Standardprosjektet 4106

Av Einar Augestad

$\boxed{1}$ $f(x) = e^x$, Finner $f'(1,5)$ for ulike verdier av h:

Finner $f'(x)$ ved $f'(x) = \frac{f(x+h) - f(x)}{h}$

$h = 0{,}01$: $f'(1{,}5) = \frac{e^{1{,}51} - e^{1{,}5}}{0{,}01} = 4{,}5042$

$h = 0{,}001$: $f'(1{,}5) = \frac{e^{1{,}501} - e^{1{,}5}}{0{,}001} = 4{,}4839$

$h = 0{,}0001$: $f'(1{,}5) = \frac{e^{1{,}5001} - e^{1{,}5}}{0{,}0001} = 4{,}4819$

$h = 0{,}00001$: $f'(1{,}5) = \frac{e^{1{,}50001} - e^{1{,}5}}{0{,}00001} = 4{,}4817$

$f'(1{,}5) = e^{1{,}5} = 4{,}4817$, Ser at for $h = 0{,}00001$ treffer man ganske godt

$\boxed{2}$ Bruker $f'(x) = \frac{f(x+h) - f(x-h)}{2h}$

$h = 0{,}1$: $f'(1{,}5) = 4{,}4892$
$h = 0{,}01$: $f'(1{,}5) = 4{,}4817$
$h = 0{,}001$: $f'(1{,}5) = 4{,}4817$

Ser her at feilen deles på 100 når h deles på 10

For $f'(x) = \frac{f(x+h) - f(x)}{h}$ betegnes stigningen som $f'(x)$ pluss resten av Taylorrekken til til f $\frac{f(x+h) - f(x)}{h} = f'(x) + \frac{f''(x)}{2}h + \frac{f'''(x)}{6}h^2$ der det som kommer etter $f'(x)$ beskriver feilen

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

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(x)}{6}h^3 + \dots$$

$$f(x-h) = f(x) + f'(x)(-h) + \frac{f''(x)(-h)^2}{2} + \frac{f'''(x)(-h)^3}{6} + \dots$$

$$f(x+h) - f(x-h) = 2f'(x)h + \frac{f'''(x)2h^3}{6} + \dots$$

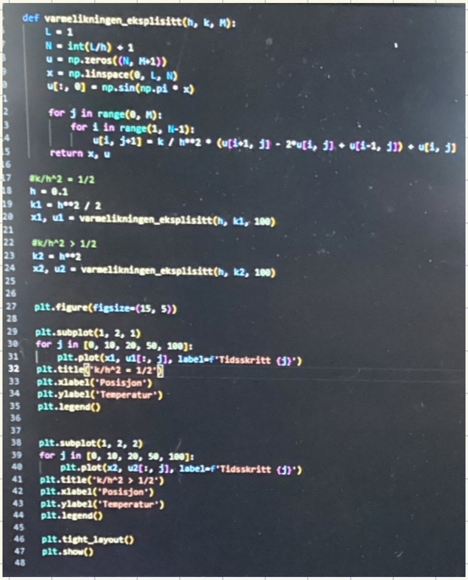
$$\frac{f(x+h)-f(x-h)}{2h} = f'(x) + \underbrace{\frac{f'''(x)h^2}{6} + \dots}_{\text{Feilen}}$$

Med dette utrykket blir feilen mindre siden leddet $\frac{f''(x)}{2}h$ og hvert andre ledd etter det ikke er med uttrykket som beskriver feilen

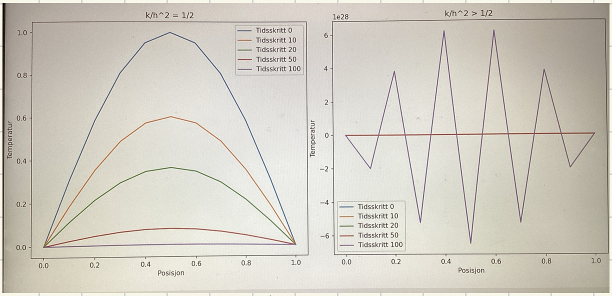
$\boxed{3}$ $f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}$

