Aflevering 4

a)

Betragt vektorerne

$$v_0 = \begin{bmatrix} 1{,}0 \\ -1{,}0 \\ 1{,}0 \\ -1{,}0 \end{bmatrix}, \quad v_1 = \begin{bmatrix} 1{,}0 \\ 1{,}0 \\ 1{,}0 \\ 1{,}0 \end{bmatrix}, \quad v_2 = \begin{bmatrix} 2{,}0 \\ 0{,}0 \\ -2{,}0 \\ 0{,}0 \end{bmatrix}$$

i $\mathbb{R}^4$.

(a) Dan Grammatricen for v_0, v_1, v_2 og bekræft at denne samling vektorer er ortogonal.

```
import numpy as np
v0 = np.array([1.0, -1.0, 1.0, -1.0])
v1 = np.array([1.0,  1.0, 1.0,  1.0])
v2 = np.array([2.0,  0.0,-2.0,  0.0])
V = np.vstack([v0, v1, v2])
G1 =  V @ V.T

print('G=\n', G1)
```

```
## G=
##  [[4. 0. 0.]
##  [0. 4. 0.]
##  [0. 0. 8.]]
```

Når vi har et sæt af vektorer og vi vil vurdere om de er ortogonale, så skal Gram matricen indgange udover diagonalen være nul, som er tilfældet i ovenstående.

b)

(b) Beregn projektionen Px af

$$x = \begin{bmatrix} 3{,}0 \\ 2{,}0 \\ 1{,}0 \\ 0{,}0 \end{bmatrix}$$

langs v_0, v_1, v_2.

Til en start opskriver jeg x.

```
x = np.array([3., 2., 1., 0.])
```

Herefter bestemmes normel.

```
v0_norm = np.linalg.norm(v0)
v1_norm = np.linalg.norm(v1)
v2_norm = np.linalg.norm(v2)
```

Nu kan vi således bestemme projektion af x langs v0.

```
prov0 = (np.dot(v0, x) / v0_norm**2) * v0
prov1 = (np.dot(v1, x) / v1_norm**2) * v1
prov2 = (np.dot(v2, x) / v2_norm**2) * v2
Px = prov0 + prov1 + prov2
print("Px =\n", Px)
```

```
## Px =
##  [3. 1. 1. 1.]
```

c)

(c) Bekræft at $v_3 := x - Px$ er ortogonal til v_0, v_1 og v_2.

```
v3 = np.round(x - Px)

print("v3 og v0 er ortogonale da deres indre produkt er", np.dot(v0,v3))
```

```
## v3 og v0 er ortogonale da deres indre produkt er 0.0
```

```
print("v3 og v0 er ortogonale da deres indre produkt er", np.dot(v1,v3))
```

```
## v3 og v0 er ortogonale da deres indre produkt er 0.0
```

```
print("v3 og v0 er ortogonale da deres indre produkt er", np.dot(v2,v3))
```

```
## v3 og v0 er ortogonale da deres indre produkt er 0.0
```

d)

(d) Brug v_0, v_1, v_2, v_3 til at bestemme en ortonormal basis for $\mathbb{R}^4$.

```
V = np.vstack([v0 / v0_norm, v1 / v1_norm, v2 / v2_norm, v3/np.linalg.norm(v3)])
print(V, '\n')
```

```
## [[ 0.5        -0.5         0.5        -0.5       ]
##  [ 0.5         0.5         0.5         0.5       ]
##  [ 0.70710678  0.         -0.70710678  0.        ]
##  [ 0.          0.70710678 -0.         -0.70710678]]
```

```
print(np.round(V.T @ V))
```

```
## [[1. 0. 0. 0.]
##  [0. 1. 0. 0.]
##  [0. 0. 1. 0.]
##  [0. 0. 0. 1.]]
```

SÅ har vi en ortonormal basis og vi ser vi får samme diagonal som i opgave a.