LinAlgDat

2018/2019

Projekt C

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 testscripts 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. Aflevér derfor i god tid, også selvom der er dele af opgaverne I ikke har nået.

Opgave 1 (25%)

Betragt underrummet (planen) $\mathcal{U} = \text{span}\{\mathbf{u}_1, \mathbf{u}_2\}$ af $\mathbb{R}^3$, hvor

$$\mathbf{u}_1 = \begin{pmatrix} 3 \\ 2 \\ 6 \end{pmatrix} \quad \text{og} \quad \mathbf{u}_2 = \begin{pmatrix} 9 \\ -1 \\ 4 \end{pmatrix} .$$

(a) Bestem projektionsmatricen $\mathbf{P}$ for underrummet $\mathcal{U}$.

(b) Bestem spejlingen af vektoren $\mathbf{e}_1 = (1 \;\; 0 \;\; 0)^\mathsf{T}$ i underrummet (planen) $\mathcal{U}$.

(c) Bestem en basis $\{\mathbf{u}_3\}$ for underrummet $\mathcal{U}^\perp$ (det ortogonale komplement til $\mathcal{U}$).

(d) Bestem forskriften for den lineære transformation $T\colon \mathbb{R}^3 \to \mathbb{R}^3$ som opfylder:

$$T(\mathbf{u}_1) = \mathbf{u}_2 \quad , \quad T(\mathbf{u}_2) = \mathbf{u}_1 \quad \text{og} \quad T(\mathbf{u}_3) = \mathbf{u}_3 \, .$$

(*Vink:* Betragt basen $\mathcal{B} = \{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ samt standardbasen $\mathcal{E} = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ for $\mathbb{R}^3$. For en vilkårlig vektor $\mathbf{x} \in \mathbb{R}^3$, benyt de givne oplysninger til at udtrykke $[T(\mathbf{x})]_\mathcal{B}$ ved $[\mathbf{x}]_\mathcal{B}$. Kombinér dette med identiteten $\mathbf{y} = [\mathbf{y}]_\mathcal{E} = \mathbf{P}_{\mathcal{E}\leftarrow\mathcal{B}}[\mathbf{y}]_\mathcal{B}$ for hhv. $\mathbf{y} = T(\mathbf{x})$ og $\mathbf{y} = \mathbf{x}$.)

Opgave 2 (25%)

Betragt matricen

$$\mathbf{A} = \begin{pmatrix} 1 & -3 & 8 & -1 \\ -4 & -3 & -2 & -2 \\ 2 & 4 & 1 & 3 \\ 2 & 4 & 16 & 1 \end{pmatrix}.$$

(a) Bestem en QR-faktorisering af $\mathbf{A}$.

Betragt en vilkårlig 4×4 øvre trekantsmatrix:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ 0 & r_{22} & r_{23} & r_{24} \\ 0 & 0 & r_{33} & r_{34} \\ 0 & 0 & 0 & r_{44} \end{pmatrix},$$

hvor produktet $d = r_{11}r_{22}r_{33}r_{44}$ er forskelligt fra nul. Følgende formel for den inverse til matricen $\mathbf{R}$ må frit benyttes i det efterfølgende:

$$\mathbf{R}^{-1} = \frac{1}{d}\begin{pmatrix} r_{22}r_{33}r_{44} & -r_{12}r_{33}r_{44} & (r_{12}r_{23} - r_{13}r_{22})r_{44} & -r_{12}(r_{23}r_{34} - r_{24}r_{33}) + r_{22}(r_{13}r_{34} - r_{14}r_{33}) \\ 0 & r_{11}r_{33}r_{44} & -r_{11}r_{23}r_{44} & r_{11}(r_{23}r_{34} - r_{24}r_{33}) \\ 0 & 0 & r_{11}r_{22}r_{44} & -r_{11}r_{22}r_{34} \\ 0 & 0 & 0 & r_{11}r_{22}r_{33} \end{pmatrix}.$$

(b) Benyt del (a) og ovenstående formel for $\mathbf{R}^{-1}$ til at beregne $\mathbf{A}^{-1}$.
(Du kan evt. tjekke dit resultat ved at benytte COMPUTATION (2.23) s. 78 i lærebogen.)

