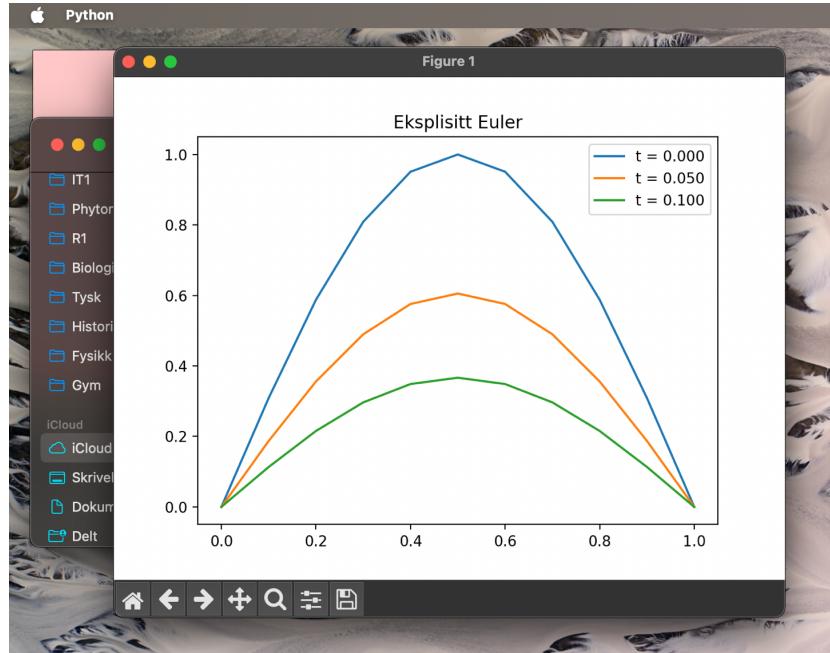


Oblig Matte 2

Eksplisitt Euler

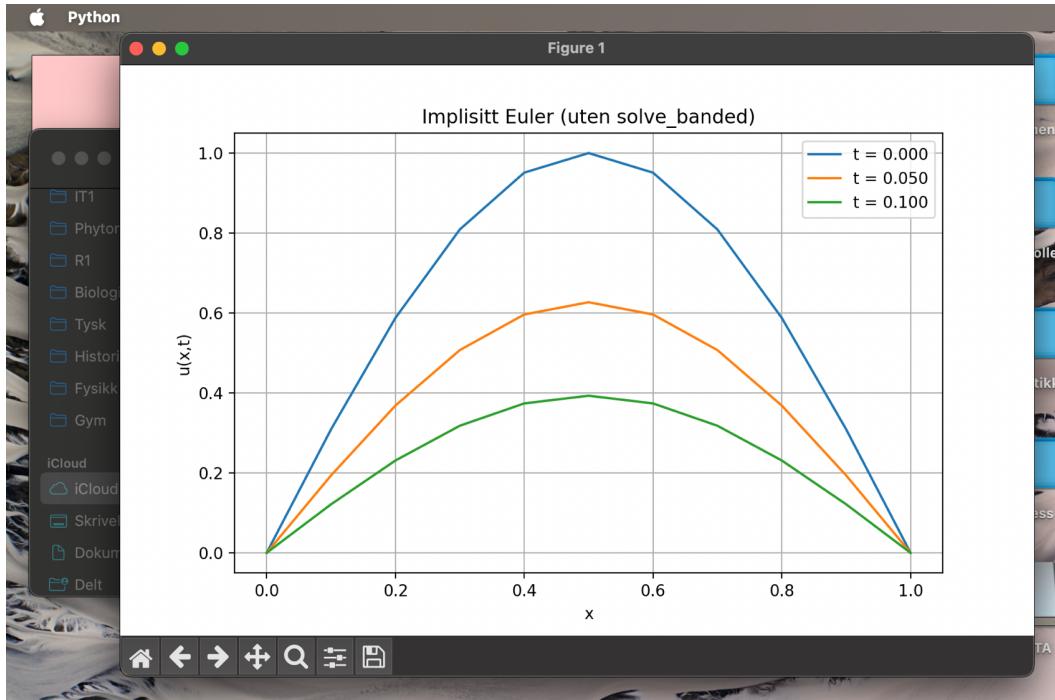


UNIMATTE2

```
oppg4.py > ...
1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  def explicit_euler(h, k, T):
5      N = int(1/h) + 1
6      M = int(T/k) + 1
7      u = np.zeros((N, M))
8      x = np.linspace(0, 1, N)
9
10     u[:, 0] = np.sin(np.pi * x)
11
12     for j in range(M-1):
13         for i in range(1, N-1):
14             u[i, j+1] = u[i, j] + (k/h**2) * (u[i+1, j] - 2*u[i, j] + u[i-1, j])
15
16     return x, u
17
18 h = 0.1
19 k = 0.005
20 T = 0.1
21 x, u = explicit_euler(h, k, T)
22
23 for j in [0, 10, 20]:
24     plt.plot(x, u[:, j], label=f"t = {j*k:.3f}")
25 plt.title("Eksplisitt Euler")
26 plt.legend()
27 plt.show()
```

