Miniprojekt 1

15. september 2011

I denne opgave skal I stifte bekendtskab med MATLAB og lære hvordan vi kan bruge MATLAB til at løse opgaver af den type, I har set indtil videre.

Nedenfor introduceres de funktioner fra MATLAB, der skal bruges til at løse dagens opgaver – du kan læse mere om funktionerne i MATLAB's dokumentation, men vær opmærksom på, at der bruges begreber, der først gennemgås senere. Desuden kan du finde informationer i appendiks D i bogen.

For at introducere funktionerne, gennemregnes Practice Problem 1 på side 46 i bogen til sidst på denne spiseseddel.

De opgaver, I skal regne, er fra de kapitler, der er blevet gennemgået indtil nu – forskellen er, at I her skal regne opgaver ved hjælp af MATLAB.

Opgaver:

- Opgave 94 og 95 side 55. I stedet for at taste "." ved hvert element (hver indgang) i matricen, kan elementerne (indgangene) i matricen i opgave 94 indtastes uden "." og derefter deles med 10.
- MATLAB-opgaver fra side 90 & 91: 2, 4, 5.

MATLAB-kommandoer vi får brug for i dag, er følgende:

`rref` Giver den række-reducerede echelonform af en matrix.

`mldivide`, `linsolve` giver én løsning til et konsistent ligningssystem; for et ikke-konsistent ligningssystem gives en *approksimativ* løsning i en forstand, vi vender tilbage til i slutningen af kurset.

`null` giver koefficienterne til de frie variable – hvis der er nogle frie variable.

Et generelt råd, når man bruger MATLAB: Læs dokumentationen for de kommandoer, du bruger. Et godt eksempel på, hvorfor det kan betale sig, er, at en kommando som `linsolve` virker anderledes, end man umiddelbart kunne forvente af navnet.

Hvis du ikke ved hvilken kommando, du kan bruge til at løse dit problem, eller hvis dokumentationen er uforståelig, så kan man næsten altid finde hjælp på nettet.

Som et eksempel på, hvordan MATLAB kan bruges, gennemgåes her Pratice Problem 1 på side 46 i bogen. For at løse Practice Problem 1, indtastes totalmatricen for ligningssystemet. Her kalder vi totalmatricen for `T`, men vi kan vælge et vilkårligt navn bestående af engelske bogstaver, samt tal. Bemærk, at der er forskel på store og små bogstaver.

```
T = [1 -1 -3 1 -1 -2 ; -2 2 6 0 -6 -6 ; 3 -2 -8 3 -5 -7]

T =

     1    -1    -3     1    -1    -2
    -2     2     6     0    -6    -6
     3    -2    -8     3    -5    -7

>> rref(T)

ans =

     1     0    -2     0     1     2
     0     1     1     0    -2    -1
     0     0     0     1    -4    -5
```

Er vi blot interesseret i en partikulær løsning til ligningssystemet, kan vi benytte `linsolve` eller `mldivide` – her skal vi dog huske at undersøge, om det rent faktisk er en løsning, der er fundet.

Først indtastes koefficientmatricen for ligningssystemet og højresiden i ligningen.

```
>> A = [1 -1 -3 1 -1 ; -2 2 6 0 -6 ; 3 -2 -8 3 -5]

A =

     1    -1    -3     1    -1
    -2     2     6     0    -6
     3    -2    -8     3    -5

>> b = -[2 ; 6 ; 7]

b =

    -2
    -6
    -7
```

Har vi indtastet oplysninger på denne måde, kan vi også lave `T` ved at samle `A` og `b`:

```
>> T = [A b]

T =

     1    -1    -3     1    -1    -2
    -2     2     6     0    -6    -6
     3    -2    -8     3    -5    -7
```

Her er det vigtigt, at `A` og `b` har samme antal rækker. Med `linsolve` finder vi en løsning til $Ax = b$:

```
>> x = linsolve(A, b)

x =

         0
    1.8750
   -0.3750
         0
    1.2500
```

For at tjekke, at det, vi har fundet, er en løsning, kan vi undersøge, om $Ax - b = 0$.

```
>> A*x - b

ans =

   1.0e-14 *

    0.1110
    0.2665
    0.0888
```

Det ses, at forskellen mellem Ax og b er af størrelsesorden 10^{-14}, hvilket kan tilskrives numeriske fejl.

Et alternativ til at benytte `rref` for at finde den fuldstændige løsning til et ligningssystem er beskrevet på hjemmesiden

`http://www.mathworks.com/help/techdoc/math/f4-983672.html`

Ideen er, at vi ved, at den fuldstændige løsning til et lineært ligningssystem består af en partikulær løsning til systemet (som kan findes med `mldivide` eller `linsolve`) plus den fuldstændige løsning til det tilhørende homogene ligningssystem (som vi kan finde med `null`). Beskrivelsen af kommandoen `null` i dokumentationen giver nok ikke mening på nuværende tidspunkt, men output er forståeligt nok

```
>> null(A)

ans =

   -0.4584    0.6799
    0.4932   -0.1239
   -0.1412    0.4120
    0.7040    0.5761
    0.1760    0.1440
```

Det vil sige, at

$$(-0.4584, 0.4932, -0.1412, 0.7040, 0.1760)$$
$$(0.6799, -0.1239, 0.4120, 0.5761, 0.1440)$$

er koefficientvektorerne til de frie variable for A.