OBLIG Matte

1) `4.5041723976187775` h = 0.01

`4.483930662007474` h = 0.001

Her gikk det skeis: `h = 0.00000000000001` `4.440892098500626`

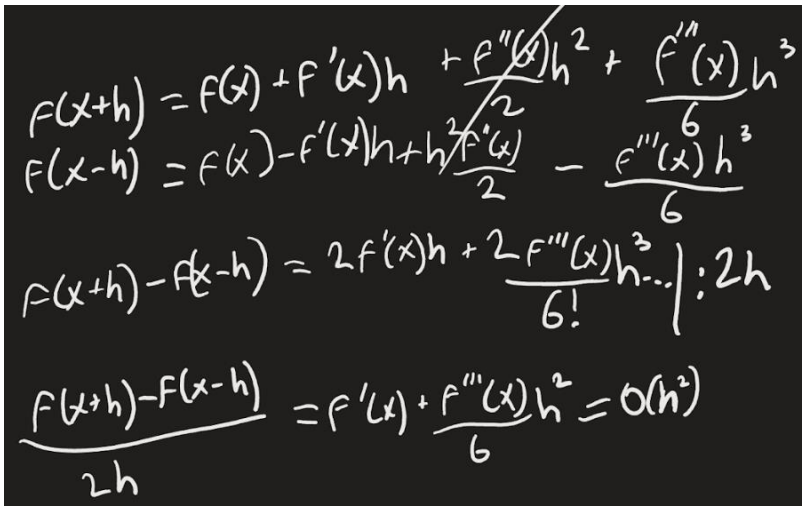
2) `4.504545316719234` h = 0.01

`4.483934396189389` h = 0.001

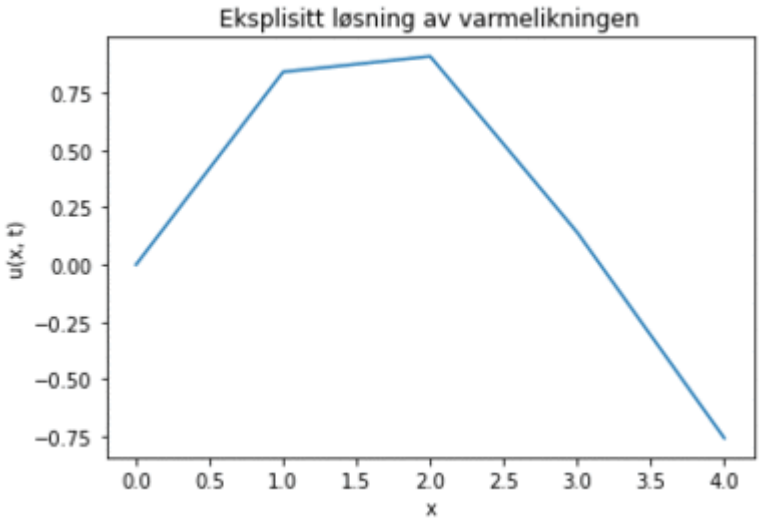
`h = 0.000000000000001` `4.476419235288631`

Går skeis her

Her er forklaring:



