

example

October 16, 2025

0.1 # NumIntro Implementeringer

Robuste implementeringer af algoritmerne fra NumIntro ## Overblik

Fungere med simple imports og har samme metodik som dem givet i Python recipies.

0.2 Eksempel Ikke-linære ligninger

Med en simpel import kan vi hurtigt køre algoritmen

```
[24]: from non_linear_solver.main import newtons_method
import pandas as pd

f = lambda x: x**2
f_prime = lambda x: 2*x
x_0 = 2.0

root, n_iters, did_converge, history = newtons_method(f=f, fprime=f_prime,
↳ x0=x_0, epsilon=10**(-6), K=200, return_history=True)
root
```

```
[24]: 0.0009765625
```

Men vi kan også hurtigt få fat på iterations historikken

```
[25]: pd.DataFrame(history)
```

```
[25]:
```

	iteration	x_k	x_k+1	f(x_k)	x_k+1 - x_k	f(x_k+1) <epsilon
0	1	2.000000	1.000000	4.000000	1.000000	False
1	2	1.000000	0.500000	1.000000	0.500000	False
2	3	0.500000	0.250000	0.250000	0.250000	False
3	4	0.250000	0.125000	0.062500	0.125000	False
4	5	0.125000	0.062500	0.015625	0.062500	False
5	6	0.062500	0.031250	0.003906	0.031250	False
6	7	0.031250	0.015625	0.000977	0.015625	False
7	8	0.015625	0.007812	0.000244	0.007812	False
8	9	0.007812	0.003906	0.000061	0.003906	False
9	10	0.003906	0.001953	0.000015	0.001953	False

```
10          11  0.001953  0.000977  0.000004          0.000977          True
```

0.3 Eksempel - LinAlg

```
[3]: from linear_systems.main import LUfactorization
import numpy as np
```

```
[11]: A = np.array([
        [1,2,3],
        [4,5,6],
        [7,8,9],
    ])
    L, U = LUfactorization(A)
    print(L, "\n", U)
```

```
[[1. 0. 0.]
 [4. 1. 0.]
 [7. 2. 1.]]
[[ 1.  2.  3.]
 [ 0. -3. -6.]
 [ 0.  0.  0.]]
```

```
[12]: L @ U
```

```
[12]: array([[1., 2., 3.],
            [4., 5., 6.],
            [7., 8., 9.]])
```