$h = 0{,}1$: $f'(1{,}5) = 4{,}4817$

$\boxed{4}$



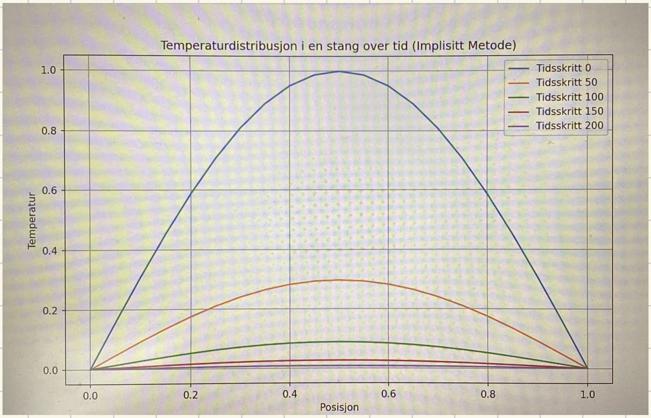
```python
def varmelikningen_eksplisitt(h, k, M):
    L = 1
    N = int(L/h) + 1
    u = np.zeros((N, M+1))
    x = np.linspace(0, L, N)
    u[:, 0] = np.sin(np.pi * x)

    for j in range(0, M):
        for i in range(1, N-1):
            u[i, j+1] = k / h**2 * (u[i+1, j] - 2*u[i, j] + u[i-1, j]) + u[i, j]
    return x, u

#k/h^2 = 1/2
h = 0.1
k1 = h**2 / 2
x1, u1 = varmelikningen_eksplisitt(h, k1, 100)

#k/h^2 > 1/2
k2 = h**2
x2, u2 = varmelikningen_eksplisitt(h, k2, 100)


plt.figure(figsize=(15, 5))

plt.subplot(1, 2, 1)
for j in [0, 10, 20, 50, 100]:
    plt.plot(x1, u1[:, j], label=f'Tidsskritt {j}')
plt.title('k/h^2 = 1/2')
plt.xlabel('Posisjon')
plt.ylabel('Temperatur')
plt.legend()


plt.subplot(1, 2, 2)
for j in [0, 10, 20, 50, 100]:
    plt.plot(x2, u2[:, j], label=f'Tidsskritt {j}')
plt.title('k/h^2 > 1/2')
plt.xlabel('Posisjon')
plt.ylabel('Temperatur')
plt.legend()

plt.tight_layout()
plt.show()
```



Ser her at temp. en synker gradvis med tiden for $k/h^2 = 1/2$.