3) Går skeis her: `h = 0.000000000000001`

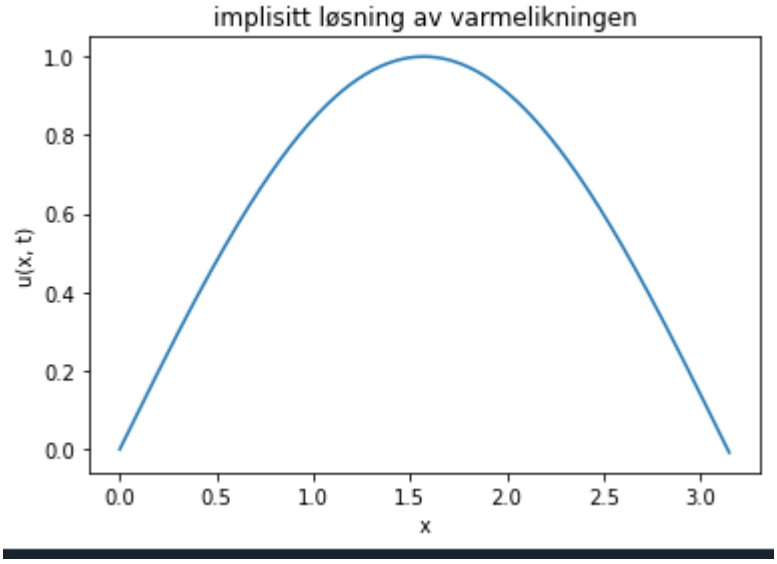


4)

```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

#Setter variabler
L = np.pi
T = 2.0
h = 0.01
k = 0.01

#Setter intervall
x = np.arange(0, L + h, h)
t = np.arange(0, T + k, k)
Nx = len(x)
Nt = len(t)

#danner null-matriser og initialbettingelse
u = np.zeros((Nx, Nt))
u[:, 0] = np.sin(x)

#iterer gjennom eksplisitt euler for x og deretter for t
for j in range(0, Nt - 1):
    for i in range(1, Nx - 1):
        u[i, j+1] = u[i, j] + k * (u[i+1, j] - 2*u[i, j] + u[i-1, j]) / h**2
#plotter
fig, ax = plt.subplots()
line, = ax.plot(x, u[:, 0])


plt.xlabel('x')
plt.ylabel('u(x, t)')
plt.title('Eksplisitt løsning av varmelikningen')
plt.show()
```

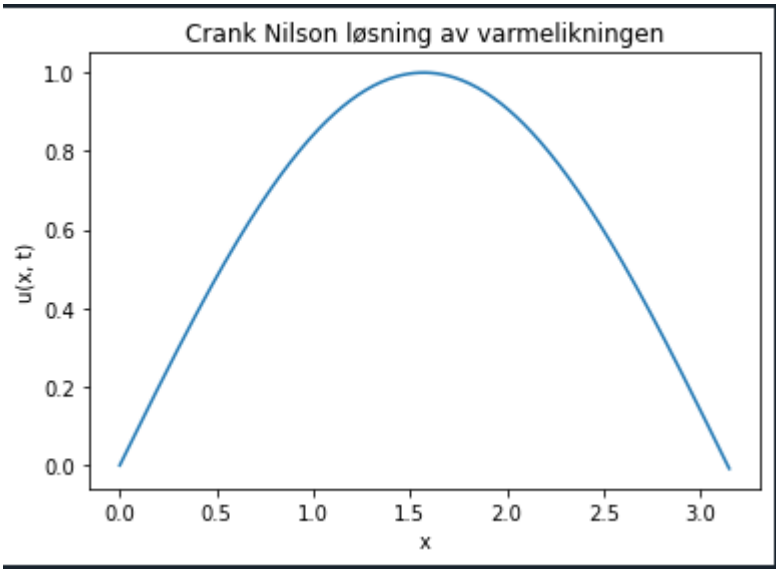
b)

