



LinAlgDat

2024/2025

Projekt A

Projektet består af fire opgaver. Opgave 1 og 2 er rene matematikopgaver (ligesom dem i den skriftlige prøve). Opgave 3 har fokus på anvendelser af lineær algebra. Opgave 4 drejer sig om at implementere metoder og algoritmer fra lineær algebra i F# eller Python.

Besvarelsen af projektet skal bestå af følgende to filer. Filerne må ikke zippes og skal afleveres i Absalon.

- Én pdf-fil, skrevet i L^AT_EX, med løsninger til opgaverne 1, 2 og 3. Første side i pdf-filen skal være en forside indeholdende forfatterens fulde navn, KU-id og holdnummer.
- Én F#- eller Python-fil med løsninger til Opgave 4 (se opgaveformuleringen for detaljer).

Ved bedømmelsen af projektet lægges naturligvis vægt på korrekthed, men det er også vigtigt, at fremstillingen er klar og overskuelig. Mellemløsningsregninger skal medtages og jeres kode skal kommenteres i passende omfang. Projektet skal laves individuelt. Afskrift betragtes som eksamenssnyd.

Programmeringsdelen rettes bl.a. ved at jeres løsning bliver afprøvet på hemmeligholdt testdata. Der vil blive udleveret tilsvarende testskripts, som I selv kan teste jeres kode på før I afleverer.

Der er adgang til hjælp ved projekthjælp-øvelserne og (evt. i lidt mindre udstrækning) ved studiecaféerne på DIKU. Tidsfrister for aflevering, retning, mm. af projektet er beskrevet i dokumentet *Kursusoversigt*. I er selv ansvarlige for at holde jer orienteret herom.

Besvarelser der er afleveret for sent vil som udgangspunkt ikke blive rettet. Der er ikke mulighed for genaflevering. Aflever derfor i god tid, også selvom der er dele af opgaverne I ikke har nået.

Opgave 1 (25%)

Betragt ligningssystemet, hvor $a \in \mathbb{R}$,

$$\begin{aligned}x_1 + 2x_2 + 8x_3 &= a \\ax_1 + ax_2 + 4ax_3 &= a \\2x_1 + 2x_2 + 2a^2x_3 &= 0.\end{aligned}$$

(a) Vis, at den reducerede rækkeechelonform af totalmatricen, for $a \in \mathbb{R} \setminus \{0, -2, 2\}$, er givet ved

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 2-a \\ 0 & 1 & 0 & (a^3-a^2-4a+8)/(a^2-4) \\ 0 & 0 & 1 & 1/(4-a^2) \end{array} \right]. \quad (1)$$

- (b) Løs ligningssystemet når $a = 0$. Kan den rækkereducerede totalmatrix i (1) bruges til at finde de løsninger der eventuelt måtte være når $a = 0$?
- (c) Løs ligningssystemet når $a = 2$. Kan den rækkereducerede totalmatrix i (1) bruges til at finde de løsninger der eventuelt måtte være når $a = 2$?
- (d) Antag, at $a = 1$. Bestem den inverse matrix til koefficientmatricen \mathbf{A} for ligningssystemet ved at bruge COMPUTATION p. 78 i lærebogen, og bestem herved samtlige løsninger til

$$\mathbf{A} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

Opgave 2 (25%)

Det oplyses, at en ukendt 3×3 matrix \mathbf{A} , hvori der indgår en parameter $a \neq 0$ ved rækkeoperationerne ero_1 , ero_2 , ero_3 og ero_4 i denne rækkefølge kan omformes til enhedsmatricen.

$\text{ero}_1: -2a\mathbf{r}_1 + \mathbf{r}_2 \rightarrow \mathbf{r}_2$ ($2a$ gange første række fratrækkes anden række),
 $\text{ero}_2: 5\mathbf{r}_3 \rightarrow \mathbf{r}_3$ (tredje række ganges med 5),
 $\text{ero}_3: \mathbf{r}_1 \leftrightarrow \mathbf{r}_3$ (første og tredje række ombyttes),
 $\text{ero}_4: (1/a^2)\mathbf{r}_3 + \mathbf{r}_1 \rightarrow \mathbf{r}_1$ ($1/a^2$ gange tredje række lægges til første række).

