# Numerisk Lineær Algebra eksamen 2023

Eks.nr: 207476, studienr: 20224617, AU-ID: au689481

Indholdsfortegnelse

[Numerisk Lineær Algebra eksamen 2023 1](#_Toc138849209)

[Opgave 1 ( 10 % ). Betragt matricen 2](#_Toc138849210)

[Opgave 2 ( 10 % ). Givet basen 2](#_Toc138849211)

[Opgave 3 ( 10 % ). Lad 𝑉 være vektorrummet, som består af alle differentiable funktioner 𝑓: [−1, 1] → R. Hvilke af de følgende mængder er ikke underrum af 𝑉? 3](#_Toc138849212)

[Opgave 4 ( 10 % ). For en symmetrisk matrix a i R100×100 udregnes en tridiagonalfaktorisering a = q @ t @ q.T. Udregningen foregår via den følgende funktion 3](#_Toc138849213)

[Opgave 5 (20 point). Lad de følgende vektorer i R6 være givet 5](#_Toc138849214)

[Opgave 6 (20 point). Betragt de følgende datapunkter 9](#_Toc138849215)

[Opgave 7 (20 point) Lad matricen 𝐴 være givet ved ( Mangler ) 12](#_Toc138849216)

## Opgave 1 ( 10 % ). Betragt matricen

Bestem en basis for søjlerummet af . Vælg en af de følgende svarmuligheder.

Matricen ønskes reduceret til echelon form.

De eneste søjler med pivotelementer fås til at være søjle 1 & 3. Med viden om, at består af de rækker med pivotelementer, så findes til at være en basis af A’s søjle 1 & 3.

Hvis disse søjler så findes fra A, så er

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

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

## Opgave 2 ( 10 % ). Givet basen

for R2, hvilken af de følgende svar muligheder er koordinatvektoren [*𝑢*]*𝐸* for *𝑢* = (0, 1) mht. *𝐸*?

Det lineære ligningssystem

Rækkeoperationer laves.

Så koordinatvektoren

Det er en svar mulighed, så der var vi heldige:

Svaret er:

===========

[D]

===========

## Opgave 3 ( 10 % ). Lad 𝑉 være vektorrummet, som består af alle differentiable funktioner 𝑓: [−1, 1] → R. Hvilke af de følgende mængder er ikke underrum af 𝑉?

## Opgave 4 ( 10 % ). For en symmetrisk matrix a i R100×100 udregnes en tridiagonalfaktorisering a = q @ t @ q.T. Udregningen foregår via den følgende funktion

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Her beregner funktionen tridiagonal\_data(a) en matrix data af samme størrelse som a, og en array s med s.shape = (100,).

1. Angiv v.shape for v i linje 8 når j = 63.

tager en liste af værdier fra række 65 til række 100, for værdier i søjle 63.

Det ligner at x bliver en vektor, med rækker og ikke søjler, derfor undre det mig lidt, at der står i stedet for . Nok om det.

Så v tager to værdier, 1 & x og lægger dem oven på hinanden.

Hvis

Så vil v.shape være

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

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

1. Når v.shape = (72,) bestem hvor mange flops bliver brugt i beregning af den nye q[j+1:, j+1:]i linje9.

Så , da er



= 



Med tabel 5.1 finder vi nu antallet af flops:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, nummer/tal

Automatisk genereret beskrivelse

indeholder n = 72, så 72 flops

Det første matrix vektorprodukt er så, , hvor , så flops.

Efterfulgt af det næste matrix vektorprodukt af samme størrelse. Det samlede antal flops er da.

=========

20808 flops

=========

## Opgave 5 (20 point). Lad de følgende vektorer i R6 være givet

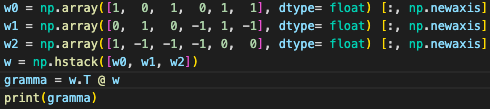
Et billede, der indeholder diagram, linje/række

Automatisk genereret beskrivelse

1. I python dan Grammatricen for , , og bekræft at vektorer er ortogonal.

Grammatricen er samlingen prikket med dens transponerede.

Min kode i python:



Som resulterer i grammatricen:



Samlingen er ortogonal, hvis er gældende.

Det eftertjekkes:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, nummer/tal

Automatisk genereret beskrivelse

Som resulterer i:

6 x 0 = 

Så denne samling er ortogonal.

1. Beregn projektionen af

Et billede, der indeholder tekst, linje/række, nummer/tal

Automatisk genereret beskrivelse

Til Span{, , }

Hele samlingen skal projiceres på u, så .

Projektionen foregår ved:

Som også kan skrives som

I python findes projektionerne.

Min kode:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

resulterer i:

Et billede, der indeholder tekst, skærmbillede, sort, Font/skrifttype

Automatisk genereret beskrivelse

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

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

1. Forklar hvorfor *𝑃*(*𝑢*) er parallel med *𝑤*2.

Det kan være at der er fejl i mine beregninger.

For at gælder der, at skal kunne skrives ved for .

Det er også næsten gældende.

Hvis s sættes til -1, så fås

Det er tæt på, men min projektion er ikke parallel med .

1. Sæt *𝑤*3 = *𝑢* − *𝑃*(*𝑢*) og bestem en ortonormal samling *𝑣*0, *𝑣*1, *𝑣*2, *𝑣*3 med hvert *𝑣𝑖* parallel med den tilsvarende *𝑤𝑖*.

Taget in mente, at min projektion ikke er helt korrekt, så ved jeg, at næste resultat ikke vil være præcis. Jeg håber at min fremgangsmåde stadigvæk giver point.

