Afleveringsopgave 4

Dette er en individuel opgave. Opgaveløsning afleveres i brightspace som én pdf fil under “Course Tools > Assignments > Aflevering 4”. Afleveringsfristen bestemmes af din instruktor, men ligger i uge 11.

Betragt vektorerne

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

i $\mathbb{R}^4$.

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

(b) Beregn projektionen Px af

$$x = \begin{bmatrix} 4{,}0 \\ 3{,}0 \\ 2{,}0 \\ 1{,}0 \end{bmatrix}$$

på samlingen v_0, v_1, v_2.

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

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

Kort forklaring af opgaveløsningen

(a) Beregning af Gram-matricen G

Vi laver Gram-matricen ved at tage skalarproduktet af vektorerne med hinanden. Gram-matricen opstilles således:

$$G = \begin{bmatrix} (v_0, v_0) & (v_0, v_1) & (v_0, v_2) \\ (v_1, v_0) & (v_1, v_1) & (v_1, v_2) \\ (v_2, v_0) & (v_2, v_1) & (v_2, v_2) \end{bmatrix}$$

(b) Beregning af projektionen (P(x))

Vi projicerer x ned på underrummet udspændt af v0, v1 og v2.

(c) Bekræftelse af ortogonalitet for (v_3 = x - P(x))

Vi tjekker, om v3 er ortogonal til v0, v1 og v2 ved at tage skalarproduktet med hver af dem. Hvis alle giver 0, er v3 ortogonal.

(d) Bestemmelse af en ortonormal basis for (R^4)

Vi bruger tager normen af vores ortogonale vektorer for at bestemme den ortonormale basis.