Vi lader \mathbf{E}_i betegne den elementære matrix, som svarer til rækkeoperationen ero_i (for $i = 1, 2, 3, 4$).

- (a) Bestem \mathbf{E}_i for hvert $i = 1, 2, 3, 4$.

Argumentér for, at et af matrixprodukterne

$$\mathbf{A}\mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1, \mathbf{A}\mathbf{E}_1\mathbf{E}_2\mathbf{E}_3\mathbf{E}_4, \mathbf{E}_1\mathbf{E}_2\mathbf{E}_3\mathbf{E}_4\mathbf{A} \text{ eller } \mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1\mathbf{A}$$

er lig enhedsmatricen. Hvilket er der tale om? Angiv endnu et blandt disse fire produkter der er lig enhedsmatricen.

- (b) Bestem for hvert $i = 1, 2, 3, 4$ den elementære matrix \mathbf{E}_i^{-1} og brug disse til at bestemme \mathbf{A} .
(c) Lad \mathbf{X} være den 2×3 matrix hvis første hhv. anden række er række 1 hhv. række 3 i matricen $\mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1$ og sæt

$$\mathbf{V} = \begin{bmatrix} 0 & 1 \\ 0 & 2a \\ 1/5 & -1/(5a^2) \end{bmatrix}.$$

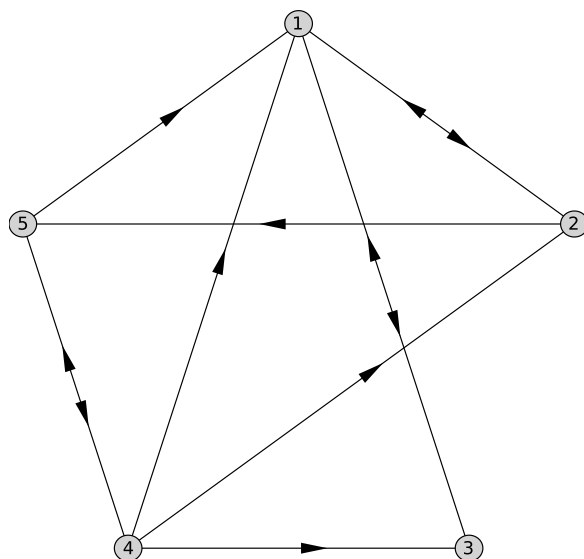
Vis, at \mathbf{X} er en venstreinvert til \mathbf{V} .

Bestem endvidere alle venstreinvertser til \mathbf{V} .

Har matricen \mathbf{V} også en højreinvert?

Opgave 3 (25%)

Nedenstående figur viser en orienteret graf med 5 knuder.



- (a) Bestem nabomatricen (eng. *adjacency matrix*) \mathbf{N} for denne orienterede graf.

Angiv antallet af veje fra knude 4 til knude 1 af længde netop 8.

Vink: det oplyses, at

$$\mathbf{N}^7 = \begin{bmatrix} 30 & 17 & 17 & 8 & 12 \\ 39 & 28 & 28 & 10 & 17 \\ 15 & 10 & 10 & 2 & 8 \\ 69 & 45 & 45 & 18 & 29 \\ 53 & 32 & 32 & 9 & 26 \end{bmatrix}.$$

Vi tænker os nu, at grafen repræsenterer et web med 5 sider. De næste spørgsmål vedrører begreber defineret i dokumentet “*Google’s page ranking*”.

- (b) Opskriv linkmatricen \mathbf{A} for grafen ovenfor.

- (c) Bestem en vektor $\mathbf{x} \neq \mathbf{0}$ som opfylder ligningen $\mathbf{Ax} = \mathbf{x}$ og foretag på grundlag af dette en rangordning af siderne i webbet.

Vink: Det kan være en god idé at starte med at multiplicere rækkerne i totalmatricen med passende tal så man opnår heltal. Men det bliver svært helt at undgå brøker!