Ln 22, Col 1 Spaces: 4 UTF-8 LF ⚡ Python 3.10.2 64-bit Go Live

Implisitt Euler



UNIMATTE2

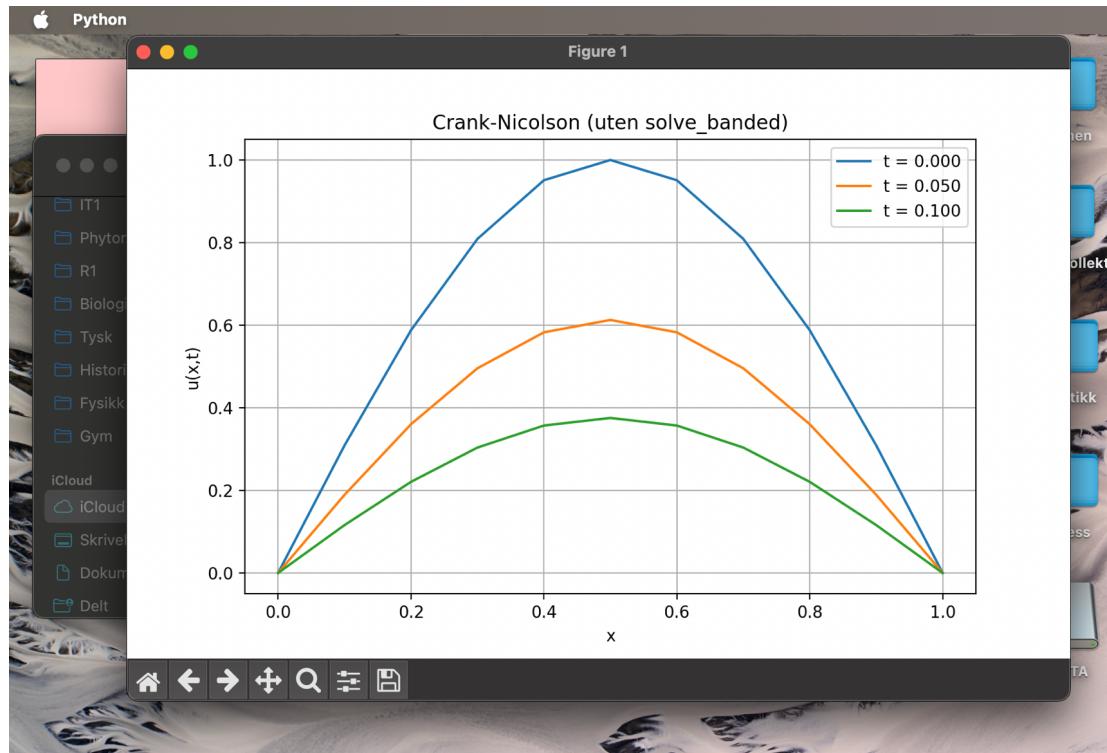
```
oppg4.py oppg5.py oppg6.py
```

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def implicit_euler_simple(h, k, T):
5     N = int(1/h) + 1
6     M = int(T/k) + 1
7     u = np.zeros((N, M))
8     x = np.linspace(0, 1, N)
9
10    u[:, 0] = np.sin(np.pi * x)
11
12    r = k / h**2
13    A = np.zeros((N-2, N-2))
14    for i in range(N-2):
15        A[i, i+1] = 1 + 2*r
16        if i > 0:
17            A[i, i-1] = -r
18        if i < N-3:
19            A[i, i+2] = -r
20
21    for j in range(M-1):
22        b = u[:, j+1]
23        u[:, j+1] = np.linalg.solve(A, b)
24
25    return x, u
26
27 h = 0.1
28 k = 0.01
29 T = 0.1
30
31 x, u = implicit_euler_simple(h, k, T)
32
33 M = u.shape[1]
34
35 plt.figure(figsize=(8, 5))
36 for j in [0, int(M/2), M-1]:
37     plt.plot(x, u[:, j], label=f't = {j*k:.3f}')
38 plt.title("Implisitt Euler (uten solve_banded)")
39 plt.xlabel("x")
40 plt.ylabel("u(x,t)")
41 plt.legend()
42 plt.grid()
43 plt.show()
```

cecilialestrom@Cecilias-MacBook-Pro UNImatte2 % cd "/Users/cecilialestrom/Library/Mobile Documents/com-apple-CloudDocs/UNImatte2"
/usr/local/bin/python3 "/Users/cecilialestrom/UNImatte2/oppg5.py"
cecilialestrom@Cecilias-MacBook-Pro UNImatte2 % /usr/local/bin/python3 "/Users/cecilialestrom/Library/Mobile Documents/com-apple-CloudDocs/UNImatte2/oppg5.py"
2025-03-26 19:32:47.149 Python[34615:2516588] +[IMKClient subclass]: chose IMKClient_Modern
2025-03-26 19:32:47.149 Python[34615:2516588] +[IMKInputSession subclass]: chose IMKInputSession_Modern

Python

Crank Nicolson



The screenshot shows a Jupyter Notebook interface with the following details:

- EXPLORER**: A sidebar showing the file structure with files `oppg4.py`, `oppg5.py`, and `oppg6.py`.
- TERMINAL**: The terminal window shows command-line history related to the execution of `oppg6.py`. It includes commands like `python oppg6.py` and `ipython oppg6.py`, along with session IDs and file paths.
