Standardprosjekt matteoblig

Maya Standahl Teien

April 2025

1 Oppgave 1

```python
import numpy as np

def derivert(f, x, h):
    return (f(x + h) - f(x)) / (h)


def f(x):
    return np.exp(x)

x0 = 1.5
print(f"Den deriverte av f i h = 0.1 er {derivert(f, x0,0.1)}. Differanse: {derivert(f,x0,0.1)-f(x0)}")
print(f"Den deriverte av f i h = 0.01 er {derivert(f, x0,0.01)}. Differanse: {derivert(f,x0,0.01)-f(x0)}")
print(f"Den deriverte av f i h = 0.001 er {derivert(f, x0,0.001)}. Differanse: {derivert(f,x0,0.001)-f(x0)}")
print(f"Den deriverte av f i h = 0.0001 er {derivert(f, x0,0.0001)}. Differanse: {derivert(f,x0,0.0001)-f(x0)}")
print(f"Den deriverte av f i h = 0.00001 er {derivert(f, x0,0.00001)}. Differanse: {derivert(f,x0,0.00001)-f(x0)}")
```

```
Den deriverte av f i h = 0.1 er 4.713433540570504. Differanse: 0.2317444702324396
Den deriverte av f i h = 0.01 er 4.5041723976187775. Differanse: 0.022483327280713006
Den deriverte av f i h = 0.001 er 4.483930662008362. Differanse: 0.0022415916702973604
Den deriverte av f i h = 0.0001 er 4.481913162264206. Differanse: 0.00022409192614158968
Den deriverte av f i h = 0.00001 er 4.4817114789097445. Differanse: 2.2408571680010425e-05
```

Du kan gjøre h så liten du vil, men h vil kun konvergere mot 4.4817

2 Oppgave 2

```python
import numpy as np

def derivert(f, x, h):
    return (f(x + h) - f(x-h)) / (2*h)


def f(x):
    return np.exp(x)

x0 = 1.5
print(f"Den deriverte av f i h = 0.1 er {derivert(f, x0,0.1)}. Differanse: {derivert(f,x0,0.1)-f(x0)}")
print(f"Den deriverte av f i h = 0.01 er {derivert(f, x0,0.01)}. Differanse: {derivert(f,x0,0.01)-f(x0)}")
print(f"Den deriverte av f i h = 0.001 er {derivert(f, x0,0.001)}. Differanse: {derivert(f,x0,0.001)-f(x0)}")
print(f"Den deriverte av f i h = 0.0001 er {derivert(f, x0,0.0001)}. Differanse: {derivert(f,x0,0.0001)-f(x0)}")
print(f"Den deriverte av f i h = 0.00001 er {derivert(f, x0,0.00001)}. Differanse: {derivert(f,x0,0.00001)-f(x0)}")
```

```
Den deriverte av f i h = 0.1 er 4.489162287752202. Differanse: 0.007473217414137423
Den deriverte av f i h = 0.01 er 4.481763765529401. Differanse: 7.469519133618263e-05
Den deriverte av f i h = 0.001 er 4.481689817286139. Differanse: 7.469480740596168e-07
Den deriverte av f i h = 0.0001 er 4.48168907780655. Differanse: 7.468485385686563e-09
Den deriverte av f i h = 0.00001 er 4.481689070434669. Differanse: 9.660450217552352e-11
```

h kan bli så lav som

$$10^{-16}$$

før pcen ikke klarer mer

$$\lim_{x \to \infty} f(x)$$

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + ...$$

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - ...$$

$$\frac{f(x+h) - f(x-h)}{2h}$$

$$\frac{(f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + ...) - (f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - ...)}{2h}$$

