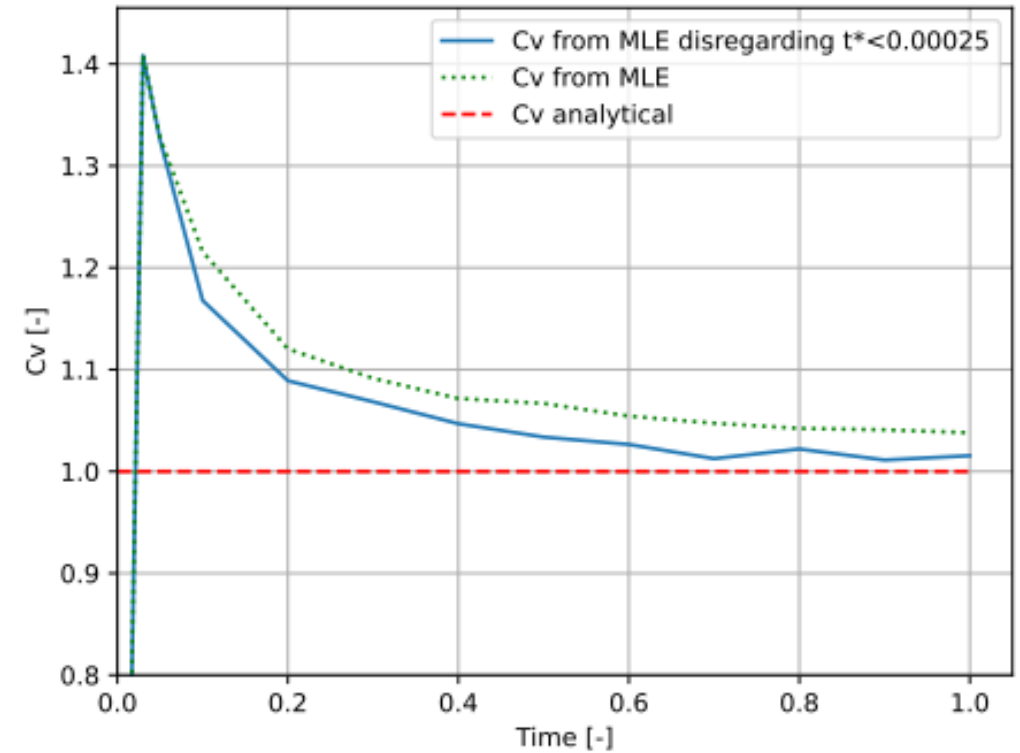
- Markov Chain Monte Carlo
- Metropolis Hastings algoritme



# Kommentar til resultat



MLE med deformasjonsdata



MLE med piezometerdata

# Opplæring i 2D koblet strømning-deformasjon

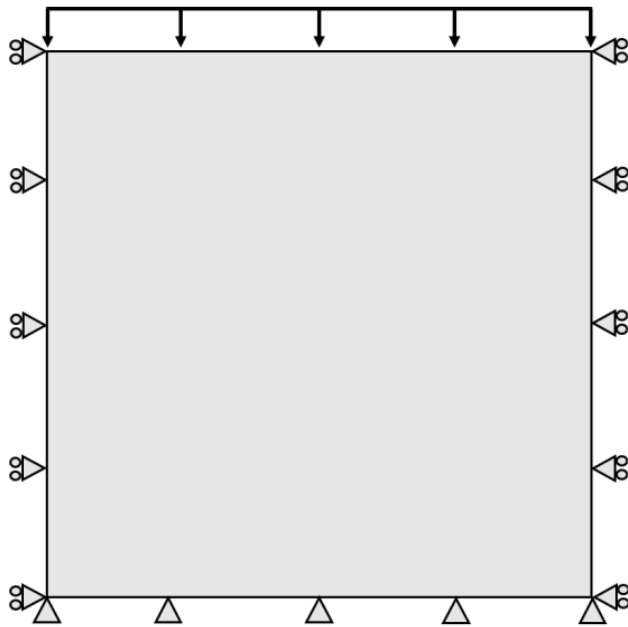
Strømning i porøse media  
(Darcy+massebevarelse)

$$\alpha \frac{\partial \varepsilon_v}{\partial t} + S \frac{\partial p_f}{\partial t} = \nabla \cdot \left( \frac{k_{ij}}{\gamma_f} \nabla p_f \right)$$