Opgave 4 [Programmering i F# eller Python] (25%)

I dette projekt skal du implementere en lille delmængde af de operationer, der er relateret til matrix og vektor algebra. Hovedformålet med denne programmeringsopgave er at få en solid grundlæggende forståelse af matrix- og vektordatastrukturer. For at komme i gang med opgaven skal man først downloade projektfilerne. F#-filerne er tilgængelige fra linket <https://absalon.ku.dk/files/9531683/>, Python-filerne er tilgængelige fra linket <https://absalon.ku.dk/files/9531684/>. For det andet skal man få et overblik over projektfilerne, prøve at åbne filerne og kigge på dem. Her er et kort resumé:

Programmering i F#

Core/ Denne mappe indeholder Matrix- og Vector-klasserne, som vil blive brugt i hele kursus. Gør dig bekendt med disse klasser.

Core/*Factory.fs Disse filer indeholder nogle fabriksmetoder. En fabriksmetode kan generere en bestemt type matrix- eller vektorobjekt. Du behøver ikke at bruge nogen af de metoder, der er defineret i disse filer, men de kan tjene som yderligere eksempler for dem, der er interesserede.

ProjectA/BasicExtensions.fs Denne fil indeholder flere ufærdige metoder. Dette er den eneste fil du har lov til at ændre og den eneste fil, du kan indsende til programmeringsdelen af projekt A.

ProjectA/RunTest.fsx Denne fil indeholder data til selvtestning.

ProjectA/TestProjectA.fs Denne fil bruges til at teste din implementering i BasicExtensions.fs i forhold til testdataene i RunTest.fsx.

ProjectA/README.md Denne fil indeholder en forklaring af kommandoerne til kompilering og kørsel af RunTest.fsx til Windows, macOS og Linux.

Din opgave er at færdiggøre de ikke-implementerede metoder i ProjectA/BasicExtensions.fs. Bemærk, at metoden `AugmentRight` allerede er implementeret som et eksempel. Man kan bruge denne som en prototype eller inspiration til at implementere de resterende metoder. Husk, at du kun er tilladt at ændre BasicExtensions.fs. Du er velkommen til at tilføje yderligere hjælpemetoder i BasicExtensions.fs, men du må ikke omdøbe eller på anden måde ændre typesignaturen for de eksisterende metoder. Når du afleverer din løsning til programmeringsdelen af projekt A, skal du kun uploade filen ProjectA/BasicExtensions.fs.

For dette første projekt har vi givet et forkortet eksempel på output efter en vellykket opbygning og kørsel af programmet i appendiks.

Opbyg og kørsel projektet. Der findes nogle build-scripts til at bygge og/eller køre koden. Du skal have en fungerende installation af dotnet. Du bør have det fra PoP. README.md-filen forklarer, hvordan du kompilerer og kører dine filer til dotnet: Windows og Linux/MacOs kompilerings-/eksekverings-scripts leveres sammen med koden.

Gå ikke i panik, hvis ingen af de ovenfor nævnte build/kompileringsmetoder lyder bekendt for dig. Vær venlig at henvende dig til kontakt TA'erne, og de vil hjælpe dig.

Når du kopierer projektfilerne til din computer, bedes du kun bruge alfanumeriske ASCII-tegn (og måske understregningen '_' og prikken '.')¹, ikke nationale sprogspecifikke (såsom dansk æ, ø, å) i din mappes navne, kan/vil det forhindre dotnet for at køre korrekt, hvis overhovedet!

Programmering i Python

Core/ Denne mappe indeholder Matrix- og Vector-klasserne, som vil blive brugt i hele kursus. Gør dig bekendt med disse klasser.

Matrix- og Vector-klasserne indeholder statiske metoder, der kan generere bestemte typer af matricer og vektorer (I F# er de kodet i separate filer.)

ProjectA/BasicExtensions.py Denne fil indeholder flere ufærdige metoder. Dette er den eneste fil du har lov til at ændre og den eneste fil, du kan indsende til programmeringsdelen af projekt A.

ProjectA/data_projectA.py Denne fil indeholder data til selvtestning.

ProjectA/TestProjectA.py Denne fil bruges til at teste din implementering i BasicExtensions.py i forhold til testdataene i data_projectA.py.

ProjectA/README.md Denne fil indeholder en kort beskrivelse af installation kommandoer til Windows, macOS og Linux.

