Matematikk 1000

Øvingsoppgaver i numerikk – leksjon 8

Matriser

Som nevnt er MATLAB en slags forkortelse for "*Matrix laboratory*". MATLAB er i utgangspunktet laget spesielt for å kunne regne med matriser; mye av skrivemåten til MATLAB er bygd opp rundt matriser, og operasjoner som kan skrives som en eller annen matrise-operasjon, går ofte veldig fort.

Vi var såvidt innom matriser i leksjon 1. I oppgave 5 i leksjon 1 gikk vi gjennom hvordan vi kan tilordne matriser og hvordan vi indekserer elementene i matrisa (en vektor har én indeks, ei matrise har to indekser). Repetér gjerne denne oppgava før du går videre med dette settet.

Oppgave 1 – Redusert trappeform og løsning av lineære likningssystemer

a) Bestem totalmatrisa til dette likningssystemet og tilordne den i MATLAB:

$$\begin{array}{rrrrcr} 5x_1 & +x_2 & +x_3 & -2x_4 & = & 2 \\ x_1 & -x_2 & +x_3 & +7x_4 & = & 30 \\ 2x_1 & +x_2 & & -5x_4 & = & -16 \\ 2x_1 & & +x_3 & +3x_4 & = & 17 \end{array} .$$

b) `rref`-funksjonen rekkereduserer ei matrise til redusert trappeform. Bruk denne funksjonen til å få totalmatrisa fra a) på redusert trappeform, og bestem løsninga på likningssystemet. (Skriv løsninga opp på papir.)

c) På samme måte som over: Finn den generelle løsninga av dette likningssystemet:

$$\begin{array}{rrrrrcr} x_1 & +2x_2 & & -2x_4 & +x_5 & = & 0 \\ 2x_1 & -x_2 & +3x_3 & +x_4 & & = & 7 \\ 3x_1 & x_2 & +3x_3 & +x_4 & -5x_5 & = & -8 \end{array} .$$

Oppgave 2 – Med og uten punktum

a) La A og B være to ulike 3×3-matriser med elementer som du bestemmer selv. Tilordne disse i kommandovinduet i MATLAB.

b) Regn ut `A.*B` og `A*B`. Forstår du forskjellen?

Oppgave 3 – Noen spesielle matriser

Noen matriser er såpass spesielle at det finnes egne funksjoner i MATLAB for å lage dem på en rask måte. Her er noen eksempler

- Identitetsmatrisa I_n kan lages slik: `eye(4)`. Denne kommandoen lager I_4.
- Ei $m \times n$-matrise med bare ett-tall kan lages slik: `ones(3,4)`. Her har vi satt $m = 3$ og $n = 4$.
- Man kan lage ei matrise med bare null-elementer med kommandoen `zeros(3,4)`.
- Komandoen `rand(m,n)` lager ei $m \times n$-matrise som består av tilfeldige tall mellom 0 og 1.
- `diag(v)`, der `v` er en vektor, gir ei diagonalmatrise med vektoren $\vec{v}$ langs diagonalen.

For ordens skyld: Ei diagonalmatrise har like mange rekker som sløyler, og den har bare null-elementer utenom diagonalen. Diagonalen er den remsa av tall som går fra oppe til venstre til nede til høgre.

I kommandovinduet i MATLAB: Tilordne følgende matriser uten å skrive de inn element for element:

a) $I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

b) $\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$.

c) $\begin{bmatrix} -2 & -2 & -2 \\ -2 & -2 & -2 \end{bmatrix}$.

d) $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$.

e) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$.

f) Ei 2×2-matrise med tilfeldige tall mellom 0 og 1.

g) Hvilken matrise blir generert med denne kommandoen:
"» `10*(rand(2,2)-0.5*ones(2,2))`" ?

Oppgave 4 – Testing av "regneregler"

Om man er usikker på om regneregler gjelder eller ikke, kan man ofte teste dem ved å sjekke noen tilfeldige eksempler. Hvis regneregelen er feil, vil det som regel fort dukke opp et moteksempel. Hvis man ikke finner noen mot-eksempler uten videre, er det mye som tyder på at regneregelen er rett. Men dette er ikke noe bevis!

Vi illustrerer tankegangen: Vi vil undersøke om $AB = BA$ for 3×3-matriser. Dersom dette skal være sant, må vi også ha at $AB - BA$ blir ei matrise med bare null-elementer.

```
>> format compact
>> A=5*(rand(3,3)-0.5*ones(3,3));
>> B=5*(rand(3,3)-0.5*ones(3,3));
>> A*B-B*A
ans =
  -14.1217   -8.1372    0.2061
    6.2475    7.7410   -4.3886
    1.6201   -2.3556    6.3807
```

Her har vi valgt å la `A` og `B` være 3×3-matriser med tilfeldige tall i intervallet $[-2.5, 2.5]$. Vi ser klart og tydelig at $AB - BA$ slett ikke ble null-matrisa; vi har vist at "regneregelen" $AB = BA$ er feil.

La oss teste denne regelen: $I_3 A = A$:

```
>> I3=eye(3);
>> A=5*(rand(3,3)-0.5*ones(3,3));
>> I3*A-A
ans =
     0     0     0
     0     0     0
     0     0     0
>> A=5*(rand(3,3)-0.5*ones(3,3));
>> I3*A-A
```

```
ans =
     0     0     0
     0     0     0
     0     0     0
>> A=5*(rand(3,3)-0.5*ones(3,3));
>> I3*A-A
ans =
     0     0     0
     0     0     0
     0     0     0
```

Merk at kommandoen "`A=5*(rand(3,3)-0.5*ones(3,3));`" lager ulike matriser hver gang. Vi ser at vi får null-matrisa til svar alle de tre gangene vi har regnet ut $I_3A - A$. Dette *beviser* ikke at $I_3A = A$, men det gir oss god grunn til å tro at det stemmer.

På samme måte: Test disse "regnereglene" for 3×3-matriser:

a) $A + B = B + A$.

b) $(AB)C = A(BC)$.

c) $(AB)^T = A^T B^T$.
"A^T" betyr A transponert. *Transponering* går ut på å bytte om på hva som er rekker og hva som er søyler i ei matrise. Dette kan gjøres med punktum-apostroff i MATLAB; vi får A^T ved "`» A.';`".

d) $(AB)^T = B^T A^T$.

e) $\det(AB) = \det A \cdot \det B$.

f) Test om A^4 er lik det du får når du opphøyer hvert element i A i fjerde.

g) Test om D^4 er lik det du får når du opphøyer hvert element i D i fjerde når D er ei diagonalmatrise.

Ekstra-oppgave

Implementér funksjonen `rref` selv. Altså: Lag ei funksjonfil som gir deg den reduserte trappeforma av den matrisa du gir når du kaller funksjonen.