



LinAlgDat

2020/2021

Projekt A

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

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

- Én pdf-fil, skrevet i \LaTeX , 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.
Opgaver som ikke er skrevet i \LaTeX (fx opgaver skrevet i Word, scannede filer mm.) vil ikke blive accepteret.
- Én F#-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. Mellemregninger skal medtages og jeres F#-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 testscripts som I selv kan teste jeres kode på før I afleverer.

Der er adgang til hjælp ved Projekthjælp-øvelserne og evt. også 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. Aflevér derfor i god tid, også selvom der er dele af opgaverne I ikke har nået.

Opgave 1 (25%)

Betragt ligningssystemet (med et ukendt reelt tal a):

$$2x_1 + (3 - a)x_2 + 2x_3 = 2 + a$$

$$2x_1 + ax_2 + 4ax_3 = 1$$

$$2x_1 + ax_2 + 2ax_3 = 1.$$

- (a) Antag, at $a \in \mathbb{R} \setminus \{0, 3/2\}$. Løs ligningssystemet ovenfor ved at bringe totalmatricen på reduceret rækkeechelonform. (Echelonformen vil afhænge af a .)
- (b) Løs ligningssystemet på samme måde for $a = 0$. Løs også ligningssystemet for $a = 3/2$.
- (c) Findes der en værdi af a så ligningssystemet ovenfor har mindst to forskellige løsninger?
- (d) Antag, at $a = 2$. Bestem den inverse matrix til koefficientmatricen \mathbf{A} for ligningssystemet ved at bruge COMPUTATION p. 78 i lærebogen, og udregn

$$\mathbf{A}^{-1} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}.$$

Opgave 2 (25%)

Det oplyses, at en ukendt 3×3 matrix \mathbf{A} ved rækkeoperationerne ero_1 , ero_2 , ero_3 og ero_4 i denne rækkefølge kan omformes til enhedsmatricen.

$\text{ero}_1: 3\mathbf{r}_1 + \mathbf{r}_2 \rightarrow \mathbf{r}_2$ (3 gange første række lægges til anden række),

$\text{ero}_2: 4\mathbf{r}_3 \rightarrow \mathbf{r}_3$ (tredje række ganges med 4),

$\text{ero}_3: \mathbf{r}_1 \leftrightarrow \mathbf{r}_2$ (første og anden række ombyttes),

$\text{ero}_4: -2\mathbf{r}_3 + \mathbf{r}_1 \rightarrow \mathbf{r}_1$ (-2 gange tredje række lægges til første række).

- (a) Bestem for hvert $i = 1, 2, 3, 4$ den elementære matrix \mathbf{E}_i , som svarer til rækkeoperationen ero_i .

Argumentér for, at et af matrixprodukterne

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

må være enhedsmatricen. Hvilket er der tale om? Angiv endnu et blandt disse fire produkter der giver 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} bestå af række nummer et og to i matricen $\mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1$ og sæt

$$\mathbf{B} = \begin{bmatrix} 0 & 1 \\ 1 & -3 \\ 0 & 0 \end{bmatrix}.$$

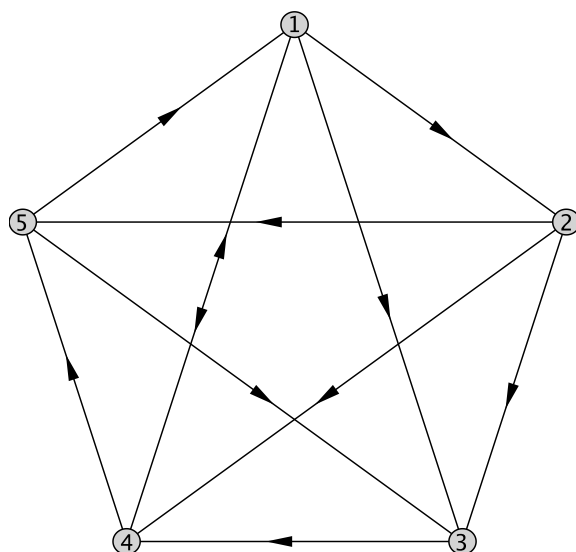
Vis, at \mathbf{X} er en venstre-invers til \mathbf{B} .