$$\frac{2f'(x)h}{2h}$$

$$f'(x)$$

3 Oppgave 3

```python
import numpy as np

def derivert(f, x, h):
    return (f(x - 2*h) - 8*f(x-h) + 8*f(x+h) -f(x+2*h)) / (12*h)


def f(x):
    return np.exp(x)

x0 = 1.5
print(f"Den deriverte av f i h = 0.1 er {derivert(f, x0,0.1)}. Differanse: {derivert(f,x0,0.1)-f(x0)}")
print(f"Den deriverte av f i h = 0.01 er {derivert(f, x0,0.01)}. Differanse: {derivert(f,x0,0.01)-f(x0)}")
print(f"Den deriverte av f i h = 0.001 er {derivert(f, x0,0.001)}. Differanse: {derivert(f,x0,0.001)-f(x0)}")
print(f"Den deriverte av f i h = 0.0001 er {derivert(f, x0,0.0001)}. Differanse: {derivert(f,x0,0.0001)-f(x0)}")
print(f"Den deriverte av f i h = 0.00001 er {derivert(f, x0,0.00001)}. Differanse: {derivert(f,x0,0.00001)-f(x0)}")
```

```
Den deriverte av f i h = 0.1 er 4.481674113579637. Differanse: -1.4956758427331351e-05
Den deriverte av f i h = 0.01 er 4.481689068844186. Differanse: -1.4938787984419832e-09
Den deriverte av f i h = 0.001 er 4.481689070337709. Differanse: -3.552713678800501e-13
Den deriverte av f i h = 0.0001 er 4.481689070338449. Differanse: 3.8458125573015423e-13
Den deriverte av f i h = 0.00001 er 4.481689070390259. Differanse: 5.219469301209756e-11
```

Blir mer presist for hver formel/ekspriment. Og differansen blir mindre fra den sanne verdien.

4 Oppgave 4

Varmelikningen:

$$\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2}$$

initialbetingelse:

$$u(x, 0) = sin(x)$$

randkrav:

$$u(0, t) = u(1, t) = 0$$

eksplisitt:

$$\frac{u_{i,j+1} - u_{ij}}{k} = \frac{u_{i+1,j} - 2 * u_{i,j} + u_{i-1,j}}{h^2}$$

$$u_{i,j+1} - u_{ij} = \frac{k}{h^2} * (u_{i+1,j} - 2 * u_{i,j} + u_{i-1,j})$$

$$u_{i,j+1} = u_{ij} + \frac{k}{h^2} * (u_{i+1,j} - 2 * u_{i,j} + u_{i-1,j})$$

Stabilitetsbetingelse: Stabil hvis:

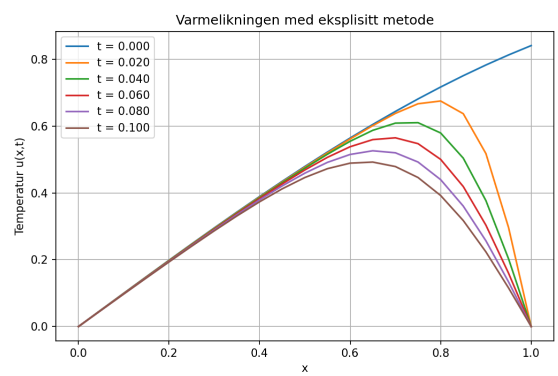
$$\frac{k}{h^2} < \frac{1}{2}$$

Ustabil hvis:

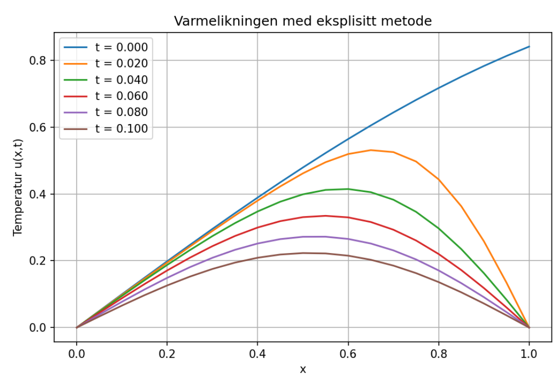
$$\frac{k}{h^2} > \frac{1}{2}$$

4.1 Grafer

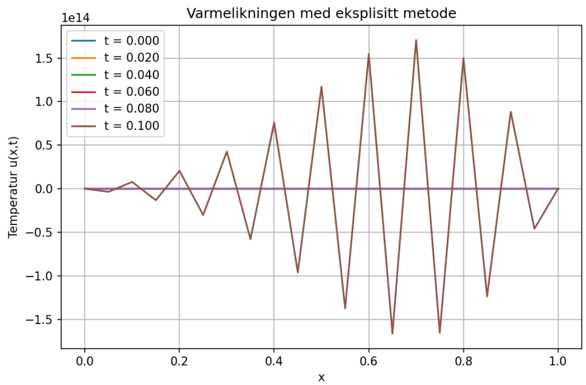
h= 0.1, k = 0.001



h= 0.05, k = 0.001



h= 0.05, k = 0.002



h = k Får feilmelding av python.

4.2 Analyse

For graf en ser vi at den er veldig spiss og forsjøvet, noe som ikke passer bra med en harmonisk funskjon.

$$h = 0.1, k = 0.001$$

$$\frac{0.001}{0.1^2} \Rightarrow 0.1$$

$$0.1 < 0.5$$

Dette er stabilt, og metoden fungerer fint med disse verdiene.