Naviers likevektsligninger  
for isotropisk plan tøyning

$$\left( K + \frac{G}{3} \right) \frac{\partial \varepsilon_{vol}}{\partial x_1} + \alpha \frac{\partial p_w}{\partial x_1} + G \nabla^2 u_1 + \rho b_1 = 0$$

$$\left( K + \frac{G}{3} \right) \frac{\partial \varepsilon_{vol}}{\partial x_2} + \alpha \frac{\partial p_w}{\partial x_2} + G \nabla^2 u_2 + \rho b_2 = 0.$$




---

## Algorithm 1 Sequential solver

---

```

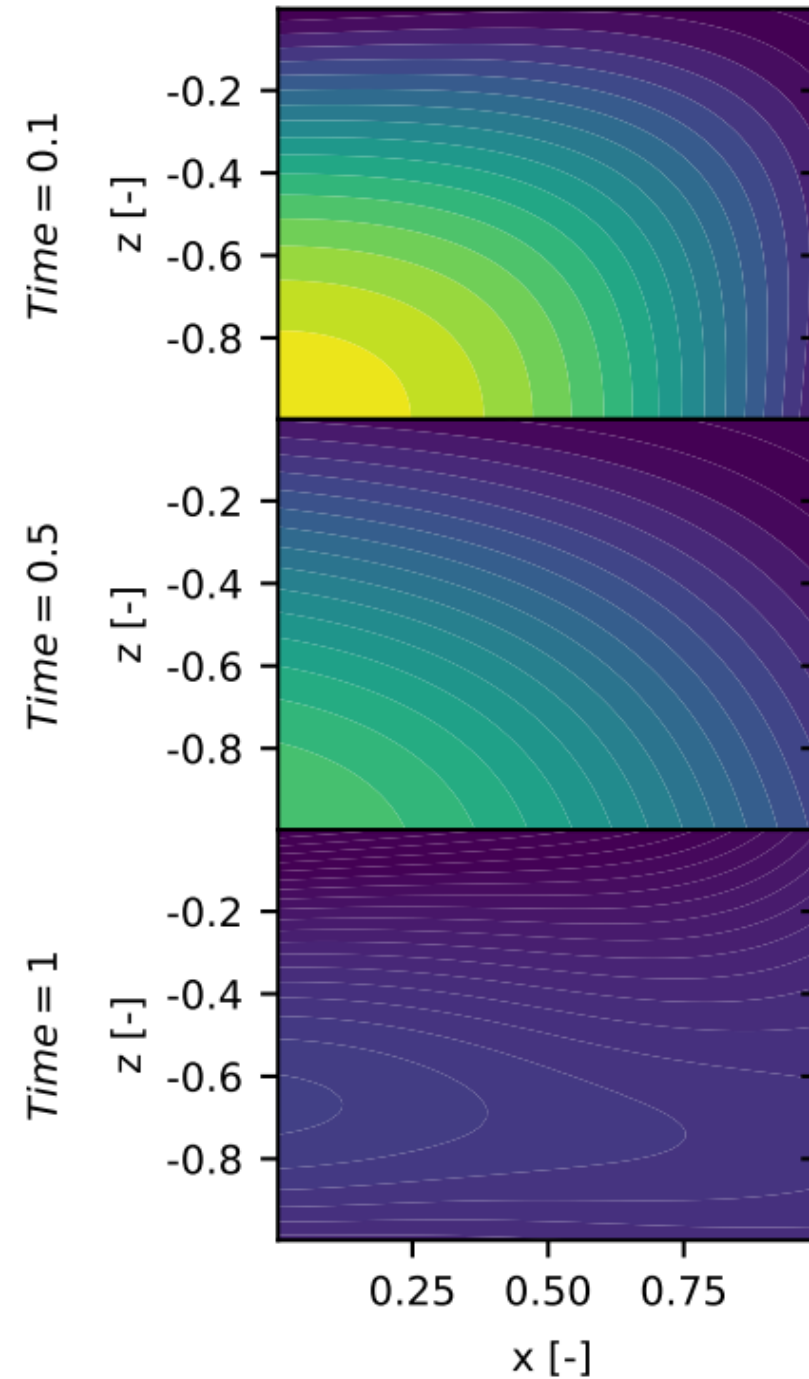
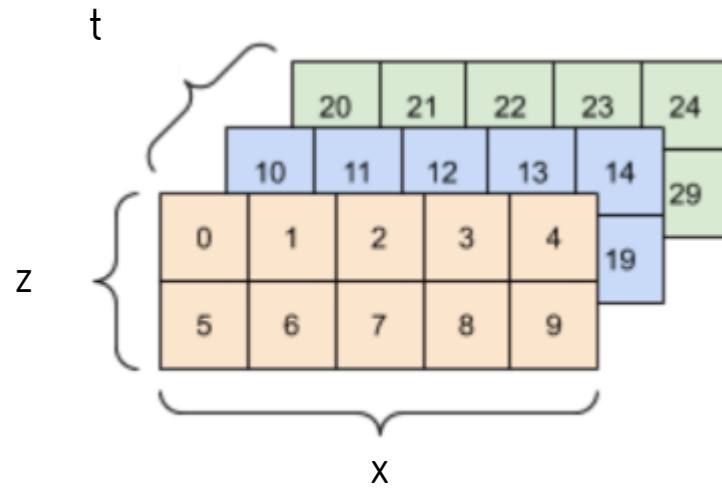
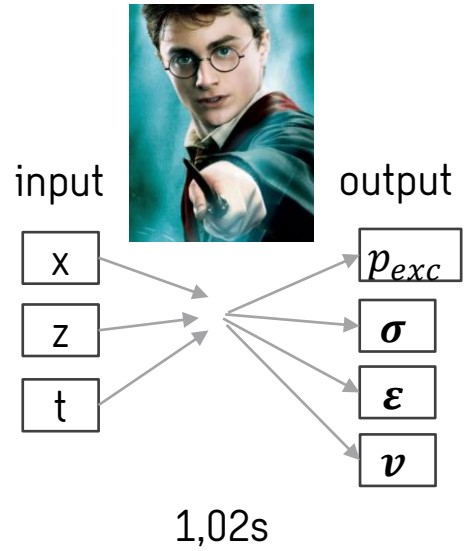
1: procedure FIXED-STRAIN SPLIT( $u_{init}, \varepsilon_{v,init}$ )
2:    $u = u_{init}$ 
3:    $\varepsilon_v = \varepsilon_{v,init}$ 
4:   for n number of iterations do
5:     Solve storage equation for  $u$  ( $\delta \varepsilon_v = 0$ )
6:      $u \leftarrow$  Storage equation
7:     Solve equilibrium equations for  $\varepsilon_v(u)$ 
8:      $\varepsilon_v \leftarrow$  Equilibrium equations

9: procedure FIXED-STRESS SPLIT( $u_{init}, p_{init}$ )
10:   $u = u_{init}$ 
11:   $p = p_{init}$ 
12:  for n number of iterations do
13:    Solve storage equation for  $u$  ( $\delta p = 0$ )
14:     $u \leftarrow$  Storage equation
15:    Solve equilibrium equations for  $p(u)$ 
16:     $p \leftarrow$  Equilibrium equations
    
```