Bestem endvidere alle venstre-inverser til \mathbf{B} .

Har matricen \mathbf{B} også en højre-invers?

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 5 af længde netop 8.

Vink: det oplyses, at

$$\mathbf{N}^4 = \begin{pmatrix} 4 & 4 & 7 & 8 & 3 \\ 4 & 2 & 5 & 7 & 5 \\ 1 & 1 & 2 & 4 & 2 \\ 6 & 1 & 4 & 4 & 5 \\ 5 & 2 & 5 & 3 & 2 \end{pmatrix}.$$

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

- (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 [Programming i F#] (25%)

In this project you have to implement a small subset of the operations related to matrix and vector algebra. The main purpose of this programming assignment is to get a solid basic understanding of matrix and vector data structures. To get started on the assignment one has to first download the project files from <https://absalon.ku.dk/courses/47283/files/folder/Projekt%20A>. Secondly, one should get an overview of the project files, try and open the files and have a look at them. Here is a brief summary:

Core/ This directory contains the `Matrix` and `Vector` classes that will be used throughout the course. Familiarize yourself with these classes.

Core/*Factory.fs These files contain some factory methods. A factory method can generate a specific type of matrix or vector object. You do not have to use any of the methods defined in these files, but they can serve as additional examples for those who are interested.

ProjectA/BasicExtensions.fs This file contains several unfinished methods. *This is the only file you are allowed to modify and the only file you may submit for the programming part of Project A.*

ProjectA/RunTest.fsx This file contains data for self-testing.

ProjectA/TestProjectA.fs This file is used to test your implementation in `BasicExtensions.fs` against the test data in `RunTest.fsx`.

ProjectA/README.md This file contains a brief description of installation and compilation commands for Windows, macOS and Linux.

Your assignment is to finish the unimplemented methods in `ProjectA/BasicExtensions.fs`. Notice that the method `AugmentRight` is already implemented as an example. One may use this as a prototype or inspiration for implementing the remaining methods. Remember that you are only allowed to modify `BasicExtensions.fs`. You are welcome to add additional helper methods in `BasicExtensions.fs`, but you are not allowed to rename or otherwise alter the type signature of the existing methods. When submitting your solution to the programming part of Project A you are only allowed to upload the file `ProjectA/BasicExtensions.fs`.

For this first project we have provided an abridged example of the output after successfully building and running the program in the appendix.

Build and run the project. Some build scripts are provided for building and/or running the code. You should have all a running installation of `mono`, used in `Programering og Problemløsning (PoP)`. Compilation scripts are provided for Windows and bash (MacOS/Linux).

- Go to ProjectA folder.
- Windows:
C:\path_to\ProjectA> compileAndRun
- Linux/MacOs
/path_to/ProjectA\$./compileAndRun.sh

These commands produce a executable program called `RunTest.exe`. It can be ran from the command line.

```
C:\PathToFolder\>RunTest (Windows)
username:PathToFolder$ ./RunTest.exe (Linux, MacOS)
```

It is also possible just to compile and not run, using the `compile.*` script

```
C:\PathToFolder\>compile (Windows)
username:PathToFolder$ ./compile.sh (Linux, MacOS)
```

Do not panic if none of the build/compile methods mentioned above sound familiar to you. Please contact the TAs and they will help you.

Appendix

This is an abridged example of the output after running the project with method `AugmentRight` implemented as given. This is the expected output if you build and run the handed out project right away without modifying anything.

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>
---	-----------------------

***** Input Matrix *****

`[[-0.5254812892]]`

***** Input Vector *****

`[|-1.477194798|]`

***** Actual result *****

[|0.0|]

***** Expected result *****

[|0.7762382266|]

MatVecProduct(Matrix, Vector) Dims [PASSED]

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)
François Bernard Lauze (francois@di.ku.dk)