Din opgave er at færdiggøre de ikke-implementerede metoder i ProjectA/BasicExtensions.py. Bemærk, at metoden AugmentRight allerede er implementeret som et eksempel. Man kan bruge denne som en prototype eller inspiration til at implementere de resterende metoder. Husk, at du kun er tilladt at ændre BasicExtensions.py. Du er velkommen til at tilføje yderligere hjælpemetoder i BasicExtensions.py, men du må ikke omdøbe eller på anden måde ændre typesignaturen for de eksisterende metoder. Når du afleverer din løsning til programmeringsdelen af projekt A, skal du må du kun uploade filen ProjectA/BasicExtensions.py.

For dette første projekt har vi givet et forkortet eksempel på output efter en vellykket kørsel af programmet i appendiks.

kør projektet. Du skal have en kørende installation af Python3 (Python 3.6 eller højere).

- Gå til mappen ProjectA.

```
C:\StiTilMappen\>python TestProjectA.py (Windows)
brugernavn:StiTilMappen$ python TestProjectA.py (Linux, MacOS)
```

Gå ikke i panik, hvis ingen af de ovenfor nævnte metoder lyder bekendt for dig. Vær venlig at henvende dig til kontakt TA'erne, og de vil hjælpe dig.

¹ASCII: American Standard Code for Information Interchange, 26 x 2 standard alfabetiske tegn (store og små bogstaver), tal samt tegnsætning, klammer og andre almindelige tegn. Ingen accent eller sprogspecifikke tegn. De alfanumeriske tegn er 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'.

Appendiks

Dette er et forkortet eksempel på resultatet efter at have kørt projektet med metoden `AugmentRight` implementeret som angivet, enten med F#-rammen eller Python-rammen. Dette er det forventede output, hvis man (bygger F# program, hvis F# bruges i projektet, og) kører det udleverede projekt rigtigt med det samme uden at ændre noget som helst.

Test results

=====

Tests for the `AugmentRight(Matrix, Vector)` function

=====

<code>AugmentRight(Matrix, Vector) Values</code>	<code>[PASSED]</code>
<code>AugmentRight(Matrix, Vector) Dims</code>	<code>[PASSED]</code>
<code>AugmentRight(Matrix, Vector) All</code>	<code>[PASSED]</code>

End of test for the `AugmentRight(Matrix, Vector)` function.

Tests for the `AugmentRight(Matrix, Vector)` function

=====

<code>AugmentRight(Matrix, Vector) Values</code>	<code>[PASSED]</code>
<code>AugmentRight(Matrix, Vector) Dims</code>	<code>[PASSED]</code>
<code>AugmentRight(Matrix, Vector) All</code>	<code>[PASSED]</code>

End of test for the `AugmentRight(Matrix, Vector)` function.

Tests for the `AugmentRight(Matrix, Vector)` function

=====

<code>AugmentRight(Matrix, Vector) Values</code>	<code>[PASSED]</code>
<code>AugmentRight(Matrix, Vector) Dims</code>	<code>[PASSED]</code>
<code>AugmentRight(Matrix, Vector) All</code>	<code>[PASSED]</code>

End of test for the `AugmentRight(Matrix, Vector)` function.

Tests for the `MatVecProduct(Matrix, Vector)` function

=====

<code>MatVecProduct(Matrix, Vector) Values</code>	<code>[FAILED]</code>
<code>MatVecProduct(Matrix, Vector) Dims</code>	<code>[PASSED]</code>

End of test for the `MatVecProduct(Matrix, Vector)` function.

Tests for the `MatVecProduct(Matrix, Vector)` function

=====

MatVecProduct(Matrix, Vector) Dims [FAILED]

End of test for the MatVecProduct(Matrix, Vector) function.

...

...

...

Summary

=====

Tests of AugmentRight(Matrix, Vector) passed/total: [3/3]

Tests of MatVecProduct(Matrix, Vector) passed/total: [0/3]

Tests of MatrixProduct(Matrix, Matrix) passed/total: [0/3]

Tests of Transpose(Matrix) passed/total: [0/3]

Tests of VectorNorm(Vector) passed/total: [0/3]

Henrik Holm (holm@math.ku.dk)
Henrik Laurberg Pedersen (henrikp@math.ku.dk)
Jon Sparring (sparring@di.ku.dk)