Skriv $\mathbf{A} = (\mathbf{a}_1|\mathbf{a}_2|\mathbf{a}_3|\mathbf{a}_4)$ og $\mathbf{Q} = (\mathbf{q}_1|\mathbf{q}_2|\mathbf{q}_3|\mathbf{q}_4)$ hvor $\mathbf{Q}$ er den matrix som du fandt i QR-faktoriseringen i (a). Betragt de to ordnede baser $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4\}$ og $\mathcal{Q} = \{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4\}$ for $\mathbb{R}^4$.

(c) Vis at $\mathbf{P}_{\mathcal{Q} \leftarrow \mathcal{A}} = \mathbf{R}$.

(d) Bestem tal $\lambda_1, \lambda_2, \lambda_4, \lambda_4$ således at $\lambda_1\mathbf{a}_1 + \lambda_2\mathbf{a}_2 + \lambda_3\mathbf{a}_3 + \lambda_4\mathbf{a}_4 = \mathbf{q}_1 + \mathbf{q}_2 + \mathbf{q}_3 - \lambda_4\mathbf{q}_4$.

Opgave 3 (25%)

En computers "regnekraft" måles i FLoating-point Operations Per Second (FLOPS). Vi vil benytte lineær algebra til at beskrive hvordan (super)computernes regnekraft har udviklet sig med tiden.

IBM Summit (2018)

TABEL 1 (http://en.wikipedia.org/wiki/History_of_supercomputing) viser, for udvalgte år, hvor mange FLOPS verdens bedste supercomputer kunne præstere i det givne år. Vi betegner med t tiden (målt i år) og med $y = y(t)$ det antal FLOPS som verdens bedste supercomputer kunne præstere i år t. Værdierne i første og anden søjle i TABEL 2 er således blot overført fra TABEL 1, mens tredje søjle er beregnet ved at tage den naturlige logaritme (ln) til værdierne i anden søjle.

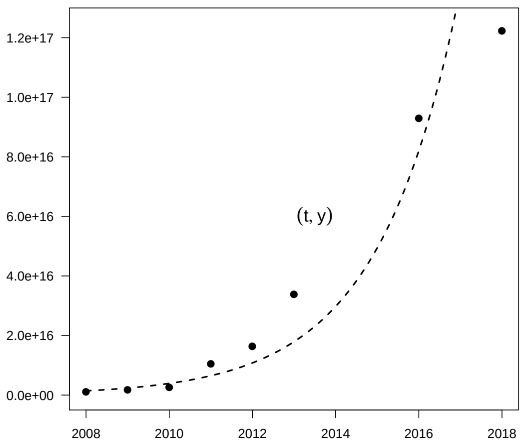
År	**Supercomputer**	**FLOPS**
2008	IBM Roadrunner	$1.026 \cdot 10^{15}$
2009	Cray Jaguar	$1.759 \cdot 10^{15}$
2010	Tianhe-IA	$2.566 \cdot 10^{15}$
2011	Fujitsu K computer	$10.51 \cdot 10^{15}$
2012	IBM Sequoia	$16.32 \cdot 10^{15}$
2013	NUDT Tianhe-2	$33.86 \cdot 10^{15}$
2016	Sunway TaihuLight	$93.00 \cdot 10^{15}$
2018	IBM Summit	$122.3 \cdot 10^{15}$

TABEL 1

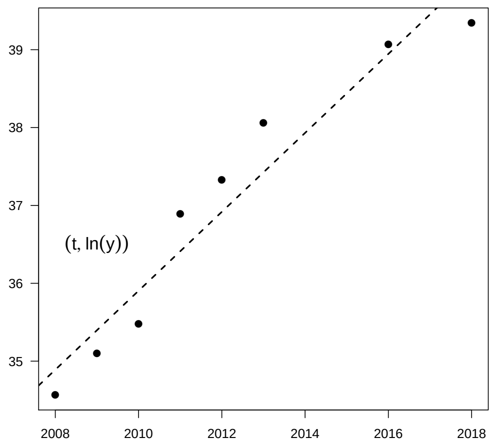
t	y	$\ln y$
2008	$1.026 \cdot 10^{15}$	34.564
2009	$1.759 \cdot 10^{15}$	35.104
2010	$2.566 \cdot 10^{15}$	35.481
2011	$10.51 \cdot 10^{15}$	36.891
2012	$16.32 \cdot 10^{15}$	37.331
2013	$33.86 \cdot 10^{15}$	38.061
2016	$93.00 \cdot 10^{15}$	39.071
2018	$122.3 \cdot 10^{15}$	39.345