- OUTPUT**: The main notebook area displays the Python code for `oppg6.py`. The code defines a function `crank_nicolson_simple(h, k, T)` which solves a banded system of equations using `np.linalg.solve`. It also contains a loop that plots the solution `u` at various time steps `t = j*k`.

OÜig mathe 2

$$1. \quad f'(x) = \frac{f(x+L) - f(x)}{L} \quad \text{für } f(x) = e^x$$

$$\Rightarrow x = 1,5$$

L	tilnærmet $f'(1,5)$	skaltes $f'(1,5) = e^{1,5}$	feil
0,1	4,7134	4,9817	0,237
0,01	4,4984		0,0167
0,001	4,4835		0,0018
10^{-4}	4,4820		0,0003
10^{-5}	4,4818		0,0001
10^{-6}	4,4817		0

für $L < 10^{-6}$ begynner numerisk etat i denne regnemaskinen

$$2. \quad f'(x) = \frac{f(x+L) - f(x-L)}{2L}$$

L	tilnærmet $f'(1,5)$	feil
0,1	4,819	0,0002
0,01	4,9817	0
0,001	4,4817	0

forhåndssetning med Taylor utvikling:

$$f(x+L) = f(x) + f'(x)L + \frac{f''(x)}{2}L^2 + O(L^3)$$

$$f(x-L) = f(x) - f'(x)L + \frac{f''(x)}{2}L^2 + O(L^3)$$

$$f(x+L) - f(x-L) = 2f'(x)L + O(L^3)$$

$$f'(x) = \frac{f(x+L) - f(x-L)}{2L} + O(L^2)$$

feilen er da proportional med L^2
derfor manger er mykhet

$$3. \quad f'(x) = \frac{f(x-2L) - 8f(x-L) + 8f(x+L) - f(x+2L)}{12L}$$

er feilt av orden $O(L^4)$, men er mykhet

$$4, 5, 6 \quad u(x, t) = u''(x, t)$$

Wendes diskretisierung:

$$u''(x_i, t) = \frac{u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)}{h^2}$$

Zeitsdiskretisierung:

• Explicit Euler:

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

$$u_{i,j+1} = u_{i,j} + \frac{\Delta t}{h^2} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j})$$

$$\text{Stabilitätskriterium: } \frac{\Delta t}{h^2} \leq \frac{1}{2}$$

• Implicit Euler

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

• Crank-Nicolson

$$\frac{u_{i,j+1} - u_{i,j}}{\Delta t} = \frac{1}{2} \left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \right.$$

$$\left. \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{h^2} \right)$$

• Abwärtsitth Euler: Lösungen nur stabil
 $\Delta t \leq h^2/2$

• auwärtitth Euler: Fehlzeiten für alle
 $\Delta t > 0$

• Crank-Nicolson: Fehlzeiten für
alle $\Delta t > 0$

Metode	Stabilität	Zeitstabilität	Bedingungssatz
Explicit Euler	$\Delta t \leq h^2/2$	$O(h^2 + \Delta t)$	kurz
Implicit Euler	alle $\Delta t > 0$	$O(h^2 + \Delta t)$	lang
Crank-Nicolson	alle $\Delta t > 0$	$O(h^2 + \Delta t^2)$	lang