

## MATLAB-föreläsning 2 (5), 8 september, 2016

## INNEHÅLL

- Sekvenser, vektorer och matriser
- Upprepning med `for`-slingor (inledning)

## VEKTORER OCH SEKVENSER

Fram till nu har vi arbetat med variabler som har varit skalärer eller textsträngar. Man kan även arbeta med variabler som utgörs av en sekvens av skalärer. Dessa kallas vektorer i MATLAB. För att skapa en variabel `a` som innehåller alla heltal från 0-10 kan vi skriva `a=[0:1:10]`. Vi kan använda de vanliga operationerna `+` och `-` enligt de definitioner som gäller för vanlig vektoralgebra.

**Uppgift 1**

Utöka programmet i **Uppgift 5 - FÖ1** så att det beräknar luftmotståndet hos objektet för hastigheterna  $v = 0, 10, 20, 30, 40, 50, 100$  m/s.

## VEKTORER OCH MATRISER

Fram till nu har vi arbetat med variabler som har varit skalärer eller textsträngar. Man kan även arbeta med variabler är **vektorer**. Vektorer skapas i MATLAB på följande sätt

```
>>v=[3,2,9]      >>u=[1;0;7]
v =              u =
    3    2    9          1
                   0
                   7
```

MATLAB skiljer på rad och kolonnvektorer. Enskilda element i en vektor refereras till på följande sätt

```
>>a=v(1)        >>b=v(2:3)      >>c=u(2)          >>d=u(1:2)
a =              b =              c =              d =
    3              2    9              0              1
                                      0
```

Addition och subtraktion av två vektorer skrivs som vanligt med tecknen + eller -. Om du vill multiplicera en vektor med en skalär,  $3\vec{v}$ , skriver du  $3*v$ .

Det finns även andra funktioner i MATLAB som opererar på vektorer. **Skalärprodukten** mellan två vektorer,  $\vec{v} \cdot \vec{u}$ , kan beräknas med funktionen `dot(u,v)`. **Vektorprodukten** (kryssprodukten) mellan två vektorer beräknas med funktionen `cross(u,v)`. Man kan också enkelt beräkna **längden av en vektor**,  $\|\vec{v}\|$ , med kommandon `norm(v)`.

**Uppgift 2** Skapa följande vektorer i MATLAB

$$\vec{u} = \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix} \quad \vec{v} = \begin{bmatrix} 2 \\ 2 \\ -4 \end{bmatrix} \quad \vec{w} = \begin{bmatrix} 3 \\ 3 \\ 3 \\ 2 \end{bmatrix}$$

I det fall de går, beräkna

- A.  $6\vec{u} + 2\vec{v}$
- B.  $5\vec{u} + 10\vec{w}$
- C.  $\vec{u} \cdot \vec{v}$ ,  $\vec{v} \cdot \vec{w}$ ,  $\vec{u} \cdot \vec{u}$ ,  $\vec{u} \cdot \vec{v} - \|\vec{w}\|$
- D.  $\vec{u} \times \vec{v}$ ,  $\vec{u} \times \vec{w}$
- E. längden av vektorerna  $\vec{u}$  och  $\vec{w}$
- F. vektorprodukten mellan  $\vec{u}$  och  $\vec{v}$
- G.  $\vec{u} \cdot \vec{v}$  utan att använda `dot`
- H. längden av  $\vec{v}$  utan att använda `norm`

**Uppgift 3** Skriv ett MATLAB-program som beräknar den ortogonal projektionen av en vektor,  $\vec{v}$ , på en linje mellan två punkter  $P$  och  $Q$ . Indata till programmet ska vara  $P$ ,  $Q$  och  $\vec{v}$  och utdata ska vara den vektor som ges av den ortogonala projektionen. (Se algebraföreläsning 3.)

MATRISER

En matris,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

skapas i MATLAB på följande sätt

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

```
A =
```

```

1     2     3
4     5     6
7     8     9
```

Enskilda element, rader och kolonner i  $A$  refereras till på följande sätt (första index refererar till rad och andra index till kolonn)

```
>>c=A(1,2)
```

```
c =
```

```
2
```

```
>>b=A(:,3)
```

```
b =
```

```
3
```

```
6
```

```
9
```

```
>>d=A(1,:)
```

```
d =
```

```
1 2 3
```

Beteckningen  $(:)$  representerar antingen en hel rad eller en hel kolonn. Man kan lika gärna skriva

```
>>b=A(1:3,3)
```

```
b =
```

```
3
```

```
6
```

```
9
```

```
>>d=A(1,1:3)
```

```
d =
```

```
1 2 3
```

Man kan också skapa delmatriser av en större matris,

```
>>B=A([1 2],[2 3])
```

```
A =
```

```
2 3
```

```
5 6
```

```
>>C=A([2 3],[1 3])
```

```
C =
```

```
4 6
```

```
7 9
```

eller sätta ihop två matriser  $A$  och  $B$ , i MATLAB, horisontellt med  $[A \ B]$  om antalet rader i de två matriserna är samma eller vertikalt med  $[A; B]$  om antal kolonner är samma.