TABEL 2

I nedenstående to koordinatsystemer er punkterne (t, y) hhv. $(t, \ln y)$ fra TABEL 2 indtegnet (sammen med stiplede grafer for nogle i første omgang ukendte funktioner):



FIGUR 1: Punkterne (t, y)



FIGUR 2: Punkterne $(t, \ln y)$

Det fremgår af FIGUR 2, at punkterne $(t, \ln y)$ tilnærmelsesvist ligger på en ret linie.

(a) Benyt mindste kvadraters metode (eng: *method of least squares*) til at bestemme forskriften for den bedste rette linie, $\ln y \simeq at+b$, gennem punkterne $(t, \ln y)$ fra TABEL 2.

(*Vink:* Se §4.4.1 Example 4 i lærebogen. Det er i orden at benytte C#-funktionen fra Projekt A eller en lommeregner til matrixmultiplikation.)

Den stiplede graf på FIGUR 2 *er grafen for den lineære funktion* $t \mapsto at+b$, *hvor* a *og* b *er de konstanter, som er bestemt ovenfor.*

(b) Begrund, at der gælder følgende tilnærmede forskrift for funktionen $y = y(t)$:

$$y = y(t) \simeq 1.42 \cdot 10^{15} \cdot e^{0.507(t-2008)} \tag{$*$}$$

(*Vink:* Man har tilnærmelsen $\ln y \simeq at+b$ for de i delspørgsmål (a) fundne konstanter a og b. Tag nu eksponentialfunktionen på begge sider af lighendstegnet. Regn med alle decimaler.)

Den stiplede graf på FIGUR 1 *er grafen for eksponentialfunktionen* $t \mapsto 1.42 \cdot 10^{15} \cdot e^{0.507(t-2008)}$ *fundet i* $(*)$ *ovenfor.*

(c) Benyt tilnærmelsen $(*)$ til at give et estimat på hvor mange FLOPS verdens bedste supercomputer kunne præstere i år 2000.

Det historiske faktum er, at verdens bedste supercomputer i år 2000 var IBM ASCI White, og denne kunne præstere $7.226 \cdot 10^{12}$ *FLOPS.*

Benyt tilnærmelsen $(*)$ til at give et estimat på hvor mange FLOPS verdens bedste supercomputer mon kan præstere i år 2025.

Opgave 4 [Programmering i C#] (25%)

In this project you have to implement a small number of functions related to the Gram–Schmidt process and calculating the determinant. To get started on the assignment one has to first download the project files from `https://absalon.instructure.com/courses/31798/files/folder/Projekt%20C`. Secondly, one should get an overview of the project files, try an open the files and have a look at them. Observe that most of the files are identical to those provided in the previous project. There are only a few new or modified files. Here is a brief summary of them:

ProjectC/AdvancedExtensions.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 C.*

ProjectC/MainClass.cs This file contains data for self-testing.

ProjectC/Program.cs This file is used to test your implementation in `AdvancedExtensions.cs` against the test data in `MainClass.cs`.

Your assignment is to finish the unimplemented methods in `ProjectC/AdvancedExtensions.cs`. You are welcome to add additional helper methods in `AdvancedExtensions.cs`, but you are not allowed to rename or otherwise alter the type signature of any of the existing methods. When submitting your solution to the programming part of Project C you are only allowed to upload the file `ProjectC/AdvancedExtensions.cs`.

Build and run the project in JetBrains Rider using `ProjectC/ProjectC.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 `ProjectC.exe` and is located in `ProjectC/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.

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