
Matematikk 1000

Øvingsoppgaver i numerikk – leksjon 4

Matriser

Løsningsforslag

Oppgave 1 – Redusert trappeform og løsning av lineære likningssystemer

a) Totalmatrisa blir

$$\begin{bmatrix} 5 & 1 & 1 & 2 & 2 \\ 1 & -1 & 1 & 7 & 30 \\ 2 & 1 & -5 & 0 & -16 \\ 2 & 0 & 1 & 3 & 17 \end{bmatrix}.$$

Vi tilordner dette i MATLAB:

```
>> format compact
>> T=[5 1 1 -2 2
1 -1 1 7 30
2 1 0 -5 -16
2 0 1 3 17]
T =
     5     1     1    -2     2
     1    -1     1     7    30
     2     1     0    -5   -16
     2     0     1     3    17
```

b) Vi lar MATLAB ta seg av rekkereduseringa:

```
>> rref(T)
ans =
     1     0     0     0     1
     0     1     0     0     2
     0     0     1     0     3
     0     0     0     1     4
```

Svar: $x_1 = 1, x_2 = 2, x_3 = 3$ og $x_4 = 4$. På vektor-form:

$$\vec{x} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

c) Vi setter opp totalmatrisa og bruker **rref**-funksjonen:

```
>> T=[1 2 0 -2 1 0; 2 -1 3 1 0 7; 3 1 3 1 -5 -8]
T =
     1     2     0    -2     1     0
     2    -1     3     1     0     7
     3     1     3     1    -5    -8
>> rref(T)
ans =
 1.0000         0  1.2000         0  0.2000  2.8000
         0  1.0000 -0.6000         0 -2.6000 -8.9000
         0         0         0  1.0000 -3.0000 -7.5000
```

Det gir

$$\begin{aligned} x_1 + 1.2x_3 + 0.2x_5 &= 2.8 \\ x_2 - 0.6x_3 - 2.6x_5 &= -8.9 \\ x_4 - 3x_5 &= -7.5 \end{aligned}$$

som igjen gir:

$$\begin{aligned} x_1 &= 2.8 - 1.2x_3 - 0.2x_5 \\ x_2 &= -8.9 + 0.6x_3 + 2.6x_5 \\ x_4 &= -7.5 + 3x_5 \end{aligned}$$

der x_3 og x_5 er frie. På parametrisk vektor-form:

$$\vec{x} = \begin{bmatrix} 2.8 \\ -8.9 \\ 0 \\ -7.5 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} -1.2 \\ 0.6 \\ 1 \\ 0 \\ 0 \end{bmatrix} + x_5 \begin{bmatrix} -0.2 \\ 2.6 \\ 0 \\ 3 \\ 1 \end{bmatrix}.$$

Oppgave 2 – Med og uten punktum

a) Vi kan for eksempel tilordne A og B slik:

```
>> A=[1 2 3; 4 5 6; 7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9
>> B=[-2 -1 0; 0 1 0; 7 -2 1]
B =
    -2    -1     0
     0     1     0
     7    -2     1
```

b) Vi regner ut produktene i kommandovinduet:

```
>> A.*B
ans =
    -2    -2     0
     0     5     0
    49   -16     9
>> A*B
ans =
    19    -5     3
    34   -11     6
    49   -17     9
```

Vi ser at det i første produktet har matrisene blitt multiplisert elementvis. Den andre produktet, uten punktum, er matriseproduktet:

$$AB = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} -2 & -1 & 0 \\ 0 & 1 & 0 \\ 7 & -2 & 1 \end{bmatrix} = \begin{bmatrix} -2+0+21 & -1+2-6 & 0+0+3 \\ -8+0+42 & -4+5-12 & 0+0+6 \\ -14+0+63 & -7+8-18 & 0+0+9 \end{bmatrix} = \begin{bmatrix} 19 & -5 & 3 \\ 34 & -11 & 6 \\ 49 & -17 & 9 \end{bmatrix}$$

Når det gjelder matriseproduktet, spiller rekkefølga vi multipliserer i en rolle:

```
>> B*A
ans =
    -6    -9   -12
     4     5     6
     6    12    18
```

– altså får vi noe helt annet enn AB . For “prikk-multiplikasjonen” i MATLAB spiller rekkefølga ingen rolle, derimot.