In [1]:
```
import numpy as np

def gram_matrix(vectors):
    """
    Danner Gram-matricen G = (v_i dot v_j) for en liste af vektorer i R^n
```

```
    ud fra form vist på notessæt side 112.

    -- KORT FORKLARING --
    For en samling vektorer [v0, v1, ..., vm-1] defineres Gram-matricen G
    som G[i,j] = <v_i, v_j>, hvor <·,·> er standard-indre produkt (Definition 8.

    Parametre:
      vectors : list[np.ndarray], hver vektor i R^n
    Returnerer:
      G : np.ndarray (m x m), hvor m = antal vektorer
    """
    print("=== gram_matrix ===")
    m = len(vectors)
    if m == 0:
        print("Tom liste af vektorer => returnerer tom matrix.")
        return np.array([[]])

    n = vectors[0].shape[0]
    print(f"Danner Gram-matrix for {m} vektorer i R^{n} ...")
    G = np.zeros((m, m), dtype=float)
    for i in range(m):
        for j in range(m):
            G[i, j] = np.dot(vectors[i], vectors[j])
    print("=> Gram-matrix konstrueret.:")
    print(G)  # Udskriv Gram-matricen
    return G


def project_x_onto_span(vectors, x):
    """
    Projekterer vektor x ortogonalt ned på span(vectors) ved at anvende
    formel (8.12) fra Definition 8.19.

    Parametre:
      vectors : list[np.ndarray] (hver i R^n)
      x       : np.ndarray (n,) - vektor der skal projekteres

    Returnerer:
      Px      : np.ndarray (n,) eller None, hvis Gram-matricen ikke er invertibe
    """
    import numpy as np

    print("=== project_x_onto_span ===")
    print("Anvender formel (8.12) fra Definition 8.19 til ortogonal projektion:\
          "  P = A (A^T A)^{-1} A^T,  hvor A har søjlerne v1, v2, ..., vk.")

    m = len(vectors)
    if m == 0:
        print("Ingen vektorer => projektion = 0.")
        return np.zeros_like(x)

    # Dan matrix A ved at sætte vektorerne som søjler
    # (alle vektorer skal være samme dimension n)
    A = np.column_stack(vectors)  # A bliver (n x m)

    # Gram-matricen G = A^T * A (størrelse m x m)
    G = A.T @ A

    # Tjek om G er invertibel (dvs. determinant != 0)
    det_G = np.linalg.det(G)
```

```
    if abs(det_G) < 1e-14:
        print("Gram-matricen er singulær (ikke-invertibel) => returnerer None.")
        return None

    # Beregn G^-1
    G_inv = np.linalg.inv(G)

    # Beregn projektionen Px = A * (G^-1 * (A^T * x))
    Px = A @ (G_inv @ (A.T @ x))

    print("Projektion P(x) =", Px)
    print("=> Projektion fuldført ved brug af (8.12).")
    return Px

def normalize_orthogonal_vectors(vectors):
    """
    Antager, at 'vectors' allerede er indbyrdes ortogonale.
    Returnerer en liste af normaliserede vektorer (ortonormale) ved at gøre dem
    """
    onb = []
    for v in vectors:
        normv = np.linalg.norm(v)
        if normv < 1e-14:
            # Hvis normen er (næsten) 0, lægges en nulvektor ind
            onb.append(np.zeros_like(v))
        else:
            onb.append(v / normv)
    return onb

# --- Definer vektorer fra opgaven ---
v0 = np.array([1, -1, 1, -1], dtype=float)
v1 = np.array([1,  1,-1, -1], dtype=float)
v2 = np.array([0,  2, 2,  0], dtype=float)
x  = np.array([4,  3, 2,  1], dtype=float)

# (a) Beregn Gram-matricen for v0, v1, v2
print("=== (a) Beregning af Gram-matrix ===")
G = gram_matrix([v0, v1, v2])  # Funktion udskriver automatisk info

# (b) Beregn projektionen P(x) af x på Span(v0, v1, v2)
print("\n=== (b) Beregning af Projektion P(x) ===")
Px = project_x_onto_span([v0, v1, v2], x)  # Funktion udskriver automatisk info

# (c) Beregn v3 = x - P(x) og tjek ortogonalitet med v0, v1, v2
print("\n=== (c) Beregning af v3 = x - P(x) ===")
v3 = x - Px
print("v3 =", v3)
print("Tjek ortogonalitet: Indre produkt mellem v3 og (v0, v1, v2):")
dot_v3_v0 = np.dot(v3, v0)
dot_v3_v1 = np.dot(v3, v1)
dot_v3_v2 = np.dot(v3, v2)
print(f"v3·v0 = {dot_v3_v0}, v3·v1 = {dot_v3_v1}, v3·v2 = {dot_v3_v2}")
print("Hvis disse er (næsten) 0, er v3 ortogonal til v0, v1, v2.")

# (d) Bestem en ortonormal basis for R^4 ved blot at normalisere (v0, v1, v2, v3
print("\n=== (d) Bestemmelse af Ortonormal Basis ved normalisering af v0, v1, v2
onb = normalize_orthogonal_vectors([v0, v1, v2, v3])

# Udskriv den ortonormale basis
print("\nOrtonormal basis for R^4:")
```

```
for i, vec in enumerate(onb):
    print(f"u_{i} = {vec}")
```

```
=== (a) Beregning af Gram-matrix ===
=== gram_matrix ===
Danner Gram-matrix for 3 vektorer i R^4 ...
=> Gram-matrix konstrueret.:
[[4. 0. 0.]
 [0. 4. 0.]
 [0. 0. 8.]]

=== (b) Beregning af Projektion P(x) ===
=== project_x_onto_span ===
Anvender formel (8.12) fra Definition 8.19 til ortogonal projektion:
  P = A (A^T A)^{-1} A^T,  hvor A har søjlerne v1, v2, ..., vk.
Projektion P(x) = [ 1.5  3.   2.  -1.5]
=> Projektion fuldført ved brug af (8.12).

=== (c) Beregning af v3 = x - P(x) ===
v3 = [2.5 0.  0.  2.5]
Tjek ortogonalitet: Indre produkt mellem v3 og (v0, v1, v2):
v3·v0 = 0.0, v3·v1 = 0.0, v3·v2 = 0.0
Hvis disse er (næsten) 0, er v3 ortogonal til v0, v1, v2.

=== (d) Bestemmelse af Ortonormal Basis ved normalisering af v0, v1, v2, v3 per D
ef 8.11===

Ortonormal basis for R^4:
u_0 = [ 0.5 -0.5  0.5 -0.5]
u_1 = [ 0.5  0.5 -0.5 -0.5]
u_2 = [0.         0.70710678 0.70710678 0.        ]
u_3 = [0.70710678 0.         0.         0.70710678]
```

Konklusion på hver opgave

a)

Da vi hermed kan se at alle indre produkter af vektorene ganget sammet med hinanden giver 0 i vores gram-matrice, ved ved vi at samlingen af vektorer er ortogonal (Se ligning 8.1 i notessæt for definition på ortogonalitet og side 112 notessætfor def af gram matrice).

b)

Hermed har vi lavet projektionen P(x) af x på samlingen {v_0,v_1,v_2} vha Def 8.19 i notessæt og fundet følgende resultat.

c)

Hermed ses vektoren v_3 er ortogonal til {v_0,v_1,v_2} da (v_3,v_i)=0 for i=0,1,2. Dette bekræfter, at projektionen er beregnet korrekt, da alt "tilbageværende" (dvs. v_3) må ligge vinkelret på det rum, vi projekterede på.

d)

Hermed tager vi vores ortogonale samling af vektorer og tager normaliserer dem, så vi får en ortonormal samling.