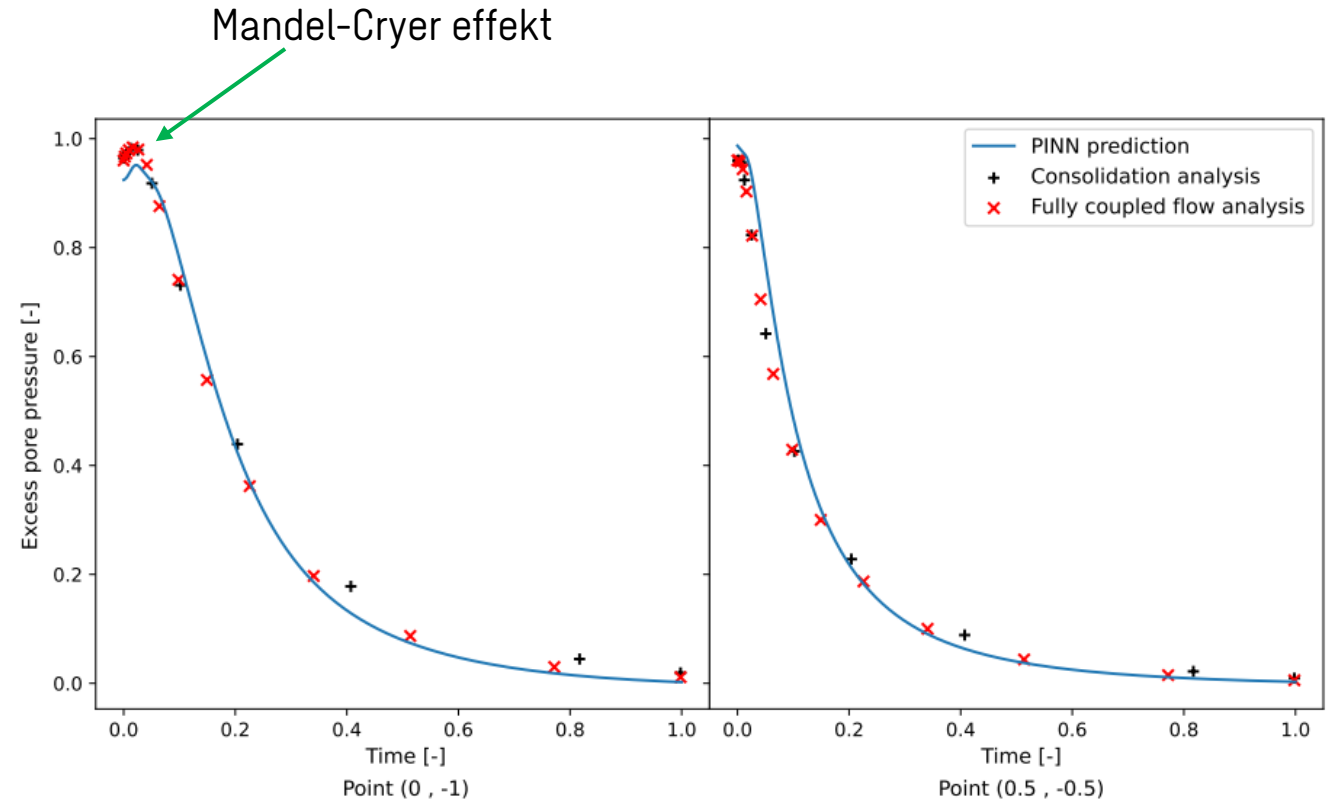