Oppgave 3 – Noen spesielle matriser

Svarene i deloppgave a) - e) blir:

```
>> eye(3)
ans =
     1     0     0
     0     1     0
     0     0     1
>> ones(2,3)
ans =
     1     1     1
     1     1     1
>> -2*ones(2,3)
ans =
    -2    -2    -2
    -2    -2    -2
>> ones(3,3)-eye(3)
ans =
     0     1     1
     1     0     1
     1     1     0
>> diag(1:4)
ans =
     1     0     0     0
     0     2     0     0
     0     0     3     0
     0     0     0     4
```

I deloppgave e) og f) lager vi matriser med tilfeldige tall.

```
>> rand(2,2)
ans =
    0.8143    0.9293
    0.2435    0.3500
>> rand(2,2)
ans =
    0.1966    0.6160
    0.2511    0.4733
>> rand(2,2)
ans =
    0.3517    0.5853
    0.8308    0.5497
```

“» `rand(2,2)`” gir ei 2×2 -matrise med tilfeldige tall mellom 0 og 1. Merk at vi får ei ny matrise hver gang selv om vi skriver den samme kommandoen.

Vi regner ut noen matriser med kommandoen gitt i deloppgave f):

```
>> 10*(rand(2,2)-0.5*ones(2,2))
ans =
    -3.3435    -2.3703
     1.0198     1.5408
>> 10*(rand(2,2)-0.5*ones(2,2))
ans =
     1.8921    -0.4946
     2.4815    -4.1618
>> 10*(rand(2,2)-0.5*ones(2,2))
ans =
    -2.7102    -3.4762
     4.1334     3.2582
>> 10*(rand(2,2)-0.5*ones(2,2))
ans =
     0.3834    -4.2182
     4.9613    -0.5732
```

Det ser ut til at vi får laga 2×2 -matriser med tilfeldige tall i intervallet $[-5, 5]$. Når vi trekker $0.5 \cdot \text{ones}(2, 2)$ fra $\text{rand}(2, 2)$, får vi tilfeldige tall mellom -0.5 og 0.5. Ganger vi dette med 10, får vi tilfeldige tall mellom -5 og 5.

Oppgave 4 – Invertering

a) I kommandovinduet:

```
>> A=[1 2; 1 1];
>> B=[4 -2 0; 1 0 1; 2 -2 -2];
>> C=[3 -7 1; 3 3 -1; 1 1 1];
>> T=[A, eye(2)]
T =
     1     2     1     0
     1     1     0     1
>> rref(T)
ans =
     1     0    -1     2
     0     1     1    -1
>> rref([B, eye(3)])
ans =
     1.0000         0     1.0000         0     1.0000         0
         0     1.0000     2.0000         0     1.0000    -0.5000
         0         0         0     1.0000    -2.0000    -1.0000
>> rref([C, eye(3)])
ans =
```

1.0000	0	0	0.1000	0.2000	0.1000
0	1.0000	0	-0.1000	0.0500	0.1500
0	0	1.0000	0	-0.2500	0.7500

Av dette ser vi at

$$A^{-1} = \begin{bmatrix} -1 & 2 \\ 1 & -1 \end{bmatrix}.$$

Videre ser vi at vi slett ikke har fått fram I_3 til venstre når vi (d.v.s. MATLAB) har rekkeredusert matrisa $[B|I_3]$ til redusert trappeform. Det betyr at B ikke er invertibel. Til sist:

$$C^{-1} = \begin{bmatrix} 0.1000 & 0.2000 & 0.1000 \\ -0.1000 & 0.0500 & 0.1500 \\ 0 & -0.2500 & 0.7500 \end{bmatrix}.$$

b) Vi inverterer matrisene ved hjelp av `inv`-funksjonen:

```
>> inv(A)
ans =
    -1     2
     1    -1
>> inv(B)
Warning: Matrix is singular to working precision.
ans =
    Inf    Inf    Inf
    Inf    Inf    Inf
    Inf    Inf    Inf
>> inv(C)
ans =
    0.1000    0.2000    0.1000
   -0.1000    0.0500    0.1500
         0   -0.2500    0.7500
```

For A og C får vi det samme som i a)¹. For B -matrisa, derimot, får vi “**Inf**” – *infinity* – for alle elementer. Dette har visst ikke gått så bra. Men så får vi da også en advarsel om at matrisa ser ut til å være *singulær* – altså at den ikke ser ut til å være invertibel.

c) Vi tilordner matrisa og vektorene:

```
>> A=[1 2 0 0; 1 -2 1 0; 0 1 -2 1; 0 0 1 -3];
>> b=[2; 3; 0; -7];
>> c=[0; 1; 0; 2];
>> d=[3; 7; pi; 2+3*i];
```

Så finner vi løsningene:

¹Det skulle bare mangle.

```

>> inv(A)*b
ans =
    1.7647
    0.1176
    1.4706
    2.8235
>> inv(A)*c
ans =
    0.8235
   -0.4118
   -0.6471
   -0.8824
>> inv(A)*d
ans =
    6.6970 + 0.3529i
   -1.8485 - 0.1765i
   -3.3941 - 0.7059i
   -1.7980 - 1.2353i

```

Vi minner om at når vi skal gjøre flere nesten identiske kommandoer, lønner det seg å bruke pil-tastene. I stedet for å skrive “`inv(A)*c`” kan man trykke pil opp, få opp kommandoen før, altså “`inv(A)*b`”, og så bare endre `b` til `c`.

d) Vi gjør det samme som i c):

```

>> A=[1.7 2.1; 3.3 4.1];
>> B=[2.5 -3.7; 2 3.2];
>> b=[0.96; 1.87];
>> inv(A)*b
ans =
    0.2250
    0.2750
>> inv(B)*b
ans =
    0.6488
    0.1789

```

Altså får vi løsningene

$$\begin{bmatrix} 0.2250 \\ 0.2750 \end{bmatrix} \quad \text{og} \quad \begin{bmatrix} 0.6488 \\ 0.1789 \end{bmatrix}$$

for de to likningene. Så gjør vi det samme med $\vec{b}$ avrunda til

$$\begin{bmatrix} 1.0 \\ 1.9 \end{bmatrix}.$$

Vi får nå løsningene

```
>> b=[1.0; 1.9];
>> inv(A)*b
ans =
    2.7500
   -1.7500
>> inv(B)*b
ans =
    0.6643
    0.1786
```

(Også her lønner det seg å bruke pil-tastane til å bla seg bakover i kommando-historikken.) Vi ser at avrundninga i $\vec{b}$ gir omtrent den samme løsninga for $B\vec{x} = \vec{b}$, mens løsninga for den første likninga, $A\vec{x} = \vec{b}$, ikke er i nærheten! Hva kan grunnen være til dette, da? Dette har med å gjøre at selv om A forsåvidt er invertibel, er den veldig *nær* å ikke være det. Med ei veldig lita justering av matrisa, ville den ha blitt singulær. Dette får den konsekvensen at likningssystemet blir veldig ustabilt i forhold til små justeringer i $\vec{b}$. Moralen er at man skal være forsiktig med å stole blindt på numeriske løsninger.

Oppgave 5 – Testing av “regneregler”

I hver oppgave genererer vi tilfeldige matriser A , B og eventuelt også C på samme måte som i deloppgave 2 f).

```
a) >> A=10*(rand(3,3)-0.5*ones(3,3));
>> B=10*(rand(3,3)-0.5*ones(3,3));
>> A+B-(B+A)
ans =
     0     0     0
     0     0     0
     0     0     0
>> A=10*(rand(3,3)-0.5*ones(3,3));
>> B=10*(rand(3,3)-0.5*ones(3,3));
>> A+B-(B+A)
ans =
     0     0     0
     0     0     0
     0     0     0
>> A=10*(rand(3,3)-0.5*ones(3,3));
>> B=10*(rand(3,3)-0.5*ones(3,3));
>> A+B-(B+A)
ans =
     0     0     0
     0     0     0
```

0 0 0

Her har vi laga tilfeldige A - og B -matriser tre ganger. Hver gang har vi fått at differansen $A + B - (B + A)$ blir nullmatrisa. Dette tyder på at regneregelen er rett.

```
b) >> A=10*(rand(3,3)-0.5*ones(3,3));
>> B=10*(rand(3,3)-0.5*ones(3,3));
>> C=10*(rand(3,3)-0.5*ones(3,3));
>> (A*B)*C-A*(B*C)
ans =
    1.0e-14 *
    0.3553    0.7105    0.5329
         0   -0.7105   -0.1776
    0.3553         0    0.1776
>> A=10*(rand(3,3)-0.5*ones(3,3));
>> B=10*(rand(3,3)-0.5*ones(3,3));
>> C=10*(rand(3,3)-0.5*ones(3,3));
>> (A*B)*C-A*(B*C)
ans =
    1.0e-13 *
    0.0711   -0.0355    0.1066
    0.1421    0.0711         0
    0.0711         0         0
>> A=10*(rand(3,3)-0.5*ones(3,3));
>> B=10*(rand(3,3)-0.5*ones(3,3));
>> C=10*(rand(3,3)-0.5*ones(3,3));
>> (A*B)*C-A*(B*C)
ans =
    1.0e-13 *
    0.1421         0   -0.0711
         0    0.1421   -0.2842
   -0.1066         0    0.0355
```