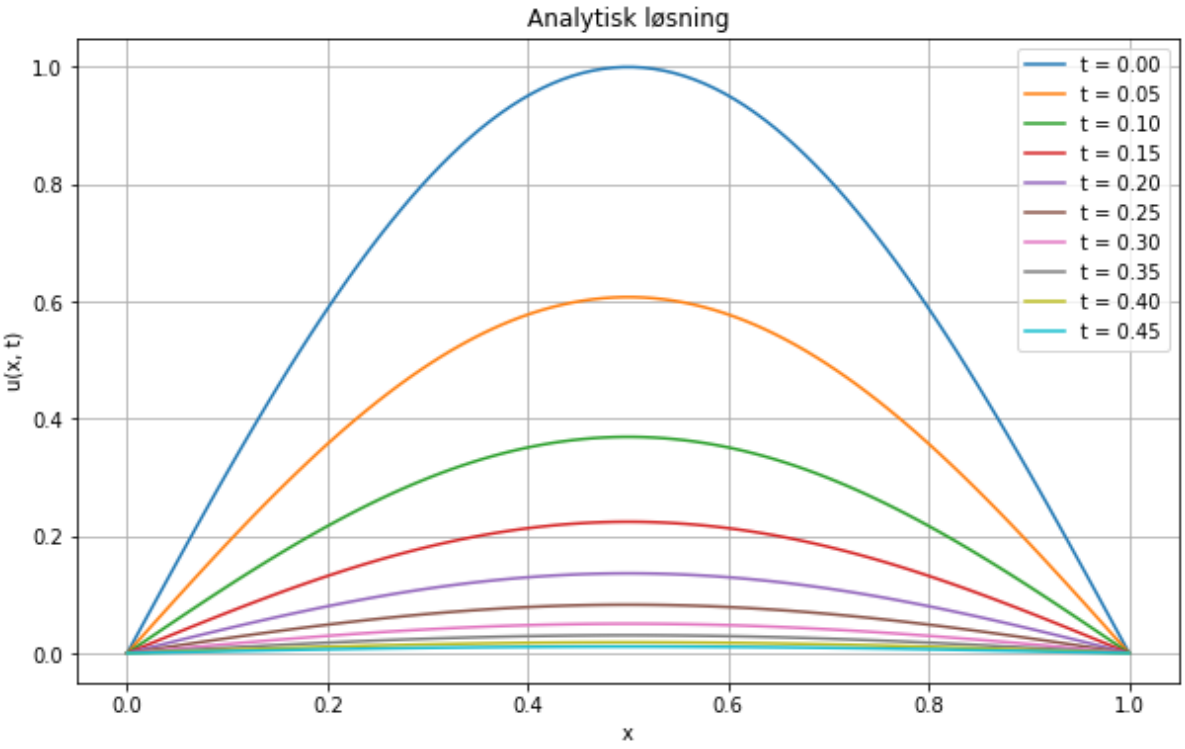
```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

#Setter variabler
L = np.pi
T = 2.0
h = 0.01
k = 0.01

#Setter intervall
x = np.arange(0, L + h, h)
t = np.arange(0, T + k, k)
Nx = len(x)
Nt = len(t)

#danner null-matriser og initialbettingelse
u = np.zeros((Nx, Nt))
u[:, 0] = np.sin(x)

#iterer gjennom eksplisitt euler for x og deretter for t
for j in range(0, Nt - 1):
    for i in range(1, Nx - 1):
        u[i, j+1] = (u[i, j] + (k / h**2) * (u[i+1, j+1] + u[i-1, j+1])) / (1 + 2 * (k / h**2))
#plotter
fig, ax = plt.subplots()
line, = ax.plot(x, u[:, 0])


plt.xlabel('x')
plt.ylabel('u(x, t)')
plt.title('implisitt løsning av varmelikningen')
plt.show()
```



c)

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

#Setter variabler
L = np.pi
T = 2.0
h = 0.01
k = 0.01

#Setter intervall
x = np.arange(0, L + h, h)
t = np.arange(0, T + k, k)
Nx = len(x)
Nt = len(t)

#danner null-matriser og initialbettingelse
u = np.zeros((Nx, Nt))
u[:, 0] = np.sin(x)

#iterer gjennom eksplisitt euler for x og deretter for t
for j in range(0, Nt - 1):
    for i in range(1, Nx - 1):
        u[i, j+1] = u[i, j] + k * ((u[i+1, j] - 2 * u[i, j] + u[i-1, j]) / (2 * h**2) + (u[i+1, j+1] - 2 * u[i, j+1] + u[i-1, j+1]) / (2 * h**2))

#plotter
fig, ax = plt.subplots()
line, = ax.plot(x, u[:, 0])


plt.xlabel('x')
plt.ylabel('u(x, t)')
plt.title('Crank Nilson løsning av varmelikningen')
plt.show()

x = np.arange(0, 2*np.pi, 0.01)
line, = ax.plot(x, np.sin(x))
```




Analytisk løsning
u(x, t)
x
0.0
0.2
0.4
0.6
0.8
1.0
t = 0.00
t = 0.05
t = 0.10
t = 0.15
t = 0.20
t = 0.25
t = 0.30
t = 0.35
t = 0.40
t = 0.45