Den andre grafen er flatere og mer avrundet, som passer bedre med en harmonisk funskjon

$$h = 0.05, k = 0.001$$

$$\frac{0.001}{0.5^2} \Rightarrow 0.4$$

$$0.4 < 0.5$$

Dette er stabilt, og metoden fungerer fint med disse verdiene.

Den siste har virkelig ”gått av skogen”, og er derfor veldig ustabil.

$$h = 0.05, k = 0.002$$

$$\frac{0.002}{0.5^2} \Rightarrow 0.8$$

$$0.8 > 0.5$$

Denne er ustabil siden vi fikk en stabilitetsverdi på 0.8, noe som er større enn 0.5. Dette fører til at feilen vokser og løsningen går av skaftet.

Hvis h = k = 0.5:

$$\frac{0.5}{0.5^2} \Rightarrow 2$$

$$2 > 0.5$$

Da får vi et tall som er større enn 0.5. Som gjør løsningen ustabil, som fører til at den feiler. Tidssteget k må være mye mindre enn kvadratet av h for at løsningen skal være stabil.

5 Oppgave 5

Varmelikningen:

$$\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2}$$

initialbetingelse:

$$u(x,0) = sin(x)$$

randkrav:

$$u(0,t) = u(1,t) = 0$$

Implisitt:

$$\frac{u_{i,j+1} - u_{ij}}{k} = \frac{u_{i+1,j+1} - 2 * u_{i,j+1} + u_{i-1,j+1}}{h^2}$$

$$u_{i,j+1} - u_{ij} = \frac{k}{h^2} * (u_{i+1,j+1} - 2 * u_{i,j+1} + u_{i-1,j+1})$$

$$u_{i,j+1} - u_{ij} = \frac{k}{h^2} * (u_{i+1,j+1} - 2 * u_{i,j+1} + u_{i-1,j+1})$$

$$\alpha = \frac{k}{h^2}$$

$$u_{i,j+1} - u_{ij} = \alpha * (u_{i+1,j+1} - 2 * u_{i,j+1} + u_{i-1,j+1})$$

$$u_{i,j+1} - u_{ij} = \alpha * u_{i+1,j+1} - \alpha * 2 * u_{i,j+1} + \alpha * u_{i-1,j+1}$$

$$-u_{ij} = -u_{i,j+1} + \alpha * u_{i+1,j+1} - \alpha * 2 * u_{i,j+1} + \alpha * u_{i-1,j+1}$$

$$u_{ij} = u_{i,j+1} - \alpha * u_{i+1,j+1} + \alpha * 2 * u_{i,j+1} - \alpha * u_{i-1,j+1}$$

$$u_{ij} = u_{i,j+1}(1 + \alpha * 2) - \alpha * u_{i+1,j+1} - \alpha * u_{i-1,j+1}$$

Må løse et lineært likningsystem på formen:

$$A * u_{j+1} = u_j$$

der

$$A = \begin{bmatrix} 1+2\lambda & -\lambda & & & \\ -\lambda & 1+2\lambda & -\lambda & & \\ & \ddots & \ddots & \ddots & \\ & & -\lambda & 1+2\lambda & -\lambda \\ & & & -\lambda & 1+2\lambda \end{bmatrix}$$

Når koden kjører, vil man se hvordan temperaturen u(x,t) utvikler seg over tid. For implisitte metoder, er det viktig at k og h er valgt slik at $\alpha = \frac{k}{h^2}$ er tilstrekkelig liten for å opprettholde stabilitet.

6 Oppgave 6

$$\frac{u_{i,j+1} - uij}{k} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{2h^2} + \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{2h^2}$$

Som vi kan skrive om til:

$$A * u_{j+1} = B * u_j$$

$$A = \begin{bmatrix} 1+\lambda & -\lambda/2 & 0 & \cdots & 0 \\ -\lambda/2 & 1+\lambda & -\lambda/2 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & -\lambda/2 & 1+\lambda & -\lambda/2 \\ 0 & \cdots & 0 & -\lambda/2 & 1+\lambda \end{bmatrix} \quad B = \begin{bmatrix} 1-\lambda & \lambda/2 & 0 & \cdots & 0 \\ \lambda/2 & 1-\lambda & \lambda/2 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \lambda/2 & 1-\lambda & \lambda/2 \\ 0 & \cdots & 0 & \lambda/2 & 1-\lambda \end{bmatrix}$$

Ut fra koden kan vi se at Crank-Nicolson er stabil for alle K, men man kan ikke se forskjell på hvilken av metodene som er best.