# Aflevering 3 - Numerisk Lineær Algebra

Af Jesper Bertelsen, AU-ID: au689481

## Plot grafen

Denne opgave var ligetil, datapunkterne plottes:



## Approksimer en funktion der går i gennem de sidste 3 punkter.

Denne opgave vred jeg midt hoved lidt rundt for at finde ud af hvordan jeg løste.

Jeg fik lavet et setup, hvor jeg kunne skrive det op som en lineær løsning.

Hvor

Og

Jeg lavede mig nogle funktioner til at lave række operationer.

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Og lavede en matrix, med alt i:

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Matrixen er kaldt A, men det er hele udtrykket. Undskyld forvirringen.

Jeg lavede operationer:

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Den inverse værdi er fundet, og af det kan jeg finde x.

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Funktionen var da fundet.



## Hvad er den laveste polynomiumgrad, som går gennem alle punkter

Til dette problem skal der forstås, hvad det er vi gør.

For hver grad haves der n + 1 koefficienter vi prøver at løse for og fra vores x og y værdier har vi 5 ligninger.

Med den lineære løsning, ax = b, som vi løser med, vil n < 4 medføre overdetermined, hvor vi har færre variabler end ligninger, n == 4 medfører en løsning, og n > 4 medfører underdetermined, hvor vi vil have frie variabler.

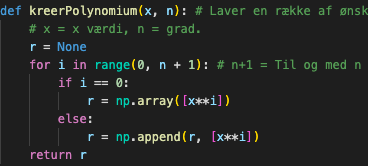
=======================================

En løsning vil da kunne findes ved 4 og herefter.

=======================================

Denne løsning laves i python:

For enkelthedens skyld laver jeg en funktion som skriver min grad ligning op for given x værdi:



For hver mine x værdier, vil jeg så indsætte disse rækker i min matrice.

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Linalg.solve drillede mig, så jeg valgte at løse den selv. Linalg.inv(matrix) virker fint.

A^-1 \* b = x

Koefficienterne blev skrevet ned, og nye x / y værdier blev lavet.



## Opstil et lineært løsningssystem for en funktion med to tredjegradsfunktioner.

Med betingelserne kan der laves ligninger for

, hvorfra vi har 6 ligninger.

Der vides at i det fælles punkt, 8, har & samme hældning.

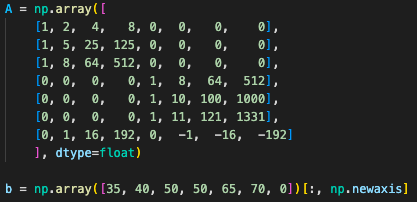
, hvilket giver en 7’ende ligning.

Når differentieret kommer vil gå ud.

Hvis vi så samler værdierne i venstre side får vi at:

Hvilket er vores sidste ligning.

Systemet laves nu i python.





## Vis at systemet har mere end en løsning.

Eftersom at mit ligningssystem ovenfor består af 8 ubekendte og 7 ligninger, vil der haves en fri variabel.

Denne variabel vil i mit ligningssystem være den koefficienten for den 3 grad i min p2.

, har en fri variabel.

Med en fri variabel, så vil denne have uendelig mange løsninger, da alle ligninger er afhængige af denne variabel.

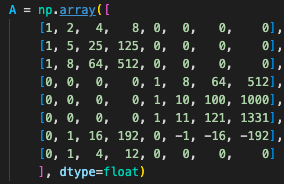
## Ved at sætte én betingelse på den afledede af *𝑝*1(*𝑥*) i 2,0, dan et system med en entydig løsning og plot den resulterende funktion *𝑓*. Gør rede for jeres valg af betingelse på *𝑝*1 (*𝑥*).

Jeg kunne godt ønske, at hældningen til x = 2 er 1,5 som virker som en reel værdi, som hældningen i polynomierne kunne have.

Jeg kunne godt ønske, at hældningen til i x = 2 er 0, så grafen i dette punkt er vandret.

Med denne betingelse vil jeg have endnu en ligning at løse, så nu 8 ligninger til 8 ubekendte, så nu burde systemet have en endelig løsning.

a\*x:



=



Med linalg.solve får jeg koefficienterne:



Med reference til dem for mine funktionsværdier, og mine x værdier lavet som følgende:

Et billede, der indeholder tekst

Automatisk genereret beskrivelse

Er jeg klar til at plotte grafen.