Til at lave en ortonormal samling, tænkte jeg at gøre, at tage brug af den klassiske gram Schmidt process. Der gælder dog, at vektoren er vinkelret på w og ikke parallelt. Det virker lidt mærkeligt, at skulle finde de parallelle, det synes jeg ikke, at vi har arbejdet så meget med.   
Jeg ignorerer det faktum og laver den klassiske gram schmidt process.

Vi ved allerede, at alle er ortogonale på hinanden, fra opgave a, så den første del af gram schmidt processen er klaret der.

For at lave alle v vektorer ortonormale sætter vi så

Min python kode:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Resultatet fra python:

Et billede, der indeholder skærmbillede, tekst, sort, Font/skrifttype

Automatisk genereret beskrivelse



## Opgave 6 (20 point). Betragt de følgende datapunkter

Et billede, der indeholder tekst, Font/skrifttype, linje/række, nummer/tal

Automatisk genereret beskrivelse

Der ønskes et polynomium *𝑝𝑠*3 + *𝑞𝑠*2 + *𝑟𝑠* + *𝑐*, der går igennem punkterne.

1. Ved hjælp af funktionen np.vander(...)opstiletlineærtligningssystem af formen *𝐴𝑣* = *𝑏*, som svarer til problemstillingen. Angiv størrelsen af *𝐴*, *𝑣* og *𝑏* og forklar hvorfor der kan ikke forventes en eksakt løsning.

Et billede, der indeholder tekst, skærmbillede, diagram, design

Automatisk genereret beskrivelse



1. Ved brug af en tynd *𝑄𝑅*-dekomponering konstrueret via den forbedrede Gram-Schmidt proces, bestem i python den mindste kvadraters løsning til systemet *𝐴𝑣* = *𝑏*.

Ved Gram-Schmidt processen, kan man vælge at bruge den forbedrede eller den klassiske. Det er noget med at den klassiske proces går søjlevist, mens den forbedrede går rækkevist og det er med til at gøre, at den forbedrede er præcis indenfor machine epsilon. Jeg har snydt lidt hjemmefra og lavet en funktion til begge, men den forbedrede bruges:

Kode:

Et billede, der indeholder tekst, skærmbillede, software, display/skærm/fremvisning

Automatisk genereret beskrivelse

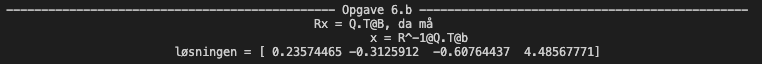
Fra notesættet side 240 kendes der løsningen til x ud fra en gram schmidt proces:

Et billede, der indeholder tekst, Font/skrifttype, kvittering, skærmbillede

Automatisk genereret beskrivelse

Min brug af koden:





1. Givenpythonplotafdatapunkterneogpolynomietbestemtafdinløsning.

Min kode:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Resultatet:

Et billede, der indeholder diagram, linje/række, Kurve

Automatisk genereret beskrivelse

1. Brug python til at bestemme singulærværdierne af *𝐴* og konditionstallet for *𝐴*. Giv desuden en øvre grænse for konditionstallet *𝜅* for beregning, som funktion af *𝐴*, af den mindste kvadraters løsning *𝑣* til *𝐴𝑣* = *𝑏*.

De singulærer værdier, lærte vi lidt til sidst, om at det kan findes på følgende måde:

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, nummer/tal

Automatisk genereret beskrivelse

Gennem kurset har vi dog mest brugt, np.linalg.svd til at finde u, s, v^T og det er det jeg er trykkest ved, så det er den metode jeg bruger.

Min python kode:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

Resultat:

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, sort

Automatisk genereret beskrivelse

Den øvre grænse for konditionstallet for A er da

## Opgave 7 (20 point) Lad matricen *𝐴* være givet ved ( Mangler )

I det følgende må der ikke bruges funktioner som np.linalg.eig eller tilsvarende computer algebra funktioner.

” Til disse opgaver vides der, at egenværdierne fås ved det(A - ) = 0.

1. Vis at *𝑣*0 = (1, 1, 1) er en egenvektor for *𝐴* og bestem den tilhørende egenværdi *𝜆*0.

Ved hjælp af rækkeoperationer vil jeg prøve at gøre dette muligt.

Så fås 3 ligninger

Hvis Lambda sættes til at være 1, så er y nødt til at være 1.

Så gælder der at

Hvis dette indsættes i topligningen:

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

Ved

Tæt på, men ikke lige med hinanden. Jeg går dog videre.

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

1. Vis at *𝜆*1 = 2 + *𝑖* er en egenværdi for *𝐴* og bestem en tilhørende egenvektor *𝑣*1.

Jeg har lært at tage determinaten ved at lægge produkterne af diagonalenerne sammen, og trække produkterne væk fra diagonalen fra.

Så

Lad os teste egenværdien 2 + *𝑖*

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

Resultatet gav 0, dermed eren egenværdi for A.

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

1. Fra dele (a) og (b) bestem en diagonal matrix Λ og en invertibel matrix *𝑉* således at

*𝐴* = *𝑉*Λ*𝑉*−1.

1. Bestem løsningen *𝑦*(*𝑡*) = (*𝑦*0(*𝑡*),*𝑦*1(*𝑡*),*𝑦*2(*𝑡*)) til systemet

*𝑦*′(*𝑡*) = *𝐴𝑦*(*𝑡*)

med startbetingelsen *𝑦*(0) = (1,0,0) og plot funktionen *𝑒*−2*𝑡𝑦*0(*𝑡*).