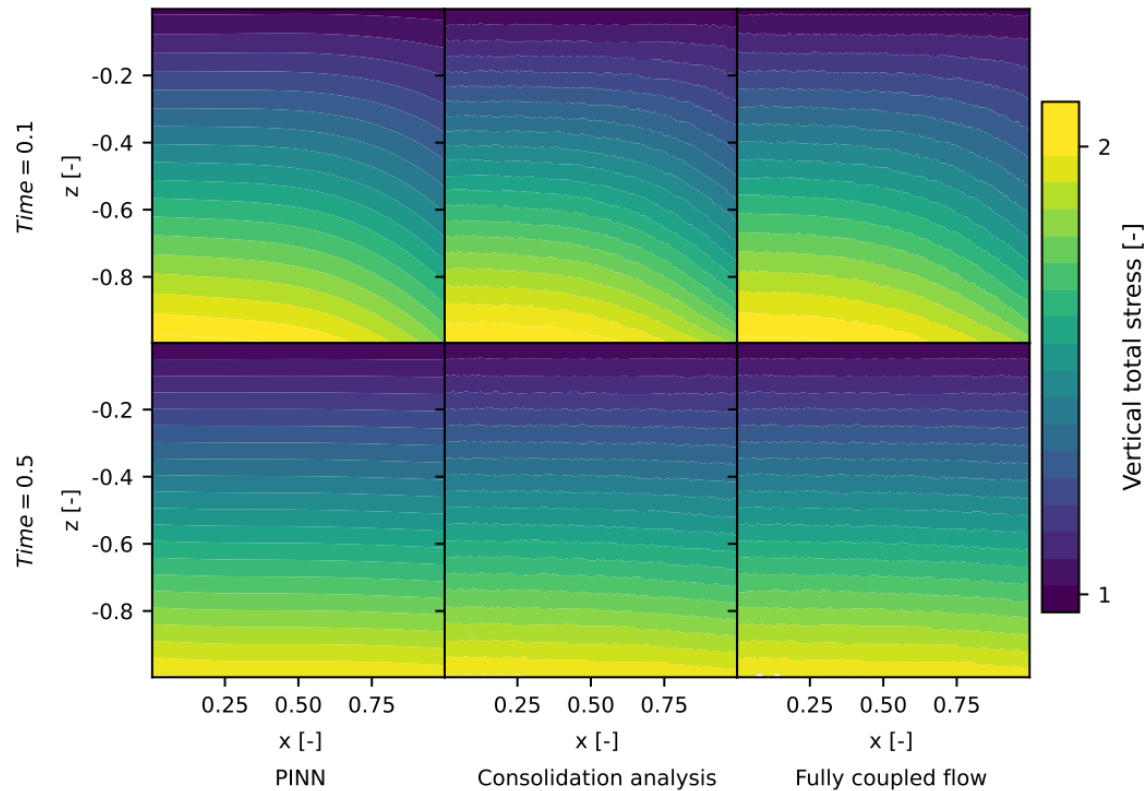
---

