



LinAlgDat

2018/2019

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 C#.

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

- En 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.
- Netop en C#-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 C#-kode skal kommenteres i passende omfang. Projektet laves og bedømmes individuelt.

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.

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 (med et ukendt reelt tal a)

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

$$x_1 + ax_2 + 2x_3 = a$$

$$ax_1 + 2x_2 + 2ax_3 = 0$$

- (a) Antag, at $a \neq \pm\sqrt{2}$. Løs ligningssystemet ovenfor ved at bringe totalmatricen på reduceret rækkeechelonform. (Echelonformen vil afhænge af a .)
- (b) Antag, at $a = \pm\sqrt{2}$. Løs ligningssystemet ovenfor på samme måde.
- (c) Findes der en værdi af a så ligningssystemet ovenfor har 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

$$\mathbf{A}^{-1} \begin{bmatrix} 0 \\ -2 \\ 0 \end{bmatrix}.$$

Hvordan passer dette resultat med (a)?

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.

$\text{ero}_1: -4\mathbf{r}_1 + \mathbf{r}_2 \rightarrow \mathbf{r}_2$ (4 gange første række lægges til anden række),

$\text{ero}_2: -1/5\mathbf{r}_3 \rightarrow \mathbf{r}_3$ (tredje række ganges med $-1/5$),

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

$\text{ero}_4: 10\mathbf{r}_3 + \mathbf{r}_1 \rightarrow \mathbf{r}_1$ (10 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 $\mathbf{A}\mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1$, $\mathbf{A}\mathbf{E}_1\mathbf{E}_2\mathbf{E}_3\mathbf{E}_4$, $\mathbf{E}_1\mathbf{E}_2\mathbf{E}_3\mathbf{E}_4\mathbf{A}$ eller $\mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1\mathbf{A}$ må være enhedsmatricen. Hvilket er der tale om?

- (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 tre i matricen $\mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1$ og sæt

$$\mathbf{B} = \begin{bmatrix} 0 & 1 \\ 0 & 4 \\ -5 & 50 \end{bmatrix}.$$

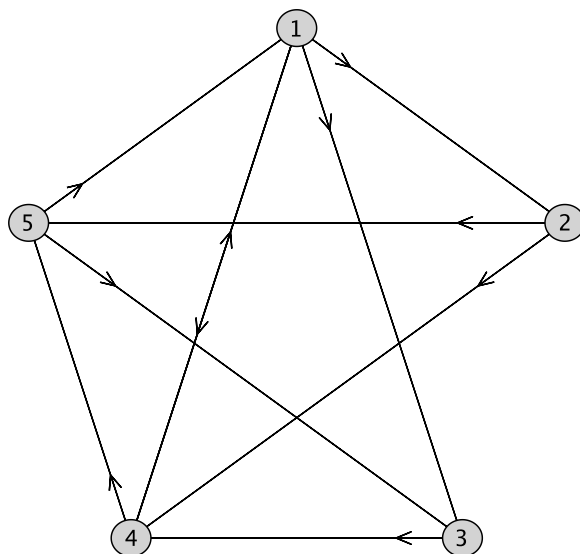
Vis, at \mathbf{X} er en venstre-invers til \mathbf{B} .

Bestem endvidere alle venstre-inverse 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 2 til knude 5 af længde netop 8. *Vink*: det oplyses, at

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

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 C#] (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.instructure.com/courses/31798/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:

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

ProjectA/Core/*Factory.cs 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.cs 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/MainClass.cs This file contains data for self-testing.

ProjectA/Program.cs This file is used to test your implementation in BasicExtensions.cs against the test data in MainClass.cs.

Your assignment is to finish the unimplemented methods in ProjectA/BasicExtensions.cs. 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.cs. You are welcome to add additional helper methods in BasicExtensions.cs, 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.cs.

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 using JetBrains Rider using ProjectA/ProjectA.csproj. Alternatively, we provide a Makefile and if you have Mono installed you can run the following commands:

- `make build` – This command builds the project using msbuild. The output executable is named ProjectA.exe and is located in ProjectA/bin/Debug/.
- `make run` – This simply runs the executable using mono.
- `make clean` – This one does what the command name says.

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.

Tests for the `Matrix.AugmentRight(Vector)` method.

```
=====
Matrix.AugmentRight(Vector) Run:                [PASSED]
Matrix.AugmentRight(Vector) Dims:                [PASSED]
Matrix.AugmentRight(Vector) Values:              [PASSED]
Matrix.AugmentRight(Vector) All:                 [PASSED]
End of test for the Matrix.AugmentRight(Vector) method.
-----
```

Tests for the `Matrix.AugmentRight(Vector)` method.

```
=====
Matrix.AugmentRight(Vector) Run:                [PASSED]
Matrix.AugmentRight(Vector) Dims:                [PASSED]
Matrix.AugmentRight(Vector) Values:              [PASSED]
Matrix.AugmentRight(Vector) All:                 [PASSED]
End of test for the Matrix.AugmentRight(Vector) method.
-----
```

Tests for the `Matrix.AugmentRight(Vector)` method.

```
=====
Matrix.AugmentRight(Vector) Run:                [PASSED]
Matrix.AugmentRight(Vector) Dims:                [PASSED]
Matrix.AugmentRight(Vector) Values:              [PASSED]
Matrix.AugmentRight(Vector) All:                 [PASSED]
End of test for the Matrix.AugmentRight(Vector) method.
-----
```

Non Square Matrix, Tests for the `Matrix.Product(Vector)` method.

```
=====
Matrix.Product(Vector) Run:                      [FAILED]
Non Square Matrix, end of test for the Matrix.Product(Vector) method.
-----
```

...

Tests for the Vector.VectorNorm() method.

=====

Vector.VectorNorm() Run:

[FAILED]

End of test for the Vector.VectorNorm() method.

=====

Success for Matrix.AugmentRight(Vector) method:	3/3 runs
---	----------

=====

=====

Success for Matrix.Product(Vector) method:	0/6 runs
(Non Square Matrix) Matrix.Product(Vector) method:	0/3 runs
(Square Matrix) Matrix.Product(Vector) method:	0/3 runs

=====

=====

Success for Matrix.Product(Matrix) method:	0/6 runs
(Non Square Matrix) Matrix.Product(Matrix) method:	0/3 runs
(Square Matrix) Matrix.Product(Matrix) method:	0/3 runs

=====

=====

Success for Matrix.Transpose() method:	0/6 runs
(Square Matrix) Matrix.Transpose() method:	0/3 runs
(Non Square Matrix) Matrix.Transpose() method:	0/3 runs

=====

=====

Success for Vector.VectorNorm() method:	0/3 runs
---	----------

=====

Henrik Holm (holm@math.ku.dk)
Henrik Laurberg Pedersen (henrikp@math.ku.dk)
François Bernard Lauze (francois@di.ku.dk)