**Uppgift 4** Skriv in följande matriser och vektor i MATLAB

$$A = \begin{bmatrix} 1 & 3 & 8 \\ 6 & -5 & 2 \\ 11 & 9 & 6 \\ 0 & 1 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 \\ 5 & 6 \\ 8 & 9 \end{bmatrix} \quad \vec{c} = \begin{bmatrix} 3 \\ 0 \\ 5 \end{bmatrix}$$

A. Låt 1:a och 4:a raden i  $A$  byta plats. Byt sedan plats på 2:a och 3:e kolonnen.

B. Utöka matrisen  $B$  med kolonnvektorn  $\vec{c}$  och sätt sedan ihop den utökade matrisen med matrisen  $A$ .

**Uppgift 5** Skapa följande kolonnvektorer i Matlab

$$\vec{c}_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} \quad \vec{c}_2 = \begin{bmatrix} 3 \\ 4 \\ 6 \end{bmatrix} \quad \vec{c}_3 = \begin{bmatrix} 9 \\ 8 \\ 7 \end{bmatrix}$$

och använd dessa för att skapa matrisen  $C = [\vec{c}_1 \ \vec{c}_2 \ \vec{c}_3]$ .

**Uppgift 6** Skapa följande matris

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 0 & 1 & 2 \\ 3 & 4 & 5 & 6 \end{bmatrix}$$

och bilda sedan de  $2 \times 2$ -delmatriserna  $A_{ij}$ ,  $i = 1, 2$  och  $j = 1, 2$  som ges av

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

genom att ta ut lämpliga delar av matrisen  $A$ . Försök sedan återskapa  $A$  med hjälp av delmatriserna  $A_{11}$ ,  $A_{12}$ ,  $A_{21}$  och  $A_{22}$ .

**Uppgift 7 - test**

Vad blir värdet på variabeln `resultat` när nedanstående rader körs i MATLAB?

```
vekA=[1, 2, 3, 4];
vekB=[-1, 0, 10, 2, 6];
resultat=vekA(1:2)+2*vekB(4:5)
```

**Uppgift 8 - test** En matris  $A$  är definierad i enligt

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

```
A =
```

```
     1     2     3
     4     5     6
     7     8     9
```

Hur kommer matrisen  $E$  att se ut efter följande kommandon

```
B=A(1:3,2);
C=A(2,:);
D=[C, A(1)];
E=[A,B;D]
```

## SEKVENSER

Man kan även se en vektor som en sekvens av skalärer. En sekvens kan exempelvis användas för att göra funktionstabeller. För att skapa en variabel  $a$  som innehåller alla heltal från 0-10 kan vi skriva  $a=[0:1:10]$ . Ett annat användbart kommando är `linspace`. Gör `help linspace` för att ta reda på hur det fungerar.

## KOMPONENTVISA OPERATIONER, .-NOTATION

Multiplikation ( $*$ ), division ( $/$ ) och upphöjt till ( $\wedge$ ) av vektorer följer vanliga vektoralgebraiska regler i MATLAB.

Ibland kan det hända att man vill utföra ovanstående operationer på varje element för sig i en vektor. Antag att vi för värdena  $x = 1, 2, 3, 4$  och  $5$  vill beräkna  $x^2$  dvs  $1^2, 2^2, 3^2, 4^2$  och  $5^2$ . Om vi i MATLAB skapar en vektor  $x=[1, 2, 3, 4, 5]$  och skriver  $x^2$  så kommer vi få ett felmeddelande som säger att operationen inte är definierad. Varför?

För att kunna hantera komponentvisa operationer för multiplikation, division och upphöjt till har man infört tre nya operationer i MATLAB.  $(.*)$  för multiplikation,  $(./)$  för division, och  $(.^)$  för upphöjt till.

MATLABS inbyggda funktioner som `sin` och `exp` osv opererar elementvis på vektorer (och matriser).

## GRAFISK REPRESENTATION AV DATA

För att rita upp grafer använder man kommandot `plot(x,y)` där  $x$  och  $y$  är två vektorer av samma dimension som innehåller de värden som ska plottas. Med kommandot `title`

ger man grafen en titel (exempelvis `title('Min graf')`) och med kommandona `xlabel` och `ylabel` döper man axlarna i grafen.

### Uppgift 9 - test

Låt  $x_1=[1, 1, 3, 4]$ ,  $x_2=[3, -1, 2, 1]$  och  $x_3=[-2, -1, 0, 1, 2]$ . Vad blir resultatet av följande uttryck

- A.  $x_1.*x_1$
- B.  $x_1./x_2$
- C.  $x_1.^2$
- D.  $x_1.*x_2$
- E.  $x_1.^x_2$
- F.  $x_2.*x_3$

### Uppgift 10

Skriv ett MATLAB-program som ritar upp funktionerna  $y_1 = \sin(4x) - 0.5x^2$  och  $y_2 = e^x - 1$  på intervallet  $x = [0, 2]$ . Lägg in en förklaring till grafen med hjälp av en titel och text på axlarna.

### Uppgift 11

Utöka programmet i **Uppgift 1** så att det beräknar luftmotståndet hos objektet för hastigheterna  $v = 10, 20, 30, 40, 50, 100$  m/s. Programmet ska även plotta luftmotståndet som funktion av hastigheten.

## UPPREPNING

Ibland vill man utföra samma beräkning eller ett antal satser flera gånger. Detta kan göras på olika sätt beroende på vad som styr hur många gånger satserna ska upprepas.

Ett sätt är att använda en så kallad **for**-slinga. Denna använder man om man på förhand vet hur många gången en viss sats ska utföras. Generellt så skriver man en **for**-slinga på följande sätt

```
for ii=1:slut
    satserna som skall utföras
end
```

I definitionen av **for**-slingan specificeras att variabeln **ii** ska börja på värdet 1 och anta

alla heltal från 1 till `slut`. För varje värde på `ii` utförs satserna som är specificerade i slingan. En `for`-slinga avslutas alltid med `end`.

Som exempel skriver följande rader ut värdet på en variabel `tal` på skärmen 5 gånger. Värdet på `tal` kommer att ges av variabeln `ii`. Vilka tal kommer vi att se på skärmen?

```
for ii=1:5
    tal=ii
end
```