Inspirert av Kim et al.  
(2011)

# Modell i bruk

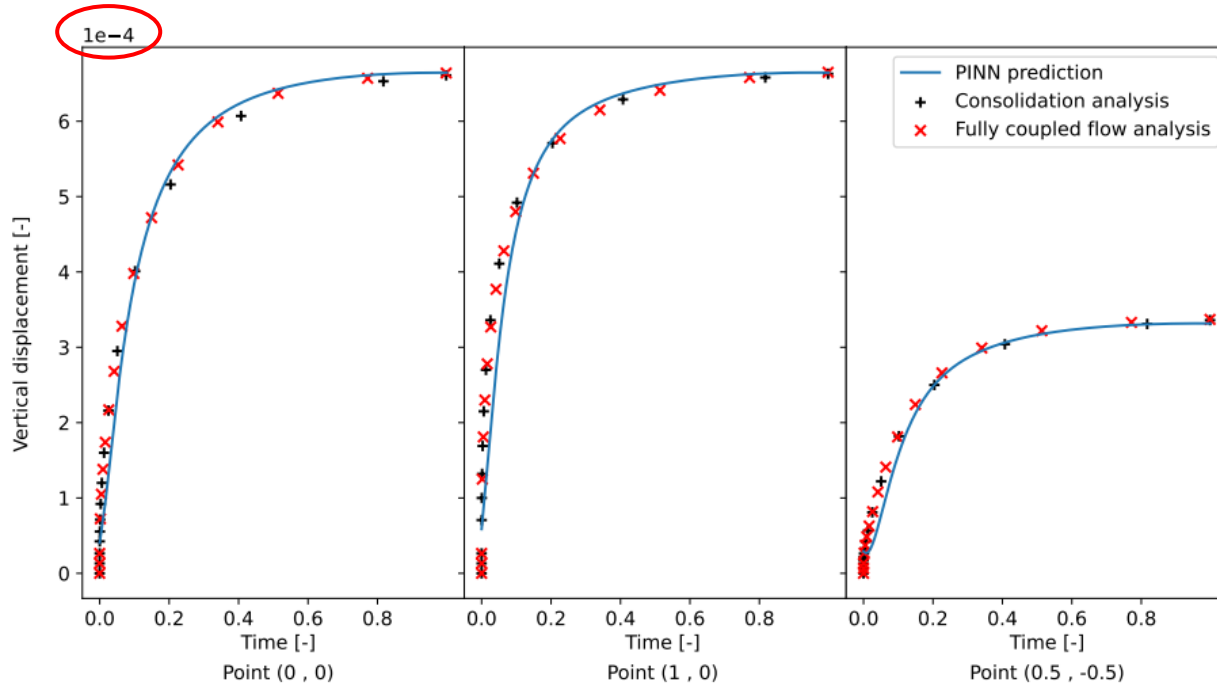


# Kommentar til resultat

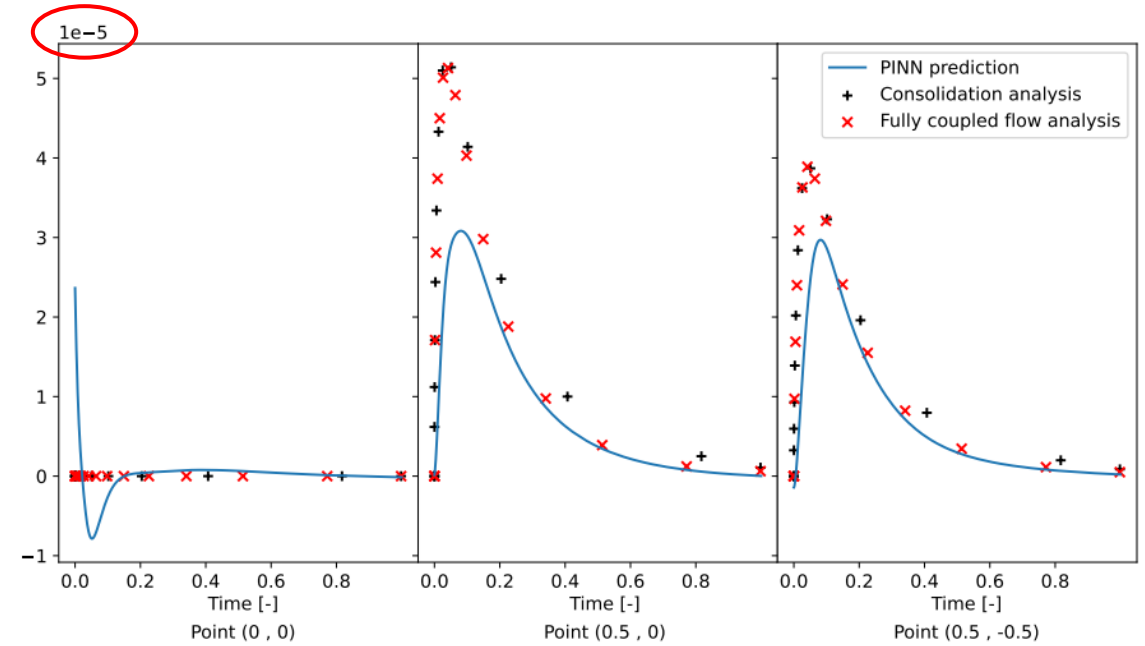


Dissipasjon av lastindusert poretrykk over tid i to forskjellige punkter.

# Kommentar til resultat



Vertikal deformasjon



Horizontal deformasjon

**Bias mot store verdier!**

# Slutt kommentar

Mitt nettverk: 23400 parametere



How large is the neural network you are made up of?



I am based on the GPT-3.5 architecture, which has 175 billion parameters. These parameters are the values that the neural network uses to understand and generate text based on the



Ca. 7,5 millioner ganger større

GPT-4 ryktes å ha over 1 trillion.

## Beregningstid

	1D	2D
FEM/Plaxis	1,58 s	4 s*
PINN	0,14 s	1,02 s

### Pros:

Veldig raske prediksjoner -> bra for parametrisk modellering og simuleringer.

Kan ha «evig» mange input dimensjoner

Optimaliseringsalgoritmen kan tilbakeregne flere parametere samtidig.

### Cons:

Litt avvikende prediksjon sammenlignet med Plaxis beregninger

Beregningstung treningsprosess