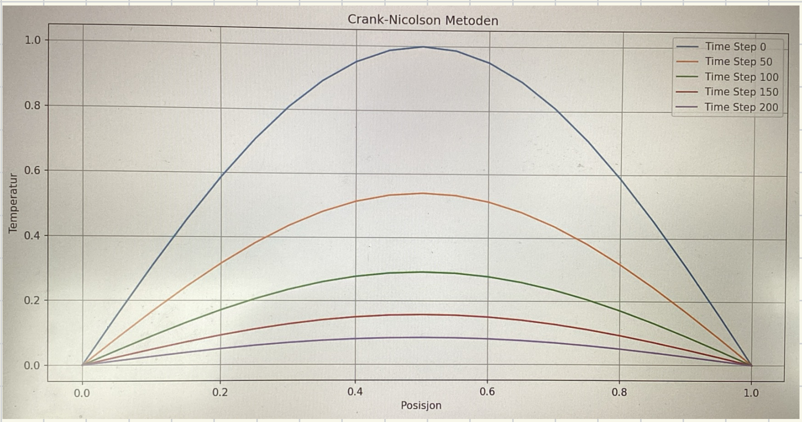
$k/h^2 > 1/2$ er ustabil

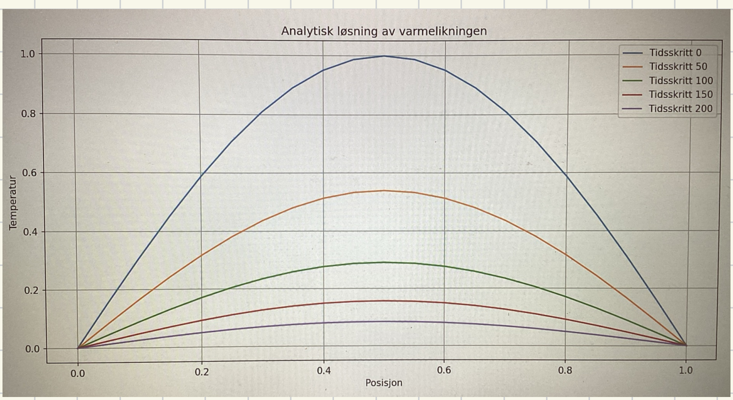
$\boxed{5}$

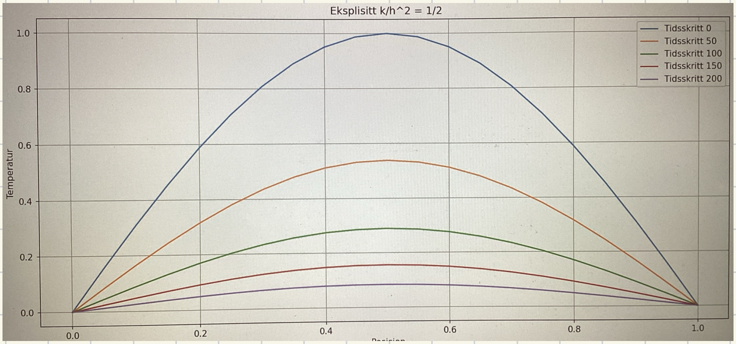


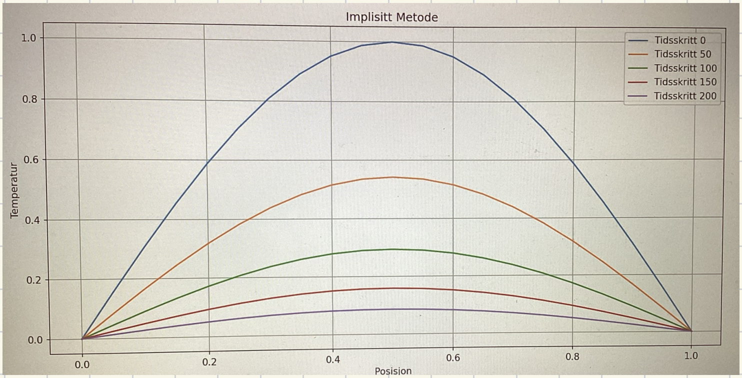
Får et liknende resultat som ved eksplisitt metode, men her uten kravet $k/h^2 \geq 1/2$.

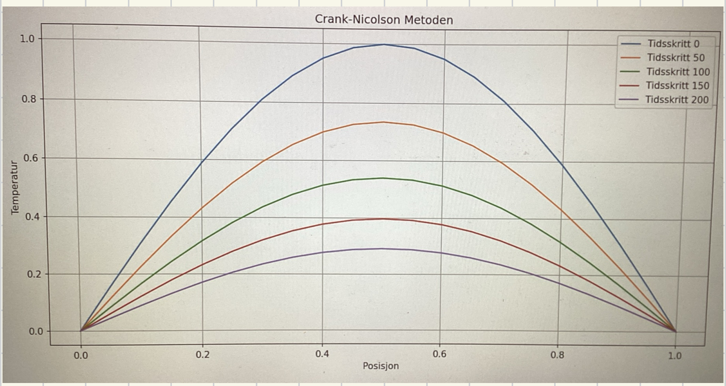
$\boxed{6}$ Crank nicholson fungerer som et kompromiss og gir både god nøyaktighet og stabilitet











Sammenlikner de tre metodene med den analytiske løsningen for $h = 0.05$ og $k = h^2/2$, slik at $k/h^2 = \frac{1}{2}$. Siden kravet $k/h^2 \geq 1/2$ fungerer eksplisitt metode bra. Implisitt metode fungerer også godt her. Crank-Nicolson metoden gir ikke en like god løsning med parameterne som ble brukt her, men for å få en god balanse mellom nøyaktighet og stabilitet over en lengere tid er visstnok denne metoden den beste.