Innlevering 1 — Lineær algebra

Oppgave 1

Siden vi vet at polynomet går gjennom punktene , , og , kan vi skrive opp følgende ligningssystem:

Om vi løser dette ved hjelp av Gauss-eliminasjon, får vi løsningsmengden . Vi kan dobbeltsjekke dette ved hjelp av denne Python-koden:

import numpy as np

import matplotlib.pyplot as plt

def p(x):

return -3/2 \* x\*\*2 + 11/2 \* x - 2

graf\_x = np.linspace(0, 4, 1000)

graf\_y = p(graf\_x)

punkter\_x = np.array([1, 2, 3])

punkter\_y = p(punkter\_x)

plt.plot(graf\_x, graf\_y)

plt.plot(punkter\_x, punkter\_y, 'bo')

Som gir følgende utskrift:

A graph with a dotted line

AI-generated content may be incorrect.

Vi ser at alle punktene er på grafen, og vi vet derfor at polynomet  
 er riktig.

Oppgave 2

Matrisen er gitt ved:

Den enkleste måten å finne inversmatrisen på vil selvfølgelig være å bruke en datamaskin — men om vi skal gjøre det for hånd, kan man bruke radoperasjoner. Slik kan man notere det:

Inversmatrisen til , , er gitt ved:

Koden nedenfor har jeg kopiert fra Jupyter Notebook. Det eneste jeg har lagt til er linjen som skriver ut B.

# her er litt syntaks for matriseregning

# vi importerer pakkene vi trenger

import numpy as np

A = np.array([[2,3,4],[3,4,5],[4,5,7]]) # her lager vi matrisa A ved å oppgi radvektorer

B = np.linalg.inv(A) # her bruker vi den innebygde rutinen i python for å finne A^(-1)

print(B)

Dette gir utskriften:

[[-3. 1. 1.]

[ 1. 2. -2.]

[ 1. -2. 1.]]

Som vi ser, er matrisen lik den vi kom fram til.

I denne oppgaven skal vi også finne et andregradspolynom som går gjennom , , og . Som i oppgave 1, kan vi sette opp et lineært ligningssystem basert på dette:

Om vi løser dette ligningssystemet får vi løsningsmengden:

Altså går følgende polynom, som jeg velger å kalle , gjennom alle de tre punktene:

Av kontinuerlige polynomer med reelle tall er dette det eneste som går gjennom punktene , , og .

*Oppgave 3 ligger på siste side.*

Oppgave 3

I oppgave 1 og 3 fant vi disse polynomene:

* , som gikk gjennom punktene , , og .
* , som gikk gjennom punktene , , og .

Vi skal finne enda et kvadratisk polynom, , som går gjennom punktene , , og . Om vi legger sammen y-verdiene i de to punktene som og går gjennom ved samme x-verdi, ser vi at vi får y-verdiene som går gjennom ved på samme sted. Med andre ord kan vi legge polynomene og sammen for å finne :

Vi sjekker at dette polynomet stemmer med følgende Python-kode:

def r(x):

return -2\*x\*\*2 + 8\*x - 3

print(f"r(1)={r(1)}")

print(f"r(2)={r(2)}")

print(f"r(3)={r(3)}")

Dette gir utskriften:

r(1)=3

r(2)=5

r(3)=3

Som vi ser, stemmer det at polynomet er gitt ved .