Også her har vi testa regneregelen med tre ulike sett av tilfeldige matriser. Matrisene har blitt skrevet på en slik måte at alle tall i matrisa skal ganges med det tallet som står øverst. Hver gang har vi fått matriser der dette tallet er 10^{-13} eller mindre. Med andre ord er alle elementene sammenlignbare med maskinpresisjonen (`eps`). Derfor kan vi ikke uten videre si at dette er ulik null. Dette resultatet tyder altså på at regneregelen er rett.

```
c) >> A=10*(rand(3,3)-0.5*ones(3,3));
>> B=10*(rand(3,3)-0.5*ones(3,3));
>> (A*B).'-(A.'*B.')
ans =
   -7.6774  -11.3022   24.8796
  -10.6053    0.2678  -15.6531
   -1.1223   38.7436    7.4096
```

Dette resultatet er slett ikke lik null-matrissa. Vi har funnet et moteksempel som viser tydelig at “regneregelen” ikke stemmer.

d) I første omgang kan vi fortsette med samme A og B:

```
>> (A*B).'-(B.'*A.')
```

```
ans =
```

```
    0    0    0
```

```
    0    0    0
```

```
    0    0    0
```

```
>> A=10*(rand(3,3)-0.5*ones(3,3));
```

```
>> B=10*(rand(3,3)-0.5*ones(3,3));
```

```
>> (A*B).'-(B.'*A.')
```

```
ans =
```

```
    0    0    0
```

```
    0    0    0
```

```
    0    0    0
```

```
>> A=10*(rand(3,3)-0.5*ones(3,3));
```

```
>> B=10*(rand(3,3)-0.5*ones(3,3));
```

```
>> (A*B).'-(B.'*A.')
```

```
ans =
```

```
    0    0    0
```

```
    0    0    0
```

```
    0    0    0
```

Mye tyder på at $(AB)^T$ faktisk *er* lik $B^T A^T$.

e) >> det(A*B)-det(A)*det(B)

```
ans =
```

```
-3.9080e-13
```

```
>> A=10*(rand(3,3)-0.5*ones(3,3));
```

```
>> B=10*(rand(3,3)-0.5*ones(3,3));
```

```
>> det(A*B)-det(A)*det(B)
```

```
ans =
```

```
-4.5475e-13
```

```
>> A=10*(rand(3,3)-0.5*ones(3,3));
```

```
>> B=10*(rand(3,3)-0.5*ones(3,3));
```

```
>> det(A*B)-det(A)*det(B)
```

```
ans =
```

```
9.0949e-13
```

Svaret, som denne gangen blir et tall, ikke ei matrise, ser ut til å bli veldig nær null. Det ser ut til at regneregelen stemmer.

f) Vi skal teste om vi får ulike svar om vi regner ut potensene med eller uten punktum:

```
>> A^4-A.^4
```

```
ans =
```

```

1.0e+03 *
    0.1356   -0.9036   -0.5729
   -0.7370    1.2437    0.8025
   -0.3747    0.4655    0.2475

```

Som vi ser av faktoren “1.0e+03”, altså 10^3 , er denne matrisa langt fra null; regnereglen er feil. A^4 er altså noe helt anna enn det vi får om vi opphøyer hvert element i A i 4.

g) Vi kan lage 3×3 -diagonalmatriser med tilfeldige tall langs diagonalen slik:

```

>> x=10*(rand(1,3)-0.5*ones(1,3))
x =
    1.9811    1.6653   -3.2187
>> D=diag(x)
D =
    1.9811         0         0
         0    1.6653         0
         0         0   -3.2187

```

Vi tester regneregelen fra f) igjen – denne gangen for diagonalmatriser:

```

>> D^4-D.^4
ans =
    1.0e-13 *
    0.0178         0         0
         0         0         0
         0         0    0.1421
>> x=10*(rand(1,3)-0.5*ones(1,3));
>> D=diag(x);
>> D^4-D.^4
ans =
    1.0e-13 *
         0         0         0
         0         0         0
         0         0   -0.1421
>> x=10*(rand(1,3)-0.5*ones(1,3));
>> D=diag(x);
>> D^4-D.^4
ans =
    1.0e-13 *
         0         0         0
         0   -0.0003         0
         0         0    0.2842

```

Det ser ut til at det er enklere å regne ut potenser av diagonalmatriser enn andre matriser.