Notessæt formler og definitioner brugt i løsning:

Definition 8.19. For $v_0, v_1, \ldots, v_{k-1}$ ortogonal, med alle $v_i \neq 0$, er *projektionen af u på samlingen $v_0, v_1, \ldots, v_{k-1}$* givet ved Pu, hvor P er matricen

$$P = \frac{1}{\|v_0\|_2^2} v_0 v_0^T + \frac{1}{\|v_1\|_2^2} v_1 v_1^T + \cdots + \frac{1}{\|v_{k-1}\|_2^2} v_{k-1} v_{k-1}^T. \tag{8.12}$$

Bemærkning 8.20. Når man beregner projektionen i python, gør man det typisk uden at opbygge selve projektionsmatricen. Ofte ønsker man blot at beregne Pw for en vektor w, og dette gives ved

$$Pw = \mathrm{pr}_{v_0}(w) + \mathrm{pr}_{v_1}(w) + \cdots + \mathrm{pr}_{v_{k-1}}(w),$$

med hver enkelt led som i (8.7). Se senere eksempler. ◇

8.1 Standard indre produkt

For vektorer $u, v \in \mathbb{R}^n$, definerer vi deres standard *indre produkt* til at være

$$\begin{aligned} \langle u, v \rangle &= u^T v \\ &= \begin{bmatrix} u_0 & u_1 & \ldots & u_{n-1} \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{bmatrix} \\ &= u_0 v_0 + u_1 v_1 + \cdots + u_{n-1} v_{n-1}. \end{aligned} \tag{8.1}$$

Vi siger at u, v er *ortogonal*, eller at v står *vinkelret* på u, og skriver $u \perp v$, hvis

$$\langle u, v \rangle = 0.$$

9.3 Ortonormale vektorer og ortogonale matricer

Givet en samling vektorer $v_0, v_1, \dots, v_{k-1}$ i $\mathbb{R}^n$, kan vi danne en matrix V med disse vektorer som søjler

$$V = \left[v_0 \mid v_1 \mid \dots \mid v_{k-1}\right] \in \mathbb{R}^{n\times k}.$$

Matricen

$$G = V^T V \in \mathbb{R}^{k\times k}$$

kaldes *Grammatricen* for $v_0, \dots, v_{k-1}$, efter Jørgen Pedersen Gram (1850–1916). Vi har

$$\begin{aligned} G \coloneqq V^T V &= \begin{bmatrix} v_0^T \\ v_1^T \\ \vdots \\ v_{k-1}^T \end{bmatrix} \left[v_0 \mid v_1 \mid \dots \mid v_{k-1}\right] \\ &= \begin{bmatrix} v_0^T v_0 & v_0^T v_1 & \dots & v_0^T v_{k-1} \\ v_1^T v_0 & v_1^T v_1 & \dots & v_1^T v_{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{k-1}^T v_0 & v_{k-1}^T v_1 & \dots & v_{k-1}^T v_{k-1} \end{bmatrix}. \end{aligned}$$

Dette sige at den (i, j)te indgang i G er det indre produkt mellem v_i og v_j:

$$G = (g_{ij}) = (v_i^T v_j) = \left(\langle v_i, v_j \rangle\right).$$

Som direkte konsekvens har vi det følgende resultat.

Definition 8.11. En samling vektorer $v_0, v_1, \dots, v_{k-1}$ er *ortogonal* hvis

$$\langle v_i, v_j \rangle = 0, \qquad \text{for alle } i \neq j.$$

Hvis alle vektorerne $v_0, v_1, \dots, v_{k-1}$ er desuden enhedsvektor, så siger vi at samlingen er *ortonormal*. Ortonormal er det sammen som

$$\langle v_i, v_j \rangle = \begin{cases} 1, & \text{for } i = j, \\ 0, & \text{for } i \neq j. \end{cases}$$